Vision System Software Manual

v1.8.0

Table of Contents

1. Welcome

Welcome to use the User Manual for the Mech-Vision and Mech-Viz software.

In the Mech-Mind Vision System, the vision processing, path planning, and robot control functions are provided by Mech-Vision and Mech-Viz.

Release Notes

Mech-Vision Release Notes

New features, improvements, and resolved issues of the latest Mech-Vision version.

See more

Mech-Viz Release Notes

New features, improvements, and resolved issues of the latest Mech-Viz version.

See more

Before You Upgrade

Upgrade notes for Mech-Vision and Mech-Viz.

See more

Installation Guide

Software downloading, installation, upgrade, uninstallation, repair and modification.

See more

Project Migration Guide

Project migration required for the software upgrade.

See more

Software User Manual

Mech-Vision

Mech-Vision Machine Vision Software

See more

Mech-Viz

Mech-Viz Robot Programming Software

See more

2. Release Notes

This document introduces the new features, improvements, and resolved issues of version 1.8 of Mech-Vision, Mech-Viz and the communication component.

> Before upgrading Mech-Vision and Mech-Viz, please read this section carefully.

For the software upgrade notes and project migration guide, refer to 1.8.0 Upgrade Notes and 1.8.0 Project Migration Guide.

For the new features, improvements, and resolved issues introduced in Mech-Vision 1.8, refer to Mech-Vision 1.8 Release Notes.

For the new features, improvements, and resolved issues introduced in Mech-Viz 1.8, refer to Mech-Viz 1.8 Release Notes.

For the new features, improvements, and resolved issues introduced in Communication Component Release Notes.

2.1. Before You Upgrade

1.8.0 Upgrade Notes

This section introduces the upgrade notes for Mech-Vision 1.8.0, Mech-Viz 1.8.0, and the communication component.

When the origin of the robot reference frame was not at the bottom of the robot base, the tool pose on the robot teach pendant would be inconsistent with the values in the software. With version 1.8.0, this problem can be solved.

- After you create a project using some robot models with the 1.8.0 software, the DH1 no longer requires manual compensation when you perform manual calibration, teach the pick point, or send the planning or recognition result in tool poses to the robot through Adapter.
- When opening projects/solutions created with versions earlier than 1.8.0, the software will automatically switch to compatibility mode, and you can see a “Compatibility mode” watermark displayed on the interface. For certain robot models, the DH1 still should be compensated manually when you perform manual calibration, teach the pick point, or send the planning or recognition result in tool poses to the robot through Adapter. (If the DH1 has already been compensated in projects created with earlier versions, you do not need to compensate again.)

When you use Mech-Vision 1.8.0 in conjunction with Mech-Viz 1.8.0, please ensure the two software is in the same mode. The software cannot function properly if their modes are different. If this is the case, please refer to the following table to operate accordingly.

	Mech-Vision Regular mode	Mech-Vision Compatibility mode
Mech-Viz Regular mode	Use as normal.	According to the previous project logic and configuration, create a new project in Mech-Vision and re-calibrate the extrinsic parameters of the camera.
Mech-Viz Compatibility mode	According to the previous project logic and configuration, create a new project in Mech-Viz.	Use as normal.

- When opening projects/solutions created with versions earlier than 1.8.0, the software will automatically switch to compatibility mode. Even though you save the project in the current software, the software will still be in compatibility mode.
- The software will be in Regular mode when you create a new project with version 1.8.0.

Mech-Vision 1.8.0 Upgrade Notes

- After Mech-Vision is upgraded to 1.8.0, pay attention to the following notes when you use the Production Interface.
  - The production interface that you have deployed with a typical application in Mech-Vision 1.7.0 or earlier cannot be used anymore.
  - The production interface that you have customized for your application in Mech-Vision 1.8.0 or earlier can still be used. However, the production interface may fail to be opened due to the change in several files. To solve these issues, please contact Mech-Mind Technical Support.
- From version 1.8.0, Mech-Eye TOF cameras, MindVision 2D cameras, and SmartRay 3D laser profilers are no longer supported in the Capture Images from Camera Step.
- Mech-Vision 1.8.0 improved the “Sampling Interval” in the 3D Fine Matching and 3D Fine Matching (Multiple Models) Steps. If the “Sampling Interval” set in the project created with earlier version software is not appropriate, the project execution may be slowed down.
- Mech-Vision 1.8.0 improved the deep learning model in the Predict Pick Points (Any Objects), so that the model optimization time is reduced during Step execution. However, when you execute the Step for the first time after the software is installed or the IPC is changed, the time it takes to optimize the model may be longer.
- Mech-Vision 1.8.0 improved the picking configuration files used in the Predict Pick Points V2 Step. Projects downloaded in the solution library of Mech-Vision 1.7.4 or earlier versions and used this Step cannot be run in Mech-Vision 1.8.0 or further versions. Please re-download the corresponding project in the new version software.
- Mech-Vision 1.8.0 improved the Step Library. Some of the Steps were removed, and the Step categories of some Steps were changed. See the following for details.

Removed Step	Steps moved to “Legacy”
Calc Disparity Image, Calc 2D Poses of Rectangles, Calc Poses and Dimensions of Rectangles, 2D Matching, Count Boxes, Deep Learning Inference (Mech-DLK 2.1.0/2.0.0), Defect Detection, Detect Fiducial Markers, Detect Edges (Deep Learning), Image Classification, Generate Rectangle Candidates, Instance Segmentation, Load 2D Templates, Object Detection, Pallet Information Recognition, Validate and Calc Parameter Compensation, Evaluate 2D Poses, Generate Rectangular Edge Templates of Specified Sizes, Remove Overlapped Polygons, Remove Polygons outside Mask, Calc Calibration Board Poses	Remove Overlapped Objects, Remove Overlapped Objects V2, 3D Coarse Matching, 3D Fine Matching (Multiple Models), 3D Coarse Matching (Multiple Models), Adjust Poses, Sort 3D Poses

Mech-Viz 1.8.0 Upgrade Notes

- The singularity detection algorithm has been improved in version 1.8.0. Once the software is upgraded, the singularity detection result may be different from that in earlier versions. However, the singularity detection result in the current version is more accurate.
- The algorithm for limiting the range of motion of the robot’s shoulder, elbow, and wrist has been improved in version 1.8.0. Once the software is upgraded, the robot path may be different from that in earlier versions. However, the improved algorithm ensures the motion range limit takes effect, enhancing the accuracy and stability of the robot motion.
- The pick point calculation algorithm has been improved in version 1.8.0. Once the software is upgraded, the robot path may be different from that in earlier versions. However, more pick points can now be calculated and more appropriate paths can be planned.
- The collision model of the end tool in STL format is no longer supported in version 1.8.0. If the collision model of end tool you use is in STL format, you should convert the STL file to the OBJ file by using the Model Editor and then configure the end tool collision model again.

Upgrade Notes on Communication Component

Changes in Communication Component Usage

- Starting from version 1.8.0, Mech-Center will only run as a background communication component of the vision system when Mech-Vision or Mech-Viz is launched.
- The user interface designed for Mech-Center was removed. Standard Interface communication and Adapter communication should be configured in Mech-Vision, while Master-Control communication should be configured in Mech-Viz.

Upgrade Notes on Adapter Project

- The Step parameters of the “Set DO” and “Check DI” Steps were modified in Mech-Viz 1.8.0. Consequently, if these two Steps were used in the Adapter project created with an earlier software version, the relevant Step parameters in the code should be modified as well. You can check the key names of the Step parameters by using the **Key Query Tool**.
- Once the software is upgraded, if the Adapter project created with an earlier software version inherits classes from the Standard Interface program, it requires the addition of initialization

parameters to function properly. For details on changes of each protocol, go to `Mech-Center\Robot_Interface\Adapter_Sample_Projects\Adapter Inherits Standard Interface` in the software installation directory.

Upgrades Notes on Standard Interface Communication with Robots

- If the Standard Interface program is not reloaded onto the robot, and no message from the "Notify" Step is needed, i.e., Command 601 does not need to be triggered, the robot program does not need to be modified.
- If the Standard Interface program is not reloaded onto the robot, and the message from the "Notify" Step is needed, i.e., Command 601 needs to be triggered, the mm_get_notify function should be coded in the robot program manually. Please contact Mech-Mind Technical Support for further assistance.
- If the Standard Interface program is reloaded onto the robot in the project, please modify the robot program according to the following instructions.
    - When establishing Standard Interface communication with ABB robots, MM_Open_Socket and MM_Close_Socket should be called in the main program to establish and terminate the connection.
    - The Standard Interface programs corresponding to certain commands of FANUC, KUKA, ABB, YASKAWA, and KAWASAKI are modified as follows:
        - The "lastdata" parameter is removed from the five programs, i.e., mm_get_vizdata, mm_get_visdata, mm_get_vispath, mm_get_plandata, and mm_get_dydata. Instead, the programs internally loop until all vision data is received.
        - The "resource" parameter is added to the mm_get_dolist and mm_get_plandata programs to distinguish the sources of DO signal or Vision Move output, i.e., from Mech-Vision or Mech-Viz.
        - The robot program should call the mm_get_notify function to receive the message set in the "Notify" Step in the project. The notification data cannot be received directly as before.

Upgrades Notes on Standard Interface Communication with PLCs

- If the Standard Interface program is not re-downloaded in the project, and no message from the "Notify" Step is needed, i.e., Command 601 does not need to be triggered, the PLC program does not need to be modified.
- If the Standard Interface program is not re-downloaded in the project, and the message from the "Notify" Step is needed, i.e., Command 601 needs to be triggered, the MM_Get_Custom_Notify function should be adjusted in the PLC program manually. Please contact Mech-Mind Technical Support for further assistance.
- If the Standard Interface program is re-downloaded in the project, please modify the robot program according to the following instructions.
    - The PLC program should enable the MM_Get_Custom_Notify function to receive the message set in the "Notify" Step in the project. The notification data cannot be received directly as before.
    - When establishing Standard Interface communication with Siemens PLC Snap 7, please note that the input and output parameters have been modified.

Upgrade Notes of Previous Versions

1.7.2 Upgrade Notes

1.7.0 Upgrade Notes

2.2. Mech-Vision Release Notes

This document introduces the new features, improvements, and resolved issues of Mech-Vision 1.8.

Mech-Vision 1.8.0 Release Notes

This document introduces the new features, improvements, and resolved issues of Mech-Vision 1.8.0.

New Features

New "Production Interface" Feature

Mech-Vision 1.8.0 has introduced a new production interface. Engineers can develop a production interface through several configuration steps, and on-site operators can execute the production stably through a graphical user interface. The production interface mainly provides the following functions:

- Get quick knowledge of production status: The production interface provides visual information panel to help people get quick knowledge of production status and project execution status, which can help Operator identify and judge production abnormalities.
- Quickly view the production results: The production interface provides various running views, including Station live, Recognition result, Deep learning result, and Picking result. Operator can quickly view the production results.
- Add new workpieces and switch workpiece types: The production interface provides various ways to create workpiece models for different types and shapes of workpieces, and set production recipes to add new workpieces and switch workpiece types.
- Facilitate maintenance and troubleshooting: The production interface provides the role-based access control to effectively reduce the risk of data leakage. It also provides the information of troubleshooting, production logs, and alert records to help the operator quickly troubleshoot and resolve faults to achieve stable production.

> After Mech-Vision is upgraded to 1.8.0, pay attention to the following notes when you use the production interface feature.
>
> - The production interface that you have deployed with a typical application in Mech-Vision 1.7.0 or earlier cannot be used anymore.
> - The production interface that you have customized for your application in Mech-Vision 1.8.0 or earlier can still be used. However, the production interface may fail to be opened due to the change in several files. To solve these issues, please contact Mech-Mind Technical Support.

New Pose Adjustment Tool

Mech-Vision 1.8.0 has introduced the "Adjust Poses V2" Step, which provides a new "pose adjustment tool". This tool improves the usability of pose adjustment, enables the visualization of poses while adjusting, and reduces the learning cost of pose adjustment. You can achieve pose adjustment, pose sorting, pose filtering, and other pose manipulation operations just through three steps, namely pose adjustment, processing rules, and general settings.

Adapter Generator

Mech-Vision 1.8.0 introduced the Adapter generator. By configuring TCP/IP commands for communication between an external device and the vision system, you can efficiently generate an Adapter program. This program seamlessly controls Mech-Vision and Mech-Viz, facilitating smooth data acquisition. As a result, the development time of the Adapter program is reduced and the overall project delivery efficiency is enhanced.

The Adapter generator provides the following functions:

- Strong control function: Four commands can be generated, covering basic operations on Mech-Vision and Mech-Viz.
  - Run the Mech-Vision or Mech-Viz project.
  - Switch the parameter recipe of the Mech-Vision project.
  - Set the exit port for the “Branch by Msg” Step in the Mech-Viz project.
  - Set Step parameters in the Mech-Vision or Mech-Viz project.
  - Eye to hand (ETH) and eye in hand (EIH) mounting modes are supported.
- Flexible configuration: The command format can be customized, enabling you to decide the parameters to be returned.
  - Optional parameters for vision point data: pose type, number of poses sent, label, custom output, etc.
  - Optional parameters for robot path data: pose type, number of poses sent, motion type, label, etc.
- Quick program generation: The Adapter program can be generated quickly and conveniently, and is open to secondary development.
  - By clicking a graphical interface, you can easily generate the Adapter program according to your needs.
  - The generated Adapter program is open to custom development, greatly improving programming efficiency.

For details on the Adapter generator, please refer to robot-integration:adapter:generator-guide.pdf.

TCP/IP Command Reference for Standard Interface

Mech-Vision 1.8.0 has provided the TCP/IP command reference for Standard Interface on the interface, which provides a convenient way for on-site engineers to check the TCP/IP command format, returned data and status codes. The examples may be used to quickly demonstrate the sent commands and returned data via Standard Interface communication.

For details on TCP/IP commands, refer to robot-integration:standard-interface-development-manual:tcp-ip-socket.pdf.

New Error Analysis Tool

Mech-Vision 1.8.0 has introduced the error analysis tool. When an error occurs during recognition and locating, this tool can be used for troubleshooting. Based on the on-site situation you described, the tool will recommend specific items to check, including camera accuracy, robot absolute accuracy and repeatability, and the result of extrinsic parameters after calibration. This tool cannot be used to check the project logic or analyze the matching accuracy.

Text Detection, Text Recognition, and Unsupervised Segmentation Are Supported in Step "Deep Learning Model Package Inference"

In Mech-Vision 1.8.0, the Deep Learning Model Package Inference Step has added the following functions:

- Enabled the "text detection" function: can locate and label text in the input images.
- Enabled the "text recognition" function: can recognize and output the text in the text area that has been located.
- Enabled the "unsupervised segmentation" function: can segment defect areas in images by training models solely on defect-free images.
- Enabled the defect determination rules: can segment defects based on the configured defect determination rules.

Added the "Predict Object Poses (Sim2Pick)" Step

Mech-Vision 1.8.0 has added the "Predict Object Poses (Sim2Pick)" Step. When the point cloud of the single object type is input, this Step will directly recognize the object from the input point cloud and output its pose. This Step is generally used in the following scenarios:

- Workobjects are randomly stacked and there is a reasonable number.
- Workobjects are highly reflective and neatly arranged.
- Workobjects are neatly arranged and closely placed.

Other New Steps

Mech-Vision 1.8.0 has added the following Steps:

No.	Step	Description
1	3D Matching and Classification (Multiple Models)	This Step can match the objects in the scene by using multiple point cloud models and output the poses and class labels of the objects.
2	Remove Overlapped Objects (Lite)	This Step is used to remove overlapped objects, i.e., when multiple vision results exist at the same position of a workpiece, ineligible vision results can be removed by adjusting parameters.
3	Remove Coinciding Objects	This Step is used to remove objects that are completely covered or covered by other objects according to the pose confidence values.
4	Sort 3D Poses V2	This Step is used to sort the 3D poses according to specified rules. Compared with the "Sort 3D Poses" Step, this Step included two new sorting orders: "Sort by S shape on plane" and "Sort by Z shape on plane."

No.	Step	Description
5	Calc Calibration Board Pose and Check Camera Intrinsic Parameters	This Step can detect the calibration circle centers on the 2D image, and then compute the pose of the calibration board and the poses of calibration circles according to the camera intrinsic parameters and the input depth map. In addition, this Step can be used to estimate the accuracy of the camera's intrinsic parameters based on the calibration circle poses.

Improvements

Improved Software Interface and User Interaction

Mech-Vision 1.8.0 improved the software interface and user interfaces in the following ways.

- The Help menu on the menu bar provides access points for the online community. Through them, you can access the online community, visit the download center, view video tutorials, and feedback comments.
- Simplified the layout of the toolbar buttons for more convenient operation.
- Improved the context menu in the graphical programming workspace, making the functions and their arrangement more logical.
- Added a quick entry for Steps under "Advanced Components." You can open the tool for these Steps by clicking the **[Open the editor]** button.
- Added an option in the context menu of the graphical programming workspace for one-click closure of all Step visualizations.

Improved Solution Library

To ensure the quality and performance of the projects and solutions in the solution library, some projects and solutions have been temporarily removed from the solution library in Mech-Vision 1.8.0. These projects and solutions will be improved and re-uploaded in subsequent versions. See the following for details.

Category	Solution name
Workpiece loading	Brake Discs, Brake Discs (Single Station), Gearbox Housings, General Workpiece Picking (Brake Discs), Metal Ingots, Rebar Marking, Track Shoes
Piece picking	Pick Packages from Assembly Line

Meanwhile, the following solutions are added.

Category	Solution name
Workpiece loading	Battery Cells, Neatly Arranged Cylindrical Shafts (With Production Interface)
Palletizing & Depalletizing	Mixed-Case Cartons_V2, Single-Case Cartons_V2, Sacks_V2, Turnover Box (With Production Interface)

Improved the “Capture Images from Camera” Step

Mech-Vision 1.8.0 improved the “Capture Images from Camera” in the following ways.

- Added an entry to open Mech-Eye Viewer in the Step Parameters panel and the “Choose the camera and the calibration parameter group to use” interface, which allows you to open Mech-Eye Viewer quickly if needed.
- Added the “Obtain Flange Pose from Parameters” parameter. In the EIH setup, the robot flange pose at image capturing can be directly input.
- From Mech-Vision 1.8.0, Mech-Eye TOF cameras, MindVision 2D cameras, and SmartRay 3D laser profilers are no longer supported in the “Capture Images from Camera” Step.

Improved the “Path Planning” Step

Mech-Vision 1.8.0 improved the Path Planning Step. An Enable option was added under the Update Scene Object Settings. Once this option is switched on, Scene Object Names, Scene Object Dimensions, and Scene Object Poses input ports will be added to the Step to update the poses and dimensions of the scene objects.

The path planning tool has been improved as well. See the following for details.

- Added a Global Configuration step, which provides the options “Plan all vision results”, “Update bin pose with vision”, and “Box depalletizing.”
- The 3D simulation area now includes an orbit control, as well as buttons for panning the view, zooming views, screen adaptation, switching between perspective/orthographic views, and displaying key buttons.
- Modified the interface for adding and selecting the workflow.

Improved Matching Model and Pick Point Editor

Mech-Vision 1.8.0 improved the Matching Model and Pick Point Editor. See the following for details.

No.	Improvement	Description
1	Enabled to “Create geometric model”	Allowing to quickly generate the point cloud model using the created geometric model.
2	Improved tool interface	Improved the 3D visualization space and interactive operations. Demo GIFs are also provided to introduce the operations in the 3D visualization space.
3	Enhanced display settings	Allowing adjustment of the manipulator thickness in Edit > Display Settings.
4	Improved the operation for selecting multiple point cloud models	The shortcut keys were changed from “Ctrl+Left mouse button” to “Shift+Left mouse button”.

Improved Camera Calibration Tool

Mech-Vision 1.8.0 improved the camera calibration tool. See the following for details.

No.	Improvement	Description
1	Improved the calibration procedure	Streamlined redundant operations and deprecated functions within the calibration tool.
2	Add more important tips	Supplemented illustrations and informative prompts for key steps and parameters within the calibration procedure.
3	Enabled to open the Error Analysis Tool	After calculating the extrinsic parameters, you can verify the robot absolute accuracy, robot repeatability, and extrinsic parameter accuracy with the Error Analysis Tool.
4	Facilitated the analysis of hand-eye calibration result	Enabled to analyze the extrinsic parameters and robot accuracy according to the calibration data, troubleshoot rapidly, and provide relevant improvement suggestions. This feature is only available for calibration on 6-axis robots (excluding gantry robots) and Mech-Eye cameras, including automatic calibration and manual calibration using “Multiple random calibration board poses” to collect data.
5	Streamlined the calibration process for gantry robots and enhance the guidance	Provided the access link for the gantry robot calibration manual and important prompts.
6	Unable to view waypoints in Mech-Viz anymore	During camera calibration, the waypoints can only be viewed in the Scene viewer in the calibration interface instead of Mech-Viz.

Improved Step Library

Mech-Vision 1.8.0 improved the Step Library. Some of the Steps were removed, and the Step categories of some Steps were changed. See the following for details.

Removed Step	Steps moved to “Legacy”
Calc Disparity Image, Calc 2D Poses of Rectangles, Calc Poses and Dimensions of Rectangles, 2D Matching, Count Boxes, Deep Learning Inference (Mech-DLK 2.1.0/2.0.0), Defect Detection, Detect Fiducial Markers, Detect Edges (Deep Learning), Image Classification, Generate Rectangle Candidates, Instance Segmentation, Load 2D Templates, Object Detection, Pallet Information Recognition, Validate and Calc Parameter Compensation, Evaluate 2D Poses, Generate Rectangular Edge Templates of Specified Sizes, Remove Overlapped Polygons, Remove Polygons outside Mask, Calc Calibration Board Poses	Remove Overlapped Objects, Remove Overlapped Objects V2, 3D Coarse Matching, 3D Fine Matching (Multiple Models), 3D Coarse Matching (Multiple Models), Adjust Poses, Sort 3D Poses

The deep learning Steps in earlier versions, such as Instance Segmentation and Object Detection, are no longer supported in Mech-Vision 1.8.0. Please use the upgraded “Deep Learning Model Package Inference” Step instead.

Improvements in Other Steps

Mech-Vision 1.8.0 improved some Steps. See the following for details.

No.	Step	Description
1	3D Workpiece Recognition	A “Matched Point Cloud” port was added to output object point clouds in which overlapping objects are removed.
2	3D Laser Profiler	Re-adapted 3D laser profilers for better scanning of objects to capture images.
3	3D Matching Related Steps	Improved the “Sampling Interval” of the “3D Fine Matching” and “3D Fine Matching (Multiple Models)” Steps and improved the matching speed.
4	Predict Pick Points V2	Improved the picking configuration file. Projects downloaded in the solution library of Mech-Vision 1.7.4 or earlier versions cannot be run in Mech-Vision 1.8.0 or further versions. Please re-download the corresponding project in the new version.
5	Predict Pick Points (Any Object)	Improved the deep learning model file. Reduced the model optimization time during Step execution. However, when you execute the Step for the first time after the software is installed or the IPC is changed, the time it takes to optimize the model can be longer.
6	Create Object Model	Rectangle and cylinder curved surface object types are supported. You can also select the point cloud model type.
7	From Depth Map to Point Cloud	Improved the execution speed of the Step.
8	Point Filter	Improved the execution speed of the Step.
9	Calc Normals and Estimate Edges of Point Cloud	Improved the edge estimation effect and improved the Step execution speed.

Improved the 3D Visualization Window and Tips

Mech-Vision 1.8.0 improved the 3D visualization window and operation tips. See the following for details.

- Function improvements: improved the operations such as zooming views, rotating views, panning views, and rotating/moving objects.
- Tip improvements: prompted 3D animation to display the preceding operations when users enter the tools such as “Matching Model and Pick Point Editor” and “3D Workpiece Recognition Visualized Configurator”, helping users to learn about the operations in the 3D visualization window.

The operation methods for the Debug Output panel remain the same.

Resolved Issues

Mech-Vision 1.8.0 has fixed the following issues:

- When the robot model was changed in the path planning tool, the soft limits of the previous robot model were wrongly applied to the new robot model.

- When a version from Mech-Vision 1.7.0 to 1.7.4 was used, hand-eye calibration for 2D cameras was not supported.
- When two Mech-Vision projects shared the same 2D camera, the images captured using auto exposure were abnormal in a project if the other project had just captured images using fixed exposure.
- When the camera failed to be connected in the “Capture Images from Camera” Step, the error code was missing.

Release Notes of Previous Versions

Mech-Vision 1.7.4 Release Notes

Mech-Vision 1.7.2 Release Notes

Mech-Vision 1.7.1 Release Notes

Mech-Vision 1.7.0 Release Notes

Mech-Vision 1.6.2 Release Notes

Mech-Vision 1.6.1 Release Notes

Mech-Vision 1.6.0 Release Notes

2.3. Mech-Viz Release Notes

This document introduces the new features, improvements, and resolved issues of Mech-Viz 1.8.

Mech-Viz 1.8.0 Release Notes

This document introduces the new features, improvements, and resolved issues of Mech-Viz 1.8.0.

New Features

File

In version 1.8.0, the functional interfaces have been reorganized and the File page has been upgraded. The File page includes example projects, file management, option settings, version information, and quick access links to the Online Community and User Manual. The example projects can help you better understand various functions and application scenarios of Mech-Viz.

Master-Control Robot

Added a configuration window for Master-Control Robot, facilitating the configuration and establishment of Master-Control communication in Mech-Viz.

The “Adaptive Relative Move” Step

Added a new Adaptive Relative Move Step. In depalletizing scenarios, this Step automatically adjusts the lifting height of the held box based on its height in the vision result.

More Robot Configurations Supported

The configurations of spray robots, gantry robots, ABB GoFa six-axis collaborative robots, SCARA robots with axis 1 as the translation axis are supported in the new version. In version 1.8.0, Mech-Viz has supported for all industrial robot configurations except for parallel robots.

> Configuration support means that Mech-Viz can conduct simulation and path planning for the robot.

Key Query Tool

Added Key Query Tool, facilitating the query of keys that can be read and written by external services. The function description and usage example of each key are also provided. This tool is available only when the Developer mode is enabled.

Improvements

Robot Z-Offset

When the origin of the robot reference frame was not at the bottom of the robot base, the tool pose on the robot teach pendant would be inconsistent with the values in the software. With the upgraded Mech-Viz and Mech-Vision of version 1.8.0, this problem can be solved. With the improved feature, when you perform manual calibration, teach the pick point, or send the planning or recognition result in tool poses to the robot through Adapter, the DH1 no longer requires manual compensation.

Step

Vision Move	For the “Reuse Vision Result” and “Share Vision Results” features, the unused vision result will be automatically overridden by newly received vision result, without the need for manual reset with the “Reset” Step.
	The “Only Use Poses with Following Label” parameter was renamed to “Use Specific Workobject Pose Only” and moved from the custom parameter panel at the bottom to the upper parameter list.
	Added the “Sorting Strategy” parameter in the “Multiple at once” mode for box depalletizing.
	When the upper surface of a single box is larger than the entire vacuum gripper, and all sections of the vacuum gripper are activated, the “box surface coverage” can reach 100%.
Set DO	The “Set DO List” Step has been merged with the “Set DO” Step to form a new “Set DO” Step.
Check DI	The “Check DI List” Step, “Check DI” Step, and “Wait for DI” Step have been merged to form a new “Check DI” Step, and more check modes were added.
Change Tool	The “Change Tool” Step will no longer interrupt the planning. When you need to switch the end tool based on the pick point labels for picking, you can classify before changing the tool instead of vice versa.
Predefined Pallet Pattern	Improved the “Automatic Layout” algorithm.
	Added the “Limit Box Count” parameter.

Robot Model Library

Robot Model Library enables you to download the robot model online and load the **online robot model list** with one click. If you find any robot model inaccurate, you can provide feedback in the software. The robot models in the online library have been expanded to over 1000, covering basically all common robots on the market.

Plan History

- The plan history was organized into three levels: level one, the planning entry; level two, the planning attempt; level three, the picking path.
- The causes of the planning failure was simplified from more than 20 to 3, i.e., waypoint unreachable, singularity, and collision.
- Reproducing the planned path is supported.

Save and Use Vision Records

Improved the workflow to save and use vision records.

- A default number of vision records will be automatically saved when the project runs. The number of saved records can be adjusted as well.
- The entry for setting vision records has been moved to the drop-down list on the side of the “Simulate” button.
- Improved the “Select Vision Record to Use” window.

Improved “Result Must Be in Specified Bin” Feature

In version 1.8.0, the “Result Must Be in Specified Bin” parameter was added in the “Vision Move” Step. You can choose whether or not to limit the effective range for vision results for each “Vision Move” Step and specify the bin in which the vision result must be located. In earlier versions, no adjustable parameters of this feature were available in the “Vision Move” Step, and the activation of the feature was completely controlled by the bin configuration. Once the feature was enabled for a bin, it would take effect for all “Vision Move” Steps in the workflow.

Model Editor

In the Model Editor of version 1.8.0:

- New **Elliptic Cylinder Selection** and **Cylinder Selection** were added to better process complex end tool models and avoid selection interference.
- Improved the workflow to set the reference frame.

3D Simulation Area

The 3D simulation area now includes a navigation gizmo, as well as buttons for panning the view, zooming views, screen adaptation, switching between perspective/orthographic views, and displaying key buttons.

Project Resource Tree

- **Vision services** and **Vision records** entries were added to the project resource tree. New buttons for collapsing and expanding items were added as well.

- A “Reload Robot” option was added to the context menu of the robot entry in the project resource tree, and modifying the robot parameters no longer requires restarting the software.

Collisions

The “Compute collision contacts” was removed from the Collisions functional panel, and “Point cloud collision calculation mode” was added in the Configuration on point cloud pane in Collision detection configuration.

Robot

- Revised the robot communication capability list.
- A common parameter to define the robot reference frame was added in the robot configuration parameters, supporting the customization of the robot reference frame position relative to its base.
- Four-axis robots can be used in the planning and picking horizontal and vertical vision results.
- The five-axis robot now includes a feature similar to the four-axis robot, automatically correcting the vision poses that deviate from the world reference frame Z-axis within ±5°.
- Allows customization of the robot flange orientation.

Others

- Improved the robot motion algorithm, reducing excessive rotation of the robot end and increasing the motion efficiency.
- Removed the Logging Level option in Settings in the menu bar has been removed. All levels of logs will be printed.
- Added visualization model similar to the buffer to the array gripper.
- Added a “Do not detect” option to the singularity detection method.

Resolved Issues

Mech-Viz 1.8.0 has fixed the following issues:

- The reduction ratio would not take effect when a singularity was caused by excessive acceleration.
- When the workobject was a custom model, the collision detection between the workobject and the scene object would not take effect.
- When you clicked one plan history entry to check the recorded tool pose, the robot pose displayed might not be the same as at the time of planning.
- Clicking the plan history entry would switch the active tool in the project resource tree.
- Scene object files with spaces at the beginning or end of their names could not be deleted or renamed.
- The software automatically added zeros to the decimal part in the input box, making it impossible to input decimals.
- Steps may overlap when using automatic format.
- After the boxes were grouped, the “Relative Move” Step that depends on the “Vision Move” Step would generate a superimposed path, resulting in abnormal visualization.
- If the collision detection was not enabled for the bin model in the scene, an error would occur

when the “Vision Move” Step updated the bin pose according to the vision result.

- The algorithm for grouping boxes along the Y-axis of the box reference frame in multi-pick depalletizing might result in an error when using the “Vision Move” Step.
- The software might crash when the “All Vision Poses in One Move” option in the “Vision Move” Step was selected.
- When a project was running, setting multiple parameters for multiple Steps at once through the Adapter took a long time.

Release Notes of Previous Versions

Mech-Viz 1.7.4 Release Notes

Mech-Viz 1.7.2 Release Notes

Mech-Viz 1.7.1 Release Notes

Mech-Viz 1.7.0 Release Notes

Mech-Viz 1.6.2 Release Notes

Mech-Viz 1.6.1 Release Notes

Mech-Viz 1.6.0 Release Notes

2.4. Communication Component Release Notes

This document introduces the new features, improvements, and resolved issues of version 1.8 of the communication component.

Communication Component 1.8.0 Release Notes

This document introduces the new features, improvements, and resolved issues of version 1.8.0 of the communication component.

> ! Starting from version 1.8.0, Mech-Center will only run as a background communication component of the vision system when Mech-Vision or Mech-Viz is launched. Specifically, Standard Interface communication and Adapter communication should be configured in Mech-Vision, while Master-Control communication should be configured in Mech-Viz.

New Features

New Features of Standard Interface Communication

The following features are now supported by Standard Interface communication.

- Multiple ports and multiple DBs (data blocks) are now supported.

  Application scenario: Connect multiple cameras to an IPC that communicates with an external device, enabling independent control of cameras and workstations through a single external device.

- Communication over TCP/IP protocols supports concurrent processing of project data with a maximum of four ports. If multiple ports on the IPC are connected with the robot side, you should modify the example program to ensure that the global variables are not shared by different ports.
- Communication over Siemens PLC Client supports concurrent processing of project data with a maximum of four DBs.

- New Standard Interface commands:
  - Command 100: Trigger Mech-Vision Project and Get Results (trigger the project + switch the parameter recipe + get vision results or the planned path)
  - Command 200: Trigger Mech-Viz Project and Get Planned Path (trigger the project + set the exit port of the “Branch by Msg” Step + get the planned path)
  - Command 106: Get Gripper DO List from Mech-Vision
  - Command 111: Get Vision Move Data from Mech-Vision
- Standard Interface communication now supports four-axis SCARA robots.

Compatibility with Robots

In version 1.8.0 of the communication component, robots listed in the following table are compatible with Master-Control communication and/or Standard Interface communication.

Robot	Master-Control communication	Standard Interface communication
Elite CS robots	√	
DENSO robots		√
INOVANCE industrial robots	√	√
SIASUN cobots	√	√
TM robots (with TMFlow2 system)		√
DOBOT CR robots		√

Improvements

Updates to Standard Interface Communication

The following are the updates to the Standard Interface communication in version 1.8.0 of the communication component.

- The example program of ABB Standard Interface communication now uses a persistent connection. You may use MM_Open_Socket or MM_Close_Socket in the main program to establish or terminate the connection.
- The Standard Interface example programs of FANUC, KUKA, ABB, YASKAWA, and KAWASAKI are modified as follows:
  - The “lastdata” parameter is removed from the five programs, i.e., mm_get_vizdata, mm_get_visdata, mm_get_vispath, mm_get_plandata, and mm_get_dydata. Instead, the programs internally loop until all vision data is received.
  - The “resource” parameter is added to the programs mm_get_dolist and mm_get_plandata to distinguish the sources of the DO signals or Vision Move output, i.e., from Mech-Vision or Mech-Viz.

- The mm_get_notify function is added to receive the message set in the "Notify" Step in the Mech-Vision or Mech-Viz project.
- The mm_stop_viz function is added to stop the running of a Mech-Viz project.
- The mm_com_test program is added to test communication.

Updates to Master-Control communication

The following are the updates to the Master-Control communication in version 1.8.0 of the communication component.

- Supports setting the blend radius for Kawasaki robots.
- Supports setting the DO signal list for AUBO robots.
- Supports setting the DO signal list for JAKA robots.

Resolved Issues

The following issues are resolved in version 1.8.0 of the communication component.

- Standard Interface communication
  - Resolved the issue that the communication component could not start when the IPC shut down abnormally and then restarted.
  - Resolved the issue that the height offset was not added to the pose data obtained with Command 210.
- Master-Control communication
  - Resolved the issue that the velocity of linear motion could not be set for DOBOT robots in Mech-Viz.
  - Resolved the issue that the gripper could not be set for EPSON robots in Mech-Viz.

2.5. Project Migration Guide

1.8.0 Project Migration Guide

Migration from 1.7.X Project to 1.8.0 Project

Starting from version 1.8.0, Mech-Center will only run as a background communication component of the vision system when Mech-Vision or Mech-Viz is launched. Specifically, Standard Interface communication and Adapter communication should be configured in Mech-Vision, while Master-Control communication should be configured in Mech-Viz.

Project Type (Divided According to the Communication Modes in Actual Use)	Actions after Software Upgrade
Standard Interface communication, Adapter communication	Keep the same usage as 1.7.X version.

Master-Control communication	Reset configuration of Master-Control communication in Mech-Viz by following these steps: 1. Start Mech-Viz, and select the robot model that corresponds to the real robot model. 2. Click **[Master-Control Robot]** in the toolbar. 3. After setting the IP address of robots, click **[Connect the robot]**.

For more notes after the upgrade, please refer to upgrade-notes-180.pdf.

Migration from 1.6.X Project to 1.8.0 Project

It is recommended to upgrade 1.6.X project to 1.7.X project first. For more detailed description, please refer to 1.7.0 Project Migration Guide.

Then you can upgrade 1.7.X project to 1.8.0 project. For more detailed description, please refer to Migration from 1.7.X Project to 1.8.0 Project.

Project Migration Guide of Previous Versions

1.7.0 Project Migration Guide

3. Installation Guide

This chapter guides you through the downloading, installation, upgrade, uninstallation, repair, and installation modification of Mech-Vision and Mech-Viz.

Mech-Mind software has the following features: (1) Provide easy-to-use setup wizard. (2) Support standard installation, upgrade and uninstallation procedures. (3) Support English, Simplified Chinese, Japanese and Korean. (4) Support automatic adaptation to various resolutions.

If you are using Mech-Vision and Mech-Viz for the first time, refer to Install Software to complete the software installation.

If you have already installed Mech-Vision and Mech-Viz, refer to Upgrade Software to upgrade the software to the latest version.

System Requirements

Mech-Vision and Mech-Viz can be used only when being installed on the IPC (industrial personal computer) or your PC (personal computer). The IPC or PC for installing software should meet the following recommended system requirements:

Authorized dongle	License dongle with Mech-Vision and Mech-Viz authorization
Operating system	Windows 10 or above
CPU	Needs to support the AVX2 instruction set and meets any of the following conditions: • IPC or PC without any discrete graphics card: Intel i5-12400 or higher. • IPC or PC with a discrete graphics card: Intel i7-6700 or higher, with the graphics card not lower than GTX 1050 Ti. IPCs with Intel CPUs are fully tested while IPCs with AMD CPUs are not yet tested. Therefore, Intel CPUs are recommended.
Memory	8GB or above
GPU	NVIDIA GTX 1050 Ti or above (if with a discrete graphics card)
Hard disk	128 GB SSD or larger

- Mech-Vision and Mech-Viz do not support discrete graphics cards other than NAVDIA.
- Ensure that the hard disk where software will be installed on IPC or PC should have at least 10 GB free space; otherwise, installation may fail.
- If the CodeMeter driver has not been installed on the IPC or PC, it will be automatically installed to the hard disk C during software installation. Do not uninstall it or move its installation files; otherwise, license authorization may fail and you are not allowed to use Mech-Vision and Mech-Viz.

When using Mech-Vision and Mech-Viz, the resolution and scaling of the screen should follow the correspondence in the table below. If the resolution and scaling of your screen do not match those in the table below, some display problems may occur.

If you are using dual monitors, make sure that both have the same resolution and scaling.

Resolution	Scaling
1280×800 (16:10)	100%
1920×1080 (16:9)	100%, 125%
2560×1440 (16:9)	125%, 150%
3840×2160 (16:9)	150%, 175%

Download Software Installation Package

Mech-Mind provides a software installation package for Mech-Vision and Mech-Viz (Mech-Vision & Mech-Viz installer).

- The Mech-Vision & Mech-Viz installer integrates the CodeMeter driver to adapt to the Dongle for software license authorization.
- The Mech-Vision & Mech-Viz installer integrates the Python 3.6.5 environment and no longer includes the running environment check tool.
- The Mech-Vision & Mech-Viz installer integrates the basic communication component Mech-Center, which includes a built-in Python 3.6.8 environment.

You can download the software installation package from Mech-Mind Download Center or contact Mech-Mind pre-sales or support engineers.

Install Software

Verify Integrity of Software Installation Package

Since the software installation package may be corrupted during transmission or downloading, you need to verify the integrity of the software installation package. The integrity of the installation package can be verified with the CRC-32 value, which is provided on the download page.

Please install and use the 7-Zip software to calculate the CRC-32 value.

To verify the integrity of the software installation package, follow these steps:

1. Copy the installation package to a specified directory on the IPC or PC, such as *D:/*.
2. Decompress the installation package. After decompression, you will obtain the installer (Mech-Vision & Mech-Viz Installer latest) and the content folder.

   After decompression, do not move the installer and the content folder.

3. Open 7-Zip, and use the address bar to navigate to the directory where the software installation package is located.
4. Select the software installation package, and in the menu bar, select File › CRC › CRC-32 to calculate the CRC-32 value.
5. Check the calculated CRC32 value against the one on the download page. The two values should be identical.
6. Repeat steps 3–5 on the installer and content file after decompression.

   If the calculated CRC32 values of the software installation file and the content folder do not match those provided on the download page, please download it again.

Install Software

To install Mech-Vision and Mech-Viz, follow these steps:

1. Double-click the software installation file(EXE file) and open **Mech-Vision & Mech-Viz Setup Wizard**. The **Welcome** window is displayed.
2. In the **Welcome** window, browse the software information and click **[Next]**.
3. In the **License Agreement** window, read the license agreement carefully, check the **[I accept the terms and conditions in the license agreement]** check box, and then click **[Next]**.
4. In the **Select Product** window, select the products(Mech-Vision x.x.x and Mech-Viz x.x.x) to install. Select the **[Create desktop shortcut(s)]** checkbox (recommended), and then click the **[Next]** button.

   - Mech-Center is forced to be installed along with the products as a fundamental communication component, but will not have a desktop shortcut.
   - Mech-Center cannot be opened individually and can only be evoked by Mech-Vision or Mech-Viz to provide communication in the background.

5. In the **Set Path** window, change the installation path as required, such as "D:/project", and then click the **[Next]** button.

   If the installation path is not changed, the default software install paths are as follows during the initial installation:

   - Mech-Vision:`C:/Mech-Mind/Mech-Vision & Mech-Viz-x.x.x/Mech-Vision`
   - Mech-Viz:`C:/Mech-Mind/Mech-Vision & Mech-Viz-x.x.x/Mech-Viz`

6. In the **Confirm** window, confirm that the installation path is correct and then click the **[Install]** button.
7. In the **Install** window, wait for the installation to complete.
8. After the software is installed, in the **Finish** window, click the **[Finish]** button.

If the installation fails, refer to the error message in the **Finish** window and the Troubleshoot Common Installation Issues section to troubleshoot the issue.

When you first open the software, if your IPC or PC displays a security alert, click **[Allow Access]** to add the software to the firewall whitelist. Otherwise, the software may not be able to detect the camera due to the firewall blocking.

Upgrade Software

The software does not support direct upgrade from a version earlier than 1.6.0 to the current version. If you want to upgrade the software running an earlier version than 1.6.0 to the current version, uninstall the earlier software version, and then install the new software version. Before upgrade, read carefully the Before You Upgrade section.

To upgrade the software, follow these steps:

1. Copy the software installation package to a specified directory on the IPC or PC, such as "D:/project", and decompress it.
2. Double-click the software installation file(.exe file) and open **Mech-Vision & Mech-Viz Setup Wizard**. The **Upgrade** window is displayed.

3. In the **Upgrade** window, click **[Upgrade and delete history versions]**.

> To preserve the historical version, rename the installation folder of the historical version before performing this step.

4. Wait until the upgrade operation is completed.

Uninstall Software

If you do not need to use Mech-Vision or Mech-Viz, you can uninstall it using any of the following ways:

- Uninstall software using the Setup Wizard (recommended)
- Uninstall software using Windows Control Panel

Uninstall Software Using the Setup Wizard

To uninstall the software using the Setup Wizard, follow these steps:

1. Double-click the decompressed software installation file(.exe file) and open **Mech-Vision & Mech-Viz Setup Wizard**. The **Maintain** window is displayed.
2. In the **Maintain** window, click the **[Uninstall]** button.
3. In the **Uninstall** window, click either **[Keep user configuration files]** or **[Do not keep user configuration files]**.
4. Wait until the uninstall operation is completed.

Uninstall Software Using Windows Control Panel

To uninstall the software using the Windows Control Panel, follow these steps:

1. Open the **Control Panel** on the IPC or PC.
2. Select Program › Uninstall or change a program.
3. Right-click the software to uninstall, and then click **[Uninstall]**.

> If you use the installation package Mech-Vision & Mech-Viz to install software, all installed software products will be displayed as “Mech-Vision & Mech-Viz x.x.x” as a whole in the program list.

4. Wait until the uninstall operation is completed.

Repair Software

If Mech-Mind software fails to work normally due to certain exception, you can use the repair function to reinstall the software.

To repair the software, follow these steps:

1. Double-click the decompressed software installation file(.exe file) and open **Mech-Vision & Mech-Viz Setup Wizard**. The **Maintain** window is displayed.
2. In the **Maintain** window, click **[Repair]**.

3. Wait until the repair operation is completed.

Modify Software Installation

After the installation package Mech-Vision & Mech-Viz is installed, to modify software installation, follow these steps:

1. Double-click the decompressed software installation file and open **Mech-Vision & Mech-Viz Setup Wizard**. The **Maintain** window is displayed.
2. In the **Maintain** window, click **[Modify]**.
3. In the **Select Product** window, select the products or components to install, or clear the products or components to uninstall.
4. Complete remaining installation operations.

Software License Agreement

For the license agreement of Mech-Vision and Mech-Viz, refer to End-User License Agreement.

Troubleshoot Common Installation Issues

Failed to Start the Setup Wizard

Symptom:

After double-clicking on the installation file, the setup wizard could not be started properly or crashed.

Possible Cause:

There is not enough space on the system drive.

Troubleshooting Procedure:

Check whether the available space on the system drive is larger than the size of the installer.

- If the available space on the system drive is smaller than the size of the installer, delete unnecessary files to free up the space, and then reinstall. If the issue still exists, contact Mech-Mind Technical Support.
- If there is enough space on the system drive, contact Mech-Mind Technical Support.

Installation Failed

Symptom:

An error message saying “Installation failed.” appeared during the installation.

Possible Causes:

- Corrupted installer or missing files.
- The current user does not have administrator access.

- Another Wizard is performing the installation or the Windows system is being upgraded.
- Other reasons.

Troubleshooting Procedure:

1. Obtain a new installer and check its integrity, and then try the installation again.
   - If the issue is resolved, the troubleshooting is completed;
   - Otherwise, proceed to step 2.
2. Right-click the installer and select “Run as administrator” to reinstall.
   - If the issue is resolved, the troubleshooting is completed;
   - Otherwise, proceed to step 3.
3. Try the installation again after the installation of other software completes or after the Windows system upgrade completes.
   - If the issue is resolved, the troubleshooting is completed;
   - Otherwise, proceed to step 4.
4. Restart your IPC or computer and try the installation again.
   - If the issue is resolved, the troubleshooting is completed.
   - If the issue still exists, click the “installation log” link in the **Finish** window to collect installation logs and send them to Mech-Mind Technical Support.

4. Mech-Vision

Welcome to use Mech-Vision User Manual.

Mech-Vision is a cutting-edge **machine vision software** independently developed by Mech-Mind. Through its fully graphical interface, advanced machine vision applications such as piece picking, high precision positioning, assembly, industrial inspection/measurement, and automatic path planning can be completed without writing code.

This manual consists of the following chapters:

No.	Chapter	Content
1	Introduction to Mech-Vision	Introduction to software functions, terms and concepts related to machine vision, user interface, and the general workflow of creating solutions in Mech-Vision
2	Quick Guide of Common Operations	Common operations with Mech-Vision
3	Step Reference Guide	Reference guide for using Steps and Procedures
4	Guide to Production Interface	Introduction to configuration and usage of the production interface
5	Hand-Eye Calibration Guide	Introduction to concepts and know-hows of hand-eye calibration
6	Measurement Mode Guide	Introduction to the basics of the measurement mode and key algorithms
7	Supplementary Tools	User guide for common tools in Mech-Vision
8	User Guide of Deep Learning	Introduction to deep learning in the 3D-vision guided robot picking scenarios

4.1. Introduction to Mech-Vision

4.1.1. Terms and Concepts

This section introduces the key terms and concepts related to machine vision.

Solution and Project

Solution library

A resource library that provides typical solutions or projects (with example data) coming from different application scenarios.

Solution

A solution is a collection of functional configurations and data for robot, communication, vision processing, and path planning required for a vision application.

Project

A project refers to a Mech-Vision project. One or more projects constitute a project. Projects cannot be used independently and must belong to a solution.

Step

Steps are the basics of a project. A Step is a minimum algorithm unit for data processing. By connecting different Steps in a project, you can achieve different data process tasks.

Procedure

A Procedure groups the Steps whose functions are closely related and achieve a specific purpose concertedly in the project.

Parameter recipe

Parameter recipes are sets of parameter settings that need to be adjusted according to different situations for the same project. With parameter recipes, you do not need to build multiple projects with the same logic and configure their parameters differently to meet different on-site requirements. Instead, you will only need to switch between parameter recipes in one project to make it applicable to various scenarios and therefore the productivity can be increased.

Hand-Eye Calibration

Hand-eye calibration

Hand-eye calibration establishes the transformation relationship between the camera and robot reference frames. With this relationship, the object poses determined by the vision system can be transformed into those in the robot reference frame, which guides the robot to perform its tasks.

Intrinsic parameters

Intrinsic parameters are measures of the properties of a camera itself, including the focal length, the lens distortion, etc. These parameters are usually calibrated and stored in the camera before the camera leaves the factory.

Extrinsic parameters

Extrinsic parameters define how poses are transformed between the camera reference frame and the world reference frame.

Euler angles

Euler angles are used to describe the orientation of an object in the 3D space. The object's rotation in the 3D space can be denoted by 3 angles, i.e., pitch, yaw, and roll.

TCP (Tool Center Point)

Tool center point is the pinpoint at the end of the robot tool. In order to complete tasks such as picking, we usually say that the robot should move to a specific point in space, which actually means its TCP should move to that point.

Vision Processing

Point cloud

A point cloud is a set of data points in space that represents a 3D shape or object.

Pose

The position and orientation information that are used to guide the robot.

Mask

Specified images, shapes, or objects are usually selected to mask the image (partial or all) to control the area to process or the processing. The particular image or object used for masking is called a mask.

ROI (Region of Interest)

The region of interest excludes surrounding parts unnecessary for vision data processing in the scene. Such parts may be background objects, pallets, bin edges, etc.

Deep Learning

Model Package

After a model has been trained in Mech-DLK, it can be exported as a model package, which contains one or more models. In Mech-Vision, the model package can be used for the inference of the image data.

Super Model Package

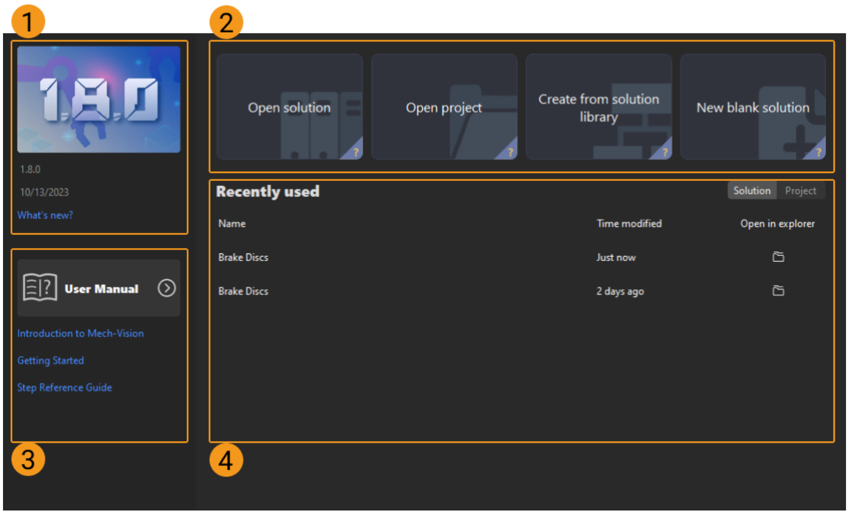
A universal model provided by Mech-Mind used to recognize cartons or sacks. If the recognition performance is unsatisfactory, Mech-DLK can be used to fine-tune the model.

4.1.2. User Interface

This part introduces Mech-Vision’s Welcome interface and Home interface.

Welcome Interface

If there is no autoloaded solution or project, Mech-Vision will display the Welcome interface when you open it, as shown below.



Name	Time modified	Open in explorer
Brake Discs	Just now	🗀
Brake Discs	2 days ago	🗀

Hover the mouse cursor over the “?” in the lower right corner of a quick entry card to check the introduction to this card.

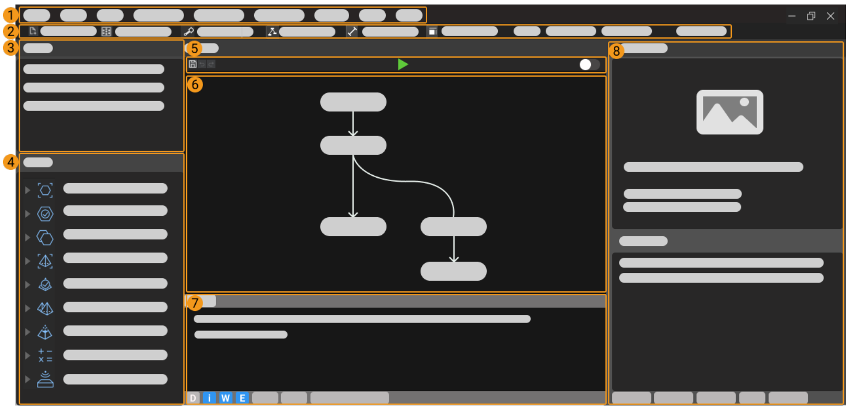
The Welcome interface of Mech-Vision consists of the following parts.

No.	Area	Description
1	Version information	Check version information and release notes.
2	Quick entry	Quick access for the user to open a solution, open a project, create a solution from solution library, and create a new blank solution.

No.	Area	Description
3	User Manual	Click the button to view Mech-Vision’s user manual.
4	Recently used	Quickly open recently used solutions or projects.

Home Interface

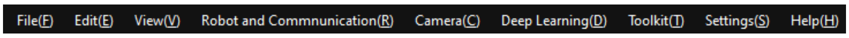
Leave the Welcome interface and you will enter the Mech-Vision Home interface, as shown below.



The Home interface of Mech-Vision consists of the following parts.

No.	Area	Description
1	Menu Bar	Include File, Edit, View, Robot and Communication, Camera, Deep Learning, Toolkit, Settings, and Help.
2	Toolbar	Include buttons to quickly access tools and functions.
3	Project List	Display opened solution, project name, and their status, including communication status and autoload status.
4	Step Library	Include all the Steps needed to create a project.
5	Project Toolbar	For project saving, running, and debug output, etc.
6	Graphical Programming Workspace	Edit the processing or logical flows of a vision project.
7	Log	Display the log in real-time while running a project; for user to find the execution record of a certain moment in this panel.
8	Project Configuration Pane	Include Debug Output, Step Parameters, Project Assistant, Step Quick Info, History, and Step Comment List.

4.1.2.1. Menu Bar



The menu bar provides basic functions related to projects, as well as supplementary tools like Camera and Deep Learning.

File

Used to manage solutions and projects.

Option	Description	Shortcut
New Solution	Create a new empty solution	Ctrl+Shift+N
New Project	Create a new empty project	Ctrl + N
Create from Solution Library	Choose and create a solution from the solution library	N/A
Open Solution	Open an existing solution	N/A
Open Project	Open an existing project	Ctrl+O
Open Recent…	Show recently opened projects. Click the project name to open it	N/A
Open Executable File in Explorer	Open the installation folder of the software	N/A
Save Solution	Save changes to the current solution	Ctrl+Shift+S
Save Project	Save changes to the current project	Ctrl+S
Save Project To JSON	Save the project, and save the VIS file as JSON	N/A
Save Project As	Save the project under a specified directory	N/A
Backup Solution	Back up the solution content within specified time range to a specified path	N/A
Close Solution	Close selected solution	N/A
Exit	Close the project and exit Mech-Vision	Ctrl+Q

Edit

Option	Description	Shortcut
Undo	Undo the previous action	Ctrl+Z
Redo	Redo the undone action	Ctrl+Y

View

Providing settings for managing components of the user interface. Check the box before each option to display the corresponding component in the software.

Option	Description
Project List	Display/Hide the Project List panel. Selected by default.
Step Library	Display/Hide the Step Library panel. Selected by default.
Debug Output	Display the Debug Output panel. Selected by default, and cannot be unselected.

Option	Description
History	Display/Hide the History panel. Selected by default.
Project Assistant	Display/Hide the Project Assistant tab. Selected by default.
Step Quick Info	Display/Hide the Step Quick Info tab. Selected by default.
Step Comment List	Display/Hide the Step Comment List tab. Selected by default.
Step Parameters	Display/Hide the Step Parameters tab. Selected by default.
Step Input Source Selection	Display/Hide the Step Input Source Selection panel. Unselected by default.
Measurement Output	Display/Hide the Measurement Output panel. Unselected by default.
Result View	Display/Hide the Result View (In Measurement Mode). Unselected by default.
Log	Display/Hide the Log panel. Selected by default.
Toolbar	Display/Hide the toolbar. Selected by default.
Default Layout	Turn Mech-Vision User Interface into default layout.
Format Grid	Display/Hide format grid and assist line. Unselected by default.
Align Type	Choose the type of assist line (Align Anchor, Align Horizontal, Align Vertical).

Robot and Communication

For configurations related to robots and communication.

Option	Description
Robot Communication Configuration	Configure the robot and the communication interfaces.
Adapter Generator	Can be used to generate Adapter programs quickly.
TCP/IP Command Reference for Standard Interface	Can be used to check the Standard Interface TCP/IP commands.
Robot Library Tools	Import/export robot models

Camera

For image capturing, camera calibration, and parameter analysis.

Option	Description	Shortcut
Camera Calibration	Standard: normal calibration procedure	N/A

Deep Learning

Option	Description	Shortcut
Deep Learning Model Package Management Tool	Open deep learning model management tool	N/A
Deep Learning Server	Start/Stop deep learning server(s)	Ctrl+Alt+D

Toolkit

Providing supplementary tools that are commonly used for debugging.

Option	Description
Matching Model and Pick Point Editor	For generating point cloud models for the object, editing point cloud models, and adding pick points
Error Analysis Tool	Check camera intrinsic parameter accuracy, robot absolute accuracy and other items, and troubleshoot the accuracy errors which occur during recognition and positioning

Options

Providing commonly used settings.

Option	Description	Shortcut
Set Mech-Center Address	Set the IP address of Mech-Center	N/A
Lock Project	Enter passwords to lock opened projects	N/A
Option	Change common settings like language, unit, etc.	Ctrl + Shift + O

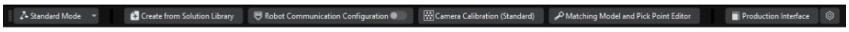
Help

Used to check the current version information, change log, manual, etc.

Option	Description
About	Check current version information
What’s New	Open the Release Notes in browser
User Manual	Open the User Manual in browser
Online Community	Access to Forum, Download Center, Video Tutorials, and Feedback
About Plugins	Select the plugins you’d like to display, and check the name, version, description, and location of the plugins

4.1.2.2. Toolbar

You can create a solution from Solution library, apply in-built tools, and change project mode.

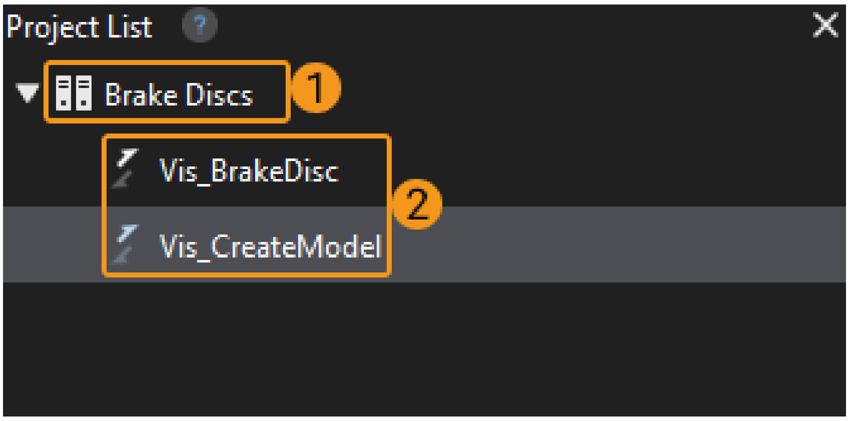

Standard Mode
Create from Solution Library
Robot Communication Configuration
Camera Calibration (Standard)
Matching Model and Pick Point Editor
Production Interface

Option	Description
Standard Mode	For switching modes between Standard Mode and Measurement Mode
Create from Solution Library	For creating a solution quickly
Robot Communication Configuration	For selecting robot models and setting interface communication
Camera Calibration (Standard)	For starting the standard camera calibration procedure
Matching Model and Pick Point Editor	For generating point cloud models for the object, editing point cloud models, and adding pick points
Production Interface	For configuring the Production Interface to monitor the Vision System status, add new workpieces or switch workpiece types, and troubleshoot errors, etc

4.1.2.3. Project List

Project list displays opened solutions and projects and their status, including interface communication status and autoload status.

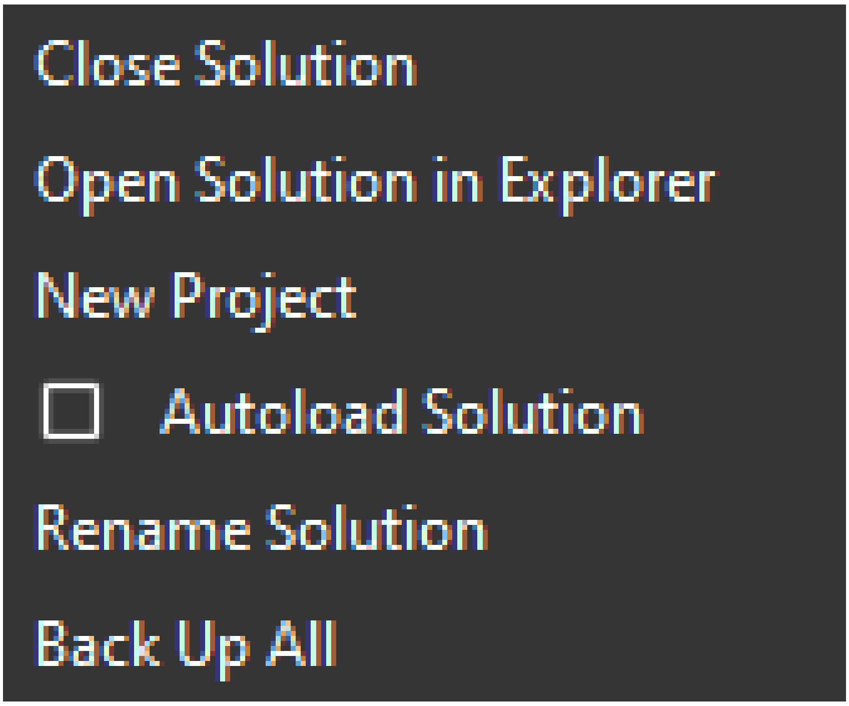
Area 1 is the solution name, area 2 is the project name, as shown in below.



- Hover the cursor over ? to check the introduction to solution and project.
- Click ? to check the user manual on “solution”.

Operating Instructions on Solutions

Select a solution, right-click it to choose actions such as **Close Solution**, **Open Solution in Explorer**, **Autoload Solution**, **Rename Solution**, etc.


Close Solution

Open Solution in Explorer

New Project

□ Autoload Solution

Rename Solution

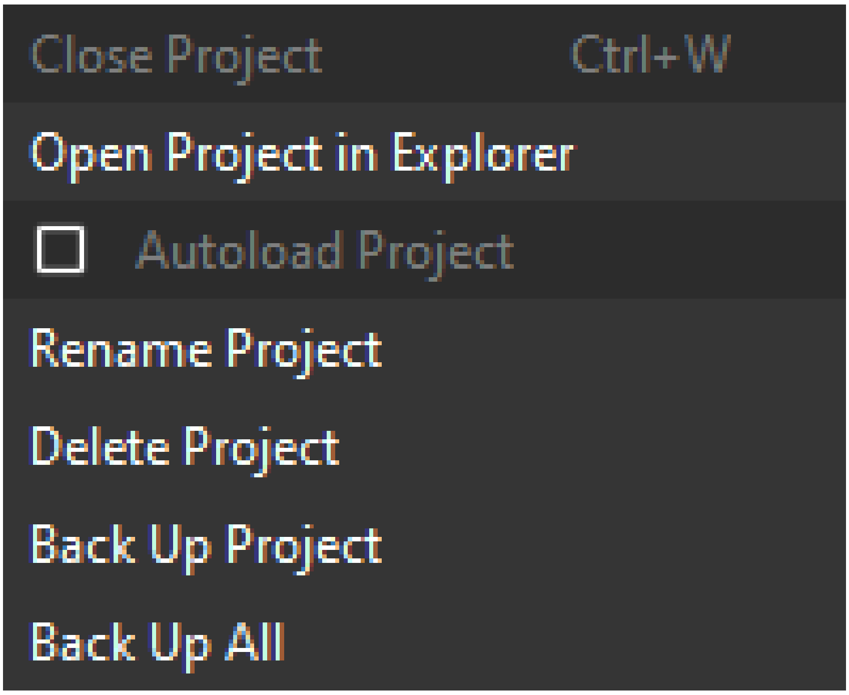
Back Up All

The options and their description are as follows.

Option	Description
Close Solution	Close current solution
Open Solution in Explorer	Open the folder in which the solution is located
Autoload Solution	Autoload all projects in the solution
Rename Solution	Rename the solution
Back Up All	Back up the solution

Operating Instructions on Projects

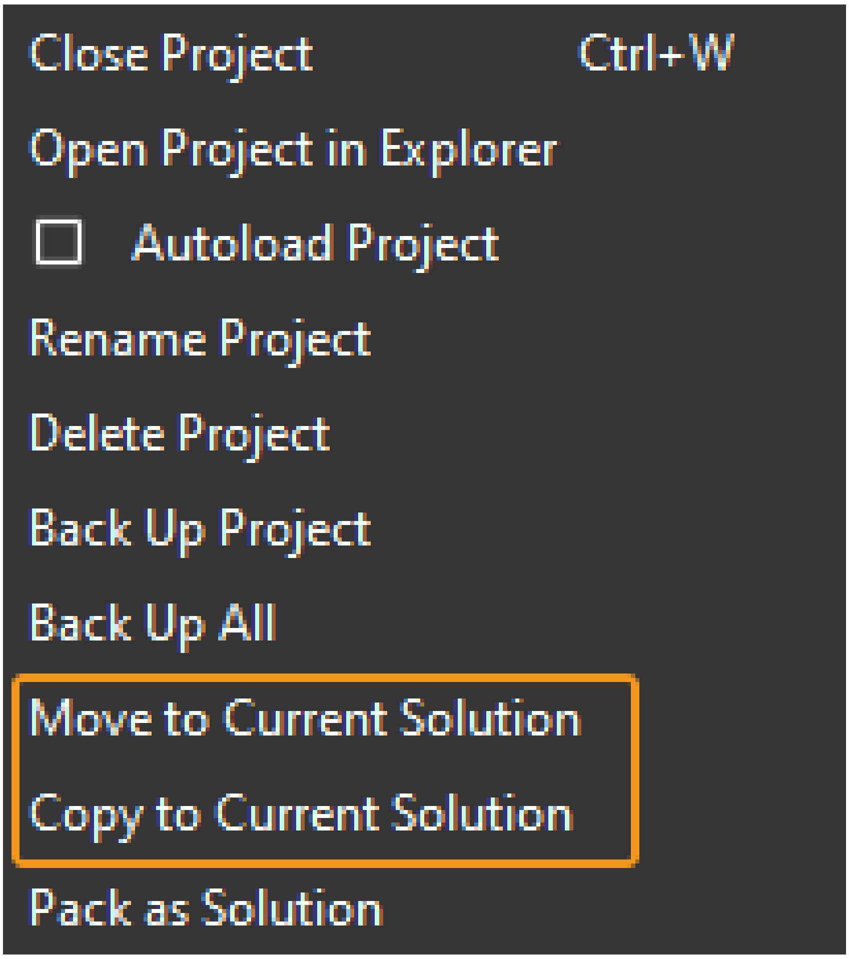
Select a project, right-click it to choose actions such as **Close Project**, **Open Project in Explorer**, **Autoload Project**, **Rename Project**, etc.



The options and their description are as follows.

Option	Description	Shortcut
Close Project	Close current project	Ctrl+W
Open Project in Explorer	Open the folder in which the project is located	N/A
Autoload Project	Autoload the selected project	N/A
Rename Project	Rename the selected project	N/A
Delete Project	Delete the selected project	N/A
Back Up Project	Back up the selected project	N/A
Back Up All	Back up all projects in the solution	N/A
Reset Project IDs	Reset the project's ID	N/A

When there is an opened solution and a project **unassigned** to any solution, you can choose **Move/Copy to Current Solution**.

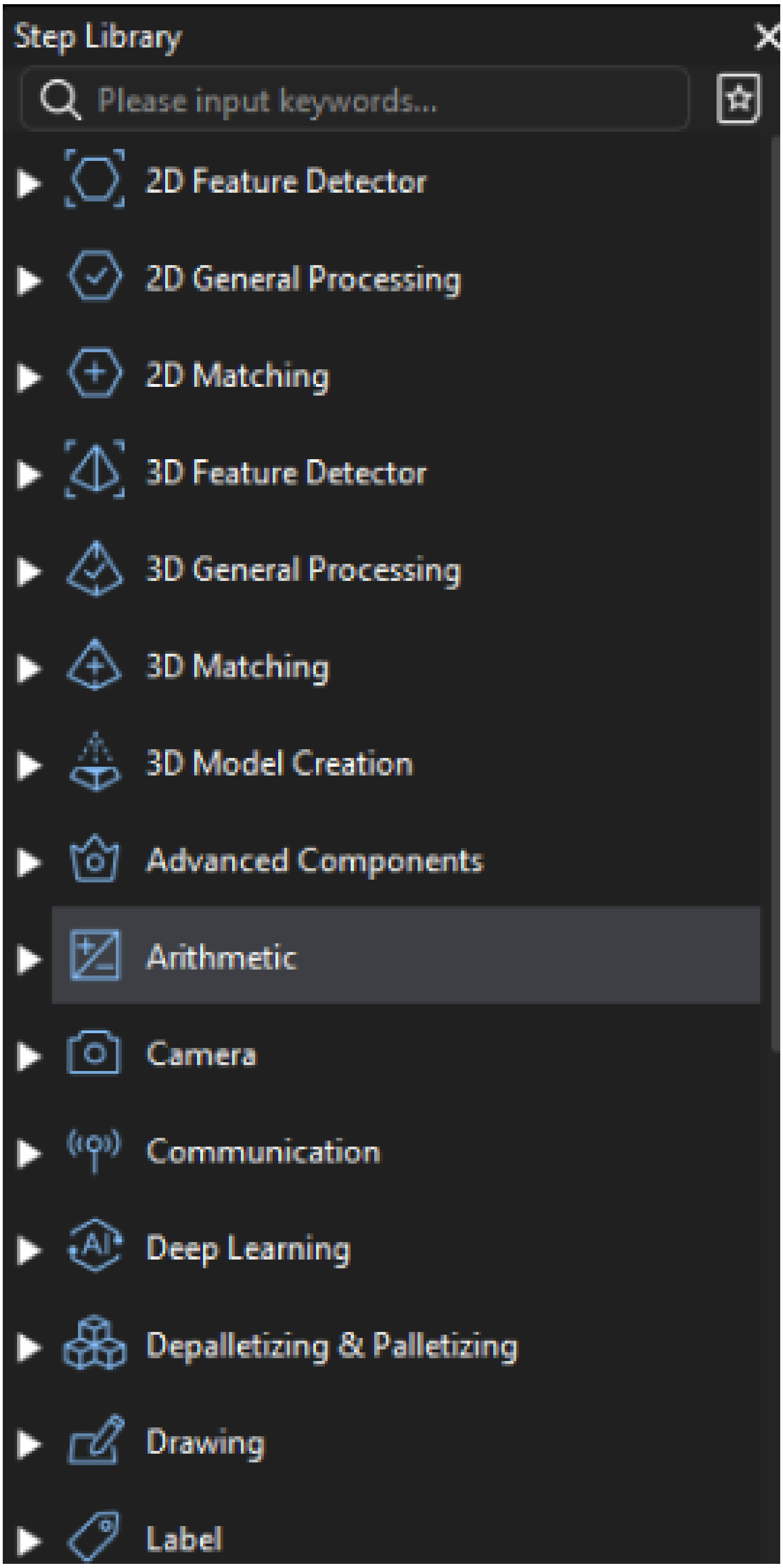


The options and their description are as follows.

Option	Description
Move to Current Solution	Move the selected project into the currently opened solution
Copy to Current Solution	Copy the selected project into the currently opened solution

4.1.2.4. Step Library

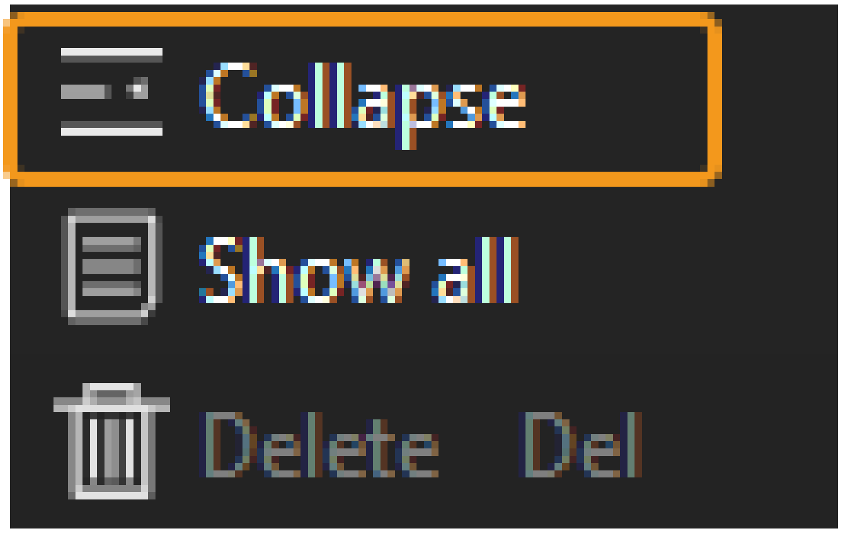
Step library includes all the Steps needed to create a project, as shown below.



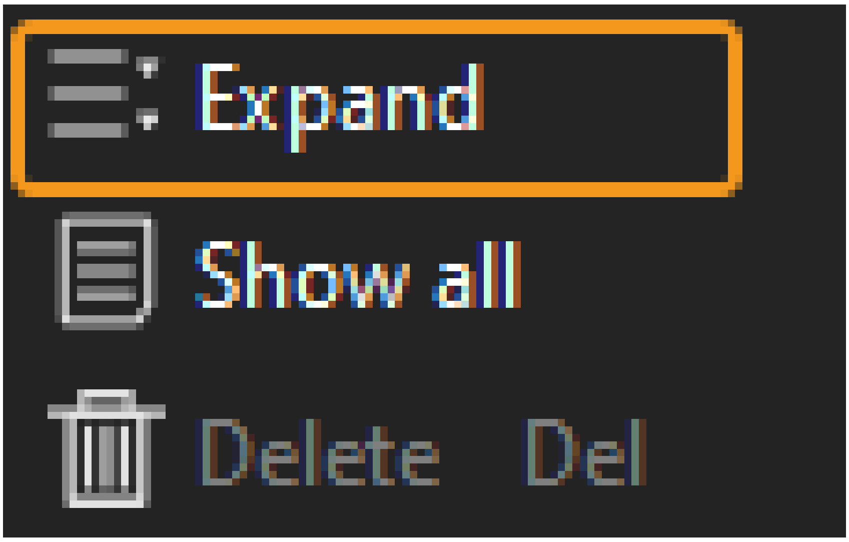
Step Display

Expand/Collapse Steps

When all groups of Steps are expanded in the Step library, click the right mouse button and click **Collapse** to fold all the groups.

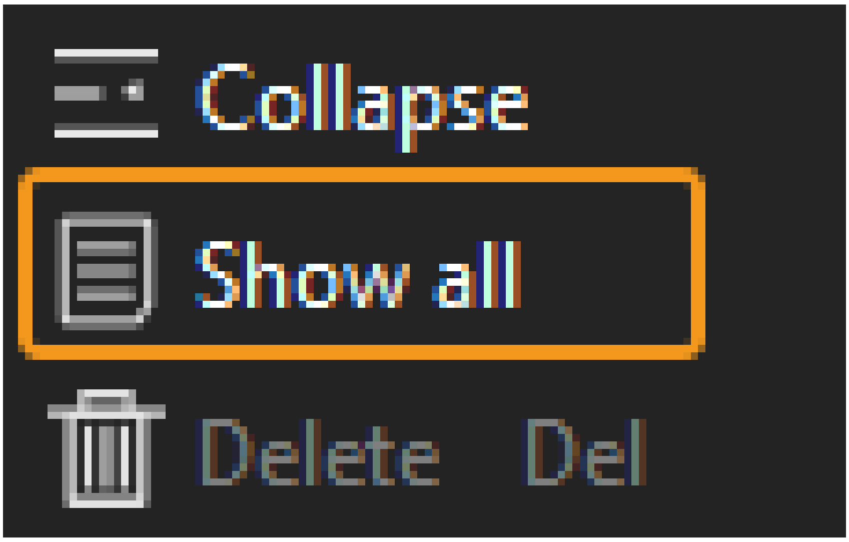


When all groups of Steps are collapsed in the Step library, click the right mouse button and click **Expand** to unfold all the groups.



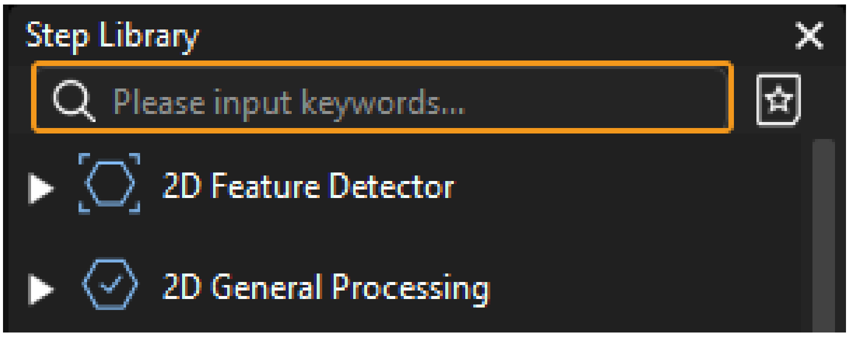
Show Frequently-Used Steps/ Show All

When the button on the right of the search bar is like ☆, the Step library displays frequently-used Steps only. For all Steps to be displayed, you can click the button ☆ on the right of the search bar. When the button turns into ☰, all Steps will be displayed in the Step library. You can also right-click in the Step Library area and select **[Show All]**.

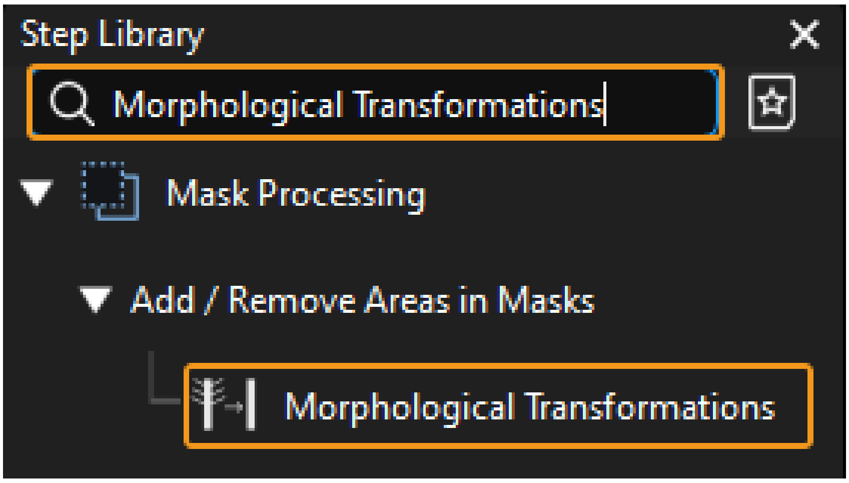


Find Steps

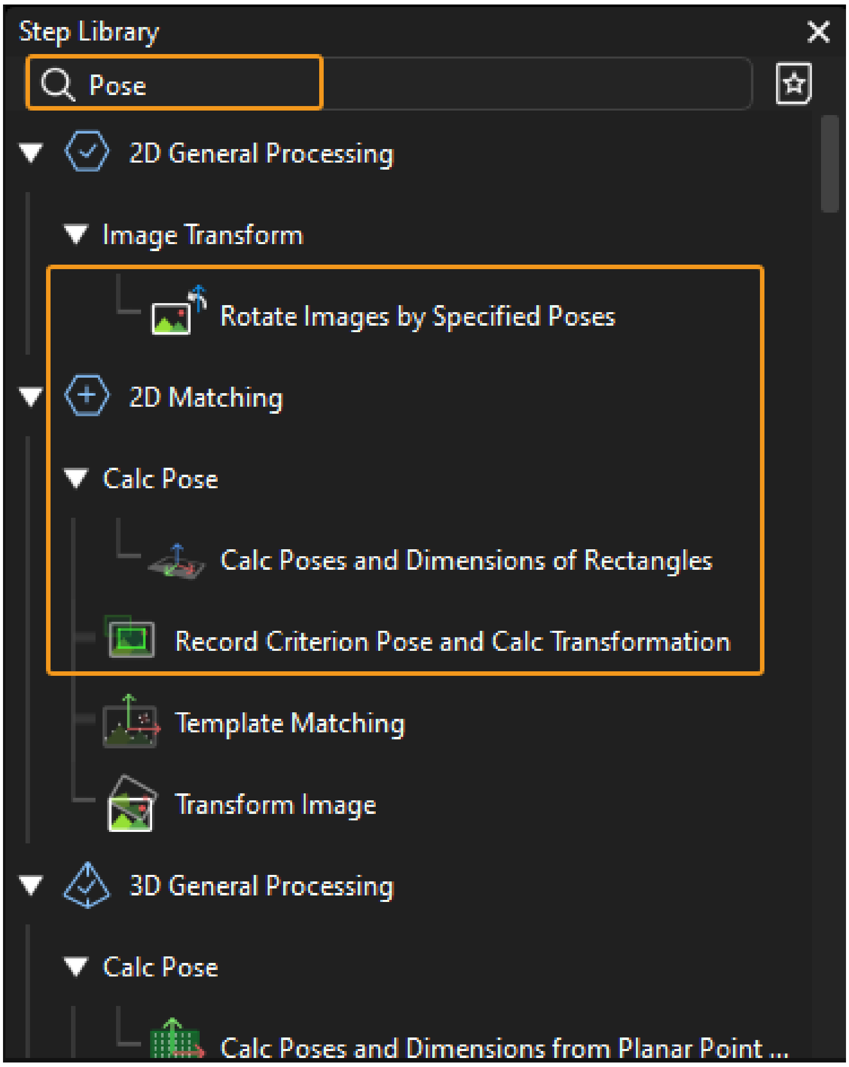
Type a Step name in the search bar to find related Steps. The Step name typed in the bar can be both the full name or key information about the name.



Type in full name of the Step and you can find the exact search result. The Step “Morphological Transformations” is taken as an example in the figure below.

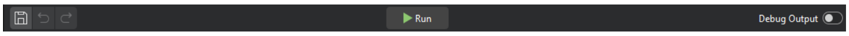

Step Library
Morphological Transformations
Mask Processing
Add / Remove Areas in Masks
Morphological Transformations

Type in key information about the Step, Steps related to the information will be displayed. The Step “Pose” is taken as an example in the figure below.


Step Library
Pose
2D General Processing
Image Transform
Rotate Images by Specified Poses
2D Matching
Calc Pose
Calc Poses and Dimensions of Rectangles
Record Criterion Pose and Calc Transformation
Template Matching
Transform Image
3D General Processing
Calc Pose
Calc Poses and Dimensions from Planar Point ...

4.1.2.5. Project Toolbar

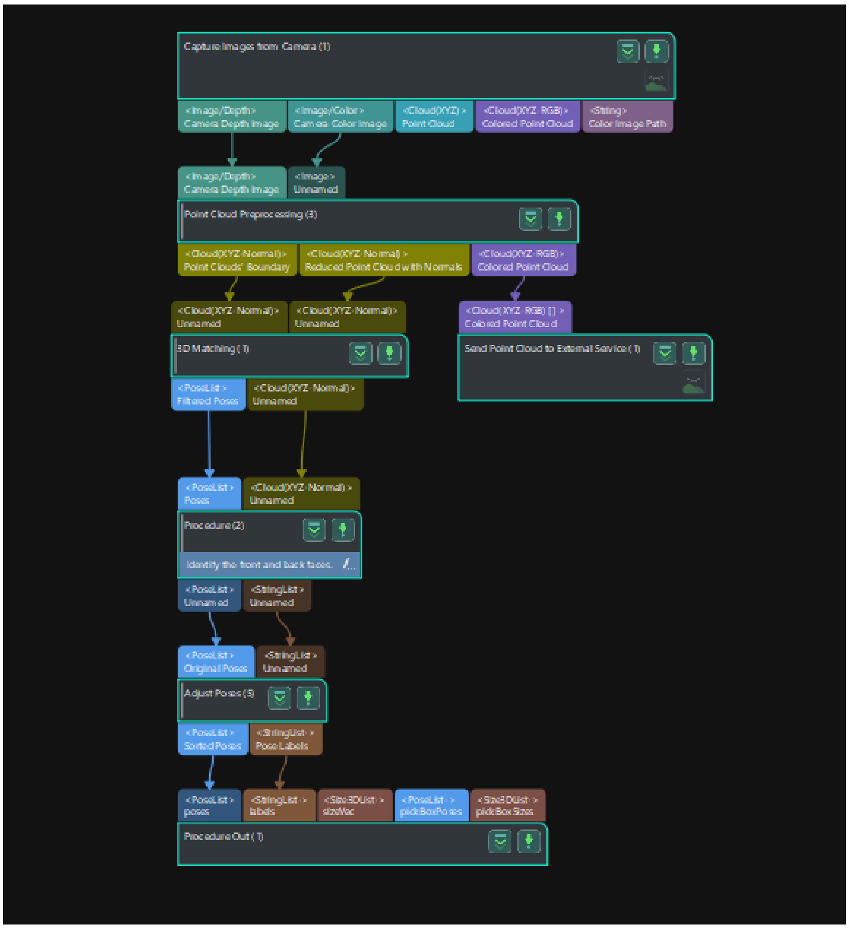
Providing buttons for running, stopping and debugging the project.



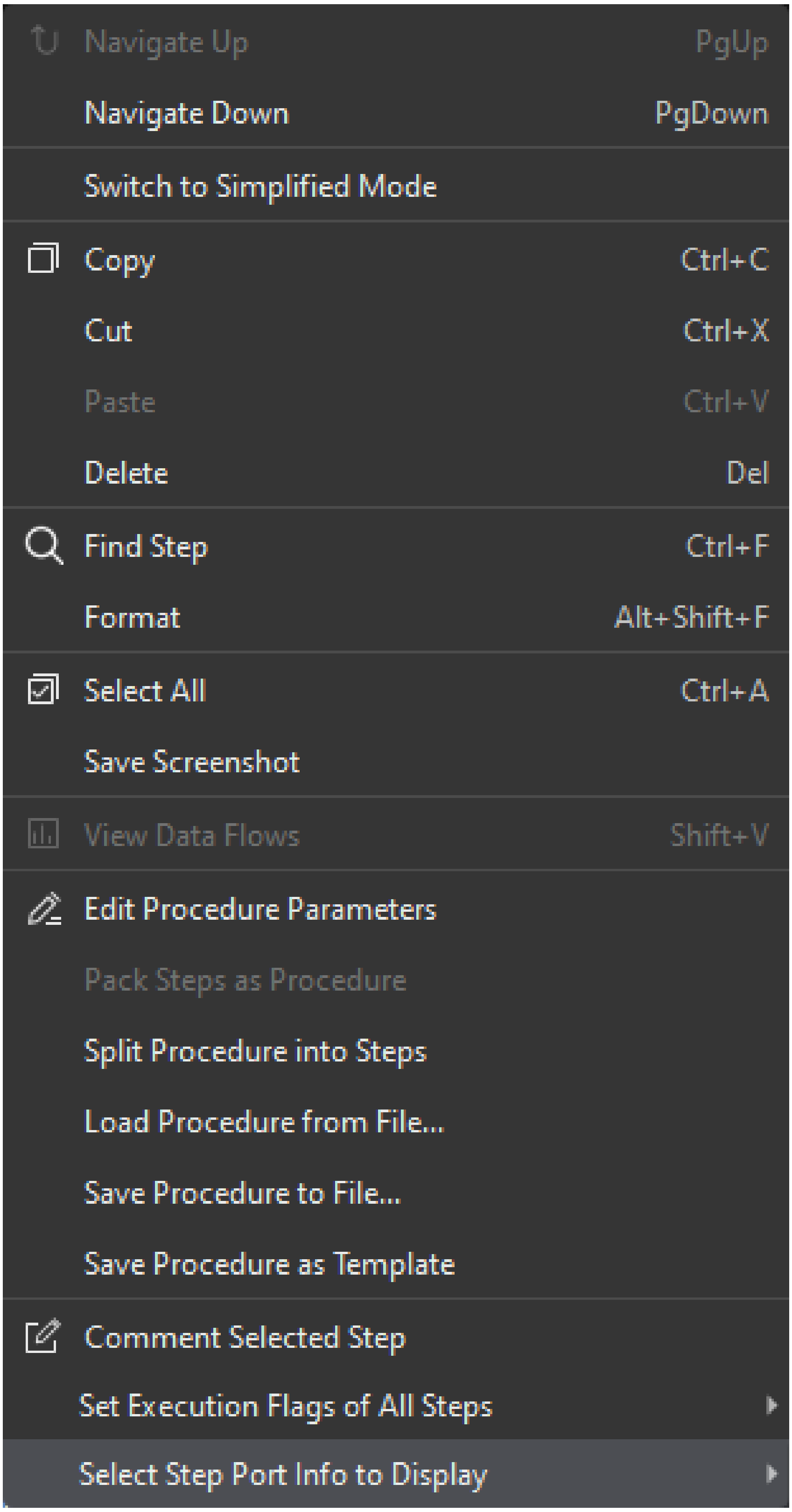
Option	Description
Save Project	Save the project
Undo	Undo the action
Redo	Redo the action
Run	Run the project
Debug Output	When switched on, additional debug information will be displayed

4.1.2.6. Graphical Programming Workspace

During software tuning, the graphical programming workspace is an integral part of editing the logical flows of a vision project. Lines between Steps indicate data flow in standard editing mode and processing procedure in measurement mode.


Capture Images from Camera (1)
<Image/Depth> Camera Depth Image
<Image/Color> Camera Color Image
<Cloud(XYZ)> Point Cloud
<Cloud(XYZ-RGB)> Colored Point Cloud
<String> Color Image Path
<Image/Depth> Camera Depth Image
<Image> Unnamed
Point Cloud Preprocessing (3)
<Cloud(XYZ-Normal)> Point Clouds' Boundary
<Cloud(XYZ-Normal)> Reduced Point Cloud with Normals
<Cloud(XYZ-RGB)> Colored Point Cloud
<Cloud(XYZ-Normal)> Unnamed
<Cloud(XYZ-Normal)> Unnamed
<Cloud(XYZ-RGB) [] > Colored Point Cloud
3D Matching (1)
Send Point Cloud to External Service (1)
<PoseList> Filtered Poses
<Cloud(XYZ-Normal)> Unnamed
<PoseList> Poses
<Cloud(XYZ-Normal)> Unnamed
Procedure (2)
Identify the front and back faces.
<PoseList> Unnamed
<StringList> Unnamed
<PoseList> Original Poses
<StringList> Unnamed
Adjust Poses (5)
<PoseList> Sorted Poses
<StringList> Pose Labels
<PoseList> poses
<StringList> labels
<Size3DList> sizeVec
<PoseList> pickBoxPoses
<Size3DList> pickBoxSizes
Procedure Out (1)

Right-click the blank space in the graphical programming workspace, a context menu with multiple options will pop up, and you can select options in this menu according to your needs.


Navigate Up PgUp
Navigate Down PgDown
Switch to Simplified Mode
Copy Ctrl+C
Cut Ctrl+X
Paste Ctrl+V
Delete Del
Find Step Ctrl+F
Format Alt+Shift+F
Select All Ctrl+A
Save Screenshot
View Data Flows Shift+V
Edit Procedure Parameters
Pack Steps as Procedure
Split Procedure into Steps
Load Procedure from File...
Save Procedure to File...
Save Procedure as Template
Comment Selected Step
Set Execution Flags of All Steps
Select Step Port Info to Display

The options and their description are as follows.

Option	Description	Shortcut
Navigate Up	Move to the previous level of the Procedure	PgUp

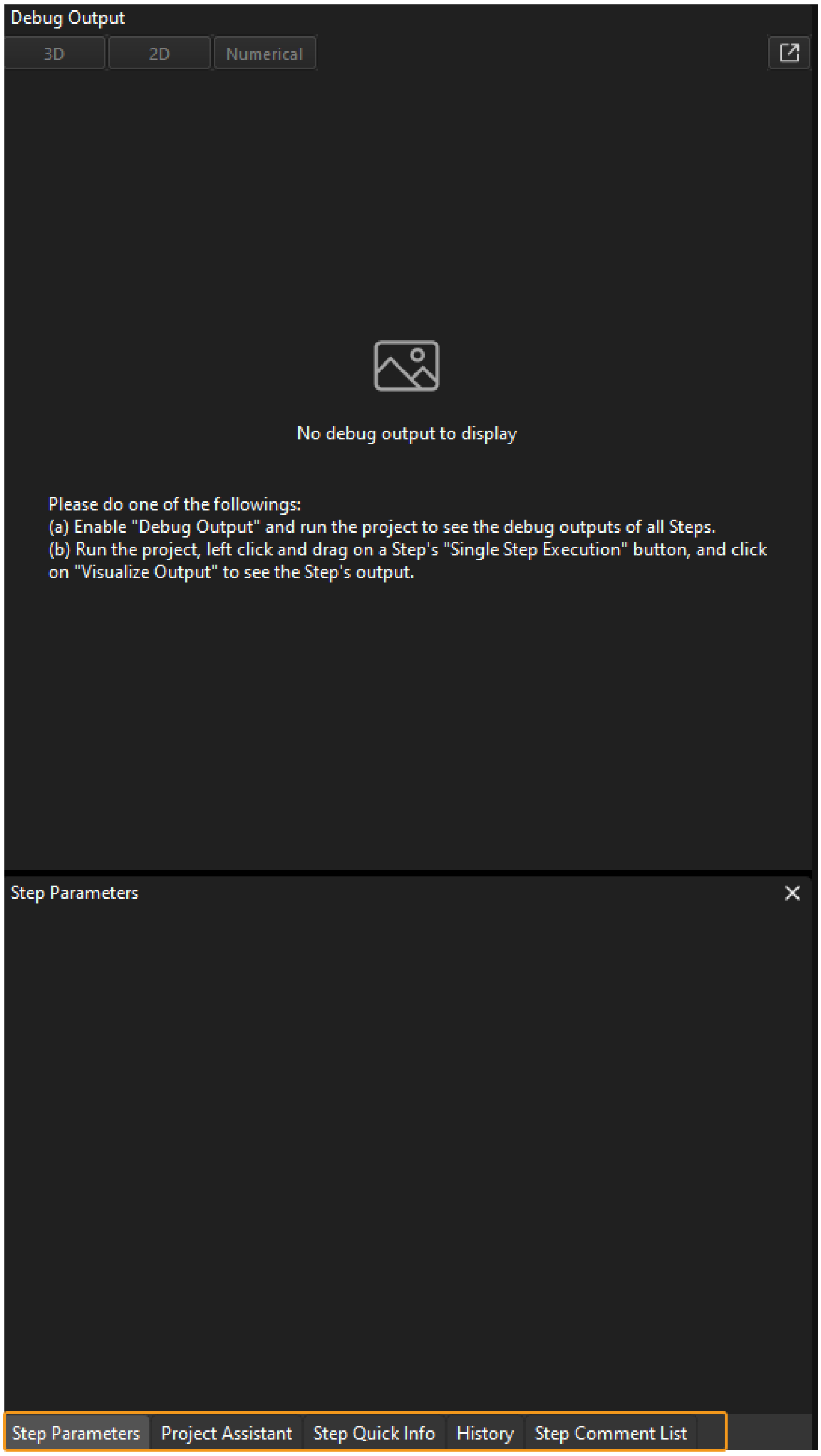
Option	Description	Shortcut
Navigate Down	Move to the next level of the Procedure (Right-click the Procedure to activate the option)	PgDown
Switch to Simplified Mode	Switch all Steps of the current project to simplified mode	N/A
Copy	Copy the Step	Ctrl+C
Cut	Cut the Step	Ctrl+X
Paste	Paste the Step	Ctrl+V
Delete	Delete Steps or data flow	Del
Find Step	Find Steps in current project	Ctrl+F
Format	Auto arrange Steps in current project	Alt+Shift+F
Select All	Select all Steps and data flows in current project	Ctrl+A
Save Screenshot	Save project screenshots to a designated folder	N/A
View Data Flows	Display data input or output in the debug output window (Right-click a data flow to activate the option)	Shift+V
Edit Procedure Parameters	Edit displayed parameters of a Procedure (Right-click the Procedure to activate the option)	N/A
Pack Steps as Procedure	Combine multiple selected Steps into a Procedure	N/A
Split Procedure into Steps	Split a selected Procedure into multiple Steps	N/A
Load Procedure from File	Load a Procedure from an existing JSON file	N/A
Save Procedure to File	Save a Procedure as a JSON file	N/A
Save Procedure as Template	Save the selected Procedure to the “Custom” category in the Step Library	N/A
Comment Selected Step	Add comments on the selected Step	N/A
Set Execution Flags of All Steps	Cancel Text Output	N/A
	Set/Cancel Reloading Files	N/A
	Set/Cancel Continuing Given No Output	N/A

Option	Description	Shortcut
Select Step Port Info to Display	Show Type and Name	Ctrl+Alt+A
	Show Name Only	Ctrl+Alt+N
	Show Type Only	Ctrl+Alt+T

4.1.2.7. Project Configuration Pane

This pane provides panels and tabs including Debug Output, Step Parameters, Project Assistant, Step Quick Info, History and Step Comment List.

> If you cannot find the tab you need in the project configuration pane, go to menu bar › View and check whether the checkbox before the option is selected.

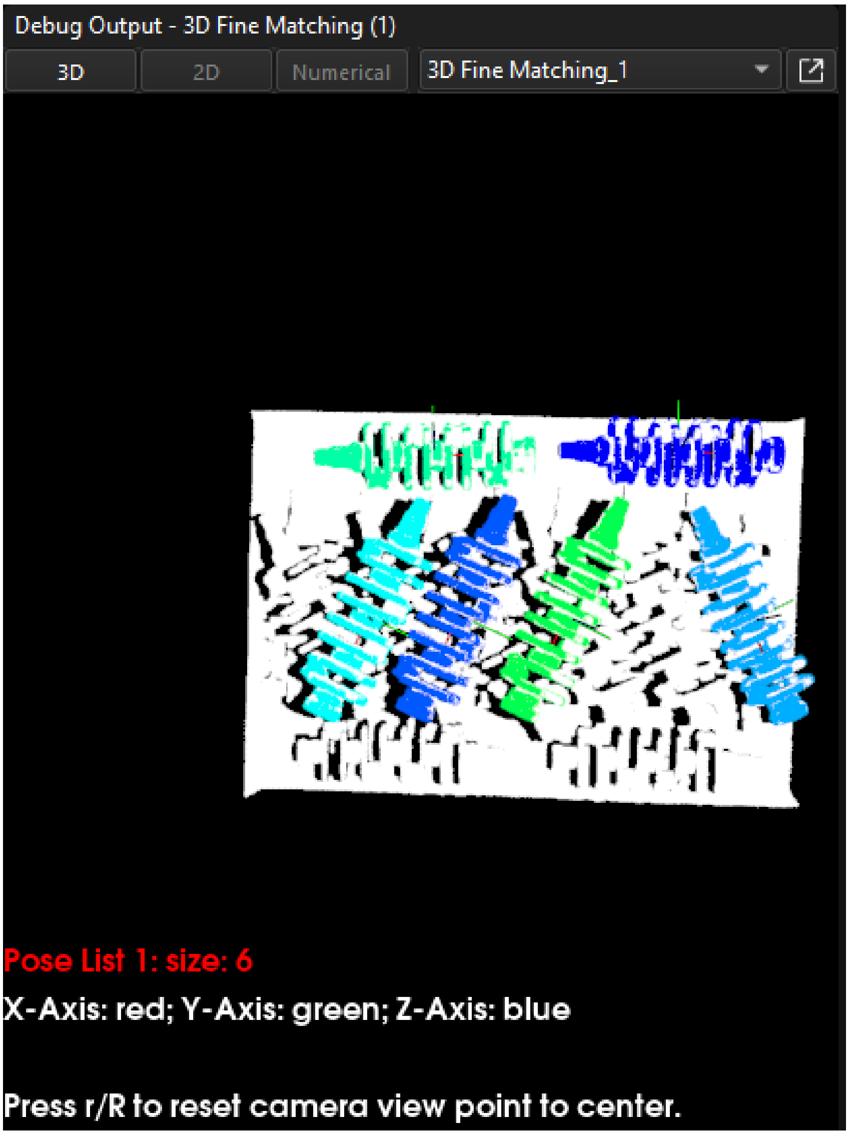


No debug output to display

Please do one of the followings:
(a) Enable "Debug Output" and run the project to see the debug outputs of all Steps.
(b) Run the project, left click and drag on a Step's "Single Step Execution" button, and click on "Visualize Output" to see the Step's output.

Debug Output

When Debug Output in the Project toolbar is switched on, this panel displays the output of Steps as the project is executed. For details, please refer to Run Steps and View Outputs.



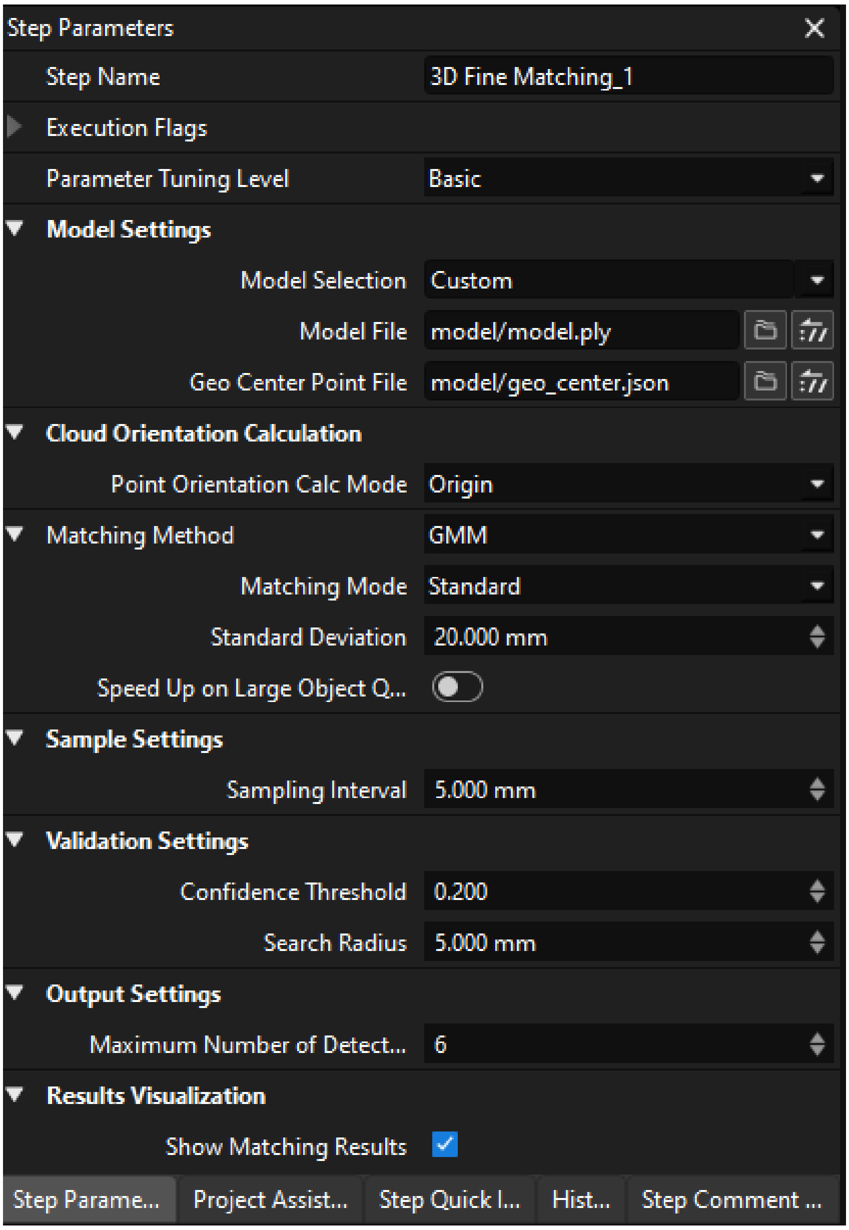
Pose List 1: size: 6

X-Axis: red; Y-Axis: green; Z-Axis: blue

Press r/R to reset camera view point to center.

Step Parameters

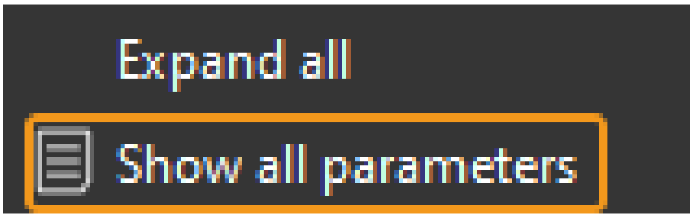
This panel displays the common parameters of the selected Step.



Step Parameters

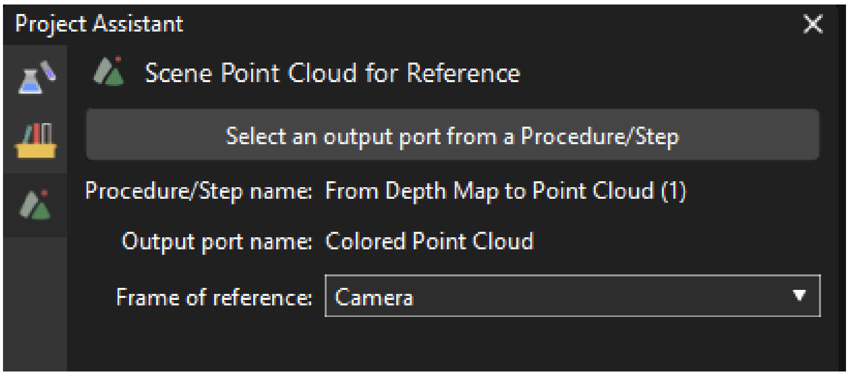
Step Name	3D Fine Matching_1
Execution Flags	
Parameter Tuning Level	Basic
Model Settings	
Model Selection	Custom
Model File	model/model.ply
Geo Center Point File	model/geo_center.json
Cloud Orientation Calculation	
Point Orientation Calc Mode	Origin
Matching Method	GMM
Matching Mode	Standard
Standard Deviation	20.000 mm
Speed Up on Large Object Q...	○
Sample Settings	
Sampling Interval	5.000 mm
Validation Settings	
Confidence Threshold	0.200
Search Radius	5.000 mm
Output Settings	
Maximum Number of Detect...	6
Results Visualization	
Show Matching Results	☑

By default, only frequently used parameters will be displayed in the **Step Parameters** panel. If you cannot find the parameter you need, please right-click on the blank area in the Step Parameters panel and select **Show all parameters** in the context menu.



Project Assistant

This tab provides tools of Parameter Recipe, Data Storage, and Scene Point Cloud for Reference.


Project Assistant
Scene Point Cloud for Reference
Select an output port from a Procedure/Step

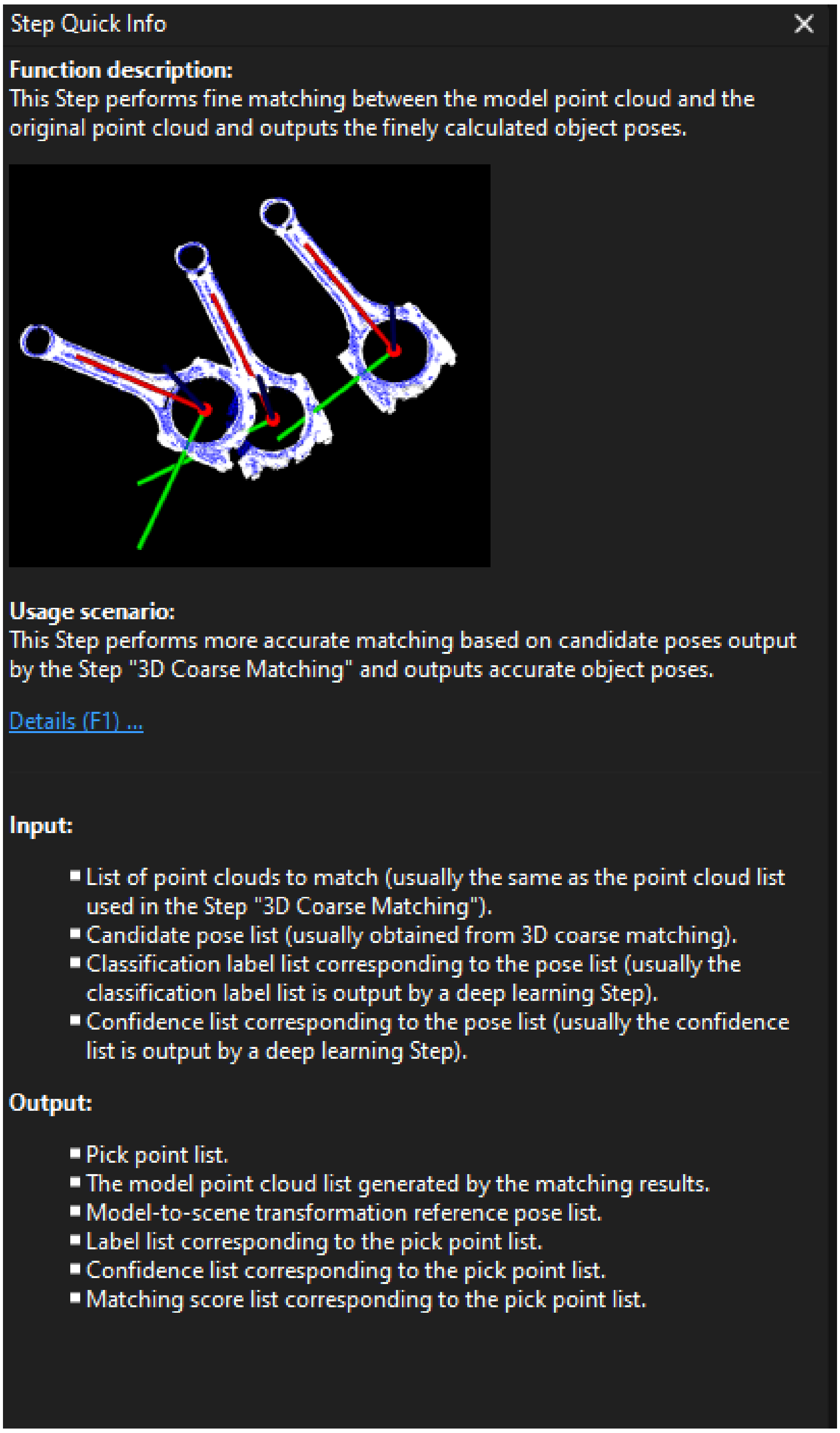
Procedure/Step name: From Depth Map to Point Cloud (1)

Output port name: Colored Point Cloud

Frame of reference: Camera

Step Quick Info

This tab displays a description of the function, usage scenario, input and output of the selected Step, as shown below.



Step Quick Info

Function description:
This Step performs fine matching between the model point cloud and the original point cloud and outputs the finely calculated object poses.

Usage scenario:
This Step performs more accurate matching based on candidate poses output by the Step "3D Coarse Matching" and outputs accurate object poses.

Details (F1) ...

Input:

- List of point clouds to match (usually the same as the point cloud list used in the Step "3D Coarse Matching").
- Candidate pose list (usually obtained from 3D coarse matching).
- Classification label list corresponding to the pose list (usually the classification label list is output by a deep learning Step).
- Confidence list corresponding to the pose list (usually the confidence list is output by a deep learning Step).

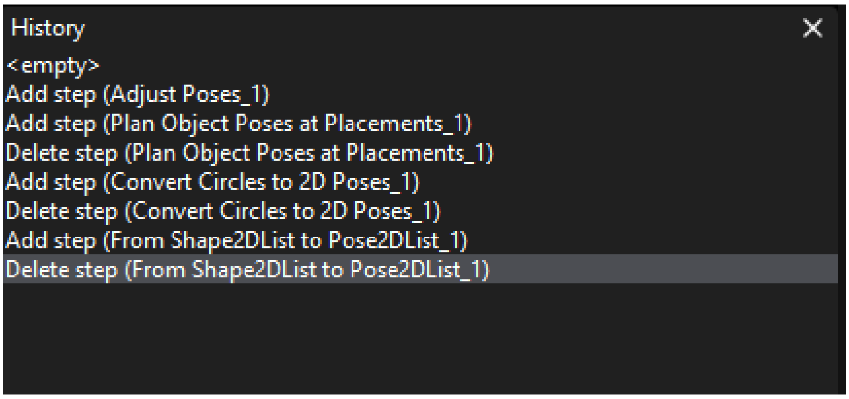
Output:

- Pick point list.
- The model point cloud list generated by the matching results.
- Model-to-scene transformation reference pose list.
- Label list corresponding to the pick point list.
- Confidence list corresponding to the pick point list.
- Matching score list corresponding to the pick point list.

Press F1 to open the corresponding chapter in the User Manual for more detailed information on the selected Step.

History

This tab provides a list of all the actions you have taken to program the project. Clicking an item to return the project to that point in history.



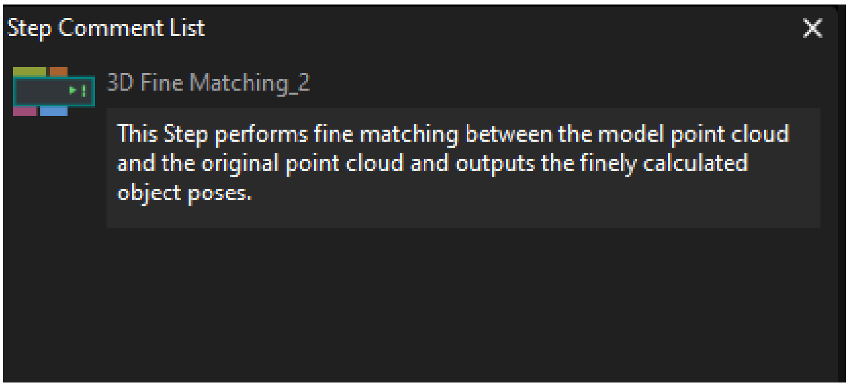
History

<empty>
Add step (Adjust Poses_1)
Add step (Plan Object Poses at Placements_1)
Delete step (Plan Object Poses at Placements_1)
Add step (Convert Circles to 2D Poses_1)
Delete step (Convert Circles to 2D Poses_1)
Add step (From Shape2DList to Pose2DList_1)
Delete step (From Shape2DList to Pose2DList_1)

If you want to use the Step Input Source Selection or Measurement Output window, you can select the corresponding option in View.

Step Comment List

This tab displays the comment added to the selected Step. For detailed instructions on Step comment, please refer to Introduction to Step Comments.



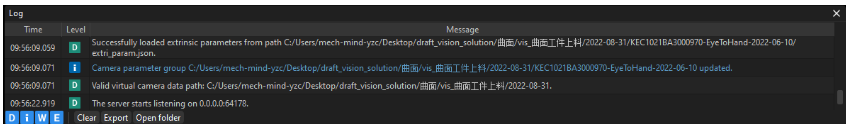
Step Comment List

3D Fine Matching_2

This Step performs fine matching between the model point cloud and the original point cloud and outputs the finely calculated object poses.

4.1.2.8. Log

This panel displays the log in real-time while running a project/solution. You can check the execution record of a certain moment in this panel.



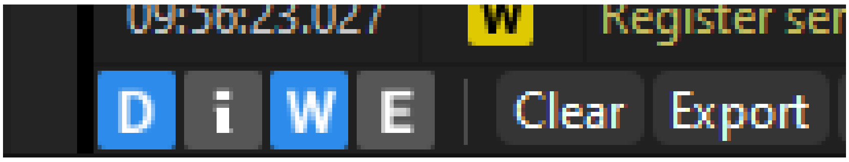
Log

```
Time          Level  Message
09:56:09.059  D      Successfully loaded extrinsic parameters from path C:/Users/mech-mind-yzc/Desktop/draft_vision_solution/曲面/vis_曲面工件上料/2022-08-31/KEC1021BA3000970-EyeToHand-2022-06-10/
                     extri_param.json.
09:56:09.071  i      Camera parameter group C:/Users/mech-mind-yzc/Desktop/draft_vision_solution/曲面/vis_曲面工件上料/2022-08-31/KEC1021BA3000970-EyeToHand-2022-06-10 updated.
09:56:09.071  D      Valid virtual camera data path: C:/Users/mech-mind-yzc/Desktop/draft_vision_solution/曲面/vis_曲面工件上料/2022-08-31.
09:56:22.919  D      The server starts listening on 0.0.0.0:64178.
```

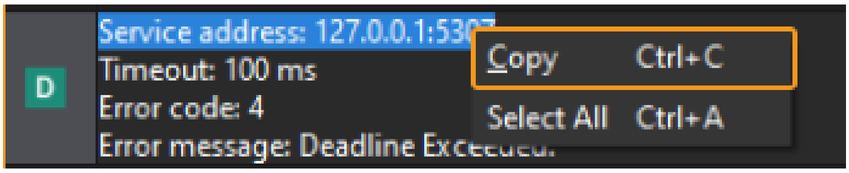
There are four levels of log, as shown below.

Level	Description
D	Debug information
i	Information
W	Warning Information
E	Error information

You can click the icons of log level in the lower left-hand corner of the log panel to choose the type of log information you'd like to display. An icon in blue indicates that log information of this level is displayed. When an icon is in gray, the log information of this level is hidden.



Double-click a single line in the log, select the content, and right-click and select **Copy** in the pop-up menu to copy the selected content.



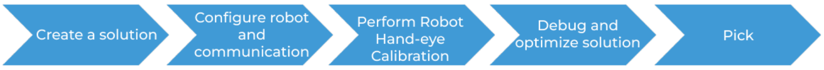
```
Service address: 127.0.0.1:530?
Timeout: 100 ms
Error code: 4
Error message: Deadline Exceeded.
```

4.1.3. Getting Started with Solution

This part introduces how to build a vision solution.

A solution is a collection of functional configurations and data for robot, communication, vision processing, and path planning required for a vision application.

A solution goes through five processes from creation to execution, as shown in the diagram below.


Create a solution
Configure robot and communication
Perform Robot Hand-eye Calibration
Debug and optimize solution
Pick

Create a Solution

A solution is a collection of functional configurations and data for robot, communication, vision processing, and path planning required for a vision application. You need to create a solution first before moving on to the next process.

1. Create and save a solution.
2. Autoload the solution.
3. Build the project.

Configure Robot and Communication

After creating a solution, you can start configuring the robot and communication.

Communication refers to the information exchange and transmission between the vision side and the robot side. To ensure normal communication between the **vision side** and the **robot side**, you need to configure the communication on both sides.

In Mech-Mind Vision System, there are two communication methods: interface communication and master-control communication. And interface communication can be divided into Standard Interface communication and Adapter communication.

Interface communication

For Interface communication, you need to configure both the vision side and the robot side, and make a trial run to ensure normal communication between the two sides.

Notably, different brands of robot require different methods of communication between the robot side and the vision side.

1. Configure communication on the vision side.
2. Configure communication on the vision side, and test whether the robot side and the vision side can communicate normally with each other.

Master-Control Communication

For master-control communication, you need to configure communication on the robot side, and then test whether the robot side and the vision side can communicate normally with each other.

1. Configure communication on the robot side.
2. Test whether the robot side and the vision side can communicate normally with each other.

Perform Robot Hand-Eye Calibration

When robot and communication configuration is completed, you can move on to robot hand-eye calibration.

Hand-eye calibration refers to the process of establishing a positional correspondence between the camera frame and the robot frame. During the process, the object poses defined by the vision system are transformed into poses under the robot frame, so as to guide the robot to perform picking tasks.

Robot Hand-eye Calibration is an essential process, as the calibration results and stability have a direct influence on the robot’s accuracy.

Tune and Improve the Solution

When the above processes are all completed, you can start running and tuning the project.

1. Run and tune the project.
2. Run the Steps and check the visualization outputs.

Pick

After running and tuning the project, you can apply it to a real robot to perform picking tasks.

- If Interface communication is adopted, you need to write a program for the robot to pick and place objects, and, through the Standard Interface, trigger the Mech-Vision solution and acquire its vision point and waypoint outputs from the Mech-Vision project.
- If master-control communication is adopted, you need to plan the waypoint by building a Mech-Viz project.

4.2. Quick Guide of Mech-Vision

This chapter provides operating instructions for functions of Mech-Vision.

Read the section below for **Quick Guide for Solution Library**.

Quick Guide for Solution Library

Read the section below for **Quick Guide for Solutions and Projects**.

Quick Guide for Solutions

Quick Guide for Projects

Read the section below for **Communication Settings Guide**.

Quick Guide for Robot and Communication Configuration

Read the section below for **Step Usage Guide**.

Basic Procedures for Using Steps

4.2.1. Quick Guide for Solution Library

In the Mech-Vision Solution Library, you can find example solutions and projects with data for different industries.

This chapter introduces you to basic procedures for using the solution library.

Open the Solution Library

Search for a Solution or Project

Check the Brief Introduction to Solutions or Projects

Open User Manual for Solutions and Projects

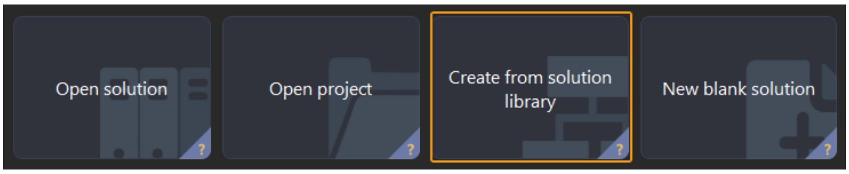
Create a Solution from the Solution Library

4.2.1.1. Open the Solution Library

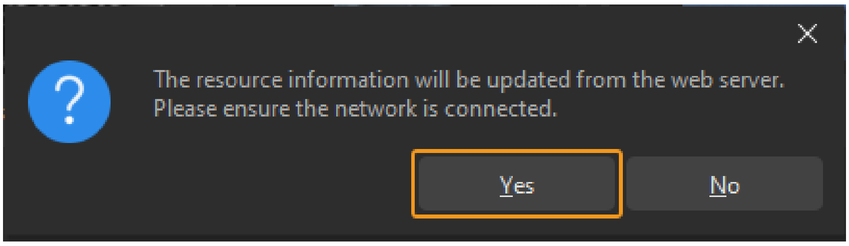
This section introduces how to open the Solution Library.

Open from the Welcome Interface

After you open Mech-Vision, click **[Create from solution library]** in the Welcome interface to open the Solution Library.

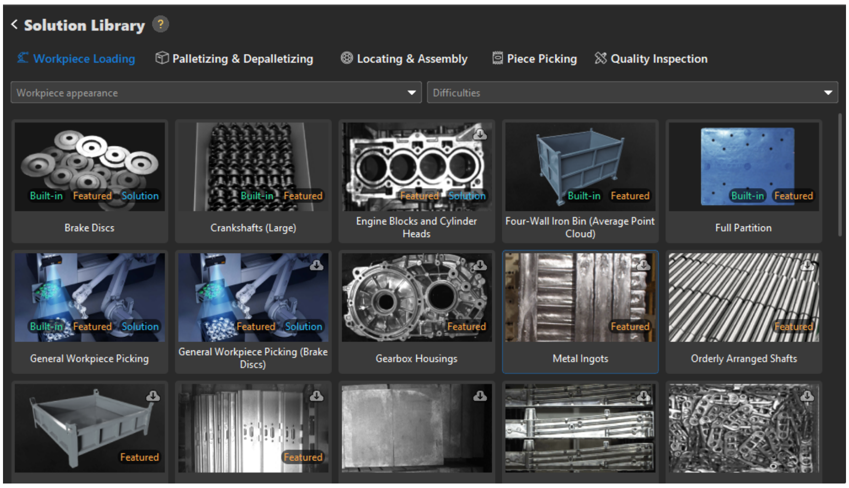


There are some built-in resources in the Solution Library. If you want to view more solutions, click **[More]** at the bottom of the interface. Click **[Yes]** in the pop-up window to start loading.



The resource information will be updated from the web server. Please ensure the network is connected.

After the resources are loaded, you can see more solutions in the Solution Library, as shown below.



- Solution: a collection of configurations and data of robot communication, vision processing, path planning and other functions for implementing the vision application.
- Featured: indicates that the solution can be widely used and its feasibility has been verified on site for many times.
- Built-in: indicates that the solution is locally installed with the software, and there is no need to download it.

Open from the Main Interface

If you have already entered the main interface, you can open the Solution Library in either of the following three ways.

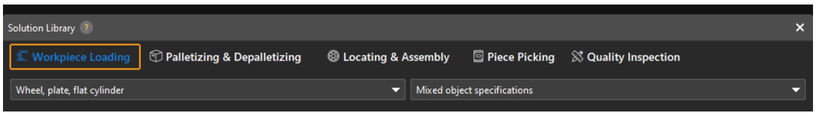
- Click **[Create from Solution Library]** in the toolbar.
- Select File › Create from Solution Library in the menu bar.

4.2.1.2. Search for a Solution or Project

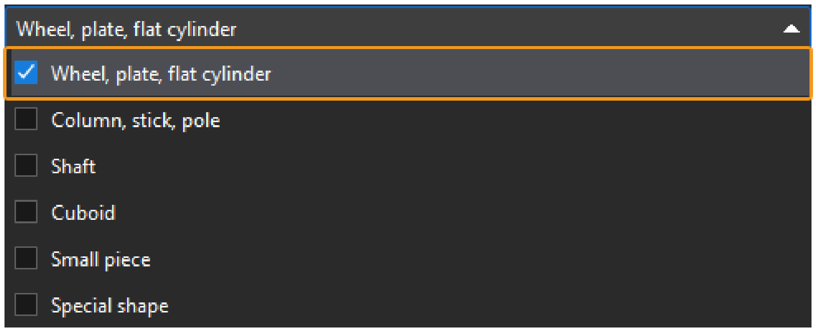
In the solution library, you can search the solution or project you need according to the **usage scenarios**. In the workpiece loading scenario, you can filter your search by the **Workpiece appearance** and **Difficulties**.

The following steps shows how to search for a **Mixed object specifications Brake Discs** project in the **Workpiece Loading** scenario.

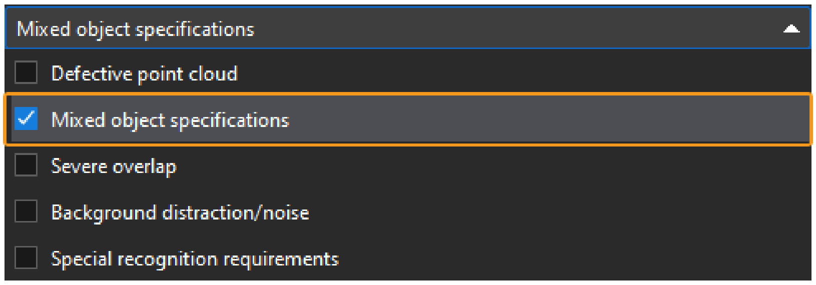
1. Filter search by usage scenarios. Click **[Workpiece Loading]** on the top of the Solution Library, and all solutions and projects of this scenario will appear in the window.



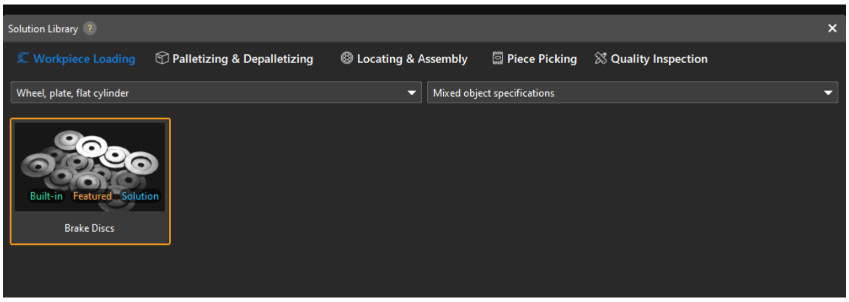
2. Filter search by workpiece appearance. Select **Wheel, plate, flat cylinder** in the drop-down menu of **Workpiece appearance** box, and solutions and projects related to the wheel, plate, and flat cylinder in the workpiece loading scenario will appear in the window.



3. Filter search by difficulties. Select **Mixed object specifications** in the drop-down menu of **Difficulties** box, and solutions and projects of mixed object specifications will appear in the window.

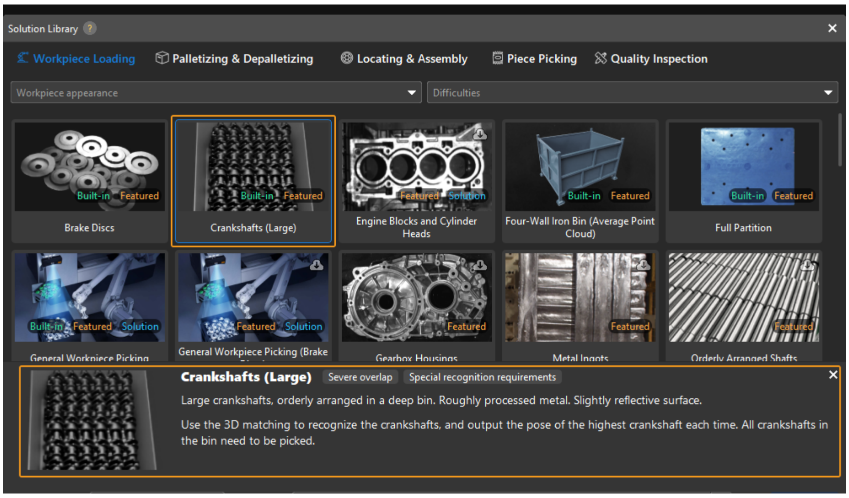

Mixed object specifications
□ Defective point cloud
☑ Mixed object specifications
□ Severe overlap
□ Background distraction/noise
□ Special recognition requirements

Then you can find the Brake Discs project in the Solution Library.



4.2.1.3. Check the Brief Introduction to Solutions or Projects

If you want to know more about the solution or project, click the solution or project in the Solution Library, and a brief introduction will appear at the bottom of the interface.

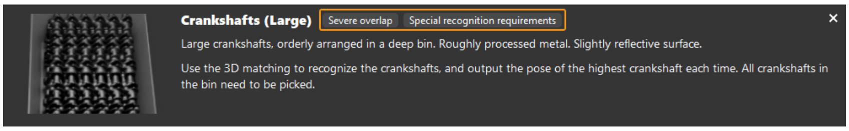


Crankshafts (Large) Severe overlap Special recognition requirements

Large crankshafts, orderly arranged in a deep bin. Roughly processed metal. Slightly reflective surface.

Use the 3D matching to recognize the crankshafts, and output the pose of the highest crankshaft each time. All crankshafts in the bin need to be picked.

In addition, tags of **project characteristics** are added next to the project names, as shown below.



Crankshafts (Large) Severe overlap Special recognition requirements

Large crankshafts, orderly arranged in a deep bin. Roughly processed metal. Slightly reflective surface.

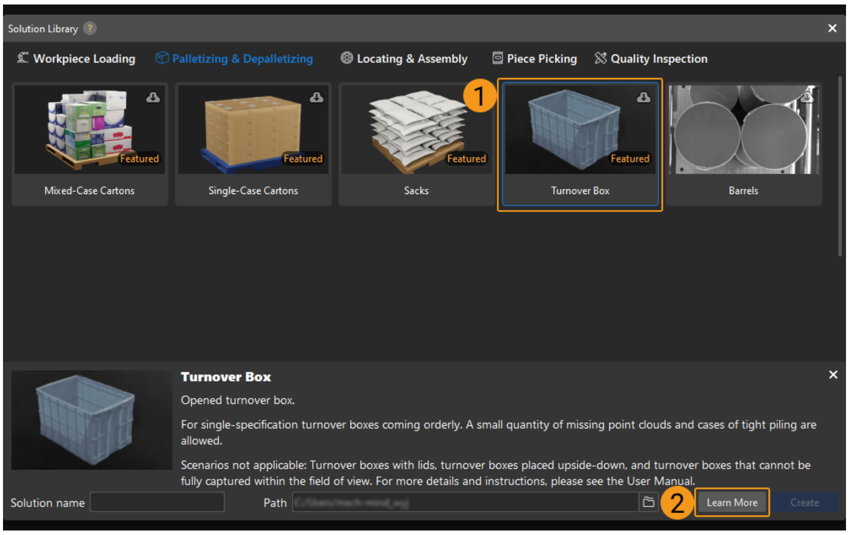
Use the 3D matching to recognize the crankshafts, and output the pose of the highest crankshaft each time. All crankshafts in the bin need to be picked.

4.2.1.4. Open User Manual for Solutions and Projects

The user manuals for solutions and projects cover the content of usage scenarios, technical specifications, project difficulties, vision solutions, parameter tuning recommendations, etc.

If there is a user manual for the solution or project, after you click the project or solution in the Solution Library, you can see a **[Learn More]** button in the lower right corner of the interface.

To check the user manual of a certain project or solution, click the **[Learn More button]** in the lower right corner.



Turnover Box

Opened turnover box.

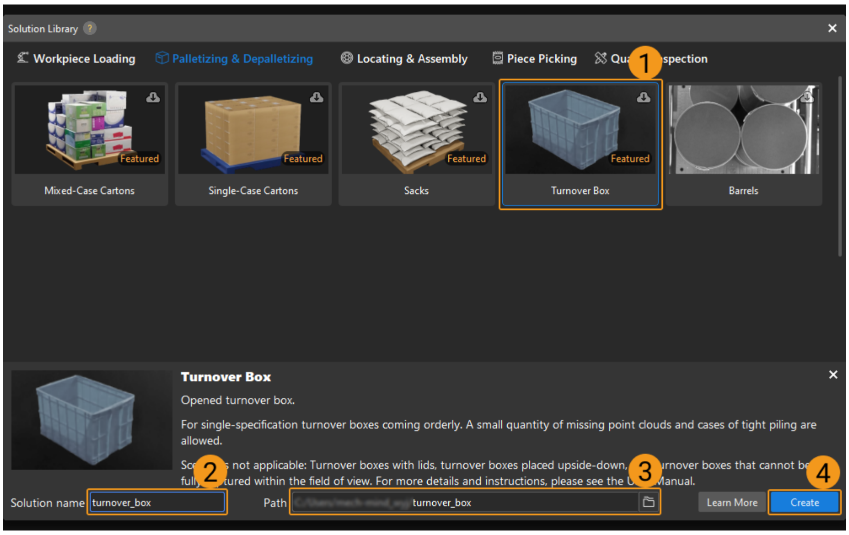
For single-specification turnover boxes coming orderly. A small quantity of missing point clouds and cases of tight piling are allowed.

Scenarios not applicable: Turnover boxes with lids, turnover boxes placed upside-down, and turnover boxes that cannot be fully captured within the field of view. For more details and instructions, please see the User Manual.

Solution name
Path

4.2.1.5. Create a Solution from the Solution Library

To create a solution from the Solution Library, open the Solution Library first. Then, select the solution, and enter a **Solution name** and **Path** to store the solution. Click **[Create]** in the end.



Turnover Box

Opened turnover box.

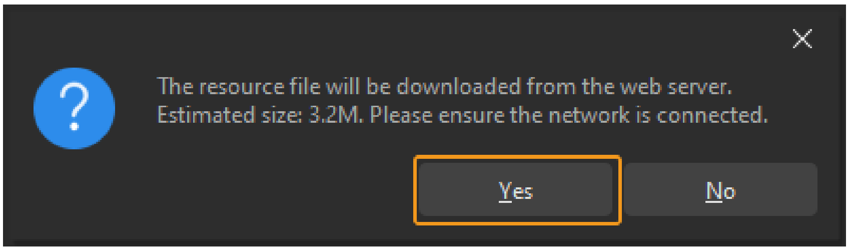
For single-specification turnover boxes coming orderly. A small quantity of missing point clouds and cases of tight piling are allowed.

Sce s not applicable: Turnover boxes with lids, turnover boxes placed upside-down, urnover boxes that cannot be fully tured within the field of view. For more details and instructions, please see the U Manual.

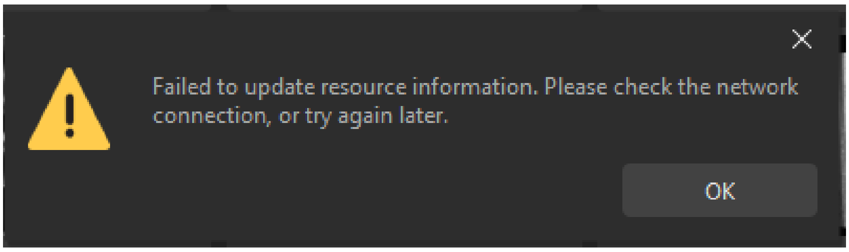
Solution name turnover_box

Path [redacted]turnover_box

Select [Yes] in the pop-up window to create the solution to the specified path.


The resource file will be downloaded from the web server.
Estimated size: 3.2M. Please ensure the network is connected.

If an error message as shown below pops up, there is no available solution on the server. Please check your network connection first, and retry after the network connection is successful. If the problem still exists, please contact Mech-Mind Technical Support.


Failed to update resource information. Please check the network connection, or try again later.

4.2.2. Quick Guide for Solutions

Solution is a collection of configurations and data of robot communication, vision processing, path planning and other functions, which are all integral components of a vision application.

This chapter introduces you to basic procedures for using projects.

Create and Save Solution

Open Solution

Autoload Solution

Rename the Solution

Close the Solution

Understand Solution Structure

4.2.2.1. Create and Save Solution

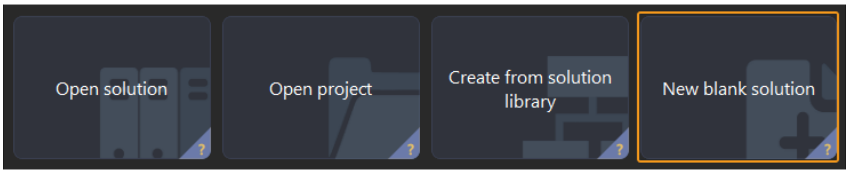
This section introduces how to create a new solution and save it.

Create a New Solution

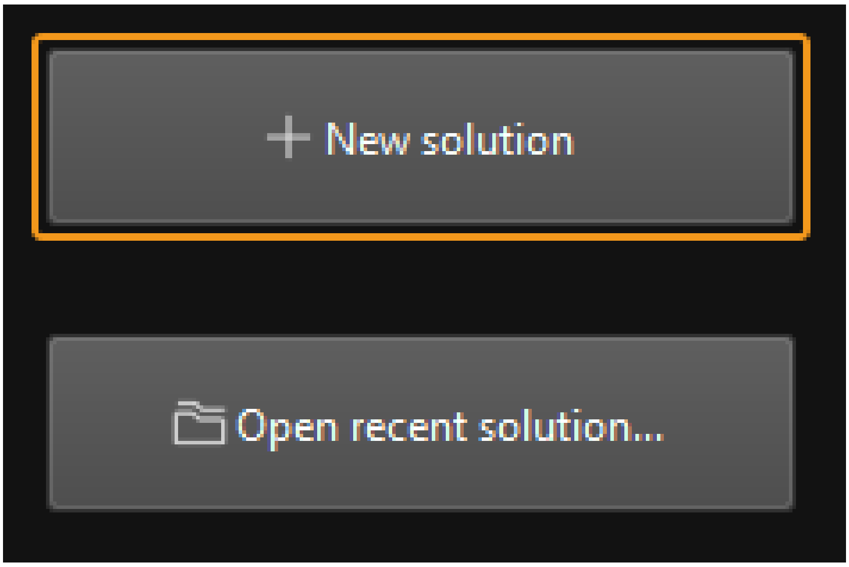
Create a New Blank Solution

There are three ways to create a new blank solution:

- Click [**New blank solution**] in the Welcome interface.



- Select File › New Solution in the menu bar of the main interface.
- If you do not open any solution or project in the main interface of Mech-Vision, click [**New Solution**] in the middle of the interface.



Create from Solution Library

Please refer to Create a new solution or project from Solution Library.

Save the Solution

After a new solution is created, it is recommended to specify a directory to save the solution in time.

There are two ways to save the solution:

- Select File › Save Solution in the menu bar.

- Press `Ctrl` + `Shift` + `S` on the keyboard.

> - If you want to rename the solution, please refer to Rename the Solution.
> - If you want the solution to be opened automatically the next time you open the software, please refer to Autoload Solution.

4.2.2.2. Open Solution

This section introduces how to open a solution.

Open an Existing Solution

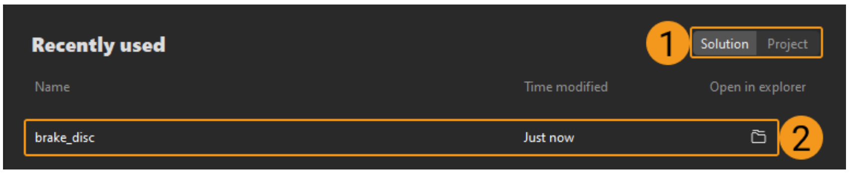
There are two ways to open an existing solution:

- Click **[Open solution]** in the Welcome interface, and select the solution folder in the Open Solution window.
- Select File › Open Solution in the menu bar of the main interface, and select the solution folder in the Open Solution window.

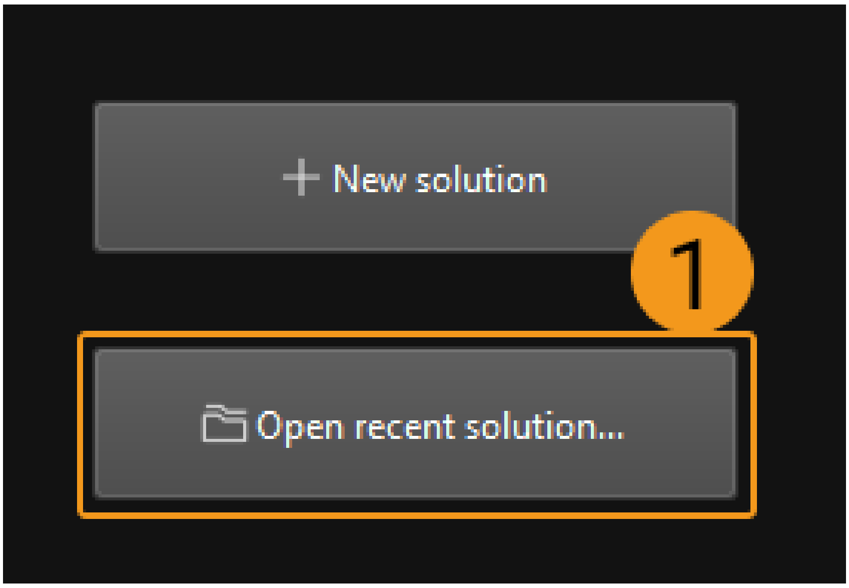
Open a Recently Opened Solution

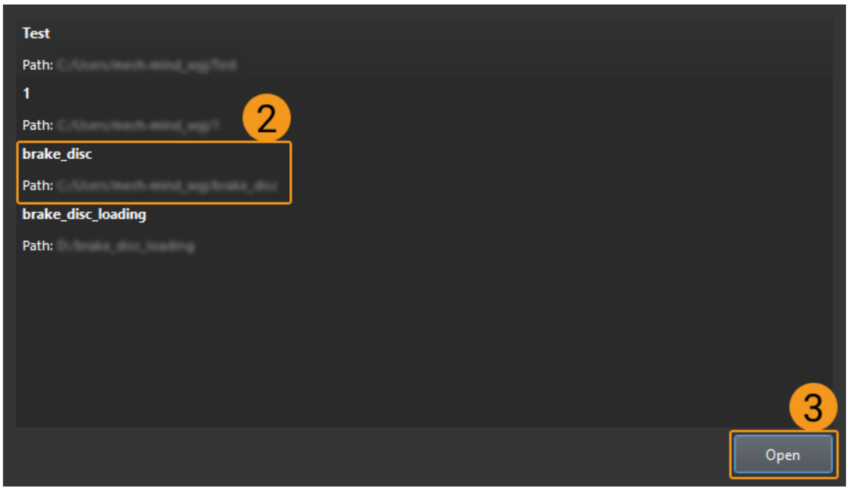
There are three ways to open a recently opened solution:

- Click **Solution** on the Solution/Project button in the Welcome interface, and double-click the solution that you wish to open.



- Click File › Open Recent… in the menu bar of the main interface, and then select the solution in the context menu to open.
- If you do not open any solution or project in the main interface of Mech-Vision, you can see two buttons in the middle of the interface. Select **[Open recent solution…]**, and select the solution you want to open in the pop-up window, and then click Open.

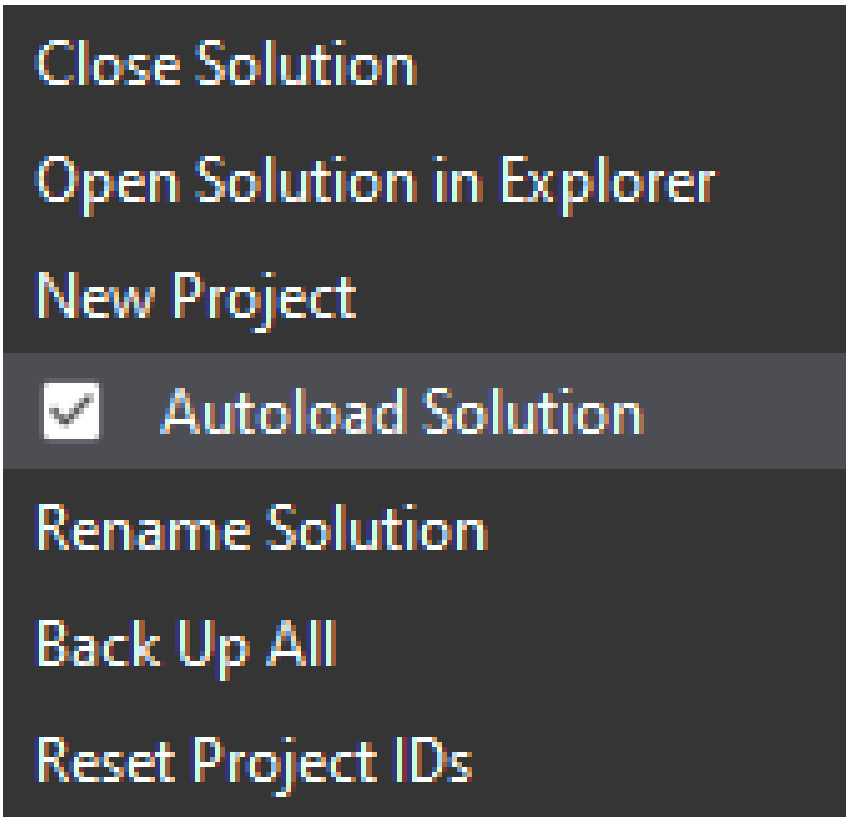




4.2.2.3. Autoload Solution

If you want a solution to be opened automatically the next time you open Mech-Vision software, please do the following step.

Right-click the solution in the Project List and select **Autoload Solution**, as shown below.

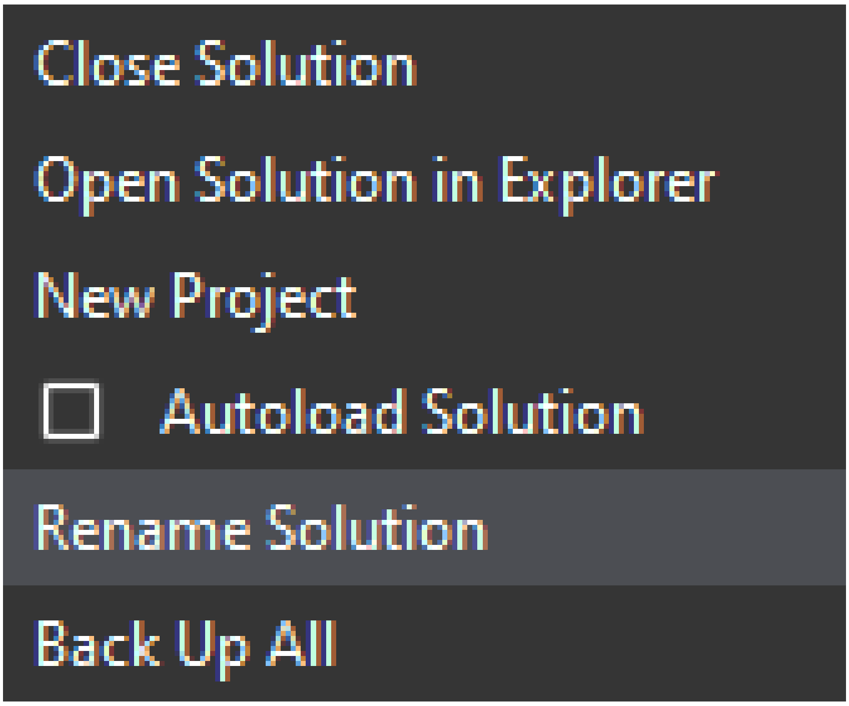


When **Autoload Solution** is selected, all the projects belonging to the solution will be autoloaded. Therefore, the next time you open the software, you will enter the main interface of the software directly, and the Welcome interface will not appear.

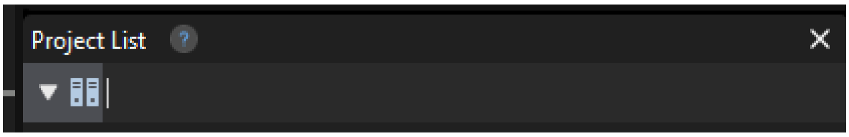
4.2.2.4. Rename the Solution

This section introduces how to rename a solution.

To rename a solution, right-click the solution name in the Project List and select **Rename Solution**.



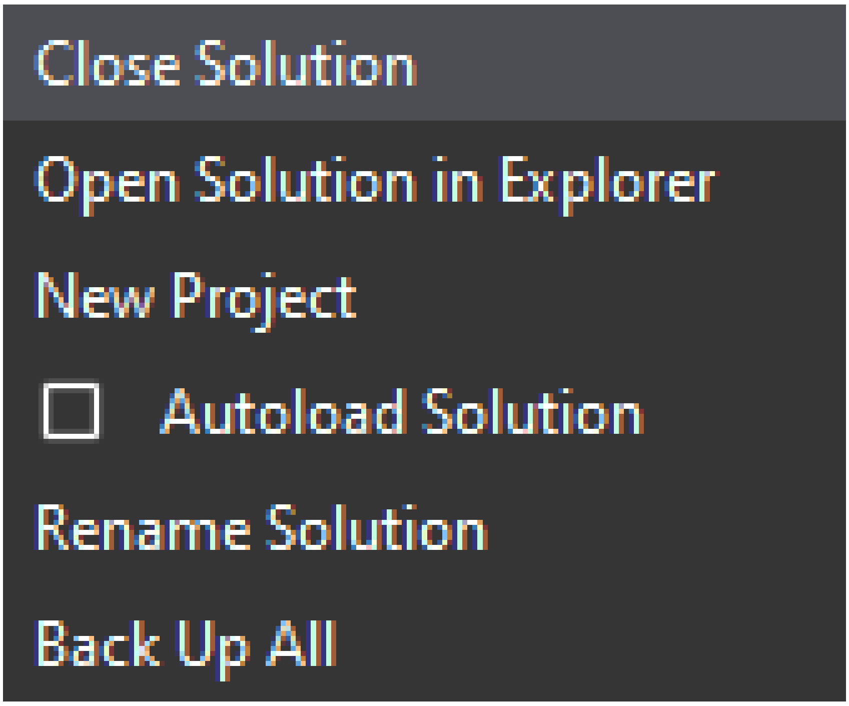
Then, input a new name for the solution and press `Enter` to confirm.



4.2.2.5. Close the Solution

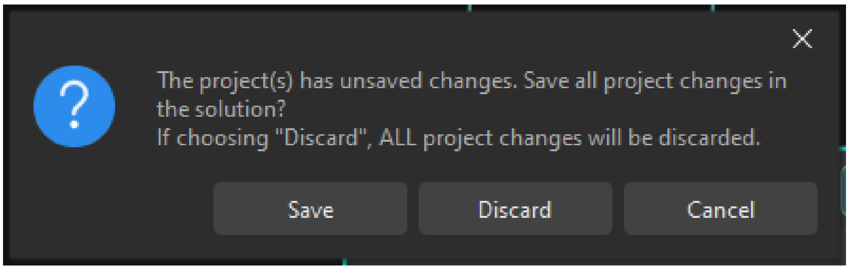
There are two ways to close a solution:

- Right-click the solution in the Project List and select **Close Solution**, as shown below.



- Select File › Close Solution in the menu bar.

If there are unsaved changes in the project, a pop-up window as shown below will appear. You can select one of the options according to your need.



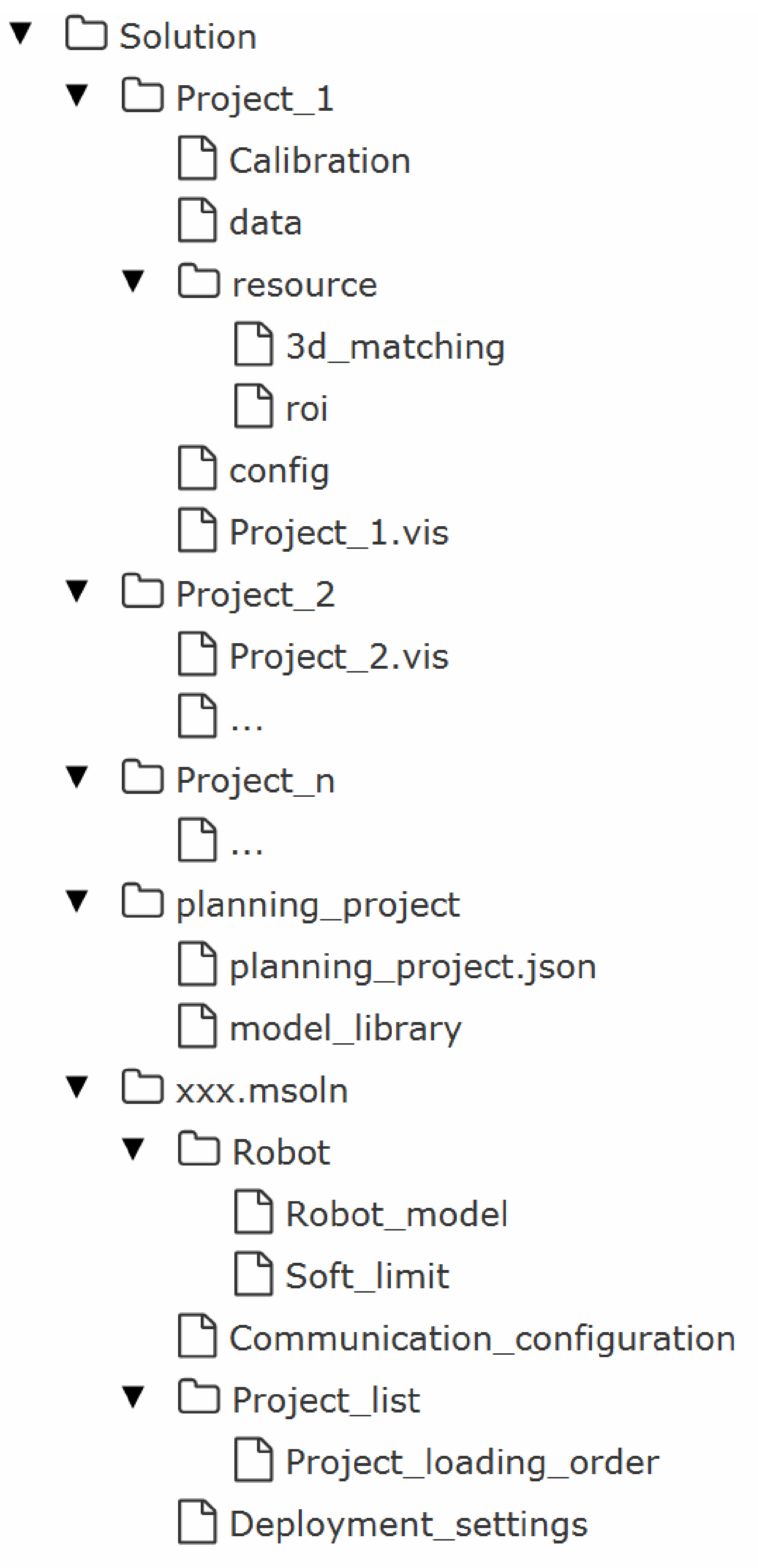
The project(s) has unsaved changes. Save all project changes in the solution?
If choosing "Discard", ALL project changes will be discarded.

Once you close the solution, all projects in the solution will be closed as well.

4.2.2.6. Understand Solution Structure

This section introduces the structure of a solution, which includes the configuration and data files for functions required by a vision application, such as robot and communication, vision processing, and path planning.

The following figure shows the structure of a solution.



```
▼ Solution
    ▼ Project_1
        Calibration
        data
        ▼ resource
            3d_matching
            roi
        config
        Project_1.vis
    ▼ Project_2
        Project_2.vis
        ...
    ▼ Project_n
        ...
    ▼ planning_project
        planning_project.json
        model_library
    ▼ xxx.msoln
        ▼ Robot
            Robot_model
            Soft_limit
        Communication_configuration
        ▼ Project_list
            Project_loading_order
        Deployment_settings
```

4.2.3. Quick Guide for Projects

Projects refer to Mech-Vision projects. A project must belong to a solution, or else it cannot be used. In addition, a solution may include multiple projects.

This chapter introduces you to basic procedures of using projects.

4.2.3.1. Create and Save Project

This section provides information on creating a new Mech-Vision project, saving the project, and the naming convention of project names.

Create New Projects

Create from Solution Library

Please refer to Create a new solution or project from Solution Library.

Create a New Blank Project

There are three ways to create a new blank project:

- After creating a new solution, create a new blank project in it.
- Select File › New Project in the menu bar to create a new blank project.
- Press `Ctrl` + `N` on the keyboard to create a new blank project.

> You can only create a new project after creating a new solution.

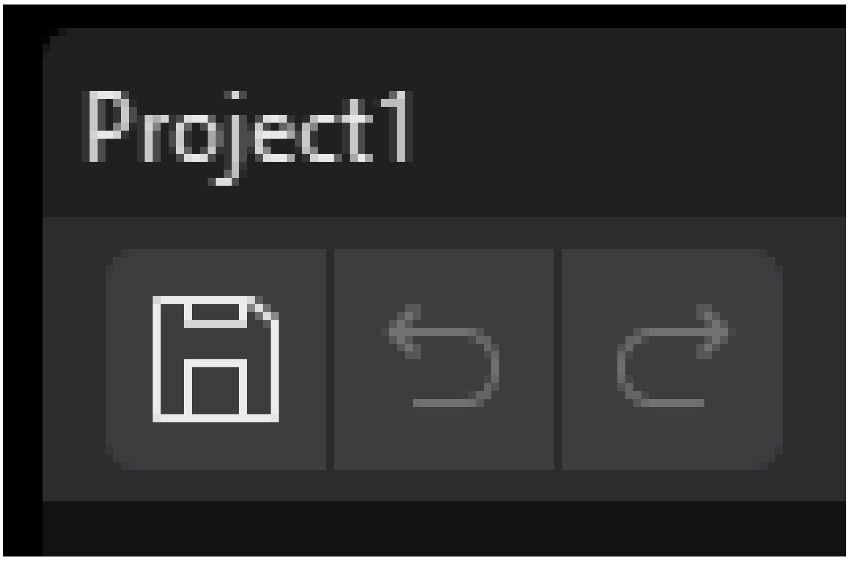
Save Projects

You can save a single project in a solution or save all projects in the solution.

Save a Single Project

There are two ways to save a single project in a solution:

- Select File › Save Project in the menu bar to save a single project.
- Click 💾 on the left end of the project toolbar or press `Ctrl` + `S` on the keyboard to save a single project.



Save all Projects in the Solution

If you want to save all projects in a solution, select File › Save Solution in the menu bar to save all projects in the solution.

Naming Convention of Project Names

The recommended naming convention of Mech-Vision projects is as shown below.

Components in the name	Description
Vis	To distinguish Mech-Vision projects from Mech-Viz projects (compulsory)
Object Name + Application Scenario	For example: CartonDepalletizing/DiscBrakeMachineTending/ElectricMotorAssembly/SpongeGluing/ParcelPicking
No. or the position of the Camera	Camera No.: 1#CrackDetection1#HeatTreating1# Camera Position: LeftCamera (optional)
Specific or special function	Specific function: Category1WorkpieceRecognition/ReflectiveWorkpieceRecognition/BinDetection/ModelMaking Special function: PointCloudStitching/ImageStitchingFromTwoCameras (optional)

The table below provides some example names.

Scenario	Example Name
Single Robot Cell + Single Camera	Vis-CartonDepalletizing
	Vis-DiscBrakeMachineTending
	Vis-ElectricMotorAssembly
	Vis-SpongeGluing
	Vis-ParcelPicking
Single Robot Cell + Multiple Cameras	Vis-WorkpieceMachineTending-LeftCamera
Multiple Robot Cells + Multiple Cameras	Vis-CrankshaftMachineTending-CrackDetection1#LeftCamera
Multiple Robot Cells + Multiple Cameras + Function	Vis-CrankshaftMachineTending-CrackDetection1#LeftCamera-PointCloudStitching
One camera in the robot cell is used to enable multiple functions	Vis-RzeppaJointMachineTending-5#LeftCamera-Category1WorkpieceRecognition
	Vis-RzeppaJointMachineTending-5#LeftCamera-Category2WorkpieceRecognition
	Vis-RzeppaJointMachineTending-5#LeftCamera-BinDetection
	Vis-RzeppaJointMachineTending-5#LeftCamera-ModelMaking
	Vis-RzeppaJointMachineTending-5#RightCamera-Category1WorkpieceRecognition
	Vis-RzeppaJointMachineTending-5#RightCamera-Category2WorkpieceRecognition

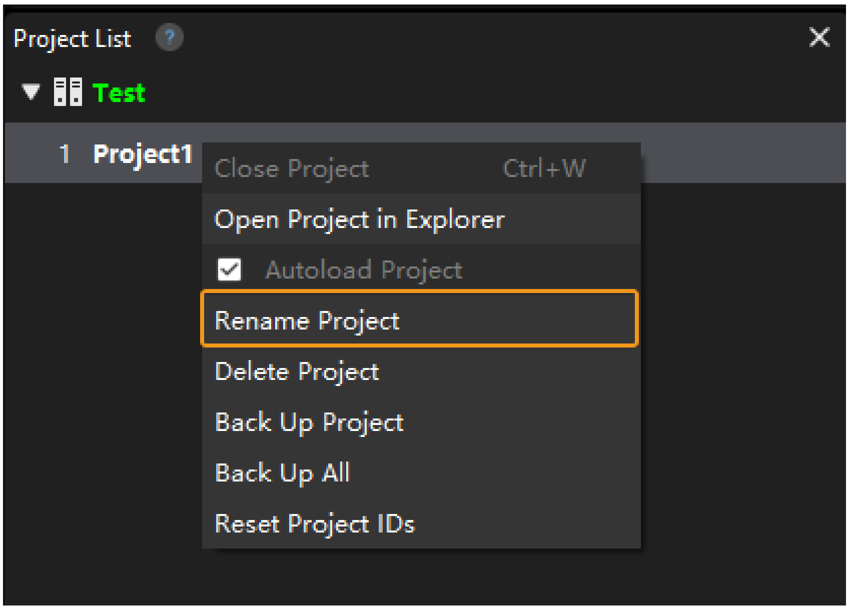
When using English for naming, note that:

- Use abbreviations instead of full names if possible.
- Try not to use *pinyin* spelling.
- Use camel case in word combinations.

4.2.3.2. Rename Projects

Follow the instruction below to rename a project.

Right-click the project in the project list, and select **[Rename Project]**. Input a new project name and press `Enter` to confirm.



After you complete the above operation, the project folder and the VIS file in the folder will be renamed as well.

4.2.3.3. Open Project

This section introduces how to open a Mech-Vision project.

Open an Existing Project

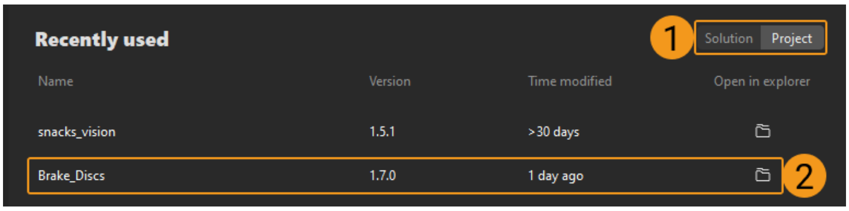
There are three ways to open an existing project:

- Click **[Open project]** on the Welcome interface, and select the project folder in the pop-up window. You do NOT need to open the folder and select the VIS file.
- Click File › Open Project (Shortcut `Ctrl` + `O`) in the menu bar of the main interface, and select the project folder in the pop-up window.
- Double-click the VIS file in the project folder to open Mech-Vision and autoload the project.

Open a Recently Opened Project

There are two ways to open a recently opened project:

- Click **Project** on the Solution/Project button in the Welcome interface, and double-click the project you want to open.



Name	Version	Time modified	Open in explorer
snacks_vision	1.5.1	>30 days	🗀
Brake_Discs	1.7.0	1 day ago	🗀

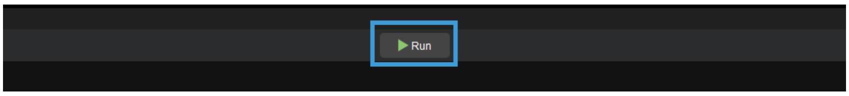
- Click File › Open Recent… in the menu bar of the main interface, and then select the project in the context menu to open.

- If you are using Mech-Vision 1.4.0 or later versions to open a project created in the previous versions, the original JSON project file will be saved as a VIS file, and the backup file will be saved as a BAK file.

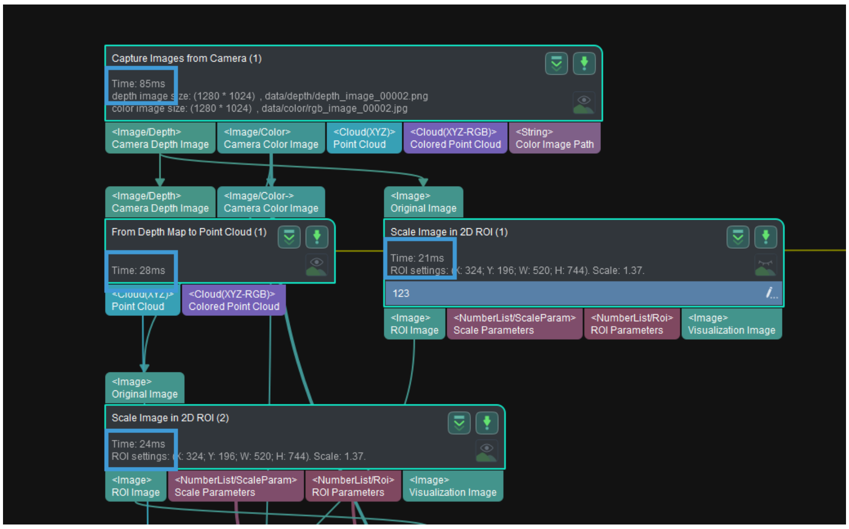
- You can click File › Save Project To JSON in the menu bar to save the project as a JSON file.

4.2.3.4. Run Project and Debug

Click [Run] in the toolbar or use the shortcut `Ctrl` + `R` to run the current Mech-Vision project, as shown below.



When the project's running result does not meet expectations, the project needs to be debugged. First, it is necessary to locate the Step(s) of error by running the Steps one by one. The parameters of Step(s) whose result does not meet expectations should be tuned. For each executed Step, the execution time of the Step is displayed in the lower-left corner of the Step box, as shown below.

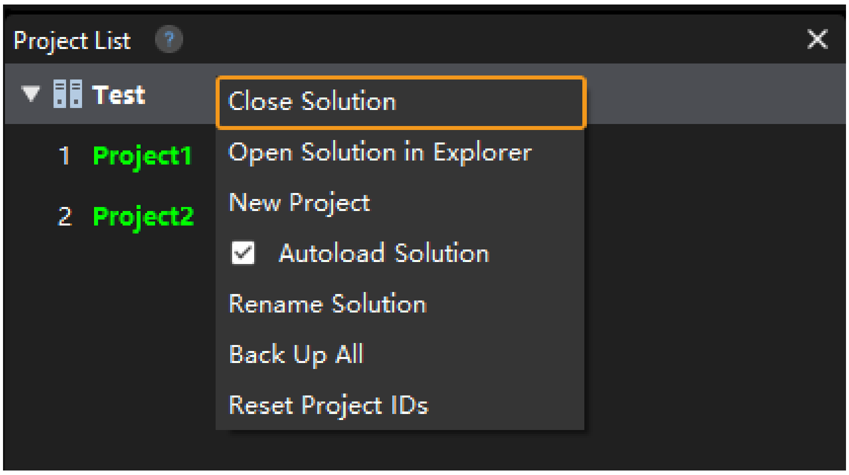

Capture Images from Camera (1)
Time: 85ms
depth image size: (1280 * 1024) , data/depth/depth_image_00002.png
color image size: (1280 * 1024) , data/color/rgb_image_00002.jpg
<Image/Depth> Camera Depth Image
<Image/Color> Camera Color Image
<Cloud(XYZ)> Point Cloud
<Cloud(XYZ-RGB)> Colored Point Cloud
<String> Color Image Path
From Depth Map to Point Cloud (1)
Time: 28ms
Scale Image in 2D ROI (1)
Time: 21ms
ROI settings: X: 324; Y: 196; W: 520; H: 744). Scale: 1.37.
123
<Image> Original Image
<Image> ROI Image
<NumberList/ScaleParam> Scale Parameters
<NumberList/Roi> ROI Parameters
<Image> Visualization Image
Scale Image in 2D ROI (2)
Time: 24ms
ROI settings: X: 324; Y: 196; W: 520; H: 744). Scale: 1.37.

For detailed instructions, please see Step Reference Guide.

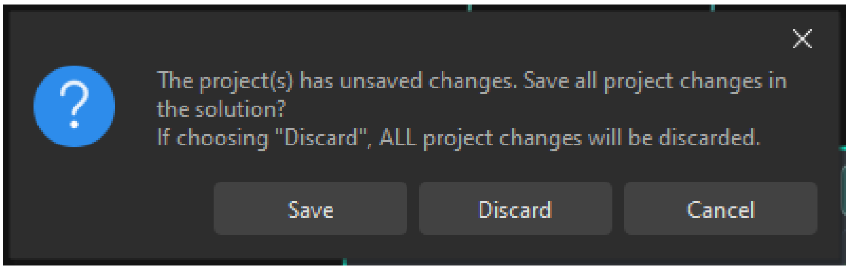
4.2.3.5. Close Project

You cannot close a single project that belongs to a solution. Instead, you can close the solution to close all projects in the solution. The detailed instructions are as follows.

Right-click the corresponding solution in the project list, and select **[Close Solution]** to close the solution and all projects within it, as shown below.



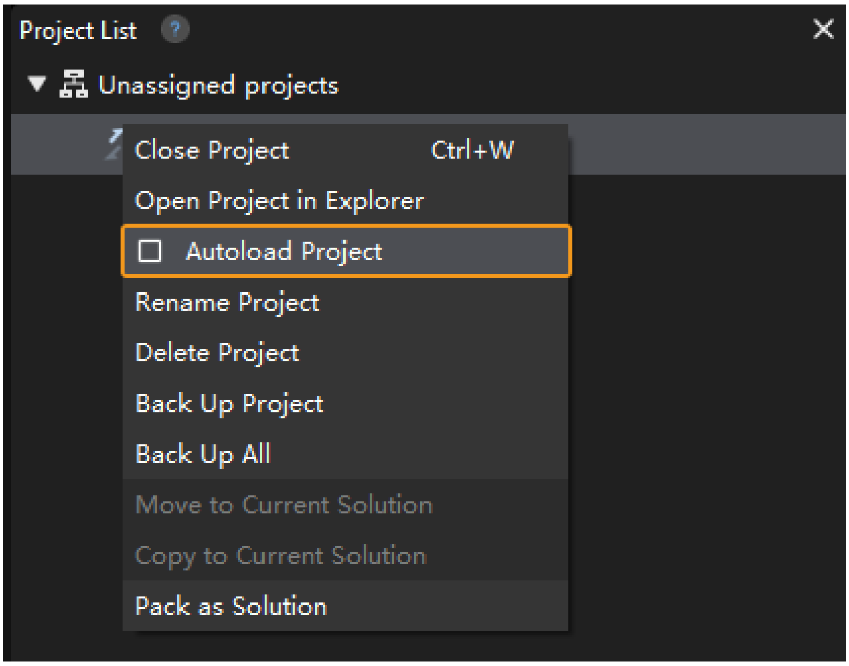
If there are unsaved changes in the project, a pop-up window as shown below will appear. You can select one of the options according to the actual requirement.


The project(s) has unsaved changes. Save all project changes in the solution?
If choosing "Discard", ALL project changes will be discarded.

4.2.3.6. Autoload Project

If you want a project to be opened automatically the next time you open Mech-Vision software, please do the following steps.

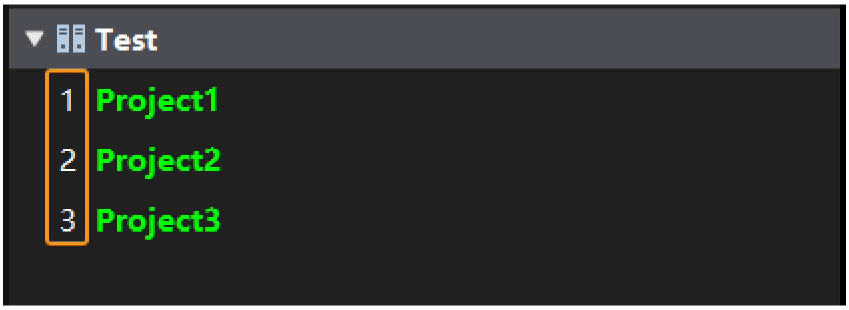
- When the project is not assigned to a solution, right-click the project in the project list and select **Autoload Project**, as shown below.



- When the project belongs to a solution, right-click the solution and select [vision-operation-guide:auto-load-solution:::autoload-solution], and all the projects belonging to the solution will be autoloaded.

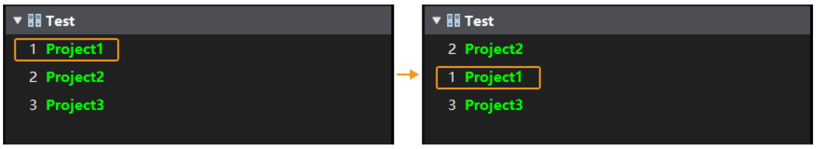
4.2.3.7. Adjust Project ID

When “Autoload Solution” is selected, the projects belonging to the solution will be autoloaded as well. Each project will be assigned a project ID, as shown below.



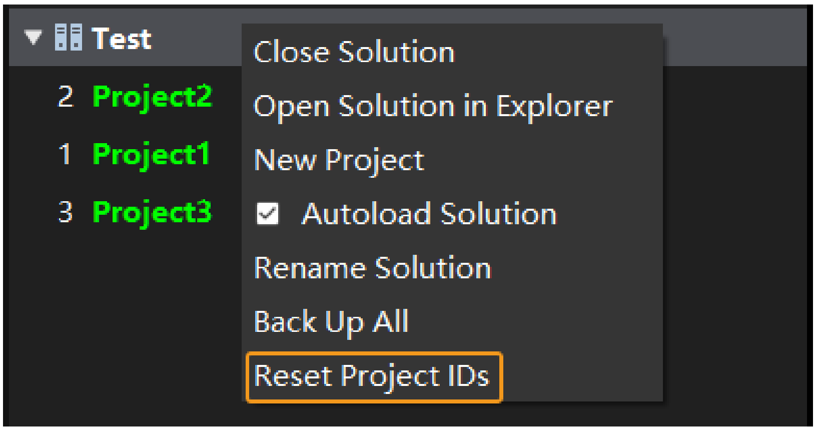
To adjust the project ID, follow these instructions:

1. Select and drag an project to adjust the display order of the project in the list. The result before and after adjusting is shown below.

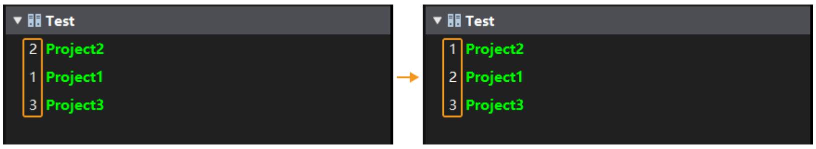


Here, only the display order of the project in the list is adjusted, and the project ID remains unchanged. Resetting project IDs is still required.

2. To reset the project ID, right-click the solution or one of the projects, and select [Reset Project IDs]. The disordered project IDs will be re-sorted in descending order, and the project corresponding to each ID will be changed.



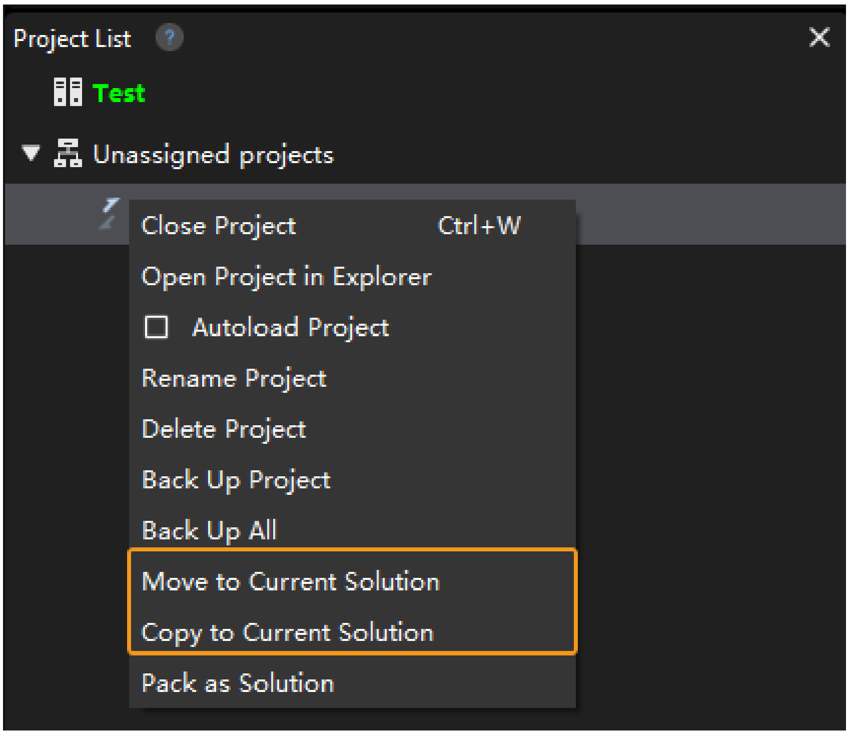
The result before and after resetting the project IDs is shown below.



4.2.3.8. Move an Existing Project to a Solution

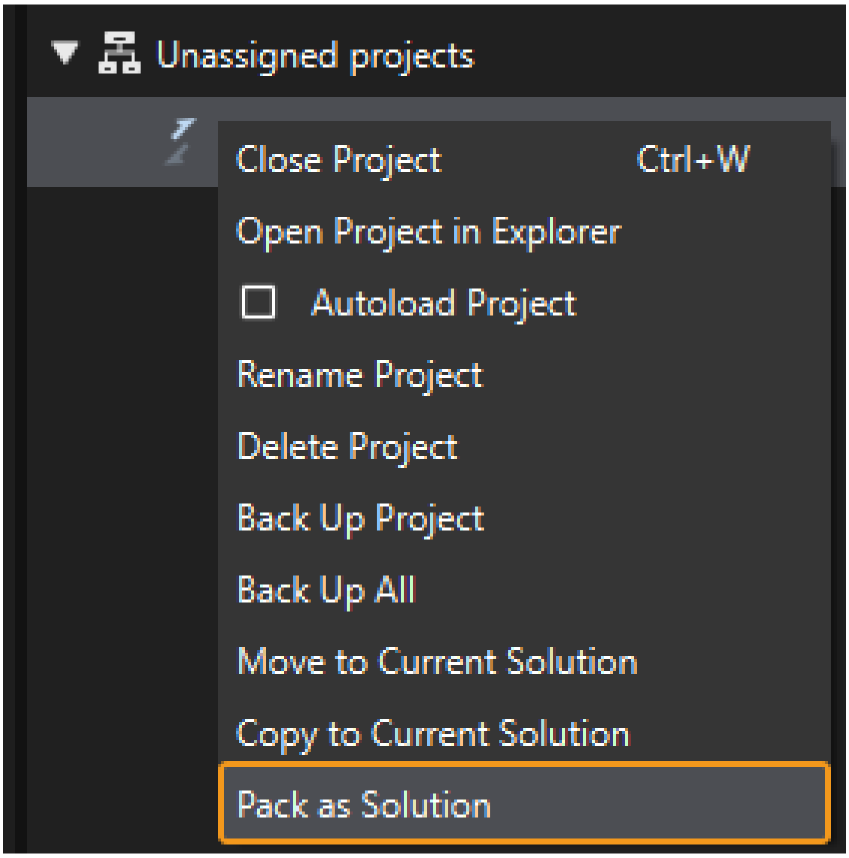
Follow the instruction below to move an existing project to a solution.

- After you create and save a new solution, right-click the unassigned project in the project list, and select [Move to Current Solution] or [Copy to Current Solution], as shown below.



After you moved the project to a solution, if the absolute path or relative path is used in the parameter of the Step, please check the path in the moved project and modify the wrong path to avoid errors.

- If you want to convert the project to a new solution, right-click an unassigned project in the project list, and select [Pack as solution], and then click [Yes] in the pop-up window. Then the project will be converted to a solution, and the parent folder of the project folder will be used as the solution folder.



▼ Unassigned projects

Close Project Ctrl+W

Open Project in Explorer

□ Autoload Project

Rename Project

Delete Project

Back Up Project

Back Up All

Move to Current Solution

Copy to Current Solution

Pack as Solution

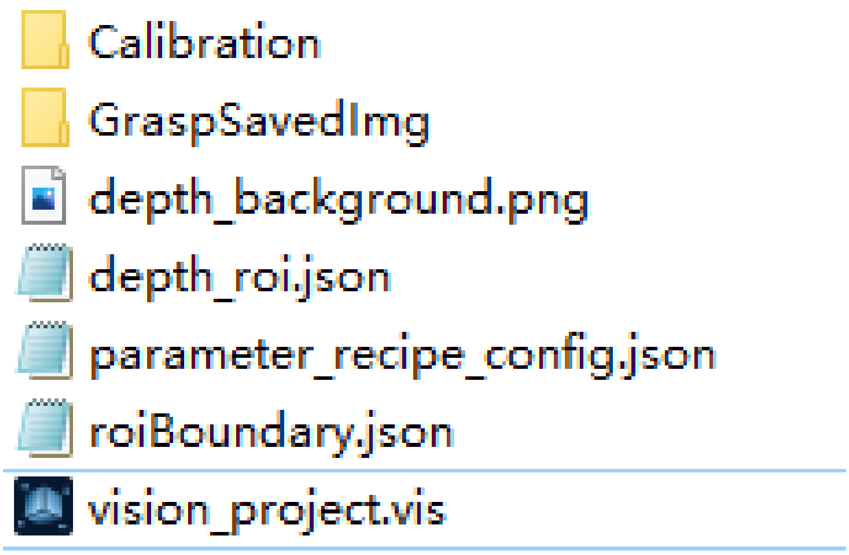
When converting the project to a new solution, the parent folder of the project folder cannot be the root folders, i.e., C drive, D drive, etc.

4.2.3.9. File Structure of a Mech-Vision Project

The figure below demonstrates a Mech-Vision project's file structure, which is mainly composed of the following files:

- An algorithm file (vision_project.vis in the figure): stores the algorithms.
- Camera parameter files (files in the folder Calibration in the figure): store the camera's intrinsic and extrinsic parameters and calibration data.
- Project configuration files (depth_background.png, depth_roi.json, roiBoundary.json, etc., in the figure): describe the setup of the working environment and add restrictions on the project.

The Mech-Vision project files will be automatically generated in the designated project path after the corresponding operations are performed, so there is no need to create these files manually.



Calibration
GraspSavedImg
depth_background.png
depth_roi.json
parameter_recipe_config.json
roiBoundary.json
vision_project.vis

4.2.4. Quick Guide for Robot and Communication Configuration

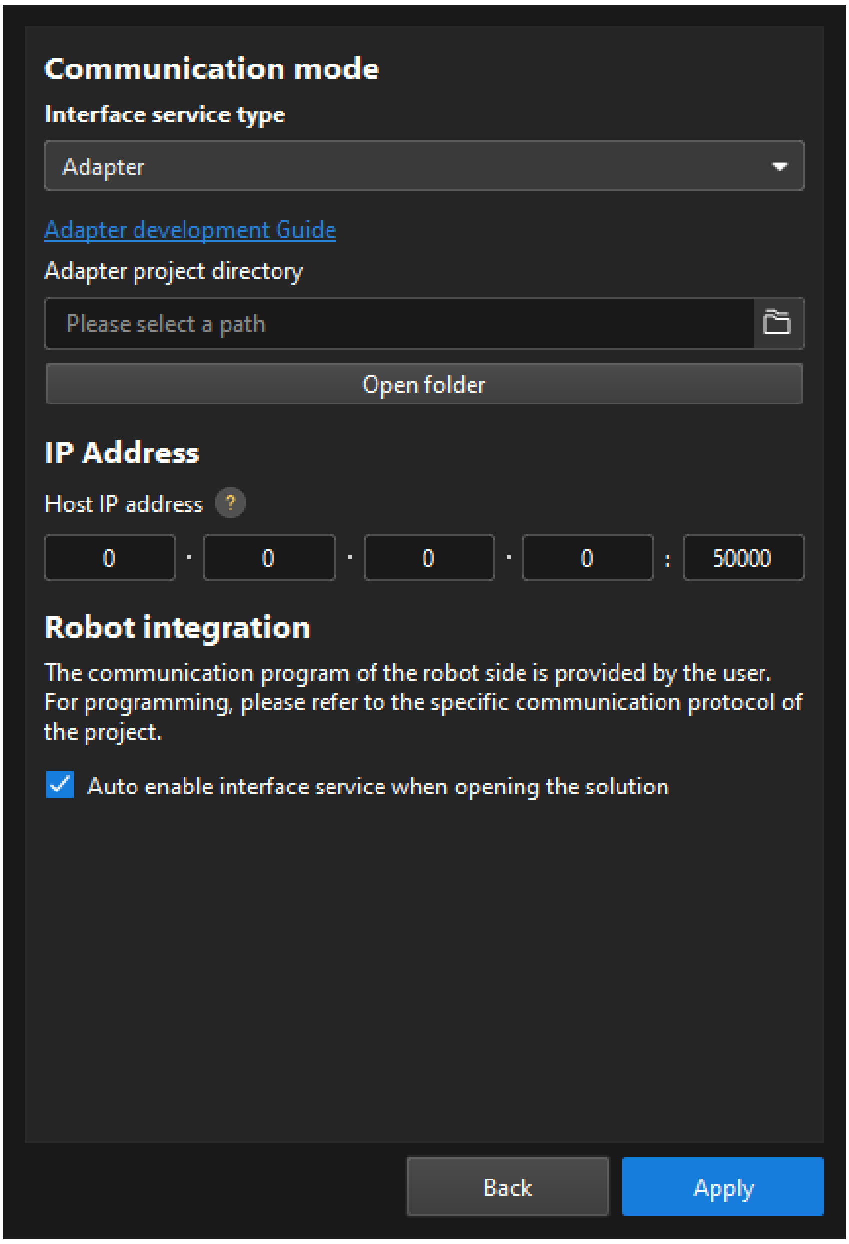
1. Operations related to robot communication configuration

   Before configuration, please confirm the communication mode and protocol used according to Communication Overview.

   - If the communication mode is “Standard Interface” or “Adapter”, please follow this section.
     - Select the Robot
     - Configure Communication:
       - Configure TCP/IP Communication
       - Configure Siemens PLC Snap 7 Communication
       - Configure PROFINET Communication
       - Configure EtherNet/IP Communication
       - Configure Modbus TCP Communication
       - Configure Adapter Communication
       - Configure Mitsubishi MC Communication
       - Configure UDP Communication
     - Start Interface Service
   - If the communication mode is Master-control communication, please refer to the section Master-control Communication.
2. view-tcp-command-set.pdf

4.2.4.1. Configure Adapter Communication

This section introduces how to configure the Adapter communication. Before configuration, please confirm that you have already selected the robot and entered the communication configuration interface.



Communication mode

Interface service type

Adapter

Adapter development Guide

Adapter project directory

Please select a path

Open folder

IP Address

Host IP address

0 . 0 . 0 . 0 : 50000

Robot integration

The communication program of the robot side is provided by the user. For programming, please refer to the specific communication protocol of the project.

☑ Auto enable interface service when opening the solution

1. On the **Communication mode** interface, set the "Interface service type" parameter to **Adapter**.
2. Click 🗀 and select the folder of the Adapter project.

   You need to write the Adapter program first. For details, refer to Adapter Programming Guide.

3. In the **IP Address** area, enter the IP address and port number of the host.

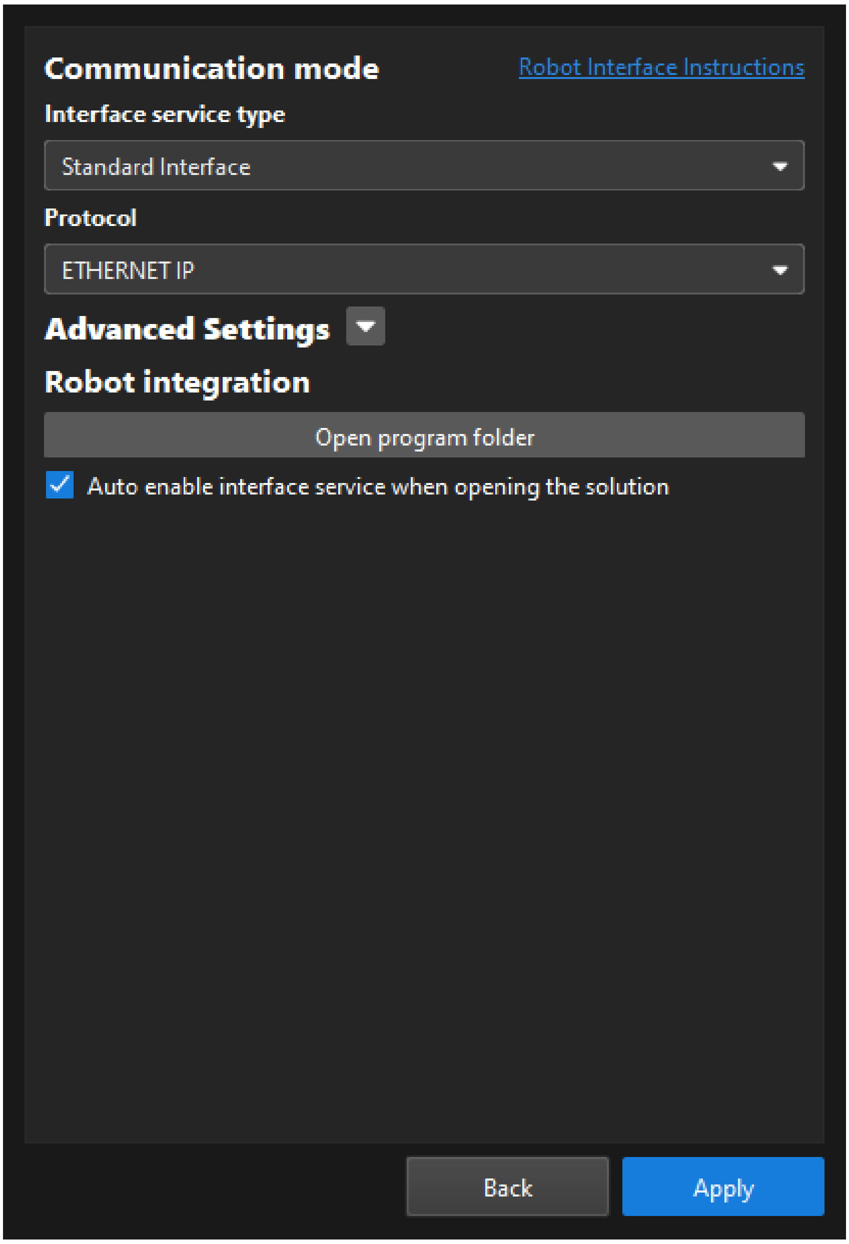
   - The host IP address and the robot IP address must be on the same subnet.
   - The port number should be set to 50000 or larger.

4. (Optional) Select **Auto enable interface service when opening the solution**. When the solution is opened, you do not need to start the interface service manually.
5. Click the **[Apply]** button.
6. Start the interface service.

Next, you need to write the robot-side communication program by referring to Standard Interface Development Manual and load it to the robot.

4.2.4.2. Configure EtherNet/IP Communication

This section introduces how to configure the EtherNet/IP communication. Before configuration, please confirm that you have already selected the robot and entered the communication configuration interface.



Communication mode

Robot Interface Instructions

Interface service type: Standard Interface

Protocol: ETHERNET IP

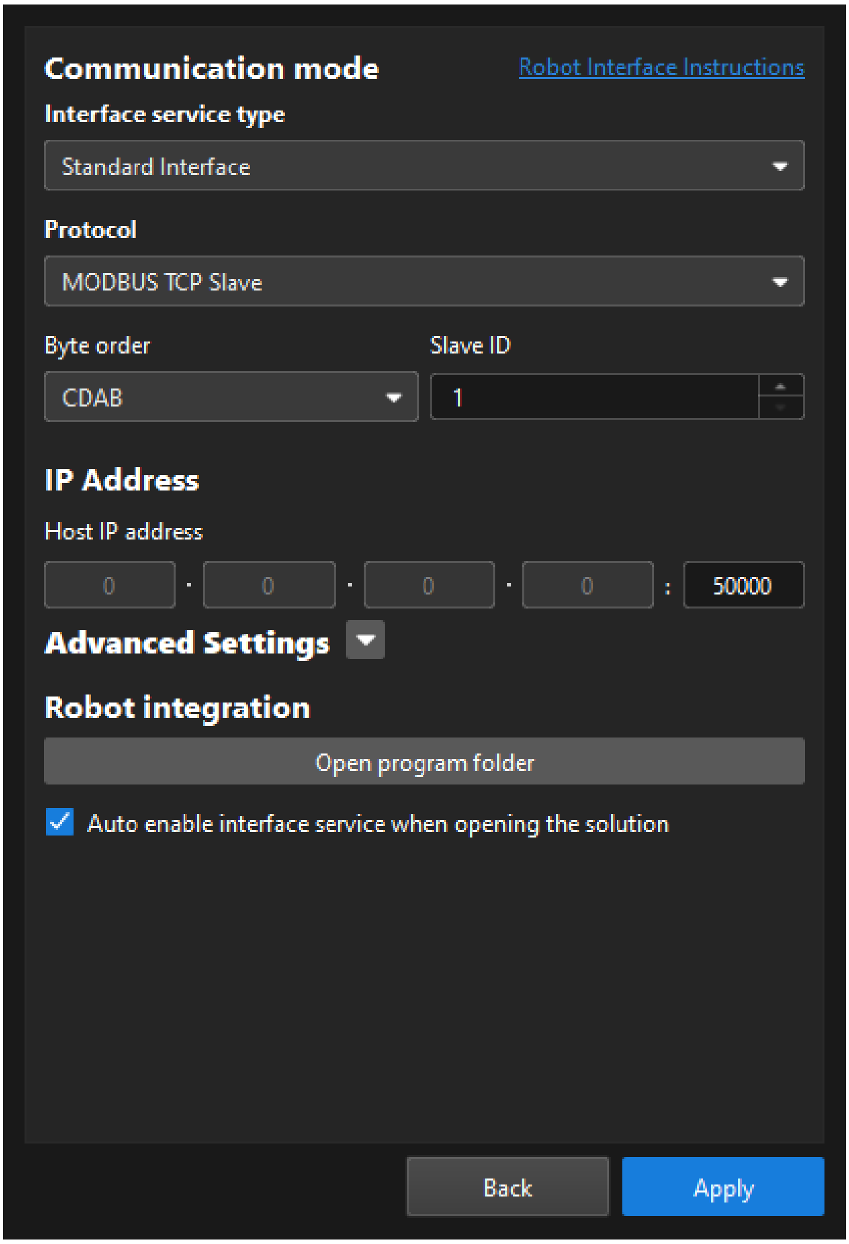
☑ Auto enable interface service when opening the solution

1. On the **Communication mode** interface, set the “Interface service type” parameter to **Standard Interface**, and set the “Protocol” parameter to **ETHERNET IP**.
2. (Optional) Select **Auto enable interface service when opening the solution**. When the solution is opened, you do not need to start the interface service manually.
3. Click the **[Apply]** button.
4. Start the interface service.

Next, you need to complete the remaining standard interface configuration according to the communication object.

4.2.4.3. Configure Modbus TCP Communication

This section introduces how to configure the Modbus TCP communication. Before configuration, please confirm that you have already selected the robot and entered the communication configuration interface.



Communication mode

Interface service type: Standard Interface

Protocol: MODBUS TCP Slave

Byte order: CDAB

Slave ID: 1

IP Address

Host IP address: 0 . 0 . 0 . 0 : 50000

Robot Interface Instructions
Advanced Settings
Robot integration
Open program folder
☑ Auto enable interface service when opening the solution
Back
Apply

1. On the **Communication mode** interface, complete the following configurations.

   a. Set the "Interface service type" parameter to **Standard Interface**, and set the "Protocol" parameter to **MODUBS TCP Slave**.

   b. Set the "Byte order" parameter. Set it based on the master device's floating point byte order. For standard big-endian data, select DCBA; for standard small-endian data, select ABCD.

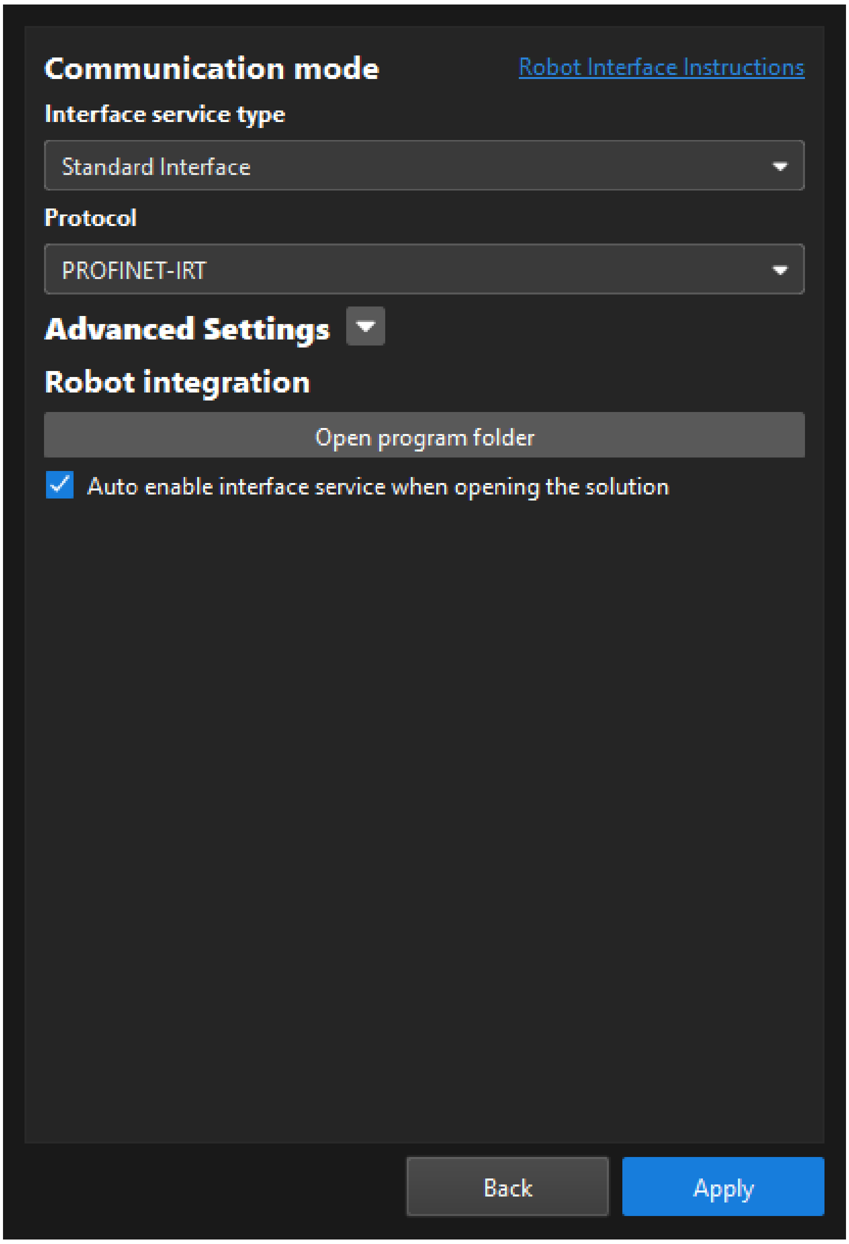
   c. Set Slave ID according to your needs.

2. In the **IP Address** area, enter the port number of the host (Modbus TCP slave). The port number should be set to 50000 or larger.
3. (Optional) Select **Auto enable interface service when opening the solution**. When the solution is opened, you do not need to start the interface service manually.
4. Click the **[Apply]** button.
5. Start the interface service.

Next, you need to complete the remaining standard interface configuration according to the communication object.

4.2.4.4. Configure PROFINET Communication

This section introduces how to configure the PROFINET communication. Before configuration, please confirm that you have already selected the robot and entered the communication configuration interface.



Communication mode

Robot Interface Instructions

Interface service type: Standard Interface

Protocol: PROFINET-IRT

Advanced Settings

Robot integration

Open program folder

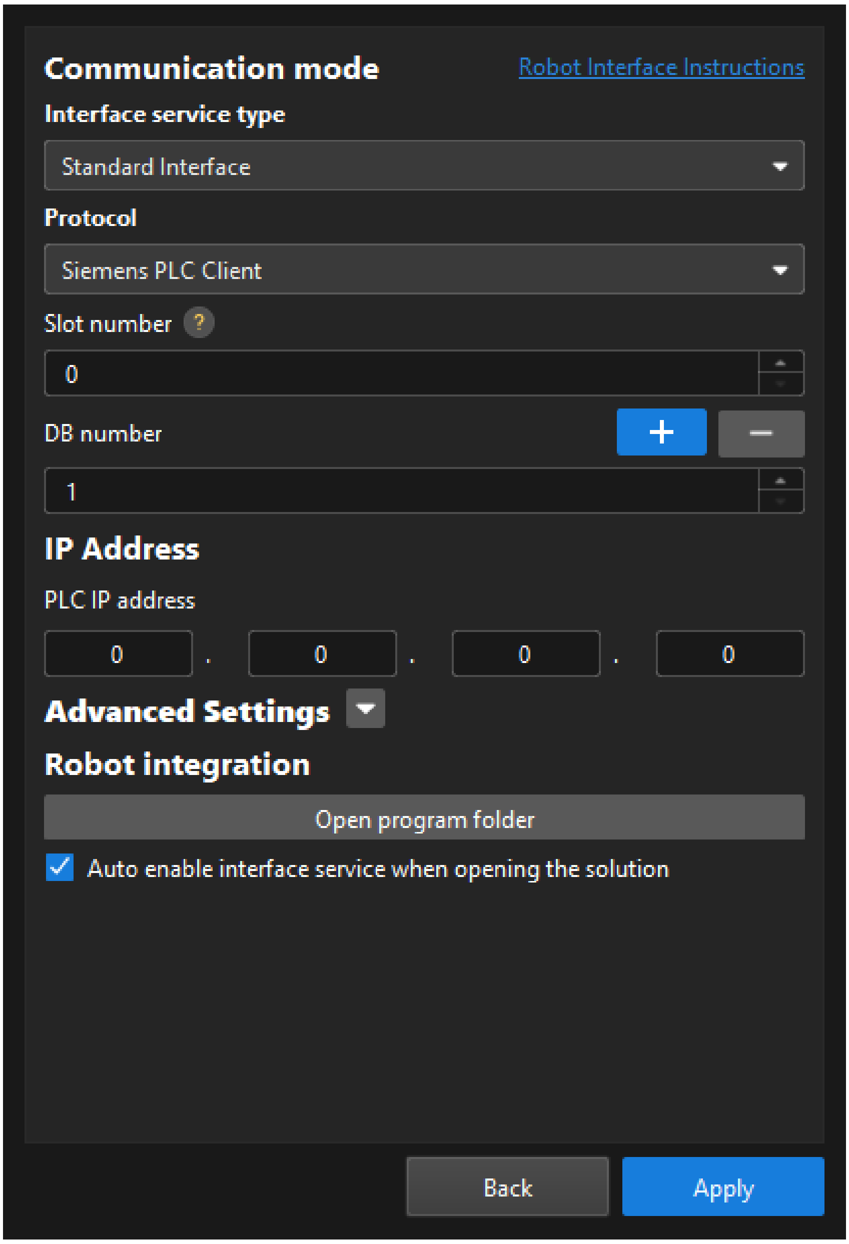
☑ Auto enable interface service when opening the solution

1. On the **Communication mode** interface, set the “Interface service type” parameter to **Standard Interface**, and set the “Protocol” parameter to **PROFINET-IRT**.
2. (Optional) Select **Auto enable interface service when opening the solution**. When the solution is opened, you do not need to start the interface service manually.
3. Click the **[Apply]** button.
4. Start the interface service.

Next, you need to complete the remaining standard interface configuration according to the communication object.

4.2.4.5. Configure Siemens PLC Snap 7 Communication

This section introduces how to configure the Siemens programmable logic controller (PLC) Snap 7 communication. Before configuration, please confirm that you have already selected the robot and entered the communication configuration interface.



Communication mode

Interface service type

Standard Interface

Protocol

Siemens PLC Client

Slot number: 0

DB number: 1

IP Address

PLC IP address: 0 . 0 . 0 . 0

Advanced Settings

Robot integration

☑ Auto enable interface service when opening the solution

1. On the **Communication mode** interface, complete the following configurations.
   a. Set the “Interface service type” parameter to **Standard Interface**, and set the “Protocol” parameter to **Siemens PLC Client**.
   b. Specify the “Slot number” parameter. Refer to the following table for instructions.

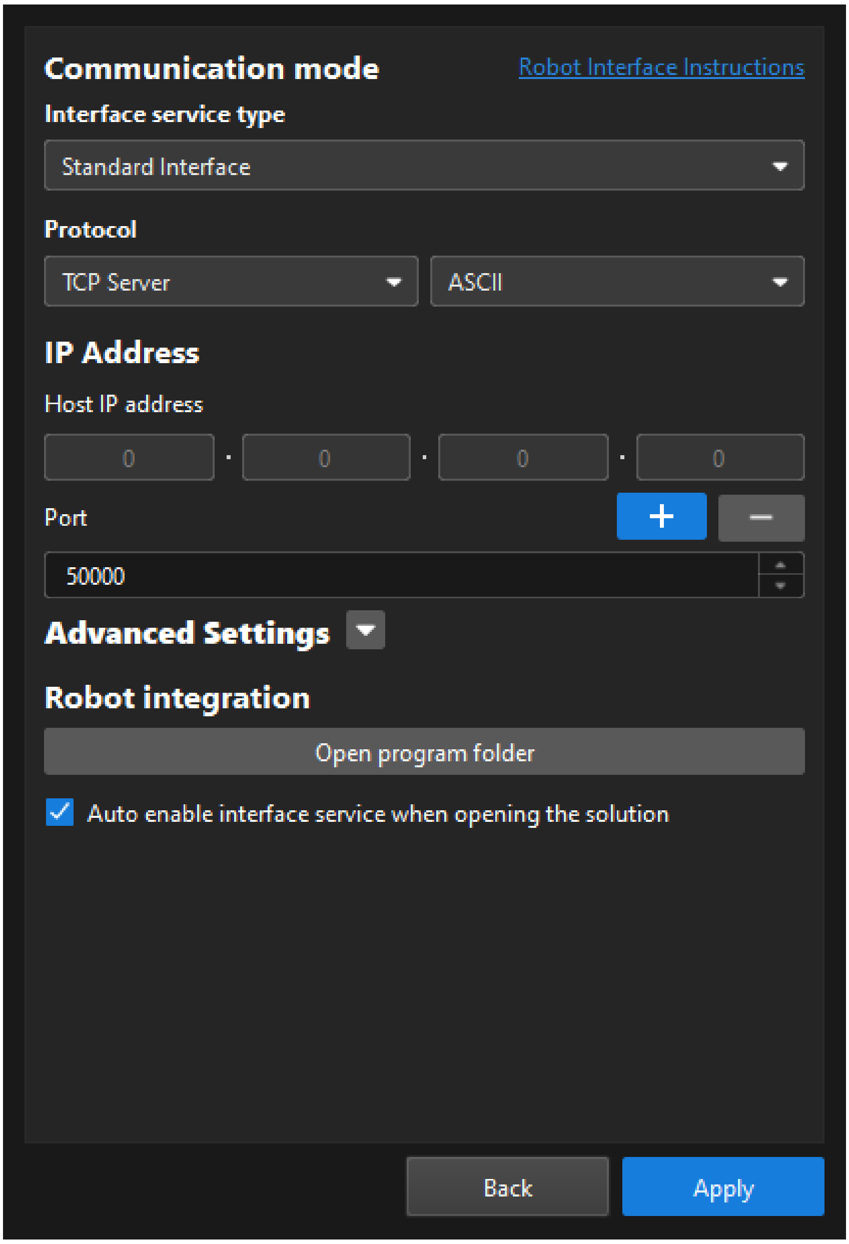
Siemens S7 Series	Slot Number
S7-300	2
S7-1200	0 or 1
S7-1500	0 or 1
Other models	0

   c. Specify the “DB number” parameter.
2. In the **IP Address** area, enter the PLC IP address. The PLC IP address and the IPC IP address must be on the same subnet.
3. (Optional) Select **Auto enable interface service when opening the solution**. When the solution is opened, you do not need to start the interface service manually.
4. Click the **[Apply]** button.
5. Start the interface service.

Next, you need to complete the remaining standard interface configuration according to the communication object.

4.2.4.6. Configure TCP/IP Communication

This guide introduces how to configure the TCP/IP communication. Before configuration, please confirm that you have already selected the robot and entered the communication configuration interface.


Communication mode

Robot Interface Instructions

Interface service type: Standard Interface

Protocol: TCP Server, ASCII

IP Address

Host IP address: 0 . 0 . 0 . 0

Port: 50000

Advanced Settings

Robot integration

Open program folder

☑ Auto enable interface service when opening the solution

1. On the **Communication mode** interface, set the “Interface service type” parameter to **Standard Interface**, set the “Protocol” parameter to **TCP Server**, and set the data transmission format according to the following table.

Robot Brand	Data Transmission Format
ABB	HEX (Little-endian)
FANUC	HEX (Big-endian)
KUKA	HEX (Little-endian)
Yaskawa	ASCII
Kawasaki	ASCII
UR	ASCII
TM	ASCII
ELITE	ASCII

Robot Brand	Data Transmission Format
JAKA	ASCII
ROKAE	ASCII
Others	You need to write the robot-side communication program. You can determine the data transmission format according to the support for HEX and ASCII.

2. In the **IP Address** area, enter the port number of the host.

   > ℹ The port number should be set to 50000 or larger.

3. (Optional) Select **Auto enable interface service when opening the solution**. When the solution is opened, you do not need to start the interface service manually.
4. Click the **[Apply]** button.
5. Start the interface service.

Next, you need to complete the following operations according to the selected robot.

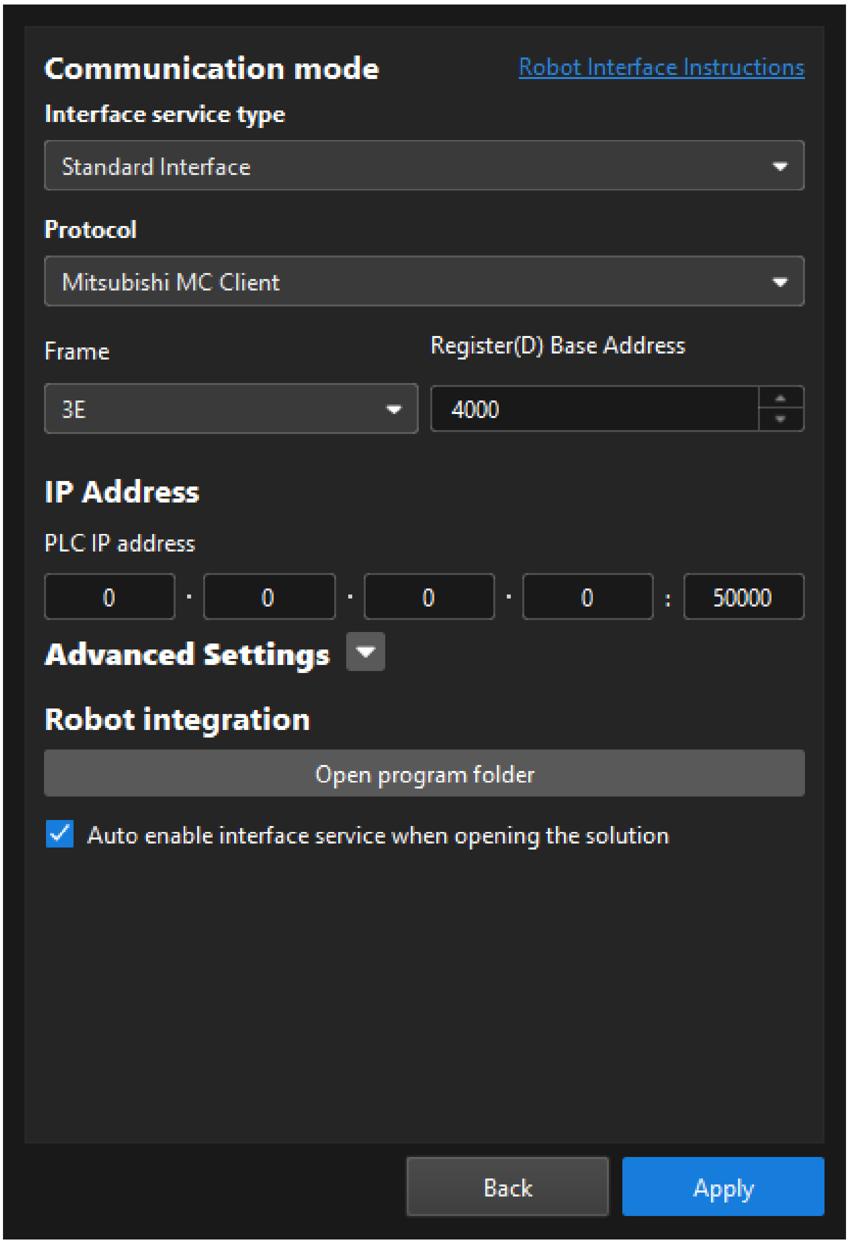
- If a listed robot is selected, complete the subsequent operations by referring to the corresponding documents in the following table.

Robot Brand	Reference Document
ABB	ABB Standard Interface Configuration
FANUC	FANUC Standard Interface Configuration
FANUC CRX	CRX Plugin Installation and Configuration
Kawasaki	Kawasaki Standard Interface Configuration
KUKA	KUKA Standard Interface Configuration
YASKAWA	YASKAWA Standard Interface Configuration
UR	URCap Plugin Installation and Configuration
TM	TM Standard Interface Configuration
LITE	ELITE Standard Interface Configuration
JAKA	JAKA Standard Interface Configuration

- If a listed robot other than the preceding robot brands is selected, you need to write the robot-side communication program by referring to Standard Interface Development Manual. For calibration, you can perform automatic calibration by referring to the Master-Control communication setup guide to the specific robot.
- If a custom robot is selected, you need to write the robot-side communication program by referring to Standard Interface Development Manual. For calibration, you can perform manual calibration.

4.2.4.7. Configure Mitsubishi MC Communication

This section introduces how to configure the Mitsubishi MC communication. Before configuration, please confirm that you have already selected the robot and entered the communication configuration interface.



Communication mode

Interface service type: Standard Interface

Protocol: Mitsubishi MC Client

Frame: 3E

Register(D) Base Address: 4000

IP Address

PLC IP address: 0 . 0 . 0 . 0 : 50000

☑ Auto enable interface service when opening the solution

1. On the **Communication mode** interface, complete the following configurations.
   a. Set the “Interface service type” parameter to **Standard Interface**, and set the “Protocol” parameter to **Mitsubishi MC Client**.
   b. In the **Frame** drop-down list, select 3E for most cases.
   c. Set the Register(D) Base Address.

      The struct types used for communication occupy a total of 728 D registers.

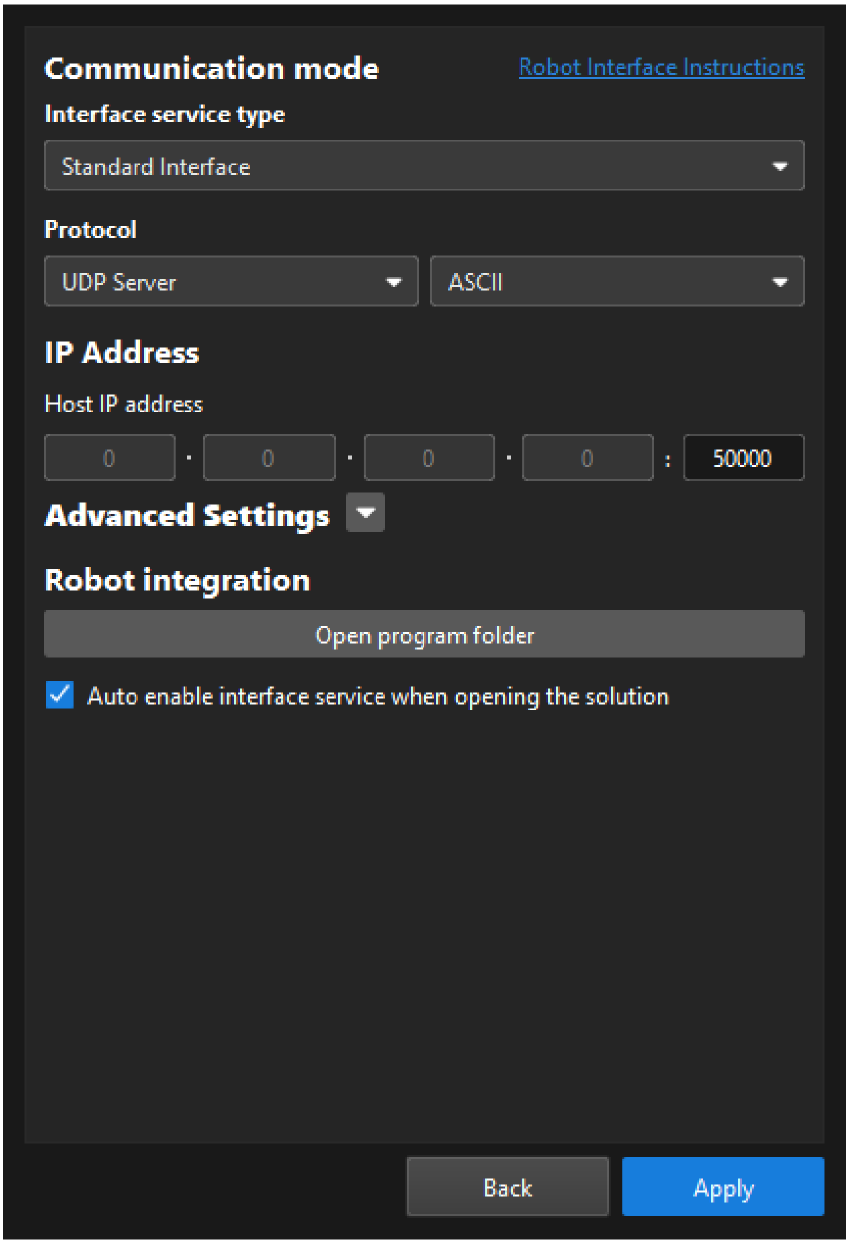
2. In the **IP Address** area, fill in the IP address and port number of the PLC. The port number should be set to 50000 or larger.
3. (Optional) Select **Auto enable interface service when opening the solution**. When the solution is opened, you do not need to start the interface service manually.
4. Click the **[Apply]** button.

5. Start the interface service.

Next, you need to complete the remaining standard interface configuration according to the communication object.

4.2.4.8. Configure UDP Communication

This section introduces how to configure the UDP communication. Before configuration, please confirm that you have already selected the robot and entered the communication configuration interface.



Communication mode

Interface service type: Standard Interface

Protocol: UDP Server, ASCII

IP Address

Host IP address: 0 . 0 . 0 . 0 : 50000

Advanced Settings

Robot integration

☑ Auto enable interface service when opening the solution

1. On the **Communication mode** interface, set the “Interface service type” parameter to **Standard Interface**, set the “Protocol” parameter to **UDP Server**, and set the data transmission format based on support for HEX and ASCII.
2. In the **IP Address** area, enter the port number of the host.

> The port number should be set to 50000 or larger.

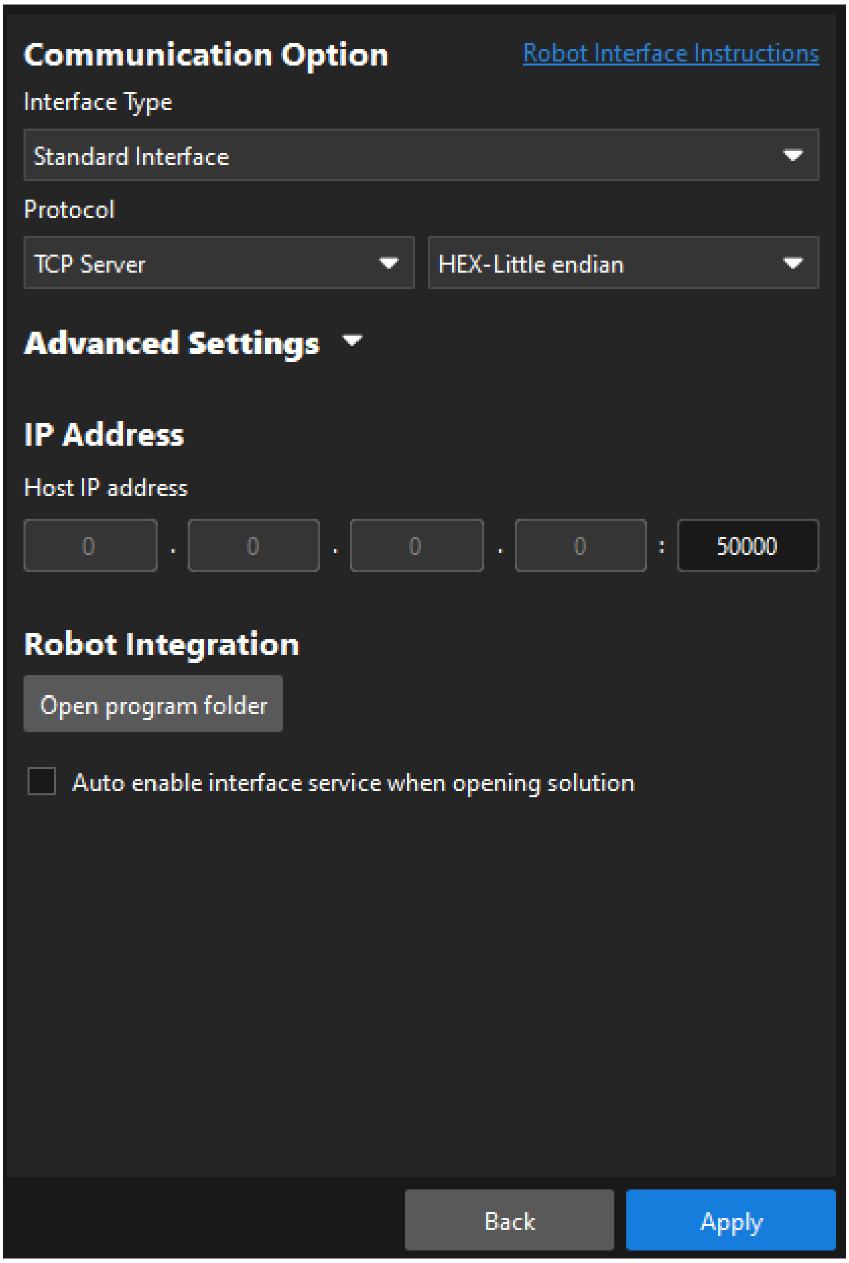
3. (Optional) Select **Auto enable interface service when opening the solution**. When the solution is opened, you do not need to start the interface service manually.
4. Click the **[Apply]** button.
5. Start the interface service.

Next, you need to complete the remaining standard interface configuration according to the communication object.

4.2.4.9. Obtain the Robot Programs

This section introduces how to obtain the robot programs.

After selecting a robot, in Communication Option window, click **[Open program folder]** to open the folder storing the robot communication program and example program.



Communication Option

Interface Type: Standard Interface

Protocol: TCP Server, HEX-Little endian

Advanced Settings

IP Address

Host IP address: 0 . 0 . 0 . 0 : 50000

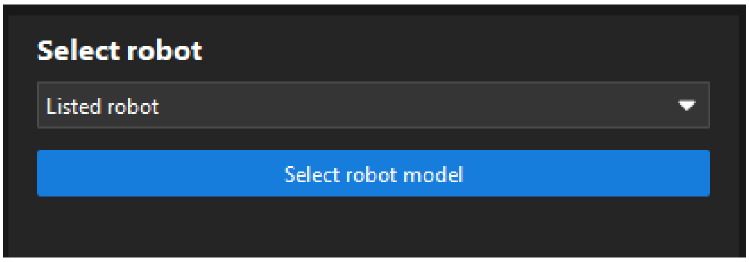
Robot Integration

□ Auto enable interface service when opening solution

4.2.4.10. Select the Robot

This section introduces how to select a robot.

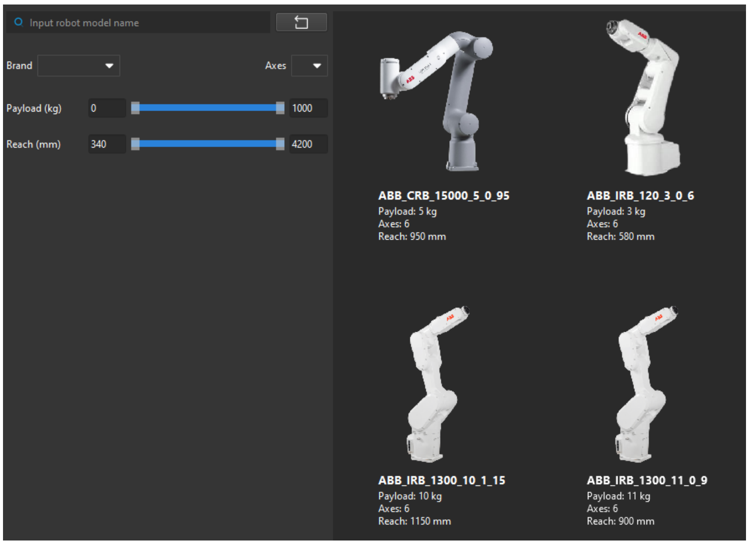
1. Click **[Robot Communication Configuration]** on the toolbar of Mech-Vision.
2. Select the robot.
   - Select a listed robot
     - Click the **Select robot** drop-down box, select **Listed robot**, and then click the **[Select robot model]** button.



Select robot

Listed robot

     - In the **Brand** drop-down box, select the robot brand. All robot models of this brand will be displayed on the right panel.



ABB_CRB_15000_5_0_95
Payload: 5 kg
Axes: 6
Reach: 950 mm

ABB_IRB_120_3_0_6
Payload: 3 kg
Axes: 6
Reach: 580 mm

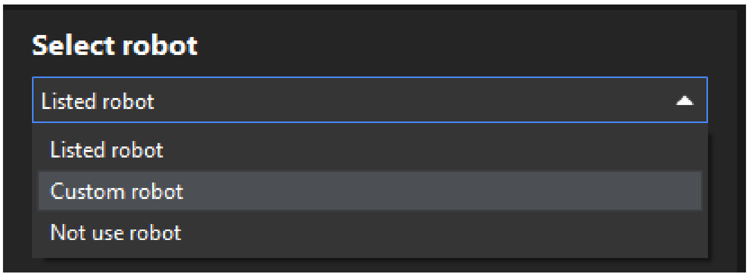
ABB_IRB_1300_10_1_15
Payload: 10 kg
Axes: 6
Reach: 1150 mm

ABB_IRB_1300_11_0_9
Payload: 11 kg
Axes: 6
Reach: 900 mm

> - Specify the **Axes**, **Payload**, and **Reach** parameters to narrow down the scope of robot models.
> - In the **search box**, enter the robot brand and model to filter the robot. Click the ⮌ button to clear the text in the search box.

> If you fail to find the required robot model in the robot library, close this window. Please import the robot model first and then perform this step again.

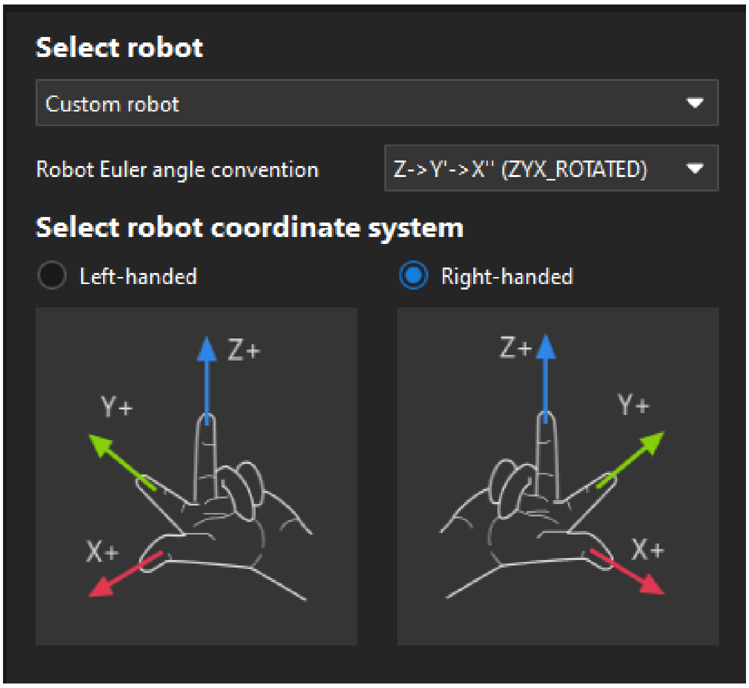
     - Hover the cursor over the robot model to select, and click the **[Select]** button when the frame turns grey-white to close the window.
     - Click the **[Next]** button to enter the Communication Option interface.
   - Select a custom robot (for example, truss robot)
     - Click the **Select robot** drop-down box and select **Custom robot**.



Select robot

Listed robot ▲

    - Specify the Robot Euler angle convention parameter.
    - Select the robot coordinate system.



Select robot

Custom robot ▼

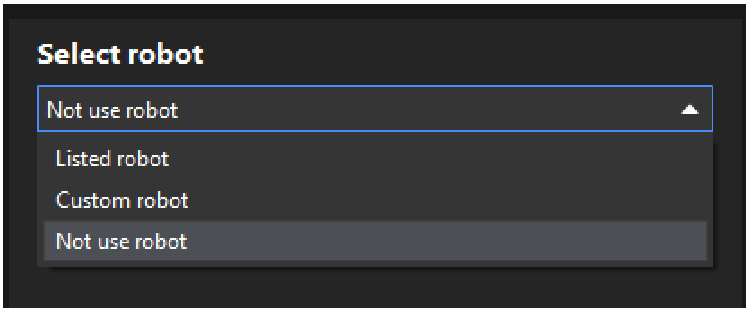
Robot Euler angle convention: Z->Y'->X'' (ZYX_ROTATED) ▼

Select robot coordinate system

○ Left-handed ● Right-handed

    - Click the **[Next]** button to enter the Communication Option interface.

  - Select not to use robot (in scenarios where no robots are used, such as measuring)
    - Click the **Select robot** drop-down box and select **Not use robot**.



Select robot

Not use robot ▲

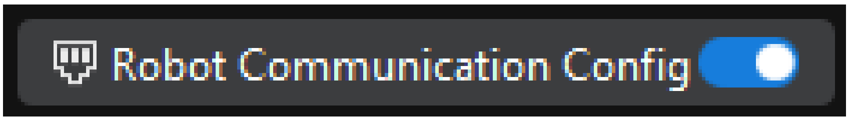
    - Click the **[Next]** button to enter the Adapter Communication Option interface.

If you select **Not use robot**, only the Adapter communication option can be used.

4.2.4.11. Start Interface Service

This section introduces how to start the interface service.

In the main window of Mech-Vision, click the “Robot Communication Configuration” switch on the toolbar. The interface service will be enabled as shown in the following figure.



If the interface service is started successfully, log “Start xxx interface” will be displayed under the Console tab of the Mech-Vision log panel.

If the interface service fails to be started, please check the **robot communication configuration**.

4.2.4.12. Guide for Robot Library Tools

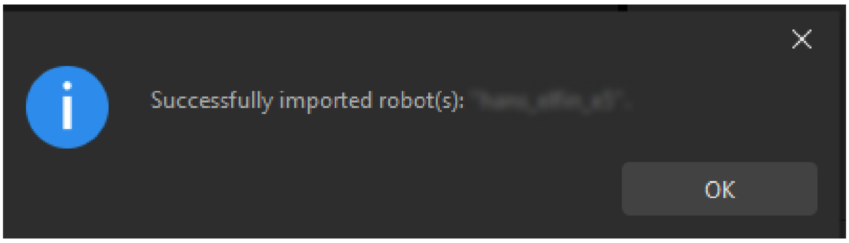
This section introduces procedures for using “Robot Library Tools”.

Import Robot

> You can export the robot model file (MROB file) from Mech-Viz menu bar > Tools > Robot Library Tools > Export Current Robot, or from Mech-Vision menu bar > Robot and Communication > Robot Library Tools > Export Current Robot.

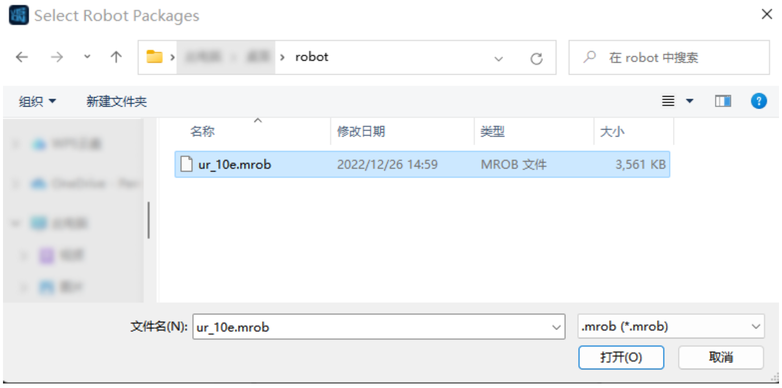
You can use either of the following methods to import the robot model.

- Method 1
  - Drag the robot model file in MROB format to the Mech-Vision software window.
  - A “Successfully imported robot(s)” message will appear. Click **[OK]** in the pop-up window.



Successfully imported robot(s): [redacted]

- Method 2
  - In Mech-Vision’s menu bar, select Toolkit › Robot Library Tools › Import Robot.
  - Select the MROB file in the Select Robot Packages window, and click **[Open]**.



Select Robot Packages

[redacted] › robot

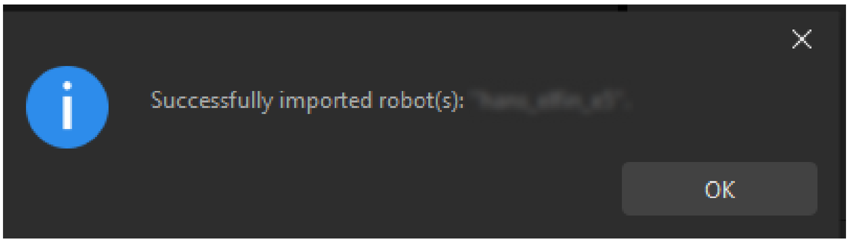
在 robot 中搜索

名称	修改日期	类型	大小
ur_10e.mrob	2022/12/26 14:59	MROB 文件	3,561 KB

文件名(N): ur_10e.mrob

.mrob (*.mrob)

- A “Successfully imported robot(s)” message will appear. Click [OK] in the pop-up window.

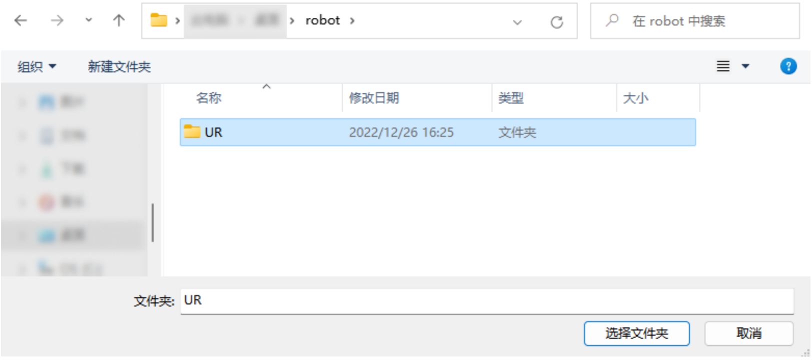


Successfully imported robot(s): [redacted].

Export Current Robot

Follow these steps to export the current robot model.

1. In Mech-Vision’s menu bar, select Toolkit › Robot Library Tools › Export Robot.
2. Select a folder to which the robot model file will be exported in the pop-up window, and click [Select Folder].



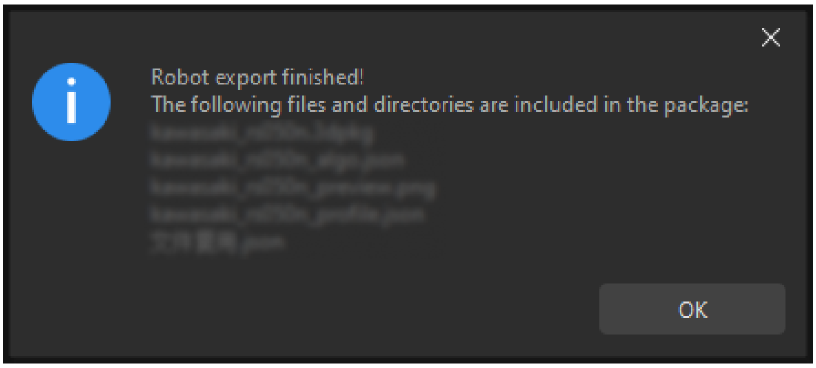
[redacted] › robot ›

在 robot 中搜索

名称	修改日期	类型	大小
UR	2022/12/26 16:25	文件夹	

文件夹: UR

3. A “Successfully exported robot(s)” message will appear. Click [OK] in the pop-up window.



Robot export finished!
The following files and directories are included in the package:
[redacted]

💡 If the robot model is exported successfully, a MROB file will appear in the specified folder.

4.2.5. Basic Procedures for Using Steps

This chapter introduces you to basic procedures of using Steps.

Search/Add/Delete Steps

Create/Delete Connections

Input/Output of Steps

Statuses of Steps

Run Steps and View Outputs

Introduction to Step Comments

A Procedure groups the Steps whose functions are closely related and achieve a specific purpose concertedly in the project.

Please check the sections below to learn about using Procedures.

Basic Features of Procedures

Modify Parameters of Procedures

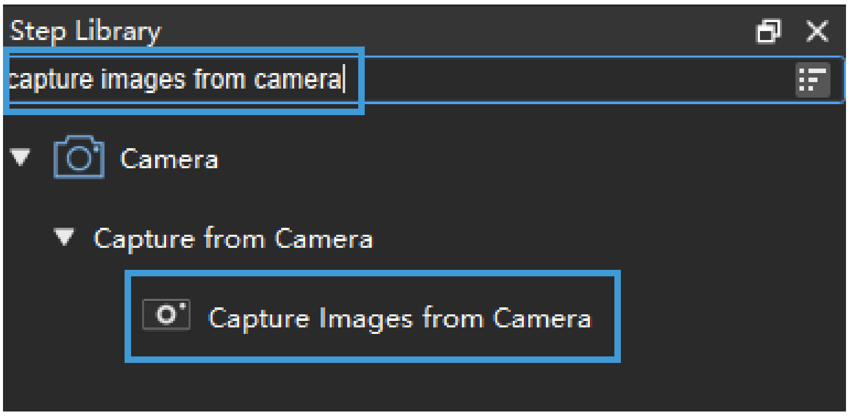
Create your own Procedure

4.2.5.1. Search/Add/Delete Steps

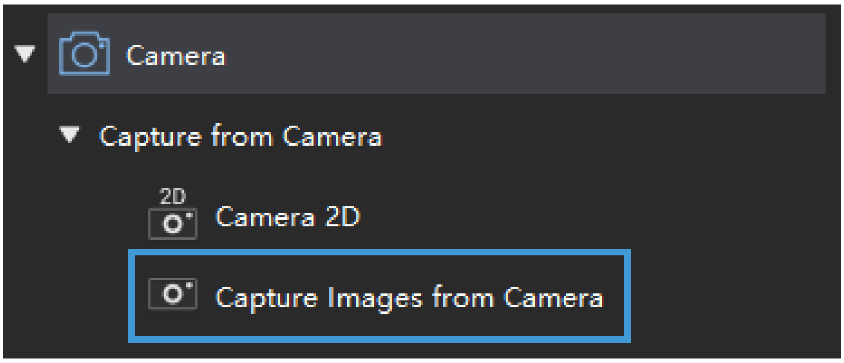
The "Capture images from camera" Step is used as an example.

Search for a Step

- Enter the name of the step directly in the search box of the step library (or enter a keyword in the name).

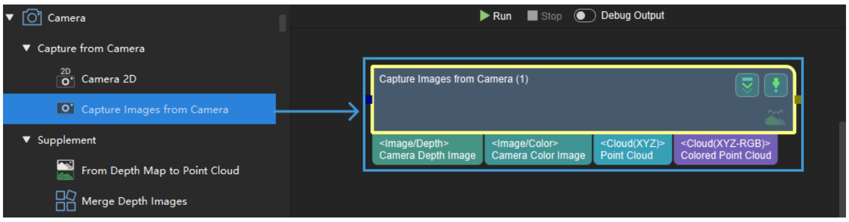


- Click the corresponding Step category to expand the hierarchical list.



Add a Step

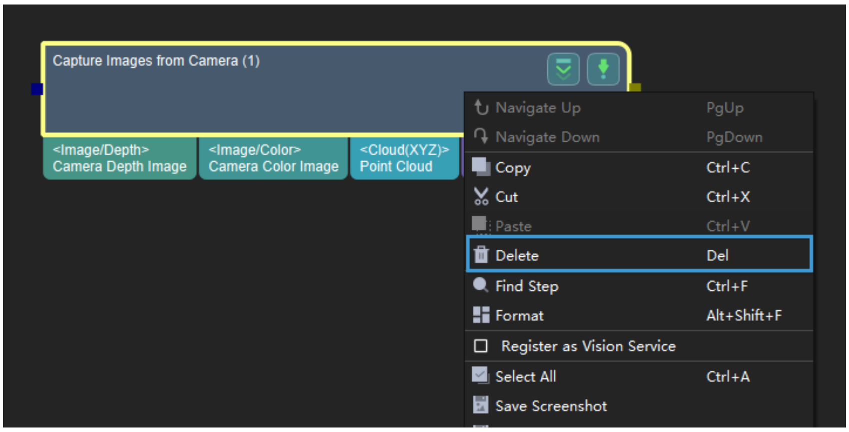
Press and hold the left mouse button to select a Step, drag it to any position in the graphical programming workspace and release it.


Camera
Capture from Camera
Camera 2D
Capture Images from Camera
Supplement
From Depth Map to Point Cloud
Merge Depth Images
Run
Stop
Debug Output
Capture Images from Camera (1)
<Image/Depth> Camera Depth Image
<Image/Color> Camera Color Image
<Cloud(XYZ)> Point Cloud
<Cloud(XYZ-RGB)> Colored Point Cloud

You can add Steps only after you have created a new project or opened an existing project.

Delete a Step

- Left-click a Step and press `Delete` on the keyboard.
- Right-click a Step and click **[Delete]** in the context menu.

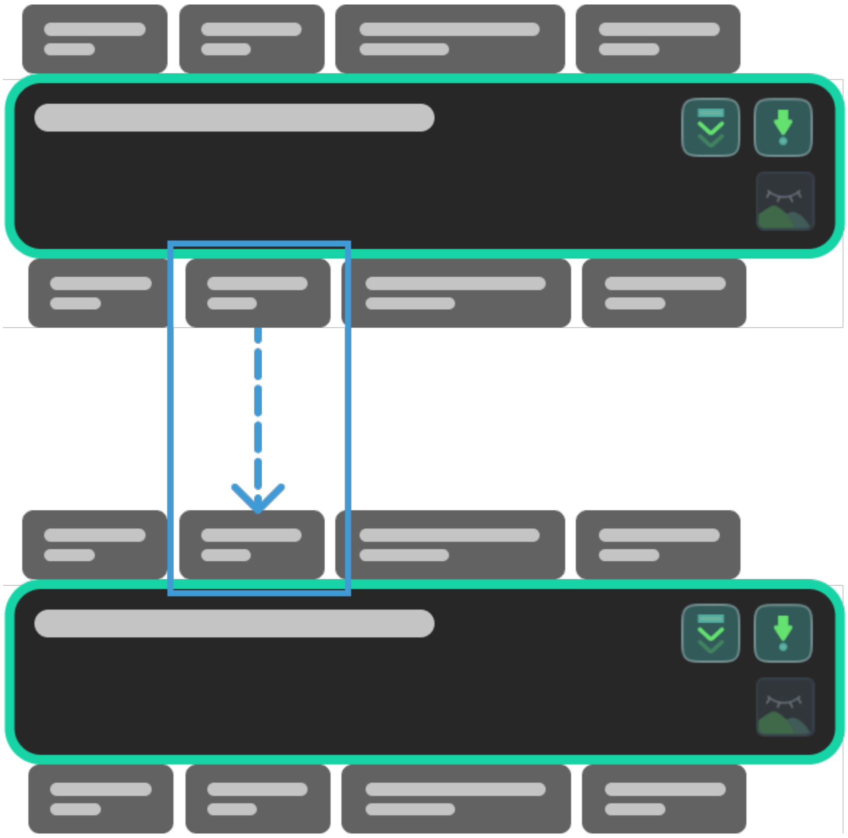


4.2.5.2. Create/Delete Connections

Create a Connection

Hover over the output port of a Step, and then hold down the left mouse button and move the mouse to the input port of a succeeding Step to create a connection.

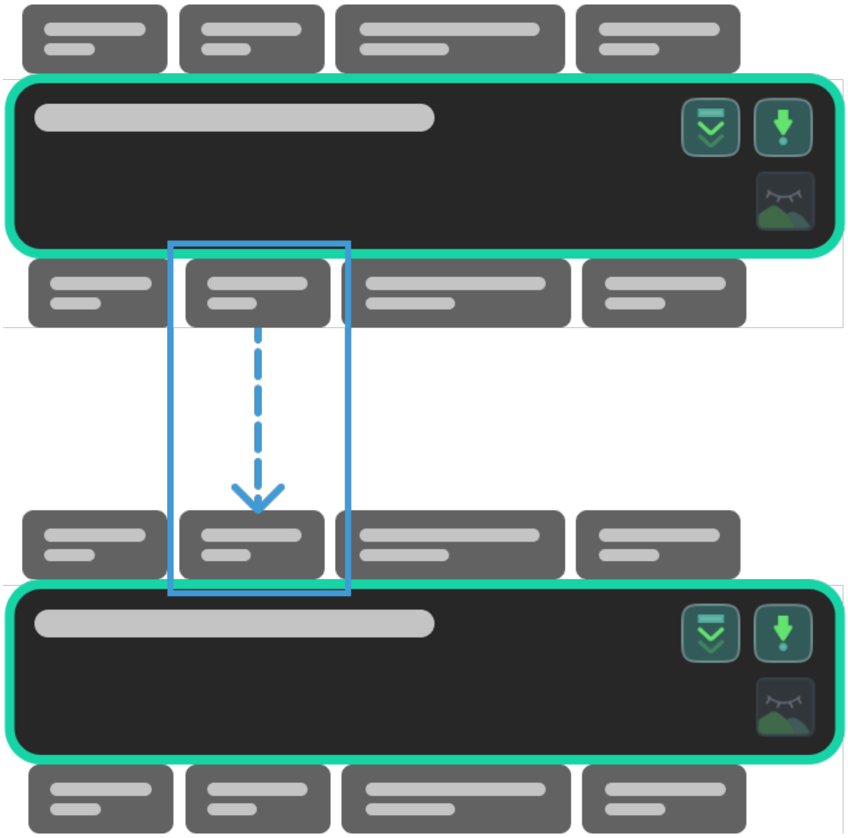
An output port can be connected to multiple input ports, but an input port can only be connected to one output port.



Connections can only be created among ports of the same data type. For example, the output port of **Cloud(XYZ-Normal)** can only be connected to the input port of **Cloud(XYZ-Normal)**. Ports of different data types can't be connected.

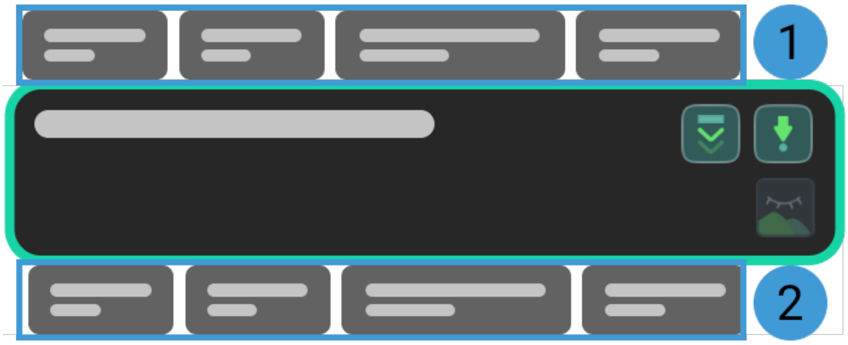
Delete a Connection

- After clicking the connection with the left mouse button, the connection will be highlighted. Press `Delete` on the keyboard to delete it.
- After right-clicking the connection, the connection will be highlighted, and a context menu bar will pop up. Then, click **[Delete]**.

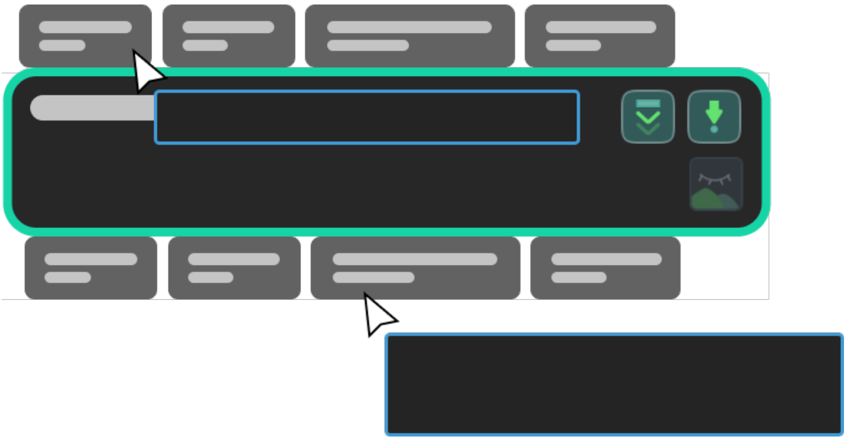


4.2.5.3. Input/Output of Steps

You can view the input/output data type of a Step by hovering the mouse over the corresponding port. The input/output ports of each step are on the top or at the bottom of the Step, as shown below, where 1 represents the input ports, and 2 the output ports.

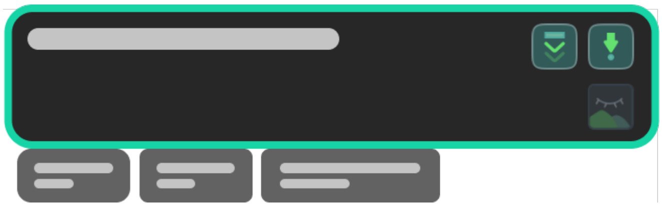
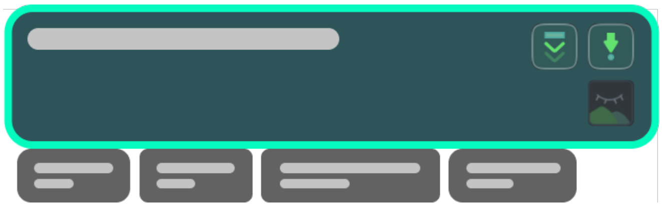
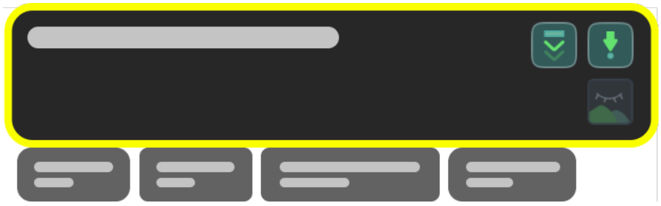
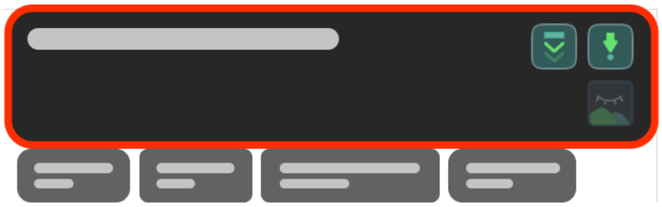


Move the mouse over the port and a detailed description of the data type as shown below will appear.



4.2.5.4. Statuses of Steps

Once the Steps are run, you can view the statuses of Steps by the frame color.

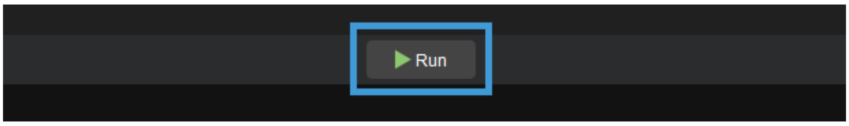
Status	Illustration
Regular	
Selected	
Warning	
Error	

4.2.5.5. Run Steps and View Outputs

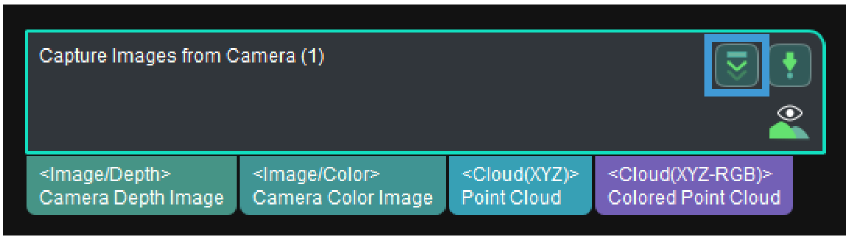
The successful functioning of an individual Step is the basis of the successful functioning of the entire project. This section will show you how to run the Step, and use the debug output feature to check the performance of the Step.

Run the Step

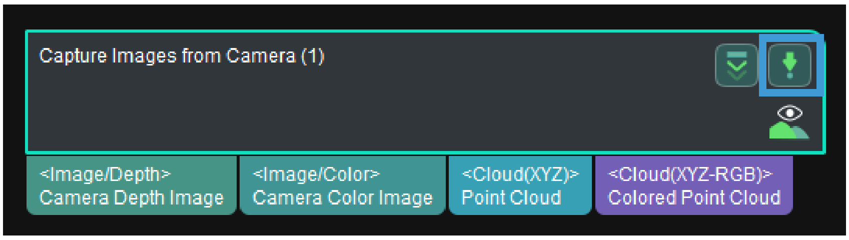
- Clicking **[Run]** in the Project toolbar will run the entire project.



- Clicking ⏬ on a Step will run the project down from the current Step.


Capture Images from Camera (1)
<Image/Depth> Camera Depth Image
<Image/Color> Camera Color Image
<Cloud(XYZ)> Point Cloud
<Cloud(XYZ-RGB)> Colored Point Cloud

- Click ⬇ to run the current Step independently.


Capture Images from Camera (1)
<Image/Depth> Camera Depth Image
<Image/Color> Camera Color Image
<Cloud(XYZ)> Point Cloud
<Cloud(XYZ-RGB)> Colored Point Cloud

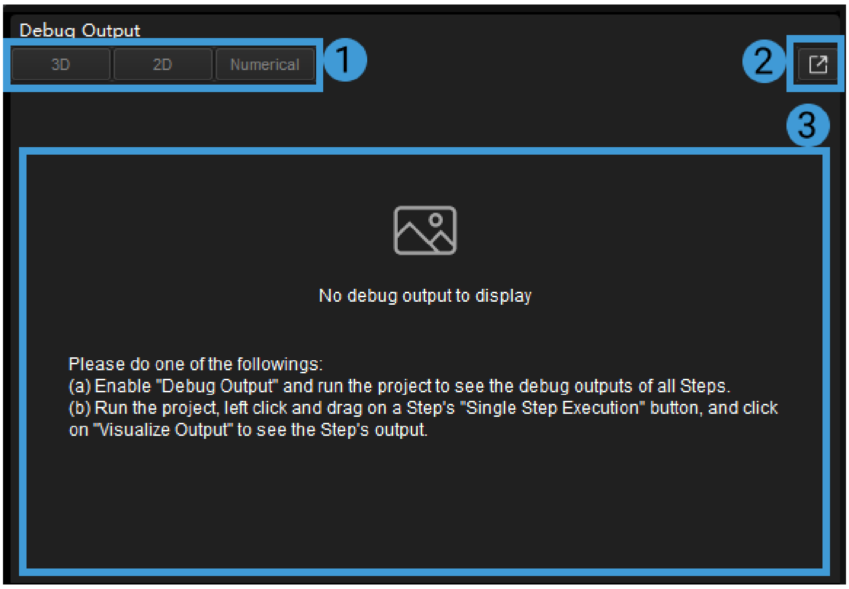
Debug Output

As an integral part of Mech-Vision, the debug output panel is where you can view the visualized output.

When debugging the project, the debug output feature enables users to visualize the output of the current Step.

Interface Introduction

The debug output panel is located in the upper right corner of the main interface, as shown below.



No debug output to display

Please do one of the followings:
(a) Enable "Debug Output" and run the project to see the debug outputs of all Steps.
(b) Run the project, left click and drag on a Step's "Single Step Execution" button, and click on "Visualize Output" to see the Step's output.

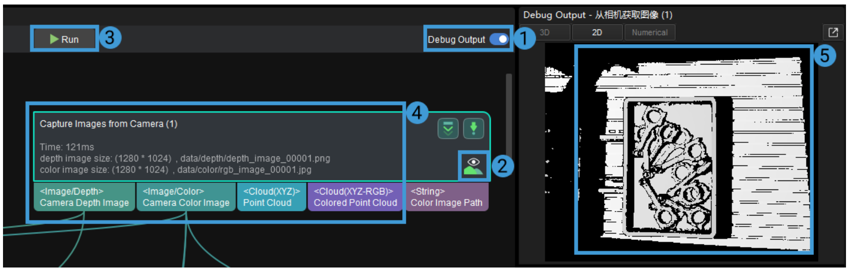
Interface description:

- The data type of the output will be shown in 1.
- Click 2 will make the panel pop up and turn it into an individual window.
- The visualized output will be displayed in 3.

Instructions

View the output of an individual Step in debug mode

Switch on **Debug Output** (1). Click 🖼 on the Step that you would like to view the output. An icon as 👁 suggests that the visualize output feature is enabled. Click **[Run]** in the Project tool bar. Then click the Step and you will see the visualized output in the debug output panel.

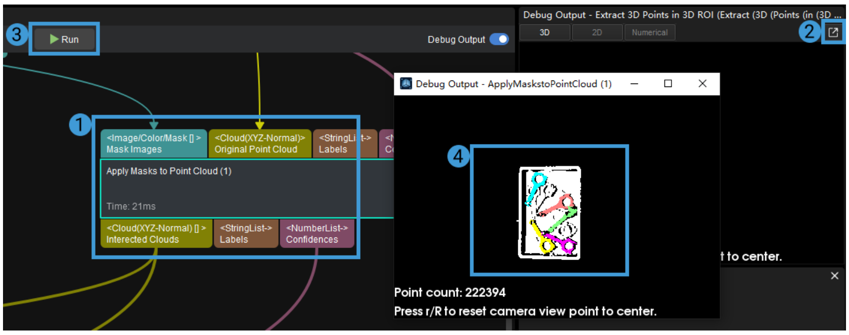


Capture Images from Camera (1)

Time: 121ms
depth image size: (1280 * 1024) , data/depth/depth_image_00001.png
color image size: (1280 * 1024) , data/color/rgb_image_00001.jpg

View outputs of multiple executions of an individual Step in debug mode

To view the visualized outputs of multiple executions of a specified Step, please select the Step first and then click ↗ to pop up the individual debug output window. Click **[Run]** in the Project tool bar and you can view the visualized outputs of multiple executions of the selected Step in the individual debug output window (4).

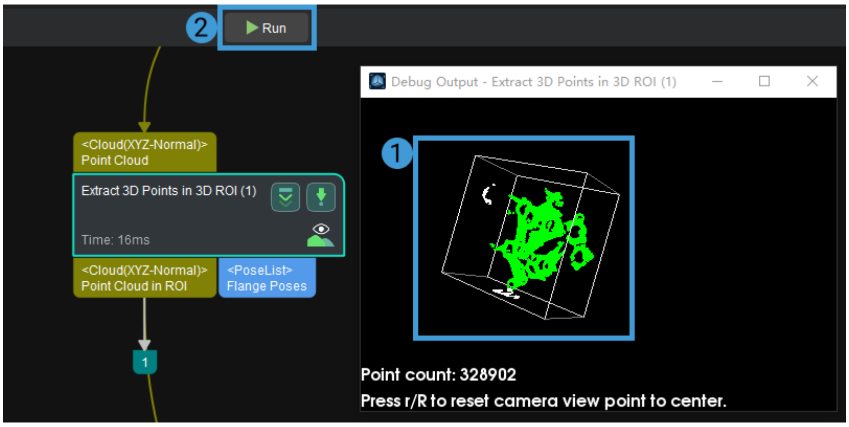


Point count: 222394
Press r/R to reset camera view point to center.

If you need to view multiple visualized outputs of multiple Steps, please follow the above instruction and open multiple individual debug output windows.

View outputs of an individual Step from other perspectives in debug mode

You can drag the 3D image in the individual debug output window to view the visualized outputs of an individual Step from other perspectives. Click [Run] in the Project tool bar and you can view the visualized outputs from your specified perspective.

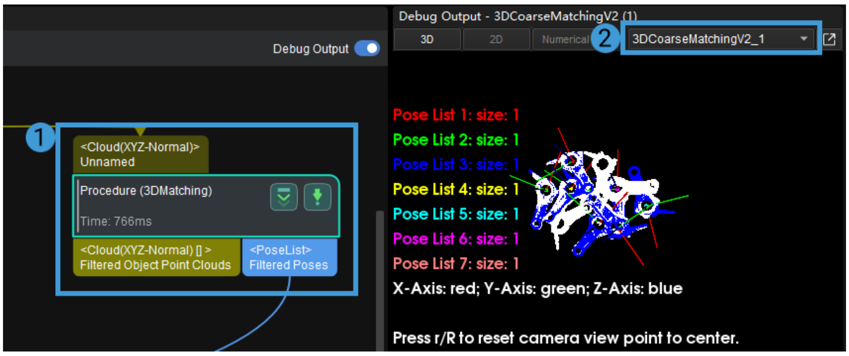


Point count: 328902
Press r/R to reset camera view point to center.

View outputs of different Steps in a Procedure

When a Procedure contains Steps whose debug output features are enabled, you can check visualized outputs of different Steps in the Procedure.

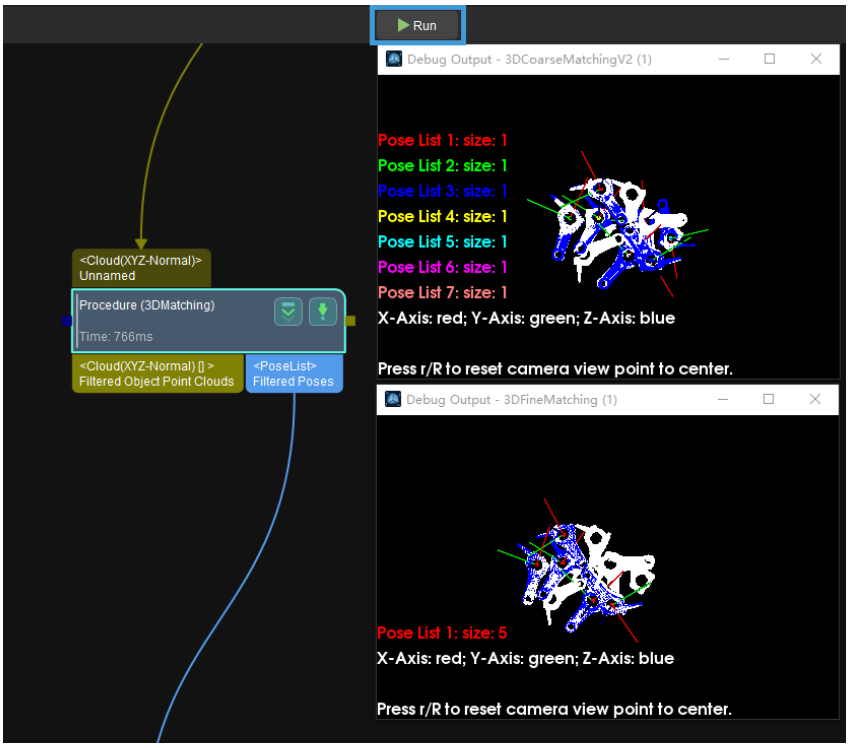
Click the Procedure, select the Step you would like to view the visualized output in the drop-down list (2) in the debug output panel.



Pose List 1: size: 1
Pose List 2: size: 1
Pose List 3: size: 1
Pose List 4: size: 1
Pose List 5: size: 1
Pose List 6: size: 1
Pose List 7: size: 1
X-Axis: red; Y-Axis: green; Z-Axis: blue

Press r/R to reset camera view point to center.

Click [Run] in the Project tool bar and you can view the visualized outputs of different Steps in the Procedure.



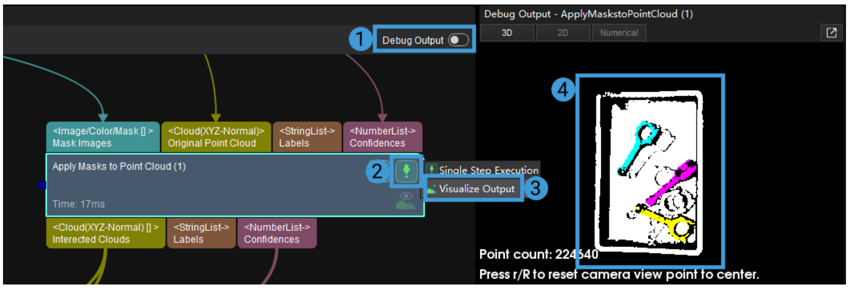
```
Pose List 1: size: 1
Pose List 2: size: 1
Pose List 3: size: 1
Pose List 4: size: 1
Pose List 5: size: 1
Pose List 6: size: 1
Pose List 7: size: 1
X-Axis: red; Y-Axis: green; Z-Axis: blue

Press r/R to reset camera view point to center.
```

```
Pose List 1: size: 5
X-Axis: red; Y-Axis: green; Z-Axis: blue

Press r/R to reset camera view point to center.
```

View the output of an individual Step in non-debug mode

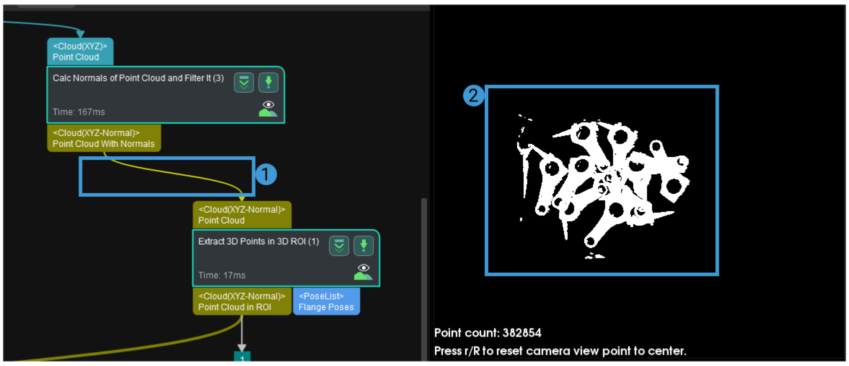
When the **Debug Output** is not switched on, you can still check the visualized output of the Step. The two methods are shown below.

- Click ⬇ and drag in any direction, select **[Visualize Output]** and release the mouse button. Then you can view the visualized output in the debug output panel.



```
Point count: 224540
Press r/R to reset camera view point to center.
```

- Click the line connecting the Steps and view the visualized output in the debug output panel.



```
Point count: 382854
Press r/R to reset camera view point to center.
```

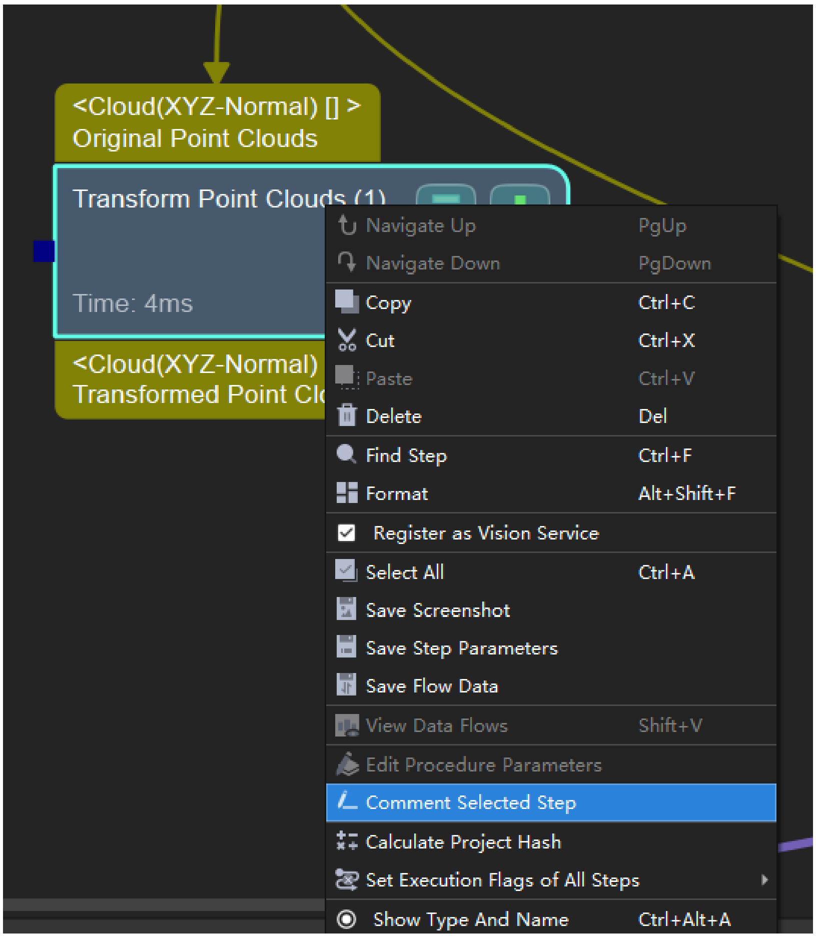
Please run the entire project or the single Step before viewing the visualized output.

4.2.5.6. Introduction to Step Comments

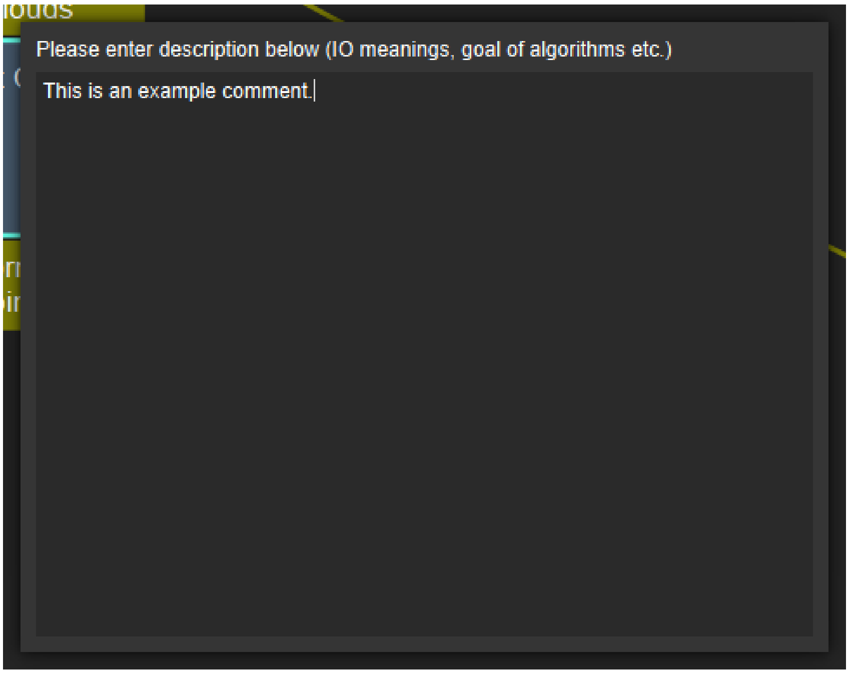
You can add a comment or description to a certain Step or Procedure in the project

How to Add Comments

1. Right-click the Step or Procedure to which you want to add a comment, and select **Comment Selected Step**.



2. Enter your comment in the pop-up window, and then click on anywhere in Mech-Vision to close the window.

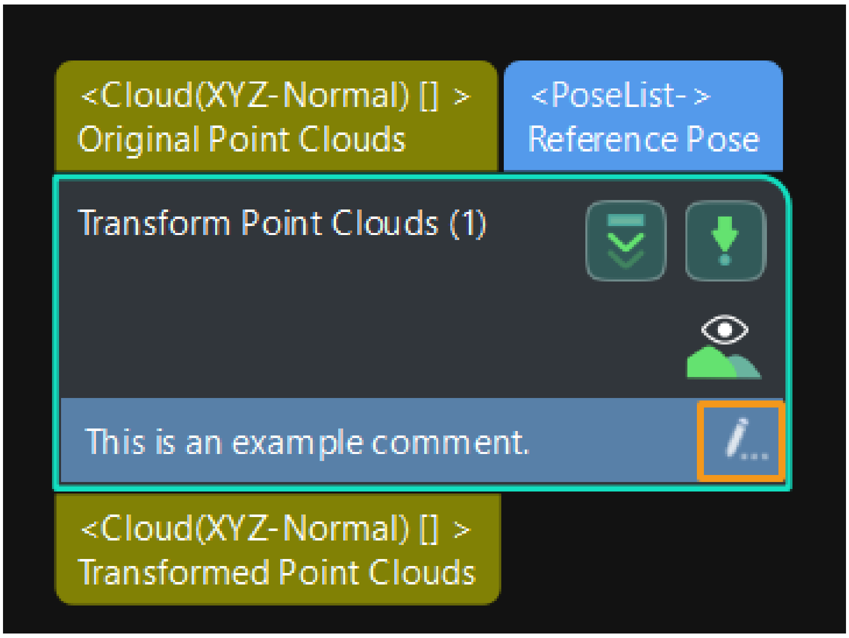


Please enter description below (IO meanings, goal of algorithms etc.)

This is an example comment.

How to Modify the Comment

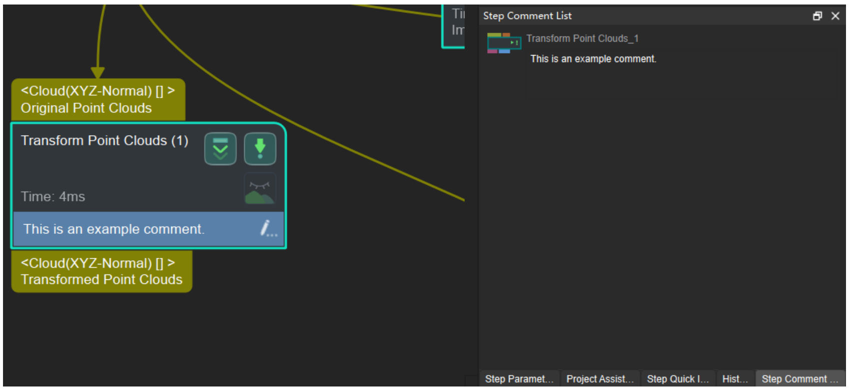
Click ✎.



How to View the Comment

Click the **Step Comment List** tab in the lower right to view all the comments in the current project. If there is no such tab, please check if View › Step Comment List is checked.

Click an comment, and the Step or Procedure to which the comment is added will be centered and flash in the Graphical Programming Workspace.



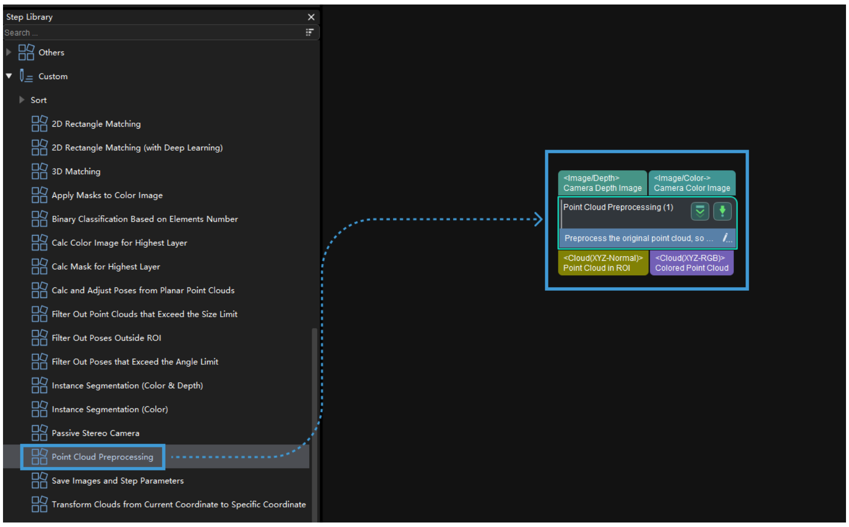
4.2.5.7. Basic Features of Procedures

Before you get to know the Procedures, you should know **Basic Features of Procedures**.

The following instructions are based on the Procedure **Point Cloud Preprocessing**. Please create a new project first.

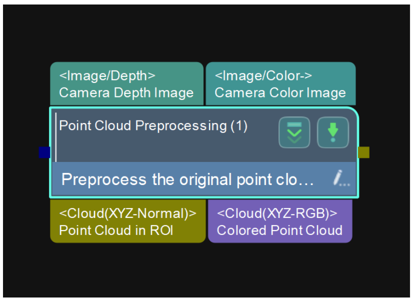
- How to add a Procedure in the project?

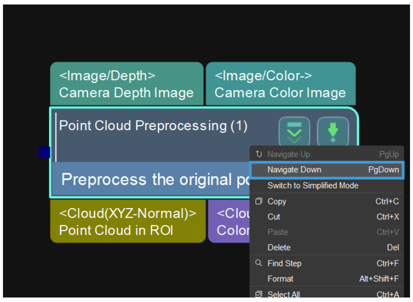
Go to Custom › Point Cloud Preprocessing in the Step library, and drag the Procedure to the graphical programming workspace.


Step Library
Search ...
Others
Custom
Sort
2D Rectangle Matching
2D Rectangle Matching (with Deep Learning)
3D Matching
Apply Masks to Color Image
Binary Classification Based on Elements Number
Calc Color Image for Highest Layer
Calc Mask for Highest Layer
Calc and Adjust Poses from Planar Point Clouds
Filter Out Point Clouds that Exceed the Size Limit
Filter Out Poses Outside ROI
Filter Out Poses that Exceed the Angle Limit
Instance Segmentation (Color & Depth)
Instance Segmentation (Color)
Passive Stereo Camera
Point Cloud Preprocessing
Save Images and Step Parameters
Transform Clouds from Current Coordinate to Specific Coordinate
<Image/Depth>
Camera Depth Image
<Image/Color->
Camera Color Image
Point Cloud Preprocessing (1)
Preprocess the original point cloud, so ...
<Cloud(XYZ-Normal)>
Point Cloud in ROI
<Cloud(XYZ-RGB)>
Colored Point Cloud

- How to view the Steps in a Procedure?

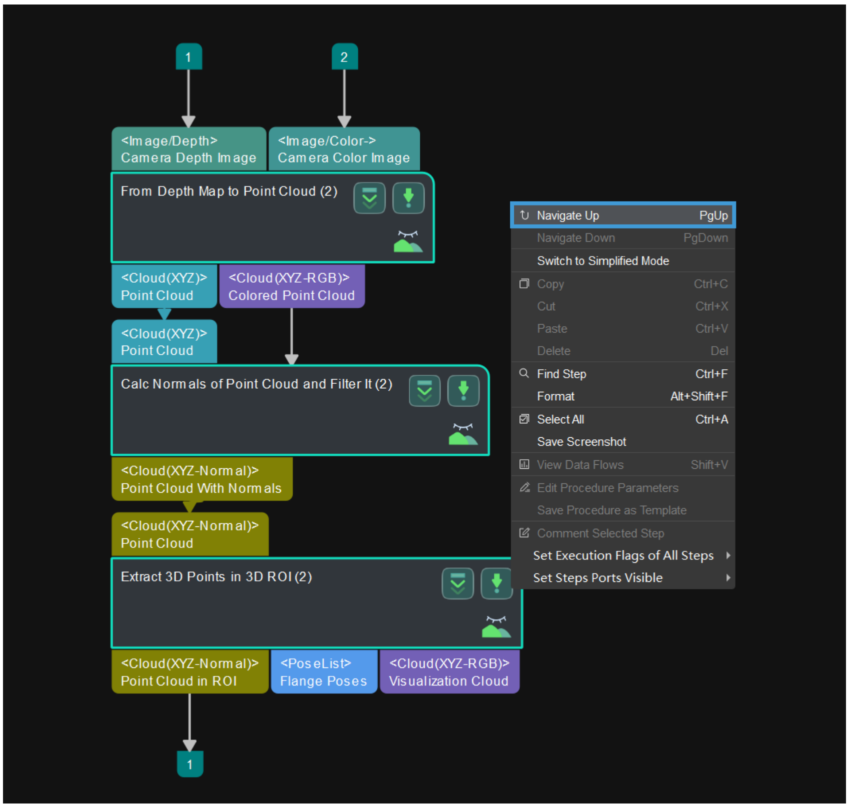
Double-click the Procedure or right-click the Procedure and select **[Navigate Down]** (shortcut key: `PgDown`) to view the Steps.





- How to go back to view the whole Procedure?

Right-click in any space in the graphical programming workspace, and select **[Navigate Up]** (shortcut key: `PgUp`).

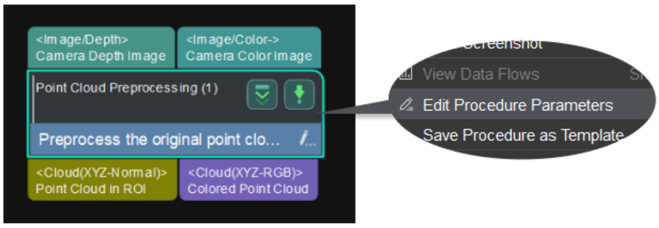


4.2.5.8. Modify Parameters of Procedures

You can modify the parameters of the Procedures according to different scenarios. The following instructions are based on the Procedure **Point Cloud Preprocessing** and mainly covers the following topics:

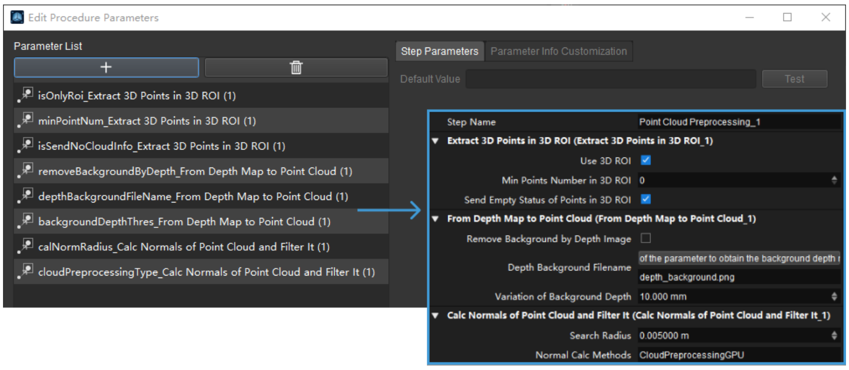
- Delete and Add Parameters
- Customize the Display Information of the Parameter

Right-click the **Point Cloud Preprocessing** and select **[Edit Procedure Parameters]** in the context menu.



Delete and Add Parameters

The parameters in the **Parameter List** correspond with those in the **Step Parameter** panel, as shown below.



Parameter List

- isOnlyRoi_Extract 3D Points in 3D ROI (1)
- minPointNum_Extract 3D Points in 3D ROI (1)
- isSendNoCloudInfo_Extract 3D Points in 3D ROI (1)
- removeBackgroundByDepth_From Depth Map to Point Cloud (1)
- depthBackgroundFileName_From Depth Map to Point Cloud (1)
- backgroundDepthThres_From Depth Map to Point Cloud (1)
- calNormRadius_Calc Normals of Point Cloud and Filter It (1)
- cloudPreprocessingType_Calc Normals of Point Cloud and Filter It (1)

Step Name: Point Cloud Preprocessing_1

▼ Extract 3D Points in 3D ROI (Extract 3D Points in 3D ROI_1)

- Use 3D ROI ☑
- Min Points Number in 3D ROI: 0
- Send Empty Status of Points in 3D ROI ☑

▼ From Depth Map to Point Cloud (From Depth Map to Point Cloud_1)

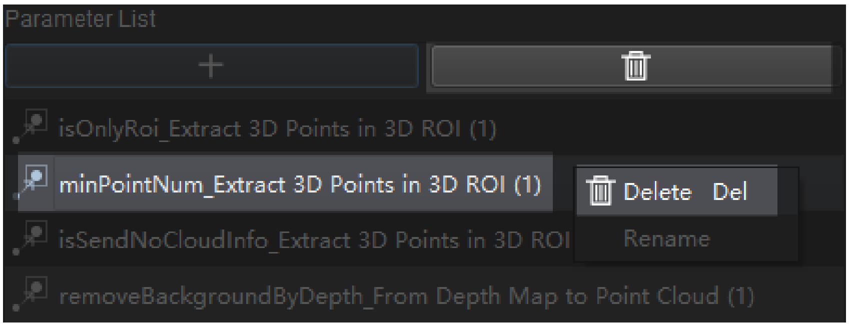
- Remove Background by Depth Image □
- Depth Background Filename: of the parameter to obtain the background depth i / depth_background.png
- Variation of Background Depth: 10.000 mm

▼ Calc Normals of Point Cloud and Filter It (Calc Normals of Point Cloud and Filter It_1)

- Search Radius: 0.005000 m
- Normal Calc Methods: CloudPreprocessingGPU

Delete a Parameter

Click the parameter that you want to delete and then click 🗑, or right-click the parameter and select **[Delete]** in the context menu to delete the parameter. Click **[Yes]** in the pop-up window.

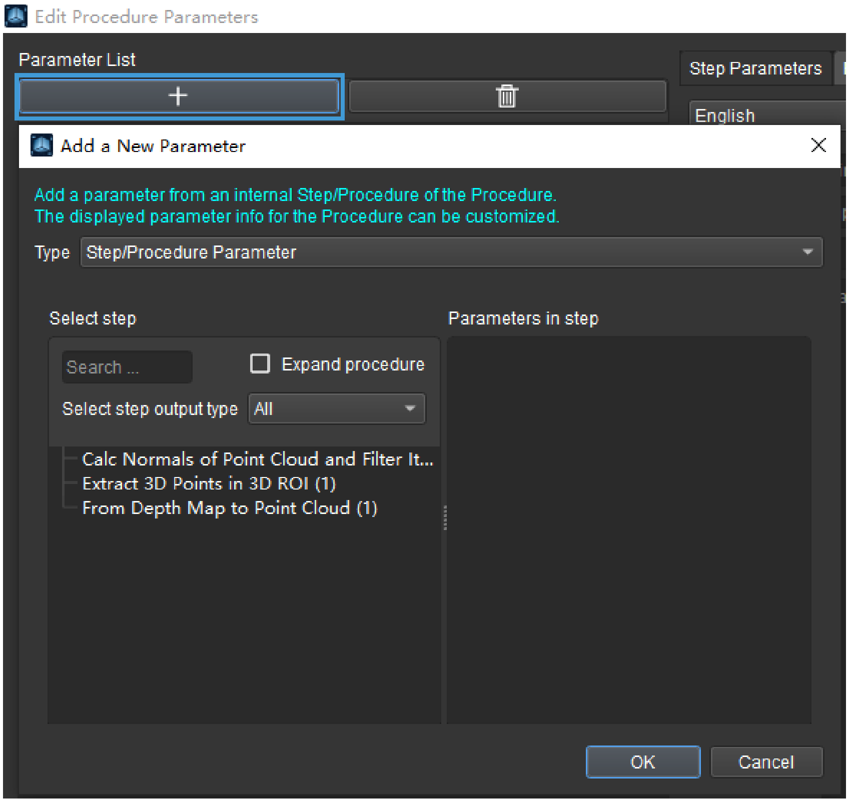


- isOnlyRoi_Extract 3D Points in 3D ROI (1)
- minPointNum_Extract 3D Points in 3D ROI (1)
- isSendNoCloudInfo_Extract 3D Points in 3D ROI
- removeBackgroundByDepth_From Depth Map to Point Cloud (1)

Click **[Apply]** to save the setting, and then you will find that the parameter in the **Step Parameter** panel has disappeared.

Add a Parameter

In the pop-up **Edit Procedure Parameters** window, click **[+]** to open the **Add a New Parameter** window.


Edit Procedure Parameters

Parameter List

Add a New Parameter

Add a parameter from an internal Step/Procedure of the Procedure.
The displayed parameter info for the Procedure can be customized.

Type: Step/Procedure Parameter

Select step

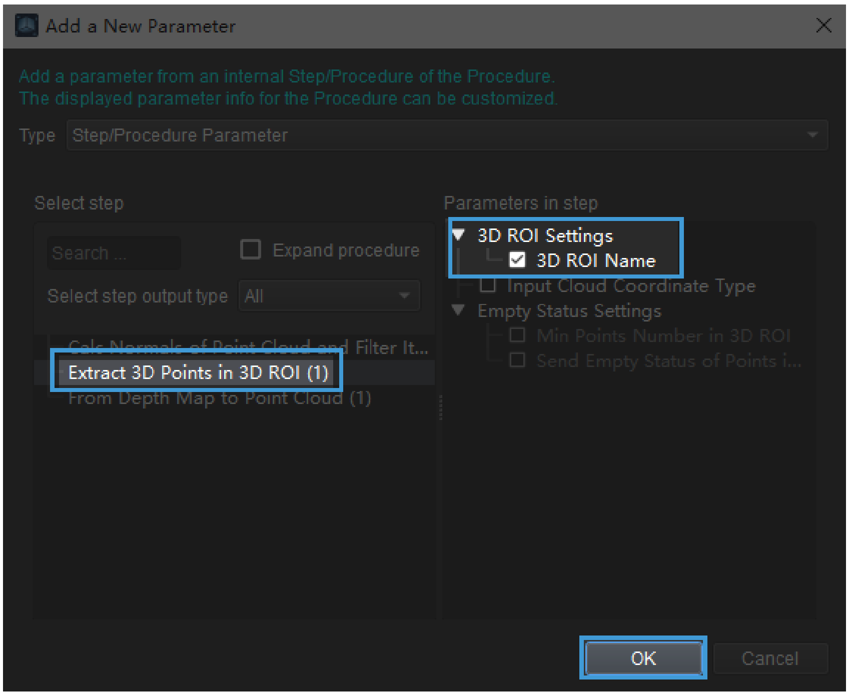
□ Expand procedure

Select step output type: All

- Calc Normals of Point Cloud and Filter It...
- Extract 3D Points in 3D ROI (1)
- From Depth Map to Point Cloud (1)

Parameters in step

Select a Step and its parameter in the window. As shown below, the Step Extract 3D Points in 3D ROI and its **3D ROI Settings** parameter is selected. Click **[OK]** to save settings.


Add a New Parameter

Add a parameter from an internal Step/Procedure of the Procedure.
The displayed parameter info for the Procedure can be customized.

Type: Step/Procedure Parameter

Select step

□ Expand procedure

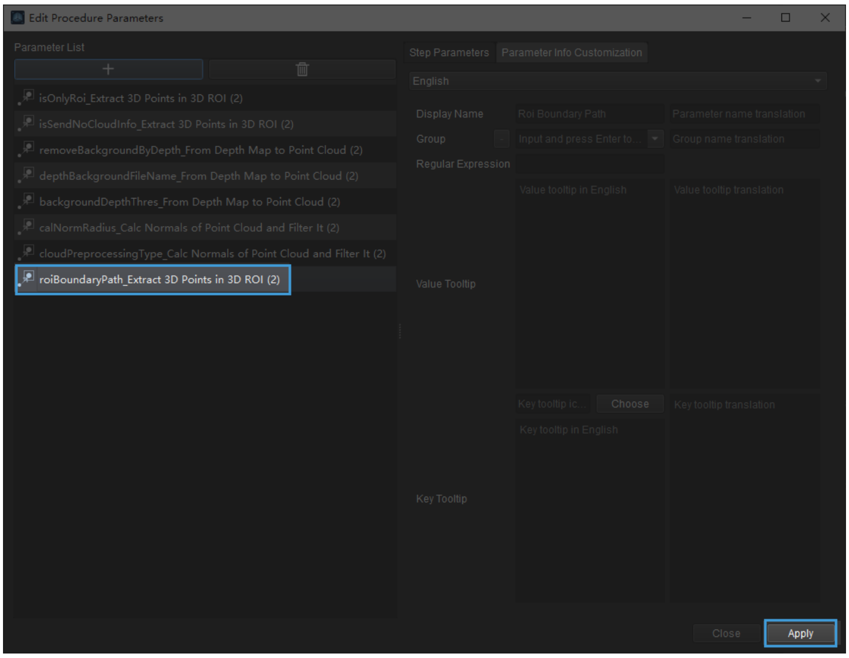
Select step output type: All

- Calc Normals of Point Cloud and Filter It...
- Extract 3D Points in 3D ROI (1)
- From Depth Map to Point Cloud (1)

Parameters in step

- 3D ROI Settings
  - ☑ 3D ROI Name
  - □ Input Cloud Coordinate Type
- Empty Status Settings
  - □ Min Points Number in 3D ROI
  - □ Send Empty Status of Points i...

The selected parameter will appear in the **Parameter List**. Click **[Apply]** to apply the settings.



Edit Procedure Parameters

Parameter List

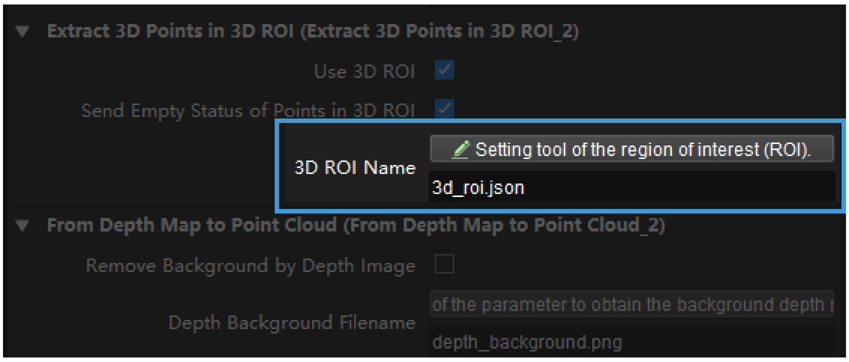
- isOnlyRoi_Extract 3D Points in 3D ROI (2)
- isSendNoCloudInfo_Extract 3D Points in 3D ROI (2)
- removeBackgroundByDepth_From Depth Map to Point Cloud (2)
- depthBackgroundFileName_From Depth Map to Point Cloud (2)
- backgroundDepthThres_From Depth Map to Point Cloud (2)
- calNormRadius_Calc Normals of Point Cloud and Filter It (2)
- cloudPreprocessingType_Calc Normals of Point Cloud and Filter It (2)
- roiBoundaryPath_Extract 3D Points in 3D ROI (2)

Step Parameters | Parameter Info Customization

English

Display Name: Roi Boundary Path

Then you can see the newly added parameter in the **Step Parameter** panel.



▼ Extract 3D Points in 3D ROI (Extract 3D Points in 3D ROI_2)

Use 3D ROI ☑

Send Empty Status of Points in 3D ROI ☑

3D ROI Name: Setting tool of the region of interest (ROI).
3d_roi.json

▼ From Depth Map to Point Cloud (From Depth Map to Point Cloud_2)

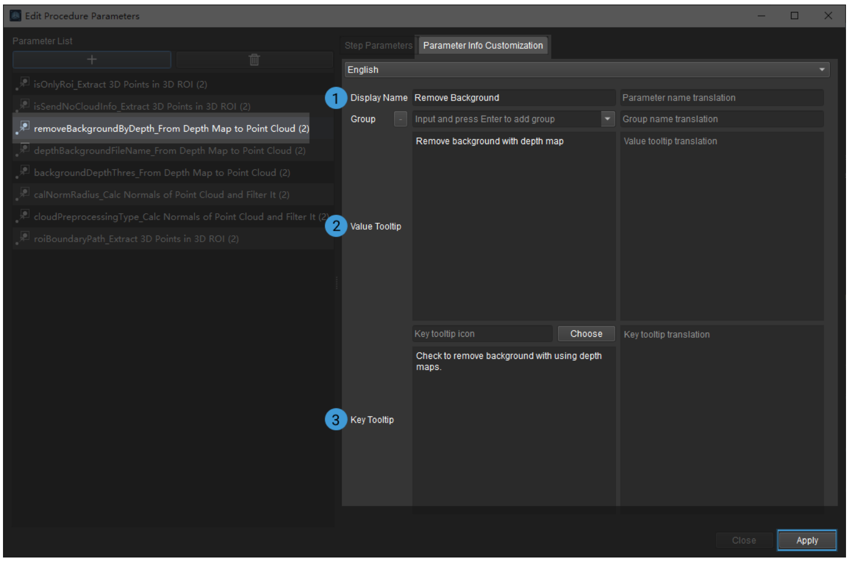
Remove Background by Depth Image □

Depth Background Filename: of the parameter to obtain the background depth i
depth_background.png

Customize the Display Information of the Parameter

Open the **Edit Procedure Parameters** window as mentioned before. Click the parameter in the **Parameter List** and go to the **Parameter Info Customization** tab.

You can edit the **Display Name**, **Value Tooltip**, and **Key Tooltip**. Click **[Apply]** to save the editing.



Edit Procedure Parameters

Parameter List

- isOnlyRoi_Extract 3D Points in 3D ROI (2)
- isSendNoCloudInfo_Extract 3D Points in 3D ROI (2)
- removeBackgroundByDepth_From Depth Map to Point Cloud (2)
- depthBackgroundFileName_From Depth Map to Point Cloud (2)
- backgroundDepthThres_From Depth Map to Point Cloud (2)
- calNormRadius_Calc Normals of Point Cloud and Filter It (2)
- cloudPreprocessingType_Calc Normals of Point Cloud and Filter It (2)
- roiBoundaryPath_Extract 3D Points in 3D ROI (2)

Step Parameters | Parameter Info Customization

English

1 Display Name: Remove Background | Parameter name translation

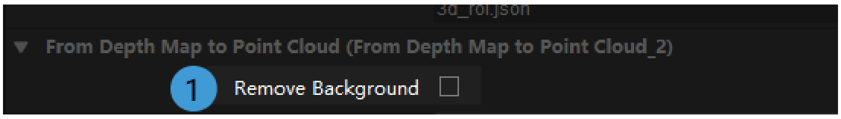
Group: Input and press Enter to add group | Group name translation

2 Value Tooltip: Remove background with depth map | Value tooltip translation

Key tooltip icon | Choose | Key tooltip translation

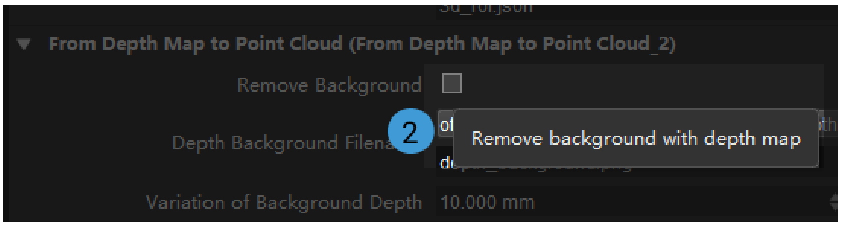
3 Key Tooltip: Check to remove background with using depth maps.

The displayed parameter information after being modified is shown below.



From Depth Map to Point Cloud (From Depth Map to Point Cloud_2)

1 Remove Background □

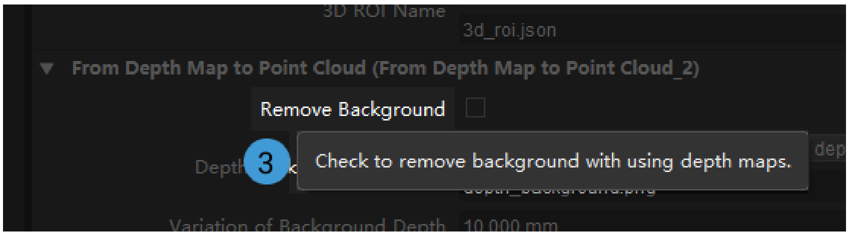


From Depth Map to Point Cloud (From Depth Map to Point Cloud_2)

Remove Background □

Depth Background Filena 2 Remove background with depth map

Variation of Background Depth 10.000 mm



3D ROI Name 3d_roi.json

From Depth Map to Point Cloud (From Depth Map to Point Cloud_2)

Remove Background □

Depth 3 Check to remove background with using depth maps.

Variation of Background Depth 10.000 mm

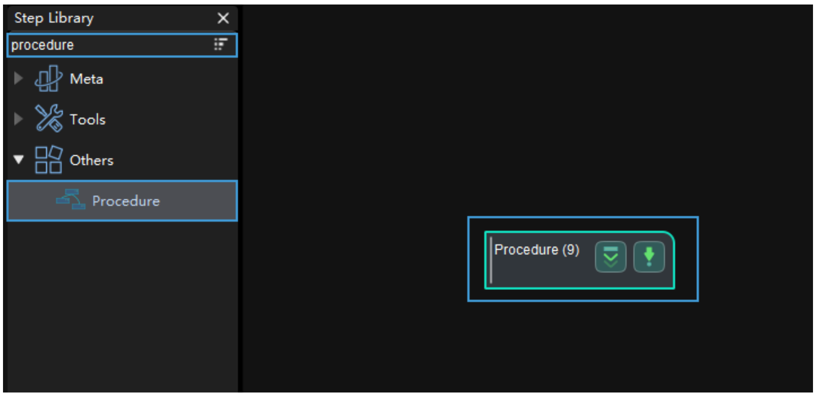
4.2.5.9. Create your own Procedure

Customizing Procedures is supported.

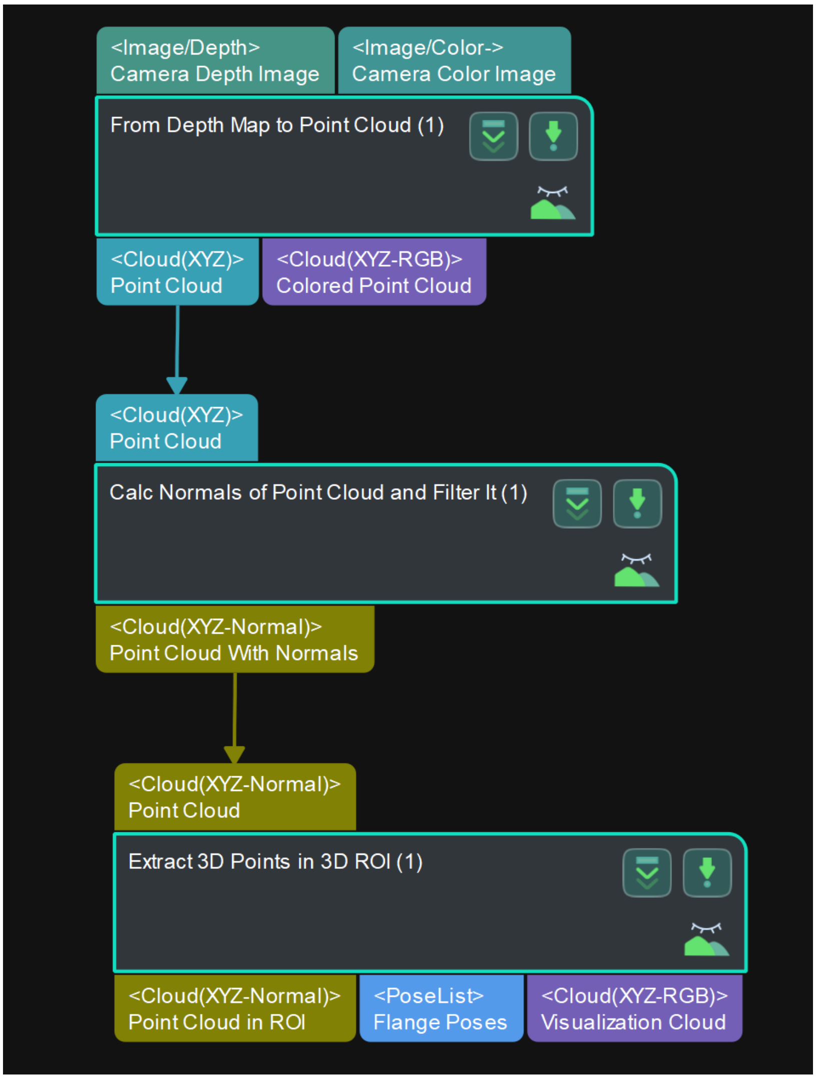
This section will show you how to package Steps to create a customized Procedure Point Cloud Processing. The Steps From Depth Map to Point Cloud, Calc Normals of Point Cloud and Filter It, and Extract 3D Points in 3D ROI are needed to create the Procedure, and the packaging should be completed with Procedure.

Add Steps

1. Locate [vision-steps:procedure:::vision-procedure] in the Step library and drag it to the graphical programming workspace, as shown below.



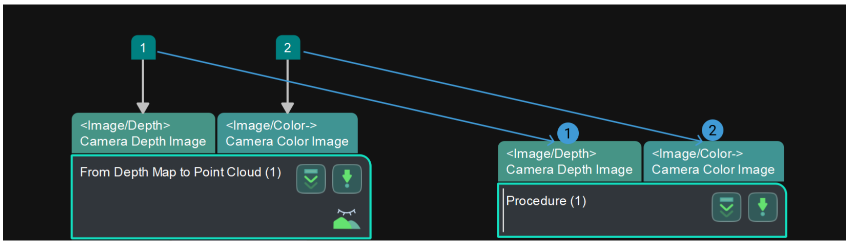
2. Click [vision-steps:procedure:::vision-procedure] to enter the Step level and then connect the Steps as shown below.



Set the Input and Output of the Procedure

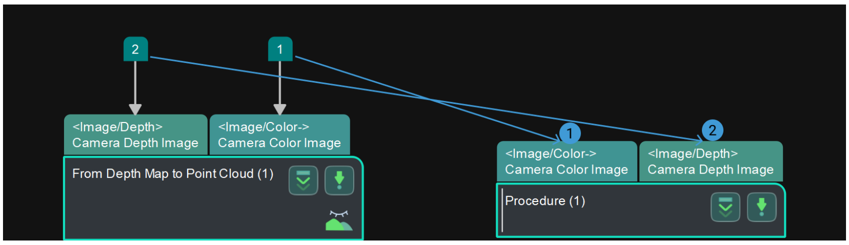
- Input:

Double-click the input port of [vision-steps:from-depth-map-to-point-cloud:::from-depth-map-to-point-cloud]. A short line with the number indicates that the port will be used as the input port of the Procedure.



The sequence of clicking the port decides the sequence of the port being displayed on the Procedure.

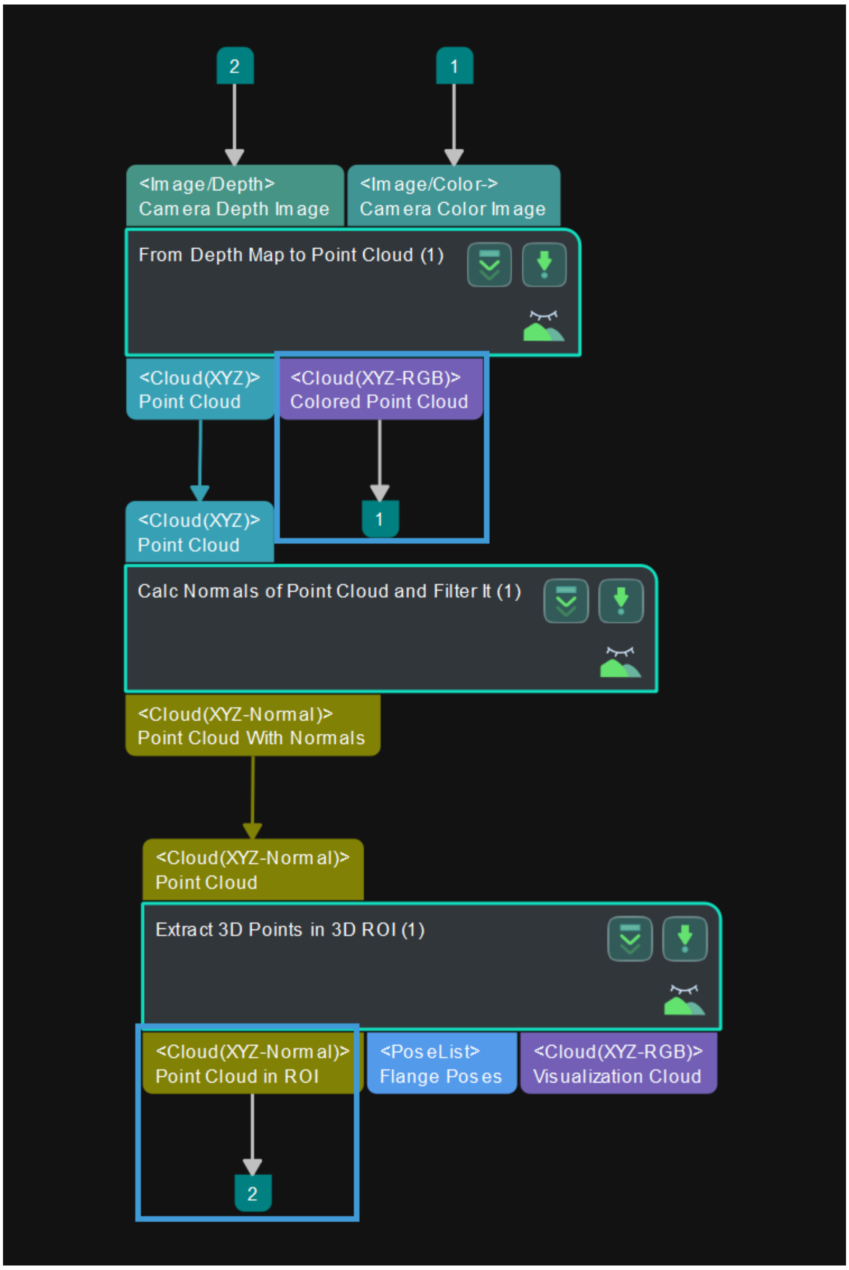
The figure above shows the layout of the input ports of the Procedure when **Camera Depth Image** is clicked before **Camera Color Image**, while the figure below shows the layout when **Camera Color Image** is clicked before **Camera Depth Image**.



Click the number on the port, press `Delete` or right-click the number and select [**Delete**] in the context menu to delete the port.

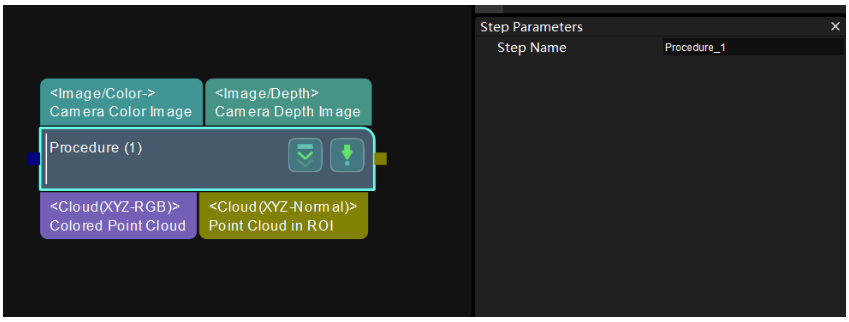
- Output:

The method to set the output ports is the same as that of the input ports.



Edit Procedure Parameters

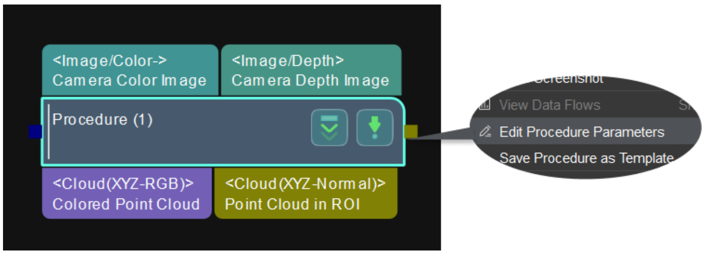
After creating a Procedure, you can only change the name of it, as shown below.



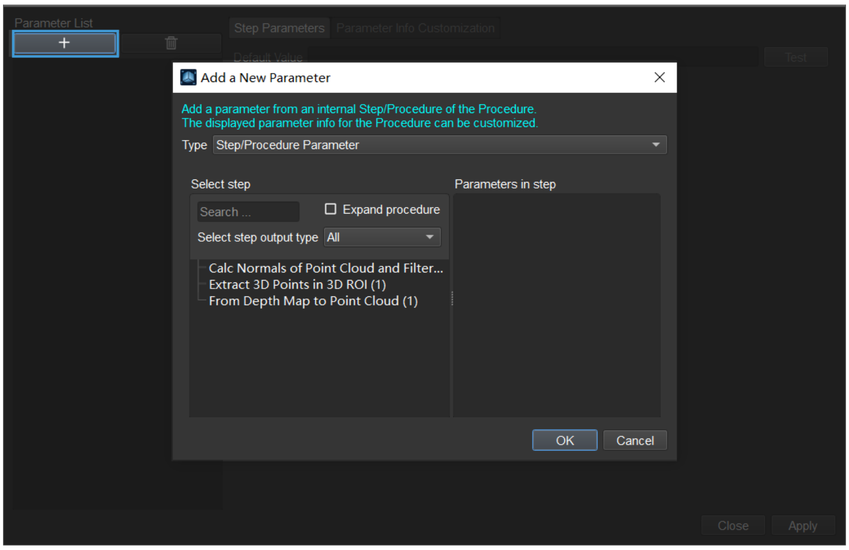
Step Name Procedure_1

To facilitate adjusting the parameters of the Procedure, you can select some key parameters of the Steps as the parameters of the Procedure.

1. Right-click the Procedure and select **[Edit Procedure Parameters]** in the context menu.


<Image/Color-> Camera Color Image
<Image/Depth> Camera Depth Image
Procedure (1)
<Cloud(XYZ-RGB)> Colored Point Cloud
<Cloud(XYZ-Normal)> Point Cloud in ROI
Screenshot
View Data Flows
Edit Procedure Parameters
Save Procedure as Template

2. In the pop-up **Edit Procedure Parameters** window, click **[+]** to open the **Add a New Parameter** window.



Add a New Parameter

Add a parameter from an internal Step/Procedure of the Procedure.
The displayed parameter info for the Procedure can be customized.

Type: Step/Procedure Parameter

Select step

Search ...

☐ Expand procedure

Select step output type: All

- Calc Normals of Point Cloud and Filter...
- Extract 3D Points in 3D ROI (1)
- From Depth Map to Point Cloud (1)

Parameters in step

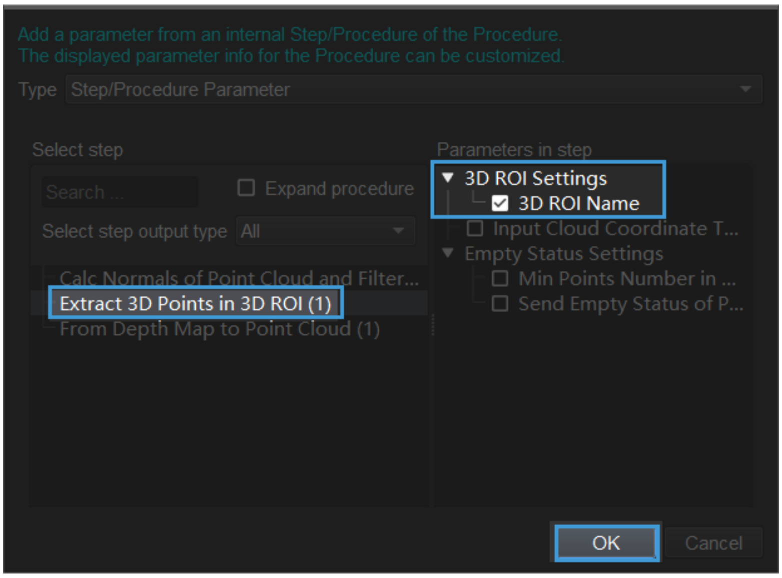
OK Cancel

There are three Procedure parameters:

- Step/Procedure Parameter (the one used here)
- Custom Mapped Parameter
- Custom Recipe Parameter

1. Select a Step and its parameter in the window. As shown below, the Step [vision-steps:extract-3d-points-in-3d-roi:::extract-3d-points-in-3d-roi] and its **3D ROI Settings** parameter is selected. Click **[OK]** to save settings.

> You can select multiple parameters according to actual needs.



Add a parameter from an internal Step/Procedure of the Procedure.
The displayed parameter info for the Procedure can be customized.

Type: Step/Procedure Parameter

Select step

Search ...

☐ Expand procedure

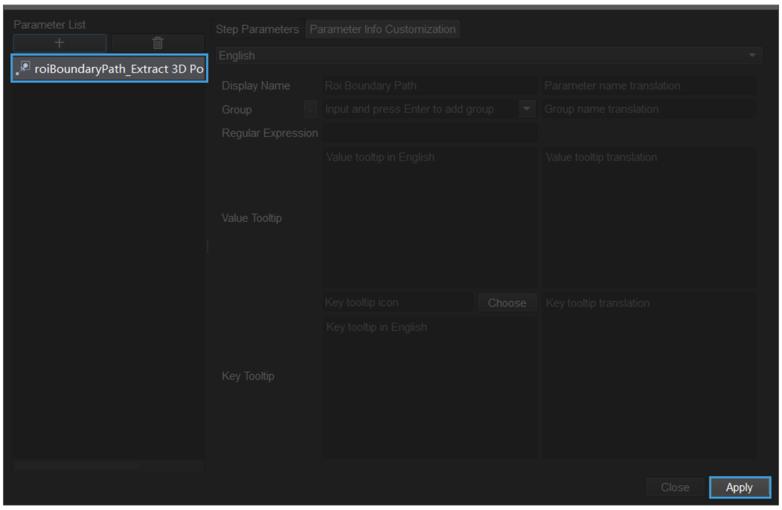
Select step output type: All

Calc Normals of Point Cloud and Filter...
Extract 3D Points in 3D ROI (1)
From Depth Map to Point Cloud (1)

Parameters in step

- ▼ 3D ROI Settings
  - ☑ 3D ROI Name
- ☐ Input Cloud Coordinate T...
- ▼ Empty Status Settings
  - ☐ Min Points Number in ...
  - ☐ Send Empty Status of P...

2. Now you can see the selected parameter in the Parameter List. Click **[Apply]** to apply the changes.



Parameter List

roiBoundaryPath_Extract 3D Po

Step Parameters | Parameter Info Customization

English

Display Name: Roi Boundary Path | Parameter name translation

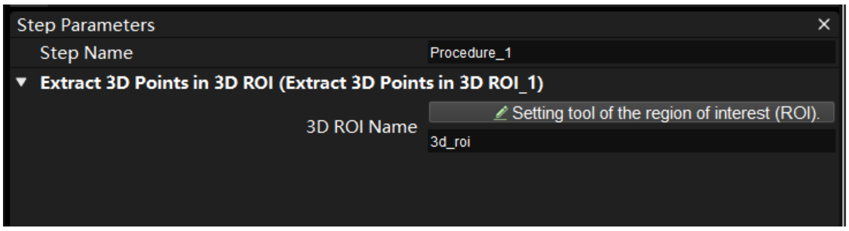
Group: Input and press Enter to add group | Group name translation

Regular Expression

Value Tooltip: Value tooltip in English | Value tooltip translation

Key Tooltip: Key tooltip icon (Choose) | Key tooltip in English | Key tooltip translation

3. The selected parameter will appear in the **Step Parameters** window.



Step Parameters

Step Name: Procedure_1

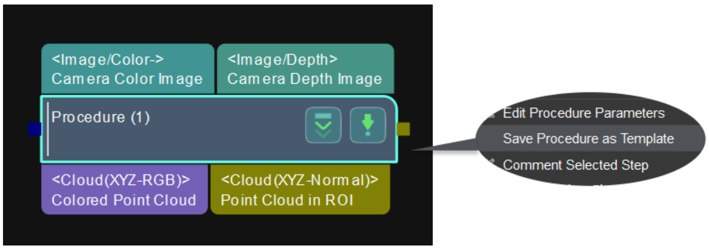
▼ **Extract 3D Points in 3D ROI (Extract 3D Points in 3D ROI_1)**

3D ROI Name: Setting tool of the region of interest (ROI).
3d_roi

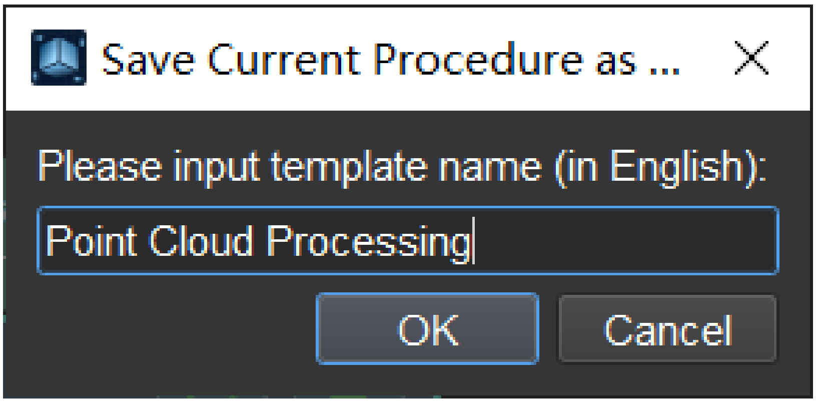
Save the Procedure

You can add the customized Procedure to the Custom directory in the Step library.

1. Right-click the Procedure and select **[Save Procedure as Template]** in the context menu.



2. Name the Procedure template.

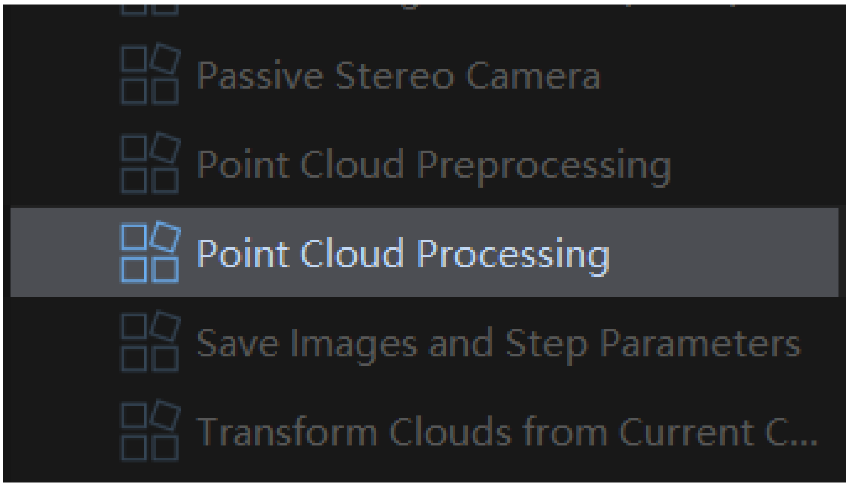


Please input template name (in English):

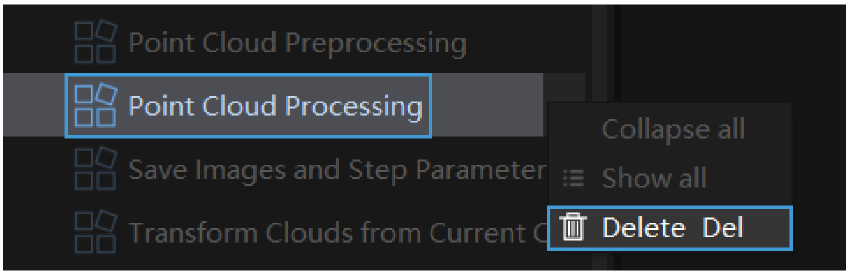
Point Cloud Processing

The name of the customized Procedure template cannot be the same as other existing Step/Procedure names.

3. Click [OK] to save the name. Then you can see the newly added Procedure in the Custom directory in the Step library.


Passive Stereo Camera
Point Cloud Preprocessing
Point Cloud Processing
Save Images and Step Parameters
Transform Clouds from Current C...

To delete a customized Procedure, you can select the Procedure and press `Delete`, or right-click the Procedure and select [Delete] in the context menu.


Point Cloud Preprocessing
Point Cloud Processing
Save Images and Step Parameter
Transform Clouds from Current C
Collapse all
Show all
Delete Del

- custom-mapped-parameter.pdf
- custom-recipe-parameter.pdf

4.3. Step Reference Guide

This chapter provides the reference guide for using Steps and Procedures in Mech-Vision.

Step Reference

- 2d-matching.pdf
- Convert Poses 2D to 3D According to Orthographic Projection
- 2D Shape Matching
- 3D Coarse Matching (Multiple Models)
- 3D Coarse Matching V2
- 3D Coarse Matching
- 3D Fine Matching Lite
- 3D Fine Matching (Multiple Models)
- 3D Fine Matching
- 3D Workpiece Recognition
- Accept All
- Add Labels to Poses
- Adjust Inaccurate Poses Caused by Camera Distortion
- Adjust Poses by Offsets
- Adjust Poses by Tilt
- Adjust Targets to Get Correct Path
- Move Poses to Point Cloud Surfaces
- Change Circular Motion Direction of Path
- Allocator
- Apply Masks to Image
- Apply Masks to Point Cloud
- Extract 2D Path Function
- Background Subtraction
- Image Thresholding
- Blob Analysis
- Boolean List Logical Operation
- Bridge
- Calc Absolute Values
- Calc Angles between Vector3Ds
- Calc Areas of Masks

- Calc Box Dimensions
- Calc Vector3D Cross Products
- Calc Diagonal Lengths
- Calc Diameter and Thickness
- Calc Distance between Two Poses along Specified Direction
- Calc Distances between Poses
- Calc Vector3D Dot Products
- Calc Edge Point Cloud Normals
- Calc Flatness Error
- Calc Histogram
- Calc Included Angles between Specified Axes of Poses
- Calc Intersection between Two Line Segments
- Calc Intersections between Line Segment and Circle
- Calc Point Cloud Spans along Axes
- Calc Lengths of Vector3Ds
- Calc Mask Distances
- Calc Mask Spans on Given Lines
- Calc Mean Gray Value
- Calc Midpoint of Specified Side of Rectangle
- Calc Minimum Bounding Rectangles Of Masks
- Calc Normalized Vector3Ds
- Calc Normals and Estimate Edges of Point Cloud
- Calc Normals of Point Cloud and Filter It
- Calc Parallelism Error
- Calc Pixel Sizes at Specified Heights
- Calc Plane Width
- Calc Point Cloud Curvatures
- Calc Poses and Dimensions from Planar Point Clouds
- calc-poses-and-dimensions-of-rectangles.pdf
- Calc Profile by Sampling
- Calc Profile
- Calc Projection Length along Reference Direction
- Calc 2D Poses of Rectangles
- Calc Results by Python
- Calc Specified Property of Point Clouds
- Calc Calibration Matrix of Structured Light Sensor
- Calc Center Points of Non-Zero Regions

- Calc Cylinder Poses
- calculate-calib-board-pose.pdf
- Caliper Tool
- Camera 2D
- Capture Images from Camera
- Classify Point Clouds by Dimensions
- Correct Point Cloud Distortion
- Filter Point Cloud by Model and Poses
- Cloud Num Limit
- Cloud Processing (GPU)
- Scale Point Cloud
- Smooth Point Cloud and Estimate Normals
- Extract Point Cloud in 3D Box
- From CloudXYZ To CloudNormal
- Cluster Point Clouds and Output Eligible Point Clouds
- Draw Coherent Lines
- Collect Poses in 3D ROI
- Compare Two Depth Maps
- Compare Values
- Compose New Poses by Combining Parts of Input Poses
- Compose Poses from Quaternions and Translation Vectors
- Compose Quaternions from Axes and Angles
- Compose Quaternions from Two Axes (Right Hand Rule)
- Compose Vector3Ds from Numbers
- Control Light Source
- Convert 2D Poses to 3D Poses
- Convert Data Type
- Convert Lengths Pixel-Wise to Physical
- Convert 2D Points to 3D Points
- Convert Circles To 2D Poses
- Convert Color Space
- Copy Images
- count-boxes.pdf
- Count Elements in Specified Dimension in Data List
- Count 3D Points
- Count Color Info
- Create Object Model

- Crop Image
- Decompose Object Dimensions
- Decompose Poses to Quaternions and Translations
- Decompose Quaternions to X- Y- Z- Axes
- Decompose Vector3D to Numbers
- deep-learning-inference.pdf
- Deep Learning Model Package Inference
- Deep Learning Result Parser
- Depth Clustering along Scan Lines
- Depth Encoding
- Detect and Measure Circle
- Detect and Measure Line
- Detect and Measure Oblong Hole
- Detect Bin (Four Sides)
- Detect Bin (Max Inscribed Rect)
- Detect Bin (Max Inscribed Rect) V2
- Detect Bin (Two Sides)
- Detect Circle Centers
- Detect Corners
- Detect Edges
- detect-fiducial-markers.pdf
- Detect Inscribed Circles
- Detect Line Segments
- Detect N Largest Rectangles in Mask
- Detect Occluded Objects
- Detect Shape Feature of Region
- Detect Vertex
- Determine Pixel Sizes
- Dichotomize Values by Threshold
- Divide Point Cloud into Smaller Parts Evenly
- Down-Sample Point Cloud
- Draw Min Circumscribed Rectangles of Masks
- draw-polygon-vertices.pdf
- Easy Frame Transformation
- Easy Create Index List
- Easy Create Number List
- Easy Create Poses

- Easy Create Quaternions
- Easy Create String List
- Easy Create Vector3Ds
- Easy Point Poses to Reference Position
- Estimate Point Cloud Edges by 2D Method
- Estimate Point Cloud Edges by 3D Method
- Evaluate Image Clarity
- Evaluate Results by HDevEngine
- Evaluate Results by JavaScript Engine
- Evaluate Depth Map Fluctuation
- evaluate-2d-poses.pdf
- External Interface
- Extract 3D Points in Cuboid
- Extract 3D Points in Cylinder
- Extract 3D Points in 3D ROI
- Extract Empty Regions in Depth Map within 3D ROI
- Fill Holes
- Filter by Labels
- Filter Masks by Poses
- Remove Poses outside Bin
- Filter
- Find Corners
- Find Correspondence between Poses and Offsets
- Find Hole 2D
- Find 2D Contour at Specified Inner-Outer Level
- Fit Circle
- Fit Line
- Flip Poses’ Axes
- From Actual Dimensions to Dimensions in Pixels
- From Cloud (XYZ-Normal) to Cloud (XYZ-RGB)
- From Cloud (XYZ-RGB) to Color Image
- From Depth Map to Point Cloud
- Convert Disparity Image to Depth Map
- From NumberList to Size3DList
- From Numbers to Variants
- From PoseList to Matrix4D
- From PoseList to PoseLists

- Convert Quaternions to Euler Angles in Poses
- From Shape2DList to Pose2DList
- From Variant to Variants
- From Variants to Labels
- From Variants to Numbers
- From Variants to Variant
- From 3D Poses to 2D Poses
- Generate Cloud Wall
- Generate Cylinder Point Cloud Model
- Generate Discrete Poses Revolving around Reference Pose
- Generate Pick Points
- Generate Point Cloud Model
- Generate Ring Point Cloud
- Generate Pose/Offset
- Generate Rect Path
- generate-rectangle-candidates.pdf
- generate-rectangular-edge-templates-of-specified-sizes.pdf
- Generate Spiral Path
- Generate Test Image
- Generate Test Point Cloud
- Generate Path from Contour
- Generate Path Given Depth Map
- Generate Zigzag Path
- Get First Image from Image List
- Get Flange Pose List
- Get Highest Layer Regions in Depth Map
- Get Highest Layer Clouds
- Get Highest-Layer Points
- Get Highest-Layer Poses
- Get Highest Score Result
- Filter Ring Point Cloud List
- Group Data
- Group 2D Poses
- Histogram Matcher
- Image Brightness and Color Balancer
- Image Filtering
- Adjust Image

- Insert End Target and Send Motion Params
- Invalidate Depth Pixels outside 3D ROI
- Invert Poses
- Get Inverses of Quaternions
- Compare Z Values of Poses with Threshold
- Join Labels
- Keep Poses Distributed in Regular Polygon
- Label Mapping
- 3D Laser Profiler
- load-2d-templates.pdf
- Load Targets in Path and Apply Affine Transform
- Load 2D Path
- Locate Calibration Spheres
- Point Poses to Reference Positions
- Make Template
- Map Depth to RGB
- Map to Multiple Pick Points
- Extract Image Regions by Mask
- Mask Clustering
- Filter Masks
- Mask Gridding
- Mask Logical Operation
- Measure Angles Segments to Segments
- Measure Circles
- Measure Distances Circles to Circles
- Measure Distances Circles to Segments
- Measure Distances Points to Circles
- Measure Distances Points to Points
- Measure Distances Points to Segments
- Measure Distances Segments to Segments
- Measure Gap Width
- Measure Height Difference Point to Point
- Measure Height Differences Points to Baseline
- Measure Height Differences Points to Plane
- Measure Longest Line Segment
- Measure Plane Height Differences Along Direction Parallel To Axes
- Measure Result

- Merge Point Cloud Lists
- Merge Data
- Merge Depth Maps
- Merge Label List
- Merge LineSegment Lists
- Merge Point Clouds with Similar Heights
- Merge Point Clouds
- Merge Pose Lists
- Merge Mask Images
- Morphological Transformations
- Move Point Cloud along Specified Direction
- Move Poses to Point Cloud Surfaces along Z-Axis
- Normal Estimation
- Notify
- Number Scaling
- Numeric Operation
- OCR
- Offset Poses in Cylinder
- Orthographic Projection Inverse
- Orthographic Projection
- Pack
- pallet-information-recognition.pdf
- Path Planning
- Periodic Trigger
- Perspective Transformation
- picking-relaxation.pdf
- Plan Object Poses at Placement
- Point Axes of Poses to Given Direction
- Point Cloud Clustering
- Point Cloud Filter
- Point Cloud Shape Detector
- Point Filter
- Adjust Poses
- Pose Filter
- Rotate Poses by Quaternion Vectors in Object Frames
- Transform Poses by Matrix in Object Frames
- Check Pose Repeatability by Statistics

- Predict Object Poses (Sim2Pick)
- Predict Pick Points (Any Object)
- Predict Pick Points V2
- Procedure Out
- Procedure
- Process 2D Shapes
- Project Points onto Plane
- Project 3D Point Cloud to 2D Image
- Read Bar Code
- Read Images V2
- Read Images
- Read Object Dimensions
- Read Point Cloud V2
- Read Point Cloud
- Read Poses from File
- Read QR Code
- Read STL
- Read 3D ROI Center
- Record Criterion Pose and Calc Transformation
- Recover Scaled Images in 2D ROI
- Rectify Image by Frame Transformation
- Rectify Ring Object Poses
- Remove All Overlapping Poses
- Remove Cloud Points from Point Cloud
- Remove Noise in Masks
- Remove Overlapped Objects V2
- Remove Overlapped Objects
- Remove Overlapped Polygons
- Remove Cloud Points from Point Cloud
- Remove Polygons outside Mask
- Reorder According to Given Indices
- Reorder by Index List
- Repeat Data and Concatenate Copies
- Replace Elements in List
- Reverse List
- Rotate Axis to Minimize Included Angle to Reference Direction
- Rotate Images by Specified Poses

- Rotate Poses around Axis by Angle
- Rotate Poses around Given Axis
- Rotate Poses Locally
- Rotate Poses to Directions Freely
- Rotate Poses’ Axes to Specified Directions under Symmetry Constraints
- Rotate Poses’ Axes to Specified Directions
- Save Images
- Save Regions around Poses as 3D ROIs
- Save Result to XML File
- Save Results to File
- Save Path Targets
- Save Step Parameters to File
- Scale Image in 2D ROI
- Segment Depth Map
- Send Point Cloud to External Service
- Set Pose Quaternions
- Set Pose Translation
- Show Images
- Show Point Clouds and Poses
- Smooth Depth Map
- Smooth Path
- Sort List and Output Index List
- Sort and Stratify
- Sort Point Clouds
- Sort Poses by Input Scores
- Sort Poses by XYZ Values
- Sort 2D Poses
- Sort 3D Poses
- Straighten Ring Shaped Image Region
- Stratify Values by Thresholds
- Subtract Real Numbers
- Suppress Neighboring Poses with Low Scores (NMS)
- symmetry-type.pdf
- Template Matching
- Test
- Path Target Matching
- Transform Image

- Align Plane Point Clouds
- Transform Point Clouds for Truss
- Transform Point Clouds
- Transform Poses for Truss
- Transform Poses
- Translate Poses along Customized Direction by Distances
- Translate Poses along Given Direction
- Translate Poses by Direction and Distance
- Translate Poses by Vector3D
- Translate Poses Locally by Distances
- Trigger Control by Flag
- Trigger
- Trim Input List
- Trim Pose List
- Unpack and Merge Data
- Unpack Data
- Validate 2D Poses by Mask
- validate-and-calc-parameter-compensation.pdf
- Validate Box Dimensions
- Validate Box Object Masks
- Validate Existence of Poses in 3D ROI
- Validate Labels and Output Flags
- Validate Masks Containing 2D Poses
- Validate Point Clouds
- Validate Poses by Included Angles to Reference Direction
- Validate Underlying Poses
- Extract Regions of Large Normal Deviations
- Vector3D Arithmetic Operations
- Verify Pick Points
- Visualize Depth in Range
- Visualize Information on Image

Common Procedures

Mech-Vision software stores the commonly used Procedures in the Custom directory in the Step library. Please check the contents below to learn about the **detailed introduction to Procedures**.

- Large Workpiece Measurement
- Sort
- 3D Matching

- Apply Masks to Color Image
- Calc Color Image for Highest Layer
- Filter Out Point Clouds That Exceed The Limit
- Filter Out Poses Out Of ROI
- Point Cloud Preprocessing
- Save Images and Step Parameters

4.3.1. Convert Poses 2D to 3D According to Orthographic Projection

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step converts 2D poses generated based on the Orthographic Projection Step to 3D poses.

Usage Scenario

This Step is generally used for measurement and follows the Orthographic Projection Step. It converts the poses in the 2D image output from Orthographic Projection into 3D poses.

Input and Output

- Input:
  1. 2D poses generated based on Orthographic Projection
  2. 3D poses converted from 2D poses
  3. The minimum X, Y, and Z of all 3D coordinates in the point cloud
  4. Scale of Orthographic Projection
  5. Border width of masks generated based on Orthographic Projection
  6. Depth map corresponding to objects in the mask generated based on Orthographic Projection
- Output:
  1. Convert 2D Poses to 3D.

4.3.2. 2D Shape Matching

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step uses 2D shape matching methods to locate objects in images.

Usage Scenario

This Step is a general 2D shape matching Step. This Step has no fixed usage scenarios.

Input and Output

- Input:
  1. Scene image in which objects will be matched against templates
  2. Object templates
- Output:
  1. Match result masks
  2. Scores corresponding to the matching results

4.3.3. 3D Coarse Matching (Multiple Models)

Function

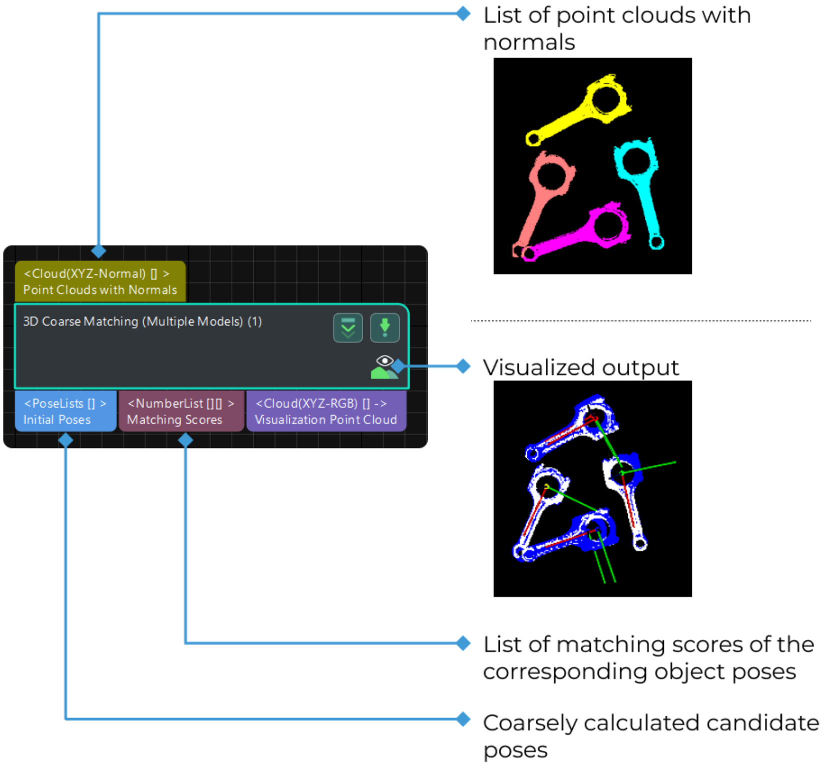
Use multiple models to roughly match objects in the scene, and output the coarsely calculated candidate poses of the target objects.

Usage Scenario

This Step calculates the original poses of objects in the scene by using multiple models. It is an extensive version of the 3D Coarse Matching Step and their parameter tuning methods are similar.

This Step should be used in multi-model scenarios to distinguish workpieces of different types. This Step is usually followed by 3D Fine Matching (Multiple Models) to obtain accurate poses.

Input and Output


List of point clouds with normals
<Cloud(XYZ-Normal) [] >
Point Clouds with Normals
3D Coarse Matching (Multiple Models) (1)
<PoseLists [] >
Initial Poses
<NumberList [][] >
Matching Scores
<Cloud(XYZ-RGB) [] ->
Visualization Point Cloud
Visualized output
List of matching scores of the corresponding object poses
Coarsely calculated candidate poses

Parameter Description

Model Settings

The path of model file and pick point file.

Model File (Required)

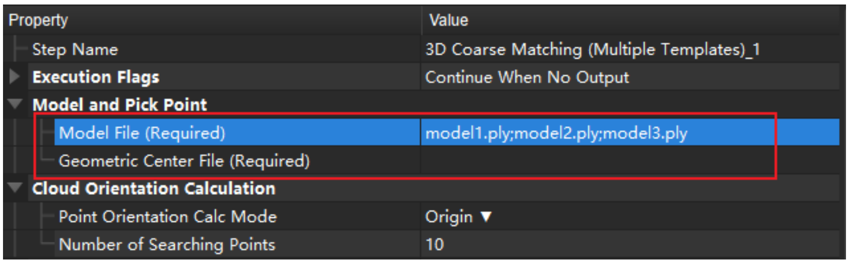
Default value: model.ply

Instruction: The path of 3D model file, the construction process can refer to the complete point cloud model stitching document. You can enter multiple file paths. Please use semicolons to divide different file paths.

Geo Center Point File (Required)

Instruction: The geometric center file in JSON format. You can enter multiple file paths. Please use semicolons to divide different file paths.

Example: Ensure that the files entered under each parameter are in the same order, meaning that the **Model File** has the same path order as the **Geometric Center File**, as shown in the figure below. Different files are separated by “;”.



Property	Value
Step Name	3D Coarse Matching (Multiple Templates)_1
Execution Flags	Continue When No Output
Model and Pick Point	
Model File (Required)	model1.ply;model2.ply;model3.ply
Geometric Center File (Required)	
Cloud Orientation Calculation	
Point Orientation Calc Mode	Origin ▼
Number of Searching Points	10

Cloud Orientation Calculation

Point Orientation Calc Mode

Default setting: Origin

List of options	Description
Origin	Use the original normal of the input point cloud directly.
StandardMode	Use the CPU to recalculate the normal direction of the input point cloud, which is recommended when the model does not have the normal direction. The k points nearest to the target were searched, and principal component analysis (PCA) is used to obtain the minimum feature vector as the normal direction of the point.
EdgeTangent	The tangent direction of the input edge point cloud is calculated as the normal direction. Objects whose outer contours are mirror images of each other can be distinguished. It is recommended to match edge point clouds of flat objects.

List of options	Description
EdgeNormal	Calculate the normal direction of the input edge point cloud, and use the tangential direction of the point as the normal direction, which is recommended for matching the edge point cloud of a flat object.

When using the **EdgeTangent** or **EdgeNormal** methods, ensure that each edge point cloud does not contain multiple objects; in other words, each object point cloud is separated.

Number of Searching Points

Default value: 10

Instruction: This parameter is used to adjust the number of adjacent points in the direction of the computed point, which is the value of K in **StandardMode** mode.

Processor Type

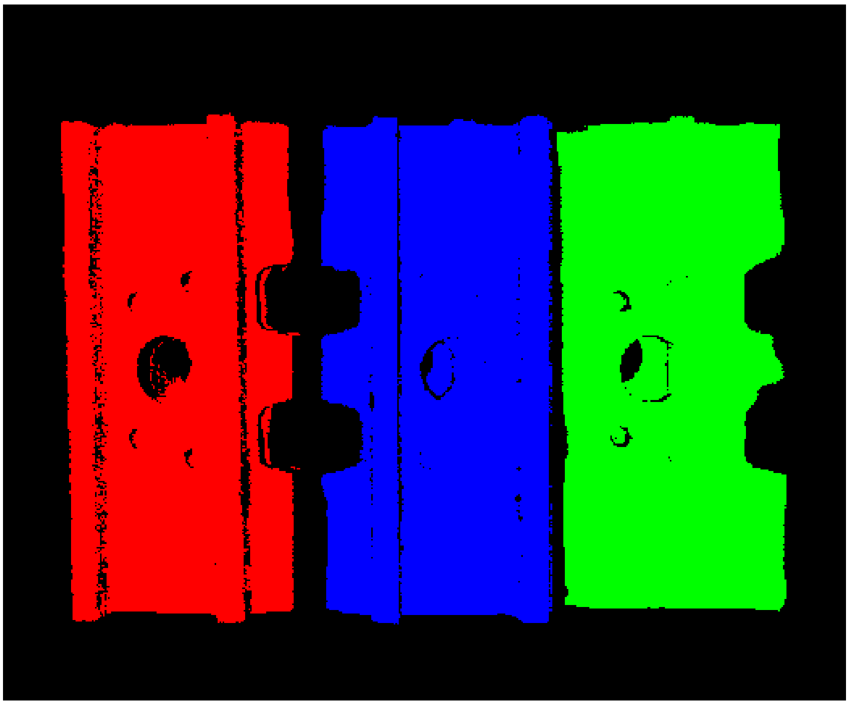
Default value: SurfaceMatchingEasyMode

List of Values: SurfaceMatchingEasyMode, SurfaceMatching

Instruction: The algorithm is categorized into two modes based on the ease of use. Both modes include the Result Visualization parameter group. As shown in the figure below, the front-side model and the back-side model will be used for the matching. Parameters in **SurfaceMatchingEasyMode** will be introduced first.

SurfaceMatchingEasyMode algorithm: The adjustable parameters module is Speed Controller and Output Settings.

SurfaceMatching algorithm: The adjustable parameters module is Sample Settings, Voting Settings, and Pose Verification Settings.



SurfaceMatchingEasyMode

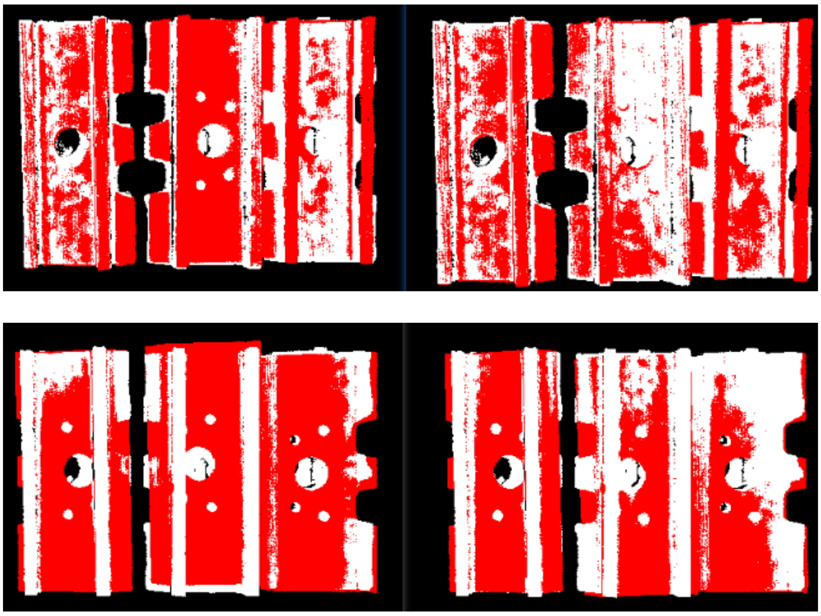
Speed Control

Main Speed Controller

Default value: 2

Instruction: This parameter is used to adjust the algorithm speed. When the value is increased, the algorithm speed becomes faster, but the matching accuracy decreases. Its effect is more obvious than **Secondary Speed Controller**. The valid range of this parameter is 1–6.

Example of adjustment: as shown in the figure below. The left figure shows the result when this value is 2 , and the right figure shows the result when this value is 6. It is obvious that the matching accuracy decreases after adjustment.

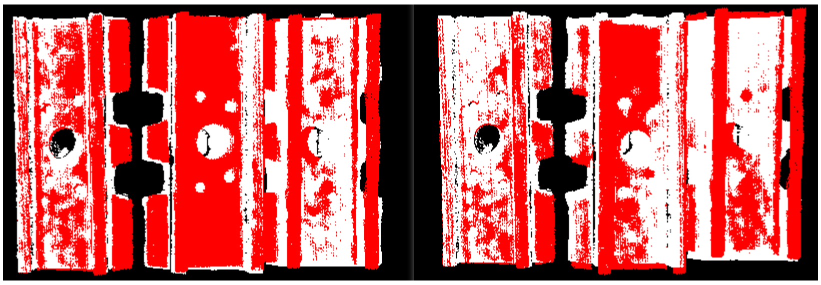


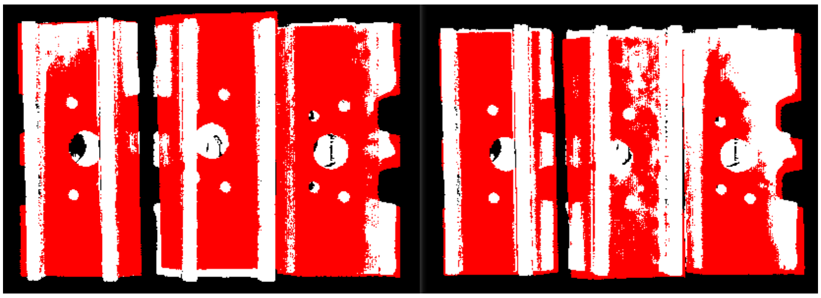
Secondary Speed Controller

Default value: 10

Instruction: This parameter is used to adjust the algorithm speed. When the value is increased, the algorithm speed becomes faster, but the matching accuracy decreases. Its effect is weaker than **Main Speed Controller**. The valid range of this parameter is 1–20.

Example of adjustment: as shown in the figure below. The figure on the left shows the result when this value is 10 , and the figure on the right shows the result when this Value is 15. It can be seen that the matching accuracy decreases after adjustment, but the influence is less than that of the main speed control parameters.





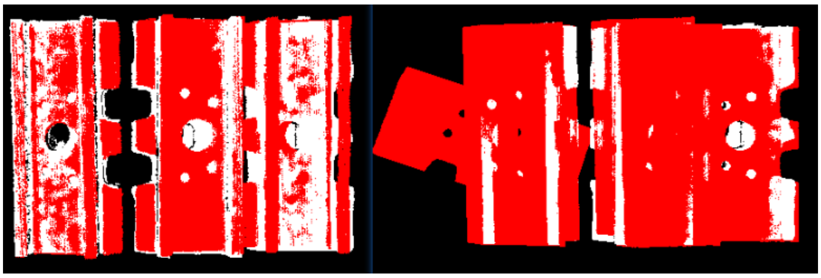
Output Settings

Maximum Number of Detected Poses in Each Point Cloud

Default value: 3

Instruction: This parameter is used to estimate the number of matching outputs per point cloud. The larger the value, the more matches are generated.

Example of adjustment: as shown in the figure below. The left picture shows the result when the parameter is 1, and the right picture shows the result when the parameter is 3.



SurfaceMatching

Sample Settings

Enable Automatic Downsampling

Default value: Selected

Instruction: This parameter is used to determine whether to use automatic downsampling. If it is selected, the sampling interval parameter of point cloud template will be automatically adjusted according to the expected points of the model after sampling.

Expected Point Number of Sampled Model

Default value: 1000

Instruction: This parameter is used to adjust the number of points of the sampling point cloud. It is effective when **Enable Automatic Downsampling** is selected, and the number of points of the point cloud is close to this value. The smaller this value is, the fewer points of sampling point cloud are, resulting in the lower accuracy of pose estimation.

Max Point Number of Sampled Model

Default value: 4000

Instruction: This parameter is used to set the maximum number of points in the point cloud model after downsampling. It sets an upper limit for the number of points in our point cloud model after downsampling. If the matching effect or matching speed is not ideal, this parameter is recommended to be increased.

Max Point Number of Sampled Scene

Default value: 3000

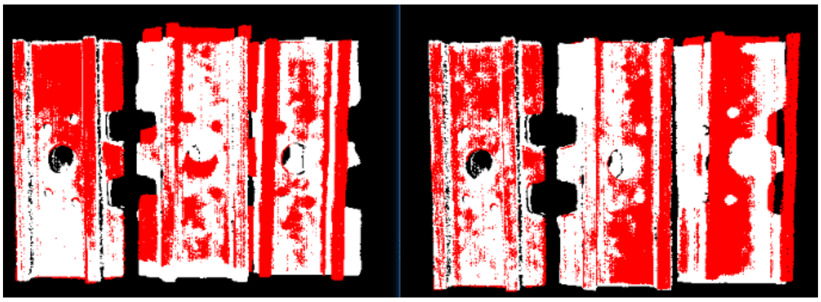
Instruction: This parameter is used to set the maximum number of points in the point cloud after the field point cloud downsampling. It sets an upper limit for the number of points in the field point cloud after the field point cloud downsampling. If the matching effect or matching speed is not ideal, this parameter is recommended to be increased.

Sampling Interval

Default value: 10.000 mm

Instruction: This parameter is used to adjust the maximum distance between points in the sampling point cloud. The unit is millimeters. When the sampling interval of point cloud model is smaller than the minimum sampling interval, the minimum sampling interval is used as the actual sampling interval. The larger the value is, the less point clouds are used for calculation after sampling, the lower the matching accuracy and the lower the algorithm execution time.

Example of adjustment: as shown in the figure below. The left picture shows the result when the parameter is 0.01, and the right picture shows the result when the parameter is 0.02.



Min Sampling Interval

Default value: 3.000 mm

Instruction: This parameter is used to calculate the sampling interval. The unit is millimeters. It is effective when the value of **Enable Automatic Downsampling** is selected. If the calculated sampling interval is smaller than this value, this value will be used as the actual sampling interval.

Voting Settings

Distance Quantification

Default value: 1

Instruction: The value for the quantification of the distance between points. As **Distance between Two Points = Distance Quantification × Sampling Interval**, the larger the value is, the larger the distance, and the less precise the result tends to be.

Angle Quantification

Default value: 60

Instruction: The value for the quantification of the angle between two vectors. As **Angle** between Two Vectors = 2 × 3.14 / Angle Quantification, increasing the parameter's value will reduce the matching accuracy.

Max Vote Ratio

Default value: 0.8

Instruction: This parameter sets the threshold for the proportion of the number of votes to the maximum number of votes. The number of votes corresponding to each pose will be obtained in the previous steps, and the maximum number of votes multiplied by this parameter will get a threshold. When the number of votes of a pose is greater than this threshold, the corresponding pose will be retained for clustering operation. The smaller the value, the more likely it is to find an accurate match, but the running time increases. The valid range of this parameter is 0–1.

Reference Point Step

Default value: 5

Instruction: This parameter is used to adjust the selection step of the reference point. The step size is taken as an interval sampling point from the point cloud. When the value is larger, the interval sampling points are fewer, and the execution speed is faster, but the matching accuracy is reduced.

Referred Point Step

Default value: 1

Instruction: This parameter is used to adjust the selection step of the referred point. The step size is taken as an interval sampling point from the point cloud. When the value is larger, the interval sampling points are fewer, and the execution speed is faster, but the matching accuracy is reduced.

- A **reference point** and a **referred point** make up a point pair. The larger the sampling step, the fewer referring points and referred points after downsampling, the fewer the point pairs, and the faster the execution.
- **Reference point** is the sampling point on the matching model. **Referred point** is the sampling point not on the matching model.

Clustering Settings

Cluster Ratio

Default value: 0.1

Instruction: This parameter is used to adjust the proportion of the number of poses used for clustering to the total computed poses. Any pose will be given a score during the calculation, and all poses will be sorted according to the score. This parameter determines how much of the pose is used for clustering, A value of 0.1 means that the top 10% pose is taken as the pose for clustering. The larger the value, the more likely it is to find an accurate match, but the running time increases accordingly.

Threshold of Angle Difference

Default value: 15

Instruction: This parameter is used to adjust the size of the Angle increment in the clustering process. In the final calculation result, the same object may calculate multiple poses, which determines the increment of the Angle parameter when the poses with very close parameters are fused. The larger the parameter is, the pose with large Angle difference will be fused into the final result, and the matching accuracy will decrease.

Threshold of Distance Difference

Default value: 0.02

Instruction: This parameter is used to adjust the size of the Distance increment in the clustering process. In the final calculation result, the same object may calculate multiple poses, which determines the increment of the Distance parameter when the poses with very close parameters are fused. The larger the parameter is, the pose with large Angle difference will be fused into the final result, and the matching accuracy will decrease.

Output First N Clusters with High Scores

Default value: 5

Instruction: This parameter is used to take the top N results with the highest score from the multiple matching results obtained after clustering adjustment as the final result.

Pose Verification Settings

Use Pose Verification

Default value: Selected

Instruction: This parameter determines whether pose validation is used. When the parameter is selected, all cluster parameters are invalid. Pose validation and clustering are two different methods for verification and screening of final matching results, which cannot be used simultaneously.

Marked Margin

Default value: 1

Instruction: This parameter is used to control the size of the verification area during pose verification. A single voxel is a unit. When the value is increased, the mark area used to verify the pose becomes larger, and more points are included to verify the final result, thus reducing the matching accuracy.

Voxel Length

Default value: 3

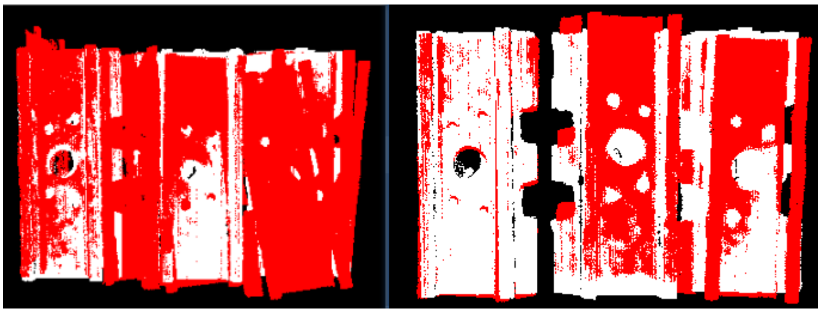
Instruction: The space where the point cloud is located is divided into a 3D grid, and the parameter is the size of the smallest unit of the 3D grid. When the value is increased, the box selection range becomes larger and there are more selected points for pose verification. In this case, the algorithm speed becomes faster, but the matching accuracy decreases.

Maximum Number of Detected Poses in Each Point Cloud

Default value: 3

Instruction: For SurfaceMatching algorithm, this parameter has the same effect as for SurfaceMatchingEasyMode algorithm. Only results are compared here.

Example of adjustment: The left side of figure below is the result when the parameter value is 3, and the right side is the result when the parameter value is 1.



Results Visualization

Show Sampled Model Cloud

Default value: Unselected

Instruction: This parameter is used to display the downsampled point cloud model.

Show Sampled Scene Cloud

Default value: Unselected

Instruction: This parameter is used to display the downsampled field point cloud.

Show Matching Results

Default value: Selected

Instruction: This parameter is used to display the matched model and field point cloud.

4.3.4. 3D Coarse Matching V2

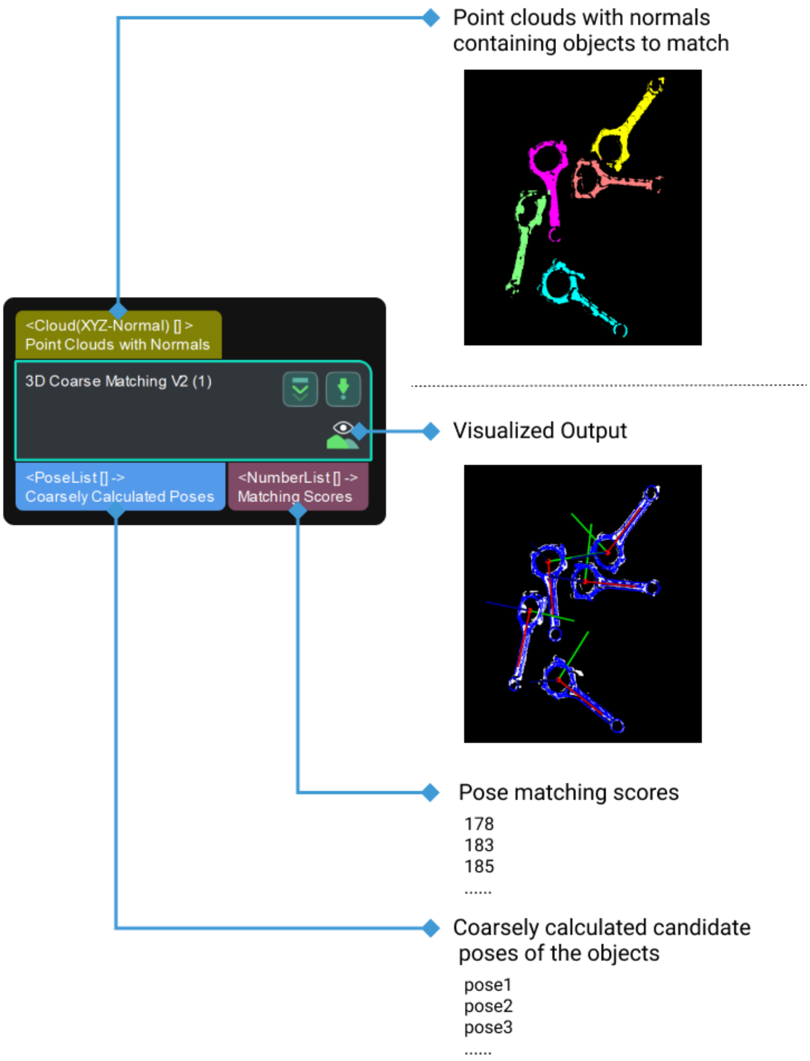
Function

This Step coarsely matches the point cloud model with the original point clouds and outputs the coarsely calculated candidate poses of the target objects.

Usage Scenario

This Step is usually used to detect the target object in the scene point cloud and obtain approximate candidate pose. This Step is usually followed by the 3D Fine Matching Step.

Input and Output


Point clouds with normals containing objects to match
<Cloud(XYZ-Normal) [] >
Point Clouds with Normals
3D Coarse Matching V2 (1)
Visualized Output
<PoseList [] ->
Coarsely Calculated Poses
<NumberList [] ->
Matching Scores
Pose matching scores
178
183
185
......
Coarsely calculated candidate poses of the objects
pose1
pose2
pose3
......

Parameters

Parameter Tuning Level

Parameter description: This parameter is used to select the Parameter Tuning Level for the Step.

Value list: Basic and Advanced.

Default value: Basic.

Basic Parameter Tuning Level

Model Settings

Model Selection

Parameter description: This parameter is used to select the point cloud model file in the model library (**project folder/resource/3d_matching**).

Tuning instruction: After making a point cloud model with the Matching Model and Pick Point Editor, click the button to the right of the parameter to expand the drop-down list to select or switch the point cloud model file.

Matching Method

Parameter description: This parameter is used to select the matching mode for 3D coarse matching.

Value list: Edge matching and Surface matching.

Default value: Edge matching.

Tuning instruction: Please select the mode according to the point cloud model in use. If the point cloud model represents the edge of the workpiece, select **Edge matching**; if the point cloud model represents the surface of the workpiece, select **Surface matching**.

Preprocessing Settings

Expected Point Count of Sampled Model

Parameter description: This parameter is used to set the expected number of points in the point cloud model after downsampling. The lower this value is, the fewer points there are in the downsampled point cloud, and the lower the matching accuracy is.

Default value: 200

Tuning recommendation: Set the value according to the actual requirement. Please refer to the tuning example for the application result.

Point Count Upper Threshold of Sampled Input Point Cloud

Parameter description: This parameter is used to set the maximum number of points in the scene point cloud after downsampling. If the number of points in the scene point cloud after downsampling exceeds this value, the scene point cloud will be ignored, and empty data will be output.

Default value: 100000

Tuning recommendation: Set the value according to the actual requirement. Please refer to the tuning example for the application result.

Pose Verification Settings

Expected Number of Detected Poses per Input Point Cloud

Parameter description: This parameter is used to set the expected number of matched poses for each input point cloud.

Default value: 3

Tuning recommendation: Set the value according to the actual requirement. Please refer to the tuning example for the application result.

Advanced Parameter Tuning Level

Model Settings

Model Selection

Parameter description: This parameter is used to select the point cloud model file in the model library (**project folder/resource/3d_matching**).

Tuning instruction: After making a point cloud model with the Matching Model and Pick Point

Editor, click the button to the right of the parameter to expand the drop-down list to select or switch the point cloud model file.

Model File

Parameter description: This parameter is used to select the point cloud model file.

Tuning instruction: Click 📁 to select a point cloud model file in PLY format.

Geo Center Point File

Parameter description: This parameter is used to select the geometric center file.

Instructions: Click 📁 to select a geometric center file in JSON format.

Matching Method

Parameter description: This parameter is used to select the matching mode for 3D coarse matching.

Value list: Edge matching and Surface matching.

Default value: Edge matching.

Tuning instruction: Please select the mode according to the point cloud model in use. If the point cloud model represents the edge of the workpiece, select **Edge matching**; if the point cloud model represents the surface of the workpiece, select **Surface matching**.

Preprocessing Settings

Expected Point Count of Sampled Model

Parameter description: This parameter is used to set the expected number of points in the point cloud model after downsampling. The lower this value is, the fewer points there are in the downsampled point cloud, and the lower the matching accuracy is.

Value range: 20–3000

Default value: 200

Tuning recommendation: Set the value according to the actual requirement. Please refer to the tuning example for the application result.

Point Count Upper Threshold of Sampled Input Point Cloud

Parameter description: This parameter is used to set the maximum number of points in the scene point cloud after downsampling. If the number of points in the scene point cloud after downsampling exceeds this value, the scene point cloud will be ignored, and empty data will be output.

Value range: 100–100000

Default value: 100000

Tuning recommendation: Set the value according to the actual requirement. Please refer to the tuning example for the application result.

Number of Neighboring Points

Parameter description: This parameter is used to set the number of neighboring points for calculating the orientation of each point. This parameter is only available in the Edge matching mode.

Value range: 1–100

Default value: 30

Voting Settings

Upper Limit of Point Pair Count per Feature

Parameter description: This parameter specifies the upper limit of the number of point pairs contained in each feature during the model analysis. The lower this value is, the faster the execution will be, but the accuracy is also lower.

Default value: 50

Distance Quantification

Parameter description: This parameter is used to quantify the distance between points in the object point cloud. DistanceInterval = DistanceQuantification × SamplingInterval. The greater this value is, the less accurate the matching result tends to be.

Default value: 1.0000

Angle Quantification

Parameter description: This parameter is used to quantify the angle between the normals of the points in the object point cloud. AngleInterval = 360°/AngleQuantification. The greater this value is, the smaller the angle interval will be, and therefore the higher the accuracy of the matching result is, but a point cloud of higher quality is required.

Default value: 60

Vote Ratio Lower Threshold

Parameter description: Poses with scores higher than the HighestScoretheVote × VoteRatioLowerThreshold will go through pose verification. The lower this value is, the more poses are used for pose verification, and therefore the more likely an exact match will be found, but the Step will also take longer to execute.

Default value: 0.80

Referring Point Sampling Step

Parameter description: This parameter is used to adjust the step size to downsample the referring point. The step size is taken as an interval sampling point from the point cloud. When the value is larger, the interval sampling points are fewer, and the execution speed is faster, but the matching accuracy is reduced.

Default value: 5

Referred Point Sampling Step

Instruction: This parameter is used to adjust the step size to downsample the referred point. The step size is taken as an interval sampling point from the point cloud. When the value is larger, the interval sampling points are fewer, and the execution speed is faster, but the matching accuracy is reduced.

Default value: 1

- **Referring point** is the sampling point on the point cloud model.
- **Referred point** is the sampling point not on the point cloud model.

A referring point and a referred point make up a point pair. The larger the sampling step is, the fewer referring points and referred points there are after downsampling, the fewer the number of point pairs is, and the faster the execution is.

Pose Filtering Settings

Use Distance NMS

Parameter description: After this feature is enabled, candidate poses whose distances to the selected poses are less than 0.1 times the diameter of the object will be filtered out.

Default value: Enabled.

Tuning instruction: Please refer to the tuning example for the application result.

Filter Poses by Model Rotation Angle

Instruction: In the Edge Matching mode, the point cloud model may need to be rotated by a certain angle to match with the scene point cloud, and the posed will be filtered by the point cloud model's rotation angle. When the model's rotation angle exceeds the Angle Difference Upper Threshold, the pose will be filtered out.

Default value: Enabled.

Angle Difference Upper Threshold

Instruction: See the tooltip of Filter Pose by Model Rotation Angle.

Default value: 135.000°

Pose Verification Settings

Expected Number of Detected Poses per Input Point Cloud

Parameter description: This parameter is used to set the expected number of matched poses for each input point cloud.

Default value: 3

Tuning recommendation: Set the value according to the actual requirement. Please refer to the tuning example for the application result.

Voxel Length Generation Strategy

Parameter description: This parameter is used to select the strategy to generate the voxel length.

Value list: Auto and Manual.

Default value: Auto

Tuning recommendation: **Auto** is recommended for new users. If **Manual** is selected, you will need to set the **Voxel Length**.

Voxel Length Lower Limit

Parameter description: This parameter is used to set the minimum voxel length in millimeters. Default value: 1.000 mm

Voxel Length Upper Limit

Parameter description: This parameter is used to set the maximum voxel length in millimeters. Default value: 15.000 mm

If you divide the space occupied by the point cloud into a 3D grid, each minimum cube in this grid is a voxel. **Voxel length** is the length of the cube edge.

Result Visualization

Visualization Options

Parameter description: This parameter is used to select the item that you want to visualize.

Value list: Sampled Model, Sampled Scene, Registration Result, and Points Used in Pose Verification.

Default value: Registration Result.

Tuning instruction: Set the value according to the actual requirement. Please refer to the tuning example for the application result.

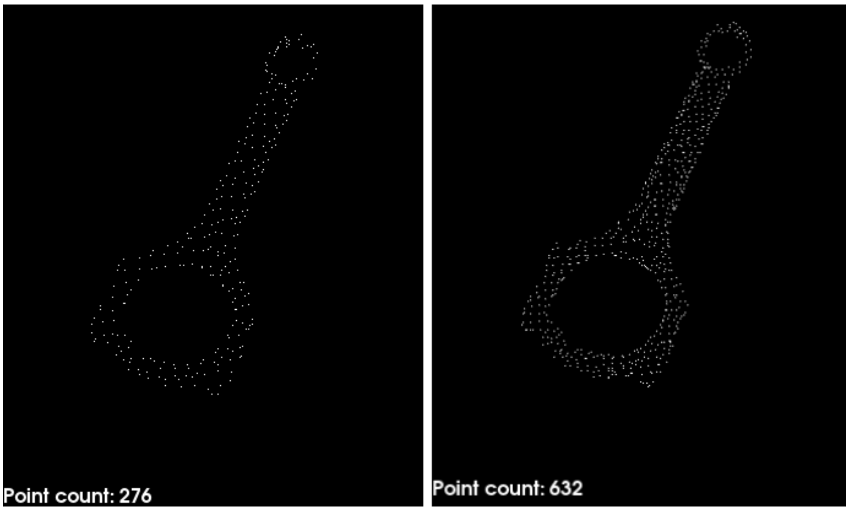
Tuning Examples

Examples in the Basic Parameter Tuning Level

Expected Point Count of Sampled Model

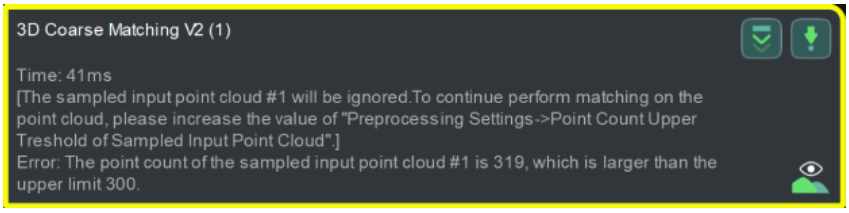
In the case of a connecting rod picking project, when the **Expected Point Count of Sampled Model** is set to 300 and 600, the corresponding point cloud models after downsampling are shown in the left and right figures below, respectively.

As you can see in the figures, the points in the left point cloud are sparse while the points in the right point cloud are dense.



Point Count Upper Threshold of Sampled Input Point Cloud

When the number of points in the input point cloud exceeds the set value, a message as shown below will appear.



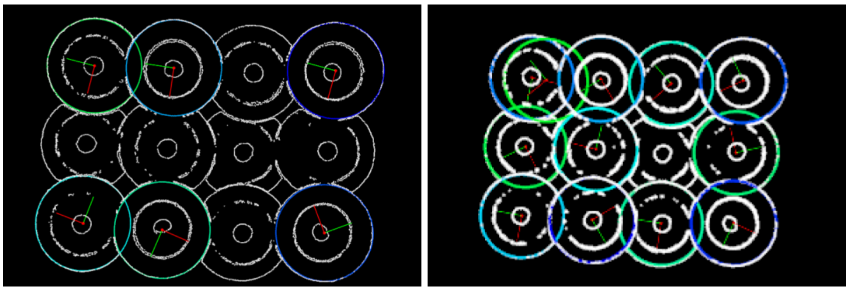
3D Coarse Matching V2 (1)

Time: 41ms
[The sampled input point cloud #1 will be ignored.To continue perform matching on the point cloud, please increase the value of "Preprocessing Settings->Point Count Upper Treshold of Sampled Input Point Cloud".]
Error: The point count of the sampled input point cloud #1 is 319, which is larger than the upper limit 300.

Expected Number of Detected Poses per Input Point Cloud

When the **Expected Number of Detected Poses per Input Point Cloud** is set to 6 and 12, the numbers of pose results output by this Step are shown in the left and right figures below, respectively.

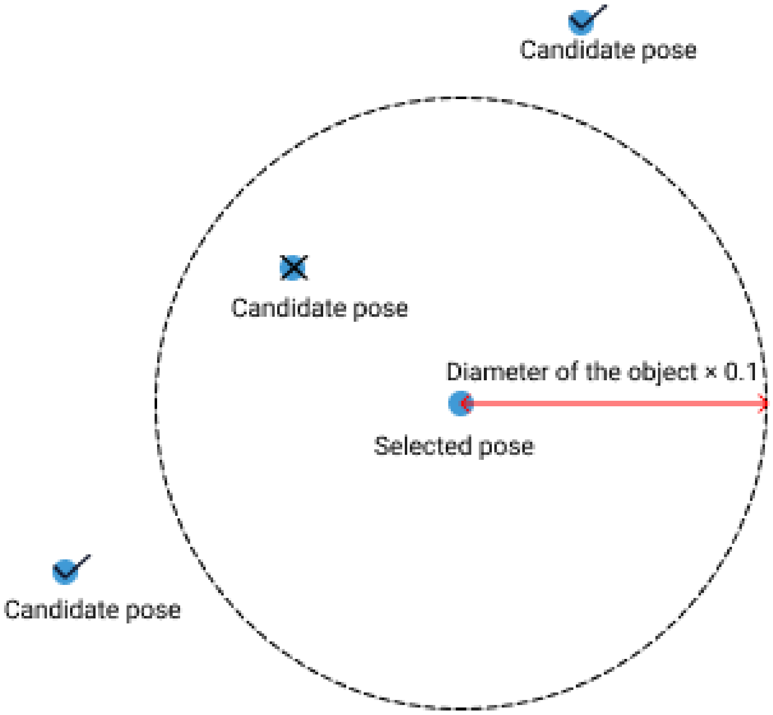
There are 6 pose results in the left figure while there are 12 pose results in the right figure.



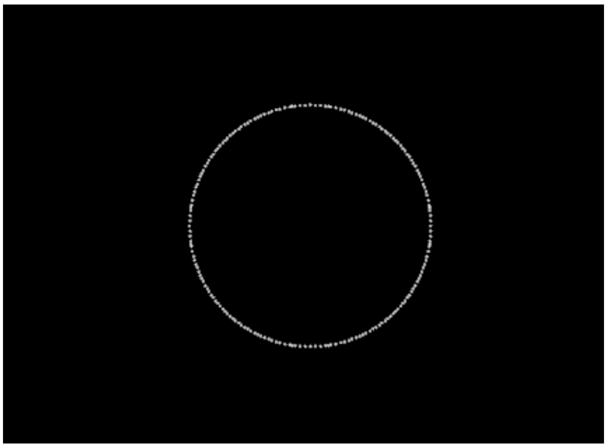
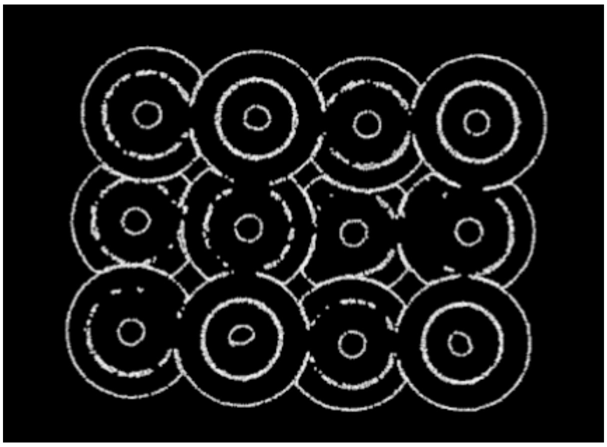
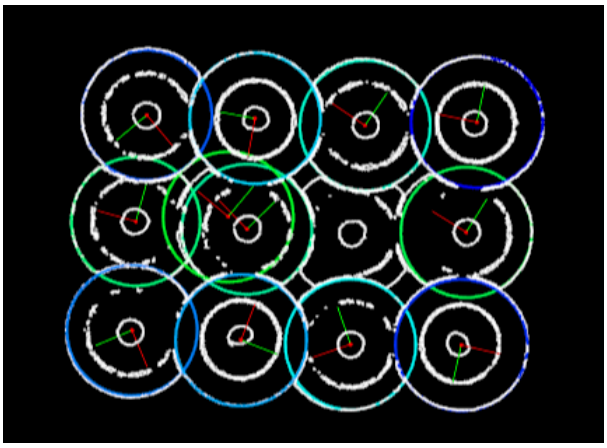
Examples in the Advanced Parameter Tuning Level

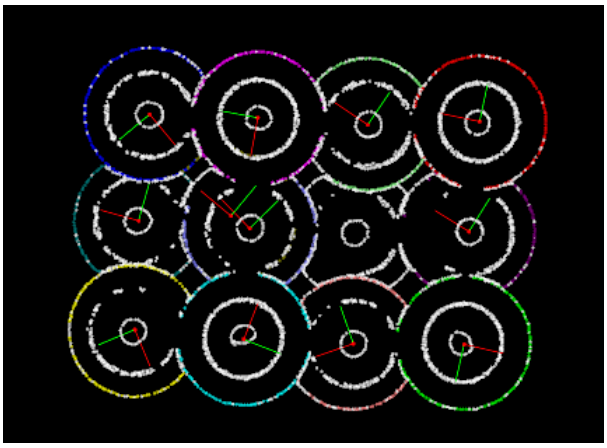
Use Distance NMS

As shown in the figure below, after this feature is enabled, candidate poses whose distances to the selected poses are less than 0.1 times the diameter of the object will be filtered out.



Result Visualization

Visualization Options	Example
Sampled Model	
Sampled Scene	
Registration Result	

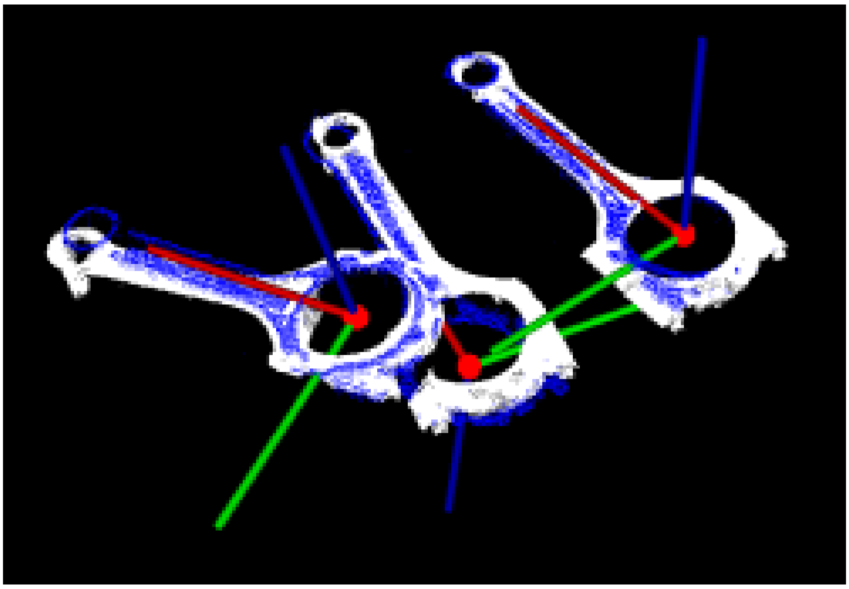
Visualization Options	Example
Points Used in Pose Verification	

4.3.5. 3D Coarse Matching

Function

Roughly match the model point cloud with the original point cloud and output approximate candidate poses of the target candidates.

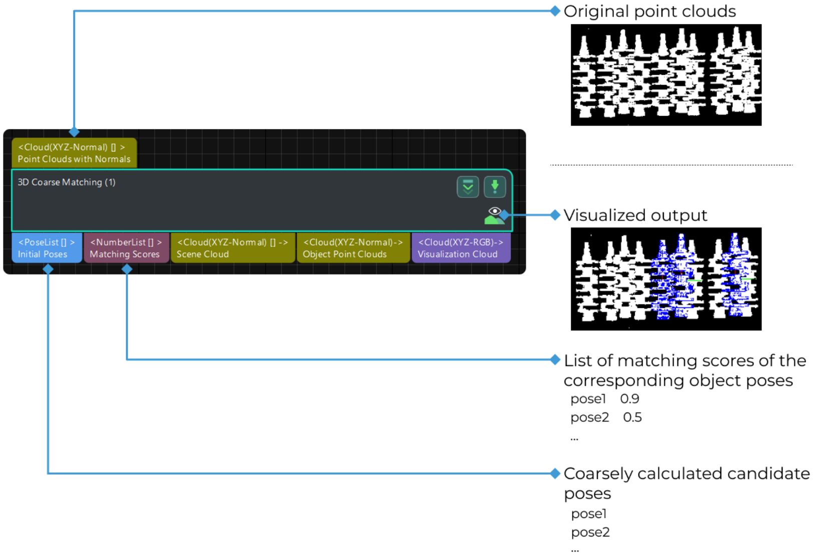
Refer to Generate Point Cloud Model for the creation of a point cloud template.)



Usage Scenario

This Step is usually used to detect the target object in the scene point cloud and obtain approximate candidate poses. This Step is usually connected to the 3D Fine Matching Step.

Input and Output



Original point clouds

Visualized output

List of matching scores of the corresponding object poses

pose1 0.9

pose2 0.5

...

Coarsely calculated candidate poses

pose1

pose2

...

Parameters

Model Settings

Model Selection

Instructions: Click the drop-down button to select and apply a point cloud template and its corresponding geometric centers.

Model file

Instructions: Click 🗀 to select a point cloud template file in PLY format. You can use Matching Model and Pick Point Editor to easily generate, edit, and save point cloud templates and the corresponding geometric centers/pick points.

Geo Center Point File

Instructions: Click 🗀 to select a geometric center file in JSON format. You can use Matching Model and Pick Point Editor to easily generate, edit and save point cloud templates and the corresponding geometric centers/pick points.

Cloud Orientation Calculation

Point Orientation Calc Mode

Default setting: Origin

List of options	Description
Origin	Use the original normal of the input point cloud directly.
StandardMode	Use the CPU to recalculate the normal direction of the input point cloud, which is recommended when the model does not have the normal direction. The k points nearest to the target were searched, and PCA (principal component analysis) is used to obtain the minimum feature vector as the normal direction of the point.

EdgeTangent	The tangent direction of the input edge point cloud is calculated as the normal direction. Objects whose outer contours are mirror images of each other can be distinguished. It is recommended to match edge point clouds of flat objects.
EdgeNormal	Calculate the normal direction of the input edge point cloud, and use the tangential direction of the point as the normal direction, which is recommended for matching the edge point cloud of a flat object.

Number of Searching Points

Default value: 10

Instruction: This parameter is used to set the number of adjacent points in the normal direction of the point, which is the value of K in **StandardMode** mode.

Processor Type

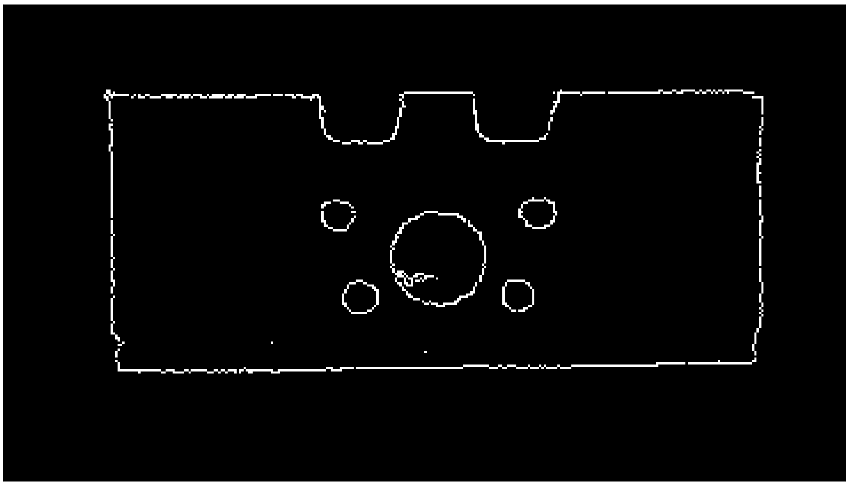
Default value: SurfaceMatchingEasyMode

List of Values: SurfaceMatchingEasyMode, SurfaceMatching

Instruction: There are two types of this algorithm. The algorithm type parameters are adjusted using the figure below as an example, starting with the adjustable parameters in the **SurfaceMatchingEasyMode** algorithm.

SurfaceMatchingEasyMode algorithm: The adjustable parameters module is Speed Controller and Output Settings.

SurfaceMatching algorithm: The adjustable parameters module is Sample Settings, Voting Settings, and Pose Verification Settings.



SurfaceMatchingEasyMode

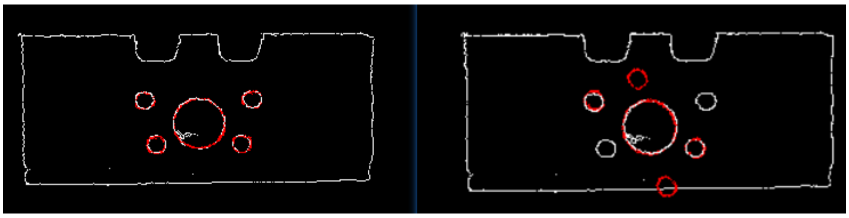
Speed Control

Main Speed Controller

Default value: 2

Instruction: This parameter is used to adjust the algorithm speed. When the value is increased, the algorithm speed becomes faster, but the matching accuracy decreases. Its effect is more obvious than **Secondary Speed Controller**. The valid range of this parameter is 1–6.

Example: As shown in the figure below. The left figure shows the result when this value is 2 , and the right figure shows the result when this value is 6. It is obvious that the matching accuracy decreases after adjustment.

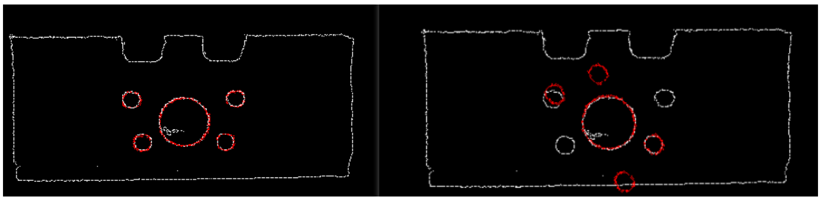


Secondary Speed Controller

Default value: 10

Instruction: This parameter is used to adjust the algorithm speed. When the value is increased, the algorithm speed becomes faster, but the matching accuracy decreases. Its effect is weaker than **Main Speed Controller**. The valid range of this parameter is 1–20.

Example: As shown in the figure below. The figure on the left shows the result when this value is 10, and the figure on the right shows the result when this value is 18. It can be seen that the matching accuracy decreases after adjustment, but the influence is less than that of the main speed control parameters.



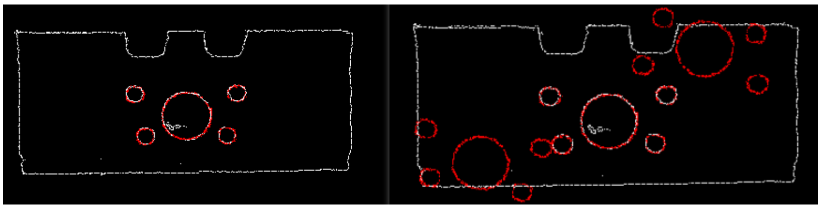
Output Settings

Maximum Number of Detected Poses in Each Point Cloud

Default value: 3

Instruction: This parameter is used to estimate the number of matching outputs per point cloud. The larger the value, the more matches are generated. The valid range of this parameter is from 1 to ∞.

Example of adjustment: As shown in the figure below. The left picture shows the result when the parameter is 1, and the right picture shows the result when the parameter is 3.



SurfaceMatching

Sample Settings

Enable Automatic Downsampling

Default value: Selected

Instruction: This parameter is used to determine whether to use automatic downsampling. If it is selected, the sampling interval parameter of point cloud template will be automatically adjusted according to the expected points of the model after sampling.

Expected Point Number of Sampled Model

Default value: 1000

Instruction: This parameter is used to adjust the number of points of the sampling point cloud. It is effective when the value of **Enable Automatic Downsampling** is selected, and the number of points of the point cloud is close to this value. The smaller this value is, the fewer points of sampling point cloud are, resulting in the lower accuracy of pose estimation.

Max Point Number of Sampled Model

Default value: 4000

Instruction: This parameter is used to set the maximum number of points in the point cloud model after downsampling. It sets an upper limit for the number of points in our point cloud model after downsampling. If the matching effect or matching speed is not ideal, this parameter is recommended to be increased.

Max Point Number of Sampled Scene

Default value: 30000

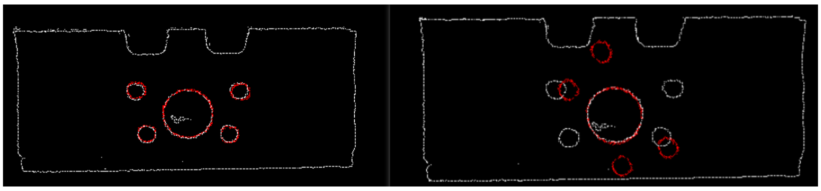
Instruction: This parameter is used to set the maximum number of points in the point cloud after the scene point cloud downsampling. It sets an upper limit for the number of points in the scene point cloud after the scene point cloud downsampling, thus limiting the time spent on running. The value of this parameter cannot be lower than the actual number of points sampled. If the maximum time spent on running needs to be limited, it is recommended to decrease the value of this parameter.

Sampling Interval

Default value: 10.000 mm

Instruction: This parameter is used to adjust the maximum distance between points in the sampling point cloud. The unit is millimeters. When the sampling interval of point cloud model is smaller than the minimum sampling interval, the minimum sampling interval is used as the actual sampling interval. The larger the value is, the less point clouds are used for calculation after sampling, the lower the matching accuracy and the lower the algorithm execution time.

Example of adjustment: As shown in the figure below. The left picture shows the result when the parameter is 0.01, and the right picture shows the result when the parameter is 0.04.



Min Sampling Interval

Default value: 3.000 mm

Instruction: This parameter is used to calculate the sampling interval. The unit is millimeters. It is effective when the value of **Enable Automatic Downsampling** is selected. If the calculated sampling interval is smaller than this value, this value will be used as the actual sampling interval.

Voting Settings

Distance Quantification

Default value: 1

Instruction: The value for the quantification of the distance between points. As DistanceInterval = DistanceQuantification × SamplingInterval, the bigger the value is, the less precise the result tends to be.

Angle Quantification

Default value: 60

Instruction: The value for the quantification of the angle between normals of points. As AngleInterval = 2 × 3.14 ÷ AngleQuantification, increasing the parameter's value will reduce the matching accuracy.

Max Vote Ratio

Default value: 0.8

Instruction: This parameter sets the threshold for the proportion of the number of votes to the maximum number of votes. The number of votes corresponding to each pose will be obtained in the previous steps, and the maximum number of votes multiplied by this parameter will get a threshold. When the number of votes of a pose is greater than this threshold, the corresponding pose will be retained for clustering operation. The smaller the value, the more likely it is to find an accurate match, but the running time increases. The valid range of this parameter is 0–1.

Reference Point Step

Default value: 5

Instruction: This parameter is used to adjust the selection step of the reference point. The step size is taken as an interval sampling point from the point cloud. When the value is larger, the interval sampling points are fewer, and the execution speed is faster, but the matching accuracy is reduced.

Referred Point Step

Default value: 1

Instruction: This parameter is used to adjust the selection step of the referred point. The step size is taken as an interval sampling point from the point cloud. When the value is larger, the interval sampling points are fewer, and the execution speed is faster, but the matching accuracy is reduced.

- A **reference point** and a **referred point** make up a point pair. The larger the sampling step, the fewer referring points and referred points after downsampling, the fewer the point pairs, and the faster the execution.
- **Reference point** is the sampling point on the matching model. **Referred point** is the sampling point not on the matching model.

Clustering Settings

Cluster Ratio

Default value: 0.1

Instruction: This parameter is used to adjust the proportion of the number of poses used for clustering to the total computed poses. Any pose will be given a score during the calculation, and all poses will be sorted according to the score. This parameter determines how much of the pose is used for clustering. A value of 0.1 means that the top 10% pose is taken as the pose for clustering. The larger the value, the more likely it is to find an accurate match, but the running time increases accordingly.

Threshold of Angle Difference

Default value: 15

Instruction: This parameter is used to adjust the size of the Angle increment in the clustering process. In the final calculation result, the same object may calculate multiple poses, which determines the increment of the Angle parameter when the poses with very close parameters are fused. The larger the parameter is, the pose with large Angle difference will be fused into the final result, and the matching accuracy will decrease.

Threshold of Distance Difference

Default value: 20.000 mm

Instruction: This parameter is used to adjust the size of the Distance increment in the clustering process. The unit is millimeters. In the final calculation result, the same object may calculate multiple poses, which determines the increment of the Distance parameter when the poses with very close parameters are fused. The larger the parameter is, the pose with large Angle difference will be fused into the final result, and the matching accuracy will decrease.

Output First N Clusters with High Scores

Default value: 5

Instruction: This parameter is used to take the top N results with the highest score from the multiple matching results obtained after clustering adjustment as the final result.

Pose Verification Settings

Use Pose Verification

Default value: Selected

Instruction: This parameter determines whether pose validation is used. If it is selected, all cluster parameters are invalid. Pose validation and clustering are two different methods for verification and screening of final matching results, which cannot be used simultaneously.

Marked Margin

Default value: 1

Instruction: This parameter is used to control the size of the verification area during pose verification. A single voxel is a unit. When the value is increased, the mark area used to verify the pose becomes larger, and more points are included to verify the final result, thus reducing the matching accuracy.

Voxel Length

Default value: 3

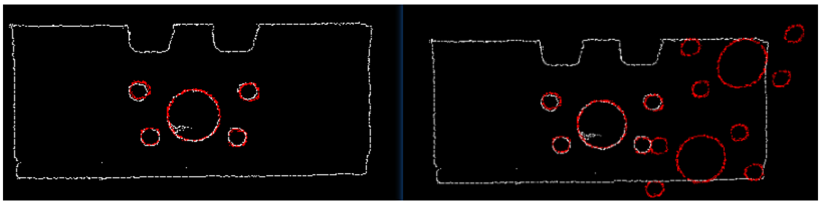
Instruction: The space where the point cloud is located is divided into a 3D grid, and the parameter is the size of the smallest unit of the 3D grid. When the value is increased, the box selection range becomes larger and there are more selected points for pose verification. In this case, the algorithm speed becomes faster, but the matching accuracy decreases.

Maximum Number of Detected Poses in Each Point Cloud

Default value: 3

Instruction: For **SurfaceMatching** algorithm, this parameter has the same effect as for **SurfaceMatchingEasyMode** algorithm.

Example of adjustment: The left side of the figure below is the result when the parameter value is 1, and the right side is the result when the parameter value is 3.



Results Visualization

Show Sampled Model Cloud

Default value: Unselected

Instruction: This parameter is used to display the downsampled point cloud model.

Show Sampled Scene Cloud

Default value: Unselected

Instruction: This parameter is used to display the downsampled field point cloud.

Show Matching Results

Default value: Selected

Instruction: This parameter is used to display the matched model and field point cloud.

4.3.6. 3D Fine Matching Lite

Function

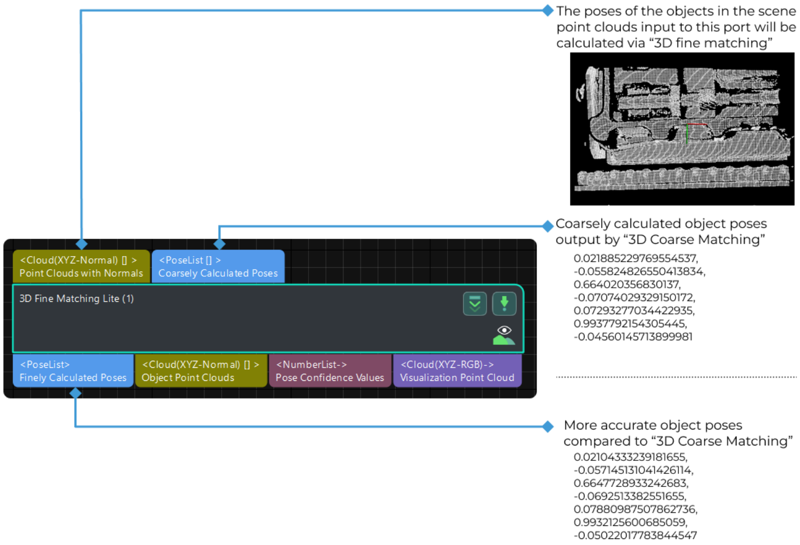
This Step is a lite version of 3D Fine Matching, in which the parameter configuration can be more intuitive.

This Step matches the point cloud and the original point cloud accurately and outputs the pose of the object.

Usage Scenario

The 3D Coarse Matching or 3D Coarse Matching V2 Step should be used before this Step to output the original candidate poses. Then, this Step will match the original poses more accurately and output the accurately calculated object poses.

Input and Output



The poses of the objects in the scene point clouds input to this port will be calculated via “3D fine matching”

Coarsely calculated object poses output by “3D Coarse Matching”

```
0.021885229769554537,
-0.055824826550413834,
0.664020356830137,
-0.07074029329150172,
0.07293277034422935,
0.9937792154305445,
-0.04560145713899981
```

More accurate object poses compared to “3D Coarse Matching”

```
0.02104333239181655,
-0.057145131041426114,
0.6647728933242683,
-0.0692513382551655,
0.07880987507862736,
0.9932125600685059,
-0.05022017783844547
```

Parameters

Parameter Tuning Level

Parameter description: This parameter is used to select the tuning level of the parameters. The parameters in different tuning levels are different.

Value list: Basic and Advanced

- Basic: basic parameter tuning level, which provides parameters that are frequently adjusted.
- Advanced: advanced parameter tuning level, which provides more adjustable parameters.

Default value: Basic.

Tuning recommendation: Select the parameter tuning level according to your need.

Basic Parameter Tuning Level

Model Settings

Model Selection

Parameter description: This parameter is used to select the point cloud model file.

Tuning instruction: Click ▼ and select an available point cloud model file in the drop-down list. Please refer to Matching Model and Pick Point Editor for how to make a point cloud model file.

Matching Method Settings

Matching Mode

Parameter description: This parameter is used to select between Edge Matching and Surface Matching modes.

Value list: Surface Matching and Edge Matching.

Surface Matching: Use the object’s surface point cloud for point cloud model matching. Edge Matching: Use the object’s edge point cloud for point cloud model matching.

Tuning recommendation: When the surface of the object has obvious recognizable features (such as crankshafts, rotors, and steel rods), it is recommended to use surface matching, and you should create a point cloud model that represents the surface features of the object. When the object is relatively flat and shows clear and regular edge features under the camera (such as panels, track pads, connecting rods, and brake discs), it is recommended to use edge matching, and you should create a point cloud that represents the edge features of the object.

> The point cloud model you select should correspond with the **Matching Mode** in use. For example, if the matching mode is Surface Matching, you should select a surface point cloud model in **Model Settings**.

Deviation Correction Capacity

Parameter description: This parameter is used to set the intensity of the deviation correction to the matching result from 3D Coarse Matching or 3D Coarse Matching V2. The greater the deviation correction capacity is, the more likely the coarsely matched poses can be corrected to the accurately matched poses. Please note that an excessive deviation correction capability may lead to a loss of matching accuracy.

Value list: Small, Medium, and Large.

Default value: Small (applicable to most cases).

Tuning recommendation: When there is a relatively large deviation of the matching result, please adjust the **Deviation Correction Capacity** first. If the deviation is too large to be corrected by this parameter, please adjust the parameters in the 3D Coarse Matching or 3D Coarse Matching V2 Step.

Operation Approach

Parameter description: This parameter specifies the operation approach of the matching process. The operation approach determines the relationship between the matching accuracy and matching speed. The higher the matching accuracy is, the longer the time required is.

Value list: HighSpeed, Standard, HighAccuracy, and ExtraHighAccuracy.

Default value: Standard.

Tuning recommendation: Please select the operation approach according to the actual requirement.

Validation Settings

Confidence Threshold

Parameter description: This parameter is used to determine whether the matching result is valid. If the confidence of the matching result is above the threshold, the matching result is valid. The higher the confidence value is, the more accurate the matching result is.

Default value: 0.300

Only Consider Visible Surface of Model

Parameter description: This parameter is used to determine whether to consider the visible part of the surface point cloud only in the matching process. You should set this parameter when the matching mode is **Surface Matching**. Once this parameter is selected, only the visible part of the surface point cloud model will be considered, which facilitates the calculation and produces a more reasonable matching confidence.

Default value: Unselected.

Tuning recommendation: It is recommended to select this parameter for object models that have parts not visible to the camera under a viewing angle, such as the complete model of a cylindrical object. For object models that are completely visible from the same viewing angle, such as the surface matching model of the upper side of a brake disc, this parameter can be ignored.

Output Settings

Maximum Number of Detected Poses in Each Point Cloud

Parameter description: This parameter sets the maximum number of output results in a single point cloud.

Default value: 1

Tuning instruction: If a single point cloud has multiple matching results, the results will be sorted in descending order according to the scores, and then the results with the highest scores will be output according to this parameter.

Advanced Parameter Tuning Level

Please refer to Advanced Parameter Tuning Level.

4.3.7. 3D Fine Matching (Multiple Models)

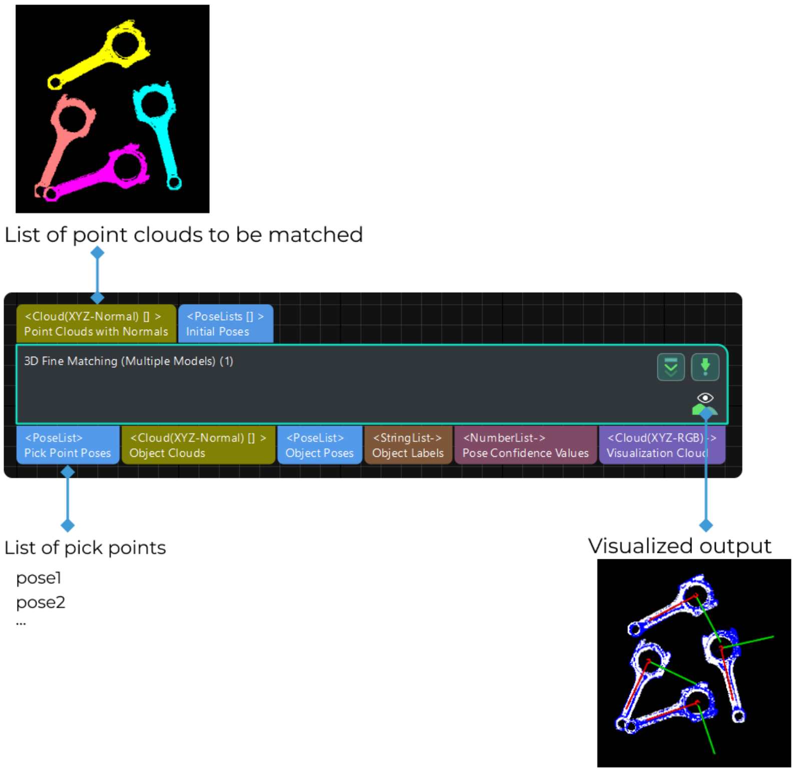
Function

Use multiple models to accurately match objects in the scene and output accurate pick points of the target objects.

Usage Scenario

This Step performs more accurate matching based on the initial candidate poses output by the Step 3D Coarse Matching (Multiple Models) and outputs accurate poses of the target object.

Input and Output



Parameter Description

Parameter Tuning Level

Default value: Basic

List of values: Basic, Advanced

Instruction: Two different parameter tuning levels can be set:

- Basic: basic adjustment, with a small amount of adjustable parameters
- Advanced: advanced adjustment, with more adjustable parameters

Basic Mode

This chapter introduces the adjustable parameters in Basic Mode. These parameters can also be configured in Advanced Mode.

Model Settings

Model File (Required)

Default value: model.ply

Instruction: The path of 3D model file. The construction process can refer to the complete point cloud model stitching document. You can enter multiple file paths. Please use semicolons to divide different file paths.

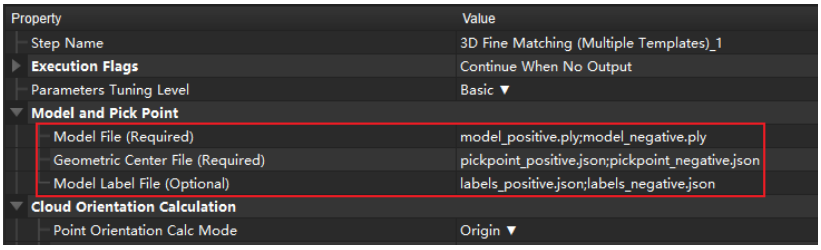
Geo Center Point File (Required)

Instruction: The geometric center file in JSON format. You can enter multiple file paths. Please use semicolons to divide different file paths.

Model Label File (Optional)

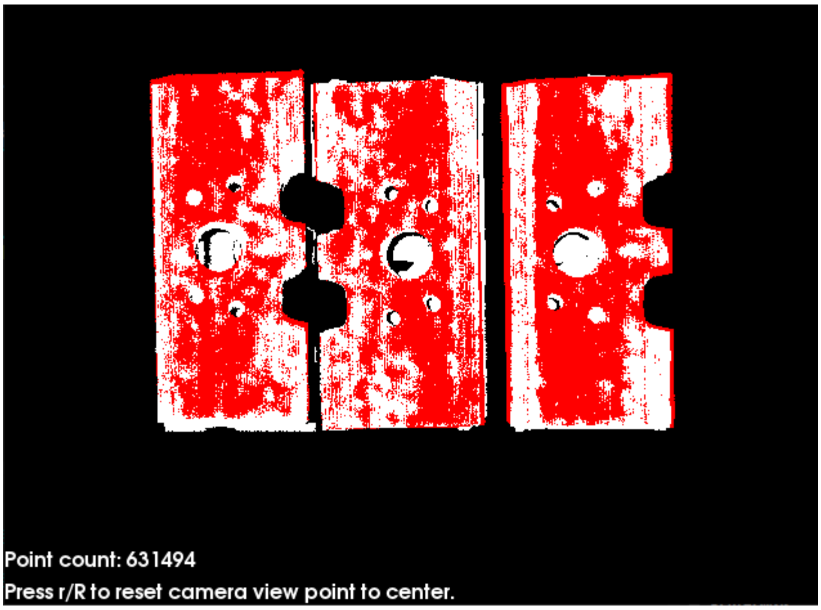
Instruction: The model label file in JSON format. You can enter multiple file paths. Please use semicolons to divide different file paths.

Example: Ensure that the order of the file paths is consistent, i.e., if the file path of an object with front side facing up is entered first in the **Model Label File**, the corresponding front side object file path should also be entered before the back side one in both **Model File** and **Geometric Center File**, as shown below.



Property	Value
Step Name	3D Fine Matching (Multiple Templates)_1
Execution Flags	Continue When No Output
Parameters Tuning Level	Basic ▼
Model and Pick Point	
Model File (Required)	model_positive.ply;model_negative.ply
Geometric Center File (Required)	pickpoint_positive.json;pickpoint_negative.json
Model Label File (Optional)	labels_positive.json;labels_negative.json
Cloud Orientation Calculation	
Point Orientation Calc Mode	Origin ▼

After inputting file paths correctly, the output point cloud image is shown as below. Both the front and back point cloud models of the object can match the scene point cloud well.



Point count: 631494
Press r/R to reset camera view point to center.

Cloud Orientation Calculation

Point Orientation Calc Mode

Default value: Origin

Instruction: This parameter is used to specify the method to calculate the orientations of points in the point cloud. There are four methods as shown below. Please select one according to the actual situation of the project.

List of options	Instruction
Origin	Directly use the original normal direction of the input point cloud.

List of options	Instruction
StandardMode	Use the CPU to recalculate the normal direction of the input point cloud. It is recommended to use this mode when the model file doesn't have normal direction: Search for the k points closest to the target, and use principal component analysis (PCA) to obtain the minimum eigenvector as the point's normal direction.
EdgeTangent	The tangent direction of the input edge point cloud is calculated as the normal direction. This mode can be used when matching the edge point cloud of a flat object. There are two usage scenarios as follows:
Scenario 1: The two sides of the object need to be distinguished. The 3D Fine Matching (Multiple Models) Step can be used and there should be two model point cloud models corresponding to the two sides of the object respectively.	Scenario 2: The two sides of the object need NOT to be distinguished. In this case, the 3D Fine Matching Step with one point cloud model is applicable.
EdgeNormal	Calculate the normal at the edge of the point cloud model. This method is recommended when edge matching is performed on relatively flat objects.

When using the **EdgeTangent** or **EdgeNormal** methods, ensure that each edge point cloud does not contain multiple objects; in other words, each object point cloud is separated.

Number of Searching Points

Default value: 10

Instruction: This parameter is used to set the number of points in the vicinity of the target point to search for. The minimum value is 1.

Correspondence Settings

Default value: GMM

List of values: GMM, nearest-neighbor

Instruction: This parameter is used to select the local matching method to match the point cloud model with the scene point cloud. The point cloud model will move and rotate according to the set range to match the point cloud in the scene obtained in the previous step match.

Example: In most scenarios, GMM has better anti-interference ability and faster matching speed, so it is recommended to use this mode; in rare cases when GMM cannot meet the demand, the nearest point (nearest-neighbor) mode is worth trying.

GMM Mode

Matching Mode

Default value: Standard

List of values: HighSpeed, Standard, HighPrecision

Instruction: There are three matching modes. Select the appropriate one according to the project requirements:

HighSpeed: the fastest mode, but with lower precision. **Standard**: a relatively stable mode. **HighPrecision**: the highest precision mode, but with slower speed.

Number of Iterations

Default value: 30

Instruction: Iteration is the process of cyclically running for a specific instruction until the conditions are met. Number of Iterations is the number of repeated instructions in this process. The parameter set here is the maximum number of iterations. The larger this value, the more matching calculation times and the longer running time. However, larger Number of Iterations also gives better matching accuracy.

Standard Deviation

Default value: 3.000 mm

Instruction: The value here needs to match with the initial pose deviation. When the input initial pose deviation after the previous step (3D Coarse Matching) is greater, a larger **Standard Deviation** is required. In this case, the matching speed is faster.

Example: The standard deviation should correspond to the deviation of the initial pose. If the set value is greater than the deviation of the initial pose, the accuracy of the output result will be greatly reduced. Normally, the value of this parameter setting will not exceed 0.01 m.

Standard Deviation Update Step Number

Default value: 3

Instruction: During the matching process, the **Standard Deviation** gradually decreases until it reaches the value of **Minimum Standard Deviation**. The number of times that the standard deviation decreases in this process is the value of this parameter. For some cases with large standard deviation, the more times of the decrease, the more stable the entire decrease process. Under normal circumstances, this parameter does not need to be adjusted.

Minimum Standard Deviation

Default value: 0.001

Instruction: the minimum **Standard Deviation**, namely, the reached minimum value after the decrease process.

Apply Weights on Pairs

Default value: Unselected.

Instruction: If it is selected, the weight of the model will be used in the iteration of the algorithm, so it is more likely to match better.

Nearest-Neighbor Mode

Iteration Settings

Number of Iterations

Default value: 30

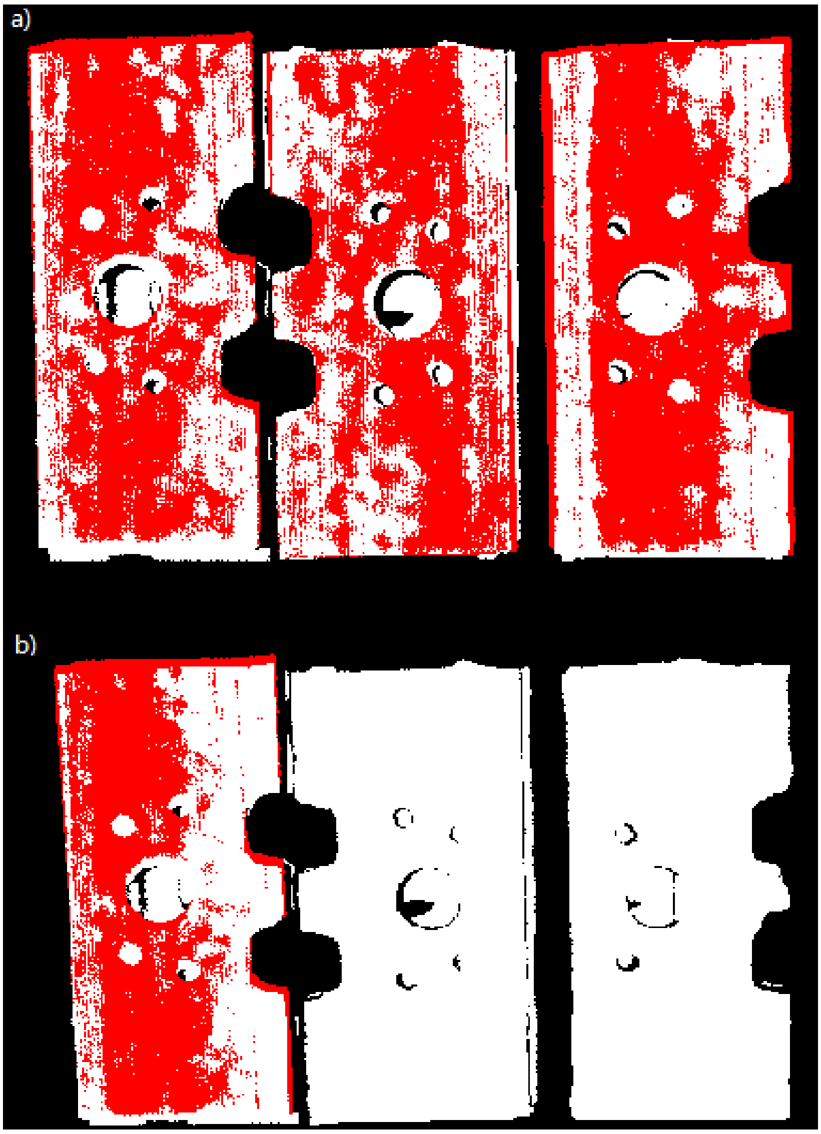
Instruction: Iteration is the process of cyclically running for a specific instruction until the conditions are met. **Number of Iterations** is the number of repeated instructions in this process. In the Nearest-Neighbor Mode, the iteration will stop when the calculation can satisfy the requirement of algorithm accuracy. The parameter set here is the maximum number of iterations. The larger this value, the more matching calculation times and the longer running time. However, larger Number of Iterations also gives better matching accuracy.

Nearest Points Search Radius

Default value: 10.000 mm

Instruction: This parameter is used to adjust the search radius of the nearest point (in mm), and the value setting should correspond to the input initial pose deviation. When the input initial pose deviation is large (the deviation between the scene point cloud and the point cloud model is large), this parameter needs to be increased; when the initial pose deviation is very small, this parameter can be adjusted to a small value. The minimum value that can be set for this parameter is 0.001. If the radius is less than this lower limit, some sparsely distributed points may not find the closest point, which could affect the output result.

Example: When the initial pose deviation is very large, and this parameter is set as the minimum value 0.001, the matching result will be incomplete, as shown by the (b) in the figure below. When this parameter is increased to 0.01, the search radius will become larger, and a complete matching result can be obtained, as shown by (a) in the figure below.



MSE Threshold

Default value: 0.001

Instruction: This parameter (Mean Square Error Threshold) is used to adjust the upper limit of the mean square error. This value is recorded after each iteration and used for comparison in subsequent iteration steps. When the mean square error of an iteration is less than the set value, it is considered that the iteration effect has met the requirements and the iteration will be terminated.

Window Size

Default value: 10

Instruction: This parameter refers to the number of consecutive iterations with small error fluctuations in the iterative optimization process, which can take a minimum value of 3. When this parameter is too small, a local optimal solution may appear, which affects the final matching accuracy. Example: If the window only contains the error of the results of three consecutive iterations, so the error trend can only be obtained by comparing the errors of the three iterations, which may lead to a local optimal solution (the error of the second iteration is less than the error of the first iteration and the first iteration) instead of a global optimal solution. According to the recommended value of the number of iterations which is 30, subsequent iterations with more significant error reduction may occur.

Show Corresponse

Default value: Unselected.

Instruction: If it is selected, the corresponding points will be displayed in each iteration.

Complex Object

Default value: Unselected.

Instruction: Selected if the target object is a complex-shaped object (not a simple geometric shape) or if the **Automatic Weight** function is required.

Automatic Weight

Default value: Unselected.

Instruction: If it is selected, the weight of iteration points will be calculated automatically. It is recommended to use this function for objects with poor coarse matching results.

Point Pair Rejection

Reject Pair

Default value: Unselected.

Instruction: This parameter decides whether to remove repeated point pairs. There are multiple point pairs in the process of searching for the nearest point, and sometimes the unwanted point pairs need to be filtered out. If this option is selected, this function will be used to remove them through configurating the following **N Sigma Threshold** and **Point Pair Angle Diff Thre** parameters; if not checked, these two parameters will not take effect during the matching process.

N Sigma Threshold

Default value: 1.0000

Instruction: This parameter is the multiplier of the standard deviation threshold of the distance between each points. The point pair whose distance is outside N sigma will be eliminated.

Point Pair Angle Diff Thre

Default value: 45°

Instruction: This parameter is the upper limit of the angle difference (in °) between the normals of two points in every matching point pair. If the angle difference is greater than this value, this point pair will be eliminated.

Apply Weights on Pairs

Default value: Unselected.

Instruction: If it is selected, the weight of the model will be used in the iteration of the algorithm, so it is more likely to match better.

Remove Repeat Correspond

Default value: Unselected.

Instruction: If it is selected, the function of removing repeating corresponding point pairs will be used, and the **Remove Repeat Correspond Type** parameter will be available and could take effect during the matching process. It is recommended to use this function when the point cloud model has more points than that of a single object in the scene.

Remove Repeat Correspond Type

Default value: MinDis

List of values: MinDis and GlobalMin

Instruction: This parameter is used to select the method of removing repeating corresponding point pairs. There are two methods in total:

MinDis: Calculate the distance between the repetitive point pairs. Then, keep the point pair with the smallest distance, and delete the remaining repetitive point pairs. More repeating corresponding point pairs will be eliminated in this method than the other. **GlobalMin**: Take the average value of the distance between the repetitive point pairs as a reference to find out the point pair whose distance is closest to the average value. Then, retain this point pair and eliminate others. Less repeating corresponding point pairs will be eliminated in this method than the other.

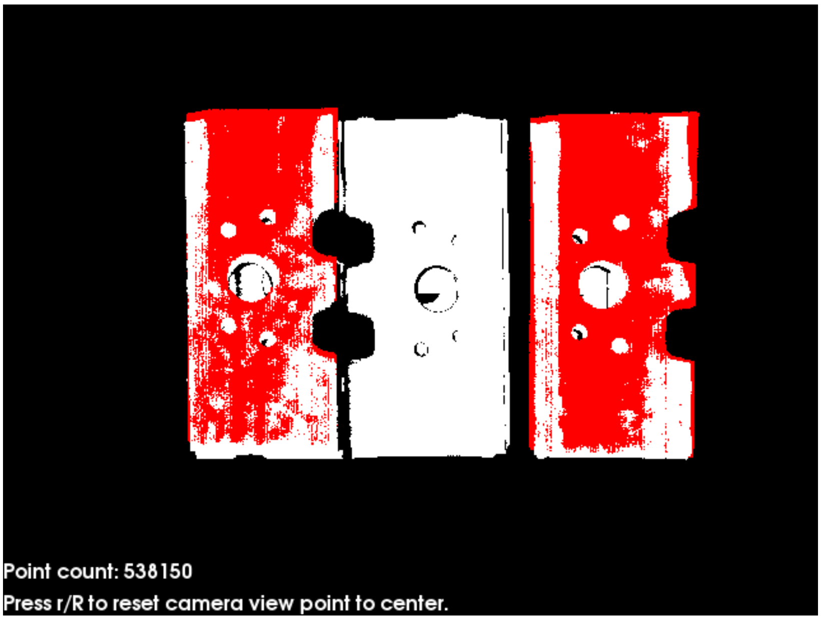
Sample Settings

Sampling Interval

Default value: 5.000 mm

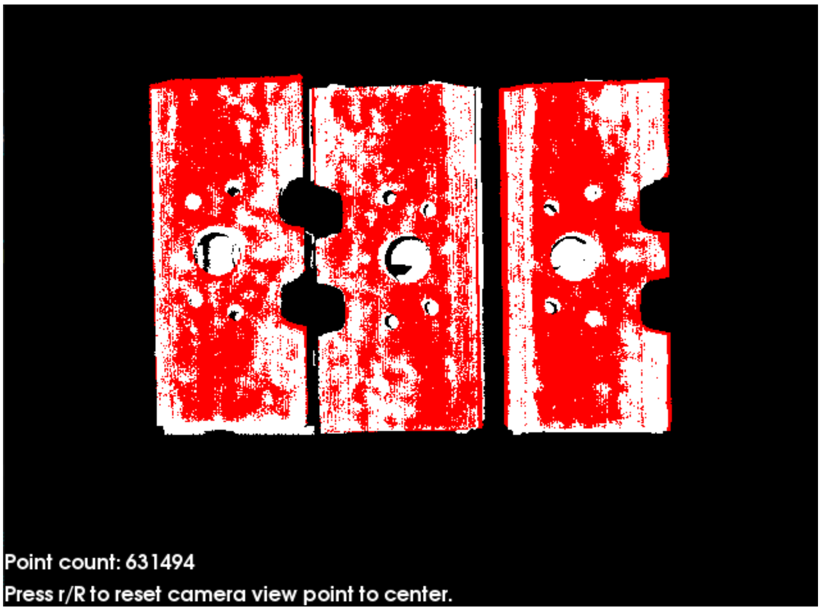
Instruction: This parameter is used for the downsampling of the model and scene point cloud. The unit is millimeters. The bigger the interval, the less the point number of the sampled point cloud, and the lower the accuracy of model estimation.

Example: The left and right figures below are the output results when this parameter is set as 0.0128 and 0.01, respectively. The white one is the input scene point cloud. It can be seen from the comparison of the two pictures that the smaller the sampling interval is, the more accurate the output result will be.



Point count: 538150

Press r/R to reset camera view point to center.



Point count: 631494
Press r/R to reset camera view point to center.

Search Points Number

Default value: 20

Instruction: When **Parameter Tuning Level** is Advanced, this parameter can be adjusted. This parameter is reserved for compatibility with old versions and is used to adjust the number of search points in the sampling process.

Pose Filtering Settings

Filter Pose by Model Rotation Angle

Instruction: In the Edge Matching mode, the point cloud model may need to be rotated by a certain angle to match with the scene point cloud, and the posed will be filtered by the point cloud model's rotation angle. When the model's rotation angle exceeds the Angle Difference Upper Threshold, the pose will be filtered out.

Default value: Disabled.

Angle Difference Upper Threshold

Instruction: See the tooltip of Filter Pose by Model Rotation Angle.

Default value: 135.000°

Validation Settings

Confidence Threshold

Default value: 0.5

Value range: [0, +∞)

Instruction: If the validation score of the result is larger than this threshold (which cannot be negative), the result is regarded as valid. The higher the value is, the more accurate results are retained.

Search Radius

Default value: 10.000 mm

Value range: [0, +∞)

Instruction: This parameter is the search radius when calculating the confidence. The unit is millimeters. Please set the value according to the actual situation of the point cloud. If the points in the point cloud are sparse, the Search Radius should be set slightly larger.

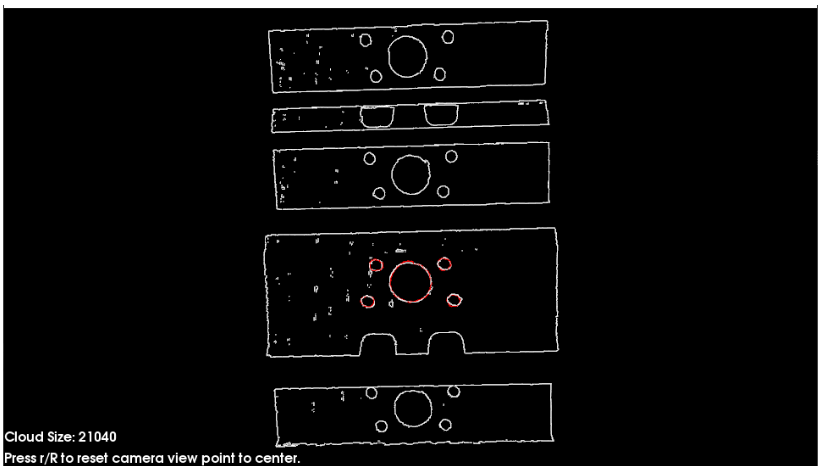
Output Settings

Maximum Number of Detected Poses in Each Point Cloud

Default value: 1

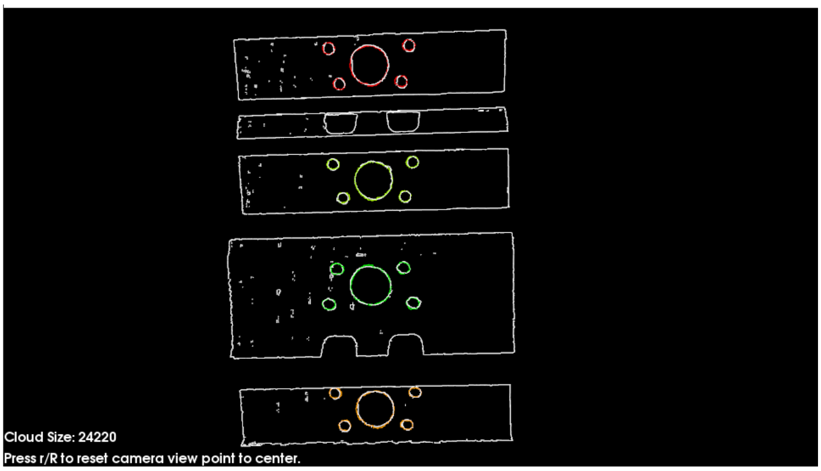
The larger the value is, the more matches are generated.

The figure below shows the output result of the parameter when the value is set to 1.



Cloud Size: 21040
Press r/R to reset camera view point to center.

The figure below shows the output when the parameter is adjusted to 4.



Cloud Size: 24220
Press r/R to reset camera view point to center.

Results Visualization

Show Matching Results

Default value: Selected

Instruction: If this option is selected, the output model and scene point cloud will be displayed.

Advanced Mode

This chapter only introduces the parameters that Basic Mode does not contain. If there are some parameters that are the same as that in the Basic Mode, please refer to Basic Mode.

Symmetry Settings

When a part of the workpiece to be recognized is symmetrical, you will need to adjust the symmetry settings to escape a local optimum of the matching result. An axis of the reference frame with the geometric center as its origin will be specified as the rotation axis, and the point cloud model will rotate around the specified axis according to the symmetry angle step and therefore the optimal matching result can be obtained.

Symmetry Axis

Default value: ROTATE_BY_Z

List of values: ROTATE_BY_X, ROTATE_BY_Y, and ROTATE_BY_Z

Instruction: This parameter is used to select the symmetry rotation axis around which the initial pose is changed.

Angle Step

Default value: 360°

Instruction: This parameter is used for changing the initial orientation from **Min Rotation Angle** to **Max Rotation Angle**.

Example: If an object looks the same before and after rotating 60° around an axis, its angle of rotational symmetry is 60°.

Min Rotation Angle

Default value: −180°

Instruction: This parameter is used to adjust the minimum rotation angle.

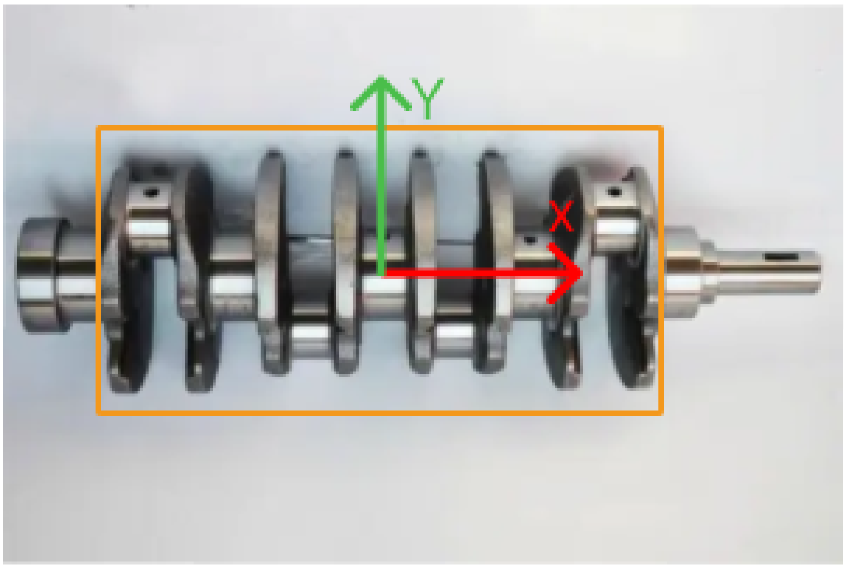
Max Rotation Angle

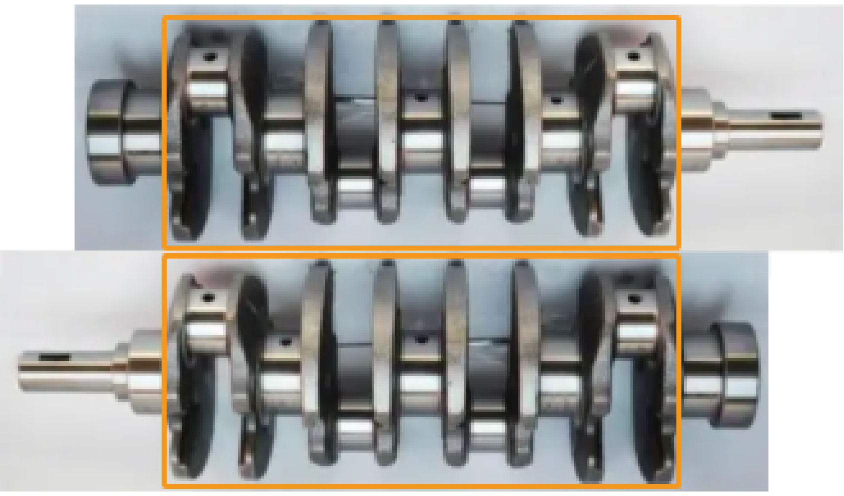
Default value: 180°

Instruction: This parameter is used to adjust the maximum rotation angle.

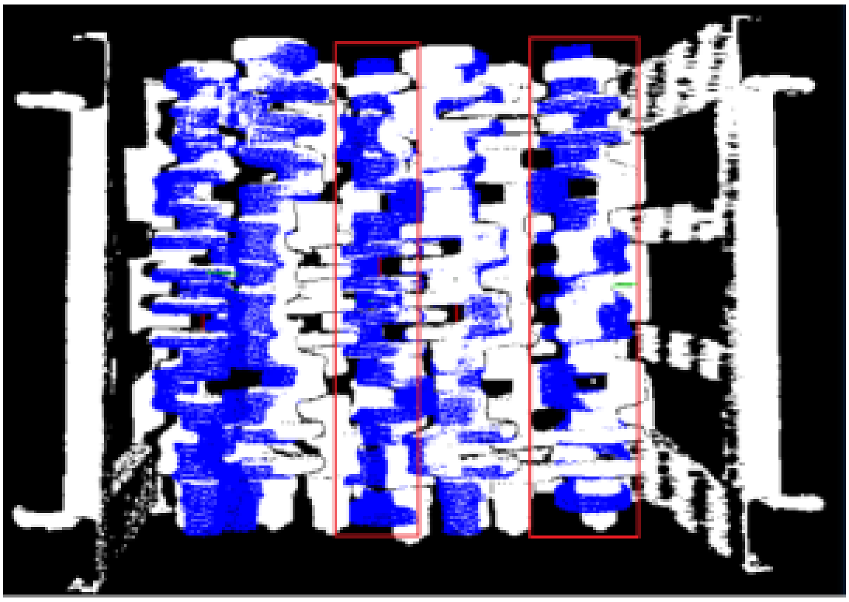
Example

As shown in the figure below, the main part of the crankshaft, i.e., the part in the frame, has 180° rotational symmetry along the Y-axis.



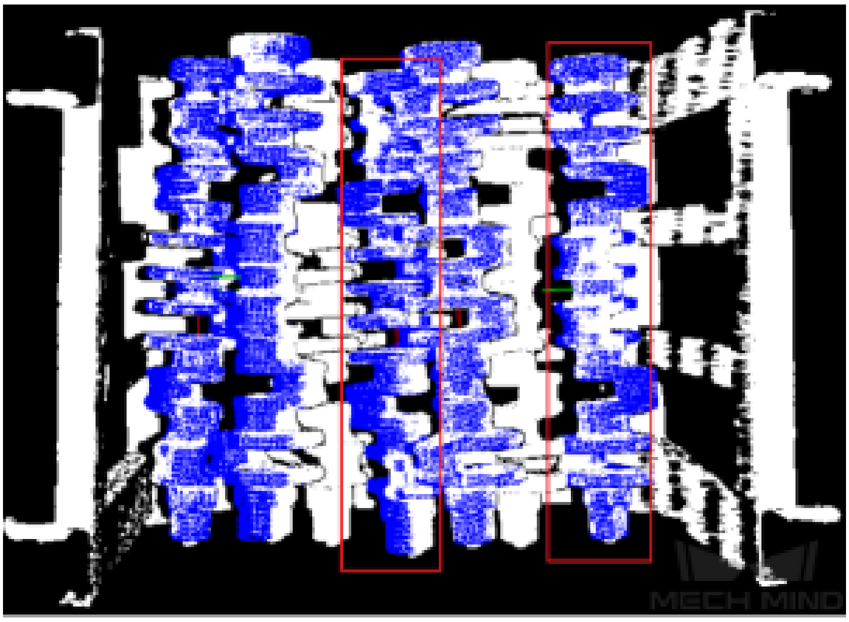


If you do not adjust the symmetry settings, there can be mismatches since the shapes of parts of the point cloud model and the crankshaft are similar, as shown below.



Only the middle part of the point cloud model and the crankshaft match with each other. However, if the point cloud model rotates 180°, it will match with the entire crankshaft, and the corresponding matching confidence will be higher.

After ROTATE_BY_Y (the Y-axis of the reference frame with the geometric center as its origin) is set as the Symmetry Axis, and the Angle Step is set to 180°, the point cloud model and the crankshaft match with each other correctly, as shown below.



Model Weight in Validation

When the workpieces to be recognized have the similar main body, while their specific parts differ greatly, the weight of the "parts" should be higher to output a correctly matched result.

File of Model Fragments with High Weight

Instruction: This file (PLY format) is a fragment of Model File. Based on the actual workpiece, determine which part is selected as the high weight file. The points in this file have a high weight set by the **Weight of Each Point**.

Weight of Each Point

Default value: 2.0

Instruction: This parameter is used to set the weight value for each point in the weight file, and there is no unit. The higher the value is, the better the matching result of the model fragments tend to be. Assuming that the weight coefficient of all initial point clouds is 1, after setting this parameter for the target cloud, the weight coefficient of the target cloud will become **1Xthisparameter**, so as to achieve the purpose of emphasis in the subsequent matching steps.

Search Radius When Set Weight

Default value: 3.000 mm

Instruction: This parameter is used to set the radius for setting weights. The unit is millimeters. The weight setting is an operation performed on the original point cloud model. The point cloud model will be down-sampled before it is used in the matching process, which causes some points to be shifted or lost. In the process of setting the weight of the points near the missing point, the search radius needs to be set, and the search radius is set by this parameter.

Example

This example is a continuation of the example in Symmetry Settings.

As the crankshafts have the similar main body, and their specific parts differ greatly, the weight of the “parts” should be higher to output a correctly matched result.

Validation Settings

Apply Angle Deviation in Validation

Default value: Unselected.

Instruction: If it is selected, the angle difference of the normal vector of the corresponding point pair will be considered in the pose validation process. In this case, although the output is often reduced, it could be more accurate.

Multiply Corresponding Points Ratio in Scene

Default value: Unselected.

Instruction: If it is selected, the matching score will be multiplied by the proportional coefficient of the corresponding point to the scene point. It is not suitable for the case of estimating poses of multiple objects in one point cloud.

Results Visualization

Show Sampled Model Cloud

Default value: Unselected.

Instruction: If it is selected, the sampled point cloud model will be visualized in the output.

Show Sampled Scene Cloud

Default value: Unselected.

Instruction: If it is selected, the sampled scene point cloud will be visualized in the output.

Show Validation Point Correspondences

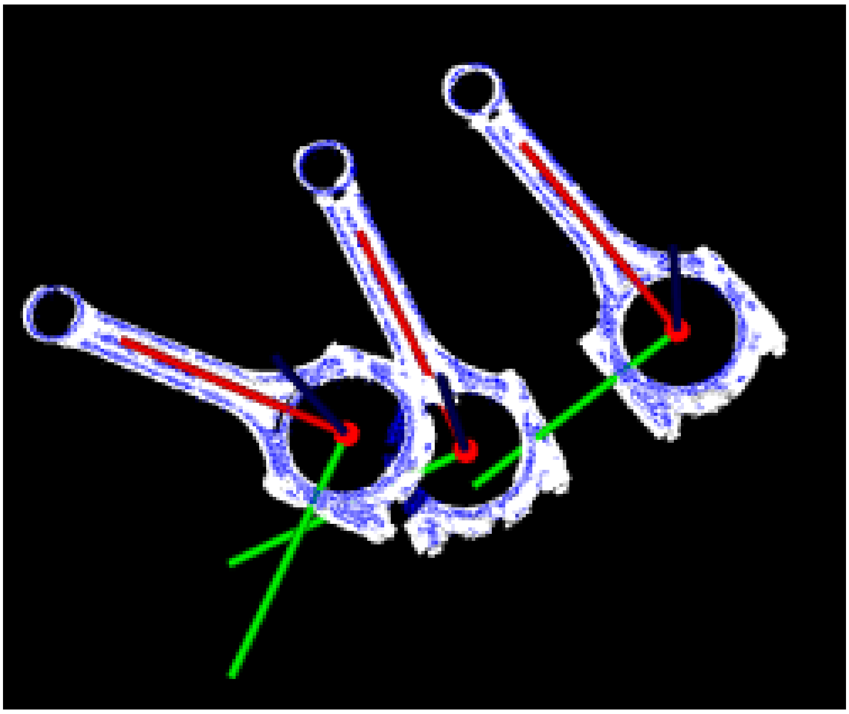
Default value: Unselected.

Instruction: If it is selected, the corresponding relationship between the model and the scene point cloud will be visualized in the output.

4.3.8. 3D Fine Matching

Function

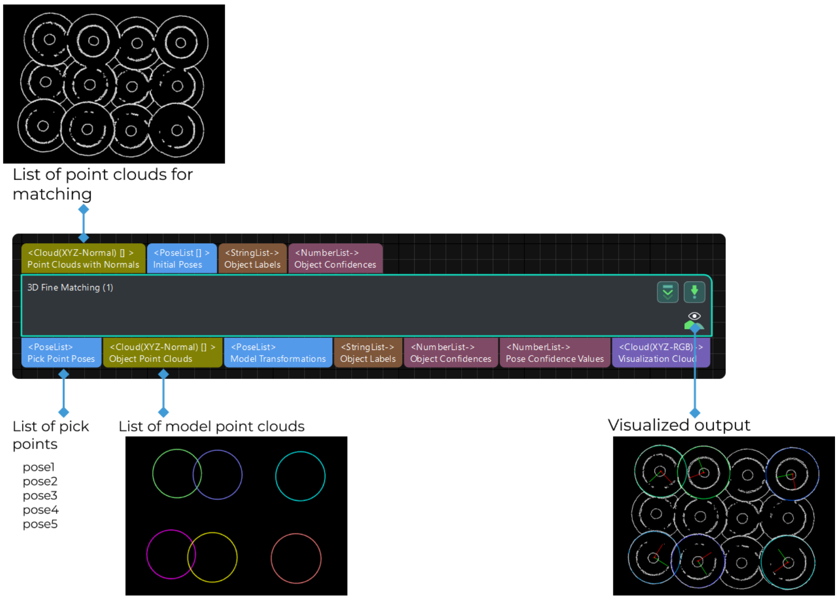
This Step can accurately match the point cloud model with the original point cloud and output more accurate poses of target objects. (To create the point cloud model, please refer to Generate Point Cloud Model.)



Usage Scenario

This Step performs fine matching based on the matching results of 3D Coarse Matching and outputs more accurate object poses that can be used as pick points.

Input and Output



The description of other input and output ports is shown in the following figure.

Other input port	Description
Initial Poses	Candidate pose list (usually obtained from 3D coarse matching).

Other input port	Description
Object Labels	The input to this port is generally the target object labels from the deep learning Steps (optional).
Object Confidences	The input to this port is generally the target object confidences from the deep learning Steps (optional).

Other output port	Description
Model Transformations	Model-to-scene transformation reference pose list.
Object Labels	Labels of target objects (related to deep learning Steps).
Object Confidences	Confidence values of target objects (related to deep learning Steps).
Pose Confidence Values	List of confidence values of the corresponding object poses.
Visualization Cloud	Show matching result with color point cloud. Points in white belong to scene cloud, while points rendered with different colors belong to different matched objects. The users can check the matching result of point cloud models by viewing the degree of overlap between white point clouds and color point clouds.

Parameters

Parameter Tuning Level

Default value: Basic

List of values: Basic, Advanced

Instruction: You can choose from the following two parameter tuning levels:

- Basic: basic adjustment, with a small amount of adjustable parameters.
- Advanced: advanced parameter tuning level, providing more adjustable parameters.

Basic Parameter Tuning Level

This section introduces the parameters in the Basic parameter tuning level, and these parameters are also available in the Advanced Parameter Tuning Level.

Model Settings

Model Selection

Instructions: Click the drop-down button to select and apply a point cloud template and its corresponding geometric centers.

Model File

Instructions: Click 📂 to select a point cloud template file in PLY format. You can use Matching Model and Pick Point Editor to easily generate, edit, and save point cloud templates and the corresponding geometric centers/pick points.

Geo Center Point File

Instructions: Click 📂 to select a geometric center file in JSON format. You can use Matching Model and Pick Point Editor to easily generate, edit and save point cloud templates and the corresponding geometric centers/pick points.

Cloud Orientation Calculation

Point Orientation Calc Mode

Default value: Origin

Instruction: This parameter is used to specify the method to calculate the orientations of points in the point cloud. There are four methods as shown below. Please select one according to the actual situation of the project.

Value list	Description
Origin	Directly obtain the normal from the point cloud model when the file contains the normal of the point cloud model directly.
StandardMode	Use the CPU to recalculate the normal direction of the input point cloud. It is recommended to use this mode when the model file doesn’t have normal direction: Search for the k points closest to the target, and use PCA (Principal Component Analysis) to obtain the minimum eigenvector as the point’s normal direction.
EdgeTangent	Calculate the tangent at the edge of the point cloud model and take the tangent as the normal. This method can distinguish different objects whose outer contours are mirror images of each other. This method is recommended when edge matching is performed on relatively flat objects. There are two example usage scenarios as follows:
	Scenario 1: The two sides of the object need to be distinguished. The 3D Fine Matching (Multiple Models) Step can be used, and there should be two model point cloud models corresponding to the two sides of the object respectively.
	Scenario 2: The two sides of the object need NOT to be distinguished. In this case, the 3D Fine Matching Step with one point cloud model is applicable.
EdgeNormal	Calculate the normal at the edge of the point cloud model. This method is recommended when edge matching is performed on relatively flat objects.

When using the **EdgeTangent** or **EdgeNormal** methods, ensure that each edge point cloud does not contain multiple objects; in other words, each object point cloud is separated.

Number of Searching Points

Default value: 10

Instruction: This parameter is used to set the number of points in the vicinity of the target point to search for. The minimum value is 1.

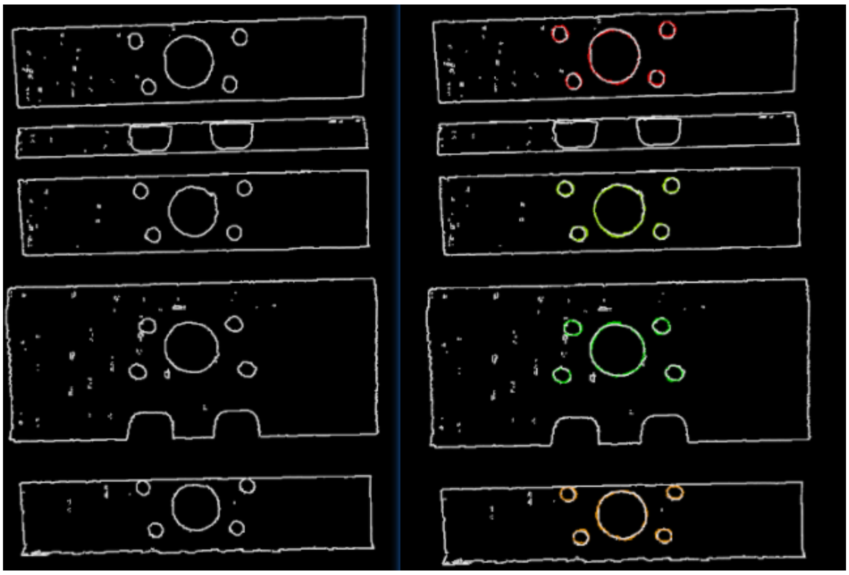
Matching Method

Default value: GMM

Value list: GMM and nearest-neighbor.

Instruction: This parameter is used to select the local matching method to match the point cloud model with the scene point cloud. The point cloud model will move and rotate according to the set range to match the point cloud in the scene obtained in the previous step match.

Example: In most scenarios, GMM has better anti-interference ability and higher matching speed, so it is generally recommended; in rare cases when GMM cannot meet the demand, nearest-neighbor can be selected.



Settings under GMM

Matching Mode

Default value: Standard

Value list: HighSpeed, Standard, and HighPrecision.

Instruction: There are three matching modes. Select the appropriate one according to the project requirements:

HighSpeed: the fastest but less accurate. **Standard**: relatively stable. **HighPrecision**: the most accurate but slower.

Number of Iterations

Default value: 30

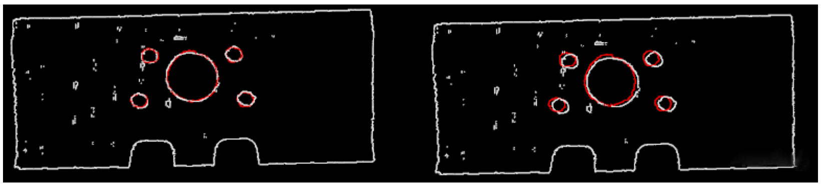
Instruction: Iteration refers to iteratively running a specific instruction until the conditions are met. The number of iterations refers to the number of repetitions of instruction running in this process. The parameter set here is the upper limit of the number of iterations. The larger the value is, the greater the number of times of matching calculations will be, which means a longer running time and higher accuracy.

Standard Deviation

Default value: 3.000 mm

Instruction: The value here needs to match with the initial pose deviation. When the input initial pose deviation after the previous step (3D Coarse Matching) is greater, a larger **Standard Deviation** is required. In this case, the matching speed is faster.

Example: When the set value is larger than the initial pose deviation, there will be a great result deviation. As shown in the figure below, the **Standard Deviation** on the right is 40.000 mm, far greater than the initial pose deviation. You can see that the matching result on the right with an overly large standard deviation is less accurate. Normally, this value will be no greater than 10 mm.



For smaller workpieces, it is recommended to adjust the standard deviation to a smaller value for more accurate matching results.

Standard Deviation Update Step Number

Default value: 3

Instruction: During the matching process, the **Standard Deviation** gradually decreases until it reaches the value of **Minimum Standard Deviation**. The number of times that the standard deviation decreases in this process is the value of this parameter. For some cases with large standard deviation, the more times of the decrease, the more stable the entire decrease process. Under normal circumstances, this parameter does not need to be adjusted.

Speed Up on Large Object Quantities

Default value: Disabled.

Instruction: It is recommended to enable this option if there is a large number of objects in the scene. If this option is enabled, the Step will perform faster when there is a large number of objects. However, if the number of objects is small, the processing speed of the Step may be slowed down.

Settings under Nearest-Neighbor

Iteration Settings

Nearest Point Search Radius

Default value: 10.000 mm

Value range: [1.000, +∞)

Instruction: This parameter is used to adjust the search radius of the nearest point (in mm), and the value setting should correspond to the input initial pose deviation. When the input initial pose deviation is large (the deviation between the scene point cloud and the point cloud model is large), this parameter needs to be increased; when the initial pose deviation is very small, this parameter can be adjusted to a small value. The minimum value that can be set for this parameter is 1.000 mm. If the radius is less than this lower limit, some sparsely distributed points may not find the closest point, which could affect the output result.

MSE Threshold

Default value: 0.001

Instruction: This parameter (Mean Square Error Threshold) is used to adjust the upper limit of the mean square error. This value is recorded after each iteration and used for comparison in subsequent iteration steps. When the mean square error of an iteration is less than the set value, it is considered that the iteration effect has met the requirements and the iteration will be terminated.

Window Size

Default value: 10

Value range: [3, +∞)

Instruction: This parameter refers to the number of consecutive iterations with small error fluctuations in the iterative optimization process. When this parameter is too small, a local optimal solution may appear, which affects the final matching accuracy. Example: If the window only contains the error of the results of three consecutive iterations, so the error trend can only be obtained by comparing the errors of the three iterations, which may lead to a local optimal solution (the error of the second iteration is less than the error of the first iteration and the first iteration) instead of a global optimal solution. According to the recommended value of the number of iterations which is 30, subsequent iterations with more significant error reduction may occur.

Show Corresponse

Default value: Unselected.

Instruction: If it is selected, the corresponding points will be displayed in each iteration.

Complex Object

Default value: Unselected.

Instruction: Selected if the target object is a complex-shaped object (not a simple geometric shape) or if the **Automatic Weight** function is required.

Automatic Weight

Default value: Unselected.

Instruction: If it is selected, the weight of iteration points will be calculated automatically. It is recommended to use this function for objects with poor coarse matching results.

Point Pair Rejection

Reject Pair

Default value: Unselected.

Instruction: This parameter decides whether to remove repeated point pairs. There are multiple point pairs in the process of searching for the nearest point, and sometimes the unwanted point pairs need to be filtered out. If this option is selected, this function will be used to remove them through configurating the following **N Sigma Threshold** and **Point Pair Angle Diff Thre** parameters; if not checked, these two parameters will not take effect during the matching process.

N Sigma Threshold

Default value: 1.0000

Instruction: This parameter is the multiplier of the standard deviation threshold of the distance between each points. The point pair whose distance is outside N sigma will be eliminated.

Point Pair Angle Diff Thre

Default value: 45°

Instruction: This parameter is the upper limit of the angle difference (in °) between the normals of two points in every matching point pair. If the angle difference is greater than this value, this point pair will be eliminated.

Apply Weights on Pairs

Default value: Unselected.

Instruction: If it is selected, the weight of the model will be used in the iteration of the algorithm, so it is more likely to match better.

Remove Repeat Correspond

Default value: Unselected.

Instruction: If it is selected, the function of removing repeating corresponding point pairs will be used, and the settings of **Remove Repeat Correspond Type** will take effect. It is recommended to use this function when the point cloud model has more points than that of a single object in the scene.

Remove Repeat Correspond Type

Default value: MinDis

List of values: MinDis and GlobalMin

Instruction: This parameter is used to select the method of removing repeating corresponding point pairs. There are two methods in total:

MinDis: Calculate the distance between the repetitive point pairs. Then, keep the point pair with the smallest distance, and delete the remaining repetitive point pairs. More repeating corresponding point pairs will be eliminated in this method than the other. **GlobalMin**: Take the average value of the distance between the repetitive point pairs as a reference to find out the point pair whose distance is closest to the average value. Then, retain this point pair and eliminate others. Less repeating corresponding point pairs will be eliminated in this method than the other.

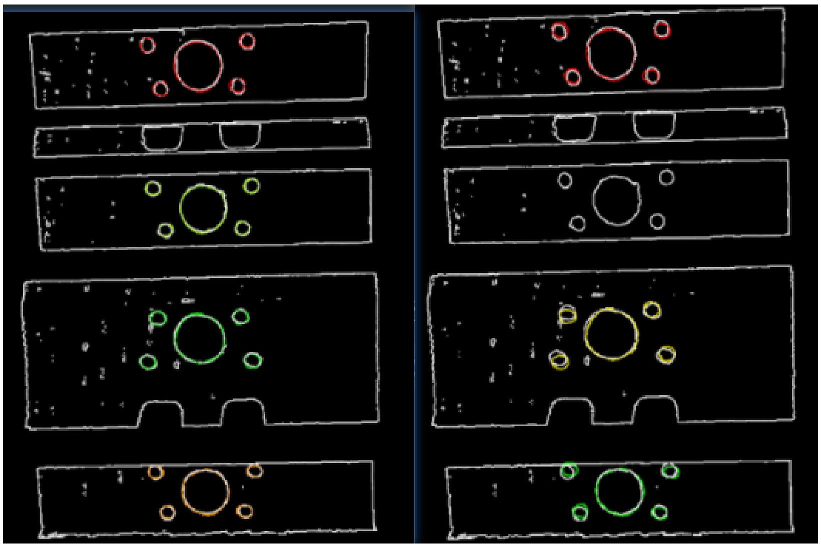
Sample Settings

Sampling Interval

Default value: 5.000 mm

Instruction: This parameter is used for the downsampling of the model and scene point cloud. The bigger the interval, the less the point number of the sampled point cloud, and the lower the accuracy of model estimation.

Example: The figure below shows the output results when Sampling Interval is set to 5.000 mm and 25.000 mm, respectively. The white parts are the input object point clouds, and the colored lines are the matching results. It can be seen from the comparison of the two pictures that the smaller the sampling interval is, the more accurate the output result will be.



Validation Settings

Confidence Threshold

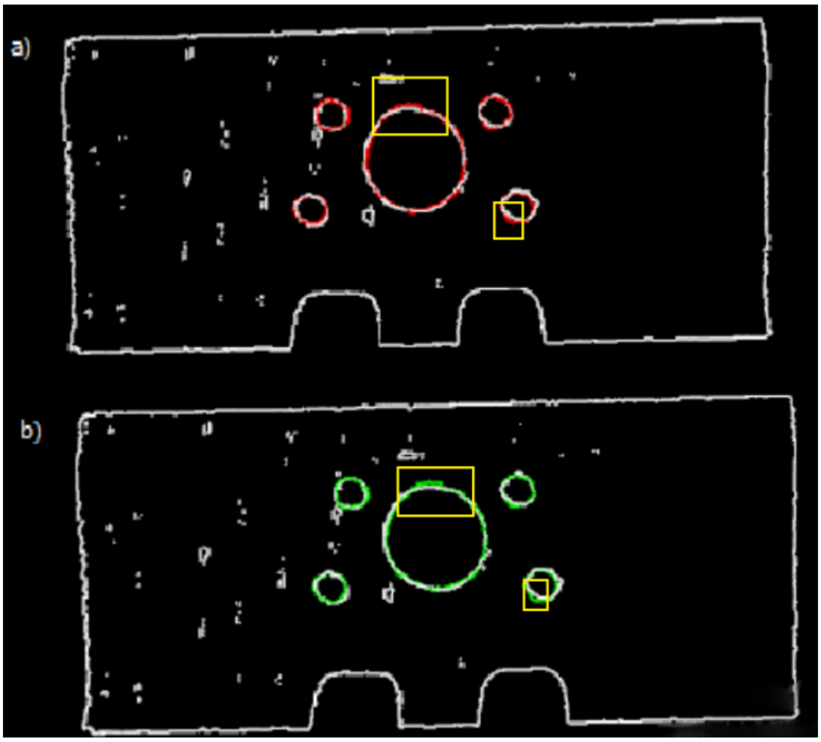
Default value: 0.500

Value range: [0, +∞)

Instruction: The confidence threshold is used to determine whether the matching result is valid. Matching results with confidences higher than the threshold are valid and those lower are not. A higher threshold leads to fewer valid matches, while the output results are more accurate. A lower threshold leads to more valid matches, while the output results are less accurate.

Example: In the figure below, the confidence threshold of (a) is 0.500, and that of (b) is 0.920. The match of (a) is more accurate than that of (b). It can be known that when the larger **Confidence Threshold** is, the more likely it is that results with higher matching accuracy are retained.

> Even when a higher threshold is set, there may be no valid matching result to output. It may be because that different parts of the target object have the same shape.



Search Radius

Default value: 10.000 mm

Value range: [0, +∞)

Instruction: The value of Search Radius affects the pose matching score. The higher the value of the Search Radius, the looser the criterion for evaluating the pose matching result will be, which means that even if the pose matching result is of poor quality, the pose matching score will be high; the lower the value of the Search Radius, the more strict the criterion for evaluating the pose matching result will be. Please set the value according to the actual situation of the point cloud. If the points in the point cloud are sparse, the Search Radius should be set slightly larger.

During parameter tuning, you can set an original **Search Radius** first and run the Step. Then you can check the matching result in the Debug Output window and compare the result with the pose matching score. If the matching result does not correspond with the matching score, please adjust the **Search Radius** according to the actual situation.

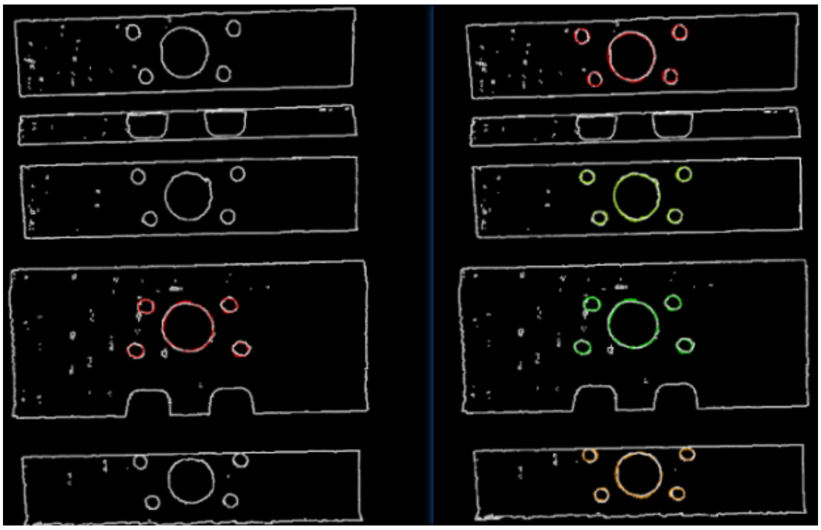
Output Settings

Maximum Number of Detected Poses in Each Point Cloud

Default value: 1

The larger the value, the more matches are generated.

Instruction: The larger the value is, the more matching pairs will be output. The figure below shows the output results when this parameter is set to 1 (left) and 4 (right).



Results Visualization

Show Matching Results

Default Value: Selected.

Instruction: If this option is selected, the output model and scene point cloud will be displayed.

Advanced Parameter Tuning Level

This chapter only introduces the parameters that Basic Mode does not contain. If there are some parameters that are the same as that in the Basic Mode, please refer to Basic.

Symmetry Settings

When a part of the workpiece to be recognized is symmetrical, you will need to adjust the symmetry settings to escape a local optimum of the matching result. An axis of the reference frame with the geometric center as its origin will be specified as the rotation axis, and the point cloud model will rotate around the specified axis according to the symmetry angle step and therefore the optimal matching result can be obtained.

Symmetry Axis

Default value: ROTATE_BY_Z

List of values: ROTATE_BY_X, ROTATE_BY_Y, and ROTATE_BY_Z

Instruction: This parameter is used to select the symmetry rotation axis around which the initial pose is changed.

Angle Step

Default value: 360°

Instruction: This parameter specifies the angle of rotational symmetry.

Example: If an object looks the same before and after rotating 60° around an axis, its angle of rotational symmetry is 60°.

Min Rotation Angle

Default value: −180°

Instruction: This parameter is used to adjust the minimum rotation angle.

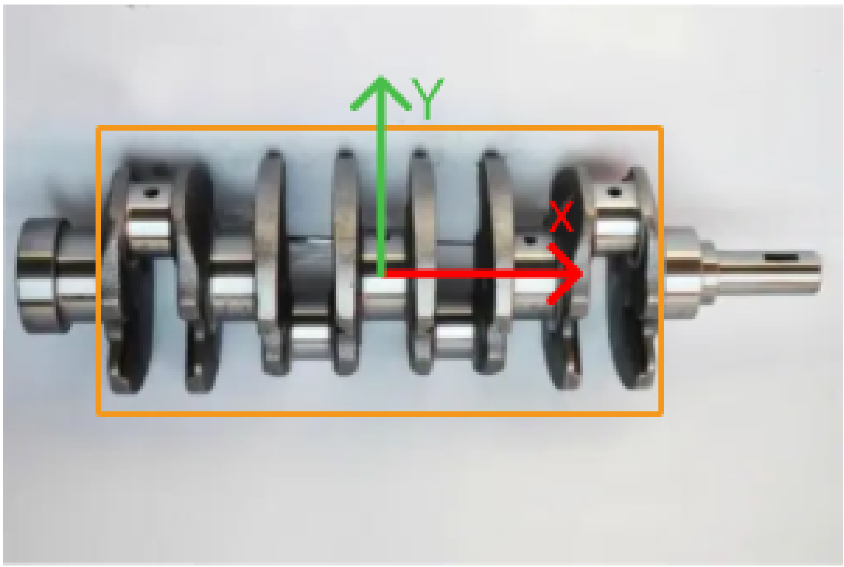
Max Rotation Angle

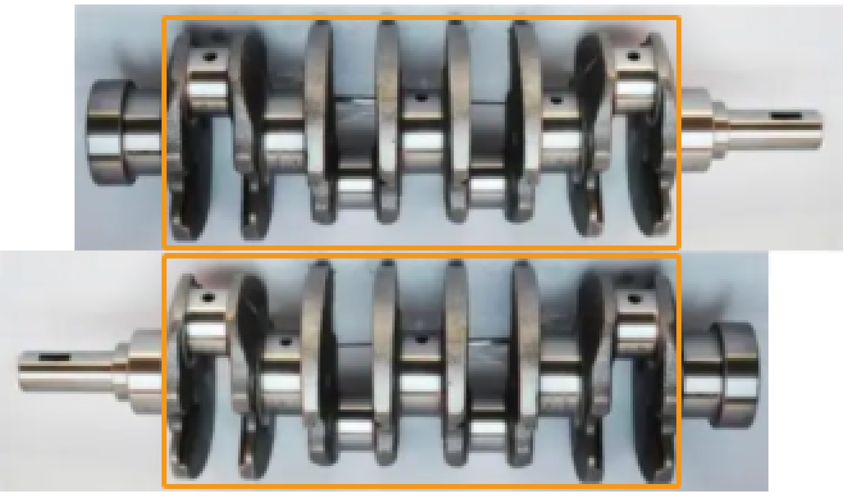
Default value: 180°

Instruction: This parameter is used to adjust the maximum rotation angle.

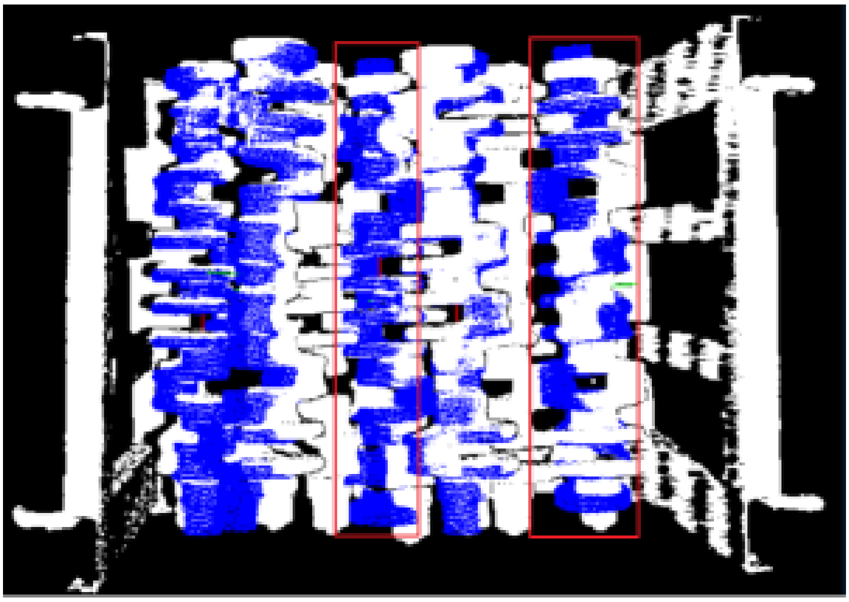
Example

As shown in the figure below, the main part of the crankshaft, i.e., the part in the frame, has 180° rotational symmetry along the Y-axis.



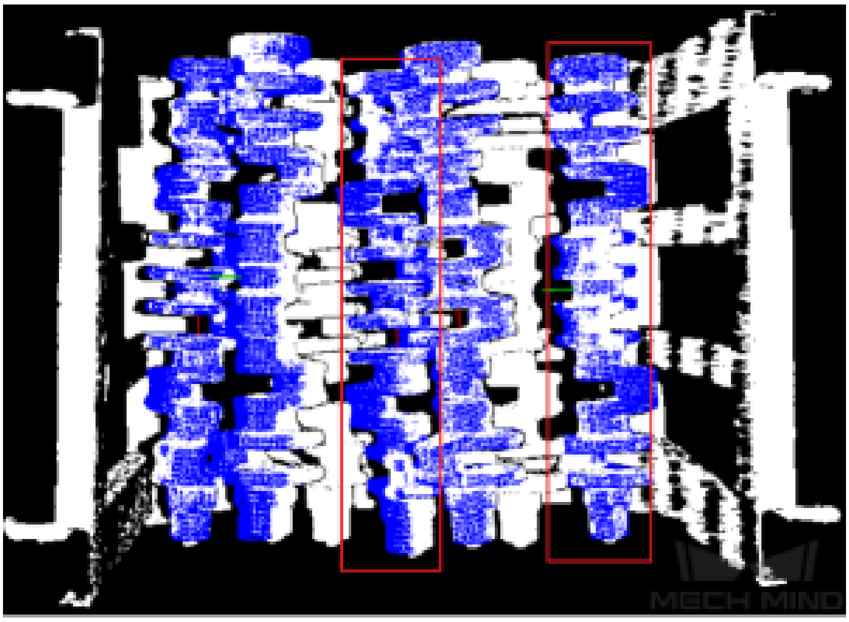


If you do not adjust the symmetry settings, there can be mismatches since the shapes of parts of the point cloud model and the crankshaft are similar, as shown below.



Only the middle part of the point cloud model and the crankshaft match with each other. However, if the point cloud model rotates 180°, it will match with the entire crankshaft, and the corresponding matching confidence will be higher.

After ROTATE_BY_Y (the Y-axis of the reference frame with the geometric center as its origin) is set as the Symmetry Axis, and the Angle Step is set to 180°, the point cloud model and the crankshaft match with each other correctly, as shown below.



Pose Filtering Settings

Filter Pose by Model Rotation Angle

Instruction: In the Edge Matching mode, the point cloud model may need to be rotated by a certain angle to match with the scene point cloud, and the posed will be filtered by the point cloud model's rotation angle. When the model's rotation angle exceeds the Angle Difference Upper Threshold, the pose will be filtered out.

Default value: Disabled.

Angle Difference Upper Threshold

Instruction: See the tooltip of Filter Pose by Model Rotation Angle.

Default value: 135.000°

Model Weight in Validation

When the workpieces to be recognized have the similar main body, while their specific parts differ greatly, the weight of the "parts" should be higher to output a correctly matched result.

File of Model Fragments with High Weight

Instruction: This file (PLY format) is a fragment of Model File. Based on the actual workpiece, determine which part is selected as the high weight file. The points in this file have a high weight set by the **Weight of Each Point**.

Weight of Each Point

Default value: 2.0000

Instruction: This parameter is to set the weight value for the parts with salient features in the point cloud model. The initial weight of all points is 1. After setting this parameter for the target cloud, the weight coefficient of the target cloud will become **1Xthisparameter**, so as to achieve the purpose of emphasis in the subsequent matching steps.

Search Radius When Set Weight

Default value: 3.000 mm

Instruction: This parameter is used to set the radius for setting weights. The unit is millimeters. The weight setting is an operation performed on the original point cloud model. The point cloud model will be down-sampled before it is used in the matching process, which causes some points to be shifted or lost. In the process of setting the weight of the points near the missing point, the search radius needs to be set, and the search radius is set by this parameter.

Validation Settings

Apply Angle Deviation in Validation

Default value: Unselected.

Instruction: If it is selected, the angle difference of the normal vector of the corresponding point pair will be considered in the pose validation process. In this case, although the output is often reduced, it could be more accurate.

Multiply Corresponding Points Ratio in Scene

Default value: Unselected.

Instruction: If it is selected, the matching score will be multiplied by the proportional coefficient of the corresponding point to the scene point. It is not suitable for the case of estimating poses of multiple objects in one point cloud.

Results Visualization

Show Sampled Model Cloud

Default value: Unselected.

Instruction: If it is selected, the sampled point cloud model will be visualized in the output.

Show Sampled Scene Cloud

Default value: Unselected.

Instruction: If it is selected, the sampled scene point cloud will be visualized in the output.

Show Validation Point Correspondences

Default value: Unselected.

Instruction: If it is selected, the corresponding relationship between the model and the scene point cloud will be visualized in the output.

4.3.9. 3D Matching and Classification (Multiple Models)

Function

This Step can match the objects in the scene by using multiple point cloud models and output the poses and class labels of the objects.

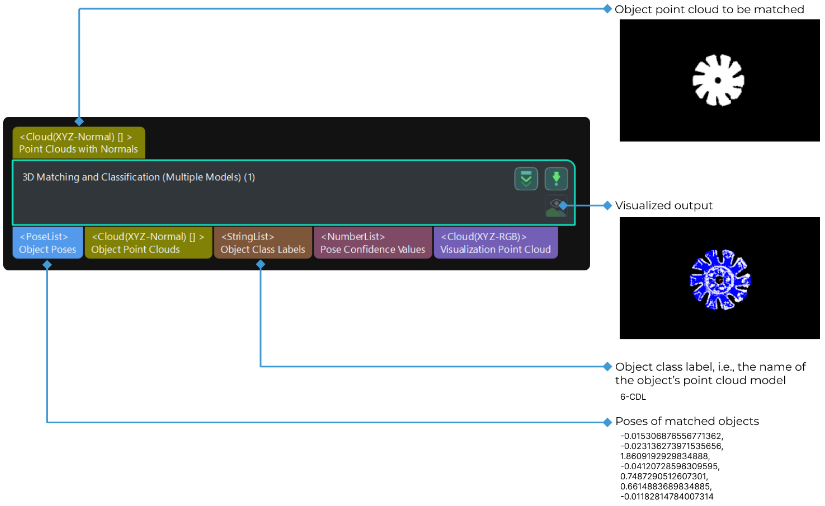
Usage Scenario

This Step is usually used for the matching and classification of a single object in the scene. If there are multiple objects in the scene, point cloud clustering or deep learning can be utilized to segment objects first and then each object can be matched and classified by using this Step.

Input and Output

After inputting the point clouds of objects, this Step will output the object poses with the same amounts as point clouds.

The input and output of this Step are shown below with the example of inputting a point cloud of a single object.



Parameter Description

Parameter Tuning Level

Parameter description: This parameter is used to select the tuning level of the parameters. The parameters in different tuning levels are different.

Value list: Basic, Advanced

- Basic: basic parameter tuning level, which provides parameters that are frequently adjusted.
- Advanced: advanced parameter tuning level, which provides more adjustable parameters.

Default value: Basic

Tuning recommendation: Select the parameter tuning level according to your need.

Basic Parameter Tuning Level

Model Settings

Select Model

Description: This parameter is used to select the point cloud model.

Instruction: Select an available point cloud model in the drop-down list. Please refer to Matching Model and Pick Point Editor to make a point cloud model file.

> - The point cloud model type should be the same. You cannot select Surface point cloud model and Edge point cloud model at the same time.
> - The sizes of the point cloud models should not differ too much. It is recommended that the difference between the largest point cloud model and the smallest one should not exceed 1.5 times. Otherwise, the matching speed will be affected. If there is a large disparity in the sizes of the point cloud models, it is recommended to classify the input scene point cloud first by using the Classify Point Clouds by Dimensions Step.

Matching Mode

Description: This parameter is used to select the mode according to the type of the point cloud model. If the point cloud model represents the edge of the object, select “Edge matching.” If the point cloud model represents the surface of the object, select “Surface matching.”

Value list: Surface matching, Edge matching.

- Surface matching: Use the object’s surface model for point cloud model matching.
- Edge matching: Use the object’s edge model for point cloud model matching.

Tuning recommendation: When the surface of the object has obvious recognizable features (such as crankshafts, rotors, and steel rods), it is recommended to use surface matching. You should create a point cloud model that represents the surface features of the object. When the object is relatively flat and shows clear and regular edge features under the camera (such as panels, track pads, connecting rods, and brake discs), it is recommended to use edge matching. You should create a point cloud model that represents the edge features of the object.

> If you select **Edge matching**, you should input the point clouds which are extracted the edges.

Coarse Matching Settings

Operation Mode

Description: This parameter is used to select the operation mode for coarse matching.

Value list: High speed, Standard, High accuracy, Extra-high accuracy, and Custom.

Instruction: Under all kinds of operation modes, the matching accuracy and speed are different due to different number of points in the point cloud models. The following table provides a detailed description.

Operation Mode	Number of Points in the Point Cloud Models	Matching Characteristics
High speed	200	High speed, low accuracy
Standard	300	Relatively stable
High accuracy	400	High accuracy, low speed
Extra-high accuracy	500	Highest accuracy
Custom	Custom	Determined by customized number of points

Expected Point Count of Model

Description: This parameter is used to set the expected number of points in the point cloud model.

Default value: 200

Instruction: Set this parameter when **Operation Mode** is **Custom**.

Tuning recommendation: Please set the parameter according to the actual requirement.

When the matching or classification results are not satisfactory, it is recommended to prioritize the adjustment of this parameter to increase the "Expected Point Count of Model."

Fine Matching Settings

Operation Mode

Description: This parameter is used to select the operation mode for fine matching.

Value list: High speed, Standard, High accuracy, and Extra-High accuracy.

Instruction: Please refer to the operation modes of the "Coarse Matching Settings."

Validation Settings

Confidence Threshold

Description: If the confidence of the matching result is above the threshold, the matching result is valid. The higher the confidence value is, the more accurate the matching result is.

Default value: 0.300

Advanced Parameter Tuning Level

Model Settings

You can refer to the Basic Parameter Tuning Level.

Coarse Matching Settings

You can refer to the Basic Parameter Tuning Level.

Expected Point Count of Model

You can refer to the Basic Parameter Tuning Level.

Upper Limit of Point Pair Count per Feature

Description: This parameter is used to specify the maximum number of point pairs contained in each feature during the model analysis. The smaller this value is, the faster the processing speed, but the lower the accuracy of the matching results.

Default value: 50

Voting Settings

Distance Quantification

Description: The parameter for the quantification of the distance between points (Distance between Two Points=Distance Quantification*Sampling Interval). The larger the value is, the less accurate the matching results.

Default value: 1.00

Angle Quantification

Description: The parameter for the quantification of the angle between the normals of the points in the point pair (Angle between Two Vectors = 360°/Angle Quantification). The larger this value is, the more accurate the matching results, but it also requires higher point cloud quality.

Default value: 60

Vote Ratio Lower Threshold

Description: In the matching process, each object pose will earn a score. When the pose score is higher than the “product of Highest Score in the Vote and Vote Ratio Lower Threshold,” the corresponding pose will be involved in the pose verification. The lower this value is, the more likely an exact match will be found, but the execution time will also be longer.

Default value: 0.80

Referring Point Sampling Step

Description: This parameter is used to adjust the step size to downsample the scene point cloud and obtain referring points that form point pairs with the referred points. When the value is larger, the interval sampling points are fewer, and the execution speed is faster, but the matching accuracy is reduced.

Default value: 5

Referred Point Sampling Step

Description: This parameter is used to adjust the step size to downsample the scene point cloud and obtain referred points that form point pairs with the referring points. When the value is larger, the interval sampling points are fewer, and the execution speed is faster, but the matching accuracy is reduced.

Default value: 1

Pose Verification Settings

Use Distance NMS

Description: After this parameter is selected, candidate poses whose distances to the selected poses are less than 0.1 times the diameter of the object will be filtered out.

Default value: selected.

Voxel Length Generation Strategy

Description: This parameter is used to select the method to generate the voxel length.

Default value: Auto.

Tuning recommendation: “Auto” is recommended to be used for new users.

Voxel Length

Description: This parameter is used to set the length (mm) of the voxel. The larger the value, the less accurate the matching results.

Default value: 3.000 mm

Voxel Length Lower Limit

Description: This parameter is used to set the minimum voxel length (mm).

Default value: 1.000 mm

Voxel Length Upper Limit

Description: This parameter is used to set the maximum voxel length (mm).

Default value: 15.000 mm

Pose Filtering Settings (only available in the edge matching mode)

Filter Poses by Model Rotation Angle

Description: In the edge matching mode, the point cloud model may need to be rotated by a certain angle to match with the scene point cloud, and the poses will be filtered by the point cloud model’s rotation angle. When the model’s rotation angle exceeds the “Angle Difference Upper Threshold,” the pose will be filtered out.

Default value: selected.

Angle Difference Upper Threshold

Description: Please refer to the parameter “Filter Poses by Model Rotation Angle” for details.

Fine Matching Settings

Deviation Correction Capacity

Description: This parameter is used to set the intensity of the deviation correction to the matching result from 3D Coarse Matching. The greater the deviation correction capacity is, the more likely the coarsely matched poses can be corrected to the accurately matched poses. Please note that an excessive deviation correction capability may lead to a loss of matching accuracy.

Value list: Small, Medium, and Large.

Default value: Small.

Operation Mode

Description: You can refer to the Basic Parameter Tuning Level.

Symmetry settings

Symmetry Axis

Description: This parameter is used to select the axis of rotational symmetry to rotate the initial pose of the object.

Value list: ROTATE_BY_X, ROTATE_BY_Y, and ROTATE_BY_Z

Default value: ROTATE_BY_Z

Symmetry Angle Step (0–360)

Description: This parameter is used to set the step to change the initial orientation from “Min Rotation Angle” to “Max Rotation Angle.” The unit is degree (°).

Default value: 360.0°

Instruction: The value of “Symmetry Angle Step” parameter should be more than 0.

Min Rotation Angle

Description: This parameter specifies the minimum rotation angle for the pose rotation according to the “Symmetry Angle Step.” The unit is degree (°).

Default value: -180.0°

Max Rotation Angle

Description: This parameter specifies the maximum rotation angle for the pose rotation according to the “Symmetry Angle Step.” The unit is degree (°).

Default value: 180.0°

Validation Settings (The parameters need to be adjusted in the surface matching mode)

Only Consider Visible Surface of Model

Description: Once this parameter is selected, only the visible part of the surface point cloud model will be involved in the matching, which facilitates the calculation and produces a more reasonable matching confidence.

Default value: Unselected.

Tuning recommendation: It is recommended to select this parameter for object models that have parts not visible to the camera under a viewing angle, such as the complete model of a cylindrical object. For object models that are completely visible from the same viewing angle, such as the surface matching model of the upper side of a brake disc, this parameter can be ignored.

Result Visualization

Visualization Options

Description: After “Debug Output” is enabled, the selected option will be generated and displayed.

Value list: Coarse matching result, Fine matching result, Downsampled model for coarse matching, Downsampled model for fine matching, Downsampled scene point cloud for coarse matching, Downsampled scene point cloud for fine matching.

Default value: Fine matching result.

Only Visualize Current Matching Result

Description: Once this parameter is selected, only the matching result corresponding to the current model will be visualized.

Default value: Unselected.

Instruction: Select the model for visualization if this parameter is unselected.

Select Model for Visualization

Description: This parameter is used to select the point cloud model that you want to visualize.

Tuning recommendation: Set this parameter after unselect **Only Visualize Current Matching Result**.

4.3.10. 3D Workpiece Recognition

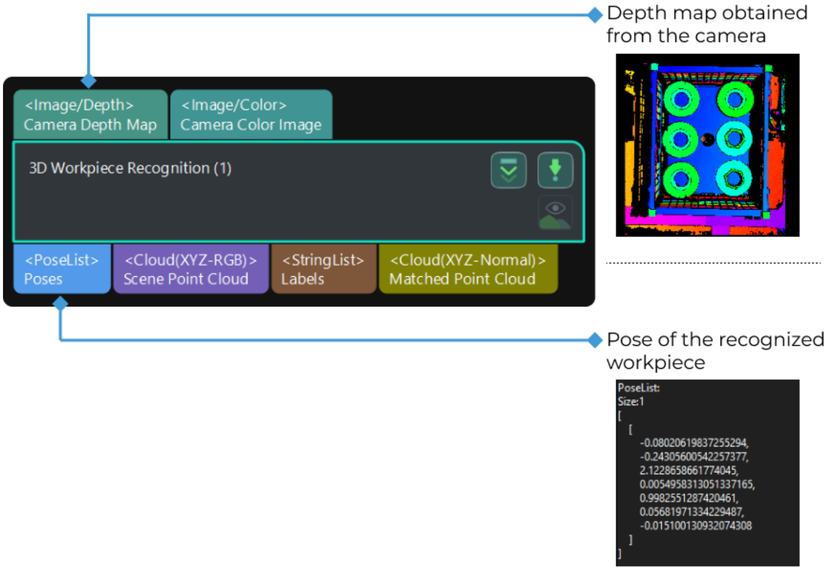
Function

This Step has integrated vision processing functions such as point cloud pre-processing, 3D model matching, and removing overlapped objects, which facilitate quick workpiece recognition.

Usage Scenario

This Step is applicable to metal workpiece loading scenarios and recognize objects with various shapes and objects placed in different ways, such as objects that are placed individually, neatly arranged on a single layer, neatly arranged on multiple layers, and randomly stacked. This Step usually follows the Capture Images from Camera Step, and the communication is established through Standard Interface or Adapter.

Input and Output


Depth map obtained from the camera
<Image/Depth> Camera Depth Map
<Image/Color> Camera Color Image
3D Workpiece Recognition (1)
<PoseList> Poses
<Cloud(XYZ-RGB)> Scene Point Cloud
<StringList> Labels
<Cloud(XYZ-Normal)> Matched Point Cloud
Pose of the recognized workpiece

```
PoseList:
Size:1
[
  [
    -0.08020619837255294,
    -0.24305600542257377,
    2.1228658661774045,
    0.0054958313051337165,
    0.9982551287420461,
    0.05681971334229487,
    -0.015100130932074308
  ]
]
```

Parameter Description

Open the editor

Description: Click the **[Open the editor]** button to launch the 3D Workpiece Recognition Visualization Configurator.

Tuning Recommendation: Please refer to 3D Workpiece Recognition Visualized Configurator for detailed instructions.

> You can also click the Step and then click the **[Open the editor]** button displayed on the Step to launch the configurator.
>
> [3d workpiece recognition click open editor] | *3d-workpiece-recognition-click-open-editor.png*

Select workpiece

Description: This parameter is used to select a workpiece to recognize in the drop-down list. The workpiece should be defined in the Configurator beforehand.

4.3.10.1. 3D Workpiece Recognition Visualized Configurator

This section introduces how to use the 3D Workpiece Recognition Visualized Configurator.

Function

3D Workpiece Recognition Visualized Configurator integrates general vision processing functions. With the following three procedures, you can complete configuring the 3D Workpiece Recognition Step quickly.

Select object
Preprocessing
Recognition

1. **Select workpiece**: select one or multiple point cloud models from Matching Model and Pick Point Editor as the target workpiece.
2. **Preprocessing**: select a region of interest (ROI) for recognition to filter the interference. The algorithm will perform basic processing on the point cloud in the ROI to improve recognition efficiency.
3. **Recognition**: adjust the 3D matching parameters in a visual way to make the recognition accuracy and takt meet the requirement. Deep learning can be used to assist the processing if necessary.

Procedure

The procedure for using the configuration tool is as follows.

Select Workpiece

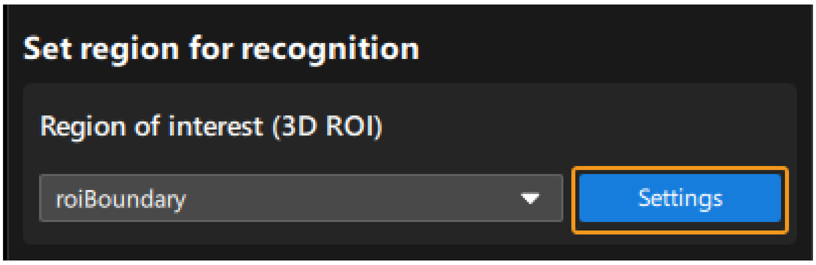
1. First, click **[Select workpiece]**, and the Workpiece Library dialog box will pop up.
2. Click **[Model Editor]** to open the Matching Model and Pick Point Editor. Create a point cloud model of the workpiece with Matching Model and Pick Point Editor and then exit the tool.
3. Select one or multiple point cloud models of the workpiece(s) in the dialog box, and then click OK. After you select the workpiece, click **[Next]** to start Preprocessing.

Only the qualified point cloud model of the workpiece will be displayed in the Workpiece Library.

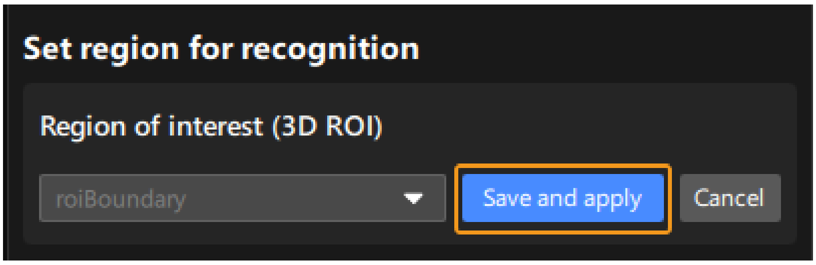
A qualified point cloud model should be added at least one pick point (which can be the geometric center of the workpiece).

Preprocessing

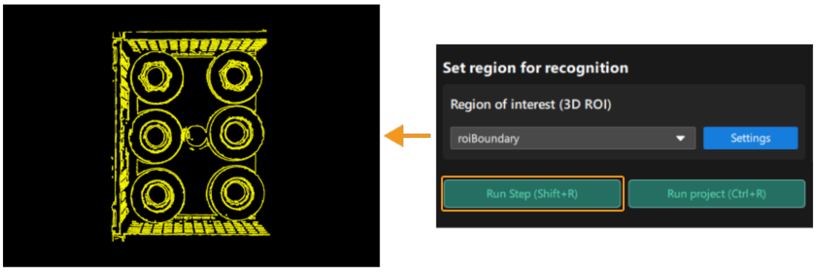
1. Click **[Settings]** and set a 3D ROI in the ROI settings window.



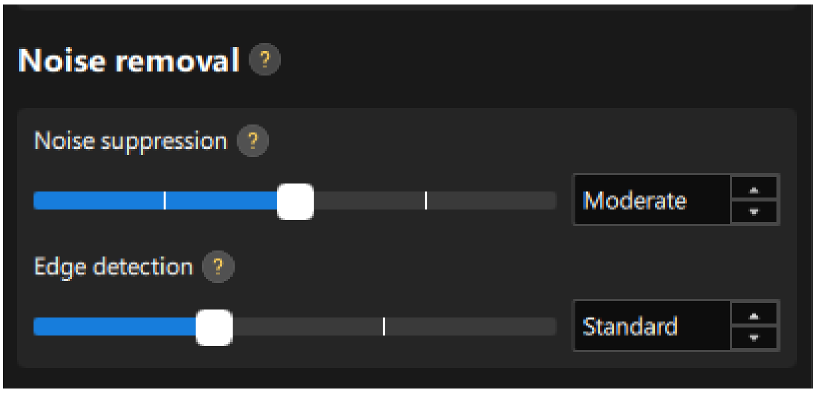
2. After you finish setting the 3D ROI, click **[Save and apply]**.



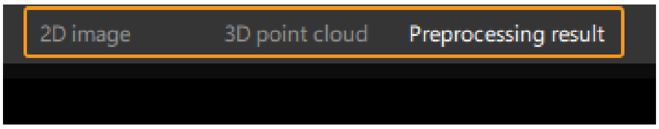
Then click **[Run Step]** and view the preprocessing result in the visualizing space on the left.



3. If noises still exist in the recognition area, you can adjust the Noise suppression and Edge detection in Noise Removal. This parameter group is only available for edge matching.



If you want to view the 2D image and point cloud after preprocessing, you can click the tabs of the visualizing space to switch the visualization type.



4. After the preprocessing, click **[Next]** to start the Recognition procedure.

Recognition

The operation workflow includes the following steps:

1. Configure parameters under 3D matching and Result judgment. Please refer to Parameter Description for detailed description of parameters and tuning recommendations.
2. (Optional) If deep learning is needed for assistance, select **Use** in the drop-down list next to **Assist processing with deep learning**, and click **[Model Package Management Tool]**. If deep learning is not required, select **Do not use**.

   Only instance segmentation is available for the “Assist processing with deep learning”.

3. In the Deep Learning Model Package Management window, import the deep learning model and exit the tool. Then select the deep learning model package in the drop-down list and adjust the Deep learning confidence threshold.
4. Click **[Run Step]** and check the 3D matching result in the visualizing space on the left.
5. If the recognition accuracy and takt meet the requirement of the project, click **[Finish]**, and the 3D Workpiece Recognition Visualized Configurator will be closed. If you select multiple target workpieces in the Select workpiece procedure, you can click **[Next workpiece]** to start configuring the next workpiece.

Parameter Description

3D matching

Operation approach

Parameter description: This parameter specifies the operation approach of the matching process. The operation approach determines the relationship between the matching accuracy and matching speed. The higher the matching accuracy, the longer the time required.

Value list: High speed, Standard, High accuracy, and Extra high accuracy.

- High speed: applicable to scenarios where workpieces are placed separately or a small number of workpieces are arranged neatly.
- Standard: an approach that balances the speed and accuracy, which is applicable to most workpiece loading scenarios.
- High accuracy: applicable to scenarios where there is a large number of workpieces or the workpieces are stacked randomly.
- Extra high accuracy: applicable to high-precision positioning scenarios.

Tuning recommendation: Please select the operation approach according to the actual requirement for recognition accuracy and speed.

Deviation correction capacity

Parameter description: This parameter is used to set the intensity of the deviation correction to the matching result. The greater the deviation correction capacity, the more likely the coarsely matched poses can be corrected to the accurately matched poses. Please note that an excessive deviation correction capability may lead to a loss of matching accuracy.

Value list: Small, Medium, and Large.

- Small: Applicable to most cases.
- Medium: Select according to the actual situation.
- Large: Select according to the actual situation.

Tuning recommendation: When there is a relatively large deviation between the matching result and the actual position of the target workpiece, please adjust this parameter.

Symmetry settings

Description: When a part of the workpiece to be recognized is symmetrical, you will need to adjust the symmetry settings to escape a local optimum of the matching result. An axis of the reference frame with the geometric center as its origin will be specified as the rotation axis, and the point cloud model will rotate around the specified axis according to the symmetry angle step and therefore the optimal matching result can be obtained.

Tuning recommendation: The tuning recommendations of parameters are as follows.

- Symmetry axis: This parameter specifies the axis of the reference frame with the geometric center as its origin, and the point cloud model will rotate around the specified axis.
- Symmetry angle (0–360): This parameter specifies the angle of rotational symmetry. If an object looks the same before and after rotating 60° around an axis, this parameter can be set to 60°.
- Min rotation angle: The minimum angle at which a workpiece can be rotated about a symmetry axis.
- Max rotation angle: The maximum angle at which a workpiece can be rotated about a symmetry axis.

Pose filtering settings

Description: When there are abnormal poses in the recognition result, this parameter can be used to filter out the abnormal poses. It is recommended to use this parameter for edge matching.

Tuning recommendation: In the edge matching mode, the point cloud model may need to be rotated by a certain angle to match with the scene point cloud, and the poses will be filtered by the point cloud model’s rotation angle. When the model’s rotation angle exceeds the “Angle difference upper threshold,” the pose will be filtered out.

Result judgment

3D matching confidence threshold

Parameter description: This parameter is used to determine whether the 3D matching recognition result is valid. If the confidence of the 3D matching recognition result is above the threshold, the result will be considered valid. The higher the confidence value, the more accurate the result is.

Default value: 0.300

Tuning recommendation: Please adjust the value according to the actual requirement.

Output count upper limit

Parameter description: This parameter sets the upper limit of the number of output 3D matching recognition results. If there are multiple 3D matching recognition results, the results will be sorted in descending order according to the 3D matching confidence, and then the results with the highest confidence will be output according to the upper limit specified by this parameter.

Default value: 1

Tuning recommendation: When there are numerous recognition results, or there is a clear requirement for the upper limit of the output results, please adjust this parameter.

- Only pick points that are also geometric center of the workpiece will be output and visualized.
- The actual number of output recognition results is not necessarily equal to the set **Output count upper limit**. If the **Output count upper limit** is set to 5, but there are 3 recognition results in total, the final number of output recognition results is 3.

4.3.11. Accept All

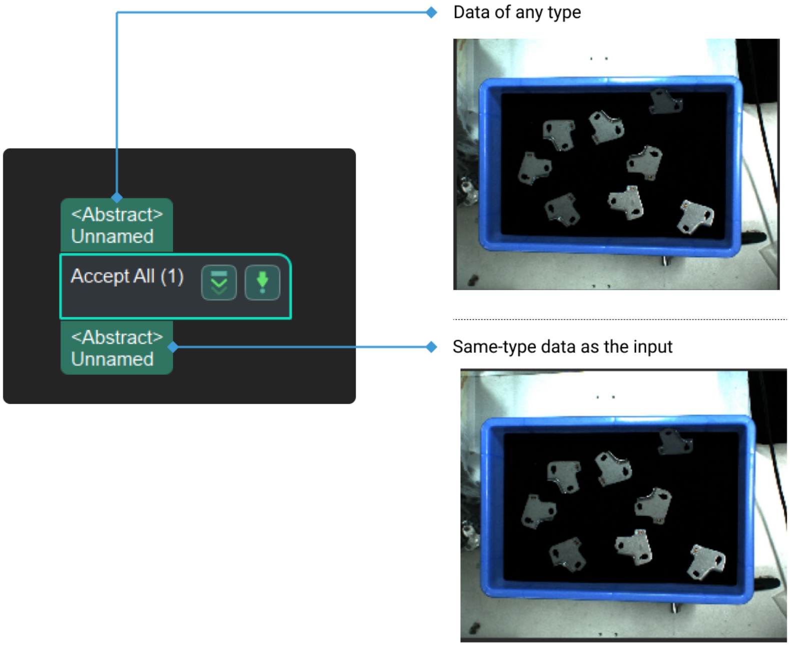
Functions

This Step can accept the output data from any other Steps.

Usage Scenario

This Step is usually used to check the data outputs from certain port during project tuning. It can also serve to temporarily store data from a specific Step with prolonged execution time, thereby reducing the overall tuning time.

Input and Output



Parameter Description

This Step does not have any parameter.

4.3.12. Add Labels to Poses

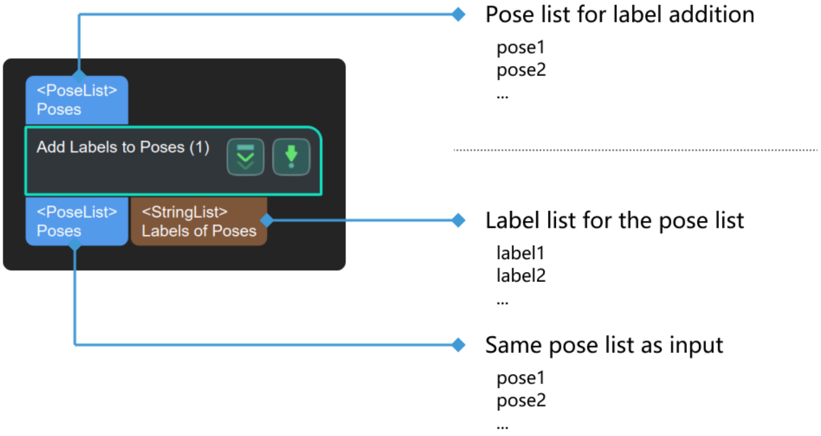
Function

Add the same label to each input pose.

Usage Scenario

General pose processing Step. No fixed usage scenarios.

Input and Output



Parameters

File Settings

Label File

Default value: addLabels.json

Instruction: The absolute path or relative path of the label file which need to include the required label names.

4.3.13. Adjust Inaccurate Poses Caused by Camera Distortion

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step corrects poses using the camera distortion coefficients (included in the intrinsic parameters).

Usage Scenario

This Step is a custom Step for special scenarios. This Step is used for camera distortion correction on poses.

Input and Output

- **Input**: Poses input to this port will be adjusted.
- **Output**: Adjusted poses.

4.3.14. Adjust Poses by Offsets

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step adjusts the pick points by calculating the pose offset between the cylinders' poses and the correct tool center points (i.e., the pick points), and outputs the actual poses of the cylinders and the offsets from the poses to the correct pick points.

Usage Scenario

This Step is usually used for picking cylindrical bars that are closely attached to each other end-to-end, preventing two bars from being picked at the same time.

Input and Output

- Input:
  1. Point clouds containing end-to-end attached cylindrical bars
  2. Robot tool center points that need to be corrected
  3. Length of a cylindrical bar
  4. Cylinder shape information
- Output:
  1. Poses of cylindrical bars
  2. Offsets of correct pick points relative to calculated poses

4.3.15. Adjust Poses by Tilt

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step translates the poses by a certain distance based on its tilt relative to the specified direction. The translation distance is the projected length of the specified vector in the direction perpendicular to the pose.

Usage Scenario

This Step is a custom Step for special scenarios.

Input and Output

- **Input**: Poses input to this port will be adjusted.
- **Output**: Adjusted poses.

4.3.16. Adjust Targets to Get Correct Path

ℹ Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step rotates and translate targets along Z axis to get the correct path.

Usage Scenario

This Step is generally used to adjust the rotation angle of the robot end effector when it moves along the path.

Input and Output

- **Input**: Path targets input to this port will be rotated and translated along the Z direction.
- **Output**: Path targets after rotation and translation along the Z direction.

4.3.17. Move Poses to Point Cloud Surfaces

ℹ Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step translates the input poses along with the specified directions so that the centers of the translated poses are on the surfaces of the input point clouds.

Usage Scenario

This Step is usually used to correct the pick points to point cloud surfaces.

Input and Output

- **Input:**
  1. Point clouds input to this port will be used as references for adjusting the poses.
  2. Poses input to this port will be adjusted to the point clouds’ surfaces.
- **Output:**
  1. Poses adjusted to the point clouds’ surfaces.
  2. Color point cloud for visualization.

4.3.18. Change Circular Motion Direction of Path

ℹ Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step adjusts the circular motion direction of a path.

Usage Scenario

This Step is usually used in door/glass gluing applications. If the robot encounters a soft limit during the movement, the start and end points will be switched, and the robot will move along the newly set targets.

Input and Output

- Input:
  1. Original path consisting of the targets input to this port will be adjusted.
  2. List of robot end effector targets in the original path.
  3. List of robot end effector targets in the original path in the flange reference frame.
- Output:
  1. Adjusted targets that form the path whose circular motion direction has been adjusted.

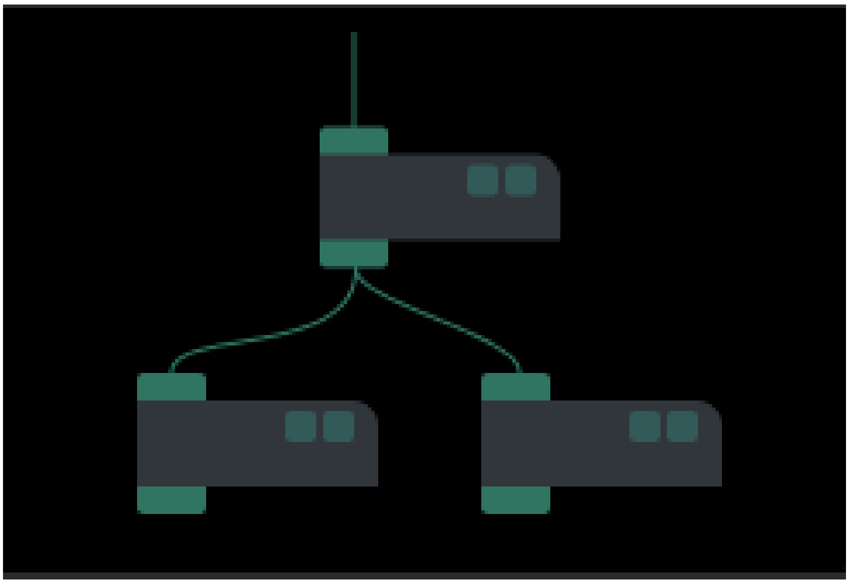
4.3.19. Allocator

Function

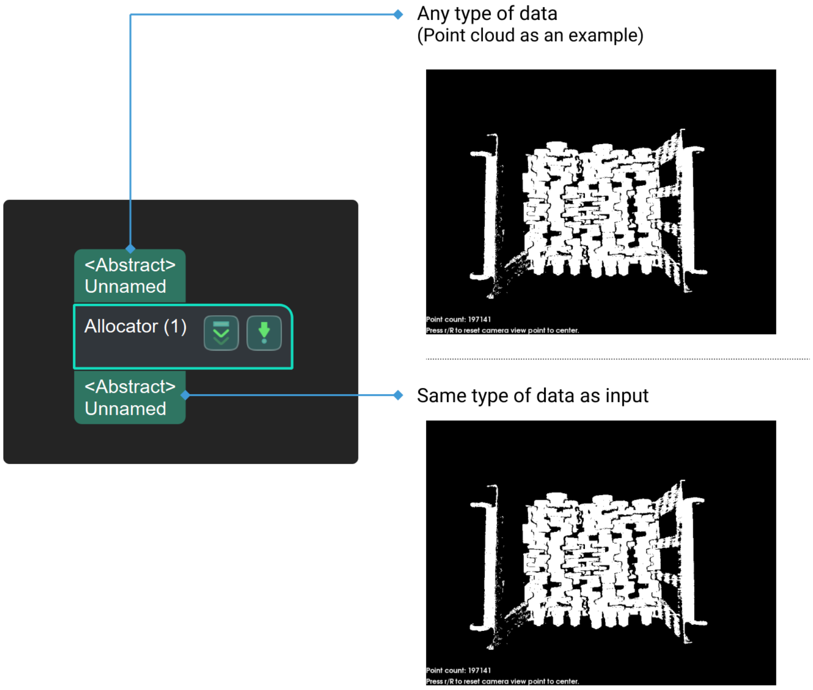
This Step is used to assign input ports for Procedures.

Usage Scenario

This Step is usually used to reduce the number of input ports of Procedures, transfer data into Procedures, and perform port assignments within Procedures.



Input and Output


Any type of data
(Point cloud as an example)
<Abstract>
Unnamed
Allocator (1)
<Abstract>
Unnamed
Same type of data as input
Point count: 197141
Press r/R to reset camera view point to center.
Point count: 197141
Press r/R to reset camera view point to center.

Parameter

This Step does not have any parameter.

4.3.20. Apply Masks to Image

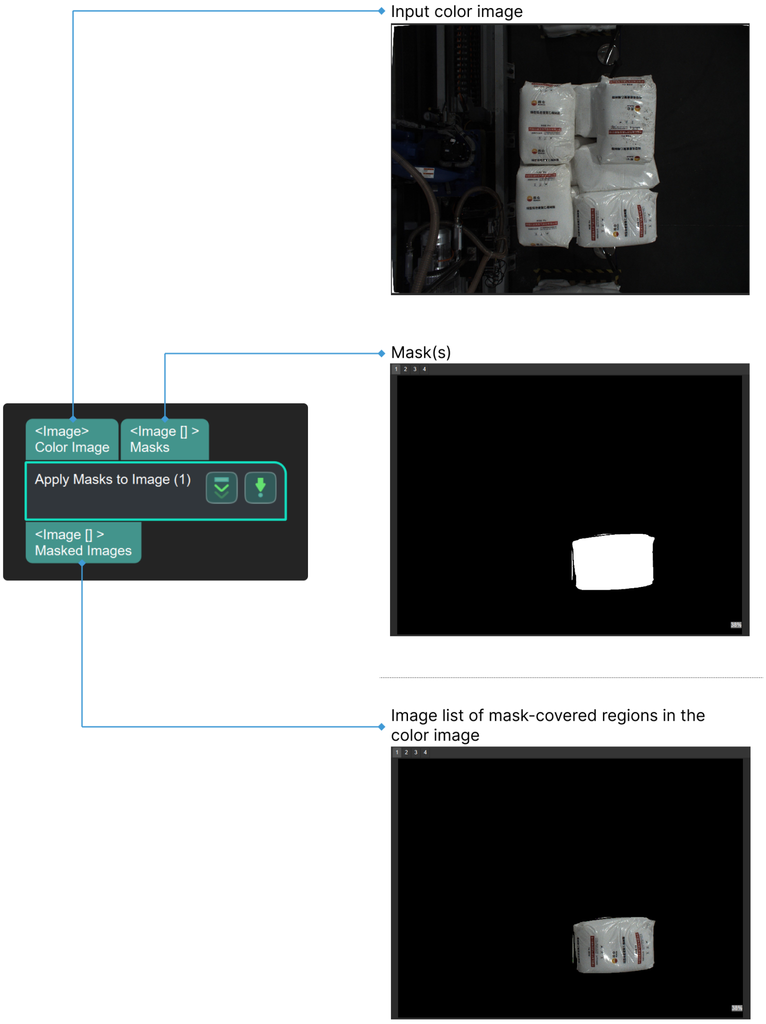
Function

Apply masks to a color image. The regions covered by the masks will be extracted and the rest of the color image will be filled with black.

Usage Scenario

Usually used with **Calc Mask for Highest Layer** to ignore irrelevant parts in the scene, avoid interference, and provide better input for subsequent deep learning Steps.

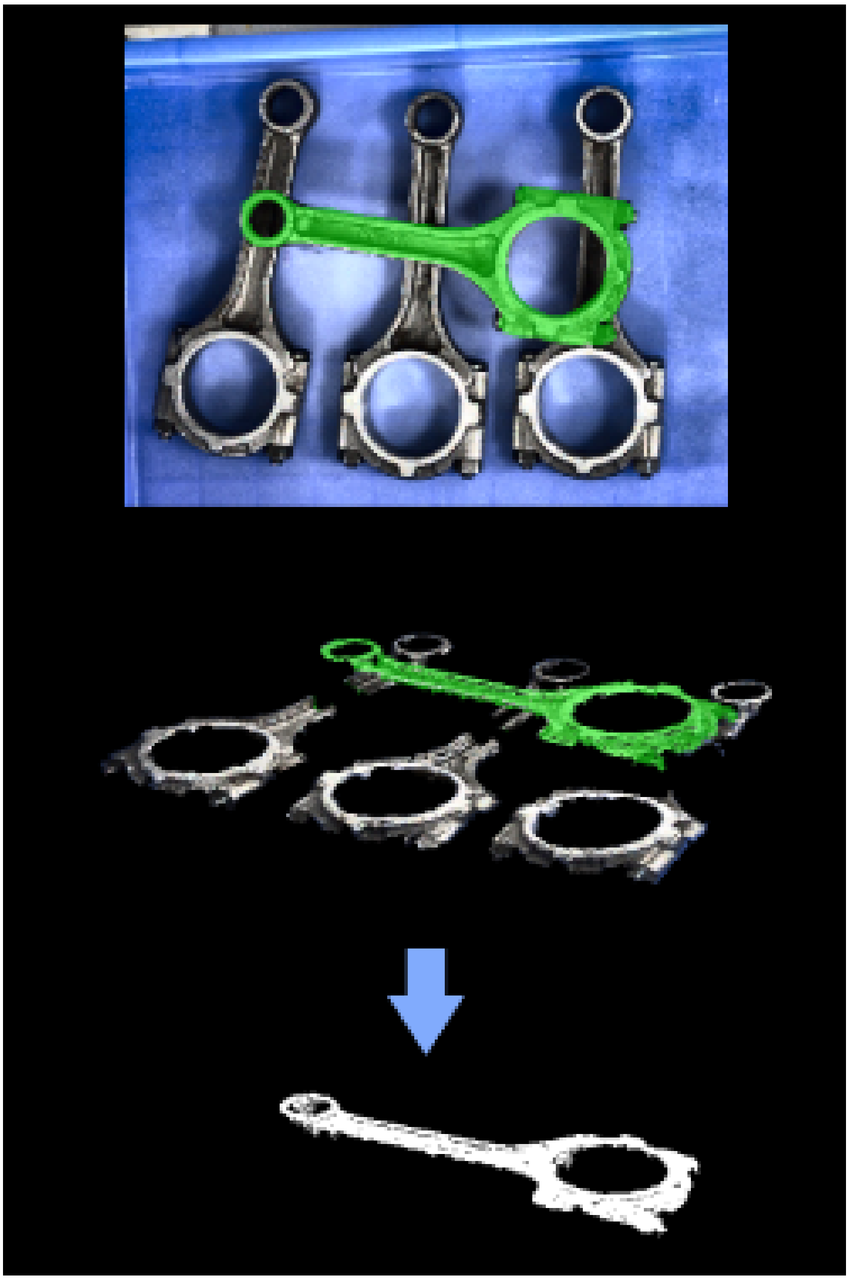
Input and Output



4.3.21. Apply Masks to Point Cloud

Function

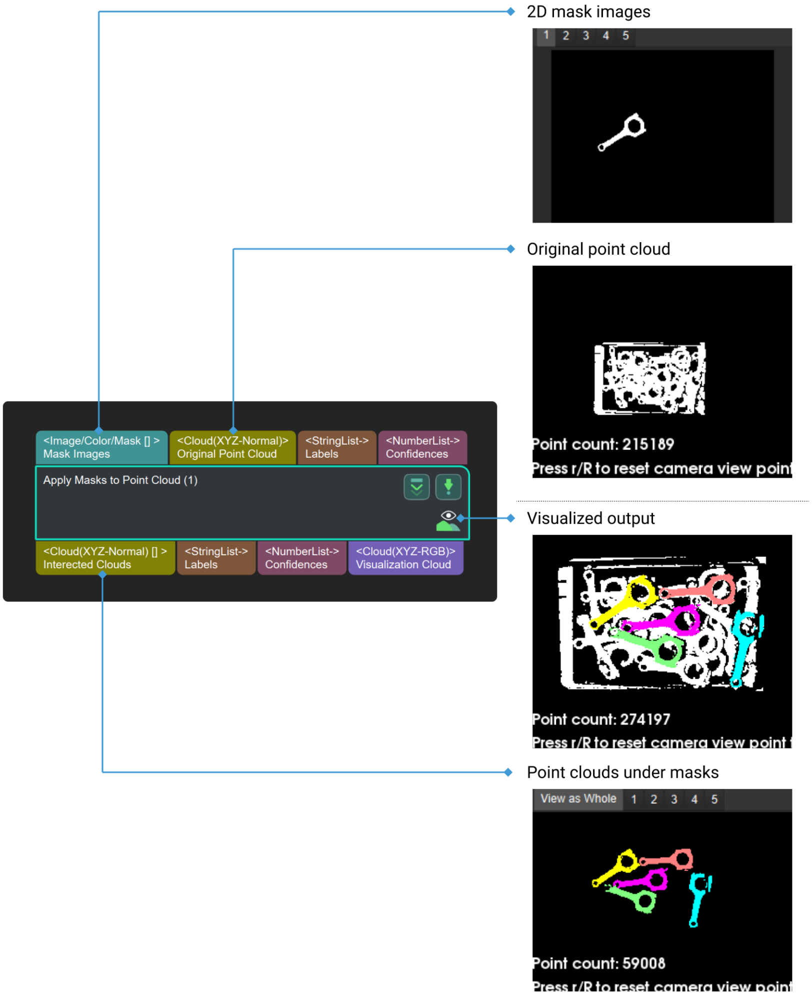
The Step applies the mask to the point cloud. Points covered by the mask will be kept while points outside the mask will be eliminated.



Usage Scenario

This Step is usually connected after the Step Instance Segmentation to extract point clouds that are within the mask.

Input and Output


2D mask images
1 2 3 4 5
Original point cloud
Point count: 215189
Press r/R to reset camera view point
<Image/Color/Mask [] > Mask Images
<Cloud(XYZ-Normal)> Original Point Cloud
<StringList-> Labels
<NumberList-> Confidences
Apply Masks to Point Cloud (1)
<Cloud(XYZ-Normal) [] > Interected Clouds
<StringList-> Labels
<NumberList-> Confidences
<Cloud(XYZ-RGB)> Visualization Cloud
Visualized output
Point count: 274197
Press r/R to reset camera view point
Point clouds under masks
View as Whole 1 2 3 4 5
Point count: 59008
Press r/R to reset camera view point

Parameter Description

This Step does not have any parameter.

4.3.22. Extract 2D Path Function

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

From the input original path image and object point cloud, this Step extracts 2D path targets.

Usage Scenario

This Step is a custom Step for special scenarios.

Input and Output

- Input:
  1. Original image of the path

2. Filtered image of the path (used to subtract from the original image to get the path in the image)
3. Object point cloud
4. Object color image

- Output:
  1. Orthographic projection 2D image of the path
  2. List of 2D targets of the generated path

4.3.23. Background Subtraction

> Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step removes the background of a 2D image, and keeps the foreground.

Usage Scenario

This Step is to avoid the interference caused by the 2D image background for the subsequent image processing.

Input and Output

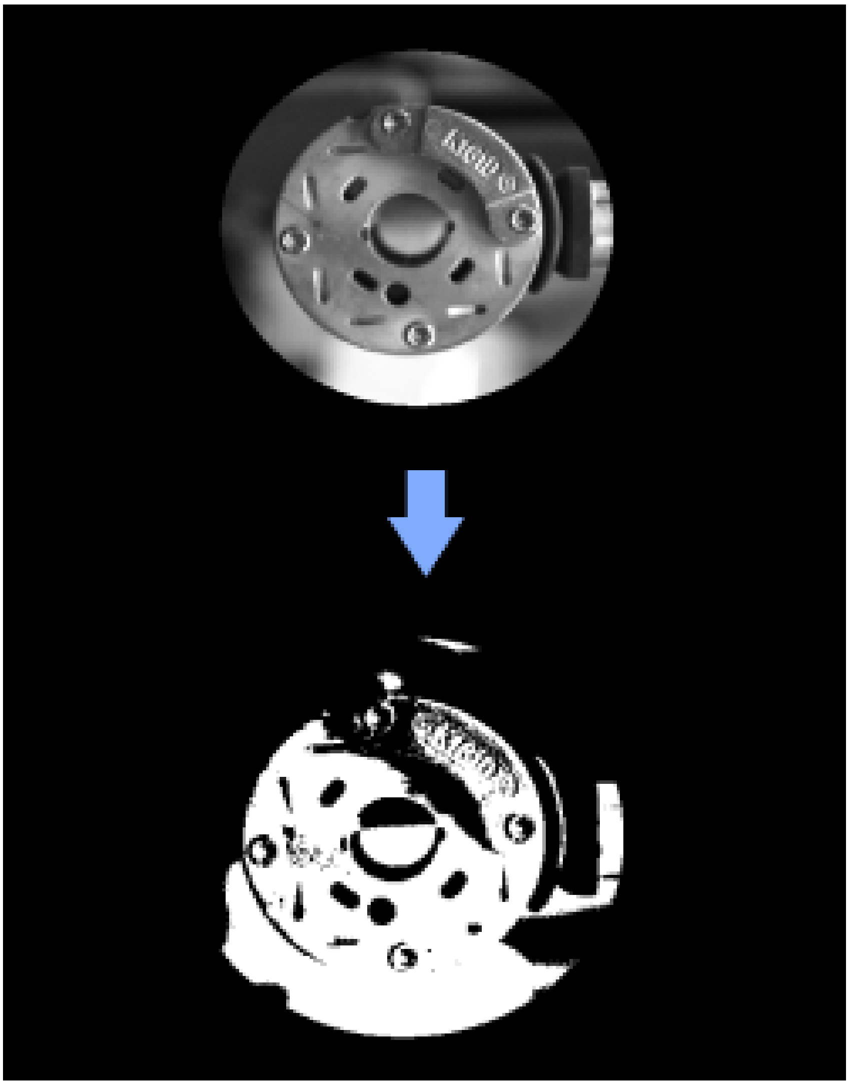
- Input:
  1. Image input to this port will have the background removed.
  2. Depth map corresponding to the input image.
- Output:
  1. Image after background removal.

4.3.24. Image Thresholding

Function

This Step filters image pixels according to the set threshold and binarizes the pixels above and below the threshold according to the set rules.

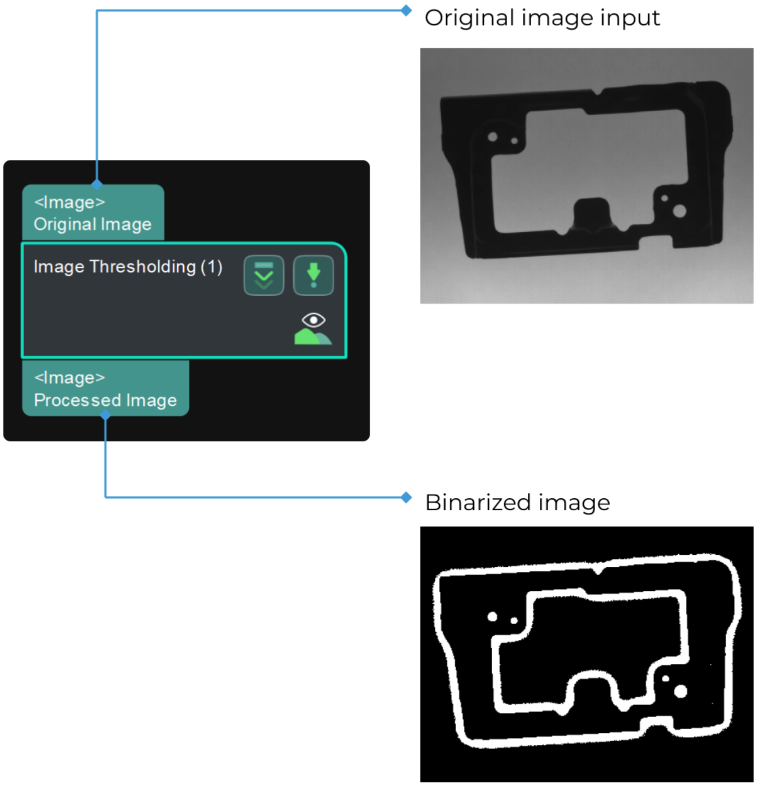
> Binary images are images whose pixels have only two possible intensity values. Numerically, the two values are often 0 for black, and 255 for white.
>
> Image binarization is the process of converting a non-binary image to a binary image.



Usage Scenario

This Step is a general image processing Step. It is generally used to segment pixels that meet threshold conditions on 2D images.

Input and Output



Parameter Description

This Step provides four segment types for separating the target object from its background.

- AdaptiveThreshold: a global adaptive thresholding method that includes two types of operation (THRESH_BINARY, THRESH_BINARY_INV).
- DualThreshold: a thresholding method with both a low threshold and a high threshold to segment pixels that meet the conditions.
- DynamicThreshold: a dynamic thresholding method that provides four thresholding types (Bright, Dark, In range, Out of range) and three image filter methods (Mean filter, Gaussian filter, Median filter). In addition, the pixel value offset and filter kernel size can be adjusted.
- Threshold: a thresholding method with a global fixed threshold. With this method, a fixed threshold is set, and seven operation types (THRESH_BINARY, THRESH_BINARY_INV, THRESH_TRUNC, THRESH_TOZERO, THRESH_TOZERO_INV, THRESH_OTSU, THRESH_TRIANGLE) are introduced to segment pixels that meet the conditions.

AdaptiveThreshold

Operation

Description: This parameter is used to select the operation type for separating the target object from its background.

Value list: THRESH_BINARY and THRESH_BINARY_INV

- THRESH_BINARY: generates a mask of pixels whose values are higher than the threshold.
- THRESH_BINARY_INV: generates a mask of pixels whose values are lower than the

threshold.

Default value: THRESH_BINARY

Tuning recommendation: Please select the operation type according to the actual requirement.

DualThreshold

Low Threshold

Description: If the low threshold is set to be lower than the high threshold, the pixels with values between the two thresholds will be set to 255, and other pixels will be set to 0. If the low threshold is set to be higher than the high threshold, the pixels with values beyond the interval set by the two thresholds will be set to 255, and other pixels will be set to 0.

Default value: 0

Tuning recommendation: Please set a proper value according to the actual requirement.

High Threshold

Description: See "Low Threshold."

Default value: 100

Tuning recommendation: Please set a proper value according to the actual requirement.

DynamicThreshold

Thresholding Type

Description: This parameter determines whether the values of pixels should be set to 255 or 0.

Value list: Bright, Dark, In Range, and Out of Range. If `P_o` is a pixel in the original image, `P_f` is the pixel corresponding to `P_o` in the filtered image, and `offset` is the set value of **Pixel Value Offset**, then the thresholding types can be explained as follows:

- Bright: If `P_o ≥ P_f + offset`, then `P_o` will be set to 255, or else it will be set to 0.
- Dark: If `P_o ≤ P_f - offset`, then `P_o` will be set to 255, or else it will be set to 0.
- In Range: If `P_f - offset ≤ P_o ≤ P_f + offset`, then `P_o` will be set to 255, or else it will be set to 0.
- Out of Range: If `P_o < P_f - offset` or `P_o > P_f + offset`, then `P_o` will be set to 255, or else it will be set to 0.

Image Filter

Description: This parameter is used to select the filter to apply to the image.

Value list: Mean filter, Gaussian filter, and Median filter

- Mean filter: Smooth the image by replacing the center value in the sliding window with the average of all the pixel values in the window.
- Gaussian filter: Smooth the image and remove detail and noise.
- Median filter: Replace each pixel with the median of neighboring pixels in the window.

Default value: Mean filter

Tuning recommendation: Please select a proper image filter method according to the actual requirement.

Filter Kernel Size

Description: This parameter sets the sliding window size (pixel-wise edge length) of the filter.

Default value: 3

Tuning recommendation: Please input an odd number because there should always be a center pixel in the window. Even numbers input will be incremented by one.

Pixel Value Offset

Description: The offset pixel value to apply to all pixels of the image after filtering.

Default value: 15.00

Threshold

Threshold (0–255)

Description: This parameter sets a threshold for filtering image pixels.

Default value: 128

Tuning recommendation: Please set a proper value according to the actual requirement.

Operation

Description: This parameter is used to select the operation type for separating the target object from its background.

Value list: THRESH_BINARY, THRESH_BINARY_INV, THRESH_TRUNC, THRESH_TOZERO, THRESH_TOZERO_INV, THRESH_OTSU, and THRESH_TRIANGLE

- THRESH_BINARY: If the original pixel value is above the set threshold, the value will be set to 255, or else it will be set to 0.
- THRESH_BINARY_INV: If the original pixel value is above the set threshold, the value will be set to 0, or else it will be set to 255.
- THRESH_TRUNC: If the original pixel value is above the set threshold, the value will be set to the same value as the set threshold, or else the original value will be retained.
- THRESH_TOZERO: If the original pixel value is above the set threshold, the original value will be retained, or else it will be set to 0.
- THRESH_TOZERO_INV: If the original pixel value is above the set threshold, the value will be set to 0, or else the original value will be retained.
- THRESH_OTSU: Find a global threshold with the Otsu’s method.
- THRESH_TRIANGLE: Find a global threshold with the triangle method.

Default value: THRESH_BINARY

Tuning recommendation: THRESH_BINARY or THRESH_BINARY_INV is recommended to use.

4.3.25. Blob Analysis

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step binarizes the input image, detects Blobs in the image, calculates their contours, pixel-wise areas, circularity, and rectangularity, and validates the Blobs one by one according to the set standard.

Usage Scenario

This Step is usually used to detect objects in 2D images in measurement scenarios.

Input and Output

- Input:
  1. Image input to this port will be binarized and have Blob(s) calculated.
  2. 2D transformation input at this port is used to transform the ROI set in a previous execution to the appropriate position.
- Output:
  1. Visualization image.
  2. Pixel-wise coordinates of Blobs' contours' vertices.
  3. Pixel-wise coordinates of the Blobs' min circumscribed rectangles' vertices.
  4. Number of Blobs output.
  5. Pixel-wise areas of Blobs.
  6. Circularity values of the Blobs obtained by dividing the areas of the Blobs by the areas of circles whose perimeters are the perimeters of the Blobs.
  7. Rectangularity values of the Blobs obtained by dividing the areas of the Blobs by their minimum circumscribed rectangles' areas.
  8. Validation result Boolean list of the output Blobs. The validation rules are set by the parameters under "Validation Settings".
  9. Mask images of the Blobs. The non-zero pixel value regions are the masks.

4.3.26. Boolean List Logical Operation

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

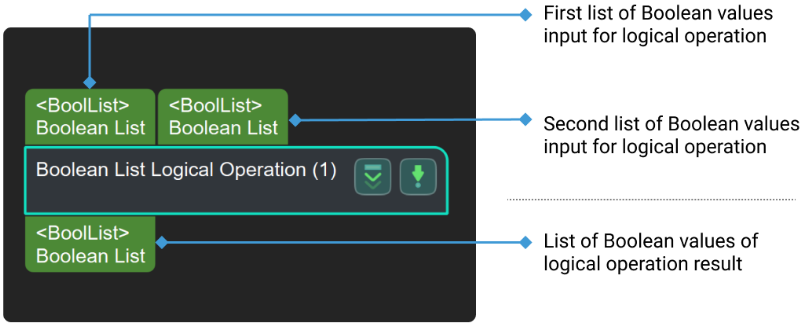
Function

According to the set logical operator, perform a logical operation on the input Boolean value (True/False) lists.

Usage Scenario

A general logic operation Step. No fixed usage scenarios.

Input and Output


<BoolList>
Boolean List
<BoolList>
Boolean List
Boolean List Logical Operation (1)
<BoolList>
Boolean List
First list of Boolean values input for logical operation
Second list of Boolean values input for logical operation
List of Boolean values of logical operation result

4.3.27. Bridge

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step transfers data to other projects.

Usage Scenario

This Step is used when saving and loading data is time-consuming and data needs to be transferred across projects.

Input and Output

- **Input**: none
- **Output**: none

4.3.28. Calc Absolute Values

The contents of this Step are no longer under maintenance. Please use the new Step Numerical Operation. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

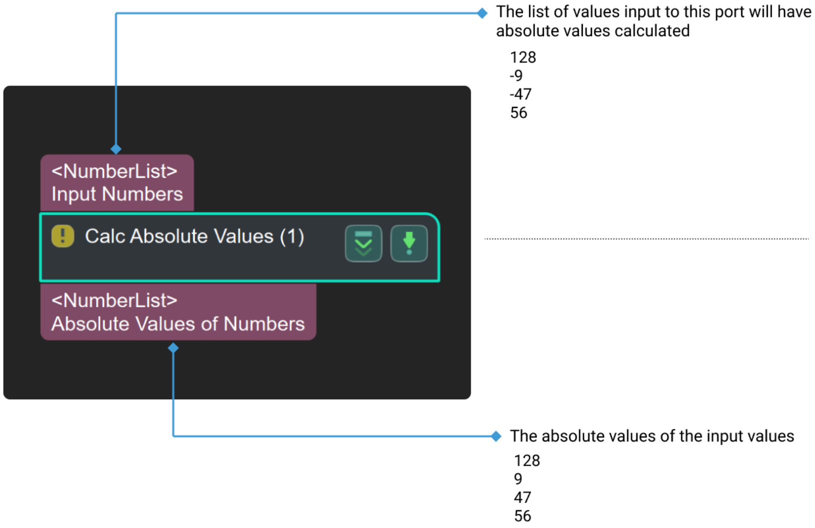
Function

Calculate the absolute values of a list of input values.

Usage Scenario

This Step is outdated. Please use the newer Step Numeric Operation that has more functions instead.

Input and Output



The list of values input to this port will have absolute values calculated

128
-9
-47
56

The absolute values of the input values

128
9
47
56

4.3.29. Calc Angles between Vector3Ds

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step calculates the angle between two Vector3D vectors. Multiple pairs of vectors can be input at a time.

Usage Scenario

This Step is a general Vector3D angle calculation Step. This Step has no fixed usage scenarios.

Input and Output

- Input:
  1. Angles between the Vector3D vectors of this port and the corresponding Vector3D vectors of the other port will be calculated and output.
  2. Angles between the Vector3D vectors of this port and the corresponding Vector3D vectors of the other port will be calculated and output.
- Output:
  1. Angles between the Vector3D vectors.

4.3.30. Calc Areas of Masks

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

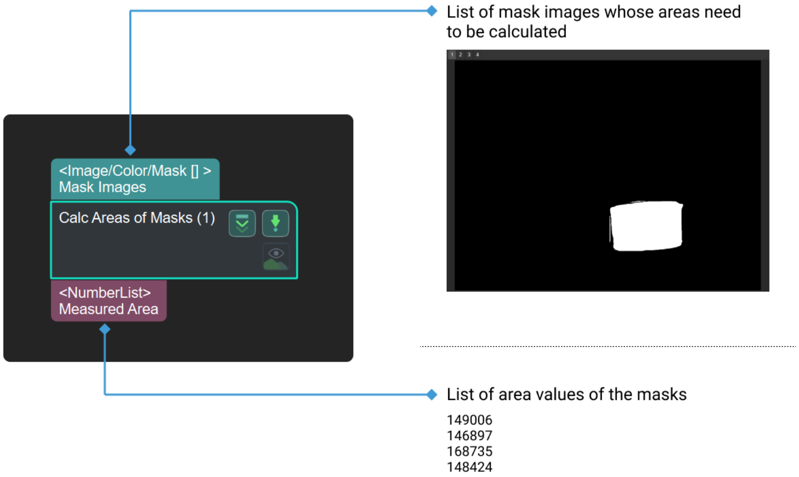
Function

Calculate the areas of multiple mask images.

Usage Scenario

This Step calculates mask areas which facilitate follow-up sorting, filtering, and other operations.

Input and Output



4.3.31. Calc Box Dimensions

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

Using the depth map and the mask of the boxes, this Step calculates the actual dimensions of the top surfaces of the boxes, and outputs the bin dimensions and top surface point cloud.

Usage Scenario

This Step is a custom Step for special scenarios.

Input and Output

- Input:
  1. Dimensions of the boxes in the depth map input to this port will be calculated
  2. Masks for the boxes in the depth map
- Output:
  1. Calculated top surface dimensions of the boxes
  2. Point clouds of the top surfaces of the boxes

4.3.32. Calc Vector3D Cross Products

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step computes the cross products of two lists of Vector3D vectors one to one.

Usage Scenario

This Step is a general vector computation Step. No fixed usage scenarios.

Input and Output

- Input:
  1. First list of Vector3D vectors for computing the cross products
  2. Second list of Vector3D vectors for computing the cross products
- Output:
  1. Computed cross products

4.3.33. Calc Diagonal Lengths

Function

After projecting the box-shaped objects to the XY plane, this Step calculates the diagonal lengths of the projections.

Usage Scenario

This Step is a custom Step for special scenarios.

Input and Output

- **Input**: The object dimensions entered at this port will be used for diagonal length calculation.
- **Output**: calculated diagonal lengths.

4.3.34. Calc Diameter and Thickness

> ℹ Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step calculates the diameters and thicknesses of the input point clouds.

Usage Scenario

This Step is a custom Step for special scenarios.

Input and Output

- Input:
  1. Point clouds input to this port will have diameters calculated.
  2. Point clouds input to this port will have thicknesses calculated.

3. Poses corresponding to point clouds.

- Output:

1. Calculated diameters
2. Calculated thicknesses
3. Labels indicating whether the corresponding calculation results are available

4.3.35. Calc Distance between Two Poses along Specified Direction

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

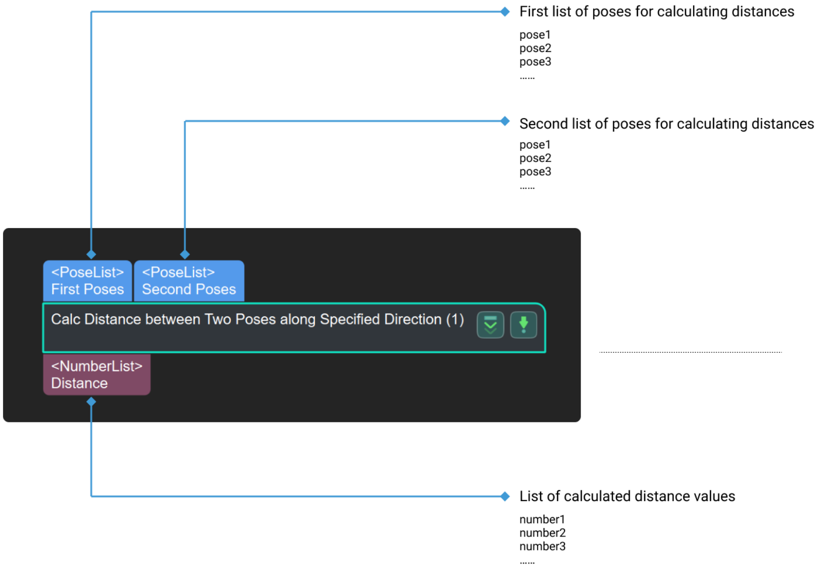
Function

Calculates the projection length of the vector formed by the two pose centers on a specified axis or combination of axes.

Usage Scenario

A general pose distance calculation Step. No fixed usage scenarios.

Input and Output


First list of poses for calculating distances
pose1
pose2
pose3
......
Second list of poses for calculating distances
pose1
pose2
pose3
......
<PoseList>
First Poses
<PoseList>
Second Poses
Calc Distance between Two Poses along Specified Direction (1)
<NumberList>
Distance
List of calculated distance values
number1
number2
number3
......

4.3.36. Calc Distances between Poses

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step forms a vector from two pose centers, and calculates the length of this vector, or the length of the vector’s projection on the XY plane. Multiple pairs of poses can be input at a time.

Usage Scenario

This Step is a general pose distance calculation Step. This Step has no fixed usage scenarios.

Input and Output

- Input:
  1. Poses input to this port will be used as the starting points of the distance vectors.
  2. Poses input to this port will be used as the endpoints of the distance vectors.
- Output:
  1. Lengths of distance vectors or lengths of projections of distance vectors on the XY planes.

4.3.37. Calc Vector3D Dot Products

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step computes the dot products of two lists of Vector3D vectors one to one.

Usage Scenario

This Step is a general vector computation Step. No fixed usage scenarios.

Input and Output

- Input:
  1. First list of Vector3D vectors for computing the dot products
  2. Second list of Vector3D vectors for computing the dot products
- Output:
  1. Dot products computed

4.3.38. Calc Edge Point Cloud Normals

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step calculates the normals of edge point cloud.

Usage Scenario

This Step is a general point cloud calculation Step. This Step has no fixed usage scenarios.

Input and Output

- **Input**: Object edge point cloud input to this port will have its normals calculated.
- **Output**: Object edge point cloud with normals.

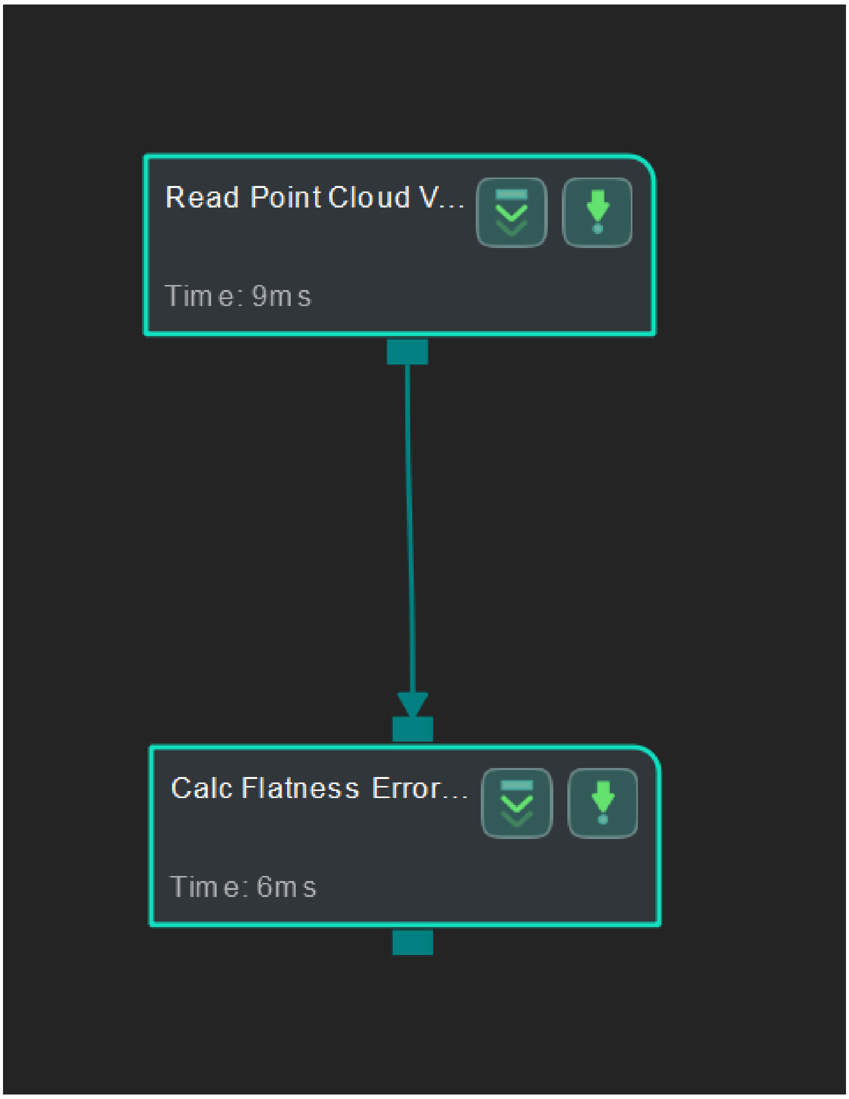
4.3.39. Calc Flatness Error

This Step is used to calculate the flatness error of the planar point cloud.

- Degree of flatness is a critical indicator to maintain quality control throughout the manufacturing process. The surfaces of products are usually required to achieve a desired degree of flatness.
- Flatness error describes the deviation of a measured plane surface from the theoretical plane surface which has the desired degree of flatness.

Before using this Step, please refer to Getting Started with Measurement Mode to learn about basics of Measurement Mode.

Sample Application



Connect Steps

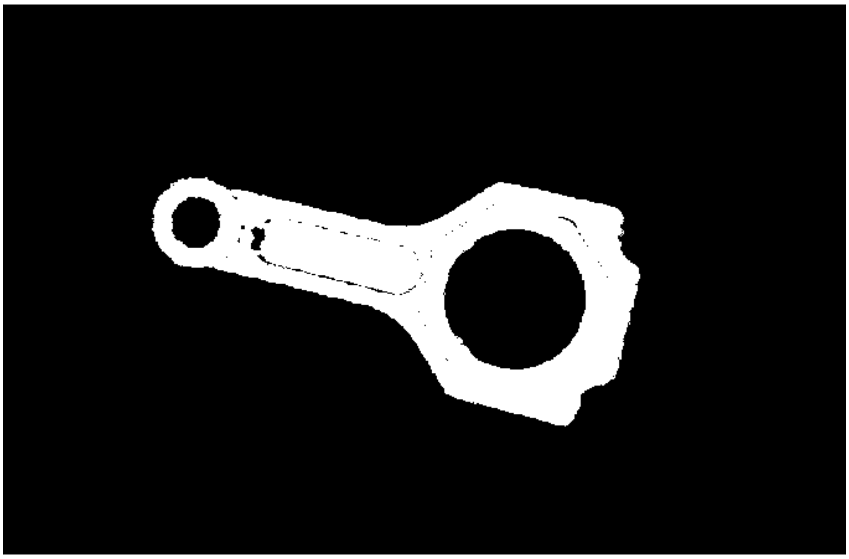
Click Calc Flatness Error, go to the **Step Input Source Selection** panel, and then select Read Point Cloud V2_1_Point Cloud With Normals as Input 1 (Planar Point Cloud).

Configure parameters

Configure in Read Point Cloud V2

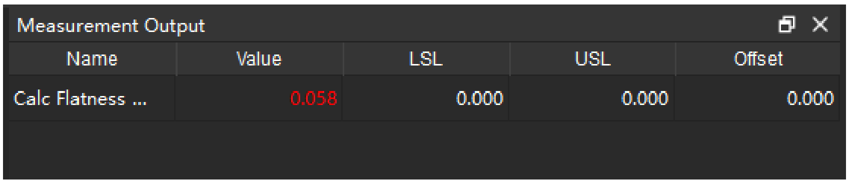
Click the Step and configure the Point Cloud Source and other parameters in the **Step Parameters** panel. After configuration, click **[Run]** to run the project.

The point cloud used in this sample application is shown below.



Read the result

After running the project, click Calc Flatness Error and check the flatness error in the **Measurement Output** panel.



Measurement Output

Name	Value	LSL	USL	Offset
Calc Flatness ...	0.058	0.000	0.000	0.000

You can set the LSL (lower specification limit) and USL (upper specification limit) in **Measurement Output** and **Result View** panels to check whether the measurement conforms to the specification.

4.3.40. Calc Histogram

This Step is outdated. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step computes a histogram of pixel intensities in a 2D image.

Usage Scenario

This Step is a custom Step for special scenarios.

Input and Output

- Input:
  1. 2D image input to this port will have a pixel intensity histogram calculated.
  2. Masks input to this port specifies the calculation range of the histogram.
- Output:
  1. Calculated histogram.

4.3.41. Calc Included Angles between Specified Axes of Poses

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

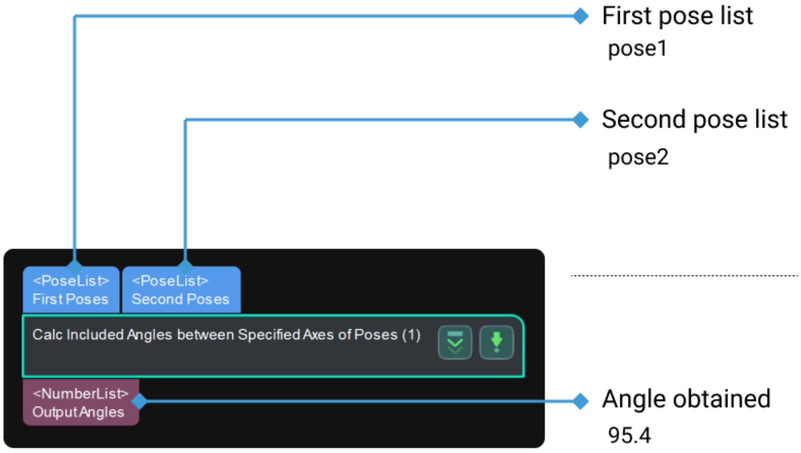
Function

Calculate angles between specified axes of input pose pairs.

Usage Scenario

General angle calculation Step. No fixed usage scenarios.

Input and Output



4.3.42. Calc Intersection between Two Line Segments

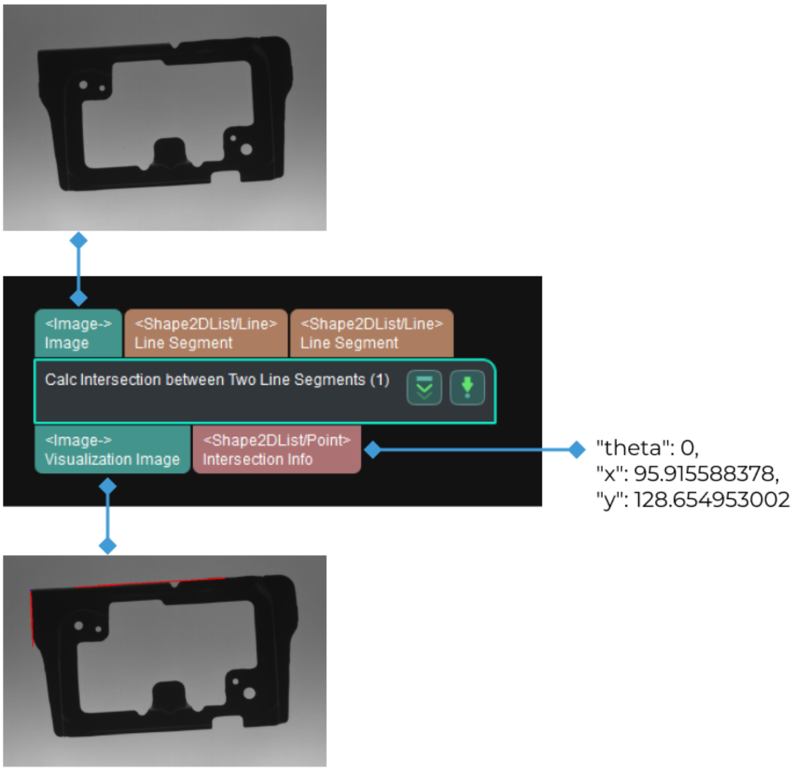
Function

This Step is used to calculate the pixel-wise coordinate of the intersection between two line segments.

Usage Scenario

This Step is generally used to locate object feature points in measurement scenarios.

Input and Output


<Image-> Image
<Shape2DList/Line> Line Segment
<Shape2DList/Line> Line Segment
Calc Intersection between Two Line Segments (1)
<Image-> Visualization Image
<Shape2DList/Point> Intersection Info

```
"theta": 0,
"x": 95.915588378,
"y": 128.654953002
```

Parameter Tuning

Intersection Type

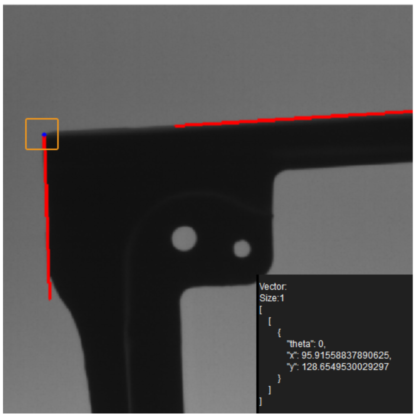
Description: Specifies the type of intersection you want to calculate.

Value list: Line Extension and Within Endpoints.

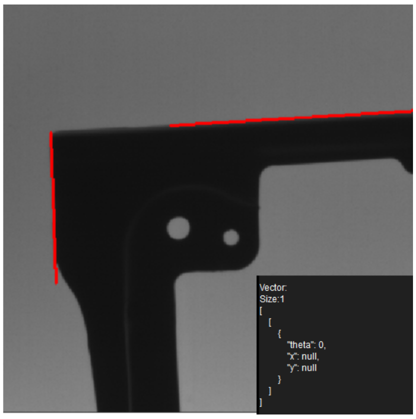
Default value: Within Endpoints

- Within Endpoints: Calculate the intersection within the endpoints of a line segment. If the two line segments do not intersect, the output intersection information will be null.
- Line Extension: Calculate the intersection on the line extension of the line segment. If the two line segments are parallel or overlap, the output intersection information will be null.

As shown below, the figure on the left shows the calculated result when the intersection type is **Line Extension**, and the figure on the right shows the calculated result when the intersection type is **Within Endpoints**. The blue point represents the intersection.



```
Vector:
Size:1
[
    [
        {
            "theta": 0,
            "x": 95.91558837890625,
            "y": 128.6549530029297
        }
    ]
]
```



```
Vector:
Size:1
[
    [
        {
            "theta": 0,
            "x": null,
            "y": null
        }
    ]
]
```

4.3.43. Calc Intersections between Line Segment and Circle

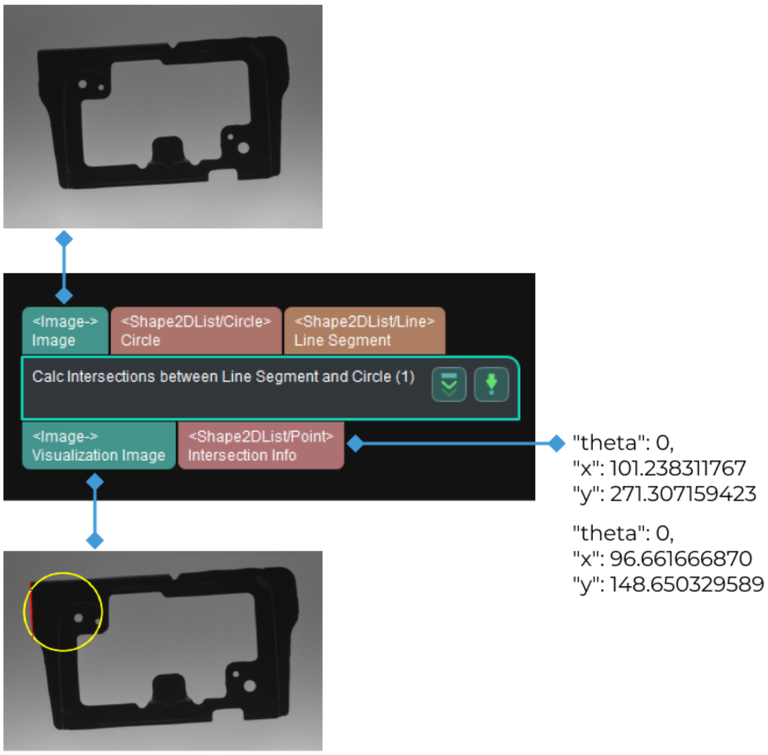
Function

This Step is used to calculate the pixel-wise coordinate of the intersections between a circle and a line segment.

Usage Scenario

This Step is generally used to locate object feature points in measurement scenarios.

Input and Output



<Image-> Image
<Shape2DList/Circle> Circle
<Shape2DList/Line> Line Segment
Calc Intersections between Line Segment and Circle (1)
<Image-> Visualization Image
<Shape2DList/Point> Intersection Info

"theta": 0,
"x": 101.238311767
"y": 271.307159423

"theta": 0,
"x": 96.661666870
"y": 148.650329589

Parameter Tuning

Intersection Type

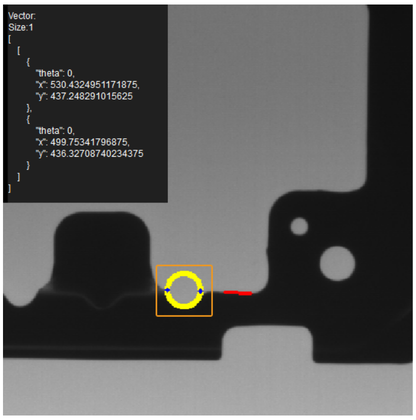
Description: Specifies the type of intersection you want to calculate.

Value list: Line Extension and Within Endpoints.

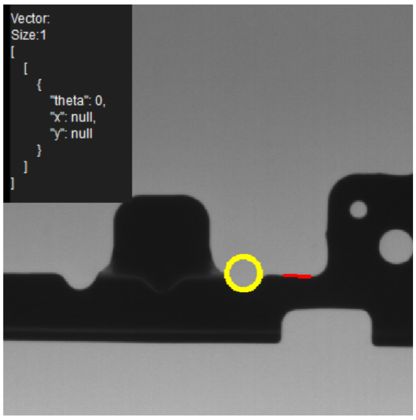
Default value: Within Endpoints

- Within Endpoints: Calculate the intersections within the endpoints of a line segment. If the line segment does not intersect with the circle, the output intersection information will be null.
- Line Extension: Calculate the intersections on the line extension of the line segment. If the line extension and the circle do not intersect, the output intersection information will be null.

As shown below, the figure on the left shows the calculated result when the intersection type is **Line Extension**, and the figure on the right shows the calculated result when the intersection type is **Within Endpoints**. The blue point represents the intersection.



```
Vector:
Size:1
[
  [
    {
      "theta": 0,
      "x": 530.4324951171875,
      "y": 437.248291015625
    },
    {
      "theta": 0,
      "x": 499.75341796875,
      "y": 436.32708740234375
    }
  ]
]
```



```
Vector:
Size:1
[
  [
    {
      "theta": 0,
      "x": null,
      "y": null
    }
  ]
]
```

4.3.44. Calc Point Cloud Spans along Axes

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

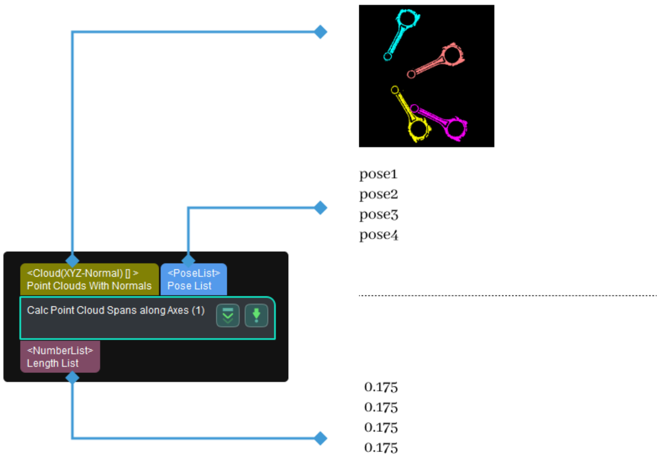
Function

This Step calculates the projected lengths of point clouds in the directions of the specified axes of the point clouds’ poses.

Usage Scenario

This Step is a general point cloud property calculation Step. There is no fixed usage scenario.

Input and Output



4.3.45. Calc Lengths of Vector3Ds

Function

This Step calculates the lengths of the Vector3D vectors.

Usage Scenario

This Step is a general vector length calculation Step. No fixed usage scenarios.

Input and Output

- **Input**: Lengths of vectors input to this port will be calculated.
- **Output**: calculated vector lengths.

4.3.46. Calc Mask Distances

> Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step calculates the distances between two lists of input masks one to one. The calculation method of the distance can be set in the parameters.

Usage Scenario

This Step is a custom Step for special scenarios.

Input and Output

- Input:
  1. Masks input at this port will have distances to masks input at the other port calculated.
  2. Masks input at this port will have distances to masks input at the other port calculated.
  3. Mask input to this port represents the center of rotation (input when the calculation method is set to “Rot180SubMassCenter”).

- Output:
  1. Calculated distances between masks

4.3.47. Calc Mask Spans on Given Lines

> Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step draws 2D lines from 3D poses. If the projected point of a pose falls inside the corresponding mask, this Step will calculate the distance between the two intersection points of the line with the farthest edges of the mask. Otherwise, this Step will consider the calculated distance invalid and set it to 0.

Usage Scenario

This Step is typically used to calculate mask spans.

Input and Output

- Input:
  1. Poses input to this port will be projected to the corresponding 2D masks.
  2. Masks input to this port will have their spans along specified lines calculated.
- Output:
  1. Spans of masks on the specified lines.

4.3.48. Calc Mean Gray Value

> Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step calculates the mean gray value of pixels in the input 2D image.

Usage Scenario

This Step is a general 2D image processing Step. No fixed usage scenarios.

Input and Output

- Input:
  1. Regions covered by the masks in the 2D image input to this port will have the mean gray value calculated.
  2. Masks for the regions where the mean gray value needs to be calculated.
- Output:
  1. Calculated mean gray value

4.3.49. Calc Midpoint of Specified Side of Rectangle

ℹ Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step calculates the midpoint of the specified side of the bounding rectangle of each non-zero pixel region in the mask image.

Usage Scenario

This Step is usually used in scenarios such as hatch door/glass gluing to calculate the starting point of the gluing path.

Input and Output

- **Input**: Non-zero pixel regions in the mask image input by this port will have bounding rectangles calculated.
- **Output**: midpoints of the specified sides of the bounding rectangles.

4.3.50. Calc Minimum Bounding Rectangles Of Masks

ℹ Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step calculates the minimum circumscribed rectangles of the non-zero pixel regions of the input masks.

Usage Scenario

This Step is a general mask processing Step. This Step has no fixed usage scenarios.

Input and Output

- **Input**: Masks input to this port will have minimum circumscribed rectangles calculated.
- **Output**: Minimum circumscribed rectangles of the non-zero pixel regions of the input masks.

4.3.51. Calc Normalized Vector3Ds

Function

This Step computes the normalized unit vectors of the input Vector3Ds.

Usage Scenario

This Step is a general vector computation Step. No fixed usage scenarios.

Input and Output

- **Input**: Vectors input to this port will have their unit vectors computed.

- **Output**: computed unit vectors.

4.3.52. Calc Normals and Estimate Edges of Point Cloud

Function

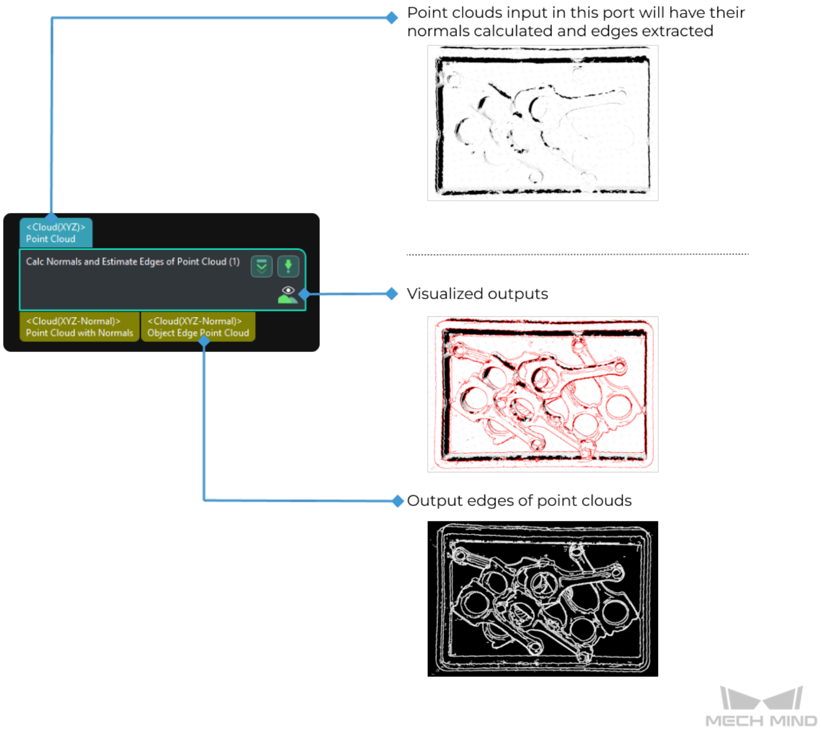
This Step is used to calculate normals of the point cloud and output the edge point cloud based on the estimated edge of the object point cloud.

The Step will convert the input point cloud to the depth map first, and then determine which points are edge points according to the normals of the point cloud and the depth information.

Usage Scenario

This Step is usually used to detect the edge features in measurement scenarios for further edge matching.

Input and Output



Point clouds input in this port will have their normals calculated and edges extracted

Visualized outputs

Output edges of point clouds

Parameter Description

Process Unorganized Point Cloud

Parameter description: This parameter is used to determine whether to process the unorganized point cloud.

Default value: Unselected (Applicable to most cases).

Tuning recommendation: If the input point cloud is an organized point cloud, you do not need to select this option; if the input point cloud is an unorganized point cloud, please select this option. Once this option is selected, you will need to set **Depth Map Resolution**, while **Search Radius** and **Half Kernel Size** no longer need to be set.

Normal Calculation Settings

Min Valid Depth for Points

Parameter description: This parameter specifies the minimum value (unit: mm) of the Z coordinate for calculating the normals. When the Z coordinate of the point is less than the parameter value, the point will be filtered out.

Default value: 100 mm

Search Radius

Parameter description: For point cloud processing, points within a circle whose radius is the value of the Search Radius parameter will be used to calculate the normals. The unit is millimeters.

Default value: 10 mm

Half Kernel Size

Parameter description: For depth map processing, points within a square with a side length of twice the value of the Half Kernel Size will be used to calculate the normals. The unit is pixels.

Default value: 2 px

Edge Computation Settings

This phase provides three ways to estimate the edge points:

- Estimate edge points based on normal variation threshold.
- Estimate edge points based on depth difference threshold.
- Estimate edge points based on the maximum angle between normal and Z axis.

In the process of determining the edge point, the Step will determine non-edge points according to the **Max Angle between Normal and Z Axis** parameter first and use the **Normal Variation Threshold** and **Depth Difference Threshold** parameters to determine the edge points in the rest points. An edge point should meet both the criteria of the **Normal Variation Threshold** and **Depth Difference Threshold** parameters.

Normal Variation Threshold

Parameter description: This parameter is used to specify the normal variation threshold used to determine the edge point, and the unit is degrees (°). A point is an edge point if the normal variation of its neighbors is above this threshold. The normal variation is a comprehensive value considering the normal variation of the adjacent points.

Default value: 10°

Depth Difference Threshold

Parameter description: This parameter is used to specify the upper limit of the depth difference (unit: mm) when determining the edge point. A point is an edge point if the maximum depth difference between the point and its eight adjacent points is greater than this threshold.

Default value: 5 mm

Max Angle between Normal and Z Axis

Parameter description: This parameter is used to specify the maximum angle (°) between the point's Z-axis and the normal. When the angle between the point's Z-axis and the normal is greater than this value, the point will not be considered an edge point.

Default value: 70° (applicable to most cases).

Edge Filtering Settings

Tolerance Gap Width

Parameter description: If there is a gap in the point cloud, this parameter is used to determine whether the ends of the gap are edges. The unit is pixels (px). If the gap width is smaller than this value and the height difference between the two ends of the gap is smaller than the **Depth Difference Threshold**, the two ends of the gap will not be considered edges.

Default value: 2 px

Min Edge Point Count

Parameter description: This parameter is used to specify the minimum number of points in the output edge point cloud. Edge point cloud part with fewer points than the set value will be ignored.

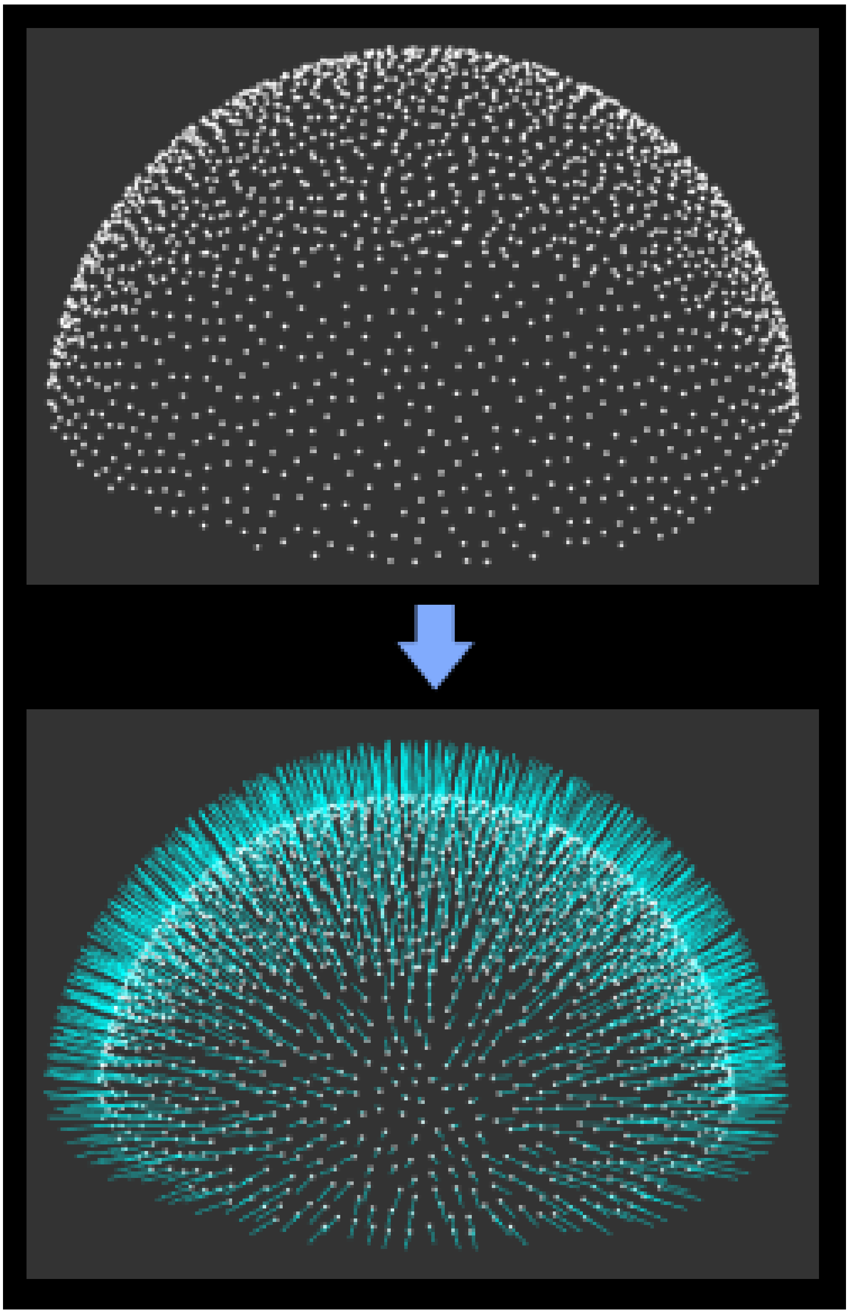
Default value: 10

4.3.53. Calc Normals of Point Cloud and Filter It

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

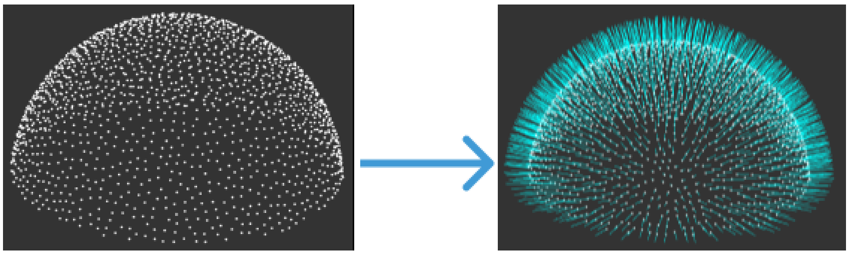
Function

This step is used to calculate point cloud normals and remove outliers.

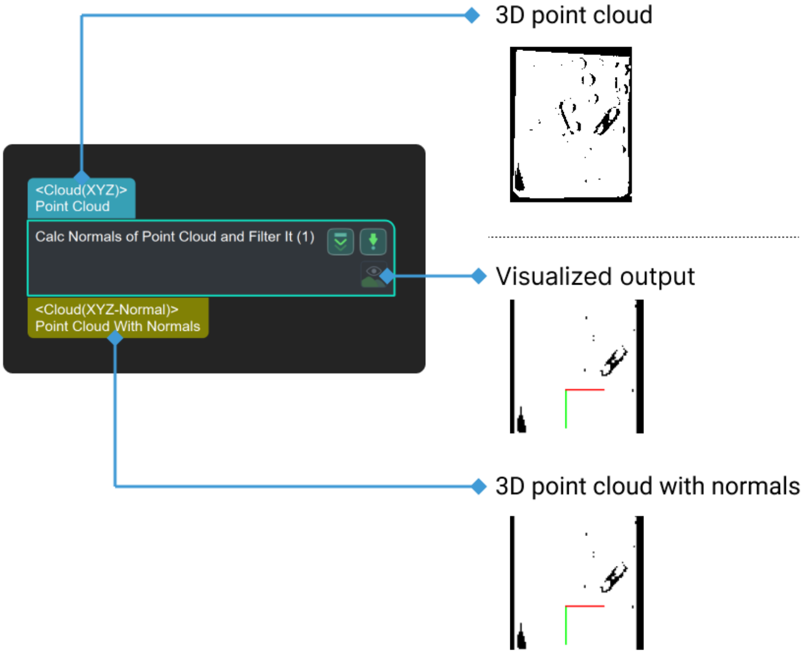


Usage Scenario

This Step is used to add normals to the original point cloud and remove outliers. It usually follows Steps Capture Images from Camera and From Depth Map to Point Cloud.



Input and Output



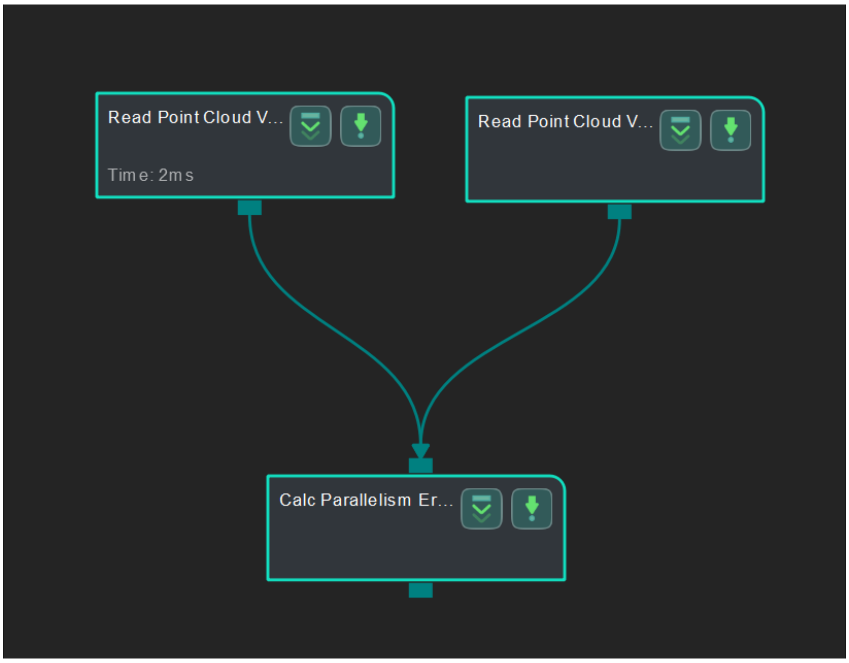
4.3.54. Calc Parallelism Error

This Step is used to calculate the parallelism error from the target planar point cloud to the reference planar point cloud.

- Parallelism is defined as the quality or condition of two planes/edges being parallel, which is measured by the maximum allowable error between one plane/edge and the other. It evaluates whether the lines, planes, or the line and the plane are parallel.
- Parallelism error is the difference between the measured plane/edge and the reference plane/edge.

Before using this Step, please refer to Getting Started with Measurement Mode to learn about basics of Measurement Mode.

Application Examples



Connect Steps

Click Calc Parallelism Error, go to the **Step Input Source Selection** panel, and then

- select Read Point Cloud V2_1_Point Cloud With Normals as Input 1 (Target Planar Point Cloud) and
- select Read Point Cloud V2_2_Point Cloud With Normals as Input 2 (Reference Planar Point Cloud).

Configure parameters

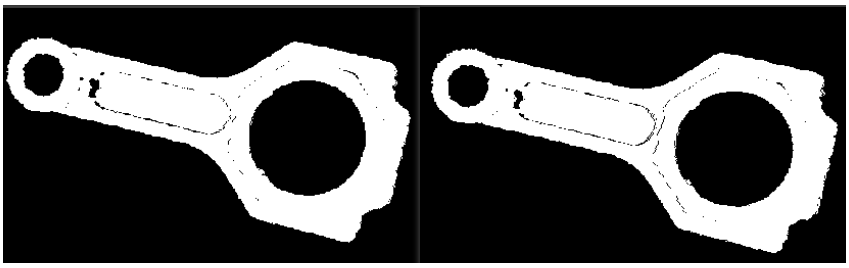
- Configure in Read Point Cloud V2 (1):

Click the Step and configure the Point Cloud Source (target cloud) and other parameters in the **Step Parameters** panel.

- Configure in Read Point Cloud V2 (2):

Click the Step and configure the Point Cloud Source (reference point cloud) and other parameters in the **Step Parameters** panel.

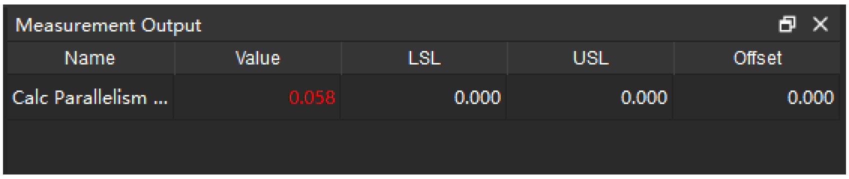
The point clouds used in this sample application are shown below (target cloud on the left and reference point cloud on the right).



Click **[Run]** to run the project after the settings.

Read the result

After running the project, click **Calc Flatness Error** and check the parallelism error in the **Measurement Output** panel.



Measurement Output

Name	Value	LSL	USL	Offset
Calc Parallelism ...	0.058	0.000	0.000	0.000

You can set the LSL (lower specification limit) and USL (upper specification limit) in the **Measurement Output** and **Result View** panels to check whether the measurement conforms to the specification.

4.3.55. Calc Pixel Sizes at Specified Heights

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step calculates the physical sizes for pixels given the heights of the corresponding 3D points.

Usage Scenario

This Step is a custom Step for special scenarios. This Step is used for depalletizing applications.

Input and Output

- **Input**: Heights of 3D points input to this port will be used to calculate the physical size for the corresponding pixels.
- **Output**: physical sizes of pixels calculated.

4.3.56. Calc Plane Width

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step clusters the depth values along the scanline direction, calculates the scanline lengths that belong to the same set, and takes the median of the lengths as the widths of planes.

Usage Scenario

This Step is a custom Step for special scenarios.

Input and Output

- **Input**:
  1. Depth map input to this port contains the planes whose widths need to be calculated.
  2. Orthographic projection scale ratio.

- Output:
  1. Plane widths calculated

4.3.57. Calc Point Cloud Curvatures

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step calculates the curvatures of all point clouds' points in the input list of point clouds.

Usage Scenario

This Step is a general point cloud calculation Step. This Step has no fixed usage scenarios.

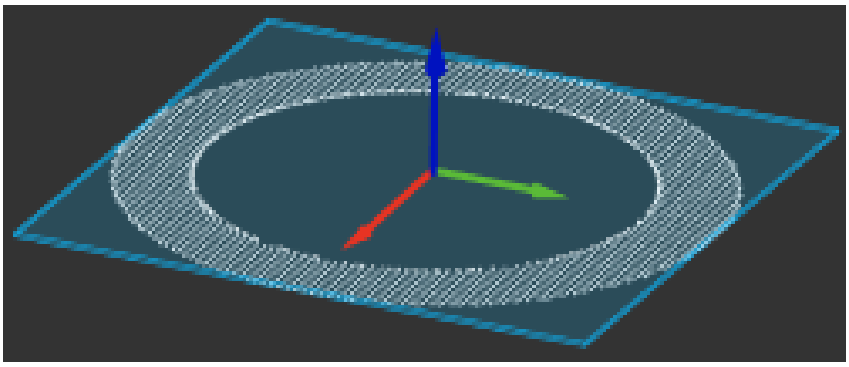
Input and Output

- Input:
  1. Point clouds input to this port will have point curvatures calculated.
- Output:
  1. Calculated curvature values
  2. Point clouds with reassigned point curvature values

4.3.58. Calc Poses and Dimensions from Planar Point Clouds

Function

This Step calculates the poses of planar point clouds and the dimensions of the point cloud bounding boxes.

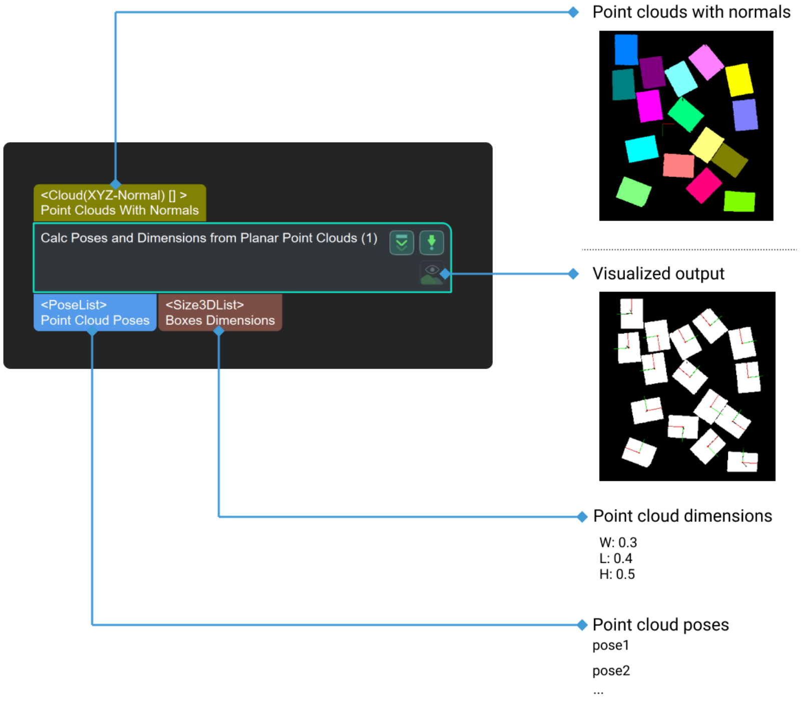


Usage Scenario

This Step is usually used to calculate pick points and dimensions of planar point clouds of objects such as cartons and sacks.

Before using this Step, please filter and cluster the point cloud to reduce the interference from unwanted points.

Input and Output



Parameter Description

Pose Calculation Settings

The selected **Method to Calc Pose Center** should correspond with the **Method to Calc Pose Orientation**.

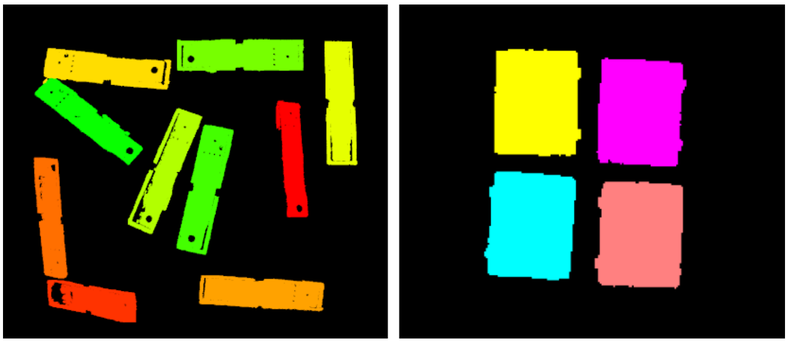
- When BoundingRectCenter is selected as the Method to Calc Pose Center, BoundingRectOrientation should be selected as the Method to Calc Pose Orientation.
- When CloudCenter is selected as the Method to Calc Pose Center, CloudOrientation should be selected as the Method to Calc Pose Orientation.

Method to Calc Pose Center

Description: This parameter specifies the center where the pose position is located.

Value list:

- CloudCenter (Center of the point cloud): Select this method when the object point clouds scatter randomly, as shown in the left figure below. BoundingRectCenter (Center of the point cloud's bounding box): Select this method when the object point clouds are evenly spread, as shown in the right figure below.
- CloudCenter (Center of the point cloud): Select this method when the object point clouds scatter randomly, as shown in the left figure below.



Tuning recommendation: Please set the parameter according to the actual requirement.

Method to Calc Pose Orientation

Description: This parameter specifies the orientation of the pose.

Value list:

- BoundingRectOrientation (Orientation of the bounding box center): Select this method when the object point clouds are evenly spread, as shown in the right figure above.
- CloudOrientation (Orientation of the point cloud's center): Select this method when the object point clouds scatter randomly, as shown in the left figure above.

Tuning recommendation: Please set the parameter according to the actual requirement.

Use Best Aligned Rect

Description: This parameter determines whether to use a bounding rectangle whose edges are as parallel as possible to the edges of the object. If this option is NOT selected, a minimum bounding rectangle will be used.

Default Value: Unselected.

Tuning recommendation: Please set the parameter according to the actual requirement.

Align Search Radius

Description: This parameter determines whether the bounding rectangle aligns with the feature points of the object. The unit is pixels. When the distance between the feature point and the bounding rectangle is less than the value, the feature points are considered aligned with the bounding rectangle.

Default value: 3 px

> This parameter will appear only after you select **Use Best Aligned Rect**.

Box Pose Calculation Settings

> This parameter group is applicable to cartons only.

Calc Boxes' Height

Description: This parameter is used to calculate the carton's height in the Z direction.

Default value: Unselected.

Tuning recommendation: Please select this parameter when you need to calculate the carton's height in the Z direction.

Fixed Height for Box Bottom

Description: This parameter specifies the height of the carton base, such as conveyor belt and pallet on which the carton is placed, for facilitating the calculation of the carton's height in the Z direction.

Default value: 0.000 mm

Tuning recommendation: Please set the parameter according to the actual requirement.

This parameter will appear only after you select **Calc Boxes' Height**.

4.3.59. Calc Profile by Sampling

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step downsamples the object surface point cloud and model point cloud, and then calculates the difference between the two to eliminate the influence on calculation caused by the inconsistent density of the object point cloud and the model point cloud.

Usage Scenario

This Step is a custom Step for special scenarios.

Input and Output

- Input:
  1. Point cloud of the object whose profile needs to be calculated
  2. Model point cloud for calculating profile
- Output:
  1. Maximum height difference between the object and the model point cloud
  2. Height difference between the object and the model point cloud at the center point
  3. Line segment used to label the location of the maximum height difference
  4. Line segment used to label the height difference at the center point
  5. Model point cloud within the region of interest
  6. Object point cloud within the region of interest
  7. Height differences between the object point cloud and the model point cloud at specified points

4.3.60. Calc Profile

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step calculates the height difference (i.e., profile) between the object surface point cloud and the model. Usually, two such Steps are used at the same time: One takes the point cloud of the object whose profile needs to be evaluated as the input, and the other takes the standard model point cloud. The two Steps respectively calculate the differences with the fitted plane (i.e., the flatness). Then, the difference between the output flatness needs to be calculated to obtain the profile. Usually, the flatness or profile is the maximum value of the height difference somewhere on the point clouds.

Usage Scenario

This Step is a custom Step for special scenarios.

Input and Output

- Input:
  1. Point cloud input to this port will be used to calculate the profile. It is the point cloud of the object to be measured or the standard model.
  2. Plane widths calculated.
- Output:
  1. Height differences between specified points in the region of interest of the input point cloud and the fitted plane
  2. Object or model point cloud in the region of interest
  3. Object or model point cloud after specifying 2D coordinates (for subsequent fitting of different point clouds to facilitate calculation)
  4. Object or model point cloud and fitted plane's point cloud after specifying 2D coordinates
  5. Cross section point cloud

4.3.61. Calc Projection Length along Reference Direction

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step connects the input poses' centers to the origin of the reference frame, generates vectors and calculates the lengths of the projections of the vectors in the reference direction.

Usage Scenario

This Step is a general pose projection distance calculation Step. This Step has no fixed usage scenarios.

Input and Output

- **Input**: Poses input at this port will have their projection lengths along the specified direction calculated.
- **Output**: length of the projections of the poses along the specified direction.

4.3.62. Calc 2D Poses of Rectangles

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step gets the poses of the rectangle center points from the rectangle vertex information.

Usage Scenario

This Step is a custom Step for special scenarios. This Step is used for depalletizing applications.

Input and Output

- Input:
  1. Rectangle vertices
- Output:
  1. Calculated rectangle center poses
  2. Calculated rectangle sizes

4.3.63. Calc Results by Python

Function

This Step runs a user-defined script through Python and outputs the calculation results to Mech-Vision.

The characteristics of this Step are as follows:

- Support multi-threading.
- Load Python script in real time.
- Support various data type conversions when transferring data between C++ and Python.
- Support to display the Python’s log in the log panel of Mech-Vision.

Usage Scenario

When a custom calculation is required, this Step can be used to run the user-defined Python script and the workflow of the Steps can be simplified in the project.

Input and Output

- **Input**: determined by the data type input in the **Input Port(s)** parameter.
- **Output**: determined by the data type input in the **Output Port(s)** parameter.

> You can specify the data type of the input/output port according to the data type of the prior or succeeding Steps’ input/output port.

Installation and Usage

Installation

Python 3.6.8 is built into Mech-Vision, and thus this Step will use the built-in Python environment of the software. If there is any missing Python library when you use this Step, you need to install the library in the Mech-Vision’s built-in Python environment. The procedures for installation are as follows:

1. Open the command prompt window.
2. Use the CD command to go to the directory where Python is stored in Mech-Vision.
3. Type “python -m pip install Python name of the library” on the command line to install the missing Python library.

> The two frequently-used Python libraries, NumPy and OpenCV, are built into Mech-Vision.

Usage Instructions

After preparing the Python script, please follow the procedure below to use the Step. For detailed description of the parameters, please refer to Parameter Description.

1. Set the data type for the input and output ports. Enter the data type for **Input Port(s)** and **Output Port(s)** according to the data type of the prior or succeeding Steps’ input/output port.
2. Specify the file path of the Python script. Select the file path of the script to be loaded in the **Script File Path** parameter.
3. Set the name of the function to be called. Once the Script File Path has been specified, this Step will retrieve the function names in the script automatically, and you can select the name of the function to be called in the drop-down list of **Func Name**.
4. Run the Step.

> - When writing the Python scripts, you can directly write the functions to process data without writing the statements in the scripts to obtain Mech-Vision data. For the specific form of the scripts, please refer to Python Example Programs.
> - When the Python script is modified, and you want to use the new script to run this Step in real time, click **Reload File** in the **Execution Flags** parameter group.

> From Mech-Vision 1.8.0, the execution mode of the “Calc Results by Python” Step is changed from serial to parallel (consistent with the effect of the threading module in Python). When calling camera APIs that do not support parallel execution within the script (such as enumerating cameras), it is necessary to add a lock at the location of that API call.

Notes

Please pay attention to the following issues when you write the Python script and run the script in this Step.

Third-Party Library is Recommended

Because running a Python script in Mech-Vision differs from running the script in Python directly, some Python libraries could not be installed successfully or they cannot function properly after being installed, it is recommended to use third-party libraries.

Use the NumPy Library

The NumPy library is used to support some complicated formats of data. If a parameter type is ndarray, but the NumPy has not been imported, an error may occur. Therefore, you should add the **import numpy** statement at the beginning of the script.

Pay Attention to the Data Dimension when Writing the Script

When writing the Python script, please note the dimension of data corresponding to each port of the Step.

- 0 dimension by default: Image; Cloud(XYZ); Cloud(XYZ-Normal)
- 1 dimension by default: NumberList; BoolList; IndexList; StringList
- 2 dimensions by default: PoseList; Pose2DList; Size3DList

Use “[]” after the data type to add a dimension.

For example, the data dimension of NumberList is 1, while the data dimension of NumberList[] is 2.

Parameter Description

Input Port(s)

Description: This parameter is used to specify the data type(s) of the input port(s). The input data types will be passed to the called function as parameters in the corresponding order.

Default value: null.

Output Port(s)

Description: This parameter is used to specify the data type(s) of the output port(s). Data returned by the function will be returned to the Step in the corresponding order and will be parsed according to the corresponding data type.

Default value: null.

The currently supported data types are as follows:

Port Type	Example
PoseList	[[10, 20, 30, 0.951, 0.255, 0.168, 0.045]], [10, 20, 30, 0.951, 0.255, 0.168, 0.045]] (In each array, the first three values are the coordinates, and the following four values are quaternions.)
Pose2DList	[[0, 0, 0]], [2, 0, 120]] (In each array, the first two values are X and Y values of the coordinates respectively, and the third value is the angle.)
NumberList	[1.1, 2, 999.9, -22]
StringList	['string_1', 'string_2', 'string_3']
Image	Image data
Cloud(XYZ)	Point cloud data
Cloud(XYZ-Normal)	Point cloud data with normals
Cloud(XYZ-RGB)	Data of textured point cloud
Size3DList	[[2.5, 5, 0.001]], [6, 5, 0.02]] (In each array, the first two values are the width and height, and the third value is the length of each pixel.)
IndexList	[45, 10, 90]
BoolList	[True, False, True]

Script File Path

Description: This parameter is used to select the file path of the script to be loaded.

Default value: Null.

Func Name

Description: This parameter is used to set the name of the function to be called.

Default value: null.

Python Example Programs

To facilitates the learning and utilizing of this Step, some Python programs are provided in the following section to help you get familiar with the Step port type.

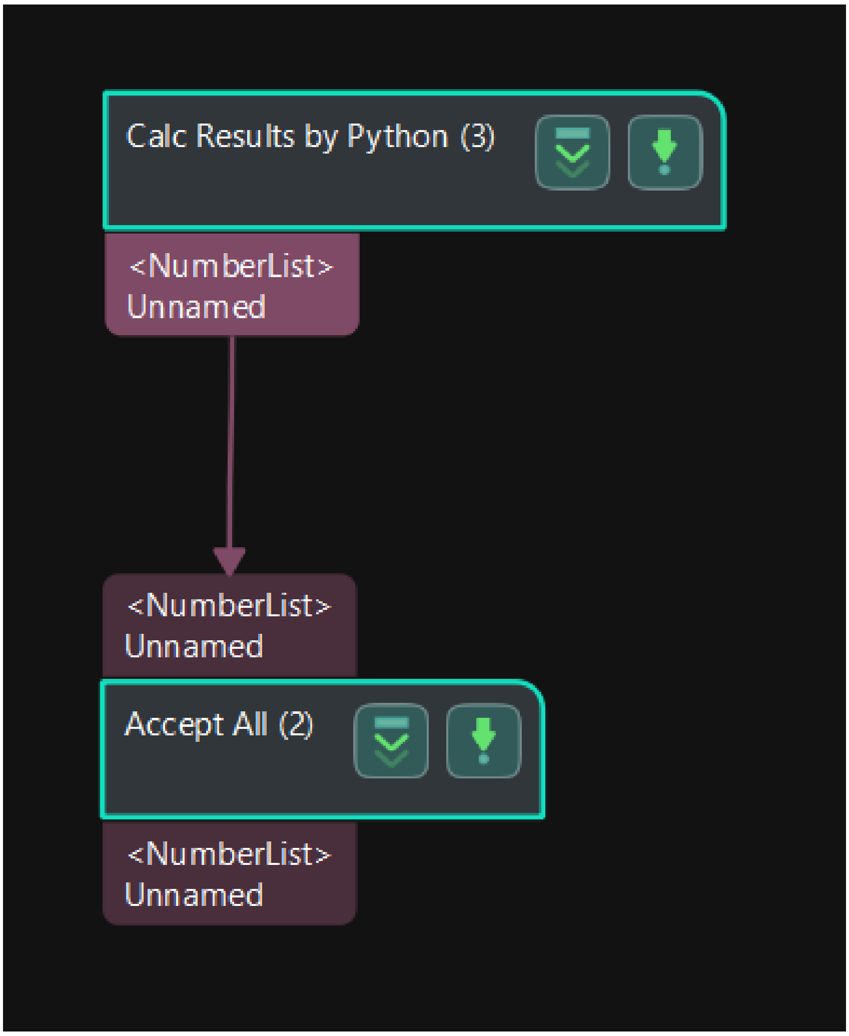
Getting Started

Output Data of One Type with Python Script

The function that **outputs data of one type** is defined in the program below. Please refer to Usage Instructions to call this function in the Step.

```python
def get_doublelist():
    return [1.1,22,3.3]
```

The Mech-Vision project corresponding to this function is shown below. You can double-click the dataflow corresponding to the output port to check the output result.

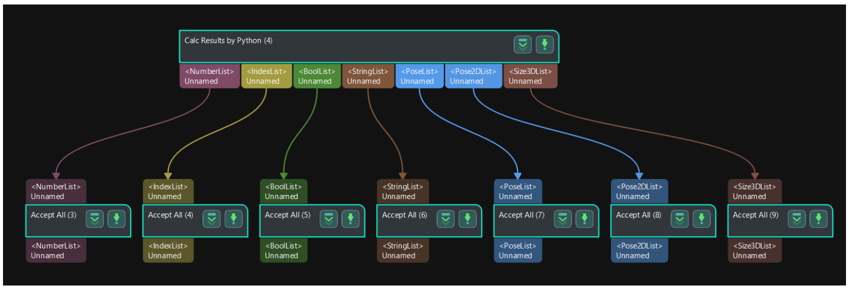


Output Data of Multiple Types with Python Script

The function that **outputs data of multiple types** s defined in the program below. Please refer to Usage Instructions to call this function in the Step.

```
def example_of_basic_portTypes():
    numList = [1.1, 2, 999.9, -22]
    indexList = [45, 10, 90]
    boolList = [True, False, True]
    strList = ['string_1', 'string_2', 'string_3']
    poseList = [[10, 20, 30, 0.951, 0.255, 0.168, 0.045], [10, 20, 30, 0.951,
0.255, 0.168, 0.045]]
    pose2dList = [[0, 0, 0], [2, 0, 0.785]]
    size3dList = [[2.5, 5, 0.001], [6, 5, 0.02]]
    return numList, indexList, boolList, strList, poseList, pose2dList,
size3dList
```

The Mech-Vision project corresponding to this function is shown below. You can double-click the dataflow corresponding to the output ports to check the output results.



Process Data with Two “Calc Results by Python” Steps

The function to output the list of numerical values is defined in Program 1. The function to calculate the sum of values within each numerical list is defined in Program 2.

Please refer to Usage Instructions to use the following programs, connect Steps, and calculate the sum of the input numerical lists.

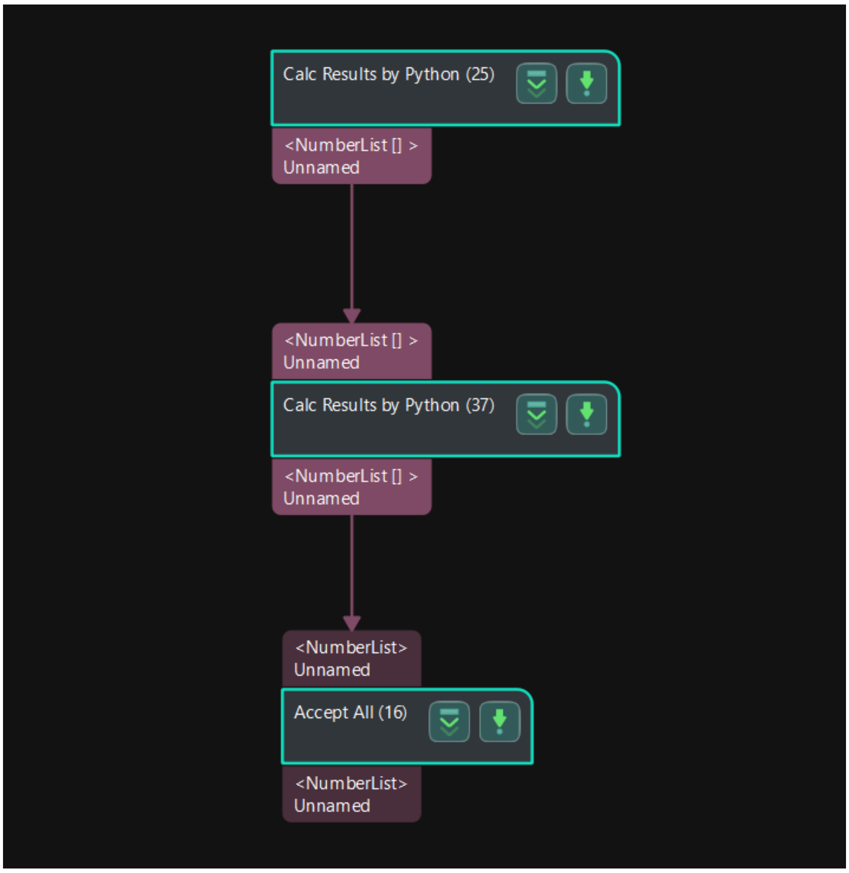
Program 1:

```
def get_numberlist_1():
    return [[1, 2, 3],[4,5,6]]
```

Program 2:

```
def cal_number_list(nums_list):
    sums = [align="center"]
    for nums in nums_list:
        sum = 0
        for num in nums:
            sum += num
        sums.append(sum)
    return [sums]
```

The Mech-Vision project corresponding to this function is shown below. You can double-click the dataflow corresponding to the output ports to check the output results.



Process Data with Multiple “Calc Results by Python” Steps

The function to output the list of numerical values is defined in Program 1. The function to output lists of Booleans is defined in Program 2. In program 3, the value “9999” is added to the numerical value, and a “False” is added to the list of Booleans, and the new list of numerical values and list of Booleans will be output.

Please refer to Usage Instructions to use the following programs in the Step to process the input data.

Program 1:

```
def get_doublelist():
    return [1.1,22,3.3]
```

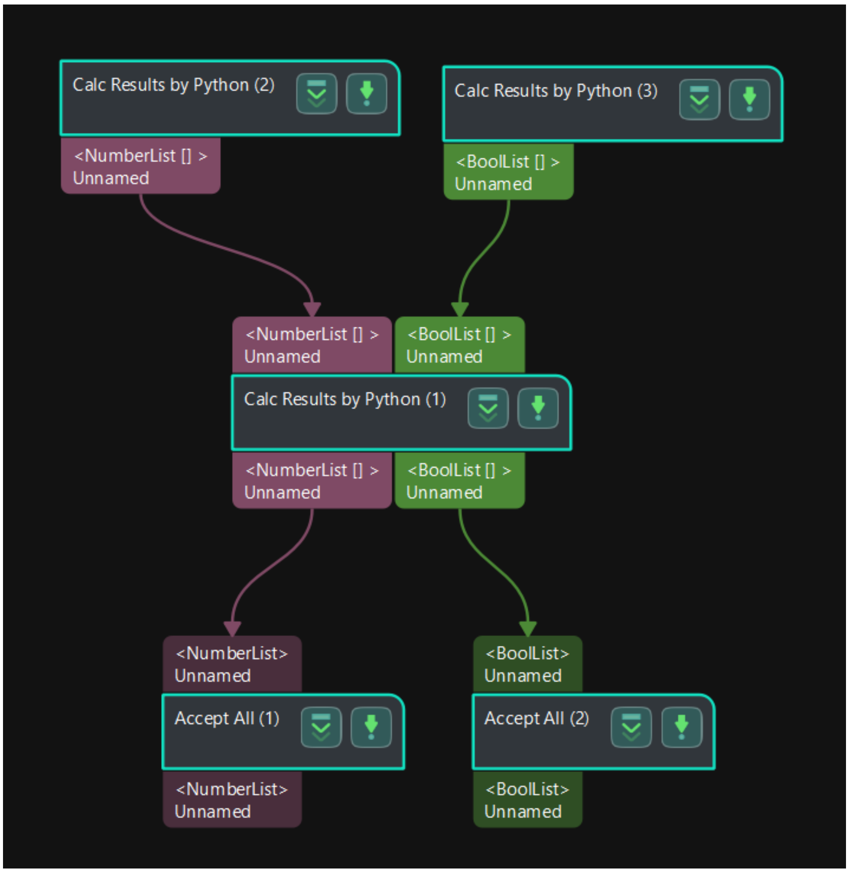
Program 2:

```
def get_bool_list():
    return [[True,False],[True,True],[False,False]]
```

Program 3:

```
def print_multi_values(numberList, boolList):
    numberList.append(9999)
    boolList.append(False)
    return numberList, boolList
```

The Mech-Vision project corresponding to this function is shown below. You can double-click the dataflow corresponding to the output ports to check the output results.



Application Examples

Saving Abnormal Data Generated During Project Execution

In actual applications, various errors may occur in the project. You can use the custom Python script together with the Data Storage feature in Mech-Vision to save the abnormal data for troubleshooting, thus improving the stability of the vision system.

The detailed instructions are as follows.

1. Enable the **Data Storage** feature.

   Enable **Save data and parameters** in Mech-Vision Project Assistant. If errors occur when the project executes, the raw data will be save under **Project folder/data/error_data**.

2. Write the Python script.

   A Python example program that determines whether the pose result output by the “3D Matching” Procedure is null is as follows:

```
def abnormal_output_detection(outputs):
    if len(outputs) == 0:
        raise Exception("NO RESULT!!!")
    else:
        pass
```

3. Please refer to Usage Instructions to use the above program in the Step, and connect the Step after “3D Matching”.

   After running the project, if the pose result output by “3D Matching” is null, an error message will pop up, and the abnormal data will be saved under **Project folder/data/error_data**.

Generate Pseudo Colored Point Cloud Based on Z Values of the Input Point Cloud

When the output point cloud does not have the textured information, you can use the custom Python script to generate pseudo colored point cloud based on the Z values of the input point cloud for better visualization of the output.

The processing process can be summarized as follows:
Firstly, the Z values will be mapped to values between 0 to 255 to obtain the grayscale values. Then OpenCV will be used to map the grayscale values to RGB values and therefore the pseudo colored point cloud can be generated.

The example program is as follows:

```
import math

import cv2
import numpy as np

def generate_color_point_could_by_depth(point_cloud):
    x = point_cloud[:, 0]
    y = point_cloud[:, 1]
    z = point_cloud[:, 2]

    z_min = np.min(z)
    z_max = np.max(z)

    if math.isclose(z_max - z_min, 0):
        return np.column_stack((x, y, z, np.full(z.shape, 0xFFFFFFFF, np.
uint32).view(np.float32)))

    color = 255 * (z - z_min) / (z_max - z_min)
    color = cv2.applyColorMap(color.astype(np.uint8), cv2.COLORMAP_JET)

    color = np.squeeze(color).astype(np.uint32)
    color = (0xFF000000 | (color[:, 0] << 16) | (color[:, 1] << 8) | color[:,
2]).view(np.float32)
```

```
    return np.column_stack((x, y, z, color))
```

4.3.64. Calc Specified Property of Point Clouds

> Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step outputs the value of a specified property of a representative point set in the parameters.

Usage Scenario

This Step is a general point cloud property calculation Step. This Step has no fixed usage scenarios.

Input and Output

- **Input**: Properties of the point clouds input to this port will be calculated.
- **Output**: values of the specified property of the point clouds, which will be used for point cloud sorting.

4.3.65. Calc Calibration Board Pose and Check Camera Intrinsic Parameters

Function

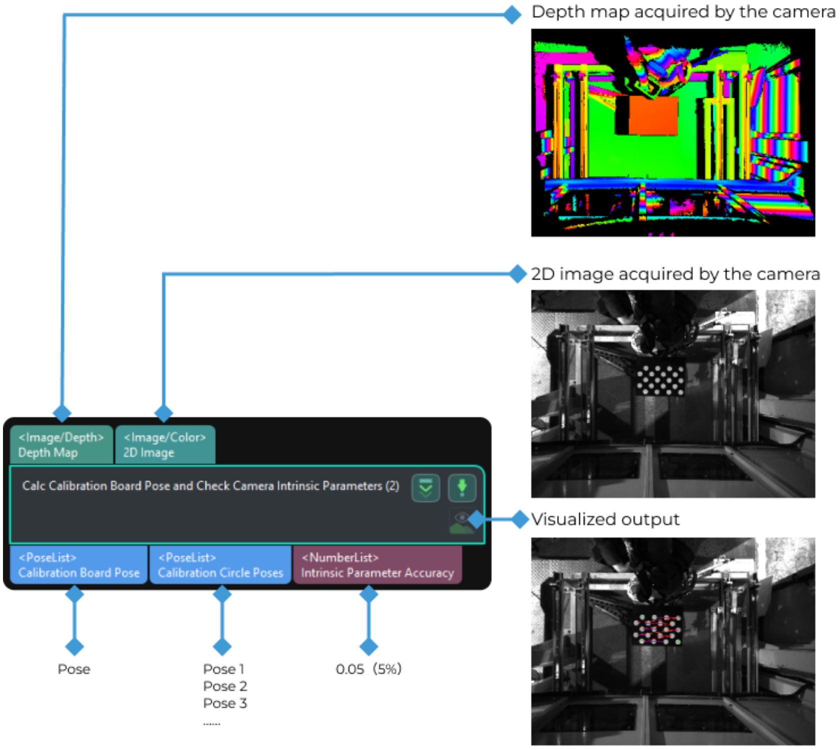
This Step can detect the calibration circle centers on the 2D image, and then compute the pose of the calibration board and the poses of calibration circles according to the camera intrinsic parameters and the input depth map. In addition, this Step can be used to estimate the accuracy of the camera's intrinsic parameters based on the calibration circle poses.

> The calibration circle pose indicates the pose of the center of the calibration circle on the calibration board.

Usage Scenario

This Step is usually used for obtaining the pose of the calibration board and check the camera's intrinsic parameters.

Input and Output



Parameter Description

Calibration Board Model

Description: This parameter is used to set the model of the calibration board.

Tuning recommendation: Please set this parameter according to the actual calibration board models in use.

2D ROI

Description: This parameter is used to set the 2D ROI.

Tuning recommendation: Click **[Open the editor]** to set the 2D ROI.

Enable Hand-Eye System Compensation

Description: If the hand-eye (robot and camera) system encounters an accuracy problem during the calibration process, you can enable the hand-eye system compensation to increase the system's accuracy. Once this option is selected, the compensation parameters will be used to reduce the error.

Default value: selected.

4.3.66. Calc Calibration Matrix of Structured Light Sensor

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step calibrates the scanning direction (relative to the laser plane) of the line laser.

Usage Scenario

This Step calibrates the relationship between the scanning direction of the laser line scan camera and the laser plane, to correct the deformation of the point cloud.

Input and Output

- **Input:** Point cloud of the calibration block.
- **Output:** Calculated calibration matrix.

4.3.67. Calc Center Points of Non-Zero Regions

> Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step computes the center points of regions of non-zero pixel values in a 2D image.

Usage Scenario

This Step is a general 2D pose calculation Step. There is no fixed usage scenario.

Input and Output

- **Input**: Centers of masks input to this port will be calculated.
- **Output**: Calculated mask center points.

4.3.68. Calc Cylinder Poses

> Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step calculates poses of cylinder center points.

Usage Scenario

If the recognized cylinder center poses are not on the axial centers, this Step can be used to correct the poses. This Step usually follows the Step Point Cloud Shape Detector.

Input and Output

- Input
  1. Cylinder point clouds input to this port will have poses of center points calculated.
  2. Cylinder shape info, usually from the output of the Step “Point Cloud Shape Detector.”
- Output
  1. Calculated center poses of the point clouds.

4.3.69. Caliper Tool

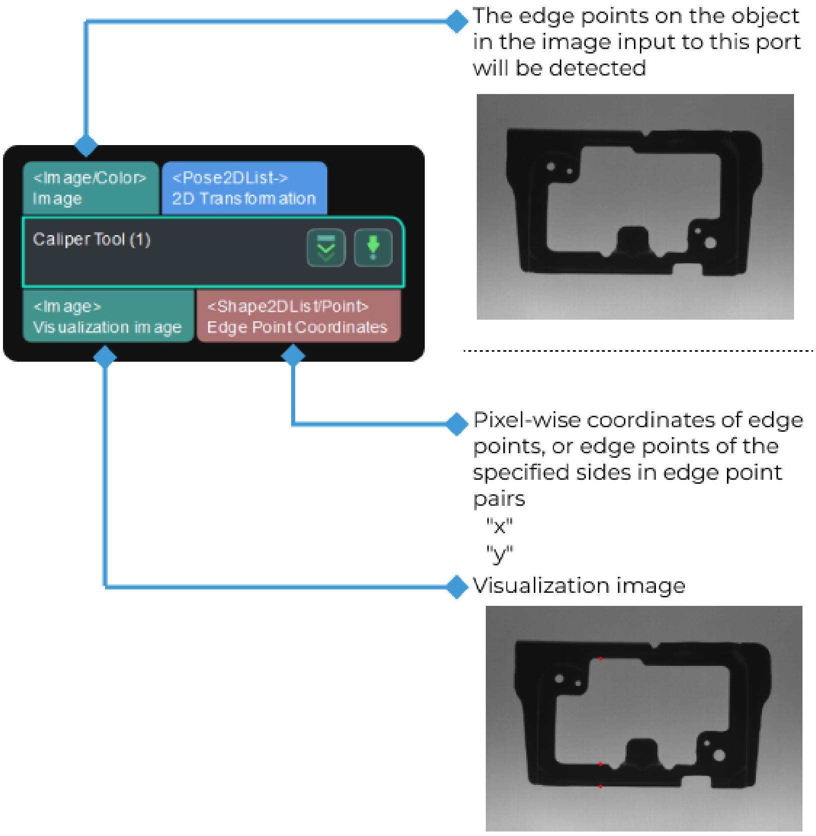
Function

This Step is used to detect edge points or edge point pairs along the vertical direction of an ROI that is usually elongated and output the pixel-wise coordinates of edge points and distances of edge point pairs (if detecting edge point pairs).

Usage Scenario

This Step is used to measure the widths of specified object parts or locate edge points along a line over an object. This Step is usually followed by the Convert Lengths Pixel-Wise to Physical Step to get the physical width values.

Input and Output


The edge points on the object in the image input to this port will be detected
<Image/Color>
Image
<Pose2DList->
2D Transformation
Caliper Tool (1)
<Image>
Visualization image
<Shape2DList/Point>
Edge Point Coordinates
Pixel-wise coordinates of edge points, or edge points of the specified sides in edge point pairs
"x"
"y"
Visualization image

Parameter Description

ROI

Description: This parameter specifies an ROI which includes the edge points or edge point pairs, and an elongated frame for edge detection will be generated.

Tuning recommendation: You should select an ROI in the **measurement mode**, please refer to Measure Circles for detailed instructions. When the image quality is good, you can set the **Width** of the ROI to 1 and adjust the **Height** to cover the area to be detected.

Position and Orientation Correction

Description: This parameter is used to adjust the ROI to a proper position.

Notes: Once this parameter is selected, the Step will adjust the ROI set before to a proper position according to the input “2D poses.”

Grayscale Change Lower Threshold

Description: This Step is used to specify the lower threshold of the grayscale change of the point in the detection area. If the point’s grayscale change is above this threshold, the point will be considered as an edge point.

Default value: 20

Sigma of Gaussian Filter

Description: Gaussian filter is used to remove noise from the image. This parameter is used to set the coefficient of the filter along the vertical direction in the edge detection area.

Default value: 1.0

Recommended value: 1.0

Tuning recommendation: If the image quality is good, you can keep the default value 1. If the image quality is poor, you can use a larger value.

Max Number of Results

Description: This parameter specifies the maximum number of edge points or edge point pairs output by this Step.

As for edge points, the edge point with greater grayscale change will be output with priority.

As for edge point pairs, the edge pair in which the distance between two edge points that is closer to the **Edge Pair Expected Distance** will be output with priority.

Default value: 1

Tuning recommendation: : Please adjust the value according to the actual requirement.

Output Order

Description: This parameter specifies in which way the detection results will be sorted and output.

Value list: Top to Bottom, Bottom to Top, and Center to Ends.

Top to Bottom: Edge points that are closer to the top of the ROI will be ranked high on the output list.

Bottom to Top: Edge points that are closer to the bottom of the ROI will be ranked high on the output list.

Center to Ends: Edge points that are closer to the center of the ROI will be ranked high on the output list.

Default value: Top to Bottom

Edge Mode

Description: This parameter specifies the mode of the edge to be detected.

Value list: Single Edge and Edge Pair.

Single Edge: Detect a single edge only and output a list of individual edge points.

Edge Pair: Detect edges on both sides, and output two lists of edge points respectively, which represent edge point pairs.

Default value: Single Edge.

Notes: Please refer to tuning examples for the visualized output.

> When the edge mode is **Edge Pair**, you will need to set all the **Edge Transition Type 0**, **Edge Transition Type 1**, and **Edge Pair Expected Distance** parameters.

Edge Transition Type 0

Description: Edge transition type is used to specify which kind of edge with the grayscale change will be considered as the object edge. This parameter is used to detect single edge points or edge points on one side in the edge point pairs.

Value list: White To Black, Black To White, and Both.

White To Black: The position in the ROI where the grayscale changes from white to black will be considered as the edge point of the object.

Black To White: The position in the ROI where the grayscale changes from black to white will be considered as the edge point of the object.

Both: The position in the ROI where the grayscale changes from black to white or from white to black will be considered as the edge point of the object.

> The grayscale change refers to the change in grayscale from top to bottom in the ROI.

Default value: White To Black

Notes: Please refer to tuning examples for the visualized output.

Edge Transition Type 1

Description: Edge transition type is used to specify which kind of edge with the grayscale change will be considered as the object edge. This parameter is used to detect the edge points on the other side in the edge point pairs. You need to set this parameter when the edge mode is **Edge Pair**.

Value list: This parameter is the same as that of **Edge Transition Type 0**.

Edge Pair Expected Distance

Description: This parameter is used to specify the expected distance in pixels between two edge points of an edge pair.

Default value: 10

Notes: The edge pair in which the distance between two edge points that is closer to this value will be output with priority.

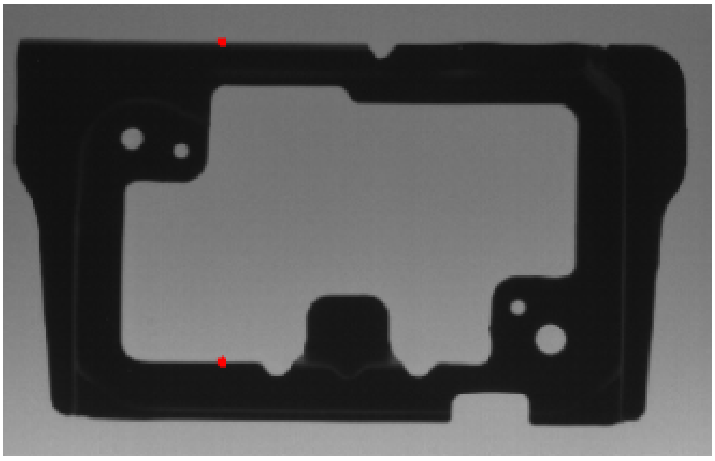
> Once a **Max Number of Results** is set, the edge point pairs with lower priority may be ignored.

Tuning Examples

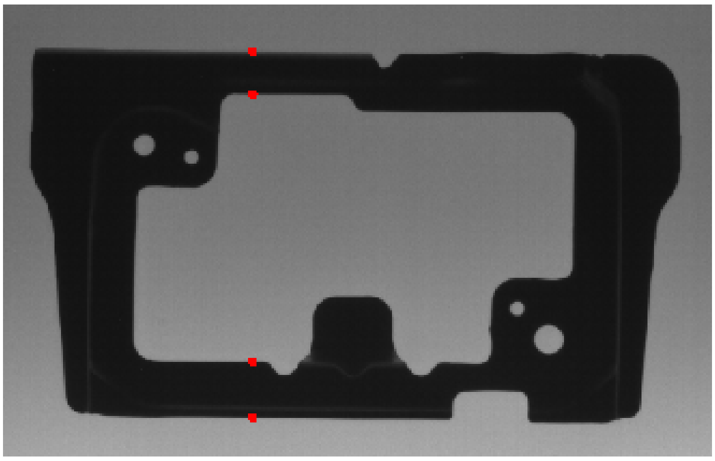
Edge Mode

When the **Edge Mode** is set to **Single Edge** and **Edge Pair**, respectively, the output results are shown in the figure 1 and 2 below.

1



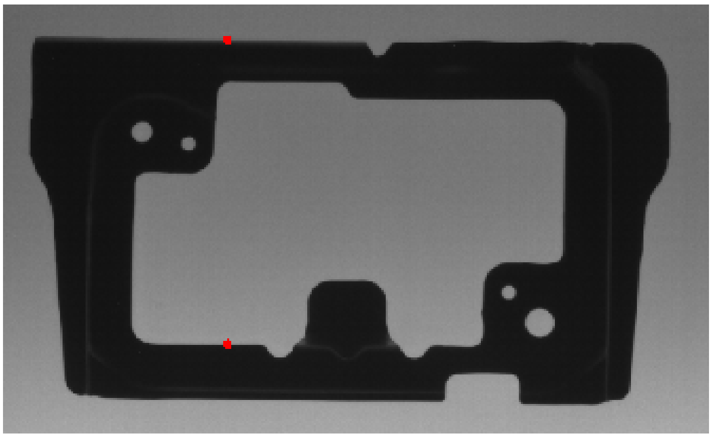
2



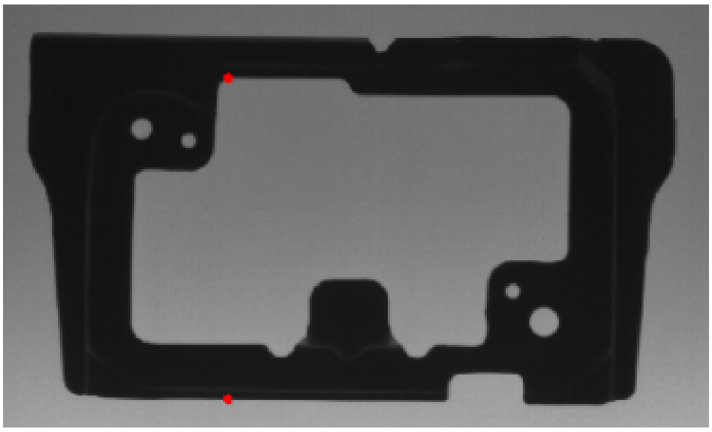
Edge Transition Type

When the **Edge Transition Type** is set to **White To Black**, **Black To White**, and **Both**, respectively, the output results are shown in the figure 1, 2, and 3 below.

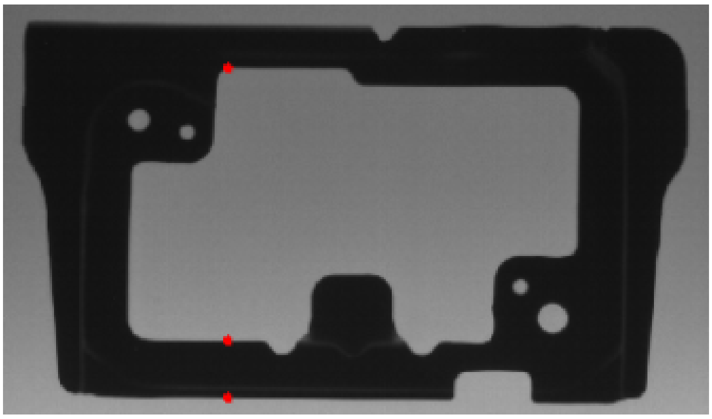
1



2



3



4.3.70. Camera 2D

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step can be used to capture images with a third-party 2D camera. The supported third-party 2D cameras are listed in the table below.

Item	Camera type	Related step	Notes	Recommended client version
Hikvision	2D camera	Camera 2D	Relative driver and the camera2d_h plugin should be installed.	MVS 3.3.1
DAHENG	2D camera	Camera 2D	The camera2d_plugin plugin should be installed.	2.7.2305.8081
iRayple	2D camera	Camera 2D	The camera2d_plugin plugin should be installed.	2.3.0_Build20210730
Cameras supporting the GenICam standard	2D camera	Camera 2D	The camera2d_plugin plugin should be installed.	eBus Player 6.2.8

Usage Scenario

This Step is generally used with the 2D camera.

Input and Output

- **Input:** none.
- **Output:**
  1. Color image captured by the camera
  2. Timestamp string corresponding to the color image acquired by the camera, which can be used as the naming prefix to save the image

4.3.71. Capture Images from Camera

Function

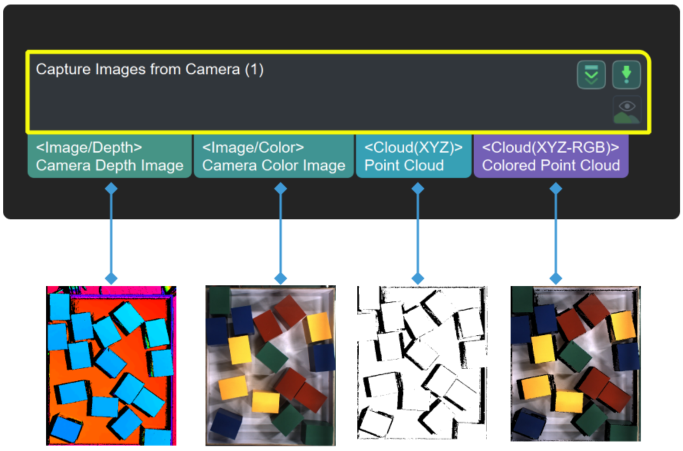
This Step is used to obtain color images, depth maps, and point clouds from a real or virtual camera.

> When this Step is used to connect a DEEP (V4) or LSR (V4) series camera, and the 2D Image Type is set to Internal IR Image, the output “color image” is actually a 2D image (Depth Source). If you want to adjust the 2D image (Depth Source), adjust the **2D Image** (Depth Source) Exposure Time of the **2D Image** (Depth Source) Exposure Mode in Mech-Eye Viewer.

Usage Scenario

Used as the initial input entry of a Mech-Vision project, this Step collects data from a Mech-Eye Industrial 3D Camera or third-party camera or acts as a virtual camera for inputting simulated data source.

Input and Output


Capture Images from Camera (1)
<Image/Depth> Camera Depth Image
<Image/Color> Camera Color Image
<Cloud(XYZ)> Point Cloud
<Cloud(XYZ-RGB)> Colored Point Cloud

Parameter Tuning

Camera Settings

Camera Type

Description: This parameter is used to select the type of camera in use.

Value list:

- External2D: external 2D camera
- LMI: LMI camera
- Mech-Eye (default value): Mech-Eye Industrial 3D Camera
- Mech-Eye TOF: customized Mech-Eye Industrial 3D Camera

- If the firmware of Mech-Eye Industrial 3D Camera is updated to 2.0.0, it is recommended to use Mech-Vision of version 1.6.2 and above. If you use Mech-Vision 1.6.1 or older versions, the software may crash when you connect the Mech-Eye Industrial 3D Camera in the **External2D EBUSCamera** category in the Choose the camera and parameter group to use window.
- The **Virtual Mode** is supported only when Mech-Eye camera type is selected.

Tuning recommendation: Different camera types have different parameters. Please adjust according to the actual situation.

Mech-Eye

When the **Virtual Mode** is disabled, a real camera can be used. When the **Virtual Mode** is enabled, local data will be used to run the Step. The Virtual Mode is disabled by default.

Use a Real Camera

Camera ID

Description: This parameter is used to connect the camera and select the camera calibration parameters.

Instructions:

1. Click **[Select camera]** to open the **Choose the camera and parameter group to use** window.
2. Select the camera you want to connect and click 🔗.

   > If the camera is connected successfully, 🔗 will become ⛓.

3. After the camera is connected successfully, you can click **Select from** and select the camera calibration parameter group in the drop-down list and then click **[OK]** in the lower right corner of the window.

   > Once the calibration parameter group is selected successfully, the **Camera ID**, **Camera Calibration Parameters**, **IP Address**, and **Port** parameters will be automatically filled.

Camera Calibration Parameters

Description: This parameter is used to specify the calibration parameter group of the camera.

IP Address

Description: This parameter is used to display the IP address of the camera. Please do not modify.

Port

Description: This parameter is used to display the port number of the camera. Please do not modify.

> **IP Address** and **Port** are only used for display. Modifying these two parameters will disconnect the camera.

Timeout

Description: This parameter is used to set the maximum response timeout period (in seconds) from the camera receives the data to it sends the data to the client. The two types of timeout are as follows:

- The camera fails to be connected successfully within the set time
- The camera fails to capture any images within the set time

Default value: 20 s

Num of Reconnection Attempts

Description: This parameter is used to specify the maximum number of attempts to reconnect the camera if the camera fails to be connected within the timeout period.

Default value: 3

Camera Config Group

Description: This parameter is used to select the configuration parameter group of the camera. The camera will capture the data according to the parameters in this parameter group.

> You can add/delete a parameter group in Mech-Eye Viewer.

Camera Model

Description: This parameter is only used to display the model of the currently connected camera.

2D Image Type

Description: This parameter is used to set the 2D image type output by the camera.

Value list:

- Internal IR Image: Default value. 2D images (depth source) whose reference frame is the same as that of the depth map and does not need to be rectified.
- External Color Image: 2D images (texture) that should be rectified before use. You should select **Rectify to Depth Map** when 2D images of this type are output.

> This parameter is only available when a DEEP (V4) or LSR (V4) series camera is used.

Rectify to Depth Map

Description: This parameter is used to rectify the reference frame of the external color image to the reference frame of the depth map.

Default value: Selected.

> This parameter is only available when **External Color Image** is set as the **2D Image Type**.

Background Removal Settings

Remove Background by Depth

Description: This parameter is used to remove the information of the background from the depth map.

Default Value: Disabled.

Depth Background Image File

Description: This parameter is used to capture or read the depth map of the background.

Default value: depth_background.png

Instructions:

1. Click the **[Open an Editor]** button to open the Set Background window.
   - If there is an available image of the background, please see step 2.
   - If there is not an available image of the background, please see step 3.
2. Click [...] and select the local image map of the background. Then, click the [OK] button to confirm.
3. Select a camera in the **Detected cameras** panel on the right and click 🔗 to connect the camera.

   > Once the camera is connected successfully, the depth map obtained from the camera will be displayed in the **Image viewer** panel on the left.

4. If you want to update the depth map, click the **[Capture once]** or **[Capture live]** button.
5. If you want to modify the file name of the depth map, enter an **Image Name** in the text box.
6. Click **[OK]** to finish selecting the depth map of the background.

Variation of Background Depth

Description: This parameter is used to compare the depth (mm) of a point in the newly-obtained depth map and the depth of the point in the background depth map. When the difference in depths between the point and its counterpart in the background is less than this value, the point will be considered a point in the background and removed.

Default value: 10.0 mm

Max Capture Attempts

Description: This parameter is used to specify the total number of attempts to capture the image if the camera fails to capture any image within the timeout period.

Default value: 3

Recommended value: 3

Robot Service Name in Mech-Center

Description: This parameter is used to specify the robot service name, which should be the same as the name of the robot connected in Mech-Center.

Default value: Null.

Use a Virtual Camera

When the **Virtual Mode** is enabled, you will need to adjust the following parameters.

Camera Calibration Parameters

Instruction: Once the **Data Path** is selected, **Camera Calibration Parameters** will be automatically filled. If there are multiple sets of camera calibration parameters under the **Data Path**, please select the one you need in the drop-down list.

Data Path

Description: This parameter is used to select the folder where images, intrinsic parameters, and extrinsic parameters are stored.

Instruction: Click the 📁 icon on the right and select the folder where images, intrinsic parameters, and extrinsic parameters are stored.

> In the virtual mode, the naming conventions for the images as follows should be followed. The serial number of the color image should be the same as its depth map counterpart.
>
> - Color images: rgb_image_xxxx.jpg
> - Depth maps: depth_image_xxxx.png
>
> If the camera mounting method is Eye-in-Hand, note that the serial number of images should correspond to that of flange poses.

Virtual Camera Assistant:

- If any one of the following descriptions is true when you select the data path, a **Virtual Camera Assistant** will be triggered to help you select the image data.
  - You have not entered a path in the **Data Path** box.
  - There are multiple sets of data in the selected folder.

- There is no color image, depth map, or intrinsic and extrinsic parameters in the selected folder.

- Please follow the steps below to configure in Virtual Camera Assistant:
  - In the **Virtual Camera Assistant** window, click 🗀 to select the data path.

    > Once the path is entered successfully, the **Camera ID** and **Parameter group** will be automatically filled.

  - Click **[Verify]**, and a **Camera parameter group updated** message will appear.
  - Click 🗀 to select the folders where color images, depth maps, and the flange pose are stored one by one. Then, click **[Confirm]** after selecting the data.

  > The flange pose is needed only when the camera is mounted in the eye-in-hand mode.

- Lastly, click **[OK]** in the pop-up **Setup Complete** window.

> In the virtual mode, please re-select the data path if there is any modification of the image in the selected data path, or else the new image data cannot be read.

Play Back Mode

Description: This parameter is used to specify the order to read the images.

Value list:

- Sequential (default value): Read the images in the order of the images in the folder.
- Repeat One: Read the current image repeatedly.
- Repeat All: Read all images in the order of the images in the folder, and then read them from the beginning after all images are read.
- Random: Read images randomly.

Tuning recommendation: Please select the mode according to image reading order you need.

Current Frame Name

Description: This parameter is used to display the serial number and timestamp of the currently read image.

Image Name Type

Description: This parameter is used to select the type of image name output by the **Color Image Path** port.

Value list: Complete Path, File Name, and Base Name.

Default value: Complete Path.

Background Removal Settings

Remove Background by Depth

Description: This parameter is used to remove the information of the background from the depth map.

Default Value: Disabled.

Depth Background Image File

Description: This parameter is used to capture or read the depth map of the background.

Default value: depth_background.png

Instructions:

1. Click the **[Open an Editor]** button to open the Set Background window.
   - If there is an available image of the background, please see step 2.
   - If there is not an available image of the background, please see step 3.
2. Click **[...]** and select the local image map of the background. Then, click the **[OK]** button to confirm.
3. Select a camera in the **Detected cameras** panel on the right and click 🔗 to connect the camera.

   > Once the camera is connected successfully, the depth map obtained from the camera will be displayed in the **Image viewer** panel on the left.

4. If you want to update the depth map, click the **[Capture once]** or **[Capture live]** button.
5. If you want to modify the file name of the depth map, enter an **Image Name** in the text box.
6. Click **[OK]** to finish selecting the depth map of the background.

Variation of Background Depth

Description: This parameter is used to compare the depth (mm) of a point in the newly-obtained depth map and the depth of the point in the background depth map. When the difference in depths between the point and its counterpart in the background is less than this value, the point will be considered a point in the background and removed.

Default value: 10.0 mm

External2D

Camera ID

Description: This parameter is used to connect the camera and select the camera calibration parameters.

Instructions:

1. Click **[Select camera]** to open the **Choose the camera and parameter group to use** window.
2. Select the camera you want to connect and click 🔗.

   > If the camera is connected successfully, 🔗 will become ⛓️‍💥.

3. After the camera is connected successfully, you can click **Select from** and select the camera calibration parameter group in the drop-down list and then click **[OK]** in the lower right corner of the window.

   > Once the calibration parameter group is selected successfully, the **Camera ID**, **Camera Calibration Parameters**, **IP Address**, and **Port** parameters will be automatically filled.

Camera Calibration Parameters

Description: This parameter is used to specify the calibration parameter group of the camera.

IP Address

Description: This parameter is used to display the IP address of the camera. Please do not modify.

Port

Description: This parameter is used to read the port number of the camera.

IP Address and **Port** are only used for reading. Modifying these two parameters will disconnect the camera.

Timeout

Description: This parameter is used to set the maximum response timeout period (in seconds) from the camera receives the data to it sends the data to the client. The two types of timeout are as follows:

- The camera fails to be connected successfully within the set time
- The camera fails to capture any images within the set time

Default value: 20 s

Num of Reconnection Attempts

Description: This parameter is used to specify the maximum number of attempts to reconnect the camera if the camera fails to be connected within the timeout period.

Default value: 3

Auto Exposure

Description: This parameter is used to determine whether to enable automatic exposure.

Default value: Selected.

Mean Gray Value

Description: This parameter is used to set the expected gray value of the image. The larger the value, the brighter the image. The smaller the value, the darker the image.

Default value: 100

Exposure Time

Description: This parameter is used to set the exposure time which affects the brightness of the image. The larger the value, the brighter the image. The smaller the value, the darker the image.

Default value: 10

Max Capture Attempts

Description: This parameter is used to specify the total number of attempts to capture the image if the camera fails to capture any image within the timeout period.

Default value: 3

Recommended value: 3

Robot Service Name in Mech-Center

Description: This parameter is used to specify the robot service name, which should be the same as the name of the robot connected in Mech-Center.

Default value: Null.

LMI

Camera ID

Description: This parameter is used to connect the camera and select the camera calibration parameters.

Instructions:

1. Click **[Select camera]** to open the **Choose the camera and parameter group to use** window.
2. Select the camera you want to connect and click 🔗.

   > If the camera is connected successfully, 🔗 will become ⛓.

3. After the camera is connected successfully, you can click **Select from** and select the camera calibration parameter group in the drop-down list and then click **[OK]** in the lower right corner of the window.

   > Once the calibration parameter group is selected successfully, the **Camera ID**, **Camera Calibration Parameters**, **IP Address**, and **Port** parameters will be automatically filled.

Camera Calibration Parameters

Description: This parameter is used to specify the calibration parameter group of the camera.

IP Address

Description: This parameter is used to display the IP address of the camera. Please do not modify.

Max Capture Attempts

Description: This parameter is used to specify the total number of attempts to capture the image if the camera fails to capture any image within the timeout period.

Default value: 3

Recommended value: 3

Robot Service Name in Mech-Center

Description: This parameter is used to specify the robot service name, which should be the same as the name of the robot connected in Mech-Center.

Default value: Null.

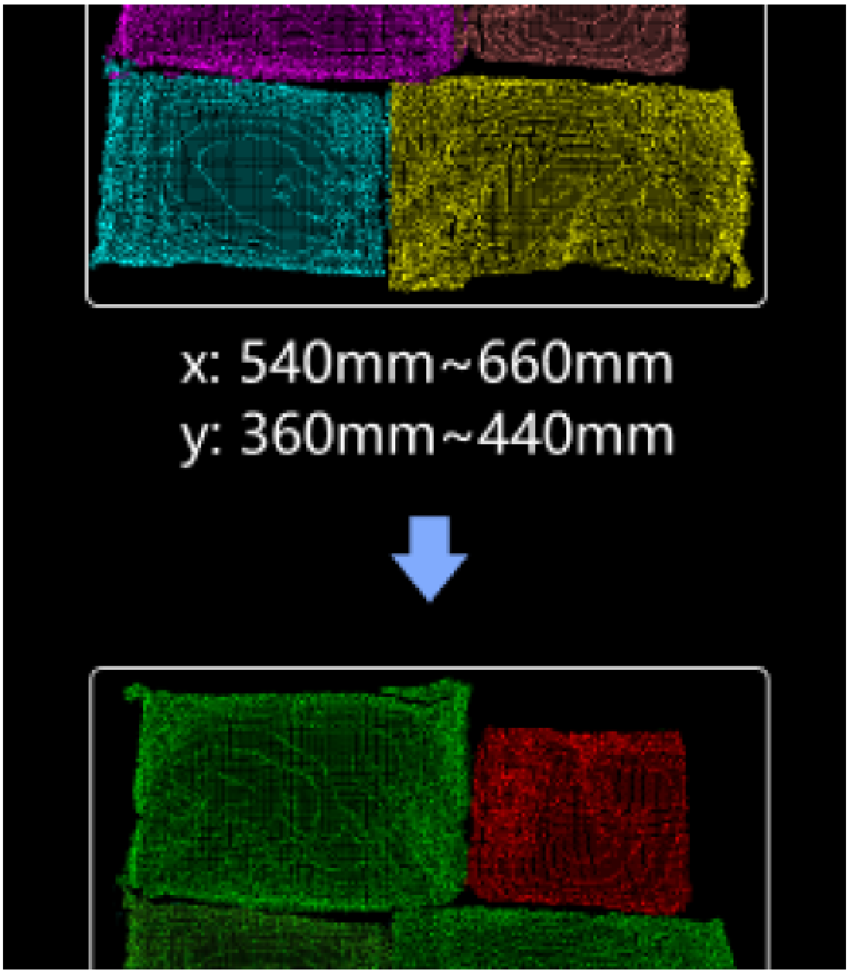
Mech-Eye TOF

Please contact Mech-Mind Technical Support for assistance in using the parameters of this mode.

4.3.72. Classify Point Clouds by Dimensions

Function

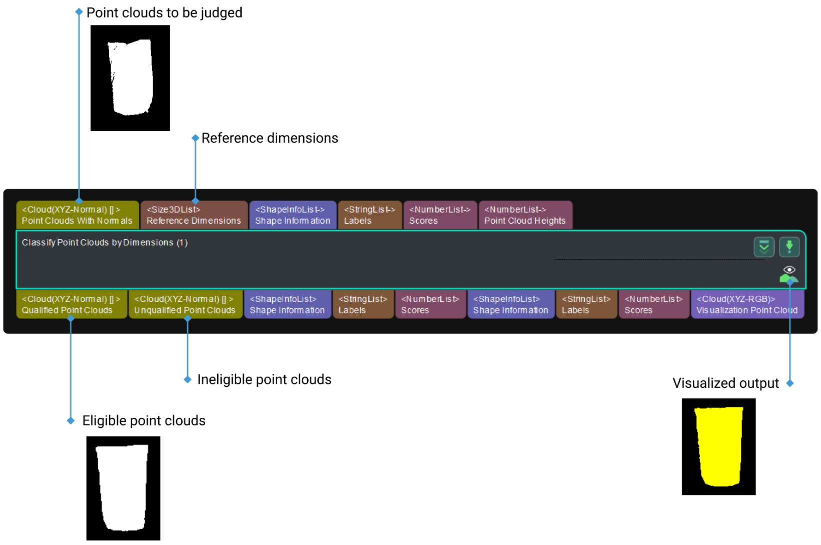
This Step judges whether each of the input point clouds is eligible based on the reference dimensions. If the deviation between the dimensions of an input point cloud and the reference dimensions is within the threshold range, the point cloud is eligible. Otherwise, it is not eligible.



Usage Scenario

This Step is usually used in sack depalletizing. To avoid being affected by errors in deep learning sack recognition results, this Step is used to add an error-proofing mechanism to the 3D algorithms by determining whether the sack point clouds from previous Steps are eligible or not. This Step is usually used with such Steps as Apply Masks to Point Cloud and Get Highest Layer Clouds.

Input and Output



Length Difference Threshold Settings

LengthDifference = ActualLengthofthePointCloud − ReferenceLength

Lower Ratio Threshold

Description: This parameter specifies the minimum ratio of the length difference to the reference length. If the ratio of the length difference to the reference length is above this threshold, the point cloud is eligible; otherwise, it is not eligible.

Default value: −0.2500

Value range: −1 to 0

Tuning recommendation: Please set the value according to the actual situation.

Upper Ratio Threshold

Description: This parameter specifies the maximum ratio of the length difference to the reference length. If the ratio of the length difference to the reference length is below this threshold, the point cloud is eligible; otherwise, it is not eligible.

Default value: 0.2500

Value range: 0 to 1

Tuning recommendation: Please set the value according to the actual situation.

- If Lower Ratio Threshold < Length Difference/Reference Length < Upper Ratio Threshold, the point cloud is eligible.
- If Length Difference/Reference Length > Upper Ratio Threshold, the point cloud is ineligible.
- If Length Difference/Reference Length < Lower Ratio Threshold, the point cloud is ineligible.

Width Difference Threshold Settings

WidthDifference = ActualWidthofthePointCloud − ReferenceWidth

Lower Ratio Threshold

Description: This parameter specifies the minimum ratio of the width difference to the reference width. If the ratio of the width difference to the reference width is above this threshold, the point cloud is eligible; otherwise it is not eligible.

Default value: −0.2500

Value range: −1 to 0

Tuning recommendation: Please set the value according to the actual situation.

Upper Ratio Threshold

Description: This parameter specifies the maximum ratio of the width difference to the reference width. If the ratio of the width difference to the reference width is below this threshold, the point cloud is eligible; otherwise it is not eligible.

Default value: 0.2500

Value range: 0 to 1

Tuning recommendation: Please set the value according to the actual situation.

- If Lower Ratio Threshold < Width Difference/Reference Width < Upper Ratio Threshold, the point cloud is eligible.
- If Width Difference/Reference Width > Upper Ratio Threshold, the point cloud is ineligible.
- If Width Difference/Reference Width < Lower Ratio Threshold, the point cloud is ineligible.

Height Difference Threshold Settings

HeightDifference = ActualHeightofthePointCloud − ReferenceHeight

Lower Ratio Threshold

Description: This parameter specifies the minimum ratio of the height difference to the reference height. If the ratio of the height difference to the reference height is above this threshold, the point cloud is eligible; otherwise it is not eligible.

Default value: −100.0000

Tuning recommendation: Please set the value according to the actual situation.

Upper Ratio Threshold

Description: This parameter specifies the maximum ratio of the height difference to the reference height. If the ratio of the height difference to the reference height is below this threshold, the point cloud is eligible; otherwise it is not eligible.

Default value: 100.0000

Tuning recommendation: Please set the value according to the actual situation.

Reference Height

Description: This parameter is used to set the reference height (in mm).

Default value: 1000.000 mm

Tuning recommendation: Please set the value according to the actual situation.

- If Lower Ratio Threshold < Height Difference/Reference Height < Upper Ratio Threshold, the point cloud is eligible.
- If Height Difference/Reference Height > Upper Ratio Threshold, the point cloud is ineligible.
- If Height Difference/Reference Height < Lower Ratio Threshold, the point cloud is ineligible.

Aspect Ratio Settings

AspectRatioDifference = ActualAspectRatioofthePointCloud − ReferenceAspectRatio

Aspect Ratio Threshold

Description: This parameter is used to set the threshold of the ratio of the aspect ratio difference to the reference aspect ratio. If the ratio of the aspect ratio to the reference height is within the range “1 ± aspect ratio difference/reference aspect ratio”, the point cloud is eligible; otherwise, it is not eligible.

Default value: 1000000.0000

Tuning recommendation: Please set the value according to the actual situation.

- If −(Aspect Ratio Threshold) < Aspect Ratio Difference/Reference Aspect Ratio < Aspect Ratio Threshold, the point cloud is eligible.
- If Aspect Ratio Difference/Reference Aspect Ratio > Aspect Ratio Threshold, the point cloud is ineligible.
- If Aspect Ratio Difference/Reference Aspect Ratio < −(Aspect Ratio Threshold), the point cloud is ineligible.

4.3.73. Correct Point Cloud Distortion

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step performs distortion correction on the input point cloud through the camera’s distortion coefficients.

Usage Scenario

This Step is a custom Step for special scenarios.

Input and Output

- **Input**: Point cloud input to this port will be corrected.
- **Output**: corrected point cloud.

4.3.74. Filter Point Cloud by Model and Poses

This Step is outdated. Please use the newer Step Remove Cloud Points from Point Cloud instead. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step compares the input point cloud with the model loaded in the parameters, and overlapped points will be removed.

Usage Scenario

This Step is usually used to remove/obtain the point cloud parts corresponding to the models from the scene.

Input and Output

- Input:
  1. Point cloud input to this port will be filtered.
  2. Poses used to transform the models into the scene.
- Output:
  1. Remaining point cloud
  2. Removed point cloud

4.3.75. Cloud Num Limit

This Step is outdated. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step retains the first N point clouds (or remove the last N point clouds) according to the order of the point clouds in the point cloud list, and the value of N can be set in the Step's parameters.

Usage Scenario

This Step is used to truncate a point cloud list.

This Step is outdated. Please use the newer Steps instead.

Input and Output

- Input:
  1. The point cloud list input to this port will be truncated.
  2. Labels corresponding to input point clouds.
  3. Confidences corresponding to input point clouds.
- Output:
  1. Point cloud list after truncating.
  2. Labels corresponding to point clouds after truncating.
  3. Confidences corresponding to point clouds after truncating.

4.3.76. Cloud Processing (GPU)

This Step is outdated. Please use the newer Step Calc Normals of Point Cloud and Filter It

> instead. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

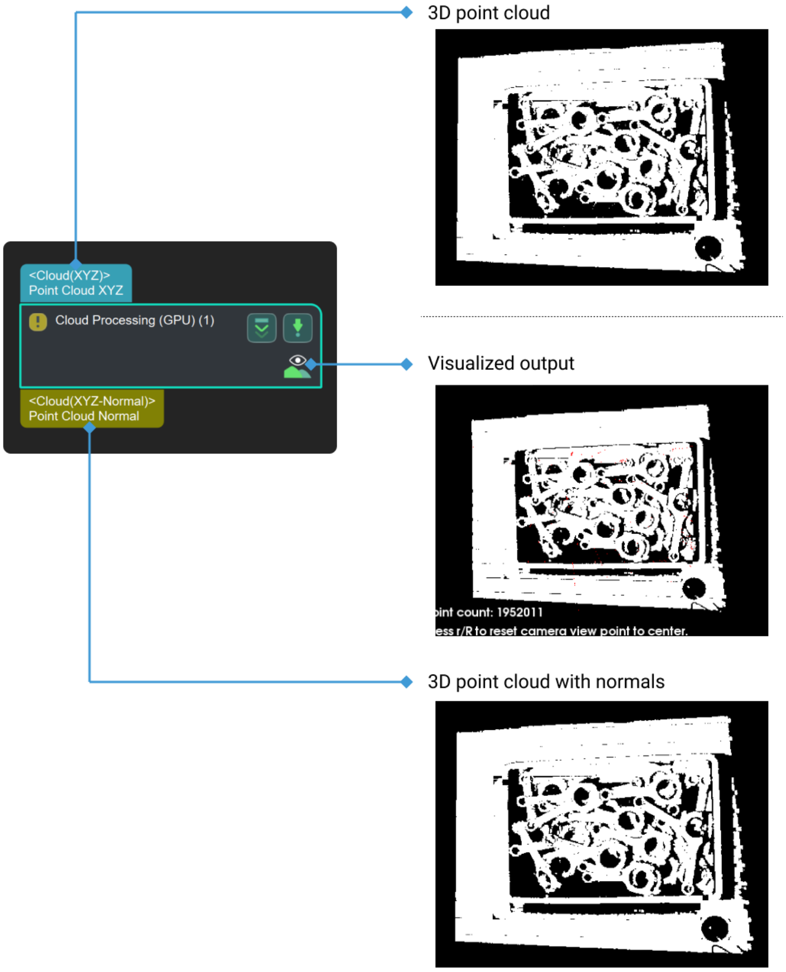
Function

Calculate normals of the input point cloud and remove outliers.

Usage Scenario

This Step is used to add normals to the original point cloud and remove outliers. It usually follows Steps Capture Images from Camera and From Depth Map to Point Cloud.

Input and Output



Parameter Description

Calculation Settings

Is Batched

Description: This parameter is used to determine whether to process the point clouds in batches.

Default setting: unselected.

Suggestion: When there are too many points in a point cloud, and the video RAM is not enough to process them, it is recommended to select this option.

Radius for Normal Calculation

Description: This parameter is used to set the search radius for normal calculation.

Default value: 5.000 mm

Removal of Sparse Points

Remove Sparse

Description: This parameter is used to determine whether to remove the sparse points in point clouds.

Default setting: unselected.

Instruction: When it is selected, sparse points can be removed according to the set threshold of sparse points.

4.3.77. Scale Point Cloud

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step multiplies the coordinates of each point in the input point cloud by the scale.

Usage Scenario

This Step is a general point cloud processing Step. This Step has no fixed usage scenarios.

Input and Output

- **Input**: Point cloud input to this port will be scaled.
- **Output**: scaled point cloud.

4.3.78. Smooth Point Cloud and Estimate Normals

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step performs point cloud smoothing and normal calculation.

Usage Scenario

This Step is usually used in scenarios with large fluctuations in point clouds.

Input and Output

- **Input**: Point cloud input to this port will be smoothed and have normals re-calculated.
- **Output**: Point cloud after smoothing and normal re-calculating.

4.3.79. Extract Point Cloud in 3D Box

The contents of this Step are outdated. Please use the new Step Extract 3D Points in 3D ROI. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step extracts the point clouds within the 3D ROI.

Usage Scenario

This Step is a general point cloud extraction Step. This Step has no fixed usage scenarios.

Input and Output

- **Input**: Point clouds input to this port will have parts in 3D ROI extracted.
- **Output**: point clouds in 3D ROI.

4.3.80. From CloudXYZ To CloudNormal

This Step is outdated. Please use the new Step Calc Normals of Point Cloud and Filter It and set the normal calc method to **CloudXYZToCloudNormal**. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step adds normals to a point cloud in the camera reference frame.

Usage Scenario

This Step is a general point cloud processing Step.

Input and Output

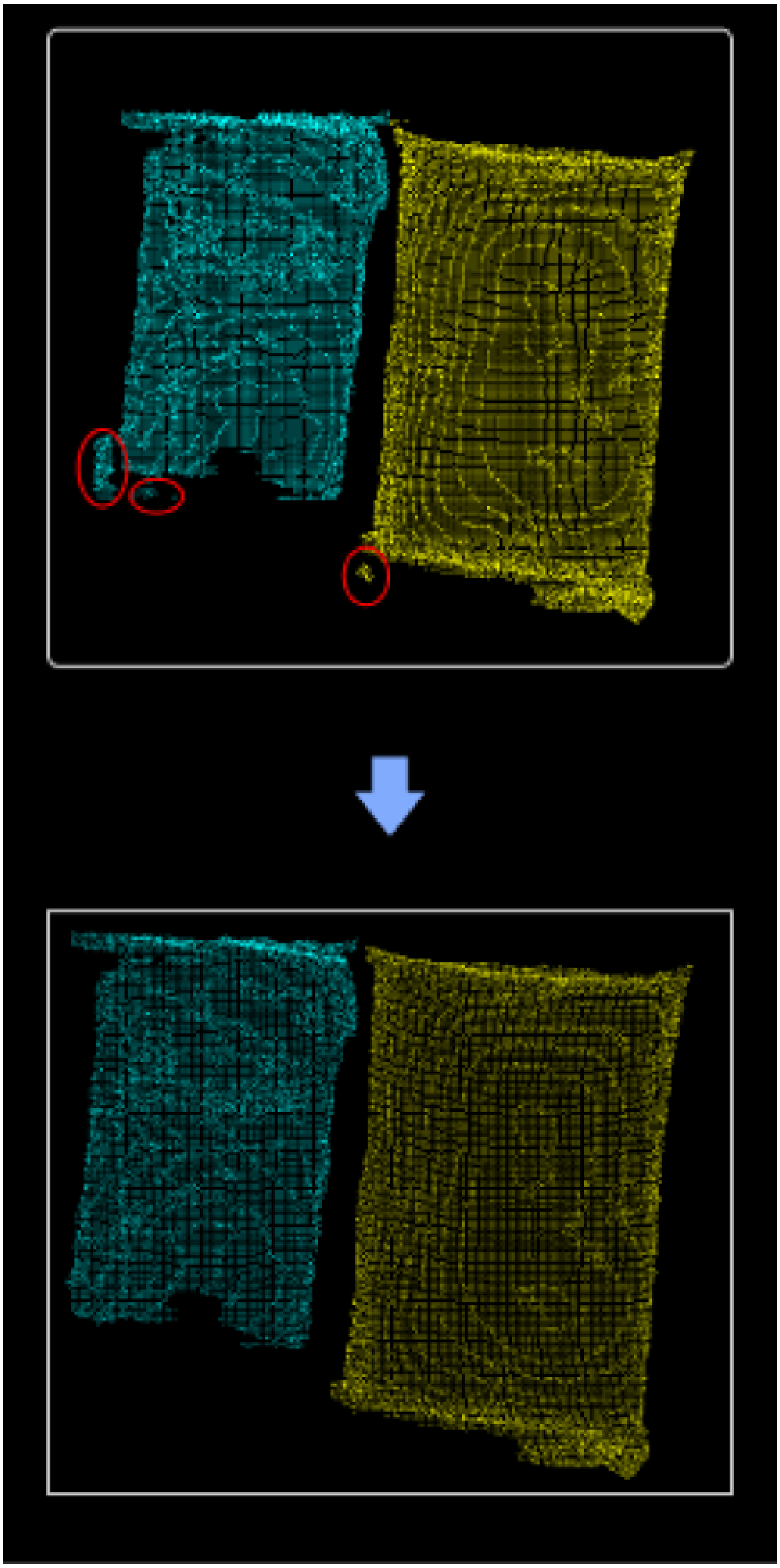
- **Input**: 3D point cloud.
- **Output**: 3D point cloud with normals.

4.3.81. Cluster Point Clouds and Output Eligible Point Clouds

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

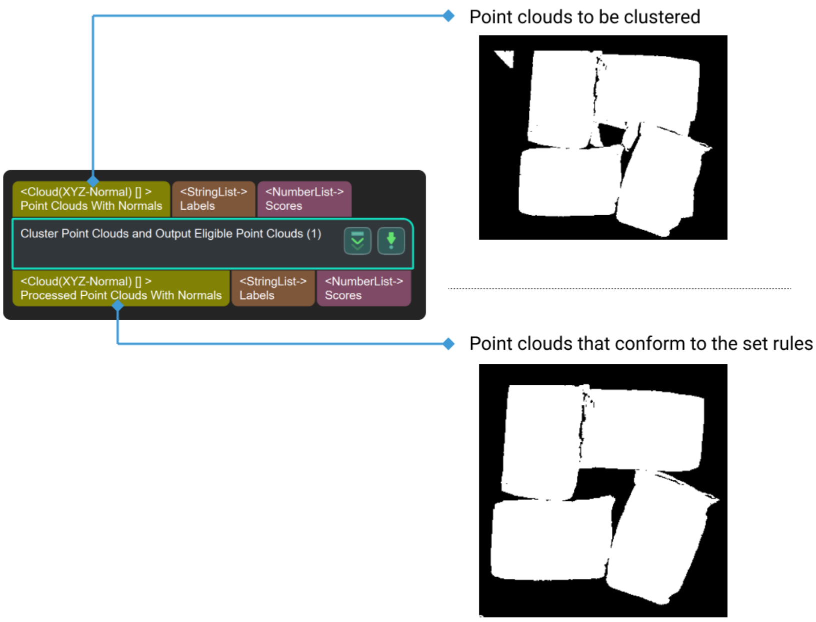
Perform clustering operation on point clouds, and output the point clouds that conform to the set rules in the clustering result.



Usage Scenario

This Step is usually used to denoise point clouds of individual objects.

Input and Output


Point clouds to be clustered
<Cloud(XYZ-Normal) [] >
Point Clouds With Normals
<StringList->
Labels
<NumberList->
Scores
Cluster Point Clouds and Output Eligible Point Clouds (1)
<Cloud(XYZ-Normal) [] >
Processed Point Clouds With Normals
<StringList->
Labels
<NumberList->
Scores
Point clouds that conform to the set rules

4.3.82. Draw Coherent Lines

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Steps detects object contour line segments in the input 2D image and draws the line segments as coherent lines.

Usage Scenario

This Step is a general 2D image edge connecting Step. This Step has no fixed usage scenarios.

Input and Output

- **Input:** Image input to this port will be drawn with coherent lines of object contours.
- **Output:** masks of coherent lines.

4.3.83. Collect Poses in 3D ROI

This Step is outdated. Please use the new Step Validate Existence of Poses in 3D ROI instead. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

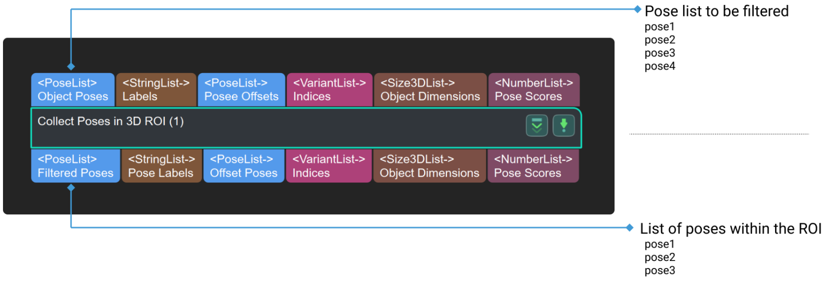
Function

Filter the input pose list by the set 3D ROI. The poses inside the ROI will be retained, and the poses outside the ROI will be discarded.

Usage Scenario

A general pose filtering Step.

Input and Output



<PoseList> Object Poses
<StringList-> Labels
<PoseList-> Posee Offsets
<VariantList-> Indices
<Size3DList-> Object Dimensions
<NumberList-> Pose Scores
Collect Poses in 3D ROI (1)
<PoseList> Filtered Poses
<StringList-> Pose Labels
<PoseList-> Offset Poses
<VariantList-> Indices
<Size3DList-> Object Dimensions
<NumberList-> Pose Scores
Pose list to be filtered
pose1
pose2
pose3
pose4
List of poses within the ROI
pose1
pose2
pose3

Parameter Description

3D ROI Settings

Set 3D ROI

Description: This parameter is used to set 3D ROI.

Instruction: Click **[Set 3D ROI]** and set the background in the pop-up window of Instructions for Setting 3D ROI.

Input Poses Coordinate Type

Description: This parameter is used to select the type of reference frame in which the input poses are located.

Options: Robot Coordinate, Camera Coordinate.

Default setting: Robot Coordinate.

4.3.84. Compare Two Depth Maps

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step compares the depth values of each pixel in two depth maps and visualizes the depth differences as a 2D image. It is used for adjusting the performance of imaging algorithms in passive and active projection modes.

Usage Scenario

This Step is for developers and is used for adjustment in the passive stereo camera mode. Regular users do not need to use this Step in general application scenarios.

Input and Output

- Input:
  1. First depth map used to calculate the difference in depth values

2. Second depth map used to calculate the difference in depth values

- Output:

1. Image representing the depth differences between the pixels of the first and second input depth maps

4.3.85. Compare Values

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

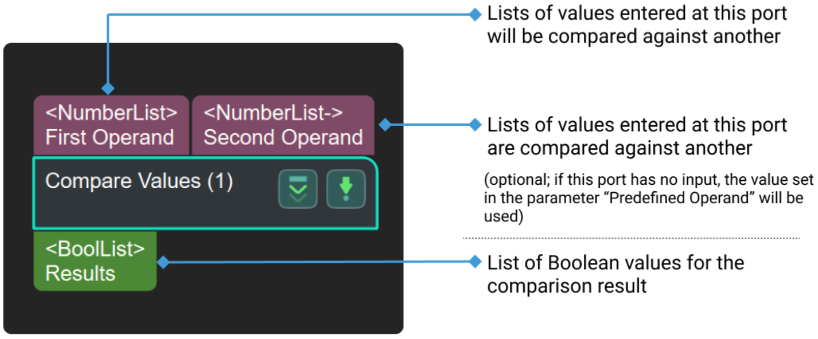
Function

According to the set operation type, compare the two input value lists and output the Boolean value (True/False) list of the comparison result.

Usage Scenario

A general value comparison Step. No fixed usage scenarios.

Input and Output



Lists of values entered at this port will be compared against another

Lists of values entered at this port are compared against another

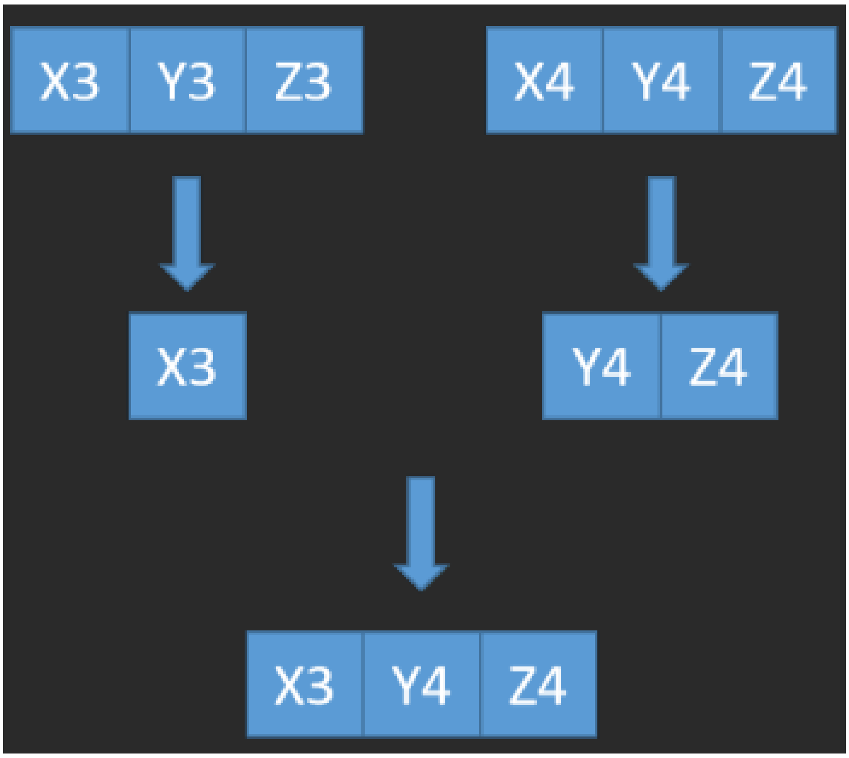
(optional; if this port has no input, the value set in the parameter “Predefined Operand” will be used)

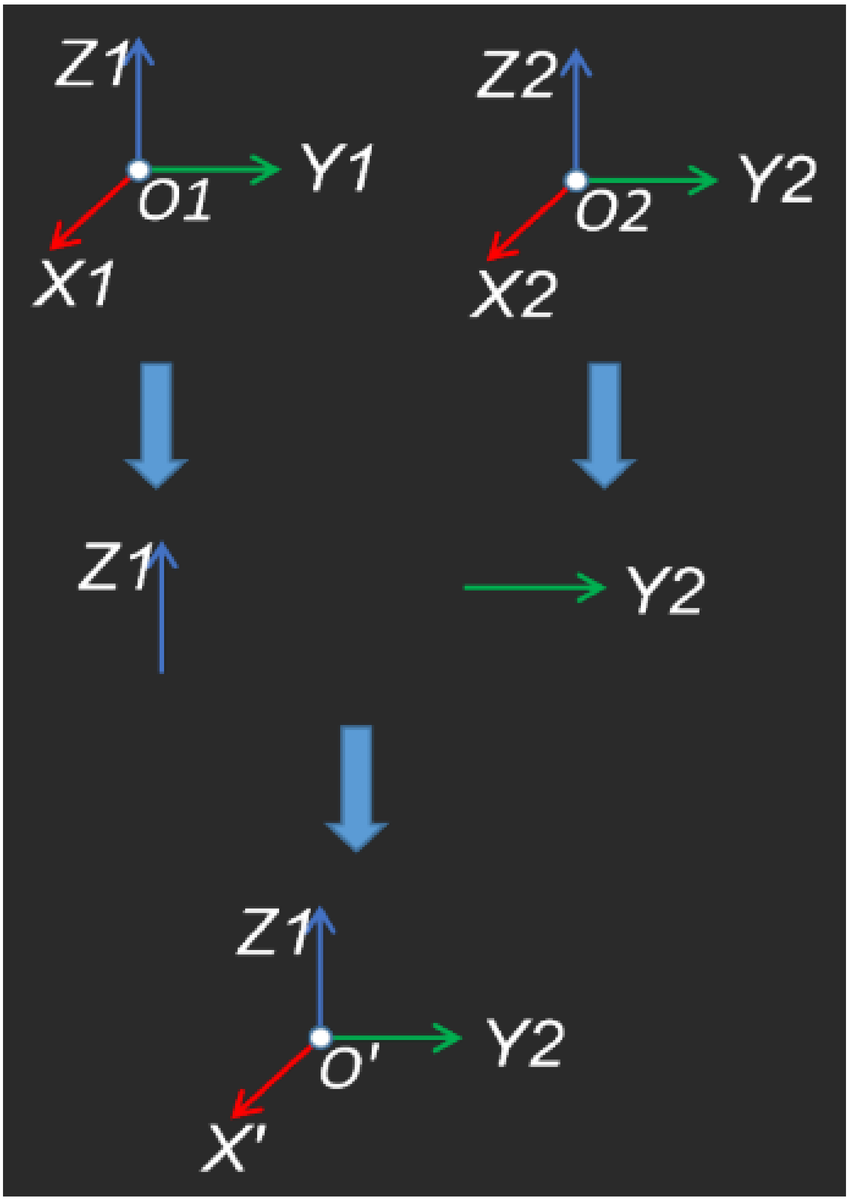
List of Boolean values for the comparison result

4.3.86. Compose New Poses by Combining Parts of Input Poses

Function

Combines the specified parts of the input poses to form a new pose.

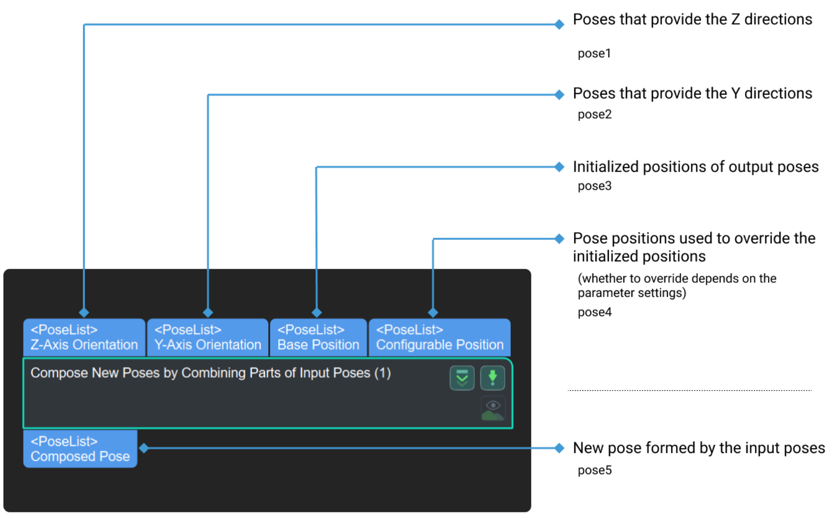

X3
Y3
Z3
X4
Y4
Z4
X3
Y4
Z4
X3
Y4
Z4



Usage Scenario

A general pose calculation Step. No fixed usage scenarios.

Input and Output


Poses that provide the Z directions
pose1
Poses that provide the Y directions
pose2
Initialized positions of output poses
pose3
Pose positions used to override the initialized positions
(whether to override depends on the parameter settings)
pose4
<PoseList> Z-Axis Orientation
<PoseList> Y-Axis Orientation
<PoseList> Base Position
<PoseList> Configurable Position
Compose New Poses by Combining Parts of Input Poses (1)
<PoseList> Composed Pose
New pose formed by the input poses
pose5

Parameters

Visualization Settings

Pose Type to Visualize

Default setting: After.

List of options: Before, After, and All.

Instruction: This parameter decides the type of pose to visualize, including pose after processing, pose before processing, and poses before and after processing.

The Axes of Fourth Input Pose to Use

Default value: X.

List of options: X, Y, Z, XY, XZ, YZ, XYZ, and None.

Instruction: Select the translation parameter. For example, for the selection of X, the corresponding operation is to set the X axis with the third input while Y and Z axes with the fourth input.

4.3.87. Compose Poses from Quaternions and Translation Vectors

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step combines the input rotation vectors (in quaternions) and translation vectors into poses.

Usage Scenario

This Step is a general pose creation Step. No fixed usage scenarios.

Input and Output

- Input:

1. Rotation vectors (in quaternions) input to this port will be used to compose the poses.
2. Pose centers (as translation vectors) input to this port will be used to compose the poses.

- Output:
  1. Poses composed of the input rotation vectors and pose centers

4.3.88. Compose Quaternions from Axes and Angles

> Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step combines the Vector3D type vectors and the rotation angles into rotation vectors.

Usage Scenario

This Step is a general rotation vector composing Step. No fixed usage scenarios.

Input and Output

- Input:
  1. Axes input at this port will be used to create the rotation vectors.
  2. Angles input at this port will be used to create the rotation vectors.
- Output:
  1. Created rotation vectors, in quaternions.

4.3.89. Compose Quaternions from Two Axes (Right Hand Rule)

> Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

Taking the input two vectors as the specified axes and using the right-hand rule, this Step generates a rotation vector in quaternions. If the two input axes are not perpendicular, after the third axis is generated by the right-hand rule, the axis from the second port will be reserved, and the rotation vector will be generated from the axis of the second port and the generated third axis using the right-hand rule again.

Usage Scenario

This Step is a general data conversion and formation Step. No fixed usage scenarios.

Input and Output

- Input:
  1. 3D vector input to this port will be used as the starting axis for rotation.
  2. 3D vector input to this port will be used as the end axis of the rotation.
- Output:

1. Quaternions representing the rotation vector

4.3.90. Compose Vector3Ds from Numbers

Function

This Step combines three numeric values (X, Y, Z) into a vector of type Vector3D. Multiple sets of (X, Y, Z) values can be input at a time to compose multiple vectors.

Usage Scenario

This Step is a general data type conversion Step. This Step has no fixed usage scenarios.

Input and Output

- Input:
  1. X values for combining into vectors
  2. Y values for combining into vectors
  3. Z values for combining into vectors
- Output:
  1. Vectors formed by combining inputs

4.3.91. Control Light Source

> ℹ Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step controls the light source by Mech-Vision.

4.3.92. Convert 2D Poses to 3D Poses

> ℹ Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step rotates the input 3D poses around the Z axes so that their X axes coincide with the input 2D poses' X axes. The X-axis direction and Z value of new 3D poses conform to those of referred 3D poses. In addition, their X-axis direction, X value, and Y value are determined by the respective values of input 2D poses.

Usage Scenario

This Step is a custom Step for special scenarios.

Input and Output

- Input:
  1. 2D poses input from this port will be used to adjust the 3D poses.

2. 3D poses input to this port will be rotated.

- Output:
  1. Rotated 3D poses

4.3.93. Convert Data Type

Function

You can customize the input/output port type of this Step, and the data type of the input port will be converted to the data type of the output port.

Usage Scenario

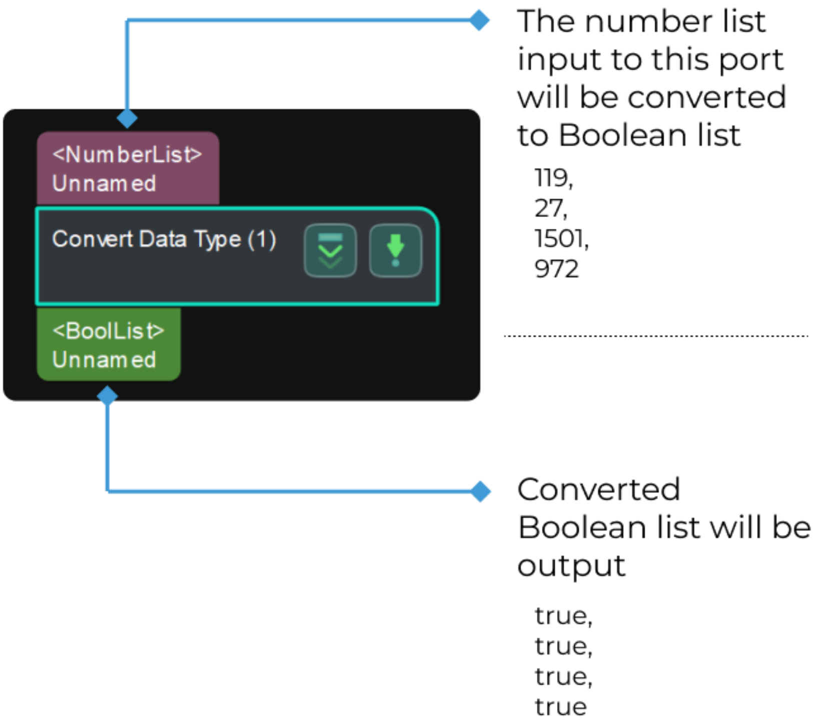
This Step is a general data conversion Step, with no fixed usage scenarios. It can be connected with other Steps according to actual data-type conversion requirements.

Input and Output

The input data type and the corresponding output data type(s) to which the input can be converted are listed in the table below. You can configure the ports of input data type and output data type according to the actual requirement.

Input Data Type	Output Data Type
Pose2DList	StringList
BoolList	StringList, String, String
StringList	BoolList, String, NumberList
String	BoolList, StringList, NumberList
NumberList	BoolList, StringList, String, Variant, VariantList
LineSegmentList	StringList
Variant	BoolList, StringList, String, NumberList, VariantList
VariantList	BoolList, StringList, String, NumberList, Variant
Shape2DList	StringList

The figure below shows an example of converting NumberList to BoolList.



The number list input to this port will be converted to Boolean list

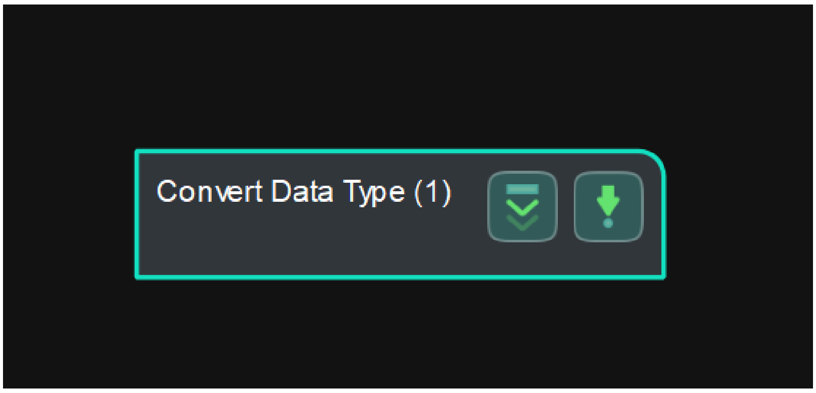
119,
27,
1501,
972

Converted Boolean list will be output

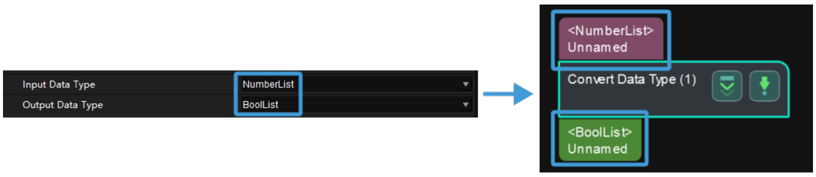
true,
true,
true,
true

Parameter Description

1. The Step does not have any input port or output port when it is just dragged to the graphical programming workspace, as shown below.

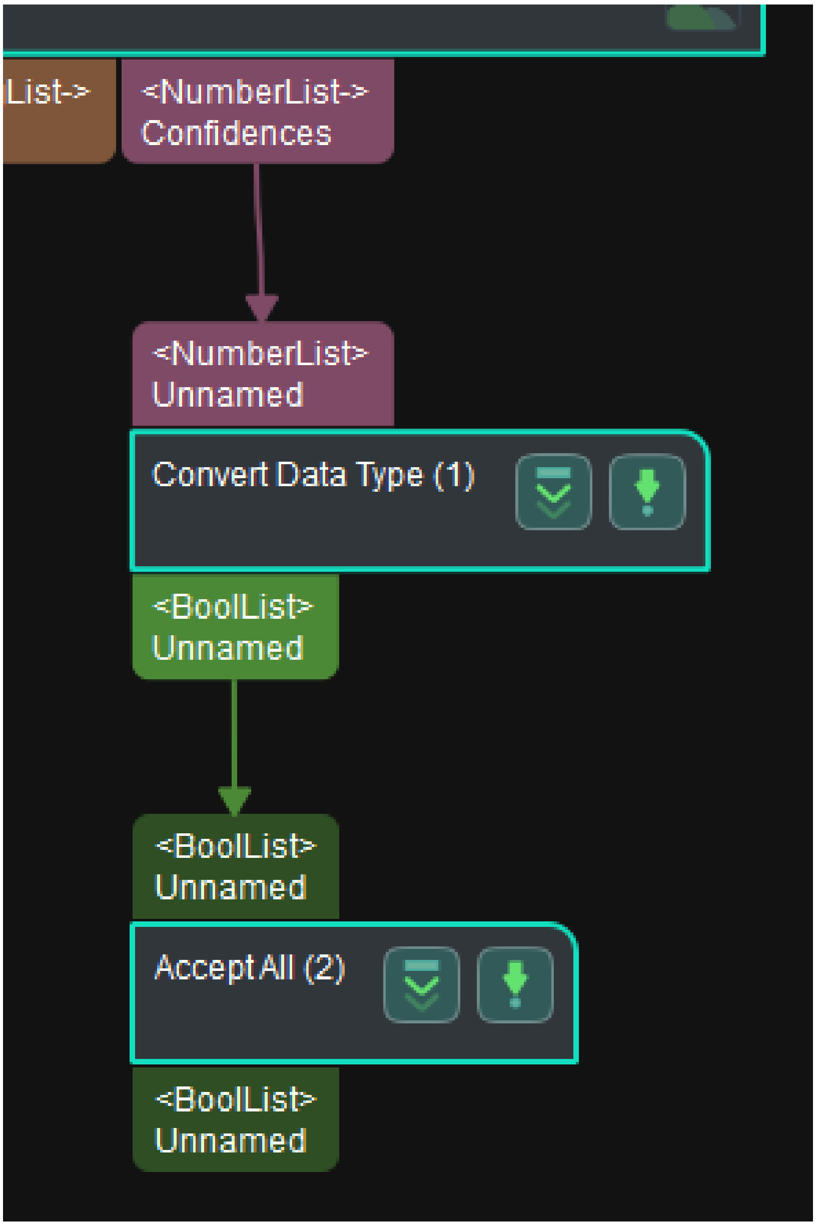


2. Go to the Step Parameters panel, select the input data type and output data type in the drop-down lists, and then corresponding ports will be added to the Step.


Input Data Type NumberList
Output Data Type BoolList

3. Connect Steps.

   An example of possible connection is shown below.



You should pay attention to the dependency relationship between input port and output port while using this Step:

- When you change the data type of either the input or the output port, if the new data type cannot be converted to/from the data type of the other port, the data type of this other port is reset.
- If the data type of a port is changed (including the reset mentioned above), connection(s) to this port is deleted, and you will need to reconnect the Steps.

4.3.94. Convert Lengths Pixel-Wise to Physical

Function

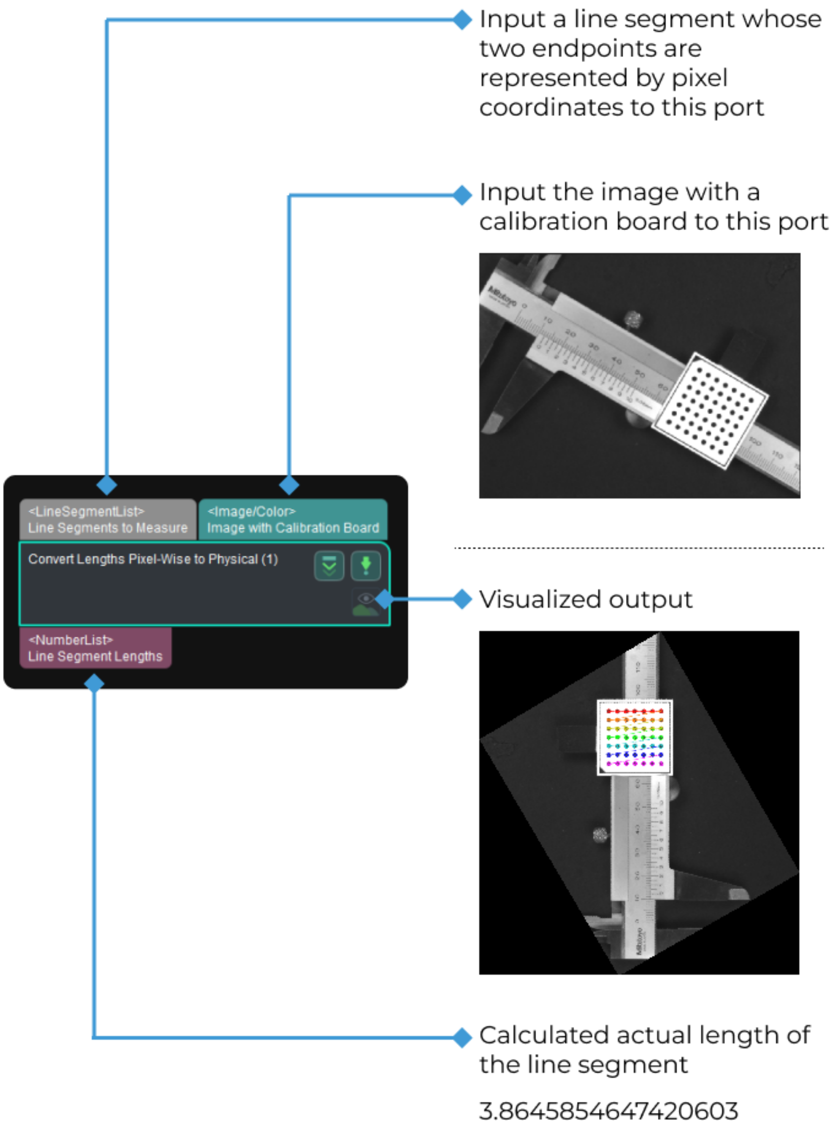
This Step is used to calculate the actual length of a specified line segment in 2D images. A calibration board is needed when using this Step, and it should be placed in the plane where the line segment to be measured is located.

Usage Scenario

This Step applies to scenarios where 2D images are used for measuring the dimensions of objects with relatively planar surfaces.

Since a line segment to be measured should be input, this Step usually follows Steps in the Measuring category with the LineSegmentList output port, such as Measure Distances Segments to Segments, Measure Distances Points to Circles, and Measure Distances Circles to Segments.

Input and Output



Input a line segment whose two endpoints are represented by pixel coordinates to this port

Input the image with a calibration board to this port

Visualized output

Calculated actual length of the line segment

3.8645854647420603

Parameter Description

Conversion Method

Description: This parameter specifies the conversion method to convert the length from pixel-wise to physical.

Options: Scale and Homography.

- Scale: The actual length of the line segment is equal to the product of the length in pixels multiplied by the scale.
- Homography: The actual length of the line segment will be calculated from the projection of the image plane onto the actual plane.

Default value: Scale.

Scale

When the conversion method is **Scale**, the actual length of the line segment is equal to the product of the length in pixels multiplied by the scale.

Scale

Description: The conversion ratio of pixel distance to actual distance.

Default value: 1.0000

Homography

When the conversion method is **Homography**, the Step will project the line segment in the image plane onto the actual plane to calculate the actual length of the line segment.

There are two ways to calculate the projection of the image plane onto the actual plane:

- **Calculate without Calibration**: Directly calculate the projection from the **Calibration Circle Center Distance**, **Number of Circle Rows**, and **Number of Circle Columns**, and then the actual length of the segment line can be obtained. The calculation with this method takes less time.
- **Calculate by Calibration**: Obtain the camera’s extrinsic and intrinsic parameters according to the **Calibration Circle Center Distance**, **Number of Circle Rows**, and **Number of Circle Columns** of the calibration board, and then calculate the projection of the image plane onto the actual plane. Lastly, **Calibration Board Plane Height** and **Measured Plane Height** will be used for the calculation of the actual length of the line segment. The calculation result using this method is more accurate.

The parameters of the **Homography** conversion method are as follows:

Calibration Circle Center Distance

Default value: 1.0000

Instruction: This parameter specifies the actual distance between two circle centers on the calibration board. Please set the value according to the actual situation. The distances between centers of adjacent circles in a row or a column on the calibration board should be the same.

Number of Circle Rows

Default value: 7

Instruction: This parameter specifies the number of rows of circles on the calibration board. Please set the value according to the actual situation.

Number of Circle Columns

Default value: 7

Instruction: This parameter specifies the number of columns of circles on the calibration board. Please set the value according to the actual situation.

Calculate by Calibration

Default Value: Unselected.

Instruction: This parameter determines whether to calculate the actual length by calibration. If this option is selected, the Step will calculate the projection of the image plane onto the actual plane by calibration and then calculate the actual length of the line segment.

Image No.

Description: When there are multiple input images, you can specify the image in which the line segment will be measured.

Default value: 1

Instruction: Once the image No. is selected, you can check the visualized output of the corresponding image in the Debug Output window.

Calibration Board Plane Height

Description: This parameter specifies the distance between the plane of the calibration board and the base plane.

Default value: 0.0000

Instruction: If the calibration board is placed on a table, and the thickness of the calibration board is about 2 cm, the Calibration Board Plane Height should be set to 2 cm.

Measured Plane Height

Description: This parameter specifies the distance between the base plane and the plane where the line segment to be measured is located.

Default value: 0.0000

Instruction: If the object to be measured is placed on a table, and the plane of the object to be measured is 2 cm away from the plane of the table, the Measured Plane Height should be set to 2 cm.

4.3.95. Convert 2D Points to 3D Points

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step converts pixels in 2D images to 3D point clouds.

Usage Scenario

This Step is usually used to match 2D shapes after converting 2D edges to 3D point clouds for matching.

Input and Output

- Input:
  1. Images input to this port will be converted to point clouds.
  2. Poses input to this port will be used to convert the images to point clouds.
- Output:
  1. 3D point clouds obtained by transforming the input 2D images by the input poses

4.3.96. Convert Circles To 2D Poses

Function Description

This Step converts circle shape info data to 2D pose data.

Usage Scenario

This Step is a general data conversion Step. This Step has no fixed usage scenarios.

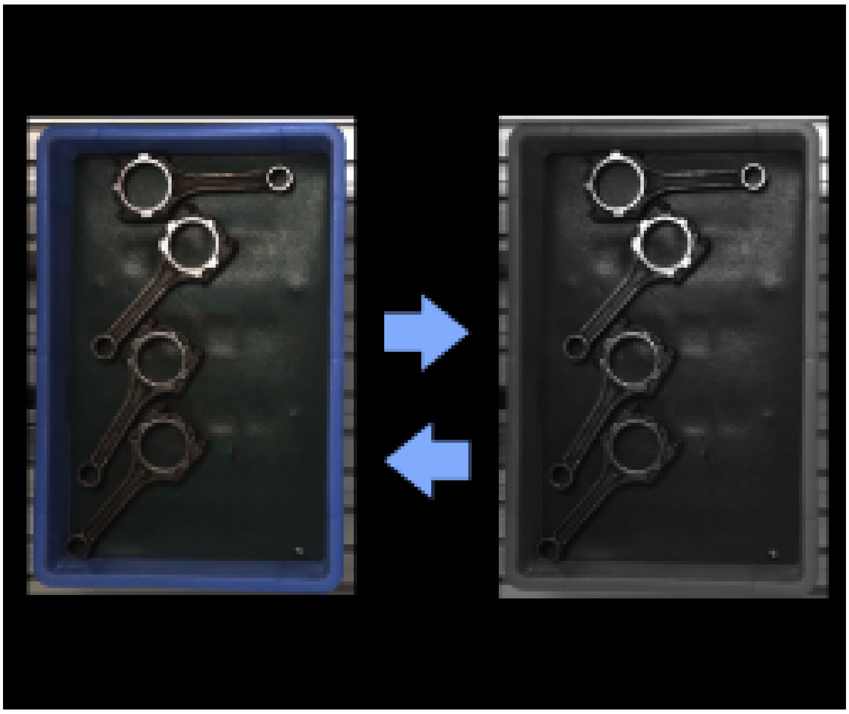
Input and Output

- **Input**: Data items in the format of circle shape info (radius, theta, X, and Y) input to this port will be converted to 2D pose data items (theta, X, and Y). The theta values of circles are normally 0.
- **Output**: 2D pose data items converted from the input data, keeping the X, Y, and theta values, discarding the radius values.

4.3.97. Convert Color Space

Function

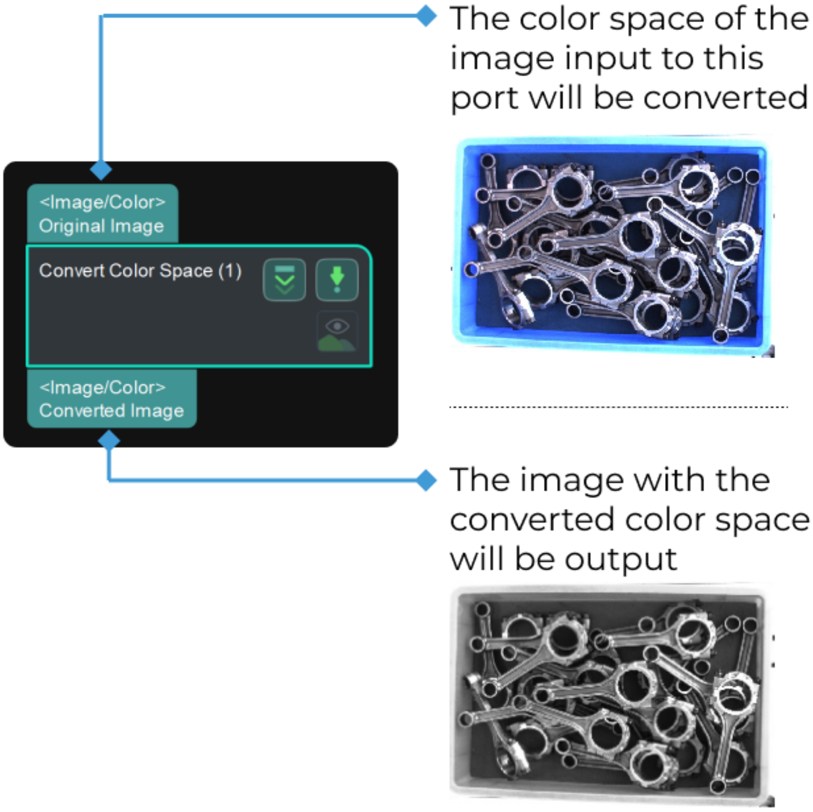
This Step is used to convert the color space of the image, such as converting RGB color image to grayscale and converting BGR color image to grayscale.



Usage Scenario

This Step is widely used to convert the color space of 2D images. No fixed usage scenarios.

Input and Output


<Image/Color>
Original Image
Convert Color Space (1)
<Image/Color>
Converted Image
The color space of the image input to this port will be converted
The image with the converted color space will be output

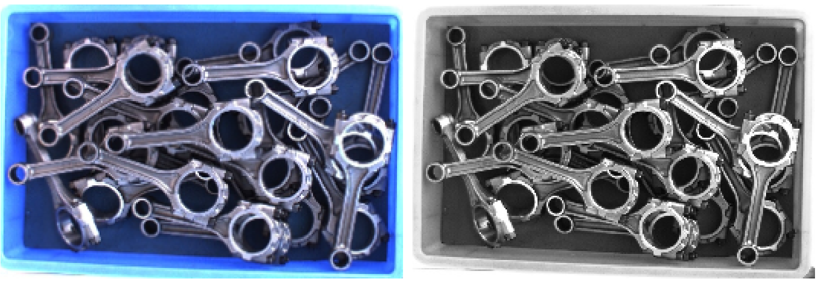
Parameter Tuning

Conversion Mode

BGR to Grayscale

Description: Convert the BGR color image to grayscale image.

Example: The conversion result is shown below. The figure on the left is the BGR color image and the figure on the right is the converted grayscale image.



RGB to Grayscale

Description: Convert the RGB color image to grayscale image. Once this option is selected, a conversion ratio needs to be set. Usually, Psychological formula can be applied. The descriptions of the conversion methods are as follows.

Conversion Method:

- Psychological formula: Use the classical psychological formula `Gray = 0.299*R + 0.587*G + 0.114*B` to get the weighted averages of the RGB channels, which conforms to the human eyes' perception sensitivity to colors.
- Average: Use the average value of R, G, and B channels for the output grayscale. The conversion formula is `Gray = (R + G + B)/3`.
- Max: For each pixel of the grayscale image, take the maximum value among the three pixels corresponding to the three channels R, G, and B.
- Min: For each pixel of the grayscale image, take the minimum value among the three pixels corresponding to the three channels R, G, and B.
- R Channel Values: Take the R channel values as the pixel values of the output grayscale image and ignore the information in other channels.
- G Channel Values: Take the G channel values as the pixel values of the output grayscale image and ignore the information in other channels.
- B Channel Values: Take the B channel values as the pixel values of the output grayscale image and ignore the information in other channels.
- Custom Weights: Customize the values' weight for the RGB channels.

Example: Select R Channel Values, G Channel Values, and B Channel Values as the conversion method. The figures below from left to right shows the conversion results when R Channel Values, G Channel Values, and B Channel Values are selected as the conversion method, respectively.

Grayscale to BGR

Description: Convert the grayscale image to BGR color image.

Example: The conversion result is shown below. Example: The conversion result is shown below. The figure on the left is the grayscale image, and the figure on the right is the converted BGR color image.

> During the conversion from the grayscale image to a BGR color image, the grayscale value is assigned to B, G, and R channels. Since the assigned values of the three channels are the same, the converted image looks like black and white. However, it is a BGR color image with three channels in nature.

Grayscale to RGB

Description: Convert the grayscale image to RGB color image.

Example: The conversion result is shown below. The figure on the left is the grayscale image, and the figure on the right is the converted RGB color image.

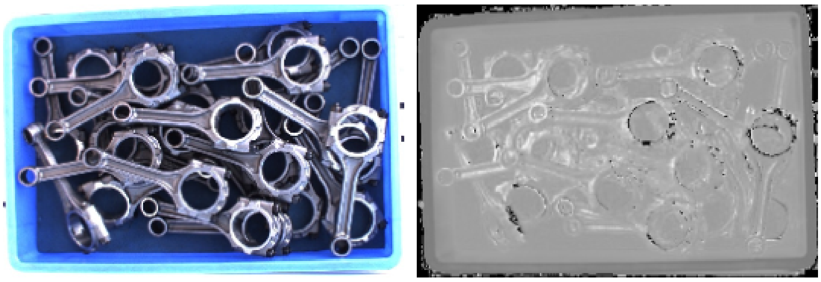
Heatmap to BGR

Description: Convert the heat map to BGR color image.

RGB to HSI

Description: Convert the RGB color image to HSI. Once this option is selected, you will need to select an output channel type.

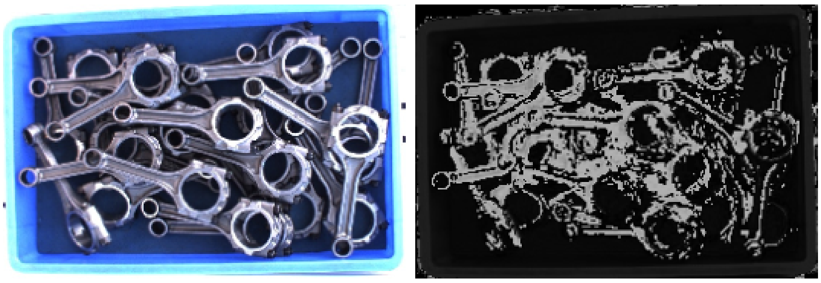
Example: The conversion result of output in the first channel is shown below. The figure on the left is the RGB color image, and the figure on the right is the converted HSI image.



RGB to HSV

Description: Convert the RGB color image to HSV. Once this option is selected, you will need to select an output channel type.

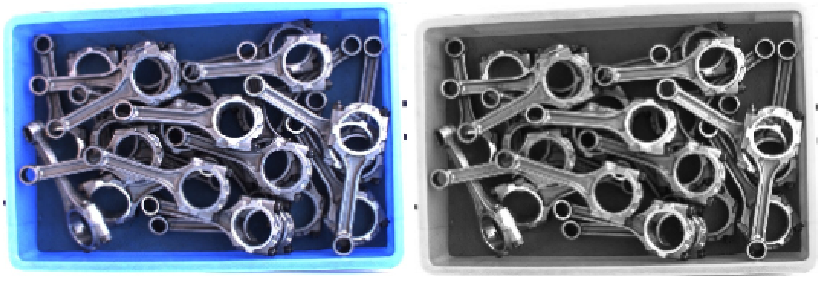
Example: The conversion result of output in the first channel is shown below. The figure on the left is the RGB color image, and the figure on the right is the converted HSV image.



RGB to YUV

Description: Convert the RGB color image to YUV. Once this option is selected, you will need to select an output channel type.

Example: The conversion result of output in the first channel is shown below. The figure on the left is the RGB color image, and the figure on the right is the converted YUV image.



4.3.98. Copy Images

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step renames the read images and save them at the specified paths.

Usage Scenario

This Step is a custom Step for special scenarios.

Input and Output

- **Input**: none
- **Output**: string used to represent the status of the replication process

4.3.99. Count Elements in Specified Dimension in Data List

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

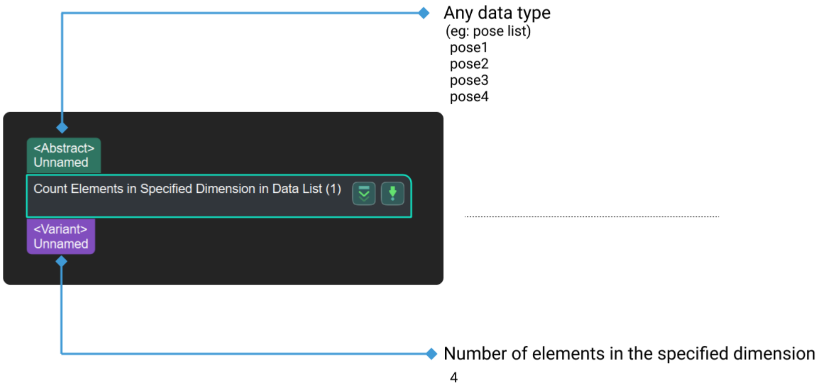
Function

Count the number of elements in a specified dimension of the input data.

Usage Scenario

A general element count Step, which can be used with the Step Repeat Data and Concatenate Copies to make the data lengths of the input ports of the subsequent Step equal.

Input and Output

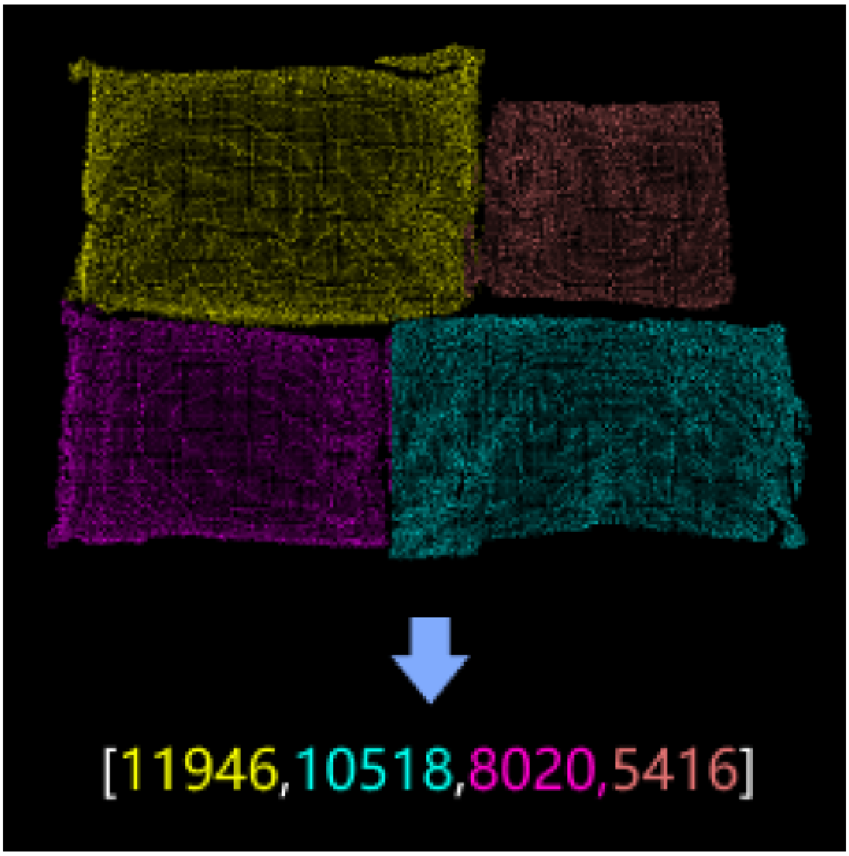

Any data type
(eg: pose list)
pose1
pose2
pose3
pose4
<Abstract>
Unnamed
Count Elements in Specified Dimension in Data List (1)
<Variant>
Unnamed
Number of elements in the specified dimension
4

4.3.100. Count 3D Points

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

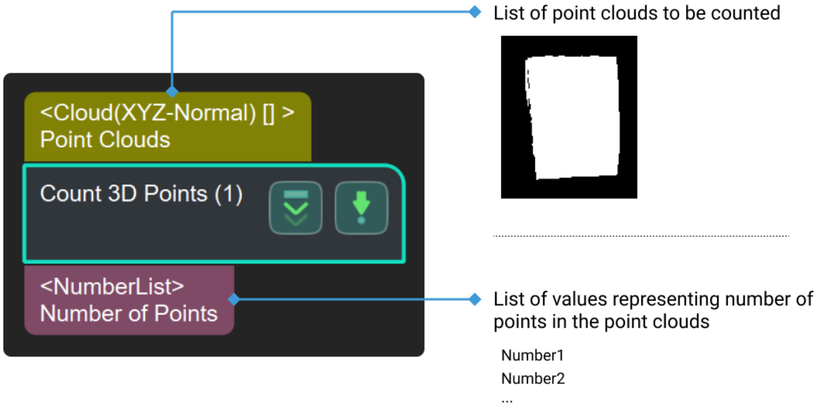
This Step counts the points in the input point clouds.



Usage Scenario

This Step is used in scenarios where multiple point clouds need to be sorted or classified. Usually used with Steps under Sort or the Step Dichotomize Values by Threshold.

Input and Output



4.3.101. Count Color Info

Function

After a color or gray image is input and a color space is selected, this Step will calculate the mean value, standard deviation, maximum, and minimum of the pixel values in the specified channel.

Usage Scenario

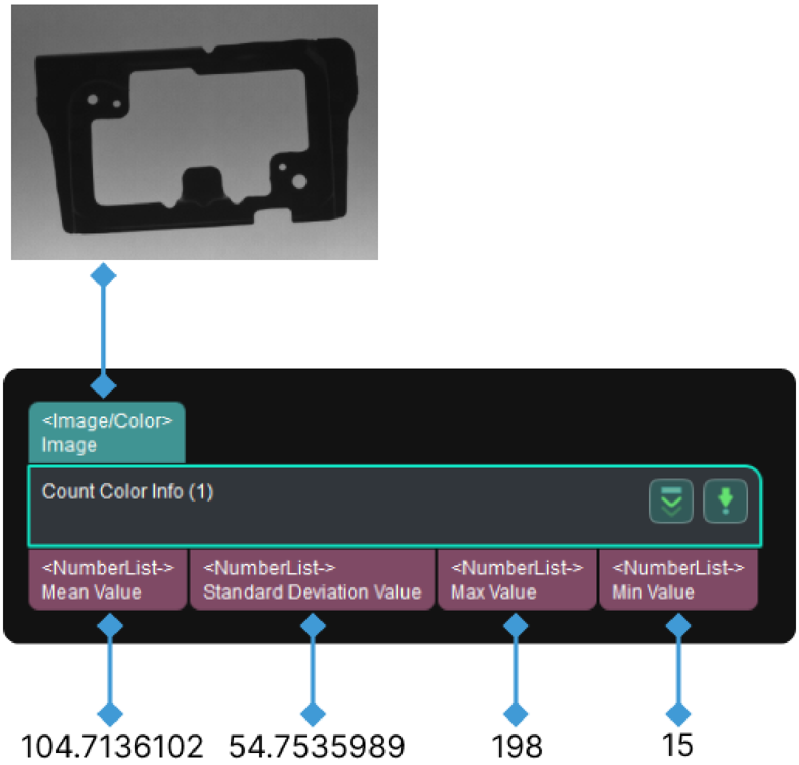
This Step is generally used to evaluate the image color or grayscale in measurement scenarios.

Input and Output

If the pixel values of the first channel in the RGB color space is input, the output data is as follows:

- The mean value of the pixel values of the first channel in the RGB color space
- The standard deviation of the pixel values of the first channel in the RGB color space
- The maximum of the pixel values of the first channel in the RGB color space
- The minimum of the pixel values of the first channel in the RGB color space

The input and output of this Step are shown as below.



Parameter Description

Color Space

Parameter description: This parameter specifies the color space of the input image.

Value list: RGB, HSV, HSI, and Gray.

- RGB (red, green, blue): A commonly used color space. This type of color space is based on red, green, and blue (primary colors), and the three primary colors are added together in various ways to reproduce a broad array of colors.
- HSV (hue, saturation, value): A color space that aligns with the way human vision perceives color-making attributes. The three letters stand for hue, saturation, and value.

- HSI (hue, saturation, intensity): A color space that aligns with the way people describe and explain color. The three letters stand for hue, saturation, and intensity.
- Gray: A most basic color space. When the image only contains luminance information without color information, the gray color space can be used.

Default value: RGB

Channel

Parameter description: This parameter is used to select in which channel of the color space the color information will be calculated.

Value list: First, Second, and Third.

Default value: First.

Notes: The sequence of the channels corresponds to the sequence of the letters in the color space name. For example, the RGB color space has three channels, i.e., red, green, and blue, which correspond to the first, second, and third channels, respectively.

> When the **Color Space** is set to **Gray**, there is only one channel, and therefore you do not need to set the **Channel** parameter.

4.3.102. Create Object Model

Function

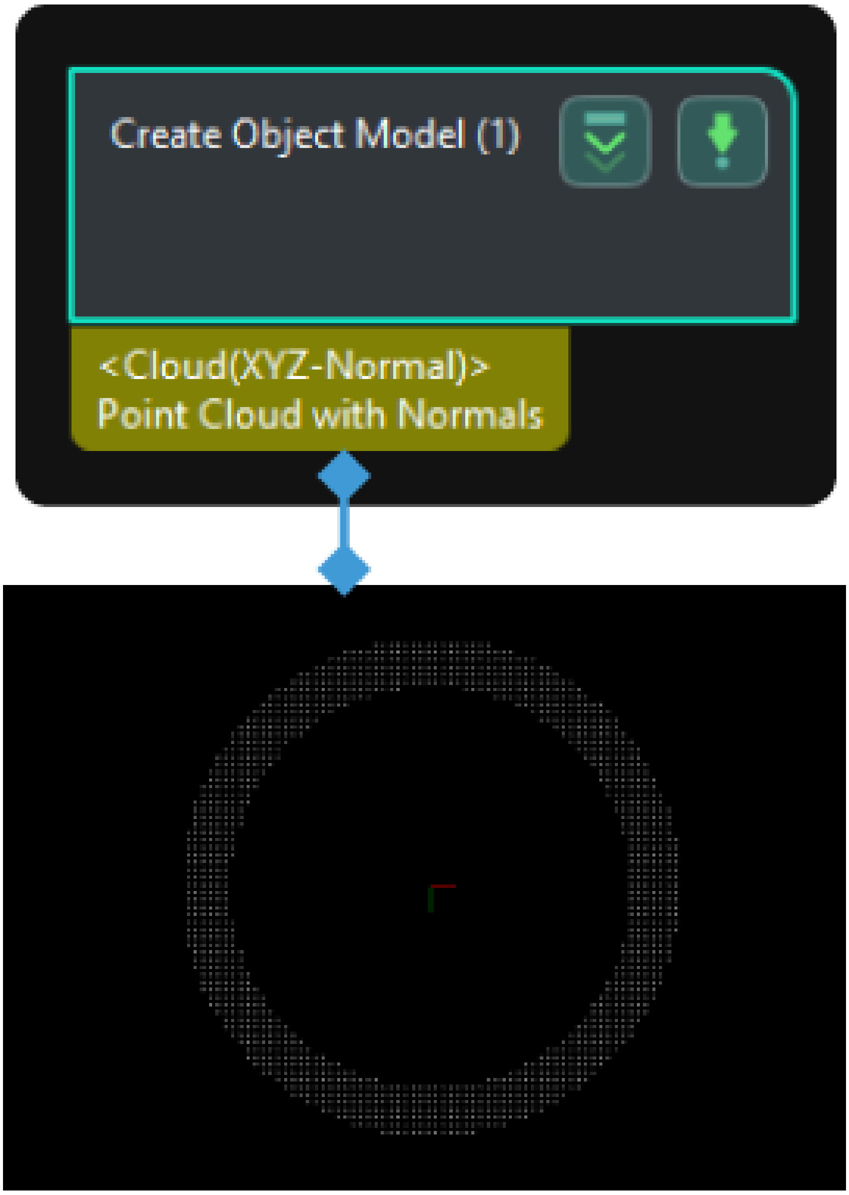
This Step is used to automatically generate point clouds of different shapes according to the specified “Model Type”.

Usage Scenario

This Step is used to automatically generate point clouds of specified shapes that can be used as point cloud models.

Input and Output

Taking the creation of the point cloud of a single ring as an example, the output of this Step is shown in the following image.



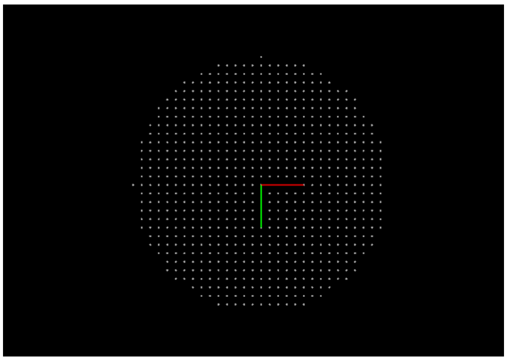
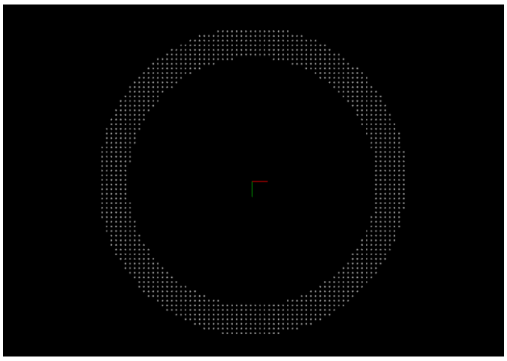
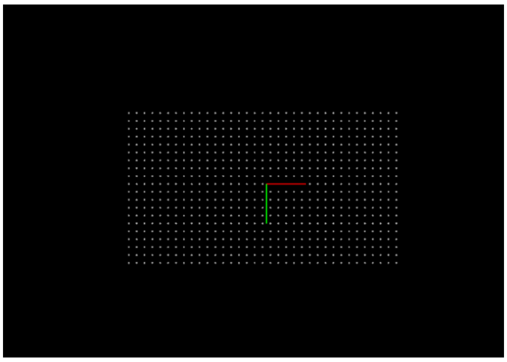
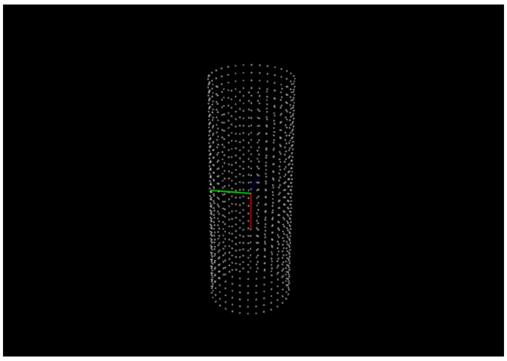
Parameter Description

General Parameters

Object Type

Description: This parameter is used to select the type of object for model creation.

Value list: Circle, Single ring, Rectangle, Cylinder curved surface. The images of various object types are presented in the table below.

Object Type	Image
Circle	
Single ring	
Rectangle	
Cylinder curved surface	

Model Type

Description: This parameter is used to select the type of model to be generated.

Value list: Surface model, Edge model

Sampling Interval

Description: This parameter is used to set the sampling interval for downsampling point cloud (in mm). A larger value produces a sparser point cloud, while a smaller value produces a denser point cloud.

Default value: 1.000 mm

Object Type-Specific Parameters

Circle

When the **Object Type** is set to **Circle**, the following parameter need to be configured.

Radius

Description: This parameter is used to set the radius of the circle (in mm).

Default value: 1.000 mm

Single ring

Outer Radius

Description: This parameter is used to set the outer radius of the ring (in mm).

Default value: 1.000 mm

Inner Radius

Description: This parameter is used to set the inner radius of the ring (in mm).

Default value: 1.000 mm

Rectangle

Length

Description: This parameter is used to set the length of the rectangle (in mm).

Default value: 1.000 mm

Width

Description: This parameter is used to set the width of the rectangle (in mm).

Default value: 1.000 mm

Angle

Description: This parameter is used to set the angle between the long side of the rectangle and the X-axis (in °).

Default value: 0°

Cylinder curved surface

Cylinder Height

Description: This parameter is used to set the height of the cylinder (in mm).

Default value: 1.0 mm

Cylinder Radius

Description: This parameter is used to set the radius of the cylinder (in mm).

Default value: 1.000 mm

4.3.103. Crop Image

ℹ Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

This Step crops the input image and outputs the cropped image and the mask corresponding to the ROI settings.

This Step is a general 2D image ROI setting Step. This Step has no fixed usage scenarios.

Input and Output

- Input:
  1. Image input to this port will be cropped according to the 2D ROI settings.
- Output:
  1. Image within 2D ROI
  2. Mask for 2D ROI

4.3.104. Decompose Object Dimensions

ℹ Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

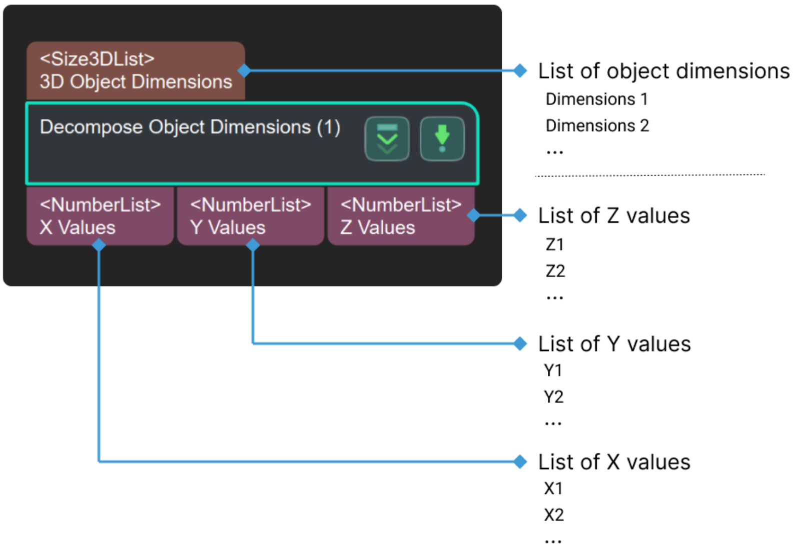
Function

Split the lengths, widths, and heights (X, Y, Z) of the objects in the input list into three separate lists.

Usage Scenario

For facilitating subsequent separate processing of object lengths, widths, and heights after obtaining the values.

Input and Output



4.3.105. Decompose Poses to Quaternions and Translations

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

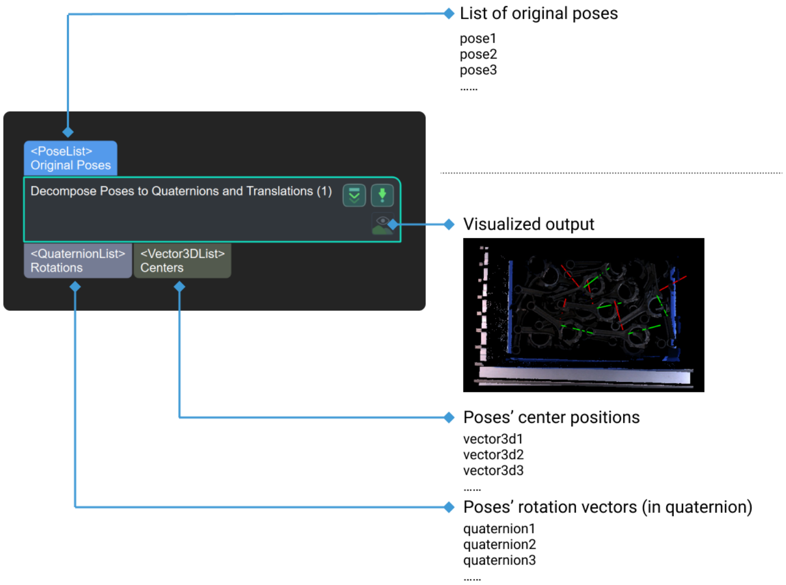
Function

Decompose Poses to Quaternions and Translations.

Usage Scenario

A general pose decomposition Step that provides data sources for subsequent Steps that require **rotation vector** and **pose center** data types.

Input and Output



4.3.106. Decompose Quaternions to X- Y- Z- Axes

Function

This Step decomposes rotation vectors in quaternions into three vectors: X-axis vector, Y-axis vector, Z-axis vector.

Usage Scenario

This Step is a general pose decomposition Step. The decomposed parts can be used as input for other Steps.

Input and Output

- Input:
  1. The X, Y, and Z axis vectors corresponding to the quaternions input at this port will be individually output.
- Output:
  1. X-axis vectors of the input vectors in quaternions
  2. Y-axis vectors of the input vectors in quaternions
  3. Z-axis vectors of the input vectors in quaternions

4.3.107. Decompose Vector3D to Numbers

Function

This Step decomposes each input vector of type Vector3D into three values of type List of Values.

Usage Scenario

This Step is a general data type conversion and decomposition Step. This Step has no fixed usage scenarios.

Input and Output

- Input:
  1. Vectors of the Vector3D data type input to this port will be decomposed.
- Output:
  1. X values of the input vectors
  2. Y values of the input vectors
  3. Z values of the input vectors

4.3.108. Deep Learning Model Package Inference

From Mech-Vision 1.7.2, the Deep Learning Model Package CPU Inference and Deep Learning Model Package Inference (Mech-DLK 2.2.0+) Steps are merged into the Deep Learning Model Package Inference Step, which supports two deep learning model files exported from Mech-DLK with the suffixes being .dlkpackC and .dlkpack.

Open the previous project with Mech-Vision 1.7.2, you can find the Deep Learning Model

> Package CPU Inference and Deep Learning Model Package Inference (Mech-DLK 2.2.0+) Steps are automatically replaced with the Deep Learning Model Package Inference Step.

Function

This Step performs inference with single model packages and cascaded model packages exported from Mech-DLK and outputs the inference result.

Currently, this Step performs model package inference in seven scenarios: classification, object detection, defect segmentation, instance segmentation, text detection, text recognition, and unsupervised segmentation.

> This Step only supports model packages exported from Mech-DLK 2.2.0 or above.

> From Mech-DLK 2.4.1, model packages can be divided into single model packages and cascaded model packages.
>
> - Single model package: There is only one deep learning model in the model package, such as an "Instance Segmentation" model.
> - Cascaded model package: Multiple models are cascaded in the model package, and the output result of the previous model is input to the next model. For example, there are two models, i.e., Object Detection and Instance Segmentation, in the model package, and the inference sequence is Object Detection › Instance Segmentation. The output of the Object Detection model is input to the Instance Segmentation model.
>
> When this Step performs inference with cascaded model packages, the Deep Learning Result Parser Step can parse the exported result.

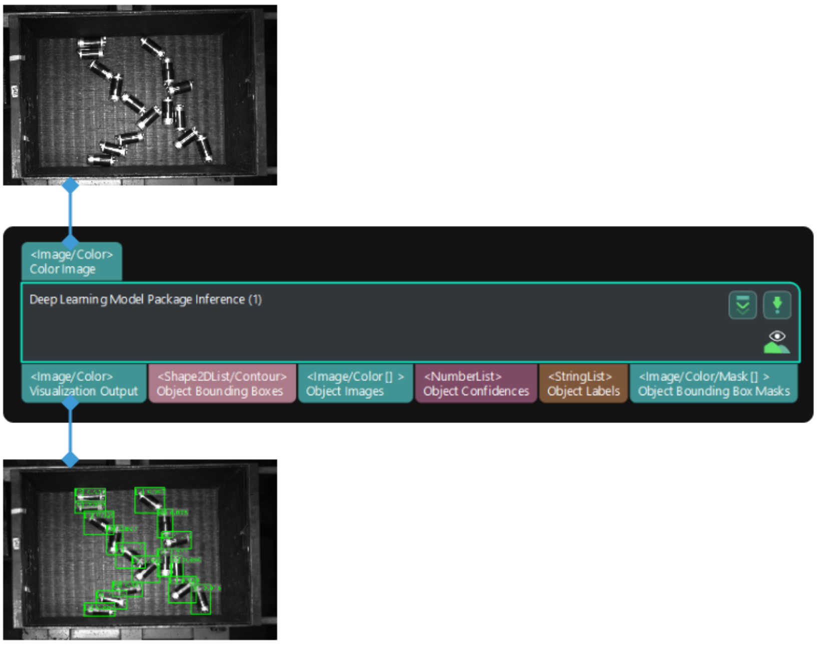
Usage Scenario

This Step is usually used in classification, object detection, and defect segmentation scenarios. For a description of the compatibility of the Step, please refer to Compatibilities of Deep Learning Steps.

Input and Output

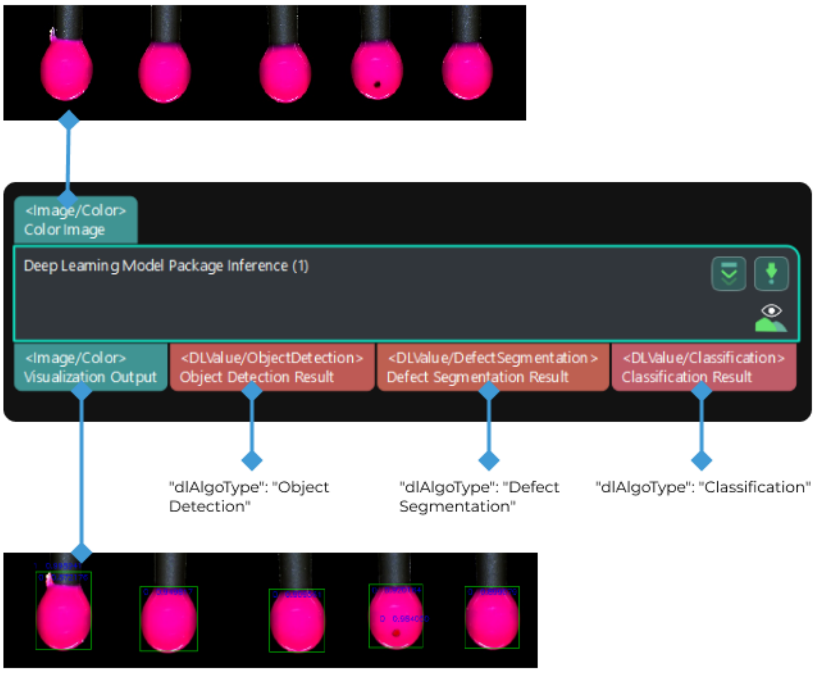
Single Model Package

When a single model package is imported, the input and output of this Step are shown below in the case of the Object Detection module.



Cascaded Model Package

When a cascaded model package is imported, the input and output of this Step are shown below in the case of the cascaded modules of Object Detection + Defect Segmentation + Classification.


<Image/Color>
Color Image
Deep Learning Model Package Inference (1)
<Image/Color>
Visualization Output
<DLValue/ObjectDetection>
Object Detection Result
<DLValue/DefectSegmentation>
Defect Segmentation Result
<DLValue/Classification>
Classification Result
"dlAlgoType": "Object Detection"
"dlAlgoType": "Defect Segmentation"
"dlAlgoType": "Classification"

System Requirements

The following system requirements need to be met when using this Step.

- **CPU**: needs to support the AVX2 instruction set and meets any of the following conditions:

- IPC or PC without any discrete graphics card: Intel i5-12400 or higher.
- IPC or PC with a discrete graphics card: Intel i7-6700 or higher, with the graphics card not lower than GeForce GTX 1660.

This Step has been thoroughly tested on Intel CPUs but has not been tested on AMD CPUs yet. Therefore, Intel CPUs are recommended.

- **GPU**: GeForce GTX 1660 or above (if with a discrete graphics card).

Parameter Description

You can click the following links to check the corresponding parameter descriptions of each function.

- Classification
- Defect Segmentation
- Instance Segmentation
- Object Detection
- Text Detection
- Text Recognition
- Unsupervised Segmentation

When using this Step for inference with the cascaded model package, you can adjust the parameters in the next Step Deep Learning Result Parser.

4.3.108.1. Classification

The following parameters need to be adjusted when the classification model package is imported into this Step.

Model Package Settings

Model Package Management Tool

Parameter description: This parameter is used to open the deep learning model package management tool and import the deep learning model package. The model package file is a ".dlkpack" or ".dlkpackC" file exported from Mech-DLK.

Tuning instruction: Please refer to Deep Learning Model Package Management Tool for the usage.

Model Name

Parameter description: This parameter is used to select the model package that has been imported for this Step.

Tuning instruction: Once you have imported the deep learning model package, you can select the corresponding model name in the drop-down list.

Model Package Type

Parameter description: Once a **Model Name** is selected, the **Model Package Type** will be filled automatically, such as Object Detection (single model package) and Object Detection + Defect Segmentation + Classification (cascaded model package).

GPU ID

Parameter description: This parameter is used to select the device ID of the GPU that will be used for the inference.

Tuning instruction: Once you have selected the model name, you can select the GPU ID in the drop-down list of this parameter.

Inference Configuration

Parameter description: This parameter is used to configure parameters related to classification model package inference. You can click **[Open the editor]** to open the inference configuration window. The parameters and their description included in this window are shown in the following table.

Parameter	Parameter Description	Tuning Instruction
Confidence Threshold	This parameter is used to set the confidence threshold in the process of classification. The results above this threshold will be kept.	Please set the parameter according to your actual needs.
Show Class Activation Map ("Show All Parameters" Enabled)	This parameter is used to display the class activation map for identifying the image regions that are most relevant to the classification. Blue indicates that the region contributes the least to the classification while red indicates that the region contributes the most to the classification.	In Mech-Vision 1.7.2 and above, when **Show Class Activation Map** is selected, the model package inference is slow.

ROI settings

ROI Path

Parameter description: This parameter is used to set or modify the ROI.

Tuning instructions:

1. Once the deep learning model is imported, a default ROI will be applied. If you need to edit the ROI, click **[Open the editor]**.
2. Edit the ROI in the pop-up Set ROI window, and fill in the ROI name.

If you would like to use the default ROI again, please delete the ROI file name below the **[Open the editor]** button.

Visualization Settings

Manually Font Scale

Parameter description: This parameter determines whether to customize the font scale in the visualized output result. Once this option is selected, you should set the **Font Scale (0–10)**.

Default value: Unselected.

Tuning recommendation: Please set this parameter according to your actual needs.

Font Scale (0–10)

Parameter description: This parameter is used to set the font scale in the visualized output result.

Default value: 3.0

Tuning recommendation: Please set this parameter according to your actual needs.

Show All Results

Parameter description: This parameter is used to visualize all inference results of the cascaded model package. It can only be set when the Deep Learning Model Package Inference Step is used for cascaded model package inference.

Tuning recommendation: Please set this parameter according to your actual needs.

4.3.108.2. Defect Segmentation

The following parameters need to be adjusted when the defect segmentation model package is imported into this Step.

Model Package Settings

Model Package Management Tool

Parameter description: This parameter is used to open the deep learning model package management tool and import the deep learning model package. The model package file is a ".dlkpack" or ".dlkpackC" file exported from Mech-DLK.

Tuning instruction: Please refer to Deep Learning Model Package Management Tool for the usage.

Model Name

Parameter description: This parameter is used to select the model package that has been imported for this Step.

Tuning instruction: Once you have imported the deep learning model by using the deep learning model package management tool, you can select the corresponding model name in the drop-down list.

Model Package Type

Parameter description: Once a **Model Name** is selected, the **Model Package Type** will be filled automatically.

GPU ID

Parameter description: This parameter is used to select the device ID of the GPU that will be used for the inference.

Tuning instruction: Once you have selected the model name, you can select the GPU ID in the drop-down list of this parameter.

Inference Configuration

Parameter description: This parameter is used to configure parameters related to defect segmentation model package inference. You can click **[Open the editor]** to open the inference configuration window.

Tuning instruction: Please refer to Defect Segmentation Determination Rule for the configuration instructions.

ROI Settings

ROI Path

Parameter description: This parameter is used to set or modify the ROI.

Tuning instructions:

1. Once the deep learning model is imported, a default ROI will be applied. If you need to edit the ROI, click **[ROI Path]**.
2. Edit the ROI in the pop-up Set ROI window, and fill in the ROI name.

> If you would like to use the default ROI again, please delete the ROI file name below the **[Open the editor]** button.

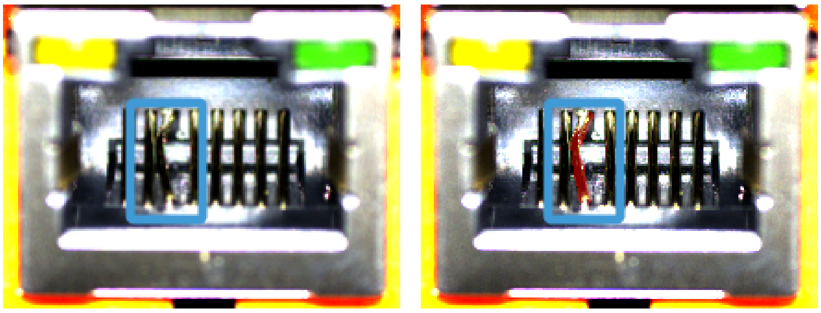
Visualization Settings

Draw Defect Mask on Image

Parameter description: This parameter is used to determine whether to draw the defect mask on the image. If selected, a defect mask will be drawn on the input image.

Default value: Unselected.

Instruction: Select to draw a mask on the input image. The figure below shows the result before and after selecting this option.



> - From Mech-Vision 1.7.2, the Defect Judgement Rules Settings, Quantity Threshold Settings, and Area Range Settings parameters are removed from the Deep Learning Model Package Inference Step. If you want to adjust these parameters, please configure them in Mech-DLK.
> - In Mech-Vision 1.7.2, when the Deep Learning Model Package Inference Step is used for inference with model packages that are exported from Mech-DLK 2.2.0 or previous versions with Defect Judgement Rules Settings configured, the old defect judgement rules will not take effect. Please configure the defect determination rules of the package model in Mech-DLK 2.4.1 or above and re-export the model package. Then, you can use the model package for inference in the Deep Learning Model Package Inference Step.

Show All Results

Parameter description: This parameter is used to visualize all inference results of the cascaded model package. It can only be set when the Deep Learning Model Package Inference Step is used for cascaded model package inference.

Tuning recommendation: Please set this parameter according to your actual needs.

4.3.108.3. Instance Segmentation

The following parameters need to be adjusted when the instance segmentation model package is imported into this Step.

Model Package Settings

Model Package Management Tool

Parameter description: This parameter is used to open the deep learning model package management tool and import the deep learning model package. The model package file is a ".dlkpack" or ".dlkpackC" file exported from Mech-DLK.

Tuning instruction: Please refer to Deep Learning Model Package Management Tool for the usage.

Model Name

Parameter description: This parameter is used to select the model package that has been imported for this Step.

Tuning instruction: Once you have imported the deep learning model package, you can select the corresponding model name in the drop-down list.

Model Package Type

Parameter description: Once a **Model Name** is selected, the **Model Package Type** will be filled automatically, such as Object Detection (single model package) and Object Detection + Defect Segmentation + Classification (cascaded model package).

GPU ID

Parameter description: This parameter is used to select the device ID of the GPU that will be used for the inference.

Tuning instruction: Once you have selected the model name, you can select the GPU ID in the drop-down list of this parameter.

Inference Configuration

Parameter description: This parameter is used to configure parameters related to instance segmentation model package inference. You can click **[Open the editor]** to open the inference configuration window. The parameter and its description included in this window are shown in the following table.

Parameter	Parameter Description	Tuning Instruction
Confidence Threshold	This parameter is used to set the confidence threshold in the process of instance segmentation. The results above this threshold will be kept.	please set the parameter according to your actual needs.

Instance Segmentation Confidence Threshold (0.0–1.0)

Parameter description: This parameter is used to set the confidence threshold for instance segmentation. The results above this threshold will be displayed in green and the results below this threshold will be displayed in red.

Default value: 0.7000

Tuning recommendation: Please set this parameter according to your actual needs.

ROI settings

ROI Path

Parameter description: This parameter is used to set or modify the ROI.

Tuning instructions:

1. Once the deep learning model is imported, a default ROI will be applied. If you need to edit the ROI, click **[ROI Path]**.
2. Edit the ROI in the pop-up Set ROI window, and fill in the ROI name.

> If you would like to use the default ROI again, please delete the ROI name below the **[Open the editor]** button.

Visualization Settings

Draw Instance on Image

Parameter description: This parameter is used to determine whether to display the segmented mask and bounding box on the image.

Default value: Unselected.

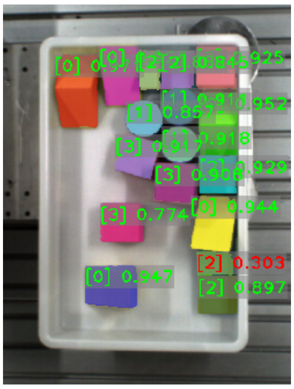
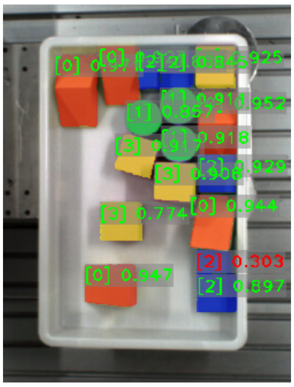
Tuning recommendation: Please set this parameter according to your actual needs.

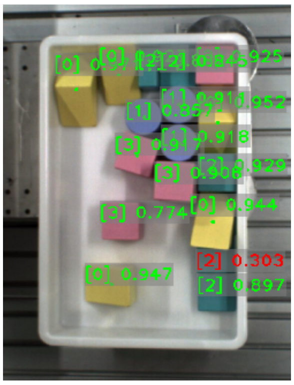
Instance Color Scheme

Parameter description: This parameter is used to specify the instance color scheme in the visualized output result.

Default value: Instances

Value list: Instances, Classes, and CentralPoint

Instance Color Scheme	Description	Image
Instances	Each detected object is displayed in an individual color.	
Classes	Objects with the same label will be displayed in the same color.	

Instance Color Scheme	Description	Image
Central Point	Display the original color of the object.	 [2] 0.303 [2] 0.897 [0] 0.947 [3] 0.774 [0] 0.944

Manually Font Scale

Parameter description: This parameter determines whether to customize the font scale in the visualized output result. Once this option is selected, you should set the **Font Scale (0–10)**.

Default value: Unselected.

Tuning recommendation: Please set this parameter according to your actual needs.

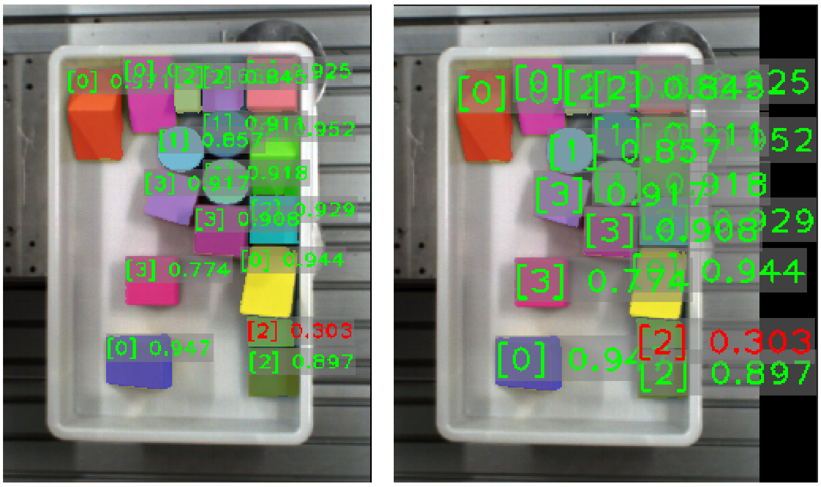
Font Scale (0–10)

Parameter description: This parameter is used to set the font scale in the visualized output result.

Default value: 3.0

Tuning recommendation: Please set this parameter according to your actual needs.

Example: The figure below shows the visualization result when the font scale is set to 3.0 and 5.0 in an instance segmentation project. The font size in the left figure is 3, and 5 in the right figure.



Show All Results

Parameter description: This parameter is used to visualize all inference results of the cascaded model package. It can only be set when the Deep Learning Model Package Inference Step is used for cascaded model package inference.

Tuning recommendation: Please set this parameter according to your actual needs.

4.3.108.4. Object Detection

The following parameters need to be adjusted when the object detection model package is imported into this Step.

Model Package Settings

Model Package Management Tool

Parameter description: This parameter is used to open the deep learning model package management tool and import the deep learning model package. The model package file is a ".dlkpack" or ".dlkpackC" file exported from Mech-DLK.

Tuning instruction: Please refer to Deep Learning Model Package Management Tool for the usage.

Model Name

Parameter description: This parameter is used to select the model package that has been imported for this Step.

Tuning instruction: Once you have imported the deep learning model package, you can select the corresponding model name in the drop-down list.

Model Package Type

Parameter description: Once a **Model Name** is selected, the **Model Package Type** will be filled automatically, such as Object Detection (single model package) and Object Detection + Defect Segmentation + Classification (cascaded model package).

GPU ID

Parameter description: This parameter is used to select the device ID of the GPU that will be used for the inference.

Tuning instruction: Once you have selected the model name, you can select the GPU ID in the drop-down list of this parameter.

Inference Configuration

Parameter description: This parameter is used to configure parameters related to object detection model package inference. You can click **[Open the editor]** to open the inference configuration window. The parameter and its description included in this window are shown in the following table.

Parameter	Parameter Description	Tuning Instruction
Confidence Threshold	This parameter is used to set the confidence threshold in the process of object detection. The results above this threshold will be kept.	Please set the parameter according to your actual needs.

Object Detection Confidence Threshold (0.0–1.0)

Default value: 0.7000

Tuning instruction: The results above this threshold will be kept.

ROI settings

ROI Path

Parameter description: This parameter is used to set or modify the ROI.

Tuning instructions:

1. Once the deep learning model is imported, a default ROI will be applied. If you need to edit the ROI, click **[ROI Path]**.
2. Edit the ROI in the pop-up Set ROI window, and fill in the ROI name.

> If you would like to use the default ROI again, please delete the ROI name below the **[Open the editor]** button.

Visualization Settings

Show Obj Bounding Box

Parameter description: This parameter is used to determine whether to display the mask and bounding box on the image.

Default value: Unselected.

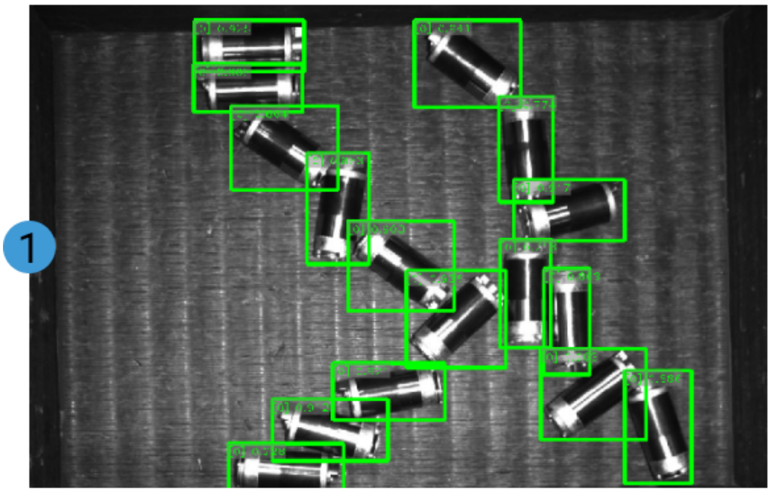
Tuning recommendation: Please set this parameter according to your actual needs.

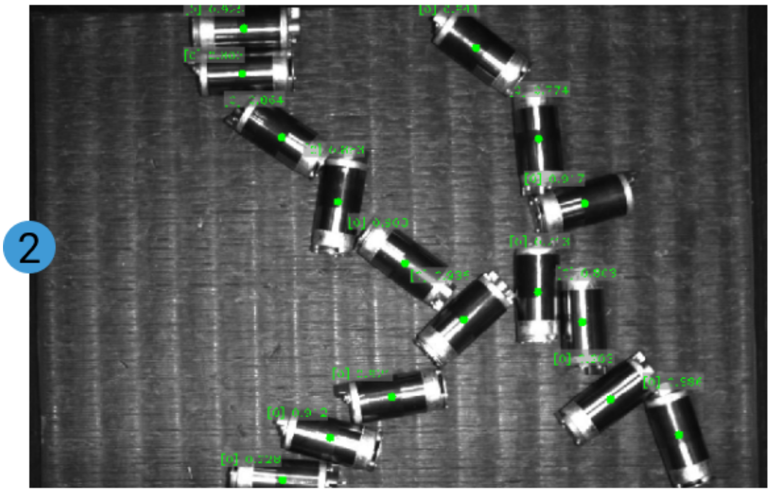
Obj Bounding Box Mode

Default value: CentralPoint

Value list: BoundingBox and CentralPoint

- Bounding Box: Display the results with bounding boxes. You can see it in the following figure 1.
- Center Point: Display the results with center points. You can see it in the following figure 2.





Tuning recommendation: Please set this parameter according to your actual needs.

Manually Font Scale

Parameter description: This parameter determines whether to customize the font scale in the visualized output result. Once this option is selected, you should set the **Font Scale (0–10)**.

Default value: Unselected.

Tuning recommendation: Please set this parameter according to your actual needs.

Font Scale (0–10)

Parameter description: This parameter is used to set the font size in the visualized output result.

Default value: 3.0

Tuning recommendation: Please set this parameter according to your actual needs.

Show All Results

Parameter description: This parameter is used to visualize all inference results of the cascaded model package. It can only be set when the Deep Learning Model Package Inference Step is used for cascaded model package inference.

Tuning recommendation: Please set this parameter according to your actual needs.

4.3.108.5. Text Detection

The following parameters need to be adjusted when the text detection model package is imported into this Step.

Model Package Settings

Model Package Management Tool

Parameter description: This parameter is used to open the deep learning model package management tool and import the deep learning model package. The model package file is a ".dlkpack" or ".dlkpackC" file exported from Mech-DLK.

Tuning instruction: Please refer to Deep Learning Model Package Management Tool for the usage.

Model Name

Parameter description: This parameter is used to select the model package that has been imported for this Step.

Tuning instruction: Once you have imported the deep learning model package, you can select the corresponding model name in the drop-down list.

Model Package Type

Parameter description: Once a **Model Name** is selected, the **Model Package Type** will be filled automatically, such as Object Detection (single model package) and Object Detection + Defect Segmentation + Classification (cascaded model package).

GPU ID

Parameter description: This parameter is used to select the device ID of the GPU that will be used for the inference.

Tuning instruction: Once you have selected the model name, you can select the GPU ID in the drop-down list of this parameter.

Inference Configuration

Parameter description: This parameter is used to configure parameters related to text detection model package inference. You can click **[Open the editor]** to open the inference configuration window. The parameter and its description included in this window are shown in the following table.

Parameter	Parameter Description	Tuning Instruction
Confidence Threshold	This parameter is used to set the confidence threshold in the process of text detection. The results above this threshold will be kept.	please set the parameter according to your actual needs.

Text Detection Confidence Threshold (0.0–1.0)

Default value: 0.7000

Tuning instruction: The results above this threshold will be kept.

ROI settings

ROI Path

Parameter description: This parameter is used to set or modify the ROI.

Tuning instructions:

1. Once the deep learning model is imported, a default ROI will be applied. If you need to edit the ROI, click **[Open the editor]**.
2. Edit the ROI in the pop-up Set ROI window, and fill in the ROI name.

> If you would like to use the default ROI again, please delete the ROI name below the **[Open the editor]** button.

Visualization Settings

Detection Result Visualization

Parameter description: This parameter is used to determine whether to display the mask and bounding box on the image.

Default value: Unselected.

Tuning recommendation: Please set this parameter according to your actual needs.

Manually Font Scale

Parameter description: This parameter determines whether to customize the font scale in the visualized output result. Once this option is selected, you should set the **Font Scale (0–10)**.

Default value: Unselected.

Tuning recommendation: Please set this parameter according to your actual needs.

Font Scale (0–10)

Parameter description: This parameter is used to set the font scale in the visualized output result.

Default value: 3.0

Tuning recommendation: Please set this parameter according to your actual needs.

Show All Results

Parameter description: This parameter is used to visualize all inference results of the cascaded model package. It can only be set when the Deep Learning Model Package Inference Step is used for cascaded model package inference.

Tuning recommendation: Please set this parameter according to your actual needs.

4.3.108.6. Text Recognition

The following parameters need to be adjusted when the text recognition model package is imported into this Step.

Model Package Settings

Model Package Management Tool

Parameter description: This parameter is used to open the deep learning model package management tool and import the deep learning model package. The model package file is a ".dlkpack" or ".dlkpackC" file exported from Mech-DLK.

Tuning instruction: Please refer to Deep Learning Model Package Management Tool for the usage.

Model Name

Parameter description: This parameter is used to select the model package that has been imported for this Step.

Tuning instruction: Once you have imported the deep learning model package, you can select the corresponding model name in the drop-down list.

Model Package Type

Parameter description: Once a **Model Name** is selected, the **Model Package Type** will be filled automatically, such as Object Detection (single model package) and Object Detection + Defect Segmentation + Classification (cascaded model package).

GPU ID

Parameter description: This parameter is used to select the device ID of the GPU that will be used for the inference.

Tuning instruction: Once you have selected the model name, you can select the GPU ID in the drop-down list of this parameter.

Inference Configuration

Parameter description: This parameter is used to configure parameters related to text recognition model package inference. You can click **[Open the editor]** to open the inference configuration window. The parameter and its description included in this window are shown in the following table.

Parameter	Parameter Description	Tuning Instruction
Max Text Length Limitation	This parameter is used to set the max text length in the process of text recognition. Once this feature is enabled, the text whose length exceeds the max text length limitation will not be recognized.	please set the parameter according to your actual needs.

ROI settings

ROI Path

Parameter description: This parameter is used to set or modify the ROI.

Tuning instructions:

1. Once the deep learning model is imported, a default ROI will be applied. If you need to edit the ROI, click **[Open the editor]**.
2. Edit the ROI in the pop-up Set ROI window, and fill in the ROI name.

If you would like to use the default ROI again, please delete the ROI name below the **[Open the editor]** button.

Visualization Settings

Manually Font Scale

Parameter description: This parameter determines whether to customize the font size in the visualized output result. Once this option is selected, you should set the **Font Scale (0–10)**.

Default value: Unselected.

Tuning recommendation: Please set this parameter according to your actual needs.

Font Scale (0–10)

Parameter description: This parameter is used to set the font scale in the visualized output result.

Default value: 3.0

Tuning recommendation: Please set this parameter according to your actual needs.

Show All Results

Parameter description: This parameter is used to visualize all inference results of the cascaded model package. It can only be set when the Deep Learning Model Package Inference Step is used for cascaded model package inference.

Tuning recommendation: Please set this parameter according to your actual needs.

4.3.108.7. Unsupervised Segmentation

The following parameters need to be adjusted when the unsupervised segmentation model package is imported into this Step.

Model Package Settings

Model Package Management Tool

Parameter description: This parameter is used to open the deep learning model package management tool and import the deep learning model package. The model package file is a ".dlkpack" or ".dlkpackC" file exported from Mech-DLK.

Tuning instruction: Please refer to Deep Learning Model Package Management Tool for the usage.

Model Name

Parameter description: This parameter is used to select the model package that has been imported for this Step.

Tuning instruction: Once you have imported the deep learning model package, you can select the corresponding model name in the drop-down list.

Model Package Type

Parameter description: Once a **Model Name** is selected, the **Model Package Type** will be filled automatically, such as Object Detection (single model package) and Object Detection + Defect Segmentation + Classification (cascaded model package).

GPU ID

Parameter description: This parameter is used to select the device ID of the GPU that will be used for the inference.

Tuning instruction: Once you have selected the model name, you can select the GPU ID in the drop-down list of this parameter.

Inference Configuration

Parameter description: This parameter is used to configure parameters related to unsupervised segmentation model package inference. You can click **[Open the editor]** to open the inference configuration window. The parameter and its description included in this window are shown in the following table.

Parameter	Parameter Description	Tuning Instruction
OK/NG Confidence Threshold	The red part represents the NG range, and the green part represents the OK range. If the confidence value of the segmentation result is below the OK confidence threshold, the segmentation result is OK. If the confidence value of the segmentation result is above the NG confidence value, the segmentation result is NG.	Please drag the slider bar to adjust the confidence thresholds.

ROI settings

ROI Path

Parameter description: This parameter is used to set or modify the ROI.

Tuning instructions:

1. Once the deep learning model is imported, a default ROI will be applied. If you need to edit the ROI, click **[ROI Path]**.
2. Edit the ROI in the pop-up Set ROI window, and fill in the ROI name.

If you would like to use the default ROI again, please delete the ROI name below the **[Open the editor]** button.

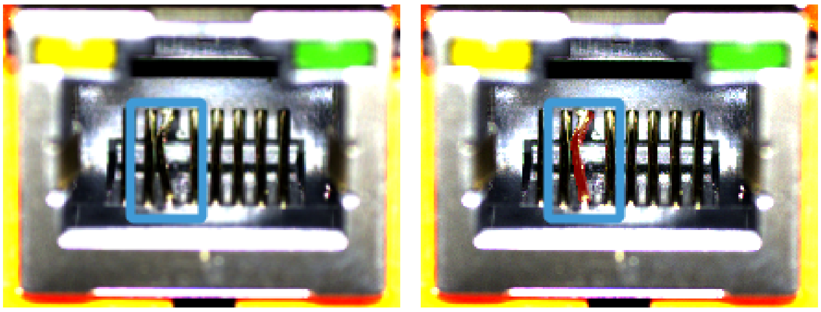
Visualization Settings

Draw Defect Mask on Image

Parameter description: This parameter is used to determine whether to draw the defect mask on the image. If selected, a defect mask will be drawn on the input image.

Default value: Unselected.

Instruction: Select to draw a mask on the input image. The figure below shows the result before and after selecting this option.



Show All Results

Parameter description: This parameter is used to visualize all inference results of the cascaded model package. It can only be set when the Deep Learning Model Package Inference Step is used for cascaded model package inference.

Tuning recommendation: Please set this parameter according to your actual needs.

4.3.109. Deep Learning Result Parser

Function

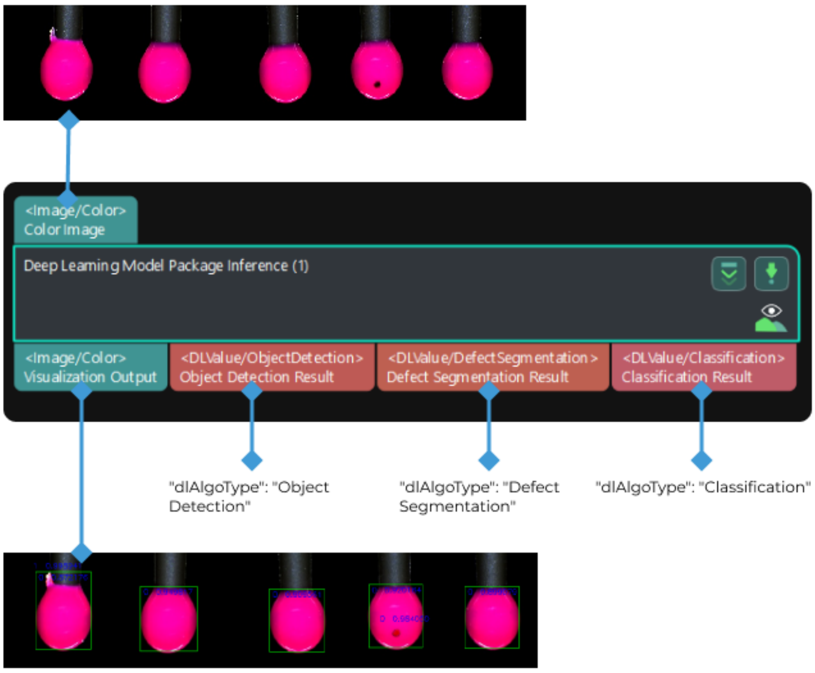
This Step can parse the cascaded model package's inference result exported from the Deep Learning Model Package Inference Step.

Usage Scenario

When the Deep Learning Model Package Inference Step is used for the inference with a cascaded model package, this Step follows the Deep Learning Model Package Inference Step.

When multiple images are input for a simultaneous inference, and you want to check the parsing result of every image, it is recommended to add an Unpack Data Step between the Deep Learning Model Package Inference and “Deep Learning Result Parser Steps.

Input and Output


<Image/Color>
Color Image
Deep Learning Model Package Inference (1)
<Image/Color>
Visualization Output
<DLValue/ObjectDetection>
Object Detection Result
<DLValue/DefectSegmentation>
Defect Segmentation Result
<DLValue/Classification>
Classification Result
"dlAlgoType": "Object Detection"
"dlAlgoType": "Defect Segmentation"
"dlAlgoType": "Classification"

System Requirements

The following system requirements need to be met when using this Step.

- CPU: needs to support the AVX2 instruction set and meets any of the following conditions:
  - IPC or PC without any discrete graphics card: Intel i5-12400 or higher.
  - IPC or PC with a discrete graphics card: Intel i7-6700 or higher, with the graphics card not lower than GTX 1660.

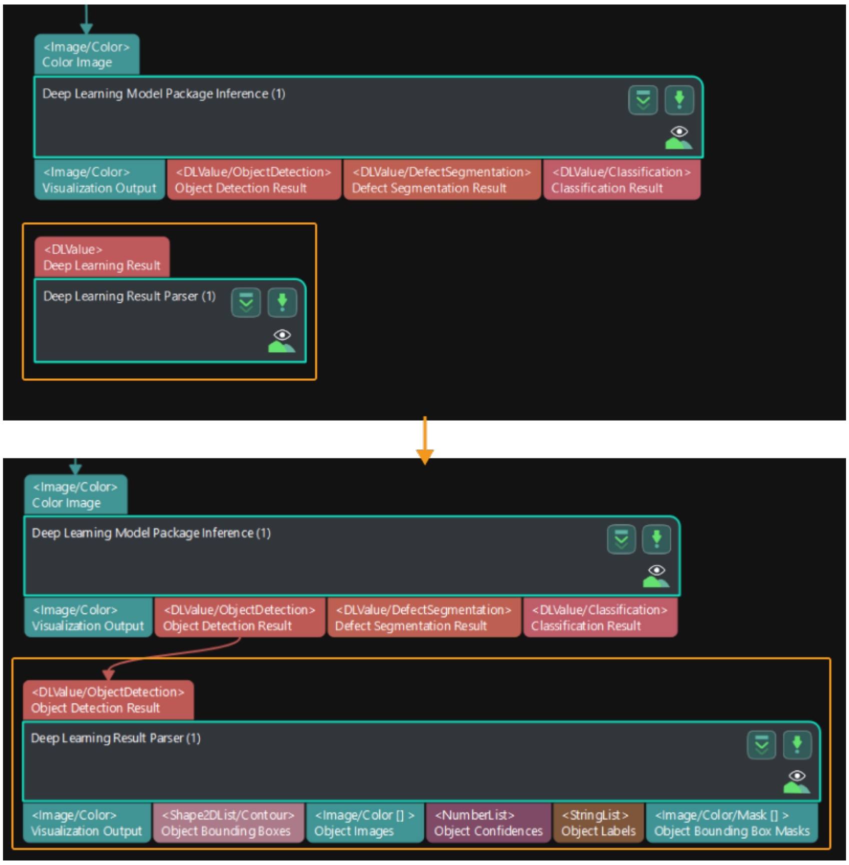
  This Step has been thoroughly tested on Intel CPUs but has not been tested on AMD CPUs yet. Therefore, Intel CPUs are recommended.
- GPU: NVIDIA GTX 1660 or above (if with a discrete graphics card).

Usage Instructions

Directly follows the Deep Learning Model Package Inference Step

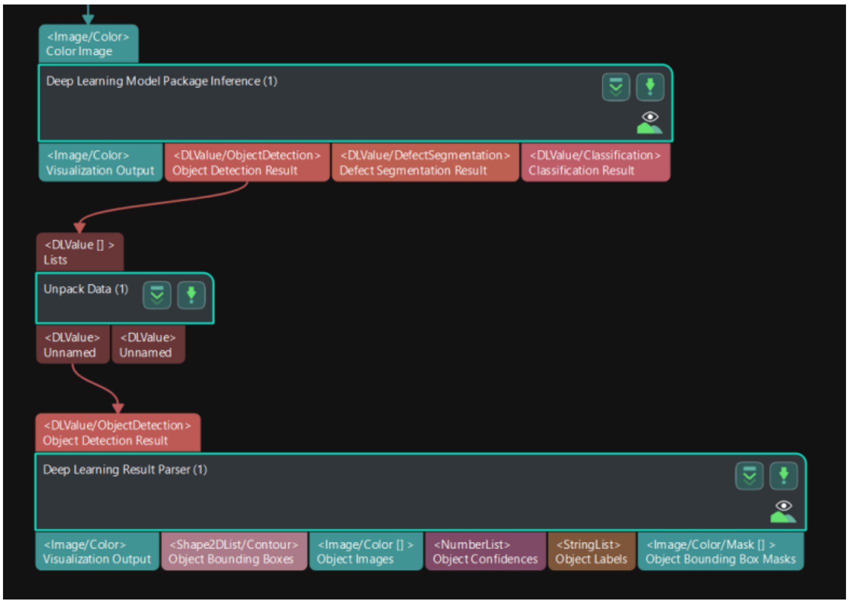
When this Step follows the Deep Learning Model Package Inference Step, this Step can display different parameters according to different Deep Learning Value Type.

For example, if this Step is followed by the Object Detection Result output port of the Deep Learning Model Package Inference Step, then this Step will display the parameters related to the object detection. A parameter description for each scene can be found in Parameter Description.



Use with an Unpack Data Step

The Unpack Data Step is added between the Deep Learning Model Package Inference and Deep Learning Result Parser Steps as shown below.



- The **Output Size** in the Unpack Data Step is the same as the input image size.
- After data unpacking, the Deep Learning Result Parser Step may not automatically determine the scenario of the model due to missing subtype data. Thus, the output port of the Deep Learning Result Parser Step will not be generated. You will need to set the **Deep Learning**

Value Type in this Step.

Once the **Deep Learning Value Type** is selected, it cannot be changed. If you want to change the **Deep Learning Value Type**, please delete and re-add the Deep Learning Result Parser Step.

4.3.110. Depth Clustering along Scan Lines

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step clusters the depth values along the direction of the laser scanline for noise removal.

Usage Scenario

This Step is a custom Step for special scenarios.

Input and Output

- **Input**: Depth map input to this port will be clustered.
- **Output**: depth maps after clustering.

4.3.111. Depth Encoding

This Step is outdated. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step encodes the depth map within the set depth range to convert the depth map to a color image.

Usage Scenario

This Step is a custom Step for special scenarios.

Input and Output

- **Input**: Depth map input to this port will be converted to a color image.
- **Output**: color image obtained from conversion.

4.3.112. Detect and Measure Circle

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step manually sets an ROI that contains the circle to detect, and this Step will detect the circle and output its info. Edge detection frames will be generated along the roughly drawn circle in the manually selected ROI, the object edge points will be detected by the grayscale changes in the edge detection frames, and in the end, the edge points will be used to fit the accurate circle. The circle for circumferential arrangement takes the center of the ROI as the center, and takes the smaller value of the half-height and half-width of the ROI as the radius.

Usage Scenario

This Step is for detecting the pixel-wise positions and radii of circles in images to prepare for subsequent calculation of physical dimensions in measurement scenarios.

Input and Output

- Input:
  1. Image input at this port will have a circle in ROI detected.
  2. 2D transformation input at this port is used to transform the ROI set in a previous execution to the appropriate position and orientations.
- Output:
  1. Visualization image.
  2. The center coordinates and radius (in pixels) of the detected circle.

4.3.113. Detect and Measure Line

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step manually sets an ROI that contains the line segment to detect, and this Step will detect the line segment and output its info. Edge detection frames arranged in parallel along the lateral direction of the ROI will be generated within the manually selected ROI, object edge points will be detected through grayscale changes in the edge detection frames, and in the end, the edge points will be used to fit the accurate line.

Usage Scenario

This Step is for detecting the pixel-wise positions of object edge line segments in images to prepare for subsequent calculation of physical dimensions in measurement scenarios.

Input and Output

- Input:
  1. Image input at this port will have a line segment in ROI detected.
  2. 2D transformation input at this port is used to transform the ROI set in a previous execution to the appropriate position.
- Output:
  1. Visualization image.
  2. Pixel-wise coordinates of the two endpoints of the detected line segment.
  3. Pixel-wise coordinates of the midpoint of the detected line segment.
  4. The tilt angle of the detected line from the horizontal direction. Range: –180° to 180°. The horizontal direction pointing to the right of the image is 0°, and the clockwise direction is positive.

4.3.114. Detect and Measure Oblong Hole

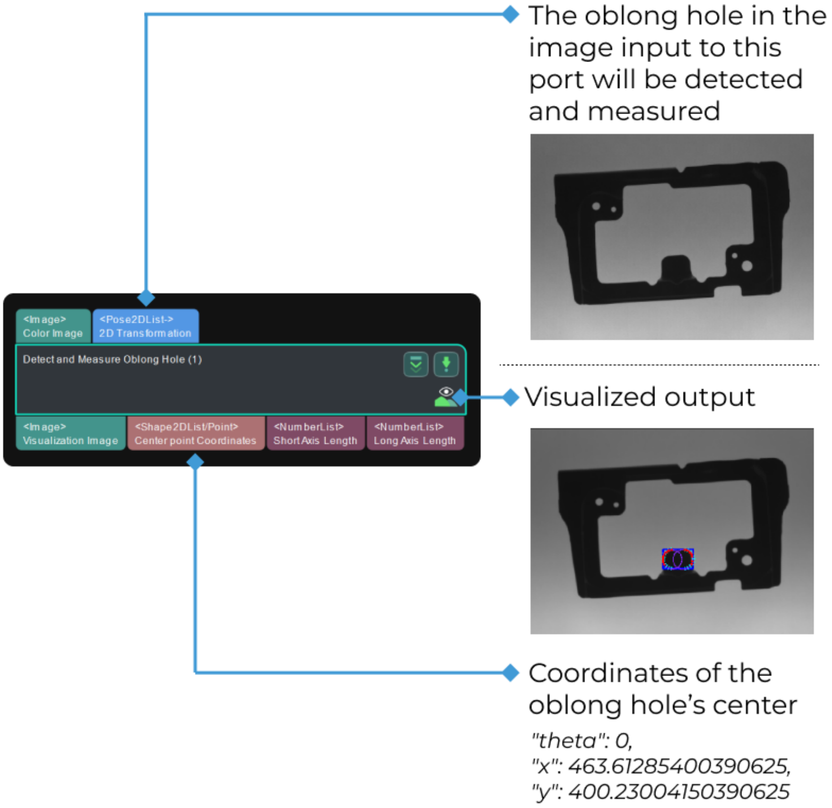
Function

This Step is used to detect and measure oblong hole in the input image.

Usage Scenario

This Step is used to detect the pixel-wise positions and sizes of oblong holes in images to facilitate subsequent calculation of physical dimensions in measurement scenarios.

Input and Output


<Image> Color Image
<Pose2DList-> 2D Transformation
Detect and Measure Oblong Hole (1)
<Image> Visualization Image
<Shape2DList/Point> Center point Coordinates
<NumberList> Short Axis Length
<NumberList> Long Axis Length

The oblong hole in the image input to this port will be detected and measured

Visualized output

Coordinates of the oblong hole's center

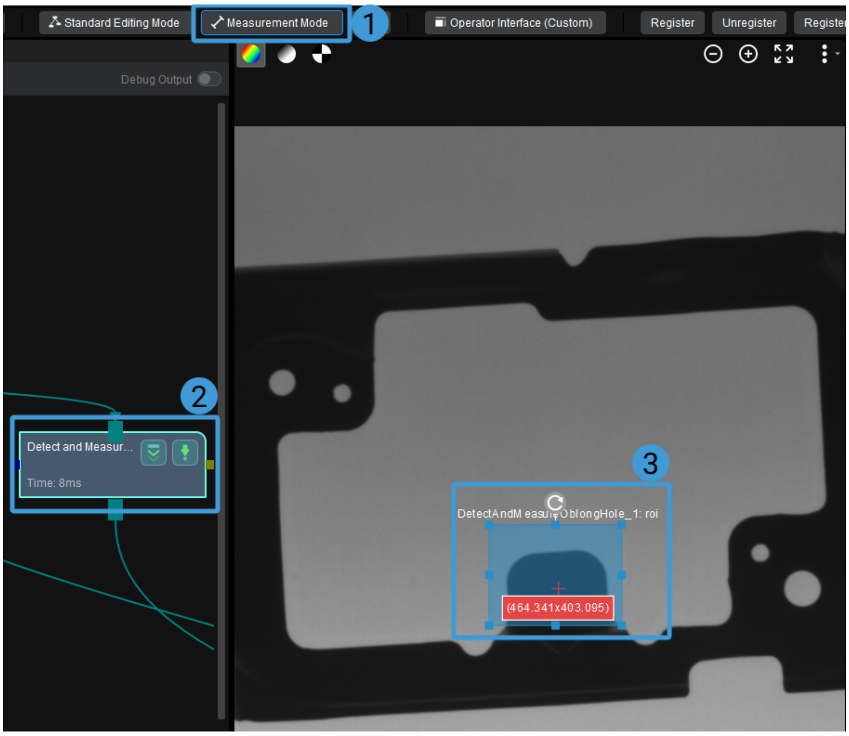
"theta": 0,
"x": 463.61285400390625,
"y": 400.23004150390625

Parameter Description

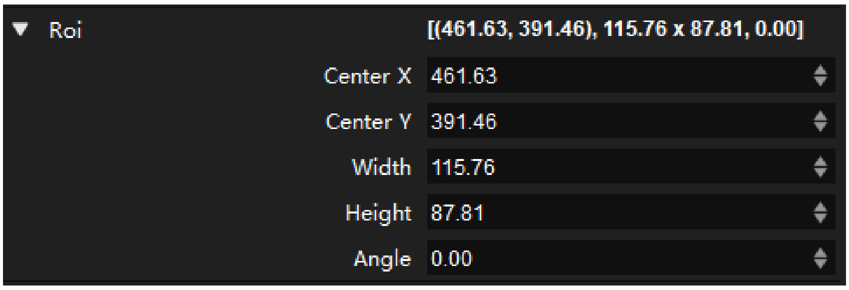
ROI

Default value: 0

Instruction: Click [**Measurement Mode**] in the toolbar to enable the measurement mode and then click this Step and select an ROI. The select frame is in the upper left corner of the image. It is recommended to zoom out the image first and hover the mouse on the frame. When the arrow icon becomes the hand icon, you can drag the select frame and adjust the frame size.



Once the selection is completed, the values will be displayed in the ROI parameters, as shown below.



Roi	[(461.63, 391.46), 115.76 x 87.81, 0.00]
Center X	461.63
Center Y	391.46
Width	115.76
Height	87.81
Angle	0.00

Position and Orientation Correction

Default value: Unselected.

Description: If this option is selected, the ROI will be adjusted to the position and orientation that fits the object in the image according to the input 2D transformation.

Canny Edge Detection Low Threshold

Default value: 40

Description: Edges with gradients greater than the high threshold are considered actual edges. Edges with gradients between high and low thresholds are considered actual edges only if they are connected to actual edges with gradients greater than the high threshold. Edges with gradients below the low threshold are ignored. The Soble operator is used to calculate the gradients, and the maximum of gradients is 1440.

Canny Edge Detection High Threshold

Default value: 100

Description: Edges with gradients greater than the high threshold are considered actual edges. Edges with gradients between high and low thresholds are considered actual edges only if they are connected to actual edges with gradients greater than the high threshold. Edges with gradients below the low threshold are ignored. The Soble operator is used to calculate the gradients, and the maximum of gradients is 1440.

Sigma of Gaussian Filter

Default value: 1.0

Description: This parameter is used to filter the image vertically within the edge detection frames. The recommended set value is 1.

Edge Transition Type

Default value: Both.

Value list: White To Black, Black To White, and Both.

Description: This parameter specifies which type of grayscale change will be considered as object edge points. The grayscale change is defined as the grayscale change vertically from top to bottom in an edge detection frame.

- White to Black: The grayscale changes from white to black in the frames will be considered points on object edges.
- Black to White: The grayscale changes from black to white in the frames will be considered points on object edges.
- Both: The grayscale changes both from black to white and from white to black in the frames will be considered points on object edges.

Edge Selection

Default value: First.

Options: First, Second, Last, and All.

Description: In each edge detection frame, this parameter sets which detected edge point should be selected as the edge point for fitting the complete edges.

First: The first point from top to bottom in the frame will be selected for fitting.

- Second: The second point from top to bottom in the frame will be selected for fitting.
- Last: The last point from top to bottom in the frame will be selected for fitting.
- All: All the points in the frame will be selected for fitting.

4.3.115. Detect Bin (Four Sides)

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step calculates the poses and dimensions of the bin and its four sidewalls.

Usage Scenario

This Step is used to recognize regular four-sided bins and send the recognition results to external services for collision detection.

Input and Output

- Input:
  1. Point cloud of bin’s top surface
  2. Mask of bin’s top surface
  3. Depth map containing the bin directly from the camera
- Output:
  1. 3D center pose of bin
  2. Poses of four sidewalls of bin
  3. 3D dimensions of bin and its four sidewalls

4.3.116. Detect Bin (Max Inscribed Rect)

This Step is outdated. Please use the new Step Detect Bin (Max Inscribed Rect) V2 instead. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step calculates the poses of the entire bin and the four sidewalls according to the maximum inscribed rectangle of the upper surface mask of the bin.

Usage Scenario

This Step is used to recognize the bin and send the results to external services for collision detection.

Input and Output

- Input:
  1. Mask on the upper surface of the bin
  2. Depth map containing the bin directly from the camera
- Output:
  1. Poses of the bin and its four sidewalls
  2. 3D Dimensions of the bin and its four sidewalls

4.3.117. Detect Bin (Max Inscribed Rect) V2

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step inputs the point cloud and pose of bin to estimate the pose and dimensions of bin. Please note that the dimensions of the output are the outer dimensions of the bin.

Usage Scenario

This Step is used to recognize comparatively irregular four-sided bins and send the recognition results to external services for collision detection.

Input and Output

- Input:
  1. Point cloud of bin's top surface
  2. Pose of bin's top surface
- Output:
  1. Detected bin pose
  2. Detected outer contour dimensions of bin

4.3.118. Detect Bin (Two Sides)

> Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

Given the poses of the two sidewalls of the bin and the settings of the collision model, this Step calculates the pose and dimensions of the collision model of each sidewall.

Usage Scenario

This Step is used to recognize the bin and send the results to an external service for collision detection.

Input and Output

- Input:
  1. Poses of the two sidewalls of the bin
- Output:
  1. Collision model poses sent to external service
  2. Poses of two sidewalls
  3. Dimensions of collision model

4.3.119. Detect Circle Centers

> Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step detects the center of the circles in the image.

Usage Scenario

This Step is a general circle center detection Step. This Step has no fixed usage scenarios.

Input and Output

- Input: Centers of circles in the image input to this port will be detected.

- **Output:** information of the detected circles (including center, and radius).

4.3.120. Detect Corners

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step detects and outputs object corners from the input images.

Usage Scenario

This Step is a general 2D feature extraction Step. This Step has no fixed usage scenarios.

Input and Output

- **Input:**
  1. Corners in the images input to this port will be detected.
  2. Image input to this port will have detected corners drawn on it.
- **Output:**
  1. Detected corners with orientation
  2. Color image with corners and contours of objects formed by the corners

4.3.121. Detect Edges

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step is used to detect edges in 2D images.

Usage Scenario

This Step is generally used to detect edges in 2D images. This Step has no fixed usage scenarios.

Input and Output

- **Input:** Edges of the target objects in the 2D image input to this port will be detected.
- **Output:** mask of detected object edges.

4.3.122. Detect Inscribed Circles

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step extracts contours from the input 2D image and calculates inscribed circles in the extracted contours.

Usage Scenario

This Step is a general 2D feature extraction Step. This Step has no fixed usage scenarios.

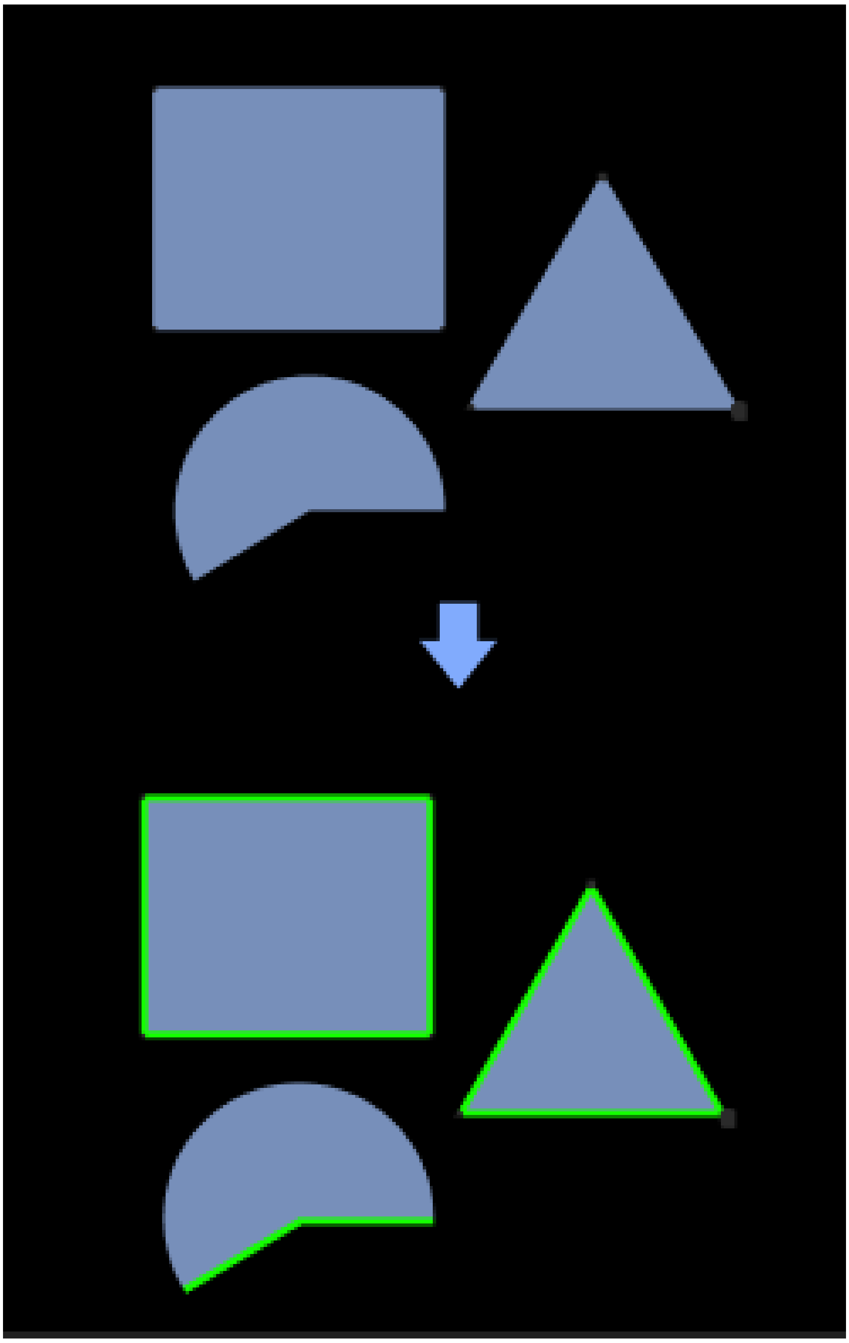
Input and Output

- Input:
  1. Masks input for this port will have inscribed circles calculated.
  2. Image used to visualize the inscribed circle.
- Output:
  1. Masks to represent inscribed circles
  2. Visualization with calculated inscribed circles

4.3.123. Detect Line Segments

Function

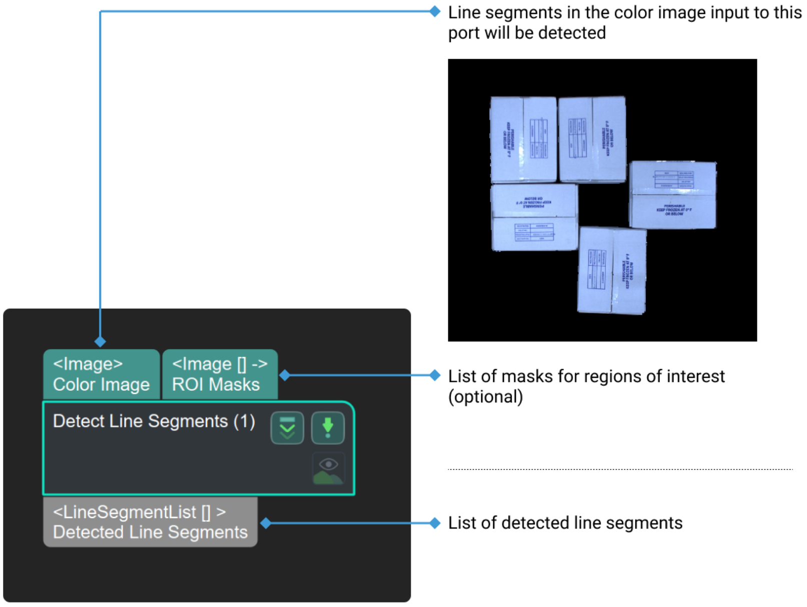
This Step detects line segments in an image.



Usage Scenario

This Step extracts the line segment features of objects in the 2D image for subsequent matching.

Input and Output



Line segments in the color image input to this port will be detected

List of masks for regions of interest (optional)

List of detected line segments

Parameters

Line Detector Setting

Line Detector Type

Parameter description: This parameter is used to designate the type of line detector.

List of value:

- LineSegmentDetector
- FastLineDetector
- EdgeDrawingDetector

Default value: LineSegmentDetector

Line Properties

Min Line Length

Parameter description: This parameter sets the minimal length of the line to be detected. Lines shorter than the set value of the parameter will be filtered out.

Default value: 10 px

Suggested value: 10 px, 20 px, 30 px

Filter Out Lines outside Masks

Parameter description: This parameter decides whether to filter out lines that are detected outside the masks.

Default value: unselected.

Suggestion: Set the parameter according to your actual needs.

4.3.124. Detect N Largest Rectangles in Mask

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step detects the first N inscribed rectangles with the largest area in the mask. The value of N can be set in the parameter.

Usage Scenario

This Step is a general mask calculation Step. This Step has no fixed usage scenarios.

Input and Output

- **Input:** Masks input by this port will have the largest inscribed rectangles detected.
- **Output:** Top N largest inscribed rectangles detected from the masks.

4.3.125. Detect Occluded Objects

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step detects the rectangular vacant area between objects, and calculates the poses and dimensions of the occluded objects.

Usage Scenario

This Step is a custom Step for special scenarios.

Input and Output

- **Input:**
  1. Point cloud containing occluded objects
  2. Object center poses
- **Output:**
  1. Center poses of occluded objects
  2. 3D dimensions of occluded objects
  3. 2D vertices of occluded objects in an image

4.3.126. Detect Shape Feature of Region

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step detects the bounding rectangle of the non-zero pixel region of the mask.

Usage Scenario

This Step is a general mask data processing Step. This Step has no fixed usage scenarios.

Input and Output

- **Input:** Mask input by this port will have the bounding rectangle of the non-zero pixel region detected.
- **Output:** Mask for the detected bounding rectangle.

4.3.127. Detect Vertex

> Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step manually sets an ROI that contains the line segment to detect, and this Step will detect a vertex and output its info.

Usage Scenario

This Step is generally used to find the vertices of objects in the measurement scenarios.

Input and Output

- Input:
  1. An object vertex will be extracted in the ROI of image input at this port.
  2. 2D transformation input at this port is used to transform the ROI set in a previous execution to the appropriate position.
- Output:
  1. Visualize the image, which can display the edge points, ROI borders, vertex, and baseline according to parameter settings.
  2. Pixel-wise coordinates of the vertex.

4.3.128. Determine Pixel Sizes

> Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step determines the physical sizes of pixels from the input circles or line segments.

Usage Scenario

This Step is a general pixel physical size calculation Step. This Step has no fixed usage scenarios.

Input and Output

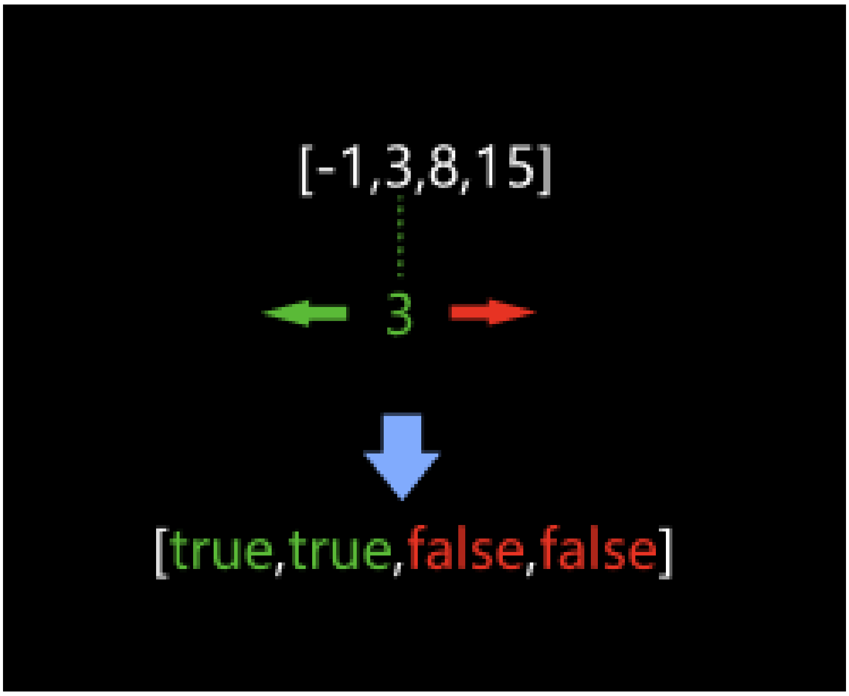
- Input:
  1. Background image for visualization
  2. Line segments or circles used to calculate pixel physical sizes (port type is set by parameter)
- Output:
  1. Visualization with line segments or circles
  2. Calculated pixel physical sizes

4.3.129. Dichotomize Values by Threshold

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

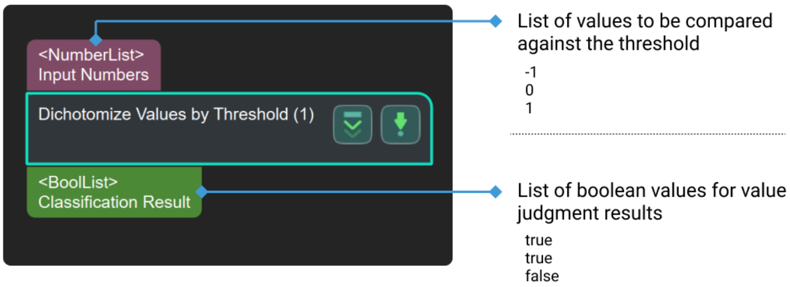
Perform a value judgment on each input list item based on a set threshold.



Usage Scenario

If a value in the list is less than or equal to the corresponding threshold, the corresponding item in the output list will be True; otherwise, it will be False. This Step is usually used with Filter.

Input and Output


<NumberList>
Input Numbers
Dichotomize Values by Threshold (1)
<BoolList>
Classification Result
List of values to be compared against the threshold
-1
0
1
List of boolean values for value judgment results
true
true
false

4.3.130. Divide Point Cloud into Smaller Parts Evenly

> Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step divides the point cloud evenly into smaller parts.

Usage Scenario

This Step is a custom Step for special scenarios.

Input and Output

- **Input:** Point cloud input to this port will be divided.
- **Output:** Point cloud segments.

4.3.131. Down-Sample Point Cloud

> Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

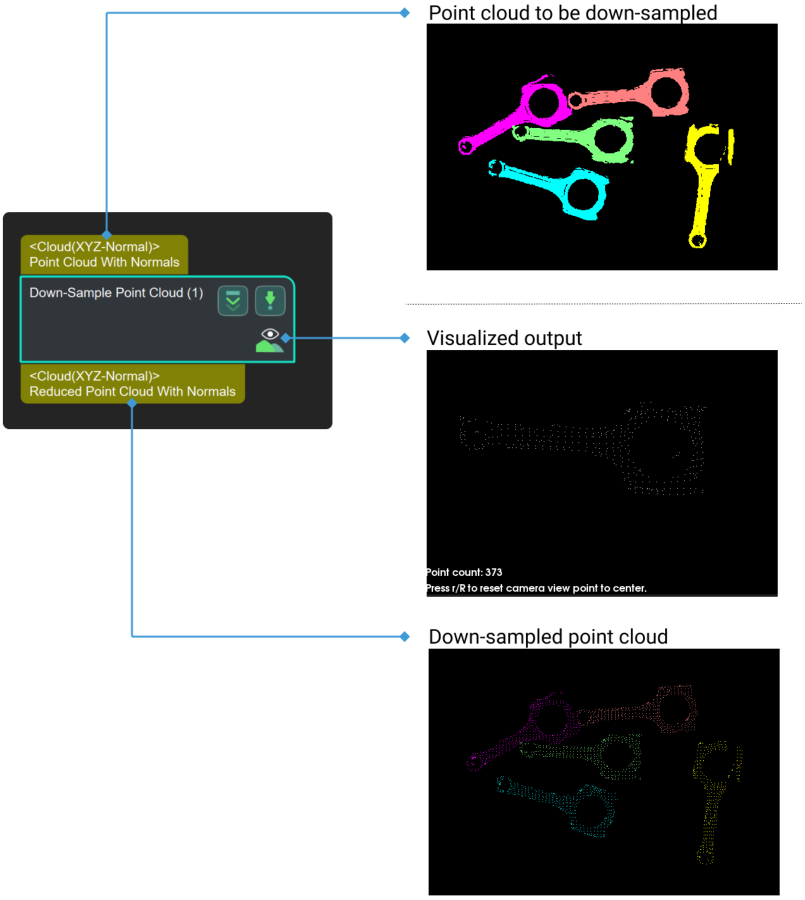
Function

This Step is to down-samples a point cloud to reduce the number of points.

Usage Scenario

This Step is used for point cloud preprocessing when there are too many cloud points to be processed, to improve the project running speed.

Input and Output


Point cloud to be down-sampled
<Cloud(XYZ-Normal)>
Point Cloud With Normals
Down-Sample Point Cloud (1)
<Cloud(XYZ-Normal)>
Reduced Point Cloud With Normals
Visualized output
Point count: 373
Press r/R to reset camera view point to center.
Down-sampled point cloud

4.3.132. Draw Min Circumscribed Rectangles of Masks

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step draws the smallest circumscribed rectangles of the input masks.

Usage Scenario

This Step is a general mask processing Step. This Step has no fixed usage scenarios.

Input and Output

- Input:
  1. Image input to this port will be used as the visualization background.
  2. Masks input by this port will have their minimum circumscribed rectangle calculated.
  3. Object dimensions used to label the visualization.
- Output:
  1. Visualized results.

4.3.133. Easy Frame Transformation

> Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step transforms the input poses between the camera reference frame and the robot reference frame.

Usage Scenario

This Step is a general pose reference frame transformation Step. This Step has no fixed usage scenarios.

Input and Output

- **Input:** Poses input to this port will be transformed to another reference frame.
- **Output:** poses transformed to another reference frame.

4.3.134. Easy Create Index List

> Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step creates an index list. The total number of elements N can be set in the parameter.

Usage Scenario

This Step is a general data creation Step. This Step has no fixed usage scenarios.

Input and Output

- **Input:** open interval cutoff range for the index (optional, if this port is not used, the range defined by the parameter will be used).
- **Output:** Index list created.

4.3.135. Easy Create Number List

> Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step creates number list data. The content can be set in Step Parameters.

Usage Scenario

This Step is a general data creation Step. This Step has no fixed usage scenarios.

Input and Output

- **Input:** the open interval cutoff range of the number list (input when the parameter input type is set to “RangeBased”; optional, if this port is not used, the range set by the parameter will be used).
- **Output:** number list created.

4.3.136. Easy Create Poses

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step creates a pose list, with poses entered in the parameter.

Usage Scenario

This Step is a general data creation Step. This Step has no fixed usage scenarios.

Input and Output

- **Input**: none.
- **Output:** created pose list.

4.3.137. Easy Create Quaternions

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step creates quaternion list data from the list of rotation vectors to set in parameters.

Usage Scenario

This Step is a general data creation Step. This Step has no fixed usage scenarios.

Input and Output

- **Input:** none.
- **Output:** created list of quaternions.

4.3.138. Easy Create String List

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

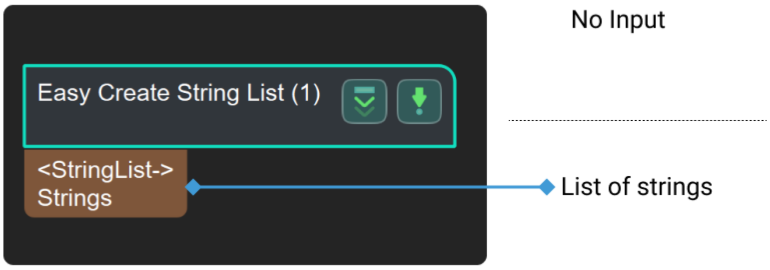
Function

Create a list of labels from a list of strings entered in the parameter.

Usage Scenario

A general label list creation Step.

Input and Output



4.3.139. Easy Create Vector3Ds

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step creates 3D vector data from parameter setting.

Usage Scenario

This Step is a general data creation Step. This Step has no fixed usage scenarios.

Input and Output

- **Input:** none.
- **Output:** created 3D vector list.

4.3.140. Easy Point Poses to Reference Position

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

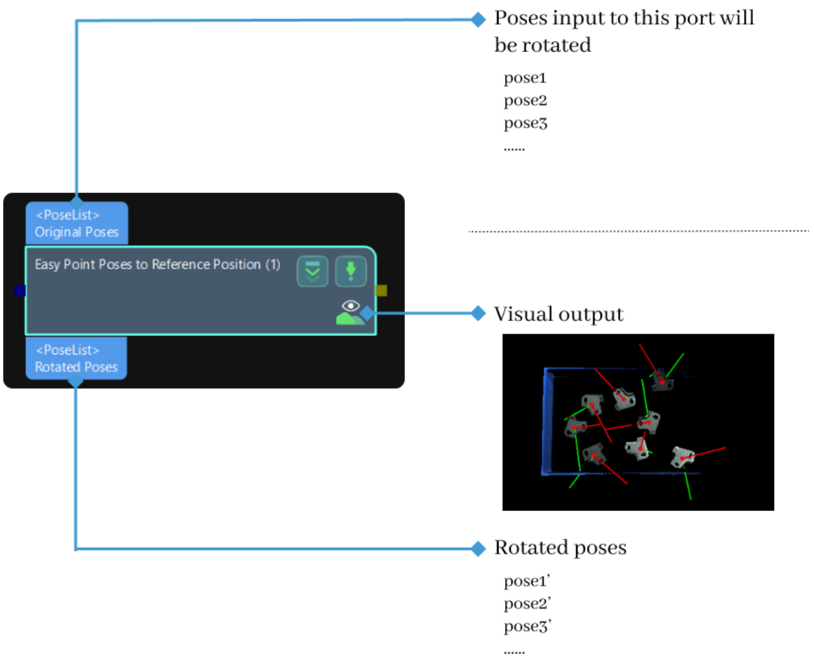
Function

Rotate the input original poses so that the specified axes point to a set reference point.

Usage Scenario

A general pose rotation Step. This Step has no fixed usage scenarios.

Input and Output

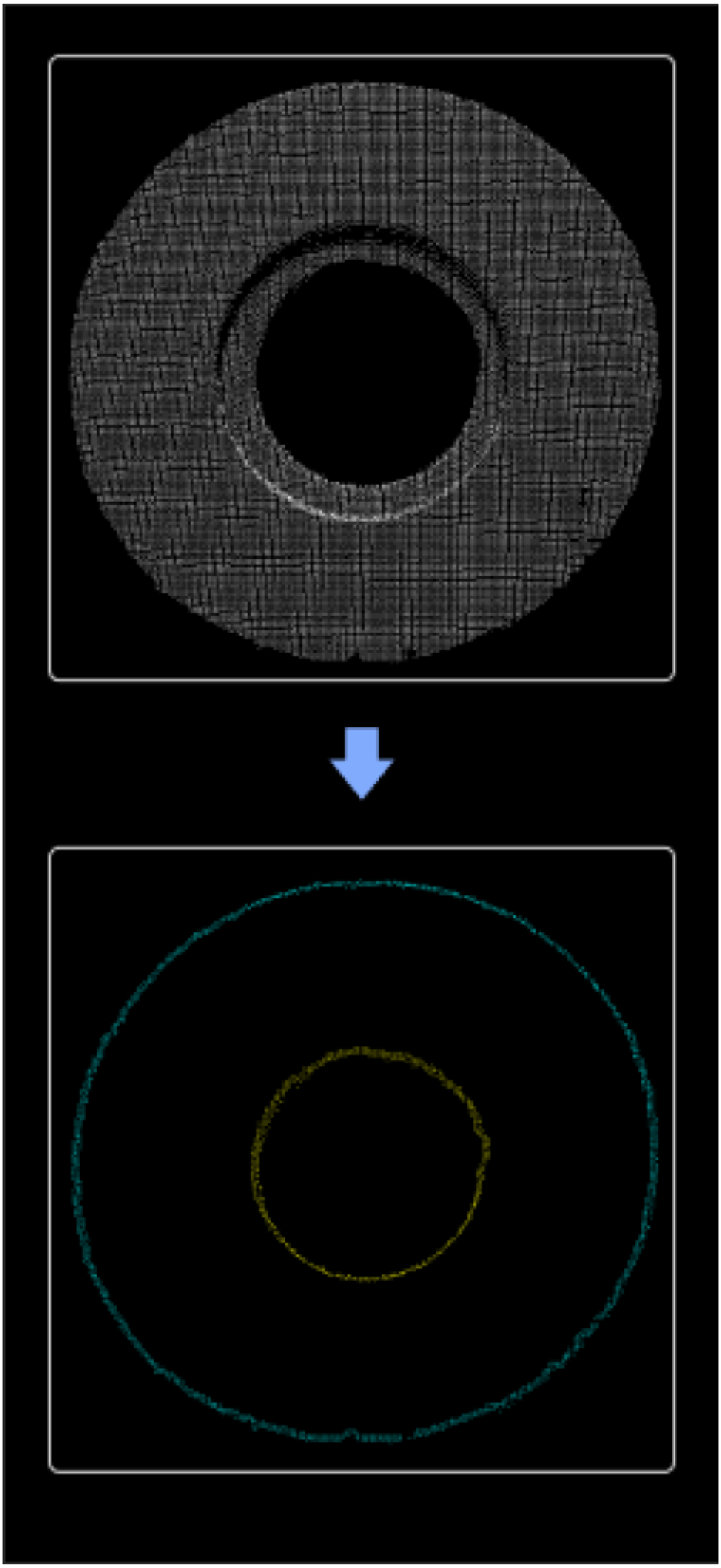


4.3.141. Estimate Point Cloud Edges by 2D Method

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

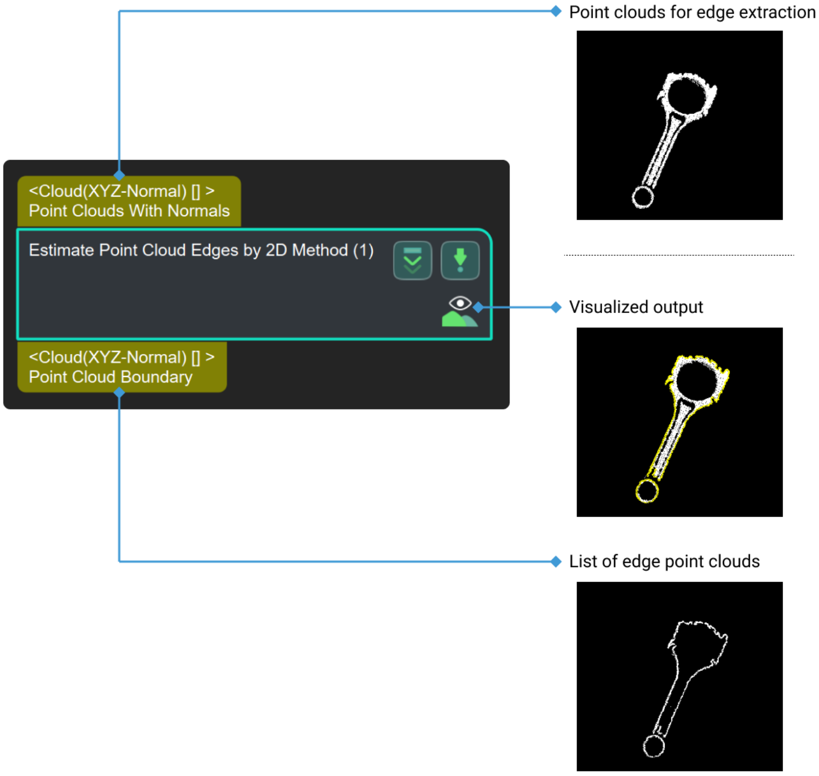
This Step extracts the edges from input point clouds.



Usage Scenario

This Step is usually used in the loading of the planar workpiece to obtain point cloud edges of planar workpieces. This step is usually used with 3D matching Steps. This Step is faster than Estimate Point Cloud Edges by 3D Method. It is recommended to use it for projects requiring short cycle times, with large quantities of 3D point cloud interference, and only requiring extracting the outermost edge points of the target object point clouds.

Input and Output

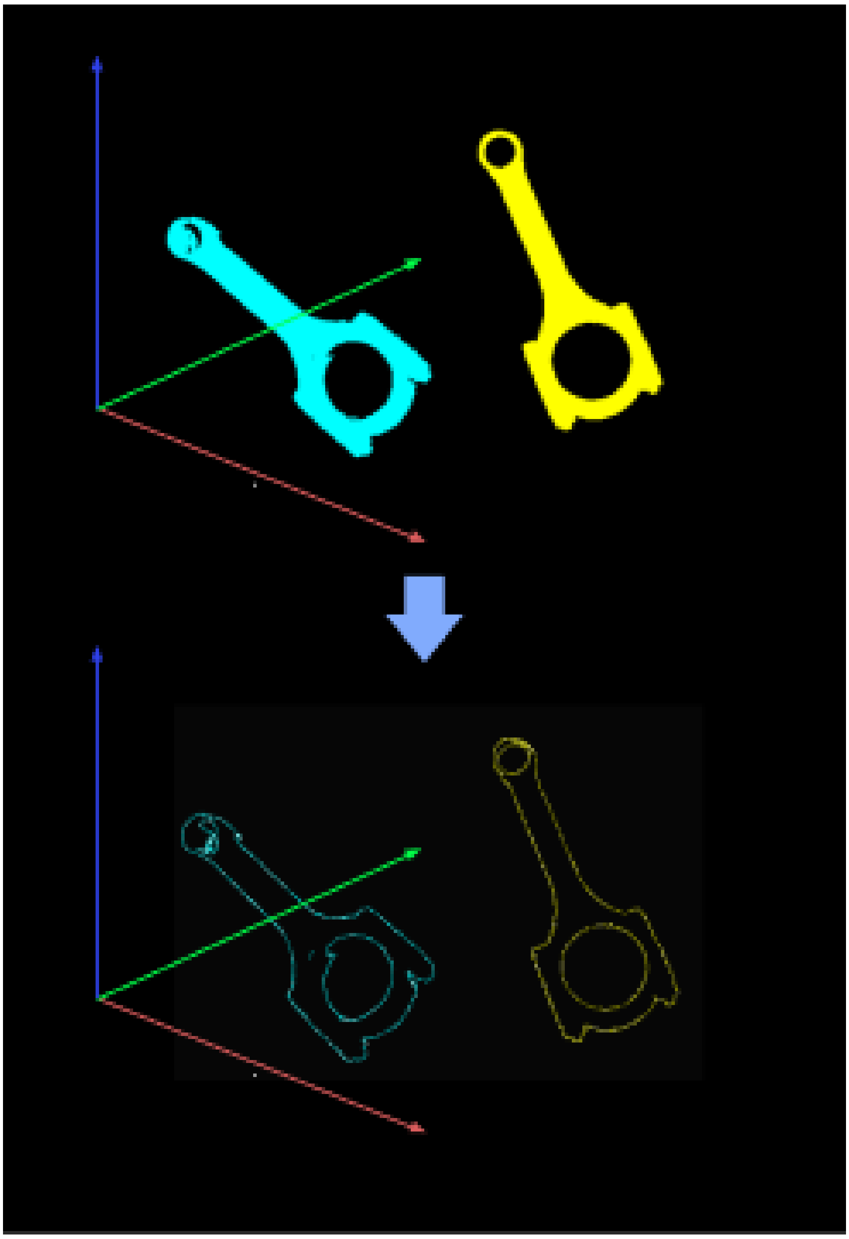

Point clouds for edge extraction
<Cloud(XYZ-Normal) [] >
Point Clouds With Normals
Estimate Point Cloud Edges by 2D Method (1)
Visualized output
<Cloud(XYZ-Normal) [] >
Point Cloud Boundary
List of edge point clouds

4.3.142. Estimate Point Cloud Edges by 3D Method

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

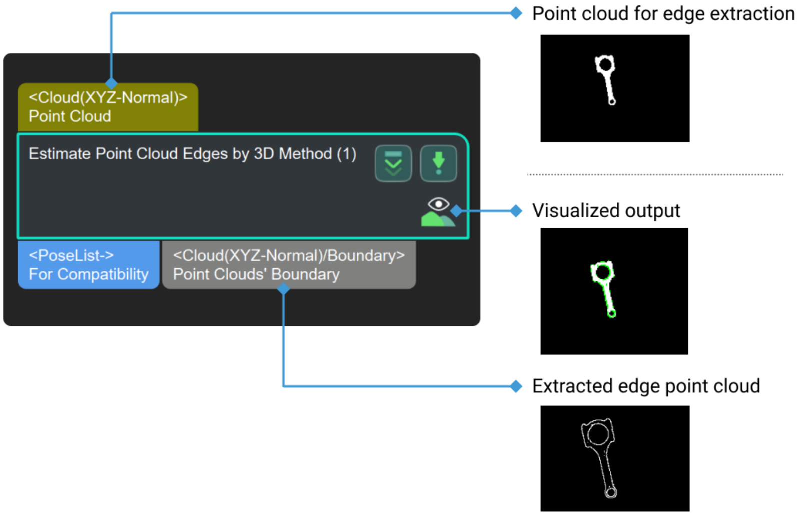
This Step extracts the edges of a 3D point cloud.



Usage Scenario

This Step is usually used for matching based on point cloud edges, which is faster than matching based on full point clouds.

Input and Output


Point cloud for edge extraction
<Cloud(XYZ-Normal)>
Point Cloud
Estimate Point Cloud Edges by 3D Method (1)
Visualized output
<PoseList->
For Compatibility
<Cloud(XYZ-Normal)/Boundary>
Point Clouds' Boundary
Extracted edge point cloud

4.3.143. Evaluate Image Clarity

Function

This Step measures the clarity of the input images according to the specified method. Different methods have different scales and standards of clarity.

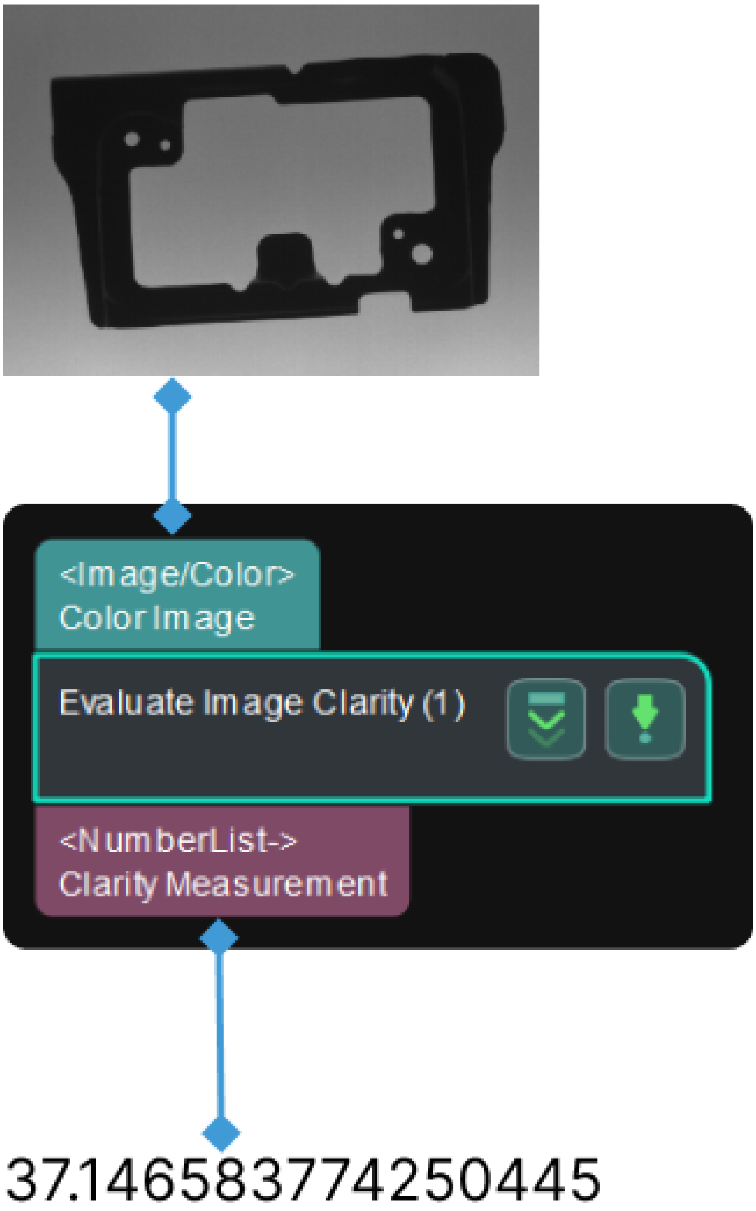
When a single image or a group of images captured by the camera under different focusing conditions are input into this Step, it can output the result that can help judge whether the camera focuses. Greater image clarity means more accurate focusing of the camera.

Since the scales for evaluating the image clarity when different methods are used are different, the clarity values with different methods cannot be compared.

Usage Scenario

This Step is usually used to assist the parameter and position adjustment of the camera in measurement scenarios.

Input and Output



37.146583774250445

Parameter Description

Method

Parameter description: This parameter selects the method to evaluate image clarity.

Value list: Energy of Gradient, Brenner, Roberts Cross, SML, Sobel, and Tenegrad.

Tuning recommendation: The tuning recommendations of using different methods are as follows.

- Energy of Gradient: This method is based on the energy of gradient (EOG) function. It is defined as follows.

$$F_{\mathrm{EOG}} = \sum_{i,j} [|\, g(i,j) - g(i,j-1)| + |g(i,j) - g(i-1),j|]$$

- Brenner: This method is based on the Brenner function. It is defined as follows.

$$F_{\mathrm{Brenner}} = \sum_{i,j} |\, g(i,j) - g(i+2,j)\, |^2$$

- Roberts Cross: The method is based on a function of the differential absolute value sum of the Robert operator (the Robert function for short). It is defined as follows.

$$F_{\mathrm{Robert}} = \sum_{i,j} [|\, g(i,j) - g(i+1,j+1)\, | + |\, g(i+1,j) - g(i,j+1)\, |]$$

- SML: The method is improved on the basis of Laplace operator and has better noise immunity. It is defined as follows.

$$F_{\mathrm{SML}} = \sum_{i,j} [|\, 2g(i,j) - g(i-1,j) - g(i+1),j\, | + |\, 2g(i,j) - g(i,j-1) - g(i,j+1|]^2$$

- Sobel: The method is based on the Sobel operator (the Sobel function for short). It is defined as follows. The formula contains the first difference of sobel operator in the X- and Y-direction of the image.

$$F_{\mathrm{Sobel}} = \sum_{i,j} [|\, G_x(i,j)\, | + |\, G_y(i,j)\, |]$$

- Tenegrad: This method is another function based on the Sobel operator. It is defined as follows.

$$F_{\mathrm{Tenengrad}} = \sum_{i,j} \left[G_x^2(i,j) + G_y^2(i,j) \right]$$

Default value: Brenner

Tuning recommendation: Select the method for evaluating the image clarity according to the actual requirement.

4.3.144. Evaluate Results by HDevEngine

Function

This Step runs a predefined Halcon vision program through HDevEngine and outputs the calculation result to Mech-Vision.

Usage Scenario

This Step is used for calling some of the procedures in Halcon.

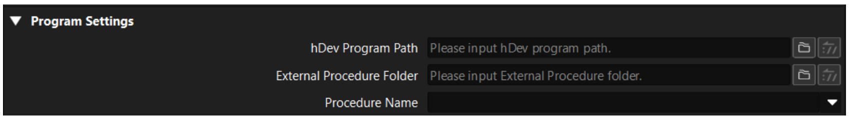
Input and Output

- **Inputs**: determined by the data type entered in the **Input Ports Desc** parameter.
- **Outputs**: determined by the data type entered in the **Output Ports Desc** parameter.

Parameter Description

Program Settings

The parameters in “Program Settings” are shown in the figure below.


▼ Program Settings
hDev Program Path
Please input hDev program path.
External Procedure Folder
Please input External Procedure folder.
Procedure Name

hDev Program Path

Description: This parameter is used to specify the path of the HDev program, i.e., the path of the main program in Halcon HDevelop.

Eternal Procedure Folder

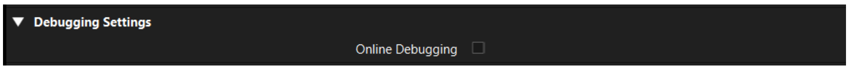
Description: This parameter is used to specify the path where the Halcon external procedures are stored (Multiple paths can be added by separating them with semicolons, “;”).

Procedure Name

Description: This parameter is used to set the name of the external procedure. You can select the name in the drop-down list.

Debugging Settings

The parameters in “Debugging Settings” are shown in the figure below.



Online Debugging

Description: Select to activate the debug server. Then you can check the data input by Mech-Vision in real-time in Halcon HDevelop and debug the program.

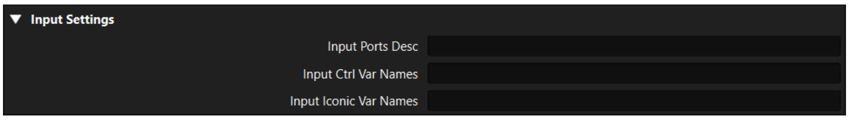
Default setting: Unselected.

Tuning instruction: Once **Online Debugging** is selected, go to the menu bar of HALCON HDevelop and click Execute › Attach To Process..., and click [OK] in the pop-up window to activate the debug server. If the debug server is not activated in Halcon, Mech-Vision may be lagging during the execution.

Tuning recommendation: Please select the option according to the actual requirement.

Input Settings

The parameters in “Input Settings” are shown in the figure below.



▼ Input Settings

Input Ports Desc

Input Ctrl Var Names

Input Iconic Var Names

The **Input Settings** can only be configured after the **Procedure Name** is entered, or else the relevant parameters will not appear.

Input Ports Desc

Description: This parameter is used to specify the data type(s) of the input port(s).

Input Ctrl Var Names

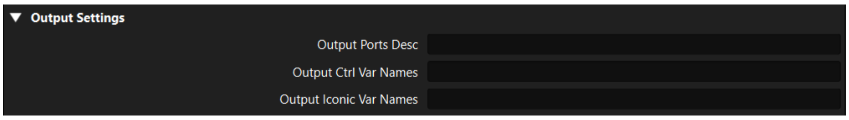
Description: This parameter is used to set the list of names of the input control data.

Input Iconic Var Names

Description: This parameter is used to set the list of names of the input iconic data.

Output Settings

The parameters in “Output Settings” are shown in the figure below.



▼ Output Settings

Output Ports Desc

Output Ctrl Var Names

Output Iconic Var Names

Output Ports Desc

Description: This parameter is used to specify the data type(s) of the output port(s).

Output Ctrl Var Names

Description: This parameter is used to set the list of names of the output control data.

Output Iconic Var Names

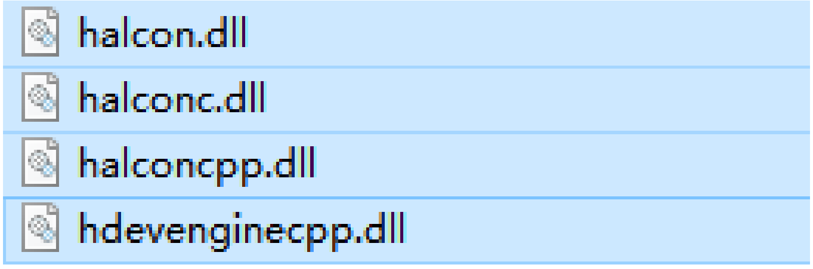
Description: This parameter is used to set the list of names of the output iconic data.

Instructions

Prerequisites

To use the “Evaluate Results by HDevEngine” Step, the following two prerequisites should be met, or else the Step cannot be found in the Step library.

1. The Halcon HDevelop software with a license has been installed on your computer.
2. Copy the related DLL files, as shown in the figure below, in the `Installation folder\bin\x64-win64` of Halcon HDevelop to the installation folder of Mech-Vision. If Mech-Vision is already opened, Mech-Vision should be restarted to make the changes take effect.


halcon.dll
halconc.dll
halconcpp.dll
hdevenginecpp.dll

Supported Methods to Call Halcon Programs

- Call an HDev main program in Mech-Vision, and there are no external procedures in the main program.
- Call an HDev main program in Mech-Vision, and there are multiple external procedures in the main program.
- Call only one external procedure in Mech-Vision.

Data Types Supported for Conversion

In the Mech-Vision project, when other Steps input data to this Step, the data types supported by Mech-Vision will be converted to data types supported by Halcon. When this Step outputs data to other Steps, the data types supported by Halcon will be converted to data types supported by Mech-Vision.

The detailed information on the supported data types for conversion is as follows.

- Iconic Data Types

Halcon data type	Mech-Vision Data type	Note
Image	Image	The data types supported by Mech-Vision and Halcon can be converted to each other.

- Control Data Types

Halcon data type	Mech-Vision Data type	Note
Tuple	NumberList, String, StringList, Variant, VariantList	When other Steps input data to this Step, the input Mech-Vision data type can be converted to tuples, which can be a collection of heterogeneous data. When this Step outputs data to other Steps, only tuples with a single type of data can be converted to the data type supported by Mech-Vision.
integer	NumberList	Only when the Step outputs data to other Steps, the data types supported by Halcon can be converted to the data types supported by Mech-Vision.

Halcon data type	Mech-Vision Data type	Note
real	NumberList	Only when the Step outputs data to other Steps, the data types supported by Halcon can be converted to the data types supported by Mech-Vision.
string	String	Only when the Step outputs data to other Steps, the data types supported by Halcon can be converted to the data types supported by Mech-Vision.

Notes

Please pay attention to the following issues when using the “Evaluate Results by HDevEngine” Step to run the Halcon program.

- Parameters of **Input Settings** do not need to be configured when only the main program is executed.
- Running the main program and external procedures at the same time is not supported.
- If you want to debug or modify the codes in the Halcon program, please select “Reload File” in **Execution Flags**, and then the latest Halcon program can be reloaded every time the project is run. If the “Reload File” option is not selected, the unmodified program will be loaded when the project is run again.

Usage Examples

Example 1: Call an HDev main program without any external procedures

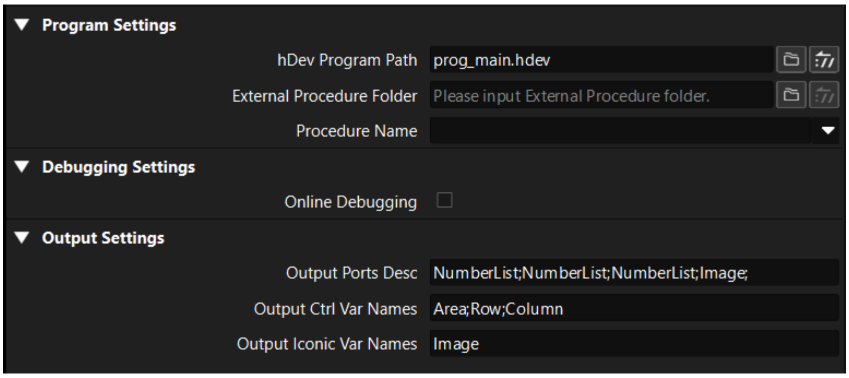
The Halcon program used in this example is shown below.

```
read_image (Image, 'fabrik')

regiongrowing (Image, Regions, 1, 1, 3, 200)

area_center (Regions, Area, Row, Column)
```

After saving the above program in Halcon HDevelop, you can call the program in Mech-Vision. The operation workflow includes the following steps.

1. Set the path of the main program. Set the **hDev Program Path** in the Step Parameters of the “Evaluate Results by HDevEngine” Step.
2. Set the output ports. According to the data types used in Halcon, set the **Output Ports Desc** parameter.
3. Set the names of the output data. According to the corresponding data names in Halcon, set **Output Ctrl Var Names** and **Output Iconic Var Names**.

After finishing the above operations, the Step parameters are shown below.



Program Settings

hDev Program Path: prog_main.hdev

External Procedure Folder: Please input External Procedure folder.

Procedure Name:

Debugging Settings

Online Debugging: □

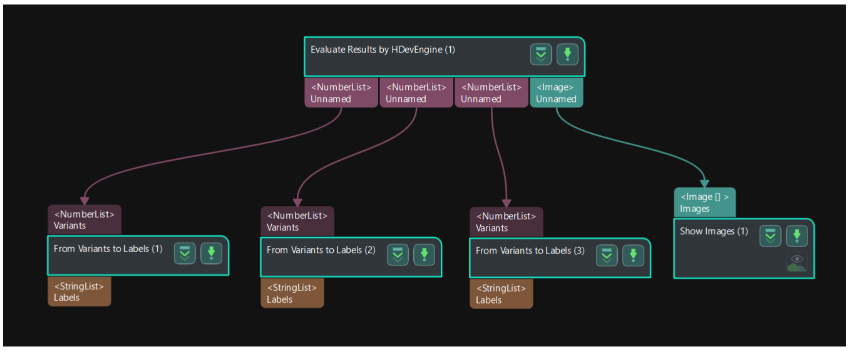
Output Settings

Output Ports Desc: NumberList;NumberList;NumberList;Image;

Output Ctrl Var Names: Area;Row;Column

Output Iconic Var Names: Image

The Mech-Vision project corresponding to this example is shown below. You can double-click the dataflow corresponding to the output ports to check the output results.



Example 2: Call an HDev main program with multiple external procedures

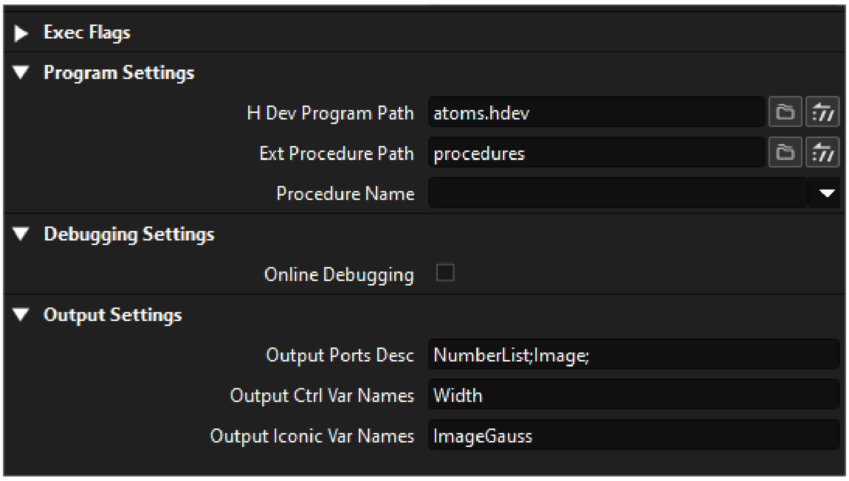
The Halcon program used in this example is shown below.

```
* atoms.hdev: Locates irregularities in an atomic grid structure
*
dev_close_window ()
dev_update_window ('off')
* ****
* Acquire image
* ****
read_image (Image, 'atoms')
get_image_size (Image, Width, Height)
crop_rectangle1 (Image, Image, Height / 2, 0, Height - 1, Width - 1)
get_image_size (Image, Width, Height)
*dev_open_window_fit_image (Image, 0, 0, -1, -1, WindowID)
*set_display_font (WindowID, 14, 'mono', 'true', 'false')
dev_set_draw ('margin')
dev_set_line_width (2)
dev_display (Image)
*disp_message (WindowID, 'Original image', 'window', 12, 12, 'black', 'true')
*disp_continue_message (WindowID, 'black', 'true')
stop ()
* ****
* Segment image
* ****
* -> Using watershed
gauss_filter (Image, ImageGauss, 5)
watersheds (ImageGauss, Basins, Watersheds)
dev_display (Image)
dev_set_colored (12)
dev_display (Watersheds)
*disp_message (WindowID, 'Watersheds', 'window', 12, 12, 'black', 'true')
*disp_continue_message (WindowID, 'black', 'true')
stop ()
* ****
* Process regions
* ****
* -> Skip regions at the border of the image
smallest_rectangle1 (Basins, Row1, Column1, Row2, Column2)
select_shape (Basins, SelectedRegions1, 'column1', 'and', 2, Width - 1)
select_shape (SelectedRegions1, SelectedRegions2, 'row1', 'and', 2, Height - 1)
select_shape (SelectedRegions2, SelectedRegions3, 'column2', 'and', 1, Width -
3)
select_shape (SelectedRegions3, Inner, 'row2', 'and', 1, Height - 3)
* -> Select irregularly shaped atoms
select_shape (Inner, Irregular, ['moments_i1','moments_i1'], 'or', [0,9.5e8],
[1.5e8,1e10])
dev_display (Image)
dev_set_line_width (1)
dev_set_color ('white')
dev_display (Inner)
dev_set_line_width (3)
dev_set_color ('red')
dev_display (Irregular)
*disp_message (WindowID, 'Defects', 'window', 12, 12, 'black', 'true')
```

After saving the above program in Halcon HDevelop, you can call the program in Mech-Vision. The operation workflow includes the following steps.

1. Set the path of the main program. Set the **hDev Program Path** in the Step Parameters of the “Evaluate Results by HDevEngine” Step.
2. Set the path of the external procedures. Set the path for the external procedures in **External Procedure Folder** in the Step Parameters of the “Evaluate Results by HDevEngine” Step.
3. Set the output ports. According to the data types used in Halcon, set the **Output Ports Desc** parameter.
4. Set the names of the output data. According to the corresponding data names in Halcon, set **Output Ctrl Var Names** and **Output Iconic Var Names**.

After finishing the above operations, the Step parameters are shown below.



▶ Exec Flags

▼ Program Settings

H Dev Program Path: atoms.hdev

Ext Procedure Path: procedures

Procedure Name:

▼ Debugging Settings

Online Debugging □

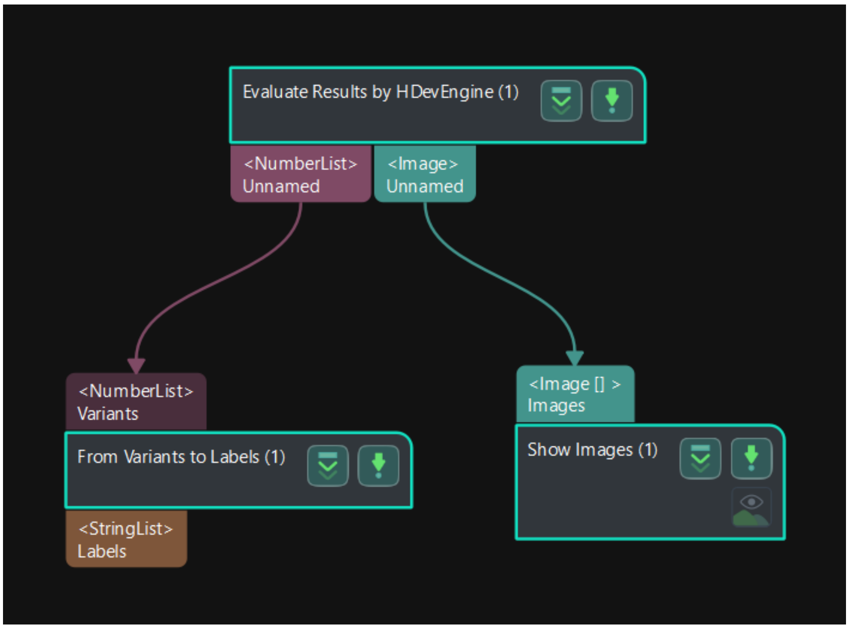
▼ Output Settings

Output Ports Desc: NumberList;Image;

Output Ctrl Var Names: Width

Output Iconic Var Names: ImageGauss

The Mech-Vision project corresponding to this example is shown below. You can double-click the dataflow corresponding to the output ports to check the output results.



Example 3: Call only one external procedure

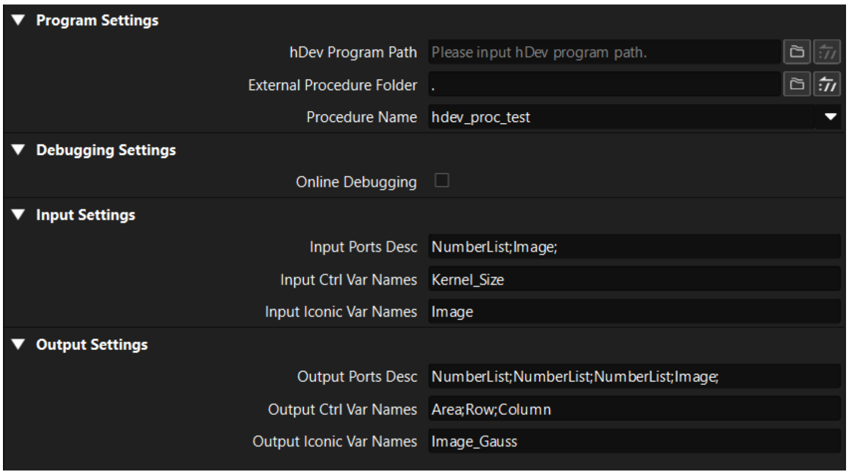
The Halcon program used in this example is shown below.

```
gauss_filter(Image, Image_Gauss, Kernel_Size)

regiongrowing (Image_Gauss, Regions, 1, 1, 3, 200)

area_center (Regions, Area, Row, Column)

return ()
```

After saving the above program in Halcon HDevelop, you can call the program in Mech-Vision. The operation workflow includes the following steps.

1. Set the path of the external procedures. Set the path for the external procedures in **External Procedure Folder** in the Step Parameters of the “Evaluate Results by HDevEngine” Step.
2. Set the name of the external procedure. Select the name of the external procedure in the drop-down list of the **Procedure Name** parameter.
3. Set the input ports. According to the parameter list of the external procedure, set the **Input Ports Desc** parameter.
4. Set the names of the input data. According to the corresponding data names in Halcon, set **Input Ctrl Var Names** and **Input Iconic Var Names**.
5. Set the output ports. According to the data types used in Halcon, set the **Output Ports Desc** parameter.
6. Set the names of the output data. According to the corresponding data names in Halcon, set **Output Ctrl Var Names** and **Output Iconic Var Names**.

After finishing the above operations, the Step parameters are shown below.



Program Settings

hDev Program Path: Please input hDev program path.
External Procedure Folder: .
Procedure Name: hdev_proc_test

Debugging Settings

Online Debugging ☐

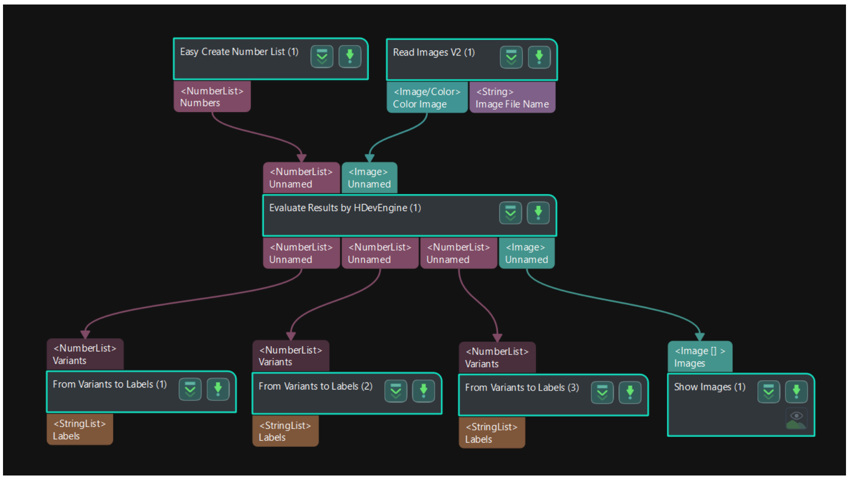
Input Settings

Input Ports Desc: NumberList;Image;
Input Ctrl Var Names: Kernel_Size
Input Iconic Var Names: Image

Output Settings

Output Ports Desc: NumberList;NumberList;NumberList;Image;
Output Ctrl Var Names: Area;Row;Column
Output Iconic Var Names: Image_Gauss

The Mech-Vision project corresponding to this example is shown below. You can double-click the dataflow corresponding to the output ports to check the output results.



4.3.145. Evaluate Results by JavaScript Engine

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step runs a user-defined script through the JavaScript engine and outputs the calculation results to Mech-Vision.

Usage Scenario

This Step can be used to simplify the project when complex logical judgments are required.

Input and Output

- **Input**: none.
- **Output**: none.

4.3.146. Evaluate Depth Map Fluctuation

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

Evaluate the fluctuation of the depth map over time on the same scene, and reflect the results in the log.

Usage Scenario

This Step is used by developers for testing. Regular users do not need to use this Step in general application scenarios.

Input and Output

- **Input:**
  1. Depth map input to this port will have fluctuations evaluated.
  2. The mask input to this port will be used to set the region for evaluation.
- **Output:**
  1. None.

4.3.147. External Interface

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step is used to receive parameter changes from an external interface.

Usage Scenario

This Step is a custom Step for special scenarios. For judgment on complex cases in the Operator Interface and transmit manual processing results back to the project.

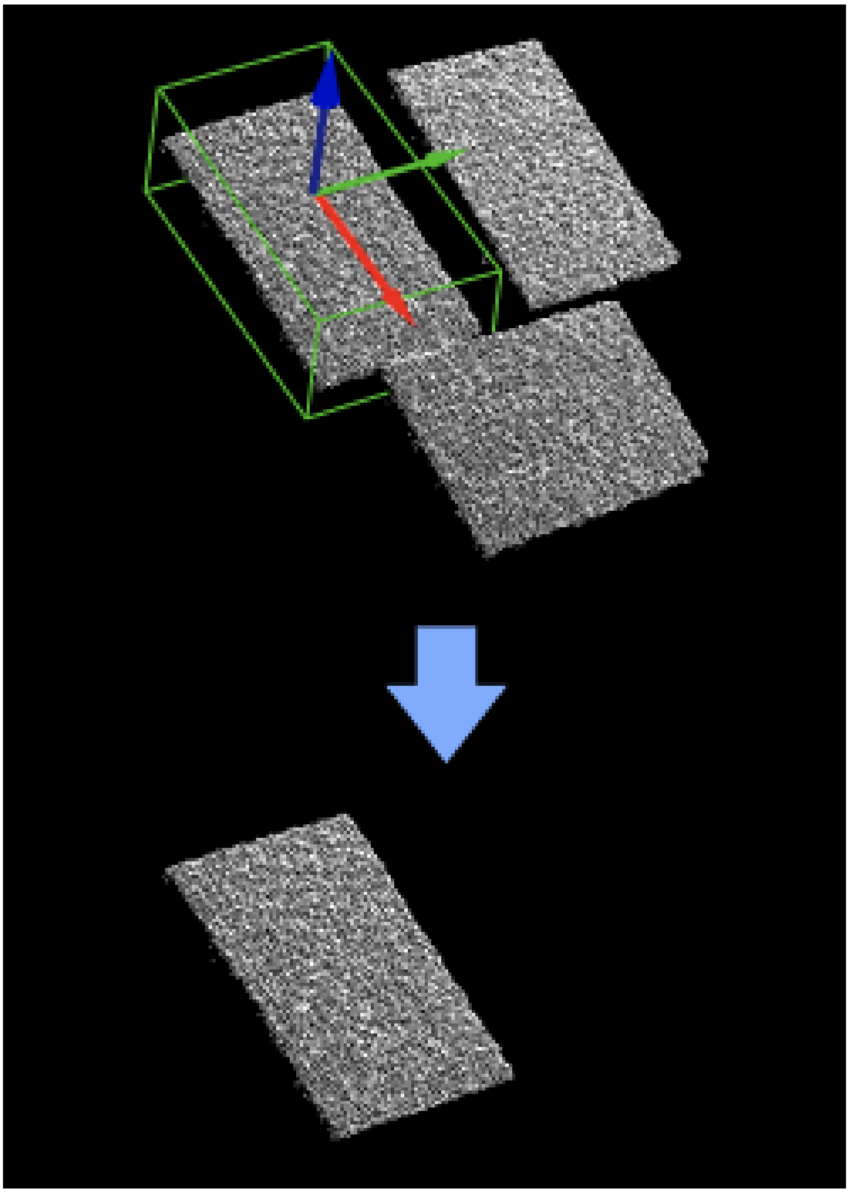
Input and Output

- **Input:** none.
- **Output:** none.

4.3.148. Extract 3D Points in Cuboid

Function

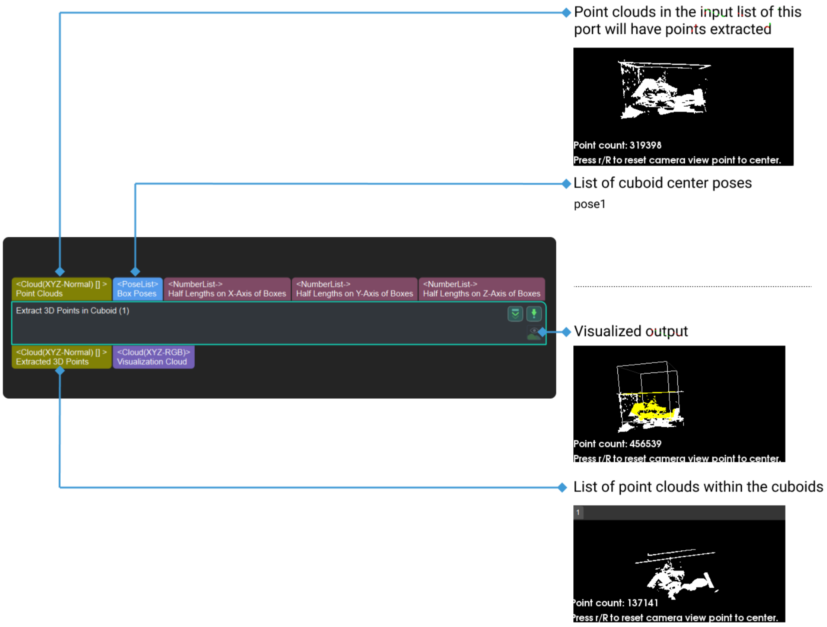
Extract point clouds inside specified cuboids and discard the point clouds outside the cuboids.



Usage Scenario

A general point cloud extraction Step. No fixed usage scenarios. The dimensions of the cuboids can be input from the port or set by parameters.

Input and Output


Point clouds in the input list of this port will have points extracted
Point count: 319398
Press r/R to reset camera view point to center.
List of cuboid center poses
pose1
<Cloud(XYZ-Normal) [] >
Point Clouds
<PoseList>
Box Poses
<NumberList->
Half Lengths on X-Axis of Boxes
<NumberList->
Half Lengths on Y-Axis of Boxes
<NumberList->
Half Lengths on Z-Axis of Boxes
Extract 3D Points in Cuboid (1)
<Cloud(XYZ-Normal) [] >
Extracted 3D Points
<Cloud(XYZ-RGB)>
Visualization Cloud
Visualized output
Point count: 456539
Press r/R to reset camera view point to center.
List of point clouds within the cuboids
1
Point count: 137141
Press r/R to reset camera view point to center.

Parameter Description

Box Dimensions’ Settings

Half Length of Box on X-Axis

Default value: 100.000

Instructions: This parameter takes the half length of the box on the X-axis in the world reference frame, corresponding to **halfX** in the figure below. The unit is millimeters.

Half Length of Box on Y-Axis

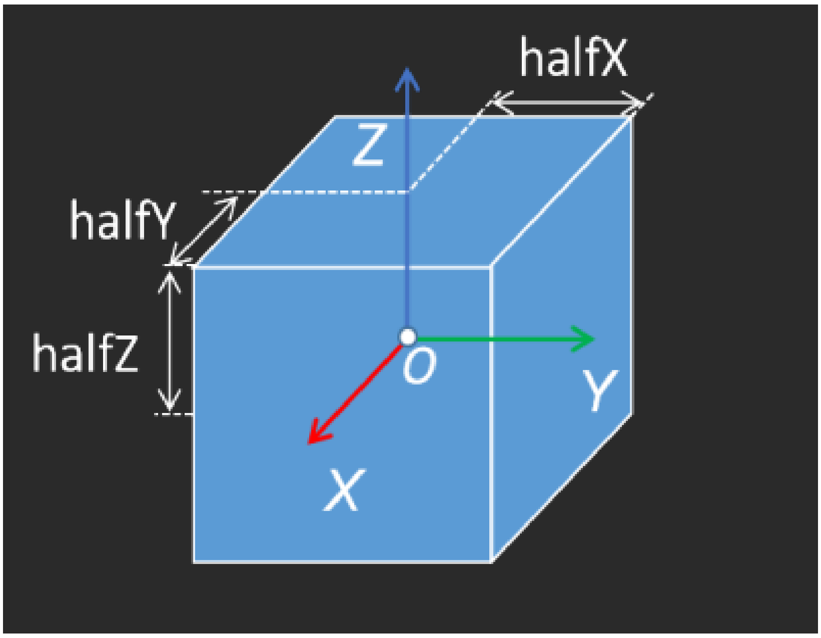
Default value: 100.000

Instructions: This parameter takes the half length of the box on the Y-axis in the world reference frame, corresponding to **halfY** in the figure below. The unit is millimeters.

Half Length of Box on Z-Axis

Default value: 100.000

Instructions: This parameter takes the half length of the box on the Z-axis in the world reference frame, corresponding to **halfZ** in the figure below. The unit is millimeters.



Usage Settings of Input Clouds

Default value: OneToOne

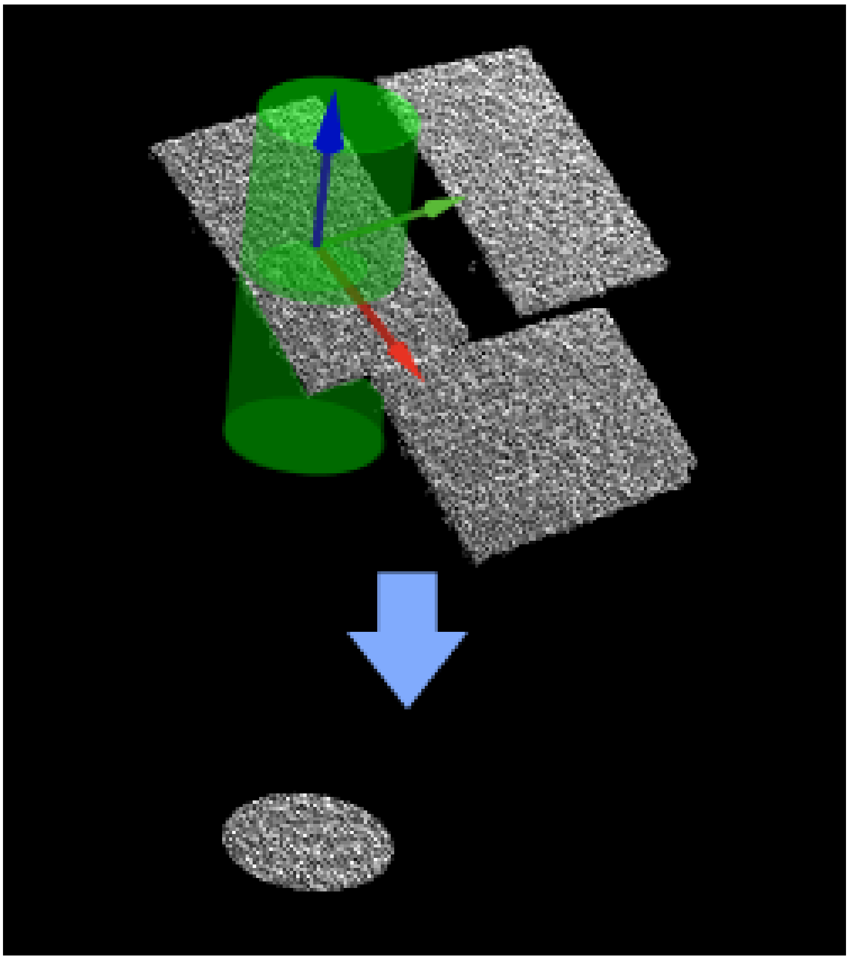
Value list: OneToOne, FirstToAll, and AllToFirst.

Instructions: Select different values to use input clouds.

4.3.149. Extract 3D Points in Cylinder

Function

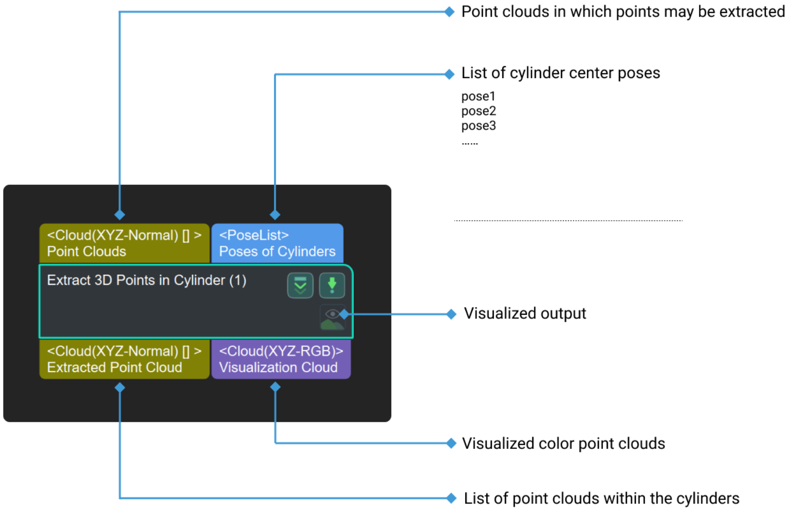
Extract point clouds inside specified cylinders. The point clouds outside the cylinders will be discarded.



Usage Scenario

A general point cloud extraction Step. No fixed usage scenarios.

Input and Output



Point clouds in which points may be extracted

List of cylinder center poses

pose1
pose2
pose3
……

Visualized output

Visualized color point clouds

List of point clouds within the cylinders

Parameter Description

Shapes Type

Default value: Ring.

List of values: Cylinder, Ring, and Sector.

Instructions: 3D points inside different shapes can be extracted according to actual needs. Once a shape is specified, 3D points inside this shape will be extracted.

Z Direction Properties

Min Z Value

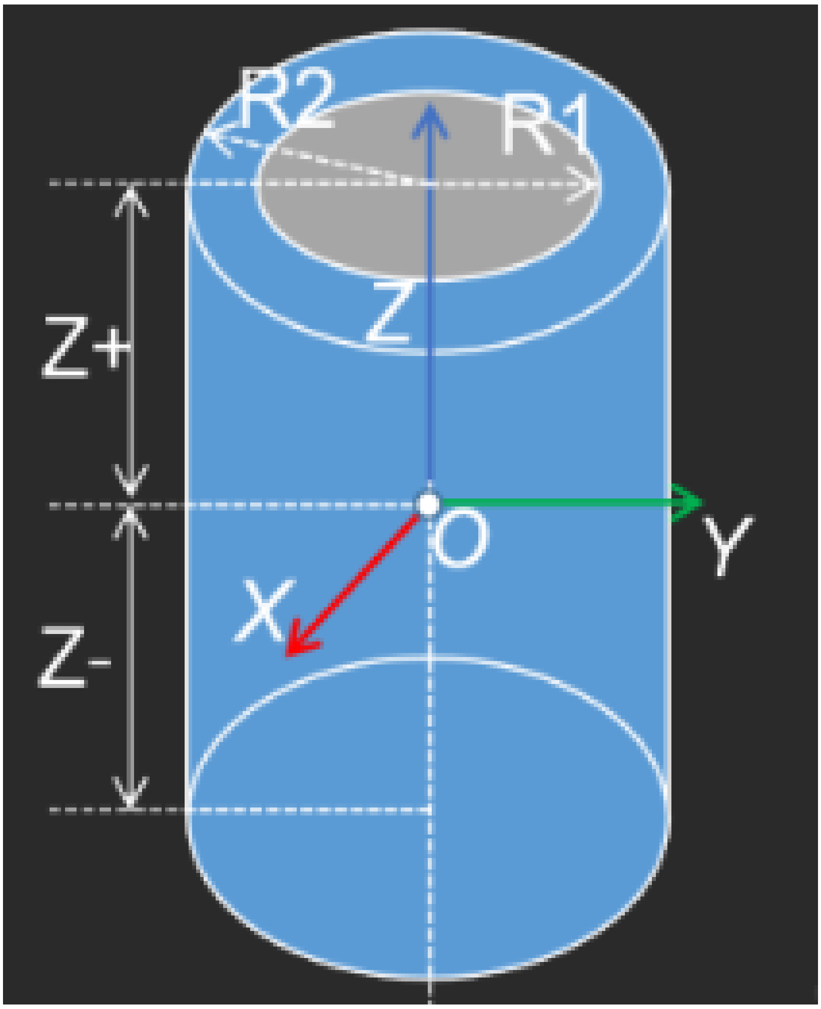
Default value: -100.000 mm

Instruction: This parameter is used to set the lower limit of distance threshold along Z axis of the reference frame, corresponding to **Z-** in the figure below. The unit is millimeters.

Max Z Value

Default value: 100.000 mm

Instruction: This parameter is used to set the upper limit of distance threshold along Z axis of the reference frame, corresponding to **Z+** in the figure below. The unit is millimeters.



Ring and Sector Radius Properties

Inner Radius

Default value: 50.000 mm

Instruction: The minimum distance on XY-plane. It corresponds to **R1** in the figure above. The unit is millimeters. To obtain a solid cylinder, this parameter needs to be set as **0**.

Outer Radius

Default value: 100.000 mm

Instruction: The maximum distance on XY-plane. It corresponds to **R2** in the figure above. The unit is millimeters.

Sector Angle Properties

The following two parameters appear after choosing **Sector** in **Shape Type**, and they are used to configure the shape of the sector.

Start Angle

Default value: 180°

Instruction: the start angle of the sector on XY-plane.

End Angle

Default value: −180°

Instruction: the end angle of the sector on XY-plane.

Usage Settings for Input Clouds

Check If Object inside Cylinder by Its Point Center

Default value: unselected.

Instruction: If this option is selected, the center of the object point cloud is used to check if it is inside the cylinder.

Input Point Cloud Usage

Default setting: All

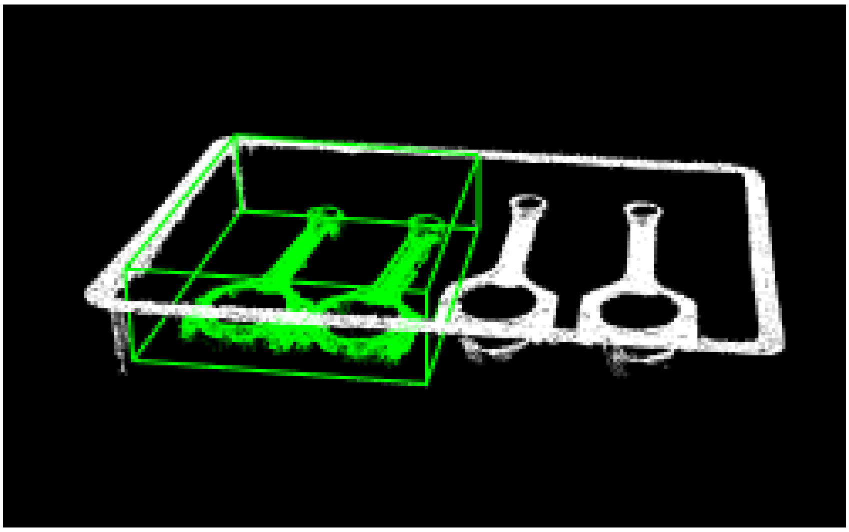
Value list: All, FirstOnly

Instruction: This parameter decides whether to only use the first input point cloud.

4.3.150. Extract 3D Points in 3D ROI

Function

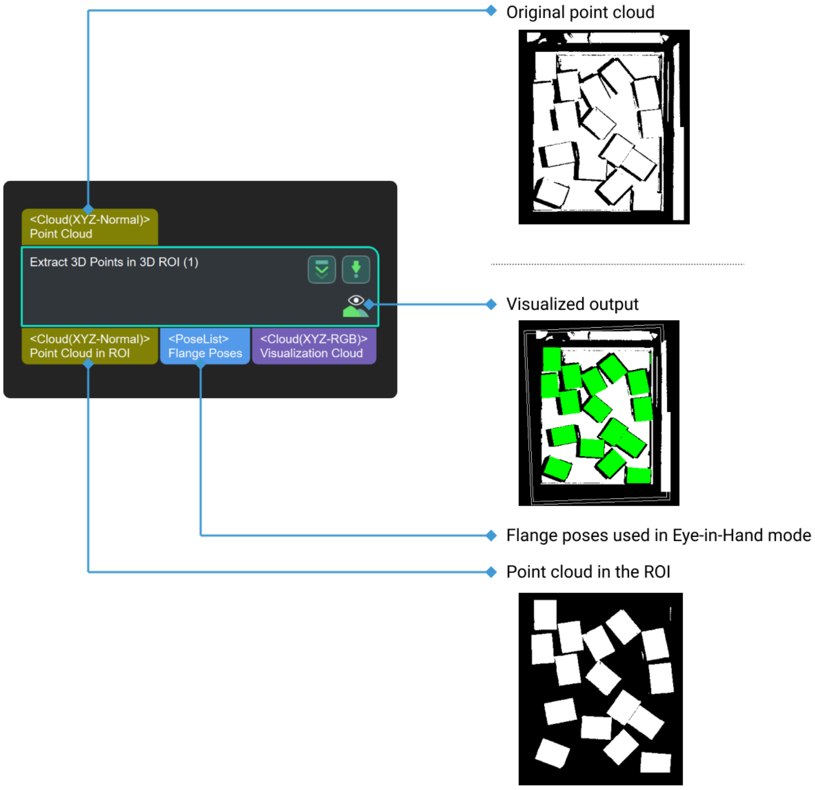
Set a Region of Interest (ROI) in 3D space, and the points within the region will be extracted and points outside the region will be eliminated.



Usage Scenario

This Step enables to extract point cloud of the target object and avoid interference from background and outliers.

Input and Output



Original point cloud

Visualized output

Flange poses used in Eye-in-Hand mode

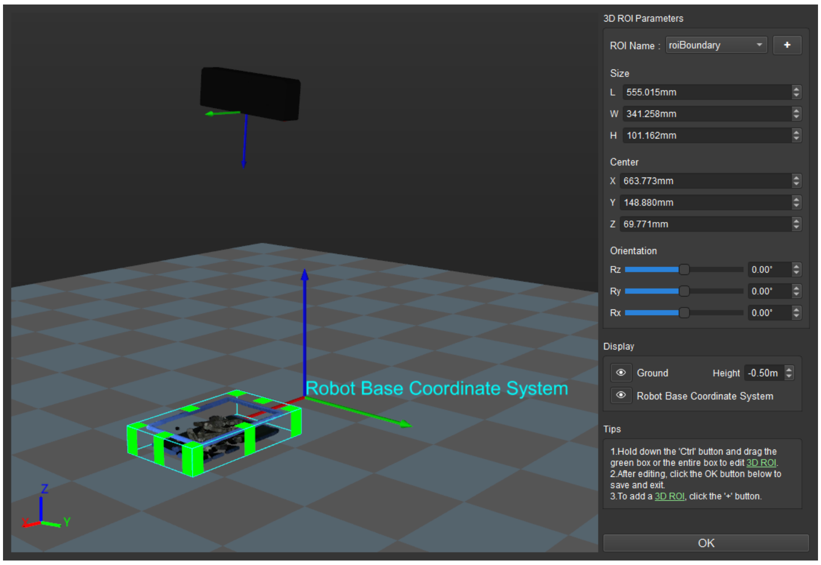
Point cloud in the ROI

Parameter Description

3D ROI Settings

3D ROI Name

Instruction: Click **[Set 3D ROI]** to open the Set ROI window, as shown below. Please refer to Instructions for Setting 3D ROI for detailed instructions.



3D ROI Parameters

ROI Name : roiBoundary +

Size

L 555.015mm

W 341.258mm

H 101.162mm

Center

X 663.773mm

Y 148.880mm

Z 69.771mm

Orientation

Rz 0.00°

Ry 0.00°

Rx 0.00°

Display

Ground Height -0.50m

Robot Base Coordinate System

Tips

1.Hold down the 'Ctrl' button and drag the green box or the entire box to edit 3D ROI.
2.After editing, click the OK button below to save and exit.
3.To add a 3D ROI, click the '+' button.

OK

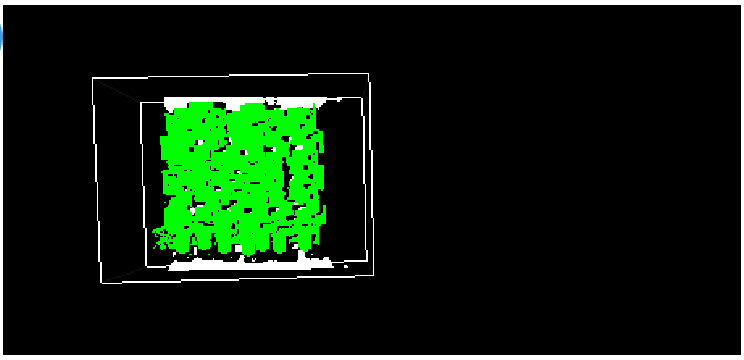
Input Point Cloud Reference Frame

Default value: Camera Frame

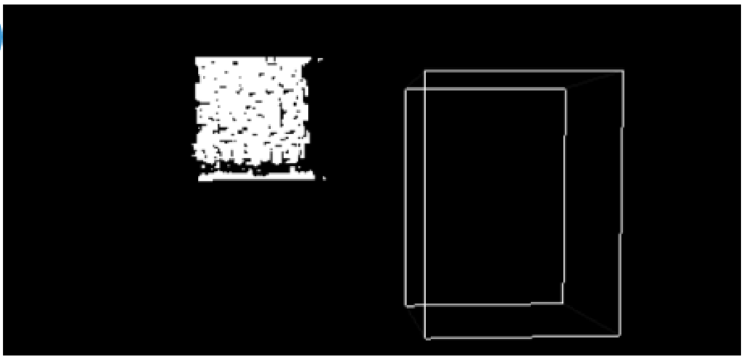
Options: Camera Frame, Robot Frame

Instruction: Change the reference frame in which the input point clouds are. The figure below on the top shows the input point cloud in the camera reference frame, and the figure at the bottom shows the input point cloud in the robot reference frame.

1



2



Empty Status Settings

Min Point Count in 3D ROI

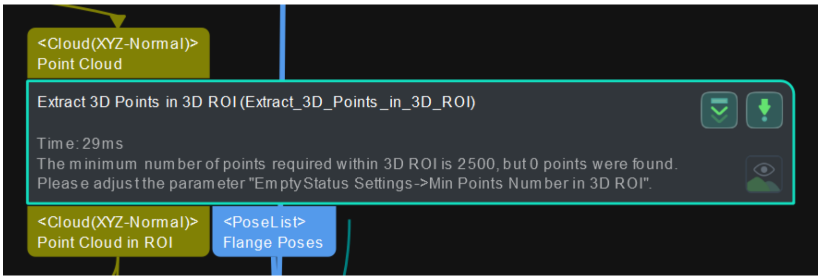
Default value: 0

Description: Adjust the minimum number of the points extracted in the 3D ROI. If the number of points extracted in the 3D ROI is less than this value, no point cloud will be returned.

Send Empty Status of Points in 3D ROI

Default value: Selected

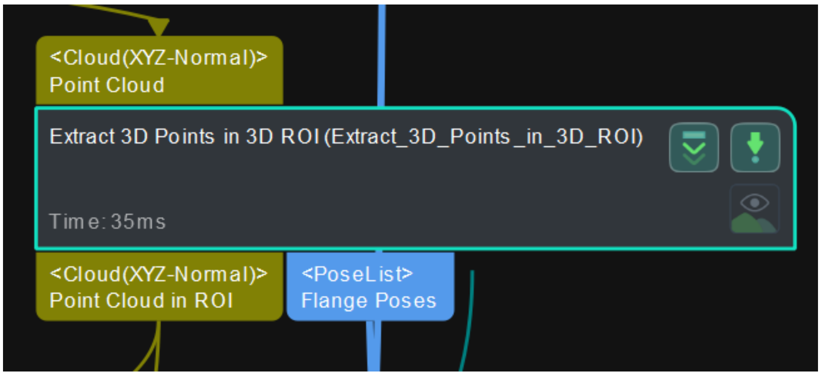
Instruction: Select to send the empty status of points in 3D ROI. By default, if the number of points in the 3D ROI is less than the set value of Min Points Number in 3D ROI, a message as shown below will appear in the Step.



Extract 3D Points in 3D ROI (Extract_3D_Points_in_3D_ROI)

Time: 29ms
The minimum number of points required within 3D ROI is 2500, but 0 points were found.
Please adjust the parameter "EmptyStatus Settings->Min Points Number in 3D ROI".

If this option is not selected, no warning message will appear in the Step.



Extract 3D Points in 3D ROI (Extract_3D_Points_in_3D_ROI)

Time: 35ms

4.3.151. Extract Empty Regions in Depth Map within 3D ROI

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

Given the 3D ROI set in the parameters, this Step outputs regions of zero depth in the ROI.

Usage Scenario

This Step is a custom Step for special scenarios.

Input and Output

- **Input:** Empty regions, i.e., regions of zero depth, of the depth map input by this port will be extracted.

- **Output:** regions with zero depth within the ROI in the depth map.

4.3.152. Fill Holes

Function

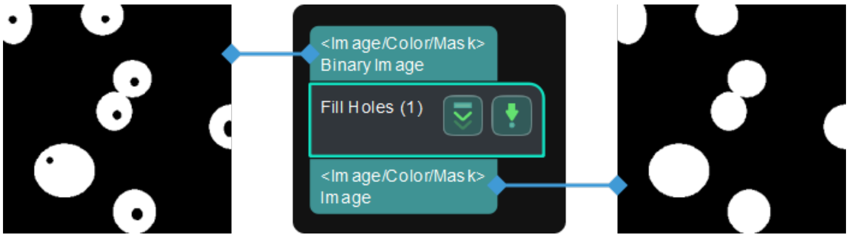
This Step is used to fill the holes in the non-zero pixel regions in the input binary image.

- The binary image means that each pixel on the image has only two possible values. Common binary images include black and white images and monochrome images.
- In this step, the non-zero pixel is a pixel with a pixel value of 255.

Usage Scenario

This step is usually used for image segmentation. It obtains a complete image of the target area and avoids interference caused by missing images in the hole section.

Input and Output



Parameter Description

Holes

Description: This parameter is used to select the type of holes to be filled.

Options: CompleteHoles, AllHoles.

- CompleteHoles: Only the complete holes surrounded by non-zero pixels will be filled. Please refer to Parameter tuning examples for more information.
- AllHoles: Both the complete holes surrounded by non-zero pixels and incomplete holes that are not entirely surrounded by non-zero pixels will be filled. Please refer to the tuning example for the corresponding result.

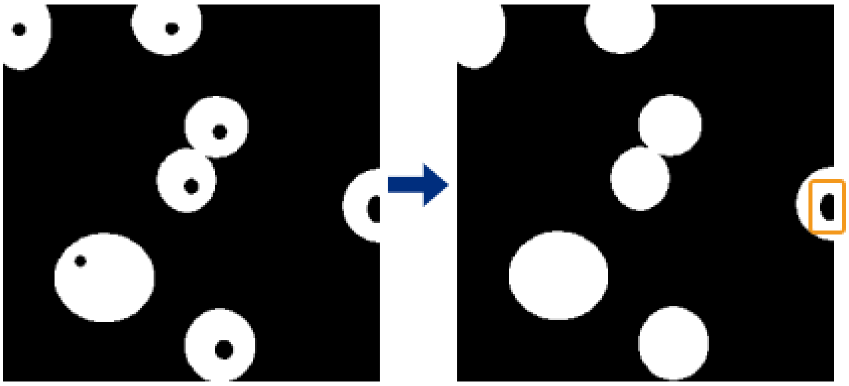
Default setting: CompleteHoles.

Instruction: Select the type of hole according to the actual situation.

Parameter tuning examples

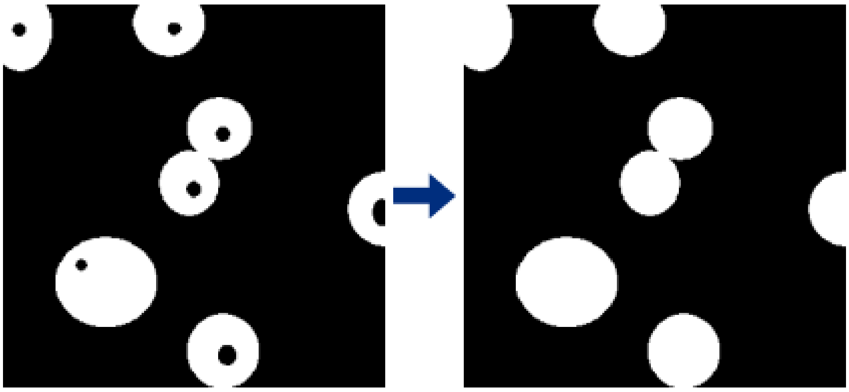
Fill Complete Holes

The effect of filling complete holes is shown in the right figure below. Only complete holes surrounded by non-zero pixels are filled, and incomplete holes at the edge of the image (as shown in the yellow frame) that are not entirely surrounded by non-zero pixels are not filled.



Fill All Holes

The effect of filling all holes is shown in the right figure below. Both complete holes surrounded by non-zero pixels and incomplete holes at the edge of the image that are not entirely surrounded by non-zero pixels are filled.



4.3.153. Filter by Labels

This Step is outdated. Please use the new Step Validate Labels and Output Flags instead. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

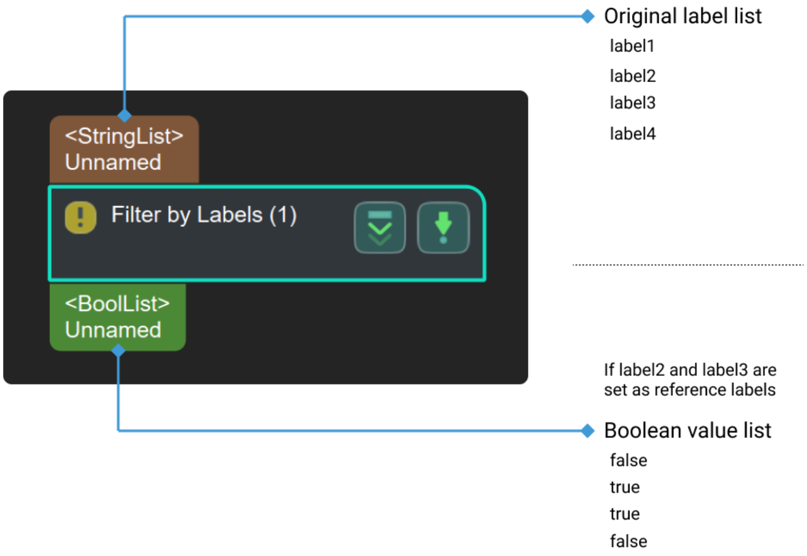
Check if each item in the original label list exists in the reference label list. The result of the check will be output as a list of Booleans.

If an original label is in the reference label, then the corresponding item in the output list is True.

Usage Scenario

It is usually used with Filter. The list of Boolean values output by this Step will be used as the basis for filtering.

Input and Output



4.3.154. Filter Masks by Poses

Function

This Step determines whether the input masks cover the reference poses. Masks covering any reference pose will be retained, and masks covering no poses will be removed.

Usage Scenario

This Step is a custom Step for special scenarios.

Input and Output

- Input:
  1. Mask list input by this port will be filtered.
  2. Poses used to filter the masks.
- Output:
  1. Remaining masks
  2. Poses in mask regions
  3. Poses outside mask regions

4.3.155. Remove Poses outside Bin

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

Given the candidate poses (and corresponding point clouds) and the poses of the two sidewalls of bin, this Step calculates and outputs candidate poses (and their corresponding point clouds) in the bin.

Usage Scenario

This Step is used to remove poses outside the bin.

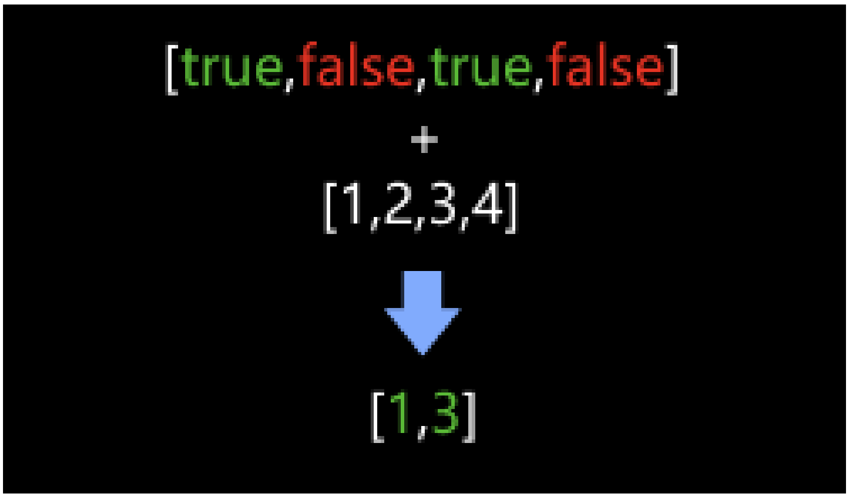
Input and Output

- Input:
  1. Out-of-the-bin poses will be filtered out of the poses input by this port.
  2. Out-of-the-bin parts of point clouds input by this port will be filtered out.
  3. 3D center poses of the two sidewalls of the bin.
- Output:
  1. Poses in the bin
  2. Point clouds corresponding to the poses in the bin
  3. Poses of the 3D diagonal points on the two sidewalls of the bin
  4. Point clouds outside the bin

4.3.156. Filter

Function

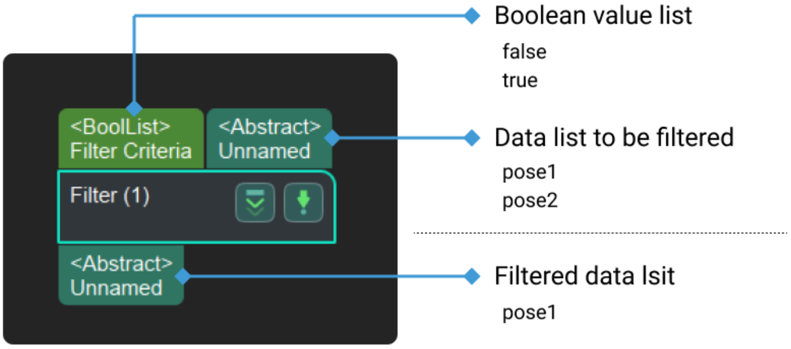
Generic filtering for any array-like variant. It takes a Boolean array of True/False as criteria and outputs the values corresponding to True values in the Boolean array as a new array.



Usage Scenario

This Step filters a data array on the basis of a Boolean list. It usually follows Steps that output Boolean lists, including Dichotomize Values by Threshold, Validate Labels and Output Flags, Validate Poses by Included Angles to Reference Direction, and Validate Point Clouds.

Input and Output



Parameters

Ports Settings

Port Number (1–15)

Default setting: 1

Range: 1–15

Description: Customize the number of the ports.

Operation Layer

Default setting: 0

Range: 0–14

Description: If the operation layer is set to 0, the operation will be performed on every element in the list.

Reverse Bool List

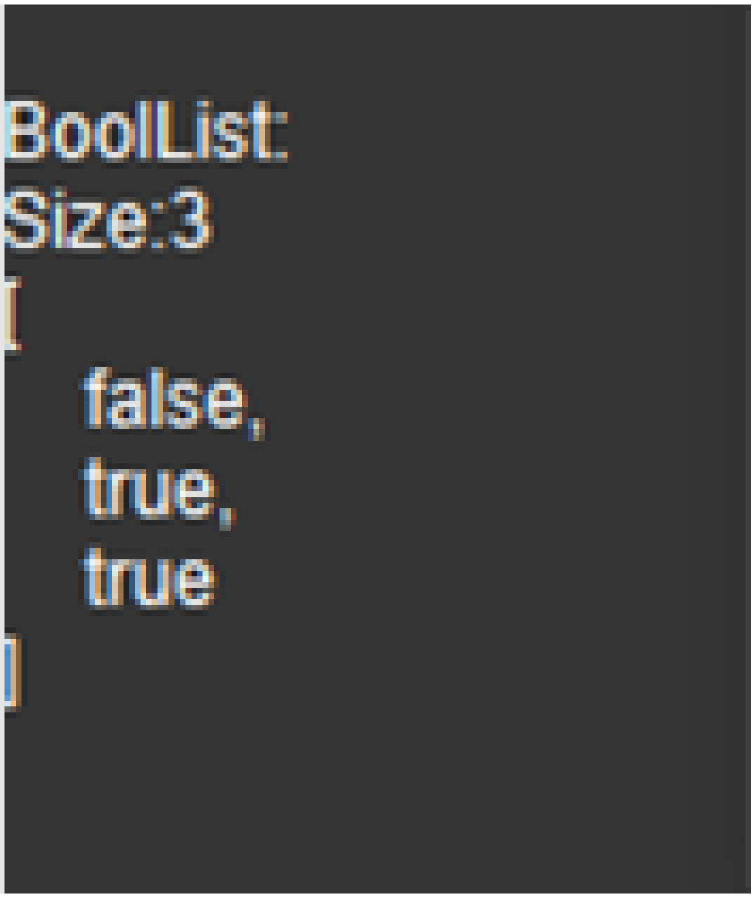
Default setting: Unselected.

Description:

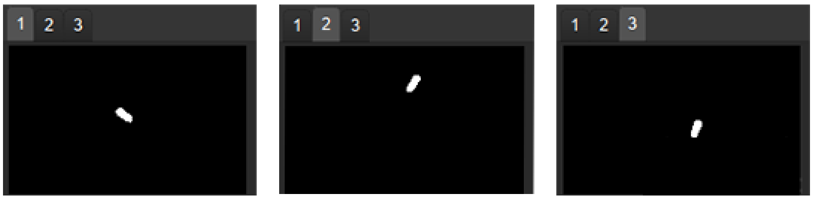
- Unselected: Elements with a value of "False" in the input Boolean list will be filtered.
- Selected: Elements with a value of "True" in the input Boolean list will be filtered.

Examples:

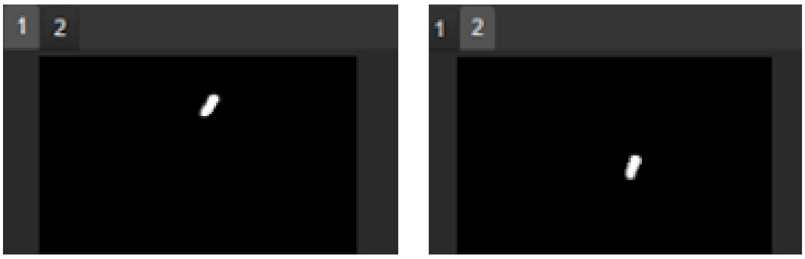
The input data is shown below.



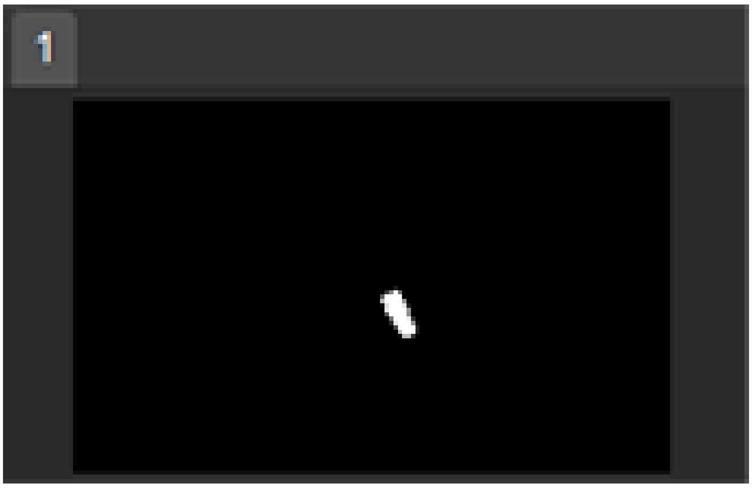
```
BoolList:
Size:3
[
    false,
    true,
    true
]
```



When **Reverse Bool List** is unselected, elements with a value of “False” have been filtered, as shown below.



When **Reverse Bool List** is selected, elements with a value of “True” have been filtered, as shown below.



4.3.157. Find Corners

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step gets contour feature points in a 2D image.

Usage Scenario

This Step is to detect corners of 2D contours.

This Step is outdated. Please use the newer Step Detect Corners.

Input and Output

- Input:
  1. Contour feature points of objects in the 2D image input to this port will be detected.
- Output:
  1. Detected object contour feature points.
  2. Image of visualizing contour feature points.

4.3.158. Find Correspondence between Poses and Offsets

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step finds the correspondence between object poses and pick points, and calculates the offsets between them. The object poses can be adjusted by the offset in the parameter.

Usage Scenario

This Step is a custom Step for special scenarios.

Input and Output

- Input:
  1. Original object poses
  2. Pick points corresponding to objects
  3. Object dimensions
- Output:
  1. Object poses
  2. Offsets from original object poses to pick points
  3. Object dimensions

4.3.159. Find Hole 2D

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step locates the round holes of the objects through the original image and the mask image.

Usage Scenario

This Step is a custom Step for special scenarios.

Input and Output

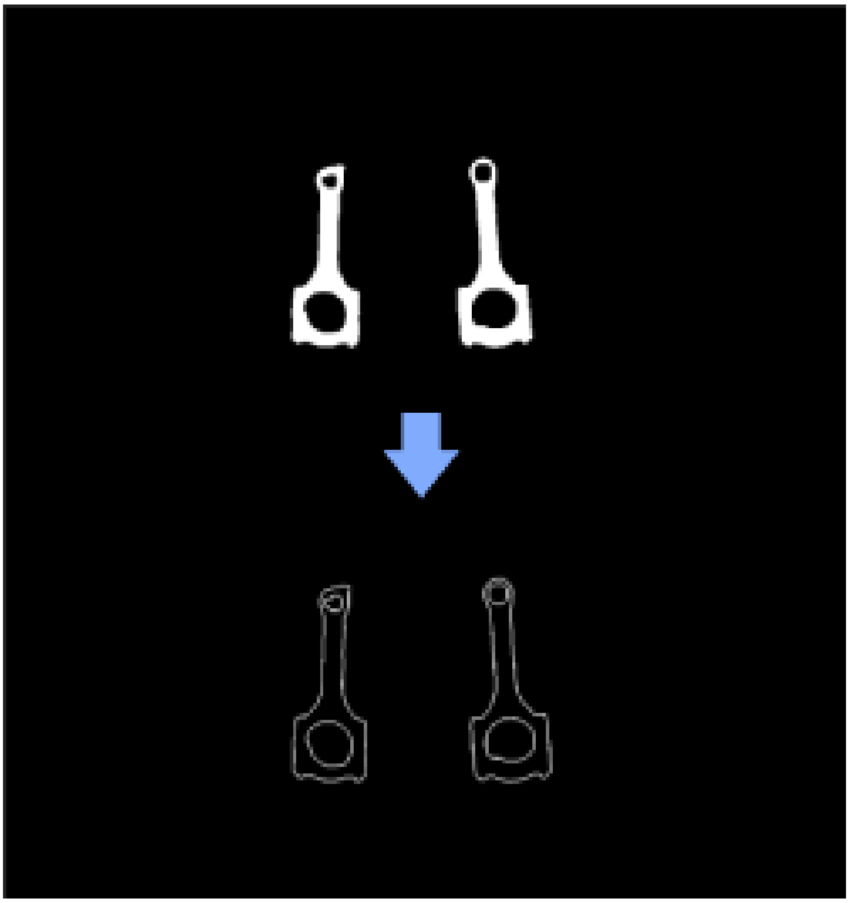
- Input:
  1. Round holes in the mask input to this port will be detected.
  2. Object contour image.
- Output:
  1. Mask of detected object round holes.

4.3.160. Find 2D Contour at Specified Inner-Outer Level

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

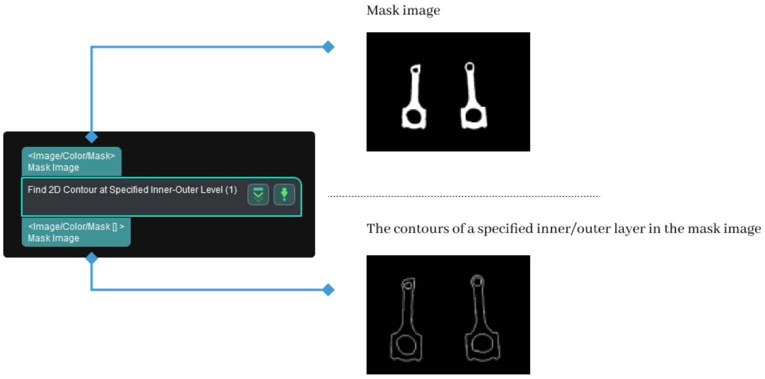
This Step sets the contours of a specified inner/outer layer in the mask image.



Usage Scenario

This Sep is a general 2D image contour extraction Step. It can be used between Steps Orthographic Projection and Orthographic Projection Inverse to obtain object contours in a 3D point cloud.

Input and Output


Mask image
<Image/Color/Mask>
Mask Image
Find 2D Contour at Specified Inner-Outer Level (1)
<Image/Color/Mask [] >
Mask Image
The contours of a specified inner/outer layer in the mask image

4.3.161. Fit Circle

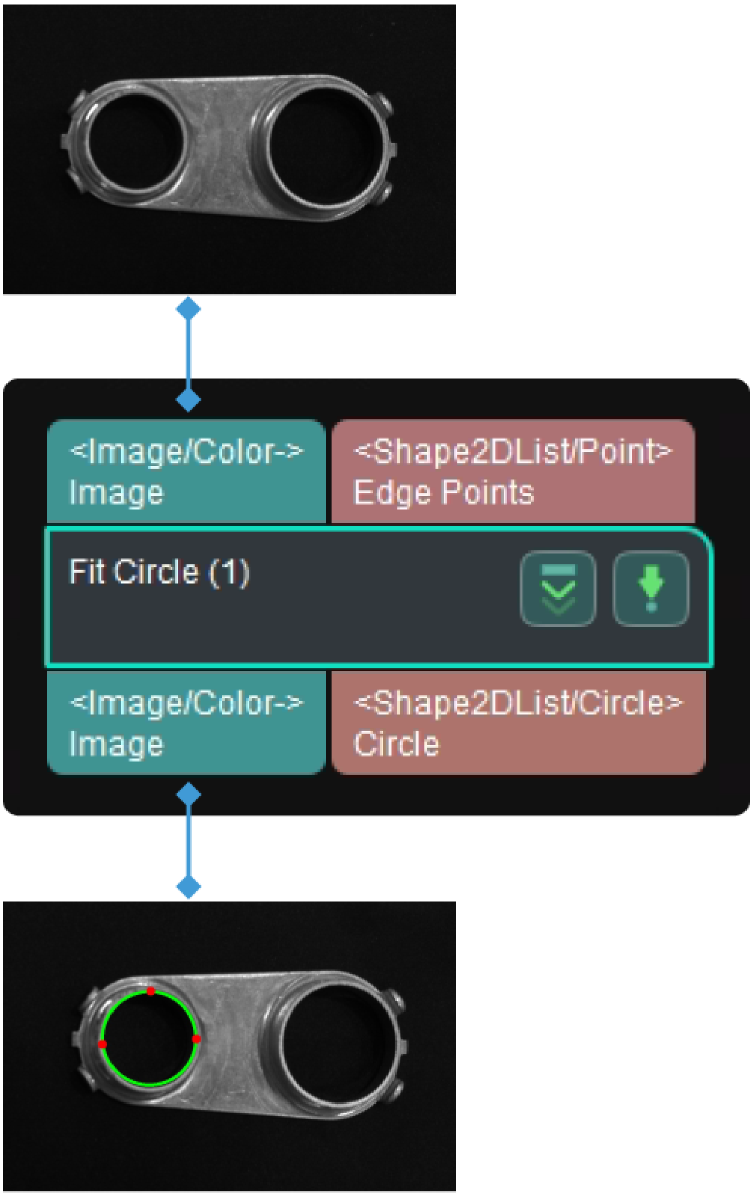
Function

This Step is used to find the fit circle for the points in the input 2D image.

Usage Scenario

This Step is generally used in measurement scenarios.

Input and Output



Parameter Description

Fitting Method

Description: Specify the fitting method.

Value list: Huber, Least Squares.

Default setting: Huber

- Least Squares: Select to use the ordinary least squares method to find the fit circle, and the sum of distances between the points and the fit circle is minimized.
- Huber: Select to use the weighted least squares method to find the fit circle. This method can reduce the impact of outliers on the fit circle.

Instruction: Generally, it is recommended to keep the default setting.

4.3.162. Fit Line

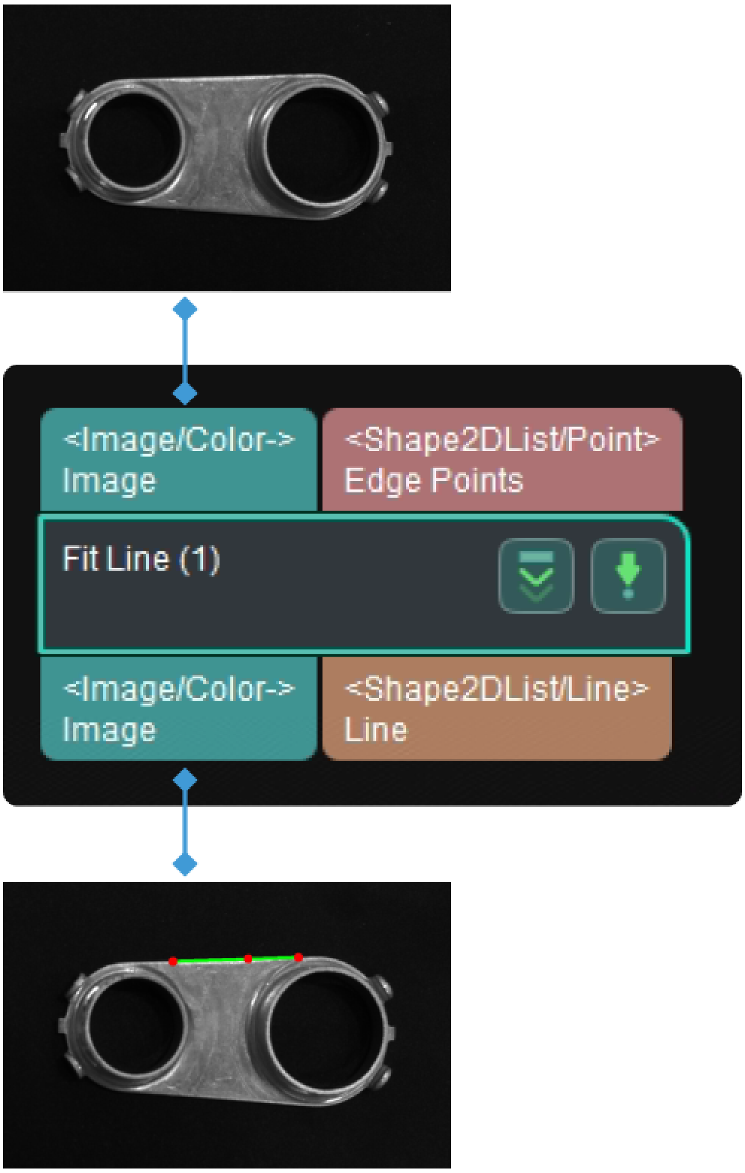
Function

This Step is used to find the fit line for the points in the input 2D image.

Usage Scenario

This Step is generally used in measurement scenarios.

Input and Output



Parameter Description

Fitting Method

Description: Specify the fitting method.

Value list: Huber, Least Squares.

Default setting: Huber

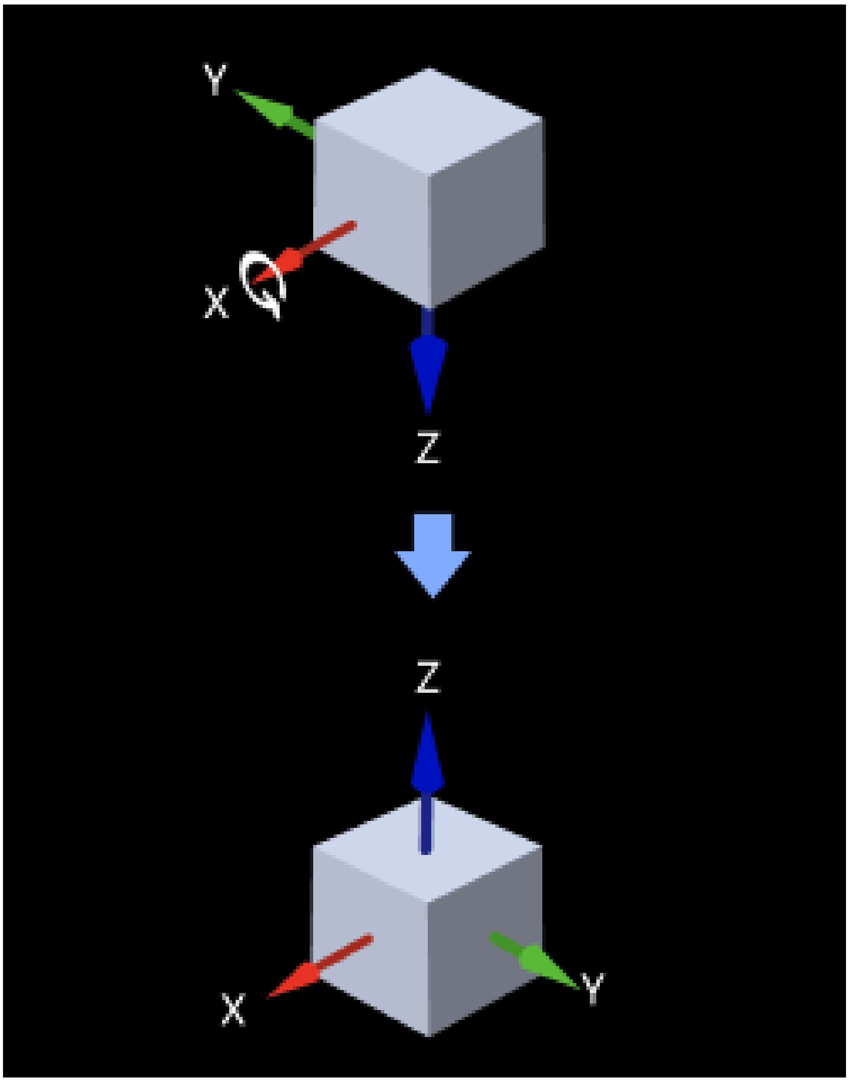
- Least Squares: Select to use the ordinary least squares method to find the fit line, and the sum of distances between the points and the fit line is minimized.
- Huber: Select to use the weighted least squares method to find the fit line. This method can reduce the impact of outliers on the fit line.

Instruction: Generally, it is recommended to keep the default setting.

4.3.163. Flip Poses' Axes

Function

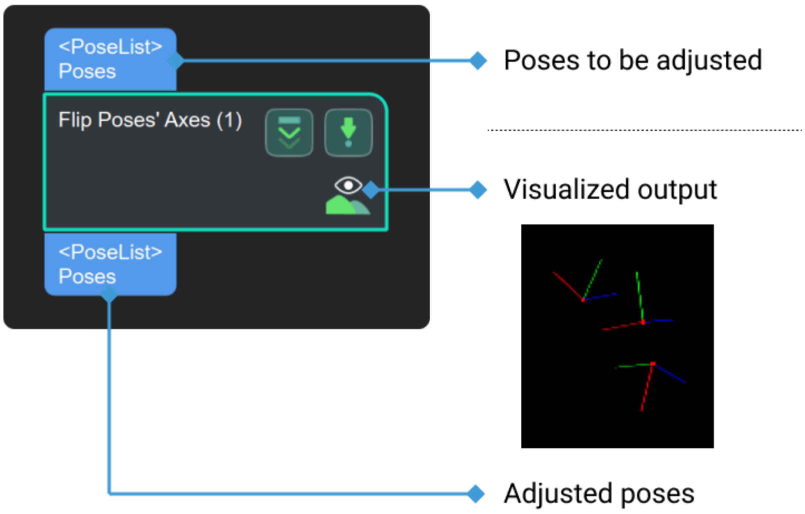
Flip the selected axis to the positive or negative directions around a specified axis of rotation, as shown below.



Usage Scenario

This Step is usually used to flip a certain axis of the poses to a specified direction in the later phase of pose processing.

Input and Output



Parameter Description

Visualization Settings

Pose Type to Visualize

Default setting: After.

Options: Before, After, and All.

Instruction: This parameter decides the type of pose to visualize, including the output flipped poses (default), the input poses to be flipped, and both the input and output poses.

Axis Settings

Axis to Flip

Default setting: Z

Options: X, Y, and Z

Instruction: Select the axis to flip.

Direction Type

Default setting: Negative.

Options: Negative, Positive.
Description: the target direction of the axis to be flipped. For example, if the **Axis to Flip** is selected as **Z**, and the Direction Type is **Positive**, the Z axis of a pose which has an included angle less than 90° with the positive axis in the world reference frame will not be adjusted, while the axis which has an included angle with the positive axis in the world reference greater than 90° will be flipped 180° around the reference axis. Likewise, when the Direction Type is selected as Negative, only axes that have included angles greater than 90° with the negative axis in the world reference frame will be flipped.

Reference Axis to Rotate Around

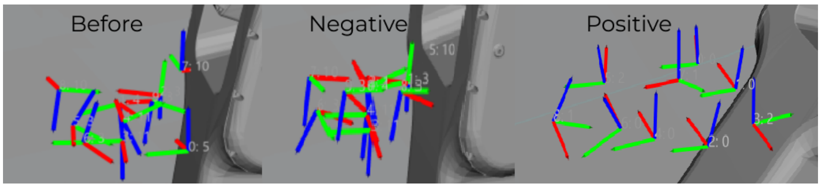
Default setting: X

Options: X, Y, and Z

Description: Select a reference axis to rotate around.

The axis to be flipped and the reference axis cannot be the same one.

Example: The figure below shows an example when Z axis is the axis to be flipped, and X-axis is the reference axis.



4.3.164. From Actual Dimensions to Dimensions in Pixels

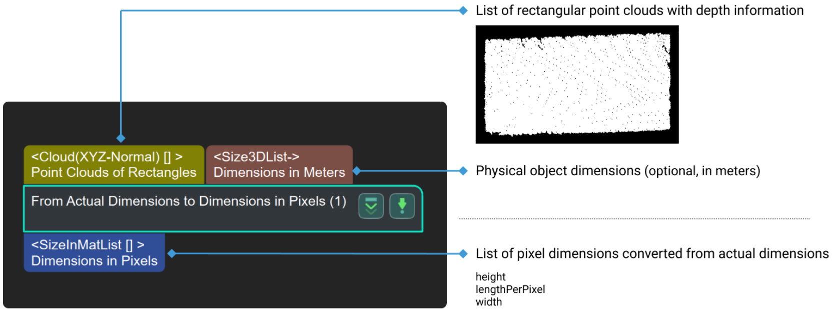
Function

Given the depth information of a rectangular point cloud and its physical dimensions, this Step calculates the dimensions of the corresponding object in pixels.

Usage Scenario

This Step is usually used to provide edge model for the subsequent 2D Matching.

Input and Output


List of rectangular point clouds with depth information
<Cloud(XYZ-Normal) [] >
Point Clouds of Rectangles
<Size3DList->
Dimensions in Meters
Physical object dimensions (optional, in meters)
From Actual Dimensions to Dimensions in Pixels (1)
<SizeInMatList [] >
Dimensions in Pixels
List of pixel dimensions converted from actual dimensions
height
lengthPerPixel
width

Parameters

Correspondence Settings

Correspondence Type

Parameter description: This parameter is used to select the type of correspondence relation between the input size and the actual point cloud size.

Value list: OneActualSizeWithAllPointCloud, OneActualSizeWithOnePointCloud.

- OneActualSizeWithAllPointCloud: The first element of the input list of dimensions corresponds to all point clouds.
- OneActualSizeWithOnePointCloud: Each element in the input list of dimensions corresponds to one actual point cloud.

Default value: OneActualSizeWithAllPointCloud

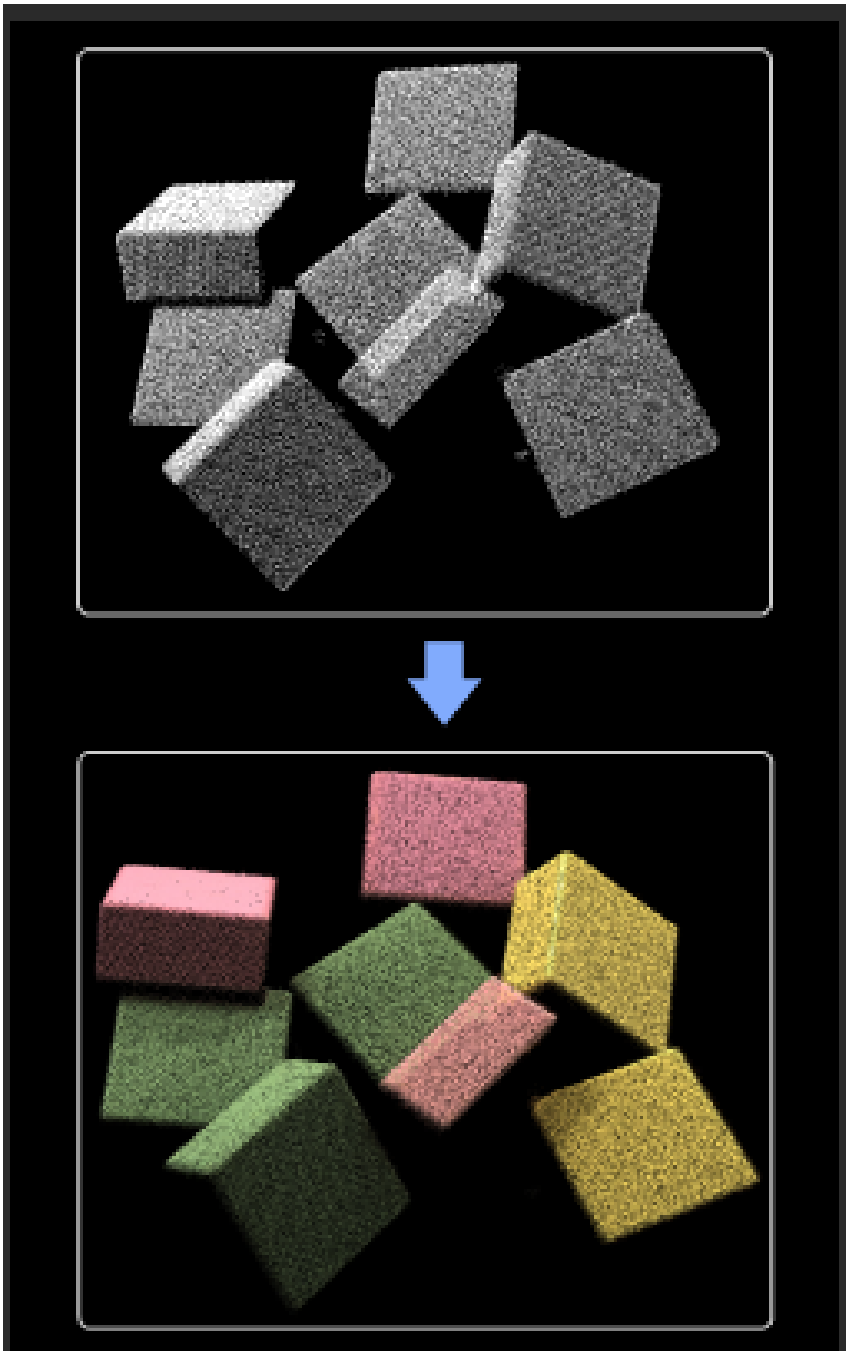
Tuning recommendation: Please set the parameter according to the actual requirement.

4.3.165. From Cloud (XYZ-Normal) to Cloud (XYZ-RGB)

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

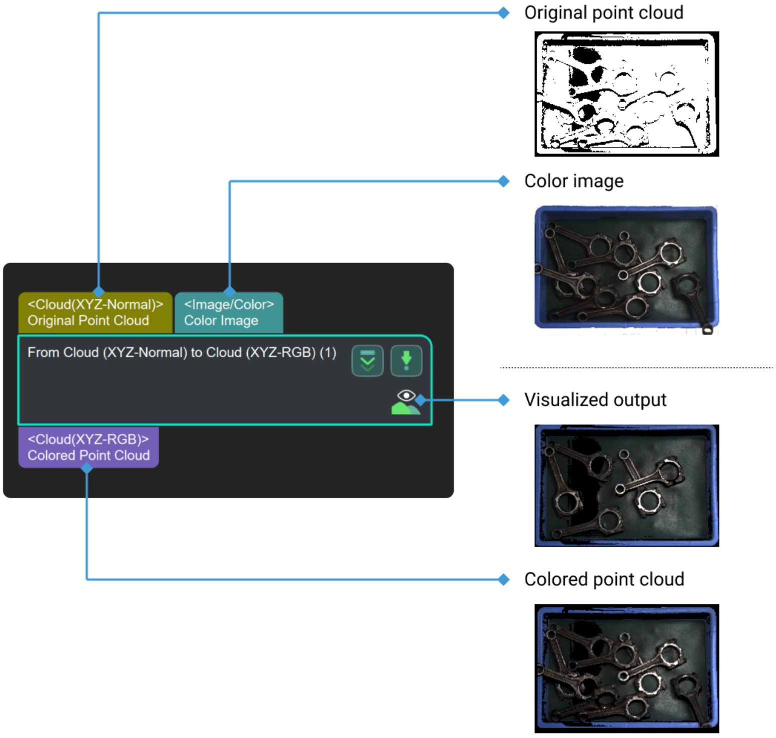
This Step converts a colorless point cloud with normals to a colored point cloud using a 2D color image.



Usage Scenario

This Step is used to view the status of colored point clouds in a simulated scene, or for other usage scenarios that require visualization of colored point clouds.

Input and Output


Original point cloud
Color image
<Cloud(XYZ-Normal)> Original Point Cloud
<Image/Color> Color Image
From Cloud (XYZ-Normal) to Cloud (XYZ-RGB) (1)
Visualized output
<Cloud(XYZ-RGB)> Colored Point Cloud
Colored point cloud

4.3.166. From Cloud (XYZ-RGB) to Color Image

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step converts a color point cloud to a color image.

Usage Scenario

This Step is a custom Step for special scenarios.

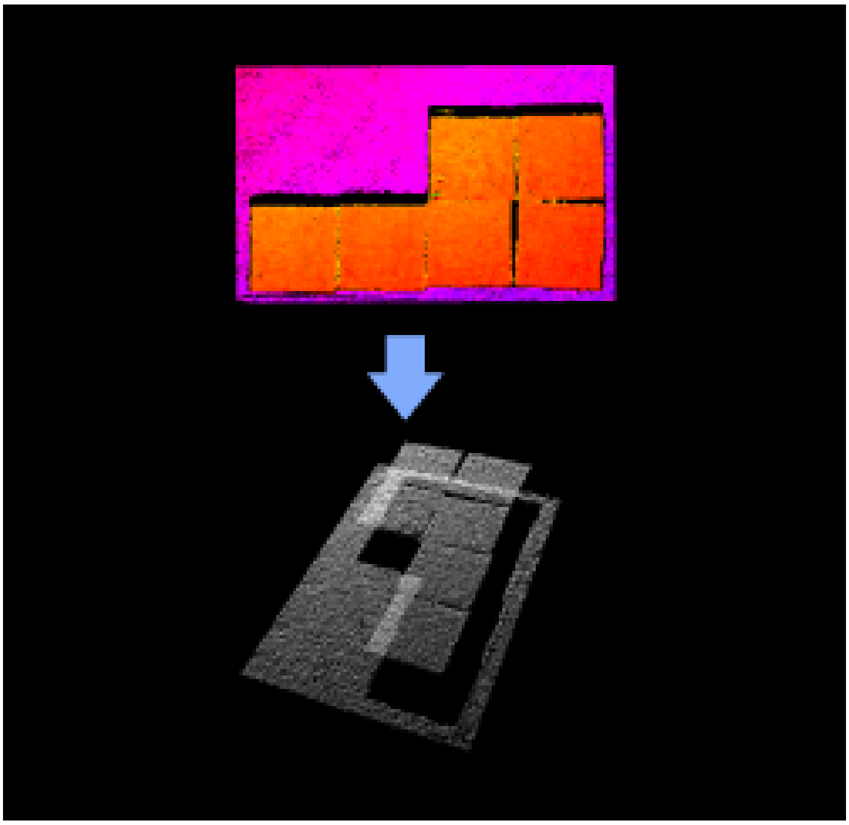
Input and Output

- **Input:** color point cloud.
- **Output:** color image.

4.3.167. From Depth Map to Point Cloud

Function

This Step generates a point cloud from a depth map and a corresponding color image.



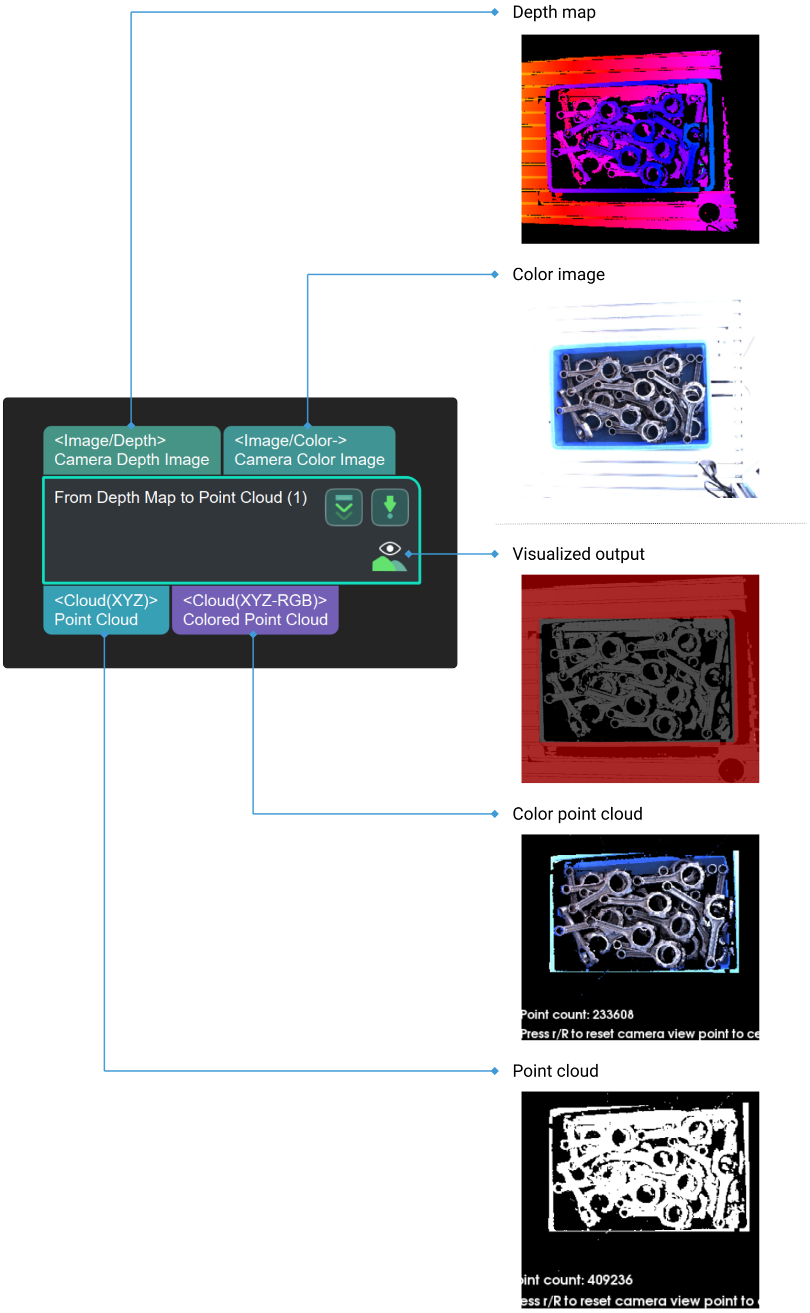
Comparison: The Step Capture Images from Camera can also directly output point clouds and color point clouds. However, the Step **From Depth Map to Point Cloud** features a higher processing speed in this regard.

You can set an ROI in the depth map in this Step to lower the quantity of unnecessary point clouds and thus improve the processing speed.

Usage Scenario

This Step is usually connected after the Step Capture Images from Camera and can generate point clouds from depth maps. You can set an ROI in the depth map in this Step to improve the processing speed.

Input and Output

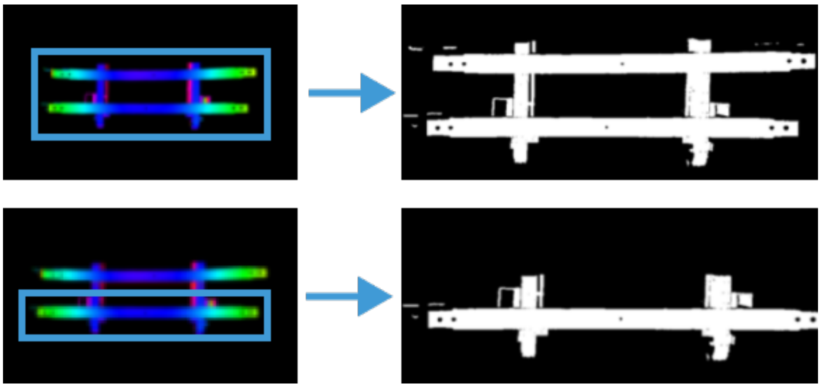


Parameter Description

Depth Roi File Name

Description: Select an ROI on the depth map.

Example: Select different ROIs.



The image on the left shows ROI in an depth map, and the image on the right shows the output point cloud.

Instructions:

1. Click the [Set 2D ROI] on the right of the parameter **Depth Roi File Name** to open the **Set ROI** window.
2. Select an ROI in the window.
3. Click [OK] to save the setting. A depth_image_roi.json file will be saved to the root directory of the project.

Background Removal Settings

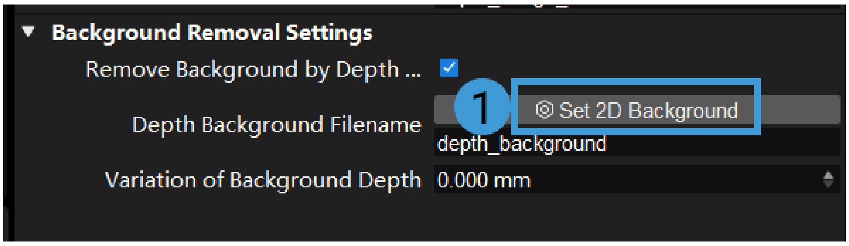
Remove Background by Depth Image

Default setting: Unselected. Description: The following parameters will only be displayed only when **Remove Background by Depth Image** is selected.

Depth Background Filename

Instructions: Please follow the instructions as follows to set a depth map of the background.

Click the [Set 2D Background] button on the right of the **Depth Background Filename** to open the **Set Background** window.



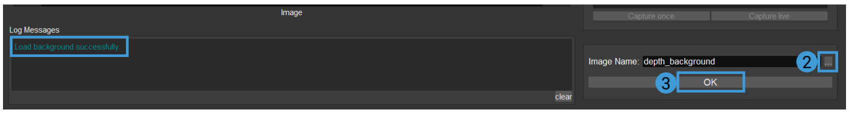
▼ Background Removal Settings

Remove Background by Depth ... ☑

Depth Background Filename: depth_background

Variation of Background Depth: 0.000 mm

Click […] and then select an image of the background that you have already prepared. Once the message Load background successfully message appears in the **Log Messages** panel, click [OK].



Log Messages

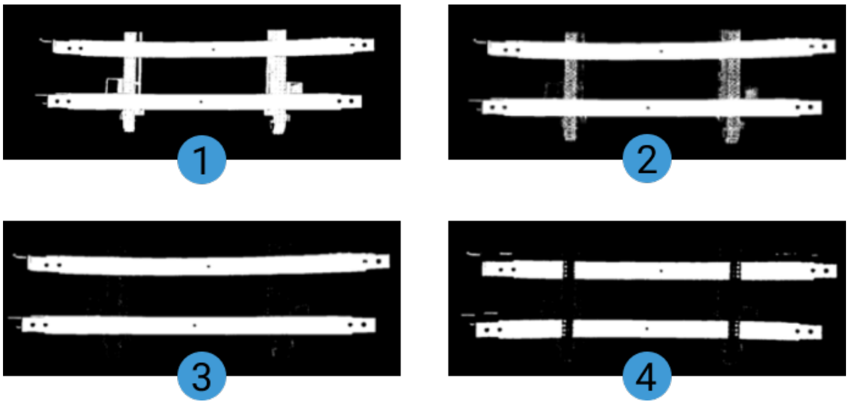
Load background successfully.

Image Name: depth_background

> You can also connect to the real camera and capture an image of the background on site.

Variation of Background Depth

Default setting: 10.000 mm Description: The input depth map will be compared with the depth map of the background, and points with a depth difference from the background lower than this value will be eliminated. Example: In the figure below, the background of image 1 has not been removed; backgrounds of images 2, 3, and 4 have been removed, and the Variation of Background Depth are set to 1 mm, 10 mm, and 30 mm, respectively.



If the value of background depth fluctuation is too small, the background cannot be removed completely. If the value of background depth fluctuation is too large, the point cloud of the object may be removed as well. Therefore, the **Variation of Background Depth should be set to a proper value according to actual requirements on site, and 10 mm is recommended to use**.

4.3.168. Convert Disparity Image to Depth Map

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step converts disparity map to depth map.

Usage Scenario

This Step is for developers and is used for adjustment in the passive stereo camera mode. Regular users do not need to use this Step in general application scenarios.

Input and Output

- Input:
  1. Disparity image input to this port will be converted to a depth map.
  2. Mapping matrix from camera disparity values to depth values (should be a 4 × 4 matrix).
- Output:
  1. Depth map converted from disparity image.

4.3.169. From NumberList to Size3DList

Function

This Step combines three number lists into a Size3DList.

Sample Scenario

This Step is a general data type conversion and composition Step. No fixed usage scenarios.

Input and Output

- Input:
  1. List of X values as Size3DList elements
  2. List of Y values as Size3DList elements
  3. List of Z values as Size3DList elements
- Output:
  1. List of Size3D data consisting of the X, Y, Z values input

4.3.170. From Numbers to Variants

Function

This Step converts data of type NumberList to type VariantList.

Sample Scenario

This Step is a general data type conversion Step. This Step has no fixed usage scenarios.

Input and Output

- **Input:** data of type NumberList.
- **Output:** Variant data obtained from conversion.

4.3.171. From PoseList to Matrix4D

Function

This Step converts data of type PoseList to type Matrix4D.

Sample Scenario

This Step is a general data type conversion Step. This Step has no fixed usage scenarios.

Input and Output

- **Input:** Pose list data input to this port will be converted into a Matrix4D matrix.
- **Output:** Matrix4D matrix transformed from the input pose list.

4.3.172. From PoseList to PoseLists

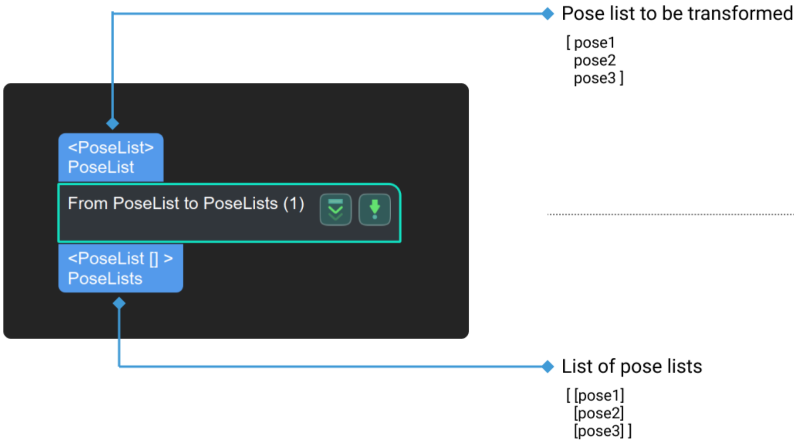
Function

Change each pose in the pose list to a one-element pose list.

Usage Scenario

A general data structure transformation Step. For instance, it is used between the Step 3D Coarse Matching and the Step 3D Fine Matching or between the Step Calc Poses and Dimensions from Planar Point Clouds and the Step Filter.

Input and Output



4.3.173. Convert Quaternions to Euler Angles in Poses

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

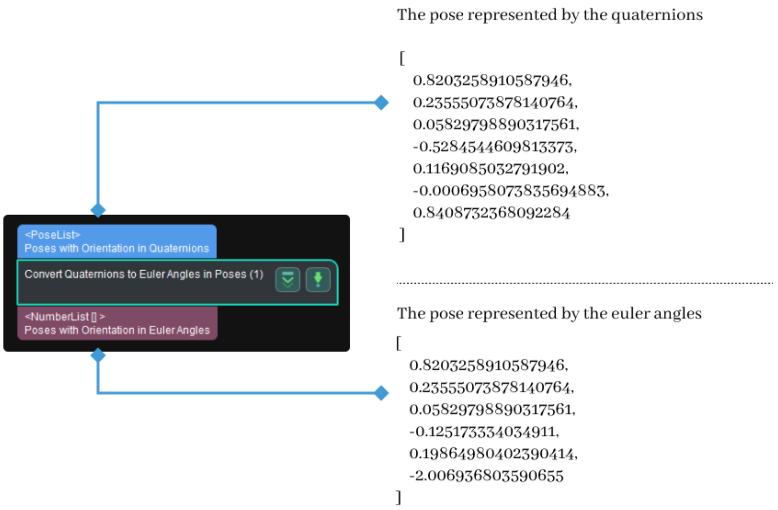
Function

Change the data type of poses from quaternions to Euler angles.

Usage Scenario

A general pose data format conversion Step. No fixed usage scenarios.

Input and Output



<PoseList>
Poses with Orientation in Quaternions
Convert Quaternions to Euler Angles in Poses (1)
<NumberList[]>
Poses with Orientation in Euler Angles

```
The pose represented by the quaternions

[
   0.8203258910587946,
   0.23555073878140764,
   0.05829798890317561,
   -0.5284544609813373,
   0.1169085032791902,
   -0.0006958073835694883,
   0.8408732368092284
]
```

```
The pose represented by the euler angles
[
  0.8203258910587946,
  0.23555073878140764,
  0.05829798890317561,
  -0.1251733334034911,
  0.19864980402390414,
  -2.006936803590655
]
```

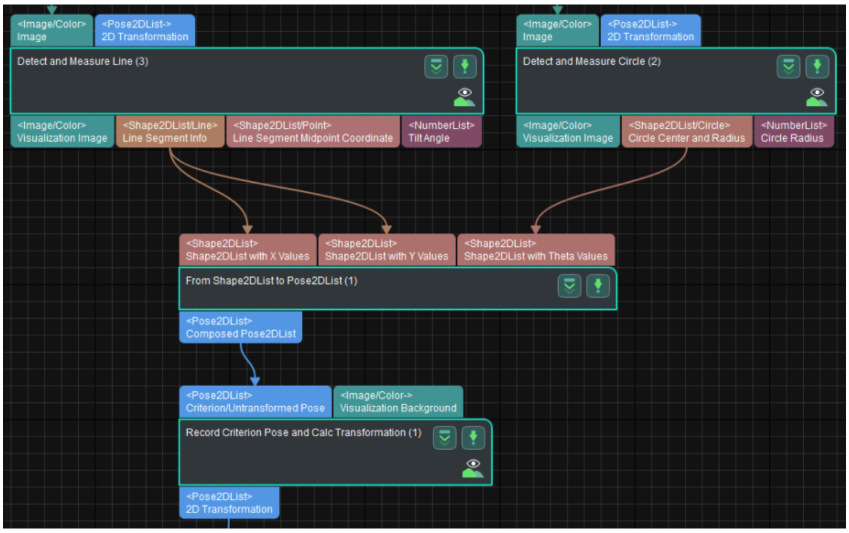
4.3.174. From Shape2DList to Pose2DList

Function

This Step is used to assemble a new 2D pose list by taking the lists of X values, Y values, and Theta values (Theta represents the tilt angle) from the three input 2D shape information lists respectively.

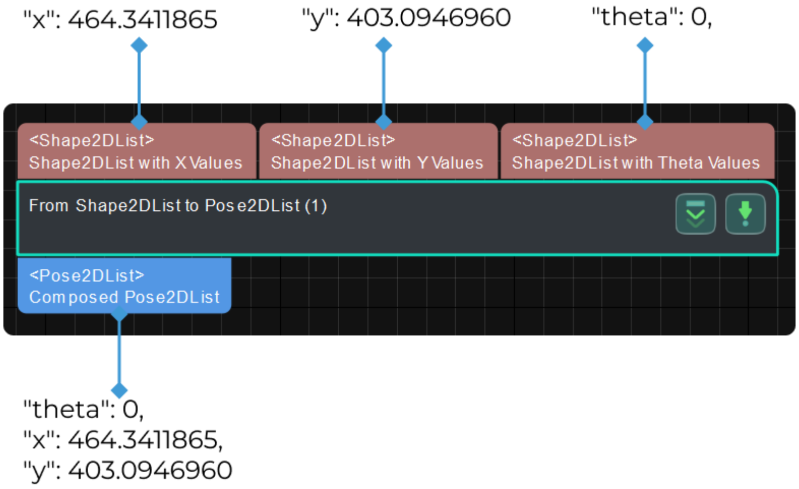
Usage Scenario

This Step is usually preceded by the Detect and Measure Line (to obtain the tilt angle of the line, Theta) and the Detect and Measure Circle Step (to obtain the X and Y coordinates of the center of the circle) to output more accurate 2D poses of the objects in the image, and then followed by the Record Criterion Pose and Calc Transformation Step to accurately align a group of objects in the image.



The above Step combination is usually applied after the Template Matching Step to make the results of the Template Matching Step more accurate. If the objects in each image are already nearly aligned, the Template Matching Step is not necessary anymore.

Input and Output



Parameter Description

None.

4.3.175. From Variant to Variants

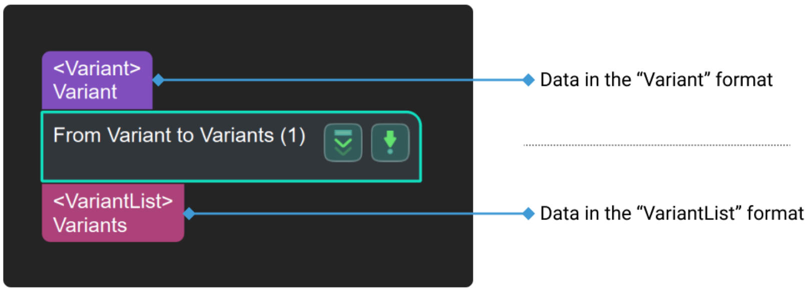
Function

Convert input from the “Variant” format to the “VariantList” format.

Usage Scenario

A general data type conversion Step. For example, it can be used between the Step Count Elements in Specified Dimension in Data List and the Step From Variants to Numbers to convert the number of a list’s elements into another list for output.

Input and Output


<Variant>
Variant
From Variant to Variants (1)
<VariantList>
Variants
Data in the “Variant” format
Data in the “VariantList” format

4.3.176. From Variants to Labels

Function

This Step converts data of any field data type to labels (currently supported data types: DoubleList, QtVariantList).

Usage Scenario

This Step is a general data type conversion Step. This Step has no fixed usage scenarios.

Input and Output

- **Input:** The data input to this port will be converted to labels.
- **Output:** Labels converted from the input.

4.3.177. From Variants to Numbers

Function

This Step converts the type of the input data from VariantList to NumberList.

Usage Scenario

This Step is a general data type conversion Step. This Step has no fixed usage scenarios.

Input and Output

- **Input:** VariantList data.
- **Output:** NumberList data converted from the input.

4.3.178. From Variants to Variant

Function

This Step converts data of type VariantList to type Variant. If the input list contains multiple elements, the first element will be used.

Usage Scenario

This Step is a general data type conversion Step. This Step has no fixed usage scenarios.

Input and Output

- **Input:** VariantList data input to this port will be converted to Variant data.
- **Output:** Variant data obtained from conversion.

4.3.179. From 3D Poses to 2D Poses

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step is used to convert 3D Poses to 2D Poses. Camera intrinsic parameters should also be considered.

Usage Scenario

This Step is a general pose transformation Step. This Step has no fixed usage scenarios.

Input and Output

- **Input:** 3D poses input to this port will be transformed.
- **Output:** 2D poses obtained by transformation.

4.3.180. Generate Cloud Wall

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step duplicates the input point cloud along the specified direction to generate a point cloud wall.

Usage Scenario

This Step is a custom Step for special scenarios.

Input and Output

- Input:
  1. Point cloud input to this port will be duplicated to generate a point cloud wall.
  2. Height of point cloud wall to generate.
  3. Specified direction of duplication (optional, if this port is not used, the direction set in the parameter will be used).
- Output:
  1. Point cloud wall generated

4.3.181. Generate Cylinder Point Cloud Model

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step generates a cylindrical point cloud. The shape of the cylinder can be set by inputting values or setting in the parameters.

Usage Scenario

This Step is used to automatically generate a point cloud in the shape of a cylinder, which will be used as a model later.

Input and Output

- Input:

1. Diameter of the cylindrical point cloud model to generate (optional; if this port is not used, the parameter settings will be used).
2. Length of the cylindrical point cloud model to generate (optional; if this port is not used, the parameter settings will be used).

- Output:
  1. Generated cylindrical point cloud model

4.3.182. Generate Discrete Poses Revolving around Reference Pose

> Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step duplicates the initial pose by rotating around the reference pose to generate a list of poses.

Usage Scenario

This Step is a general pose generating Step. No fixed usage scenarios.

Input and Output

- Input:
  1. Initial pose input to this port will be rotated to form a set of poses.
  2. Pose input to this port will be used as the reference pose for rotation.
- Output:
  1. A list of poses formed by duplicating the initial pose while rotating it around the reference pose.

4.3.183. Generate Pick Points

> Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

Generate pick points using a 2D template image.

Usage Scenario

Generally used in steel plate picking applications; based on the defined suction cup shape and dimensions, this Step will generate the pick points for the flat steel plate objects.

Input and Output

- Input:
  1. Object template binary mask input to this port will be used to generate pick points.
  2. Suction cup area binary mask input by this port will be used to generate pick points (if the overlap area between the object template and the suction cup area mask is less than the

set threshold, no pick points will be generated).

- **Output:**
  1. Generated pick points

4.3.184. Generate Point Cloud Model

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step converts a 2D image to a 3D point cloud.

Usage Scenario

This Step is usually used for steel plate picking, to convert 2D image into 3D point cloud, to use as a model.

Input and Output

- **Input:** 2D image used to generate the template.
- **Output:** 3D point cloud model generated from 2D image.

4.3.185. Generate Ring Point Cloud

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step generates a ring-shaped point cloud. The dimensions of the ring can be set in the parameters.

Usage Scenario

This Step is used to automatically generate a point cloud in the shape of a ring to use as a model later.

Input and Output

- **Input:** none.
- **Output:** ring-shaped point cloud (with normals) generated.

4.3.186. Generate Pose/Offset

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step generates pose offsets for adjusting the original object poses. Only offsets within the object range will be preserved.

Usage Scenario

This Step is a custom Step for special scenarios.

Input and Output

- Input:
  1. Original poses input to this port will be offset.
  2. List of object dimensions corresponding to poses.
- Output:
  1. List of poses after offsetting.
  2. List of offsets (depending on parameter settings).
  3. Object index list corresponding to poses (one object may have multiple poses, i.e., pick points).

4.3.187. Generate Rect Path

> Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step generates a rectangular path based on the input object point cloud.

Usage Scenario

This Step is typically used in edge gluing applications for objects such as rectangular hatches/glass.

Input and Output

- Input:
  1. Point cloud of the object
  2. Scene point cloud with the object
  3. Starting target of the rectangular path
- Output:
  1. List of targets on the rectangular path
  2. 2D orthographic projection image of the path

4.3.188. Generate Spiral Path

> Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step generates spiral path targets based on the input point cloud.

Usage Scenario

This Step is usually used for automatic spraying on flat objects.

Input and Output

- Input:
  1. Surface of the object in the point cloud input to this port will have a spiral path generated.
  2. Depth map corresponding to the point cloud.
  3. Robot tool center point in the flange reference frame.
- Output:
  1. Outside-to-in targets of generated path (except for the end target).
  2. End target of generated path.
  3. Orthographic projection 2D image of the path

4.3.189. Generate Test Image

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step generates an image of the specified format and size.

Usage Scenario

This Step is used by developers for testing. Regular users do not need to use this Step in general application scenarios.

Input and Output

- **Input:** none.
- **Output:** image generated.

4.3.190. Generate Test Point Cloud

Function

This Step generates point cloud data for testing other features.

Usage Scenario

This Step is used by developers for testing. Regular users do not need to use this Step in general application scenarios.

Input and Output

- **Input:** none.
- **Output:** point cloud generated.

4.3.191. Generate Path from Contour

Function

This Step generates a path from the input edge point cloud.

Usage Scenario

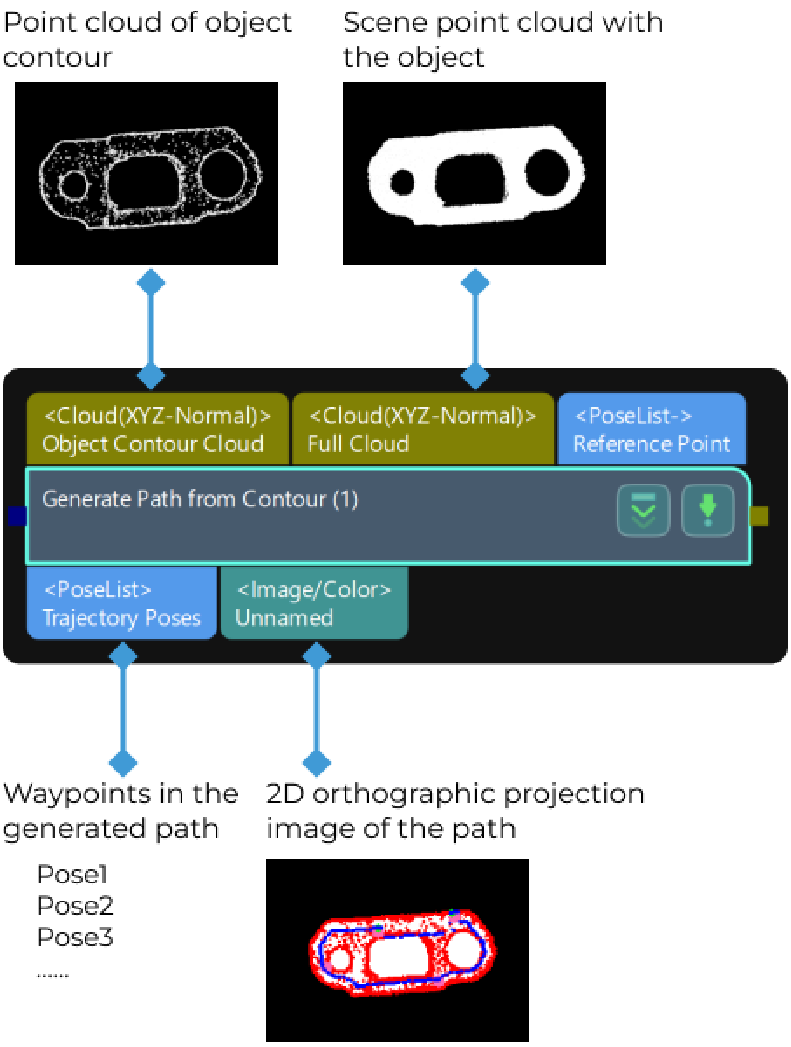
This Step is typically used in edge gluing applications for objects such as rectangular hatches/glass.

In the actual application process, due to factors such as the camera mounting angle and lighting, point clouds may lack necessary parts or have excessive noise. This can lead to changes or deviations in the starting point of the generated path, so it is recommended to use this Step only as an aid for path generation.

For objects of the same type, you can obtain a stable path by following the steps below.

1. Use the object’s point cloud in conjunction with this Step to generate an initial path, and adjust the waypoints in the path.
2. Take the object’s point cloud as a point cloud model, use the adjusted waypoints as multiple pick points for the point cloud model, and set the geometric center point in the point cloud model.
3. Perform 3D matching on the object using the point cloud model and geometric center point, obtain the object’s pick points, and then use the “Map to Multiple Pick Points” Step to obtain multiple pick points for the object, which make up a stable path altogether.

Input and Output


Point cloud of object contour
Scene point cloud with the object
<Cloud(XYZ-Normal)> Object Contour Cloud
<Cloud(XYZ-Normal)> Full Cloud
<PoseList-> Reference Point
Generate Path from Contour (1)
<PoseList> Trajectory Poses
<Image/Color> Unnamed
Waypoints in the generated path
Pose1
Pose2
Pose3
......
2D orthographic projection image of the path

Parameter Description

This Step includes four parameter groups, and the descriptions are as follows:

- General Params: This parameter group is used for general settings. Please adjust the parameters according to the actual requirement.
- Smooth Points: When the edge point cloud of the object is jagged, you may need to adjust this parameter group to make the edge point cloud smoother.
- Special Side Erode: This parameter group is designed for special applications, and it does not need to be adjusted in general.
- Find Corners: When there are obvious corners on the edge point cloud, you will need to adjust this parameter group.

General Params

Mask Closing Kernel Size

Description: This parameter is used to set the size of the kernel when a closing operation is applied to the image mask. The unit is pixels.

Default value: 5 px

Points Sample Size

Description: This parameter is used to specify the interval between waypoints in the generated path. The unit is pixels. The larger the value, the larger the interval between waypoints. Please refer to the tuning example for the application result.

Default value: 10 px

Edge Zoom Type

Description: This parameter is used to select the method to generate a path based on the edge point cloud.

Value list: ByErode, ByVertical

- ByErode: Generates a path by erosion operation.
- ByVertical: Generates a path by calculating vertical lines of the point tangent lines on the edge point cloud.

Default value: ByErode.

+ Tuning recommendation: It is recommended to select **ByErode**.

Erode Kernel Size

Description: When the **Edge Zoom Type** is set to **ByErode**, this parameter is used to determine the distance between the path and the object edge and whether the path is within or outside the object edge. The unit is pixels. When the parameter value is positive, the generated path will be within the object edge. When the parameter value is negative, the generated path will be outside the object edge. Please refer to the tuning example for the corresponding result.

Default value: 15 px

Vertical Move Real Dis

Description: When the **Edge Zoom Type** is set to **ByVertical**, this parameter is used to determine the distance between the path and the object edge and whether the path is within or outside the object edge. The unit is millimeters. When the parameter value is positive, the generated path will be within the object edge. When the parameter value is negative, the generated path will be outside the object edge. Please refer to the tuning example for the corresponding result.

Default value: 8.000 mm

Smooth Points

Use Gaussian Filter

Description: When the edge point cloud is jagged, this parameter is used to determine whether to use the Gaussian filter to make the edge point cloud smoother.

Default Value: Unselected.

Tuning recommendation: Please select this option according to the actual requirement.

Gaussian Filter Radius

Description: This parameter is used to specify the radius of the Gaussian filter. The unit is pixels.

Default value: 20 px

Tuning recommendation: It is recommended to use the default value.

Gaussian Filter Sigma

Description: This parameter is used to specify the sigma of the Gaussian filter.

Default value: 5.0000

Tuning recommendation: It is recommended to use the default value.

Special Side Erode

This parameter group is designed for special applications, and it does not need to be adjusted in general.

Find Corners

Sample Size for Corner

Description: This parameter is used to set the coefficient of the downsampling on the edge point cloud. The unit is pixels. Downsampling facilitates the detection of the corner points in the edge point cloud.

Default value: 3

Quality Level

Description: This parameter is used to detect the corner points of the edge point cloud. It ensures that waypoints are located in the corners no matter how large the waypoint interval is.

Default value: 0.0100

Tuning recommendation: It is recommended to use the default value.

Min Distance

Description: This parameter is used to specify the minimum distance (in pixels) between corners when detecting the corner points.

Default value: 50.0000

Max Corners

Description: This parameter is used to specify the maximum number of corners of the object.

Default value: 4

Block Size

Description: This parameter is used to specify the size of the detection area. When the corner points are not obvious enough or the edge point cloud is too smooth, you can increase the value.

Default value: 10

Use Harris Detector

Description: This parameter is used to determine whether to use Harris corner detector. Once this option is selected, you will need to set the **K** value.

Default Value: Unselected.

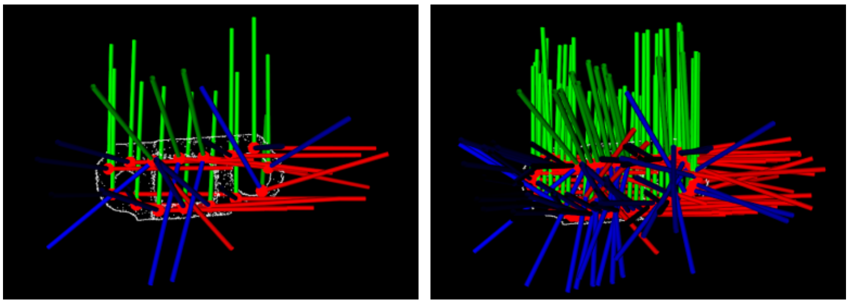
Tuning recommendation: In general, this option does not need to be used.

Tuning Examples

Points Sample Size

When **Points Sample Size** is set to **20** and **5**, respectively, the waypoints in the generated path are shown below.

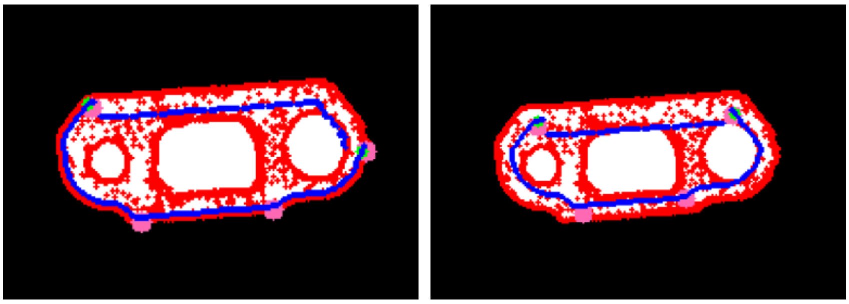
For the left figure, **Points Sample Size** is set to **20**, and the waypoints are relatively sparse; for the right figure, **Points Sample Size** is set to **5**, and the waypoints are relatively dense.



Erode Kernel Size

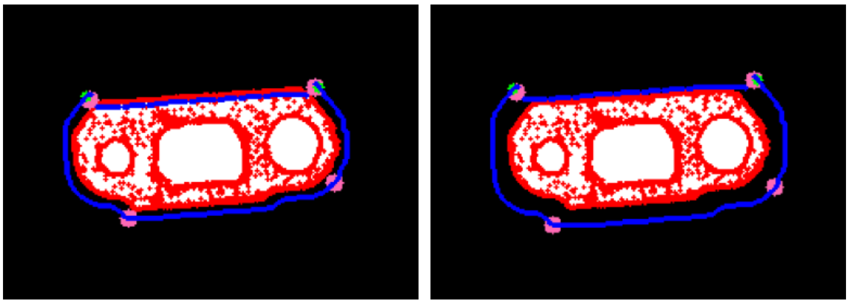
When **Erode Kernel Size** is set to **5** and **15**, respectively, the generated paths are shown below.

For the left figure, **Erode Kernel Size** is set to **5**; for the right figure, **Erode Kernel Size** is set to **15**. As the positive value increases, the generated path will shrink toward the inside of the object's edge.



When **Erode Kernel Size** is set to **-5** and **-20**, respectively, the generated paths are shown below.

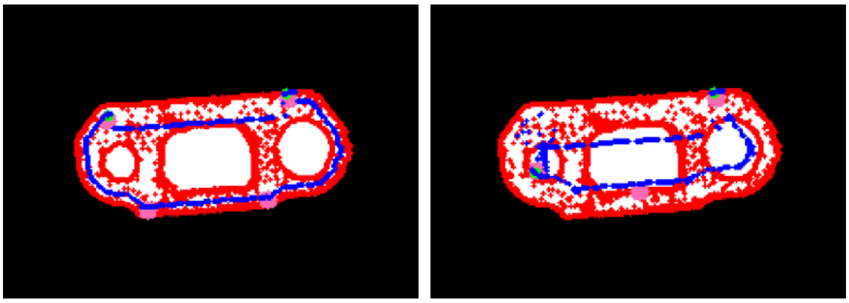
For the left figure, **Erode Kernel Size** is set to **-5**; for the right figure, **Erode Kernel Size** is set to **-20**. As the negative value decreases, the generated path will expand towards the outside of the object's edge.



Vertical Move Real Dis

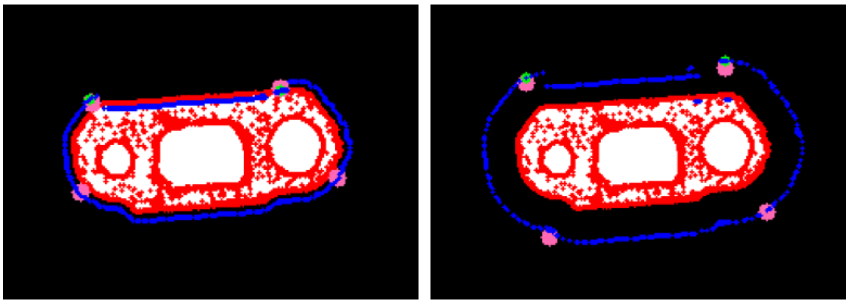
When **Vertical Move Real Dis** is set to **5** and **15**, respectively, the generated paths are shown below.

For the left figure, **Vertical Move Real Dis** is set to **5**; for the right figure, **Vertical Move Real Dis** is set to **15**. As the value of the positive number increases, the generated path will shrink towards the inside of the object edge.



When **Vertical Move Real Dis** is set to **−5** and **−20**, respectively, the generated paths are shown below.

For the left figure, **Vertical Move Real Dis** is set to **−5**; for the right figure, **Vertical Move Real Dis** is set to **−20**. As the negative value decreases, the generated path will expand towards the outside of the object edge.



4.3.192. Generate Path Given Depth Map

> Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step generates inward spiraling path’s targets based on an object’s point cloud edge and depth map.

Usage Scenario

This Step is generally used for gluing the edges of flat objects.

Input and Output

- Input:
  1. Object mask
  2. Scene depth map

3. Object point cloud

- Output:
  1. List of targets used to represent the generated path

4.3.193. Generate Zigzag Path

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step generates targets for a zigzag path based on the input point cloud.

Usage Scenario

This Step is usually used for automatic spraying on flat objects.

Input and Output

- Input:
  1. Object mask
  2. Scene depth map
  3. Object point cloud
- Output:
  1. List of targets of the generated path

4.3.194. Get First Image from Image List

The contents of this step are outdated. Please use the new Steps Trim Input List and Unpack Data in sequence. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

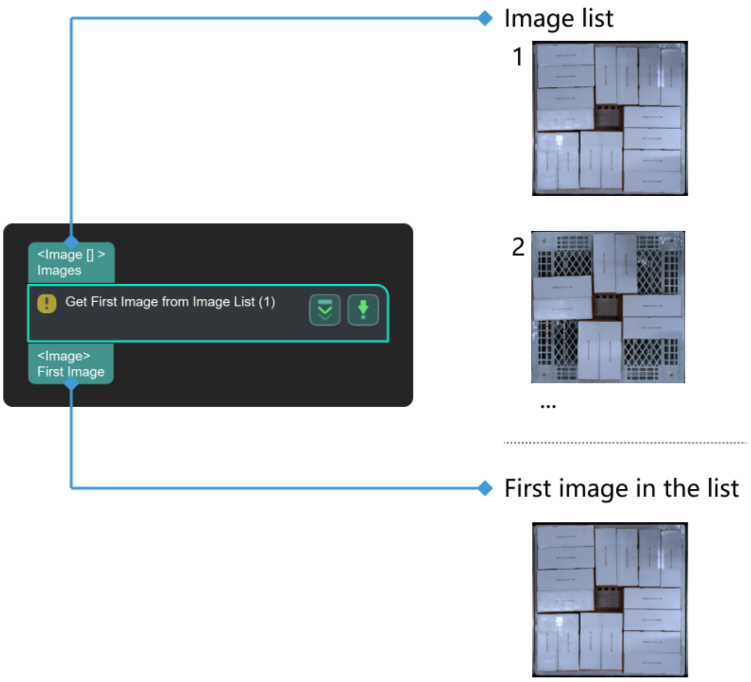
Function

Get the first image in an image list.

Usage Scenario

This Step is outdated. Please use the new Step mentioned above.

Input and Output



4.3.195. Get Flange Pose List

Function

This Step gets the flange pose of the robot.

Usage Scenario

In the eye-in-hand mode (a camera installation type), this Step obtains the flange pose of the robot when taking photos.

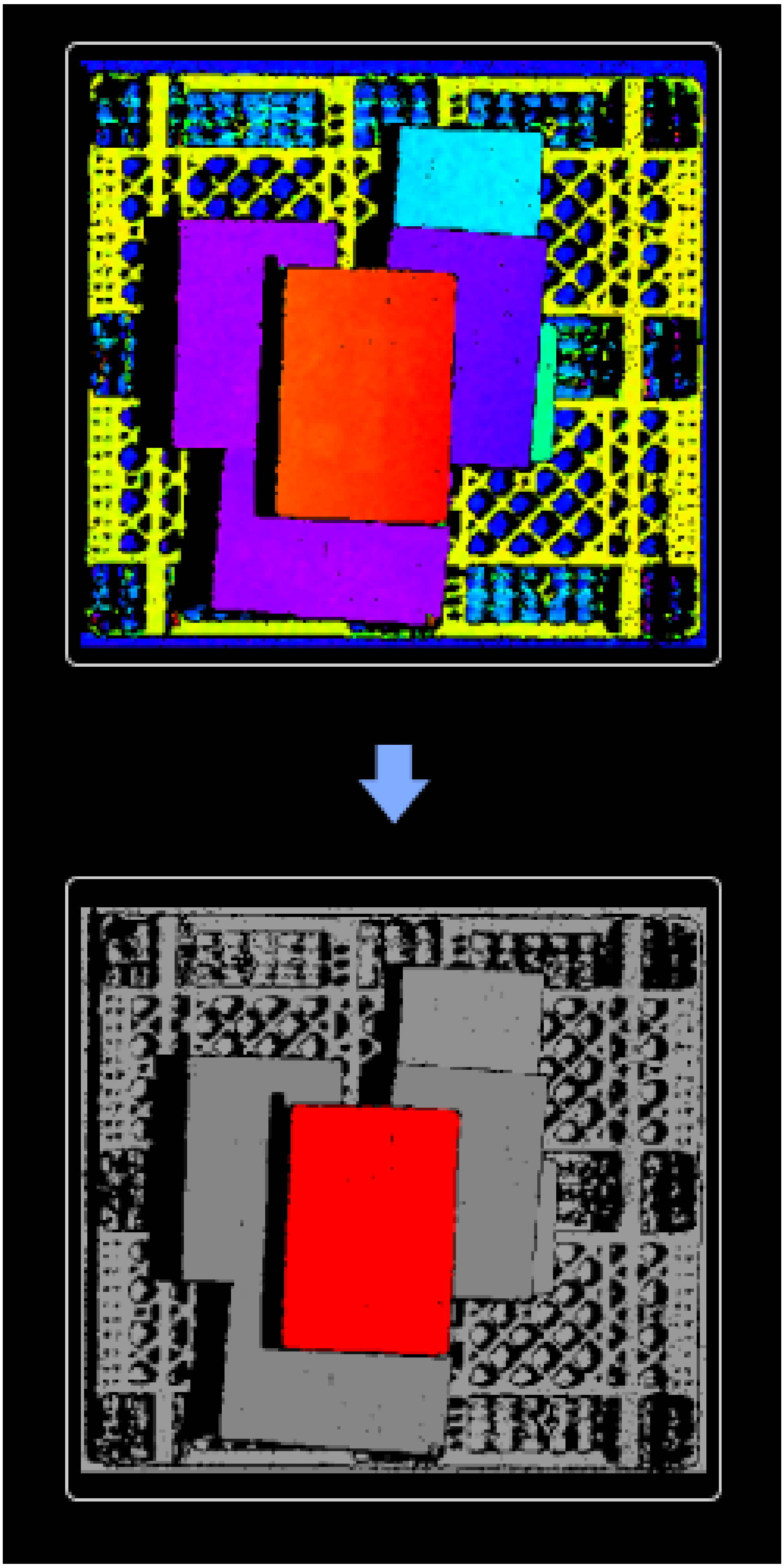
Input and Output

- **Input:** none.
- **Output:** obtained robot flange pose.

4.3.196. Get Highest Layer Regions in Depth Map

Function

This Step uses the given depth map and mask list to segment the image into multiple areas, outputs the area that is closest to the camera and discards the others.

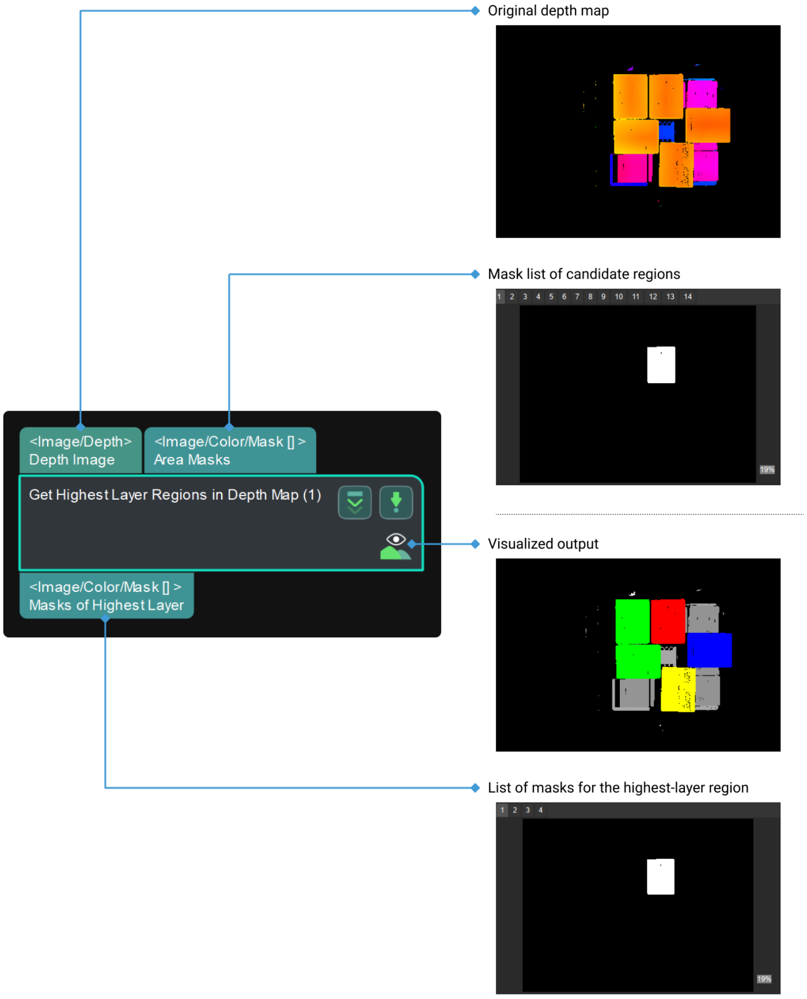


Usage Scenario

This Step is usually used in carton depalletizing applications to identify the objects on the highest layer. It is usually connected with Steps such as Invalidate Depth Pixels outside 3D ROI, Merge

Mask Images, and Segment Depth Map.

Input and Output



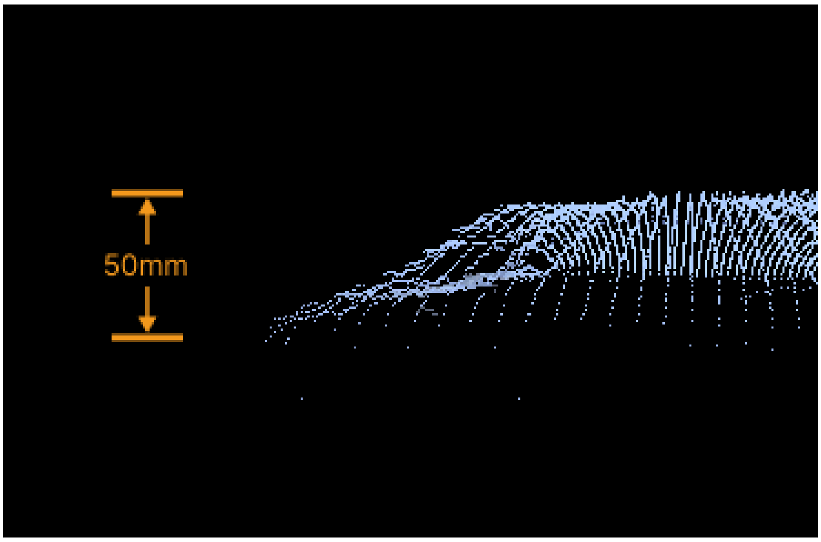
Parameters

Highest Layer Settings

Layer Height

Default value: 50 mm

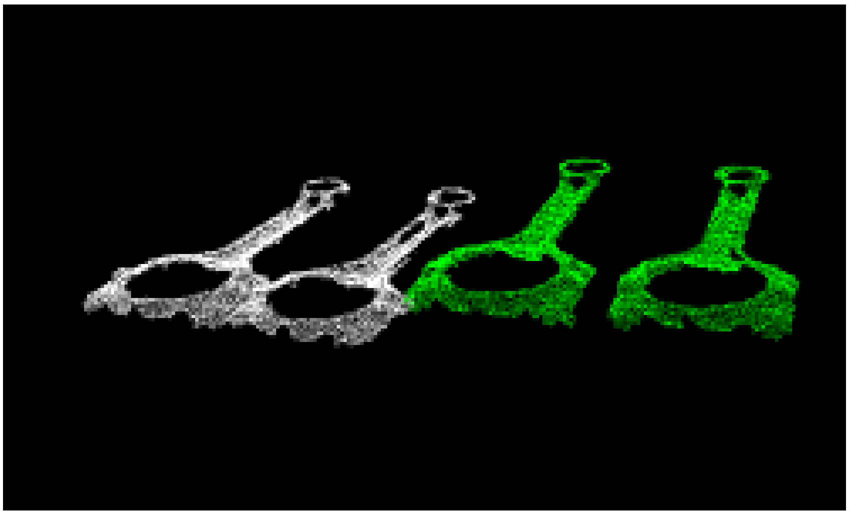
Instruction: This parameter refers to the range of height of the highest layer. It sets the range of height perpendicular to the highest layer of the objects. Masks of objects that are within the height range will be output. If there are objects of other layers included in this range, you should adjust the value of the parameter until there are only objects on the highest layer within the range.



4.3.197. Get Highest Layer Clouds

Function

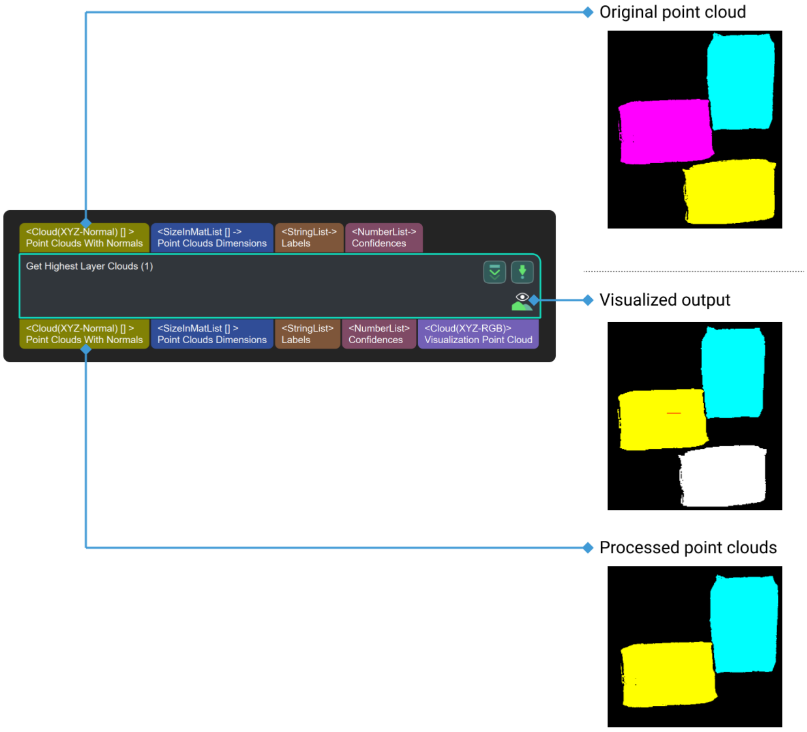
This Step sorts the centers of point cloud parts in descending order along a specified direction, and then obtains the highest-layer point cloud.



Usage Scenario

Keep point clouds that meet the requirements along the user-defined reference direction. Unlike Get Highest-Layer Points, this Step process the point cloud directly and usually follows Steps such as Point Cloud Clustering, and Apply Masks to Point Cloud.

Input and Output



Parameter Description

Reference Direction

X, Y, Z

Default setting: X = 0; Y = 0; Z = 1.0

Description: Set the X, Y, or Z value to define a reference direction.

Layer Settings

Only Get Highest One

Default Value: Unselected.

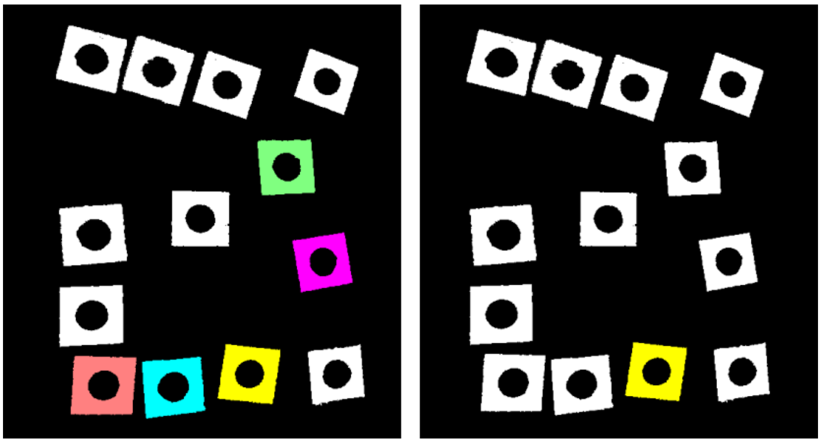
Description: Select to obtain the point cloud on the highest layer along the specified direction. Otherwise, all point clouds on the highest layer will be obtained.

Layer Height

Description: This parameter is used to set the layer height of the highest-layer point clouds. Point clouds within this height will be retained.

Default value: 100.000 mm

The figure below shows the output effects when the layer height is set to 100.000 mm and 35.000 mm, respectively.



Layer Tolerance

Default value: 1.0000

Description: Set the upper limit of height difference between the highest layer and the rest of the layers, which is equal to the product of layer height and layer tolerance. If the distance from the point to the highest layer is less than this upper limit, it is on the highest layer.

Cloud Settings

Use Max Size Cloud Normal

Default setting: Unselected.

Description: Select to use the normal of the largest point cloud as the reference direction. Once this option is selected, the **Reference Direction** will be invalid.

4.3.198. Get Highest-Layer Points

Function

This Step is used to get the points in highest layer along the specified direction.

Usage Scenario

A general point cloud filtering Step. No fixed usage scenarios.

Input and Output

- Input
  1. Highest layer points of point cloud input to this port will be extracted.
  2. Pose used to define the vertical direction (optional, if this port is not used, the parameter “Reference Direction” will be used; if inputting multiple poses, only the first will be used).
- Output
  1. Highest layer points extracted

Parameter Description

Reference Direction

Instruction: This parameter is used to select the reference direction, x, y, and z included.

Default value: x = 0; y = 0; z = 1.0

Suggested value: Set according to the actual scenarios.

Layer Settings

Only Get Highest One

Instruction: This parameter decides whether to only get the highest point along the specified direction or not.

Default setting: unselected.

Suggested value: Set according to the actual scenarios.

Layer Height

Description: This parameter is used to set the height of every layer (in mm).

Default value: 100.000 mm

Suggested value: Set according to the actual scenarios.

Layer Tolerance

Description: Set the upper limit of height difference between the highest layer and the rest of the layers, which is equal to the product of layer height and layer tolerance. If the height difference is less than the parameter, they belong to the same layer (the highest layer).

Default value: 1.0000

Suggested value: Set according to the actual scenarios.

4.3.199. Get Highest-Layer Poses

> Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

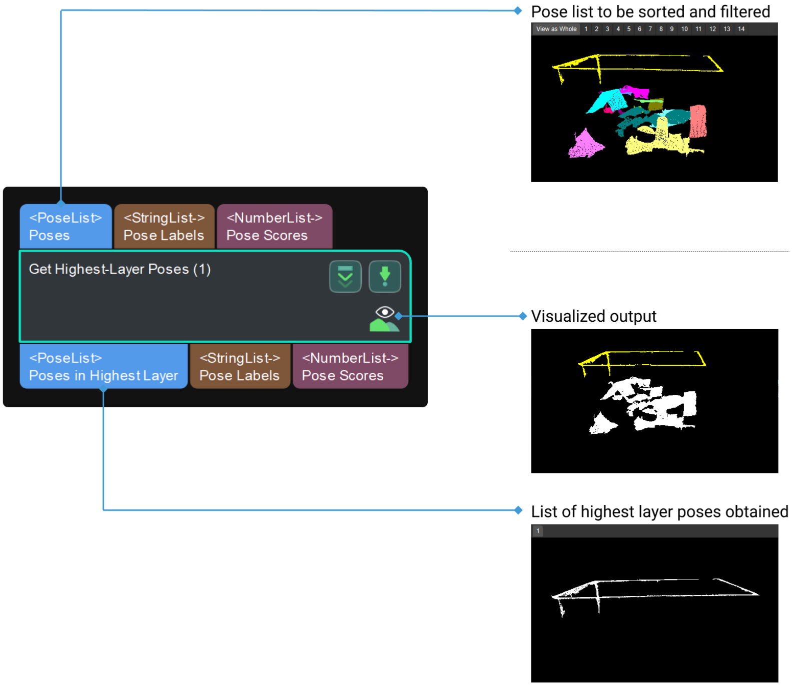
Function

Sort the poses in descending order along the specified direction, and then output the highest layer poses.

Usage Scenario

A general pose filtering Step. It outputs poses that meet the requirements according to the set parameters.

Input and Output



4.3.200. Get Highest Score Result

This Step is outdated. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step performs one-to-one comparison of two input point cloud lists with the corresponding poses and scores, and extracts the point clouds and poses with higher scores to form a list for output.

Usage Scenario

This Step is generally used to compare the results obtained by different matching methods to obtain the optimal list of matching results.

his Step is outdated. Please use newer Steps instead.

Input and Output

- Input:
  1. First list of point clouds
  2. Poses of first list of point clouds
  3. Matching scores of first list of point clouds
  4. Classification labels of first list of point clouds
  5. Confidences of first list of point clouds
  6. Second list of point clouds
  7. Poses of second list of point clouds

8. Matching scores of second list of point clouds
9. Classification labels of second list of point clouds
10. Confidences of second list of point clouds

- Output:
  1. List of point clouds with higher scores
  2. List of poses with higher scores
  3. List of higher scores
  4. List of classification labels with higher scores
  5. List of confidences with higher scores

4.3.201. Filter Ring Point Cloud List

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step filters circular point clouds according to the settings in the parameters. Point clouds within the threshold will be retained, and point clouds beyond the threshold will be discarded.

Usage Scenario

This Step is a custom Step for special scenarios.

Input and Output

- Input:
  1. Ring point clouds input to this port will be filtered.
  2. List of labels corresponding to the point clouds.
  3. List of scores corresponding to the point clouds.
- Output:
  1. A filtered list of ring point clouds
  2. List of labels corresponding to the remaining point clouds
  3. List of scores corresponding to the remaining point clouds
  4. List of masks corresponding to the remaining point clouds

4.3.202. Group Data

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step groups elements within a data list by index list. Multiple input ports can be set to perform grouping operations with the same index list within each data list at the same time. If the index list is \[[0, 2], [1, 3, 4]] and the input data is [100, 20, 300, 40, 500], the output is \[[100, 300], [20, 40, 500]].

Usage Scenario

This Step is a general data sorting Step.

Input and Output

- Input:
  1. Index list input at this port will be used for grouping.
  2. Data input to this port will be grouped.
- Output:
  1. Grouped data

4.3.203. Group 2D Poses

Function

This Step groups the 2D poses. Poses within the same mask will be grouped into a list, and poses not within any mask will be grouped into a list.

Usage Scenario

This Step is a custom Step for special scenarios.

Input and Output

- Input:
  1. 2D poses input to this port will be grouped.
  2. Masks for grouping 2D poses.
- Output:
  1. Grouped 2D poses.
  2. Grouping index lists.

4.3.204. Histogram Matcher

> This Step is outdated. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step compares the input 2D image histograms with the template histogram set in the parameter, and outputs the corresponding labels according to the matching score.

Usage Scenario

This Step is a custom Step for special scenarios.

Input and Output

- Input: Histograms input to this port will be matched against the template.

- **Output:** Matching result labels.

4.3.205. Image Brightness and Color Balancer

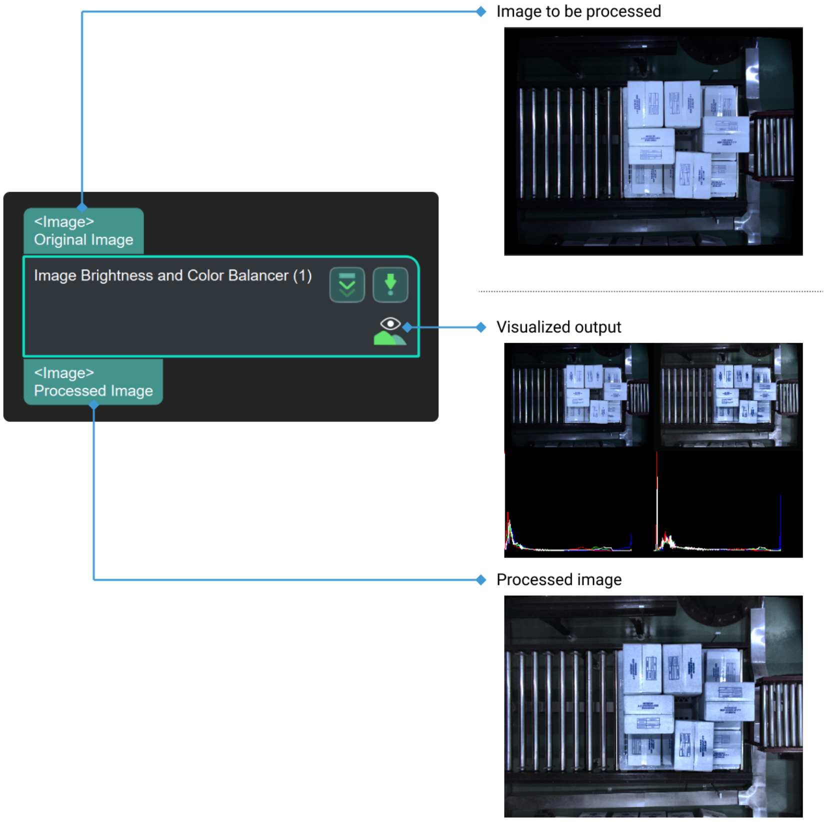
Function

This Step uses different algorithms to process images. It is mainly used to adjust the contrast of images and also used to adjust the color balance, which facilitates further processing such as edge detection and identification.

Usage Scenario

This Step is used in scenes with large variations in light intensity. It performs a balancing operation on the image, so that the brightness or color of the image is within an appropriate range to facilitate subsequent processing.

Input and Output



Parameters

The following five types of balancers are available, each requires different parameter settings:

- AdaptiveColorBalancer
- CLAHE
- ColorBalancer
- GammaCorrection

- Illumination Normalization

AdaptiveColorBalancer

This balancer performs a balancing operation based on the RGB channels' values of the image, while its effect on grayscale images is not obvious.

There are no parameters for this kind of balancer.

CLAHE

The default balancer is CLAHE.

Clip Limit

Instruction: This parameter is used to adjust the color balance. When the value is increased, the balancing effect is more obvious, and the degree of contrast decreases.

Default value: 4

ColorBalancer

Lightness

Instruction: This parameter is used to adjust the lightness of the image. When the value is increased, the image will be brighter.

Default value: 0

Green to Red

Instruction: This parameter is used to adjust the color of the image. When the value is increased, the image will tend to be reddish.

Default value: 0

Blue to Yellow

Instruction: This parameter is used to adjust the color of the image. When the value is increased, the image will tend to be yellowish.

Default value: 0

GammaCorrection

Gamma

Instruction: When the value is increased, the degree of contrast decreases. Only a few pixels will be displayed if this value is set to 0.

Default value: 1.00

Illumination Normalization

There are three kinds of methods in Illumination Normalization, and the parameters of each method are different.

- Retinex_SSR

- Retinex_MSR
- BgAdjust

Retinex_SSR

Gaussian Kernel Settings

Kernel Size

Instruction: This parameter is used to set the Gaussian kernel size, and the value can only be an odd number. When the value is increased, the edge of objects in the image will be more obvious.

Default value: 21

Tuning recommendation: Please set the value according to the actual situation.

General Settings

Mean Illumination

Instruction: This parameter is used to adjust the mean value of illumination. When the value is increased, the image will be brighter, and is suitable for processing images acquired under darker lighting conditions.

Default value: 100

Tuning recommendation: Please set the value according to the actual situation.

Retinex_MSR

Gaussian Kernel Settings

This method combines three kinds of Gaussian kernel with different kernel sizes to process the given image. Adjusting the sizes of three kernels could bring images with different edge effects. Usually, the Retinex MSR method could obtain more edge information than Retinex SSR.

Small/Medium/Large Kernel Size

Instruction: This parameter is used to set the Gaussian kernel size, and the value can only be an odd number. When the value is increased, the edge of objects in the image will be more obvious.

Default value of small size: 15

Default value of medium size: 81

Default value of large size: 201

Tuning recommendation: Please set the value according to the actual situation.

General Settings

Mean Illumination

Instruction: This parameter is used to adjust the mean value of illumination. When the value is increased, the image will be brighter, and is suitable for processing images acquired under darker lighting conditions.

Default value: 100

Tuning recommendation: Please set the value according to the actual situation.

BgAdjust

Background Adjustment Settings

ROI

> The coordinate of the upper left corner of the original image is (0, 0).

X

Instruction: This parameter refers to the X coordinate of the ROI’s upper left vertex.

Default value: 30

Y

Instruction: This parameter refers to the Y coordinate of the ROI’s upper left vertex.

Default value: 700

Width

Instruction: This parameter refers to the width of the ROI.

Default value: 200

Height

Instruction: This parameter refers to the height of the ROI.

Default value: 200

General Settings

Mean Illumination

Instruction: This parameter is used to adjust the mean value of illumination. When the value is increased, the image will be brighter, and is suitable for processing images acquired under darker lighting conditions.

Default value: 100

Tuning recommendation: Please set the value according to the actual situation.

4.3.206. Image Filtering

Function

This Step can perform filtering process on the input images with different kinds of filters.

Usage Scenario

This Step is widely used for image processing, and there is no specific usage scenario.

Parameters

The following types of filters are available:

- GaussianFilter
- BilateralFilter
- Boxfilter
- ConditionalMeanFilter
- DftFilter
- GuidedFilter
- HaarFilter
- MedianFilter

GaussianFilter

Utilize Gaussian function to process images.

Size X

Instruction: This parameter is used to set the width of the Gaussian kernel size. When the value is increased, the image is blurred.

Default value: 3

Tuning recommendation: Please set the value according to the actual situation.

Size Y

Instruction: This parameter is used to set the height of the Gaussian kernel size. When the value is increased, the image is blurred.

Default value: 3

Tuning recommendation: Please set the value according to the actual situation.

Sigma X

Instruction: This parameter is used to set the standard deviation of the Gaussian kernel in X direction. When the value is increased, the image is blurred.

Default value: 3

Tuning recommendation: Please set the value according to the actual situation.

Sigma Y

Instruction: This parameter is used to set the standard deviation of the Gaussian kernel in Y direction. When the value is increased, the image is blurred.

Default value: 3

Tuning recommendation: Please set the value according to the actual situation.

BilateralFilter

Diameter

Instruction: This parameter determines the range of filtering area. When the value is increased, more original image information will be retained, so that the image is clearer.

Default value: 3

Tuning recommendation: Please set the value according to the actual situation.

Standard Deviation of Color Space Filter and **Standard Deviation of Coordinate Space Filter** affect the filter result jointly.

Standard Deviation of Color Space Filter

Instruction: This parameter is used to set the standard deviation of pixel values.

Default value: 3

Tuning recommendation: Please set the value according to the actual situation.

Standard Deviation of Coordinate Space Filter

Instruction: This parameter is used to set the standard deviation of the spatial distance between pixels.

Default value: 3

Boxfilter

Kernel Size

Instruction: This parameter is used to set the kernel size during the filtering process and its unit is pixel. When the value is increased, the image obtained is blurred.

Default value: 3

Tuning recommendation: Please set the value according to the actual situation.

ConditionalMeanFilter

Kernel Size

Instruction: This parameter is used to set the kernel size.

Default value: 3

Tuning recommendation: Please set the value according to the actual situation.

Threshold

Instruction: This parameter is used to set the threshold of the distance between pixels and its unit is pixel (px). When the distance is greater than this threshold, the pixel will be retained. When it is less than this threshold, the pixel will not be included in filtering.

Default value: 100 px

Tuning recommendation: Please set the value according to the actual situation.

DftFilter

Use discrete Fourier transform to process images and filter out the signals with high frequencies, noise and edges for example. This filter is capable of noise reduction and edge enhancement.

GuidedFilter

Kernel Size

Instruction: This parameter is used to set the kernel size. When the value is increased, the image smoothing effect is more obvious.

Default value: 3

Tuning recommendation: Please set the value according to the actual situation.

Eps

Instruction:This parameter is used to set the regularization parameter of the guided filter.

Default value: 0.01

Tuning recommendation: Please set the value according to the actual situation.

HaarFilter

Smooth the image and remove some noise.

MedianFilter

Kernel Size

Instruction: This parameter is used to set the kernel size. When the value is increased, the image smoothing effect is more obvious.

Default value: 3

Tuning recommendation: Please set the value according to the actual situation.

4.3.207. Adjust Image

> Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

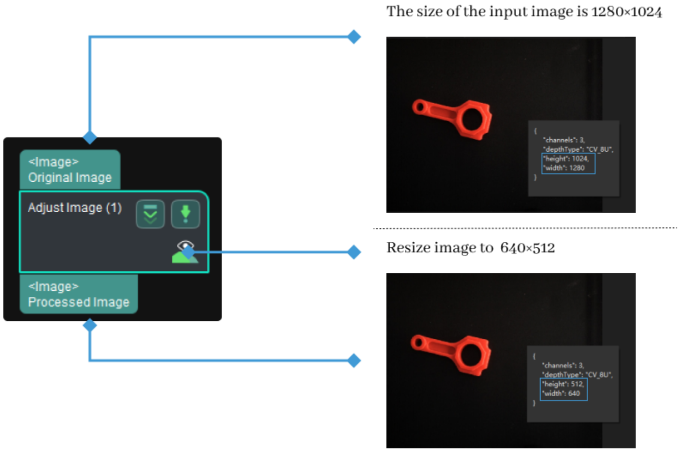
Function

This Step scales the input 2D image or rectifies the image with the camera’s distortion coefficients.

Usage Scenario

This Step is a general image processing Step. No fixed usage scenarios.

Input and Output



The size of the input image is 1280×1024

```
{
    "channels": 3,
    "depthType": "CV_8U",
    "height": 1024,
    "width": 1280
}
```

Resize image to 640×512

```
{
    "channels": 3,
    "depthType": "CV_8U",
    "height": 512,
    "width": 640
}
```

4.3.208. Insert End Target and Send Motion Params

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step inserts a single path target at the end of the input path.

Usage Scenario

This Step is usually used in applications involving path planning.

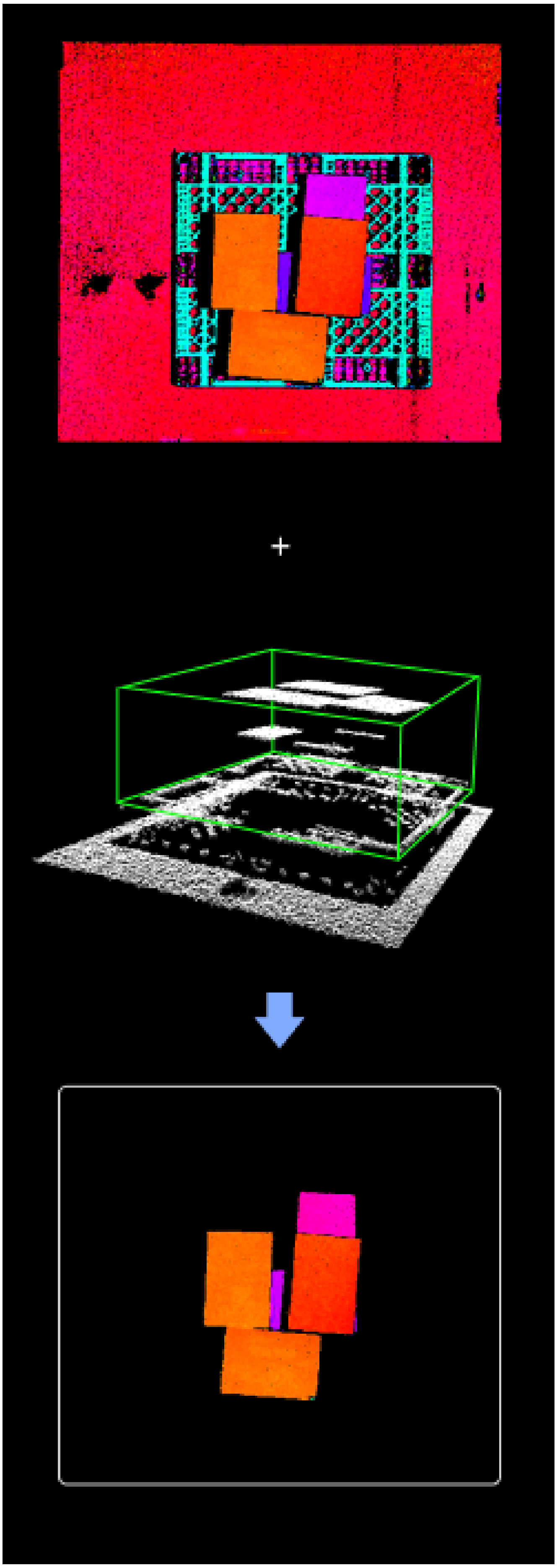
Input and Output

- Input:
  1. A new path target will be inserted at the end of the path target list input at this port.
  2. New path target input at this port will be inserted at the end of the list.
- **Output:** Path target list after the new track point is inserted at the end.

4.3.209. Invalidate Depth Pixels outside 3D ROI

Function

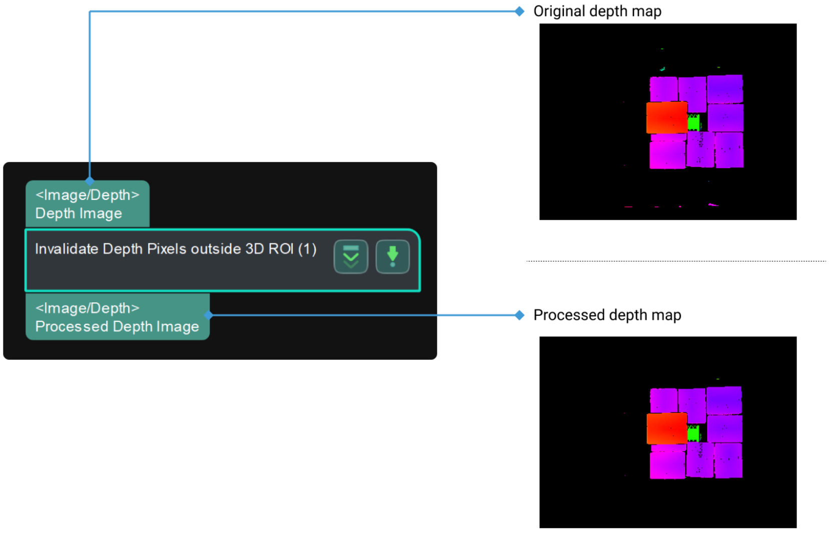
Set a Region of Interest (ROI) in the depth image and set the pixel values outside the region to zero.



Usage Scenario

This Step can be used to extract a target area in the original depth image and therefore avoid interference from background and other regions. It is usually connected with Steps such as Segment Depth Map, Get Highest Layer Regions in Depth Map, and Merge Mask Images.

Input and Output



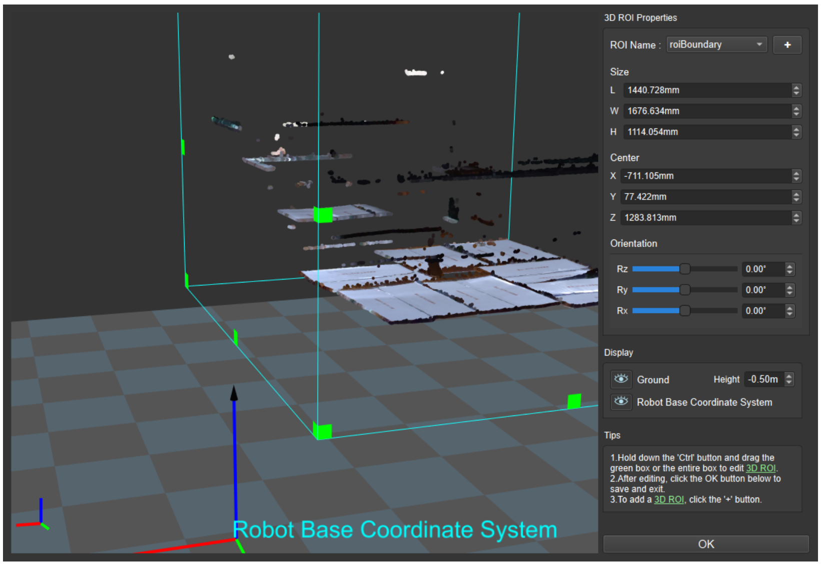
Parameter Description

3D ROI Settings

3D ROI

Description: This parameter is used to set 3D ROI.

Instruction: Click **[Set 3D ROI]** to open the **Set ROI** window, as shown below. For detailed settings, please refer to Instructions for Setting 3D ROI.



3D ROI Properties

ROI Name : roiBoundary +

Size

L 1440.728mm

W 1676.634mm

H 1114.054mm

Center

X -711.105mm

Y 77.422mm

Z 1283.813mm

Orientation

Rz 0.00°

Ry 0.00°

Rx 0.00°

Display

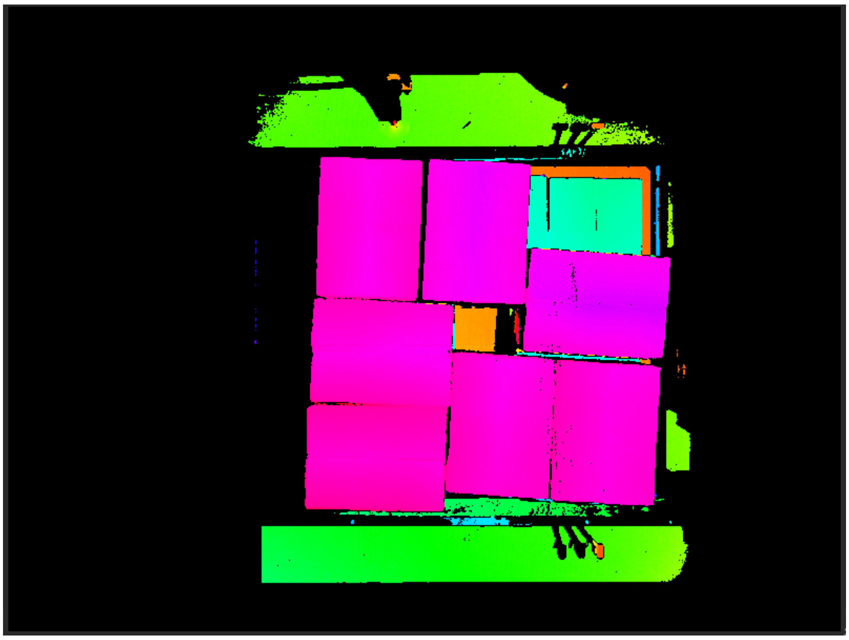
Ground Height -0.50m

Robot Base Coordinate System

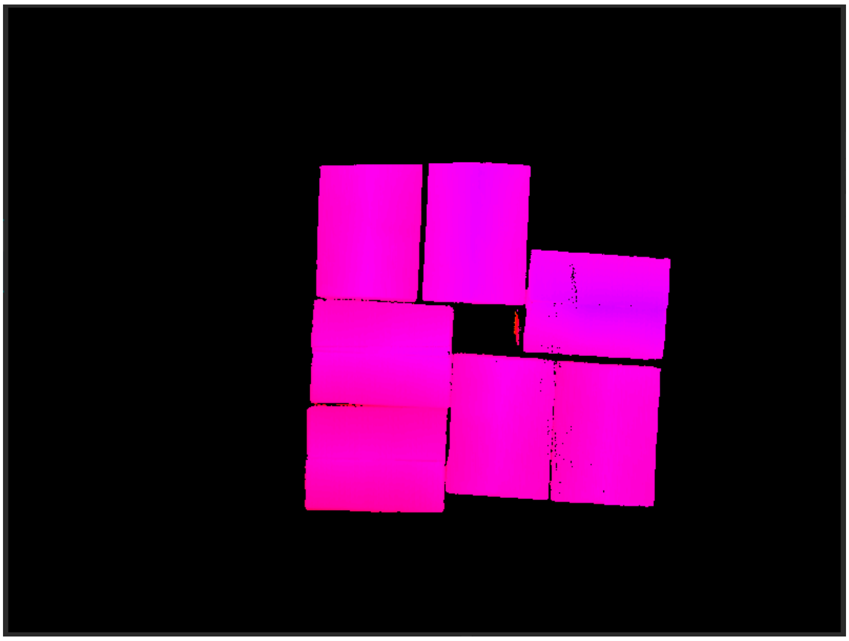
Tips

1.Hold down the 'Ctrl' button and drag the green box or the entire box to edit 3D ROI.
2.After editing, click the OK button below to save and exit.
3.To add a 3D ROI, click the '+' button.

After the depth ROI interception, the pixel values outside the region of the depth image’s interest will be set to zero. The figure below shows the input depth map.



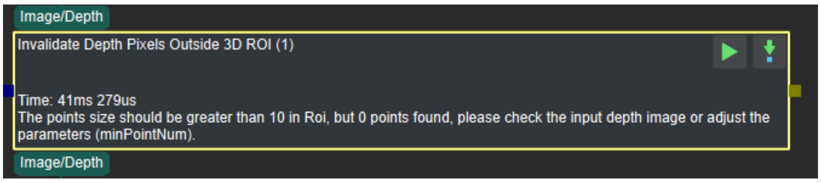
The figure below shows the depth map after interception, where invalid information around the boxes is filtered out.



Threshold Setting

Min Point Count in ROI

Description: 3D ROI displays the threshold of the number of points in the middle. If the number of points is threshold, it will warn, as shown below.



Image/Depth
Invalidate Depth Pixels Outside 3D ROI (1)

Time: 41ms 279us
The points size should be greater than 10 in Roi, but 0 points found, please check the input depth image or adjust the parameters (minPointNum).
Image/Depth

Instruction: The threshold can be set according to the point cloud requirement posed by actual engineering.

4.3.210. Invert Poses

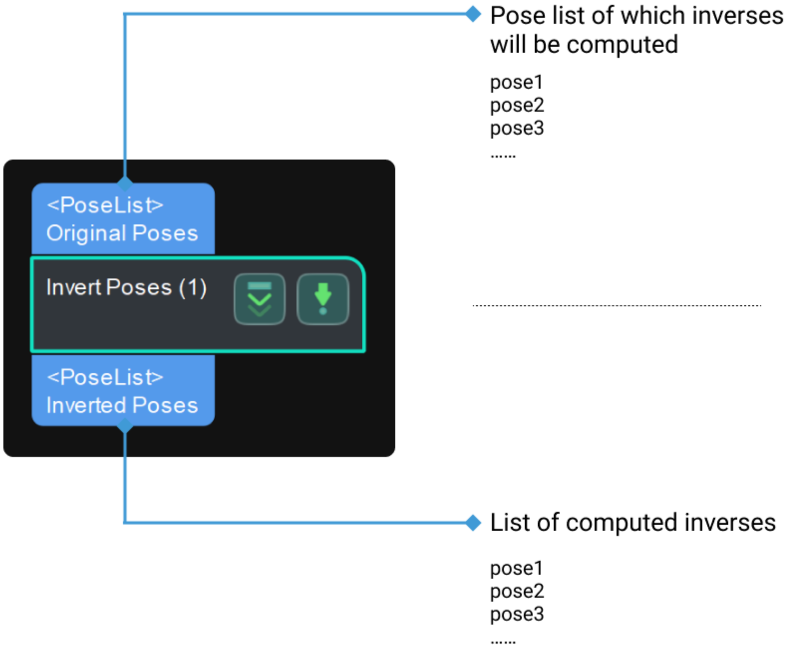
Function

This Step obtains the inverses of the poses in the input list.

Usage Scenario

This Step is a general pose calculation Step. This Step can be used for transformation between reference frames.

Input and Output



4.3.211. Get Inverses of Quaternions

Function

This Step computes the inverses of the input quaternions.

Usage Scenario

This Step is a general quaternion inverse computation Step. No fixed usage scenarios.

Input and Output

- **Input:** Quaternions input to this port will have their inverses computed.
- **Output:** computed inverses of the quaternions.

4.3.212. Compare Z Values of Poses with Threshold

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

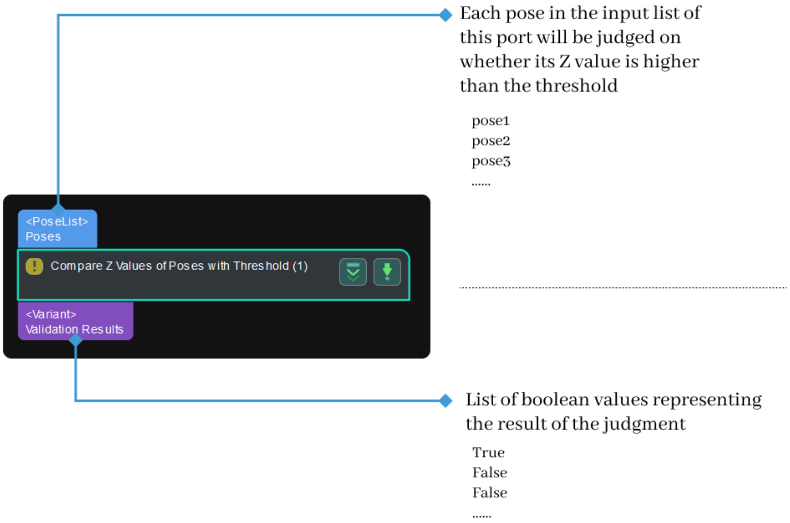
Function

Compare the Z values of the input poses with the threshold set in the parameters and output the judgment result. If a pose’s Z value is greater than or equal to the threshold, output “True”; otherwise, output “False.”

Usage Scenario

A custom Step for special scenarios.

Input and Output



Each pose in the input list of this port will be judged on whether its Z value is higher than the threshold

pose1
pose2
pose3
……

List of boolean values representing the result of the judgment

True
False
False
……

4.3.213. Join Labels

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step joins the labels from the two input lists one to one. If the input of the first port is [“a”, “b”], and the input of the second port is [“c”, “d”], the output is [“ac” , “bd”].

Usage Scenario

This Step is a general label editing Step. This Step has no fixed usage scenarios.

Input and Output

- Input:
  1. Labels entered for this port will be joined.
  2. Labels entered for this port will be joined.

- Output:
  1. Joined labels

4.3.214. Keep Poses Distributed in Regular Polygon

ℹ Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

Given a list of poses, this Step keeps only the desired poses. The desired poses are those at the vertices of regular polygons.

Usage Scenario

This Step is a custom Step for special scenarios; for wheel hub mounting hole recognition.

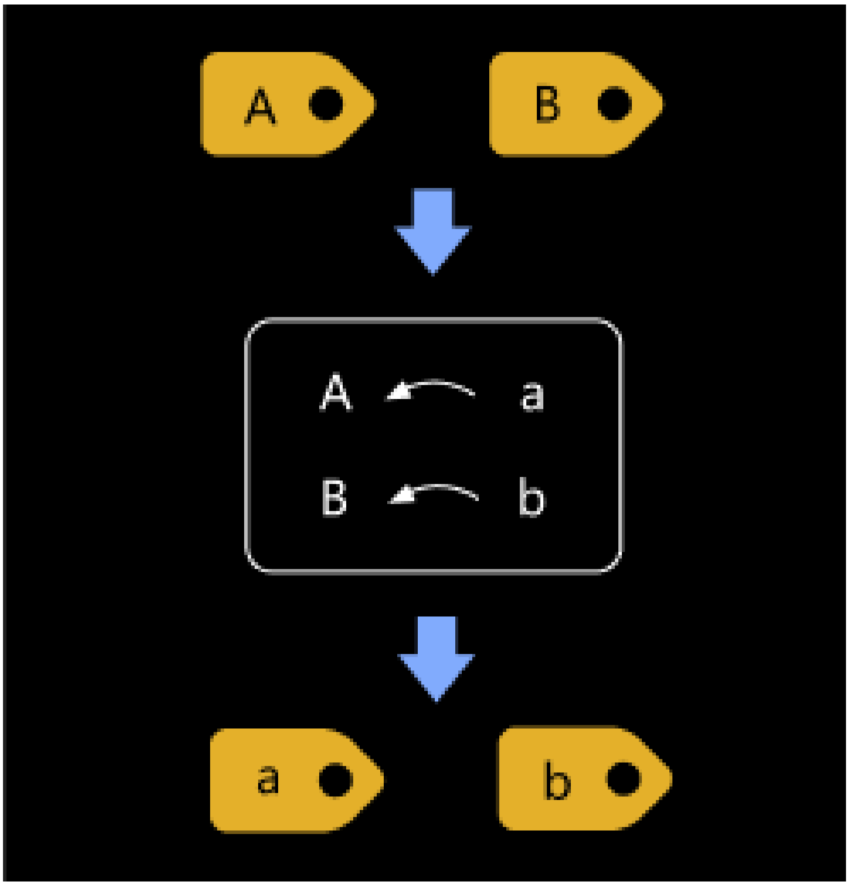
Input and Output

- **Input**: List of poses input to this port will be filtered.
- **Output:** Poses at the vertices of regular polygons.

4.3.215. Label Mapping

Function

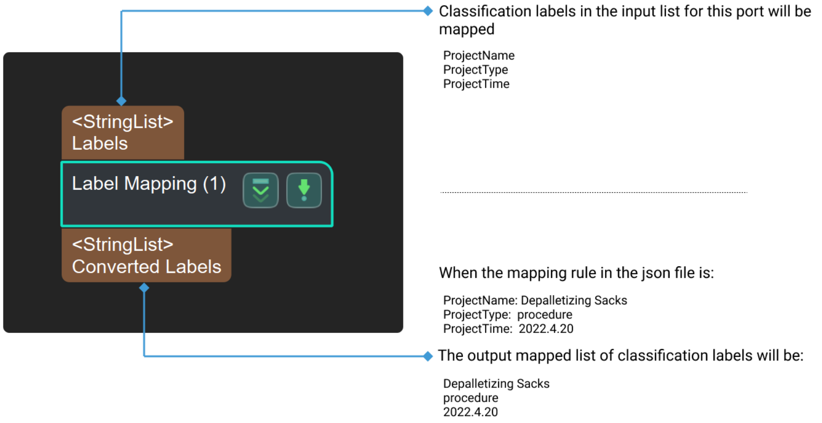
This Step renames labels through one-to-one mapping of the input classification labels.



Usage Scenario

This Step is used to rename labels.

Input and Output



Classification labels in the input list for this port will be mapped

ProjectName
ProjectType
ProjectTime

When the mapping rule in the json file is:

ProjectName: Depalletizing Sacks
ProjectType: procedure
ProjectTime: 2022.4.20

The output mapped list of classification labels will be:

Depalletizing Sacks
procedure
2022.4.20

Parameters

Port Settings

Size of Input Ports (1–8)

Parameter description: To set the number of input port in this Step.

Default value: 1

Value range: 1–8

File Settings

> **Set label mapping** is available only when the value of Size of Input Ports is 1.

Label File

Parameter description: To set the label files after label mapping.

Default value: labelmap.json

Instructions:

1. Click [**Set label mapping**] to enter the JSON editor page.
2. Click +, type the label to be renamed in **Label before conversion**, and click [**OK**]. Type the renamed label in **Label after conversion** and click [**OK**] to save the setting. The label is successfully renamed, and the mapping panel will display the correspondence between the two names in the bar.

   > If a label includes decimals or special characters, this label cannot be added for conversion and there will be a hint below the mapping panel suggesting that the user input is invalid.

3. If you still need to rename the label, please repeat the second step until you have added all the labels to be named.
4. **Directory** refers to the directory of the current project. Click [**...**] on the right to change the directory of the project according to actual needs.
5. Name the file in **JSON File Name** and click [**Save**] to save the label in the file.
6. Click [**OK**] to complete the setup.

   > If the label file is already renamed beforehand, you can just click the [**...**] in **JSON File Name** and select the file to read the mapping relation.

4.3.216. 3D Laser Profiler

Function

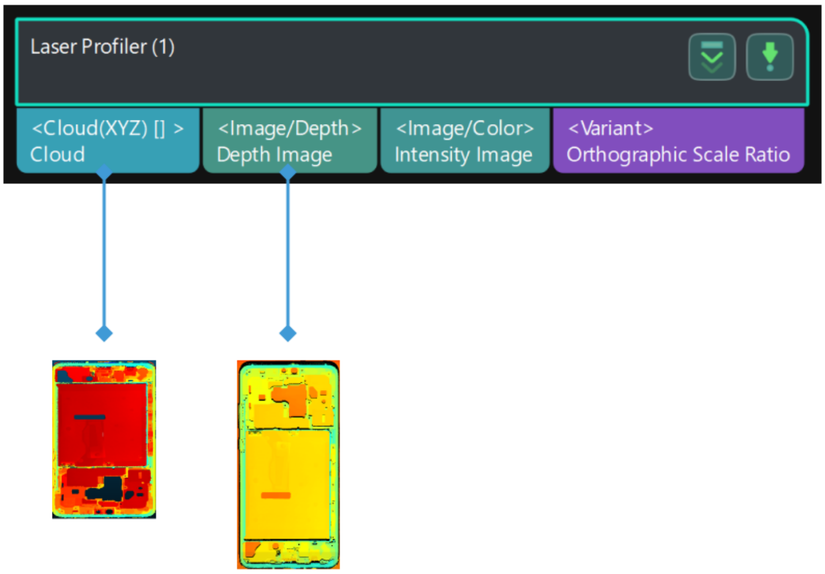
This Step can control 3D laser profiler (hereinafter referred to as laser profiler) to acquire data by using Mech-Vision.

> Currently, this Step is only available for Mech-Eye 3D Laser Profiler and SSZN 3D Laser Profiler.

Usage Scenario

This Step is usually used as the initial input for a vision project. This Step is suitable for the inspection and measurement applications in industries include consumer electronics, EV battery, automobile, and home electronics with a high requirement for precision.

Input and Output



Usage Instructions

This Step should be used in conjunction with Mech-Eye Viewer. The usage instruction is as follows.

1. Connect the laser profiler in Mech-Eye Viewer.
2. Adjust the parameters and save the parameter group in the Profile Mode and Scan Mode.
    - Profile Mode: Adjust the exposure time, analog gain, digital gain or laser power to ensure a good quality of the profile, intensity image and depth map.
    - Scan Mode: Adjust “Trigger setting” and “Scan setting” to prepare for collecting images.
3. Set the IP address of the selected laser profiler, and select the saved parameter group in the Mech-Vision.
4. Adjust the parameters of the “3D Laser Profiler” Step in the Mech-Vision.
5. Run the Step.

Parameters

Laser Profiler Model

Parameter description: This parameter is used to select the laser profiler model you want to use.

Value list: LNXCamera, SSZNCamera

Default value: LNXCamera

Tuning instruction: If you need to use Mech-Eye 3D Laser Profiler, please select “LNXCamera.” If you need to use SSZN 3D Laser Profiler, please select “SSZNCamera.”

Basic Settings

IP Address

Parameter description: The IP address of the laser profiler.

Tuning instruction: Click **[Open the editor]**, and select the laser profiler that you want to use and its IP address in the pop-up window. After selecting OK, the IP address will be automatically filled in this parameter frame.

Parameter Group

Parameter description: This parameter is used to select the parameter group used in the laser profiler.

Trigger Settings

Data Acquisition Trigger Source

Parameter description: This parameter is used to select the source of the signals that trigger data acquisition. In one round of data acquisition, multiple lines are scanned, multiple profiles are generated, and one intensity image and one depth map are generated using the profile data.

Value list: External, Software.

Tuning instruction: If you use externally input signals to trigger data acquisition, select “External.” Otherwise, select “Software.”

Line Scan Trigger Source

Parameter description: This parameter is used to select the source of the signals that trigger the scan of a single line.

Value list: Encoder, Fixed rate.

Tuning instruction: If you use an encoder to trigger scanning, select “Encoder.” If you need to trigger scanning at a fixed rate, select “Fixed rate.”

Scan Line Count

Parameter description: This parameter is used to set the number of profiles needed to generate one intensity image/depth map.

Default value: 1600

Tuning instruction: Make sure that the set value can cover one target object completely.

Scan Duration

Parameter description: This parameter is used to set the duration for each round of data acquisition (in ms). The profiles acquired in this duration will be used to generate one intensity image/depth map.

Default value: 5000 ms

Y-Axis Resolution

Parameter description: This parameter is used to set the resolution in the Y direction, which is the distance between two neighboring points along the travel direction of the target object.

Trigger Interval

Parameter description: This parameter is used to set the number of trigger signals needed for scanning one line.

Pulse Equivalent

Parameter description: This parameter is used to set the distance or displacement (in mm) represented by an encoder signal period.

Default value: 0.004000 mm

Tuning instruction: This parameter is related to the encoder hardware specifications.

Other Settings

Movement Mode

Description: This parameter is used to select the movement mode which determines how the object moves relative to the laser profiler.

Value list: One direction, Back and forth.

Tuning Instruction: Select the movement mode according to actual situation.

X Direction Translation

Parameter description: When you select “Back and forth,” Mech-Vision will output multiple acquired point clouds that represent parts of the object, and therefore an entire point cloud of the object can be generated. This parameter is used to set the distance (in mm) between the data acquired from two scans on the X-axis.

Default value: 100.000 mm

Cache Setting

Maximum Cached Data

Description: This parameter is used to show the maximum data volume that can be cached. With each scan, the cached data can generate depth maps, intensity maps, and point clouds. When this value is exceeded, you can process the cached data using the “Cache Overflow Solution” or manually clear the cache.

Cache Overflow Solution

Description: This parameter is used to select the solution to the cache overflow.

Value list: Delete earliest acquired data, Pause current data acquisition.

Clear Cache

Description: This parameter is used to clear cached image data manually.

4.3.217. Load Targets in Path and Apply Affine Transform

ℹ Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step loads the targets of a path and transforms the targets from the camera reference frame to the robot reference frame.

Usage Scenario

This Step is usually used in door/glass gluing applications.

Input and Output

- Input:
  1. Rigid body transformation matrix input to this port will be used for transformation between reference frames.
  2. Scene point cloud. For adjusting targets to the point cloud surface (if the parameter “Refine Poses” is set to True).
- Output:
  1. Targets after reference frame transformation

4.3.218. Load 2D Path

> Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step loads 2D path targets from a file and maps them to 3D path targets represented by poses.

Usage Scenario

This Step is usually used in applications involving path planning.

Input and Output

- Input:
  1. 2D image with objects
  2. Depth map with objects
  3. Point cloud with objects
- Output:
  1. Calculated 3D path targets

4.3.219. Locate Calibration Spheres

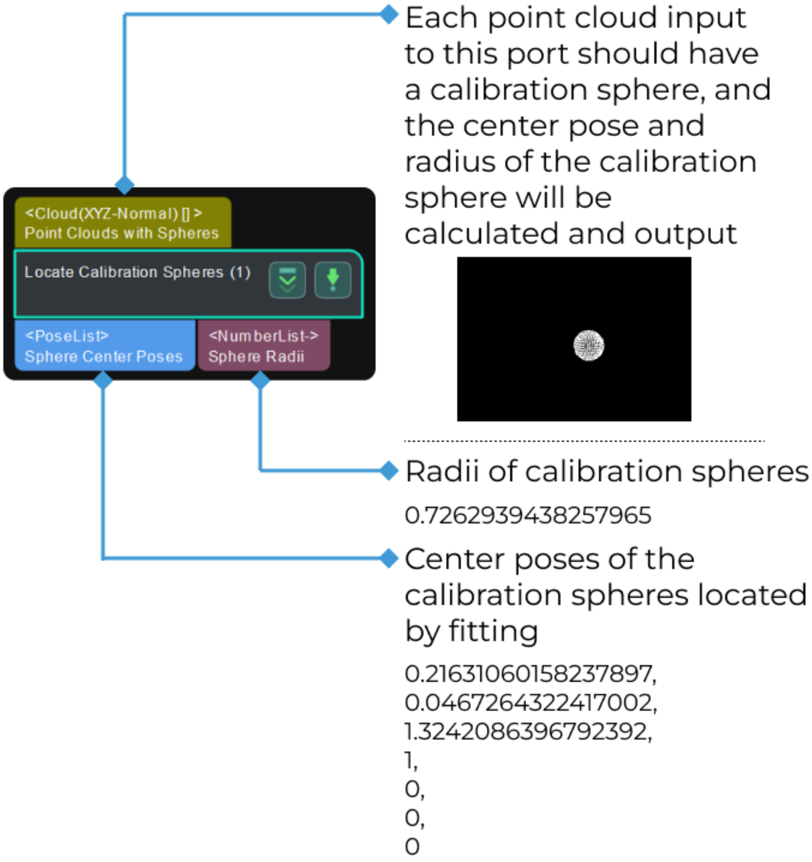
Function

This Step is used to fit a sphere (calibration sphere) in each input point cloud, and output the center pose and radius of each calibration sphere.

Usage Scenario

This Step is used to compensate for the temperature drifts of the camera and robot by locating the calibration spheres in the scene in measurement scenarios.

Input and Output



Each point cloud input to this port should have a calibration sphere, and the center pose and radius of the calibration sphere will be calculated and output

Radii of calibration spheres

0.7262939438257965

Center poses of the calibration spheres located by fitting

0.21631060158237897,
0.0467264322417002,
1.3242086396792392,
1,
0,
0,
0

Parameter Description

This Step does not have any parameter.

4.3.220. Point Poses to Reference Positions

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step rotates specified axes of input poses to specified position(s).

Usage Scenario

This Step is a general pose rotation Step. No fixed usage scenarios.

Input and Output

- Input:
  1. Poses input to this port will have specified axes rotated to specified position(s).
  2. Specified positions for rotating the poses (optional, if this port is not used, the settings of parameters under “Reference Translation Settings” will be taken as a specified position).
- Output:

1. Poses with specified axes pointing to specified positions(s)

4.3.221. Make Template

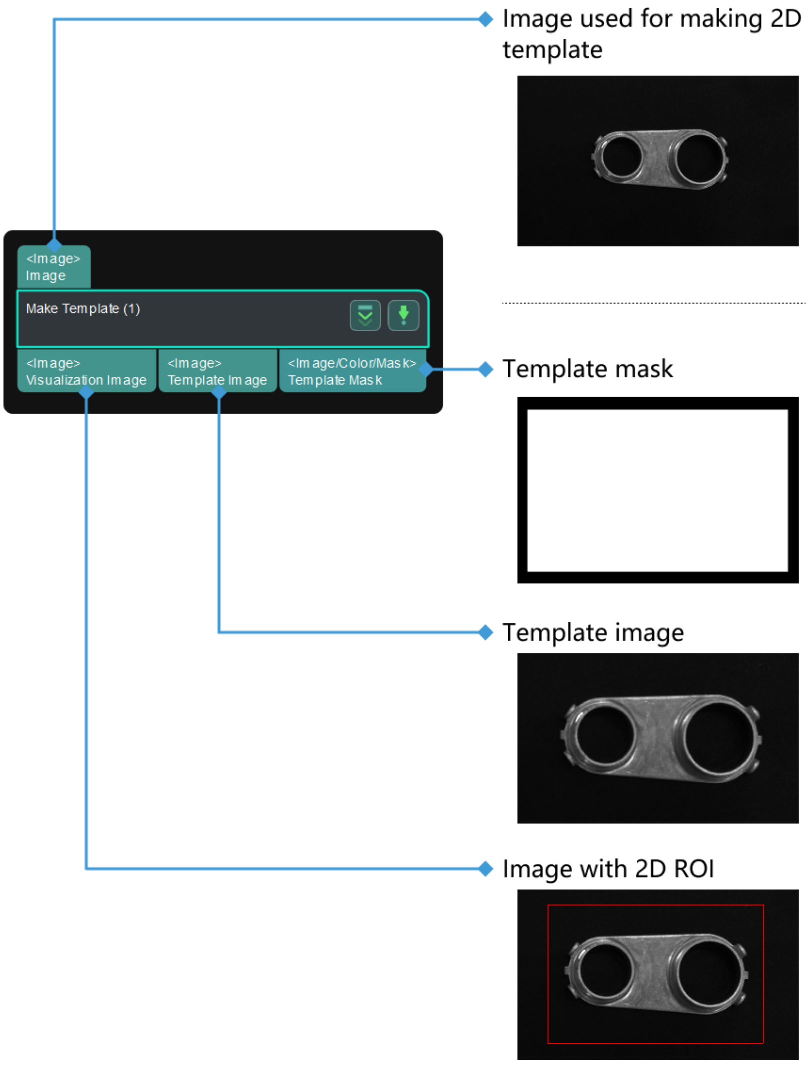
Function

Select a 2D ROI on the input image in the sketchpad in measurement mode. The ROI is output as the template image, and the mask of the ROI as the template mask.

Usage Scenario

Usually used in measurement scenarios for making the templates used in the Template Matching Step, which calculates the 2D pose of an object in a 2D image.

Input and Output

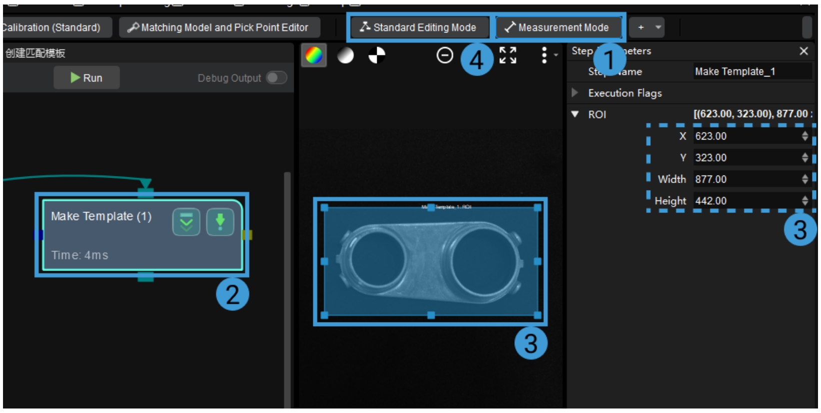


Parameter Description

ROI

Default value: [(−1.00, −1.00)，0.00 × 0.00]

Instructions: Click **[Measurement Mode]** on the toolbar to enter measurement mode. Select the **Make Template** Step. The ROI select frame is in the upper left corner of the sketchpad. Drag it to adjust the position, and drag its vertices to adjust the size of the ROI. Click Standard Editing Mode on the toolbar,and the ROI you selected is recorded as parameter values.



Step Name: Make Template_1

ROI: [(623.00, 323.00), 877.00 :

X: 623.00

Y: 323.00

Width: 877.00

Height: 442.00

4.3.222. Map Depth to RGB

Function

This Step adjusts the pixel coordinates of the depth map to correspond to the color image.

Usage Scenario

This Step is a custom Step for special scenarios.

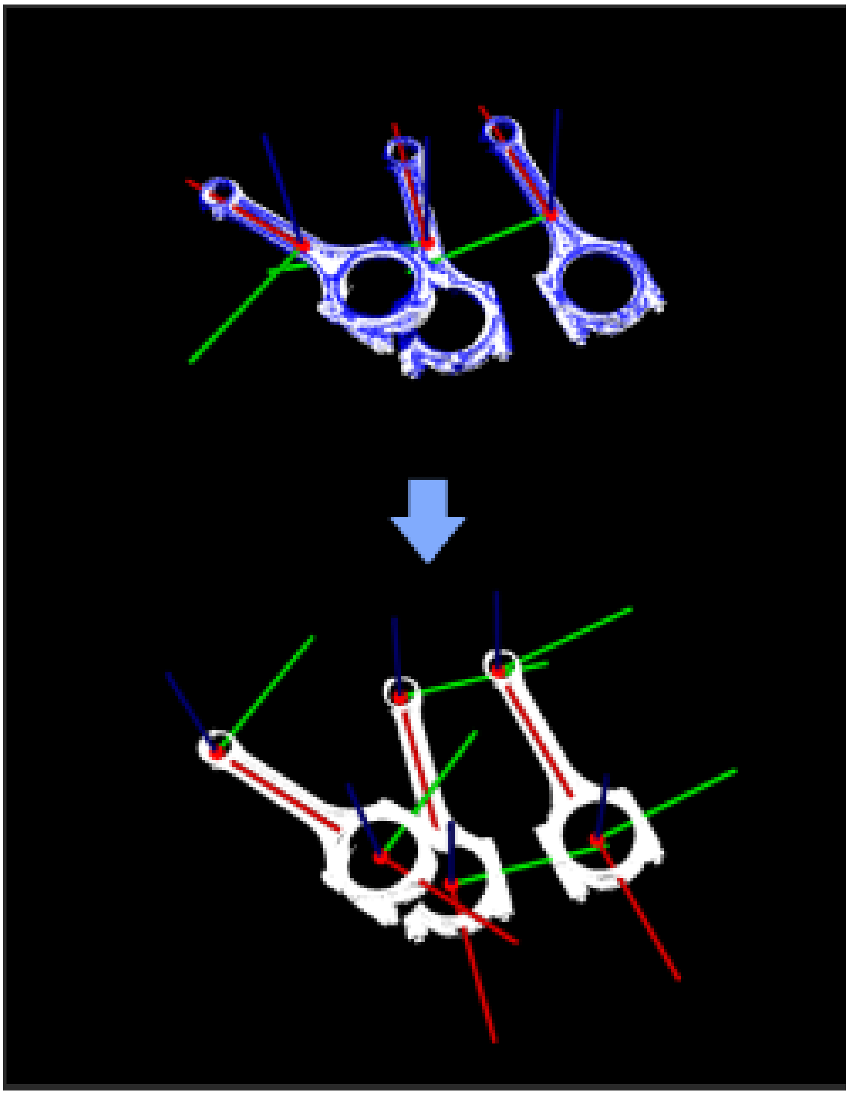
Input and Output

- Input:
  1. Depth map input to this port will be adjusted to have pixel coordinates correspond to the color image.
  2. Color image corresponding to the depth map.
  3. Point cloud corresponding to the depth map.
- Output:
  1. Adjusted depth map

4.3.223. Map to Multiple Pick Points

Function

Calculate other pick points of objects through object geometric center points and their mapping relationships with the multiple pick points.



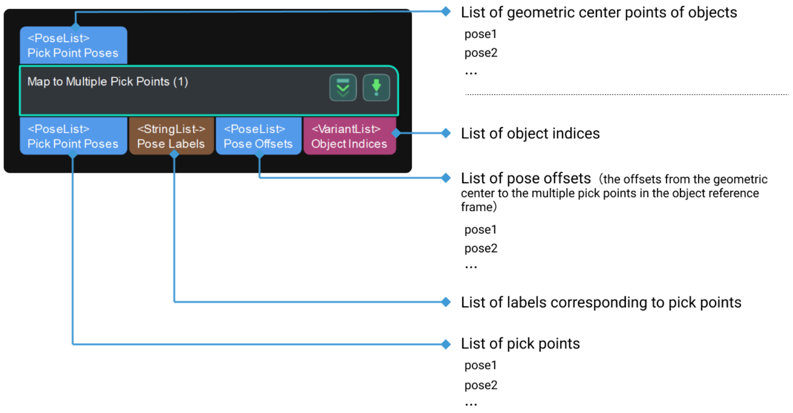
Usage Scenario

This Step is required if the objects' pick points do not coincide with their geometric center points, or if there is more than one pick point on each object. Please follow the fixed usage instructions to add multiple pick points to a scene object to send to the robot control software.

The Pick Point output port of this Step needs to be connected to the Reference Poses input port of the Transform Poses Step.

The Pose Offsets output port of this Step needs to be connected to the Original Poses input port of the Transform Poses Step.

Input and Output


<PoseList>
Pick Point Poses
Map to Multiple Pick Points (1)
<PoseList>
Pick Point Poses
<StringList>
Pose Labels
<PoseList>
Pose Offsets
<VariantList>
Object Indices
List of geometric center points of objects
pose1
pose2
...
List of object indices
List of pose offsets（the offsets from the geometric center to the multiple pick points in the object reference frame）
pose1
pose2
...
List of labels corresponding to pick points
List of pick points
pose1
pose2
...

Parameters

Pose Files Path

Geometric Center File

The pose of the object’s geometric center point on the point cloud model.

Instruction: the absolute path or relative path of the geometric center point file.

Placing Point File

Pose of object placement point.

Instruction: the absolute path or relative path of object placement point file.

Pick Points File

The pose of the object’s pick point on the point cloud model.

Instruction: the absolute or relative path of the pick point file.

Import Labels (Optional)

Pose Label File

Label file corresponding to the pick point.

Instruction: The absolute path or relative path of label file.

4.3.224. Extract Image Regions by Mask

This Step is outdated. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

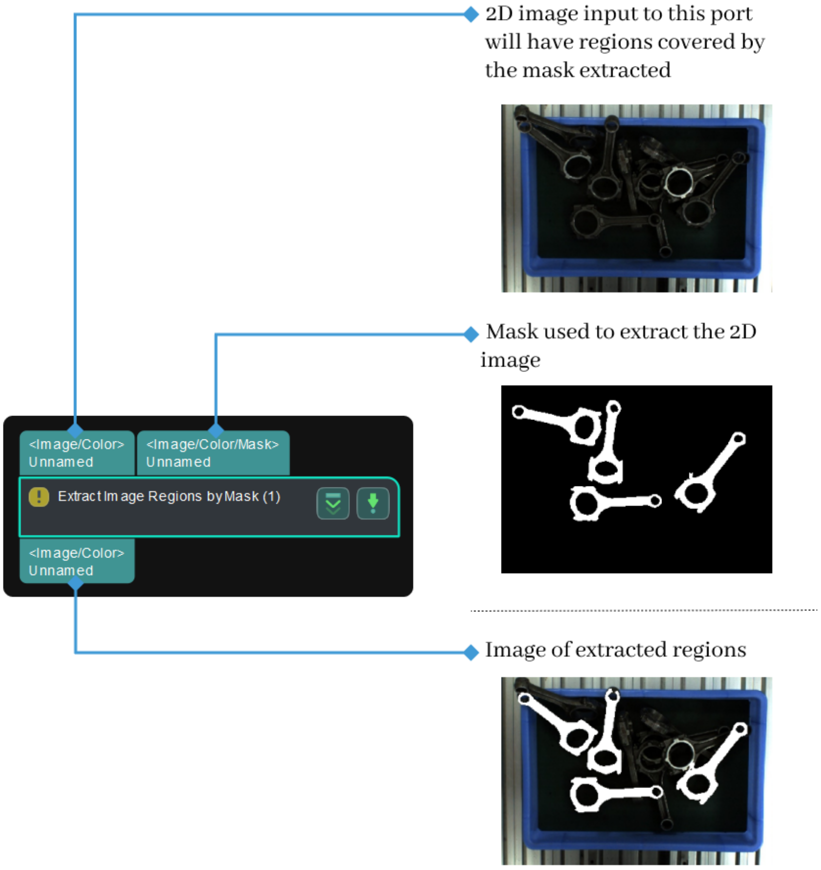
Function

Extract the regions covered by masks in the input 2D image. The pixels in other regions are set to 0.

Usage Scenario

A custom Step for special scenarios.

Input and Output



2D image input to this port will have regions covered by the mask extracted

Mask used to extract the 2D image

Image of extracted regions

4.3.225. Mask Clustering

Function

This Step clusters points whose pixel values are not zero in masks.

Usage Scenario

This Step is usually used to extract regions in an image that belong to the same object.

Input and Output

- **Input:** Masks in the input list of this port will be clustered.
- **Output:** list of masks after clustering.

4.3.226. Filter Masks

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step filters mask regions based on the number of pixels in each region.

Usage Scenario

This Step is a general mask filtering Step. No fixed usage scenarios.

Input and Output

- **Input:** Mask list input to this port will be filtered.
- **Output:** filtered list of masks.

4.3.227. Mask Gridding

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

For each input mask, this Step meshes the mask and generate multiple masks with valid points.

Usage Scenario

This Step is a custom Step for special scenarios.

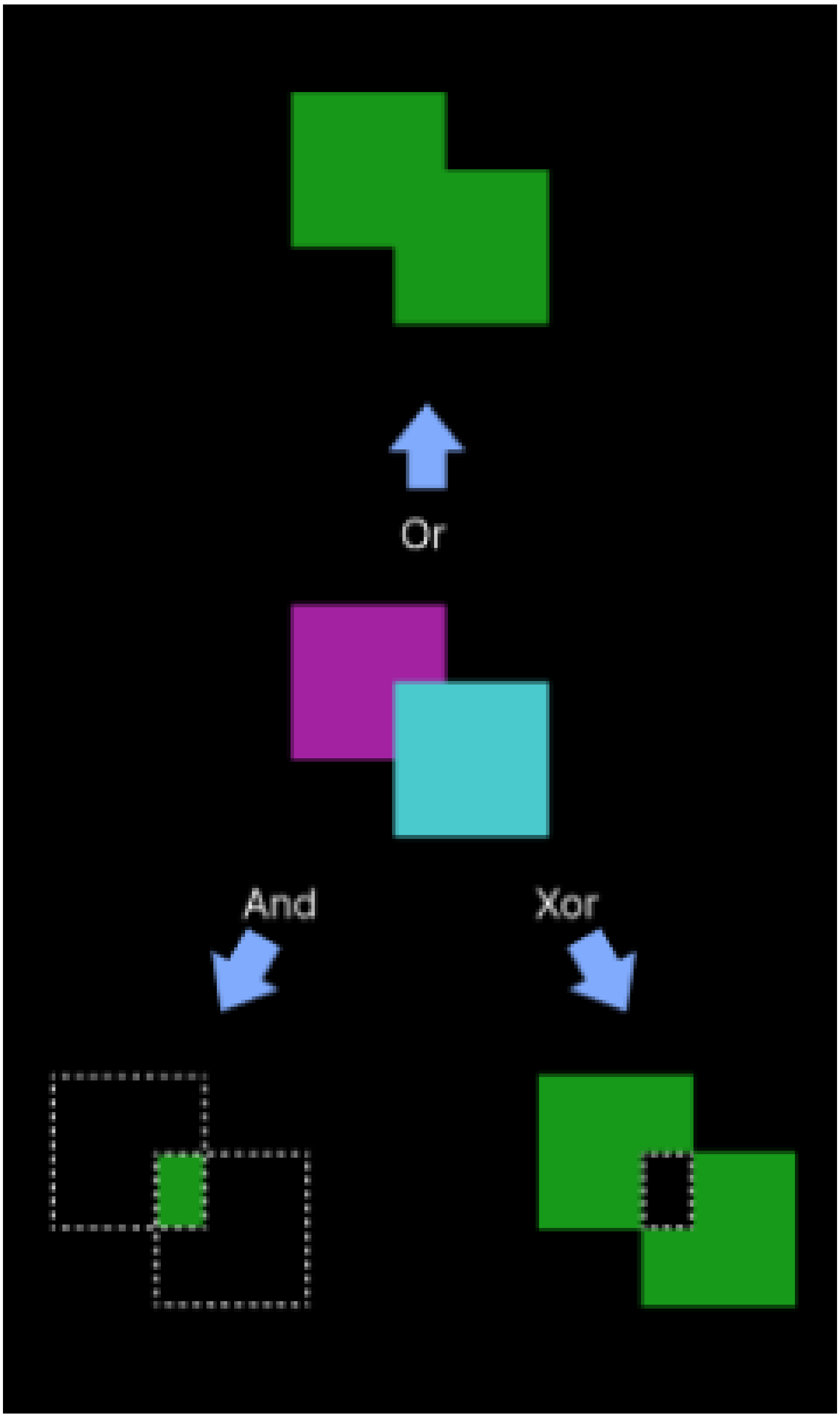
Input and Output

- **Input:** Masks input to this port will be meshed.
- **Output:** Meshed masks.

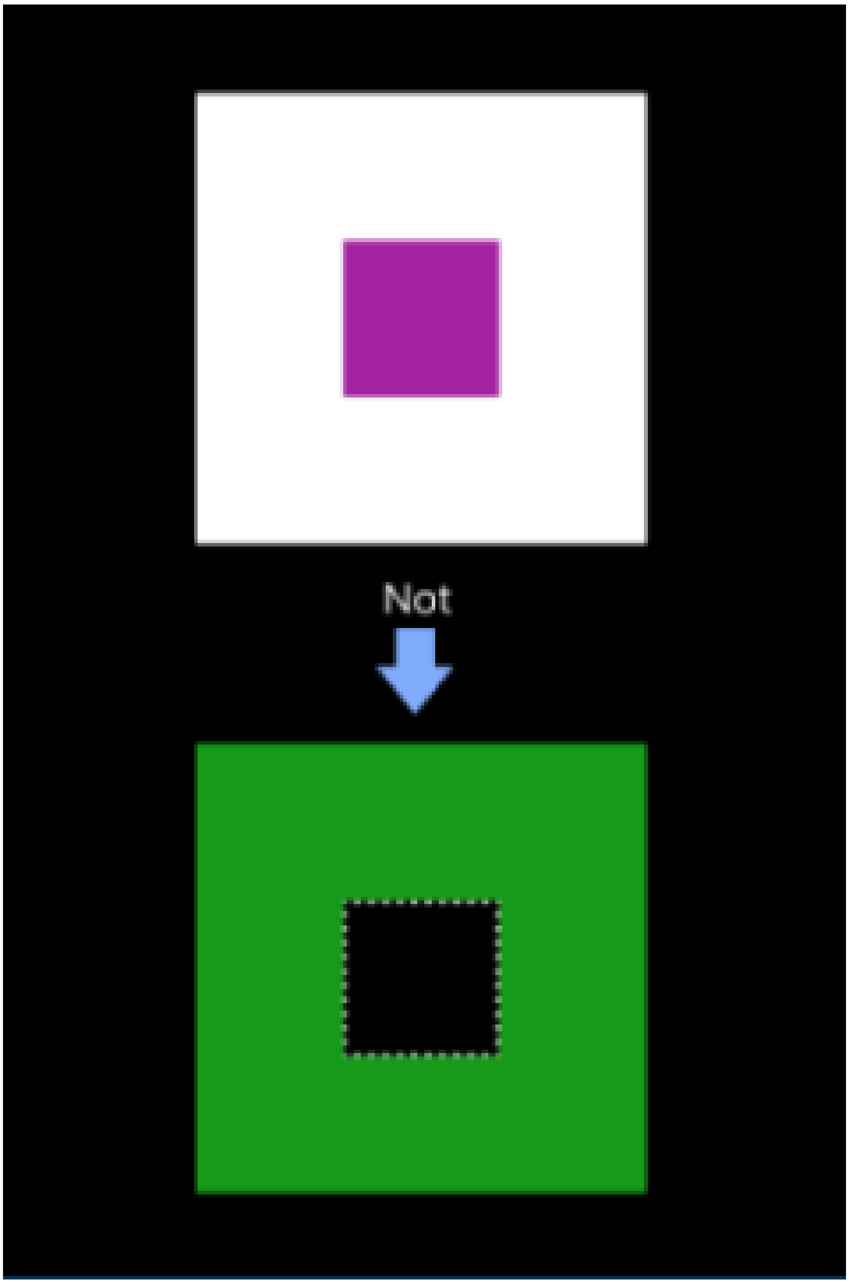
4.3.228. Mask Logical Operation

Function

This Step is used to perform logical operations on two mask lists of the same size.



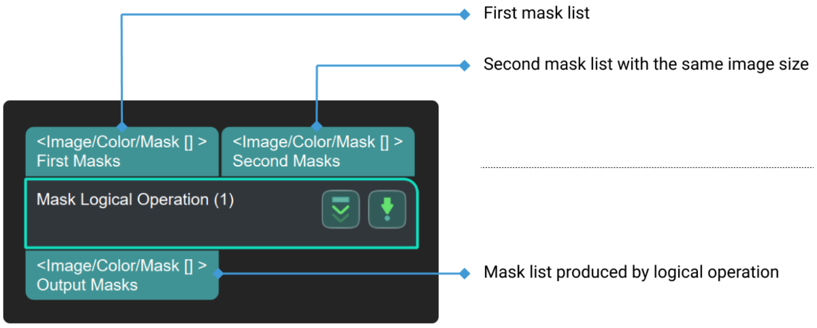
When there is only one mask, it will negate the image of the mask itself.



Usage Scenario

This Step is widely used in all scenarios concerning mask logical operations.

Input and Output



Parameter Description

Operation Settings

Logical Operation

Parameter description: This parameter is used to select the type of logical operation on masks.

Options:

- And: The default option. Extract the shared part of the two mask lists.
- Or: Combine the two mask lists input.
- Xor: Remove the shared part of two mask lists.
- Not: Negate an input mask, turning the black part (usually the background) into white and the white part (usually the contour or edges of the object) into black.

Usage Settings of Input Masks

The parameter setting is not available when the type of **Logical Operation** is **Not**.

Correspondence Type

Parameter description: This parameter is used to specify the correspondence between masks in the two input mask lists.

Options:

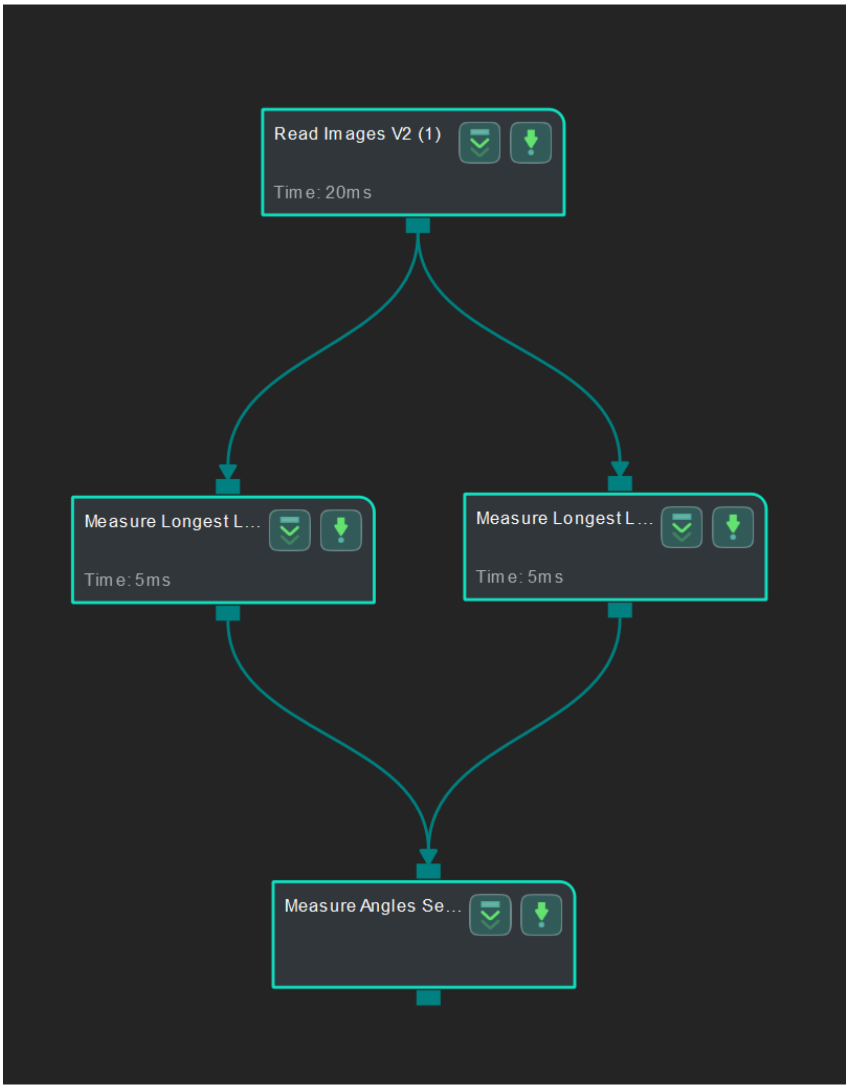
- FirstToAll: The first mask in the second mask list will correspond to all masks in the first mask list.
- OneToOne (default value): All masks in the second mask list will correspond to all masks in the first mask list one by one. Use all masks of the second mask group to correspond to those in the first mask group.

4.3.229. Measure Angles Segments to Segments

This Step is used to measure the angles between line segments.

Before using this Step, please refer to Getting Started with Measurement Mode to learn about basics of Measurement Mode.

Sample Application



Connect Read Images V2, Measure Longest Line Segment, and Measure Angles Segments to Segments to measure the angle between two line segments.

Connect Steps

1. Click Measure Longest Line Segment (1) , go to the **Step Input Source Selection** panel, and then select Read Images V2_1_Color Image as Input 1 (Color Image).
2. Click Measure Longest Line Segment (2) , go to the **Step Input Source Selection** panel, and then select Read Images V2_1_Color Image as Input 1 (Color Image).
3. Click Measure Angles Segments to Segments, go to the **Step Input Source Selection** panel, and then select Measure Longest Line Segment_1_Line Segments as Input 2 (Line Segments) and Measure Longest Line Segment_2_Line Segments as Input 3 (Line Segments).

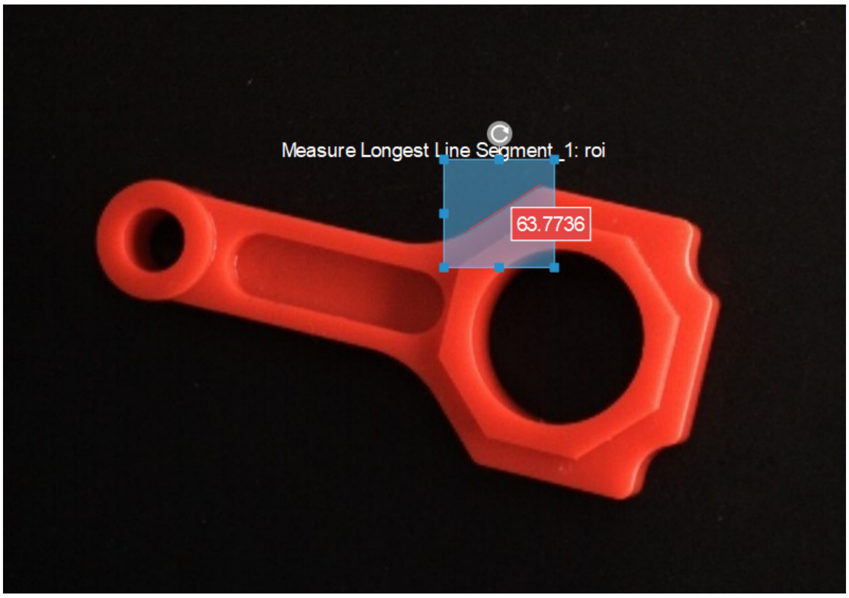
Configure parameters and read the result

- Configure in Read Images V2

Click the Step and configure the Image Source, Image File/Folder Path, and other parameters in the **Step Parameters** panel. After the configuration, click **[Run]** to run the project.

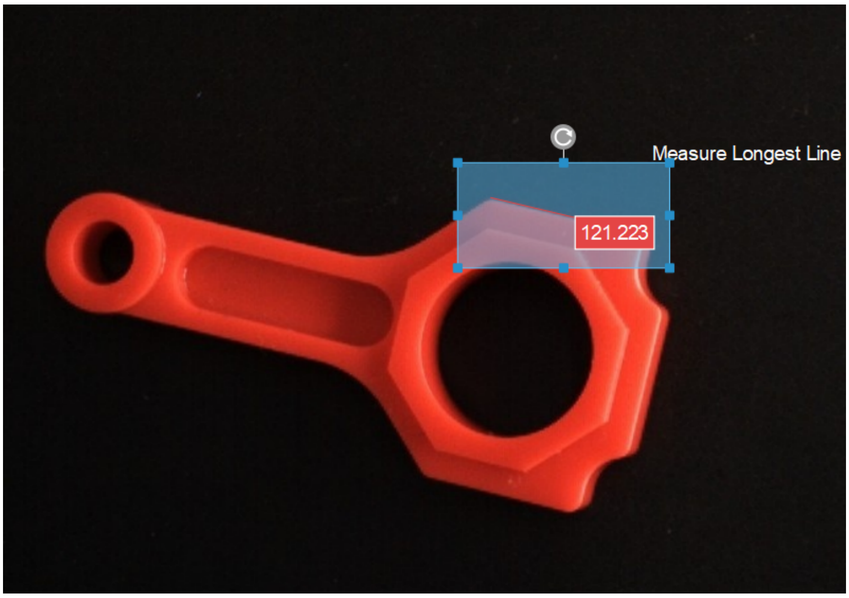
- Configure in Measure Longest Line Segment (1)

Click the Step and select an ROI in the sketchpad area.

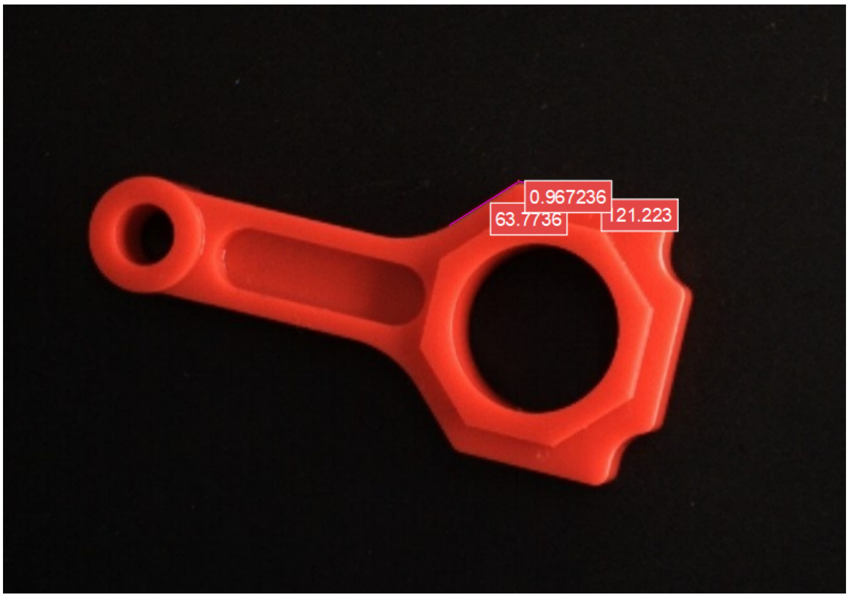


- Configure in Measure Longest Line Segment (2)

Click the Step and select an ROI in the sketchpad area.



- The measurement result will be displayed in **Measurement Output** and **Result View** panels as well. You can also set the LSL (lower specification limit) and USL (upper specification limit) to check whether the measurement conforms to the specification.

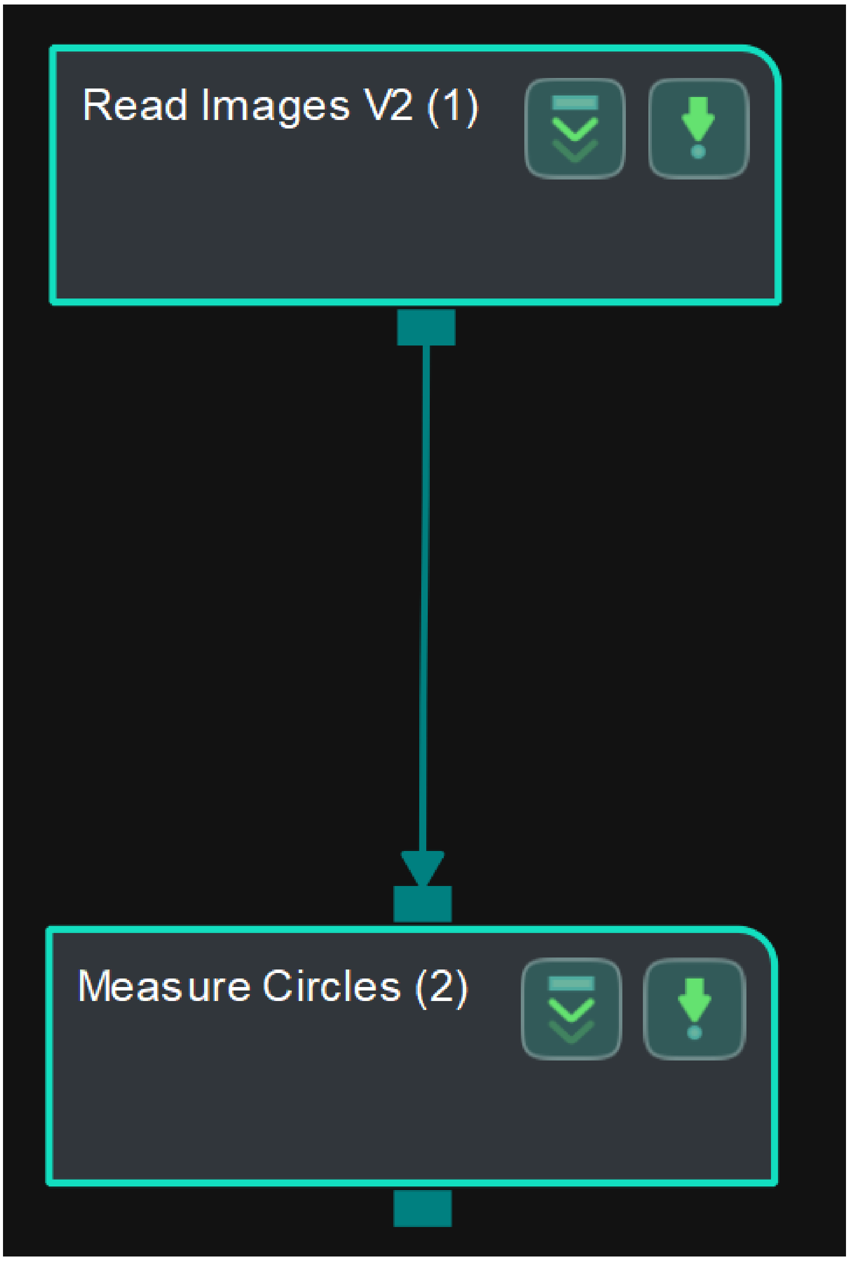


4.3.230. Measure Circles

This Step is mainly used to measure the radius of a circle.

Before using this Step, please refer to Getting Started with Measurement Mode to learn about basics of Measurement Mode.

Sample Application



Connect Read Images V2 with Measure Circles to locate the circle in the ROI and measure its radius.

Connect Steps

Click Measure Circles, go to the **Step Input Source Selection** panel, and then select Read Images V2_1_Color Image as Input 1 (Color Image).

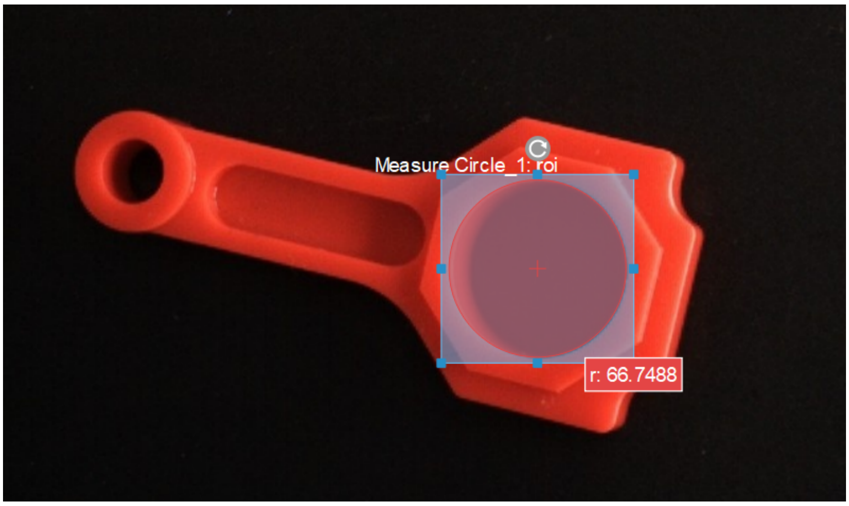
Configure parameters and read the result

- Configure in Read Images V2

Click the Step and configure the Image Source, Image File/Folder Path, and other parameters in the **Step Parameters** panel. Click **[Run]** to run the project after the settings.

- Configure in Measure Circles

Click the Step and select an ROI in the sketchpad area.



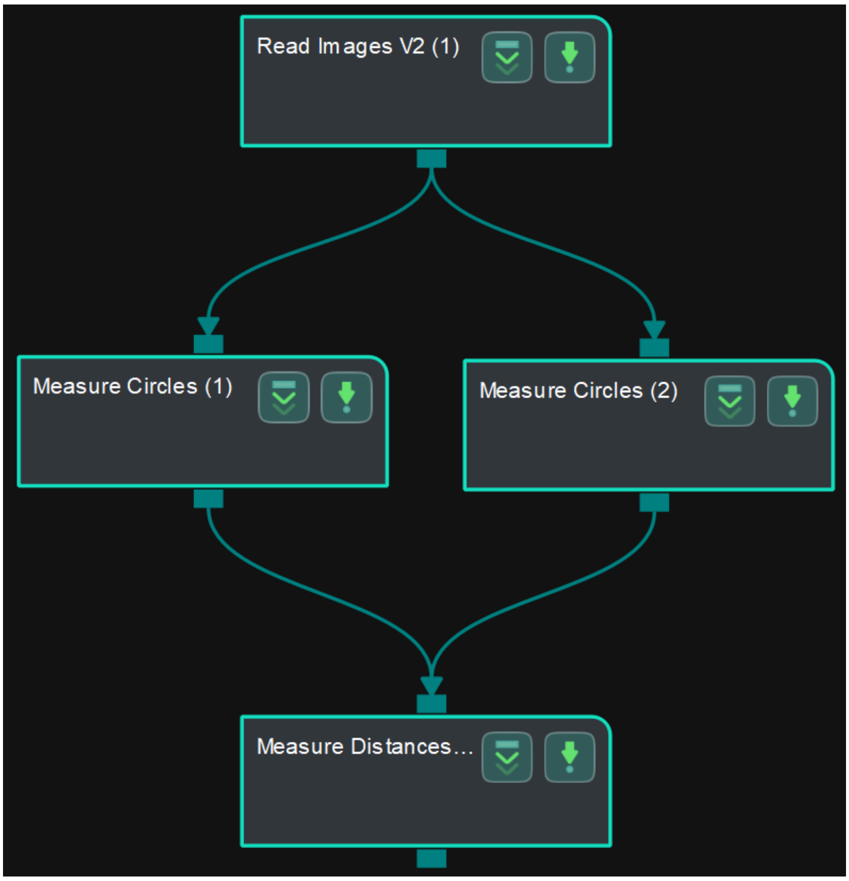
- The measured radius (in pixels) will be displayed in **Measurement Output** and **Result View** panel. You can also set the LSL (lower specification limit) and USL (upper specification limit) to check whether the measurement conforms to the specification.

4.3.231. Measure Distances Circles to Circles

This Step is used to measure the distances from circles to circles.

Before using this Step, please refer to Getting Started with Measurement Mode to learn about basics of Measurement Mode.

Sample Application



Connect Read Images V2, Measure Circles, and Measure Distances Circles to Circles to measure the distance between the centers of two circles or the nearest distance between two circles.

Connect Steps

1. Click Measure Circles (1), go to the **Step Input Source Selection** panel, and then select Read Images V2_1_Color Image as Input 1 (Color Image).
2. Click Measure Circles (2), go to the **Step Input Source Selection** panel, and then select Read Images V2_1_Color Image as Input 1 (Color Image).
3. Click Measure Distances Circles to Circles, go to the **Step Input Source Selection** panel, and then select Measure Circles_1_Circles as Input 2 (Circles) and Measure Circles_2_Circles as Input 3 (Circles).

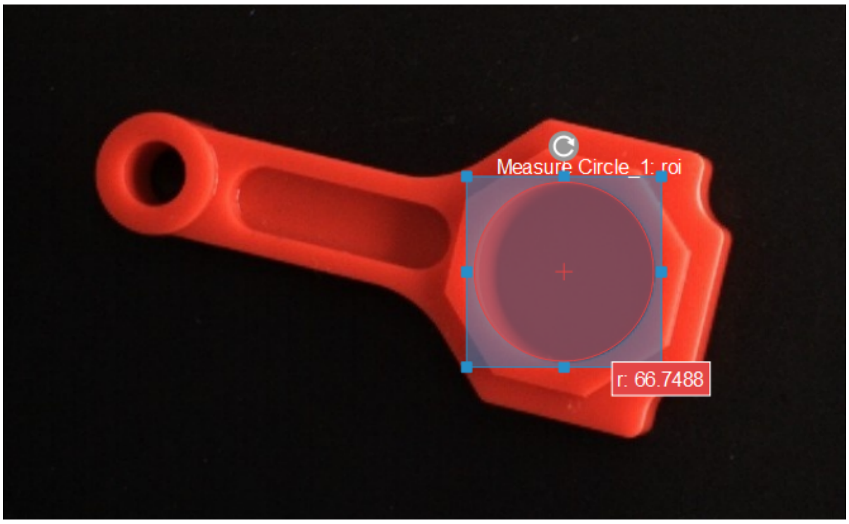
Configure parameters and read the result

- Configure in Read Images V2

Click the Step and configure the Image Source, Image File/Folder Path, and other parameters in the **Step Parameters** panel. Click **[Run]** to run the project after the settings.

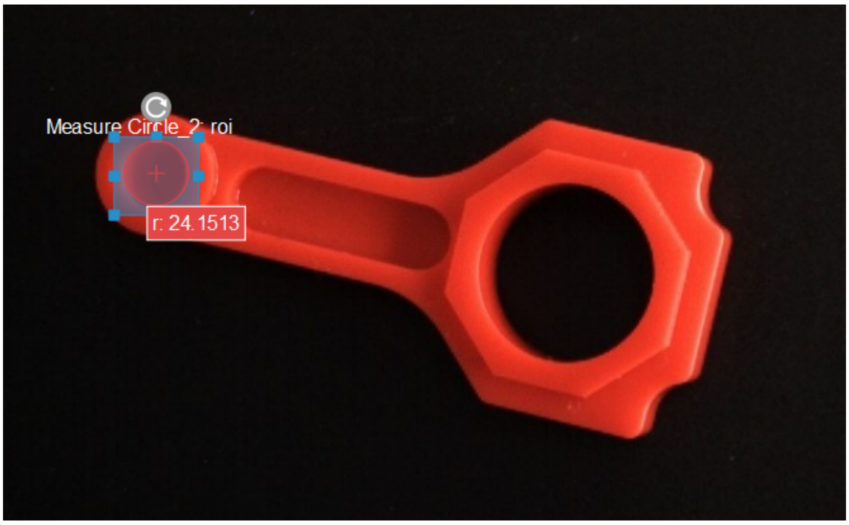
- Configure in Measure Circles (1)

Click the Step and select an ROI in the sketchpad area.



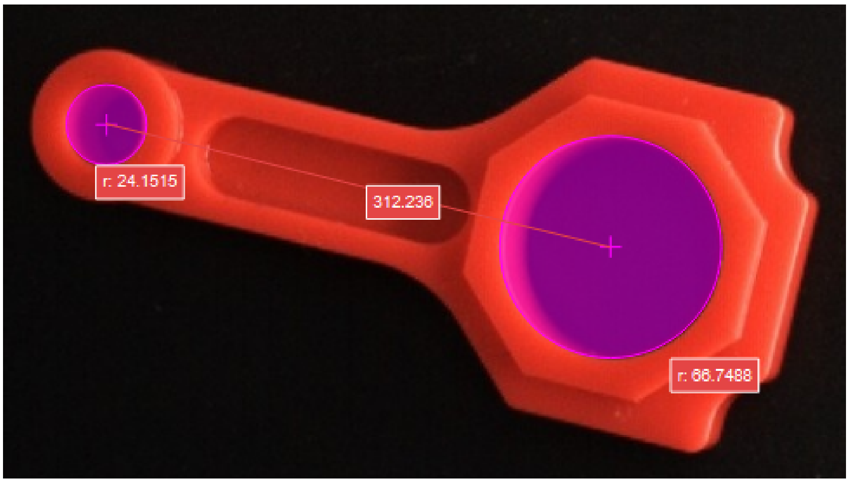
- Configure in Measure Circles (2)

Click the Step and select an ROI in the sketchpad area.

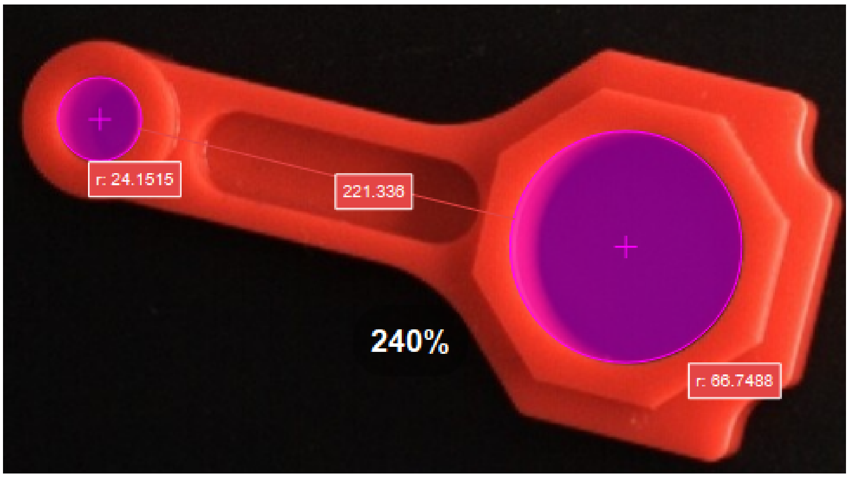


- Configure in Measure Distances Circles to Circles

Click the Step, and the distance (in pixels) between the centers of two circles by default will be displayed in the sketchpad area.



You can switch the Distance Mode to Nearest in the Step Parameters panel. Run the project again to measure the nearest distance from the circle to the line segment.



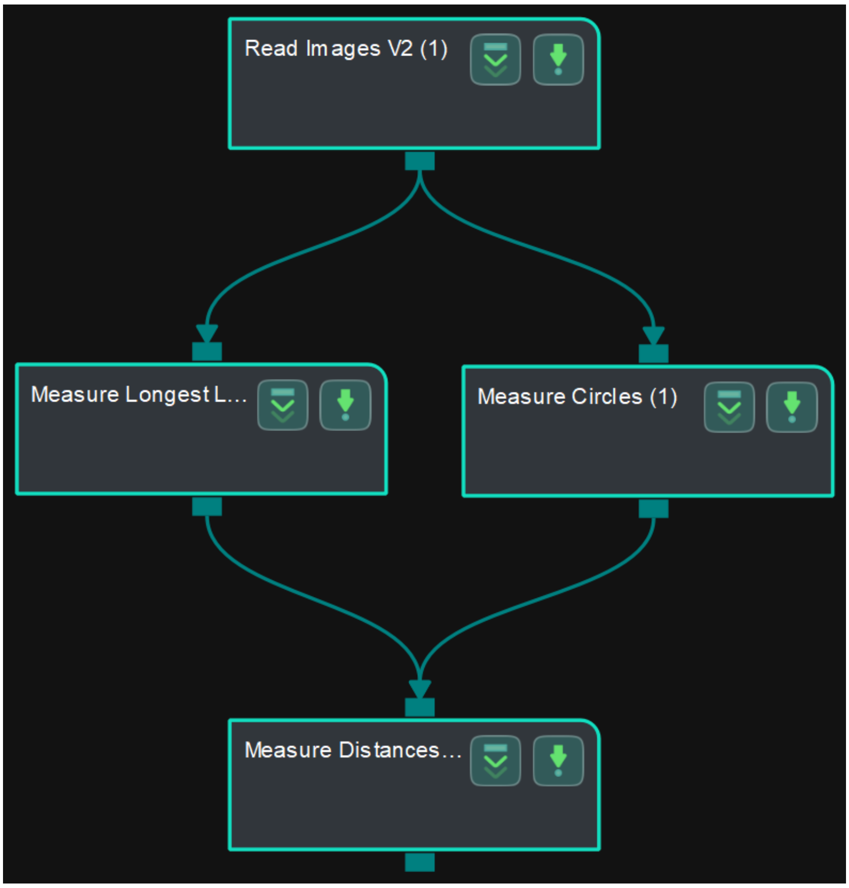
- The measurement result will be displayed in **Measurement Output** and **Result View** panels as well. You can also set the LSL (lower specification limit) and USL (upper specification limit) to check whether the measurement conforms to the specification.

4.3.232. Measure Distances Circles to Segments

This Step is used to measure the distances from circles to line segments.

Before using this Step, please refer to Getting Started with Measurement Mode to learn about basics of Measurement Mode.

Sample Application



Connect Read Images V2, Measure Circles, Measure Longest Line Segment, and Measure Distances Circles to Segments to measure the distance from circles to line segments.

Connect Steps

1. Click Measure Circles, go to the **Step Input Source Selection** panel, and then select Read Images V2_1_Color Image as Input 1 (Color Image).
2. Click Measure Longest Line Segment, go to the **Step Input Source Selection** panel, and then select Read Images V2_1_Color Image as Input 1 (Color Image).
3. Click Measure Distances Circles to Segments, go to the **Step Input Source Selection** panel, and then select Measure Circles_1_Circles as Input 2 (Circles) and Measure Longest Line Segment_1_Line Segments as Input 3 (Line Segments).

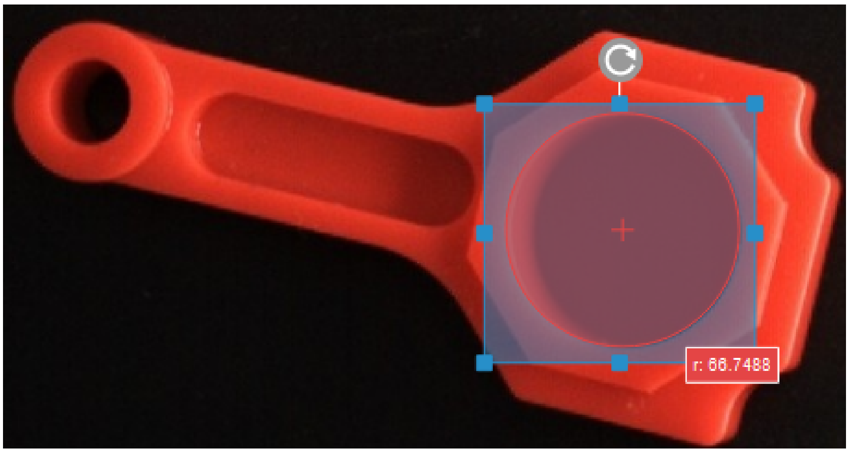
Configure parameters and read the result

- Configure in Read Images V2

Click the Step and configure the Image Source, Image File/Folder Path, and other parameters in the **Step Parameters** panel. Click **[Run]** to run the project after the settings.

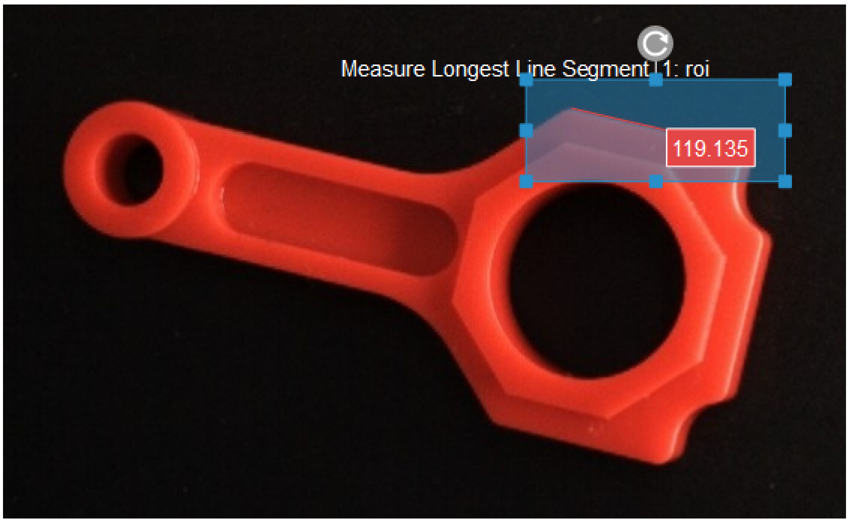
- Configure in Measure Circles

Click the Step and select an ROI in the sketchpad area.



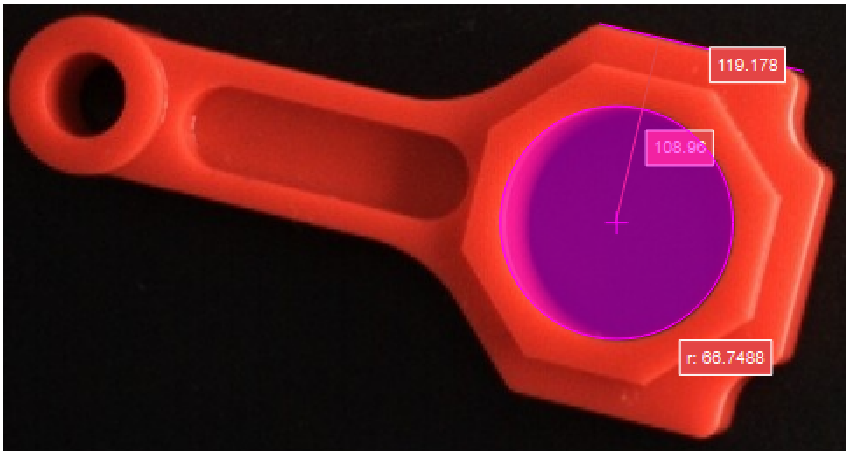
- Configure in Measure Longest Line Segment

Click the Step and select an ROI in the sketchpad area.

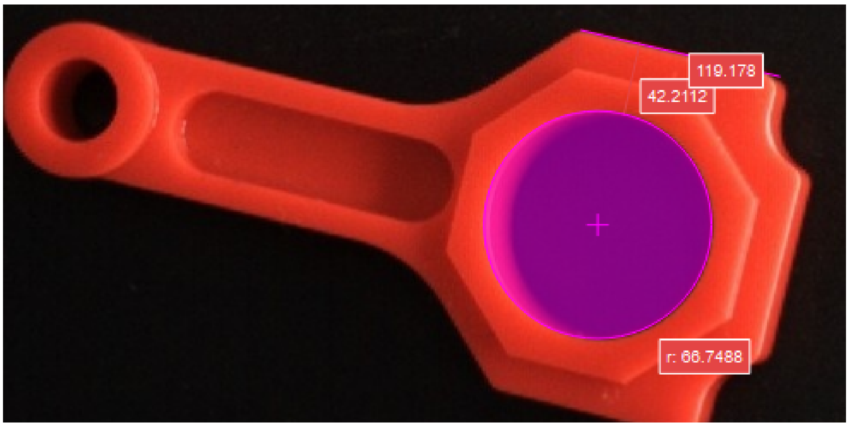


- Configure in Measure Distances Circles to Segments

Click the Step, and the distance (in pixels) from the center of the circle to the line segment by default will be displayed in the sketchpad area.



You can switch the Distance Mode to Nearest in the Step Parameters panel. Run the project again to measure the nearest distance from the circle to the line segment.



- The measurement result will be displayed in **Measurement Output** and **Result View** panels as well. You can also set the LSL (lower specification limit) and USL (upper specification limit) to check whether the measurement conforms to the specification.

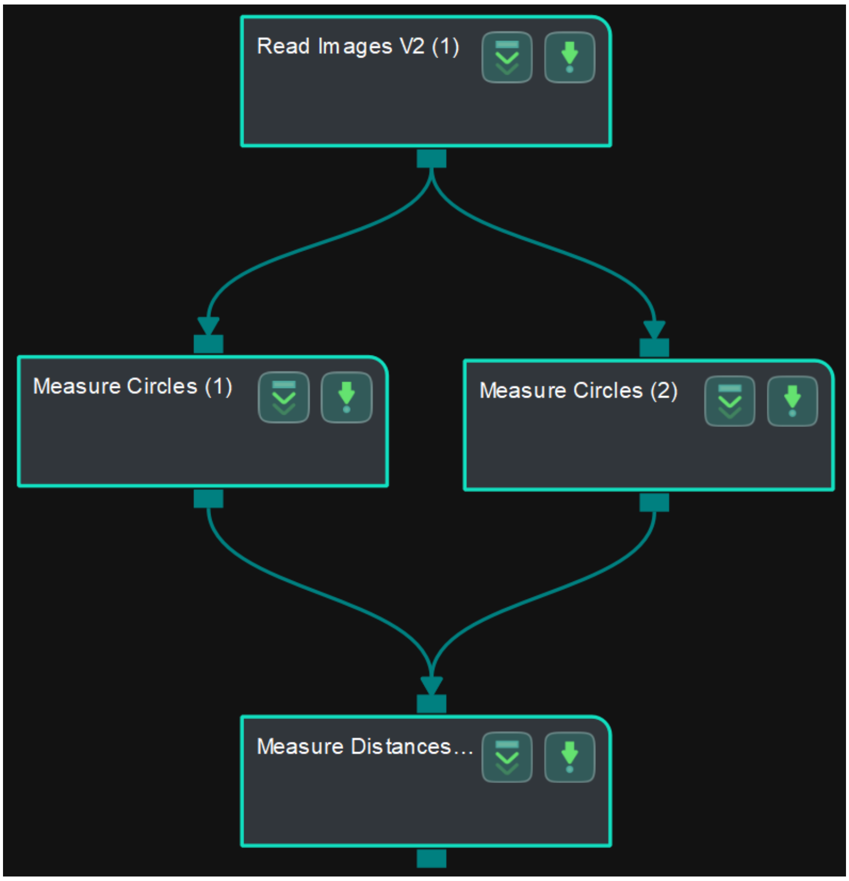
4.3.233. Measure Distances Points to Circles

This Step is used to measure the distances from points to circles.

Before using this Step, please refer to Getting Started with Measurement Mode to learn about basics of Measurement Mode.

> This Step does not support custom specified points, the points can only be obtained from Measure Circles or Detect Circle Centers.

Sample Application



Connect Read Images V2, Measure Circles, and Measure Distances Points to Circles to measure the distance from points to circles.

Connect Steps

1. Click Measure Circles (1), go to the **Step Input Source Selection** panel, and then select Read Images V2_1_Color Image as Input 1 (Color Image).
2. Click Measure Circles (2), go to the **Step Input Source Selection** panel, and then select Read Images V2_1_Color Image as Input 1 (Color Image).
3. Click on Measure Distances Points to Circles, go to the **Step Input Source Selection** panel, and then select Measure Circles_1_Circles as Input 2 (Points) and Measure Circles_2_Circles as Input 3 (Circles).

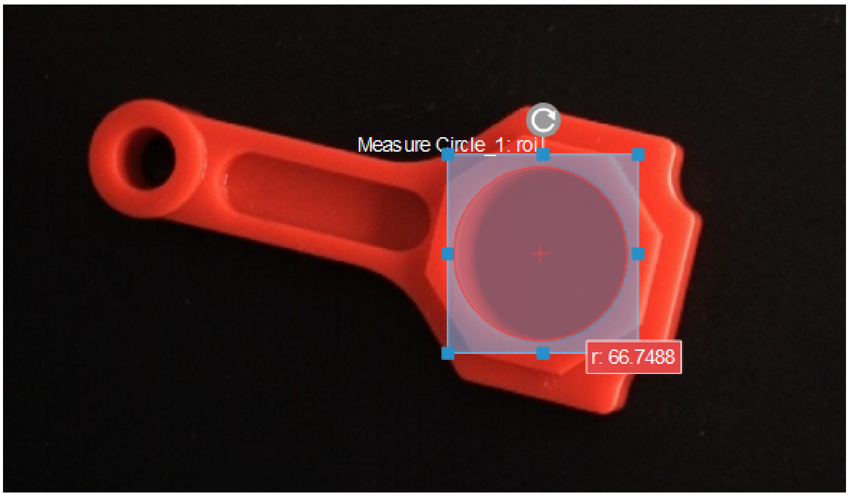
Configure parameters and read the result

- Configure in Read Images V2

Click the Step and configure the Image Source, Image File/Folder Path, and other parameters in the **Step Parameters** panel. Click **[Run]** to run the project after the settings.

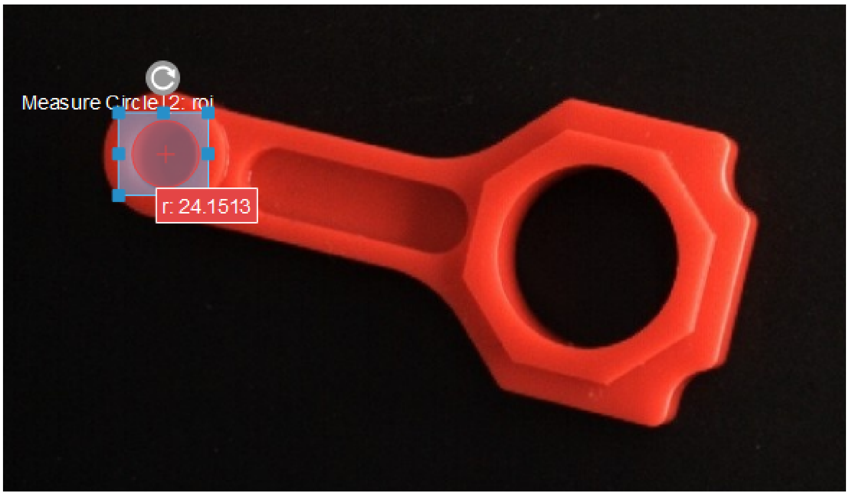
- Configure in Measure Circles (1)

Click the Step and select an ROI in the sketchpad area.



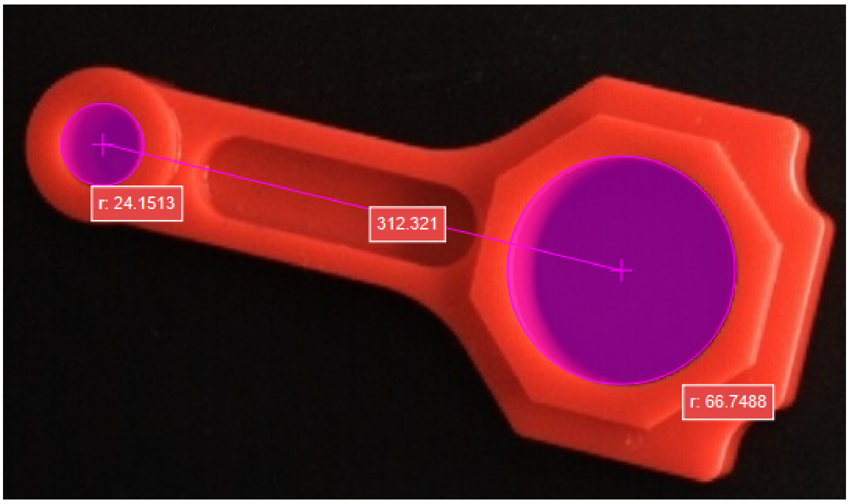
- Configure in Measure Circles (2)

Click the Step and select an ROI in the sketchpad area.



- Configure in Measure Distances Points to Circles

Click the Step, and the distance (in pixels) between the centers of two circles will be displayed in the sketchpad area.



- The measurement result will be displayed in **Measurement Output** and **Result View** panels as well. You can also set the LSL (lower specification limit) and USL (upper specification limit) to check whether the measurement conforms to the specification.

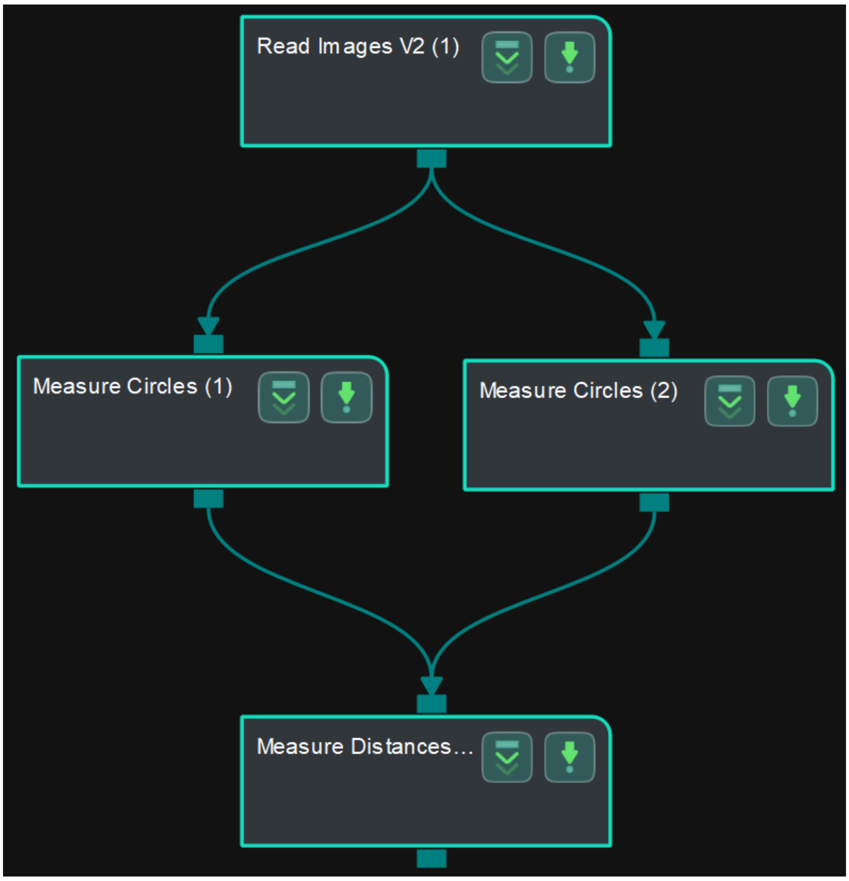
4.3.234. Measure Distances Points to Points

This Step is used to measure the distances between two points.

Before using this Step, please refer to Getting Started with Measurement Mode to learn about basics of Measurement Mode.

> This Step does not support custom specified points, the points can only be obtained from Measure Circles or Detect Circle Centers.

Sample Application



Connect Read Images V2, Measure Circles, and Measure Distances Points to Points to measure the distance between two points.

Connect Steps

1. Click Measure Circles (1), go to the **Step Input Source Selection** panel, and then select Read Images V2_1_Color Image as Input 1 (Color Image).
2. Click Measure Circles (2), go to the **Step Input Source Selection** panel, and then select Read Images V2_1_Color Image as Input 1 (Color Image).
3. Click Measure Distances Points to Points, go to the **Step Input Source Selection** panel, and then select Measure Circles_1_Circles as Input 2 (Points) and Measure Circles_2_Circles as Input 3 (Points).

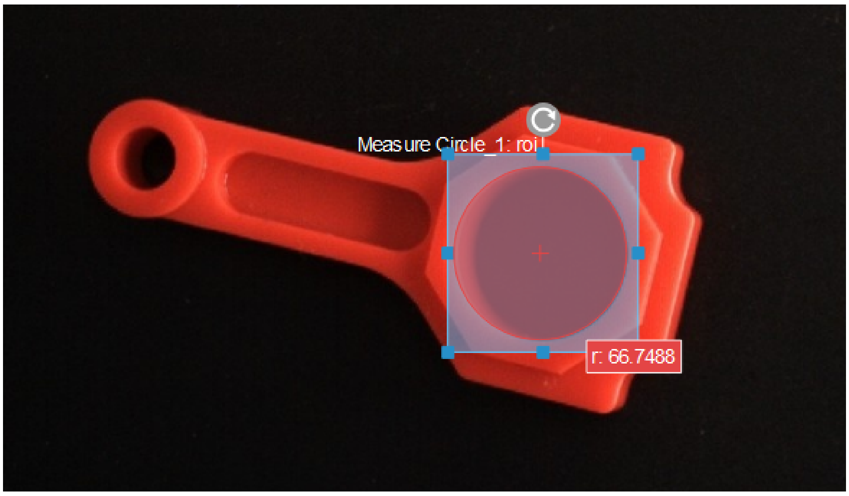
Configure parameters and read the result

- Configure in Read Images V2

Click the Step and configure the Image Source, Image File/Folder Path, and other parameters in the **Step Parameters** panel. Click **[Run]** to run the project after the settings.

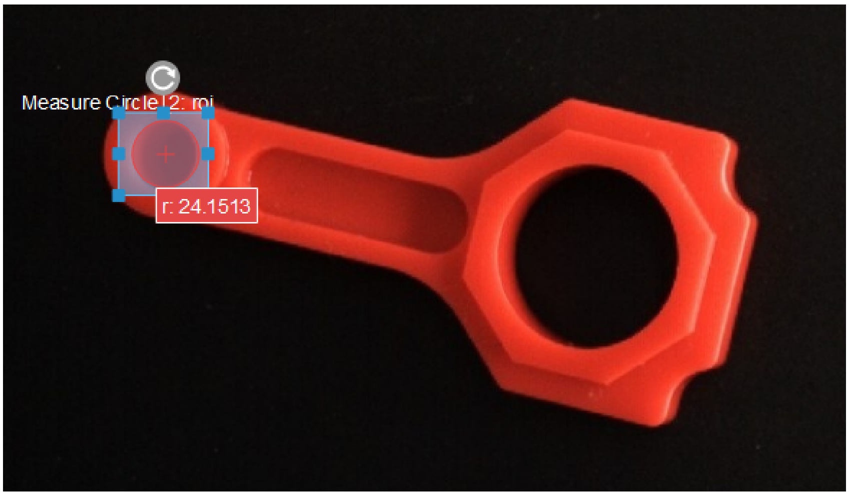
- Configure in Measure Circles (1)

Click the Step and select an ROI in the sketchpad area.



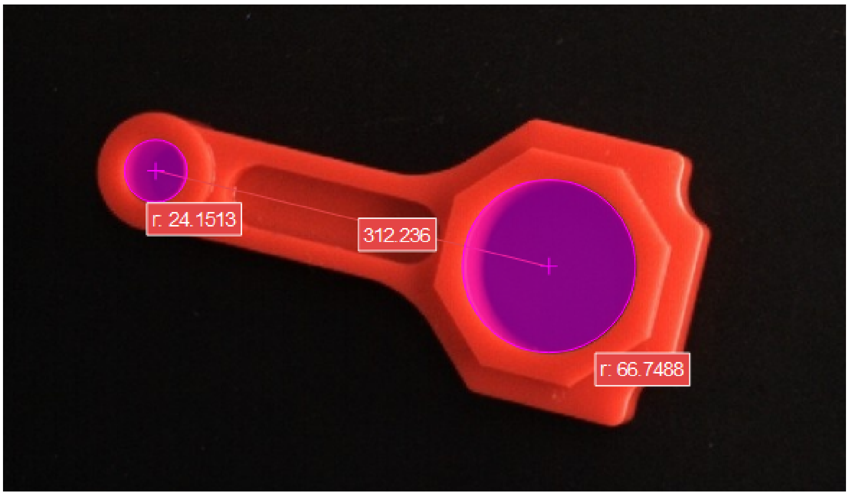
- Configure in Measure Circles (2)

Click the Step and select an ROI in the sketchpad area.



- Configure in Measure Distances Points to Points

Click the Step, and the distance (in pixels) between the centers of two circles will be displayed in the sketchpad area.



- The measurement result will be displayed in **Measurement Output** and **Result View** panels as well. You can also set the LSL (lower specification limit) and USL (upper specification limit) to check whether the measurement conforms to the specification.

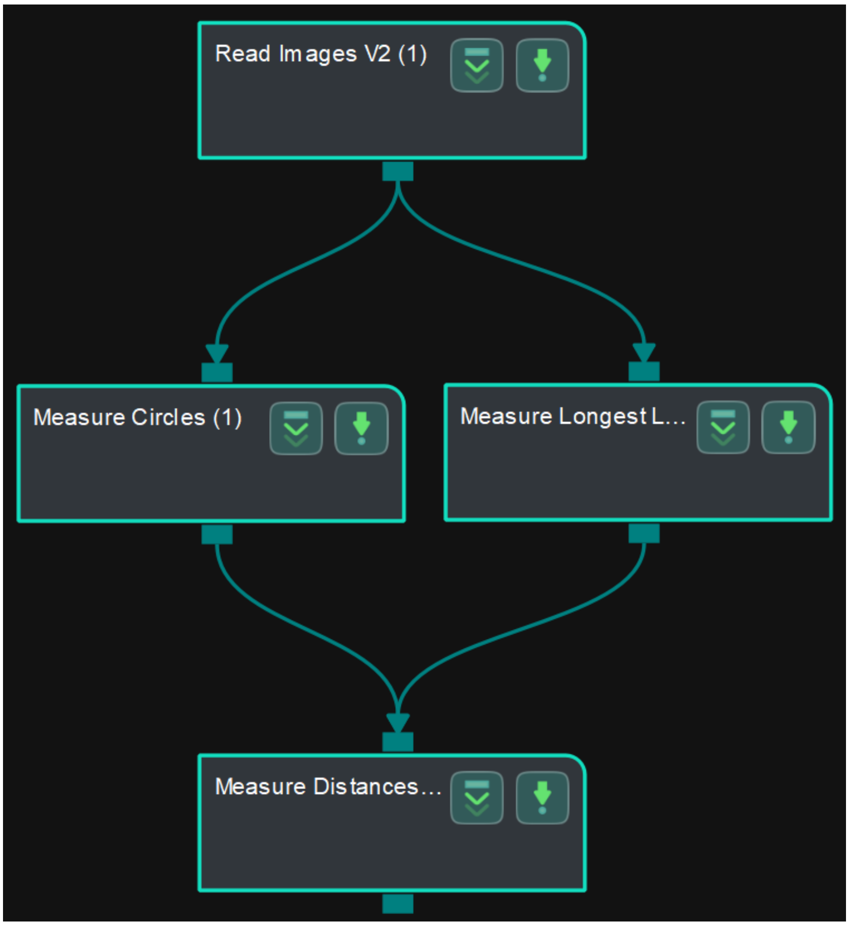
4.3.235. Measure Distances Points to Segments

This Step is used to measure the distances from points to line segments.

Before using this Step, please refer to Getting Started with Measurement Mode to learn about basics of Measurement Mode.

> This Step does not support custom specified points, the points can only be obtained from Measure Circles or Detect Circle Centers.

Sample Application



Connect Read Images V2, Measure Circles, Measure Longest Line Segment, and Measure Distances Points to Segments to measure the distance from points to line segments.

Connect Steps

1. Click Measure Circles, go to the **Step Input Source Selection** panel, and then select Read Images V2_1_Color Image as Input 1 (Color Image).
2. Click Measure Longest Line Segment, go to the **Step Input Source Selection** panel, and then select Read Images V2_1_Color Image as Input 1 (Color Image).
3. Click Measure Distances Points to Segments, go to the **Step Input Source Selection** panel, and

then select Measure Circles_1_Circles as Input 2 (Points) and Measure Longest Line Segment_1_Line Segments as Input 3 (Line Segments).

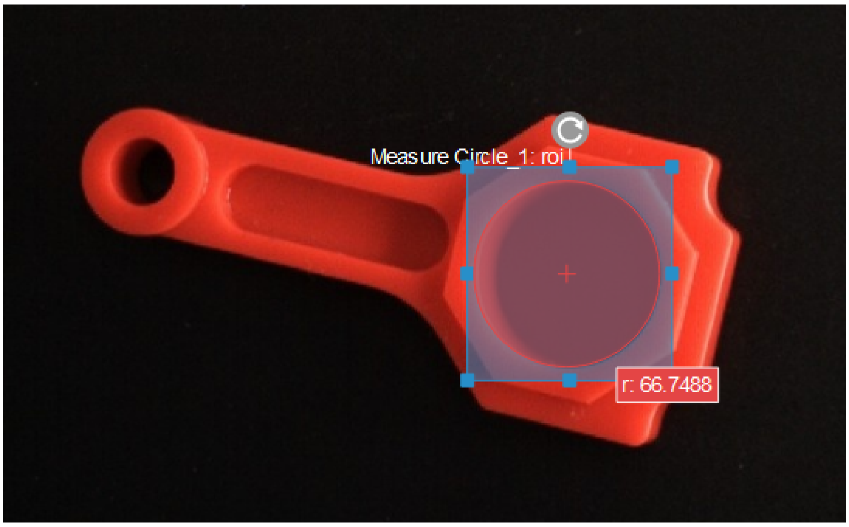
Configure parameters and read the result

- Configure in Read Images V2

Click the Step and configure the Image Source, Image File/Folder Path, and other parameters in the **Step Parameters** panel. Click **[Run]** to run the project after the settings.

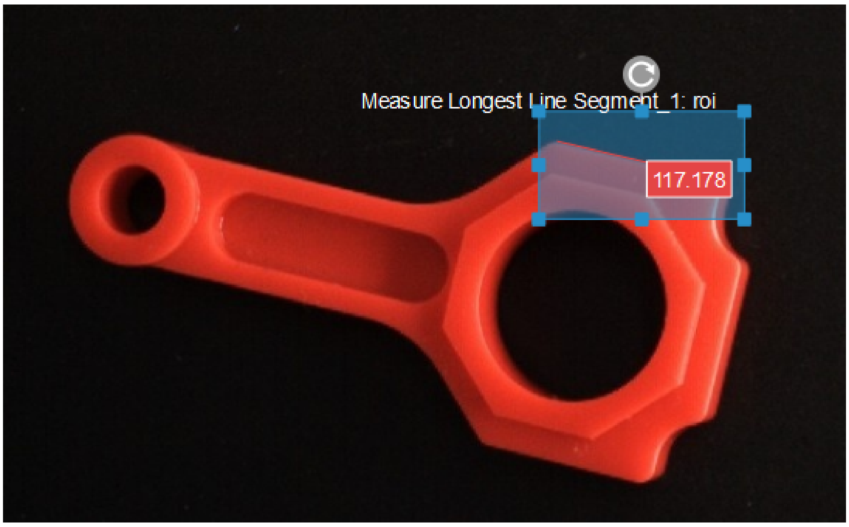
- Configure in Measure Circles

Click the Step and select an ROI in the sketchpad area.



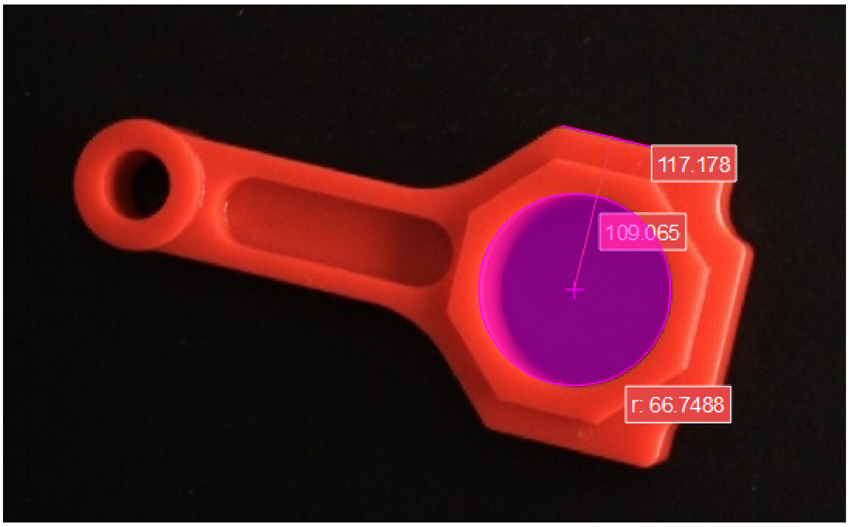
- Configure in Measure Longest Line Segment

Click the Step and select an ROI in the sketchpad area.



- Configure in Measure Distances Points to Segments

Click the Step, and the distance (in pixels) from the center of the circle to the line segment by default will be displayed in the sketchpad area.



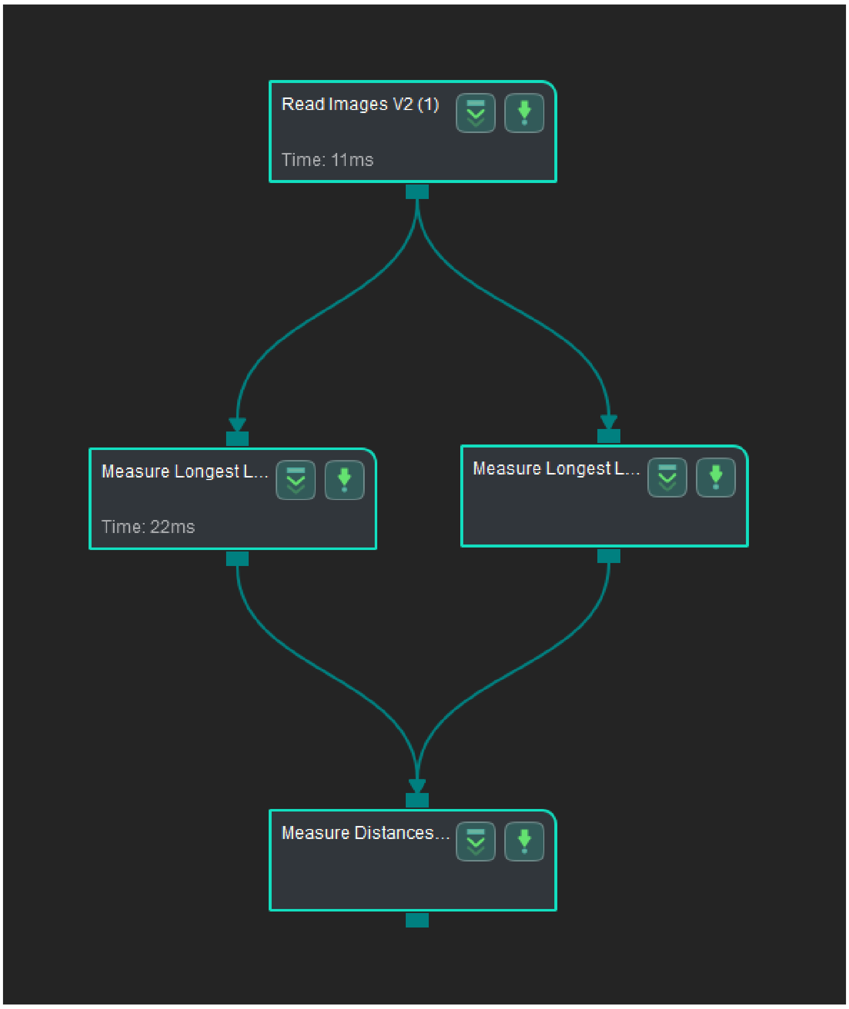
- The measurement result will be displayed in **Measurement Output** and **Result View** panels as well. You can also set the LSL (lower specification limit) and USL (upper specification limit) to check whether the measurement conforms to the specification.

4.3.236. Measure Distances Segments to Segments

This Step is used to measure the distances between line segments.

Before using this Step, please refer to Getting Started with Measurement Mode to learn about basics of Measurement Mode.

Sample Application



Connect Read Images V2, Measure Longest Line Segment, and Measure Distances Segments to Segments to measure the distance between line segments.

Connect Steps

1. Click on Measure Longest Line Segment (1), go to the **Step Input Source Selection** panel, and then select Read Images V2_1_Color Image as Input 1 (Color Image).
2. Click on Measure Longest Line Segment (2), go to the **Step Input Source Selection** panel, and then select Read Images V2_1_Color Image as Input 1 (Color Image).
3. Click on Measure Distances Segments to Segments, go to the **Step Input Source Selection** panel, and then select Measure Longest Line Segment_1_Line Segments as Input 2 (Line Segments) and Measure Longest Line Segment_2_Line Segments as Input 3 (Line Segments).

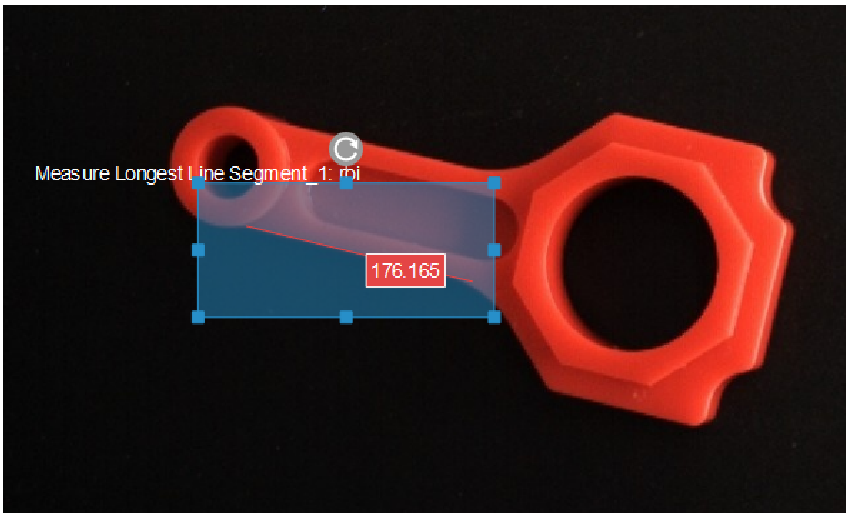
Configure parameters and read the result

- Configure in Read Images V2

Click the Step and configure the Image Source, Image File/Folder Path, and other parameters in the **Step Parameters** panel. Click **[Run]** to run the project after the settings.

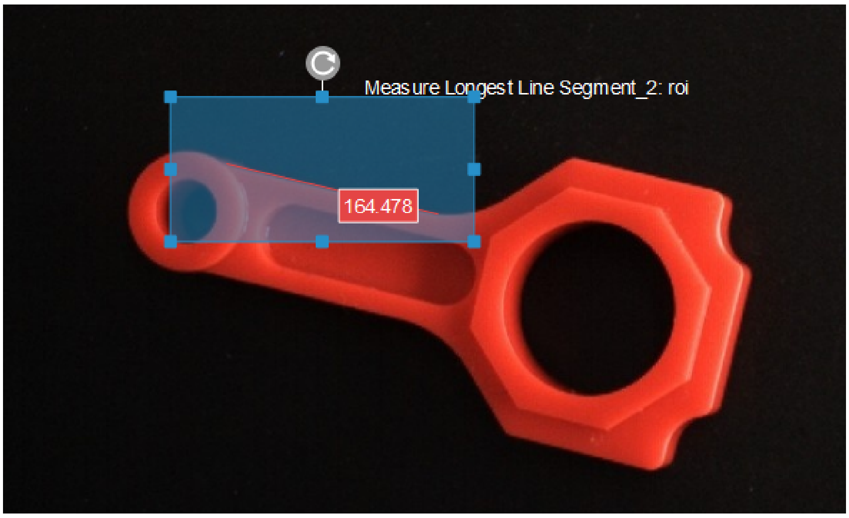
- Configure in Measure Longest Line Segment (1)

Click the Step and select an ROI in the sketchpad area.



- Configure in Measure Longest Line Segment (2)

Click the Step and select an ROI in the sketchpad area.



- Configure in Measure Distances Segments to Segments

Click the Step, and then the distance (in pixels) between line segments will be displayed in the sketchpad area.

- The measurement result will be displayed in **Measurement Output** and **Result View** panels as well. You can also set the LSL (lower specification limit) and USL (upper specification limit) to check whether the measurement conforms to the specification.

4.3.237. Measure Gap Width

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

In a 2D image, this Step measures the width of a gap in the target object.

Usage Scenario

This Step is a custom Step for special scenarios.

Input and Output

- Input:
  1. Color image input at this port will be used as the visualization background.
  2. 2D pose of the target object (optional, used to fit the template to the original image; if the image content of the selected ROI coincides with the template, no input is required).
  3. Physical size represented by a single pixel (optional, if this port is not used, the output size will be in pixels).
- Output:
  1. Image for visualization.
  2. Two line segments representing the two edges of the gap.
  3. Gap width.

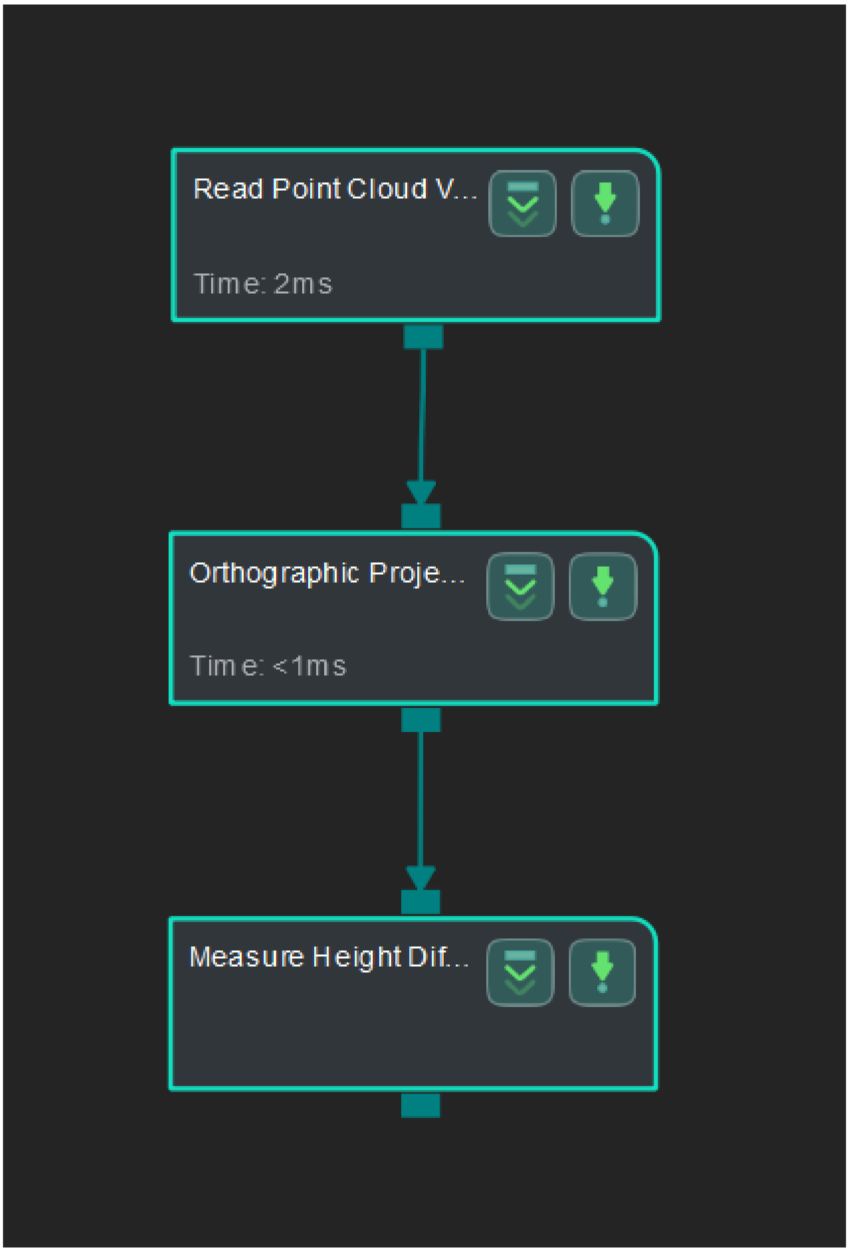
4.3.238. Measure Height Difference Point to Point

This Step is used to calculate the height difference from the detect point to the base point.

A line segment connecting the two specified points will be automatically drawn, and the variation of heights along the line will be shown in the height view widget.

Before using this Step, please refer to Getting Started with Measurement Mode to learn about basics of Measurement Mode.

Application Examples



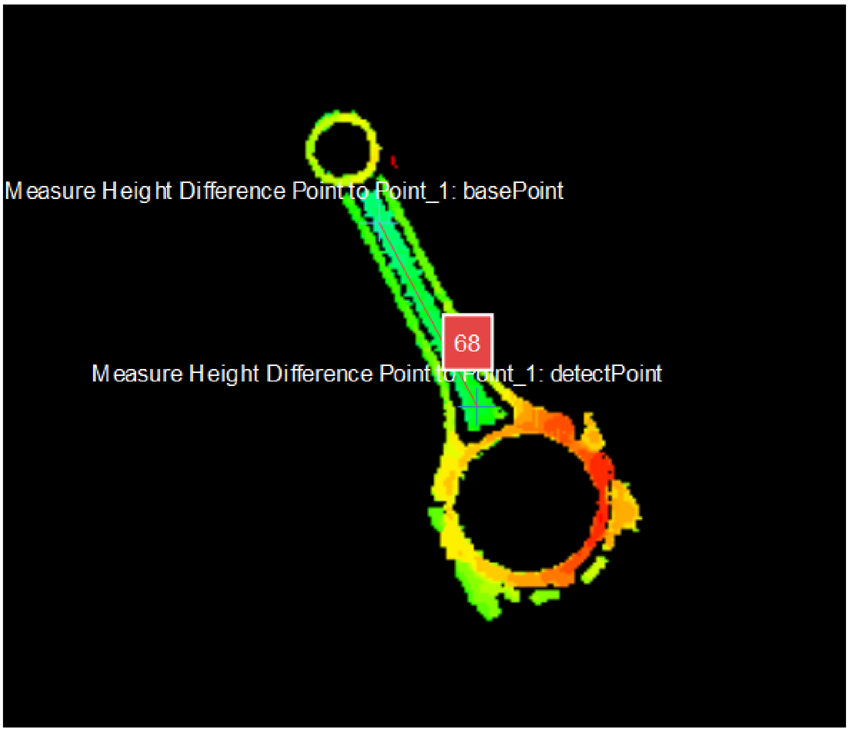
Connect Read Point Cloud V2, Orthographic Projection, and Measure Height Difference Point to Point to calculate the height difference from the **detect point** to the **base point**.

Connect Steps

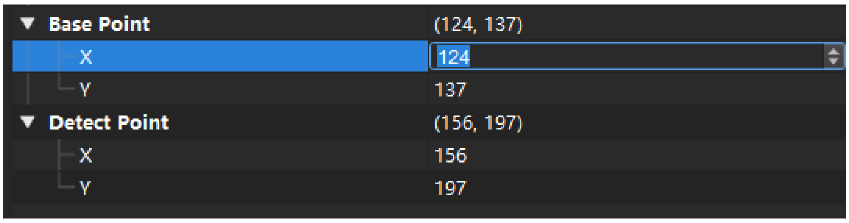
1. Click Orthographic Projection, go to the **Step Input Source Selection** panel, and then select Read Point Cloud V2_1_Point Cloud With Normals as Input 1 (Point Cloud With Normals).
2. Click Measure Height Difference Point to Point, go to the **Step Input Source Selection** panel, and then select Orthographic Projection_1_Depth Image as Input 1 (Visualization Background).

Configure parameters

- Configure in Read Point Cloud V2 Click the Step and configure the Point Cloud Source and other parameters in the **Step Parameters** panel. After configuration, click **[Run]** to run the project after the settings.
- Configure in Measure Height Difference Point to Point The baseline point and detect point are located in the upper left corner in the sketchpad area by default. Please drag them onto the point cloud.


Measure Height Difference Point to Point_1: basePoint
68
Measure Height Difference Point to Point_1: detectPoint

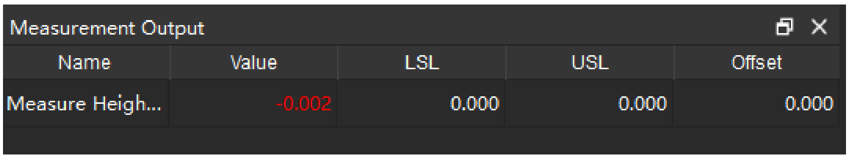
You can also adjust the position of the points in the **Step Parameters** panel, as shown below.



▼ **Base Point**	(124, 137)
X	124
Y	137
▼ **Detect Point**	(156, 197)
X	156
Y	197

Read the result

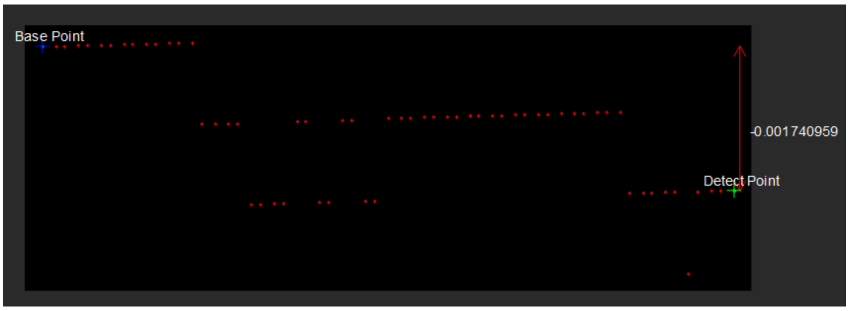
The distance between two points will be displayed in the sketchpad area. Click the Step Measure Height Difference Point to Point and check the measurement result in the **Measurement Output** panel.



Measurement Output

Name	Value	LSL	USL	Offset
Measure Heigh...	-0.002	0.000	0.000	0.000

The variation of heights along the line will be shown in the height view widget.



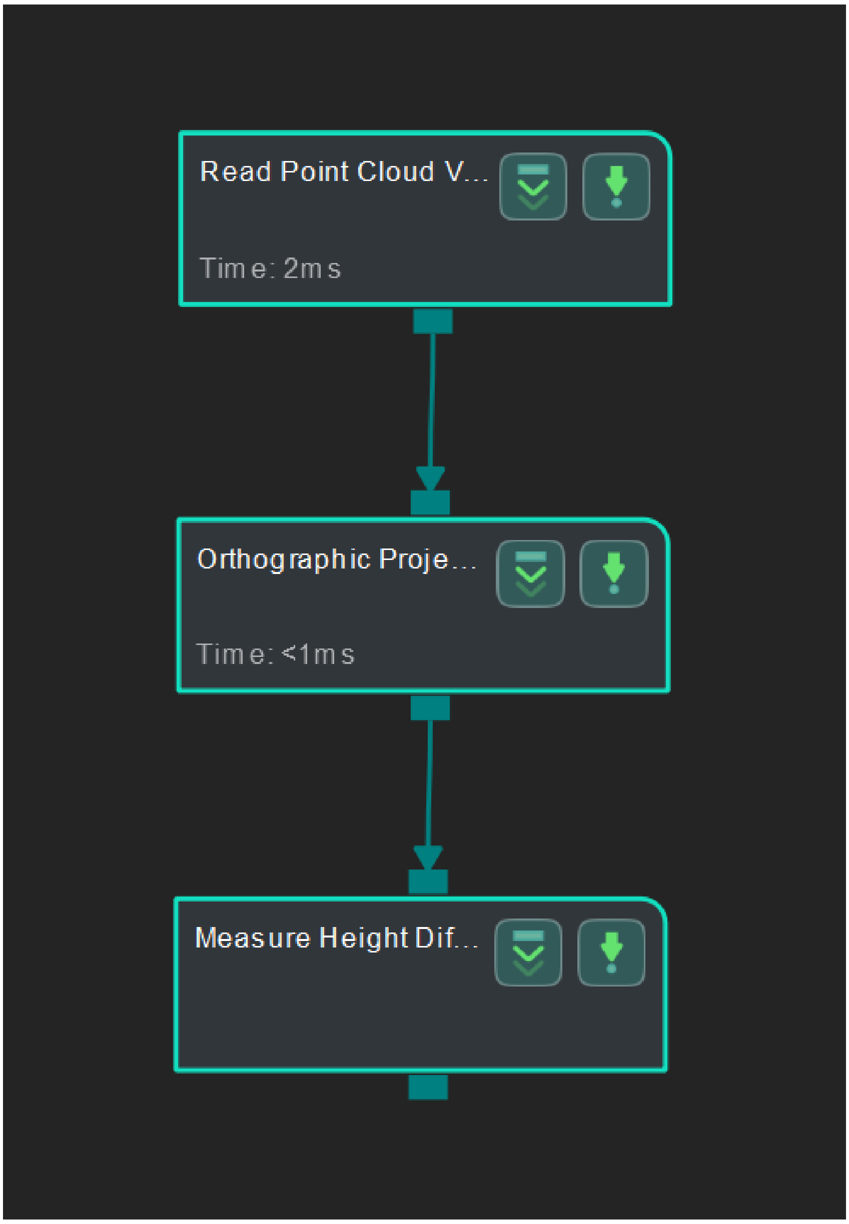
The measurement result will be displayed in **Measurement Output** and **Result View** panels as well. You can also set the LSL (lower specification limit) and USL (upper specification limit) to check whether the measurement conforms to the specification.

4.3.239. Measure Height Differences Points to Baseline

This Step is used to calculate the height difference from points to the baseline.

Before using this Step, please refer to Getting Started with Measurement Mode to learn about basics of Measurement Mode.

Sample Application



Connect Read Point Cloud V2, Orthographic Projection, and Measure Height Differences Points to Baseline to calculate the height difference from **points** to the **baseline**.

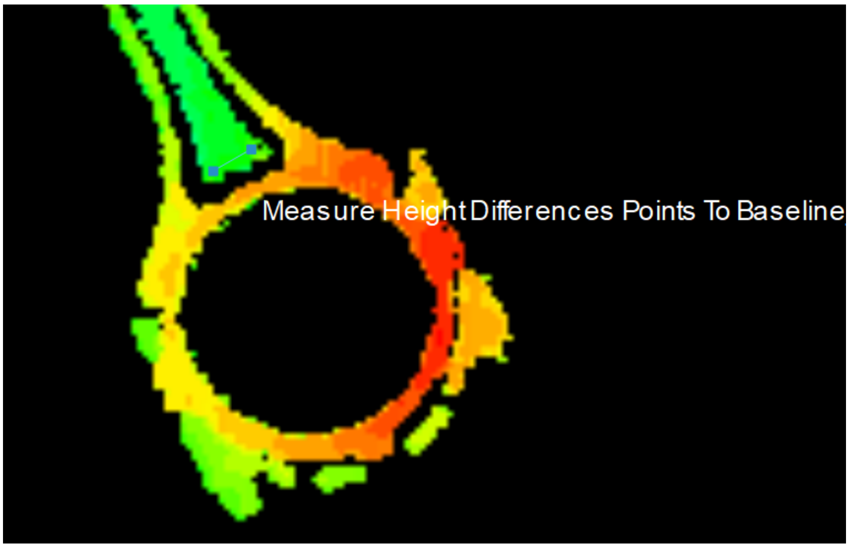
Connect Steps

1. Click Orthographic Projection, go to the **Step Input Source Selection** panel, and then select Read Point Cloud V2_1_Point Cloud With Normals as Input 1 (Point Cloud With Normals).
2. Click Measure Height Differences Points to Baseline, go to the **Step Input Source Selection** panel, and then select Orthographic Projection_1_Depth Image as Input 1 (Visualization Background).

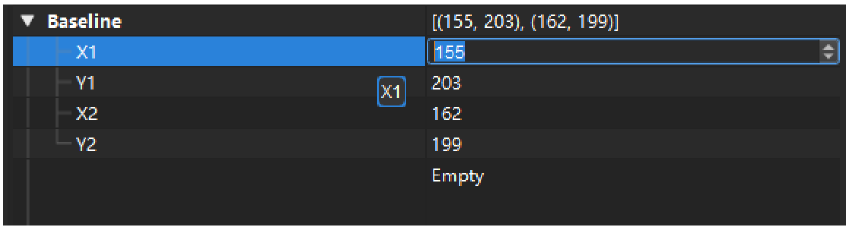
Configure parameters

- Configure in Read Point Cloud V2 Click the Step and configure the Point Cloud Source and other parameters in the **Step Parameters** panel. Click [**Run**] to run the project after the settings.
- Configure in Measure Height Differences Points to Baseline The baseline is located in the

upper left corner in the sketchpad area by default, drag it onto the point cloud and adjust the length.

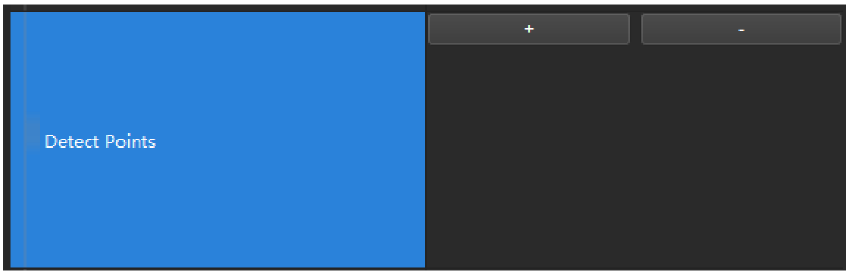


You can also adjust the position of the baseline in the **Step Parameters** panel, as shown below.

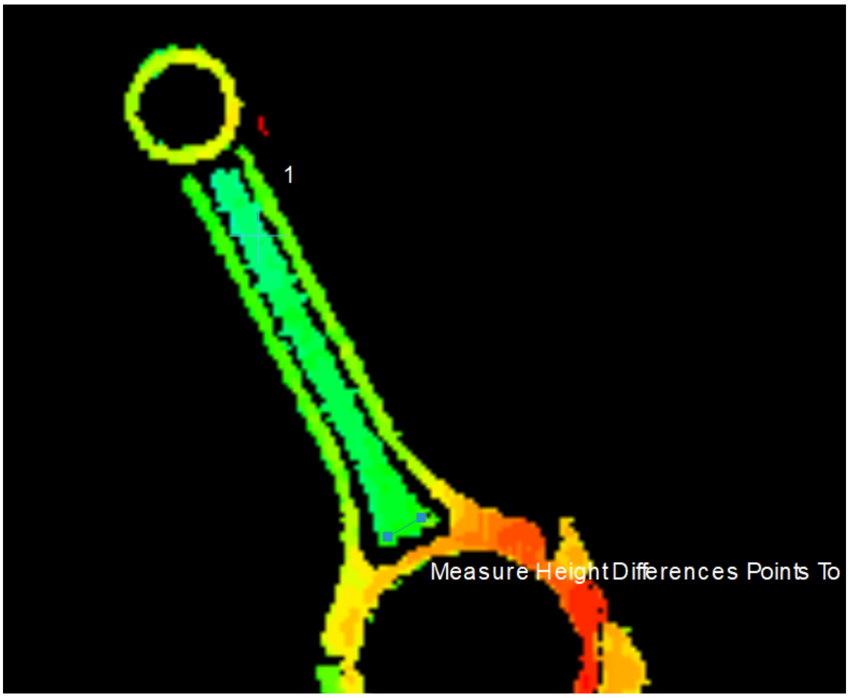


▼ Baseline	[(155, 203), (162, 199)]
X1	155
Y1	203
X2	162
Y2	199
	Empty

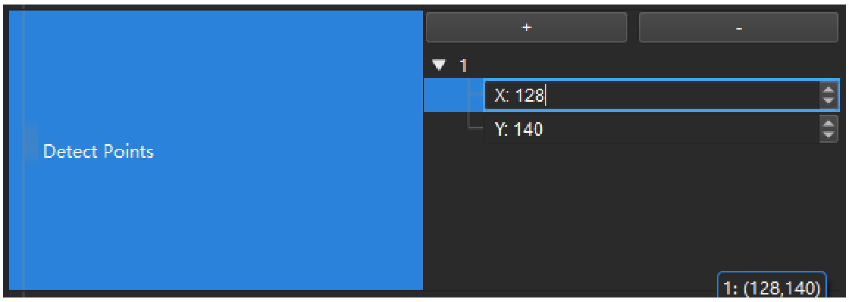
Click the **[+]** button to add one or more detect points:



Drag the newly added points from the upper left corner in the sketchpad area onto the point cloud:



You can also adjust the position of the detect points in the **Step Parameters** panel, as shown below.



Detect Points

+ -

▼ 1

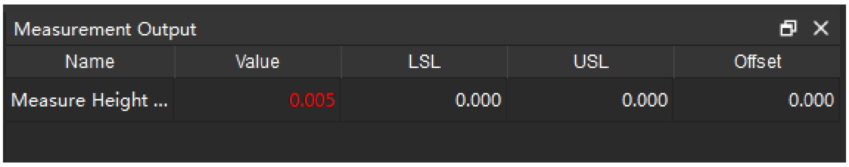
X: 128

Y: 140

1: (128,140)

Read the result

Click the Step Measure Height Differences Points to Baseline and check the measurement result in the **Measurement Output** panel.



Measurement Output

Name	Value	LSL	USL	Offset
Measure Height ...	0.005	0.000	0.000	0.000

If multiple detect points have been added, there will be multiple measurement results.

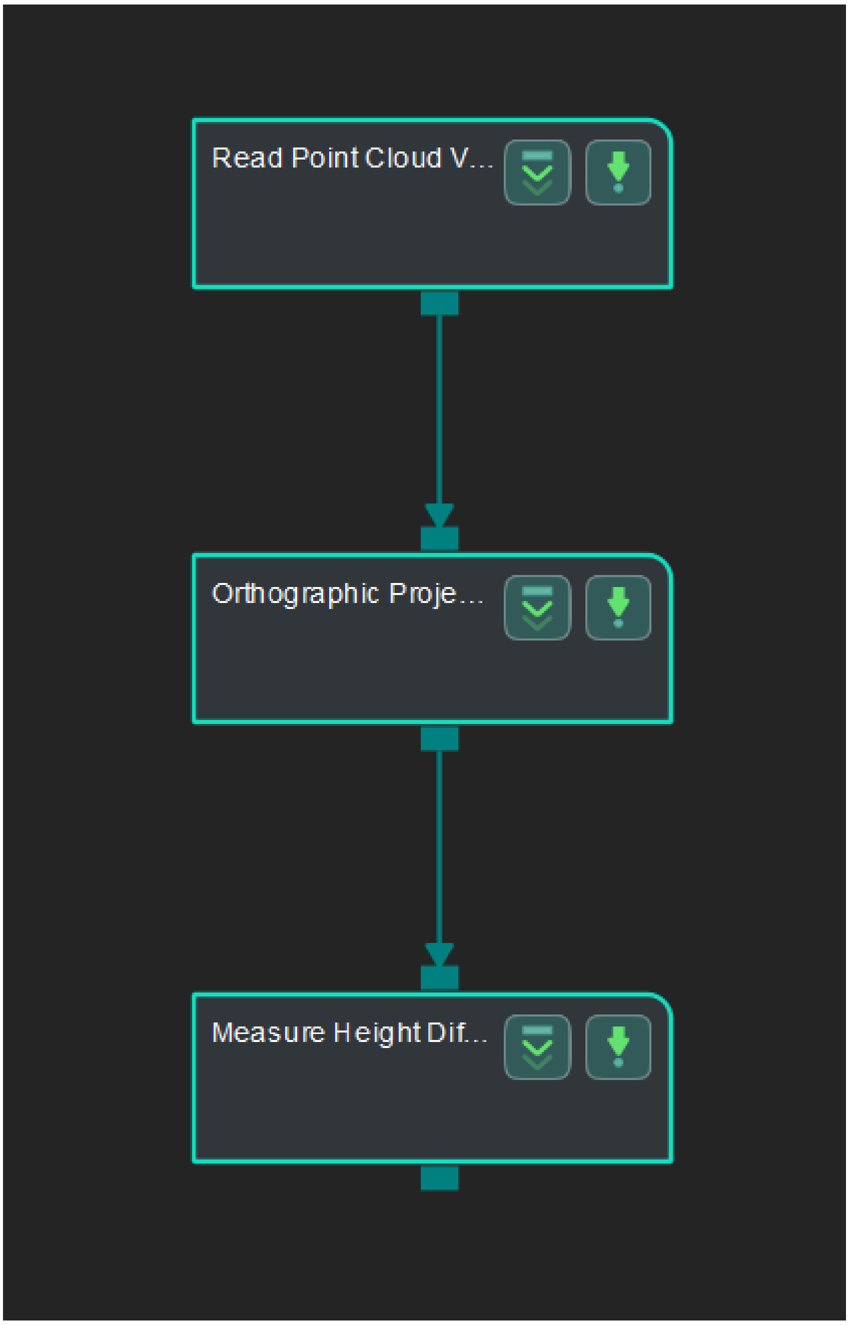
The measurement result will be displayed in **Measurement Output** and **Result View** panels as well. You can also set the LSL (lower specification limit) and USL (upper specification limit) to check whether the measurement conforms to the specification.

4.3.240. Measure Height Differences Points to Plane

This Step is used to calculate the height difference from **points** to the **plane** in a point cloud.

Before using this Step, please refer to Getting Started with Measurement Mode to learn about basics of Measurement Mode.

Application Examples



Connect Read Point Cloud V2, Orthographic Projection, and Measure Height Differences Points to Plane to calculate the height difference from **points** to the **plane** in the point cloud.

Connect Steps

1. Click Orthographic Projection, go to the **Step Input Source Selection** panel, and then select Read Point Cloud V2_1_Point Cloud With Normals as Input 1 (Point Cloud With Normals).
2. Click Measure Height Differences Points to Plane, go to the **Step Input Source Selection** panel, and then select Orthographic Projection_1_Depth Image as Input 1 (Visualization Background)

and Orthographic Projection_1_Orthographic Scale Ratio as Input 2 (Orthographic Scale Ratio).

Configure parameters

- Configure in Read Point Cloud V2

Click the Step and configure the Point Cloud Source and other parameters in the **Step Parameters** panel. Click **[Run]** to run the project after the settings.

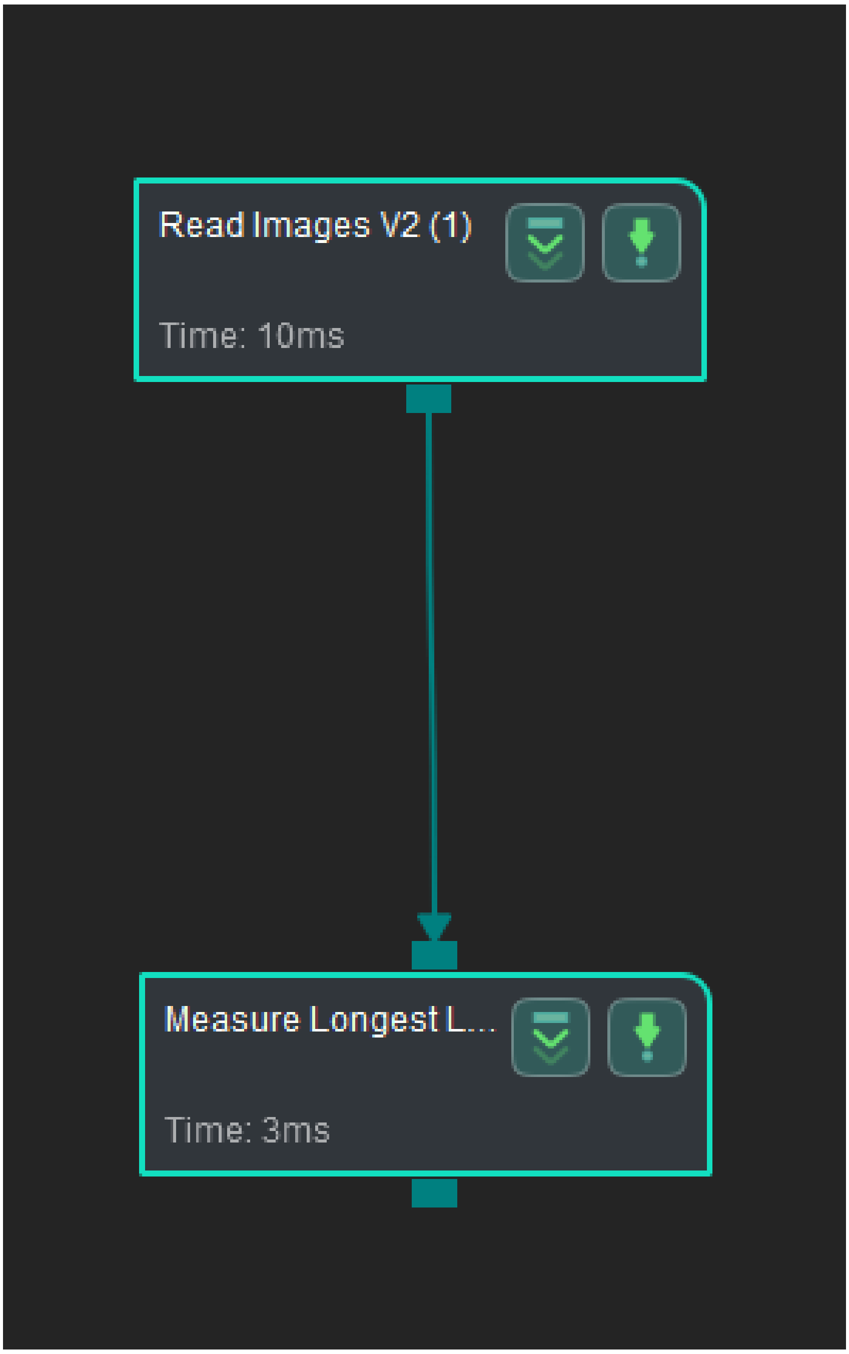
- Configure in Measure Height Differences Points to Plane
  - Keep the default height computation settings.
  - Add three Base Points to define a plane in the point cloud.
  - Add one Detect Point.
- The measured height difference from **points** to the **plane** will be displayed in **Measurement Output** and **Result View** panel. You can also set the LSL (lower specification limit) and USL (upper specification limit) to check whether the measurement conforms to the specification.

4.3.241. Measure Longest Line Segment

This Step is used to locate and measure the longest line segment in the ROI.

Before using this Step, please refer to Getting Started with Measurement Mode to learn about basics of Measurement Mode.

Sample Application



Connect Read Images V2 with Measure Longest Line Segment to locate the longest line segment in the ROI and measure its length (in pixels in this sample application).

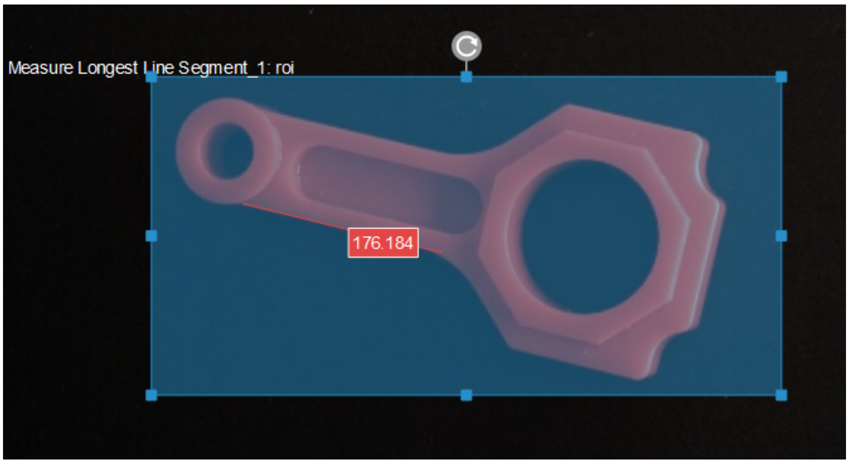
Connect Steps

Click Measure Longest Line Segment, go to the **Step Input Source Selection** panel, and then select Read Images V2_1_Color Image as Input 1 (Color Image).

Configure parameters and read the result

- Configure in Read Images V2: Click the Step and configure the Image Source, Image File/Folder Path, and other parameters in the **Step Parameters** panel. Click **[Run]** to run the project after the settings.

- Configure in Measure Longest Line Segment: Click the Step and select an ROI in the sketchpad area.



- The measurement result will be displayed in **Measurement Output** and **Result View** panel. You can also set the LSL (lower specification limit) and USL (upper specification limit) to check whether the measurement conforms to the specification.

4.3.242. Measure Plane Height Differences Along Direction Parallel To Axes

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step calculates the height difference at the point where the largest height drop occurs in the specified direction in the depth map.

Usage Scenario

This Step is a custom Step for special scenarios.

Input and Output

- Input:
  1. Depth map that defines the height of each coordinate point
- Output:
  1. Flatness (i.e. height difference at the largest height drop)
  2. Two parts of a line segment that extends from the largest height drop to the edges of the image, perpendicular to the fissure of the largest height drop

4.3.243. Measure Result

Function

This Step calculates and outputs measurement results according to the input depth map, images and poses.

Usage Scenario

This Step is used by developers for testing. Regular users do not need to use this Step in general application scenarios.

Input and Output

- **Input:**
  1. Depth map
  2. Masks
  3. Poses
  4. Poses
- **Output:**
  1. Color image

4.3.244. Merge Point Cloud Lists

> This Step is outdated. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step merges two point cloud lists into one.

Usage Scenario

This Step is a general point cloud list editing Step.

This Step is outdated. Please use newer Steps instead.

Input and Output

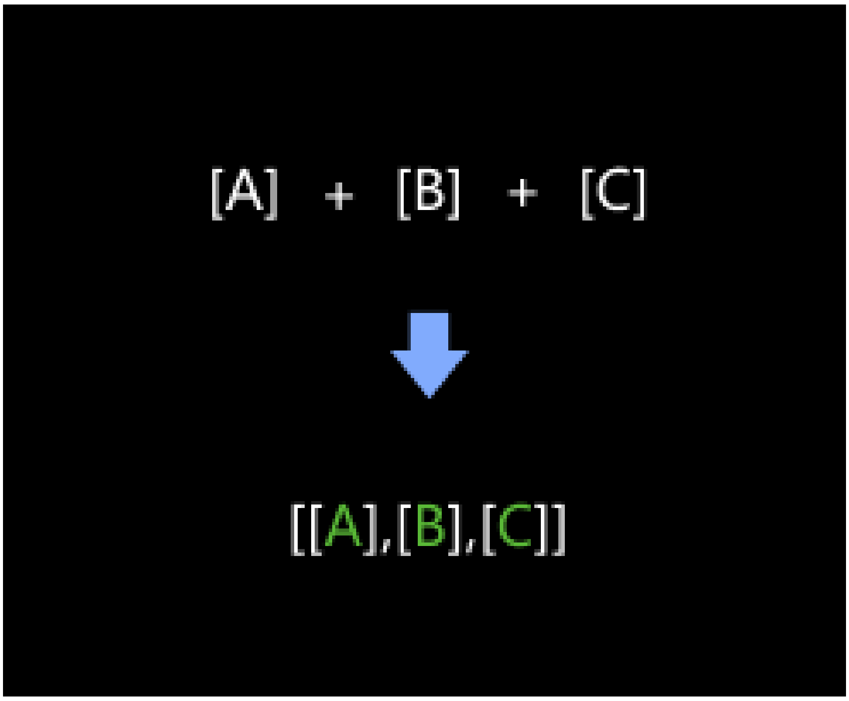
- **Input:** List of point clouds for merging.
- **Output:** Merged list of point clouds.

4.3.245. Merge Data

> Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

General data merging Step.

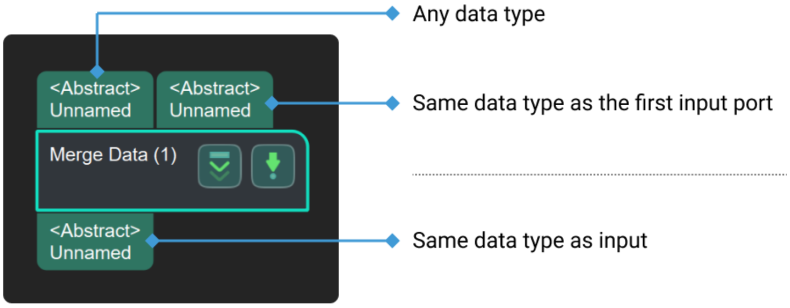


Usage Scenario

This Step is used to merge data of types, such as pose and number.

This Step is new, and it is recommended to replace earlier version Steps including Merge Pose Lists and Merge Point Cloud Lists with this Step.

Input and Output



4.3.246. Merge Depth Maps

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step merges two camera depth maps into one.

Usage Scenario

This Step is for developers and is used for adjustment in the passive stereo camera mode. Regular users do not need to use this Step in general application scenarios.

Input and Output

- Input:
  1. First depth map from camera for merging
  2. Second depth map from camera for merging
  3. ROI mask list for the left view
  4. ROI mask list for the right view
  5. Mapping matrix from camera disparity values to depth values (should be a 4 × 4 matrix)
- Output:
  1. Depth map generated from merging

4.3.247. Merge Label List

> The contents of this Step are no longer under maintenance. Please use the new Step Merge Data. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step combines the input two label lists into one.

Usage Scenario

This Step is a general label data combination Step.

Input and Output

- Input:
  1. List of labels input to this port will be combined into the output.
  2. List of labels input to this port will be combined into the output.
- Output:
  1. Label list obtained from combining inputs.

4.3.248. Merge LineSegment Lists

> The contents of this Step are no longer under maintenance. Please use the new Step Merge Data. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step joins two line segment lists.

Usage Scenario

This Step is a general line segment list joining Step. This Step has no fixed usage scenarios.

Input and Output

- Input:
  1. Line segment list input to this port will be joined.
  2. Line segment list input to this port will be joined (to the end of the first input list).
- Output:
  1. List of line segments formed by joining the two input lists.

4.3.249. Merge Point Clouds with Similar Heights

ℹ Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step merges point clouds with similar height values in the specified direction.

Usage Scenario

This Step merges the list of point clouds belonging to the same object. This Step can be used after the Step Point Cloud Clustering.

Input and Output

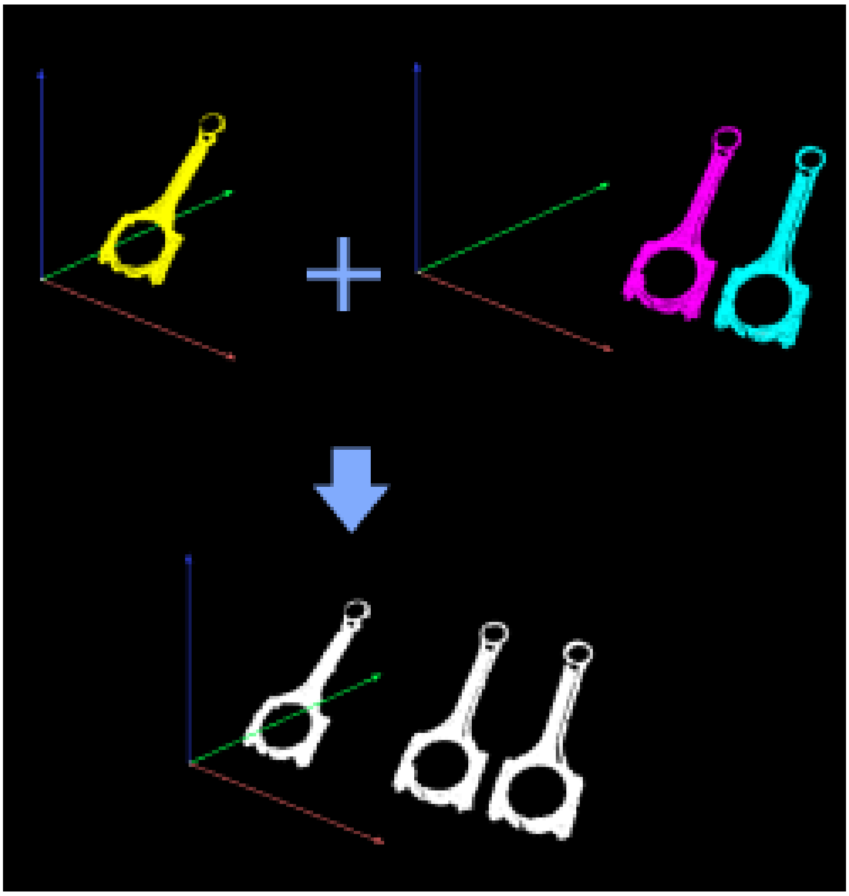
- **Input:** among point clouds input to this port, those with similar height values will be merged.
- **Output:** list of point clouds after merging.

4.3.250. Merge Point Clouds

ℹ Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

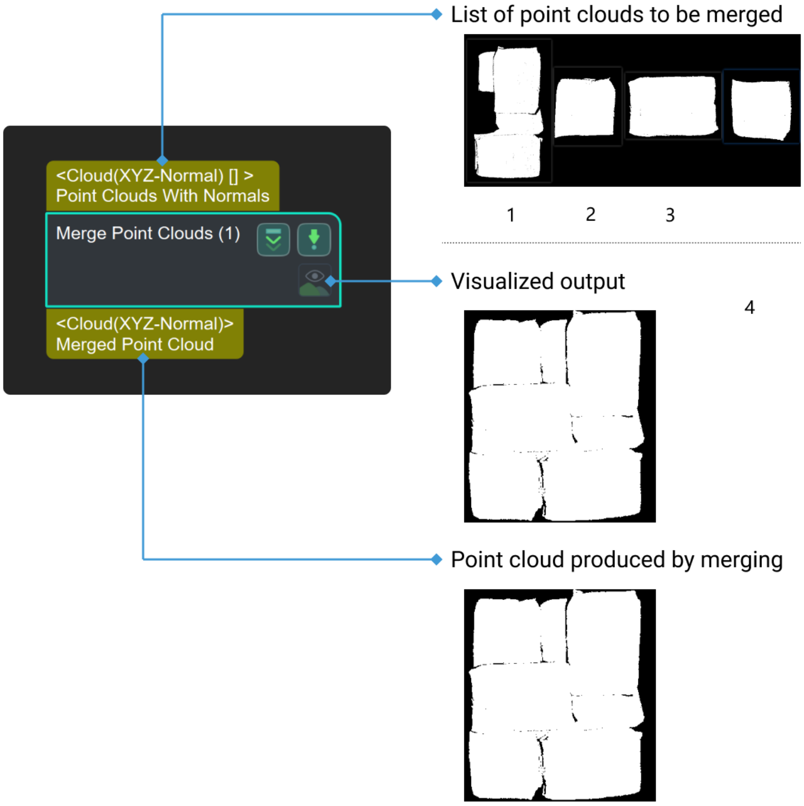
This Step merges multiple point clouds into one and outputs it.



Usage Scenario

This Step usually follows Steps such as Point Cloud Clustering and Apply Masks to Point Cloud. It can merge multiple point clouds in the point cloud list into a single point cloud to facilitate further processing.

Input and Output



4.3.251. Merge Pose Lists

This Step belongs to an earlier version; it can only merge lists of pose-related data types and cannot work with other data types. Please use Merge Data instead. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

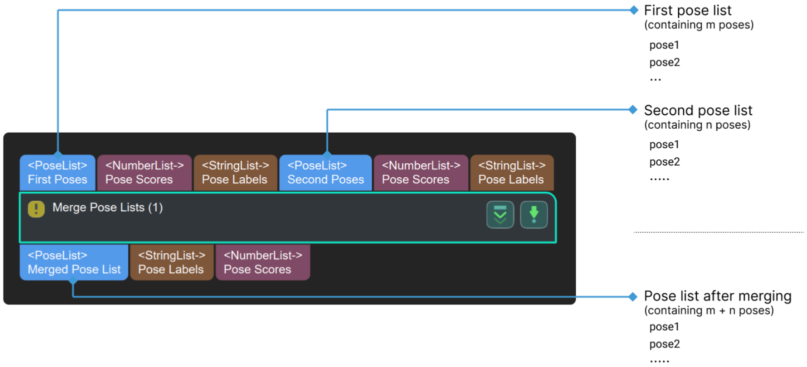
Function

Concatenate the two pose lists to merge them into one.

Usage Scenario

For various scenarios where pose data structures need to be adjusted.

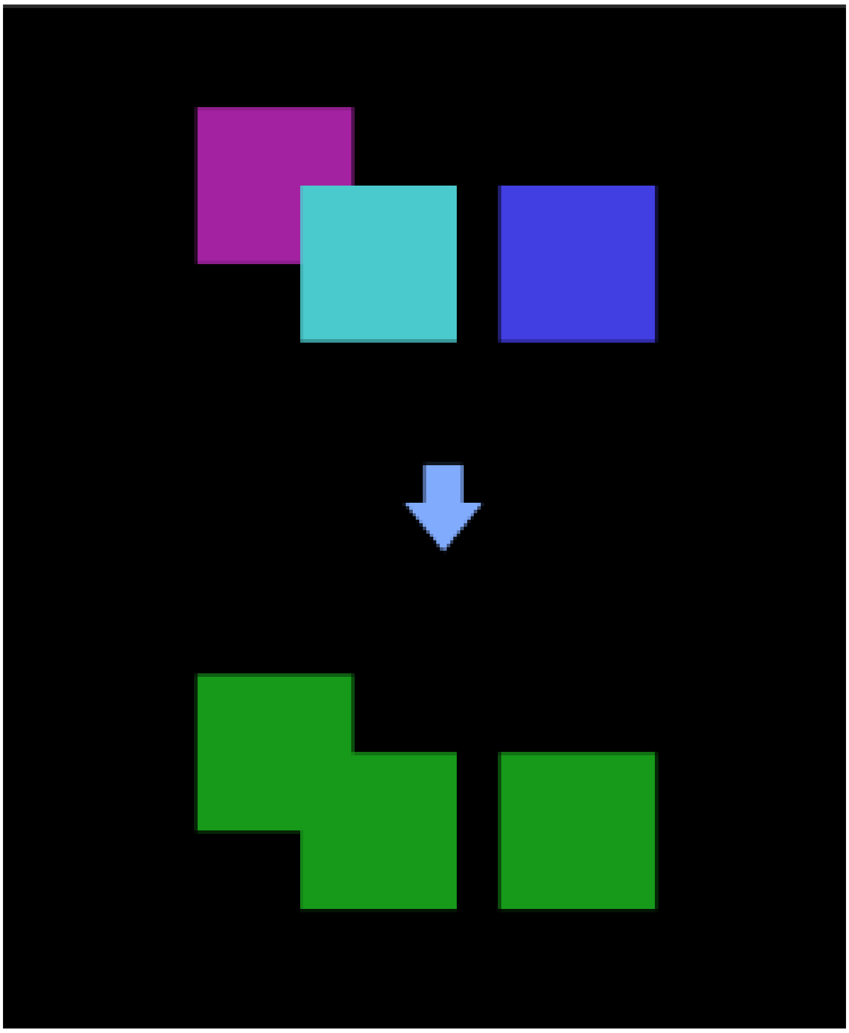
Input and Output



4.3.252. Merge Mask Images

Function

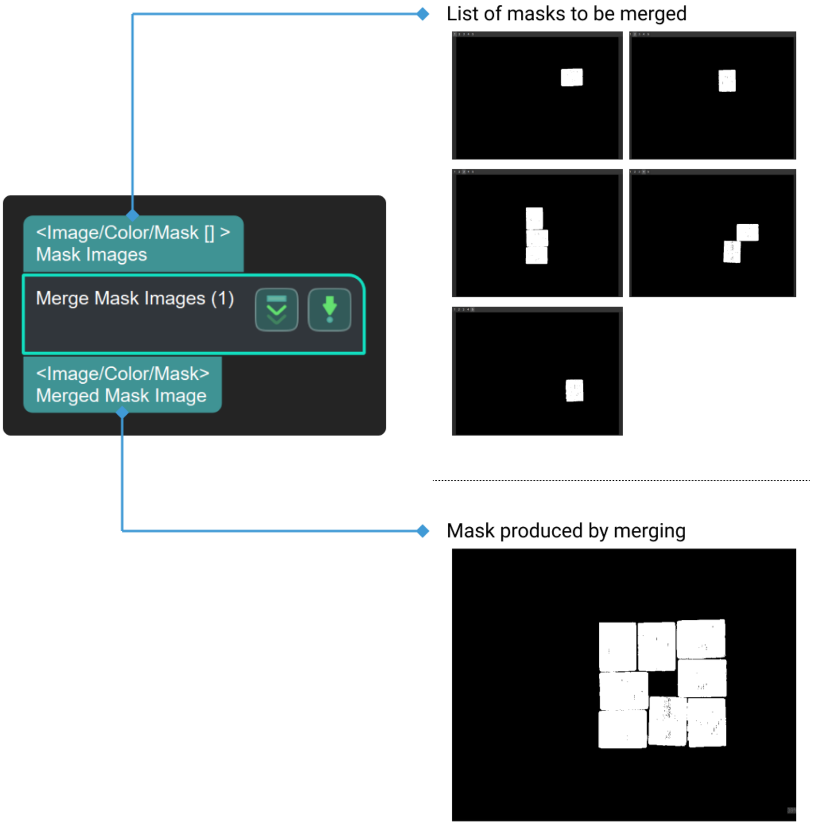
Merge multiple masks into on mask image.



Usage Scenario

This Step is widely used in depth map processing. It can merge multiple masks in the list into one mask, which subsequent processing of the masks as a whole.

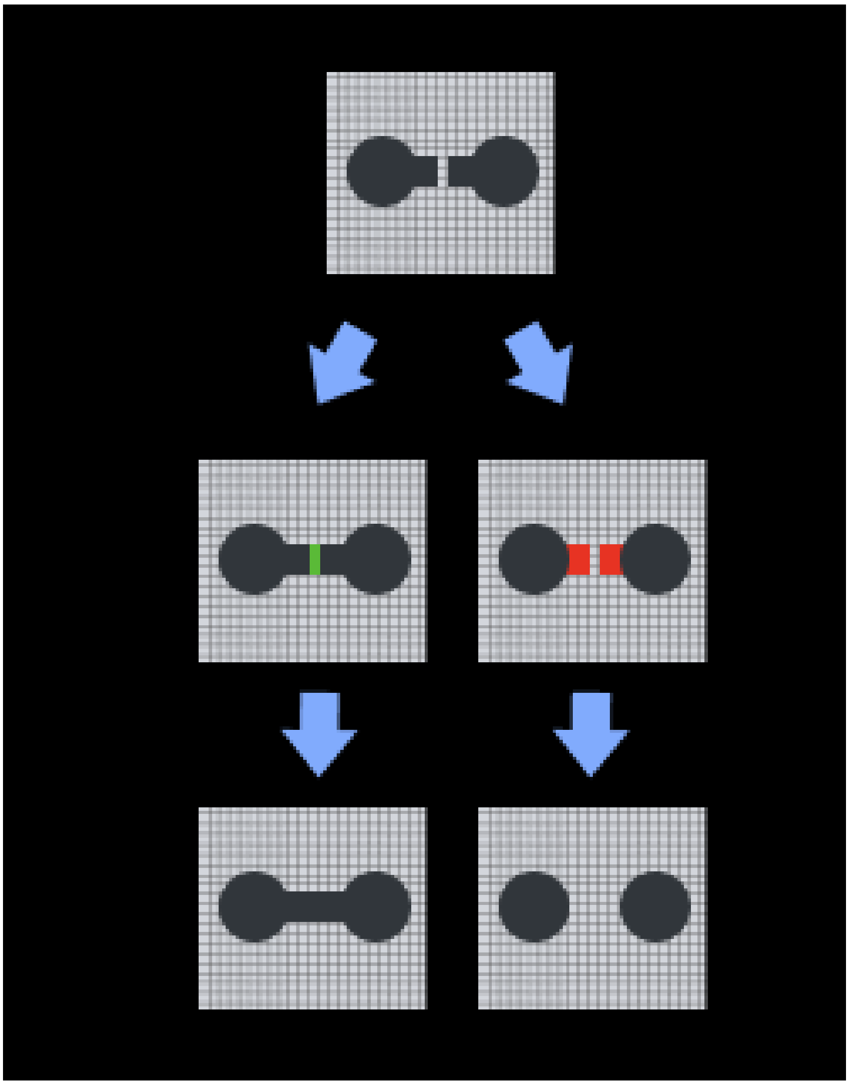
Input and Output



4.3.253. Morphological Transformations

Function

This Step processes an image using morphological operations.

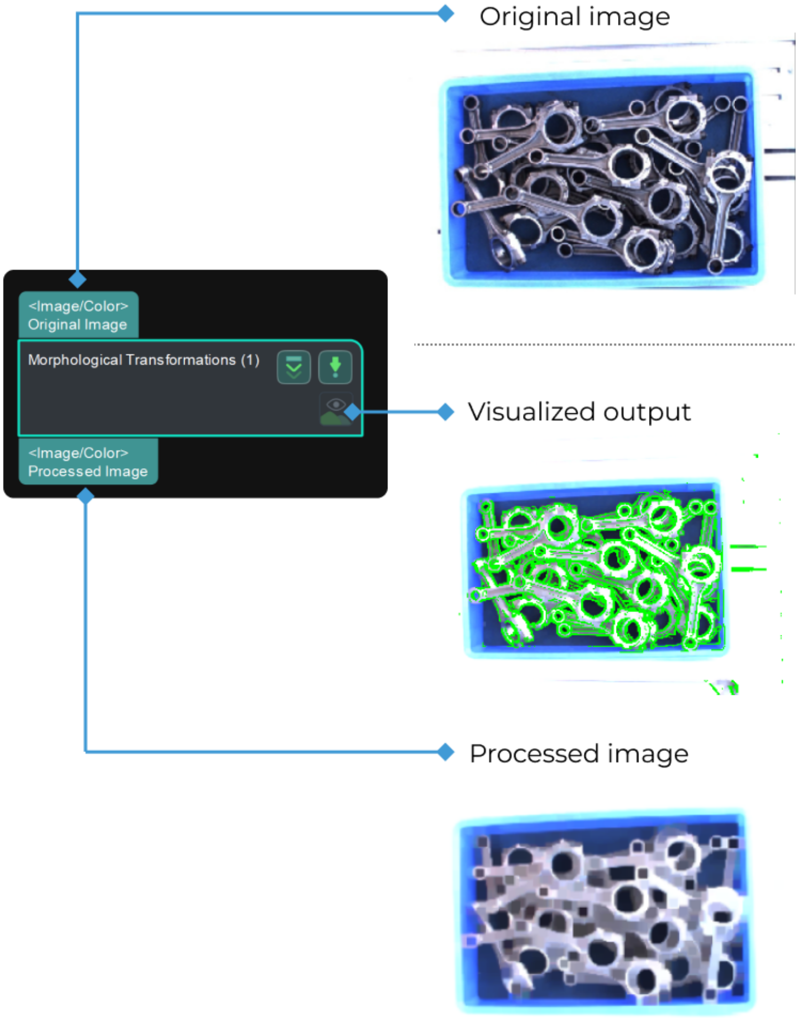


Usage Scenario

This Step is usually used for image pre-processing. When the quality of images is not ideal and further adjustment is needed, you need to apply operations such as dilation, erosion, and opening and closing to denoise, fill, etc. based on actual requirements.

Input and Output

When ClosingOperator is selected, the input and output are as shown in the figure below.



Parameter Description

Morphological Transformations

Description: This parameter is used to choose the type of morphological transformation.

Default setting: DilateOperator

List of options: ClosingOperator, DilateOperator, ErodeOperator, OpeningOperator, and Skeleton.

Instruction:

1. ClosingOperator: Closing is reverse of Opening, Dilation followed by Erosion. It is useful in closing small holes inside the foreground objects, or small black points on the object.

   Kernel Size

   Default value: 21 px

   Instruction: The size of the kernel should be set according to the measurement requirements. A larger size is accompanied by better erosion performance after dilation. Please refer to the configuration example for the corresponding result.

Kernel Shape

Default setting: Rectangle (MORPH_RECT)

Options: Rectangle (MORPH_RECT), Cross shape (MORPH_CROSS), and Ellipse (MORPH_ELLIPSE).

Instruction: It is recommended to keep the default setting. However, you may configure the setting according to specific requirements.

Performance on image detail restoration: MORPH_RECT < MORPH_CROSS < MORPH_ELLIPSE

- Time required for operation: MORPH_RECT < MORPH_CROSS < MORPH_ELLIPSE

2. DilateOperator: Dilation. It is mainly used to increase the object area and join broken parts of an object.

Kernel Size

Default value: 3 px

Instruction: The size of the kernel should be set according to the measurement requirements. The larger the kernel size, the more dilated the object area. Please refer to the configuration example for the corresponding result.

Kernel Shape

Default setting: Rectangle (MORPH_RECT)

Options: Rectangle (MORPH_RECT), Cross shape (MORPH_CROSS), and Ellipse (MORPH_ELLIPSE).

Instruction: It is recommended to keep the default setting. However, you may configure the setting according to specific requirements.

Performance on image detail restoration: MORPH_RECT < MORPH_CROSS < MORPH_ELLIPSE

- Time required for operation: MORPH_RECT < MORPH_CROSS < MORPH_ELLIPSE

3. ErodeOperator: Erosion. It is mainly used to remove small noises and detach two connected objects.

Kernel Size

Default value: 3 px

Instruction: The size of the kernel should be set according to the measurement requirements. The larger the kernel size, the greater the erosion of the object area. Please refer to the configuration example for the corresponding result.

Kernel Shape

Default setting: Rectangle (MORPH_RECT)

Options: Rectangle (MORPH_RECT), Cross shape (MORPH_CROSS), and Ellipse (MORPH_ELLIPSE).

Instruction: It is recommended to keep the default setting. However, you may configure the setting according to specific requirements.

Performance on image detail restoration: MORPH_RECT < MORPH_CROSS < MORPH_ELLIPSE

- Time required for operation: MORPH_RECT < MORPH_CROSS < MORPH_ELLIPSE

4. OpeningOperator: Opening is another name of erosion followed by dilation. It is useful in removing noise.

   Kernel Size

   Default value: 21 px

   Instruction: The size of the kernel should be set according to the measurement requirements. Please refer to the configuration example for the corresponding result.

   Kernel Shape

   Default setting: Rectangle (MORPH_RECT)

   Options: Rectangle (MORPH_RECT), Cross shape (MORPH_CROSS), and Ellipse (MORPH_ELLIPSE).

   Instruction: It is recommended to keep the default setting. However, you may configure the setting according to specific requirements.

   Performance on image detail restoration: MORPH_RECT < MORPH_CROSS < MORPH_ELLIPSE

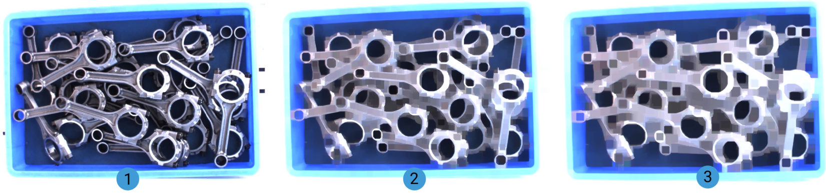
   - Time required for operation: MORPH_RECT < MORPH_CROSS < MORPH_ELLIPSE

5. Skeleton: Extract the contour of the object. Please refer to the configuration example for the corresponding result.

Configuration Examples

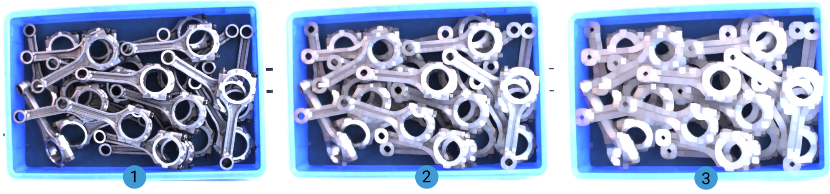
ClosingOperator

When **ClosingOperator** is selected, the size increase of the kernel is shown in the images below (the kernel size grows by degrees and from image 1 to 3):



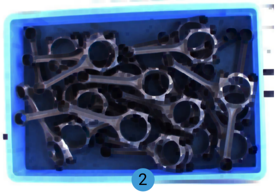
DilateOperator

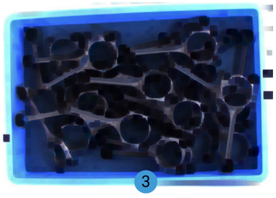
When **DilateOperator** is selected, the size increase of the kernel is shown in the images below (The kernel size grows by degrees and from image 1 to 3):



ErodeOperator

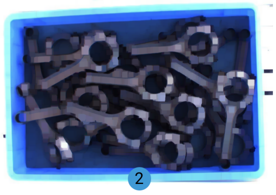
When **ErodeOperator** is selected, the size increase of the kernel is shown in the images below (The kernel size grows by degrees and from image 1 to 3):

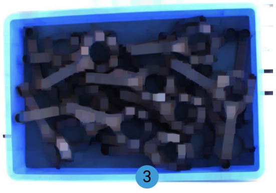




OpeningOperator

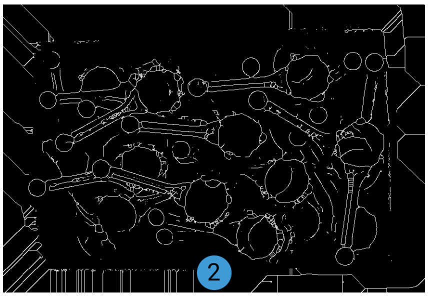
When **OpeningOperator** is selected, the size increase of the kernel is shown in the images below (The kernel size grows by degrees and from image 1 to 3):





Skeleton

The images before and after Skeleton is selected are as follows:



4.3.254. Move Point Cloud along Specified Direction

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step translates point clouds in the specified directions. The translation distance can be set by the input or the parameter.

Usage Scenario

This Step is a general point translation Step. No fixed usage scenarios.

This Step is outdated. Please use newer Steps instead.

Input and Output

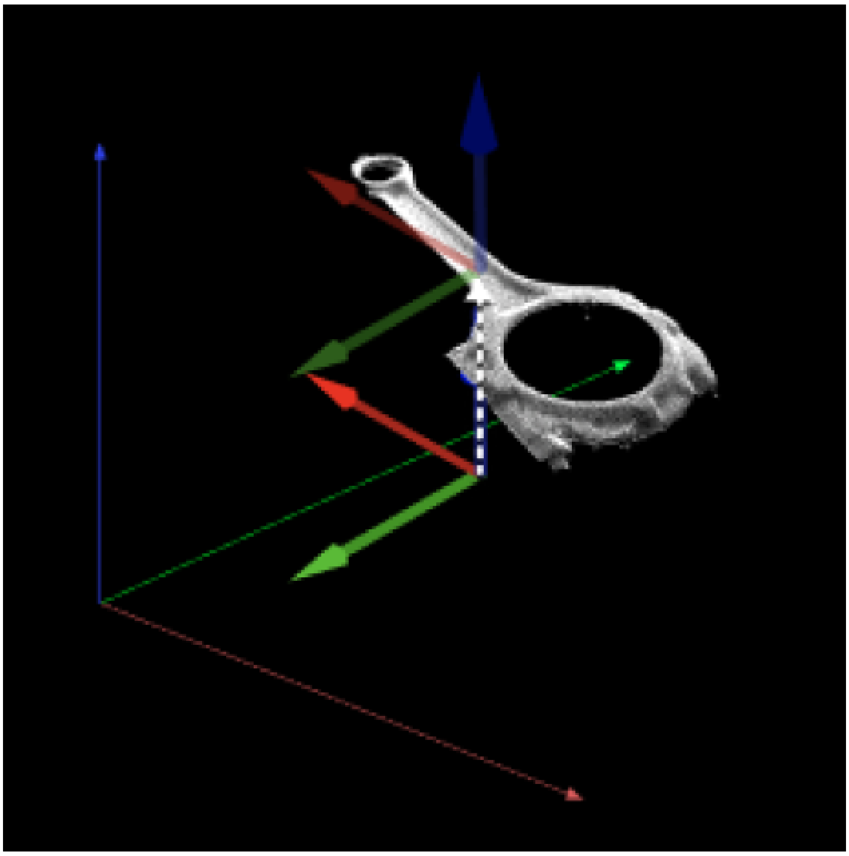
- Input:
    1. Point clouds input to this port will be translated.
    2. Object reference frame used as a reference for translation (optional, used when the translation reference frame is set as the object reference frame in the parameter).
    3. Translation distance (optional, if this port is not used, the distance set in the parameter will be used).
- Output:
    1. Translated point cloud

4.3.255. Move Poses to Point Cloud Surfaces along Z-Axis

> Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

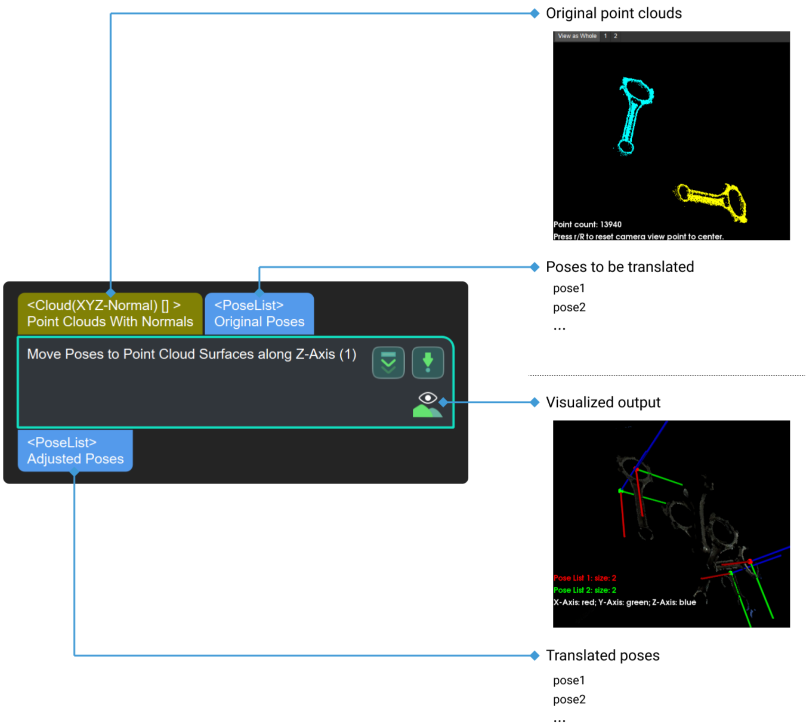
Move poses to point cloud surface along the Z-axis of a specified reference frame according to user-set rules.



Usage Scenario

This step is generally used to adjust the poses which are not on object surfaces. Causes of producing such poses include tilted target object and concave/convex surface. It is usually used after the Step Calc Poses and Dimensions from Planar Point Clouds.

Input and Output



4.3.256. Normal Estimation

> The Step is outdated. Please use the **NormalEstimation** method in the new Step Calc Normals of Point Cloud and Filter It to calculate normals. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

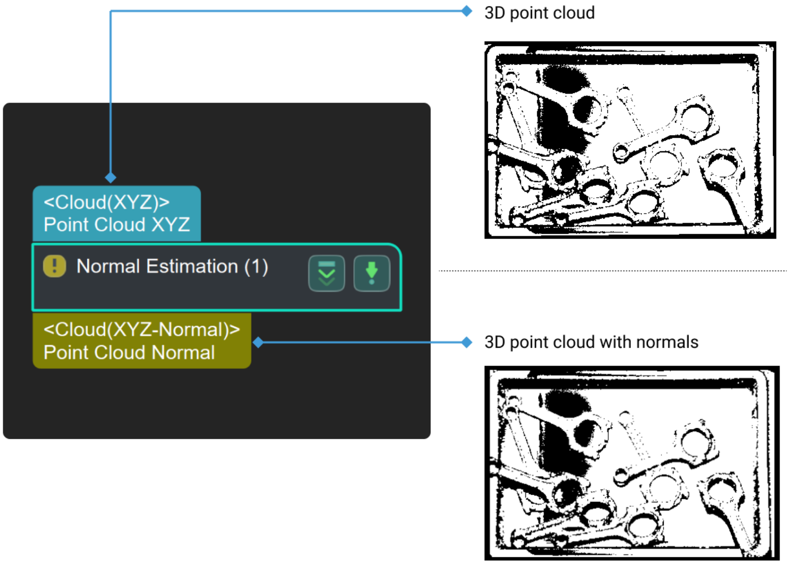
Function

Compute point cloud normal vectors.

Usage Scenario

This Step is outdated. Please use the new Step.

Input and Output



4.3.257. Notify

> Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step sends custom notifications to external services.

Usage Scenario

This Step is used to communicate with external services. Once the Step is connected to other Steps via control flow, the status of the Step can be monitored and messages can be sent to external services.

Input and Output

- **Input:** none.
- **Output:** none.

4.3.258. Number Scaling

> The contents of this Step will not be updated anymore. Please use the new Step Numerical Operation. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

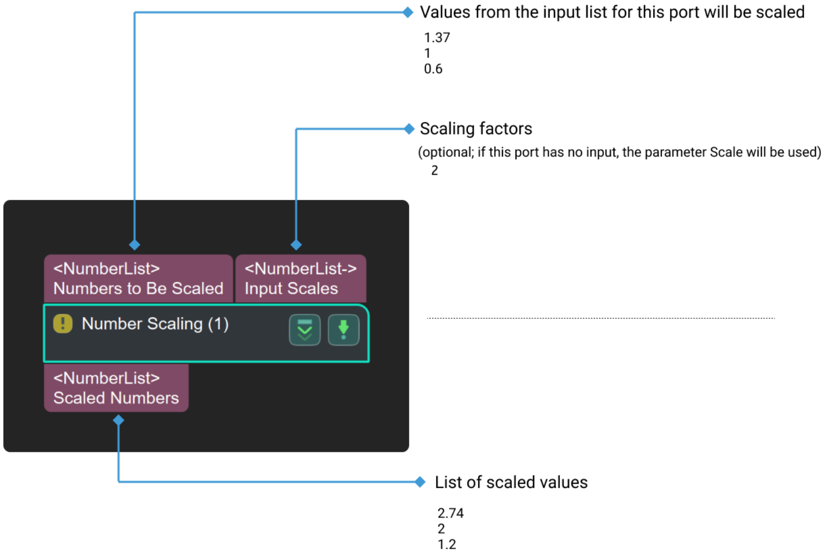
Function

Scale the list of values input by the first port. Both the second port and the parameter can be used to set the scaling factor.

Usage Scenario

This Step is outdated, please use the new Step.

Input and Output



4.3.259. Numeric Operation

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

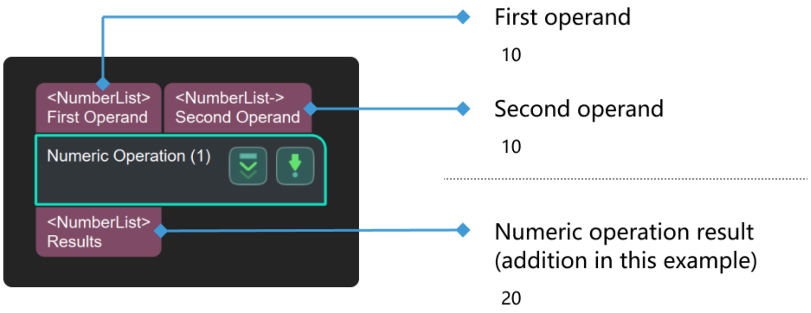
Perform numeric operations on the input values. Supported operations include addition, subtraction, multiplication, and division.

Usage Scenario

General numeric operation Step where the first operand is usually added, subtracted, multiplied, or divided by the second operand.

If you need to perform complex numeric operations, you can use the Evaluate Results by JavaScript Engine Step.

Input and Output



4.3.260. OCR

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step recognizes Chinese and English characters in the input image.

Usage Scenario

This Step is a general image processing Step. No fixed usage scenarios.

Input and Output

- Input:
  1. Chinese and English characters in the image input from this port will be recognized.
- Output:
  1. Recognized characters
  2. Confidence of recognized characters
  3. Positions of recognized characters in the image

4.3.261. Offset Poses in Cylinder

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step moves the input poses along the Z-axis of the reference frame to the surfaces of cylinder point clouds.

Usage Scenario

This Step is usually used to adjust pick points to point cloud surfaces.

This Step is outdated. Please use the newer Step “Move Poses to Point Cloud Surfaces along Z-Axis.”

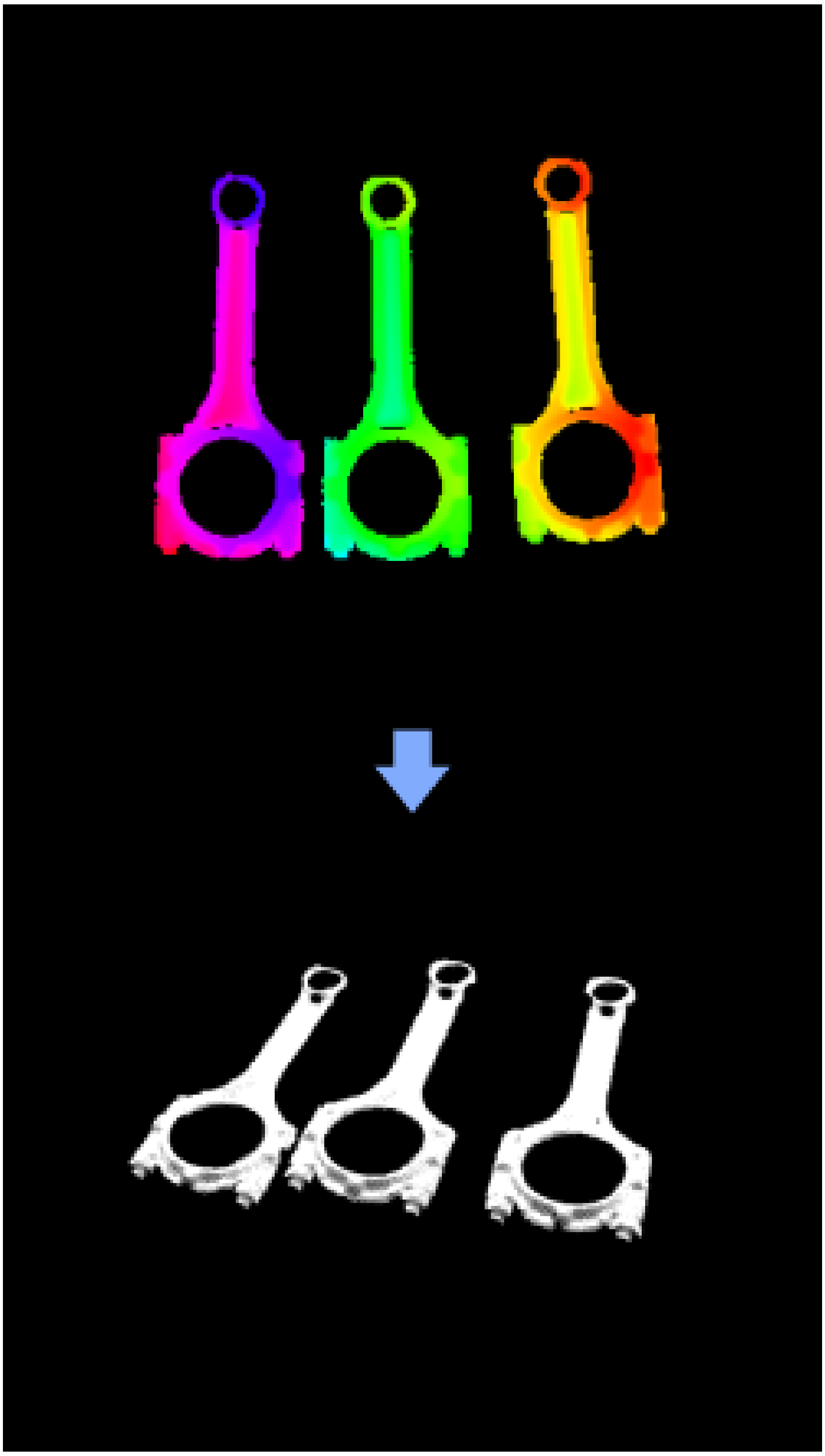
Input and Output

- Input:
  1. Cylinder point clouds input to this port will be used as the reference for adjusting the poses.
  2. Poses input to this port will be adjusted.
- Output:
  1. Poses adjusted to cylinder point cloud surfaces

4.3.262. Orthographic Projection Inverse

Function

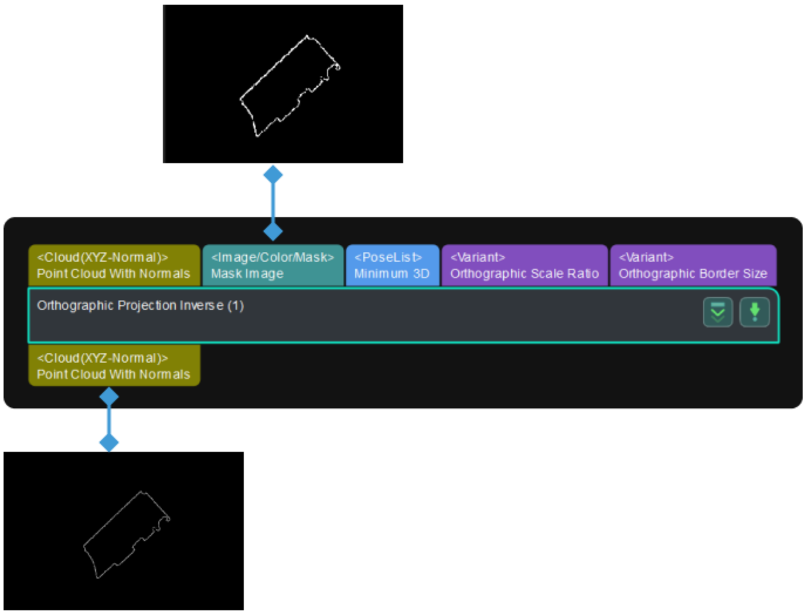
This Step converts the 2D points in the mask to 3D point clouds.



Usage Scenario

This Step is usually used for measurement scenarios. It usually follows the Orthographic Projection Step to convert the 2D image obtained from orthographic projection into the 3D point cloud.

Input and Output



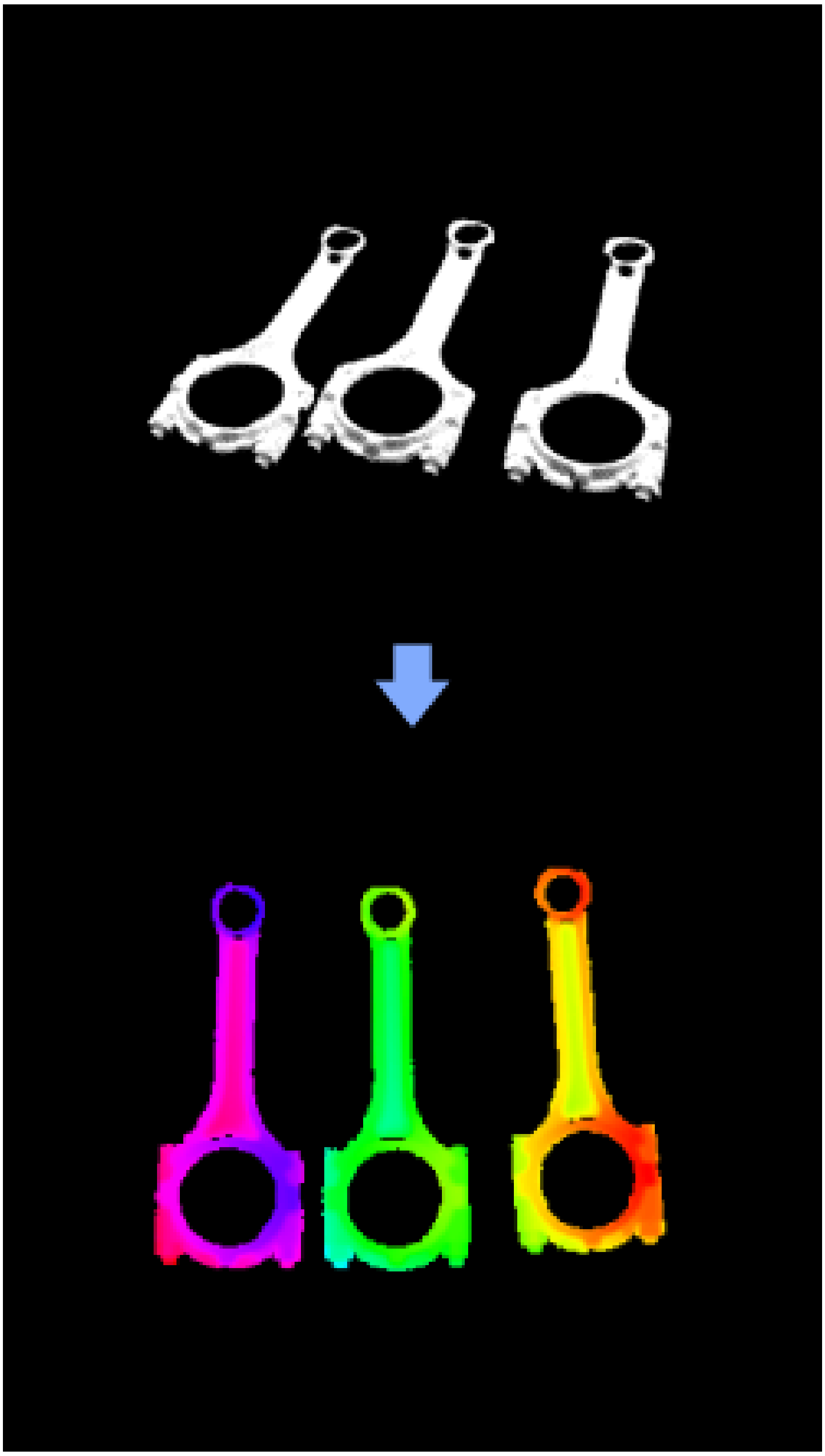
Parameter Description

None.

4.3.263. Orthographic Projection

Function

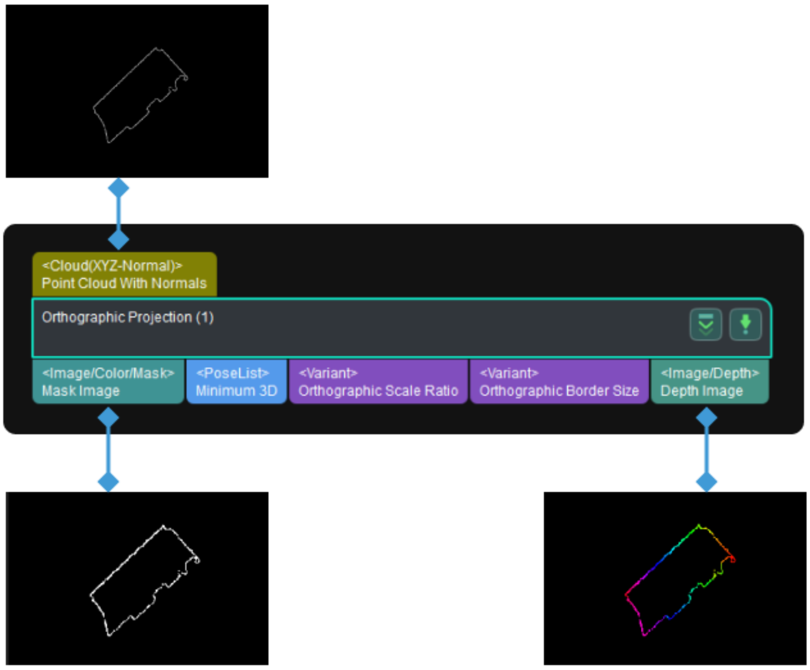
This Step orthographically projects a 3D point cloud into a 2D depth map, which can avoid the perspective distortion in ordinary projection.



Usage Scenario

This Step is usually used for measurement. It projects a 3D point cloud into a 2D depth map for further processing in succeeding measurement Steps.

Input and Output



Parameter Description

Please right-click and select **Show all parameters** to view all parameters of this Step.

Orthographic Projection Settings

Resolution

Parameter description: This parameter represents the number of pixels per unit length on the 2D image formed by the orthographic projection. The unit length is 1 meter.

Default value: 1000.0000

Tuning instruction: Please refer to the tuning example for the application result.

Mask Blank Border Width

Parameter description: A white frame will be added to the mask of the 2D image formed by the projection. This parameter specifies the width of the frame.

Default value: 100 px

Tuning instruction: Please refer to the tuning example for the application result.

Specify Min Point

Parameter description: This parameter specifies the vertex with the minimum X and Y coordinates on the bounding box of the point cloud. Once a minimum point and a maximum point are specified, the orthogonal projection will be performed on the range between the maximum and minimum points.

Default Value: Unselected.

Notes: If you select this parameter, you will need to set the **Min Point X Coordinate** and **Min Point Y Coordinate**.

Min Point X Coordinate

Default value: 0.000 mm

Tuning recommendation: If you run the project with the **Specify Min Point** unselected, reference values will be output in the log panel. You can adjust the parameter according to the reference values until the effect of the orthographic projection outputs a desired result. Generally, it is expected that there is no inference point cloud and only the area where the object is located is kept after the orthographic projection.

[orthographic projection log x y] | *orthographic-projection-log_x_y.png*

Min Point Y Coordinate

Default value: 0.000 mm

Tuning recommendation: See **Min Point X Coordinate**.

Specify Max Point

Parameter description: This parameter specifies the vertex with the maximum X and Y coordinates on the bounding box of the point cloud. Once a minimum point and a maximum point are specified, the orthogonal projection will be performed on the range between the maximum and minimum points.

Default Value: Unselected.

Notes: If you select this parameter, you will need to set the **Max Point X Coordinate** and **Max Point Y Coordinate**.

Max Point X Coordinate

Default value: 0.000 mm

Tuning recommendation: See **Min Point X Coordinate**.

Max Point Y Coordinate

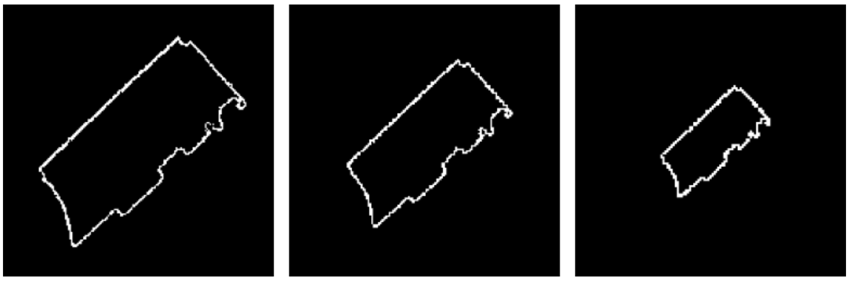
Default value: 0.000 mm

Tuning recommendation: See **Min Point X Coordinate**.

Tuning Example

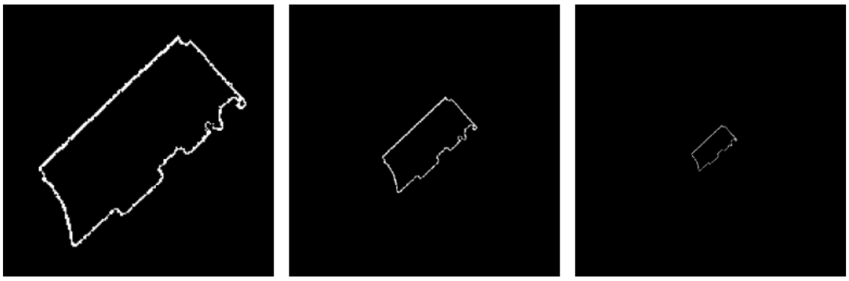
Resolution

When the resolution is set to 1000, 700, and 400, respectively, the output masks after the orthographic projection are shown below.



Mask Blank Border Width

When the mask blank border width is set to 100, 400, and 700, respectively, the output masks after the orthographic projection are shown below.



4.3.264. Pack

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

Pack multiple data elements into a data list.

Usage Scenario

A general data composition Step to combine data and increase data structure dimensionality.

Input and Output

- Input:
  1. Data input to this port will become an element in the output list.
  2. Data input to this port will become an element in the output list.
- Output:
  1. List formed by taking the data input as elements

4.3.265. Path Planning

Function

This Step uses the input vision points to plan the robot path and outputs the successfully planned robot path.

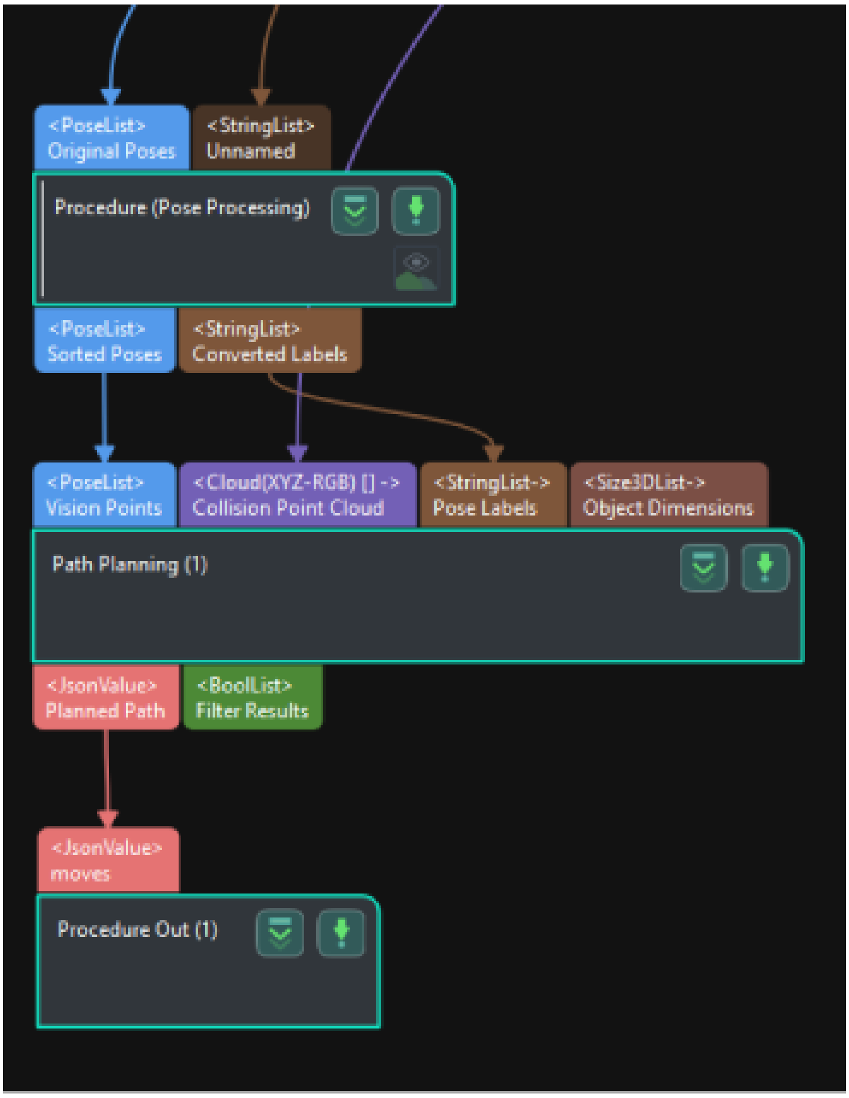
- Please ensure that you have successfully established the communication between the robot and the software before using this Step.
- When using the Standard Interface communication for communication, you will need to start the Mech-Vision project with command 101. The specific command parameters you should send are: *101, Mech-Vision project ID, expected number of vision points, robot pose type, robot pose.*

Usage Scenario

This Step is usually used in projects where the Standard Interface or Adapter is used for communication, and only the robot motion path near the vision point needs to be planned. After you build a scene and input the vision points, this Step will output a collision-free robot motion path after point cloud collision detection and path planning.

The common predecessors and successor of this Step are as follows:

- Predecessors: Steps related to pose adjustment and pose processing.
- Successor: the Procedure Out Step (used to output the successfully planned robot path).


<PoseList> Original Poses
<StringList> Unnamed
Procedure (Pose Processing)
<PoseList> Sorted Poses
<StringList> Converted Labels
<PoseList> Vision Points
<Cloud(XYZ-RGB) [] -> Collision Point Cloud
<StringList-> Pose Labels
<Size3DList-> Object Dimensions
Path Planning (1)
<JsonValue> Planned Path
<BoolList> Filter Results
<JsonValue> moves
Procedure Out (1)

Input and Output

Input

- Vision Points: Poses input to this port will be used for robot path planning.
- Collision Point Cloud: Point clouds input to this port will be used for collision detection with the tool.
- Pose Labels: Object labels corresponding to vision points will be input to this port.
- Object Dimensions: Object dimensions corresponding to vision points will be input to this port.

The following three input ports will appear after the Update Scene Object Settings is enabled.

- Scene Object Names: The names of the scene objects will be input through this port.
- Scene Object Dimensions: The dimensions of the scene objects will be input through this port.
- Scene Object Poses: The poses of the scene objects will be input through this port.

Output

- Planned Path: Successfully planned robot path will be output via this port.
- Filter Results: The list of the filtered vision points. True indicates that the corresponding vision point meet the requirement while False indicates that the vision point does not meet the requirement.

Parameter Description

Path Planning Settings

Workflow Configuration

Description: Open the path planning tool and select the workflow created with the path planning tool in the drop-down list.

Update Scene Object Settings

This feature is usually used in scenarios where the bin pose should be updated dynamically. To ensure its proper functioning, you should enable **Update bin pose with vision** under Global configuration in the **path planning tool**.

Once this option is switched on, Scene Object Names, Scene Object Dimensions, and Scene Object Poses input ports will be added to the Step.

Collision Point Cloud Settings

Input Cloud Type

Description: the point cloud type input to the path planning tool.

Value list: CloudXYZRGB (colored point cloud), CloudXYZ (point cloud), and CloudXYZNORMAL (point cloud with normals).

Input Point Cloud in Camera Frame

Description: If the input point cloud of this Step is in the camera reference frame, please select this option. Then the point cloud in the camera reference frame will be converted to the robot reference frame.

4.3.266. Periodic Trigger

Function

This Step outputs trigger signals periodically.

Usage Scenario

This Step is generally used for cyclic testing to trigger the project to run according to the set period.

Input and Output

- **Input:** none.
- **Output:** none.

4.3.267. Perspective Transformation

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step performs perspective transformation on the input image based on the two lists of points correspondingly before and after the perspective transformation.

Usage Scenario

This Step is generally used to correct oblique images to horizontal images.

Input and Output

- **Input:**
  1. Image input to this port will go through perspective transformation.
  2. Points before perspective transformation.
  3. Points corresponding to those in the second input port after perspective transformation.
- **Output:**
  1. Image after perspective transformation

4.3.268. Plan Object Poses at Placement

Function

This Step utilizes the input scene images and information of the bin and objects to be placed to plan the object poses at placements.

Usage Scenario

This Step is usually used for planning the object poses at placements in piece picking and palletizing/depalletizing scenarios.

Input and Output

- **Input:**
  1. Scene color image in the ROI. Usually from Step Scale Image in 2D ROI.
  2. Scene depth map in the ROI. Usually from Step Scale Image in 2D ROI.

3. ROI parameters. Usually from Step Scale Image in 2D ROI.
4. The 3D dimensions of the objects to be placed. In palletizing/depalletizing, please input the box dimensions. In piece picking, please input the dimensions of the objects' 3D bounding boxes.
5. Pose of the bin's top plane.

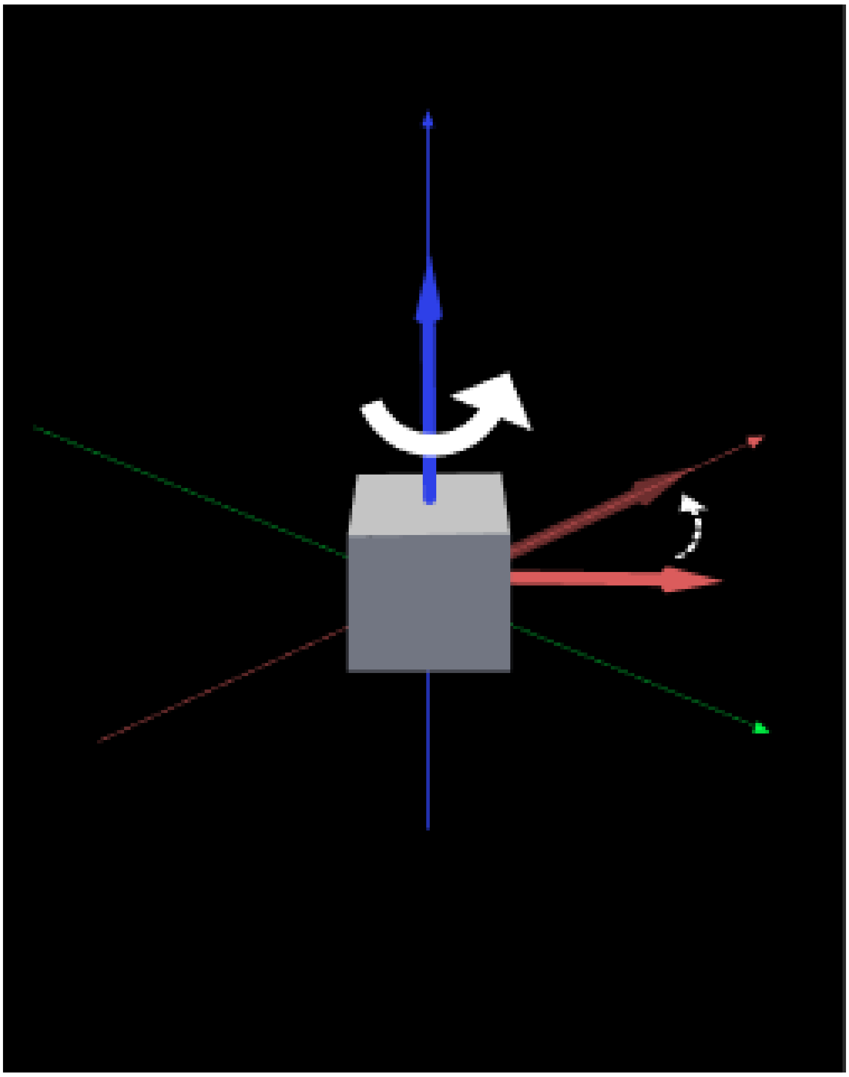
- **Output:**

1. Planned object poses after placement. Follow the planned placement order.
2. Visualization image.

4.3.269. Point Axes of Poses to Given Direction

Function

Adjust the orientations of axes of poses by a set reference direction.

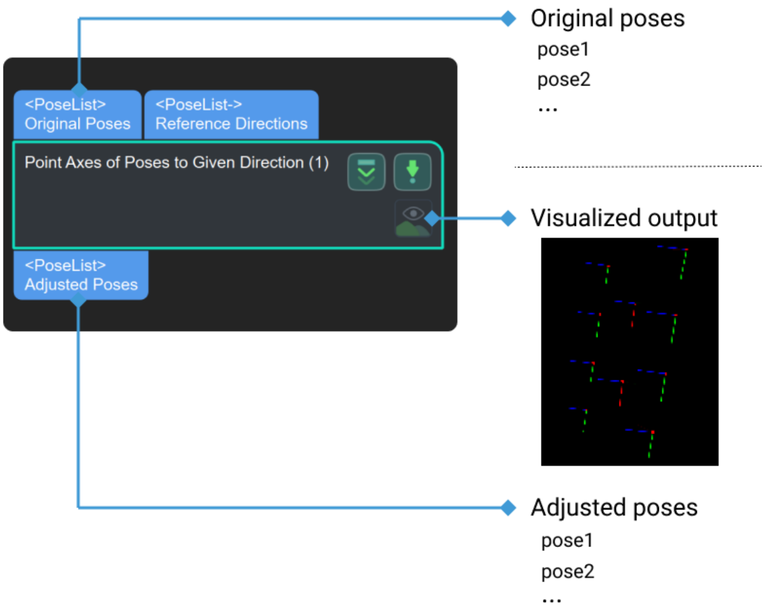


Usage Scenario

For various scenarios where pose orientations need to be adjusted.

This Step is outdated. If you need to rotate poses by object symmetricity (that is, to set the parameter Rectification Method of this Step to ROTATION), please use the new Step Rotate Poses' Axes to Specified Directions under Symmetry Constraints together with Step Easy Create Vector3Ds.

Input and Output



Parameter Description

Visualization Settings

Pose Type to Visualize

Default setting: After.

List of options: After, Before, and All.

Instruction: This parameter decides the type of pose to visualize, including pose after processing, pose before processing, and poses before and after processing.

Method Selection

Rectification Method

- HARD_RECTIFY: The default value. If there exist a second input (reference pose), and useFirstRefPose is selected, the reference direction is the direction of the corresponding axis type of the first reference pose. Otherwise the reference direction will be the corresponding axis type of all reference poses. If there isn't a second input (reference pose), "reference Dir" will determine the reference direction.
- ROTATION: Set the symmetrical angle according to the symmetry of the object, so that the rotation axis to be adjusted will be roughly in the same direction with the specified reference axis.

Axis Settings

Axis to Be Adjusted

Instruction: the rotation axis that needs adjustment.

List of option: X, Y, and Z (default setting).

Reference Direction

X-axis

Default value: 0

Instruction: the direction vector of the reference axis X.

Y-axis

Default value: 0

Instruction: the direction vector of reference axis Y.

Z-axis

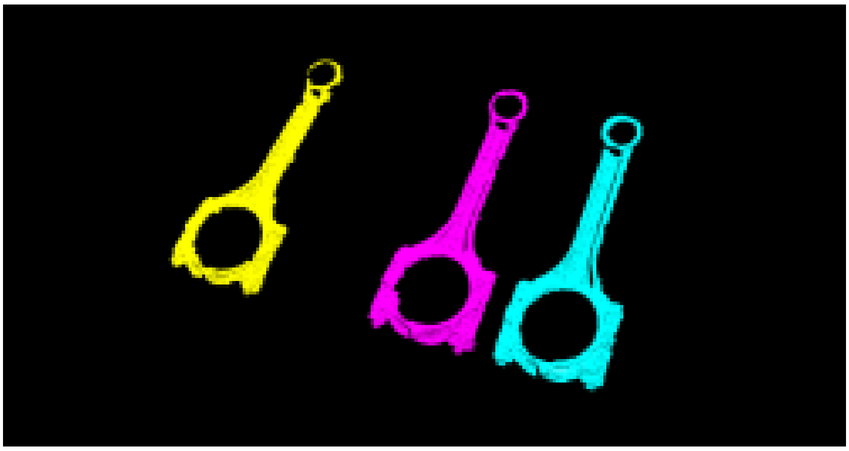
Default value: 1

Instruction: the direction vector of the reference axis Z.

4.3.270. Point Cloud Clustering

Function

Perform point cloud clustering according to specified rules; often used for segmentation of target objects.



Usage Scenario

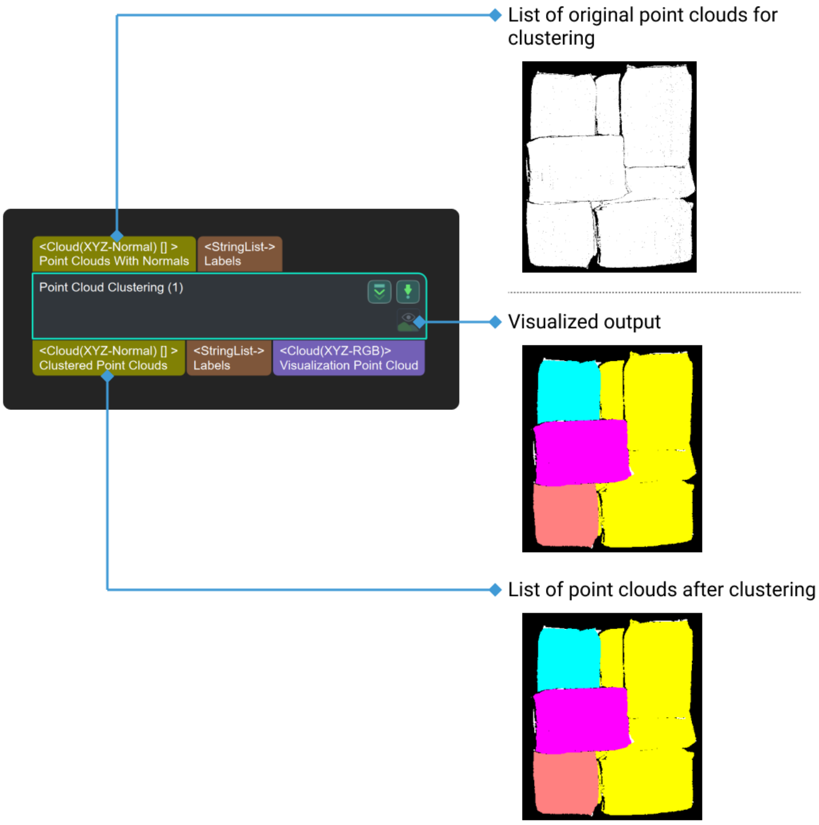
This Step is usually used for point cloud pre-processing to eliminate interference from unwanted point clouds.

It supports two point cloud clustering methods: clustering using Euclidean distance (EuclideanCluster) and clustering using region growing segmentation (RegionGrowingSeg).

When the point cloud has obvious spatial separation, it is recommended to use Euclidean clustering (EuclideanCluster); when the point cloud is spatially continuous, but the curvatures of the connection parts change greatly, it is recommended to use region growing segmentation

(RegionGrowingSeg).

Input and Output



List of original point clouds for clustering

Visualized output

List of point clouds after clustering

Parameter Description

Cluster Algorithm

Default value: EuclideanCluster

List of values: EuclideanCluster, RegionGrowingSeg.

Instruction: This parameter is used to select the clustering method, and there are two clustering methods in total. In practical engineering, it is recommended to use the EuclideanCluster algorithm, and this algorithm parameter is introduced first.

- EuclideanCluster: Determine whether a class belongs to based on proximity.
- RegionGrowingSeg: Determine whether a class belongs based on normality and curvature.

EuclideanCluster

Max Distance of Neighbouring Points in Output Clusters

Default value: 3.000 mm

Description: This parameter is used to set the cluster tolerance (in mm).

Instruction: When this parameter is adjusted up, points that are farther apart will be grouped into the same class; when this value is adjusted down, points that are closer together will be grouped into different categories.

Example of adjustment: as shown in the figure below. The left picture shows the result when the parameter is 3.000 mm, and the right picture shows the result when the parameter is 5.000 mm. It can be seen that after adjusting the parameters, the color of the point clouds indicated by the orange arrows are all green, which means that they are classified in the same category.

[point cloud clustering adjacent point distance] | *point-cloud-clustering-adjacent-point_distance.png*

Cores

Default value: 4°

Instruction: This parameter is used to adjust the normal angle difference threshold of adjacent points during region growth. When this value is adjusted to a larger value, the tolerance for normal angle difference of adjacent image points becomes larger and points with larger angle difference will still be grouped together.

RegionGrowingSeg

Number of Neighbours

Default value: 30

Instruction: This parameter is used to adjust the number of pixels searched during the region growing process. When the value is adjusted to a higher value, the number of searched pixel points becomes larger and the region grows faster, which results in a smaller number of classes obtained by clustering.

Smoothness Threshold

Default value: 4°

Instruction: This parameter is used to adjust the threshold for the difference in normal angle between neighboring points during region growth. When this value is adjusted upwards, the tolerance for the normal phase angle difference of adjacent image points becomes larger and points with larger angle differences will still be grouped together.

Curvature Threshold

Default value: 1.0000

Instruction: This parameter is used to adjust the upper threshold value for the curvature of points in the region growth process.

Min Points Num in Cluster

Default value: 800

Instructions: This parameter is for filtering the results after clustering. Only the classes with number of points greater than the minimum number of points can be output. When this parameter is large, the number of final output classes will decrease; when the parameter is small, the number of final output classes will increase.

Max Points Num in Cluster

Default value: 3000000

Instructions: This parameter is for filtering the results after clustering. Only the classes with number of points lower than the minimum number of points can be output. When this parameter is small, the number of final output classes will decrease; when the parameter is large, the number of final output classes will increase.

Example: Suppose that the point cloud is clustered into five classes, and their numbers of points are 10,000, 20,000, 30,000, 40,000, and 50,000, respectively. If the maximum number of points of the class is set to 45,000, and the minimum number of points is set to 15,000, then the classes with 10,000 and 50,000 points will be filtered out, and only the classes with 20,000, 30,000, and 40,000 points will be output.

Use GPU

Default setting: unselected.

Description: This parameter decides whether to accelerate calculation with GPU.

4.3.271. Point Cloud Filter

This Step is outdated. Please use the new Step Validate Point Clouds instead. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

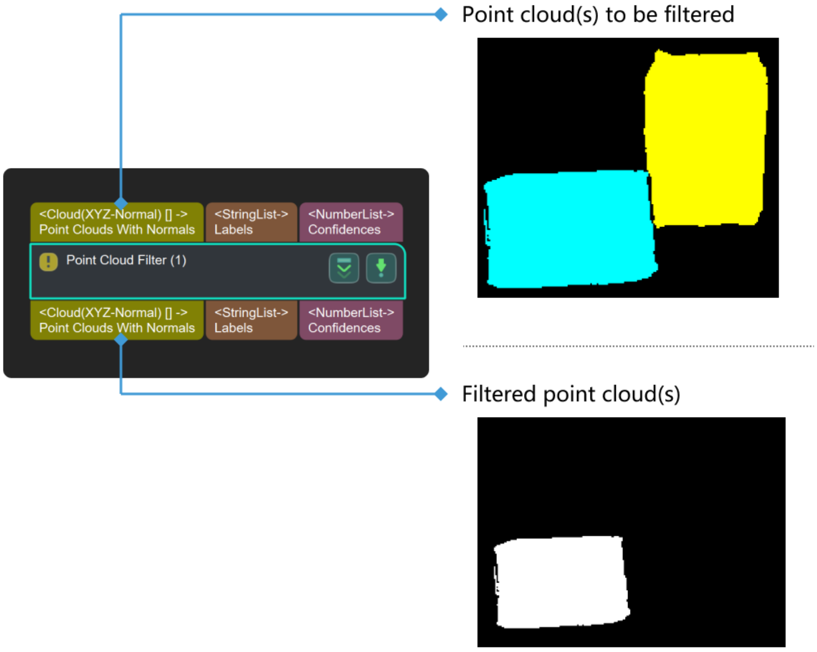
Function

Filter the input point clouds according to the set rules.

Usage Scenario

Usually used to filter point clouds based on the number of points (setting the Filter parameter to CloudCapacityFilter).

Input and Output


Point cloud(s) to be filtered

Filtered point cloud(s)

<Cloud(XYZ-Normal) [] -> Point Clouds With Normals
<StringList-> Labels
<NumberList-> Confidences
Point Cloud Filter (1)
<Cloud(XYZ-Normal) [] -> Point Clouds With Normals
<StringList-> Labels
<NumberList-> Confidences

Parameter Description

Filter Type

Instruction: This parameter is used to select the filter type. There are four types of filter, i.e., CircleCloudsFilter, CloudCapacityFilter, RecCloudsFilter, and SolidCloudsFilter.

Default Value: CircleCloudsFilter

Suggested value: To set according to the actual scenarios.

List of Values: CircleCloudsFilter, CloudCapacityFilter, RecCloudsFilter, and SolidCloudsFilter.

CircleCloudsFilter

CircleCloudsFilter could transfer the 3D point clouds into 2D mask image to get the contour of the mask and calculate the circularity and perimeter in order to filter out the point clouds that are not covered in the range.

Contour Retrieval Mode

Description: This parameter is used to select the contour retrieval mode and two kinds of mode are included:RETR_EXTERNAL and RETR_HULL.

Default value: RECT_EXTERNAL

Suggested Value: To set according to the actual scenarios.

Min Area Threshold

Description: This parameter is the lower limit of the area of the circle clouds (in pixel).

Default value: 100

Suggested value: Set according to the actual scenarios.

Max Area Threshold

Description: This parameter is the upper limit of the area of the circle clouds (in pixel).

Default value: 1000

Suggested value: Set according to the actual scenarios.

Min Circularity

Description: This parameter is the lower limit of the circularity of the circle.

Default value: 0.5

Suggested value: Set according to the actual scenarios.

Max Circularity

Description: This parameter is the upper limit of the circularity of the circle.

Default value: 1

Suggested value: Set according to the actual scenarios.

Min Convexity

Description: This parameter is the lower limit of the convexity of the circle.

Default value: 0.8

Suggested value: Set according to the actual scenarios.

Max Convexity

Description: This parameter is the upper limit of the convexity of the circle.

Default value: 1

Suggested value: Set according to the actual scenarios.

Min Inertia

Description: This parameter is the lower limit of the ratio between the length of short axis and the length of long axis.

Default value: 0.6

Suggested value: Set according to the actual scenarios.

Max Inertia

Description: This parameter is the upper limit of the ratio between the length of short axis and the length of long axis.

Default value: 1

Suggested value: Set according to the actual scenarios.

CloudCapacityFilter

CloudCapacityFilter could filter out the point clouds based on the number of the point in the point clouds.

Min Number of Points

Description: This parameter is the lower limit of the number of the point in the point clouds.

Default value: 6000

Suggested value: Set according to the actual scenarios.

Valid range: (0, maxPointsNum)

Max Number of Points

Description: This parameter is the upper limit of the number of the point in the point clouds.

Default value: 360000

Suggested value: Set according to the actual scenarios.

Valid range: (minPointsNum, +∞)

RecCloudsFilter

Min Rectangularity

Description: This parameter is the lower limit of the rectangularity of the rectangle. The rectangularity equals the ratio of the actual area of the point cloud to the area of the minimum circumscribed rectangle.

Default value: 0.9000

Suggested value: Set according to the actual scenarios.

SolidCloudsFilter

Ratio

Description: This parameter is the upper lower of the size ratio between the point cloud and the solid.

Default value: 0.9000

Suggested value: Set according to the actual scenarios.

Valid range: (0, 1]

Remove Solid Clouds

Description: This parameter is used to decides whether to filter out the solid point clouds or not. Solid points clouds will be removed once this parameter is selected, or else hollow point clouds will be removed.

Default value: selected.

Suggested value: Set according to the actual scenarios.

4.3.272. Point Cloud Shape Detector

Function

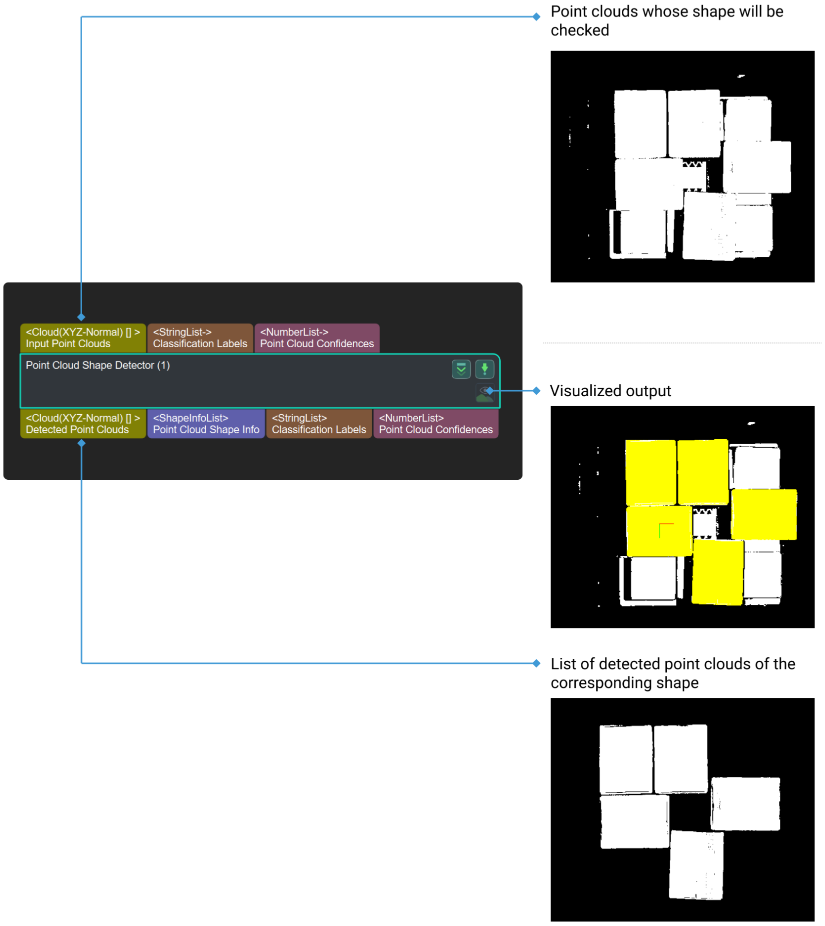
This Step detects and outputs point clouds of a plane or cylindrical shape; the shape can be selected in the Step Parameters.

Usage Scenario

This Step is usually used to detect objects with flat and cylindrical shapes.

For calculating the pick points of cylinder point clouds, this Step can be followed by the Step Calc Cylinder Poses or the Step Adjust Poses by Offsets.

Input and Output



Point clouds whose shape will be checked

Visualized output

List of detected point clouds of the corresponding shape

Parameter Description

Shapes to Detect

Instruction: This parameter is used to choose the target shape type, plane, cylinder, and sphere included.

Default setting: Plane.

Suggested setting: Set according to the actual scenarios.

Detection Settings

Detector

Instruction: This parameter is used to choose the type of detector used during detection, including Method 1 and Method 2. (Upon optimization, there will be only Method 1.)

Default setting: Method 1

Suggested setting: Method 2

Normal Deviation

Instruction: This parameter is used to set the upper limit of the angle difference (in °) between the normal of point in the image and the normal of projected theoretically corresponding standard shape.

Default value: 30°

Suggested value: Set according to the actual scenarios.

Acceptable Point Cloud Fluctuation

Instruction: This parameter is the upper limit (in mm) of the acceptable range of the point cloud fluctuation error.

Default value: 5.000 mm

Suggested value: [3.000, 5.000].

Min Cluster Interval

Instruction: This parameter is the lower limit (in mm) of the interval of point cloud connectivity judgement. When this parameter becomes large, points far away from each other will be clustered into one class; when it becomes smaller, however, points close to each other will be clustered into different classes.

Default value: 5.000 mm

Suggested value: Set according to the actual scenarios.

Point Cloud Quantization Resolution

Instruction: This parameter is used to control the number and intensity of points. The unit is millimeters. Greater number and intensity corresponds to higher accuracy and longer computing time.

Default value: 5.000 mm

Suggested value: Usually twice the value of clusterEpsilon.

Number of Results per Point Cloud

Instruction: This parameter determines the number of final outputted result for every point cloud.

Default value: 1

Suggested value: Set according to the actual scenarios.

Sort by Point Counts

Instruction: This parameter decides whether to sort the final result according to the size of point cloud or not.

Default setting: selected.

Suggested value: Set according to the actual scenarios.

Success Probability (0–1.0)

Instruction: This parameter is used to adjust the success rate of detection.

Default value: 0.9900

Suggested value: Set according to the actual scenarios.

Candidates Filter

Min Shape Point Count Ratio (0–1.0)

Instruction: This parameter is used to set the lower limit of the ratio between the detected points belonging to one shape and the total number of points with shape labels. The points that have shape labels and are under this limit will be filtered out.

Default value: 0.1000

Suggested value: Set according to the actual scenarios.

Min Mead Dist Error (0–1.0)

Instruction: This parameter is used to set the lower limit of the mean distance deviation(in mm) between real points and the points of corresponding detected theoretical shape. When this parameter becomes larger, point clouds with huge deviations will be retained; when it becomes smaller, however, point clouds with small deviations will be filtered out.

Default value: 10.000 mm

Suggested value: Set according to the actual scenarios.

Min Mean Angle Error (0–90)

Instruction: This parameter is used to set the lower limit of the mean angle deviation (in °) between real points and the points of corresponding detected theoretical shape. When this parameter becomes larger, point clouds with huge deviations will be retained; when it becomes smaller, however, point clouds with small deviations will be filtered out.

Default value: 45.0000

Suggested value: Set according to the actual scenarios.

Cylinder Property

Min Radius

Instruction: This parameter is used to set the lower limit of the cylinder (in mm). Cylinders with a radius lower than this limit will be filtered out.

Default value: 10.000 mm

Suggested value: Set according to the actual scenarios.

Max Radius

Instruction: This parameter is used to set the upper limit of the cylinder (in mm). Cylinders with a radius greater than this limit will be filtered out.

Default value: 100.000 mm

Suggested value: Set according to the actual scenarios.

Optimize Cylinder Positions

Instruction: This parameter decides whether to optimize the shape parameters of cylinder, center point and axis included.

Default Value: Unselected.

Suggested Setting: Selected.

4.3.273. Point Filter

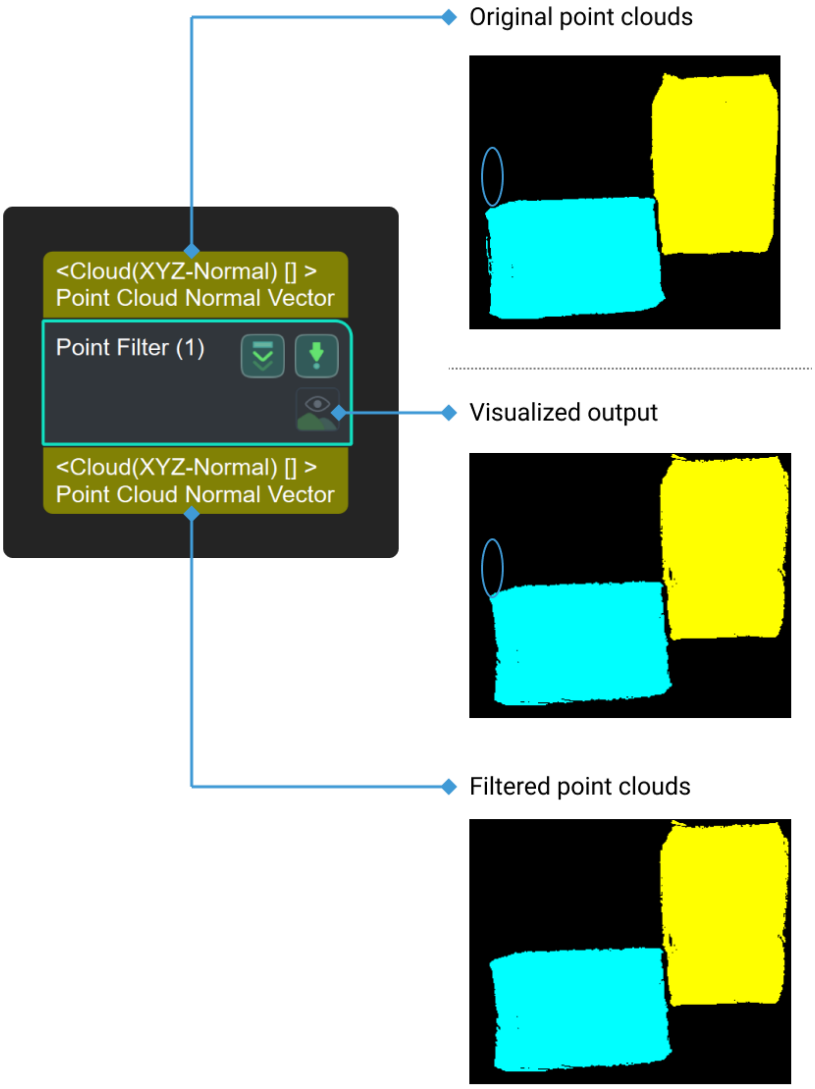
Function

Filter points based on the rules specified by the user.

Usage Scenario

This Step is usually used in the point cloud pre-processing phase. It can remove outliers and filter other unwanted points to facilitate subsequent clustering operations on the point clouds.

Input and Output

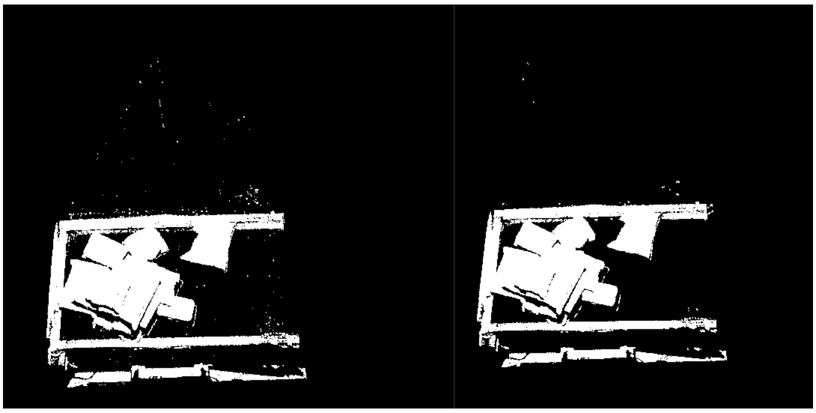


Parameter Description

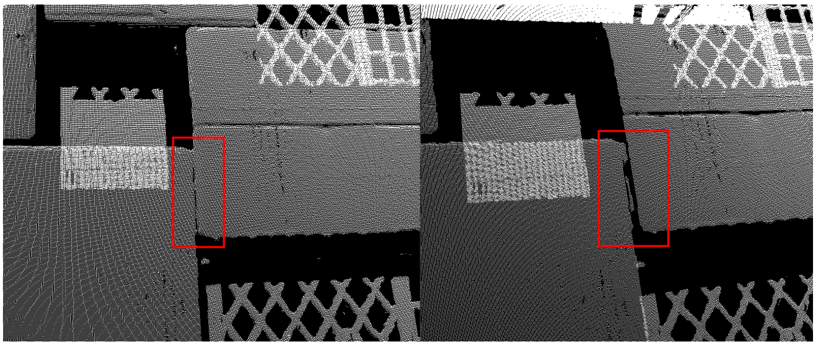
Points Filter Type

Instruction: There are two methods for filtering points in the point cloud, which can be selected according to the actual situation of the project:

- StatisticalOutlierFilter: Points out of range are excluded according to a given parameter. This method is recommended when the point cloud has obvious outliers, as shown below.



- NormalsFilter: The angle between each point in the point cloud and the reference direction is calculated according to the given reference direction. Points outside the range will be filtered. It is recommended to be used when point clouds need to be layered to avoid conglutination, as shown in the red box below.



StatisticalOutlierFilter

Mean K

Default value: 30

Instruction: This parameter is used to control the number of nearby points for each search. The larger the parameter is, the more adjacent points are used to calculate the average distance value, and the wider the range is covered.

Standard Deviation Multiplier

Default value: 2

Instruction: This parameter is used to adjust the multiplier of standard deviation used in calculating the threshold. When the parameter is reduced, the lower the threshold, the fewer points are reserved. However, the number of points is too sparse, the information will also be reduced.

NormalsFilter

Reference Direction

Use Robot Z in Camera

Default value: unselected.

Instruction: This parameter decides whether to use the Z axis of robot in camera or not. If this parameter is selected, **Align X, Align Y, and Align Z** are invalid.

Align X

Default value: 0

Instruction: This parameter is used to determine the reference axis of normal filtering. A vector consisting of X,Y, and Z parameters sets the direction of the reference axis. When **Use Robot Z in Camera** is selected, the three parameters are invalid and the reference axis is robot Z-axis.

Align Y

Default value: 0

Instruction: same as the adjustment method of **Align X**.

Align Z

Default value: −1

Instruction: same as the adjustment method of **Align X**.

Angle Thresholding Settings

Min Angle

Default value: 0

Instruction: This parameter is used to control the lower limit of Angle difference in the process of filtering point, in °. When the angular difference between the normal vector of a point and the reference vector is below this threshold, the point is filtered out.

Max Angle

Default value: 90

Instruction: This parameter is used to control the upper limit of Angle difference in the process of filtering point, in °. When the angular difference between the normal vector of a point and the reference vector exceeds this threshold, the point is filtered out.

4.3.274. Adjust Poses

Function

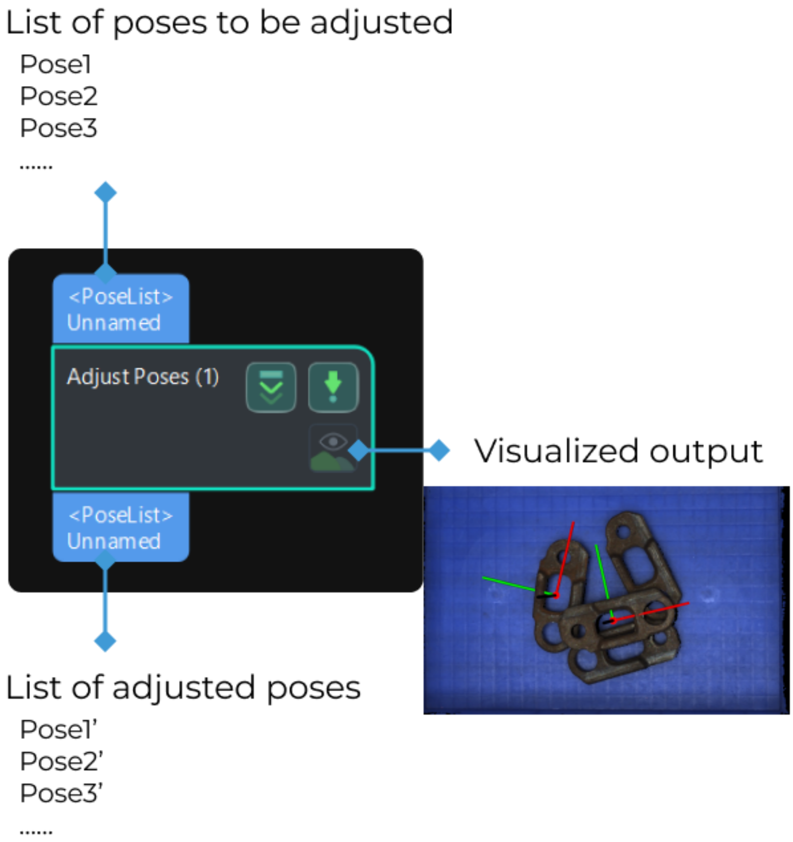
This Step enables adjustments such as coordinate system conversion, pose translation, pose rotation, and pose sorting.

Usage Scenario

This Step is a general pose adjustment Step.

This Step is outdated. Please use the new Step Adjust Poses V2 which is integrated with a pose adjustment tool with a wider array of features.

Input and Output



Parameter Description

Operations

Description: This parameter is the entrance for the pose editor.

Tuning recommendation: Click **[Open Pose Editor]** to enter the pose editor. For more information, please refer to Pose Editor.

Entry Port Description

Description: This parameter is used to set the description of the entry ports, which need to be separated by a semicolon.

Tuning recommendation: For example, Pose; Index correspond to the Poselist; IndexList data types, respectively.

Exit Port Description

Description: This parameter is used to set the description of the entry ports. The description should be separated by semicolons.

Tuning recommendation: For example, Pose; Index correspond to the Poselist; IndexList data types, respectively.

Complete Description for Internal Sub-Steps

Description: This parameter is used to enter the full description of the internal Steps.

Complete Description for Sub-Steps IO Connections

Description: This parameter is used to enter the full description of the internal Step IO connection.

4.3.274.1. Pose Step Editor

Function

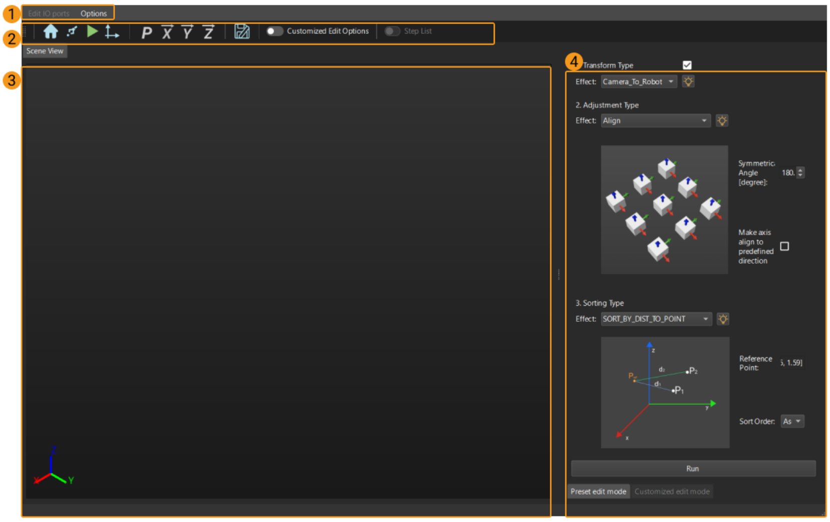
It is used to improve the usability of pose adjustment, the visualization of poses while editing, and reduce the learning cost of pose adjustment.

There are two modes of the pose editor:

1. Preset Edit Mode: Can edit poses quickly by common functions such as reference frame transformation, adjustment, and sorting.
2. Customized Edit Mode: Adjust poses by adding corresponding Steps and the edited poses can be seen.

Introduction to Interface

The interface of the pose editor, as shown below, consists of four parts:



Transform Type ☑
Effect: Camera_To_Robot
2. Adjustment Type
Effect: Align
Symmetric Angle [degree]: 180.
Make axis align to predefined direction □
3. Sorting Type
Effect: SORT_BY_DIST_TO_POINT
Reference Point: 5, 1.59]
Sort Order: As

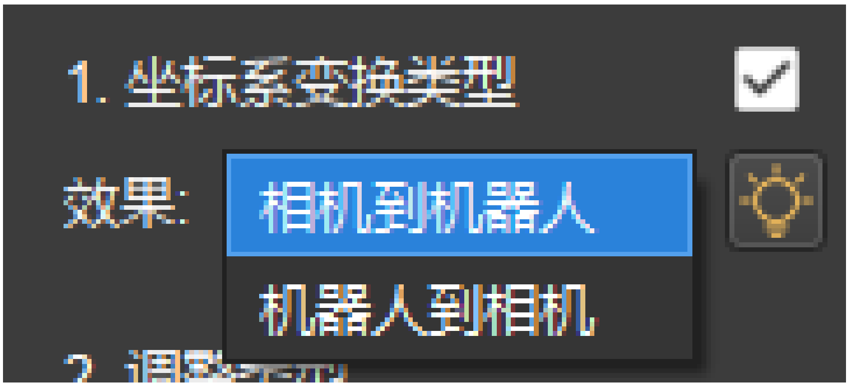
Interface	Description
Menu Bar	Edit IO ports, set dragger type, etc.
Toolbar	Edit and adjust poses.
Scene View (visualization area)	View edited poses for further adjustments.
Editing Area	Adjust poses and switch edit modes (the editing area differs with the different mode selected).

Mode

Preset Edit Mode

There are three types of functions:

Transform Type: Two transform types, which can transform the reference frame of the pose by selecting **camera to robot** or **robot to camera**.



1. 坐标系变换类型 ☑

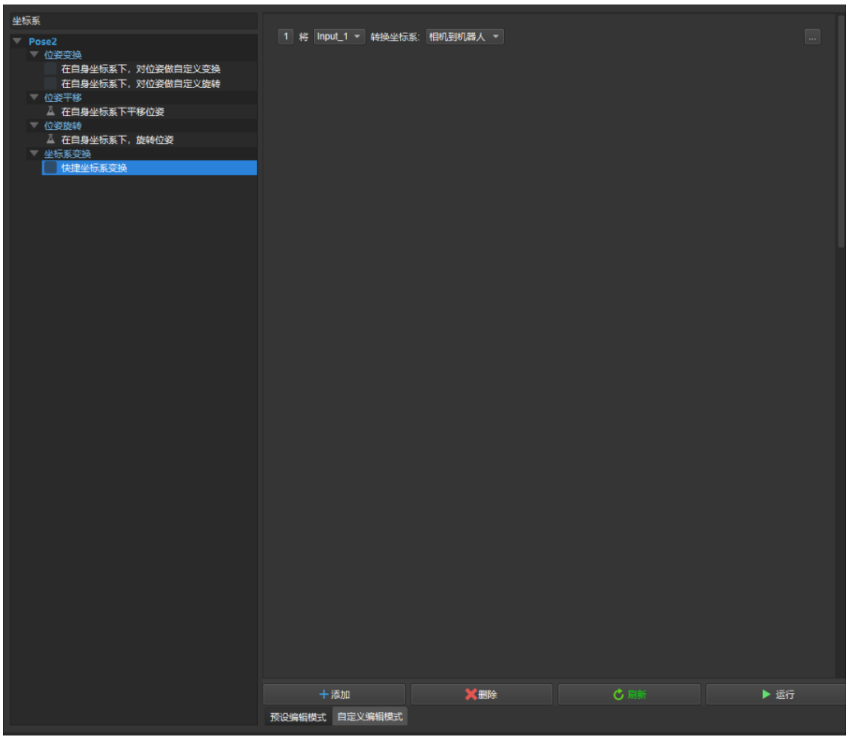
效果: 相机到机器人

Adjustment Type: Three adjustment types: 1. Align; 2. outwards; 3. point to.

Sorting Type: Five sorting types: 1. Sort by custom direction; 2. sort by dist to point; 3. sort by dist to point on XY plane; 4. sort by position XYZ; 5. sort by row or column.

Customized Edit Mode

First, click **[Customized Edit Options]** on the toolbar and click Step List. Then, select the corresponding Step from the Step List and click **[Add]**.

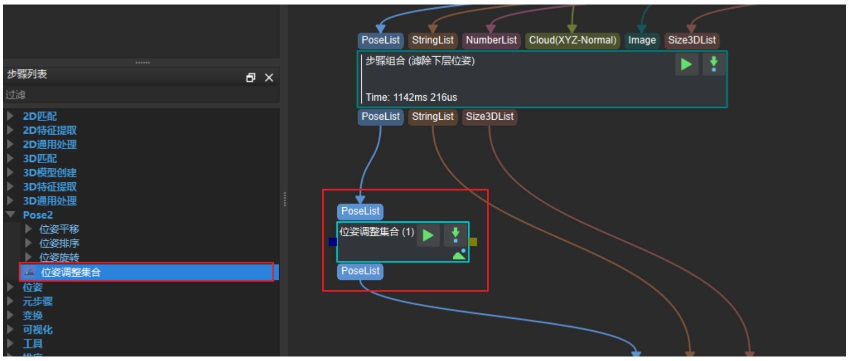


1 将 Input_1 转换坐标系: 相机到机器人

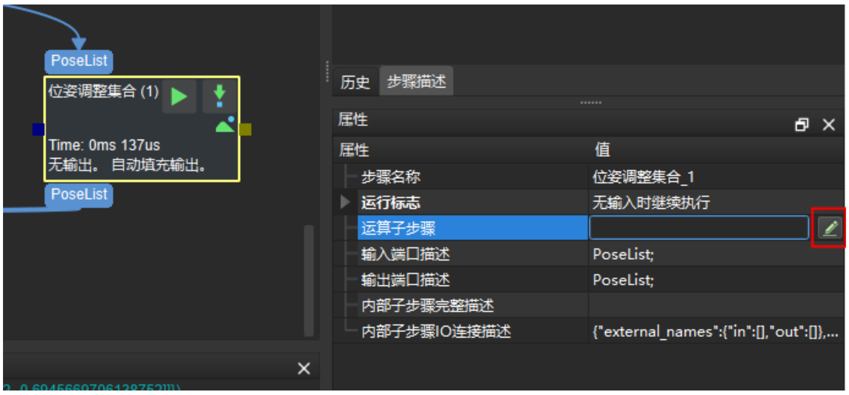
Usage

The following section introduces how to use the Pose Step Editor.

Step 1: Add the Step Adjust Poses in the project, as shown in the figure below.

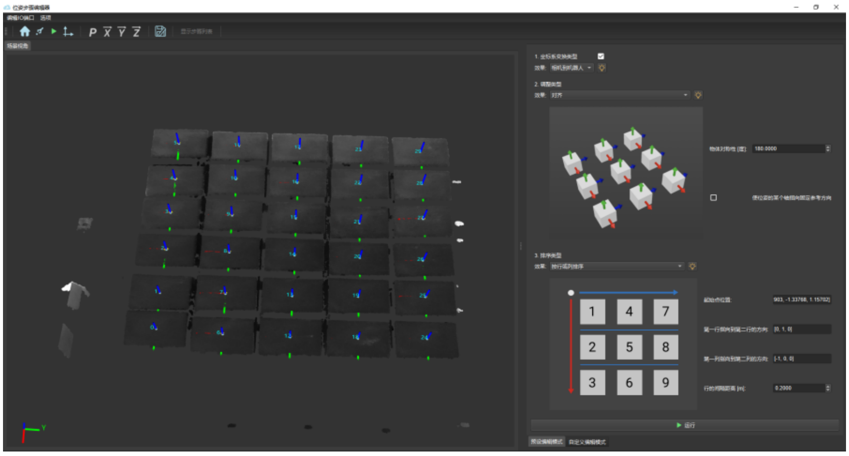


Step 2: Run the project to get the input pose and click ✎ to enter the editor, as shown below.



属性	值
步骤名称	位姿调整集合_1
运行标志	无输入时继续执行
运算子步骤	
输入端口描述	PoseList;
输出端口描述	PoseList;
内部子步骤完整描述	
内部子步骤IO连接描述	{"external_names":{"in":[],"out":[]},...

Step 3: Select and set an edit mode. In Preset Edit Mode, please set Transform Type, Adjustment Type, and Sorting Type. Then click Run to preview the results. The result is shown in the figure below.



1. 坐标系变换类型 ☑
效果: 相机到机器人

2. 调整类型
效果: 对齐

物体对称性 [度]: 180.0000

□ 使位姿的某个轴指向固定参考方向

3. 排序类型
效果: 按行或列排序

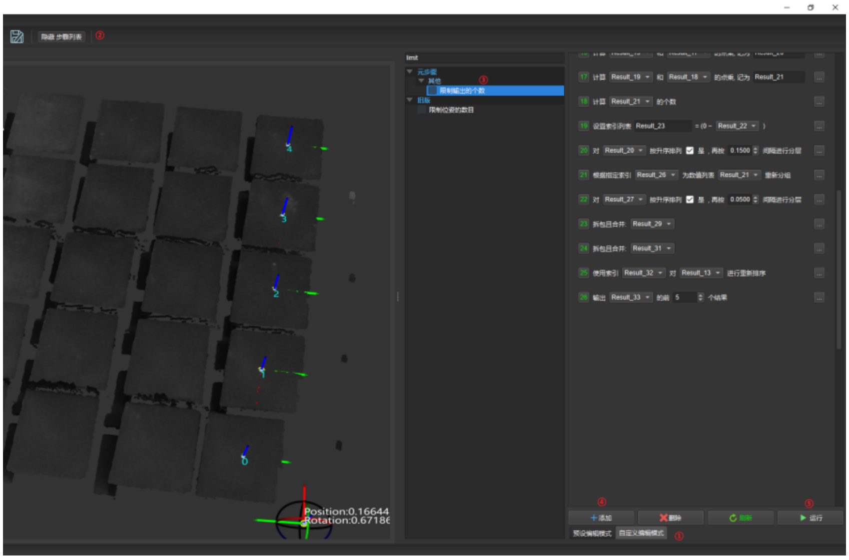
起始点位置: 903, -1.33768, 1.15702]

第一行指向到第二行的方向: [0, 1, 0]

第一列指向到第二列的方向: [-1, 0, 0]

行的间隔距离 [m]: 0.2000

Step 4: In Customized Edit Mode, please add output quantity limit: 1. Click the Customized Edit Options. 2. Click Step List. 3. Select the number of displayed outputs. 4. Click Add. 5. Fill in the limit number. 6. Run and preview the results. The result is shown in the figure below.



17 计算 Result_19 和 Result_18 的点乘, 记为 Result_21

18 计算 Result_21 的个数

19 设置索引列表 Result_23 = (0 ~ Result_22)

20 对 Result_20 按升序排列 ☑ 是，再按 0.1500 间隔进行分层

21 根据指定索引 Result_26 为数值列表 Result_21 重新分组

22 对 Result_27 按升序排列 ☑ 是，再按 0.0500 间隔进行分层

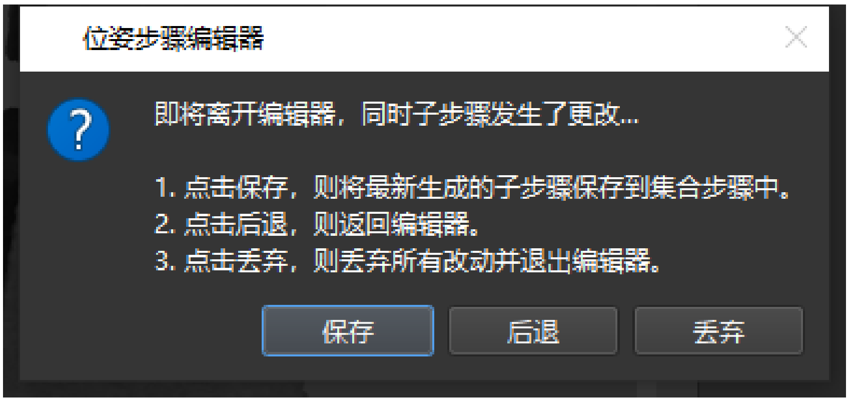
23 拆包且合并: Result_29

24 拆包且合并: Result_31

25 使用索引 Result_32 对 Result_13 进行重新排序

26 输出 Result_33 的前 5 个结果

Step 5: After finishing the pose editing, click the exit button to save the edited pose.



位姿步骤编辑器

即将离开编辑器，同时子步骤发生了更改...

1. 点击保存，则将最新生成的子步骤保存到集合步骤中。
2. 点击后退，则返回编辑器。
3. 点击丢弃，则丢弃所有改动并退出编辑器。

Step 6: Run the project to review the results.

4.3.275. Adjust Poses V2

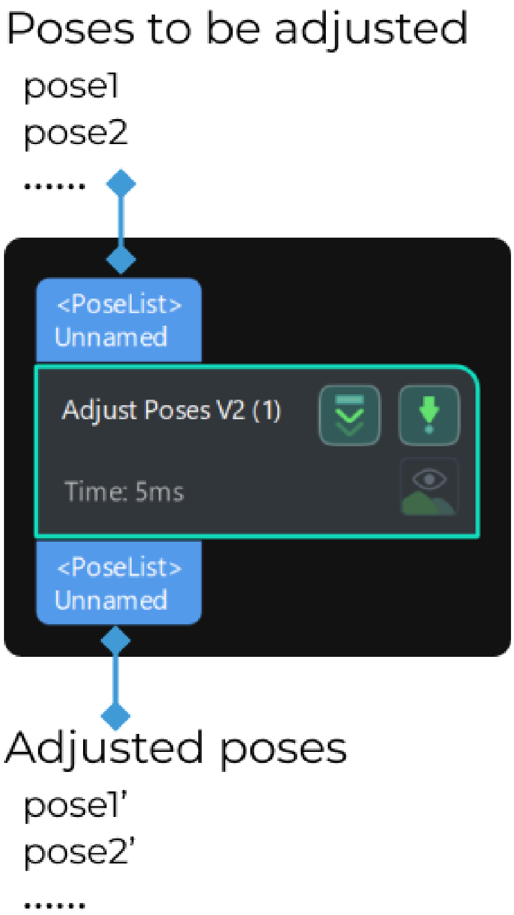
Function

This Step allows you to manipulate poses, including transforming them into the robot reference frame, as well as performing translation, rotation, and sorting operations.

Usage Scenario

This Step can be widely used in pose adjustment scenarios.

Input and Output



Parameter Description

Open the editor

Description: This parameter is the entry for the new pose adjustment tool.

Tuning recommendation: Click **[Open the editor]** to open the new pose adjustment tool. Refer to User Interface Introduction for the introduction of the user interface and features. Refer to Pose Adjustment Tool for detailed instructions.

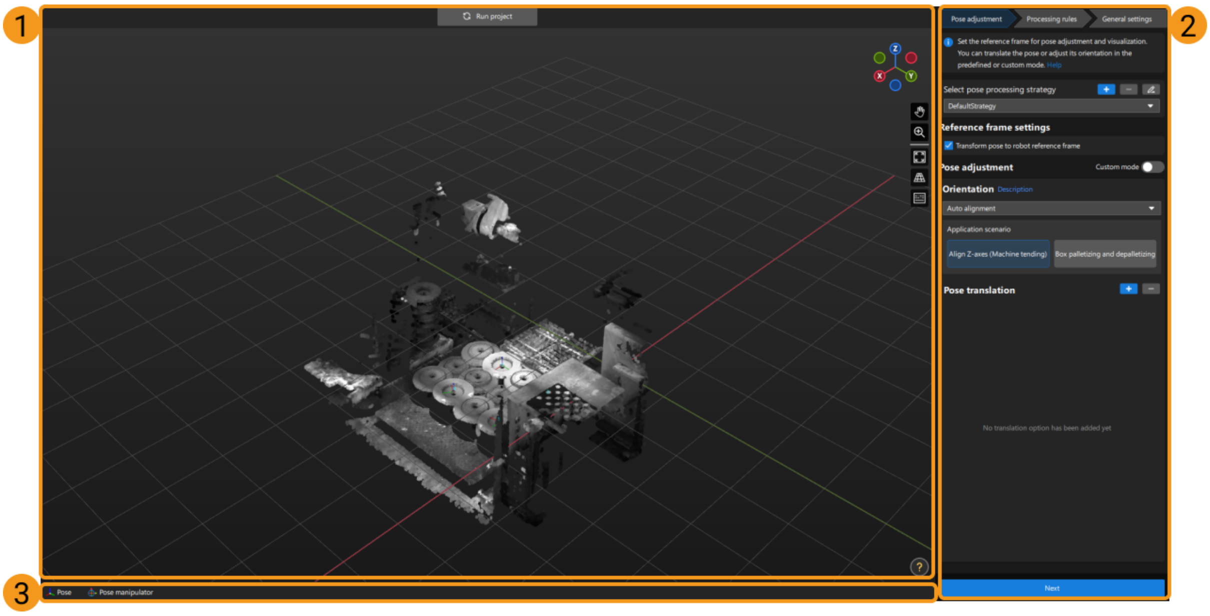
Select Pose Processing Strategy

Description: This parameter specifies the pose processing strategy.

Note: If you need to switch the pose processing strategy when the pose adjustment tool is opened, switch it in the tool, or close the tool and retry.

4.3.275.1. User Interface Introduction

The interface of the pose adjustment tool is shown below.



Set the reference frame for pose adjustment and visualization. You can translate the pose or adjust its orientation in the predefined or custom mode. Help

Select pose processing strategy: DefaultStrategy

Reference frame settings

☑ Transform pose to robot reference frame

Pose adjustment — Custom mode

Orientation Description: Auto alignment

Application scenario: Align Z-axes (Machine tending) | Box palletizing and depalletizing

Pose translation

No translation option has been added yet

The following table describes the functions of each area in the figure above.

No.	Area	Description
1	Visualizing space	Displays of the processed poses.
2	Parameter adjustment area	Displays parameters related to pose adjustment, sorting, and filtering.
3	Reference information area	Displays reference information such as view operation instructions.

4.3.275.2. Pose Adjustment Tool

Function

The pose adjustment tool is used to improve the usability of pose adjustment, enable the visualization of poses while adjusting, and reduce the learning cost of pose adjustment. The **Pose adjustment, Processing rules, and General settings** three steps provide settings related to pose adjustment, pose sorting, pose filtering, etc.

Pose adjustment → Processing rules → General settings

1. Pose adjustment: for translating poses and adjusting the orientations of the poses.
2. Processing rules: sort poses according to the actual requirement and filter out ineligible poses.
3. General settings: provides settings other than pose processing settings. Currently, this step only supports sorting other data together with the corresponding poses.

Operation Workflow

The procedure for using this tool is as follows.

Pose Adjustment

This Step is used for translating poses and adjusting the orientations of the poses. See the following for details.

Select Pose Processing Strategy

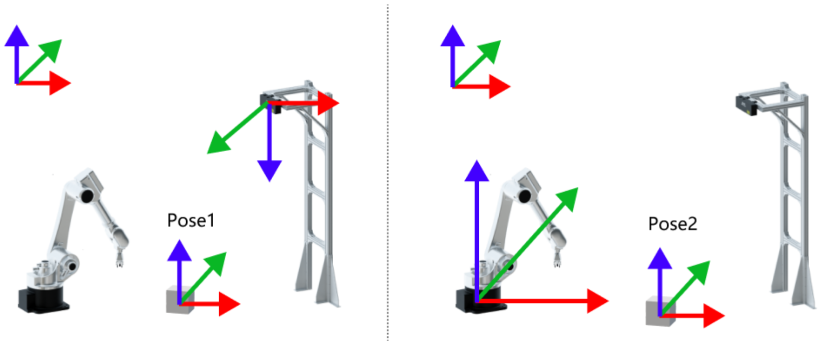
Before adjusting the orientation and position of the pose, please select a **Pose processing strategy** first. Click the **[+]** button on the right to add a pose processing strategy.

Once you finish all configurations or want to exit the tool, click **[Save]** in the “General Settings” window to save all the configurations to the selected pose processing strategy.

Reference Frame Settings

This setting is used to transform the poses from the camera reference frame to the robot reference frame. Once **Transform pose to robot reference frame** is enabled, poses in the robot reference frame can be output.

As shown below, the figure on the left shows the pose in the camera reference frame, and the figure on the right shows the pose in the robot reference frame.



Pose Adjustment

You can adjust the orientations and positions of the poses in this section. Two modes, **Custom mode** and **predefined mode**, are available. If the Custom mode is not enabled, the tool remains in the predefined mode.

- If you want to adjust the orientations or positions of the poses with the predefined options, you can stay in the default **predefined mode**.
- If you want to make more complex adjustments to the orientations or positions of the poses, you can enable **Custom mode**.

Predefined Mode

In the predefined mode, the predefined options can be used to make simple adjustments to the position and orientation of the poses.

- Orientation

  Select the option for orientation adjustment according to the actual requirement, including **Auto alignment**, **Point to reference pose**, **Point to/away from pallet center (depalletizing)**. Refer to Adjust Pose Orientation (Predefined Mode) for detailed instructions.

- Pose translation

  Click the [+] button on the right to add an option for translation. Only one translation option, “Translate pose along specified direction”, is available in the predefined mode. Refer to Translate Poses (Predefined Mode) for detailed instructions.

Custom Mode

Once the Custom mode is enabled, you can click the [+] button on the right to add the following pose adjustment options. Please click the links below to view detailed instructions.

- Flip pose and minimize the angle between the flip axis and target direction
- Rotate pose and minimize the angle between the rotation axis and target direction
- Rotate pose axis to target direction
- Point pose axis to reference pose
- Align the pose orientation with the target pose orientation
- Rotate pose around specified axis
- Translate pose along specified direction

Processing Rules

Sort poses according to the actual requirement and filter out ineligible poses.

Sorting Type

The following seven sorting types are provided.

- Do not sort
- Sort by S shape on plane
- Sort by Z shape on plane
- Sort by X/Y/Z value of pose
- Sort by distance from pose to reference pose
- Sort by distance from pose to reference pose on XOY plane
- Sort by pose confidence

Filter Pose

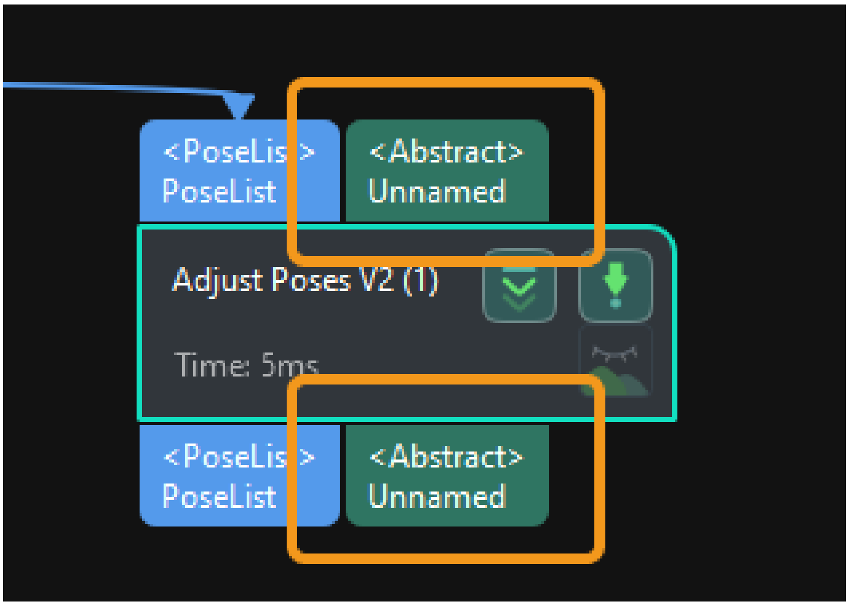
Two options for pose filtering, **Filter by angle** and **Filter out poses out of ROI**, are provided. Please refer to Filter Poses by Predefined Options for detailed instructions.

General Settings

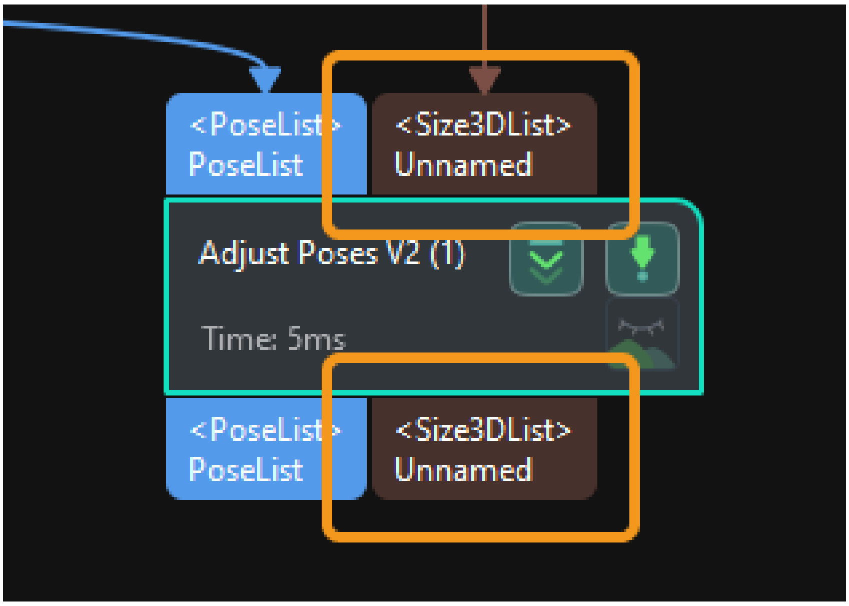
Once the number of new ports is specified, input ports compatible with all data types will be added to the “Adjust Poses V2” Step. Then you can connect the port outputting the data that needs to be sorted with the newly added input port.

> Up to 5 new ports can be added.

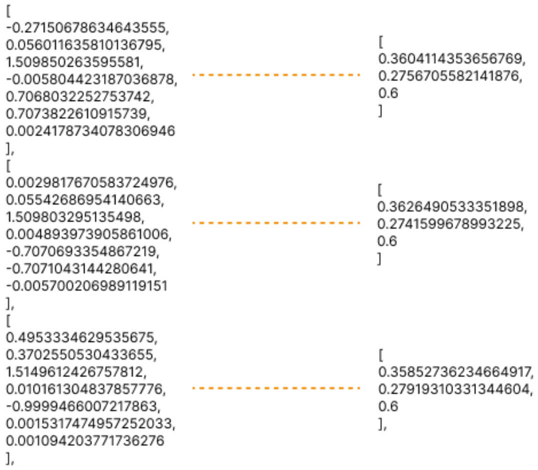
For example, when the **Set number of new ports** is **1**, one new port compatible with all data types is added, as shown below.



If you want to sort the object dimensions, connect the output port of the Step outputting the object dimensions with this new input port.



Then click **[Get data]** to sort the object dimensions according to the sorting result of the poses. As shown below, the dimensions and their corresponding poses match each other after sorting.



```
[
-0.27150678634643555,
0.056011635810136795,                     [
1.509850263595581,                        0.3604114353656769,
-0.005804423187036878,  - - - - - - - -   0.2756705582141876,
0.7068032252753742,                       0.6
0.7073822610915739,                       ]
0.0024178734078306946
],
[
0.0029817670583724976,                    [
0.05542686954140663,                      0.3626490533351898,
1.509803295135498,      - - - - - - - -   0.2741599678993225,
0.004893973905861006,                     0.6
-0.7070693354867219,                      ]
-0.7071043144280641,
-0.005700206989119151
],
[
0.4953334629535675,                       [
0.3702550530433655,                       0.35852736234664917,
1.5149612426757812,     - - - - - - - -   0.27919310331344604,
0.010161304837857776,                     0.6
-0.9999466007217863,                      ],
0.0015317474957252033,
0.001094203771736276
],
```

4.3.276. Pose Filter

This Step will no longer be updated. Please use the new Step Validate Poses by Included Angles to Reference Direction instead. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

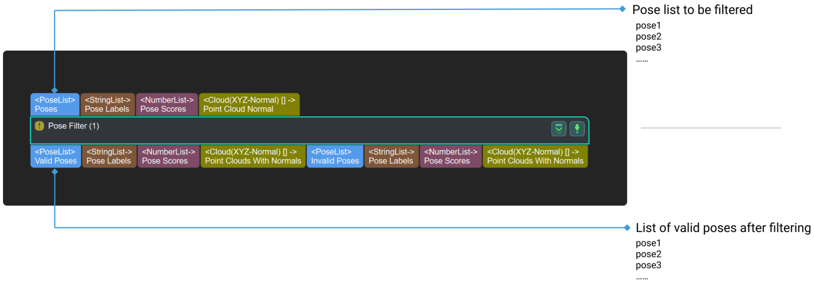
Function

Calculate the angle between the Z-axis of the input pose and the reference direction. Those having angles smaller than the threshold will be retained, and those having angles larger than the threshold will be removed.

Usage Scenario

This Step is outdated. Please use the new Step Validate Poses by Included Angles to Reference Direction. This Step can only calculate the angle between the Z-axis and a reference direction. The new Step supports filtering poses by calculating the angle between any axes and a reference direction.

Input and Output


Pose list to be filtered
pose1
pose2
pose3
......
<PoseList> Poses
<StringList-> Pose Labels
<NumberList-> Pose Scores
<Cloud(XYZ-Normal) [] -> Point Cloud Normal
Pose Filter (1)
<PoseList> Valid Poses
<StringList-> Pose Labels
<NumberList-> Pose Scores
<Cloud(XYZ-Normal) [] -> Point Clouds With Normals
<PoseList> Invalid Poses
<StringList-> Pose Labels
<NumberList-> Pose Scores
<Cloud(XYZ-Normal) [] -> Point Clouds With Normals
List of valid poses after filtering
pose1
pose2
pose3
......

4.3.277. Rotate Poses by Quaternion Vectors in Object Frames

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

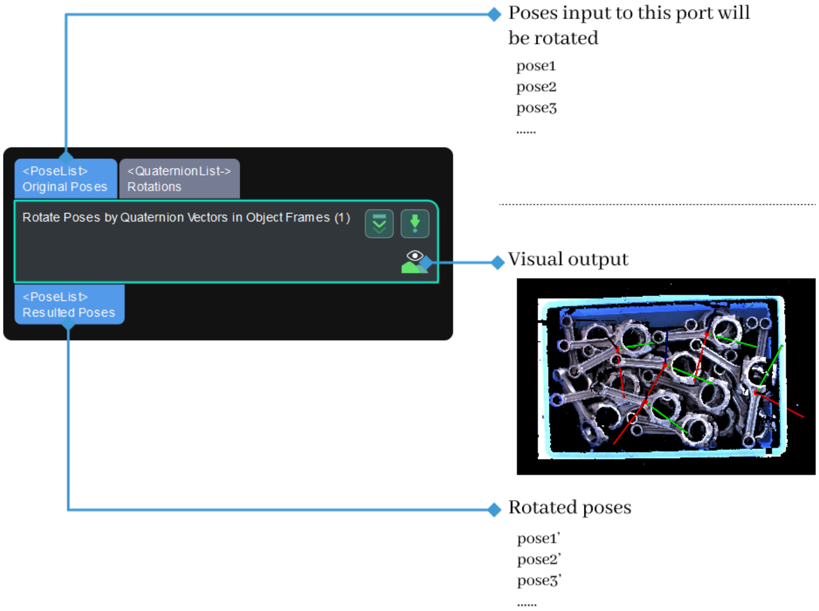
Function

Rotate the poses using the rotation vectors.

Usage Scenario

A general pose rotation Step. No fixed usage scenarios.

Input and Output



4.3.278. Transform Poses by Matrix in Object Frames

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

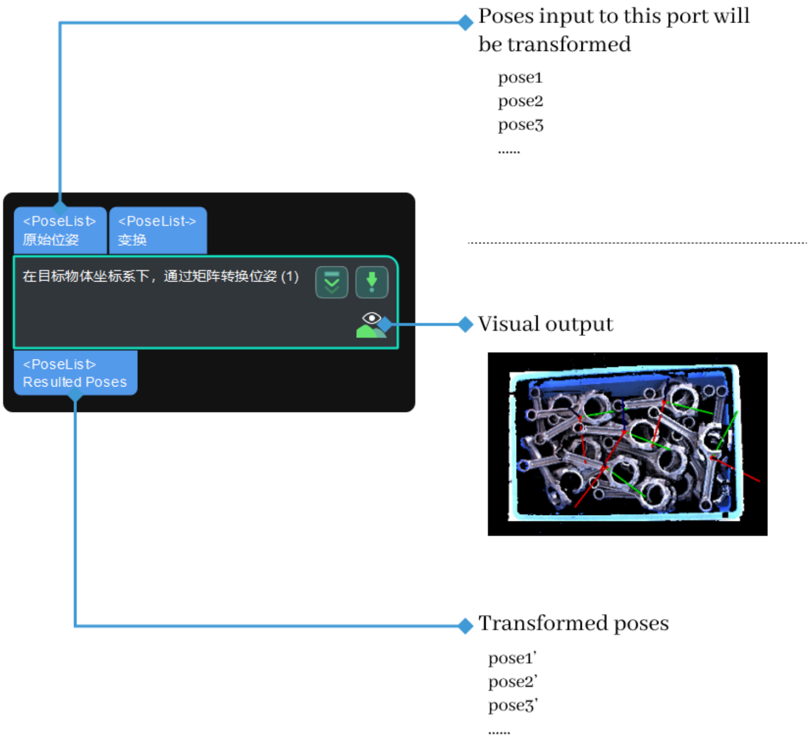
Function

Transform the poses in the reference frames of the poses themselves.

Usage Scenario

A general pose transformation Step. No fixed usage scenarios.

Input and Output



4.3.279. Check Pose Repeatability by Statistics

Function

This Step is used to evaluate the stability of the poses in the vision projects.

In the actual application, the camera will capture 10–20 images on the same workpiece at each position first, and this Step will calculate the minimum, maximum, mean, range, and standard deviation of the workpiece poses at each position, and then you can determine whether the project runs stably according to the statistics.

> No matter whether the camera is mounted in the eye-in-hand or eye-to-hand mode, images of the workpiece at different positions should be captured to verify the pose stability at different positions. However, when the camera captures 10–20 images at the same position, the relative position between the camera and the workpiece should not change.
>
> - When the camera is mounted in the eye-to-hand mode, only the workpiece can be moved.
> - When the camera is mounted in the eye-in-hand mode, both the workpiece and camera can be moved. Under this circumstance, please ensure that the relative position between the camera and the robot does not change when the camera continuously captures images on the same workpiece at one of the positions.

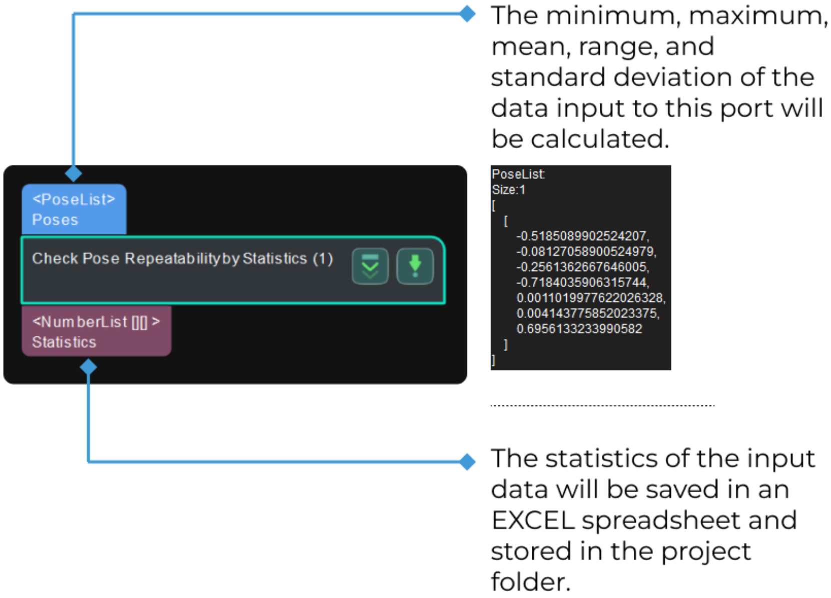
Usage Scenario

This Step can be used to evaluate the robustness of the camera or verify the effectiveness of the vision project parameter tuning.

It can follow 3D matching Steps to evaluate the robustness of matching. You can also connect it before the Procedure Out Step in a project to verify the stability of positioning and recognition deviation. In the latter case, please keep the workpieces unmoved and then capture images.

After running the project, this Step will output the statistics spreadsheet to a specified directory. You can check the statistics of X, Y, Z values of the poses and X, Y, Z values of Euler angles in this file.

Input and Output



The minimum, maximum, mean, range, and standard deviation of the data input to this port will be calculated.

```
PoseList:
Size:1
[
  [
    -0.5185089902524207,
    -0.08127058900524979,
    -0.2561362667646005,
    -0.7184035906315744,
    0.0011019977622026328,
    0.004143775852023375,
    0.6956133233990582
  ]
]
```

The statistics of the input data will be saved in an EXCEL spreadsheet and stored in the project folder.

Parameter Description

Data for Making Statistics

Input Data Type

Description: This parameter is used to select the input data type which corresponds to the input port.

Value list: PoseListInput, NumberListInput.

- PoseListInput: Input one or more pose lists.
- NumberListInput: Input other data.

Default value: PoseListInput

Tuning recommendation: Please select the input data type according to the actual requirement. Please refer to the tuning example for the corresponding result.

Output Data Type

Description: This parameter specifies the output data type.

Default value: Statistics of spatial positions and Euler angles.

Tuning recommendation: This parameter does not need any adjustment. Please keep the default setting.

Position Coordinate Deviation Range (0–1000)

Description: This parameter specifies the acceptable upper limit of position coordinate deviation, in mm. If the difference between the maximum and the minimum exceeds this threshold, an error may be reported according to the setting of the **Outlier Handling** parameter. This checking process is performed independently for the X, Y, and Z coordinates.

Value range: 0–1000

Default value: 1.0 mm

Tuning instruction: Please set the value according to the actual requirement. Please refer to the tuning example for the corresponding result.

Euler Angle Deviation Range (0–360)

Description: This parameter specifies the acceptable upper limit of Euler angle deviation, in °. If the difference between the maximum and the minimum exceeds this threshold, an error may be reported according to the setting of the **Outlier Handling** parameter. This checking process is performed independently for the α, β, and γ values.

Value range: 0–360

Default value: 0.100°

Tuning instruction: Please set the value according to the actual requirement. Please refer to the tuning example for the corresponding result.

Statistics Saving

Restart

Description: When the workpiece's position changes or outliers occur in the statistics, this parameter is used to determine whether to recollect the statistics. Once this option is selected, the output data will be marked as "Restart". In the statistics file, data collected before the "Restart" mark will be excluded from the statistics and shown in gray.

Default Value: unselected.

Tuning recommendation: If you want to recollect the statistics, select this option. Please refer to the tuning example for the corresponding result.

- After the Step successfully executes again, this option will be reset to unselected automatically.
- If this option is not selected, the outlier will remain in the saved statistics.

Outlier Handling

Description: This parameter is used to specify how to handle outliers when they occur in the statistics.

Value list: Raise Error on Outlier, Record and Mark Outlier.

- Raise Error on Outlier: An error will be immediately reported when an outlier is detected and the outlier will not be recorded. The possible three causes of the error are as follows.
  - The number of elements in the data is inconsistent. In this case, you can try to use the Unpack Data Step to fix the problem.
  - The position coordinate deviation or Euler angle deviation exceeds the set threshold. In this case, you can try to adjust the **Position Coordinate Deviation Range** and **Euler Angle Deviation Range** parameters.

    - The statistics file has been opened and the new data cannot be saved. In this case, please close the statistics file and re-run the project.
- Record and Mark Outlier: No error will be reported if an outlier occurs. The outlier will be recorded in the file and marked in red, and it will also be included in the statistics.

Default value: Raise Error on Outlier.

Tuning instruction: Please set the value according to the actual requirement. Please refer to the tuning example for the corresponding result.

Number of Decimal Places

Description: This parameter specifies the number of decimal places used for the statistics.

Default value: 3

Value range: 0–8

Tuning instruction: Please set the value according to the actual requirement.

File Base Name

Description: This parameter is used to set the name of the statistics file. If you do not name the file, the output file will be named “posesStatistics” by default. If you modified the file base name, the **Restart** parameter will be automatically selected and the statistics will be recollected.

Default setting: none.

Tuning instruction: Please set the value according to the actual requirement.

> If the position of the workpiece changes, or outliers occur in the statistics, you can select **Restart** and reset the **File Base Name** to re-create a statistics file to record the data.

Saving Directory Path

Description: This parameter specifies the saving directory of the statistics file. If you do not select the saving directory, the statistics file will be saved in the project folder by default.

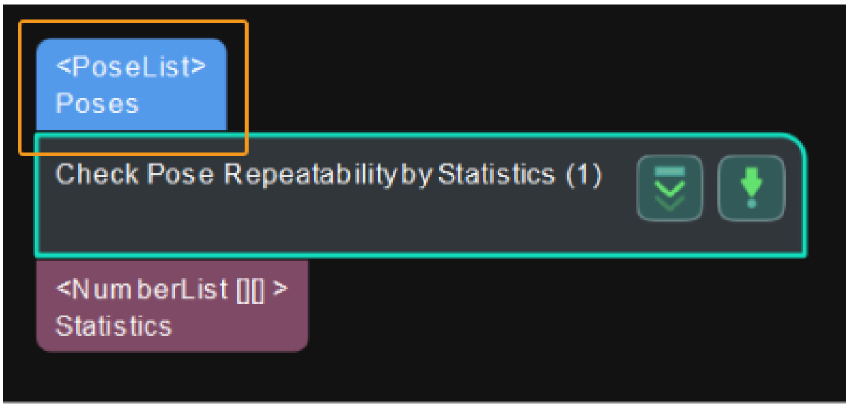
Default setting: none.

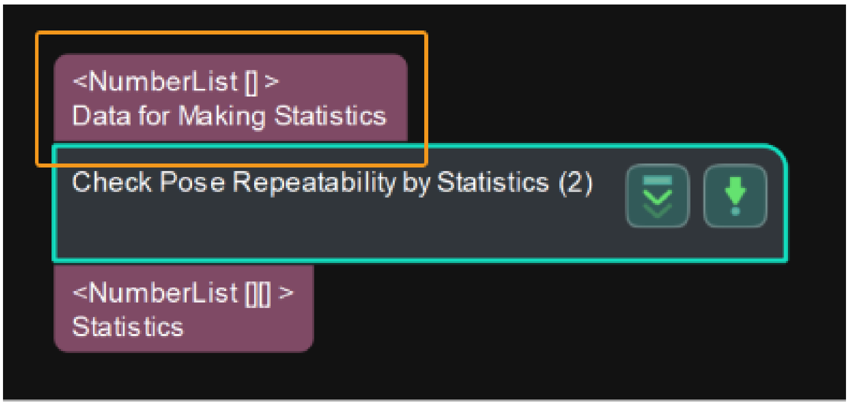
Tuning recommendation: Click the 🗀 button to choose a saving directory.

Tuning Examples

Input Data Type

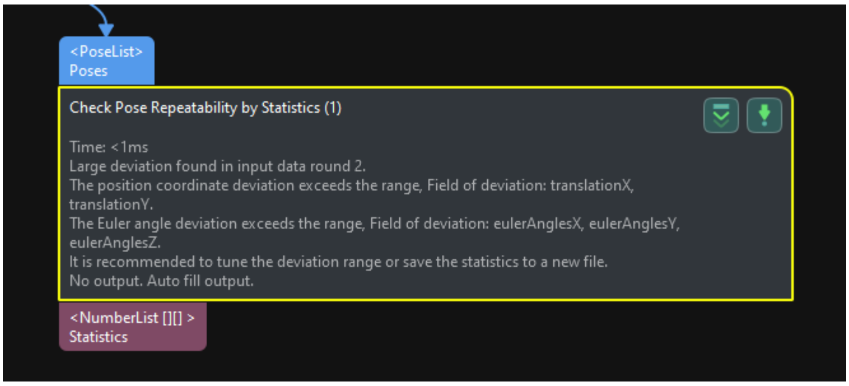
As shown below, when different **Input Data Types** is selected, the input port of the Step will change accordingly.




<NumberList [] >
Data for Making Statistics
Check Pose Repeatability by Statistics (2)
<NumberList [][] >
Statistics

Position Coordinate/Euler Angle Deviation Range

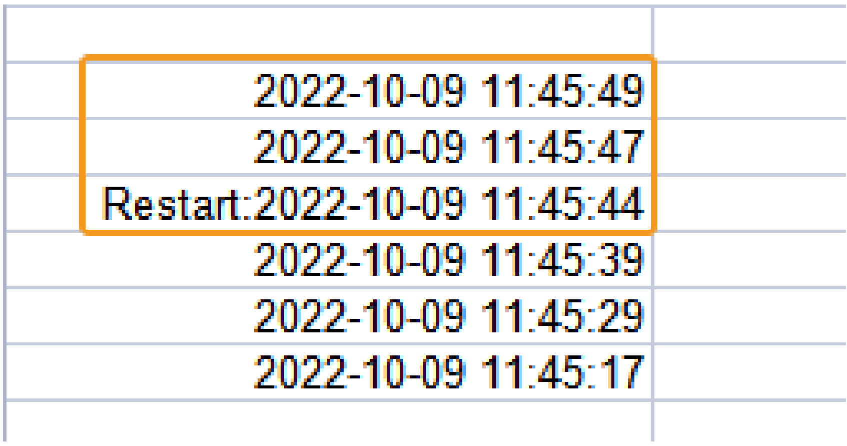
When the difference between the current set of data and the last set of data exceeds the set value, an error message as shown below will appear on the Step.



Time: <1ms
Large deviation found in input data round 2.
The position coordinate deviation exceeds the range, Field of deviation: translationX, translationY.
The Euler angle deviation exceeds the range, Field of deviation: eulerAnglesX, eulerAnglesY, eulerAnglesZ.
It is recommended to tune the deviation range or save the statistics to a new file.
No output. Auto fill output.

Restart

When **Restart** is selected, data collected after the “Restart” mark (including the data with the Restart mark) will be included in the calculation while the data collected before the “Restart” mark will be excluded and shown in gray in the statistics file.



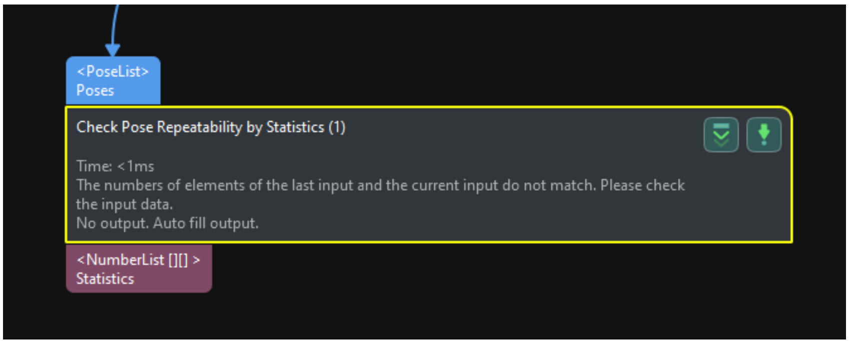
2022-10-09 11:45:49
2022-10-09 11:45:47
Restart:2022-10-09 11:45:44
2022-10-09 11:45:39
2022-10-09 11:45:29
2022-10-09 11:45:17

Outlier Handling

Raise Error on Outlier

The three types of error messages are shown below:

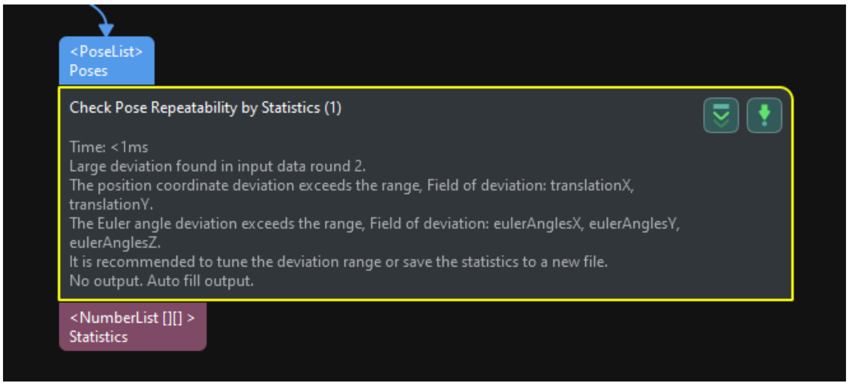
- The number of elements in the data is inconsistent.



Check Pose Repeatability by Statistics (1)

Time: <1ms
The numbers of elements of the last input and the current input do not match. Please check the input data.
No output. Auto fill output.

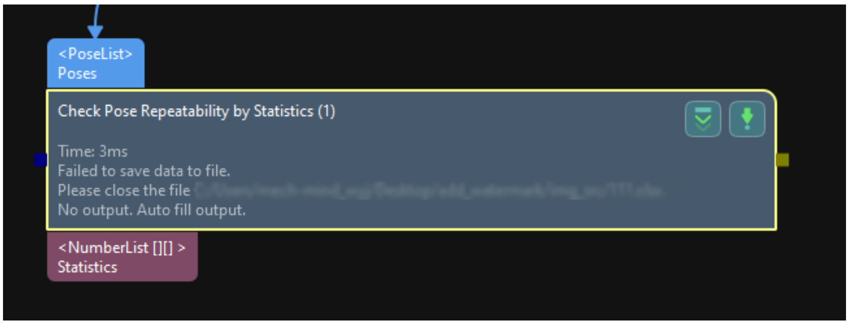
- The position coordinate deviation or Euler angle deviation exceeds the set threshold.



Check Pose Repeatability by Statistics (1)

Time: <1ms
Large deviation found in input data round 2.
The position coordinate deviation exceeds the range, Field of deviation: translationX, translationY.
The Euler angle deviation exceeds the range, Field of deviation: eulerAnglesX, eulerAnglesY, eulerAnglesZ.
It is recommended to tune the deviation range or save the statistics to a new file.
No output. Auto fill output.

- The statistics file has been opened and the new data cannot be saved.

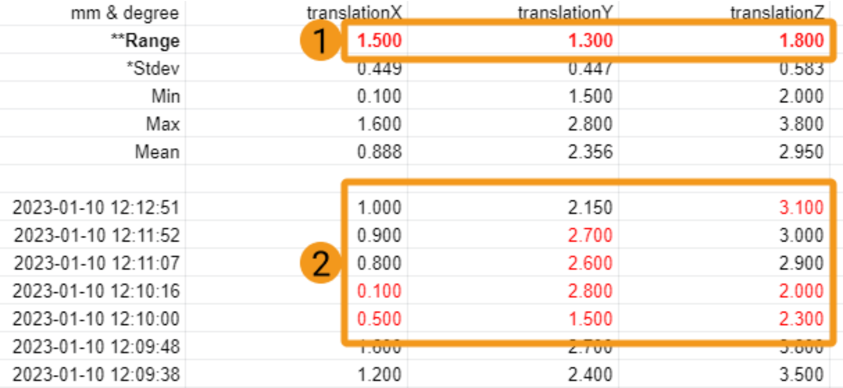


Check Pose Repeatability by Statistics (1)

Time: 3ms
Failed to save data to file.
Please close the file [redacted]
No output. Auto fill output.

Record and Mark Outlier

When this option is selected, the outliers (1 in the figure below) and the corresponding statistics (2 in the figure below) will be marked in red.



mm & degree	translationX	translationY	translationZ
**Range	1.500	1.300	1.800
*Stdev	0.449	0.447	0.583
Min	0.100	1.500	2.000
Max	1.600	2.800	3.800
Mean	0.888	2.356	2.950
2023-01-10 12:12:51	1.000	2.150	3.100
2023-01-10 12:11:52	0.900	2.700	3.000
2023-01-10 12:11:07	0.800	2.600	2.900
2023-01-10 12:10:16	0.100	2.800	2.000
2023-01-10 12:10:00	0.500	1.500	2.300
2023-01-10 12:09:48	1.000	2.700	3.000
2023-01-10 12:09:38	1.200	2.400	3.500

4.3.280. Predict Object Poses (Sim2Pick)

Function

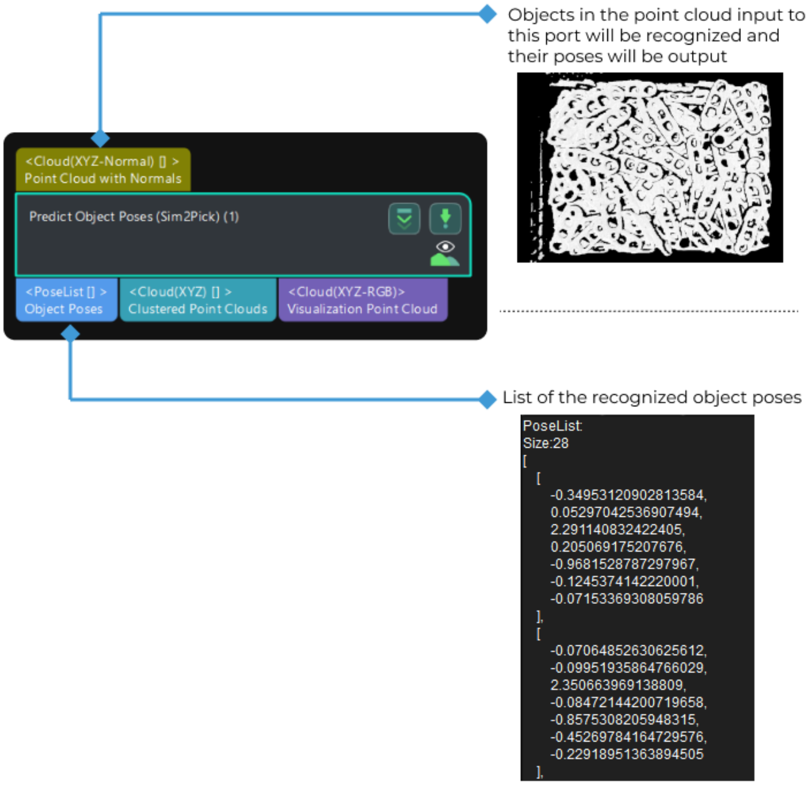
When the point cloud of the single object type is input, this Step will directly recognize the object from the input point cloud and output its pose.

Usage Scenario

This Step is used to recognize and locate objects of a single type. This Step is used to recognize and locate objects of a single type. The Step Extract 3D Points in 3D ROI is usually used before this Step to provide the pre-processed point cloud. The Step Show Point Clouds and Poses usually follows this Step to further process the outputs poses of objects in this Step.

For a high accuracy of the output object poses, the 3D ROI set in the preceding Extract 3D Points in 3D ROI Step should only include the target object and exclude the frame and bottom of the bin as much as possible.

Input and Output



Objects in the point cloud input to this port will be recognized and their poses will be output

List of the recognized object poses

```
PoseList:
Size:28
[
   [
      -0.34953120902813584,
      0.05297042536907494,
      2.291140832422405,
      0.205069175207676,
      -0.9681528787297967,
      -0.1245374142220001,
      -0.07153369308059786
   ],
   [
      -0.07064852630625612,
      -0.09951935864766029,
      2.350663969138809,
      -0.08472144200719658,
      -0.8575308205948315,
      -0.45269784164729576,
      -0.22918951363894505
   ],
```

Prerequisites

The prerequisites for using this Step are as follows.

Requirement of Graphics Card

To execute this Step successfully, it is recommended to use a graphics card between NVIDIA GTX 1650 Ti and RTX 3060 Ti (including GTX 1650 Ti and RTX 3060 Ti).

Requirement of Driver

To execute this Step, the minimum version requirement for the graphics driver is 472.50.

Obtain the Deep Learning Model File

Use Mech-DLK Sim2Pick to train deep learning model. For more specific operation, please refer to Use Mech-DLK Sim2Pick to Train Model.

After obtaining deep learning model, please refer to Use Deep Learning Model for the model using “Predict Object Poses (Sim2Pick)” Step.

Parameters

Server

Server IP

Description: This parameter is used to specify the IP address of the deep learning server. Keep the default parameter settings.

Default value: 127.0.0.1

Server Port (1–65535)

Description: This parameter is used to specify the port number of the deep learning server.

Default value: 60101

Tuning recommendation: All port numbers are valid, with a maximum value of 65535.

> After opening the project, please wait for the deep learning server to start. If the deep learning server is started successfully, a message saying that **Deep learning server started successfully at xxx** will appear in the log panel, and then you can run the project.

Inference Setting

Inference Mode

Description: This parameter is used to select the inference mode for deep learning. Value list: GPU, CPU

- GPU: Use GPU for deep learning model inference after optimizing the model. The inference speed is relatively fast. It takes about 10 to 30 minutes to optimize the model for the first time.
- CPU: Use CPU for deep learning model inference, which will increase the inference time and reduce the recognition accuracy compared with GPU.

Default value: GPU

Tuning recommendation: GPU inference is faster than CPU. Please restart the deep learning server after switching the inference mode.

Model Configuration

Deep Learning Model File

Description: This parameter is used to specify the path (vis_xxx\resource\models) where the deep learning model file is stored. The model file is in the ONNX format.

> - After modifying the **Deep Learning Model File**, you need to restart the deep learning server to make the parameter modification take effect.
> - When you execute this Step for the first time, it takes some time to load the model, and the time depends on the performance of the GPU of your PC. Please wait for a while.

Pose Optimization

This Step provides two methods to optimize poses:

- Adjust poses based on **Geometric Center File**: Based on the selected geometric center file, adjust the object poses output by deep learning to more pickable poses.
- **Pose Correction**: Based on the selected object point cloud file, further optimize the object poses for precise correction.

Geometric Center File

Description: This parameter is used to specify the path where the geometric center file is stored.

Pose Correction

Description: This parameter is used to correct the initially calculated poses to more accurate poses.

Default value: Unselected.

Tuning recommendation: Once this parameter is selected, you need set the path for the **Object Ply File**. Since the point count of the cloud will affect the pose correction, it is recommended to input a PLY file with fewer points. When there is no “3D Matching” related Steps in the subsequent Steps, it is recommended to enable this feature.

Object Ply File

Description: This parameter is used to specify the path where the object point cloud file is stored. Point cloud files in PLY or STL format are supported.

Object Clustering

Three cluster algorithms, **MeanShift**, **RegionGrowing**, and **Dbscan** are provided. The speeds for each algorithm are listed in the table below.

Algorithm	Speed
MeanShift	Low
RegionGrowing	High
Dbscan	Medium

When the **Inference Mode** is set to **CPU**, **RegionGrowing** cannot be used.

Cluster Method

Description: This parameter is used to select a cluster algorithm.

Value list: MeanShift, RegionGrowing, Dbscan

When the cluster method is **MeanShift**, you can adjust the following parameters.

Cluster Bandwidth

Description: This parameter is used to define the maximum distance between two points that can be grouped into the same cluster. The larger the value, the more distant points can be grouped into one cluster. When this parameter is relatively large, points that belong to two or more objects may fall into the same cluster; while this parameter is relatively small, points belong to the same object may be divided into several clusters.

Default value: 10

Value range: 1–100

Minimum value: 2

Tuning recommendation: Please adjust the parameter according to the actual situation.

Min Point Count per Cluster

Description: This parameter is used to set the lower limit of points in a cluster. When the number of points in a cluster is less than the lower limit, the cluster will be ignored.

Value range: 1–50

Default value: 30

Tuning recommendation: Please adjust the parameter according to the actual situation.

When the cluster method is **RegionGrowing**, you can adjust the following parameters.

Min Point Count per Cluster

Description: This parameter is used to set the lower limit of points in a cluster. When the number of points in a cluster is less than the lower limit, the cluster will be ignored.

Default value: 30

Value range: 1–50

Tuning recommendation: Please adjust the parameter according to the actual situation.

Cluster Radius

Description: This parameter is used to define the maximum distance between two points that can be grouped into the same cluster. The larger the value, the more distant points can be grouped into one cluster. When this parameter is relatively large, points that belong to two or more objects may fall into the same cluster; while this parameter is relatively small, points belong to the same object may be divided into several clusters.

Default value: 0.5 mm

Value range: 0.1–20

Default value: 0.4 mm

Tuning recommendation: Please adjust the parameter according to the actual situation.

Number of Neighbours

Description: This parameter is used to determine the number of neighboring points around a specified point within the Cluster Radius, and the neighboring points and that point will be grouped into the same cluster.

Default value: 30

Value range: 1–200

Minimum value: 10

Tuning recommendation: Please adjust the parameter according to the actual situation.

When the cluster method is **Dbscan**, you can adjust the following parameters.

Cluster Bandwidth

Description: This parameter is used to define the maximum distance between two points that can be grouped into the same cluster. The larger the value, the more distant points can be grouped into one cluster. When this parameter is relatively large, points that belong to two or more objects may fall into the same cluster; while this parameter is relatively small, points belong to the same object may be divided into several clusters.

Default value: 10

Value range: 1–100

Minimum value: 2

Tuning recommendation: Please adjust the parameter according to the actual situation.

Min Point Count per Cluster

Description: This parameter is used to set the lower limit of points in a cluster. When the number of points in a cluster is less than the lower limit, the cluster will be ignored.

Value range: 1–50

Default value: 30

Tuning recommendation: Please adjust the parameter according to the actual situation.

Visualization

Enable

Description: This parameter facilitates the visualization of object instances, with each distinct object being represented by a unique color.

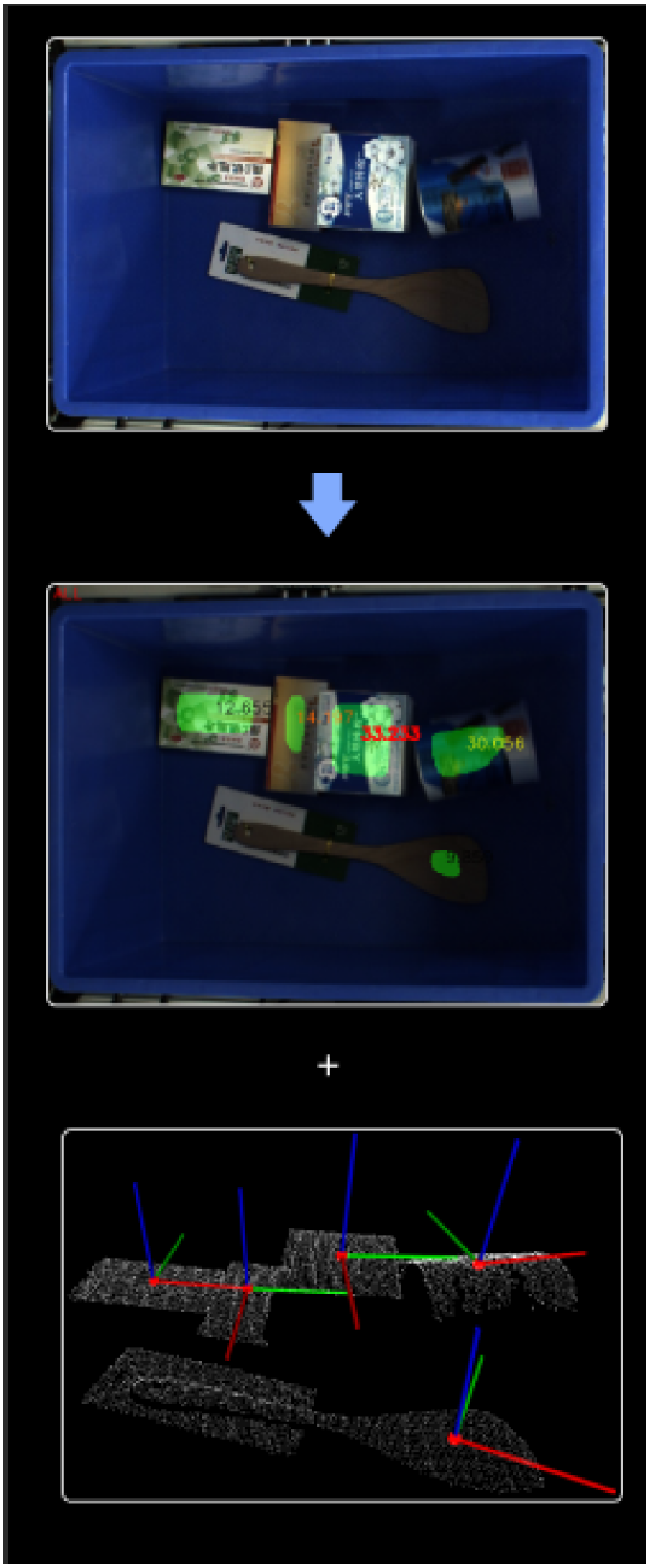
Default value: Unselected.

Tuning recommendation: Please set the parameter according to the actual requirement.

4.3.281. Predict Pick Points (Any Object)

Function

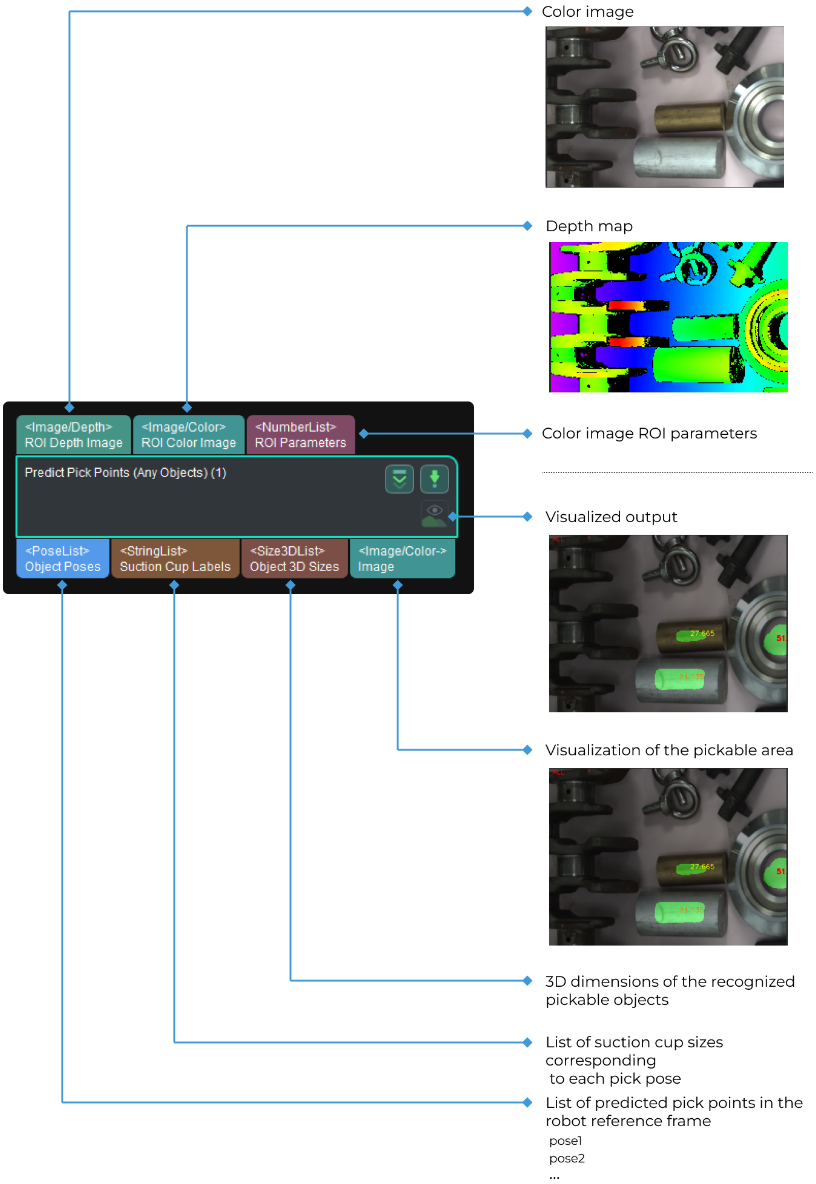
Based on 2D images and depth maps, this Step recognizes the pickable objects in the image and outputs corresponding picking poses.


12.655
14.197
33.233
30.056

Usage Scenario

This Step is usually used for sorting different objects heaped or scattered.

Input and Output


Color image
Depth map
<Image/Depth>
ROI Depth Image
<Image/Color>
ROI Color Image
<NumberList>
ROI Parameters
Color image ROI parameters
Predict Pick Points (Any Objects) (1)
Visualized output
<PoseList>
Object Poses
<StringList>
Suction Cup Labels
<Size3DList>
Object 3D Sizes
<Image/Color>
Image
Visualization of the pickable area
3D dimensions of the recognized pickable objects
List of suction cup sizes corresponding to each pick pose
List of predicted pick points in the robot reference frame
pose1
pose2
...

Requirement of Graphics Card

This Step requires a graphics card of NVIDIA GTX 1650 Ti or higher to be used.

Parameter Description

Server

Server IP

Default value: 127.0.0.1

Instruction: Please set the IP address of deep learning server according to the actual requirement.

Server Port (1–65535)

Default value: 60052

Instruction: For Pick Anything and Pick Anything (Without Bin) projects, you should use a port number of 60000 or above.

Working Distance

Minimum Working Distance

Default value: 0 mm

Instruction: The shortest distance between the camera and the object(s) in the scene. If a bin is used, please set this value to the distance from the bottom of the camera to the top edge of the bin. Please configure the value according to the actual situation.

Maximum Working Distance

Default value: 3000 mm

Instruction: The longest distance between the camera and the object(s) in the scene. If a bin is used, please set this value to the distance from the bottom of the camera to the bottom edge of the bin. Please configure the value according to the actual situation.

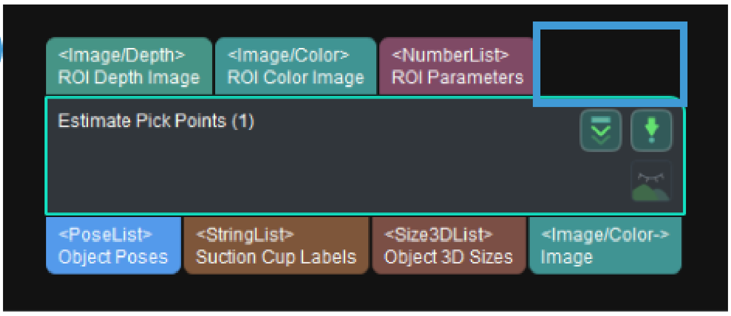
Bin Settings

Use Bin

Default setting: Unselected.

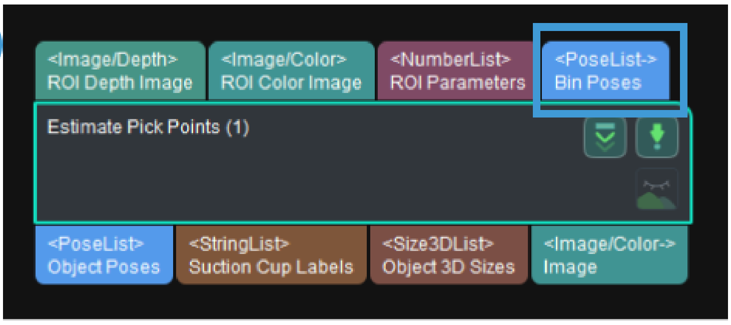
Instruction: Please select this option when a bin is used. Once this option is selected, a Bin Poses port will be added. The first and second figures below show the situation before and after the selection, respectively.

1



<Image/Depth> ROI Depth Image
<Image/Color> ROI Color Image
<NumberList> ROI Parameters
Estimate Pick Points (1)
<PoseList> Object Poses
<StringList> Suction Cup Labels
<Size3DList> Object 3D Sizes
<Image/Color-> Image

2



<Image/Depth> ROI Depth Image
<Image/Color> ROI Color Image
<NumberList> ROI Parameters
<PoseList-> Bin Poses
Estimate Pick Points (1)
<PoseList> Object Poses
<StringList> Suction Cup Labels
<Size3DList> Object 3D Sizes
<Image/Color-> Image

Length

Default setting: 100 mm

Instruction: Once **Use Bin** is selected, set the value according to the dimensions of the bin.

Width

Default setting: 100 mm

Instruction: Once **Use Bin** is selected, set the value according to the dimensions of the bin.

Contour Detection (Smart Placement, Special Shape)

Contour Detection

Default setting: Unselected.

Instruction: Select to detect the contour of the object. Once this option is selected, the **Special Shape Filter** parameter will appear.

Special Shape Filter (frequently used for parcel picking scenarios)

Filter Out Narrow Objects

Default setting: Unselected.

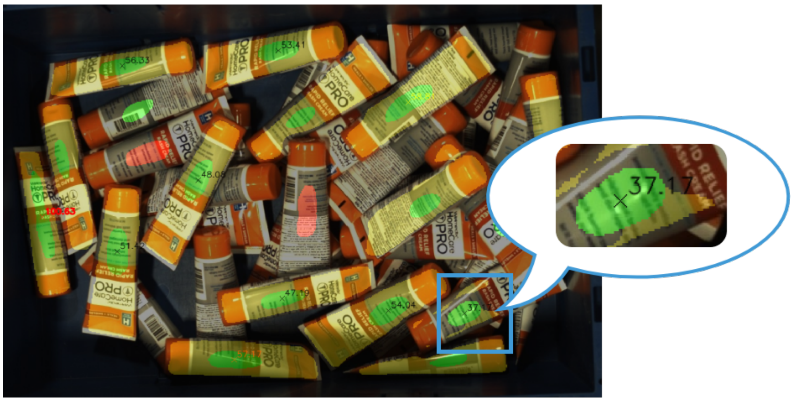
Instruction: Once this option is selected, object whose smallest cuboid bounding box’s shortest edge is below the set threshold will be considered as the narrow object and therefore ignored.

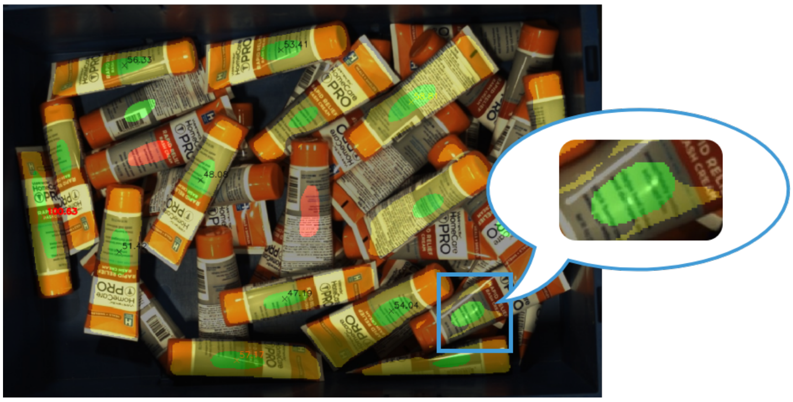
Short Edge Length Lower Limit

Default value: 0 mm

Instruction: This option will appear once the **Filter Out Narrow Objects** is selected. Object whose smallest cuboid bounding box’s shortest edge is below this value will be considered as the narrow object and therefore ignored. Please configure the value according to the actual situation.

Example: When the value is set to 30, objects whose actual lengths are less than 30 mm will be filtered, as shown below. The top and bottom figures show the recognition result before and after filtering, respectively.





Filter Out Long Objects

Default setting: Unselected.

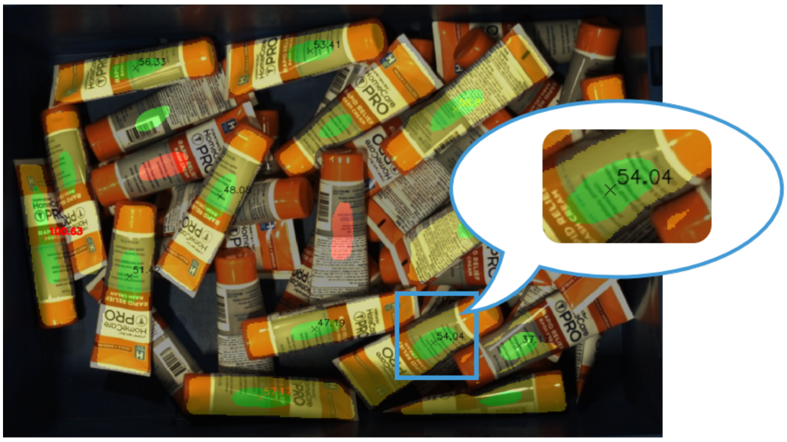
Instruction: Once this option is selected, object whose smallest cuboid bounding box’s longest edge is higher than the set threshold will be considered as the long object and therefore ignored.

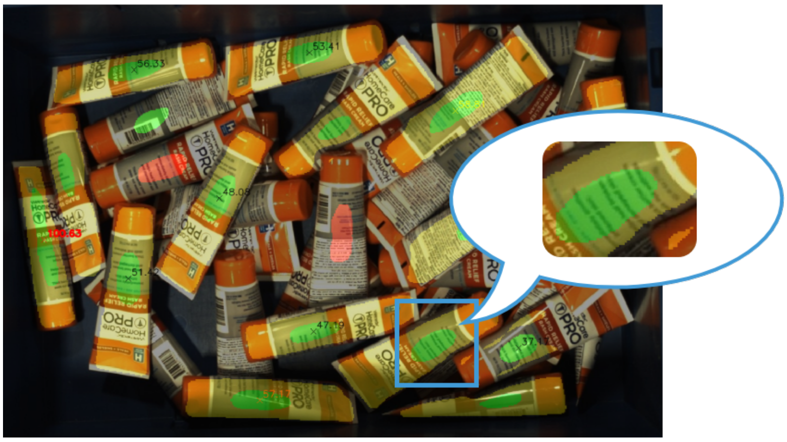
Long Edge Length Upper Limit

Default value: 0 mm

Instruction: This option will appear once the **Filter Out Long Objects** is selected. Object whose smallest cuboid bounding box’s longest edge is higher than this value will be considered as the long object and therefore ignored. Please configure the value according to the actual situation.

Example: When the value is set to 180, objects whose actual lengths are longer than 180 mm will be filtered, as shown below. The top and bottom figures show the recognition result before and after filtering, respectively.





Overlap Detection

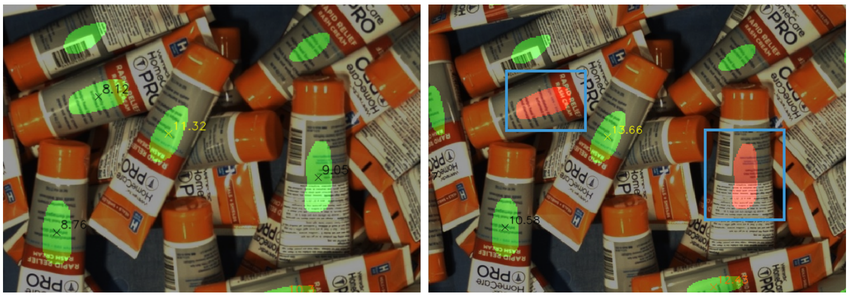
Overlap Detection
: Default setting: Unselected.

Instruction: Please select this option when there are overlapping objects. Once this option is selected, the overlapping objects will be detected and their picking priority will be reduced.

Max Number of Objects
: Default value: 6

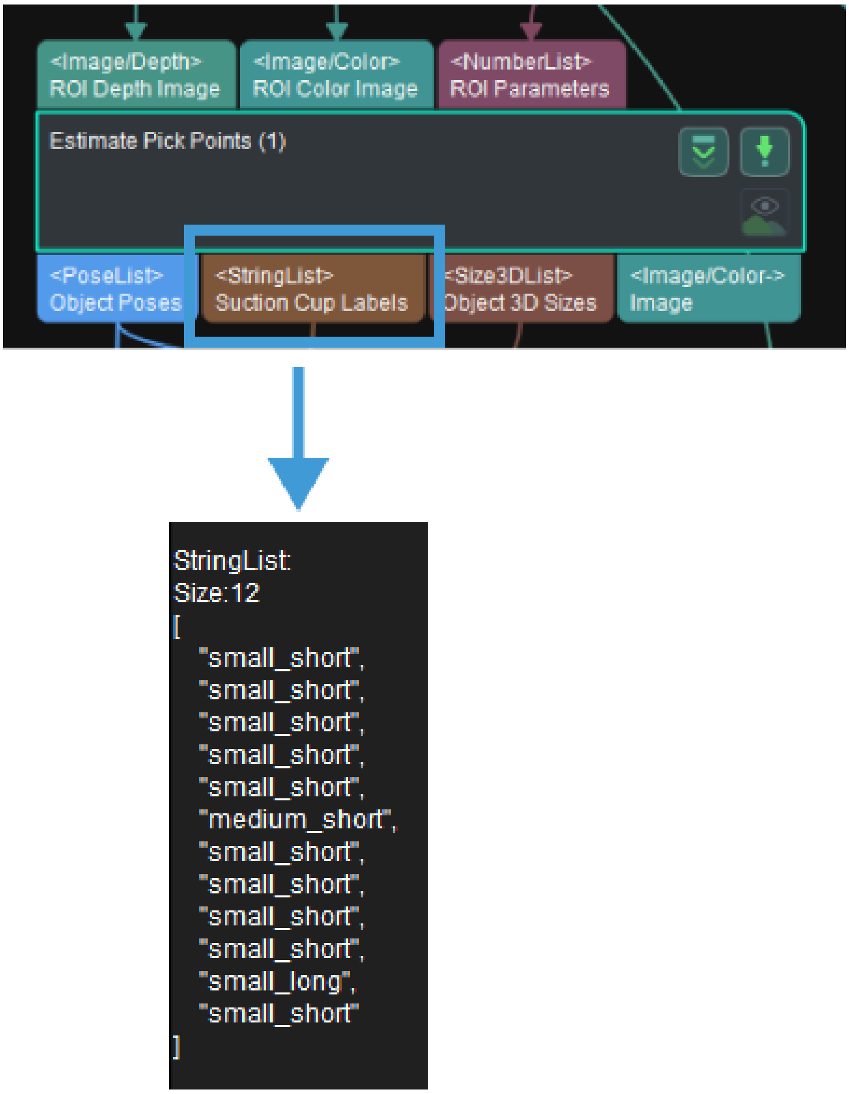
Instruction: This option will appear once **Overlap Detection** is selected. It defines the maximum number of objects which are considered overlapping objects. Please configure the value according to the actual situation. The higher this value, the more objects will be considered overlapping, and therefore it will be harder to complete the picking task. However, the target objects are less likely to be damaged.

Example: When the value is set to 6, no more than 6 overlapping objects will be detected, as shown below. The left and right figure shows the recognition results before and after setting the value. The objects in red are overlapping objects.



Suction Cup Configuration

This parameter determines the output result of suction cup labels.



```
StringList:
Size:12
[
  "small_short",
  "small_short",
  "small_short",
  "small_short",
  "small_short",
  "medium_short",
  "small_short",
  "small_short",
  "small_short",
  "small_short",
  "small_long",
  "small_short"
]
```

Allocate by Mask Heftiness

Default setting: Disable.

Options: Disable, Into Two Groups, and Into Three Groups.

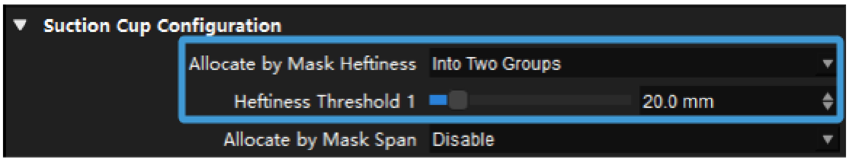
Instruction: The objects will be allocated to different groups according to the smallest inscribed circle’s radii of their masks, and the suction cup configuration of different groups will be different.

Heftiness Threshold 1

Default setting: 0.0 mm

Instruction: The first threshold of the smallest inscribed circle’s radii. Objects with radii below this threshold will be allocated to group 1, while objects with radii above this threshold will be allocated to group 2.

Example: As shown in the figure below, when the detected inscribed circle’s radius of the object mask is below 20 mm, the suction cup’s label will be labeled as “Small.”



▼ Suction Cup Configuration

Allocate by Mask Heftiness: Into Two Groups

Heftiness Threshold 1: 20.0 mm

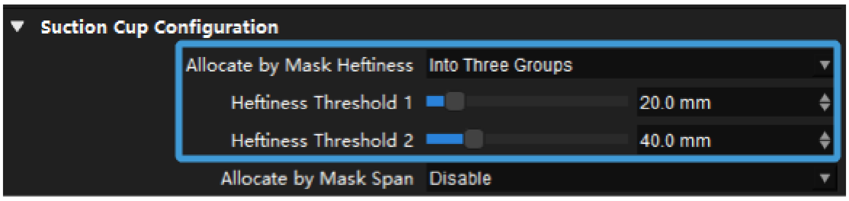
Allocate by Mask Span: Disable

Heftiness Threshold 2

Default setting: 0.0 mm

Instruction: The second threshold of the smallest inscribed circle’s radii. Objects with radii below this threshold will be allocated to group 2, while objects with radii above this threshold will be allocated to group 3.

Example: As shown in the figure below, when the detected inscribed circle’s radius of the object mask is greater than 40 mm, the suction cup’s label will be labeled as “Large”, and if the radius is between 20 mm and 40 mm, the label will be labeled as “Medium.”



▼ Suction Cup Configuration

Allocate by Mask Heftiness: Into Three Groups

Heftiness Threshold 1: 20.0 mm

Heftiness Threshold 2: 40.0 mm

Allocate by Mask Span: Disable

Allocate by Mask Span

Default setting: Disable.

Options: Disable and Into Two Groups.

Instruction: The objects will be allocated to different groups according to the smallest bounding box’s diagonal of their masks, and the suction cup configuration of different groups will be different.

Mask Span Threshold

Default value: 80

Instruction: Object(s) with diagonals below this threshold will be marked as “Short”, or else will be marked as “Long.”

Example: As shown in the figure below, the detected object will be labeled as “Long” when its length is greater than 80 mm, or else it will be labeled as “Short.”

[grasp pose estimation length paragraph 2] | *grasp-pose-estimation-length_paragraph_2.png*

Visualization

Enable

Default setting: Selected.

Instruction: Select to enable visualization.

Visualization Type

Default setting: Final Score.

Options: Final Score, Show Suction Cup Diameter, Show Object Length, and Show Pose Height.

Instruction: Select the options for visualization.

Example:

- When Final Score is selected as the Visualization Type, the visualization result is shown below. The object with red text on it will be picked first.

[grasp pose estimation open visualization 1] | *grasp-pose-estimation-open_visualization_1.png*

- When Show Suction Cup Diameter is selected as the Visualization Type, the visualization result is shown below. The object with red text on it will be picked first.

[grasp pose estimation open visualization 2] | *grasp-pose-estimation-open_visualization_2.png*

- When Show Object Length is selected as the Visualization Type, the visualization result is shown below. The object with red text on it will be picked first.

[grasp pose estimation open visualization 3] | *grasp-pose-estimation-open_visualization_3.png*

- When Show Pose Height is selected as the Visualization Type, the visualization result is shown below. The object with red text on it will be picked first.

[grasp pose estimation open visualization 4] | *grasp-pose-estimation-open_visualization_4.png*

Pose Sorting Logic

Pose Height Weight

Default setting: 3

Options: 1, 2, 3

Suction Cup Size Weight

Default setting: 1

Options: 0, 1, 2, 3

Object Length Weight

Default setting: 1

Options: 0, 1, 2, 3

Instruction: When the Pose Height Weight, Suction Cup Size Weight, and Object Length Weight are specified, the final score is calculated according to the equation: Final Score = Pose Height

× Pose Height Weight + Suction Cup Size × Suction Cup Size Weight + Object Length × Object Length Weight, and objects with higher final score will be prioritized for picking.

4.3.282. Predict Pick Points V2

Function

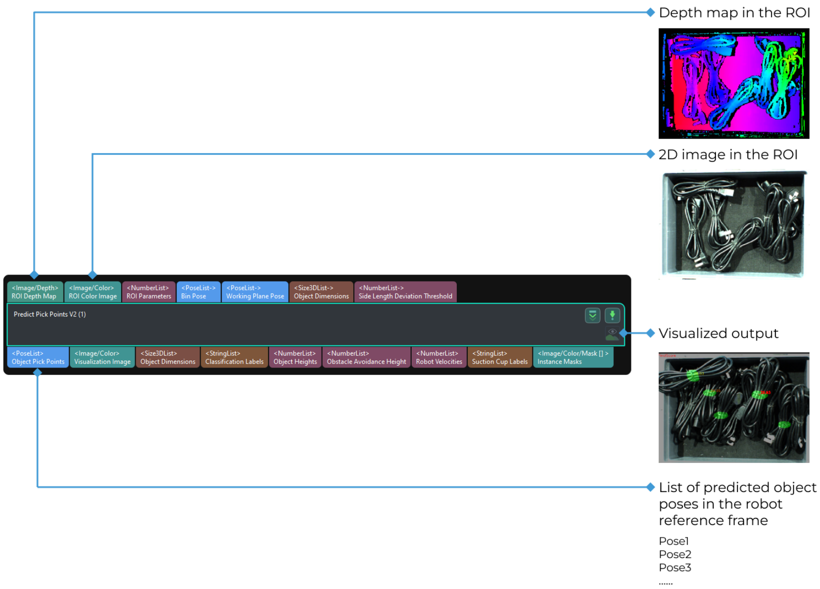
This Step recognizes the pickable objects based on the 2D images and depth maps and outputs the corresponding pick points.

- Before using this Step, please wait for the deep learning server to start. If the deep learning server is started successfully, a message saying that Deep learning server started successfully at xxx will appear in the log panel, and then you can run the Step.
- When you run the Step for the first time, you should load a **Picking Configuration Folder**. It takes some time to load the file. Please wait with patience.

Usage Scenario

This Step is designed for piece picking in logistics, supermarket, and cables industry. This Step follows the Scale Image in 2D ROI Step to obtain the information of the scaled depth map, point cloud, and ROI.

Input and Output


Depth map in the ROI
2D image in the ROI
<Image/Depth> ROI Depth Map
<Image/Color> ROI Color Image
<NumberList> ROI Parameters
<PoseList-> Bin Pose
<PoseList-> Working Plane Pose
<Size3DList-> Object Dimensions
<NumberList-> Side Length Deviation Threshold
Predict Pick Points V2 (1)
<PoseList> Object Pick Points
<Image/Color> Visualization Image
<Size3DList> Object Dimensions
<StringList> Classification Labels
<NumberList> Object Heights
<NumberList> Obstacle Avoidance Height
<NumberList> Robot Velocities
<StringList> Suction Cup Labels
<Image/Color/Mask [] > Instance Masks
Visualized output
List of predicted object poses in the robot reference frame
Pose1
Pose2
Pose3
......

Requirement of Graphics Card

This Step requires a graphics card of NVIDIA GTX 1650 Ti or higher to be used.

Parameter Description

Server

Server IP

Description: This parameter is used to set the IP address of the deep learning server.

Default value: 127.0.0.1

Tuning recommendation: Please set the parameter according to the actual requirement.

Server Port (1–65535)

Description: This parameter is used to set the port number of the deep learning server.

Default value: 60054

Value range: 60000–65535

Tuning recommendation: Please set the parameter according to the actual requirement.

Inference Setting

Inference Mode

Description: This parameter is used to select the inference mode for deep learning.

Value list: GPU (optimization), GPU (default), CPU

- GPU (default): Use GPU for deep learning model inference without optimizing the model, and the model inference will not be accelerated.
- GPU (optimization): Use GPU for deep learning model inference after optimizing the model. The inference speed is relatively fast. It takes about 10 to 30 minutes to optimize the model for the first time.
- CPU: Use CPU for deep learning model inference, which will increase the inference time and reduce the recognition accuracy compared with GPU.

Default value: GPU (default)

Tuning recommendation: Inference speed: GPU (optimization)>GPU (default)>CPU. Please restart the deep learning server after switching the inference mode.

Picking Configuration

Picking Configuration Folder Path

Description: This parameter is used to select the path where the picking configuration folder is stored.

Tuning recommendation: Before you run the project, please load the **Picking Configuration Folder** first. Four kinds of Picking Configuration Folders used in the scenarios including logistics, supermarket, cables and medicine boxes are provided in the following table.

Usage Scenario	Picking Configuration Folder Name	Request
Logistics	Logistics_Seg_RGBSuction	Contact Mech-Mind Technical Support to request.
Supermarket	Supermarket_Seg_RGBSuction	Contact Mech-Mind Technical Support to request.
Cables	Cable_Seg_RGBGrasp	Contact Mech-Mind Technical Support to request.

Usage Scenario	Picking Configuration Folder Name	Request
Medicine Boxes	MedicineBox_Instance_3DSize_RGB Suction	The Picking Configuration Folder can be downloaded along with the solutions in the solution library.

There are two JSON files and one model folder in the picking configuration folder. The deep learning model is stored in the model folder. The folder path should NOT contain the model folder, or else this Step cannot function properly.

For example, a correct path can be D:/ConfigurationFiles/Cable_Seg_RGBGrasp.

If you are not sure about which type of deep learning model you should use, you can consult Mech-Mind Technical Support for some advice.

Logistics

Please refer to Parameter Adjustment in the Logistics Scenario.

Supermarket

Please refer to Parameter Adjustment in the Supermarket Scenario.

Cables

Please refer to Parameter Adjustment in the Cables Scenario.

Medicine Boxes

Please refer to Parameter Adjustment in the Medicine Boxes Scenario.

It is recommended to use a GeForce GTX 10 Series graphics card with a memory of at least or above 4G when you use the model for the above scenarios. When you run this Step for the first time, the deep learning model will be optimized according to the hardware type and the one-time optimization process takes about 15 to 35 minutes.

4.3.282.1. Parameter Adjustment in the Logistics Scenario

In the logistics (semantic segmentation) scenario, the **Logistics_Seg_RGBSuction** picking configuration folder should be used. This model can be used to obtain all pixels of the object in the image, and therefore obtain the contour and status (overlapped or not) of the object.

Parameter Adjustment Order

1. Set a **Working Distance** first to limit the range for the camera to capture the data.
2. Adjust the **Contour Detection (Smart Placement, Special Shape)** parameter to determine whether to process objects whose contours are detected with priority.
3. The objects in the logistics scenario are usually stacked randomly. The **Overlap Detection** can divide objects into overlapped objects and non-overlapped objects.
4. Adjust the **Pose Processing** parameter group to control the upward offset along the Z-axis to avoid damaging the object.
5. Adjust the **Dynamic Place** parameter group to adjust the speed of the robot.
6. Adjust the **Priority Picking Setting** and **Pose Sorting Logic** parameter group to determine the picking sequence.

Parameter Description

Working Distance

Minimum Working Distance

Description: This parameter is used to specify the minimum distance (in millimeters) between the camera and the object in the scene. If a bin is used, this parameter should be the distance from the camera to the top edge of the bin. Please set the value according to the actual situation.

Default value: 0 mm

Value range: 0–3000 mm

Maximum Working Distance

Description: This parameter is used to specify the maximum distance (in millimeters) between the camera and the object in the scene. If a bin is used, this parameter should be the distance from the camera to the bottom edge of the bin. Please set the value according to the actual situation.

Default value: 3000 mm

Value range: 0–3000 mm

Contour Detection (Smart Placement, Special Shape)

Overlap Detection

Box Object Rectangularity Threshold

Description: This parameter is used to specify the rectangularity threshold for boxes. When the rectangularity of the box is lower than this threshold, the box will be considered to be overlapped, while a rectangularity higher than the threshold will be considered not overlapped.

Default value: 10%

Recommended value: 60%

Value range: 0–100%

Picking Area Ratio Threshold for Boxes

The picking area refers to the region on the object’s surface predicted by deep learning to be suitable for picking. For box-shaped objects, if the ratio of the picking area to the object’s surface area in the camera’s field of view exceeds this threshold, the object is considered not overlapped. Otherwise, the object is considered overlapped.

Default value: 10%

Recommended value: 60%

Value range: 0–100%

Envelope Object Rectangularity Threshold

Description: This parameter is used to specify the rectangularity threshold for envelope-shaped objects. When the rectangularity of the envelope-shaped object is lower than this threshold, the object will be considered to be overlapped, while a rectangularity higher than the threshold will be considered not overlapped.

Default value: 10%

Recommended value: 75%

Value range: 0–100%

Picking Area Ratio Threshold for Envelopes

The picking area refers to the region on the object’s surface predicted by deep learning to be suitable for picking. For envelope-shaped objects, if the ratio of the picking area to the object’s surface area in the camera’s field of view exceeds this threshold, the object is considered not overlapped. Otherwise, the object is considered overlapped.

Default value: 10%

Recommended value: 75%

Value range: 0–100%

Bag-Shaped Object Rectangularity Threshold

Description: This parameter is used to specify the rectangularity threshold for bag-shaped objects. When the rectangularity of the bag-shaped object is lower than this threshold, the object will be considered to be overlapped, while a rectangularity higher than the threshold will be considered not overlapped.

Default value: 10%

Recommended value: 70%

Value range: 0–100%

Picking Area Ratio Threshold for Bags

The picking area refers to the region on the object’s surface predicted by deep learning to be suitable for picking. For bag-shaped objects, if the ratio of the picking area to the object’s surface area in the camera’s field of view exceeds this threshold, the object is considered not overlapped. Otherwise, the object is considered overlapped.

Default value: 10%

Recommended value: 70%

Value range: 0–100%

Pose Processing

Max Number of Output Poses

Description: This parameter is used to specify the maximum number of output poses.

Value range: 0–20

Upward Offset along Z-Axis

Description: This parameter is used to specify the offset distance (in millimeters) along Z-axis above the pick point to prevent the robot from squeezing the object. The unit is millimeters.

Default value: 0

Value range: 0–10 mm

Tuning recommendation: Please set the parameter according to the actual situation.

Point Cloud Downsampling

Point Cloud Downsampling

Description: This parameter is used to reduce the number of object point clouds and increase the processing speed of the Step. The larger the value, the fewer object point clouds, and the faster the Step processing speed.

Default value: 0.0010

Dynamic Place

This parameter group can be used to determine the height of the robot end tool when the robot places the object according to the height of the object, and therefore the following two situations can be avoided.

- If the robot drops the object at a very high height, the object may roll on the fast-running conveyor belt.
- If the robot places the object at a very low height, the object may collide with the conveyor belt, and both the object and the conveyor belt may be damaged.

When the robot picks the object, it needs to determine the weight of the object. The algorithm in this parameter estimates the object’s weight according to its volume. If the robot picks a heavy object, and the robot is moving at a fast speed, the following consequences may occur:

- The object cannot be picked successfully.
- The picked object is thrown out.
- The package of the object is torn, but the object is not picked up successfully.

2D Area Threshold

Description: This parameter is used to detect the area of the detected object. If the area of the detected object is larger than this value, the robot speed will be reduced when placing the object. When the area of the detected object is smaller than this value, the robot will place the object with the **Robot Placing Velocity**.

Value range: 0–10000000 px

Robot Placing Velocity

Description: This parameter is used to specify the speed when the robot places the object. When the area of the detected object is smaller than the set **2D Area Threshold**, the robot will place the object with the **speed set in this parameter**; when the area of the detected object exceeds the set **2D Area Threshold**, the robot reduces the placement speed.

Default value: 100%

Value range: 0–100%

Priority Picking Setting

Priority Picking Direction

Description: This parameter is used to set the direction in which the object will be prioritized for picking.

Value list: Up, Down, Left, Right

Default setting: Right

Pose Sorting Logic

The detected poses in this Step will be sorted according to the pose score in descending order. Pose Score = Object Height × Pose Height Weight + Object Horizontal Position × Object Horizontal Position Weight + Rectangularity × Rectangularity Weight

Pose Height Weight

Description: This parameter is used to set the weight of the pose height when the score of a pose is calculated. If this weight is set to a larger value, objects that are at a high height will be more likely to be picked first.

Default value: 5

Value range: 0–5

Object Horizontal Position Weight

Description: This parameter is used to set the weight of the horizontal position when the score of a pose is calculated. If this weight is set to a larger value, objects that are close to the end in the **Priority Picking Direction** will be more likely to be picked first.

Default value: 1

Value range: 0–5

Visualization

The **Visualization** parameter group is only used for selecting what will be displayed in the Debug Output window and will not affect the adjustment of the above parameters.

Enable

Description: Once this option is selected, you can select the items you want to visualize in the **Debug Output** window.

Default value: selected.

Visualization Options

Description: This parameter is used to specify the items you want to visualize.

Value list: Picking sequence, Pose height, 2D size, Object class, Rectangularity

- Picking sequence: display the picking sequence as numbers in the central region of the pickable objects.
- Pose height: display the object pose height in the central region of the pickable object.
- 2D size: display the number of pixels of the object in the image in the central region of the pickable object.

- Object class: display the class of the object in the central region of the object.
- Rectangularity: display the object’s rectangularity in the central region of the pickable object. Rectangularity = Size of the object/Size of its minimum bounding rectangle

Default value: Picking sequence

This parameter will only be displayed after the visualization is **enabled**.

4.3.282.2. Parameter Adjustment in the Supermarket Scenario

In the supermarket scenario, the **Supermarket_Seg_RGBSuction** picking configuration folder should be used.

It is recommended to use a GeForce GTX 10 Series graphics card with a memory of at least or above 4G when you use the model for the medicine boxes scenario. When you run this Step for the first time, the deep learning model will be optimized according to the hardware type, and the one-time optimization process takes about 15 to 35 minutes.

Parameter Adjustment Order

1. Set a **Working Distance** first to limit the range for the camera to capture the data.
2. Adjust the **Contour Detection (Smart Placement, Special Shape)** parameter to determine whether to process objects whose contours are detected with priority.
3. The objects in the supermarket scenario are usually stacked randomly, adjusting the **Overlap Detection** parameter group can divide the objects into objects that are overlapped and not overlapped.
4. Adjust the **Pose Processing** parameter group to control the upward offset along the Z-axis to avoid damaging the object.
5. Adjust the **Suction Cup Configuration** parameter group to divide the suction cups into sections.
6. Adjust the **Pose Sorting Logic** parameter group to determine the picking sequence.

Parameter Description

Working Distance

Minimum Working Distance

Description: This parameter is used to specify the minimum distance (in millimeters) between the camera and the object in the scene. If a bin is used, this parameter should be the distance from the camera to the top edge of the bin. Please set the value according to the actual situation.

Default value: 0 mm

Value range: 0–3000 mm

Maximum Working Distance

Description: This parameter is used to specify the maximum distance (in millimeters) between the camera and the object in the scene. If a bin is used, this parameter should be the distance from the camera to the bottom edge of the bin. Please set the value according to the actual situation.

Default value: 3000 mm

Value range: 0–3000 mm

Contour Detection (Smart Placement, Special Shape)

Prioritize Contours

Description: This parameter is used to detect the contours of objects. The more complete the contour of an object is, the higher its priority for picking; the less complete the contour, the lower its priority for picking.

Default setting: selected.

Tuning recommendation: If the project does not have a high requirement for the overlap detection, you can enable this feature to increase the picking success rate and avoid changing bins frequently. If there is a high requirement on the overlap detection and only the objects that are NOT overlapped should be picked, it is recommended to disable this feature.

Overlap Detection

Max Number of Objects

Description: This parameter is used to specify the maximum number of objects that will be considered overlapped.

Default value: 6

Value range: 0–10

Tuning recommendation: Please set the parameter according to the actual situation.

Pose Processing

Upward Offset along Z-Axis

Description: This parameter is used to specify the offset distance (in millimeters) along Z-axis above the object to prevent the robot from squeezing the object .

Default value: 0

Value range: 0–10 mm

Tuning recommendation: Please set the parameter according to the actual situation.

Suction Cup Configuration

Allocate by Mask Heftiness

Description: This parameter is used to allocate objects to different groups according to the heftiness of their masks, and the suction cup configurations for each group are different.

Options:

- Default setting: Disable. Do not allocate objects to multiple groups according to the mask heftiness.
- Into Two Groups: Allocate the objects to two groups according to **Heftiness Threshold 1**.
- Into Three Groups: Allocate the objects to three groups according to **Heftiness Threshold 1**

and **Heftiness Threshold 2**.

Heftiness Threshold 1 / Heftiness Threshold 2

Description: This parameter is used to set the heftiness threshold (in millimeters) for sorting masks.

Default value: 0 mm

Value range: 0–200 mm

- When **Allocate by Mask Heftiness** is set to **Into Two Groups**, you will need to set **Heftiness Threshold 1**. The allocation logic is as follows:
  - If the mask heftiness of the object is less than Heftiness Threshold 1, the object will be put into the first group.
  - If the mask heftiness of the object is greater than Heftiness Threshold 1, the object will be put into the second group.
- When **Allocate by Mask Heftiness** is set to **Into Three Groups**, you will need to set **Heftiness Threshold 1** and **Heftiness Threshold 2**. The **Heftiness Threshold 2** must be greater than **Heftiness Threshold 1**. The allocation logic is as follows:
  - If the mask heftiness of the object is less than Heftiness Threshold 1, the object will be put into the first group.
  - If the mask heftiness of the object is between Heftiness Threshold 1 and Heftiness Threshold 2, the object will be put into the second group.
  - If the mask heftiness of the object is greater than Heftiness Threshold 2, the object will be put into the third group.

Allocate by Mask Span

Description: This parameter is used to allocate objects to different groups according to the span of their masks, and the suction cup configurations for each group are different.

Options:

- Default setting: Disable. Do not allocate objects to multiple groups according to the mask heftiness.
- Into Two Groups: Allocate the objects to two groups according to **Mask Span Threshold**.

Mask Span Threshold

Description: This parameter is used to set the span threshold (in millimeters) for sorting masks.

Default value: 0 mm

Value range: 0–200 mm

When the **Allocate by Mask Span** is set to **Into Two Groups**, you will need to set this parameter. The allocation logic is as follows:

- If the mask span of the object is less than Mask Span Threshold 1, the object will be put into the first group.
- If the mask span of the object is greater than Mask Span Threshold 1, the object will be put into the second group.

Pose Sorting Logic

The detected poses in this Step will be sorted according to the pose score in descending order. Pose Score = Object Height × Pose Height Weight + Suction Cup Size × Suction Cup Size Weight +

Object Length × Object Length Weight

Pose Height Weight

Description: This parameter is used to set the weight of the pose height when the score of a pose is calculated. If this weight is set to a larger value, objects that are at a high height will be more likely to be picked first.

Default value: 3

Value range: 0–5

Suction Cup Size Weight

Description: This parameter is used to set the weight of the suction cup size when the score of a pose is calculated. If this weight is set to a larger value, objects that should be picked with larger suction cups will be more likely to be picked first.

Default value: 1

Value range: 0–5

Object Length Weight

Description: This parameter is used to set the weight of the object length when the score of a pose is calculated. If this weight is set to a larger value, objects with longer lengths will be more likely to be picked first.

Default value: 1

Value range: 0–5

Visualization

The **Visualization** parameter group is only used for selecting what will be displayed in the Debug Output window and will not affect the adjustment of the above parameters.

Enable

Description: Once this option is selected, you can select the items you want to visualize in the **Debug Output** window.

Default value: selected.

Visualization Type

Description: This parameter is used to specify the items you want to visualize.

Options:

- Final Score
- Show Pose Height
- Show Suction Cup Diameter
- Show Object Length

Default value: Final Score

This parameter will only be displayed after the visualization is **enabled**.

4.3.282.3. Parameter Adjustment in the Cables Scenario

In the cables scenario, the **Cable_Seg_RGBGrasp** picking configuration folder should be used, and then you can see all related parameters.

Parameter Adjustment Order

1. Set a **Working Distance** first to limit the range for the camera to capture the data.
2. Adjust the **Pose Processing** parameter group to control the upward offset along the Z-axis to avoid damaging the object.

Parameter Description

Working Distance

Minimum Working Distance

Description: This parameter is used to specify the minimum distance (in millimeters) between the camera and the object in the scene. If a bin is used, this parameter should be the distance from the camera to the top edge of the bin. Please set the value according to the actual situation.

Default value: 0 mm

Value range: 0–3000 mm

Maximum Working Distance

Description: This parameter is used to specify the maximum distance (in millimeters) between the camera and the object in the scene. If a bin is used, this parameter should be the distance from the camera to the bottom edge of the bin. Please set the value according to the actual situation.

Default value: 3000 mm

Value range: 0–3000 mm

Pose Processing

Upward Offset along Z-Axis

Description: This parameter is used to specify the offset distance (in millimeters) along Z-axis above the object to prevent the robot from squeezing the object. The unit is millimeters.

Default value: 0

Value range: 0–10 mm

Tuning recommendation: Please set the parameter according to the actual situation.

Visualization

> The **Visualization** parameter group is only used for selecting what will be displayed in the Debug Output window and will not affect the adjustment of the above parameters.

Enable

Description: Once this option is selected, you can select the items you want to visualize in the **Debug Output** window.

Default value: Selected.

Visualization Type

Description: This parameter is used to specify the items you want to visualize.

Value list:

- Final Score
- Show Pose Height

Default value: Final Score

4.3.282.4. Parameter Adjustment in the Medicine Boxes Scenario

In the medicine boxes scenario, the **MedicineBox_Instance_3DSize_RGBSuction** picking configuration folder should be used. After the configuration folder is loaded, all parameters will appear.

It is recommended to use a GeForce GTX 10 Series graphics card with a memory of at least or above 4G when you use the model for the medicine boxes scenario. When you run this Step for the first time, the deep learning model will be optimized according to the hardware type, and the one-time optimization process takes about 15 to 35 minutes.

Parameter Adjustment Order

1. Select **Picking Configuration Folder Path** first.
2. Set a **Working Distance** to limit the range for the camera to capture the data.
3. Set parameters in the **Bin Settings** parameter group.
4. Adjust the **Pose Processing** parameter group to control the upward offset along the Z-axis to avoid damaging the object.
5. Adjust the **Point Cloud Downsampling** to reduce the number of object point clouds and increase the Step processing speed.
6. Lastly, set the **Plane Segmentation Threshold** to segment the object’s planar point cloud.

Parameter Description

Picking Configuration

Picking Configuration Folder Path

Description: This parameter is used to select the path where the picking configuration folder is stored.

Tuning recommendation: Click the folder button on the right of the parameter and select the path where the picking configuration folder is stored.

Working Distance

Minimum Working Distance

Description: This parameter is used to specify the minimum distance (in millimeters) between the camera and the object in the scene. If a bin is used, this parameter should be the distance from the camera to the top edge of the bin. Please set the value according to the actual situation.

Default value: 0 mm

Value range: 0–3000 mm

Maximum Working Distance

Description: This parameter is used to specify the maximum distance (in millimeters) between the camera and the object in the scene. If a bin is used, this parameter should be the distance from the camera to the bottom edge of the bin. Please set the value according to the actual situation.

Default value: 3000 mm

Value range: 0–3000 mm

Bin Settings

Length

Description: This parameter is used to set the length (mm) of the bin.

Width

Description: This parameter is used to set the width (mm) of the bin.

Pose Processing

Max Number of Output Poses

Description: This parameter is used to specify the maximum number of output poses.

Default value: 15

Upward Offset along Z-Axis

Description: This parameter is used to specify the offset distance (in millimeters) along Z-axis above the object to prevent the robot from squeezing the object. The unit is millimeters.

Default value: 0

Value range: 0–10 mm

Tuning recommendation: Please set the parameter according to the actual situation.

Pose Rotation Angle Threshold

Description: If the rotation angle of the object pose’s X-axis or Y-axis exceeds this threshold, the object pose will be filtered out. (Unit: °)

Default value: 45.000°

Point Cloud Downsampling

Point Cloud Downsampling

Description: This parameter is used to reduce the number of object point clouds and increase the processing speed of the Step. The larger the value, the fewer object point clouds, and the faster the Step processing speed.

Default value: 0.0010

Plane Segmentation

Plane Segmentation Threshold

Description: This parameter is used to segment the planar point cloud of the object. The smaller the value, the fewer planar point clouds after the segmentation, and the higher the segmentation precision. The larger the value, the more planar point clouds after the segmentation, and the lower the segmentation precision.

Default value: 0.0020

Visualization

Enable

Description: This parameter is used to select whether to visualize the output result of this Step.

Default value: selected.

4.3.283. Procedure Out

Function

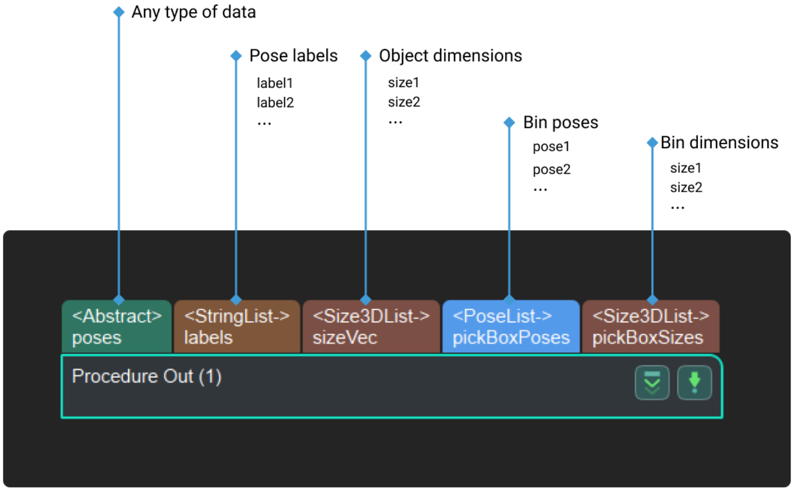
This Step sends the results of the current project to the server.

Usage Scenario

This Step is usually used to send various vision results to Mech-Viz or other software or services that need to use the vision results.

Input and Output

When no Steps related to path planning are involved in the project, the input and output are shown as follows.



Parameter Description

Port Type

Determine the input port type.

Options:

- Custom: The user defines the data types of the project output.
- Predefined (vision result): Default value. It is selected when no Steps related to path planning are involved in the project.
- Predefined (robot path): Selected when Steps related to path planning are involved in the project.

Customized Ports Name

Description: Set the data type output for external communication.

Steps to set this parameter:

1. Click **[Set data type output to external communication]**, and the **Procedure Out Setting Assistant** window will pop up.
2. Add a port in the **Predefined Keys** or **Customized Keys** panel.
   - Predefined Keys: Select the check boxes of ports you need.
   - Customized Keys: Click the blank space of the window, then click **[Insert]**, and double-click to rename the port.

   > The inserted customized key must be renamed, or else the added port is invalid.
3. Click **[OK]** to save the changes.

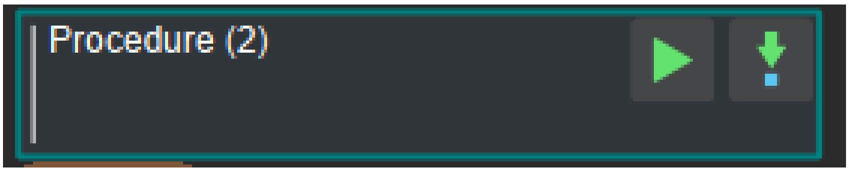
4.3.284. Procedure

Description

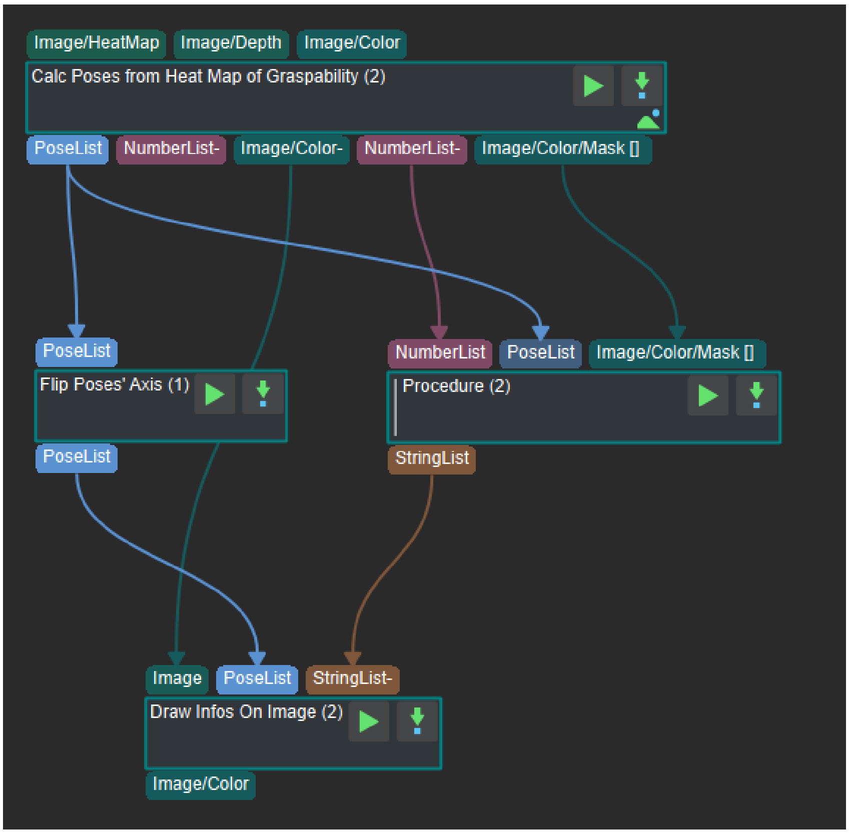
The combination of multiples Steps.

Usage Scenario

Usually, multiple Steps can be combined into a “procedure” to achieve complex functions. By doing so, we can make a project simpler, clearer, and more logical.



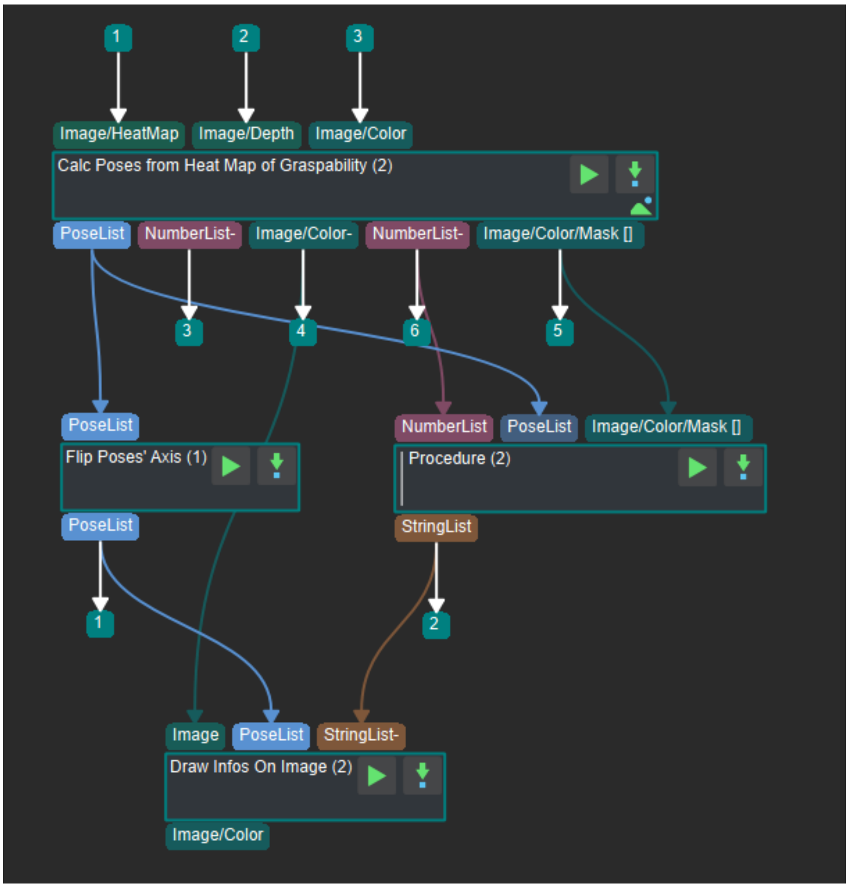
Multiples Steps can be put into a “procedure.” Double-click “procedure” or right-click it and select “Navigate Down” to entre this “procedure.” Add the required Steps into the “procedure” and connect them.

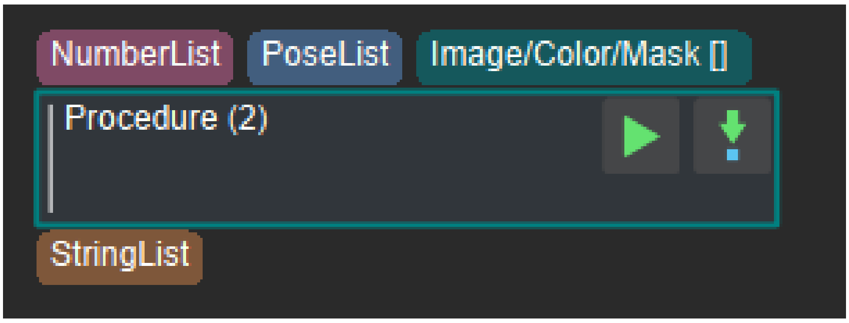


Right-click on the blank and select “Navigate Up” to exit this procedure.

Set the Input and Output of Procedure

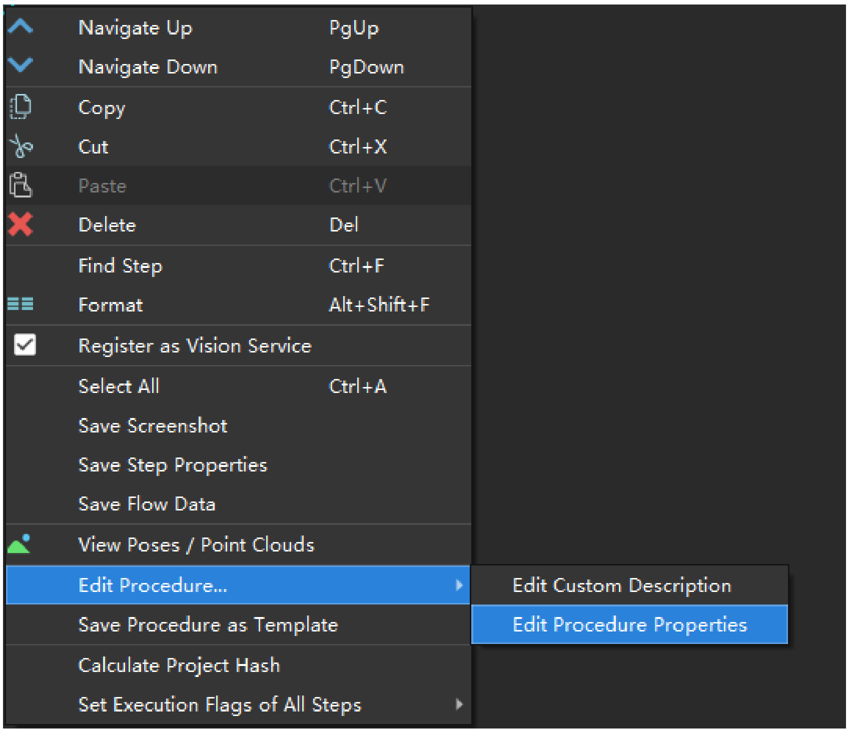
After the “procedure” is created, you need to set the input and output of this procedure. The setting method is as follows: Directly double-click the input or output of the “Step” to use the input or output of the “Step” as the input or output of the “procedure.” The sequence of the input and output depends on the order of the double-click. The setting method and the “procedure” after setting are shown in the figure below.

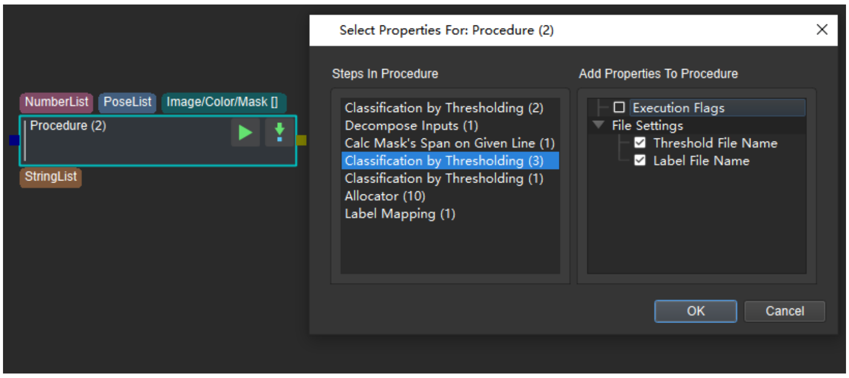




Set the Parameters of Procedure

In order to facilitate the adjustment of the “procedure”, choose the key parameter(s) of each Step in the “procedure” as the attribute of the “procedure.” Right-click the “procedure” and select Edit Procedure Parameters. Select the attributes to be displayed in the opened window and click OK after selecting. The setting steps and effects are shown in the figure below.





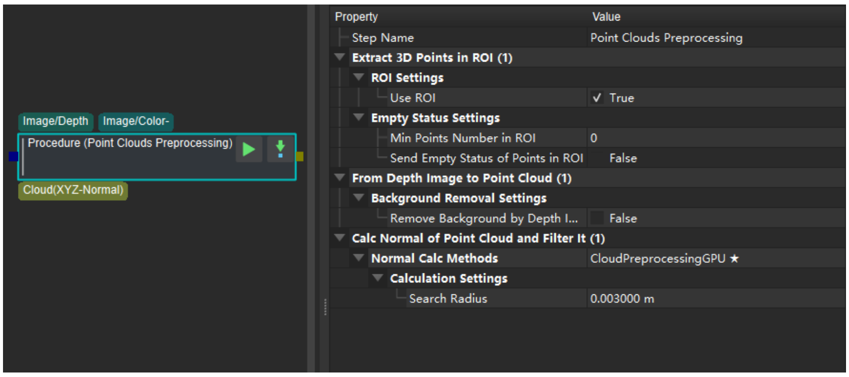
Select Properties For: Procedure (2)

Steps In Procedure

- Classification by Thresholding (2)
- Decompose Inputs (1)
- Calc Mask's Span on Given Line (1)
- Classification by Thresholding (3)
- Classification by Thresholding (1)
- Allocator (10)
- Label Mapping (1)

Add Properties To Procedure

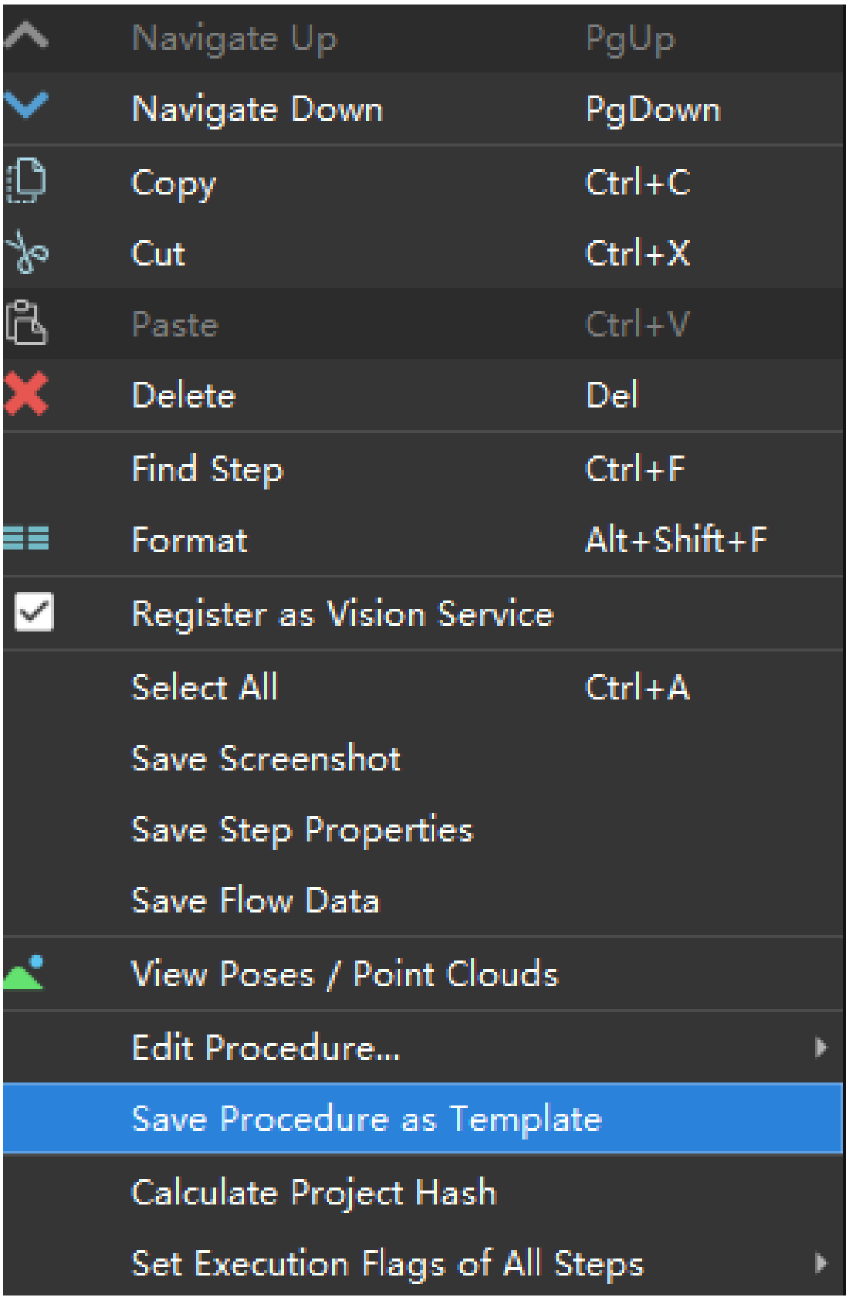
- □ Execution Flags
- File Settings
  - ☑ Threshold File Name
  - ☑ Label File Name

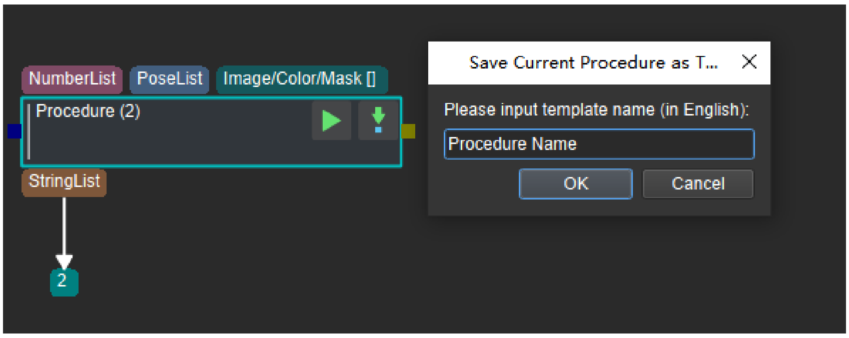


Property	Value
Step Name	Point Clouds Preprocessing
Extract 3D Points in ROI (1)	
ROI Settings	
Use ROI	✓ True
Empty Status Settings	
Min Points Number in ROI	0
Send Empty Status of Points in ROI	False
From Depth Image to Point Cloud (1)	
Background Removal Settings	
Remove Background by Depth I...	False
Calc Normal of Point Cloud and Filter It (1)	
Normal Calc Methods	CloudPreprocessingGPU ★
Calculation Settings	
Search Radius	0.003000 m

Save and Call a Procedure

To reuse the Procedure in the future, you can save it as a template. Right-click it and select Save Procedure as Template. Enter the template name in the pop-up interface. The saved Procedure can be found under the “Custom group” in the “Step Library.”

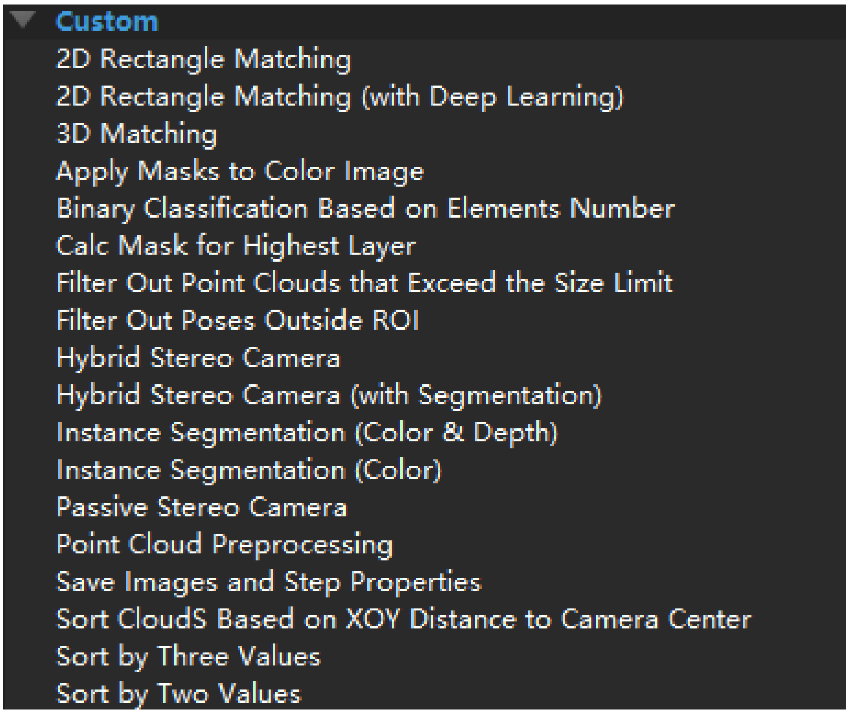




Save Current Procedure as T...

Please input template name (in English):

Procedure Name



- Custom
  - 2D Rectangle Matching
  - 2D Rectangle Matching (with Deep Learning)
  - 3D Matching
  - Apply Masks to Color Image
  - Binary Classification Based on Elements Number
  - Calc Mask for Highest Layer
  - Filter Out Point Clouds that Exceed the Size Limit
  - Filter Out Poses Outside ROI
  - Hybrid Stereo Camera
  - Hybrid Stereo Camera (with Segmentation)
  - Instance Segmentation (Color & Depth)
  - Instance Segmentation (Color)
  - Passive Stereo Camera
  - Point Cloud Preprocessing
  - Save Images and Step Properties
  - Sort CloudS Based on XOY Distance to Camera Center
  - Sort by Three Values
  - Sort by Two Values

4.3.285. Process 2D Shapes

Function

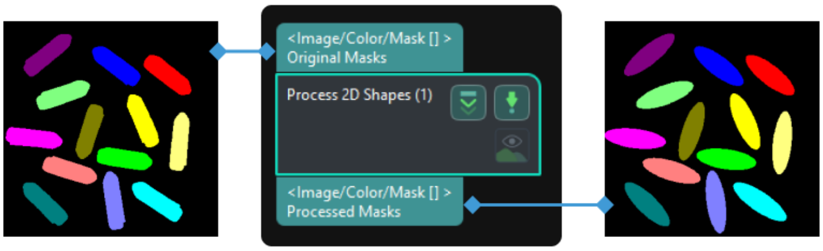
This Step processes the shapes in the input binary images by the specified method.

Usage Scenario

This Step is usually used to process object contours to facilitate various calculations in measurement scenarios.

Input and Output

For example, when **Ellipse** is selected, the input and output of this Step are shown as below.



Parameter Description

Target Shape

Description: This parameter specifies the method to process the 2D image.

Value list: Convex, Ellipse, Circumcircle, Incircle, BoundingBox, CircumscribedRectangle, InscribedRectangle, and InnerCenter.

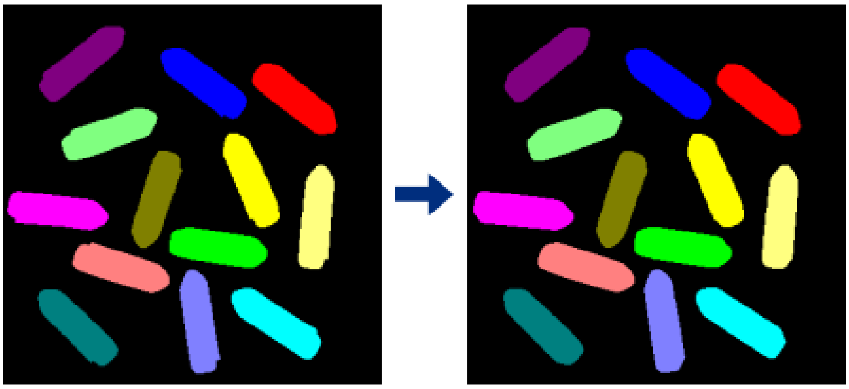
- Convex: Generate a convex hull of the original shape.
- Ellipse: Generate an ellipse with the same first-order moment and second-order moment as the original shape.
- Circumcircle: Generate a minimum circumscribed circle of the original shape.
- Incircle: Generate a maximum inscribed circle of the original shape.
- BoundingBox: Generate a minimum bounding box with edges parallel to the image borders.
- CircumscribedRectangle: Generate a minimal circumscribed rectangle of the original shape.
- InscribedRectangle: Generate a maximum inscribed rectangle with edges parallel to the image borders.
- InnerCenter: Generate a point that is closest to the center of gravity on the original shape's skeleton.

Tuning recommendation: Please select a method according to the actual requirement.

Tuning Examples

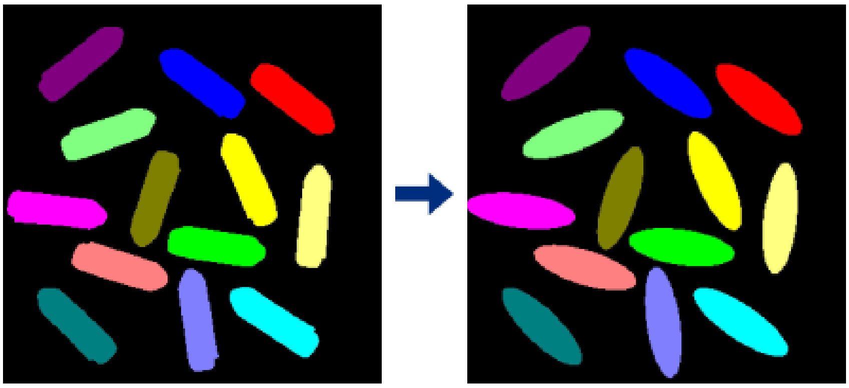
Convex

When **Convex** is selected as the Target Shape, the 2D shape before and after processing is shown below.



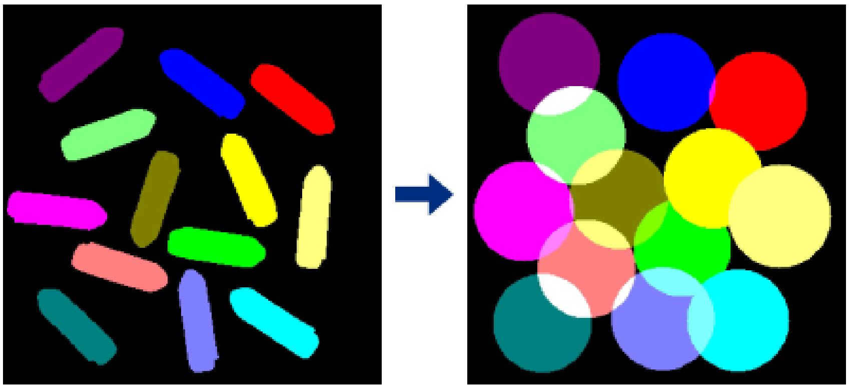
Ellipse

When **Ellipse** is selected as the Target Shape, the 2D shape before and after processing is shown below.



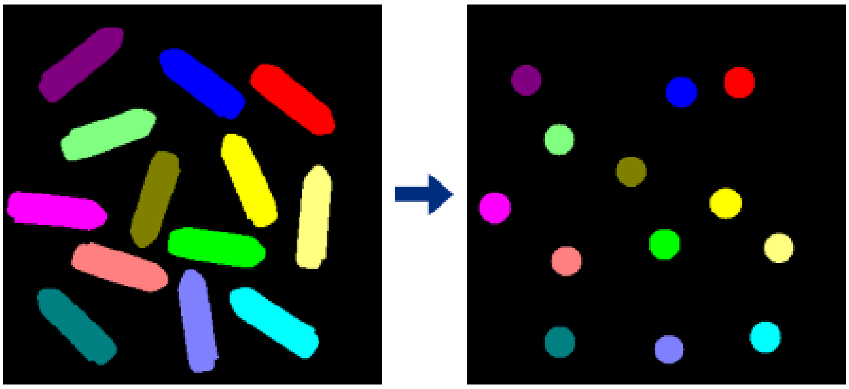
Circumcircle

When **Circumcircle** is selected as the Target Shape, the 2D shape before and after processing is shown below.



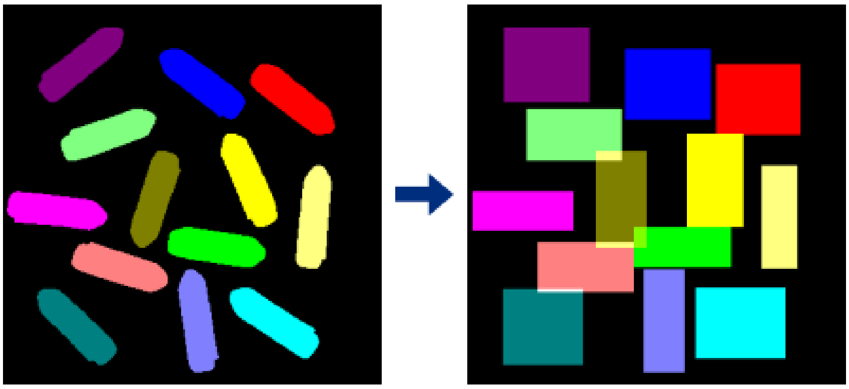
Incircle

When **Incircle** is selected as the Target Shape, the 2D shape before and after processing is shown below.



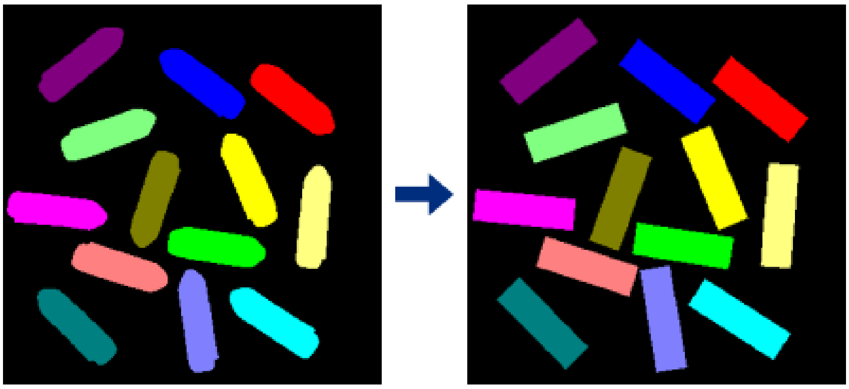
BoundingBox

When **BoundingBox** is selected as the Target Shape, the 2D shape before and after processing is shown below.



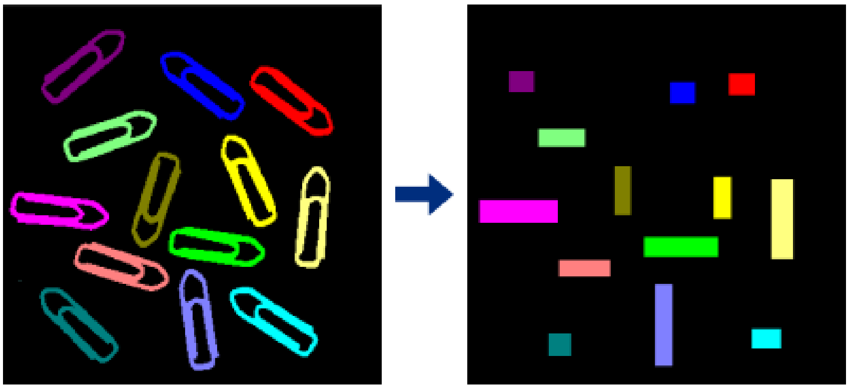
CircumscribedRectangle

When **CircumscribedRectangle** is selected as the Target Shape, the 2D shape before and after processing is shown below.



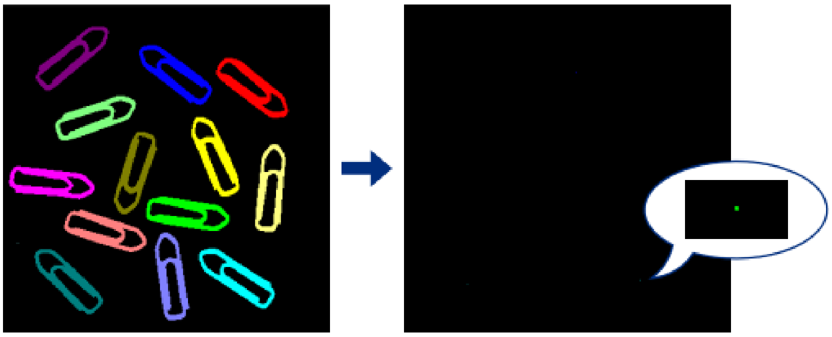
InscribedRectangle

When **InscribedRectangle** is selected as the Target Shape, the 2D shape before and after processing is shown below.



InnerCenter

When **InnerCenter** is selected as the Target Shape, the 2D shape before and after processing is shown below.



4.3.286. Project Points onto Plane

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step projects a point to the plane of the input pose.

Usage Scenario

This Step is a custom Step for special scenarios.

Input and Output

- Input:
  1. Poses input from this port will be used to define the plane.
  2. Poses input from this port will be used to define the points, which will be projected onto the plane.
- Output:
  1. Points of projection on the plane

4.3.287. Project 3D Point Cloud to 2D Image

Function

This Step converts a point cloud to a 2D image according to the specified projection type.

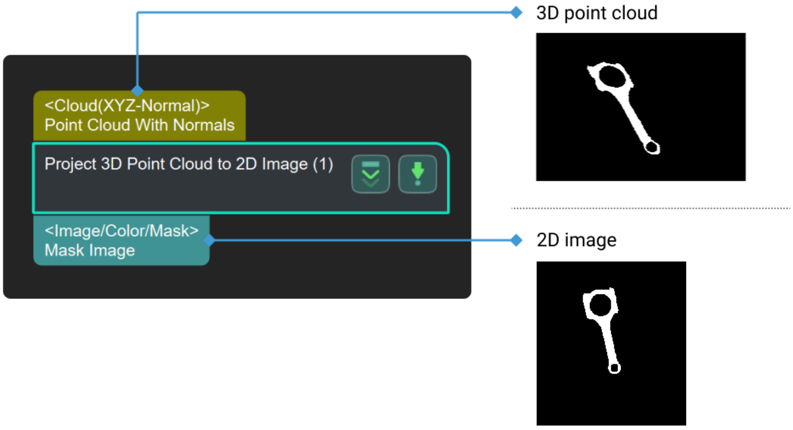
Usage Scenario

This Step is used for point cloud preprocessing. First, the point cloud will be projected onto a 2D image. Then, 2D image processing methods (such as 2D morphological operations) will be used to process the image, and the 2D image can be converted back to the point cloud by the Apply Masks to Point Cloud Step.

For picking scenarios, it is recommended to set the Projection Type to **PerspectiveProjection**.

For measurement scenarios, set the Projection Type to **OrthographicProjection**. However, using the Orthographic Projection Step to replace this Step is more recommended in this case.

Input and Output



Parameter Description

Projection Type

Parameter description: This parameter specifies the projection type to convert the point cloud to the 2D image.

Value list:

- PerspectiveProjection: perspective projection suitable for picking scenarios.
- OrthographicProjection: orthographic projection suitable for measurement scenarios.

Default value: PerspectiveProjection

PerspectiveProjection

Image Type

Parameter description: This parameter specifies the type of the output 2D image.

Value list:

- Color: colored image
- Depth: depth map

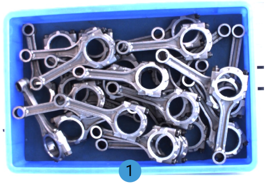
Default value: Color

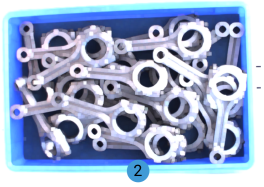
Dilate Kernel Size

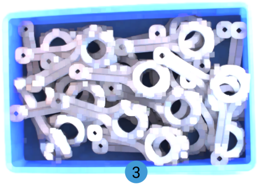
Parameter description: This parameter is used to set the size of dilate kernel (px). Dilate kernel is used to increase the area of the object, and connect the damaged parts. The larger the value, the more region dilates; the smaller the value, the less region dilates.

Default value: 21 px

Example: From Figure 1 to 3, the set kernel size is gradually increased.





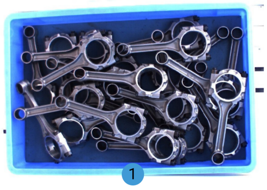


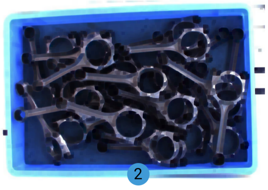
Erode Kernel Size

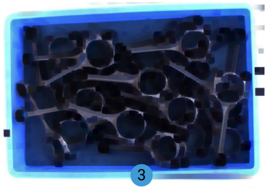
Parameter description: This parameter is used to set the size of erode kernel (px). Erode kernel is used to remove the small image noises, and separate connected objects. The larger the value, the more region erodes; the smaller the value, the less region erodes.

Default value: 21 px

Example: From Figure 1 to 3, the set kernel size is gradually increased.







OrthographicProjection

Projection Operation

Project Along Object Z Axis

Parameter description: Once this option is enabled, the point cloud will be projected along its Z-axis.

Default setting: selected.

Scale Ratio

Parameter description: This parameter specifies the scale ratio of the image after the projection.

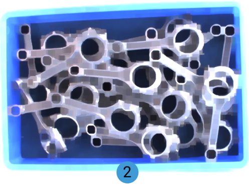
Default value: 1000

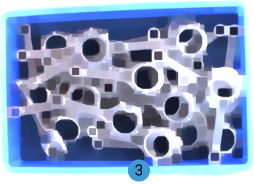
Closing Operation Kernel Size

Parameter description: This parameter is used to set the closing operation kernel size (px). It can fill small cavities in the object, connect neighboring objects, and smooth their boundaries without significantly altering their area. The larger the value, the more obvious the effect.

Default value: 7 px

Example: From Figure 1 to 3, the set kernel size is gradually increased.





4.3.288. Read Bar Code

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step reads barcodes in 2D images. Multiple code types are supported.

Usage Scenario

This Step is a general image processing Step. No fixed usage scenarios.

Input and Output

- Input:
  1. Barcodes in the input image of this port will be read.
- Output:
  1. Contents of barcodes read
  2. Positions of barcodes in the image

4.3.289. Read Images V2

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step reads 2D images from files or folders.

Usage Scenario

This Step is a general image reading Step. No fixed usage scenarios.

Input and Output

- **Input:** none.
- **Output:**
  1. Image read
  2. Filename of image read

4.3.290. Read Images

The contents of this Step are no longer under maintenance. Please use the new Step Read Images V2. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step reads 2D images.

Usage Scenario

This Step is a general image reading Step. No fixed usage scenarios.

Input and Output

- **Input:**

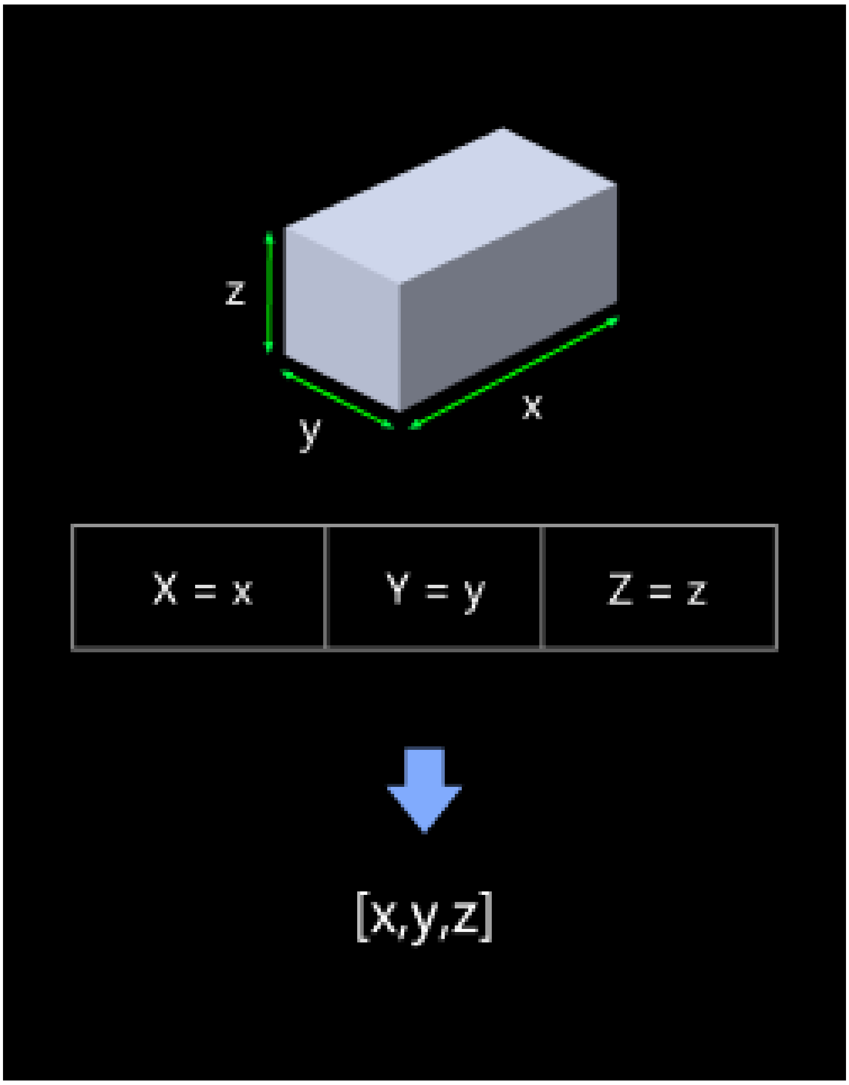
None

- **Output:**
  1. 2D image read
  2. Filename of the 2D image read

4.3.291. Read Object Dimensions

Function

Read a list of object dimensions.



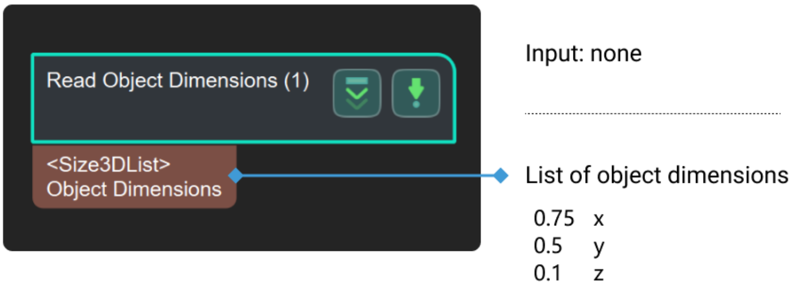
Usage Scenario

There are two ways to input the object dimensions in this Step:

1. A JSON file containing dimensions of object(s)
2. Parameters

This Step is usually used for dynamically setting box dimensions in **depalletizing projects**, or for the subsequent point cloud classification operations that require object dimensions.

Input and Output



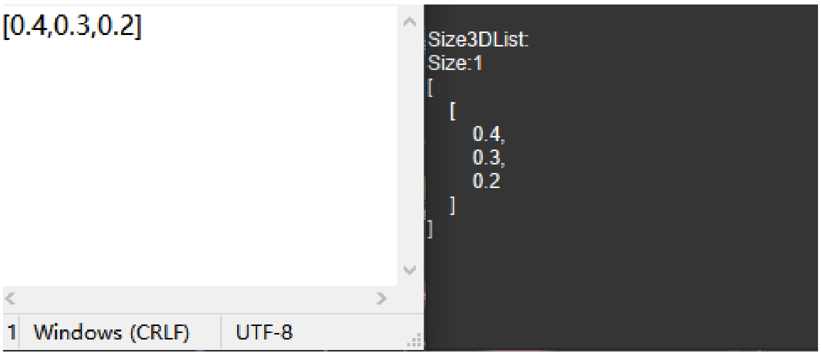
Parameter Description

Read Settings

Read Size from Properties

Default setting: selected.

Instruction: When the parameter is selected, the actual length of the box on the X, Y, and Z axes can be filled in the **Box Size Settings**. The filled-in size value will be read from the properties when running this Step. When it is not selected, the size information of the object will be read through the parameters in the specified file “boxSizesList.json”, as shown below. The content of the file is [0.4, 0.3, 0.2], and the length, width, and height of the corresponding box are 0.4 m, 0.3 m, and 0.2 m, respectively.



```
[0.4,0.3,0.2]
```

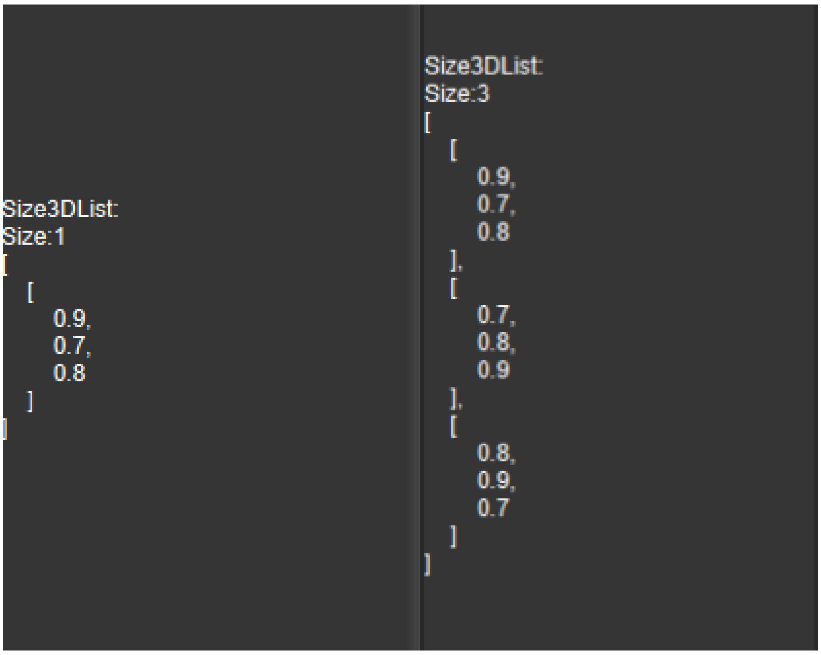
```
Size3DList:
Size:1
[
  [
    0.4,
    0.3,
    0.2
  ]
]
```

Box Size Settings

Auto Generate Box Sizes

Default Value: unselected.

Instruction: This parameter is used for multiple template matching and is not commonly used. By default, one set of size values are output, as shown in the figure on the left. When this parameter is selected, three sets of size values are output. The original x/y/z value in the parameter is xyz/yzx /zxy output, as shown in the figure on the right.



```
Size3DList:
Size:1
[
  [
    0.9,
    0.7,
    0.8
  ]
]
```

```
Size3DList:
Size:3
[
  [
    0.9,
    0.7,
    0.8
  ],
  [
    0.7,
    0.8,
    0.9
  ],
  [
    0.8,
    0.9,
    0.7
  ]
]
```

Length on X-axis

Default value: 100.000 mm

Instruction: Fill in the actual length of the object on the X axis (the unit is millimeters). The valid range is (0, +∞).

Length on Y-axis

Default value: 100.000 mm

Instruction: Fill in the actual length of the object on the Y axis (the unit is millimeters). The valid range is (0, +∞).

Length on Z-axis

Default value: 100.000 mm

Instruction: Fill in the actual length of the object on the Z axis (the unit is millimeters). The valid range is (0, +∞).

4.3.292. Read Point Cloud V2

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

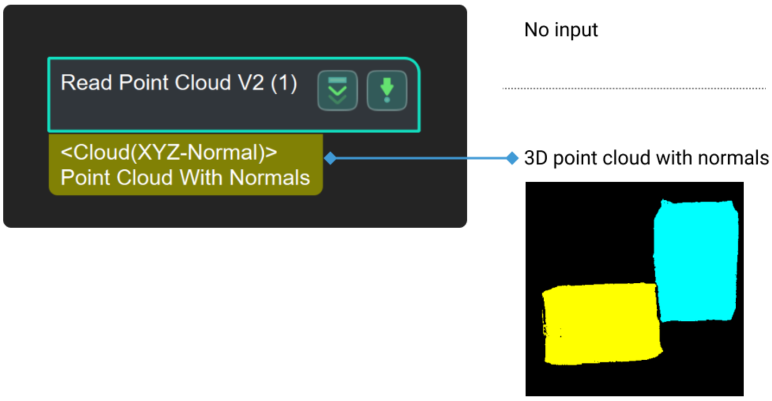
Function

Read point cloud from a file or files in a folder.

Usage Scenario

A general point cloud reading Step. Read the point cloud in PLY/PCD format from a specified path.

Input and Output



4.3.293. Read Point Cloud

The contents of this Step are no longer under maintenance. Please use the new Step Read Point Cloud V2. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

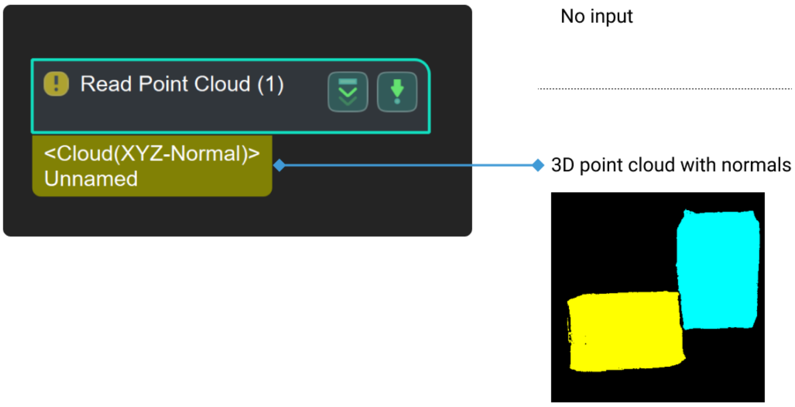
Function

Read point cloud in the PLY format.

Usage Scenario

Read the point cloud in PLY format from a specified path.

Input and Output



4.3.294. Read Poses from File

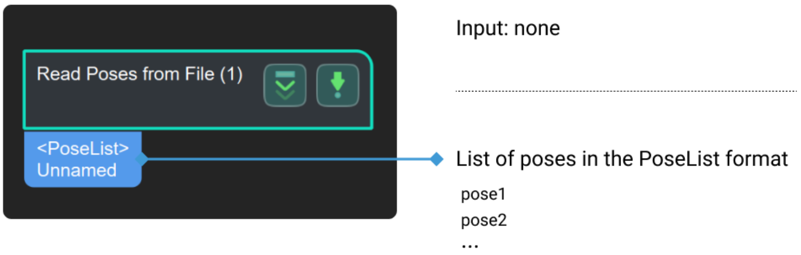
Function

Read a pose list from a local file.

Usage Scenario

This Step is a frequently used source of reference poses. If the current project does not have a specific pose file to read, it is recommended to use Easy Create Poses instead.

Input and Output



Parameter Description

Pose File Path

Description: This parameter is used to choose the path of pose files.

Instruction: Click 📂 to select the file you need.

4.3.295. Read QR Code

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step parses the QR code in the 2D image and outputs the information in the QR code in the form of a string.

Usage Scenario

This Step is a general QR code parsing Step. This Step has no fixed usage scenarios.

This Step is outdated. Please use the new Step Evaluate Results by HDevEngine and install Halcon.

Input and Output

- **Input:** QR code in the 2D image input by this port will be parsed.
- **Output:** parsed QR code content.

4.3.296. Read STL

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

Read point cloud data from an STL file.

Usage Scenario

This Step is a general point cloud reading Step. No fixed usage scenarios.

Input and Output

- **Input:** none.
- **Output:** point cloud read.

4.3.297. Read 3D ROI Center

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step reads a 3D ROI file and outputs the center point of the 3D ROI.

Usage Scenario

This Step is a custom Step for special scenarios.

Input and Output

- **Input**: none.
- **Output**: 3D ROI center as a pose.

4.3.298. Record Criterion Pose and Calc Transformation

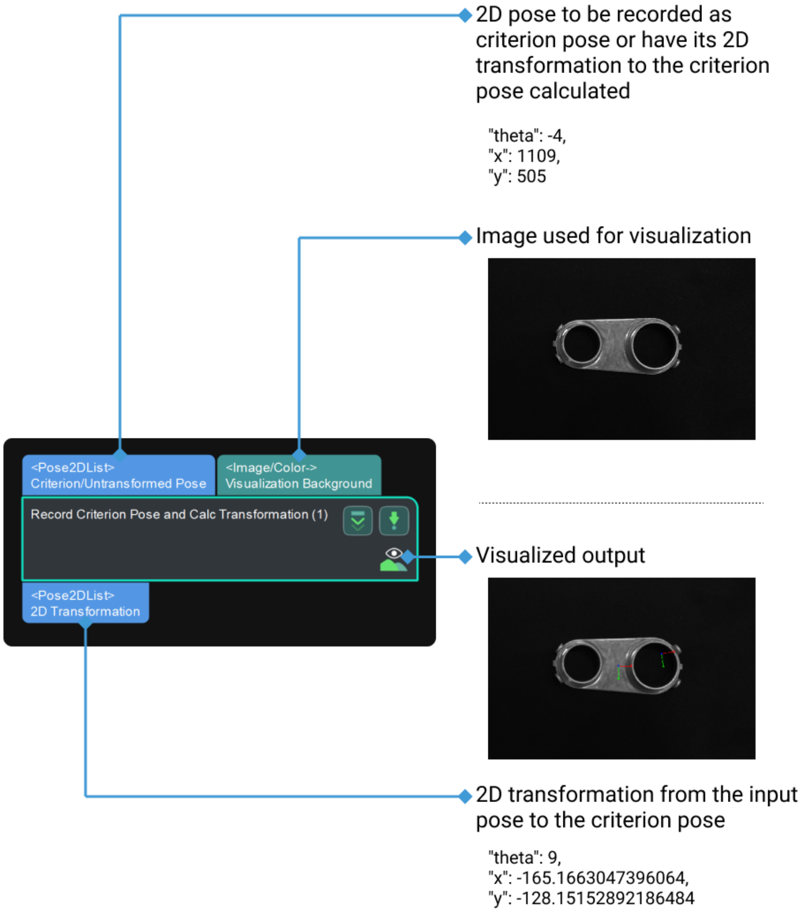
Function

This Step sets the input 2D pose from an execution of this Step as the criterion pose for images, and calculates the 2D transformation from the subsequently input 2D pose to the criterion pose.

Usage Scenario

This Step is generally used for correcting images in measurement scenarios so that the objects in all images are aligned in position and orientation, or for adjusting the position and orientation of the ROI so that it fits the object in the image. This Step is usually used with the Transform Image Step.

Input and Output



2D pose to be recorded as criterion pose or have its 2D transformation to the criterion pose calculated

"theta": -4,
"x": 1109,
"y": 505

Image used for visualization

Visualized output

2D transformation from the input pose to the criterion pose

"theta": 9,
"x": -165.1663047396064,
"y": -128.15152892186484

Parameter Description

Criterion Created

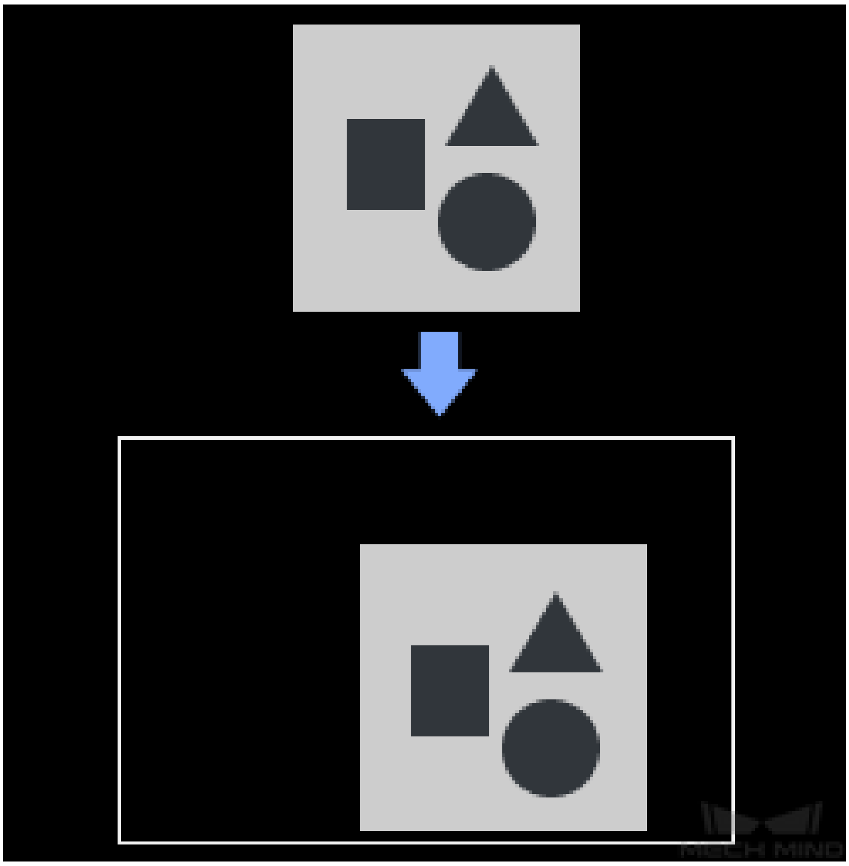
Instruction: The checkbox of the parameter leads to two scenarios:

- Unselected: Record the 2D pose input as the criterion pose.
- Selected: Calculate 2D transformations from input 2D poses to the criterion pose.

4.3.299. Recover Scaled Images in 2D ROI

Function

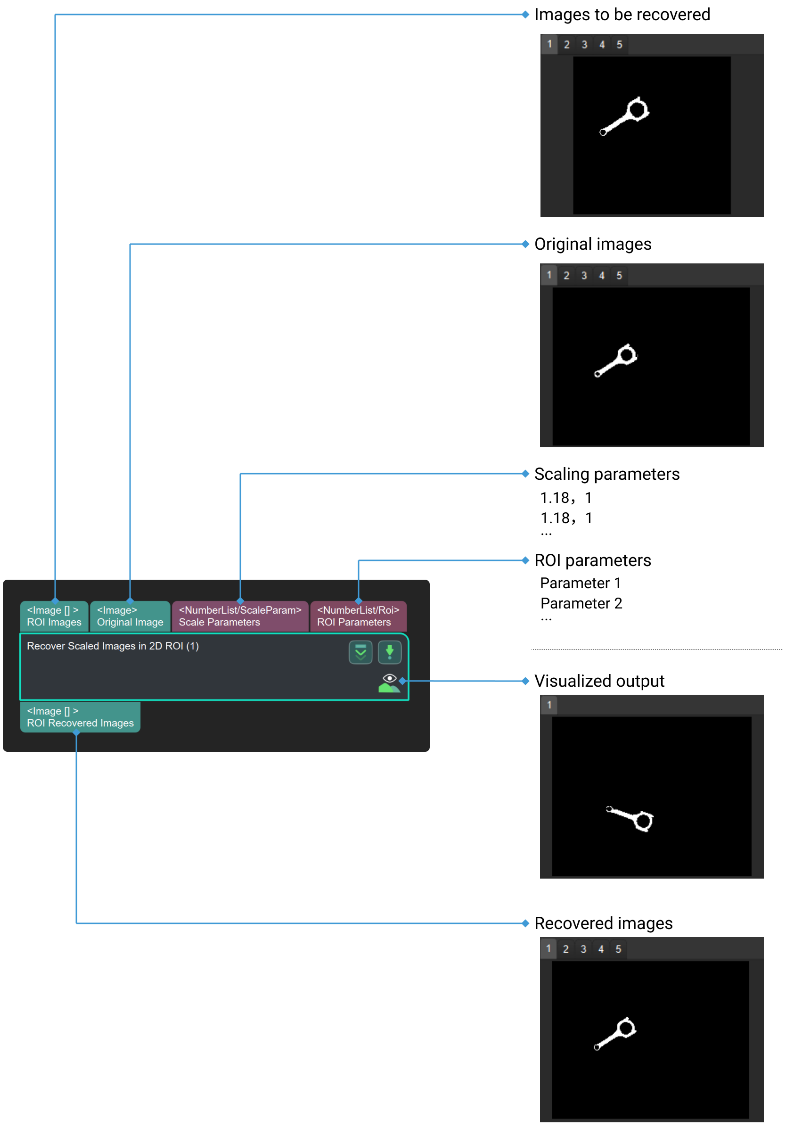
This Step can recover the input scaled 2D image in an ROI to its original size based on scaling parameters, and the recovered image will be displayed on a black background with a specified size.



Usage Scenario

This Step is usually connected with Scale Image in 2D ROI and other Steps related to deep learning.

Input and Output



Parameter Description

This Step does not have any parameter.

4.3.300. Rectify Image by Frame Transformation

Function

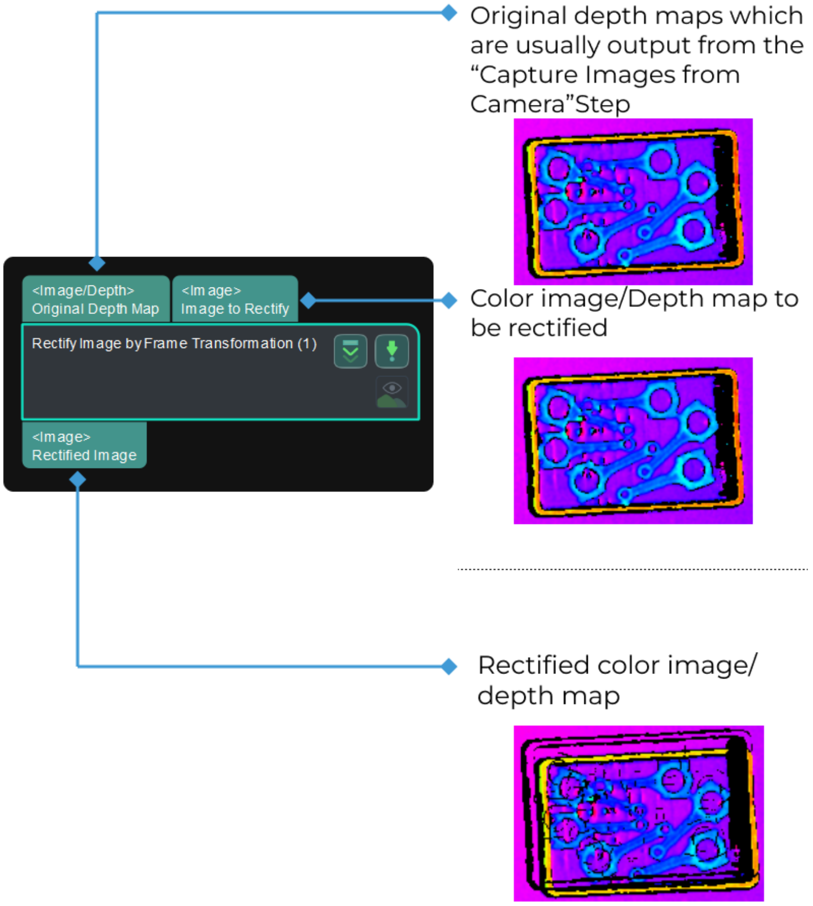
This Step is used to rectify the color image to the depth map frame, or vice versa, to make the pixels of the color image and the depth map have one-to-one correspondences, thus facilitating subsequent image processing.

Usage Scenario

This Step is used to solve the problem that the pixels of the original color image and depth map do not correspond due to the difference in camera reference frames.

It is not recommended to use this Step if the relationship between the reference frame of the depth map and the reference frame of the color image in the project is not clearly determined.

Input and Output



Original depth maps which are usually output from the “Capture Images from Camera”Step

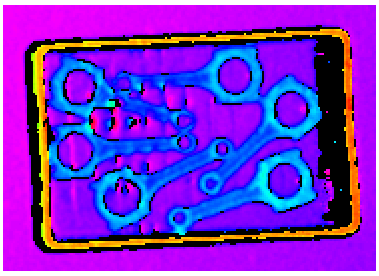
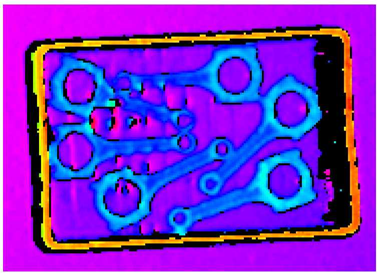
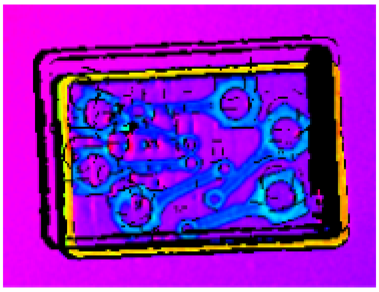
Color image/Depth map to be rectified

Rectified color image/ depth map

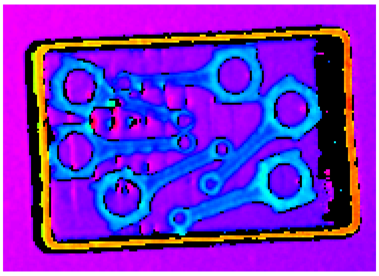
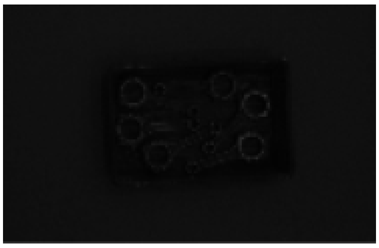
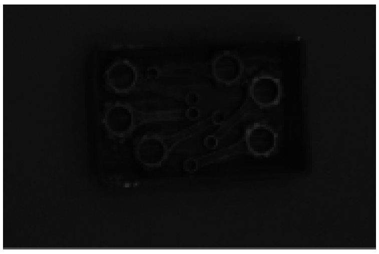
Parameter Description

Parameter		Description
Rectification Mode	Depth to Color Frame	Rectify the depth map to the color image reference frame.
	Color to Depth Frame	Rectify the color image to the depth map reference frame.

Rectify the depth map to the color image reference frame:

Original depth map	Depth map to be rectified	Rectified depth map
		

Rectify the color image to the depth map reference frame:

Original depth map	Color image to be rectified	Rectified color image
		

4.3.301. Rectify Ring Object Poses

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step corrects the input ring object pick points to the ring centers.

Usage Scenario

This Step is a custom Step for special scenarios.

Input and Output

- Input:
  1. Ring object pick points input at this port will be corrected to the ring centers.
  2. Masks of ring objects.
- Output:
  1. Corrected pick points

4.3.302. Remove All Overlapping Poses

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

When the distance between two poses is less than the set threshold, both poses will be removed.

Usage Scenario

This Step is a general object pose filtering Step. No fixed usage scenarios. No fixed usage scenarios.

Input and Output

- Input:
  1. Original pose list input to this port will be filtered.
  2. Labels corresponding to original poses.
  3. Confidences corresponding to original poses.
- Output:
  1. List of remaining poses
  2. Labels corresponding to remaining poses
  3. Confidences corresponding to remaining poses

4.3.303. Remove Cloud Points from Point Cloud

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step removes specified points from the input point cloud.

Usage Scenario

This Step is usually used to remove/retain the point clouds of models from the scene.

Input and Output

- Input:
  1. Original point cloud.
  2. Point cloud including the portion that needs to be removed from the original point cloud.
- Output:
  1. Original point cloud after removing part of it.
  2. Point cloud portion removed from the original point cloud.

4.3.304. Remove Noise in Masks

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step removes noise clusters whose area is smaller than the threshold. The threshold of the noise cluster area can be set in the parameters.

Usage Scenario

This Step is a general image processing Step. This Step has no fixed usage scenarios.

Input and Output

- **Input:** Noise in masks input at this port input will be removed.
- **Output:** masks after noise removal.

4.3.305. Remove Coinciding Objects

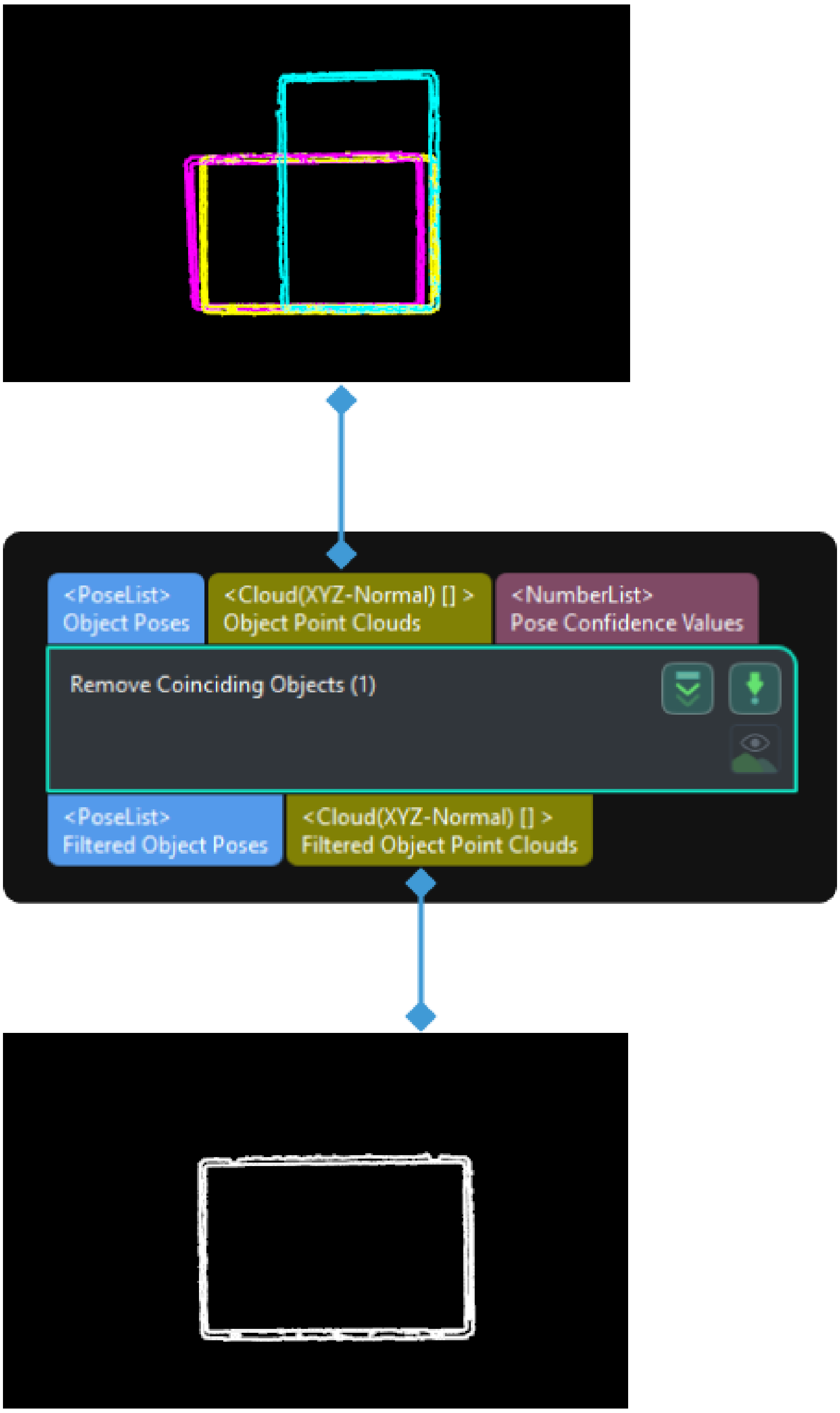
Function

Remove objects that are completely covered or covered by other objects according to the pose confidence values.

Usage Scenario

This Step is usually used after the Step 3D Fine Matching Lite (recommended) or the Step 3D Fine Matching to remove incorrect matching results.

Input and Output



Parameter Description

Parameter Tuning Level

Description: This parameter is used to set the tuning level of the parameters. The parameters in different tuning levels are different.

Value list: Basic, Advanced

Instruction: If the Parameter Tuning Level is set to Basic, please complete parameter settings under “Object Model Type” and “Threshold Setting.” If the Parameter Tuning Level is set to Advanced, parameter settings under **Advanced Settings** will also need to be configured.

Object Model Type

Object Model Type

Description: This parameter is used to select the type of the object's point cloud model.

Value list: Surface model, Edge model

Instruction: Select the type of the point cloud model according to the actual requirements. If Edge model is selected, please input the scene point cloud that represents the entire surface of the object instead of the scene point cloud that represents the extracted edges only.

Threshold Setting

Coincidence Ratio Threshold

Description: This parameter is used to set the threshold of the coincidence ratio between the object and another object. When the coincidence ratio is above this value, the objects with lower pose confidence will be judged as coinciding objects.

Default value: 30%

Advanced Settings

Method to Generate Pixel Size

Description: When the coincidence ratio is calculated based on the 2D orthographic projection of the object point cloud, this parameter is used to select the method to generate the size per pixel on the projected 2D image.

Value list: Auto, Manual

- Auto: Automatically determine the size per pixel on the projected 2D image.
- Manual: Set the **Pixel Size** manually.

Instruction: If the "Method to Generate Pixel Size" is set to "Auto," the input point clouds must belong to the same object type. This feature does not support inputting point clouds of different object types. If the input point clouds belong to different object types, please select "Manual" in the "Method to Generate Pixel Size."

Pixel Size

Description: Set the "Pixel Size" in millimeters.

Default value: 2.5 mm

Instruction: It is recommended to set the "Pixel Size" the same as the "Sampling Interval" when making the point cloud model. If the value is set too large, the accuracy of the calculated coincidence ratio may be reduced. If the value is set too small, the calculated coincidence ratio may be lower.

4.3.306. Remove Overlapped Objects (Lite)

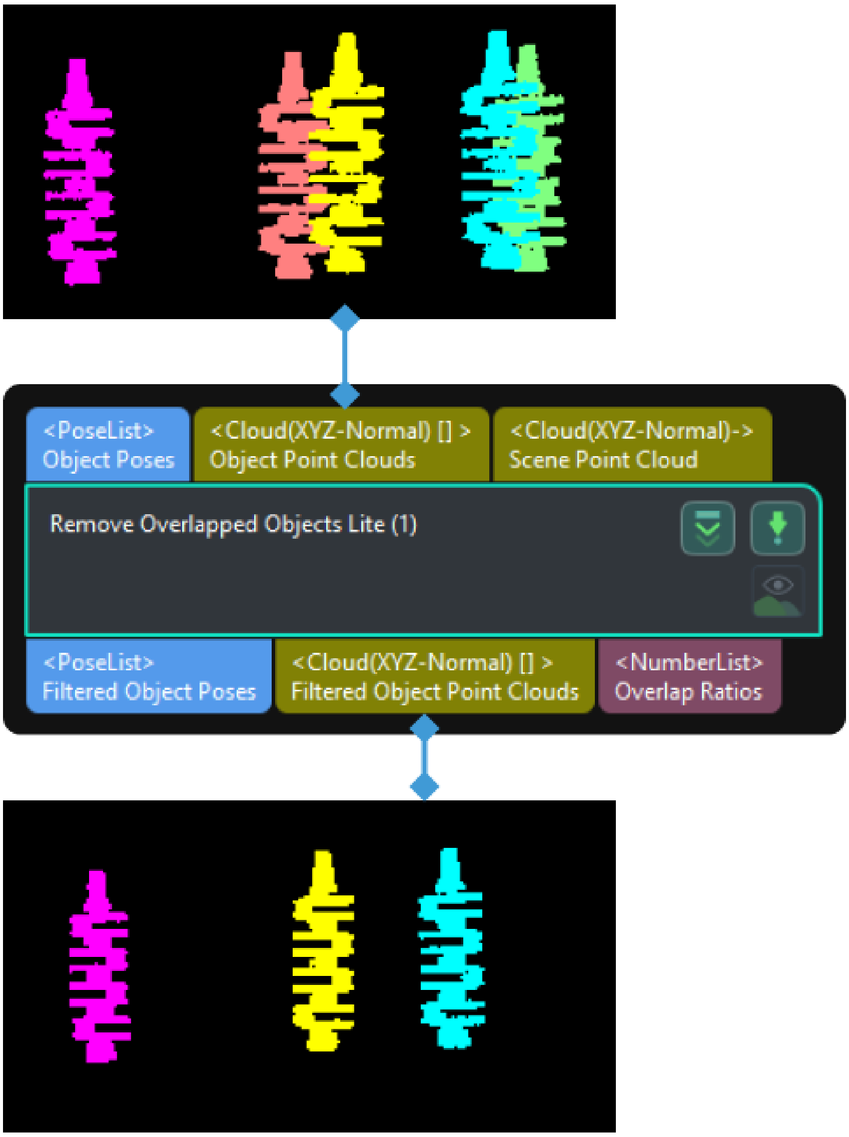
Function

This Step is used to remove overlapped objects, i.e., when multiple vision results exist at the same position of a workpiece, ineligible vision results can be removed by adjusting parameters.

Usage Scenario

This Step is usually used after 3D Fine Matching Lite (recommended) or 3D Fine Matching to remove overlapped objects that are not suitable for picking.

Input and Output



Parameter Description

Parameter Tuning Level

Description: This parameter is used to set the tuning level of the parameters. The parameters in different tuning levels are different.

Value list: Basic, Advanced

Instruction: If the Parameter Tuning Level is set to Basic, please complete parameter settings under “Object Model Type” and “Threshold Setting”. If the Parameter Tuning Level is set to Advanced, parameter settings under **Advanced Settings** will also need to be configured.

Object Model Type

Object Model Type

Description: This parameter is used to select the type of the object's point cloud model.

Value list: Surface model, Edge model

Instruction: Select the type of the point cloud model according to the actual requirements. If Edge model is selected, please input the scene point cloud that represents the entire surface of the object instead of the scene point cloud that represents the extracted edges only.

Threshold Setting

Overlap Ratio Threshold

Description: The threshold of the overlap ratio between the object and other objects. If the overlap ratio is above this value, the object will be considered overlapped.

Default value: 30%

Advanced Settings

Method to Generate Pixel Size

Description: When the overlap ratio is calculated based on the 2D orthographic projection of the object point cloud, select the method to generate the size per pixel on the projected 2D image.

Value list: Auto, Manual

- Auto: Automatically generate the size per pixel on the projected 2D image.
- Manual: Set the **Pixel Size** manually.

Instruction: If the "Method to Generate Pixel Size" is set to "Auto", the input point clouds must belong to the same object type. This feature does not support input point clouds of different object types. If the input point clouds belong to different object types, please adjust the "Method to Generate Pixel Size" to "Manual".

Pixel Size

Description: Set the pixel size in millimeters.

Default value: 2.5 mm

Instruction: It is recommended to set the "Pixel Size" the same as the "Sampling Interval" when making the point cloud model. If the value is set too large, the accuracy of the calculated overlap ratio may be reduced. If the value is set too small, the calculated overlap ratio may be lower.

Range of Scene Point Removal around Object

Description: Within this range, points in the scene point cloud around the object will be removed, and the rest scene point cloud will be involved in the overlap detection. The unit is millimeters.

Default value: 3 mm

Visualization Option

Description: This parameter is used to select the item that you want to visualize.

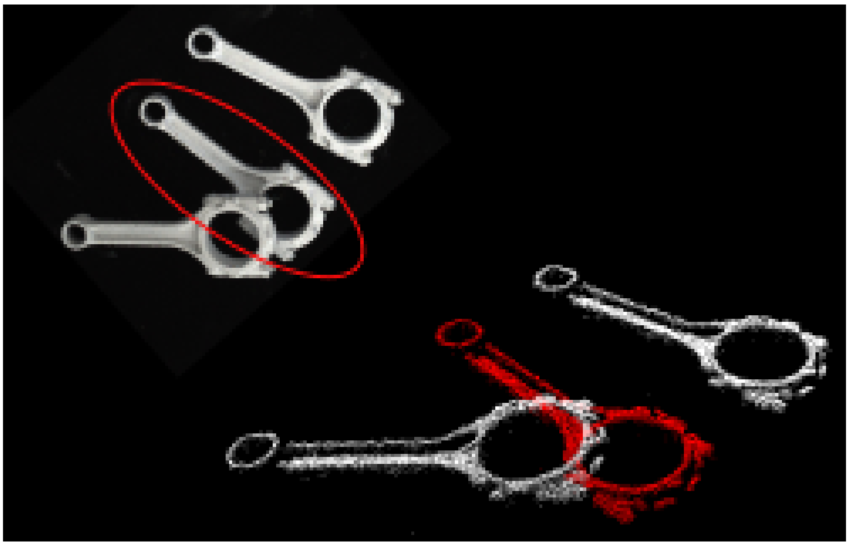
Options:

- Filtered result: The scene point cloud is displayed in white, overlapped objects with overlap ratios above the threshold are displayed in red, and objects that are not overlapped are displayed in other colors.
- Object projection image: 2D image of an individual object after the projection.
- Background depth map without objects: The depth map of the background after objects are removed.
- Bounding box point cloud of edge model: Bounding box point cloud of the object’s edge model.

4.3.307. Remove Overlapped Objects V2

Function

This Step removes the poses of overlapped objects according to user-defined rules. As shown in the figure below, the red object in the lower right corner is the removed overlapped object.



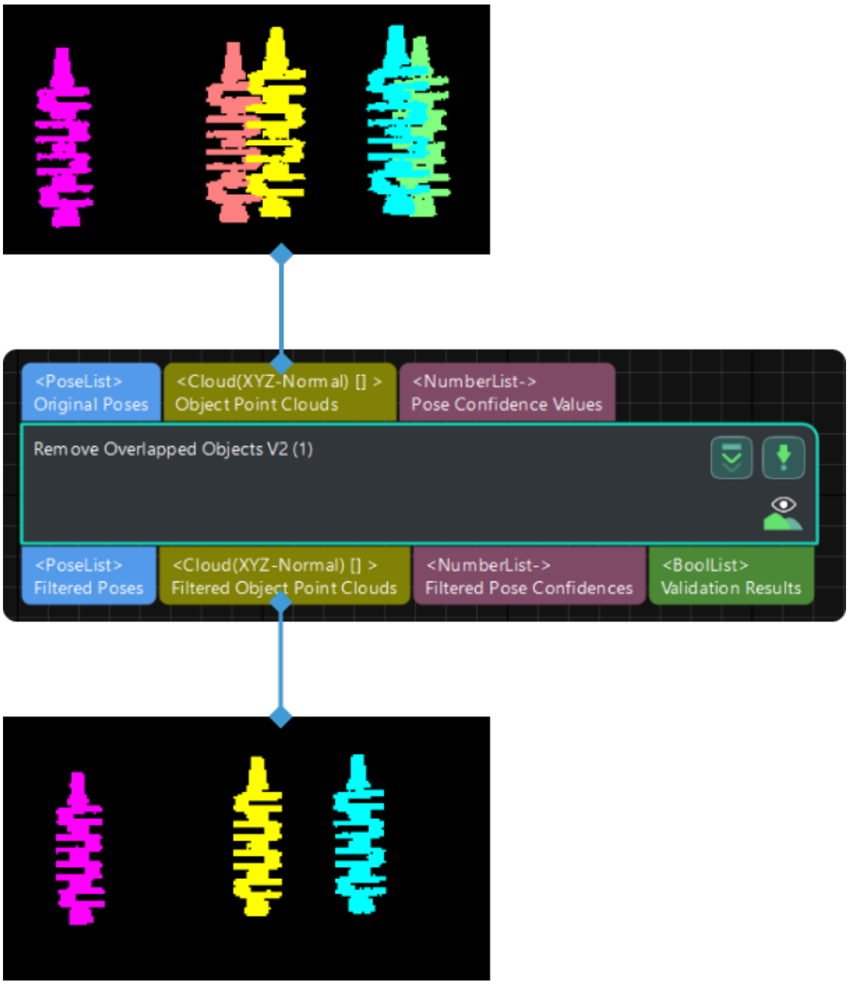
Usage Scenario

This Step is usually used after the 3D Fine Matching Step. It filters matched poses according to different requirements and therefore the vision recognition result of the overlapped object can be removed.

The Projection (2D) method in Remove Overlapped Objects V2 has been improved, and therefore it is recommended to use the new Remove Overlapped Objects V2 Step if the Projection (2D) method is used to remove overlapped objects.

The Bounding Box (3D) method in Remove Overlapped Objects V2 is the same as the BoundingBoxOfObjectIn3DBased method in Remove Overlapped Objects, and therefore you can use either the Remove Overlapped Objects or Remove Overlapped Objects V2 Step if the Bounding Box (3D) method is used to remove overlapped objects.

Input and Output



Parameter Description

This Step provides two methods to determine whether the object is overlapped:

- Bounding Box (3D): The bounding box of the point cloud will be divided into numerous cubes, and the overlap ratio is calculated according to the number of overlapped cubes.
- Projection (2D): The overlap ratio is calculated according to the size of the object's overlapped area projected on a 2D plane.

Method Setting

Method

Description: This parameter is used to select the method to determine whether the object is overlapped.

Value list: Bounding Box (3D) and Projection (2D).

Bounding Box (3D): If 3D matching is not used in the project or the edge matching method for 3D matching is used, please select this method. The method uses the 3D bounding boxes of the point clouds to determine whether the objects are overlapped. A 3D bounding box refers to a cuboid with the object pose as the center and with edges parallel to the X, Y, and Z axes of the poses.

- Projection (2D): If 3D matching is used in the project, it is recommended to use this method. The method works by calculating the overlap area ratios of object projections on a

2D plane. Once this method is selected, you only need to set the **Overlap Ratio Threshold** parameter.

Tuning recommendation: Please select the method according to the actual requirement.

Bounding Box (3D)

Threshold Setting

Overlap Ratio Threshold (0–1.0)

Description: This parameter is used to determine whether to remove the vision recognition result of the overlapped object. The overlap ratio refers to the volume of a bounding box that overlaps with another bounding box to the total volume of this bounding box. If the overlap ratio of an object's point cloud is above this threshold, the vision recognition result of this object will be removed.

Value range: 0.00–1.00

Default value: 0.30

Tuning recommendation: You can adjust according to the actual requirement and take the step of 0.01 to adjust the **Overlap Ratio Threshold**. Please refer to the tuning example for the application result.

Point Cloud Resolution Settings

Point Cloud Diagonal Ratio

Description: This parameter is used to divide the bounding box into little cubes, and the Overlap Ratio Threshold will be calculated from the number of the overlapped cubes. The side length of the cube = Point cloud diagonal ratio × Diagonal length of the object point cloud.

Default value: 2.00%

Tuning recommendation: Please set this parameter according to the actual situation.

Object Dimension Settings

Method to Calc Object Height

Description: This parameter is used to select the method to calculate the object's height.

Value list: Specified Height, and Calc from Point Cloud (default).

- Specified Height: Specify a fixed object height by setting the **Specified Height** parameter value.
- Calc from Point Cloud: The Step will calculate the object height from the point cloud automatically.

Tuning recommendation: If the point cloud is relatively planar or cannot represent the complete shape of the object, you should specify the object height yourself.

Specified Height

Description: This parameter is used to specify the object's height in millimeters. When the **Method to Calc Object Height** is set to **Specified Height**, please set a value for this parameter.

Default value: 100.000 mm

Tuning recommendation: Please enter a value according to the actual situation.

Bounding Box Expansion

Expansion Ratio along X-Axis

Description: Expanding the bounding boxes along the X-Axis of object poses allows for more sensitive detection of overlap.

Default value: 1.0000

Tuning recommendation: Please set this parameter according to the actual situation.

Expansion Ratio along Y-Axis

Description: Expanding the bounding boxes along the Y-Axis of object poses allows for more sensitive detection of overlap.

Default value: 1.0000

Tuning recommendation: Please set this parameter according to the actual situation.

Expansion Ratio along Z-Axis

Description: Expanding the bounding boxes along the Z-Axis of object poses allows for more sensitive detection of overlap. When the **Method to Calc Object Height** is set to **Specified Height**, you do not need to set this parameter.

Default value: 3.0000

Tuning recommendation: Please set this parameter according to the actual situation.

Projection (2D)

Threshold Setting

Overlap Ratio Threshold (0–1.0)

Description: This parameter is used to determine whether to remove the vision recognition result of the overlapped object. The overlap ratio refers to the mask area that overlaps with another mask to the total area of this mask. If the overlap ratio of an object's point cloud is above this threshold, the vision recognition result of this object will be removed.

Default value: 0.30

Value range: 0.00–1.00

Tuning recommendation: You can adjust according to the actual requirement and take the step of 0.01 to adjust the **Overlap Ratio Threshold**. Please refer to the tuning example for the application result.

Advanced Settings

This parameter group will only be displayed after you right-click and select **Show all parameters** in the context menu.

Projection Type

Description: This parameter is used to select the type of the projection.

Value list: Orthographic Projection and Perspective Projection.

Orthographic Projection: Objects are not stretched along the depth of view.

- Perspective Projection: Objects are stretched along the depth of view.

Tuning recommendation: Please set this parameter according to the actual situation.

Dilation Kernel Size

Description: This parameter is used to specify the kernel size for dilating the projected image, and zero-value pixels on the projected 2D image can be avoided.

Default value: 1

Tuning recommendation: Please set this parameter according to the actual situation.

Downsampling Factor

Description: This parameter is used to set a factor for downsampling the projected image to improve the processing speed. When the **Projection Type** is **Perspective Projection**, you need to set this parameter. For example, if the factor is 2, a 100*100 image will be downsampled to 50*50.

Default value: 2

Tuning recommendation: Please set this parameter according to the actual situation.

Orthographic Projection Resolution

Description: This parameter is used to set the resolution for orthographic projection, i.e., the number of pixels per meter (m) on the 2D image formed by orthographic projection. When the **Projection Type** is **Orthographic Projection**, you need to set this parameter.

Default value: 500.0000

Tuning recommendation: Please set this parameter according to the actual situation.

Point Cloud Removal Range around Model

Description: When object point cloud is generated from the input poses and the model, scene point clouds around the generated point cloud within this removal range will be removed, and the remaining scene point cloud will be used in the overlap detection. The unit is millimeters.

Default value: 3.000 mm

> ℹ Scene point clouds refer to all point clouds in the camera's field of view.

Visualization Options

Parameter description: This parameter is used to select the item that you want to visualize.

Value list: Filtered result, Scene point cloud with removed parts highlighted, and Projection image.

Filtered result: View the filtered result, in which the point cloud of overlapped objects are removed, in the Debug Output window.

- Scene point cloud with removed parts highlighted: The scene point cloud needs to be involved in overlap detection, especially those parts of unrecognized objects. Before performing overlap detection, the recognized object parts of the scene point cloud need to be removed to make space for the object point clouds. The object point cloud is the point cloud generated from the input poses and the model.

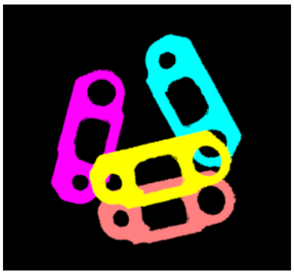
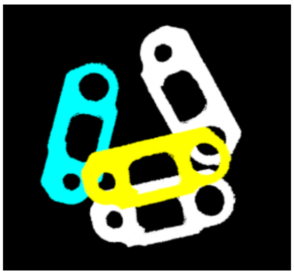
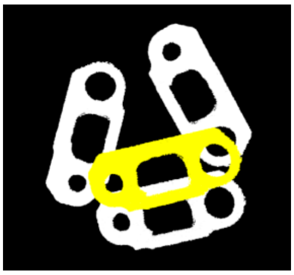
- Projection image: View the projection image of the first object in the list to preview the projection effect.

Tuning recommendation: Please set this parameter according to the actual situation. Please refer to the tuning example for the corresponding result.

Tuning Examples

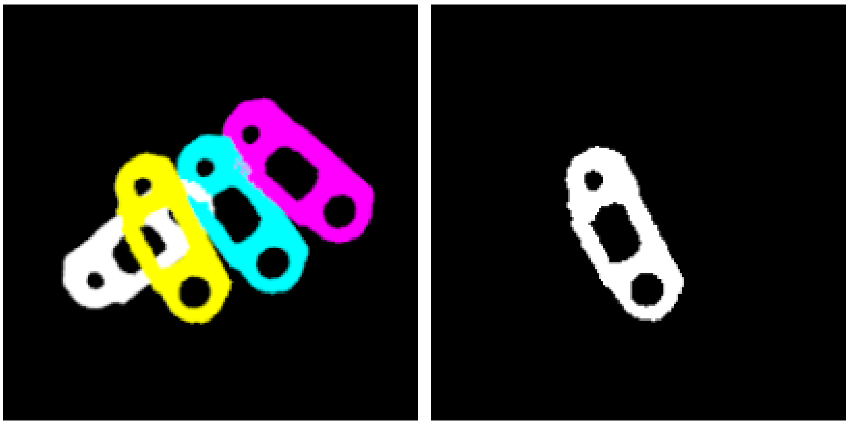
Overlap Ratio Threshold

When the **Overlap Ratio Threshold** is set to 0.60, 0.10, and 0.03, the visualized outputs are shown in the table below. The removed point clouds of the overlapped objects are displayed in white.

Threshold value	0.60	0.10	0.03
Visualized output			

Visualization Options

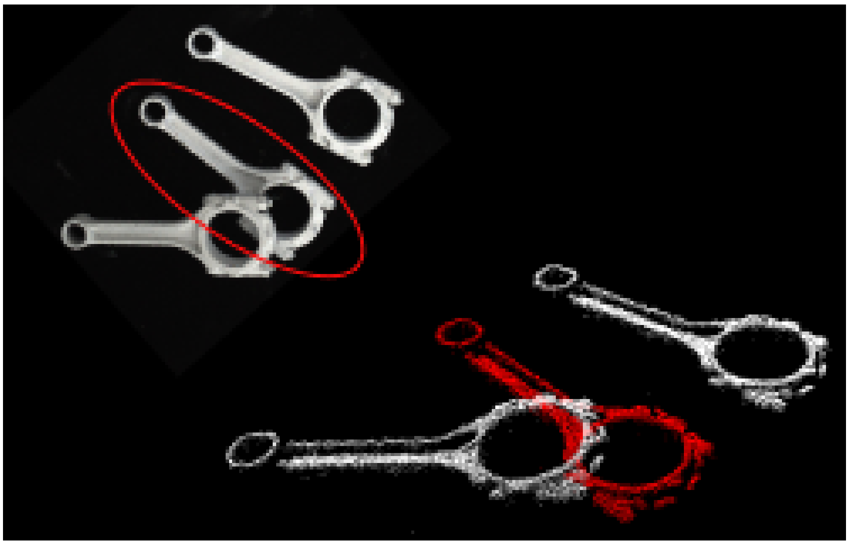
When the **Visualization Options** is set to **Filtered result** and **Projection image**, the visualized outputs are shown in the figure below.



4.3.308. Remove Overlapped Objects

Function

This Step is used to remove the poses of the overlapped objects according to user-defined rules. As shown in the figure below, the red object in the lower right corner is the removed overlapped object.

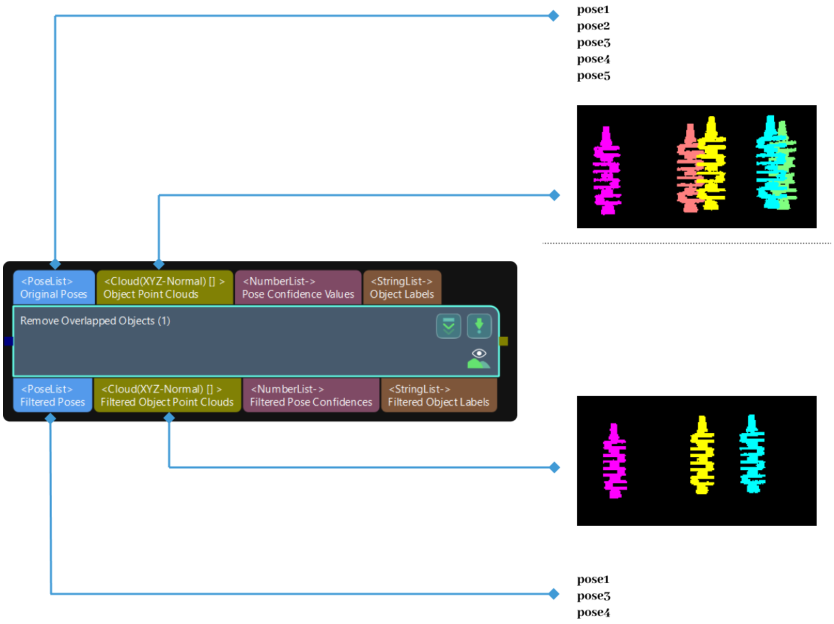


Usage Scenario

This Step is usually used after the Step 3D Fine Matching to filter the matched poses according to different requirements and therefore the poses of overlapped objects can be removed.

> The Projection (2D) method in Remove Overlapped Objects V2 has been improved, and therefore it is recommended to use the new Remove Overlapped Objects V2 Step if the Projection (2D) method is used to remove overlapped objects.
>
> The Bounding Box (3D) method in Remove Overlapped Objects V2 is the same as the BoundingBoxOfObjectIn3DBased method in Remove Overlapped Objects, and therefore you can use either the Remove Overlapped Objects or Remove Overlapped Objects V2 Step if the Bounding Box (3D) method is used to remove overlapped objects.

Input and Output



Parameter Description

There are two types of algorithms used in this Step. Please adjust the parameters according to the selected algorithm.

BoundingBoxOfObjectIn3DBased (Calculation with Bounding Boxes)

This algorithm divides the bounding box of the object point cloud into little cubes, and the number of overlapped cubes between objects is used to calculate the overlap ratio of the overlapped part of the entire object point cloud.

Threshold Settings

Overlap Ratio Threshold (0–1.0)

Description: This parameter is used to determine whether to remove the vision recognition result of the overlapped object. The overlap ratio refers to the volume of a bounding box that overlaps with another bounding box to the total volume of this bounding box. If the overlap ratio of an object's point cloud is above this threshold, the vision recognition result of this object will be removed. Please refer to Overlap Ratio Threshold for the corresponding results.

Value range: 0–1.0

Default value: 0.60

Tuning recommendation: You can adjust according to the actual requirement and take the step of 0.01 to adjust the **Overlap Ratio Threshold**.

Cloud Resolution Settings

Leaf Size

Tuning recommendation: This parameter is used to set the dimensions of the point cloud bounding box. Then the quantity of the point cloud bounding box can be used to calculate the overlap ratio between the object and overlapped object.

Default value: 3 mm

Tuning recommendation: Please set the value according to the actual situation.

Object Height Settings

Method to Calc Object Height

Description: This parameter is used to select the method to calculate the object's height.

Value list:

UsingInputCloudHeight (default value): The Step will calculate the object height from the point cloud automatically.

- UsingSpecifiedHeight: Specify a fixed object height by setting the Specified Height parameter value. It is recommended to use this method when the point cloud is relatively planar or cannot represent the complete shape of the object.

Specified Height

Description: This parameter is used to specify the object's height in millimeters. When the **Method to Calc Object Height** is set to **Specified Height**, please set a value for this parameter.

Default value: 100 mm

Tuning recommendation: Please set this parameter according to the actual situation.

Expansion Settings

Expanding Ratio Along X-axis

Description: Expanding the bounding boxes along the X-Axis of object poses allows for more sensitive detection of overlap.

Default value: 1.0000

Tuning recommendation: Please set this parameter according to the actual situation.

Expanding Ratio Along Y-axis

Description: Expanding the bounding boxes along the Y-Axis of object poses allows for more sensitive detection of overlap.

Default value: 1.0000

Tuning recommendation: Please set this parameter according to the actual situation.

Expanding Ratio Along Z-axis

Description: Expanding the bounding boxes along the Z-Axis of object poses allows for more sensitive detection of overlap. When the **Method to Calc Object Height** is set to **Specified Height**, you do not need to set this parameter.

Default value: 3.0000

Tuning recommendation: Please set this parameter according to the actual situation.

ProjectionOfObjectIn2DBased (Calculation by 2D Projection)

This algorithm uses the size of the object’s overlapped area projected on a 2D plane to claculate the overlap ratio and then determines whether the object is overlapped.

Threshold Settings

Overlap Ratio Threshold (0–1.0)

Description: This parameter is used to determine whether to remove the vision recognition result of the overlapped object. The overlap ratio refers to the volume of a bounding box that overlaps with another bounding box to the total volume of this bounding box. If the overlap ratio of an object’s point cloud is above this threshold, the vision recognition result of this object will be removed. Please refer to Overlap Ratio Threshold for the corresponding results.

Value range: 0–1.0

Default value: 0.60

Tuning recommendation: You can adjust according to the actual requirement and take the step of 0.01 to adjust the **Overlap Ratio Threshold**.

Object Height Settings

Method to Calc Object Height

Description: This parameter is used to select the method to calculate the object’s height.

Value list:

UsingInputCloudHeight (default value): The Step will calculate the object height from the point cloud automatically.

- UsingSpecifiedHeight: Specify a fixed object height by setting the Specified Height parameter value. It is recommended to use this method when the point cloud is relatively planar or cannot represent the complete shape of the object.

Specified Height

Description: This parameter is used to specify the object’s height in millimeters. When the **Method to Calc Object Height** is set to **Specified Height**, please set a value for this parameter.

Default value: 100 mm

Tuning recommendation: Please set this parameter according to the actual situation.

Advanced Calculation Settings

Mask Closing Kernel Size

Description: This parameter is used to specify the kernel size for closing the projected mask.

Default value: 3

Tuning recommendation: Please set the value according to the actual situation.

Depth Img Dilate Kernel Size

Description: This parameter is used to specify the kernel size for dilating the projected depth map.

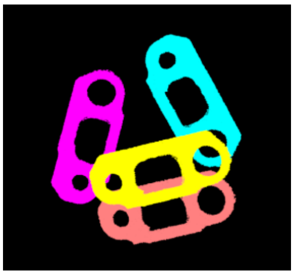
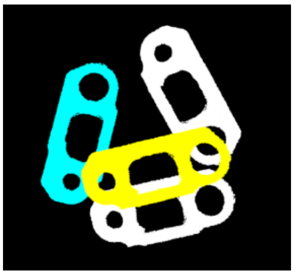
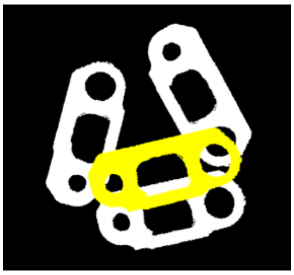
Default value: 4

Tuning recommendation: Please set the value according to the actual situation.

Tuning Example

Overlap Ratio Threshold

When the **Overlap Ratio Threshold** is set to 0.60, 0.10, and 0.03, the visualized outputs are shown in the table below. The removed point clouds of the overlapped objects are displayed in white.

Threshold value	0.60	0.10	0.03
Visualized output			

4.3.309. Remove Overlapped Polygons

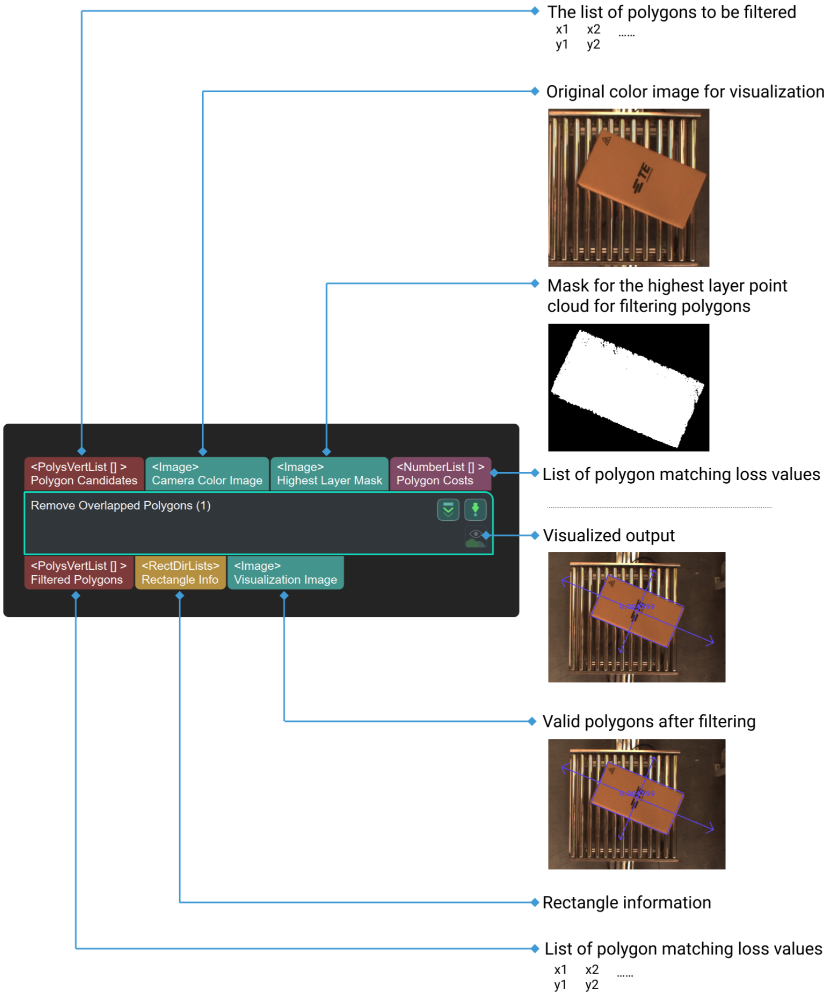
Function

This Step filters out invalid polygons that overlap or are outside the masks.

Usage Scenario

It should be used with the Step Remove Polygons outside Mask to filter out polygons that do not represent the actual contour of the object.

Input and Output


The list of polygons to be filtered
x1 x2 ......
y1 y2
Original color image for visualization
Mask for the highest layer point cloud for filtering polygons
List of polygon matching loss values
Visualized output
Valid polygons after filtering
Rectangle information
List of polygon matching loss values
x1 x2 ......
y1 y2
<PolysVertList [] > Polygon Candidates
<Image> Camera Color Image
<Image> Highest Layer Mask
<NumberList [] > Polygon Costs
Remove Overlapped Polygons (1)
<PolysVertList [] > Filtered Polygons
<RectDirLists> Rectangle Info
<Image> Visualization Image

Parameters

Mask Settings

Usage of Input Masks

Default value: SingleMask

List of values: SingleMask (one mask applies to all polygons), CorrespondingMasks (One mask applies to one polygon)

Instruction: To change the way the mask is used. You can choose to apply one mask to all polygons or apply different masks to each polygon accordingly.

Inspection Criteria

Min Out-Of-Bound Area Ratio (0–1.0)

Default value: 0.4500

Instruction: The mask of the box is translated by one-half of the length and one-half of the width along the positive and negative directions of the long and short axes. The ratio of the area of the shifted mask beyond the highest layer of the mask to the area of the mask itself will be calculated, and if it is greater than the threshold, the box is considered to be on the boundary of the highest layer. Generally speaking, if the rectangular box mask is located at the corners of the highest layer mask, as long as the mask is shifted by one-half of the length or width in the direction of removing the highest layer mask, the ratio should be 0.5, so the minimum threshold default value is 0.45 . When the value is increased, generally speaking, it exceeds 0.5, and all polygons will be filtered out. This Step will give priority to the mask of the box at the corner point, so the carton mask area at the border needs to be determined first.

Max Out-Of-Bound Area Ratio (0–1.0)

Default value: 0.6000

Instruction: The mask of the box is translated by one-half length and one-half width along the positive and negative directions of the long and short axes. The ratio of the area of the shifted mask beyond the highest layer of the mask to the area of the mask itself will be calculated, and if it is less than the threshold, the box is considered to be on the boundary of the highest layer. Generally speaking, if the rectangular box mask is located at the corners of the highest layer mask, as long as the mask is shifted by one-half of the length or width in the direction of removing the highest layer mask, the ratio should be 0.5, so the default maximum threshold is 0.6. This Step will give priority to the mask of the box at the corner point, so the carton mask area at the border needs to be determined first.

Max Overlap Ratio (0–1.0)

Default value: 0.2000

Instruction: If the overlap ratio of two polygons exceeds the threshold. one of the polygons should be filtered out.

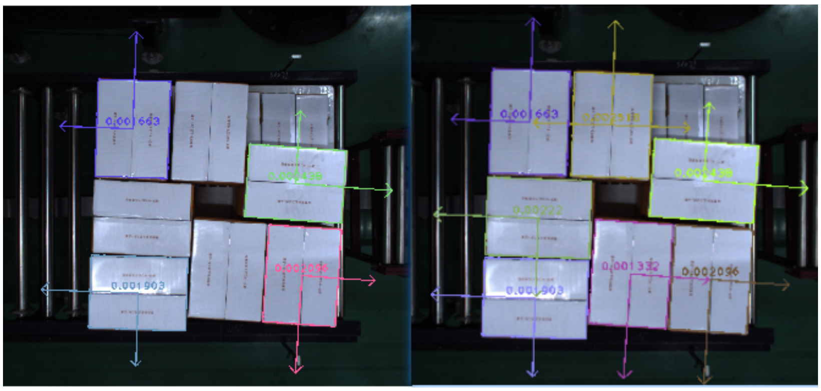
Output Strategy

Position Restraint

Default value: AtCorners

List of values: AtCorners, None

Instruction: As shown below, the left side is the output image under the default setting, and the right side is the output image when the parameter is set to **None**.



4.3.310. Remove Cloud Points from Point Cloud

Function

This Step removes one cloud from other one.

4.3.311. Remove Polygons outside Mask

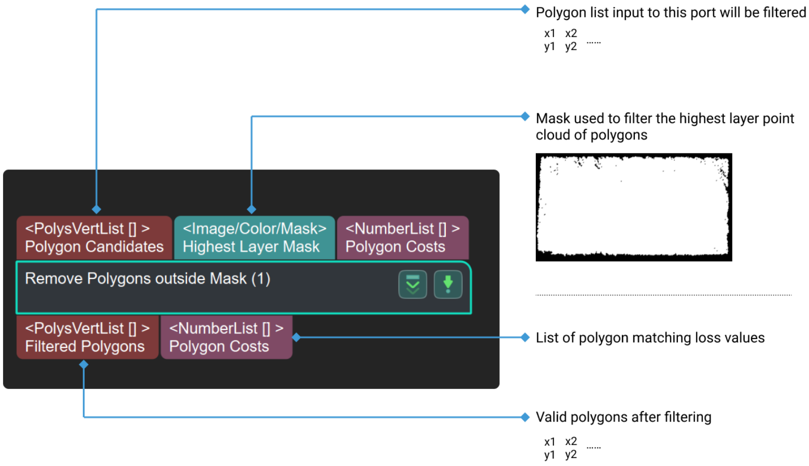
Function

This Step filters out interfering polygons whose overlapping area with the mask is smaller than the set threshold.

Usage Scenario

This Step is fixedly used with the Step Remove Overlapped Polygons to filter out the interfering polygons from matching by the mask.

Input and Output


<PolysVertList [] > Polygon Candidates
<Image/Color/Mask> Highest Layer Mask
<NumberList [] > Polygon Costs
Remove Polygons outside Mask (1)
<PolysVertList [] > Filtered Polygons
<NumberList [] > Polygon Costs
x1 x2 y1 y2 ......
x1 x2 y1 y2 ......

Polygon list input to this port will be filtered

Mask used to filter the highest layer point cloud of polygons

List of polygon matching loss values

Valid polygons after filtering

Parameter Description

Usage of Input Masks

Default value: SingleMask

Value list: SingleMask (one input image corresponds to one mask image), CorrespondingMask (all the input images corresponds to one mask image).

Instruction: To change the way the mask is used. You can choose to apply one mask to all polygons or apply different masks to each polygon accordingly.

Min Overlap Ratio (0–1.0)

Default value: 0.8000

Description: This parameter is used to adjust the threshold of the overlap area ratio with the mask. If the overlap area ratio between the polygon and the mask is greater than the threshold, the polygon is retained. If the overlap area ratio of the polygon and the mask is less than the threshold, the polygons outside the mask are filtered out.

4.3.312. Reorder According to Given Indices

The contents of this step will no longer be updated. Please use the new Step Reorder by Index List. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step sorts a list of values according to the input index list.

Usage Scenario

This Step is a general numerical sorting Step. No fixed usage scenarios.

Input and Output

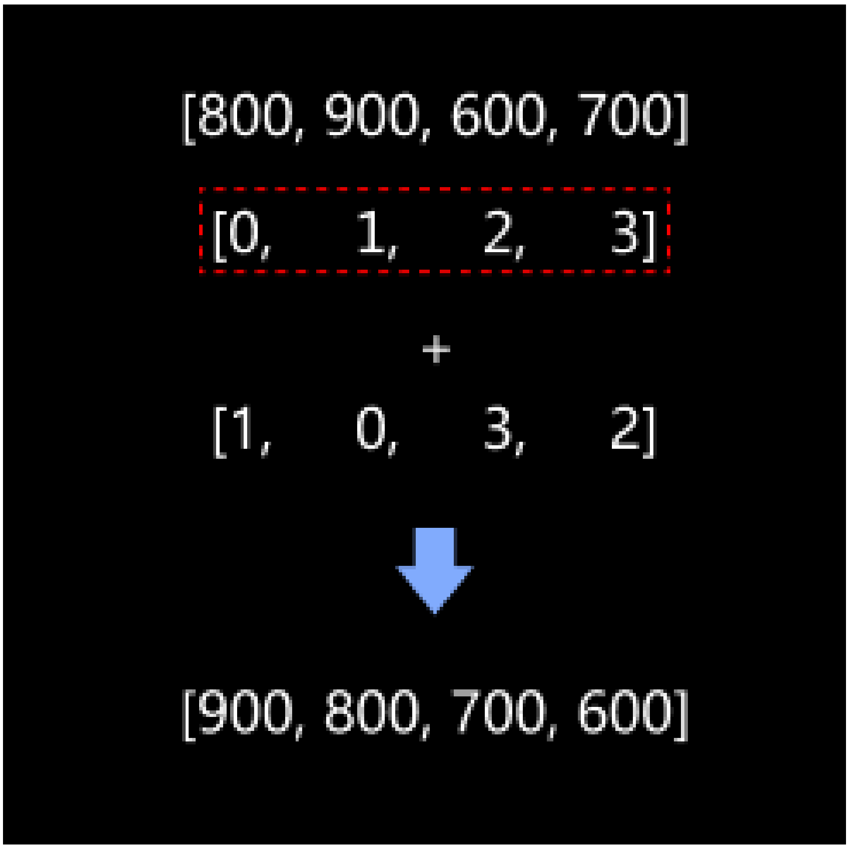
- Input:
  1. List of data input to this port will be sorted according to the input index list.
  2. Index list used to sort the input data.
- Output:
  1. Sorted data list

4.3.313. Reorder by Index List

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

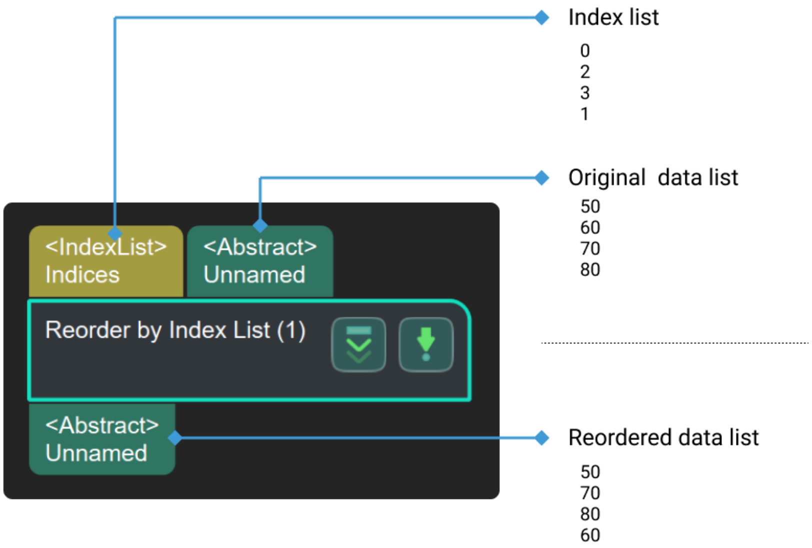
Reorder the original data according to the index.



Usage Scenario

Reorder the original data according to the input index. This Step usually follows Steps in the Sort category.

Input and Output



4.3.314. Repeat Data and Concatenate Copies

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

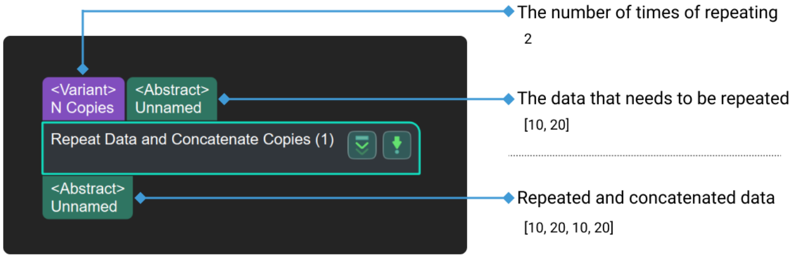
Function

Copy the input data by the set number of times, concatenate the copies, and output. For example, if the original data list is [10, 20] and it needs to be repeated 2 times, the output will be [10, 20, 10, 20].

Usage Scenario

A general data list modifying Step. Usually used with the Step Count Elements in Specified Dimension in Data List to make the numbers of elements input to a subsequent Step's ports equal.

Input and Output


The number of times of repeating
2
<Variant>
N Copies
<Abstract>
Unnamed
The data that needs to be repeated
[10, 20]
Repeat Data and Concatenate Copies (1)
<Abstract>
Unnamed
Repeated and concatenated data
[10, 20, 10, 20]

4.3.315. Replace Elements in List

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

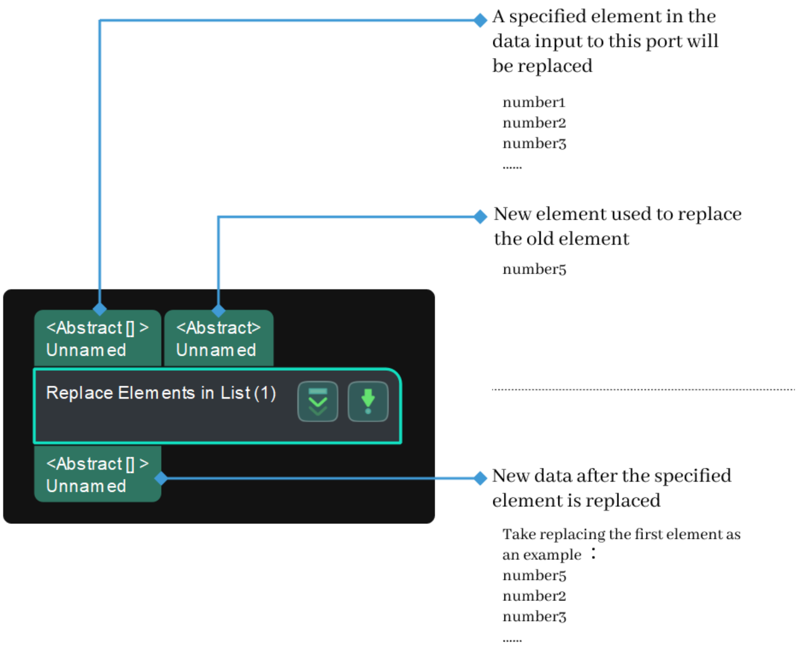
Function

This Step replaces elements in the input data. The dimension of the element to replace and the replacement method can be set in the parameters.

Usage Scenario

This Step is used by developers for testing. Regular users do not need to use this Step in general application scenarios.

Input and Output


A specified element in the data input to this port will be replaced
number1
number2
number3
......
New element used to replace the old element
number5
<Abstract[]>
Unnamed
<Abstract>
Unnamed
Replace Elements in List(1)
<Abstract[]>
Unnamed
New data after the specified element is replaced
Take replacing the first element as an example：
number5
number2
number3
......

4.3.316. Reverse List

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step arranges the elements in the input data list in reverse order.

Usage Scenario

This Step is a general element sorting Step. No fixed usage scenarios.

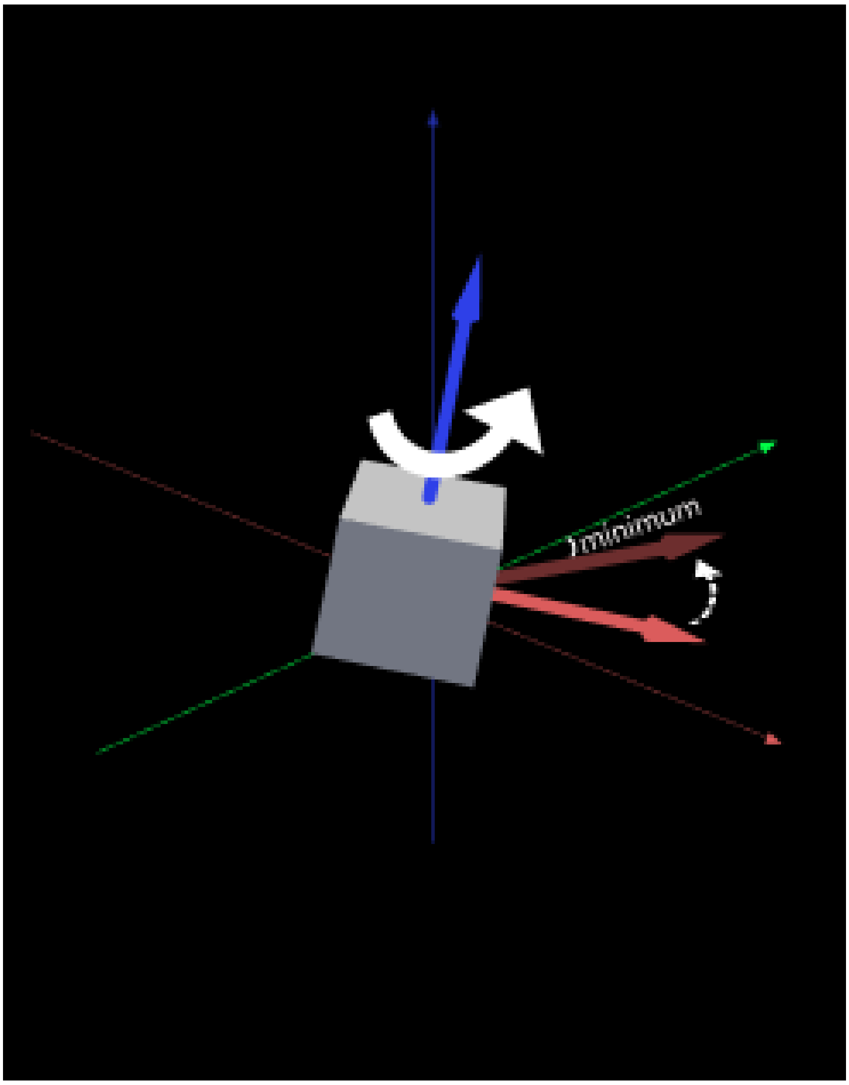
Input and Output

- Input:
  1. Data list input to this port will be reversed.
- Output:
  1. List of data in reverse order

4.3.317. Rotate Axis to Minimize Included Angle to Reference Direction

Function

Adjust axes of poses to the direction with the smallest angles to the reference directions around selected pose axes within symmetry constraints.

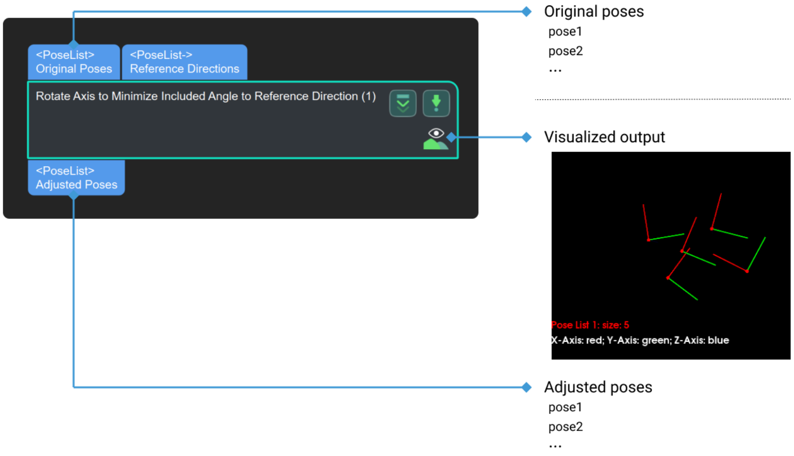


Usage Scenario

For various scenarios where pose orientations need to be adjusted.

This Step belongs to an earlier version. Please instead use the new Step Rotate Poses to Directions Freely that has a wider range of functions. Use it with Easy Create Vector3Ds.

Input and Output



Parameter Description

Visualization Settings

Pose Type to Visualize

Instruction: This parameter decides the type of pose to visualize.

Options:

- Before (pose before processing)
- After (pose after processing)
- All (all poses before and after processing)

Default setting: After.

Use Reference Point

Description: This parameter is used to select whether to use the base point to calculate reference direction. When the option is selected, you need to set the parameters in **Reference Point Settings**.

Default setting: unselected.

Instruction: Select the option only when you use the base point to calculate reference direction. The direction constructed starts from the input pose and points to the reference point.

Reference Point Settings

Use Relative Z

Description: This parameter decides whether to replace the Z value of the reference point. Z value of reference point = original pose Z value + relative value.

Default setting: unselected.

X

Description: the X coordinate value of the reference point.

Default value: 0.0000

Y

Description: the Y coordinate value of the reference point.

Default value: 0.0000

Z

Description: the Z coordinate value of the reference point.

Default value: 0.0000

Relative Z

Description: the relative translation value in the Z direction. When **Use Relative Z** is selected, you need to set this parameter.

Default value: 0.0000

Axis Settings

Fixed Axis

Description: This parameter specifies the axis that remains invariant during the pose rotation.

Options: X, Y, Z.

Default setting: Y

Rotated Axis

Description: This parameter is used to select the axis that rotates to a specified direction.

Options: X, Y, Z.

Default setting: Z

Reference Direction Settings

X

Description: the direction vector of the reference axis X.

Default value: 0.0000

Y

Description: the direction vector of reference axis Y.

Default value: 0.0000

Z

Description: the direction vector of the reference axis Z.

Default value: 1.0000

4.3.318. Rotate Images by Specified Poses

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step rotates 2D images to the specified orientations with the input 3D poses.

Usage Scenario

This Step is to rotate images corresponding to the objects after the objects are located, to position the object images where the objects are in the original image.

Input and Output

- Input:
  1. Object images input to this port will be rotated and placed at the specified positions.
  2. Object poses for positioning the images.
- Output:
  1. Positioned images

4.3.319. Rotate Poses around Axis by Angle

> Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step rotates the input poses by the specified angles around the specified axes.

Usage Scenario

This Step is a general pose rotation Step. No fixed usage scenarios.

Input and Output

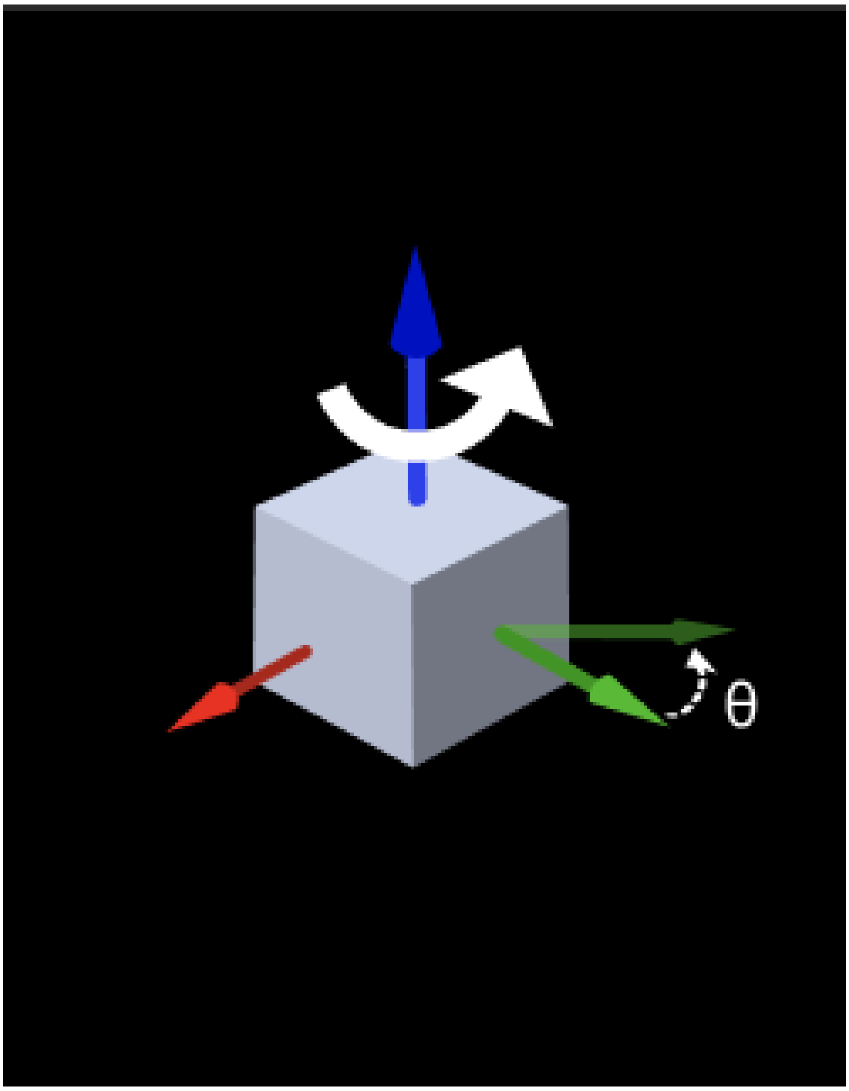
- Input:
  1. Poses input to this port will be rotated.
  2. Poses will be rotated around the axes input in this port.
  3. Rotation angles of the pose.
- Output:
  1. Rotated poses

4.3.320. Rotate Poses around Given Axis

> Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

Rotate poses around a specified axis.

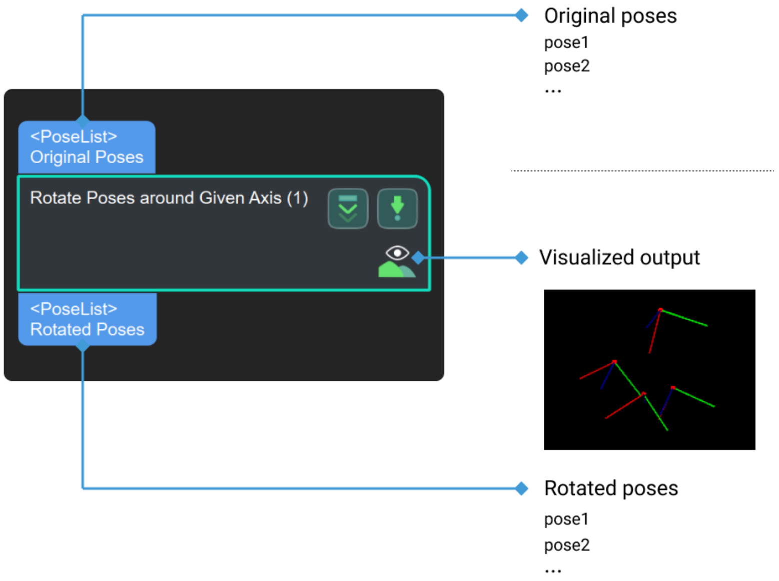


Usage Scenario

General pose rotation Step. No fixed usage scenarios.

This Step is outdated. Please use the new Step Rotate Poses around Axis by Angle instead and use it with the Step Easy Create Vector3Ds.

Input and Output



4.3.321. Rotate Poses Locally

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step rotates poses around specified axes in objects' reference frames.

Usage Scenario

This Step is a general pose rotation Step. No fixed usage scenarios.

Input and Output

- **Input**: Poses input to this port will be rotated.
- **Output:** rotated poses.

4.3.322. Rotate Poses to Directions Freely

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step rotates the input original poses to align the directions of the specified axes with the reference directions. The reference directions can be obtained from the input or can be set in parameters.

Usage Scenario

This Step is a general pose rotation Step. No fixed usage scenarios.

Input and Output

- Input:
  1. Poses input to this port will be rotated.
  2. Target orientations for rotation.
- Output:
  1. Rotated poses

4.3.323. Rotate Poses' Axes to Specified Directions under Symmetry Constraints

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

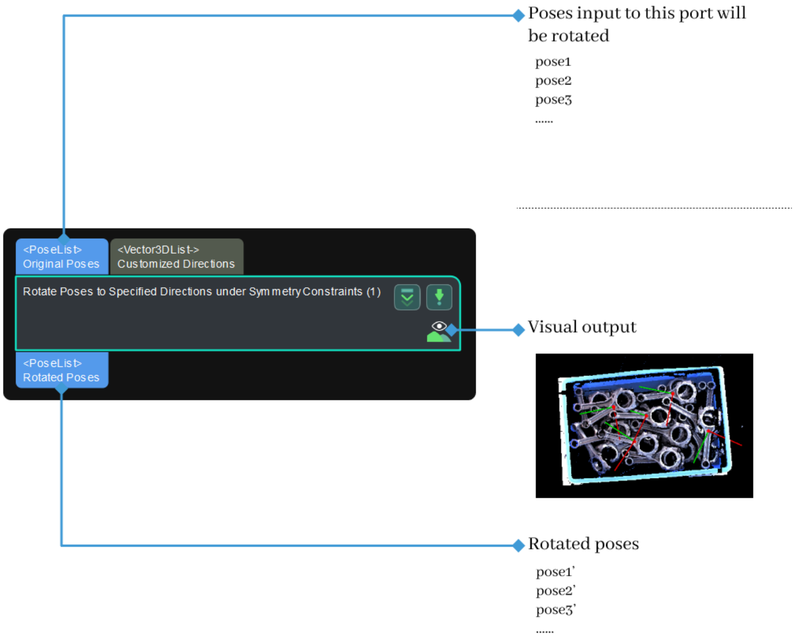
Function

This Step rotates the input original poses so that the specified axes are aligned to the reference directions. The object symmetry can be set in the parameters.

Usage Scenario

A general pose rotation Step. This Step has no fixed usage scenarios.

Input and Output

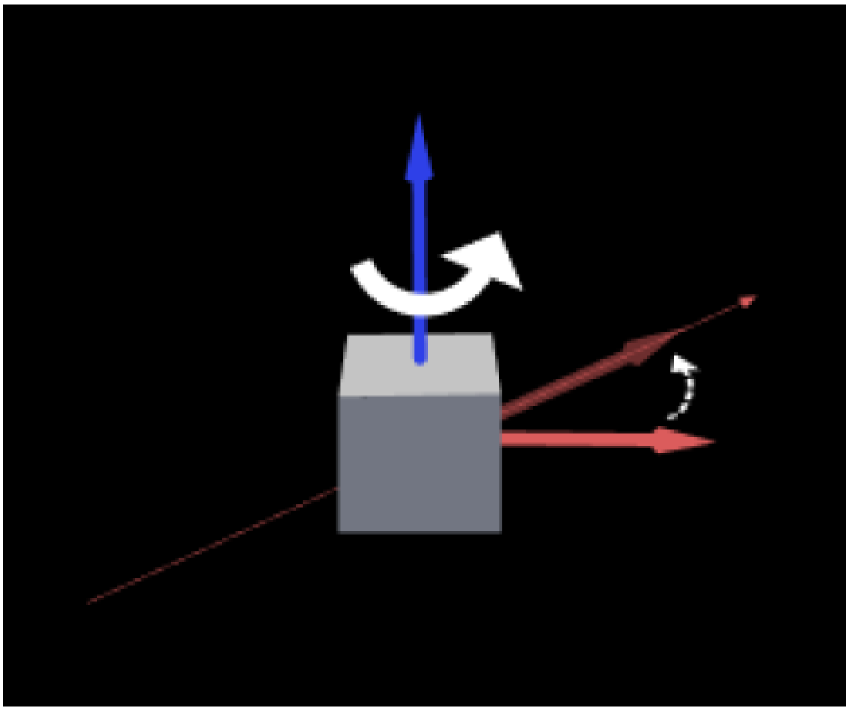


4.3.324. Rotate Poses' Axes to Specified Directions

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

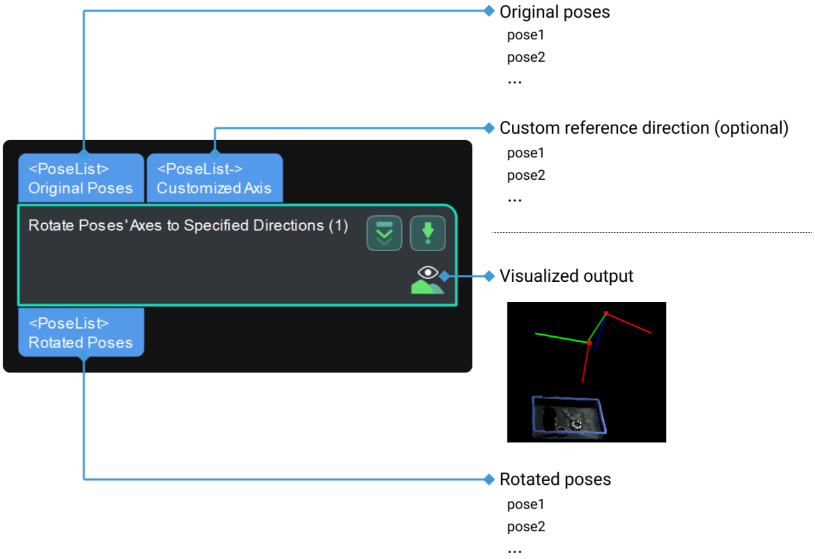
Rotate poses around fixed axes, aligning the specified axes with the reference direction.



Usage Scenario

This Step is a general pose rotation Step. No fixed usage scenarios.

Input and Output



4.3.325. Save Images

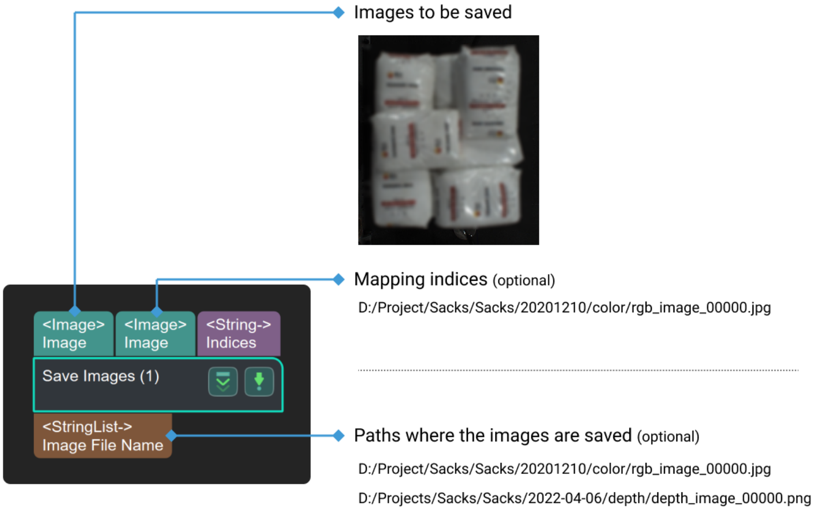
Function

Save the color images and depth maps to a specified path.

Usage Scenario

It is often used to save the original color images and depth maps from the camera and can also be used to save instance segmentation results. Please change the **save path**, **subfolder name**, etc. in the Parameters as needed.

Input and Output


Images to be saved
Mapping indices (optional)
D:/Project/Sacks/Sacks/20201210/color/rgb_image_00000.jpg
<Image> Image
<Image> Image
<String-> Indices
Save Images (1)
<StringList-> Image File Name
Paths where the images are saved (optional)
D:/Project/Sacks/Sacks/20201210/color/rgb_image_00000.jpg
D:/Projects/Sacks/Sacks/2022-04-06/depth/depth_image_00000.png

Parameter Description

Directory Settings

Save Path

Description: This parameter is used to set the path to save images.

Instruction: Please set the image saving path according to actual needs.

Target Name

Description: This parameter is used to create a subfolder named after the target.

Default value: null.

Name of Subdirectories

Description: Subfolders can save images in different formats. To create it, use a semicolon between subfolder names; for instance, color; depth.

Format: XXXX;XXXX;XXXX;XXXX

Default value: depth;color

Date Related Settings

Create Subdirectories by Date

Description: This parameter is used to determine whether to create a subfolder named with the date.

Default value: selected.

When **Create Subdirectories by Date** is not selected, you do not need to set **Max Subdirectories' Number** and **Delete Expired Subdirectories**.

Max Subdirectories' Number

Description: This parameter is used to set the maximum number of days a file can be kept.

Default value: 7 days

Delete Expired Subdirectories

Description: This parameter is used to determine whether to delete the folders that have been retained for more than **Max Subdirectories' Number**(the maximum retention days).

Default value: selected.

Save Contents Settings

Save Camera Parameters

Description: This parameter is used to determine whether to save camera parameters to the folder. If this parameter is checked, camera parameters will be saved along with the captured images.

Default setting: selected.

Instruction: It is recommended to use the default value.

Save Flange pose

Description: This parameter is used to determine whether to save flange poses to the folder. If this parameter is checked, flange poses will be saved along with the captured images.

Default setting: selected.

Instruction: It is recommended to use the default value.

Not Save If Using Virtual Camera

Description: This parameter is used to determine whether to save data from the virtual camera. When checked, the input images will not be saved if a virtual camera is used in the project.

Default setting: selected.

Instruction: It is recommended to use the default value.

Image Index Settings

Overwrite Images with Same Name

Description: This parameter is used to set whether to automatically overwrite images with the same name. If checked, images with the same name will be automatically overwritten; if not checked, a prompt will appear asking whether to overwrite the images with the same name.

Default value: selected.

Max Images' Number

Description: This parameter is used to set the maximum number of images to be saved. If the number of images exceeds the maximum value, the next image index will be reset to 0.

Default value: 1000

Value range: [1, +∞)

Custom Start Index

Description: This parameter is used to define the start index for saving images.

Default value: −1

File Name Settings

Save Color Images in PNG Format

Description: This parameter is used to determine whether to save the color image in the **PNG** format.

Default setting: unselected.

Custom Prefix of Image Files

Description: This parameter is used to set the prefix to the name of image files.

Default value: null.

4.3.326. Save Regions around Poses as 3D ROIs

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step saves the regions near the input poses as a 3D ROI file.

Usage Scenario

This Step is a legacy 3D ROI format file generation Step.

This Step is outdated. Please use new Steps related to 3D ROI instead.

Input and Output

- **Input:** Regions with the center as the input poses of this port will be saved as a 3D ROI file.
- **Output:** none.

4.3.327. Save Result to XML File

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step saves the output of other Steps to an XML file. You can set the file name in the field of File Path.

Usage Scenario

This Step is a custom Step for special scenarios.

Input and Output

- Input:
  1. Poses input by this port will be saved.
  2. Poses input by this port will be save.
  3. Strings input by this port will be saved.
  4. Numbers input by this port will be saved.
  5. Strings input by this port will be saved.
  6. String input by this port will be saved.
- Output:
  1. None.

4.3.328. Save Results to File

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

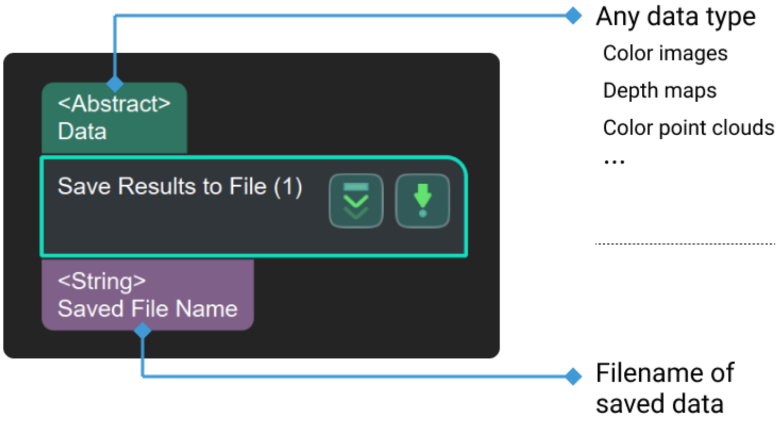
Function

Save project data to local files.

Usage Scenario

This Step saves data of all types, but if you need to save a 2D image, it is recommended to use the Step Save Images instead.

Input and Output



4.3.329. Save Path Targets

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step saves path targets to a JSON file.

Usage Scenario

This Step is a general path target saving Step. No fixed usage scenarios.

Input and Output

- **Input:** Path targets input at this port will be saved.
- **Output:** none.

4.3.330. Save Step Parameters to File

Function

Save the parameters of a specified Step to a local file.

Usage Scenario

Export the parameters of other Steps. This Step must follow the Save Images Step.

Input and Output

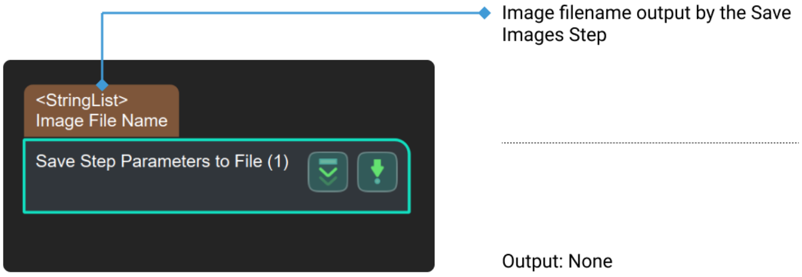
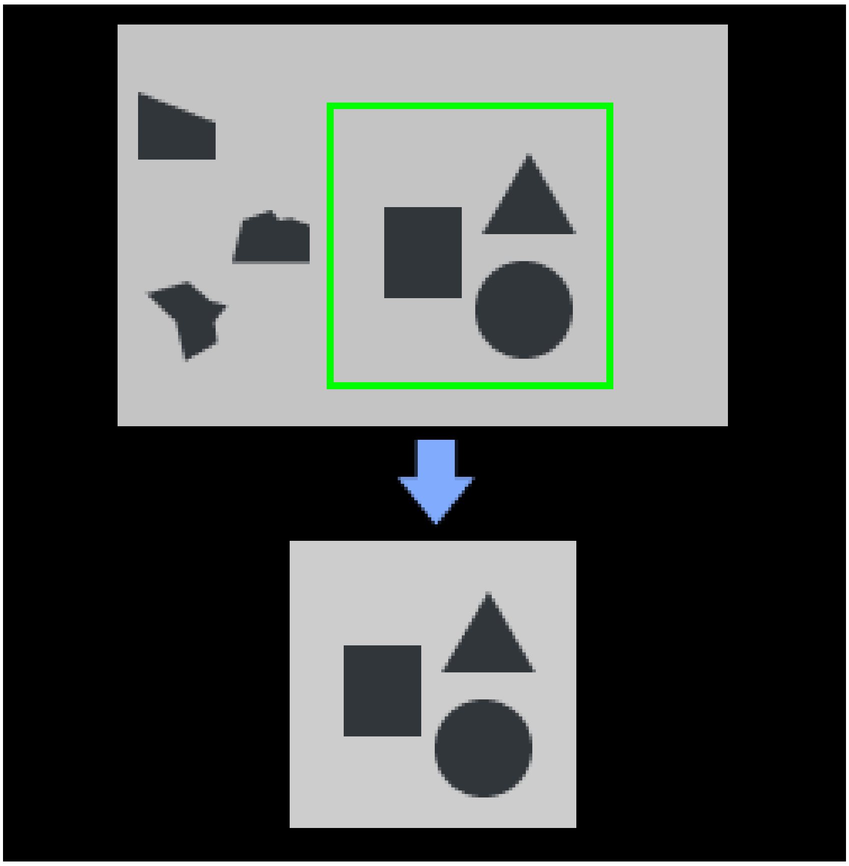


Image filename output by the Save Images Step
<StringList>
Image File Name
Save Step Parameters to File (1)
Output: None

4.3.331. Scale Image in 2D ROI

Function

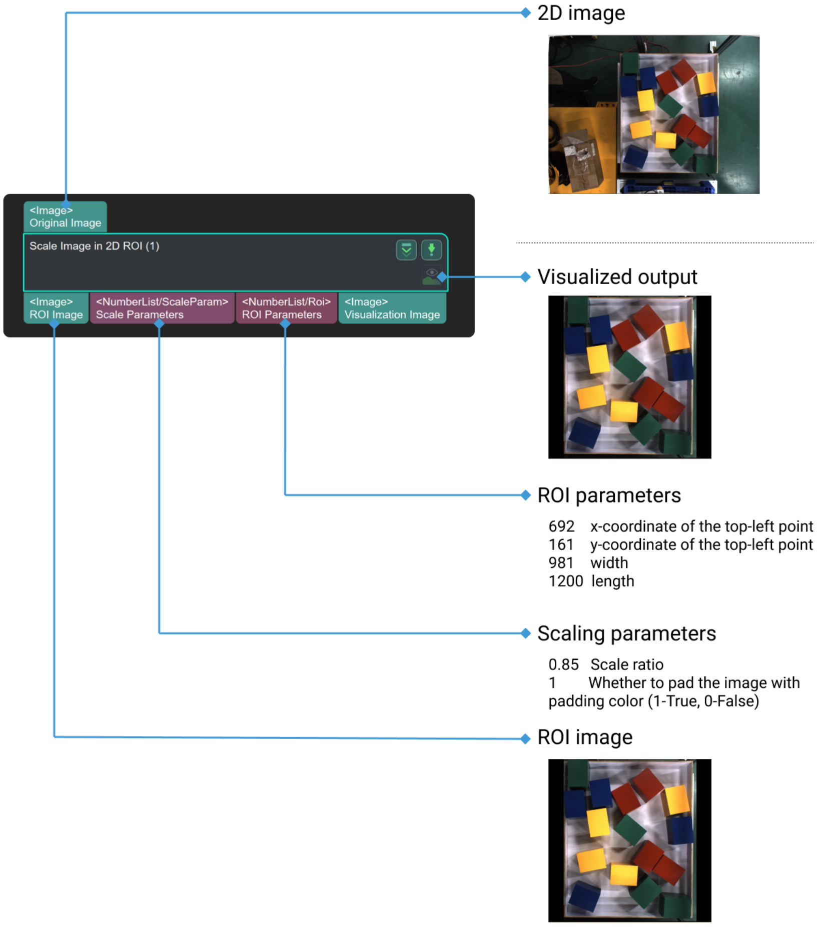
Scale the region of interest (ROI) in an image to a specified size.



Usage Scenario

It is usually used for image preprocessing in deep learning related Steps and is usually used with Recover Scaled Images in 2D ROI and deep learning related steps.

Input and Output



692	x-coordinate of the top-left point
161	y-coordinate of the top-left point
981	width
1200	length

0.85	Scale ratio
1	Whether to pad the image with padding color (1-True, 0-False)

Instruction

This Step is used to scale images in the ROI into a designated size by setting the parameter **Color ROI Settings** and **Ideal Destination Resolution**.

Auto Scale

To auto scale the ROI, please follow these steps:

1. Select the checkbox **Auto Scale** under **Color ROI Settings**.
2. Select the checkbox **Same as Input Color ROI & Skip Scaling** under **Ideal Destination Resolution**.

Output Initial ROI Image

To output the initial ROI image, please follow these steps:

1. Disable **Auto Scale** under **Color ROI Settings**.
2. Set 1 as the value of **Customized Scale**.
3. Select the checkbox **Same as Input Color ROI & Skip Scaling** under **Ideal Destination Resolution**.

Scale ROI Image by a Particular Degree

To output ROI image of a particular size, please follow these steps:

1. Disable **Auto Scale** under **Color ROI Settings**.
2. Set **Customized Scale** according to your need. When the value of this parameter is higher than 1, image will be magnified. If the value is lower than 1, image will be reduced.
3. Select the checkbox **Same as Input Color ROI & Skip Scaling** under **Ideal Destination Resolution**.

Output ROI Image of Designated Width and Height

To out put ROI image of designated width and height, please follow these steps:

1. Disable the **Same as Input Color ROI & Skip Scaling** under **Ideal Destination Resolution**.
2. Set the value of **Width** and **Height** according to your need.
3. Choose whether to select **Padding** according to your need.

> Please continue your reading to view the description of parameters.

Parameter Description

Color ROI Settings

Description: This parameter decides the way to set color ROI.

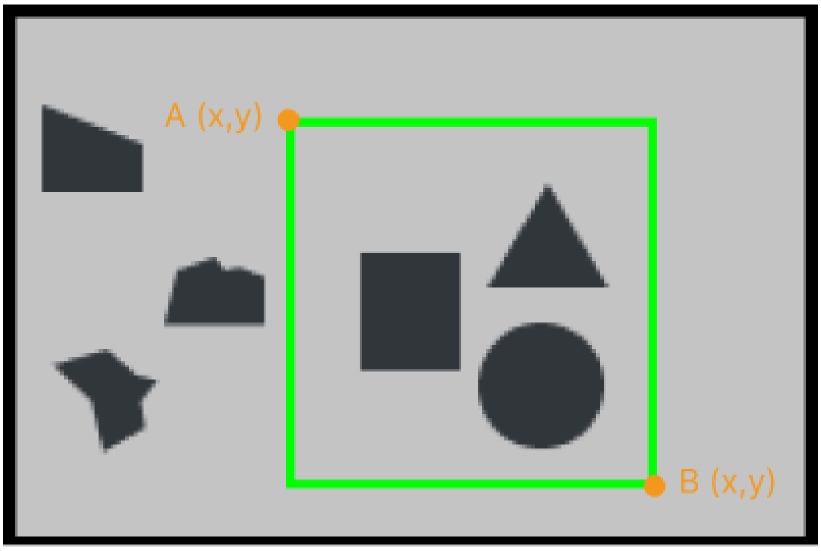
Options:

- ROIByFile: The default setting. Set and identify ROI in a visualize way according to the coordinates of the top left vertex and bottom right vertex.
- ROIByParam: Set the ROI by configuring the parameters manually, including the coordinate of the top left vertex and the width and height of the ROI.

ROIByFile

Color ROI File

Description: This parameter sets the ROI of color image. Please refer to Set 2D ROI for more information. In the image below, point A is the top left vertex, and B is the bottom right vertex.



The coordinate of the upper left corner of the original image is (0, 0).

ROIByParam

Start X

Description: the x-coordinate of the ROI’s top left vertex.

Default value: 0

Instruction: Start X + Width < Initial image width

Start Y

Description: the y-coordinate of the ROI’s top left vertex.

Default value: 0

Instruction: Start Y + Height < Initial image height

Width

Instruction: This parameter refers to the width of the ROI.

Default value: 0

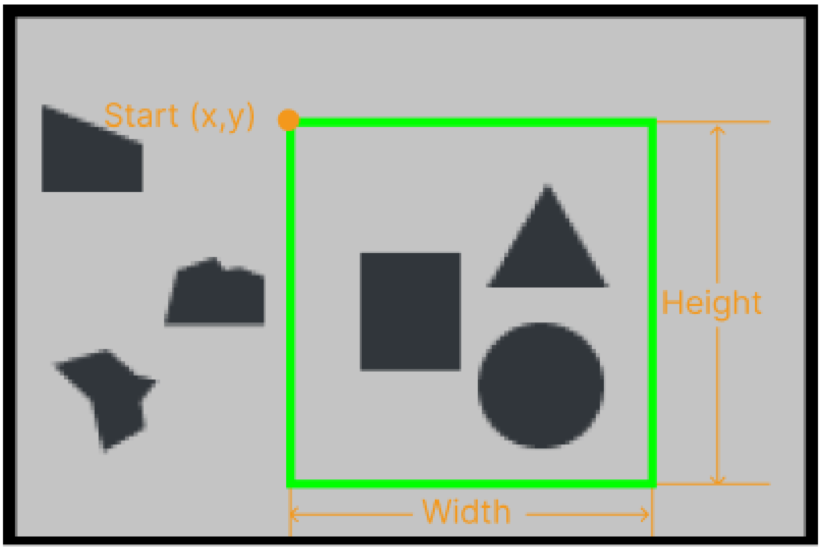
Instruction: Start X + Width < Initial image width

Height

Instruction: This parameter refers to the height of the ROI.

Default value: 0

Instruction: Start Y + Height < Initial image height



The coordinate of the upper left corner of the original image is (0, 0).

Method to Update Color ROI

Description: This parameter decides the type of ROI.

Options:

Origin: the initial color image of the ROI.

AdaptivemodifiedRoiOrigin: Use the border of the image to adjust the color ROI and adaptively adjust the center of the non-zero point area of the image to the center of the ROI.

BoundRect: Extract the bounding rectangle area of the input image and replace the original image with the circumscribed rectangle of the non-zero point of the image to calculate the zoom ratio.

Color ROI Scaling

Auto Scale

Description: This parameter automatically adjusts the size of the ROI image.

Default setting: selected.

Customized Scale

Description: This parameter designates the degree to which the ROI is scaled.

Instruction: Scaled image size = Initial ROI size × Customized scaling factor. As shown below, the image on the left is of the initial size, with a customized scaling factor of 0.5.

You can set the parameter only when **Auto Scale** is not selected.

Ideal Destination Resolution

Same as Input Color ROI & Skip Scaling

Description: This parameter decides whether the size of the image output is set the same as that of the image after **Color ROI Scaling** setting.

Default setting: unselected.

> When this checkbox is not selected, the size of the image output will be scaled.

Width/Height

Description: This parameter sets the width and height of the ROI (in px).

Default value: 1024 px

Tuning recommendation: Please set the parameter according to the actual situation.

> You can set the parameter only when **Ideal Destination Resolution** is not selected.

Padding Color

> - **Padding** is needed only when the ROI width and height are customized.
> - The value of R, G, and B can be set only when the **Padding** checkbox is selected.

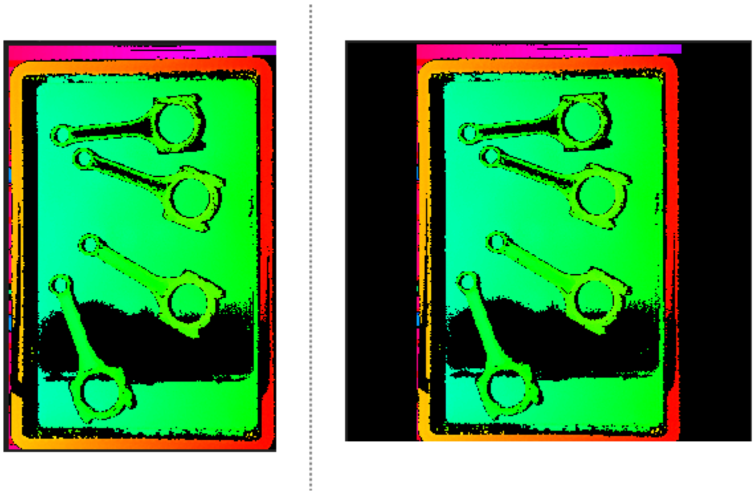
Padding

Description: This parameter is used to fill the extra space of the ROI when it is scaled up to a particular size.

Default setting: selected.

Instruction: It is recommended to fill the space with black (0,0,0) or grey (128,128,128).

Adjustment Example: The image on the left below is the scaled-up ROI without padding. When padding is used, the size of the ROI image must be designated, and the extra space will be filled up by color, as the image on the right side shows below.



R/G/B

Description: Color image is made up of three colors: R, G, and B. Different R/G/B values can fill the image with different colors.

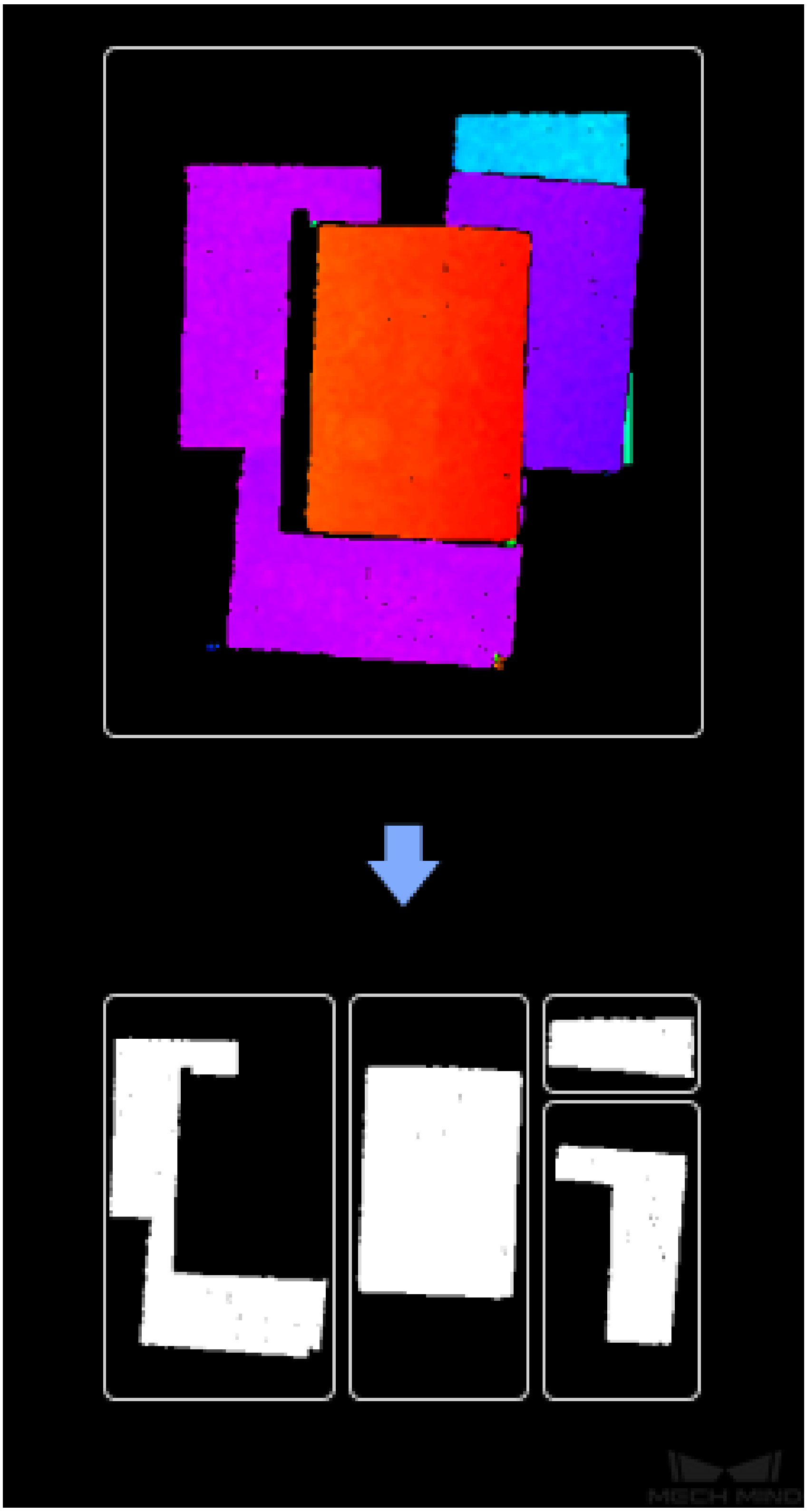
Default value: 0

Value range: 0–255.

4.3.332. Segment Depth Map

Function

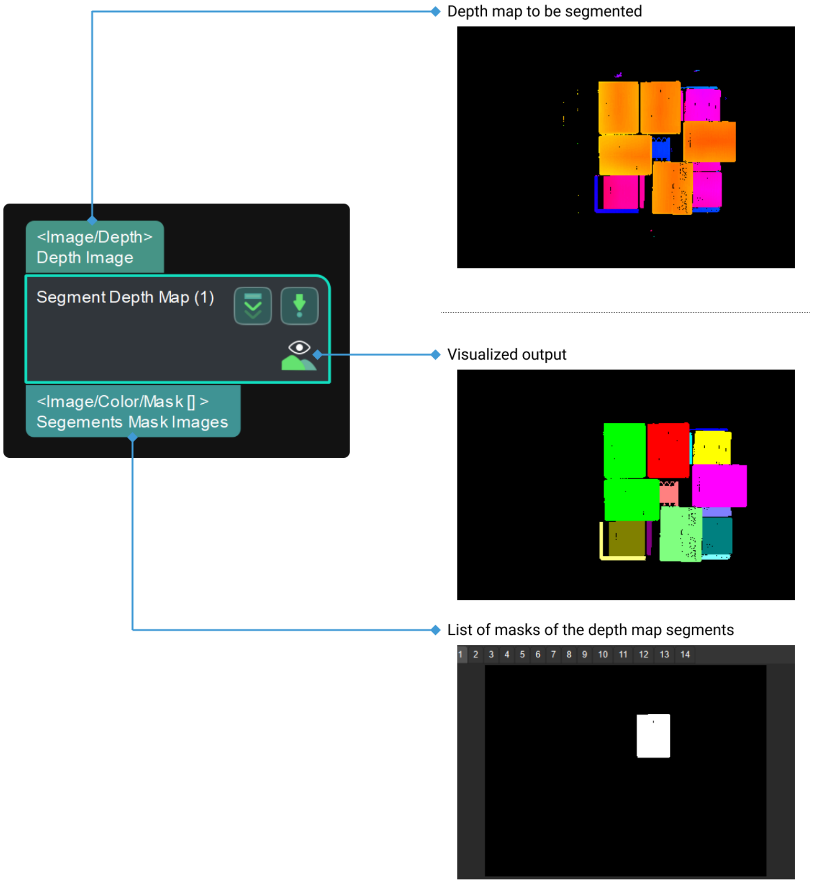
This Step is used to segment depth map according to specified rules and output areas that meet the rules.



Usage Scenario

This Step is widely used for depth map processing and is usually connected with Steps such as Invalidate Depth Pixels outside 3D ROI, Get Highest Layer Regions in Depth Map, and Merge Mask Images.

Input and Output



Parameters

Segmentation Settings

Max Height Diff between Neighbouring Pixels

Description: This parameter is used to determine the maximum tolerance of depth difference (in mm) between neighboring pixels in the process of segmenting the depth map. When the depth difference between neighboring pixels is less than the set value, the two pixels will be considered as belonging to the same region; otherwise, the pixels will be considered as belonging to different segmented areas. The larger the value, the more segmented areas.

Default value: 7.000 mm

Tuning recommendation: Please set the value according to the actual situation.

Min Segment Area

Description: This parameter specifies the minimum size of the segmented areas (in pixel). The segmented areas smaller than this value will be discarded. The smaller the value, the more segmented areas.

Default value: 1500 px

Tuning recommendation: Please set the value according to the actual situation.

Depth ROI File Name

Description: This parameter is used to select an ROI on the color image.

Tuning recommendation: Please refer to Instructions for Setting 2D ROI for detailed instructions.

4.3.333. Send Point Cloud to External Service

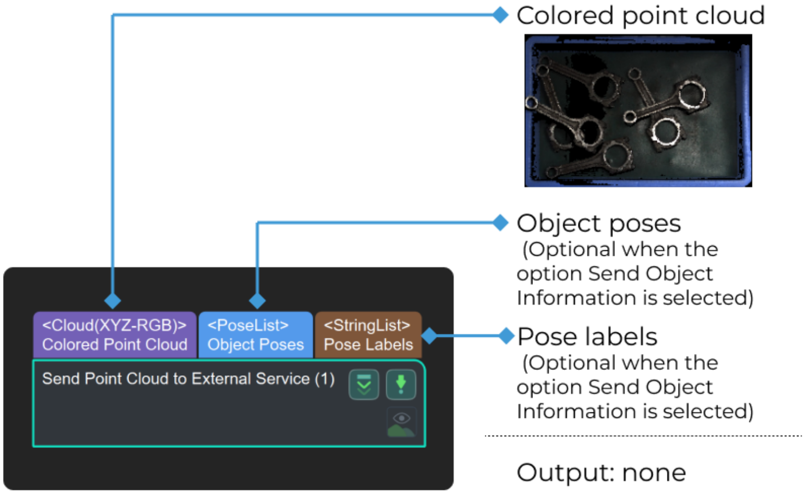
Function

Send point cloud information to Mech-Viz.

Usage Scenario

Send point cloud to Mech-Viz for simulation and debugging or for checking the actual point cloud processing results.

Input and Output


Colored point cloud
Object poses
(Optional when the option Send Object Information is selected)
Pose labels
(Optional when the option Send Object Information is selected)
Output: none
<Cloud(XYZ-RGB)> Colored Point Cloud
<PoseList> Object Poses
<StringList> Pose Labels
Send Point Cloud to External Service (1)

Parameter Description

Cloud Settings

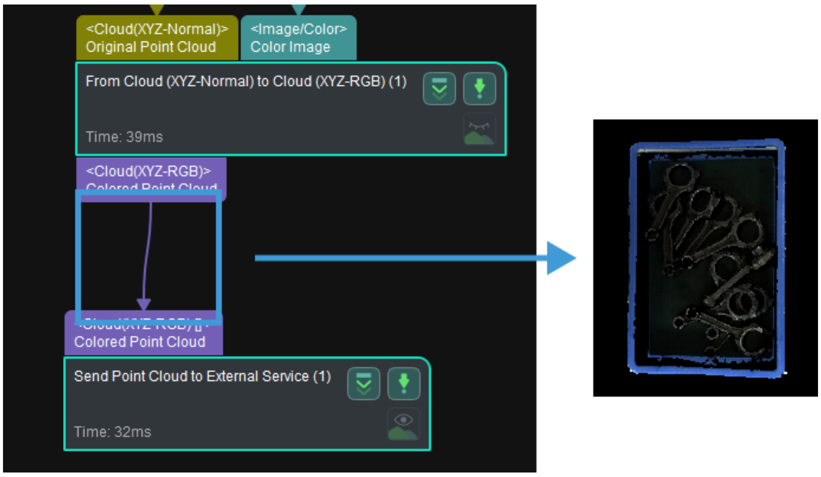
Input Point Cloud Type

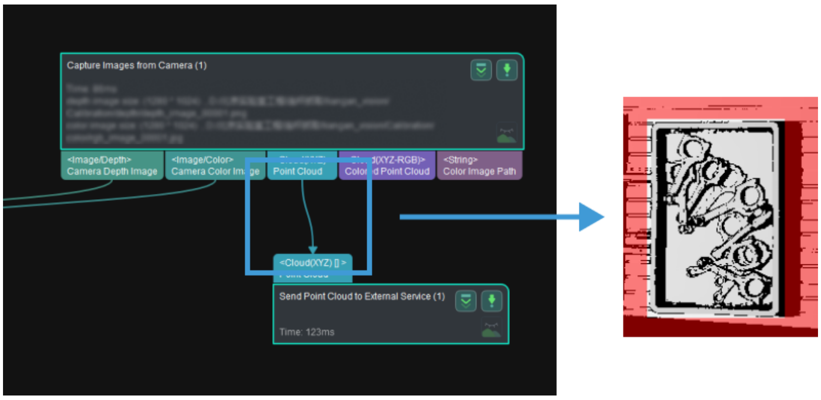
Default setting: CloudXYZRGB.

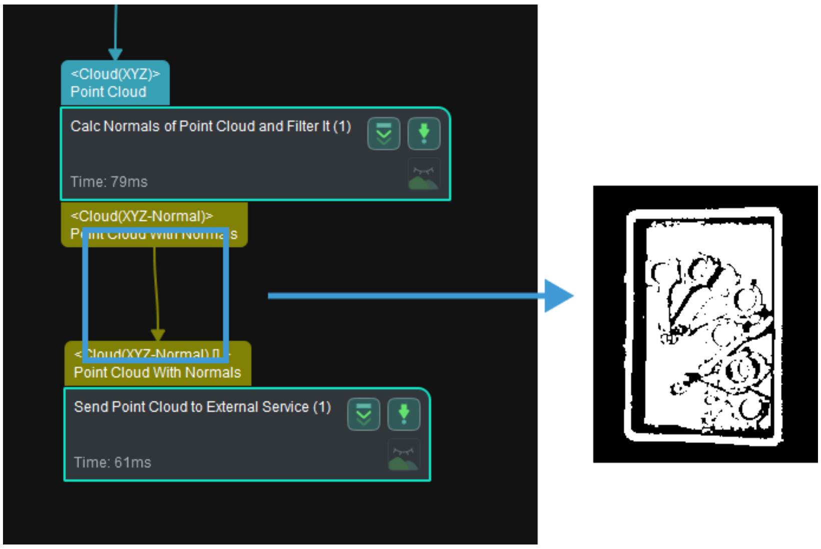
Options: CloudXYZRGB, CloudXYZ, and CloudNORMAL.

Instruction: Select the point cloud type to be sent according to the actual requirement.

Example: When CloudXYZRGB, CloudXYZ, and CloudNORMAL are selected, respectively, the input point clouds are shown below.







Object Information Settings

Send Object Information

Default setting: unselected.

Instruction: When this option is not selected, **Object Poses** and **Pose Labels** ports will not appear. When this option is selected, **Object Poses** and **Pose Labels** ports will be available. Please select if you need to send **Object Poses** and **Pose Labels** to Mech-Viz. The figure below shows the Step when this option is selected and unselected.

[send point cloud to external service send object information] | *send-point-cloud-to-external-service-send_object_information.png*

Input Point Cloud in Camera Frame

Default value: selected.

Instruction: Once this option is selected, the input point cloud will be converted to the robot reference frame and then sent to Mech-Viz. If the option is not selected, the input point cloud in the camera reference frame will be sent to Mech-Viz directly.

4.3.334. Set Pose Quaternions

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

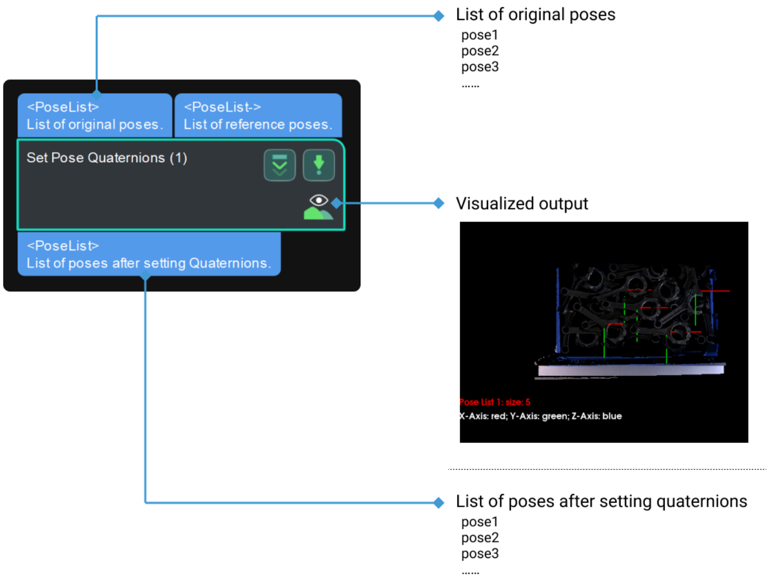
Function

Set the orientations of poses (in quaternion) by the reference pose input from the second port, or the quaternion set in the parameter.

Usage Scenario

A general pose quaternion setting Step. If the input reference pose list contains multiple poses, only the quaternion of the first pose will be used.

Input and Output



4.3.335. Set Pose Translation

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

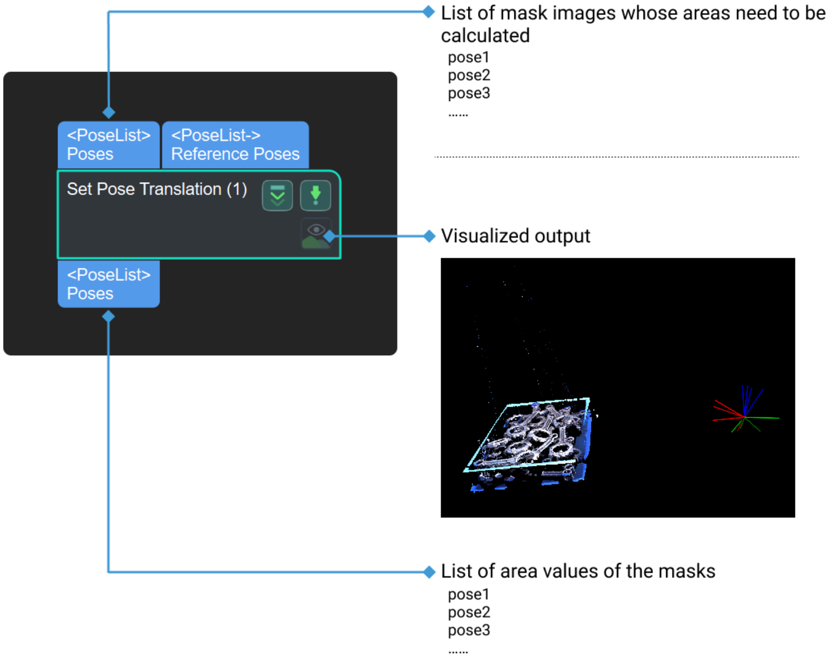
Function

Set the translation vectors of the input poses. The translation vectors can be either from the input reference poses or from the parameter Reference Direction.

Usage Scenario

A general pose translation vector setting Step. If the input reference pose list contains multiple poses, only the translation vector of the first pose will be used.

Input and Output


List of mask images whose areas need to be calculated
pose1
pose2
pose3
……
<PoseList> Poses
<PoseList-> Reference Poses
Set Pose Translation (1)
Visualized output
<PoseList> Poses
List of area values of the masks
pose1
pose2
pose3
……

4.3.336. Show Images

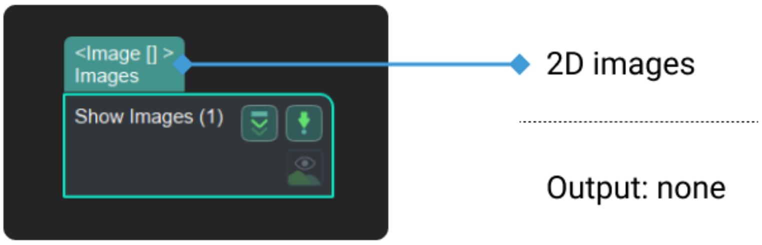
Function

This Step is used to display 2D images output by other Steps. The input Steps can output 2D images, including some Steps in the **2D Matching**, **2D Feature Detector**, and **2D General Processing** categories.

Usage Scenario

You can tune parameters of previous Steps according to the results of output images until the output results meet requirements.

Input and Output



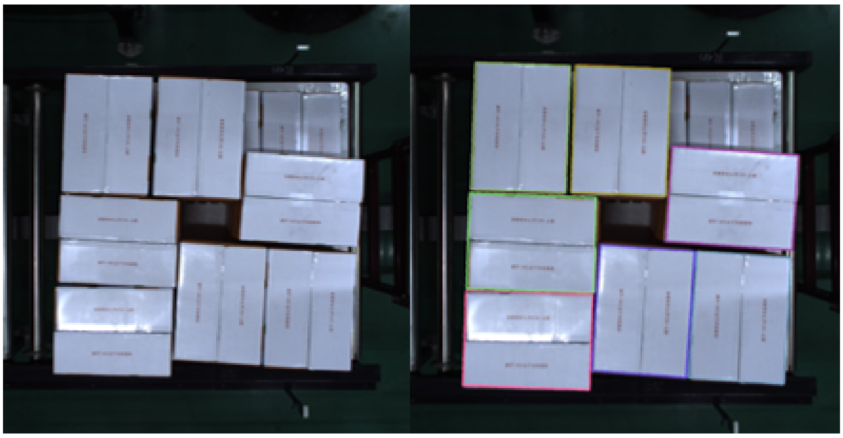
Parameter Description

Display Type

Default setting: MultiImages

Options: MultiImages, ImgAndPolyVerts

Instruction: When **MultiImages** is selected, only images will be displayed, as shown in the figure on the left. When **ImgAndPolyVerts** is selected, both images and candidate polygons will be displayed, as shown in the figure on the right.



4.3.337. Show Point Clouds and Poses

Function

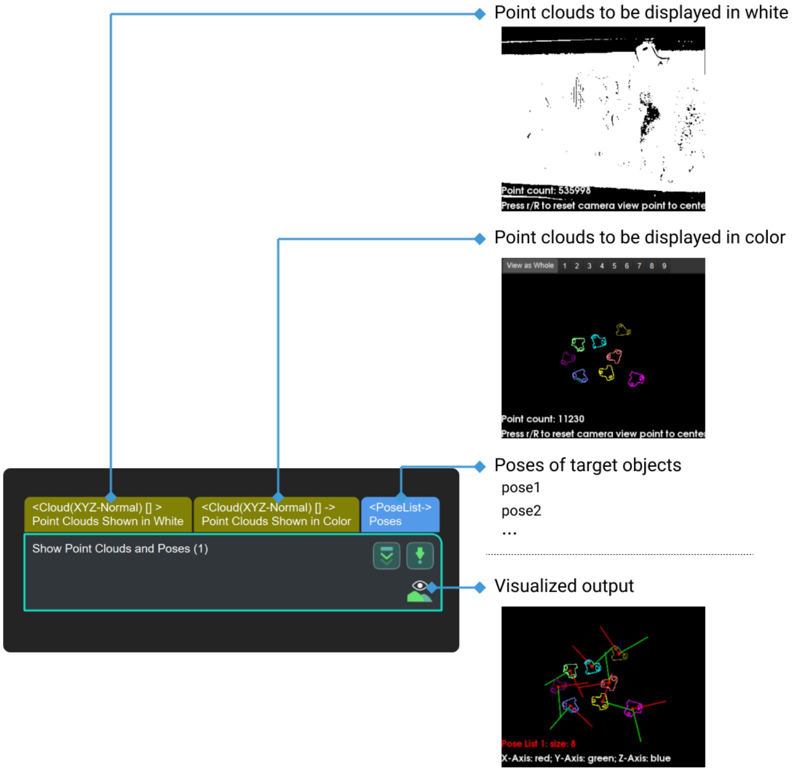
Display Point Clouds and Poses.

> Comparison of Similar Steps
>
> The Step **Show Point Clouds and Poses** is used to show point clouds and poses while the Step Show Images is used to show color images and masks.

Usage Scenario

This Step can display the relative position between the point cloud(s) and pose(s). For example, in a picking application, it can be used to check if the poses are correct. It is usually used in debugging phase.

Input and Output


Point clouds to be displayed in white
Point count: 535998
Press r/R to reset camera view point to cente
Point clouds to be displayed in color
View as Whole 1 2 3 4 5 6 7 8 9
Point count: 11230
Press r/R to reset camera view point to cente
Poses of target objects
pose1
pose2
...
Visualized output
Pose List 1: size: 8
X-Axis: red; Y-Axis: green; Z-Axis: blue
<Cloud(XYZ-Normal) [] >
Point Clouds Shown in White
<Cloud(XYZ-Normal) [] ->
Point Clouds Shown in Color
<PoseList->
Poses
Show Point Clouds and Poses (1)

Parameter Description

Normal Vector Visualization Settings

Show Normals

Description: Select to show normals of the point cloud.

Default setting: unselected.

Instruction: Once this parameter is selected, the normals of the point cloud will be shown in the visualized result. Please refer to the tuning example for the corresponding result.

Display Interval of Normals

Description: This option will only be available when the checkbox of **Show Normals** is selected.

Default value: 20

Instruction: It is used to adjust the interval of normals on the point cloud, and the unit is millimeters. The higher the value, the sparser the displayed normals. Please refer to the tuning example for the corresponding result.

Z Value Visualization Settings

Z Value Visualization Settings

Description: Select to display the point clouds in different grayscales according to the Z value.

Default setting: unselected.

Instruction: When this parameter is selected, point clouds can be displayed in different grayscales according to the Z value.

Upper Bound

Description: Maximum Z value of point clouds to be colored. This option will only be available when the checkbox of **Z Value Visualization Settings** is selected.

Default value: 0

Instruction: This value denotes the maximum Z value of point clouds to be colored. The higher the value, the darker and less obvious the point cloud.

Lower Bound

Description: Minimum Z value of point clouds to be colored. This option will only be available when the checkbox of **Z Value Visualization Settings** is selected.

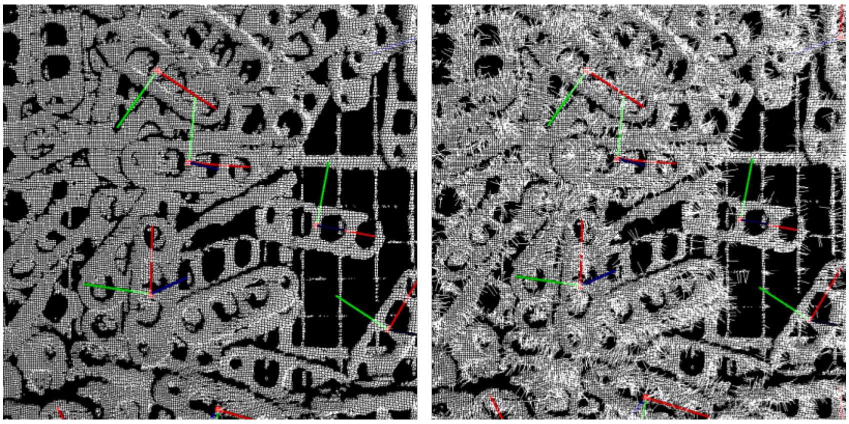
Default value: 0

Instruction: This value denotes the minimum Z value of point clouds to be colored. The smaller the value, the more obvious the point cloud.

Tuning Examples

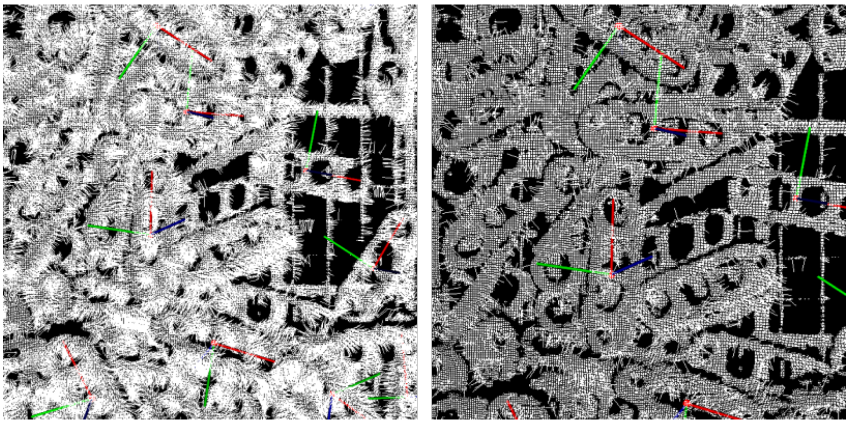
Show Normals of Point Clouds

The figure on the left is the case when normals are not displayed while that on the right is the case when normals are displayed, and the display interval of normals is 20 mm.



Change Display Interval of Normals

The figure on the left shows the case when the display interval of normals is 5 mm, where normals are dense, while that on the right shows the case when the interval is 40 mm, where normals are sparse.



4.3.338. Smooth Depth Map

This Step is outdated. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step smooths depth map with a specified method.

Usage Scenario

This Step is usually used when the point cloud has great fluctuations.

Input and Output

- **Input**: Depth map input to this port will be converted to a color image.
- **Output:** smoothed depth map.

4.3.339. Smooth Path

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step smooths the motion path consisting of the input list of poses.

Usage Scenario

This Step is used to prevent the robot motion from fluctuating too violently.

Input and Output

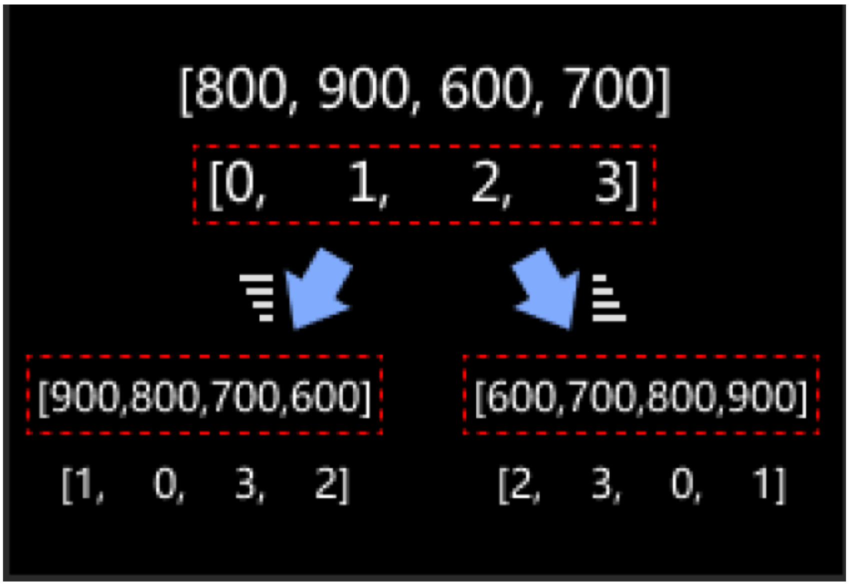
- **Input:** Motion path composed of target list input by this port will be smoothed.
- **Output:** list of targets of the motion path after smoothing.

4.3.340. Sort List and Output Index List

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

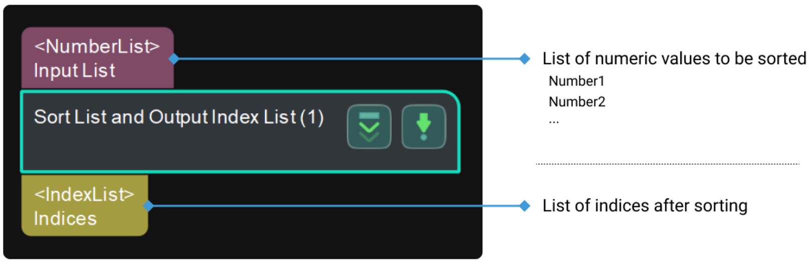
Sort the input list in ascending or descending order and output the index list according to the sorting.



Usage Scenario

This Step is used in scenarios where various types of data need to be sorted in ascending/descending order, usually used with the Step Reorder by Index List.

Input and Output


<NumberList>
Input List
Sort List and Output Index List (1)
<IndexList>
Indices
List of numeric values to be sorted
Number1
Number2
...
List of indices after sorting

4.3.341. Sort and Stratify

Function

This Step sorts and stratifies the input data according to the specified rules, moves the original indices of the data according to the sorting and stratifying process, and outputs the processed data and indices, and sort and stratification indices.

If the multidimensional data is input, each array of the last dimension of the data will be processed independently in turn.

Usage Scenario

This Step is used for multidimensional sorting. This Step can be used with other Steps to sort the specified dimension of the data.

Input and Output

The input data to be processed is shown in the table below:

Data to Be Processed	\[[30, 10, 20], [20, 10]]
Original Indices of Data	\[[4, 5, 6], [3, 2]]

The output data after processing is shown in the table below:

Processed Data	[\[[10], [20], [30]], \[[10], [20]]]
Processed Data Indices	[\[[5], [6], [4]],\[[2], [3]]]
Sort and Stratification Indices	[\[[1], [2], [0]], \[[1], [0]]]

Parameter Description

Ascending

Description: This parameter specifies whether to sort the data by ascending or descending order. Select to sort the data in ascending order, or else in descending order.

Default value: selected.

Layer Interval

Description: This parameter specifies the layer interval for the stratification. The unit is millimeters.

Default value: 50.0 mm

Starting Point of Each Stratum

Description: This parameter specifies the starting point of each stratum for the stratification.

Options: End of Previous Stratum Range, First Number of This Stratum

- End of Previous Stratum Range: The ranges of the strata are connected end to end and do not depend on the data. There will be no spacing between strata.
- First Number of This Stratum: Where the range of each stratum lies depends on the data. There will be spacing between strata, and the spacing will be larger if the numbers in the data are wide apart.

Default value: End of Previous Stratum Range

4.3.342. Sort Point Clouds

This Step is outdated. Please use the new Step Calc Specified Property of Point Clouds instead. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

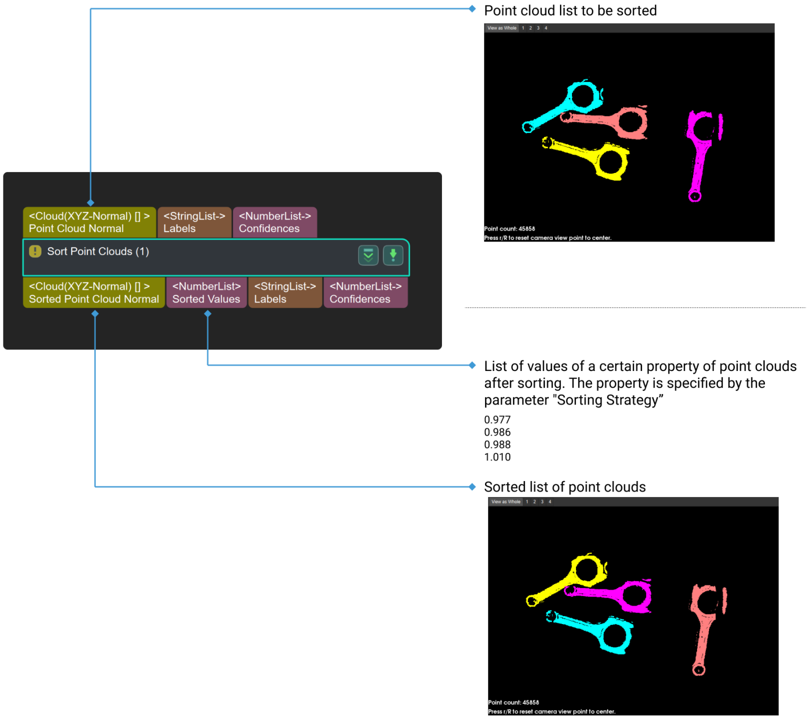
Function

This Step sorts the point clouds by specified rules, and outputs the corresponding labels and confidences.

Usage Scenario

This Step is used to set the order for picking.

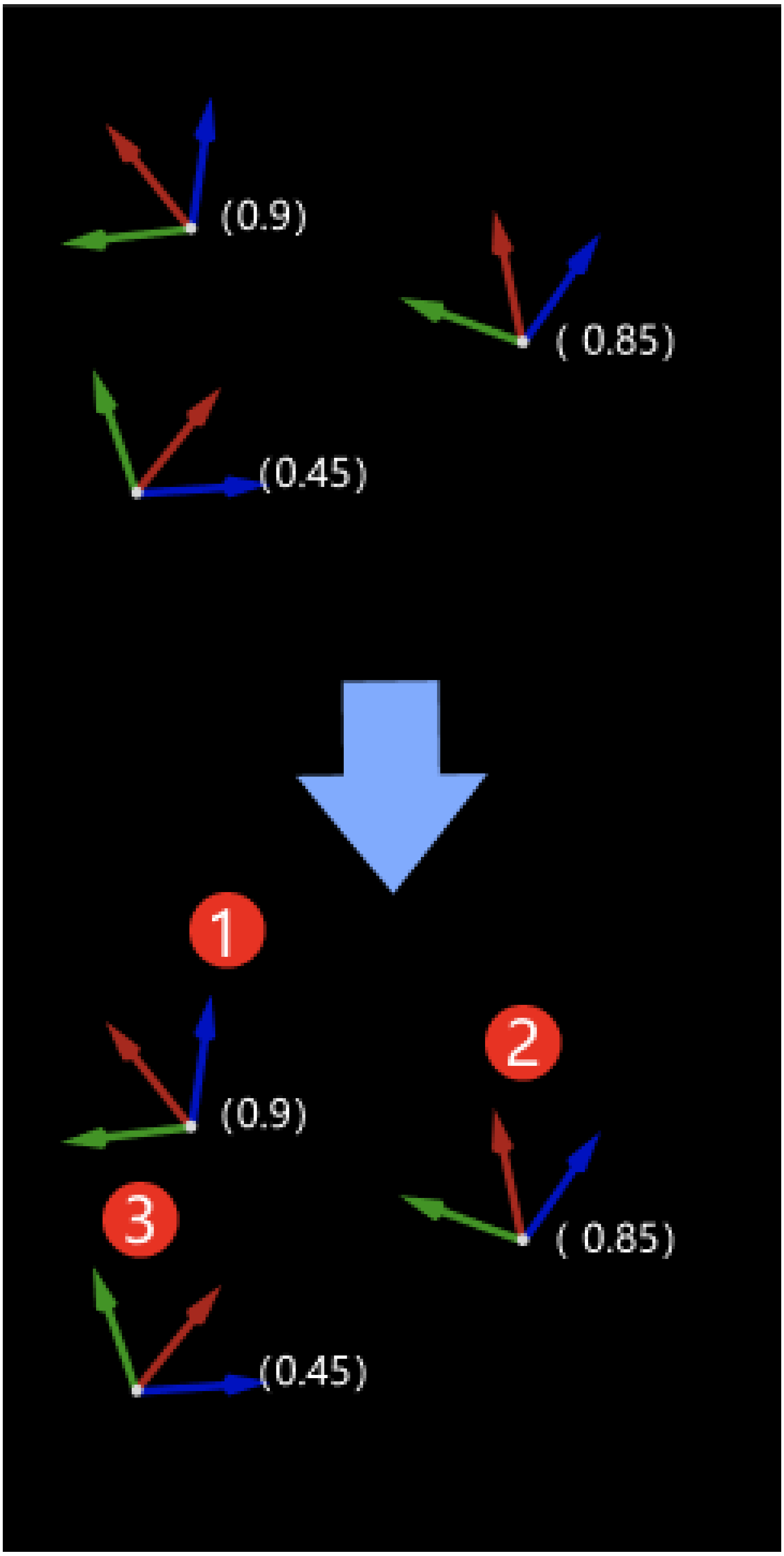
Input and Output


Point cloud list to be sorted
Point count: 45858
Press r/R to reset camera view point to center.
<Cloud(XYZ-Normal) [] >
Point Cloud Normal
<StringList->
Labels
<NumberList->
Confidences
Sort Point Clouds (1)
<Cloud(XYZ-Normal) [] >
Sorted Point Cloud Normal
<NumberList>
Sorted Values
<StringList->
Labels
<NumberList->
Confidences
List of values of a certain property of point clouds after sorting. The property is specified by the parameter "Sorting Strategy”
0.977
0.986
0.988
1.010
Sorted list of point clouds
Point count: 45858
Press r/R to reset camera view point to center.

4.3.343. Sort Poses by Input Scores

Function

This Step sorts poses by their scores and outputs the sorted list and its index list.



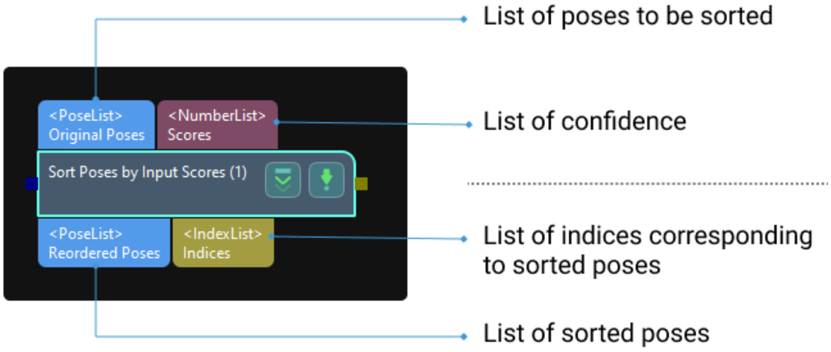
Usage Scenario

The scores usually come from the output port Pose Matching Scores of a 3D Fine Matching Step, or output ports for confidences from deep learning Steps.

Usage Scenario

- Machine tending tasks of metal workpieces: Poses can be sorted by the matching scores, and therefore workpieces with the most accurately calculated poses can be prioritized for picking.
- Deep learning: Sort poses and associated masks according to the mask areas output from a deep learning Step.

Input and Output



Parameter Description

Sort Order

Description: Select a pose sorting method. Decide whether to sort in ascending or descending order according to the input scores (confidence).

Options:

- Ascending: Sort in ascending order.
- Descending: Sort in descending order.

Default value: Descending

4.3.344. Sort Poses by XYZ Values

Function

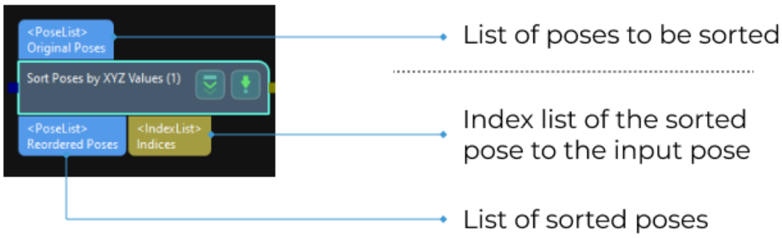
This Step sorts poses from the input pose list by any of the X, Y, or Z values and outputs the sorted list and its index list.

Other data lists related to poses, such as masks, can be reordered based on the index list of the sorted poses.

Usage Scenario

This Step is a general Step for pose sorting. It has no fixed usage scenarios.

Input and Output



List of poses to be sorted

Index list of the sorted pose to the input pose

List of sorted poses

Parameter Description

Sort Order

Description: This parameter decides whether to sort in ascending or descending order.

Options:

- Ascending: Sort in ascending order.
- Descending: Sort in descending order.

Default value: Descending

Value for Sorting

Description: This parameter decides whether X, Y, or Z values of the poses is the basis of the sorting.

Options:

- X: Sort by the X values of the poses.
- Y: Sort by the Y values of the poses.
- Z: Sort by the Z values of the poses.

Default setting: X

4.3.345. Sort 2D Poses

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step sorts the input 2D poses by X or Y coordinates.

Usage Scenario

This Step is a general 2D pose sorting Step. No fixed usage scenarios.

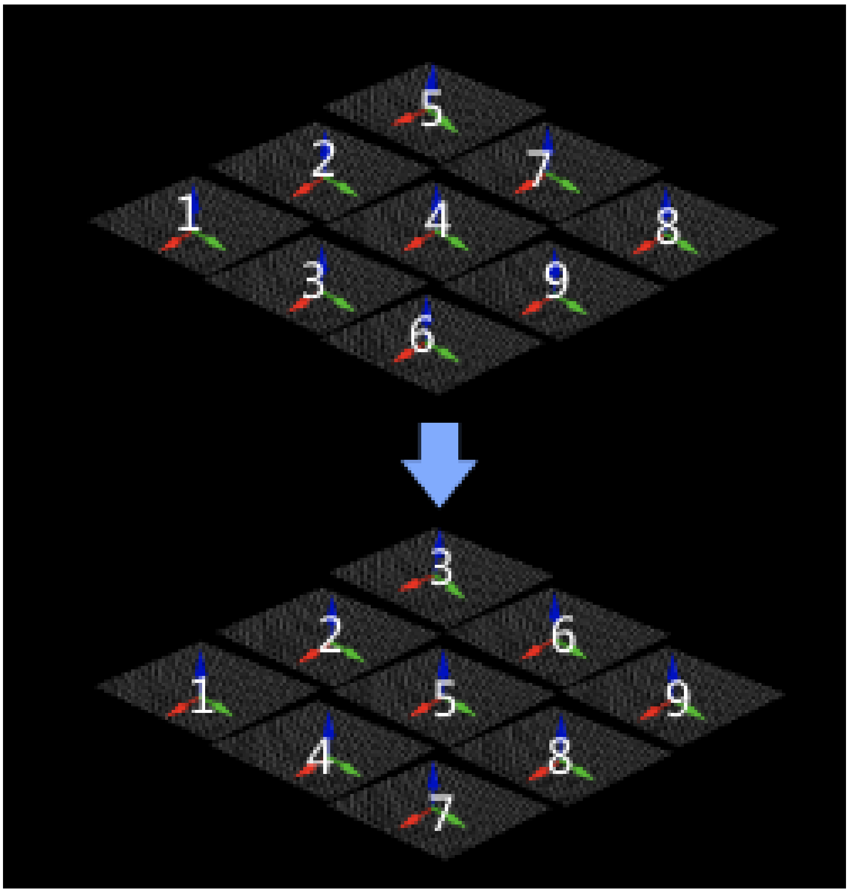
Input and Output

- **Input**: Poses input to this port will be sorted.
- **Output:** sorted poses.

4.3.346. Sort 3D Poses

Function

This Step is used to sort the 3D poses according to specified rules.

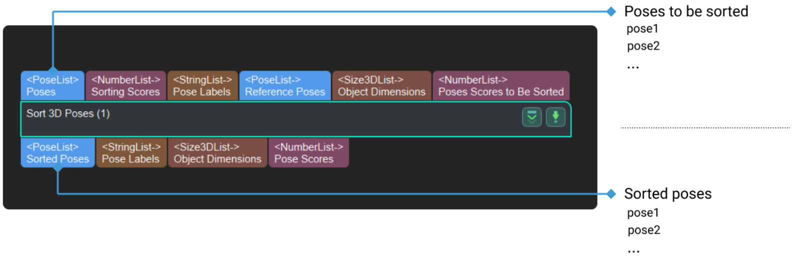


Usage Scenario

Sort the poses output from other Steps according to the required rules for subsequent pose processing or robot path planning.

This Step is used in the previous versions. It is recommended to use the new Step Sort 3D Poses V2 which is equipped with more comprehensive functions.

Input and Output


<PoseList> Poses
<NumberList-> Sorting Scores
<StringList-> Pose Labels
<PoseList-> Reference Poses
<Size3DList-> Object Dimensions
<NumberList-> Poses Scores to Be Sorted
Sort 3D Poses (1)
<PoseList> Sorted Poses
<StringList-> Pose Labels
<Size3DList-> Object Dimensions
<NumberList-> Pose Scores
Poses to be sorted
pose1
pose2
...
Sorted poses
pose1
pose2
...

Parameter Description

Sorting Strategy Settings

Sort Method

Description: Select a pose sorting method.

Default setting: NONE

Options: SORT_BY_X, SORT_BY_Y, SORT_BY_Z, SORT_BY_DIS2BasePt, SORT_BY_DIS2BasePtOnXYPlane, SORT_BY_CONFIDENCE, SORT_BY_CUSTOM_DIR, and SORT_BY_DIAGONAL_LENGTH.

Instructions:

- SORT_BY_X: Sort the poses and scores by the X value of the pose translation. Please refer to the tuning example for the corresponding result.
- SORT_BY_Y: Sort the poses and scores by the Y value of the pose translation. Please refer to the tuning example for the corresponding result.
- SORT_BY_Z: Sort the poses and scores by the Z value of the pose translation. Please refer to the tuning example for the corresponding result.
- SORT_BY_DIS2BasePt: Sort the poses and scores by the distance between the reference pose and the input pose. You need to set a reference pose when this option is selected. Please refer to the tuning example for the corresponding result.
- SORT_BY_DIS2BasePtOnXYPlane: Sort the poses and scores by the distance between the reference pose and the input poses on the XY plane. You need to set a reference pose when this option is selected. Please refer to the tuning example for the corresponding result.
- SORT_BY_CONFIDENCE: Sort the poses and scores by the score criteria. When this option is selected, you need to input the sorting score. Please refer to the tuning example for the corresponding result.
- SORT_BY_CUSTOM_DIR: Sort the poses and scores by the pose translation along a customized direction. When this option is selected, you need to specify a reference direction. Please refer to the tuning example for the corresponding result.
- SORT_BY_DIAGONAL_LENGTH: Sort the poses and scores by the diagonal length of the input dimensions. When this option is selected, you need to input the object dimensions. Please refer to the tuning example for the corresponding result.

Ascending

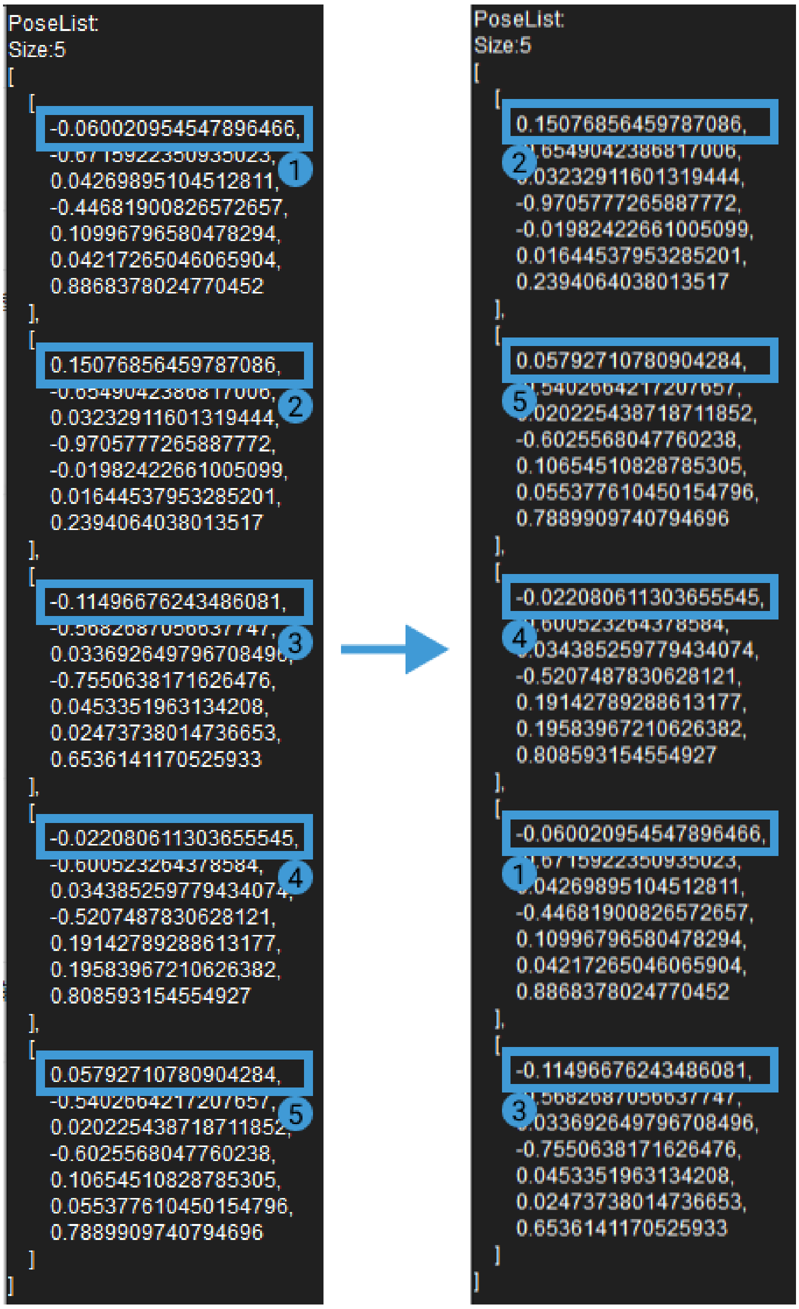
Default setting: selected.

Instruction: Select to sort in ascending order according to actual requirements.

Tuning Examples

SORT_BY_X

The sorting result is shown in the figure below (in descending order).

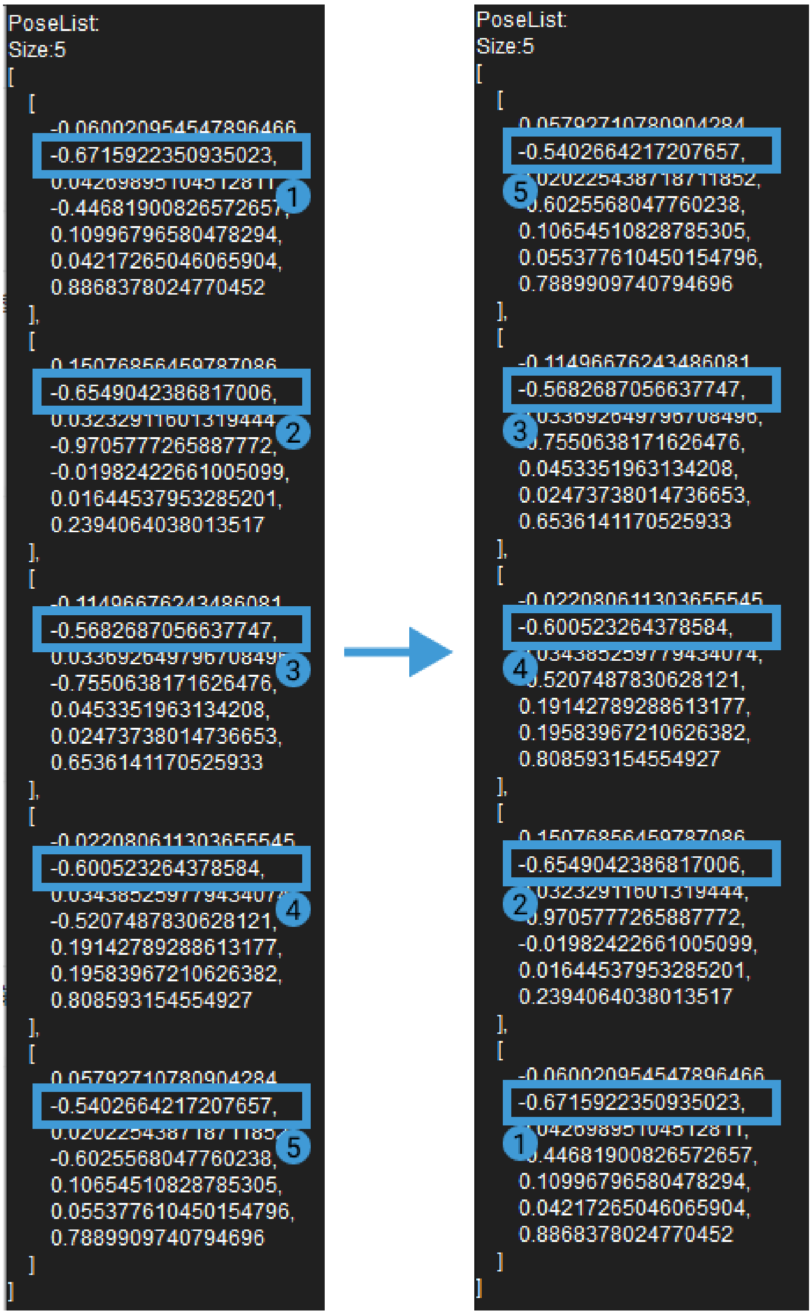


```
PoseList:
Size:5
[
   [
      -0.060020954547896466,
      -0.6715922350935023,
      0.04269895104512811,
      -0.44681900826572657,
      0.10996796580478294,
      0.04217265046065904,
      0.8868378024770452
   ],
   [
      0.15076856459787086,
      -0.6549042386817006,
      0.03232911601319444,
      -0.9705777265887772,
      -0.01982422661005099,
      0.01644537953285201,
      0.2394064038013517
   ],
   [
      -0.11496676243486081,
      -0.5682687056637747,
      0.033692649796708496,
      -0.7550638171626476,
      0.0453351963134208,
      0.02473738014736653,
      0.6536141170525933
   ],
   [
      -0.022080611303655545,
      -0.600523264378584,
      0.034385259779434074,
      -0.5207487830628121,
      0.19142789288613177,
      0.19583967210626382,
      0.808593154554927
   ],
   [
      0.05792710780904284,
      -0.5402664217207657,
      0.020225438718711852,
      -0.6025568047760238,
      0.10654510828785305,
      0.055377610450154796,
      0.7889909740794696
   ]
]
```

```
PoseList:
Size:5
[
   [
      0.15076856459787086,
      -0.6549042386817006,
      0.03232911601319444,
      -0.9705777265887772,
      -0.01982422661005099,
      0.01644537953285201,
      0.2394064038013517
   ],
   [
      0.05792710780904284,
      -0.5402664217207657,
      0.020225438718711852,
      -0.6025568047760238,
      0.10654510828785305,
      0.055377610450154796,
      0.7889909740794696
   ],
   [
      -0.022080611303655545,
      -0.600523264378584,
      0.034385259779434074,
      -0.5207487830628121,
      0.19142789288613177,
      0.19583967210626382,
      0.808593154554927
   ],
   [
      -0.060020954547896466,
      -0.6715922350935023,
      0.04269895104512811,
      -0.44681900826572657,
      0.10996796580478294,
      0.04217265046065904,
      0.8868378024770452
   ],
   [
      -0.11496676243486081,
      -0.5682687056637747,
      0.033692649796708496,
      -0.7550638171626476,
      0.0453351963134208,
      0.02473738014736653,
      0.6536141170525933
   ]
]
```

SORT_BY_Y

The sorting result is shown in the figure below (in descending order).

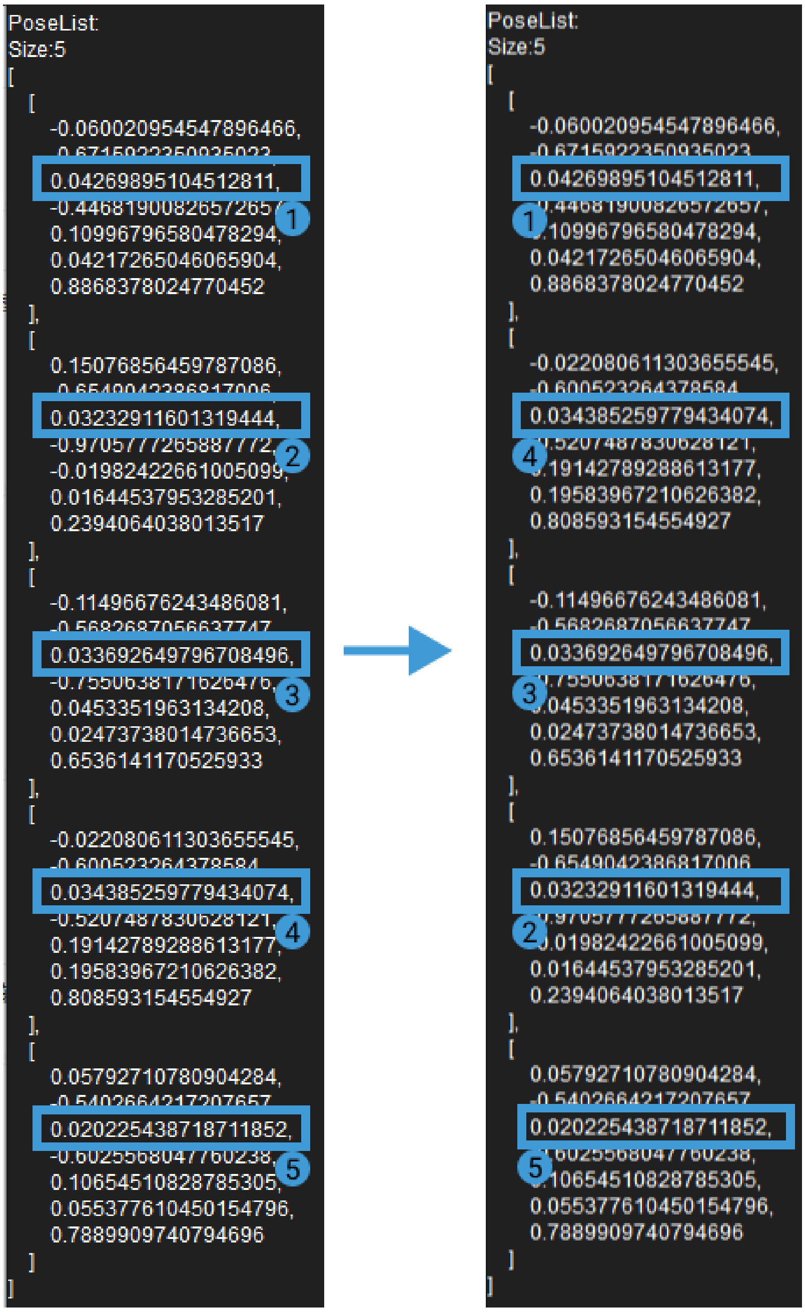


```
PoseList:
Size:5
[
  [
    -0.060020954547896466,
    -0.6715922350935023,
    0.04269895104512811,
    -0.44681900826572657,
    0.10996796580478294,
    0.04217265046065904,
    0.8868378024770452
  ],
  [
    0.15076856459787086,
    -0.6549042386817006,
    0.03232911601319444,
    -0.9705777265887772,
    -0.01982422661005099,
    0.01644537953285201,
    0.2394064038013517
  ],
  [
    -0.11496676243486081,
    -0.5682687056637747,
    0.033692649796708496,
    -0.7550638171626476,
    0.0453351963134208,
    0.02473738014736653,
    0.6536141170525933
  ],
  [
    -0.022080611303655545,
    -0.600523264378584,
    0.034385259779434074,
    -0.5207487830628121,
    0.19142789288613177,
    0.19583967210626382,
    0.808593154554927
  ],
  [
    0.05792710780904284,
    -0.5402664217207657,
    0.0202225438718711852,
    -0.6025568047760238,
    0.10654510828785305,
    0.055377610450154796,
    0.7889909740794696
  ]
]
```

```
PoseList:
Size:5
[
  [
    0.05792710780904284,
    -0.5402664217207657,
    0.0202225438718711852,
    -0.6025568047760238,
    0.10654510828785305,
    0.055377610450154796,
    0.7889909740794696
  ],
  [
    -0.11496676243486081,
    -0.5682687056637747,
    0.033692649796708496,
    -0.7550638171626476,
    0.0453351963134208,
    0.02473738014736653,
    0.6536141170525933
  ],
  [
    -0.022080611303655545,
    -0.600523264378584,
    0.034385259779434074,
    -0.5207487830628121,
    0.19142789288613177,
    0.19583967210626382,
    0.808593154554927
  ],
  [
    0.15076856459787086,
    -0.6549042386817006,
    0.03232911601319444,
    -0.9705777265887772,
    -0.01982422661005099,
    0.01644537953285201,
    0.2394064038013517
  ],
  [
    -0.060020954547896466,
    -0.6715922350935023,
    0.04269895104512811,
    -0.44681900826572657,
    0.10996796580478294,
    0.04217265046065904,
    0.8868378024770452
  ]
]
```

SORT_BY_Z

The sorting result is shown in the figure below (in descending order).

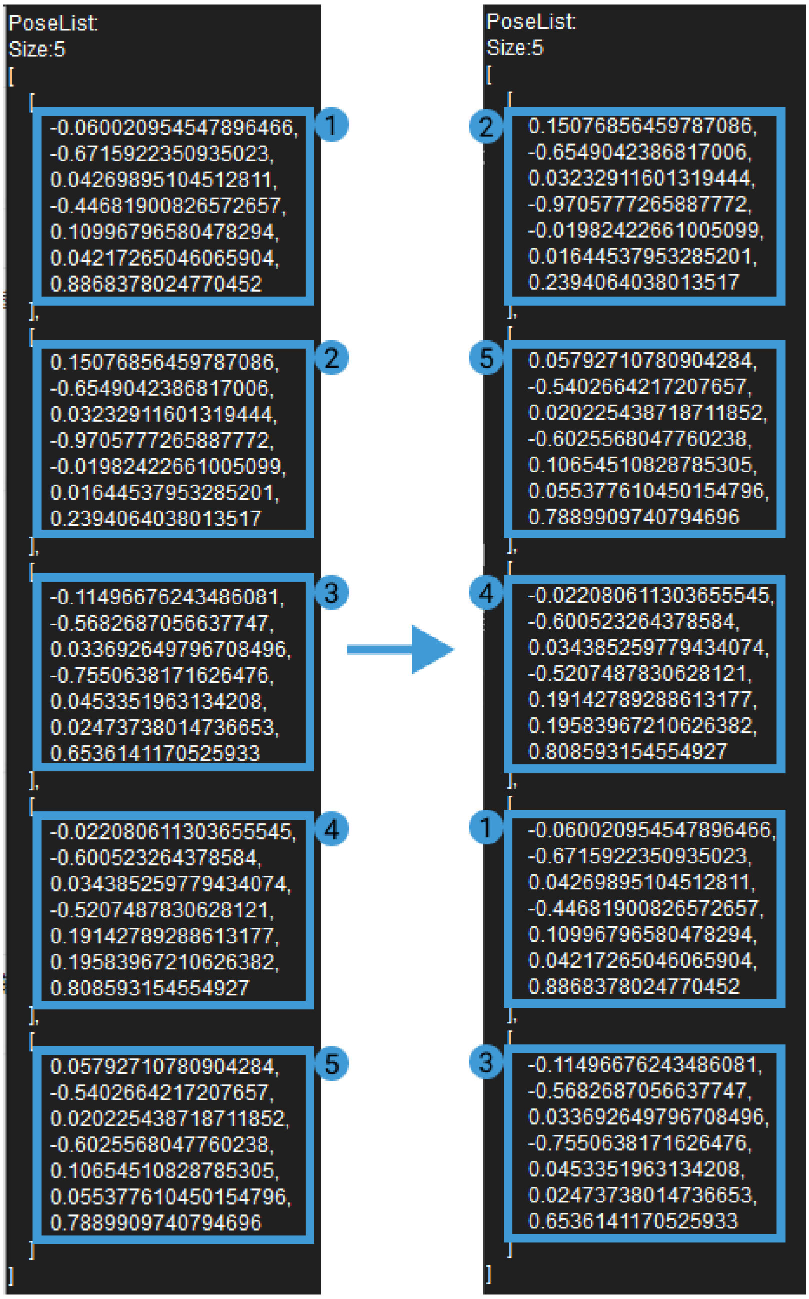


```
PoseList:
Size:5
[
    [
        -0.060020954547896466,
        -0.6715922350935023,
        0.04269895104512811,
        -0.4468190082657265,
        0.10996796580478294,
        0.04217265046065904,
        0.8868378024770452
    ],
    [
        0.15076856459787086,
        -0.6549042386817006,
        0.03232911601319444,
        -0.9705777265887772,
        -0.01982422661005099,
        0.01644537953285201,
        0.2394064038013517
    ],
    [
        -0.11496676243486081,
        -0.5682687056637747,
        0.033692649796708496,
        -0.7550638171626476,
        0.0453351963134208,
        0.02473738014736653,
        0.6536141170525933
    ],
    [
        -0.022080611303655545,
        -0.600523264378584,
        0.034385259779434074,
        -0.5207487830628121,
        0.19142789288613177,
        0.19583967210626382,
        0.808593154554927
    ],
    [
        0.05792710780904284,
        -0.5402664217207657,
        0.0202225438718711852,
        -0.6025568047760238,
        0.10654510828785305,
        0.055377610450154796,
        0.7889909740794696
    ]
]
```

```
PoseList:
Size:5
[
    [
        -0.060020954547896466,
        -0.6715922350935023,
        0.04269895104512811,
        -0.4468190082657265,
        0.10996796580478294,
        0.04217265046065904,
        0.8868378024770452
    ],
    [
        -0.022080611303655545,
        -0.600523264378584,
        0.034385259779434074,
        -0.5207487830628121,
        0.19142789288613177,
        0.19583967210626382,
        0.808593154554927
    ],
    [
        -0.11496676243486081,
        -0.5682687056637747,
        0.033692649796708496,
        -0.7550638171626476,
        0.0453351963134208,
        0.02473738014736653,
        0.6536141170525933
    ],
    [
        0.15076856459787086,
        -0.6549042386817006,
        0.03232911601319444,
        -0.9705777265887772,
        -0.01982422661005099,
        0.01644537953285201,
        0.2394064038013517
    ],
    [
        0.05792710780904284,
        -0.5402664217207657,
        0.0202225438718711852,
        -0.6025568047760238,
        0.10654510828785305,
        0.055377610450154796,
        0.7889909740794696
    ]
]
```

SORT_BY_DIS2BasePt

When the input reference pose is [−1,−1,−1], the sorting result is shown in the figure below (in descending order).

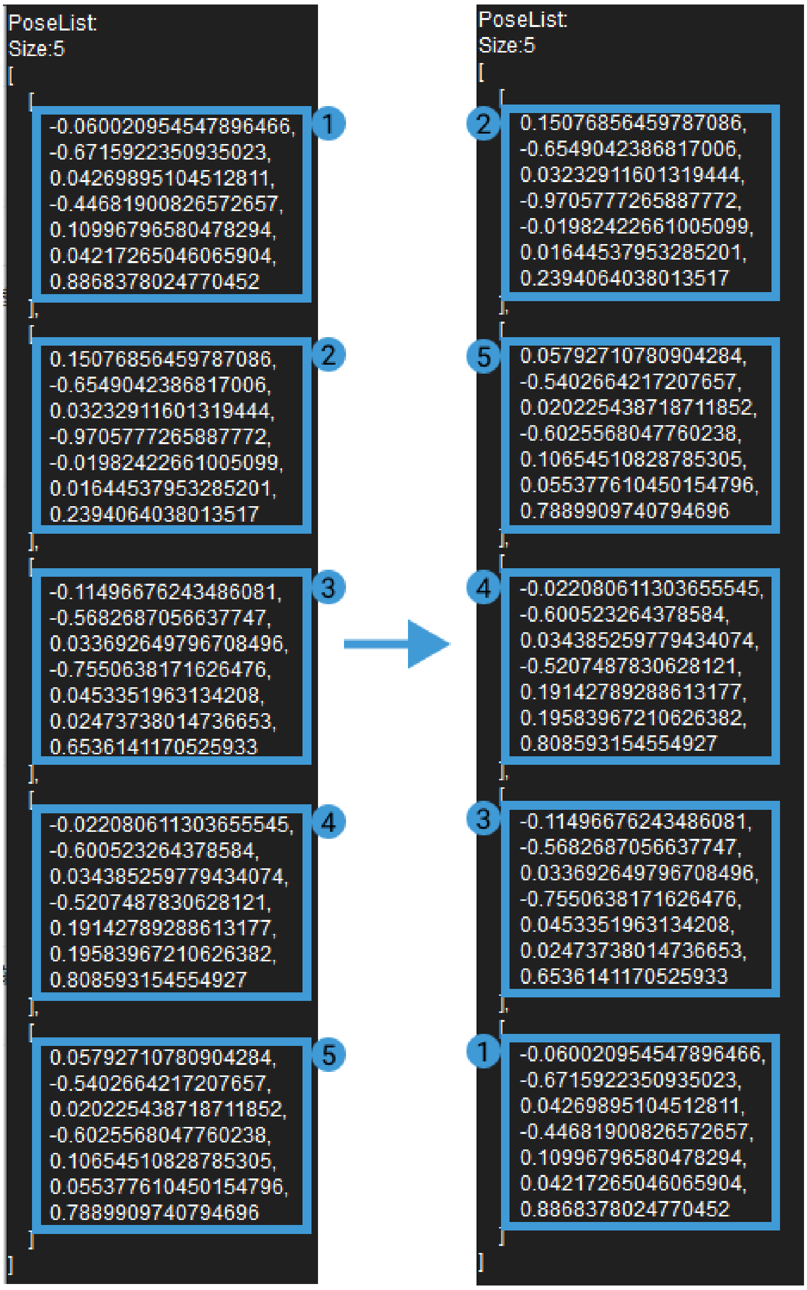


```
PoseList:
Size:5
[
  [
    -0.06002095454789646б,
    -0.6715922350935023,
    0.04269895104512811,
    -0.44681900826572657,
    0.10996796580478294,
    0.04217265046065904,
    0.8868378024770452
  ],
  [
    0.15076856459787086,
    -0.6549042386817006,
    0.03232911601319444,
    -0.9705777265887772,
    -0.01982422661005099,
    0.01644537953285201,
    0.2394064038013517
  ],
  [
    -0.11496676243486081,
    -0.5682687056637747,
    0.033692649796708496,
    -0.7550638171626476,
    0.0453351963134208,
    0.02473738014736653,
    0.6536141170525933
  ],
  [
    -0.022080611303655545,
    -0.600523264378584,
    0.034385259779434074,
    -0.5207487830628121,
    0.19142789288613177,
    0.19583967210626382,
    0.808593154554927
  ],
  [
    0.057927107809042840,
    -0.5402664217207657,
    0.020225438718711852,
    -0.6025568047760238,
    0.10654510828785305,
    0.055377610450154796,
    0.7889909740794696
  ]
]
```

```
PoseList:
Size:5
[
  [
    0.15076856459787086,
    -0.6549042386817006,
    0.03232911601319444,
    -0.9705777265887772,
    -0.01982422661005099,
    0.01644537953285201,
    0.2394064038013517
  ],
  [
    0.05792710780904284,
    -0.5402664217207657,
    0.020225438718711852,
    -0.6025568047760238,
    0.10654510828785305,
    0.055377610450154796,
    0.7889909740794696
  ],
  [
    -0.022080611303655545,
    -0.600523264378584,
    0.034385259779434074,
    -0.5207487830628121,
    0.19142789288613177,
    0.19583967210626382,
    0.808593154554927
  ],
  [
    -0.06002095454789646б,
    -0.6715922350935023,
    0.04269895104512811,
    -0.44681900826572657,
    0.10996796580478294,
    0.04217265046065904,
    0.8868378024770452
  ],
  [
    -0.11496676243486081,
    -0.5682687056637747,
    0.033692649796708496,
    -0.7550638171626476,
    0.0453351963134208,
    0.02473738014736653,
    0.6536141170525933
  ]
]
```

SORT_BY_DIS2BasePtOnXYPlane

When the input reference pose is [−1,−1,0], the sorting result is shown in the figure below (in descending order).

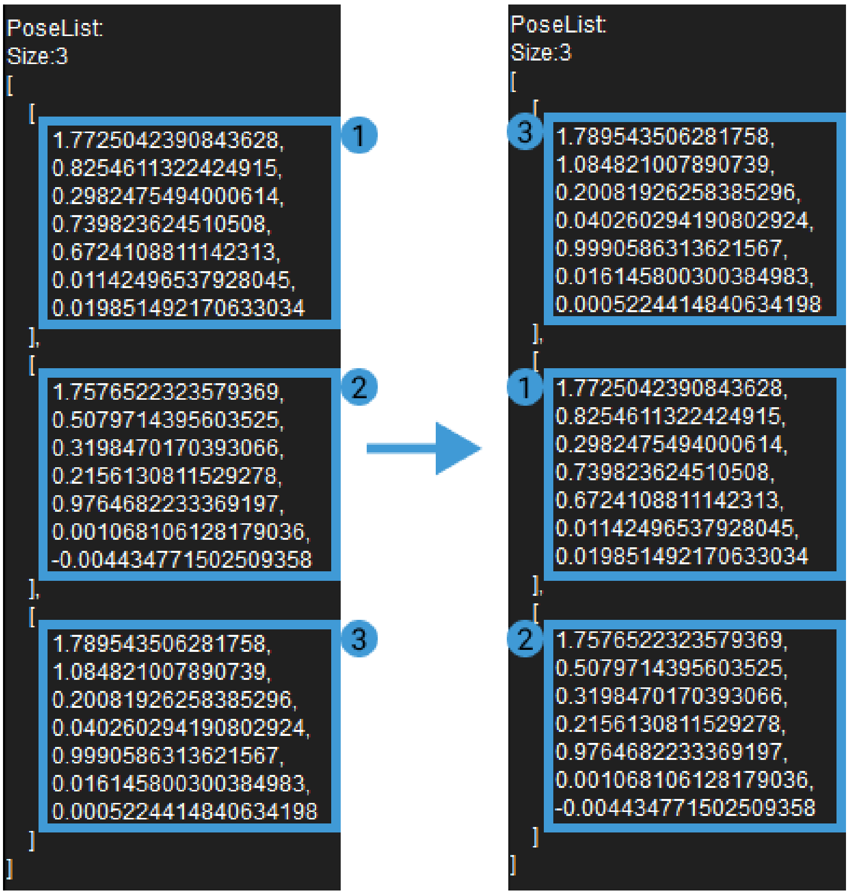


```
PoseList:
Size:5
[
    [
        -0.060020954547896466,
        -0.6715922350935023,
        0.04269895104512811,
        -0.44681900826572657,
        0.10996796580478294,
        0.04217265046065904,
        0.8868378024770452
    ],
    [
        0.15076856459787086,
        -0.6549042386817006,
        0.03232911601319444,
        -0.9705777265887772,
        -0.01982422661005099,
        0.01644537953285201,
        0.2394064038013517
    ],
    [
        -0.11496676243486081,
        -0.5682687056637747,
        0.033692649796708496,
        -0.7550638171626476,
        0.0453351963134208,
        0.02473738014736653,
        0.6536141170525933
    ],
    [
        -0.022080611303655545,
        -0.600523264378584,
        0.034385259779434074,
        -0.5207487830628121,
        0.19142789288613177,
        0.19583967210626382,
        0.808593154554927
    ],
    [
        0.05792710780904284,
        -0.5402664217207657,
        0.020225438718711852,
        -0.6025568047760238,
        0.10654510828785305,
        0.055377610450154796,
        0.7889909740794696
    ]
]
```

```
PoseList:
Size:5
[
    [
        0.15076856459787086,
        -0.6549042386817006,
        0.03232911601319444,
        -0.9705777265887772,
        -0.01982422661005099,
        0.01644537953285201,
        0.2394064038013517
    ],
    [
        0.05792710780904284,
        -0.5402664217207657,
        0.020225438718711852,
        -0.6025568047760238,
        0.10654510828785305,
        0.055377610450154796,
        0.7889909740794696
    ],
    [
        -0.022080611303655545,
        -0.600523264378584,
        0.034385259779434074,
        -0.5207487830628121,
        0.19142789288613177,
        0.19583967210626382,
        0.808593154554927
    ],
    [
        -0.11496676243486081,
        -0.5682687056637747,
        0.033692649796708496,
        -0.7550638171626476,
        0.0453351963134208,
        0.02473738014736653,
        0.6536141170525933
    ],
    [
        -0.060020954547896466,
        -0.6715922350935023,
        0.04269895104512811,
        -0.44681900826572657,
        0.10996796580478294,
        0.04217265046065904,
        0.8868378024770452
    ]
]
```

SORT_BY_CONFIDENCE

When the input scores of the poses are a, b, and c, respectively, and c > a > b, the sorting result is shown in the figure below (in descending order).

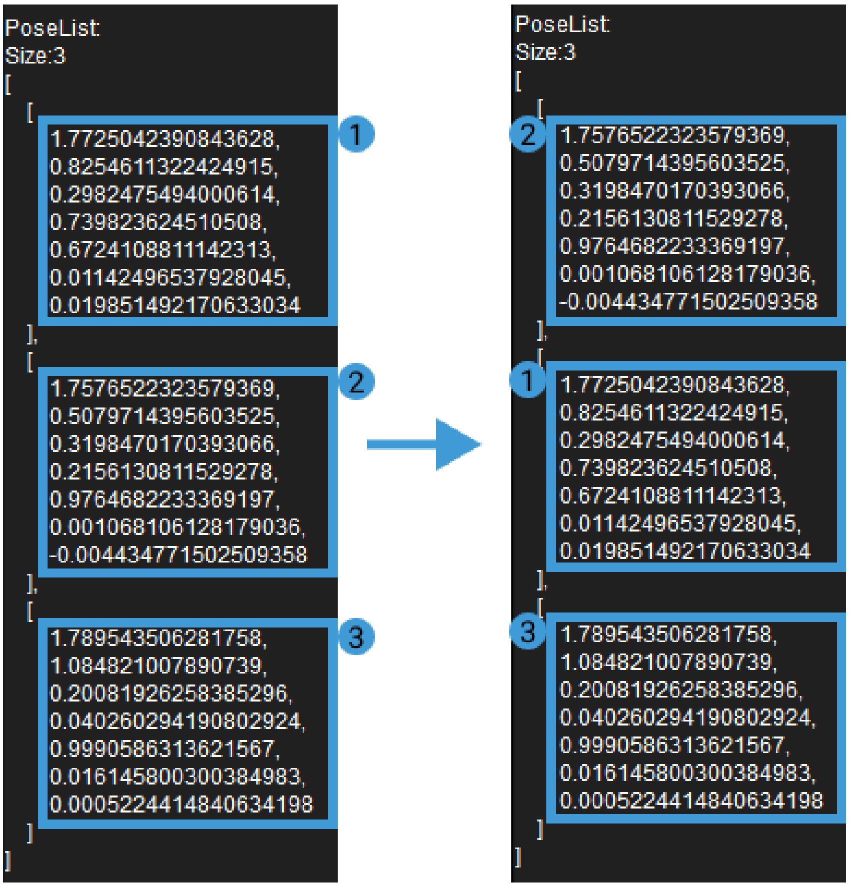


```
PoseList:
Size:3
[
  [
    1.7725042390843628,
    0.8254611322424915,
    0.2982475494000614,
    0.739823624510508,
    0.6724108811142313,
    0.01142496537928045,
    0.019851492170633034
  ],
  [
    1.7576522323579369,
    0.5079714395603525,
    0.3198470170393066,
    0.2156130811529278,
    0.9764682233369197,
    0.0010681061281790 36,
    -0.004434771502509358
  ],
  [
    1.789543506281758,
    1.084821007890739,
    0.20081926258385296,
    0.040260294190802924,
    0.9990586313621567,
    0.016145800300384983,
    0.0005224414840634198
  ]
]
```

```
PoseList:
Size:3
[
  [
    1.789543506281758,
    1.084821007890739,
    0.20081926258385296,
    0.040260294190802924,
    0.9990586313621567,
    0.016145800300384983,
    0.0005224414840634198
  ],
  [
    1.7725042390843628,
    0.825461132242491 5,
    0.2982475494000614,
    0.739823624510508,
    0.6724108811142313,
    0.01142496537928045,
    0.019851492170633034
  ],
  [
    1.7576522323579369,
    0.5079714395603525,
    0.3198470170393066,
    0.2156130811529278,
    0.9764682233369197,
    0.001068106128179036,
    -0.004434771502509358
  ]
]
```

SORT_BY_CUSTOM_DIR

When the values of specified reference directions X, Y, and Z are set to 0, 0, and 1, respectively, the sorting result is shown in the figure below (in descending order).

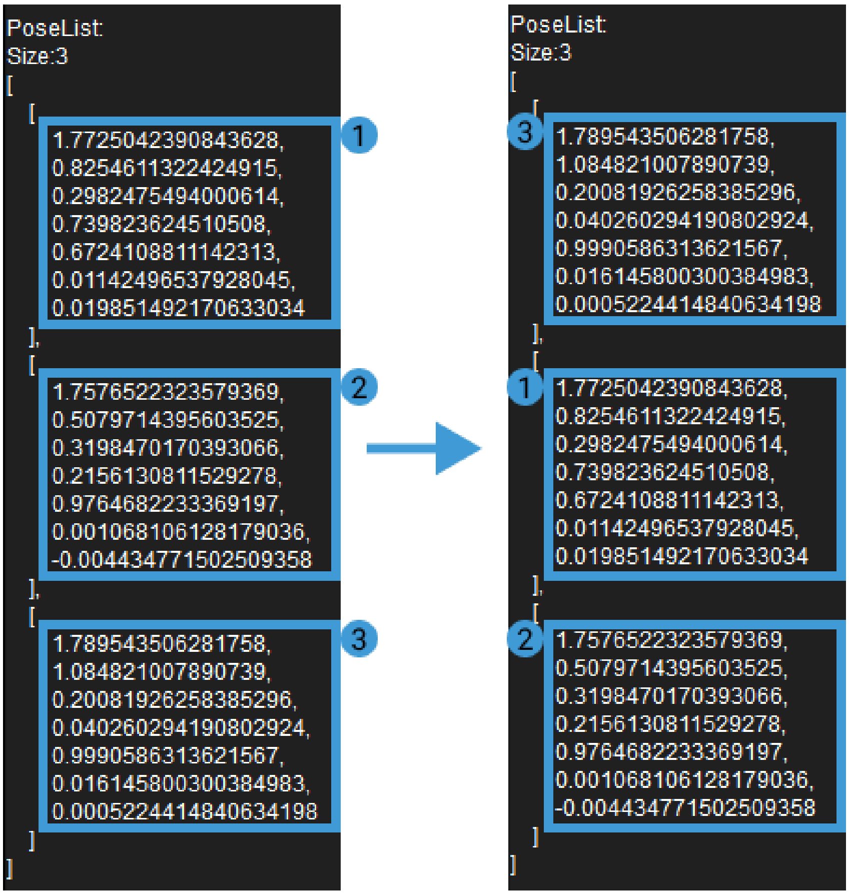


```
PoseList:
Size:3
[
  [
    1.7725042390843628,
    0.8254611322424915,
    0.2982475494000614,
    0.739823624510508,
    0.6724108811142313,
    0.01142496537928045,
    0.019851492170633034
  ],
  [
    1.7576522323579369,
    0.5079714395603525,
    0.3198470170393066,
    0.2156130811529278,
    0.9764682233369197,
    0.001068106128179036,
    -0.004434771502509358
  ],
  [
    1.789543506281758,
    1.084821007890739,
    0.20081926258385296,
    0.040260294190802924,
    0.9990586313621567,
    0.016145800300384983,
    0.0005224414840634198
  ]
]
```

```
PoseList:
Size:3
[
  [
    1.7576522323579369,
    0.5079714395603525,
    0.3198470170393066,
    0.2156130811529278,
    0.9764682233369197,
    0.001068106128179036,
    -0.004434771502509358
  ],
  [
    1.7725042390843628,
    0.8254611322424915,
    0.2982475494000614,
    0.739823624510508,
    0.6724108811142313,
    0.01142496537928045,
    0.019851492170633034
  ],
  [
    1.789543506281758,
    1.084821007890739,
    0.20081926258385296,
    0.040260294190802924,
    0.9990586313621567,
    0.016145800300384983,
    0.0005224414840634198
  ]
]
```

SORT_BY_DIAGONAL_LENGTH

When the input diagonal lengths are a, b, and c, respectively, and c > a > b, the sorting result is shown in the figure below (in descending order).



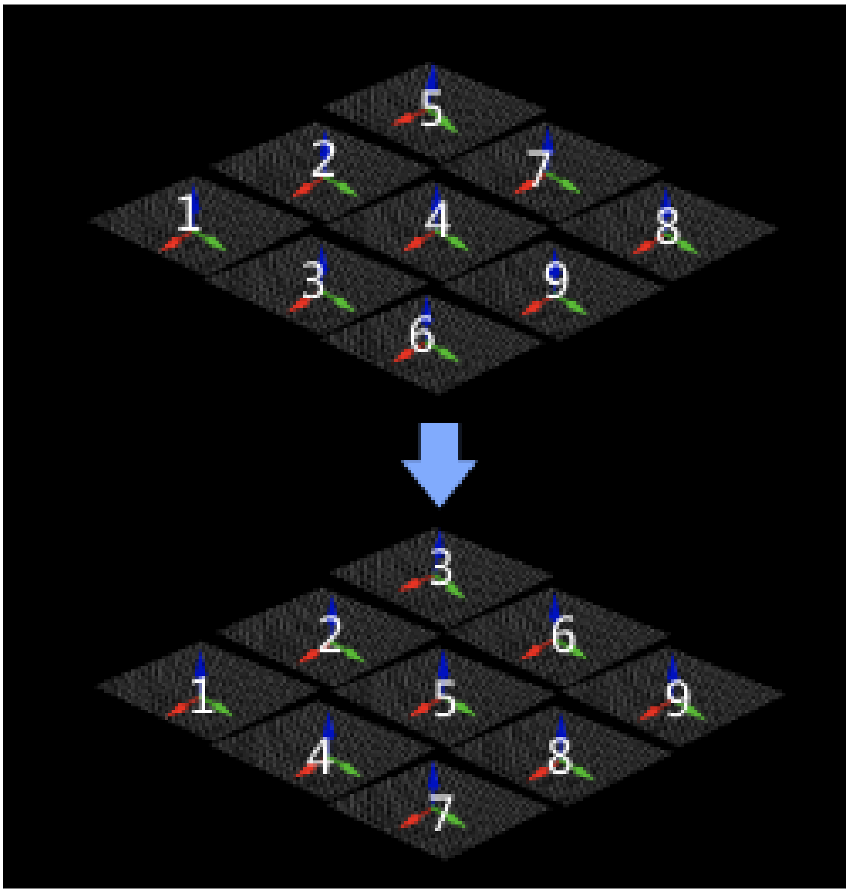
```
PoseList:
Size:3
[
  [
    1.7725042390843628,
    0.8254611322424915,
    0.2982475494000614,
    0.739823624510508,
    0.6724108811142313,
    0.01142496537928045,
    0.019851492170633034
  ],
  [
    1.7576522323579369,
    0.5079714395603525,
    0.3198470170393066,
    0.2156130811529278,
    0.9764682233369197,
    0.001068106128179036,
    -0.004434771502509358
  ],
  [
    1.789543506281758,
    1.084821007890739,
    0.20081926258385296,
    0.040260294190802924,
    0.9990586313621567,
    0.016145800300384983,
    0.0005224414840634198
  ]
]
```

```
PoseList:
Size:3
[
  [
    1.789543506281758,
    1.084821007890739,
    0.20081926258385296,
    0.040260294190802924,
    0.9990586313621567,
    0.016145800300384983,
    0.0005224414840634198
  ],
  [
    1.7725042390843628,
    0.8254611322424915,
    0.2982475494000614,
    0.739823624510508,
    0.6724108811142313,
    0.01142496537928045,
    0.019851492170633034
  ],
  [
    1.7576522323579369,
    0.5079714395603525,
    0.3198470170393066,
    0.2156130811529278,
    0.9764682233369197,
    0.001068106128179036,
    -0.004434771502509358
  ]
]
```

4.3.347. Sort 3D Poses V2

Function

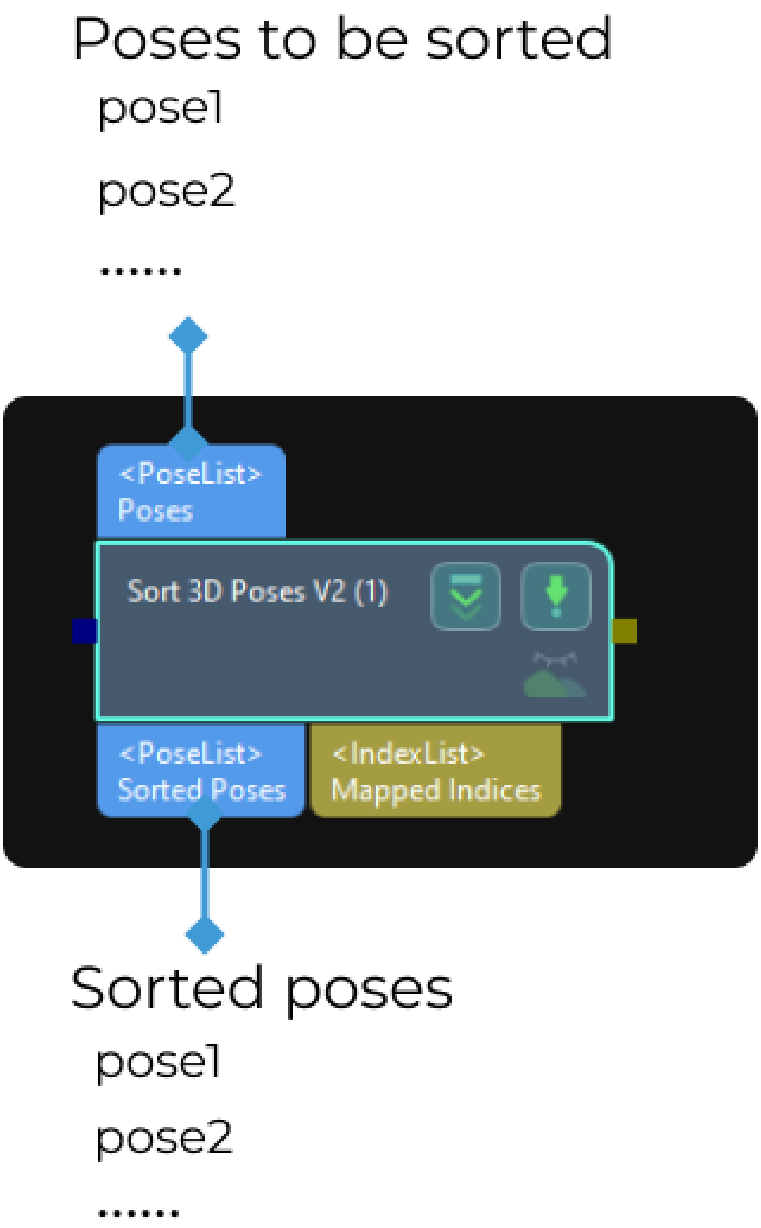
This Step sorts 3D poses according to specified rules.



Usage Scenario

This Step sorts poses according to specified rules for subsequent pose processing or picking strategy.

Input and Output



Parameters

This Step provides eleven methods to sort poses. The detailed description is as follows.

Sort by None

Description

The input poses will not be sorted when you select “None” next to “Sort by.”

Parameters

Show Point Cloud in Selected Frame

Parameter description: This parameter is used to select the reference frame that display the point cloud.

Value list: Robot Coordinate, Camera Coordinate.

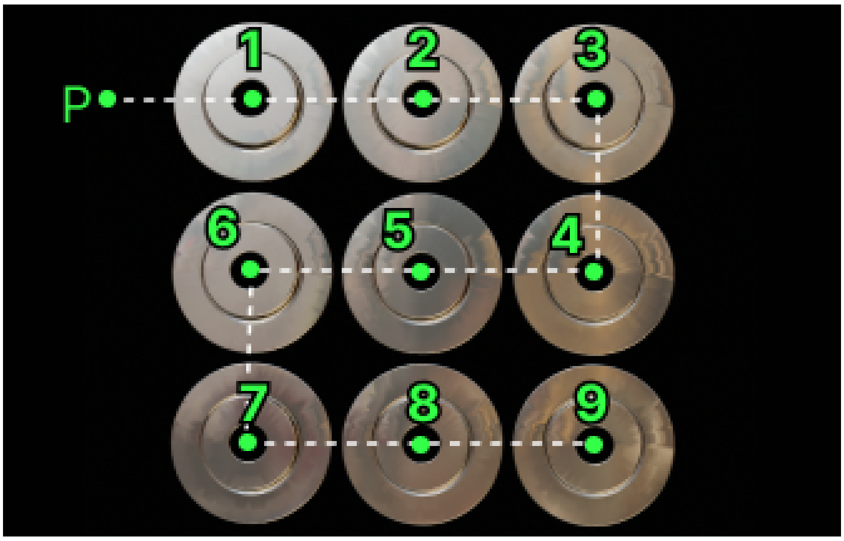
Default value: Robot Coordinate.

Tuning instruction: Select this parameter according to actual situation.

Sort by S shape on plane

Description

Sort the poses in rows and columns based on the given reference pose (with priority given to sorting poses according to “Row Direction”). This method is usually used for unloading neatly arranged workpieces or depalletizing.



Parameters

Row direction

Parameter description: This parameter is used to set the row direction when the poses are sorted on the plane.

Value list: Positive X-direction, Positive Y-direction, Positive Z-direction, Negative X-direction, Negative Y-direction, Negative Z-direction.

Default value: Positive X-direction.

Column direction

Parameter description: This parameter is used to set the column direction when the poses are sorted on the plane.

Value list: Positive X-direction, Positive Y-direction, Positive Z-direction, Negative X-direction, Negative Y-direction, Negative Z-direction.

Default value: Positive Y-direction.

Row Interval

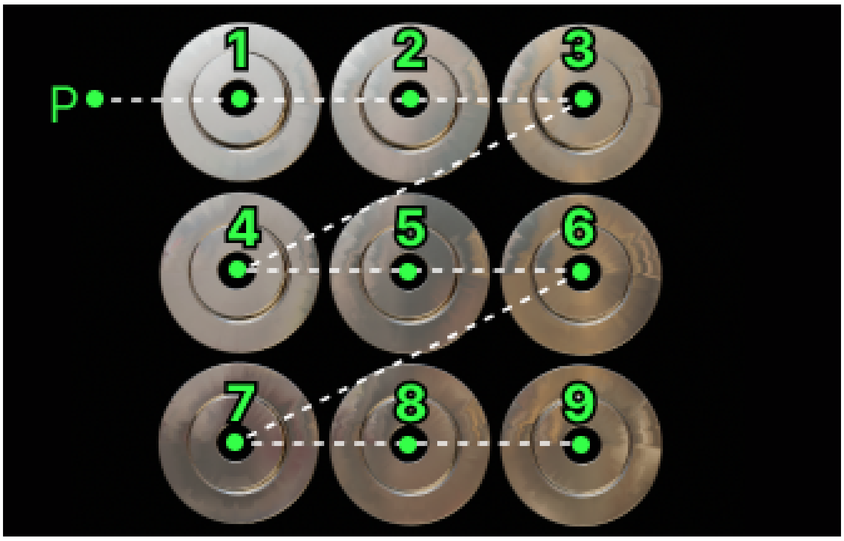
Parameter description: This parameter is used to set the interval (in mm) between rows when the poses are sorted on the plane.

Default value: 100.000 mm

Sort by Z shape on plane

Description

Sort the poses in rows and columns based on the given reference pose (with priority given to sorting poses according to “Row Direction”). This method is usually used for unloading neatly arranged workpieces or depalletizing.



Parameters

Row direction

Parameter description: This parameter is used to set the row direction when the poses are sorted on the plane.

Value list: Positive X-direction, Positive Y-direction, Positive Z-direction, Negative X-direction, Negative Y-direction, Negative Z-direction.

Default value: Positive X-direction.

Column direction

Parameter description: This parameter is used to set the column direction when the poses are sorted on the plane.

Value list: Positive X-direction, Positive Y-direction, Positive Z-direction, Negative X-direction, Negative Y-direction, Negative Z-direction.

Default value: Positive Y-direction.

Row Interval

Parameter description: This parameter is used to set the interval (in mm) between rows when the poses are sorted on the plane.

Default value: 100.000 mm

Sort by X value of pose

Description

Sort all poses by X value of the pose. You can switch between ascending and descending order.

Parameters

Ascending

Parameter description: Once this option is selected, the poses will be sorted in order of X values from smallest to largest.

Default value: selected.

Sort by Y value of pose

Description

Sort all poses by Y value of the pose. You can switch between ascending and descending order.

Parameters

Ascending

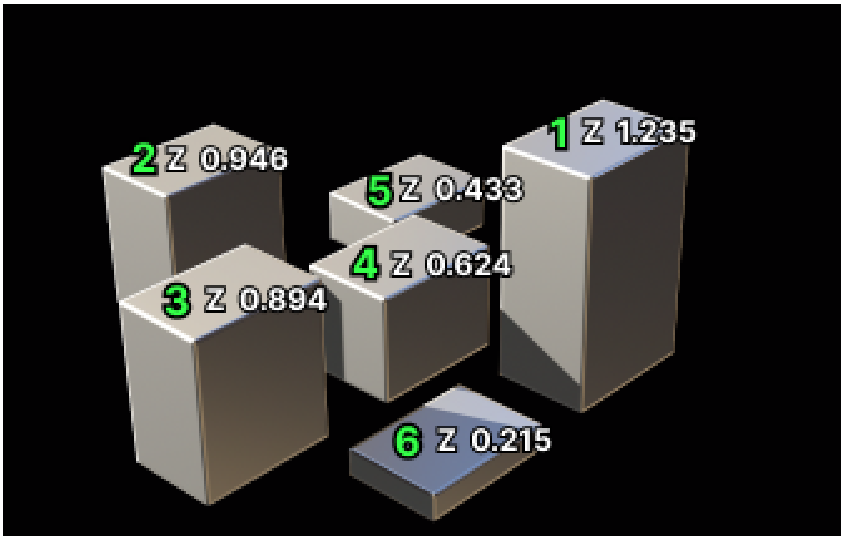
Parameter description: Once this option is selected, the poses will be sorted in order of Y values from smallest to largest.

Default value: selected.

Sort by Z value of pose

Description

Sort all poses by Z value of the pose. You can switch between ascending and descending order.



Parameters

Ascending

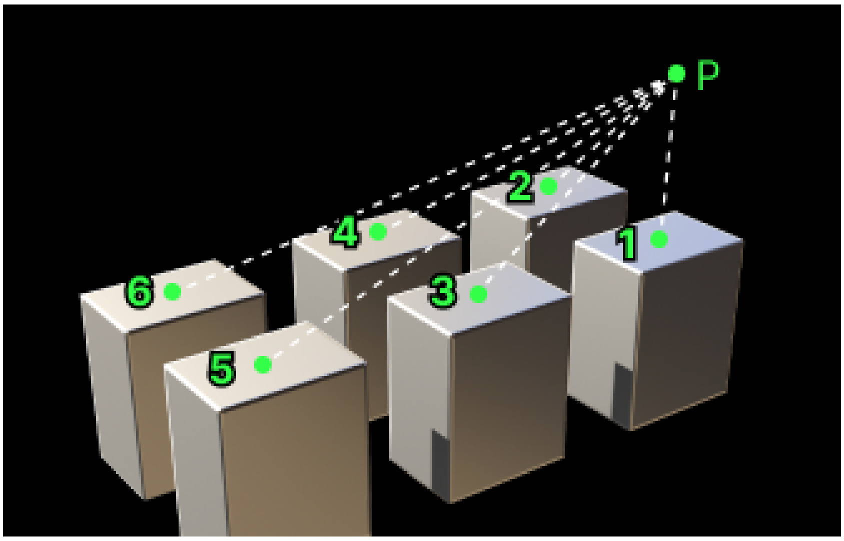
Parameter description: Once this option is selected, the poses will be sorted in order of Z values from smallest to largest.

Default value: selected.

Sort by Distance from pose to reference pose

Description

This sorting method calculates distances from all poses to be sorted to the reference pose and sort the poses according to the distance.



Parameters

Ascending

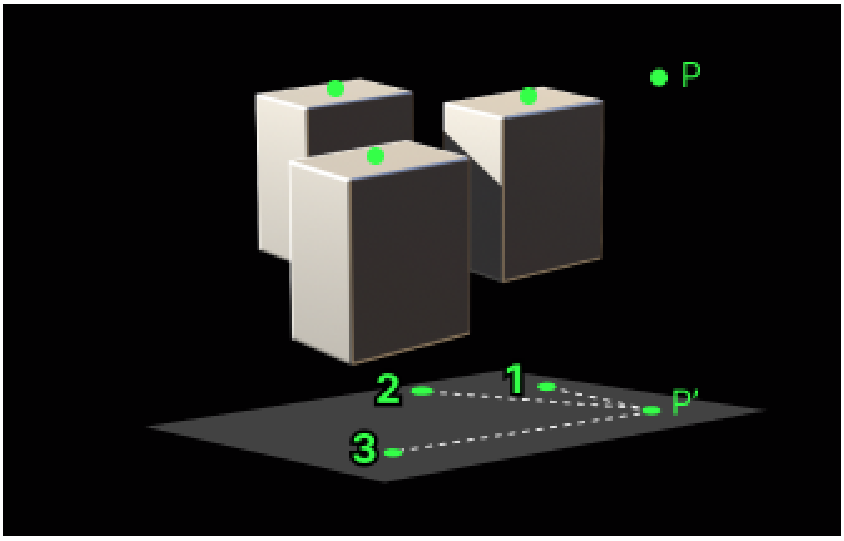
Parameter description: Once this option is selected, the poses will be sorted in order of their distances to the reference pose from shortest to longest.

Default value: selected.

Sort by Distance from pose to reference pose on XOY plane

Description

Project poses to be sorted and the specified reference pose onto the XOY plane, calculate the distances from poses to the reference pose on the plane, and then sort the poses according to the distance.



Parameters

Ascending

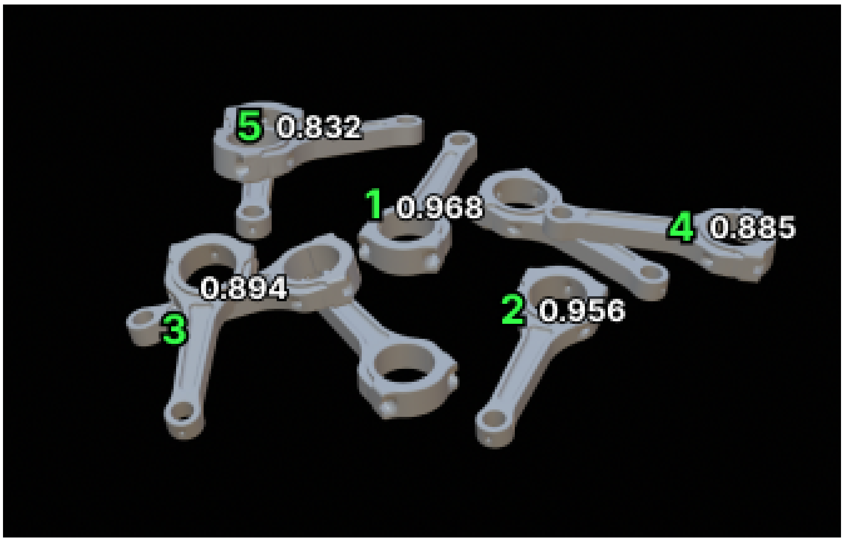
Parameter description: Once this option is selected, the poses will be sorted in order of their distances to the reference pose from shortest to longest.

Default value: selected.

Sort by Pose confidence values

Description

This sorting method sorts poses according to the corresponding confidence. You can switch between ascending and descending order.



Parameters

Ascending

Parameter description: Once this option is selected, the poses will be sorted in order of their confidence values from smallest to largest.

Default value: selected.

Sort by Projection of pose in reference direction

Description

Project the pose onto the axes of the reference pose, calculate the projection distance, and then sort the poses according to the distance.

Parameters

Ascending

Parameter description: Once this option is selected, the poses will be sorted in order of projection distances from shortest to longest.

Default value: selected.

Reference direction

Parameter description: This parameter is used to select the axis of the reference pose and its direction used for pose projection.

Value list: Positive X-direction, Positive Y-direction, Positive Z-direction, Negative X-direction, Negative Y-direction, Negative Z-direction.

Default value: Positive X-direction.

Sort by Cuboid diagonal lengths

Description

Calculate the diagonal lengths according to the input cuboid size, and then sort the poses according to the diagonal lengths. This sorting method is usually used in the carton depalletizing scenarios.

Parameters

Ascending

Parameter description: Once this option is selected, the poses will be sorted in order of the diagonal lengths from shortest to longest.

Default value: selected.

4.3.348. Straighten Ring Shaped Image Region

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step straightens a ring-shaped region or a sector of a ring-shaped region in the input image and output.

Usage Scenario

This Step is used to read characters arranged in a ring.

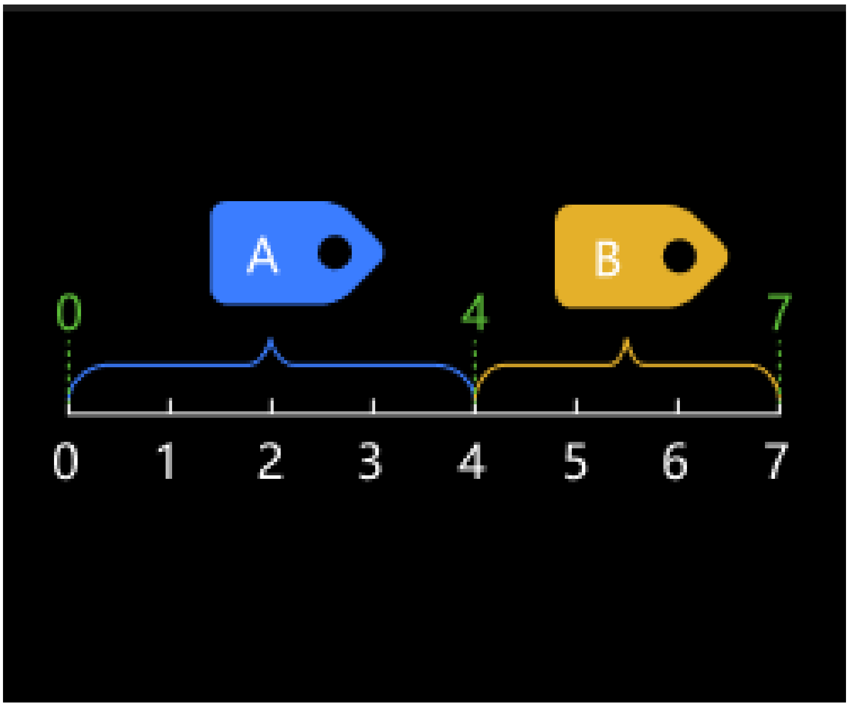
Input and Output

- Input:
  1. The specified region in the image input to this port will be straightened.
  2. 2D transformation input at this port is used to transform the ROI set in a previous execution to the appropriate position.
- Output:
  1. Image formed by straightening the ring-shaped region or a sector of a ring-shaped region

4.3.349. Stratify Values by Thresholds

Function

This Step stratifies input values by the set thresholds and output the labels.

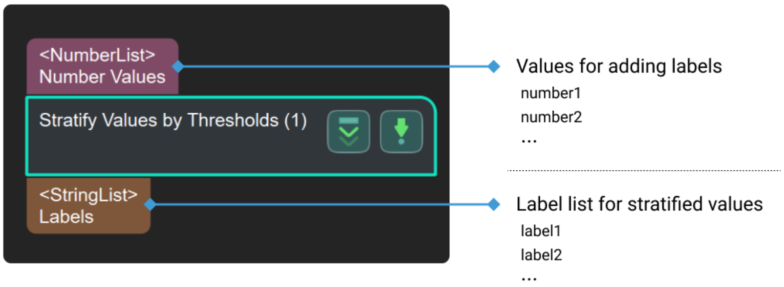


Usage Scenario

This Step sorts and labels data by values. It is usually used with Validate Labels and Output Flags to filter the data list by value intervals.

If you need to divide the data list into two parts by a threshold and filter out one of them, please use Dichotomize Values by Threshold together with Filter instead.

Input and Output



Parameter Description

File Settings

Threshold File Name

Description: This parameter is used to set one or more thresholds and save the thresholds as a JSON file. The input values will be sorted based on the threshold.

Value type: integer or float.

> Data of other types will be considered invalid and cannot be input successfully.

Tuning recommendation: You can set one or more thresholds.

Setting procedures:

1. Click [**Set thresholds**] to open the JSON editor window.
2. Click ＋, input value in the pop-up window, and click [**OK**] to add a single threshold.
3. To add multiple thresholds, please repeat Step 2.

> If you have added multiple thresholds, please ensure that the added thresholds are sorted in ascending order. You can select the threshold and click ⌃ or ⌄ to adjust the order.

4. Click [**…**] in **Directory** to select a directory to store the JSON file. The default directory is the project folder.
5. Enter a **JSON File Name** to name the threshold file and click [**Save**].
6. Click [**OK**] to finish adding the threshold.

> If there is an existing threshold file, you can click [**…**] in **JSON File Name** to open the file directly.

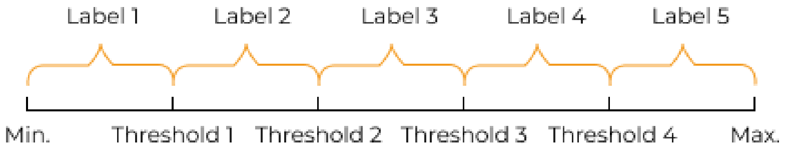
Label File Name

Description: This parameter is used to set labels after sorting the data and save the labels as a JSON file.

Data type: strings (without special characters)

> Except strings, other types of data will be considered invalid here and cannot be input successfully.

Instruction: the number of labels = the number of thresholds + 1. The correspondence between thresholds and labels is determined by their sorting, irrelevant to the label names.



Setting procedures:

1. Click **[Set labels]** to open the JSON Editor window.
2. Click ➕, input a string in the pop-up window, and click **[OK]** to add a single label.
3. To set multiple labels, please repeat Step 2. The number of labels should be one more than the number of added thresholds.

> To align the labels with the added thresholds, please ensure that the order of the labels corresponds with that of the thresholds displayed in the JSON Editor window. You can click ⌃ or ⌄ to adjust the order of the labels.

4. **Directory** refers to the directory of the current project. Click **[…]** on the right to change the directory of the project according to actual needs.
5. Name the file in **JSON File Name** and click **[Save]** to save the label in the file.
6. Click **[OK]** to finish adding the label.

> If there is an existing label file, you can click **[…]** in **JSON File Name** to open the file directly.

4.3.350. Subtract Real Numbers

> The contents of this Step are no longer under maintenance. Please use the new Step Numerical Operation. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

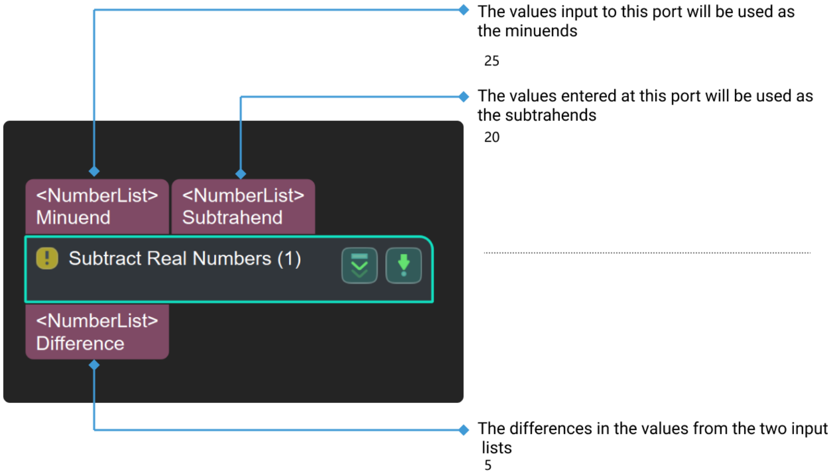
Function

Subtracts each value input at the second port from each value input at the first port to output the differences between the elements of the two input lists one-to-one.

Usage Scenario

This Step is outdated. Please use the newer Step Numeric Operation that has more functions instead.

Input and Output



The values input to this port will be used as the minuends

25

The values entered at this port will be used as the subtrahends

20

The differences in the values from the two input lists

5

4.3.351. Suppress Neighboring Poses with Low Scores (NMS)

The contents of this Step are outdated. Please use the new Step 3D Coarse Matching. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

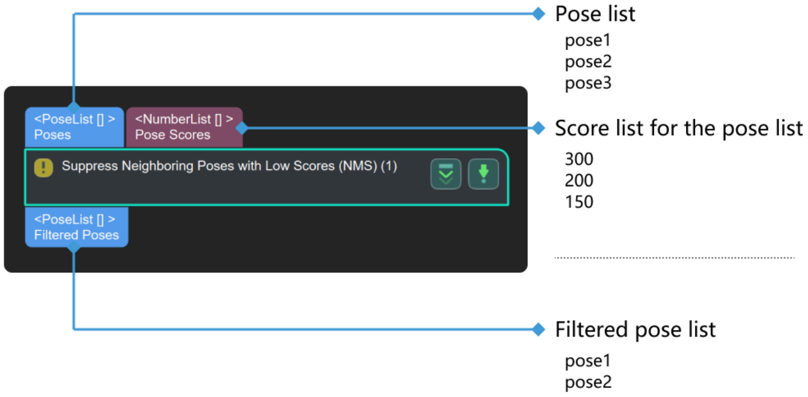
Function

Find the pose with the highest score within the set range and filter out the poses with low scores in its vicinity according to the set threshold.

Usage Scenario

One of the methods for filtering poses.

Input and Output



Pose list

pose1
pose2
pose3

Score list for the pose list

300
200
150

Filtered pose list

pose1
pose2

Parameter Description

Neighboring Poses Search Settings

Default value: 50.000 mm

Instruction: Define the range of the local search (mm). If it is larger, more poses are involved in the calculation, which may result in a longer processing time for later step.

Threshold Settings

Default value: 0.02

Instruction: When the pose score is less than the threshold, it is not considered.

4.3.352. Template Matching

Function

This Step matches the objects in the input image with the 2D template to calculate the 2D poses of the objects.

Usage Scenario

This Step is used to locate the target objects in the image, in order to calculate the poses or the 2D transformation for image position and orientation correction in subsequent Steps.

Input and Output

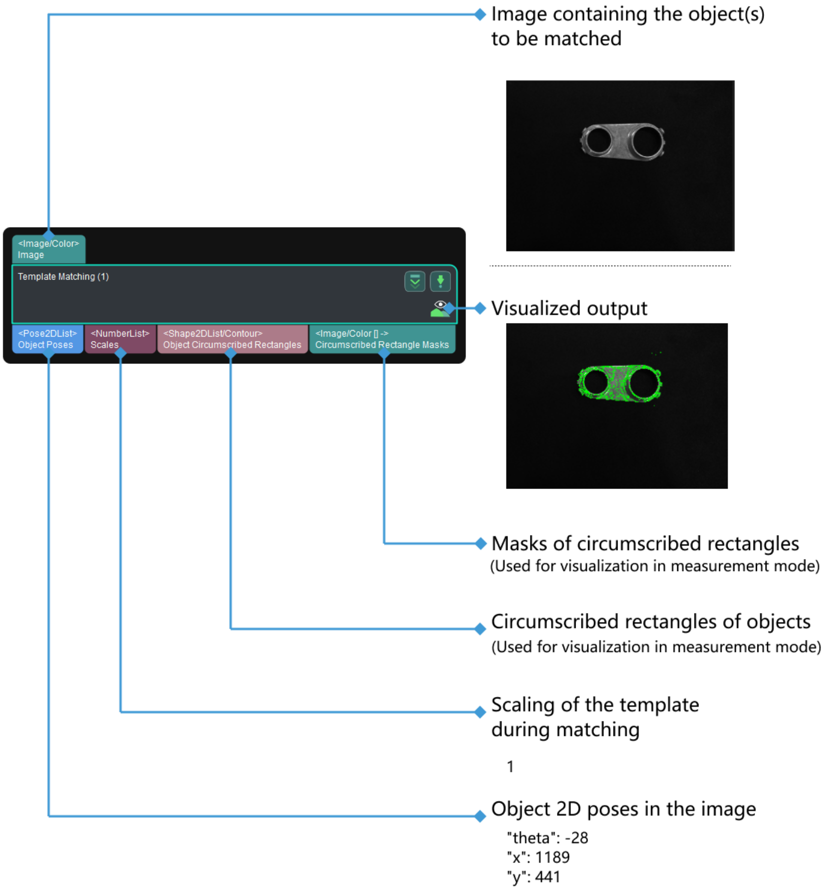
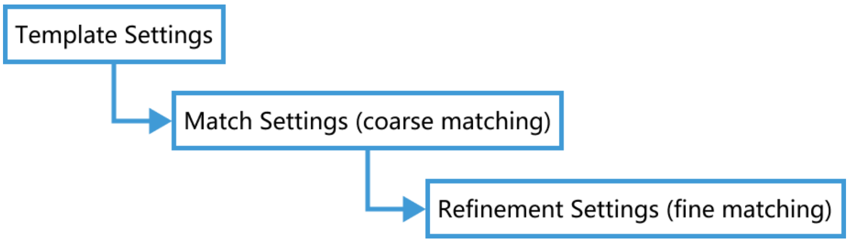


Image containing the object(s) to be matched
<Image/Color>
Image
Template Matching (1)
Visualized output
<Pose2DList>
Object Poses
<NumberList>
Scales
<Shape2DList/Contour>
Object Circumscribed Rectangles
<Image/Color [] ->
Circumscribed Rectangle Masks
Masks of circumscribed rectangles
(Used for visualization in measurement mode)
Circumscribed rectangles of objects
(Used for visualization in measurement mode)
Scaling of the template during matching
1
Object 2D poses in the image
"theta": -28
"x": 1189
"y": 441

Parameter Description

The workflow of adjusting the parameters of this Step is as follows:



Template Settings

Match Settings (coarse matching)

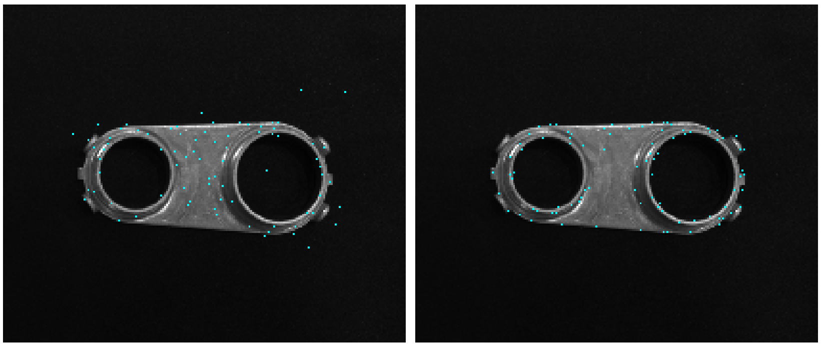
Refinement Settings (fine matching)

Gradient Magnitude Lower Threshold

Default value: 60

Instructions: Object features must be extracted before matching. Pixels with gradient magnitudes greater than this threshold are regarded as feature pixels for matching. The following figure shows the extracted feature pixels of the same object when Gradient Magnitude Lower Threshold is set to 10 (left) and 60 (right).

When it is set to 10, the feature pixel extraction is not quite accurate, and some feature pixels fall on the background. When it is set to 60, the feature pixel extraction is more accurate. Therefore, an appropriate increase in the threshold is conducive to raising the accuracy of feature pixel extraction.



Template Features Already Extracted

Default value: Unchecked.

Instructions: When this Step is run for the first time, this option is unchecked by default. Therefore, the feature pixels of the imported template image (from Template Image File Path) are extracted and saved to Template Feature File Path. Then, the feature pixels of the input image are extracted, and matching is performed. When this Step is run again, this option is automatically checked, and matching is performed using the saved feature pixels of the template image. If you need to extract the feature pixels of the template image, such as when you changed the template image, please uncheck this option and run this Step.

Template Settings (displayed when Template Features Already Extracted is unchecked)

Template Feature File Path

Instructions: Select the directory for saving the template feature files.

Template Image File Path

Instructions: Select the template image output by Make Template.

Template Mask File Path

Instructions: Select the template mask output by Make Template.

Starting Angle

Default value: −180.0

Instructions: Starting Angle and End Angle specify the rotation range of the template in the input image during matching.

End Angle

Default value: 180.0

Instructions: Starting Angle and End Angle specify the rotation range of the template in the input image during matching.

Angle Step

Default value: 1.0

Instructions: Angle Step determines the rotation increment of matching attempts.

Starting Scale

Default value: 0.80

Instructions: Starting Scale and End Scale specify the scaling range of the template in the input image during matching.

End Scale

Default value: 1.20

Instructions: Starting Scale and End Scale specify the scaling range of the template in the input image during matching.

Scale Step

Default value: 0.20

Instructions: Scale Step determines the scaling increment of matching attempts.

Match Settings (displayed when **Template Features Already Extracted** is checked)

Matching Score Lower Threshold

Default value: 50.0

Instructions: Matching results with scores lower than this threshold are discarded, and the remaining results will be output after being filtered in the refinement process of this Step.

- If some objects are blocked by others, this threshold should be lowered accordingly.
- If this threshold is set too high, all results might be discarded and none will be output.

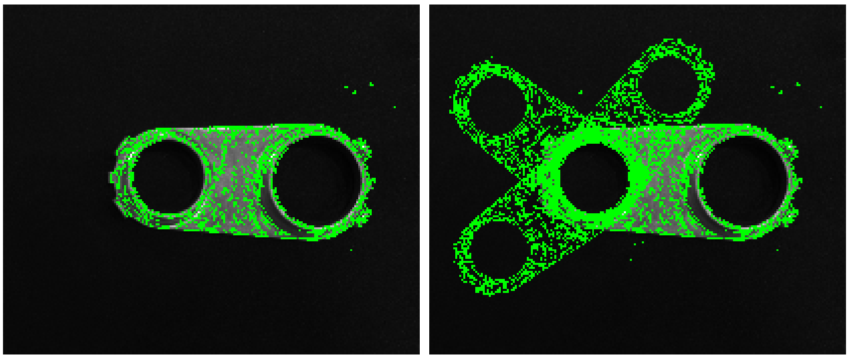
Upper Limit for Number of Matches

Default value: 3

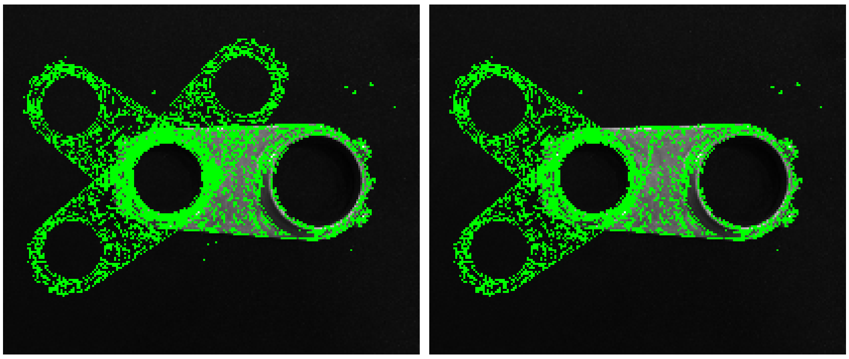
Instructions: If the actual number of matching results is greater than this value, the results at the end of the list are discarded.

- When **Matching Score Lower Threshold** is kept constant, the higher the value of **Upper Limit for Number of Matches**, the more matching results may be output.

  - When **Matching Score Lower Threshold** is set to 25, and **Upper Limit for Number of Matches** is set to 1 or 4, the matching results are shown in the following figures. The left figure shows the matching result when **Upper Limit for Number of Matches** is set to 1, while the right figure shows the matching result when **Upper Limit for Number of Matches** is set to 4.



- When **Upper Limit for Number of Matches** is kept constant, the higher the value of **Matching Score Lower Threshold**, the fewer matching results may be output. When **Upper Limit for Number of Matches** is set to 4, and **Matching Score Lower Threshold** is set to 25 or 50, the matching results are shown in the following figures. The left figure shows the matching result when **Matching Score Lower Threshold** is set to 25, while the right figure shows the matching result when **Matching Score Lower Threshold** is set to 50.



In the figure above, because only 3 matching results have scores higher than the set value of **Matching Score Lower Threshold**, so only 3 results at most can be output.

Overlap Ratio Upper Threshold

Default value: 0.40

Instructions: If two matching results overlap and the overlap ratio exceeds this value, the result with the higher matching score is kept, and the other one is discarded.

Refinement Settings (displayed when **Template Features Already Extracted** is checked)

Search Radius

Default value: 10.0

Instructions: The search radius is used to find the correspondences in pose refinement. When the matching results are not very good, please increase this value accordingly.

4.3.353. Test

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step is used for testing by developers.

Usage Scenario

This Step is used by developers for testing. Regular users do not need to use this Step in general application scenarios.

Input and Output

- **Input:** none.
- **Output:** none.

4.3.354. Path Target Matching

This Step is outdated. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step transforms the model and matches the model with the input point cloud.

Usage Scenario

This Step is a custom Step for special scenarios.

Input and Output

- **Input:**
  1. Scene Point Cloud
- **Output:**
  1. Transformed model point cloud
  2. Transformation matrix

4.3.355. Transform Image

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step corrects the input image’s position by the input transformation pose so that the corrected image is aligned with the template image.

Usage Scenario

This Step is generally used for correcting images in measurement scenarios so that the objects in all images are aligned in position and orientation. This Step is usually used with the Step Record Criterion Pose and Calc Transformation.

Input and Output

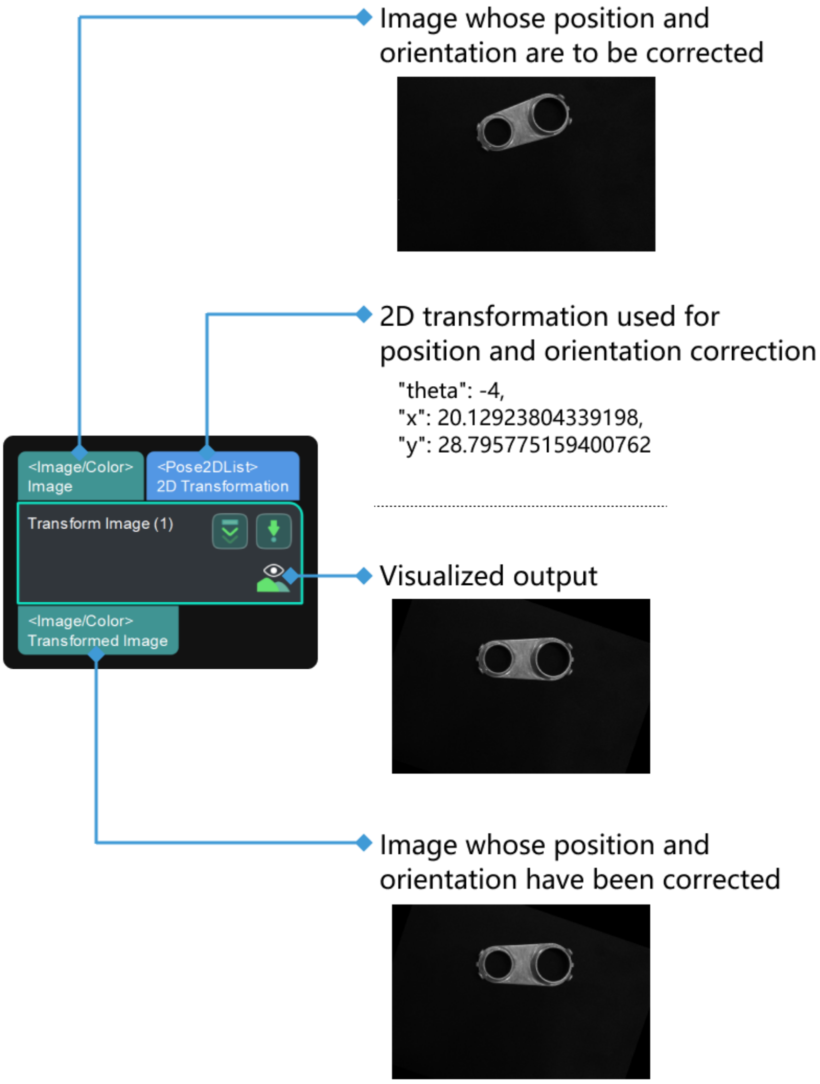


Image whose position and orientation are to be corrected
2D transformation used for position and orientation correction
<Image/Color> Image
<Pose2DList> 2D Transformation
Transform Image (1)
Visualized output
<Image/Color> Transformed Image
Image whose position and orientation have been corrected

"theta": -4,
"x": 20.12923804339198,
"y": 28.795775159400762

4.3.356. Align Plane Point Clouds

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

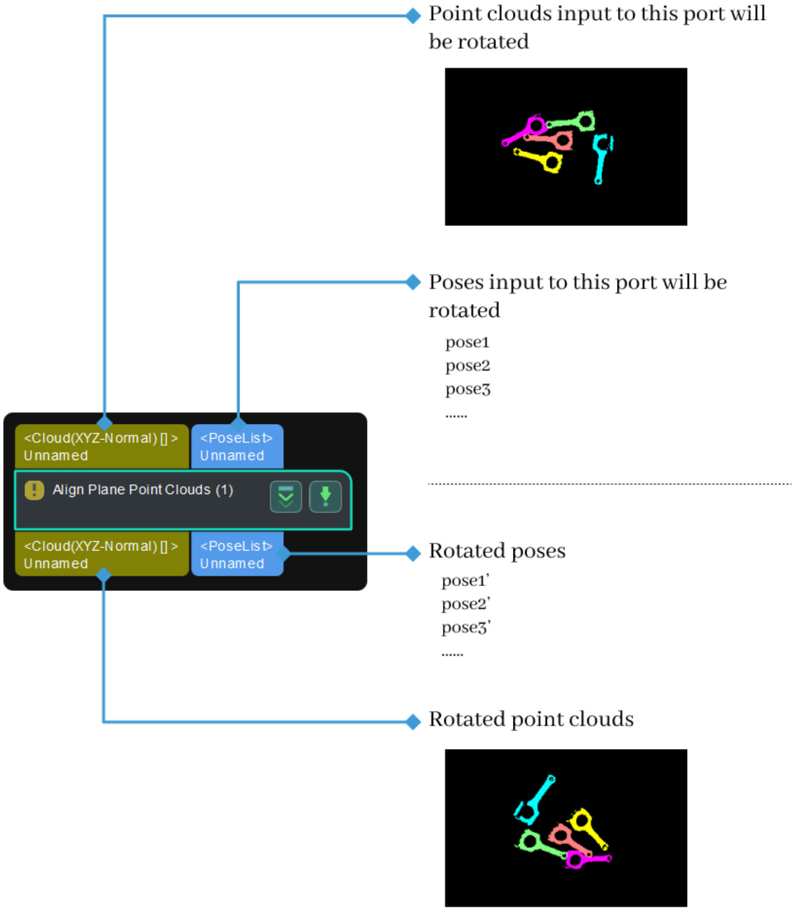
Function

This Step rotates the input point clouds and the corresponding poses according to the set directions.

Usage Scenario

This Step is a custom Step for special scenarios.

Input and Output



4.3.357. Transform Point Clouds for Truss

Function

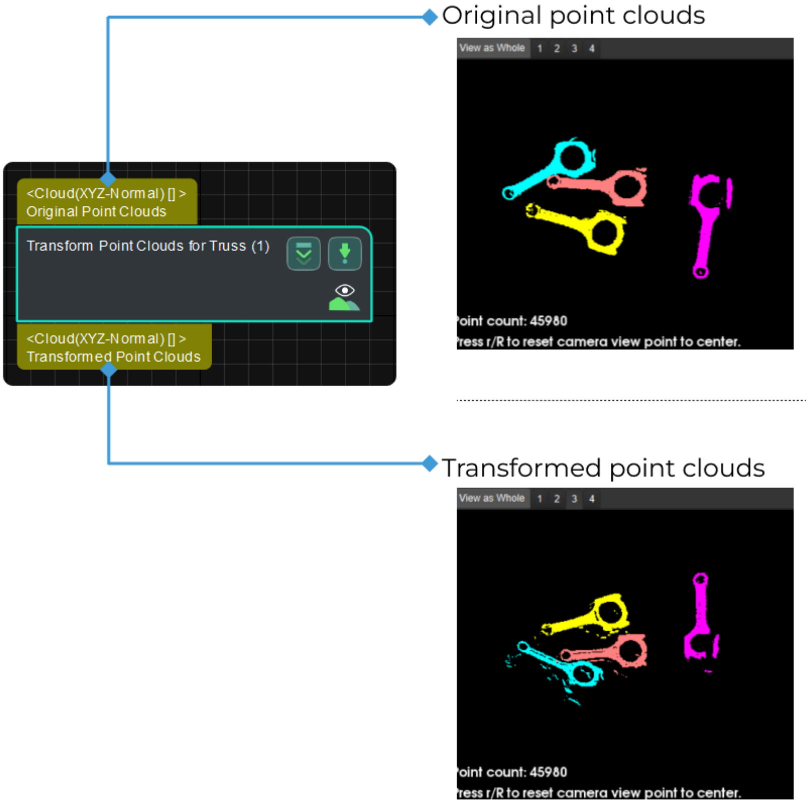
This Step is used to transform the input point clouds to the camera reference frame or the truss system reference frame and then output the transformed point clouds.

Usage Scenario

This Step is designed to transform the reference frame of point clouds in automated truss system projects.

> If you are using a robot other than the automated truss system, please use the Transform Point Clouds Step.
>
> If you use the wrong transform Step, the position of the point cloud after the transformation will be incorrect, and therefore incorrect pick points will be output.

Input and Output



Parameter Description

Transformation Type

Parameter description: This parameter is used to select the type of point cloud transformation.

Options: CameraToRobot, RobotToCamera.

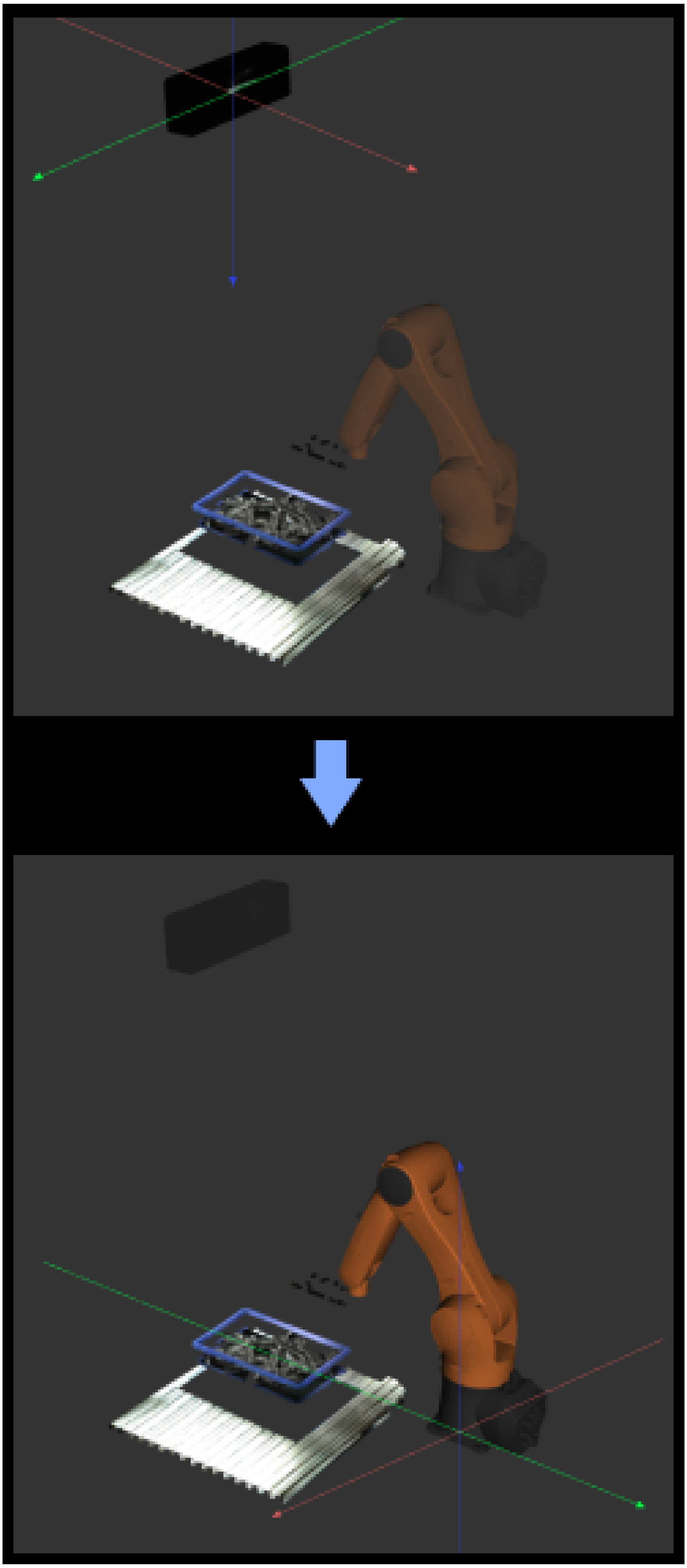
- CameraToRobot: Transform the point clouds from the camera reference frame to the truss system reference frame. Please refer to the tuning example for the corresponding result.
- RobotToCamera: Transform the point clouds from the truss system reference frame to the camera reference frame. Please refer to the tuning example for the corresponding result.

Default setting: CameraToRobot.

4.3.358. Transform Point Clouds

Function

This Step can transform point clouds from one reference frame to another, or transform within its own reference frame according to the reference pose.

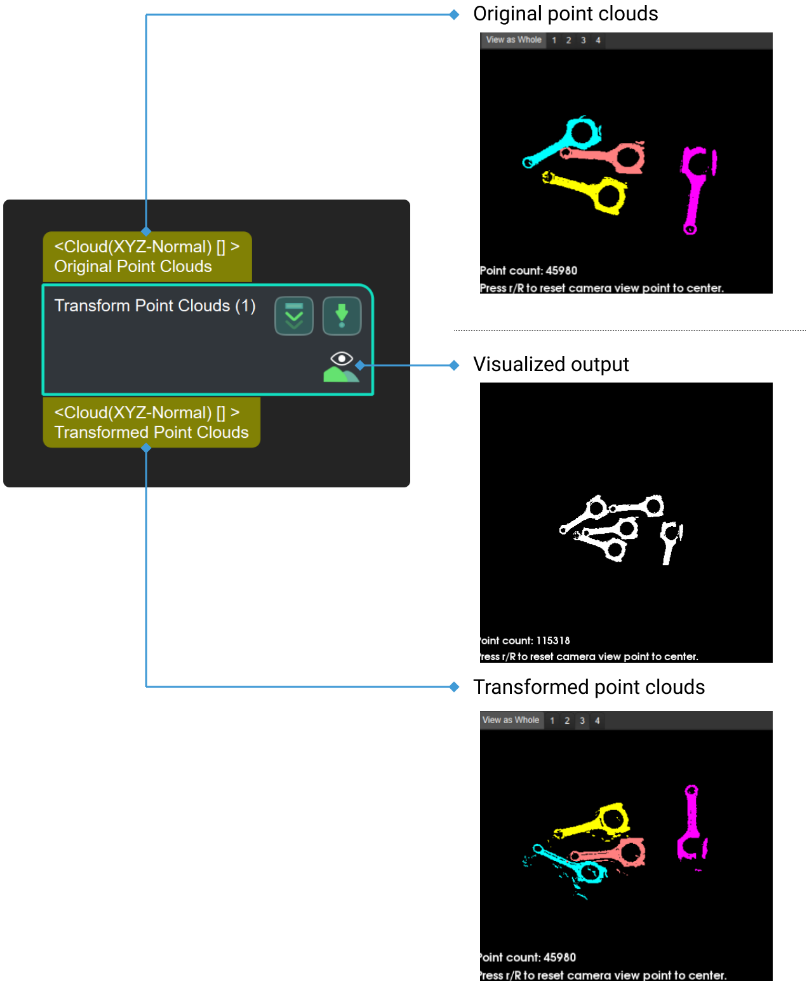


Usage Scenario

This Step is usually used to transform point clouds between camera reference frame and robot reference frame.

If you are using a truss robot, please use the Transform Point Clouds for Truss Step.

Input and Output



Parameter Description

Transformation Settings

Transformation Type

Parameter description: This parameter is used to select the type of point cloud transformation.

Default setting: CameraToRobot.

Options: CameraToRobot, RobotToCamera, AllWithFirst, FirstWithAll, UseCorrespondenceInput, and AffineTransform.

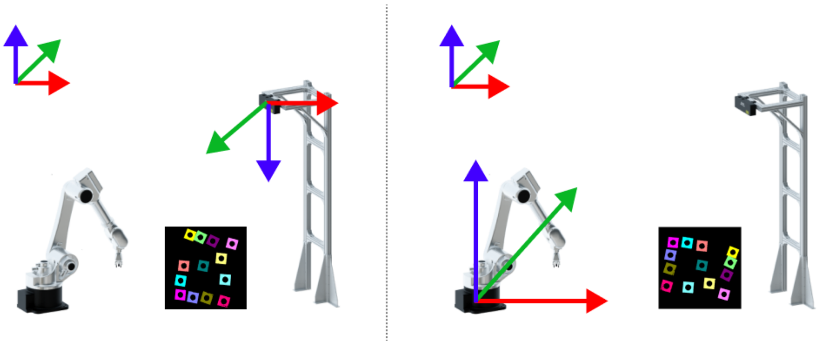
Instructions:

- CameraToRobot: Transform the point clouds from the camera reference frame to the robot reference frame. Please refer to the tuning example for the corresponding result.
- RobotToCamera: Transform the point clouds from the robot reference frame to the camera reference frame. Please refer to the tuning example for the corresponding result.
- AllWithFirst: Transform the original point clouds to another reference frame according to the first reference pose input. Please refer to the tuning example for the corresponding result.
- FirstWithAll: Transform the first original point clouds input to another reference frame according to reference poses in the reference pose list which should be input. Please refer to the tuning example for the corresponding result.
- UseCorrespondenceInput: Transform the original point clouds to another reference frame according to the corresponding reference poses respectively. Please refer to the tuning example for the corresponding result.
- AffineTransform: Transform the point clouds with an affine matrix. The pose list should be converted to matrix4D format with the Step From PoseList to Matrix4D first and then input. Please refer to the tuning example for the corresponding result.

Tuning Examples

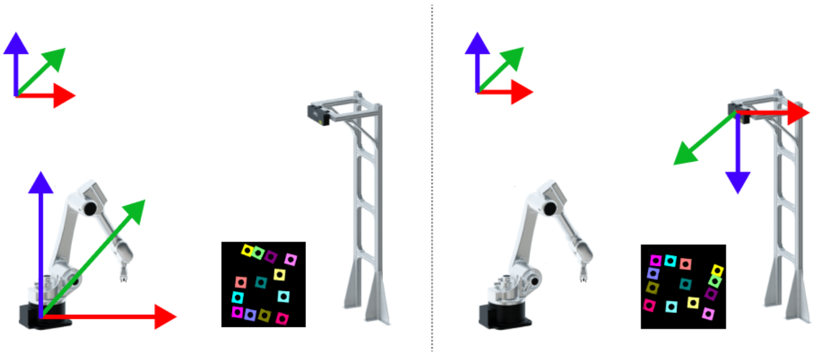
CameraToRobot

The following figure shows the results before (left) and after (right) point cloud transformation of **CameraToRobot**.



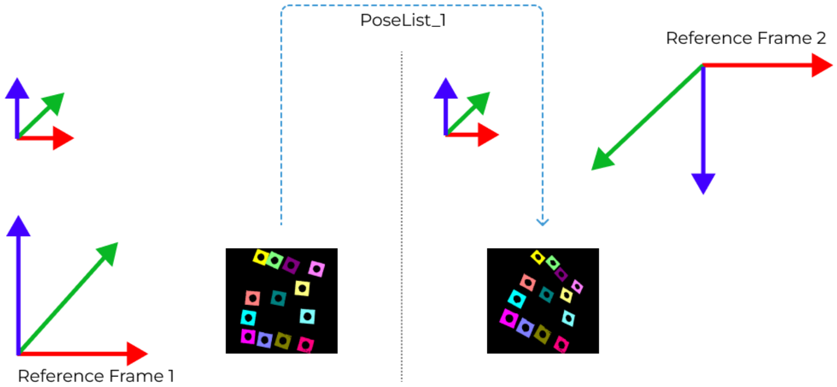
CameraToRobot

The following figure shows the results before (left) and after (right) point cloud transformation of **RobotToCamera**.



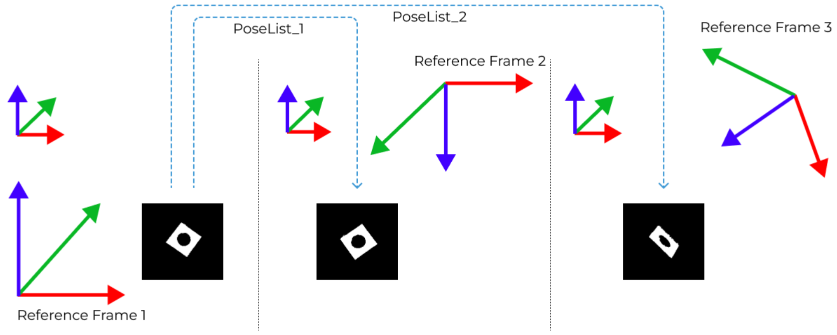
AllWithFirst

The following figure shows the results before (left) and after (right) point cloud transformation of **AllWithFirst**.



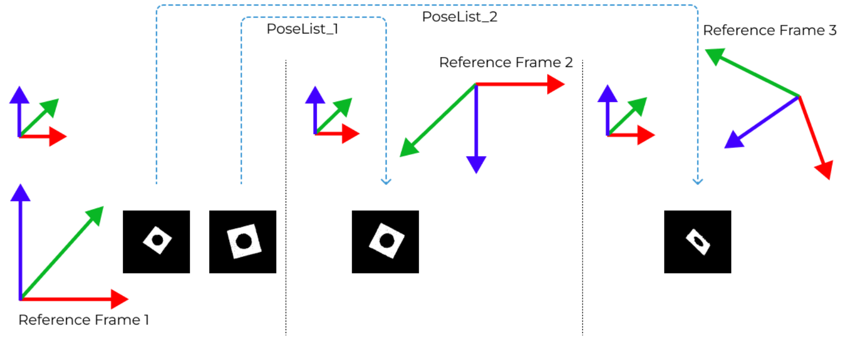
FirstWithAll

The following figure shows the point-cloud transformation result of **FirstWithAll**.



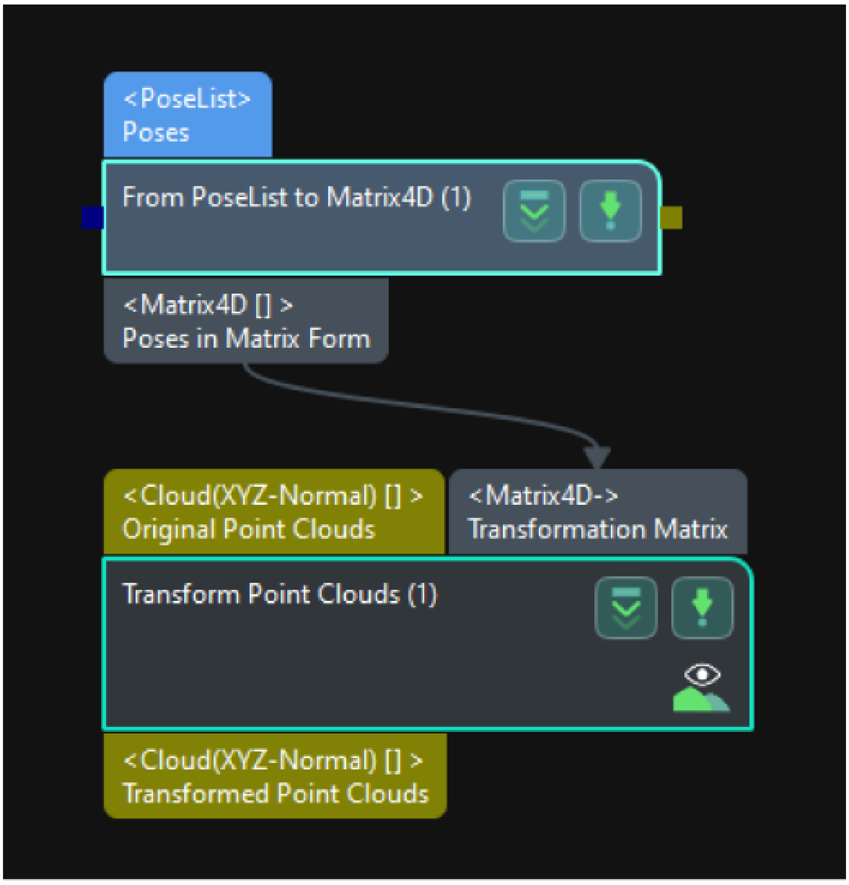
UseCorrespondenceInput

The following figure shows the point-cloud transformation result of **UseCorrespondenceInput**.



AffineTransform

The following figure shows the point-cloud transformation result of **AffineTransform**.



4.3.359. Transform Poses for Truss

Function

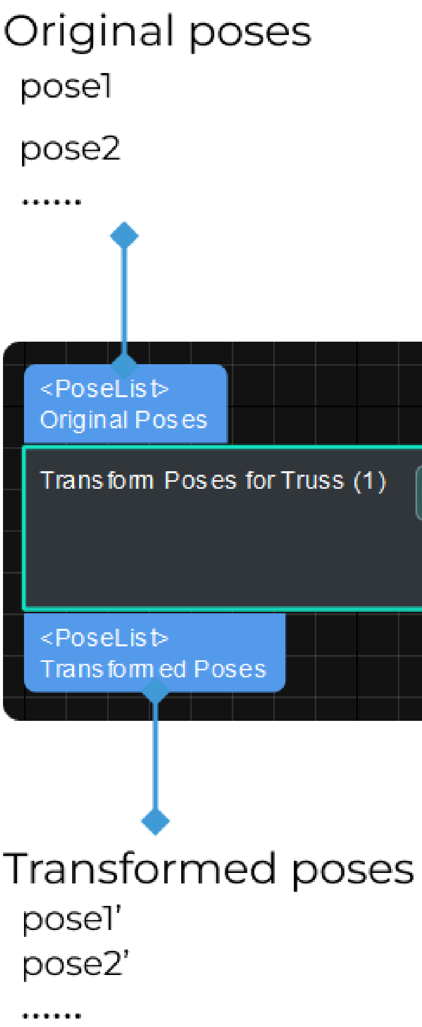
This Step is used to transform the input poses to the camera reference frame or the truss system reference frame and then output the transformed poses.

Usage Scenario

This Step is designed to transform the reference frame of poses in automated truss system projects.

> If you are using a robot other than the automated truss system, please use the Transform Poses Step.
>
> If you use the wrong transform Step, the transformed poses will be incorrect and therefore incorrect pick points will be output.

Input and Output



Parameter Description

Transformation Type

Parameter description: This parameter is used to select the type of pose transformation.

Options: CameraToRobot, RobotToCamera.

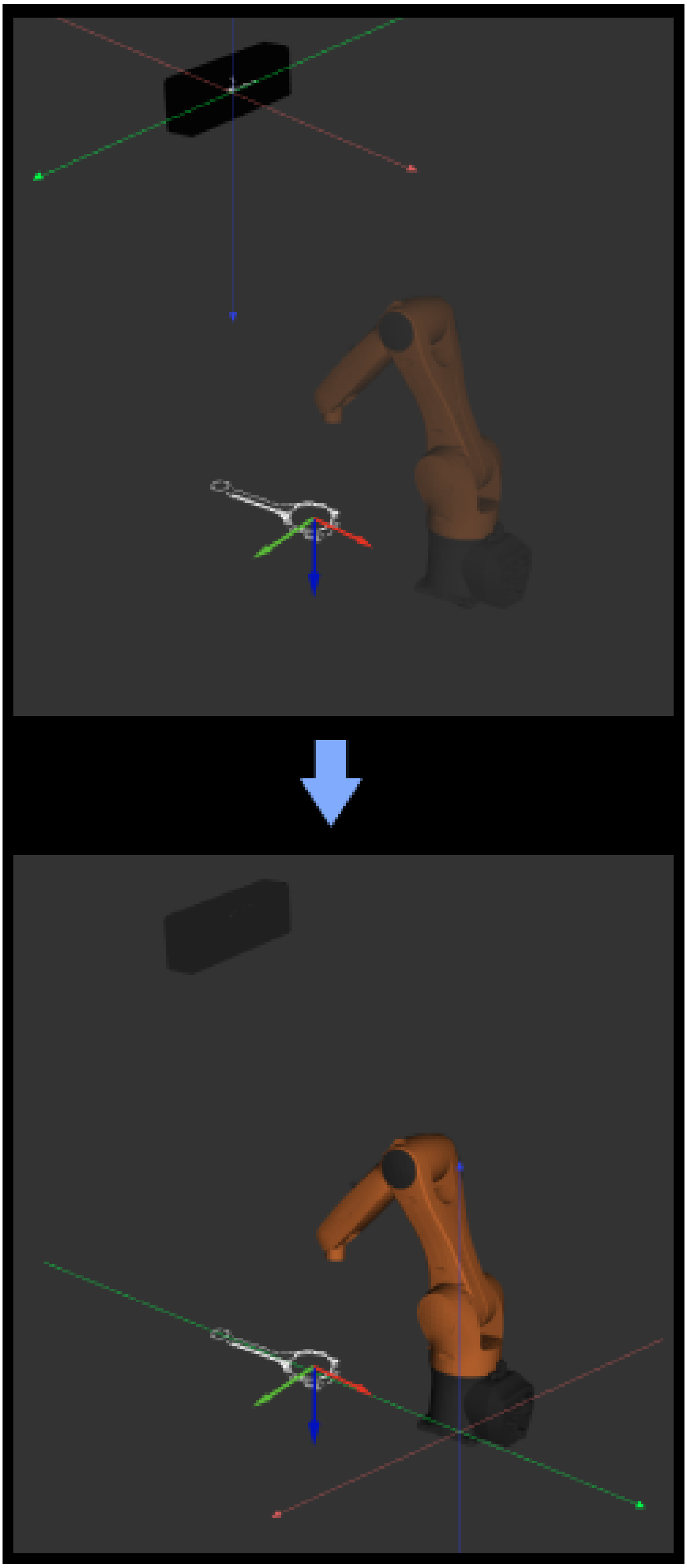
- CameraToRobot: Assuming that the object pose in the camera reference frame is Pose 1, when the transformation type is **CameraToRobot**, the transformed pose Pose2 in the truss system reference frame will be output. Please refer to the tuning example for the corresponding result.
- RobotToCamera: Assuming that the object pose in the truss system reference frame is Pose 1, when the transformation type is **RobotToCamera**, the transformed pose Pose2 in the camera reference frame will be output. Please refer to the tuning example for the corresponding result.

Default setting: CameraToRobot.

4.3.360. Transform Poses

Function

This Step can transform poses from one reference frame to another, or transform within its own reference frame according to the reference pose.

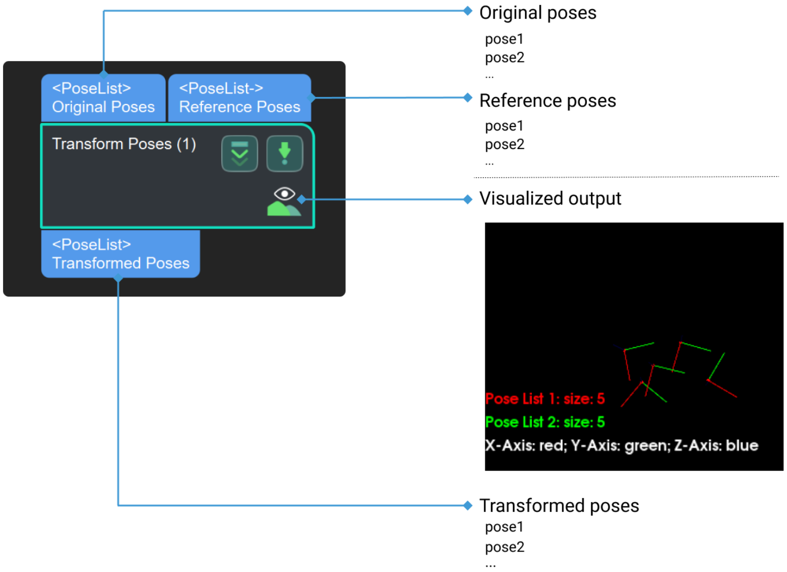


Usage Scenario

This Step is usually used to transform poses between camera reference frame and robot reference frame.

If you are using a truss robot, please use the Transform Poses for Truss Step.

Input and Output



Description

Visualization Settings

Show Point Cloud In Selected Frame

Description: Display point cloud under the robot reference frame or the camera reference frame.

Default setting: Robot Coordinate.

Options: Robot Coordinate, Camera Coordinate.

Instruction: When this parameter is set to **Robot Coordinate**, the Debug Output panel will display the visualized output under the camera reference frame; when it is set to **Robot Coordinate**, the panel will display the visualized output under the robot reference frame. Please refer to the tuning example for the corresponding result.

This parameter will appear only when the scene point cloud is set in the project.

- Under the **Robot Coordinate**, the Debug Output panel is black, and you need to adjust the view to show the point cloud and coordinates.
- Under the **Camera Coordinate**, the Debug Output panel only shows the point cloud, and you need to adjust point cloud orientation to show the coordinates.

Pose Type to Visualize

Parameter description: This parameter is used to select the pose type to visualize.

Default setting: After.

Options: Before, After, and All.

Instruction: Select an appropriate option according to actual needs. Please refer to the tuning example for the corresponding result.

Transformation Settings

Transformation Type

Parameter description: This parameter is used to select the type of pose transformation.

Default setting: CameraToRobot.

Options: CameraToRobot, RobotToCamera, AllWithFrst, FirstWithAll, and UseCorrespondenceInput.

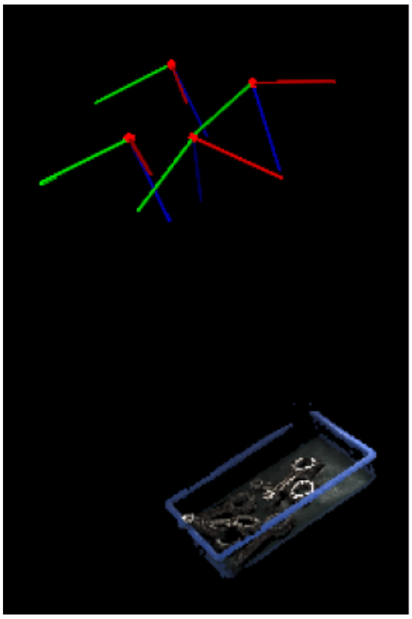
Instructions:

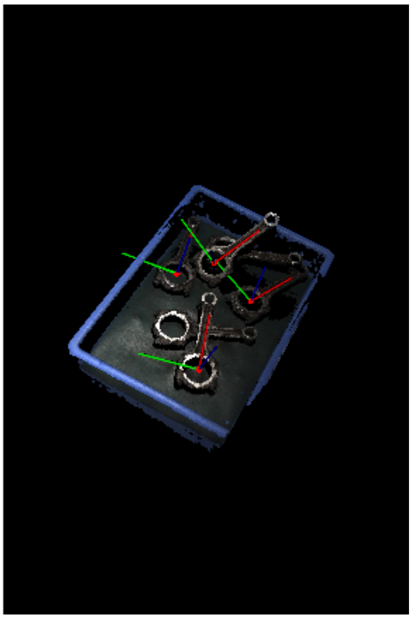
- CameraToRobot: Assuming that the object pose in the camera reference frame is Pose 1, when the transformation type is **CameraToRobot**, the transformed pose Pose2 in the robot reference frame will be output. Please refer to the tuning example for the corresponding result.
- RobotToCamera: Assuming that the object pose in the robot reference frame is Pose 1, when the transformation type is **RobotToCamera**, the transformed pose Pose2 in the camera reference frame will be output. Please refer to the tuning example for the corresponding result.
- AllWithFirst: The poses of objects A and B are denoted as Pose1A and Pose1B, respectively, in the reference frame 1. When the transformation type is selected as **AllWithFirst**, and a reference pose list has been input, the transformed poses of object A and B, namely, Pose2A and Pose2B, respectively, in the reference frame 2 will be output. The first reference pose in the pose list is used in the transformation of all original poses. Please refer to the tuning example for the corresponding result.
- FirstWithAll: The poses of objects A and B are denoted as Pose1A and Pose1B, respectively, in the reference frame 1. When the transformation type is selected as **FirstWithAll**, and reference pose lists have been input, the transformed poses of object A, namely, Pose2A and Pose3A, in the reference frame 2 and reference frame 3, respectively, will be output. All reference poses in the pose lists are used in the transformation of the first original pose, which is Pose1A in this example. Please refer to the tuning example for the corresponding result.
- UseCorrespondenceInput: The poses of objects A and B are denoted as Pose1A and Pose1B, respectively, in the reference frame 1. When the transformation type is selected as **UseCorrespondenceInput**, and reference pose lists have been input, the transformed poses of object A and B, namely, Pose2A and Pose3B, in the reference frame 2 and reference frame 3, respectively, will be output. The reference poses in the pose list are used in the transformation of each original pose correspondingly. Please refer to the tuning example for the corresponding result.

Tuning Examples

Show Point Cloud In Selected Frame

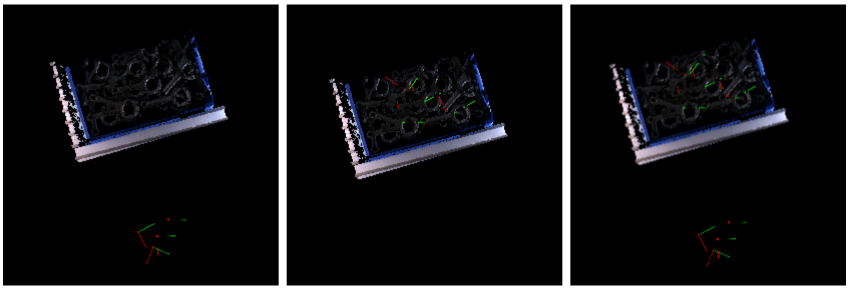
As shown below, the figure on the left shows the point cloud in the camera reference frame, and the figure on the right shows the point cloud in the robot reference frame.





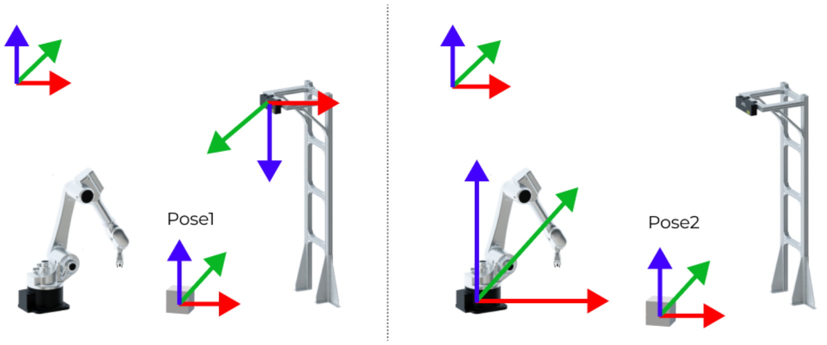
Pose Type to Visualize

In the case of the robot reference frame, the visualized poses are as follows when the Pose Type to Visualize is set to **After**, **Before**, and **All**.



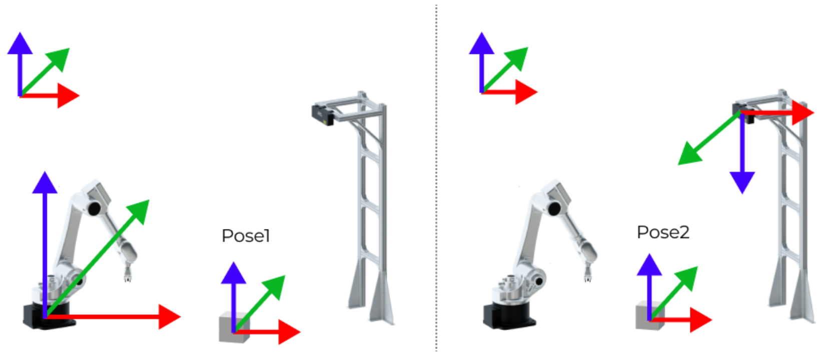
CameraToRobot

The following figure shows the results before (left) and after (right) pose transformation of **CameraToRobot**.



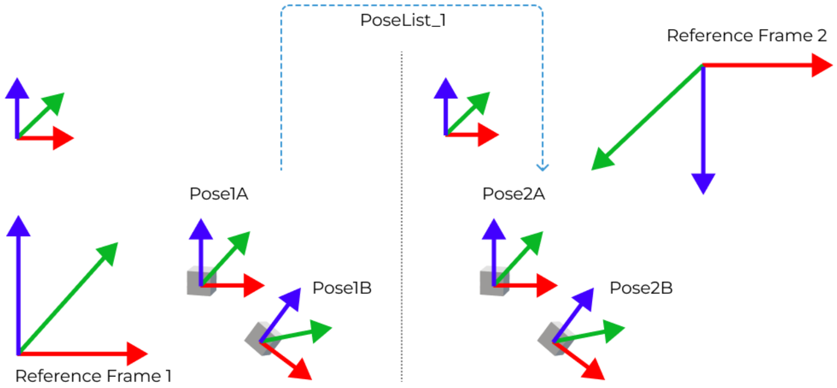
RobotToCamera

The following figure shows the results before (left) and after (right) pose transformation of RobotToCamera.



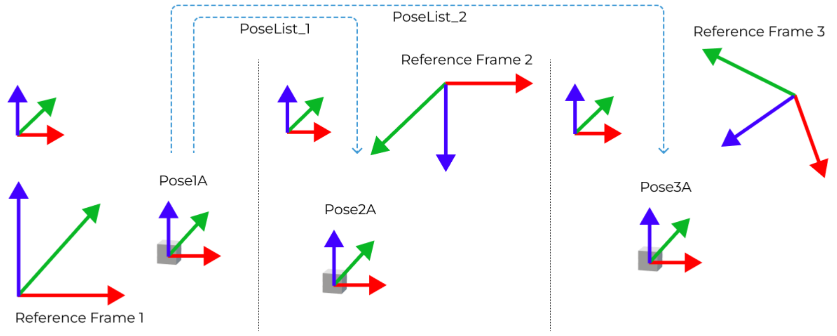
AllWithFirst

The following figure shows the results before (left) and after (right) pose transformation of AllWithFirst.



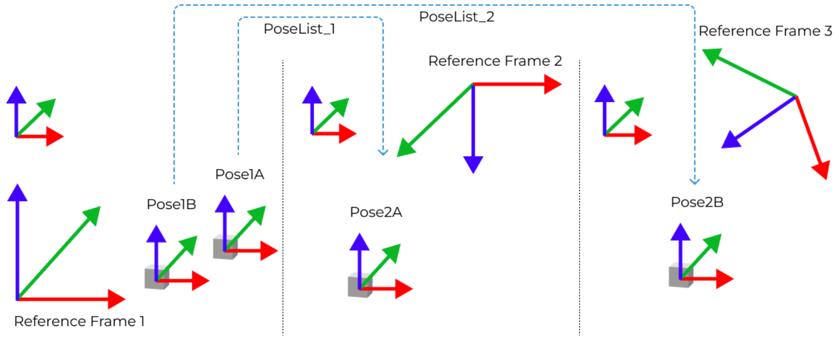
FirstWithAll

The following figure shows the pose transformation result of FirstWithAll.



UseCorrespondenceInput

The following figure shows the pose transformation result of UseCorrespondenceInput.



4.3.361. Translate Poses along Customized Direction by Distances

> Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step translates the input poses. The directions and distances of the translation can be set via the input or parameters.

Usage Scenario

This Step is a general pose translation Step. No fixed usage scenarios.

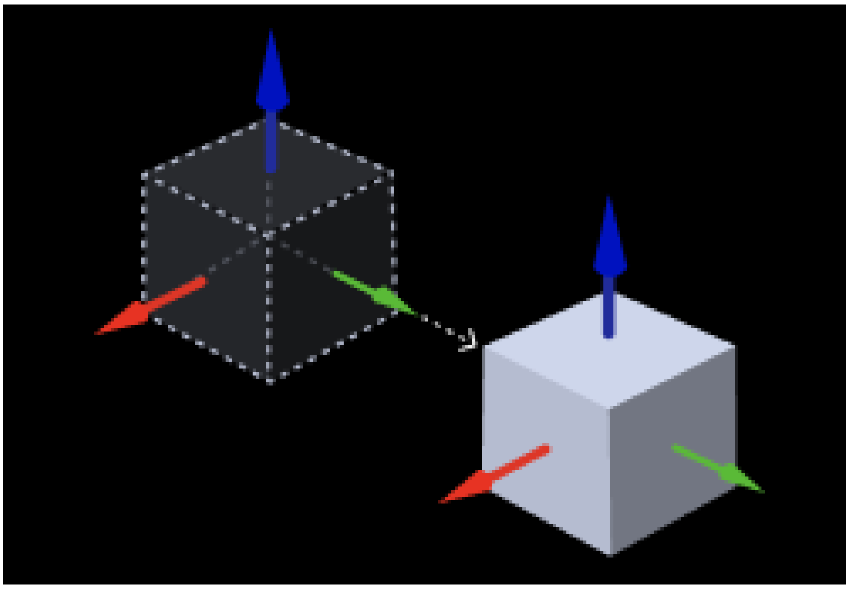
Input and Output

- Input:
  1. Pose input to this port will be translated.
  2. Custom pose translation distances (optional, if this port is not used, the setting in the parameter will be used).
- Output:
  1. Translated poses

4.3.362. Translate Poses along Given Direction

Function

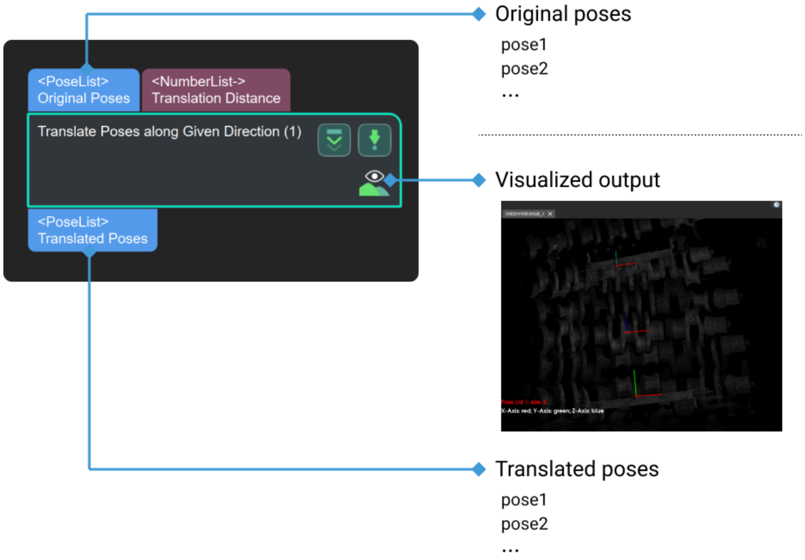
This Step translates poses along a given direction.



Usage Scenario

This Step translates poses according to the direction and distance set by the user.

Input and Output



Parameter Description

Visualization Settings

Show Point Cloud In Selected Frame

Default setting: Robot Coordinate.

List of options: Robot Coordinate, Camera Coordinate.

Instruction: Please refer to Show Point Cloud In Selected Frame for detailed instructions.

Pose Type to Visualize

Default setting: After.

Options: Before, After, and All.

Instruction: Please refer to Show Point Cloud In Selected Frame for detailed instructions.

Translation Direction Settings

Translation Distance

Default value: 0

Instruction: Set the parameter according to actual situation.

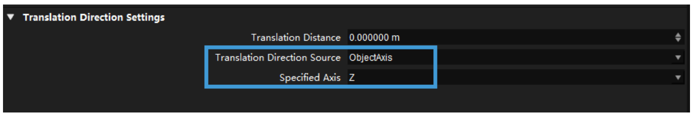
Translation Direction Source

Default setting: CustomizedDirection (translate along the customized direction).

Options: CustomizedDirection (translate along the customized direction), ObjectAxis (translate along the direction of the specified axis of the pose).

Instruction: Set the parameter according to actual situation.

When ObjectAxis is selected, you will need to select an axis according to the actual situation.



▼ Translation Direction Settings

Translation Distance 0.000000 m

Translation Direction Source ObjectAxis

Specified Axis Z

CustomizedDirection (this option will only appear when CustomizedDirection is selected).

X

Default value: 0

Y

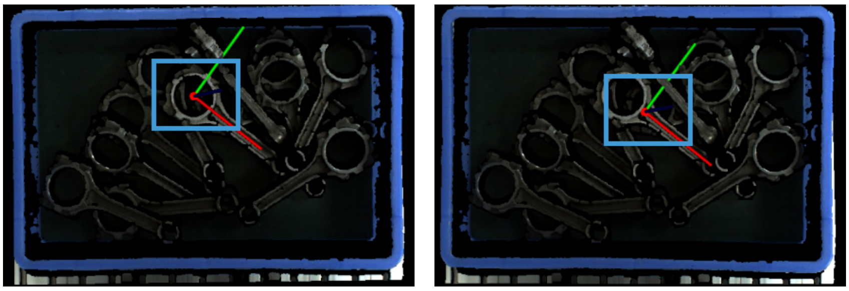
Default value: 0

Z

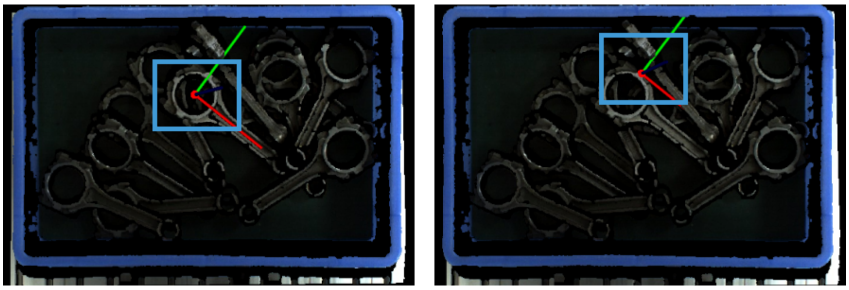
Default value: 1

Examples

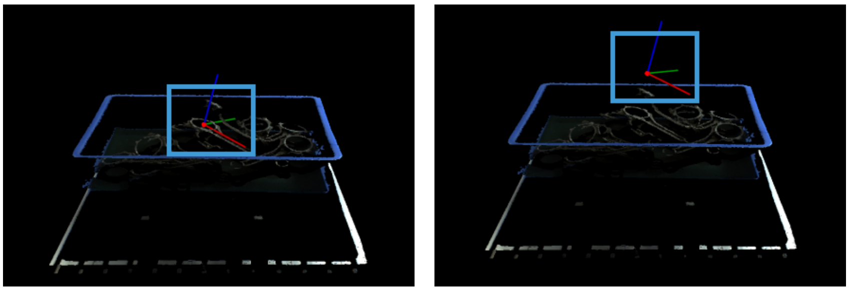
- The figure below shows the pose before and after being translated 0.03 mm along the X axis of the object reference frame. In this example, ObjectAxis is selected as the Translation Direction Source; the specified axis is the X axis; and the Translation Distance is set to 0.03 mm.



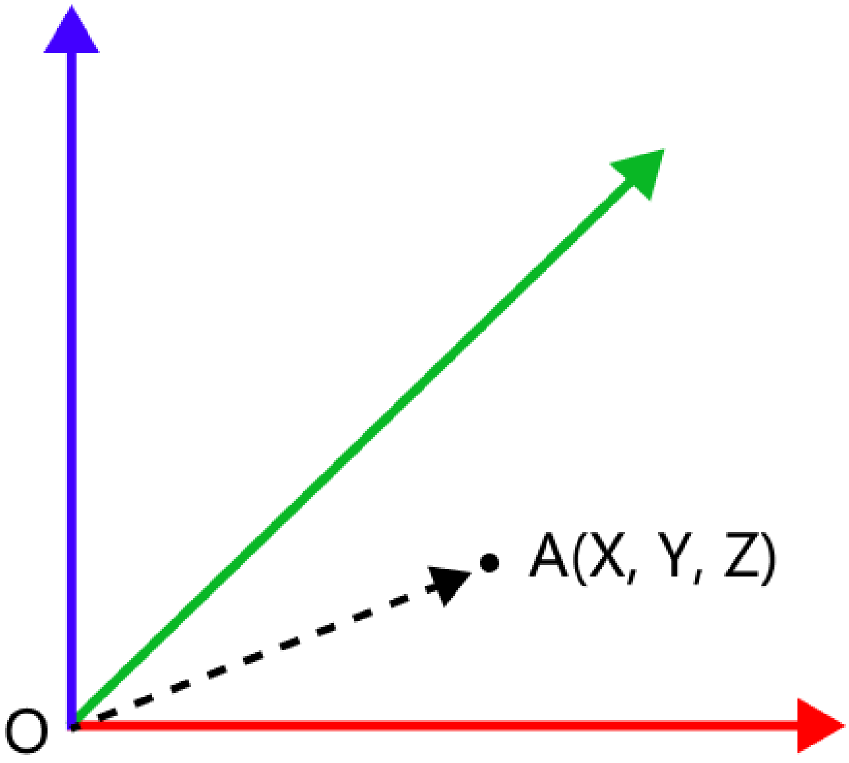
- The figure below shows the pose before and after being translated 0.03 mm along the Y axis of the object reference frame. In this example, ObjectAxis is selected as the Translation Direction Source; the specified axis is the Y axis; and the Translation Distance is set to 0.03 mm.



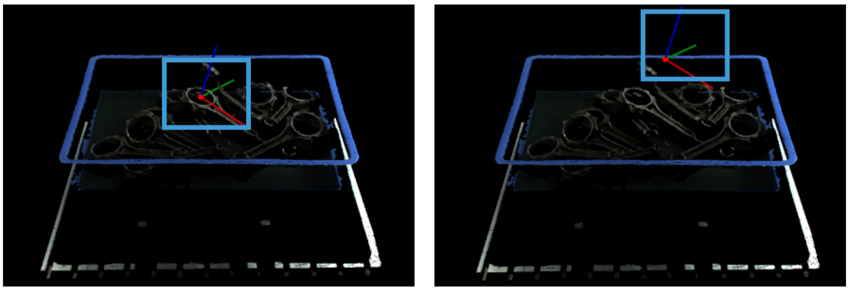
- The figure below shows the pose before and after being translated 0.08 mm along the Z axis of the object reference frame. In this example, ObjectAxis is selected as the Translation Direction Source; the specified axis is the Z axis; and the Translation Distance is set to 0.08 mm.



- When CustomizedDirection is selected as the Translation Direction Source, you will need to set the values of X, Y, and Z in CustomizedDirection. As shown in the figure below, the path from the origin O to A (X, Y, Z) is the direction for the pose translation.



As shown below, in this example, the values of X, Y, and Z are all set to 5, and the Translation Distance is set to 0.08 mm. The figure shows the pose before (left) and after (right) being translated 0.08 mm along the customized direction.



4.3.363. Translate Poses by Direction and Distance

> Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step translates the input poses. The translation directions and distances can be set through the input or parameters.

Usage Scenario

This Step is a general pose translation Step. No fixed usage scenarios.

Input and Output

- Input:
  1. Pose input to this port will be translated.
  2. Custom vectors that define the translation directions of the poses (optional, if this port is not used, the setting in the parameter will be used).
  3. Custom pose translation distances (optional, if this port is not used, the setting in the parameter will be used).
- Output:
  1. Translated poses

4.3.364. Translate Poses by Vector3D

> Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step translates the input poses using custom translation vectors.

Usage Scenario

This Step is a general pose translation Step. No fixed usage scenarios.

Input and Output

- Input:
  1. Poses input to this port will be translated.
  2. Custom vectors used to translate the poses.
- Output:
  1. Translated poses.

4.3.365. Translate Poses Locally by Distances

> Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

Translate the input original poses along the directions of the specified axes. The translation distances can be set by input or parameter.

Usage Scenario

This Step is a general pose translation Step. No fixed usage scenarios.

Input and Output

- Input:
  1. Poses input to this port will be translated.
  2. Translation distances of the poses (optional, if this port is not used, the setting in the parameter will be used).
- Output:
  1. Translated poses

4.3.366. Trigger Control by Flag

> Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

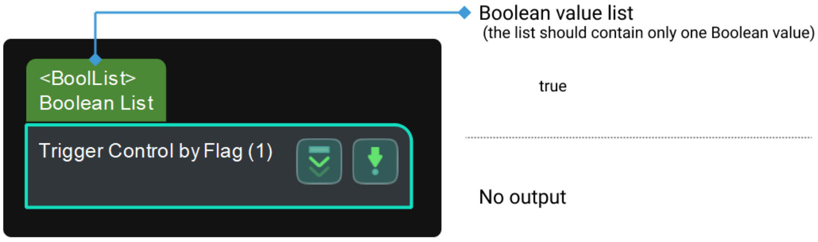
Function

This Step triggers control flow based on an input Boolean value (True/False) and a specified rule.

Usage Scenario

This Step is used in the branch structure of a project to trigger different subsequent Steps according to different rules. Often used with Steps that output Boolean values, such as Dichotomize Values by Threshold.

Input and Output


<BoolList>
Boolean List
Trigger Control by Flag (1)
Boolean value list
(the list should contain only one Boolean value)
true
No output

4.3.367. Trigger

Function

This Step triggers the execution of succeeding Step through control flow.

Usage Scenario

This Step is a custom Step for special scenarios.

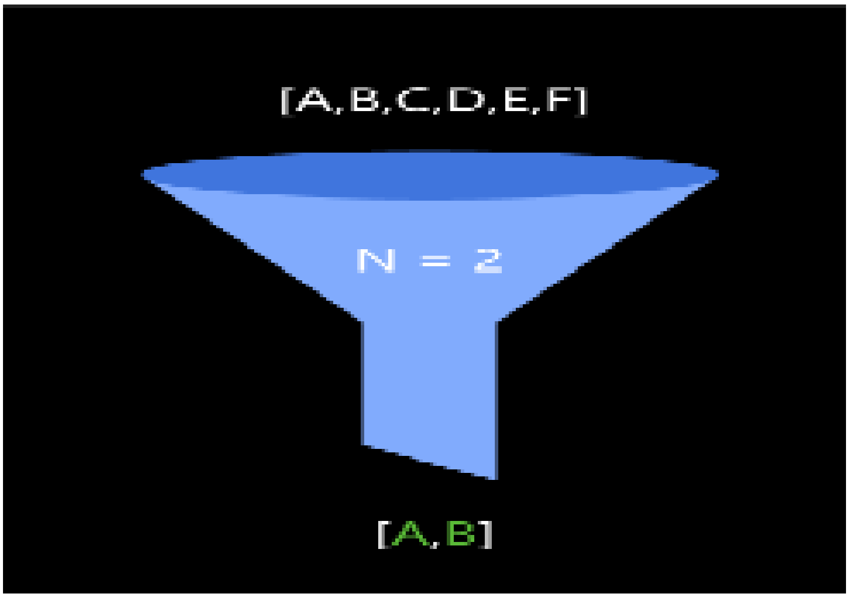
Input and Output

- **Input:** none.
- **Output:** none.

4.3.368. Trim Input List

Function

This Step keeps the first N items in a data list and output. Items after the Nth item will be discarded.

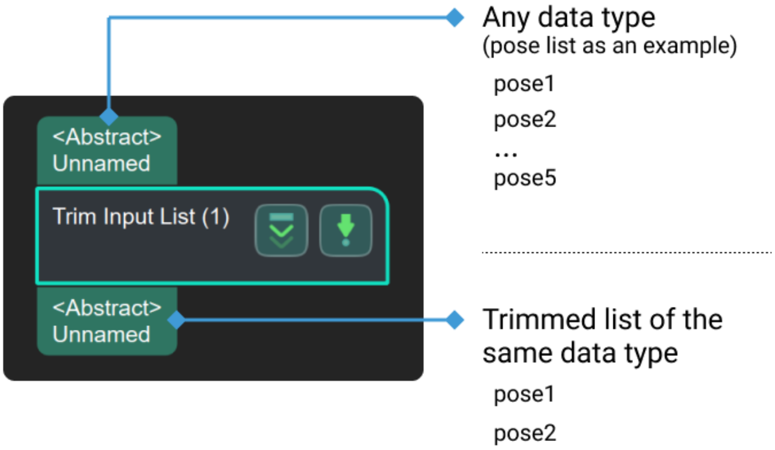


Usage Scenario

This Step extracts the first N items in a list, and the N value can be set in the parameters by project requirements.

Input and Output

For this Step, the input and output are of the same data type. For example, when the input is pose data, the input and output of this Step are shown as follows:



Parameter Description

Ports Settings

Port Number (1–15)

Description: Set the number of input and output ports in this Step.

Default value: 1

Value range: 1–15

Operation Layer

Description: Specify the layer of elements on which operations should be performed in the list. When the value of this parameter is 0, operations will be performed on every element in the list; otherwise, operations will be performed only on elements of the specified layer.

Default value: 0

Output Size

Description: Set the number of outputs in this Step.

Default value: 1

Tuning recommendation: Please set the value according to the actual situation.

4.3.369. Trim Pose List

> ℹ The contents of this step are outdated. Please use the new Steps Trim Input List. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

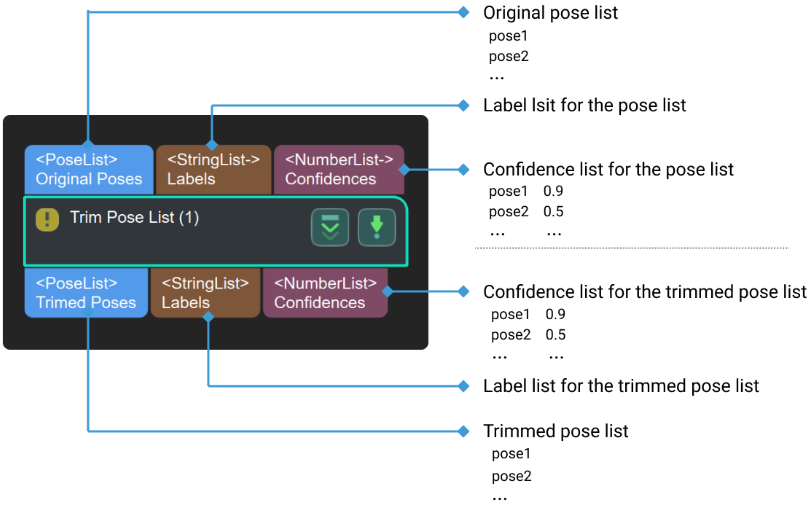
Function

This Step keeps the first N items in a data list and output. Items after the N-th item will be discarded.

Usage Scenario

Extract the top N items in the pose list, and the N value can be set in the Parameter panel by requirements.

Input and Output



Parameter Description

Output Poses Size

Default value: 1

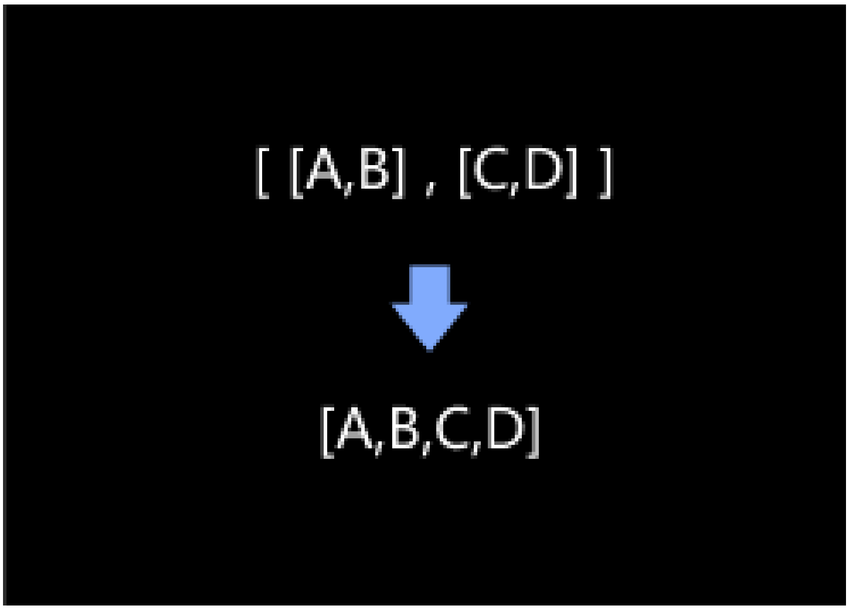
Instruction: Specify the number of output poses.

4.3.370. Unpack and Merge Data

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

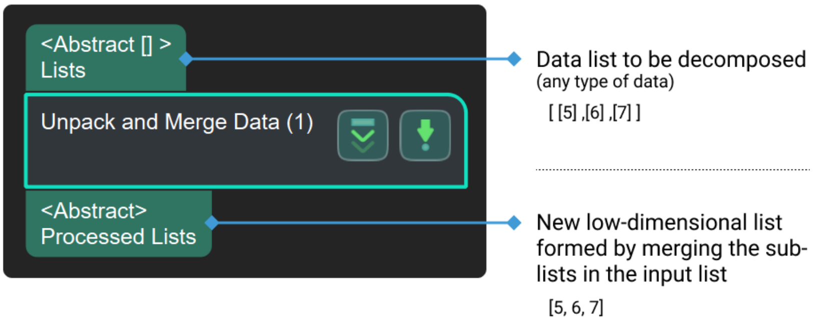
Decompose a high-dimensional data list and merge the sub-lists in it.



Usage Scenario

This Step is widely used for dimensionality reduction. It can decompose the data list, and merge the elements after decomposition. It is usually used to reduce dimensionality of high-dimensional data. For example, if the input list is \[[5], [6], [7]], the output can be [5, 6, 7].

Input and Output

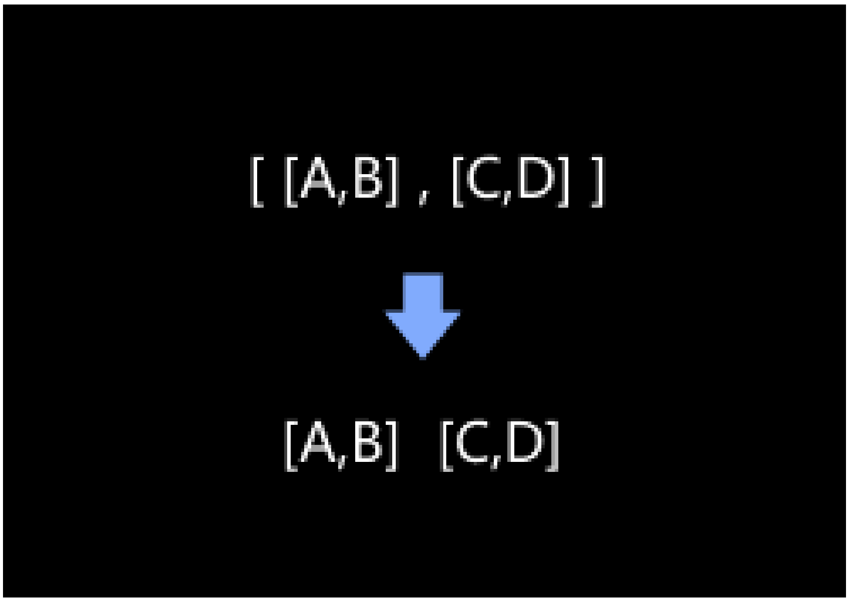

<Abstract [] >
Lists
Unpack and Merge Data (1)
<Abstract>
Processed Lists
Data list to be decomposed
(any type of data)
[[5] ,[6] ,[7]]
New low-dimensional list formed by merging the sub-lists in the input list
[5, 6, 7]

4.3.371. Unpack Data

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

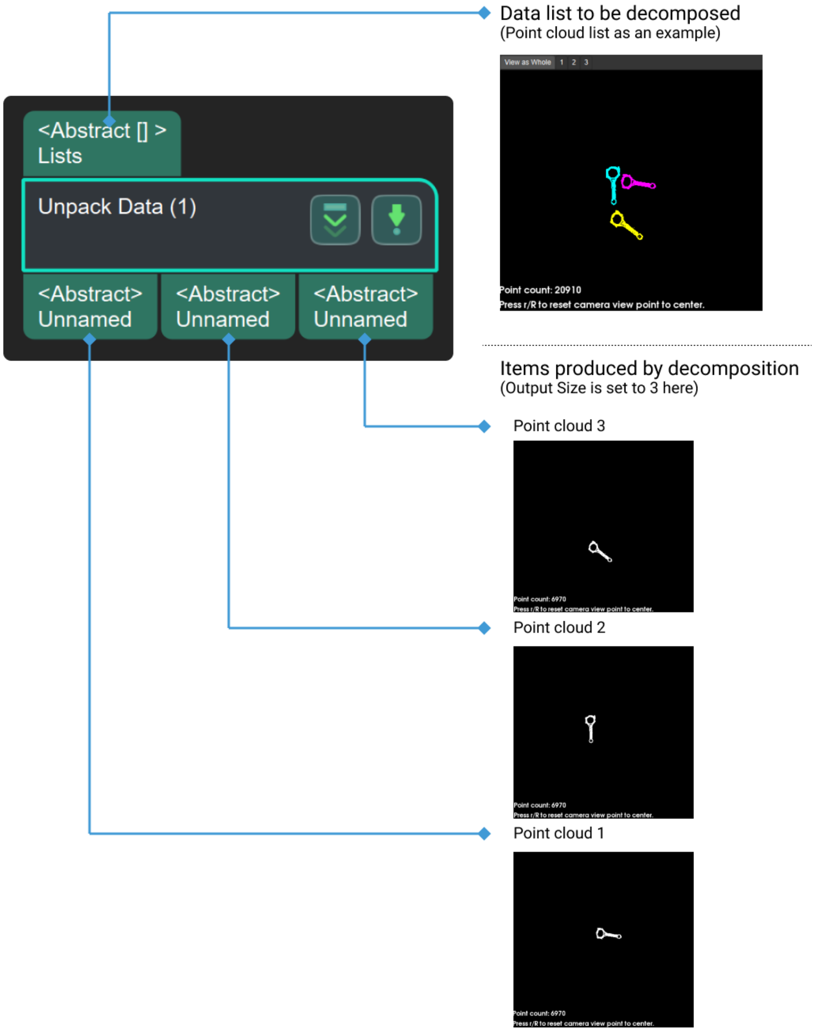
Decompose the list and obtain the first N elements in the list. The N value can be set in Step Parameters.



Usage Scenario

This Step is widely used for dimensionality reduction. It can decompose image list, point cloud list, dimensions list and other data lists. For example, if the input list is \[[5], [6], [7]], the output can be [5], [6], [7].

Input and Output


Data list to be decomposed
(Point cloud list as an example)
View as Whole 1 2 3
Point count: 20910
Press r/R to reset camera view point to center.
<Abstract [] > Lists
Unpack Data (1)
<Abstract> Unnamed
<Abstract> Unnamed
<Abstract> Unnamed
Items produced by decomposition
(Output Size is set to 3 here)
Point cloud 3
Point count: 6970
Press r/R to reset camera view point to center.
Point cloud 2
Point count: 6970
Press r/R to reset camera view point to center.
Point cloud 1
Point count: 6970
Press r/R to reset camera view point to center.

4.3.372. Validate 2D Poses by Mask

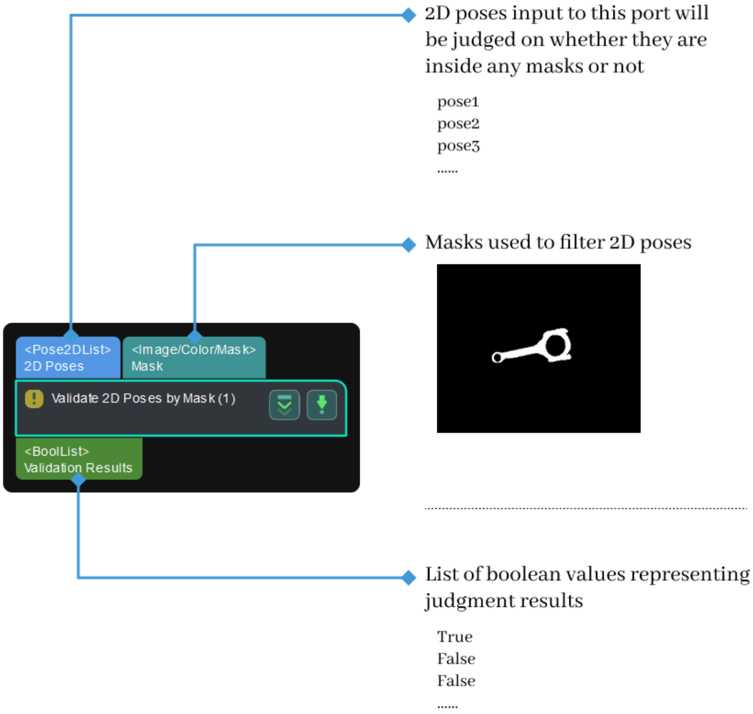
Function

Check that each 2D pose of input is within the input masks.

Usage Scenario

A custom Step for special scenarios.

Input and Output


2D poses input to this port will be judged on whether they are inside any masks or not
pose1
pose2
pose3
......
Masks used to filter 2D poses
<Pose2DList>
2D Poses
<Image/ColorMask>
Mask
Validate 2D Poses by Mask (1)
<BoolList>
Validation Results
List of boolean values representing judgment results
True
False
False
......

4.3.373. Validate Box Dimensions

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

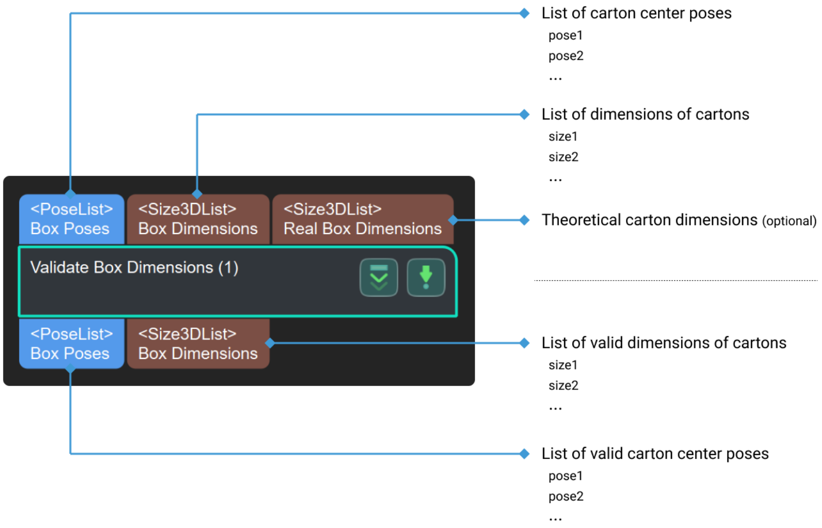
Function

Determine whether the carton dimensions corresponding to the pose list are valid based on the reference dimensions. Cartons with a significant dimensional error will be labeled as abnormal cartons.

Usage Scenario

This Step is usually used in carton depalletizing applications to determine whether recognized cartons are eligible. It is usually used after the Step Calc Poses and Dimensions from Planar Point Clouds.

Input and Output


List of carton center poses
pose1
pose2
...
List of dimensions of cartons
size1
size2
...
<PoseList> Box Poses
<Size3DList> Box Dimensions
<Size3DList> Real Box Dimensions
Theoretical carton dimensions (optional)
Validate Box Dimensions (1)
<PoseList> Box Poses
<Size3DList> Box Dimensions
List of valid dimensions of cartons
size1
size2
...
List of valid carton center poses
pose1
pose2
...

4.3.374. Validate Box Object Masks

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

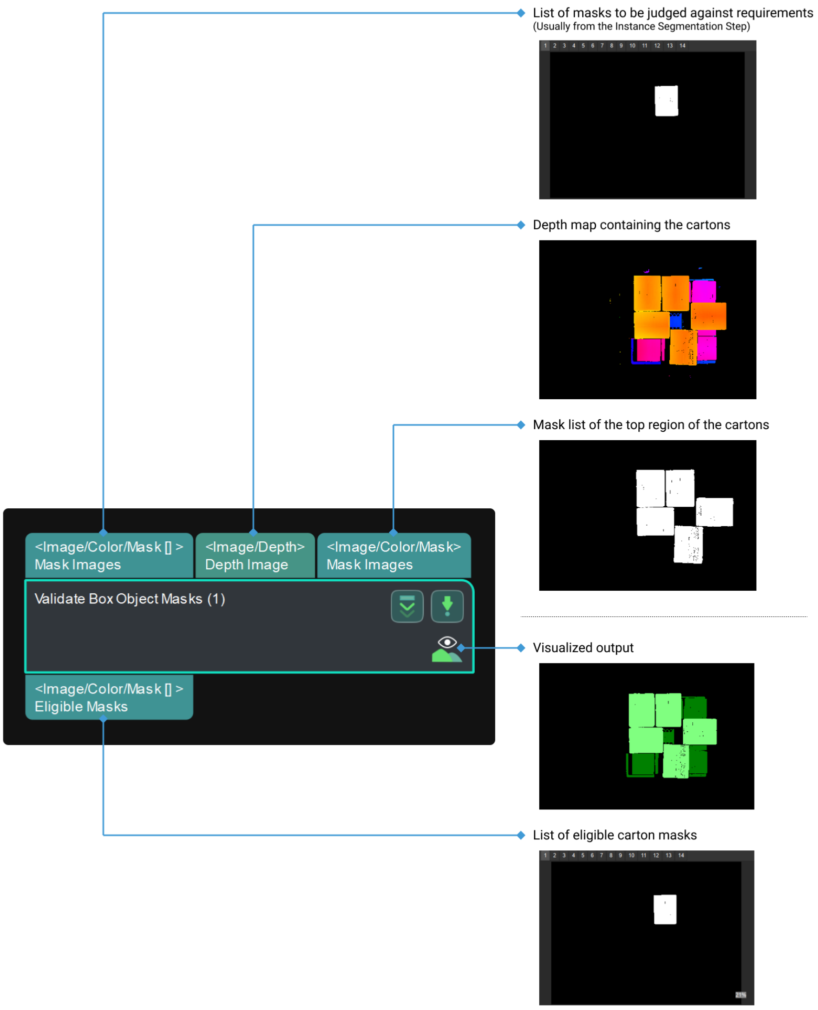
Function

Determine whether the carton masks meet the requirement, and output a list of eligible masks.

Usage Scenario

This Step is usually used in carton depalletizing applications to check whether the carton masks meet the requirement.

Input and Output



List of masks to be judged against requirements
(Usually from the Instance Segmentation Step)

Depth map containing the cartons

Mask list of the top region of the cartons

Visualized output

List of eligible carton masks

4.3.375. Validate Existence of Poses in 3D ROI

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

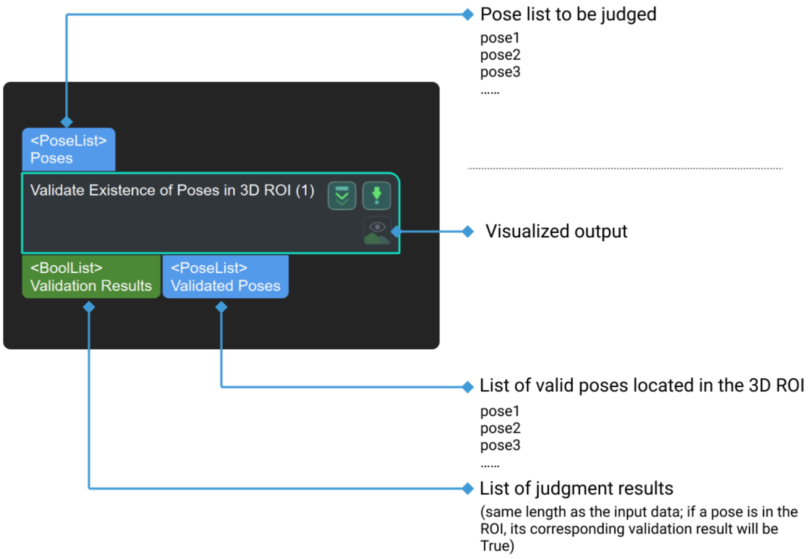
Function

Determine whether the input poses are in the set 3D ROI, and output the validation result and the poses located in the 3D ROI.

Usage Scenario

A general pose filtering Step, filtering by 3D ROI and outputting the validation results.

Input and Output

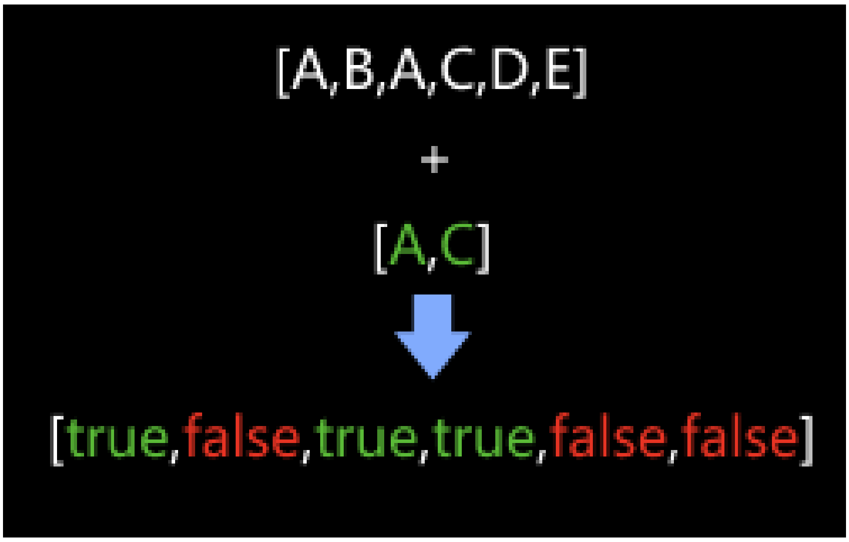

Pose list to be judged
pose1
pose2
pose3
......
<PoseList>
Poses
Validate Existence of Poses in 3D ROI (1)
Visualized output
<BoolList>
Validation Results
<PoseList>
Validated Poses
List of valid poses located in the 3D ROI
pose1
pose2
pose3
......
List of judgment results
(same length as the input data; if a pose is in the ROI, its corresponding validation result will be True)

4.3.376. Validate Labels and Output Flags

Function

Determine whether each label of the input label list appears in the standard list, and output the corresponding Boolean value list.

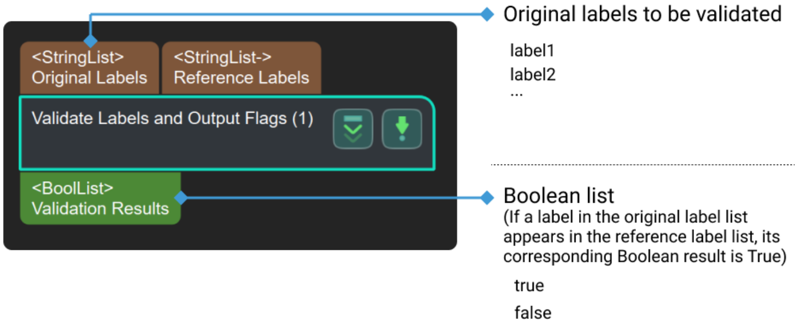
- If a label in the original label list appears in the reference label list, its corresponding Boolean result is "True."
- Otherwise, the result is "False."



Usage Scenario

This Step is usually followed by the Step Filter, and the Boolean list it outputs will be used as the criterion to filter the data array.

Input and Output


<StringList>
Original Labels
<StringList->
Reference Labels
Validate Labels and Output Flags (1)
<BoolList>
Validation Results
Original labels to be validated
label1
label2
...
Boolean list
(If a label in the original label list appears in the reference label list, its corresponding Boolean result is True)
true
false

Parameter Description

Label File Name

Description: This parameter is used to set the reference label file.

Default setting: labelFilter.json

Data type: strings (without special characters)

Data that does not meet requirements is invalid and cannot be input.

Instructions:

1. Click [**Set reference labels**] to open the JSON editor, as shown below.
2. Click ⊞, input a reference label name in the pop-up **Add Item** window, and then click [OK]. A reference label is added.

If a label includes decimals or special characters, this label cannot be added for conversion, and there will be a hint below the mapping panel suggesting that the user input is invalid.

1. If multiple reference labels are to be set, please repeat Step 2.
2. **Directory** refers to the directory of the current project. Click [...] on the right to change the directory of the project according to actual needs.
3. Name the file in **JSON File Name** and click [**Save**] to save the label in the file.
4. Click [OK] to finish the settings.

If the label file is already renamed beforehand, you can just click the [...] in **JSON File Name** and select the file to read the mapping relation.

4.3.377. Validate Masks Containing 2D Poses

Function

This Step checks whether each input mask contains any 2D pose from the input.

Usage Scenario

This Step is a custom Step for special scenarios.

Input and Output

- Input:
  1. Masks input to this port will be verified on whether they contain poses.
  2. Poses used to validate the masks.
- Output:
  1. A list of Boolean values representing the validation results. “True”: There is at least one pose in the mask. “False”: There is no pose in the mask.

4.3.378. Validate Point Clouds

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

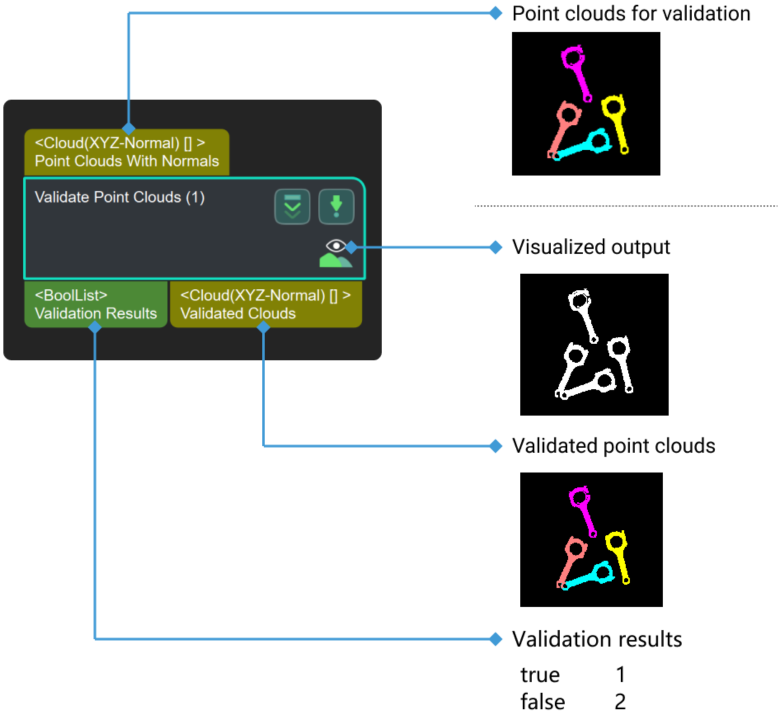
This Step uses the set rules to judge whether the input point clouds meet the requirements.

Usage Scenario

This Step is usually used to judge whether each of the input point clouds meets the requirements based on a threshold on the number of points (setting the parameter **Filter** to **CloudCapacityFilter**).

After validation, you can use the Filter Step to filter out point clouds that do not meet the requirements.

Input and Output



4.3.379. Validate Poses by Included Angles to Reference Direction

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

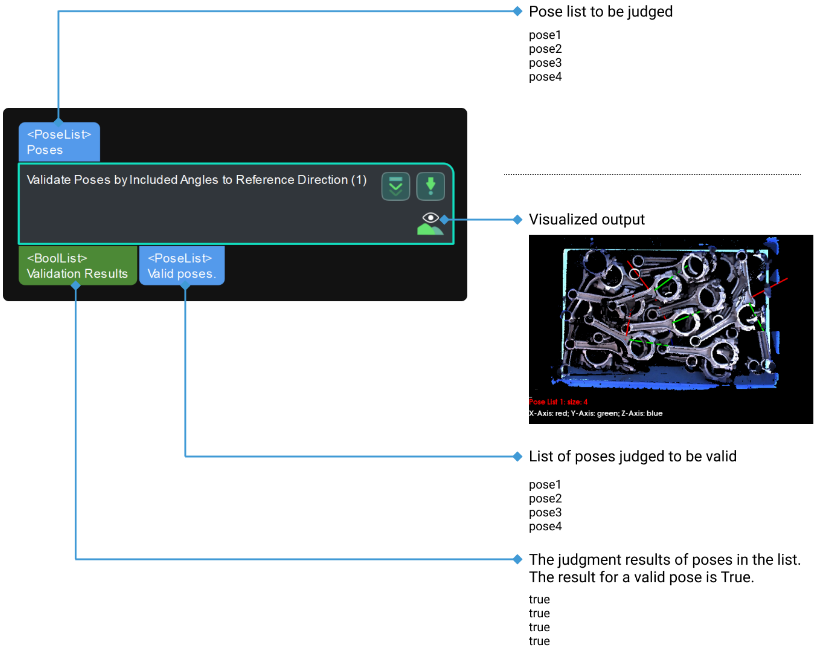
Function

Calculate the angles between the specified axes of the poses and the reference direction, the poses having such angles within the set maximum angle difference threshold will be retained and will have their output validation results set to True; those having such angles beyond the threshold will be removed and will have their results set to False.

Usage Scenario

A general pose filtering Step. No fixed usage scenarios.

Input and Output



<PoseList>
Poses
Validate Poses by Included Angles to Reference Direction (1)
<BoolList>
Validation Results
<PoseList>
Valid poses
Pose List 1: size: 4
X-Axis: red; Y-Axis: green; Z-Axis: blue

Pose list to be judged

pose1
pose2
pose3
pose4

Visualized output

List of poses judged to be valid

pose1
pose2
pose3
pose4

The judgment results of poses in the list.
The result for a valid pose is True.

true
true
true
true

4.3.380. Validate Underlying Poses

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step determines whether the input poses are located below other poses. The input poses will be divided into different groups, and only if the current group of poses is below the previous group of poses, the current group of poses will be used to check whether the corresponding objects are occluded.

Usage Scenario

This Step is a custom Step for special scenarios.

Input and Output

- Input:
  1. Poses input by this port will be judged on whether the corresponding objects are occluded (judgment between groups, not within groups).
  2. The radius value used to determine whether objects are occluded.
- Output:
  1. List of Boolean values representing the result of the judgment.

4.3.381. Extract Regions of Large Normal Deviations

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

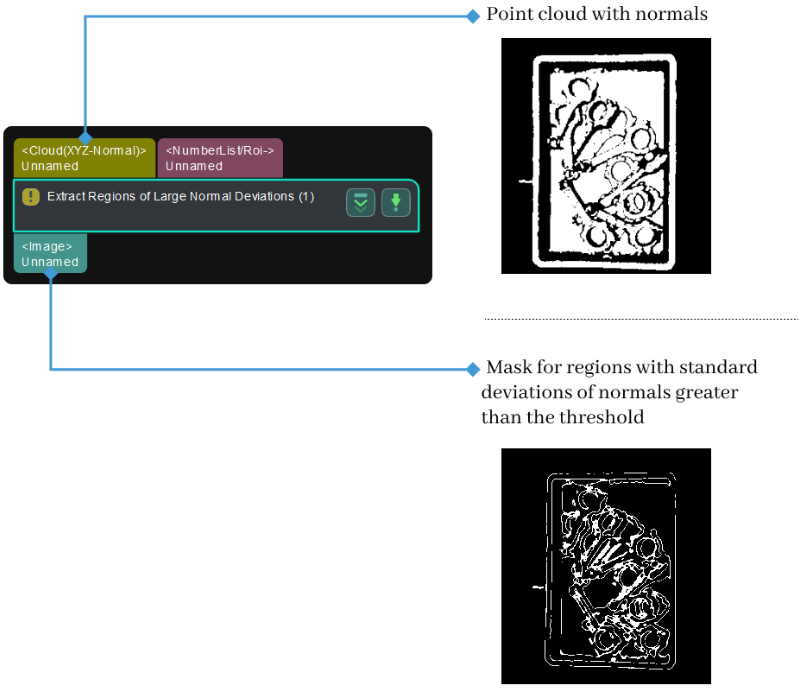
Function

This Step extracts regions where the normals have large standard deviations. It maps the normals onto a 2D image first, and then uses a 2D filter to calculate the local standard deviations and generate a mask of regions with standard deviations larger than the set threshold.

Usage Scenario

A custom Step for special scenarios.

Input and Output


<Cloud(XYZ-Normal)>
Unnamed
<NumberList/Roi->
Unnamed
Extract Regions of Large Normal Deviations (1)
<Image>
Unnamed
Point cloud with normals
Mask for regions with standard deviations of normals greater than the threshold

4.3.382. Vector3D Arithmetic Operations

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step performs simple operations on a list of 3D vectors of the Vector3D data type with another list of 3D vectors (or numbers).

Usage Scenario

This Step is a general 3D vector calculation Step. No fixed usage scenarios.

Input and Output

- Input:
  1. First list of 3D vectors for the operation
  2. Second list of 3D vectors (or numbers) for the operation
- Output:
  1. 3D vectors obtained from operation

4.3.383. Verify Pick Points

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step calculates the deviation between the input pick points and the pick points in the file.

Usage Scenario

This Step is a custom Step for special scenarios.

Input and Output

- Input:
  1. The deviation between the pick points input at this port and the pick points in the file specified in the parameter will be calculated.
- Output:
  1. Calculated deviations

4.3.384. Visualize Depth in Range

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step converts the depth map in the ROI to a color image, with different depth values corresponding to different color pixel values.

Usage Scenario

This Step is usually used in measurement scenarios to convert depth maps into color images for subsequent measurement or display.

Input and Output

- **Input**: Depth map input to this port will be converted to a color image.
- **Output**: color image obtained from conversion.

4.3.385. Visualize Information on Image

Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step draws masks, poses, and labels on a color image.

Usage Scenario

This Step is generally used for subsequent display.

Input and Output

- Input:
  1. Information input from other ports will be visualized on the image input to this port.
  2. Mask list input by this port will be filtered.
  3. Poses input to this port will be visualized.
  4. Classification labels input to this port will be visualized.
- Output:
  1. Result of visualizing the information input by each port on image.

4.3.386. Large Workpiece Measurement

Large workpieces refer to parts that are usually used in automobile, aircraft, ship, and other manufacturing industries, with lengths longer than 0.5 m. In large workpiece measurement scenarios, inspections typically focus on circular holes, oblong holes, planes, etc., to determine the qualification of the parts. For large workpiece measurement scenarios, Mech-Vision provides corresponding Procedures that can facilitate the creation and deployment of the project.

Large workpiece measurement has a high requirement on the precision and a high-precision camera is needed. You can use the UHP series cameras provided by Mech-Mind.

This chapter introduces the project creation framework and related Procedures of large workpiece measurement.

Check the content below to learn about **how to create a large workpiece measurement project**.

Large Workpiece Measurement Project Creation

Check the contents below to learn about the **Procedures** related to large workpiece measurement.

- Select Point Cloud from Depth Map
- Extract Planar Point Clouds
- Calc Hole Center Poses and Diameters
- Calc Oblong Hole Center Poses and Axis Lengths
- Transform Poses to Custom Frame

The next section will introduce how to create a large workpiece measurement project.

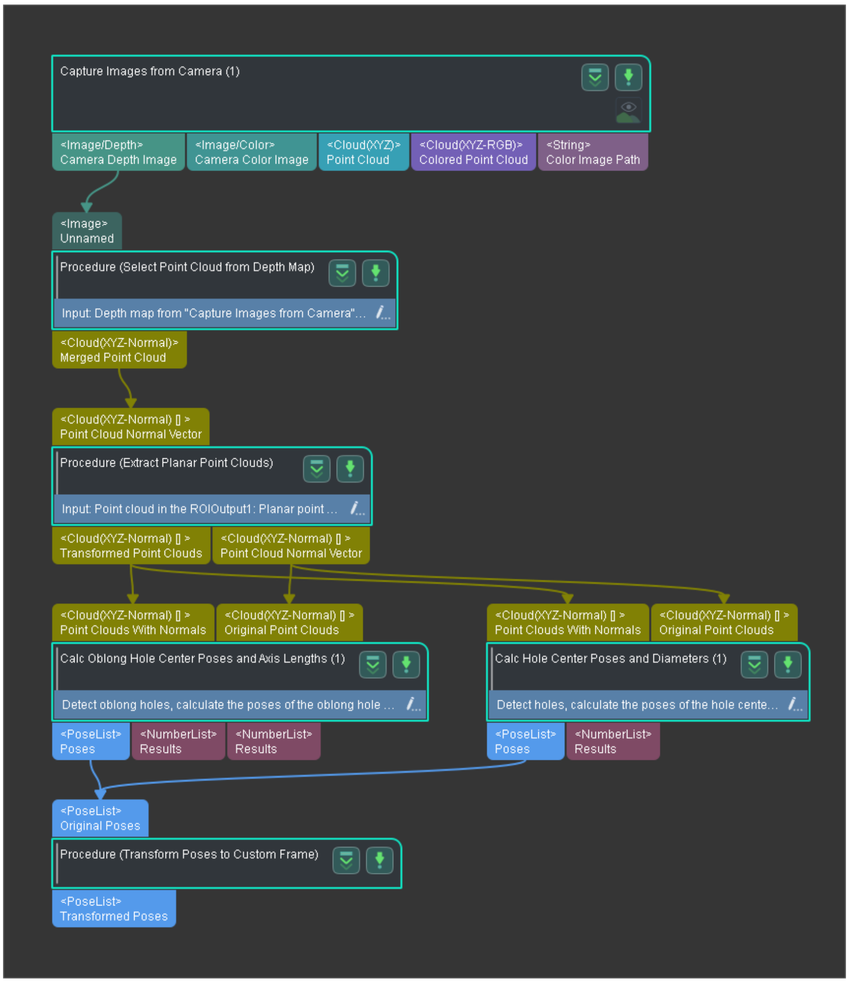
4.3.386.1. Large Workpiece Measurement Project Creation

This section introduces the project creation framework, processing principles, and notes of creating a LARGE WORKPIECE measurement project.

Project Creation Framework

The following Steps and Procedures can be used in sequence to create a large workpiece measurement project:

1. Capture Images from Camera
2. Select Point Cloud from Depth Map
3. Extract Planar Point Clouds
4. Calc Hole Center Poses and Diameters or Calc Oblong Hole Center Poses and Axis Lengths
5. Transform Poses to Custom Frame


Capture Images from Camera (1)
<Image/Depth> Camera Depth Image
<Image/Color> Camera Color Image
<Cloud(XYZ)> Point Cloud
<Cloud(XYZ-RGB)> Colored Point Cloud
<String> Color Image Path
<Image> Unnamed
Procedure (Select Point Cloud from Depth Map)
Input: Depth map from "Capture Images from Camera"...
<Cloud(XYZ-Normal)> Merged Point Cloud
<Cloud(XYZ-Normal) [] > Point Cloud Normal Vector
Procedure (Extract Planar Point Clouds)
Input: Point cloud in the ROIOutput1 : Planar point ...
<Cloud(XYZ-Normal) [] > Transformed Point Clouds
<Cloud(XYZ-Normal) [] > Point Cloud Normal Vector
<Cloud(XYZ-Normal) [] > Point Clouds With Normals
<Cloud(XYZ-Normal) [] > Original Point Clouds
<Cloud(XYZ-Normal) [] > Point Clouds With Normals
<Cloud(XYZ-Normal) [] > Original Point Clouds
Calc Oblong Hole Center Poses and Axis Lengths (1)
Detect oblong holes, calculate the poses of the oblong hole ...
Calc Hole Center Poses and Diameters (1)
Detect holes, calculate the poses of the hole cente...
<PoseList> Poses
<NumberList> Results
<NumberList> Results
<PoseList> Poses
<NumberList> Results
<PoseList> Original Poses
Procedure (Transform Poses to Custom Frame)
<PoseList> Transformed Poses

You should select the Procedure according to the feature of the workpiece. For example, Calc Hole Center Poses and Diameters should be used to measure holes, while Calc Oblong Hole Center Poses and Axis Lengths should be used to measure oblong holes. An incorrectly used Procedure will lead to incorrect measurement results.

Processing Principles

The large workpiece measurement project converts the workpiece poses from the camera reference frame to the **BASE REFERENCE FRAME** (The reference frame of the object where the workpiece is installed. For example, if the large workpiece is an automobile part, the reference frame of the car body is the base reference frame). The data workflow of the project is as follows:

1. Capture images with a high-precision camera, and then Mech-Vision will process the images and output the poses of the hole or oblong hole on the workpiece in the camera reference frame.
2. Create a workpiece reference frame based on the benchmark in the drawing of the workpiece provided by the customer, and then convert the poses from the camera reference frame to the workpiece reference frame.
3. Convert the poses from the workpiece reference frame to the **BASE REFERENCE FRAME**.

   The processing in Mech-Vision is completed after the poses in the base reference frame are output.

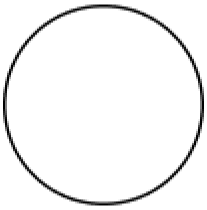
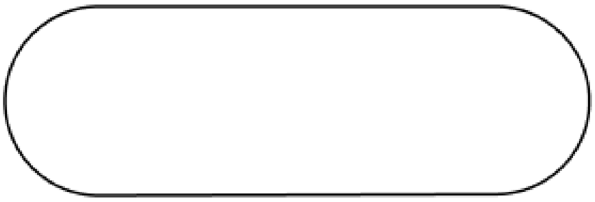
4. Calculate the deviation of the result output by Mech-Vision from the values in the drawings.

Notes

Large workpieces
: Large workpieces refer to parts that are usually used in automobile, aircraft, ship, and other manufacturing industries, with lengths longer than 0.5 m.

Base reference frame
: The reference frame of the object where the workpiece is installed. For example, if the large workpiece is an automobile part, the reference frame of the car body is the base reference frame.

Hole and oblong hole
: The illustrations are shown below:

Hole	Oblong Hole
	

Since each large workpiece may include multiple holes/oblong holes, you can add projects or parameter recipes according to the sequence or feature types of the measurement.

4.3.386.2. Calc Hole Center Poses and Diameters

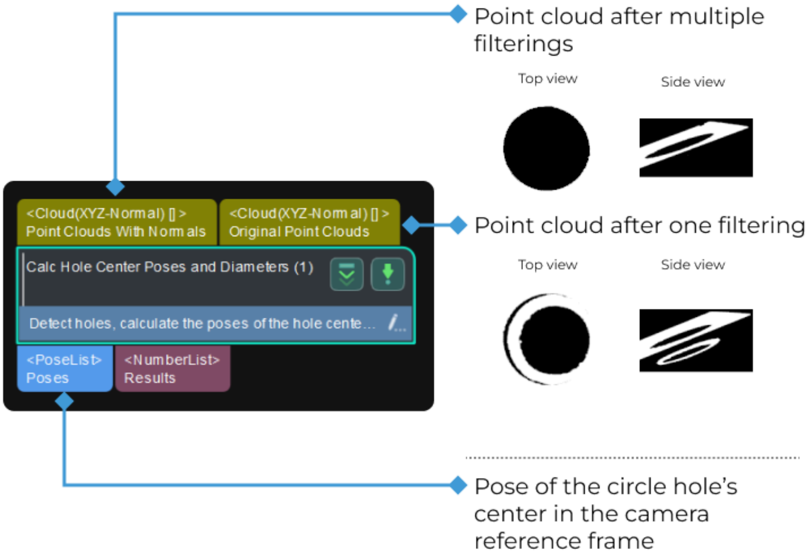
Function

This Procedure is used to detect the circular hole and calculate the diameter of the circular hole and the pose of the circular hole's center in the camera reference frame.

Usage Scenario

This Procedure is usually used in measurement scenarios for **LARGE WORKPIECES**. Usually, the Extract Planar Point Clouds Procedure is used before this Procedure to obtain the planar point cloud, and the Transform Poses to Custom Frame Procedure follows this Procedure to obtain the pose in the BASE REFERENCE FRAME.

Input and Output


Point cloud after multiple filterings
Top view
Side view
<Cloud(XYZ-Normal) []>
Point Clouds With Normals
<Cloud(XYZ-Normal) []>
Original Point Clouds
Calc Hole Center Poses and Diameters (1)
Detect holes, calculate the poses of the hole cente...
<PoseList>
Poses
<NumberList>
Results
Point cloud after one filtering
Top view
Side view
Pose of the circle hole's center in the camera reference frame

Parameter Tuning

Detect and Measure Circle

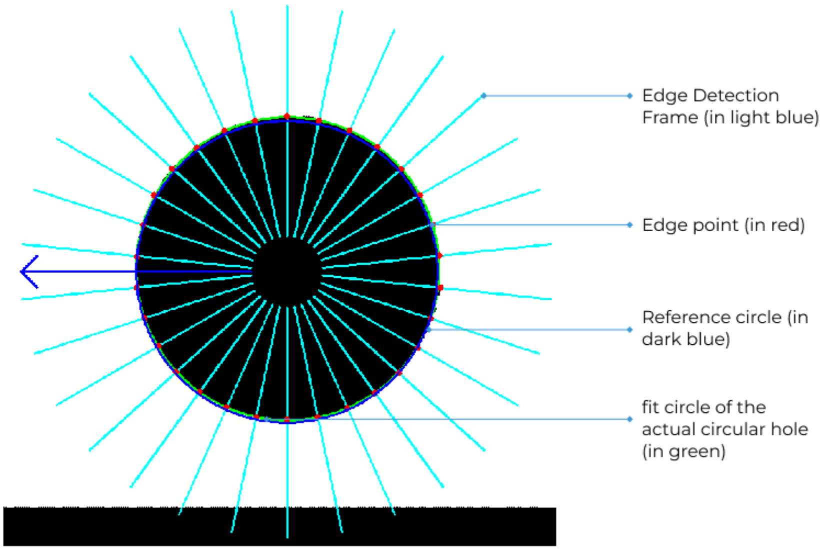
ROI

Parameter description: This parameter is used to set an ROI and read the center pose and size information of the circular hole.

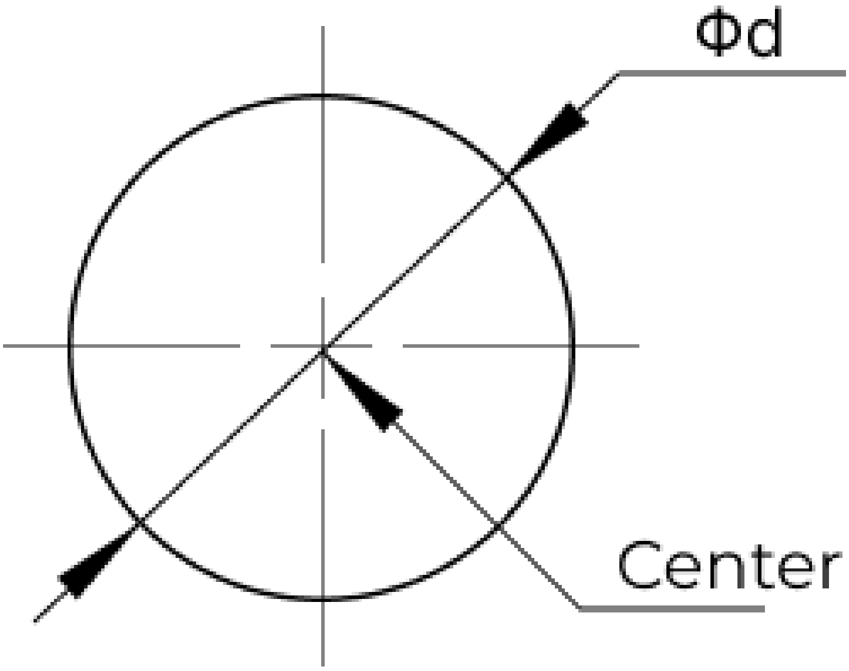
Tuning recommendation: Please select an ROI in the **Measurement Mode** and set an accurate ROI.

When selecting an ROI, please make the selection frame as close to the circle as possible. An inaccurate ROI will affect the accuracy of the measured circle.

Principle: Once an ROI is determined with the selection frame, a reference circle with the center of the frame as the center and the short side of the frame as its diameter will be generated, and edge detection frames will be generated along the direction of the reference circle's radius. Since the width of the edge detection frames are very small, they look like line segments, as shown below. In the edge detection frames, if the neighboring pixels change from black to white along the direction from the center outward, an edge point will be detected, and therefore a fit circle of the actual circular hole can be found with the detected edge points. The more the edge detection frames, the more the edge points and a more accurate circle can be calculated. The longer the length of the edge detection frame, the larger the detection area, and therefore even if the circle's position changes in the image, it can still be accurately calculated.


Edge Detection Frame (in light blue)
Edge point (in red)
Reference circle (in dark blue)
fit circle of the actual circular hole (in green)

- Center X/Y: the coordinates of the circle’s center.
- Width/Height: the actual size of the circle (diameter d in the figure below).
- Angle: Set the rotation angle of the ROI selection frame.



Number of Edge Detection Frames

Parameter description: This parameter specifies the number of edge detection frames of the measured circle. The larger the value, the more accurate the measured circle will be and the longer calculation time required.

Default value: 500

Tuning recommendation: If the circle is regular in shape, you can reduce the default value; if the circle is not regular, you can increase the default value.

Edge Detection Frame Half Height

Parameter description: This parameter specifies the length of the edge detection frame. The longer the length of the edge detection frame, the larger the detection area, and the longer calculation time required. The larger the parameter value, the longer the computation time. The smaller the parameter value, the shorter the computation time. An overly large value may lead to an incorrect fit circle, because the single edge detection frame may have two intersections with the circle and unwanted edge points may be detected.

Default value: 110

Tuning recommendation: If the ROI can exactly enclose the entire circle (not applicable to semicircles), it is recommended to set this value to the radius of the circle, i.e., half of the width/height in ROI. If the ROI cannot enclose the entire circle or there are other edges in the image, please set the value according to the actual situation.

Extract 3D Points in Cylinder

> When this Procedure is executed, the point cloud will be converted from the camera reference frame to the point’s reference frame. The following parameters are used to set the range of the Z coordinates in the point’s reference frame.

Min Z Value

Parameter description: This parameter specifies the minimum value (unit: mm) of the Z coordinate in the point’s reference frame. The smaller the value, more points will be extracted.

Default value: −0.500 mm

Max Z Value

Parameter description: This parameter specifies the maximum value (unit: mm) of the Z coordinate in the point’s reference frame. The larger the value, more points will be extracted.

Default value: 0.500 mm

4.3.386.3. Calc Oblong Hole Center Poses and Axis Lengths

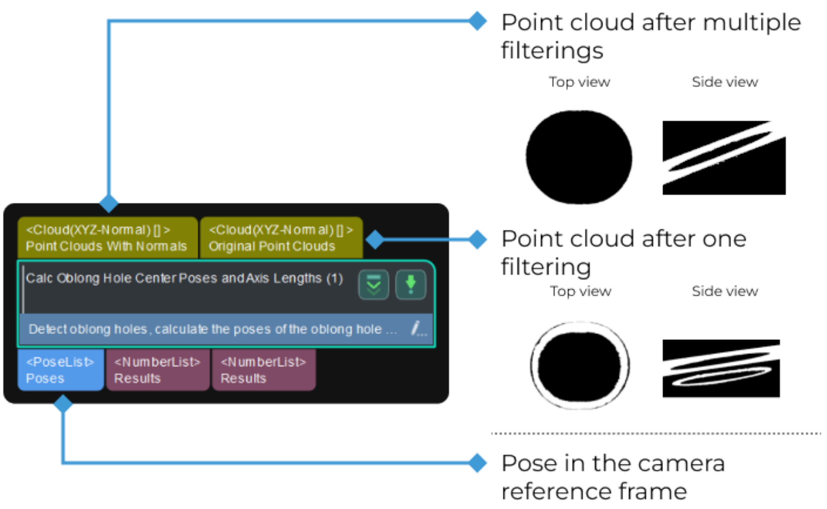
Function

This Procedure calculates the longitudinal axis and radii of the oblong hole and outputs the pose of the oblong hole’s center in the camera reference frame.

Usage Scenario

This Procedure is usually used in measurement scenarios for LARGE WORKPIECES. Usually, the Extract Planar Point Clouds Procedure is used before this Procedure to obtain the planar point cloud, and the Transform Poses to Custom Frame Procedure follows this Procedure to obtain the pose in the BASE REFERENCE FRAME.

Input and Output


Point cloud after multiple filterings
Top view
Side view
<Cloud(XYZ-Normal)[]> Point Clouds With Normals
<Cloud(XYZ-Normal)[]> Original Point Clouds
Calc Oblong Hole Center Poses and Axis Lengths (1)
Detect oblong holes, calculate the poses of the oblong hole ...
<PoseList> Poses
<NumberList> Results
<NumberList> Results
Point cloud after one filtering
Top view
Side view
Pose in the camera reference frame

Parameter Tuning

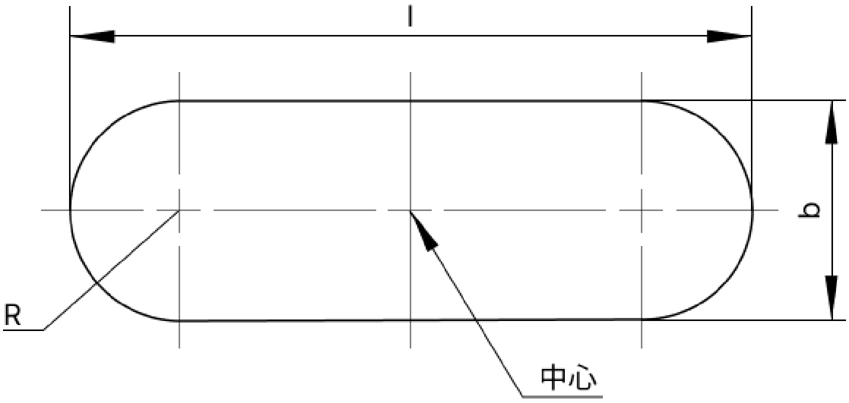
Detect and Measure Oblong Hole

ROI

Parameter description: This parameter is used to set an ROI and read the center pose and size information of the oblong hole. Tuning recommendation: Please select an ROI in the **Measurement Mode** and set an accurate ROI.

> When selecting an ROI, please make the selection frame as close to the oblong hole as possible. An inaccurate ROI will affect the accuracy of the measured oblong hole.

- Center X/Y: the coordinates of the oblong hole’s center.
- Width: the width of the oblong hole (l in the figure below).
- Height: the height of the oblong hole (b in the figure below).
- Angle: Set the rotation angle of the ROI selection frame. When the oblong hole is in a horizontal or vertical direction, you do not need to set this parameter.



Extract 3D Points in Cylinder

> When this Procedure is executed, the point cloud will be converted from the camera reference frame to the point’s reference frame. The following parameters are used to set the range of the Z coordinates in the point’s reference frame.

Min Z Value

Parameter description: This parameter specifies the minimum value (unit: mm) of the Z coordinate in the point's reference frame. The smaller the value, more points will be extracted.
Default value: −0.500 mm

Max Z Value

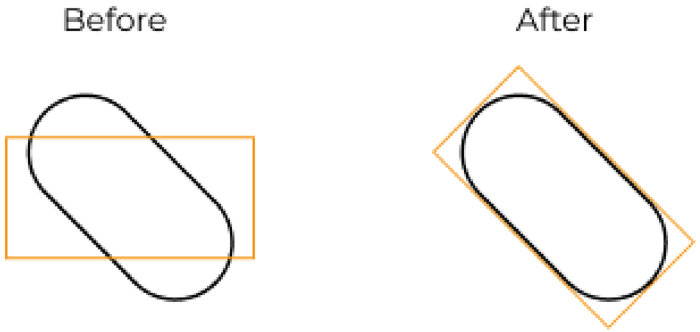
Parameter description: This parameter specifies the maximum value (unit: mm) of the Z coordinate in the point's reference frame. The larger the value, more points will be extracted.
Default value: 0.500 mm

Tuning Example

Angle

This parameter is used to set the rotation angle of the ROI selection frame. In the actual large workpiece measurement scenarios, the oblong holes may be in varied directions, and the ROI selection frame can only be in the horizontal or vertical direction. You will need to set the **Angle** if the oblong hole is not in the horizontal or vertical direction to make the ROI selection frame as close to the oblong hole as possible.

The figure below shows the effect before and after the **Angle** is adjusted. The orange frame represents the ROI selection frame.



4.3.386.4. Extract Planar Point Clouds

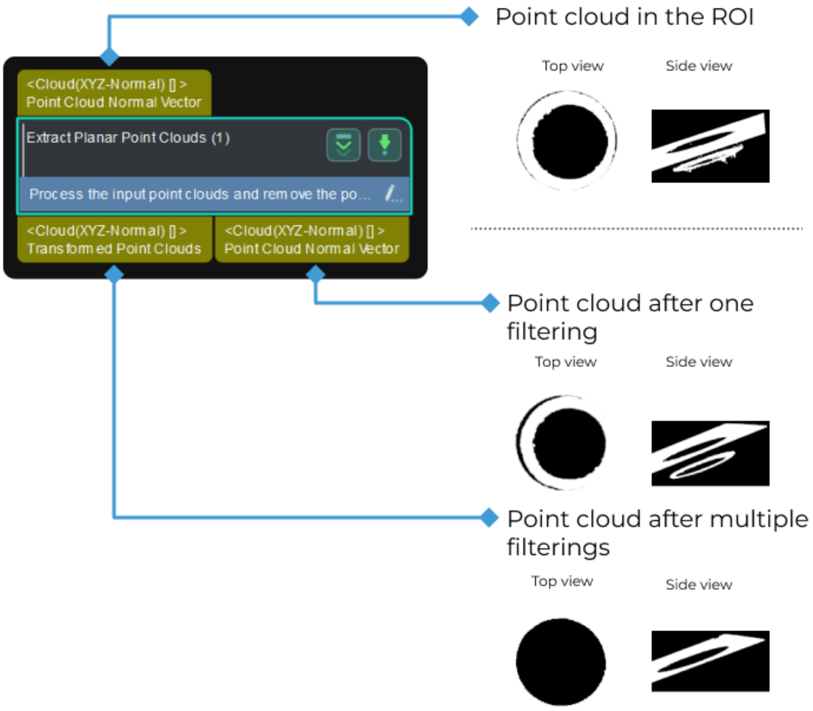
Function

This Procedure can be used to remove points that deviate from the plane of the point cloud and obtain a relatively flat planar point cloud.

Usage Scenario

This Procedure is usually used in measurement scenarios for LARGE WORKPIECES. Usually, the Select Point Cloud from Depth Map Procedure is used before this Procedure to obtain the point cloud. The Calc Hole Center Poses and Diameters and Calc Oblong Hole Center Poses and Axis Lengths Procedures follow this Procedure to measure the circular holes and oblong holes, respectively.

Input and Output


Point cloud in the ROI
Top view
Side view
<Cloud(XYZ-Normal) [] >
Point Cloud Normal Vector
Extract Planar Point Clouds (1)
Process the input point clouds and remove the po...
<Cloud(XYZ-Normal) [] >
Transformed Point Clouds
<Cloud(XYZ-Normal) [] >
Point Cloud Normal Vector
Point cloud after one filtering
Top view
Side view
Point cloud after multiple filterings
Top view
Side view

Parameter Tuning

Point Filter / Point Filter (Coarse Filtering 2) / Point Filter (Fine Filtering 2)

Mean K

Parameter description: This parameter is used to remove points that are near to the planar point cloud. Points that are nearest to the planar point cloud will be removed first.

Default value: 30

Standard Deviation Multiplier

Parameter description: This parameter is used to calculate the threshold of distance. The smaller the value, the more points will be removed.

Default value: 2.0

Point Filter (Coarse Filtering) / Point Filter (Fine Filtering)

Max Angle

Parameter description: This parameter specifies the maximum angle (°) between the Z-axis of the reference point and the normal of the measurement point. Points with the angle between the reference direction and the point normal greater than the maximum angle will be discarded. The larger the value, the more points will be kept.

Default value: 5°

4.3.386.5. Select Point Cloud from Depth Map

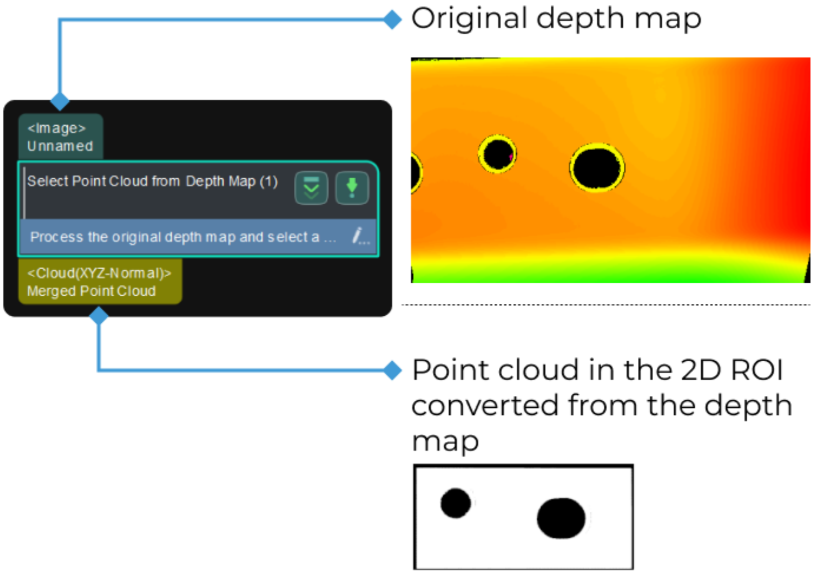
Function

This Procedure selects a 2D ROI on the original depth map, and then converts the depth map in the 2D ROI to the point cloud to facilitate further calculation.

Usage Scenario

This Procedure is usually used in the measurement scenario for LARGE WORKPIECES. Usually, the Capture Images from Camera Step is used before this Procedure to obtain the depth map, and the Extract Planar Point Clouds Procedure follows this Procedure to convert the point cloud to planar point cloud.

Input and Output


Original depth map
<Image>
Unnamed
Select Point Cloud from Depth Map (1)
Process the original depth map and select a ...
<Cloud(XYZ-Normal)>
Merged Point Cloud
Point cloud in the 2D ROI converted from the depth map

Parameter Description

Segment Depth Map

The following parameters are used to segment the depth map to obtain the desired point cloud.

Max Height Diff between Neighbouring Pixels

Parameter description: In the process of segmenting the depth map, this parameter specifies the maximum height difference between neighboring pixels. If the height difference between two pixels is smaller than this value, the two pixels belong to the same area. The larger the value, the more the segmented area.

Default value: 0.500 mm

Tuning recommendation: If you want to segment a depth map of a flat surface, you can adjust the value to a larger value. However, an overly large value may lead to overly small segmented areas and no valid point cloud will be output. If you want to segment the depth map of a curved surface or threaded holes, you can adjust the value to a smaller value to avoid a loss of desired point clouds. If the point clouds are closely adjacent to each other, adjusting the value may not be able to remove the unwanted point cloud.

Min Segment Area

Parameter description: This parameter specifies the minimum area of the segmented depth map. Segmented depth map with pixels less than this value will be removed. The smaller the value, the more the segmented area; the larger the value, the larger the removed area.

Default value: 1500 px

Depth ROI File Name

Parameter description: This parameter is used to select an ROI on the depth map.

Tuning recommendation: Click the **[Set 2D ROI]** button to select an accurate 2D ROI.

Get Highest Layer Regions in Depth Map

Layer Height

Parameter description: After the depth map is converted to the point cloud, this parameter specifies the distance (unit: mm) between the camera and the planar point cloud along the Z-axis in the camera reference frame. The larger the value, the more points will be kept.

Default value: 10.000 mm

Tuning recommendation: When the point cloud of the object to be measured has multiple layers, you can adjust this value to obtain the desired point cloud.

4.3.386.6. Transform Poses to Custom Frame

Function

This Procedure converts the object center pose to the BASE REFERENCE FRAME for further calculation. The process of the conversion is as follows:

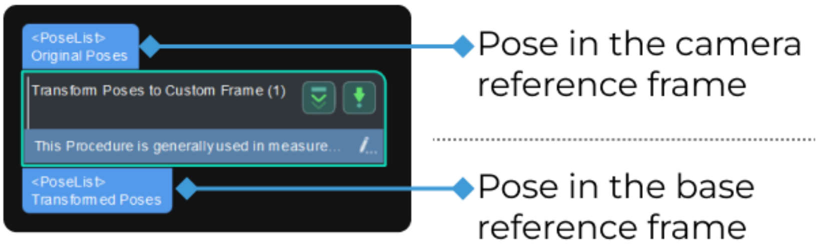
1. Convert the pose from the camera reference frame to a customized reference frame.
2. Convert the pose from the customized reference frame to the base reference frame.

> When this Procedure is used in large workpiece measurement scenarios, the customized reference frame is usually the workpiece reference frame.

Usage Scenario

This Procedure is usually used in the measurement scenario for LARGE WORKPIECES. Usually, the Calc Hole Center Poses and Diameters or Calc Oblong Hole Center Poses and Axis Lengths Procedures are used before this Procedure to output the center pose of the circular hole or oblong hole in the camera reference frame.

Input and Output


<PoseList>
Original Poses
Transform Poses to Custom Frame (1)
This Procedure is generally used in measure...
<PoseList>
Transformed Poses

Pose in the camera reference frame

Pose in the base reference frame

Parameter Tuning

Read Poses from File

Pose File Path

Click 🗀 to select **JSON** file of the first customized reference frame.

> When this Procedure is used in large workpiece measurement scenarios, the customized reference frame used here is usually the workpiece reference frame. Please contact Mech-Mind Technical Support to obtain the file.

Read Poses from File

Pose File Path

Click 🗀 to select **JSON** file of the second customized reference frame.

> When this Procedure is used in large workpiece measurement scenarios, the customized reference frame used here is usually the base reference frame. Please contact Mech-Mind Technical Support to obtain the file.

4.3.387. Sort

This section introduces Procedures related to sorting.

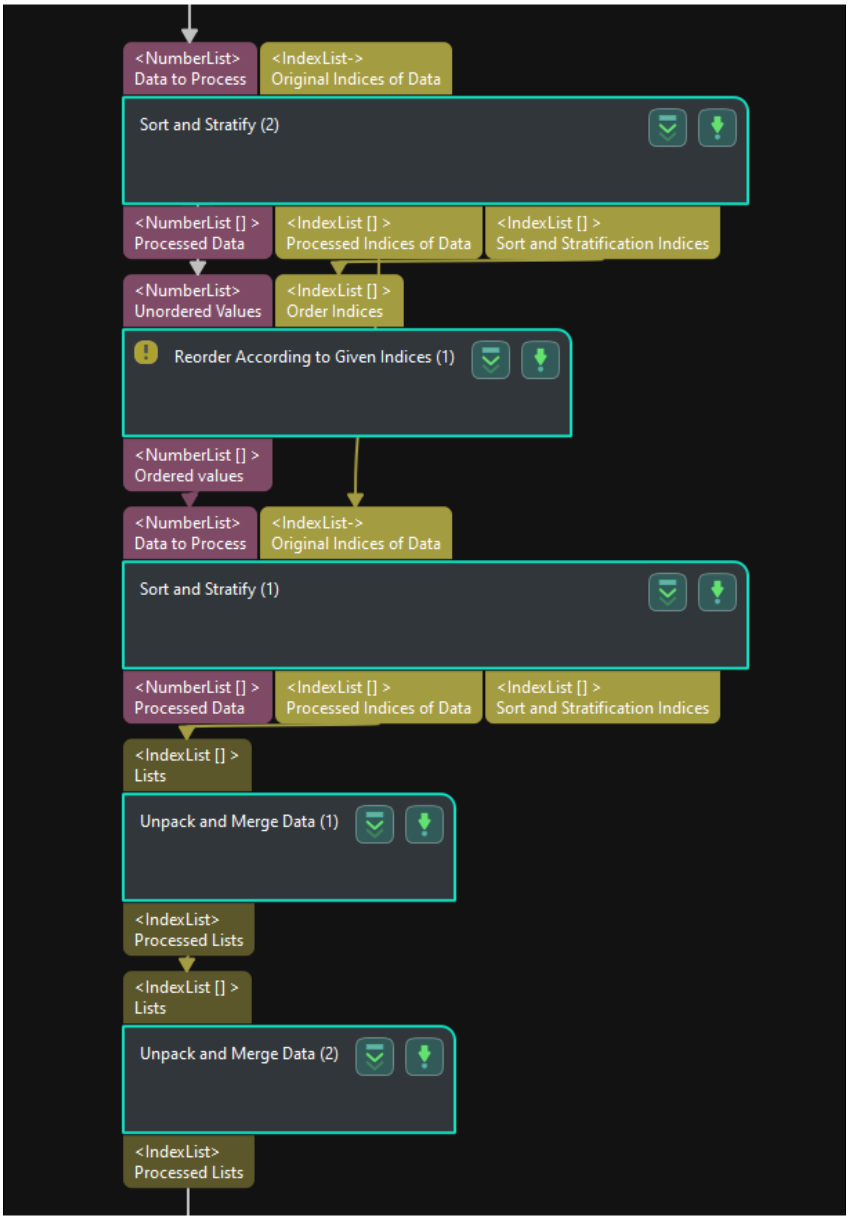
Sort by Two Values

Sort by Three Values

Sort Pick Points

4.3.387.1. Sort by Two Values

Input two elements with the same length, then Mech-Vision would sort them twice with different standards and finally output the indexes after sorting. The combination is as follows:

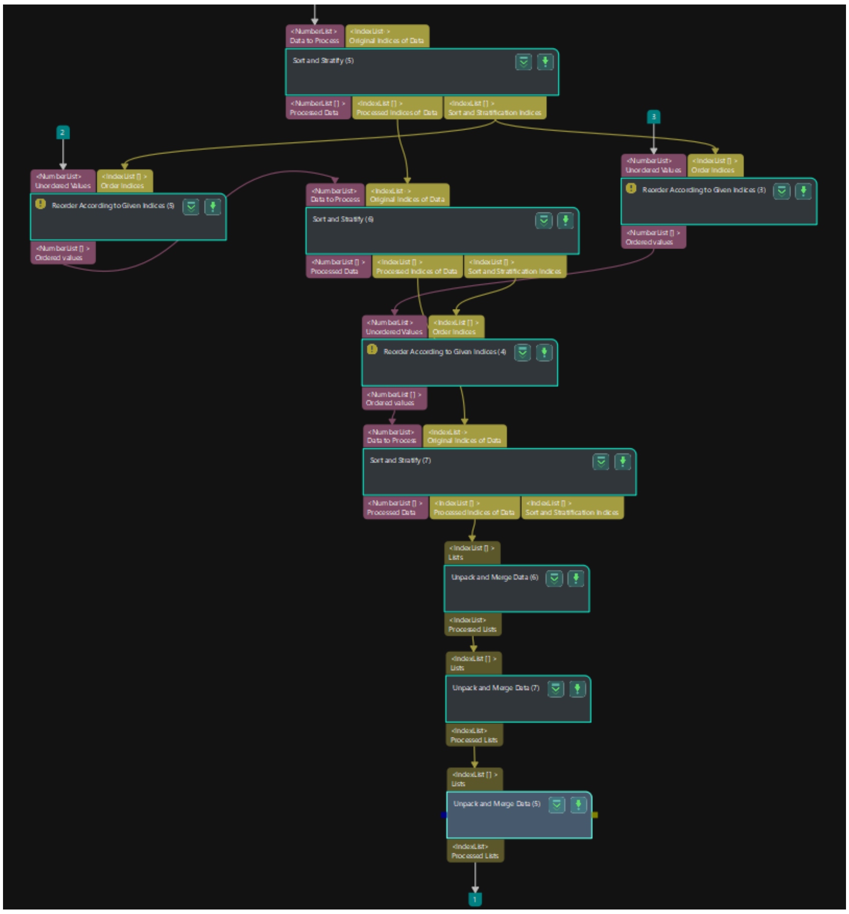


The introduction of each Steps is as follows:

1. Sort and Stratify Sort and classify the first group of elements with the specified layer spacing (Standard 1) and the output the index.
2. Reorder According to Given Indices Using the index of the first sorting, Mech-Vision would sort the second group of elements and then output values after sorting.
3. Sort and Stratify Based on the sorted second group and the index of the first sorting, this Step would sort the second group again with the specified layer spacing (Standard 2) and output the index. With this index, the index of the first sorting is sorted again and then the indexes after sorting twice will be output.
4. Unpack and Merge Data This Step turns three-dimensional array-like variables into two-dimensional variables based on the indexes after sorting twice.
5. Unpack and Merge Data After last Step “Unpack And Merge Data”, this Step could turn the two-dimensional array-like variables into a one-dimensional array as the final output.

4.3.387.2. Sort by Three Values

Input two elements with the same length, then Mech-Vision would sort them three times with different standards and finally output the indexes after sorting. The combination is as follows:


Sort and Stratify (5)
Reorder According to Given Indices (5)
Sort and Stratify (6)
Reorder According to Given Indices (3)
Reorder According to Given Indices (4)
Sort and Stratify (7)
Unpack and Merge Data (6)
Unpack and Merge Data (7)
Unpack and Merge Data (5)

The functions of each Steps are described below:

1. Sort and Stratify Sort and classify the first group of elements with the specified layer spacing (Standard 1) and the output the index.
2. Reorder According to Given Indices Using the index of the first sorting, Mech-Vision would sort the second group of elements and then output values of the second group after sorting.
3. Reorder According to Given Indices Using the index of the first sorting, Mech-Vision would sort the third group of elements and then output values of the second group after sorting.
4. Sort and Stratify Based on the sorted second group (Input 1) and the index of the first sorting (Input 2), this step would sort and classify Input 1 again with the specified layer spacing (Standard 2). This step outputs the index after sorting twice (Output 2) and the index of the second sorting (Output 3).
5. Reorder According to Given Indices Using the index of the second sorting, Mech-Vision would sort the second group of elements again and then output values of the third group after sorting twice.
6. Sort and Stratify Based on the third group after sorting twice (Input 1) and its index (Input 2), this step would sort and classify Input 1 again with the specified layer spacing (Standard 3). This step outputs the index after sorting three times.
7. Unpack and Merge Data This step turns four-dimensional array-like variables into three-dimensional variables based on the indexes after sorting three times.

8. Unpack and Merge Data After last step "Unpack And Merge Data", this step could turn the three-dimensional array-like variables into two-dimensional variables.
9. Unpack and Merge Data After last step "Unpack And Merge Data" twice, this step could turn the two-dimensional array-like variables into a one-dimensional array as the final output.

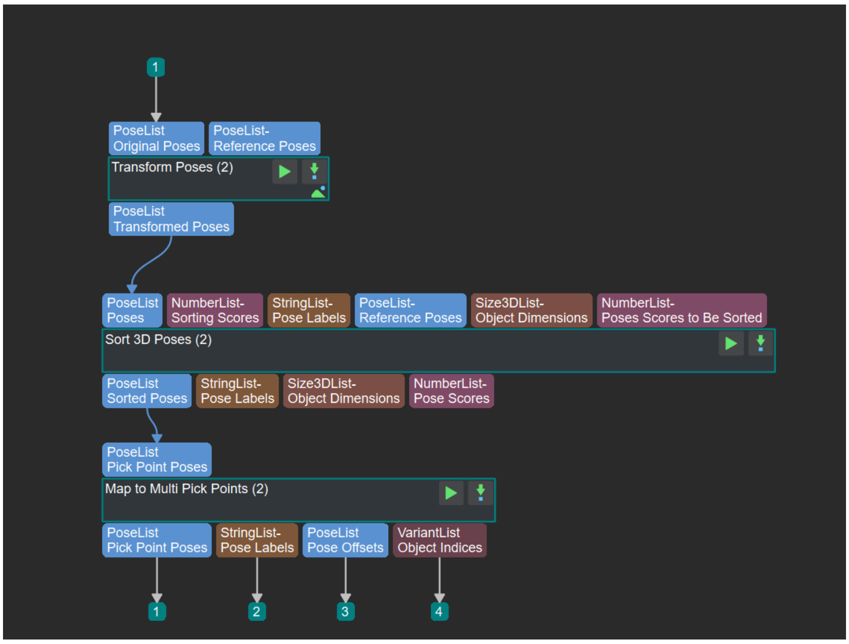
4.3.387.3. Sort Pick Points

Function

This Procedure converts poses under the camera reference frame to the robot reference frame, then sorts poses according to a specified rule, and finally maps poses to multiple pick points based on a pick point model.

- Input: the pose list to be sorted.
- Output: the pick point pose list, the pose classification label list, the pose offsets, and the object indices.

The graphical programming of the Procedure is shown below.



The functions of the Steps involved are as follows (please click the links for detailed descriptions of the Steps):

Transform Poses transforms the input pick point poses under the camera reference frame to the robot reference frame to enable robot picking.

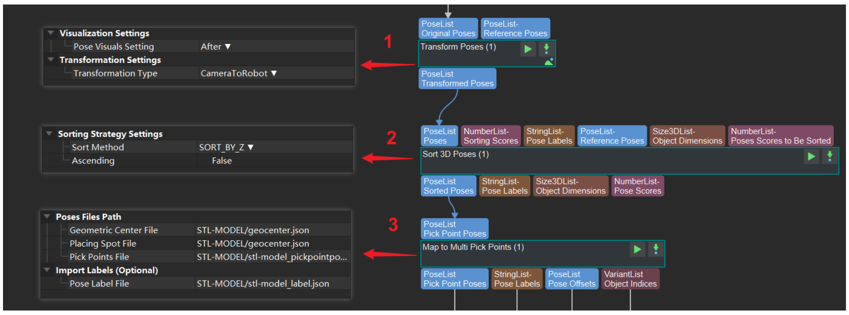
Sort 3D Poses sorts the input transformed pick point pose list based on some specific rules, such as sorting by the X values in descending order and sorting by the pose scores in ascending order.

Map to Multiple Pick Points maps each sorted pose to multiple pick points based on a pick point model to increase the feasibility of picking.

Usage Example

In the example, this Procedure converts poses under the camera reference frame to the robot reference frame, then sorts poses by the Z values in descending order, and maps each pose to multiple pick points for output in the end.

The figure below shows the settings of parameters.



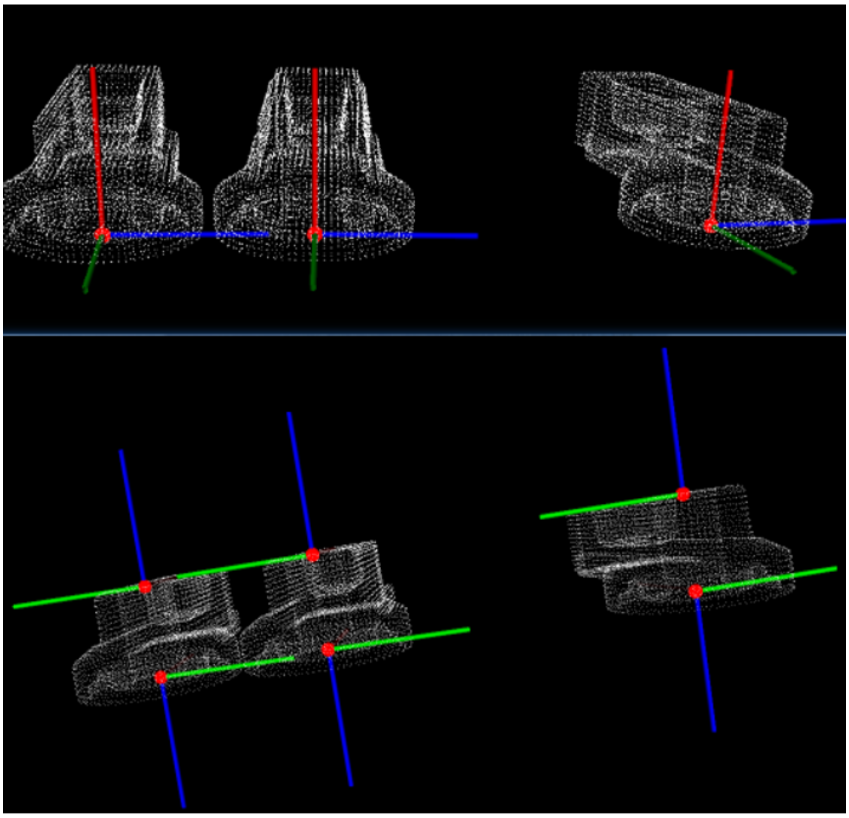
Visualization Settings
Pose Visuals Setting: After
Transformation Settings
Transformation Type: CameraToRobot

Sorting Strategy Settings
Sort Method: SORT_BY_Z
Ascending: False

Poses Files Path
Geometric Center File: STL-MODEL/geocenter.json
Placing Spot File: STL-MODEL/geocenter.json
Pick Points File: STL-MODEL/stl-model_pickpointpo...
Import Labels (Optional)
Pose Label File: STL-MODEL/stl-model_label.json

1. Transform poses to the robot reference frame.
2. Sort poses by the Z values in descending order.
3. Map each pose to multiple pick points.

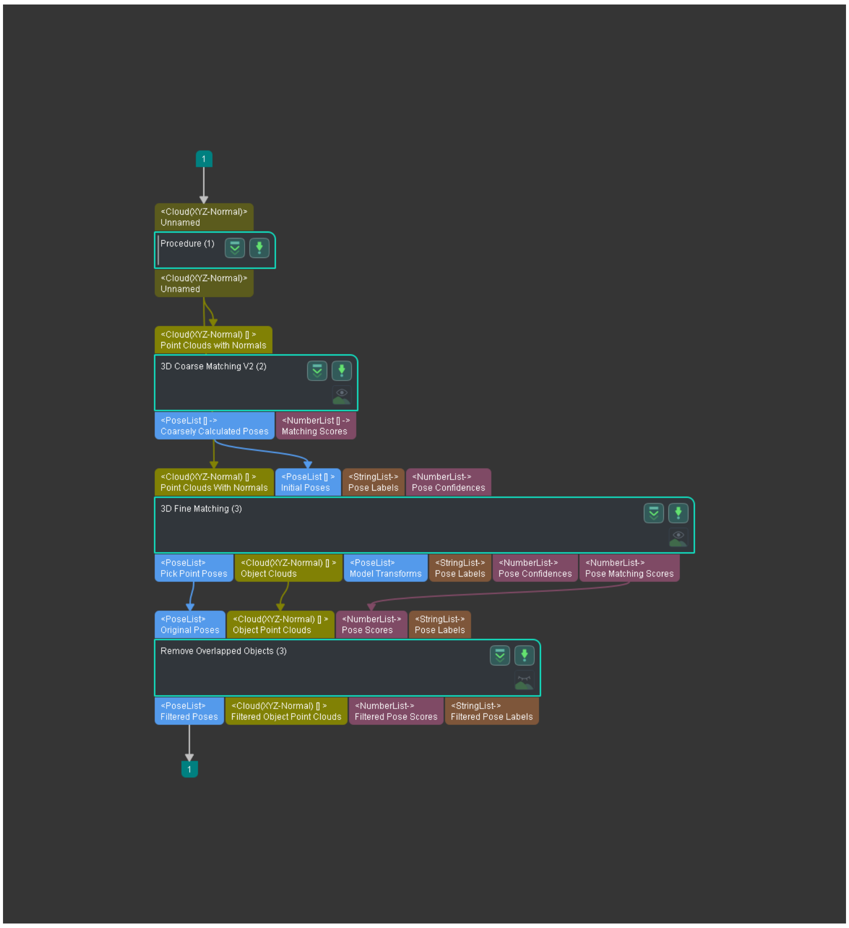
The upper and lower part of the figure below shows the poses and the mapped multiple pick points, respectively.



4.3.388. 3D Matching

Function

This Procedure is used to match the point cloud model with the scene point clouds and detect the object in the scene, and then output the object poses. The components of the Procedure are shown as below.

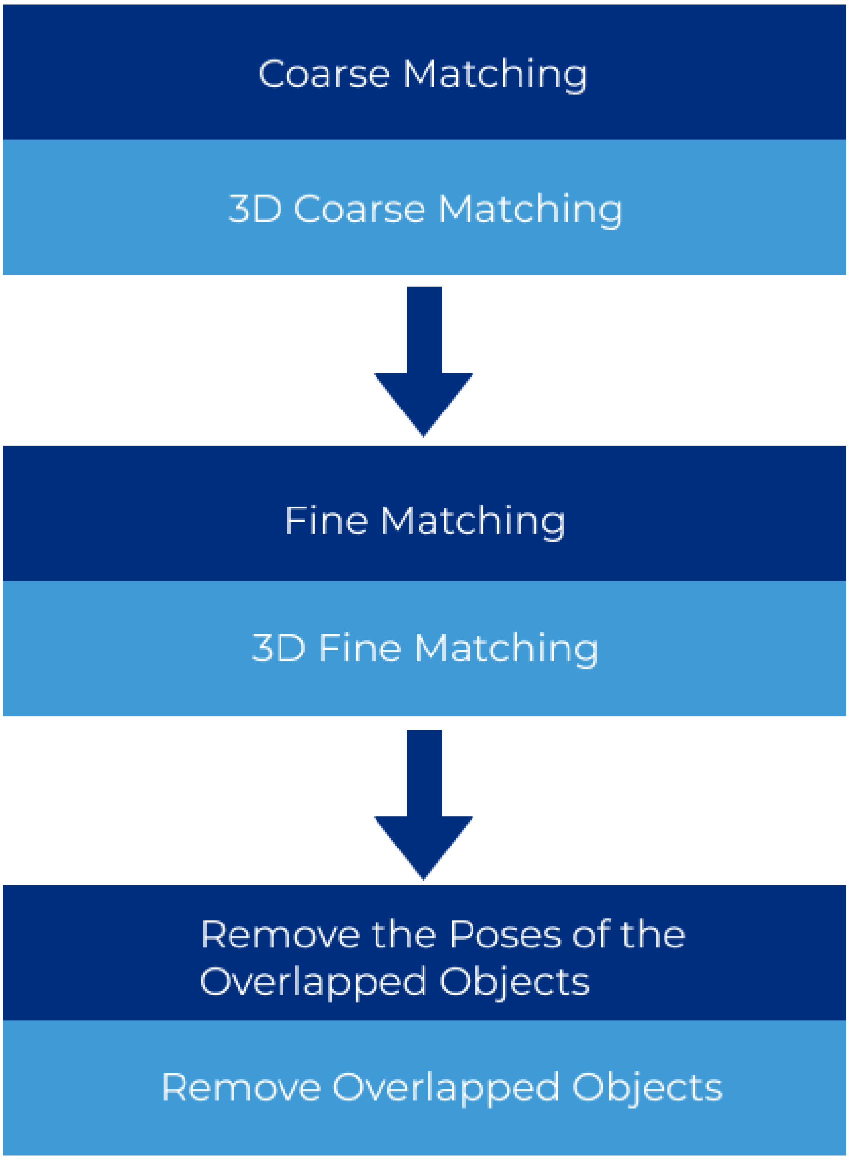

1
<Cloud(XYZ-Normal)> Unnamed
Procedure (1)
<Cloud(XYZ-Normal)> Unnamed
<Cloud(XYZ-Normal) [] > Point Clouds with Normals
3D Coarse Matching V2 (2)
<PoseList [] -> Coarsely Calculated Poses
<NumberList [] -> Matching Scores
<Cloud(XYZ-Normal) [] > Point Clouds With Normals
<PoseList [] > Initial Poses
<StringList-> Pose Labels
<NumberList-> Pose Confidences
3D Fine Matching (3)
<PoseList> Pick Point Poses
<Cloud(XYZ-Normal) [] > Object Clouds
<PoseList> Model Transforms
<StringList-> Pose Labels
<NumberList-> Pose Confidences
<NumberList-> Pose Matching Scores
<PoseList> Original Poses
<Cloud(XYZ-Normal) [] > Object Point Clouds
<NumberList-> Pose Scores
<StringList-> Pose Labels
Remove Overlapped Objects (3)
<PoseList> Filtered Poses
<Cloud(XYZ-Normal) [] > Filtered Object Point Clouds
<NumberList-> Filtered Pose Scores
<StringList-> Filtered Pose Labels
1

Usage Scenario

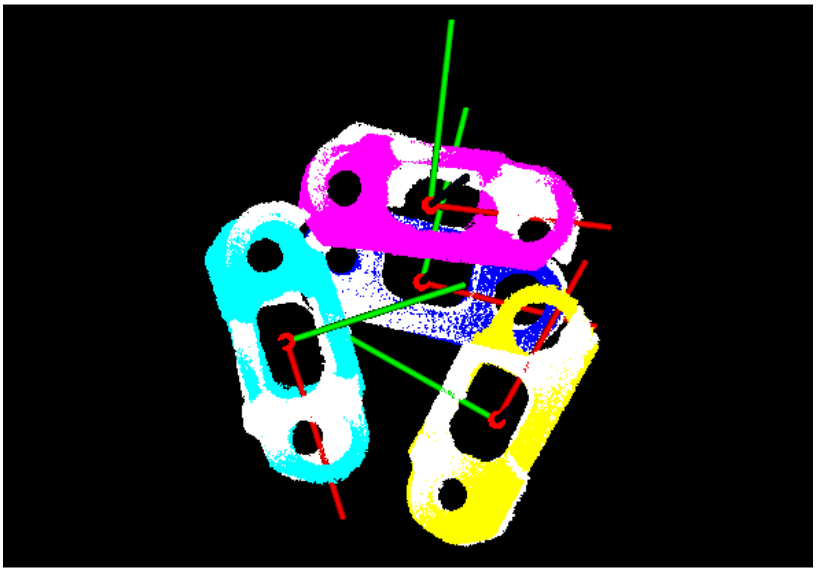
This Procedure usually follows the Filter Out Point Clouds That Exceed The Limit Procedure. It uses the 3D Coarse Matching V2 and 3D Fine Matching Steps to gradually optimize the pick poses and remove the poses of the overlapped objects.

Matching Process

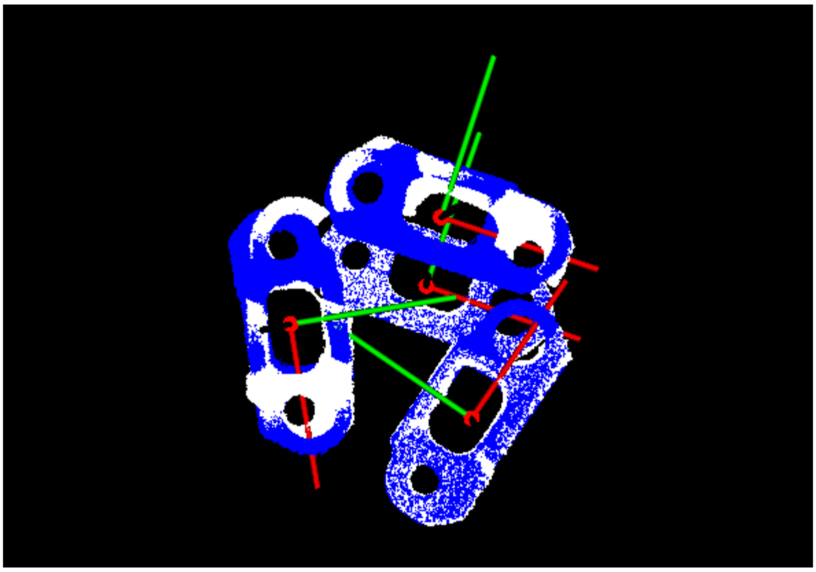
The matching workflow of this Procedure is shown below:


Coarse Matching
3D Coarse Matching
Fine Matching
3D Fine Matching
Remove the Poses of the Overlapped Objects
Remove Overlapped Objects

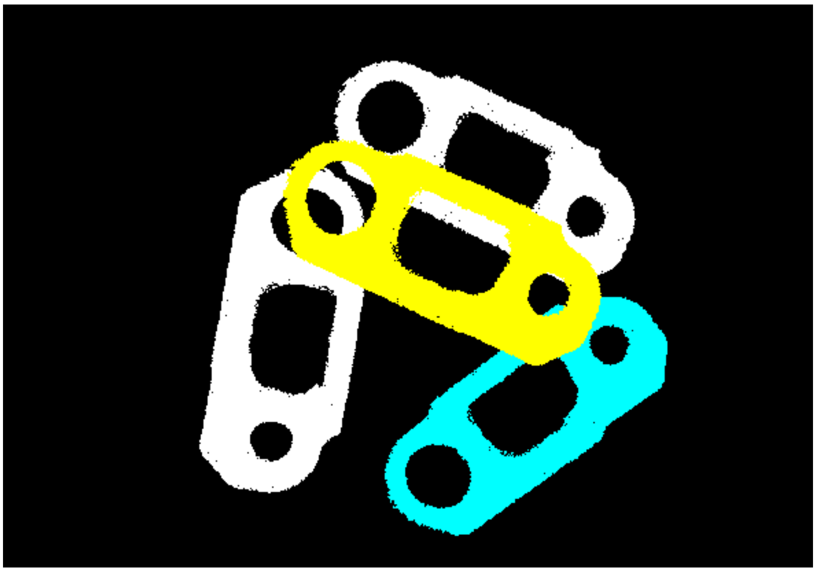
1. Once you obtain the filtered point cloud with the Filter Out Point Clouds That Exceed The Limit Procedure, you can input the filtered point cloud to this Procedure. The 3D Coarse Matching V2 Step can calculate the original pose of the object, as shown below. There can be multiple original poses.



2. Then the 3D Fine Matching Step will accurately match the original poses and output more accurate poses, as shown below.



3. Lastly, object poses will be selected in the horizontal direction, and a Remove Overlapped Objects Step is used to remove the overlapped objects along the Z-axis direction in the object reference frame and output the pick poses on the top layer. As shown in the figure below, the white point clouds are removed.



Parameter Tuning

Model Setting

Model Name

Default value: Null.

Notes: Before you use this Procedure, you should add a point cloud model of the object. Then the point cloud model will be matched with the scene point clouds. In this way, the object in the scene can be detected and their poses can be calculated. After you make a point cloud model and save it with the Matching Model and Pick Point Editor, you can select the saved point cloud model in the drop-down list.

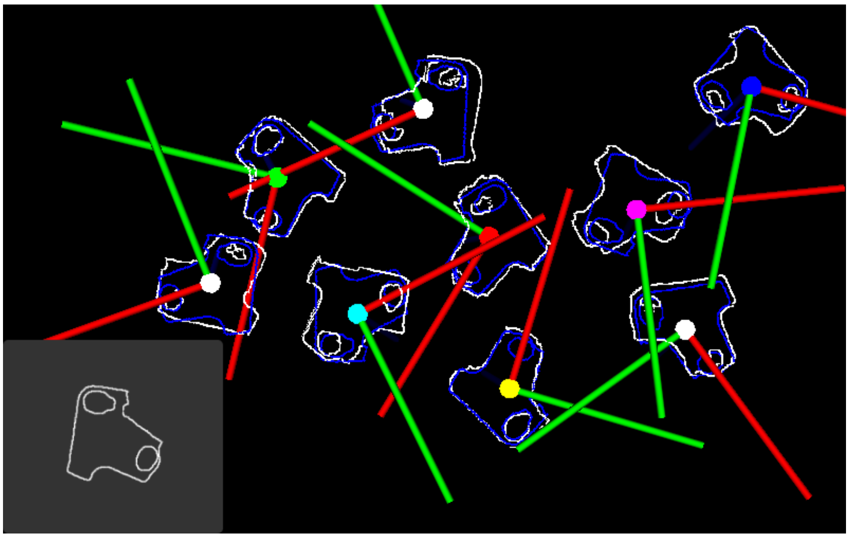
Matching Settings

Matching Method

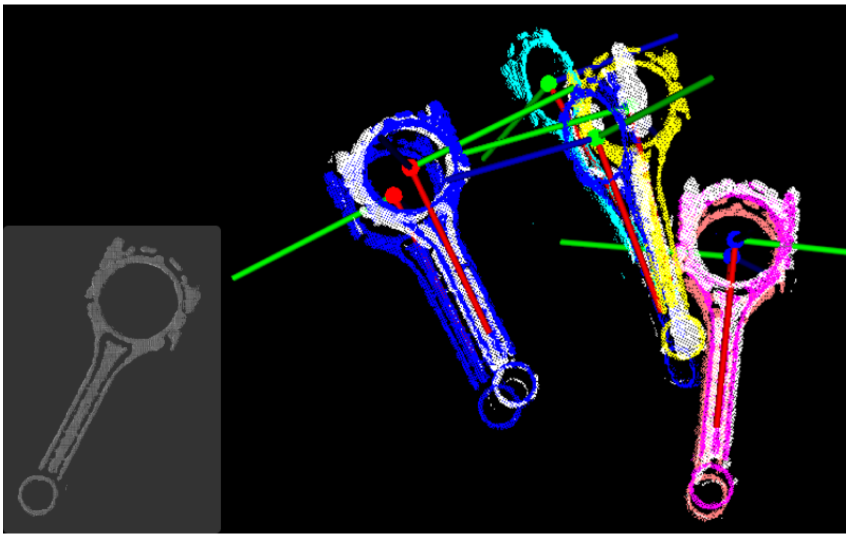
Default value: Edge

Value list: Surface and Edge.

Instruction: Matching methods include Surface Matching and Edge Matching. If the matching mode is Edge Matching, you should select an edge point cloud model in Model Settings, as shown below.



If the matching mode is Surface Matching, you should select an surface point cloud model in Model Settings, as shown below.



Expected Point Number of Sampled Model

Default value: 300

Instruction: This parameter specifies the number of points in the point cloud model after the automatic downsampling. The smaller the value, the less points in the point cloud, and the lower the matching accuracy will be.

Parameter	300	600
Illustration		
Number of points	276	632

Matching Speed

Default value: Standard

Value list: HighSpeed, Standard, and HighPrecision.

Instruction: Please select the appropriate mode according to the requirement of the project.

- HighSpeed: high speed but with less matching accuracy.
- Standard: a relatively balanced mode.
- HighPrecision: high matching accuracy with low speed.

Confidence Threshold

Default value: 0.6

Instruction: This parameter is used to compare with the pose matching score to obtain more accurate pose matching results. It is usually used with the **Search Radius** parameter. If the pose matching score is above this value, the pose matching result is valid. When all other parameters remain unchanged, the higher the value, the more accurate the pose matching result will be.

In the actual application, you should set an appropriate **Search Radius** first, and set the **Confidence Threshold** according to the actual situation.You can double-click the Procedure to view the Steps it contains, and set the Search Radius in the 3D Fine Matching Step.

The value of **Search Radius** affects the pose matching score. The higher the value of the Search Radius, the looser the criterion for evaluating the pose matching result will be, which means that even if the pose matching result is of poor quality, the pose matching score will be high; the lower the value of the Search Radius, the more strict the criterion for evaluating the pose matching result will be. Please set the value according to the actual situation of the point cloud. If the points in the point cloud are sparse, the **Search Radius** should be set slightly larger.

During parameter tuning, you can set an original **Search Radius** first and run the Step. Then you can check the matching result in the Debug Output window and compare the result with the pose matching score. If the matching result does not correspond with the matching score, please adjust the **Search Radius** according to the actual situation.

Once you set an appropriate **Search Radius**, you can set the **Confidence Threshold** according to the actual requirement to obtain the pose matching results with scores above the threshold.

Result Settings

Maximum Output Number

Default value: 10

Value range: 1–100

Instruction: This parameter specifies the maximum number of pose matching results output by the Procedure. If there are multiple pose matching results, the results will be sorted according to the pose matching score in an descending order, and the pose matching results with the highest scores of the **Maximum Output Number** will be output.

> The actual number of output pose matching results is not necessarily equal to the set **Maximum Output Number**. If the **Maximum Output Number** is set to 5, but there are 3 pose matching results in total, the final number of output pose matching results is 3.

4.3.389. Apply Masks to Color Image

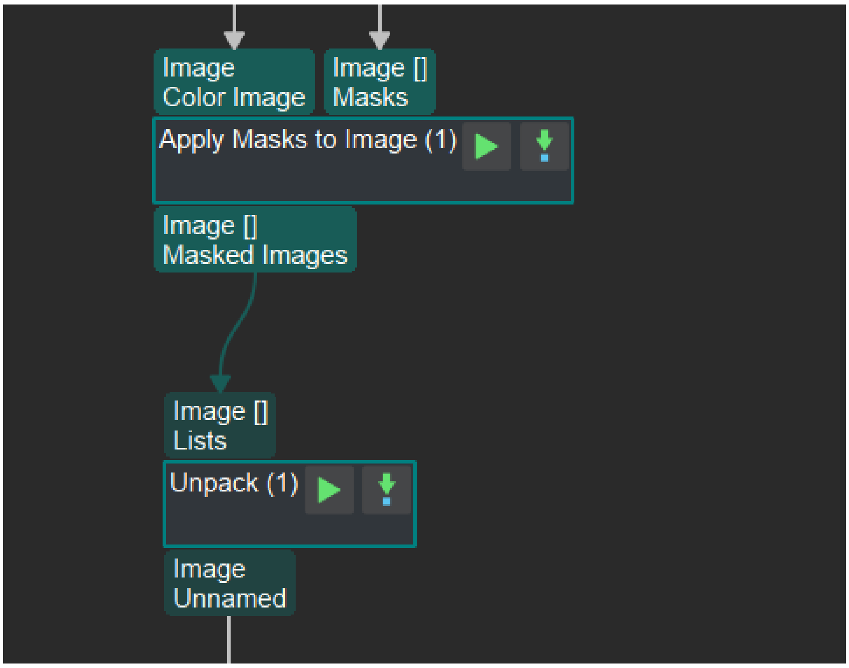
Function

This Procedure obtains the color image part under the mask of the first index in the input mask list.

- Input: color image, mask list.

- Output: the color image part under the mask of the first index in the input mask list.

The graphical programming of this Procedure is shown in the figure below.



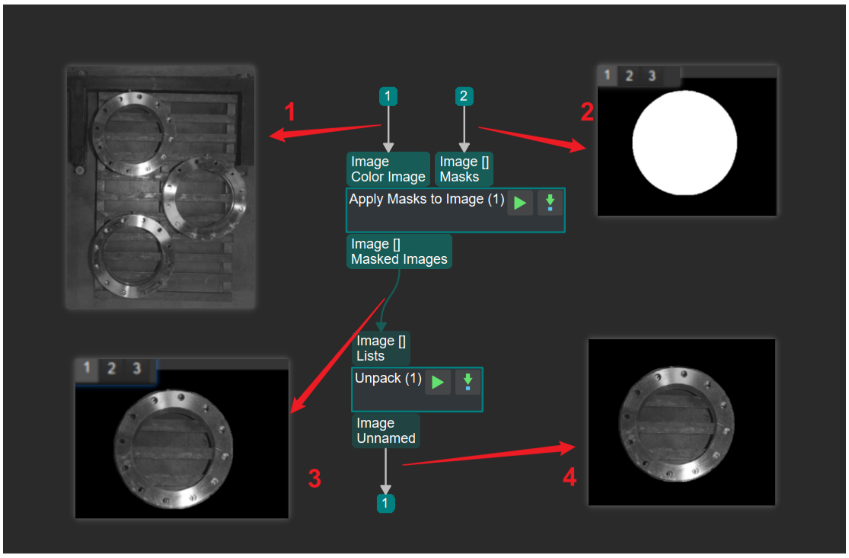
Functions of Involved Steps

The functions of the Steps are as follows (please click the link for detailed descriptions):

1. Apply Masks to Image obtains the image parts corresponding to the masks in the mask list.
2. Obtain the first image from the output of **Apply Masks to Image**.

Usage Scenario

This Procedure can obtain the image part under the mask of the first index in the mask list output from an Instance Segmentation Step.



1. Input color image.

2. Workpiece mask list output from an Instance Segmentation Step, including three masks in this example.
3. List of color image parts under the masks, including three color image parts in this example.
4. The color image part corresponding to the mask of the first index in the mask list.

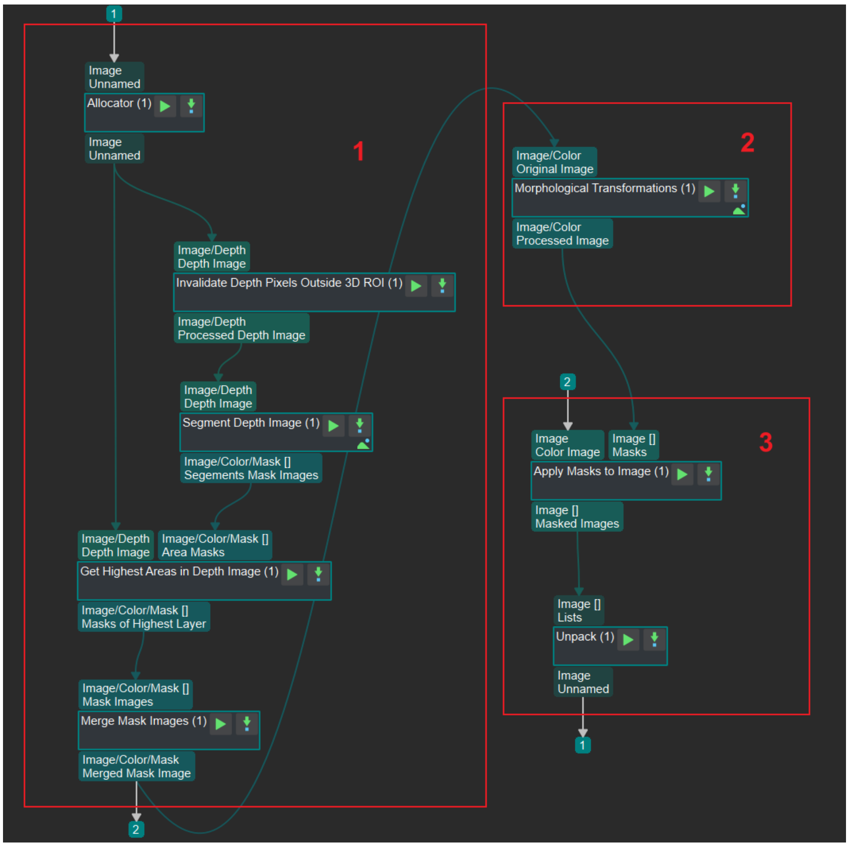
Comparison of Similar Procedures

Apply Masks to Color Image outputs the image part under the mask of the first index in the input mask list. Apply Masks to Image outputs the list of image parts under all masks in the input mask list.

4.3.390. Calc Color Image for Highest Layer

Function

This Procedure detects the area of the highest layer in the scene based on the depth map and then outputs the 2D image and the mask of the highest-layer area. The Steps in the Procedure are shown below.

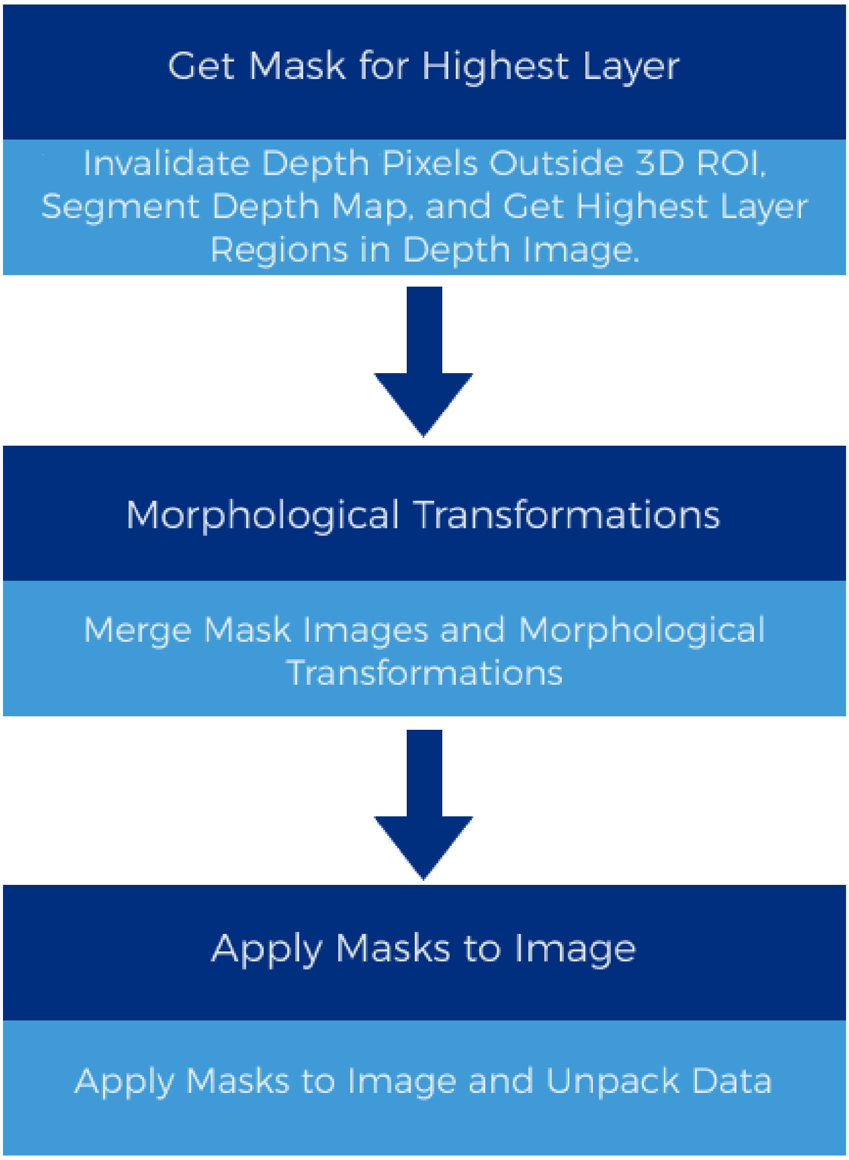


Usage Scenario

This Procedure can extract the color image segment of the highest layer for deep learning recognition.

Processing Workflow and Parameter Tuning

The workflow of this Procedure is shown below:

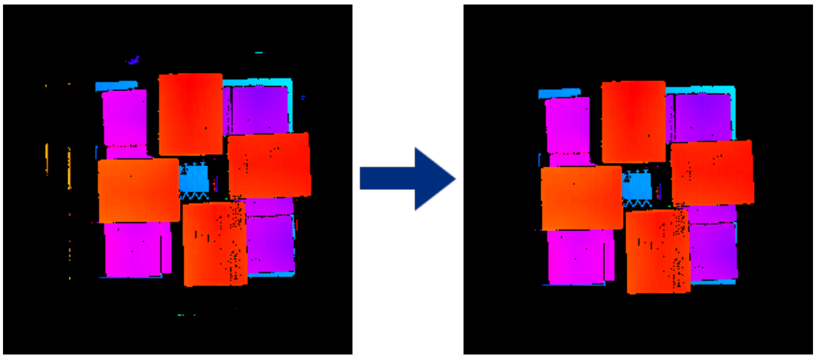

Get Mask for Highest Layer
Invalidate Depth Pixels Outside 3D ROI, Segment Depth Map, and Get Highest Layer Regions in Depth Image.
Morphological Transformations
Merge Mask Images and Morphological Transformations
Apply Masks to Image
Apply Masks to Image and Unpack Data

Get Mask for Highest Layer

1. To filter out unwanted point clouds, an ROI should be selected in the original depth map, thus the processing efficiency in the subsequent Steps can be improved.

   Double-click to enter this Procedure. Click the **[Set 3D ROI]** button in the Step Parameters panel of Invalidate Depth Pixels Outside 3D ROI and select 3D ROI in the pop-up Set ROI window.

   The following show the point clouds before and after 3D ROI selection. The image on the left is the original depth map, containing many irrelevant point clouds; the image on the right is the depth map after 3D ROI selection, which no longer contains point clouds of regions outside the selected ROI.

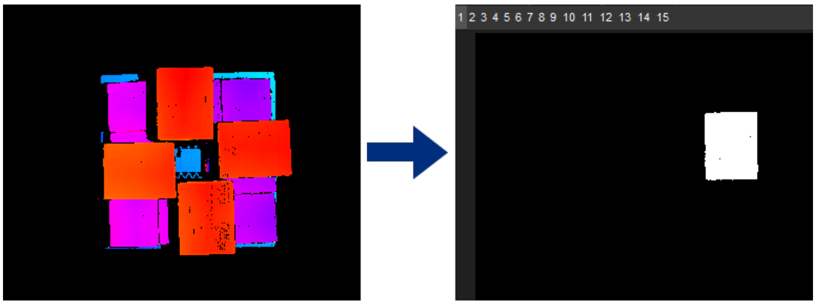


2. Segment Depth Map segments the depth map within the ROI based on each pixel’s depth and generates the corresponding mask(s).

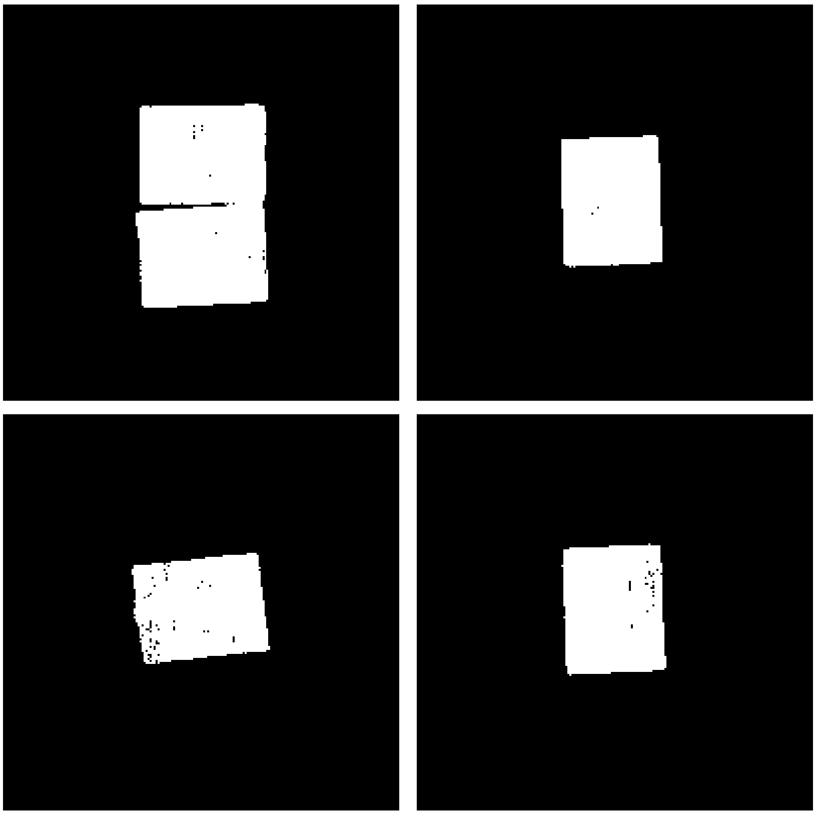
   In the Step Parameters Panel of Segment Depth Map, click the [Set 2D ROI] button and select 2D ROI in the pop-up Set ROI window.

   Then, set **Max Height Diff between Neighbouring Pixels** to determine whether the neighboring pixels belong to the same segment; set **Min Segment Area** to determine the valid segment.

   As shown below, the image on the left is the depth map before segmentation, and that on the right is the depth image after segmentation.

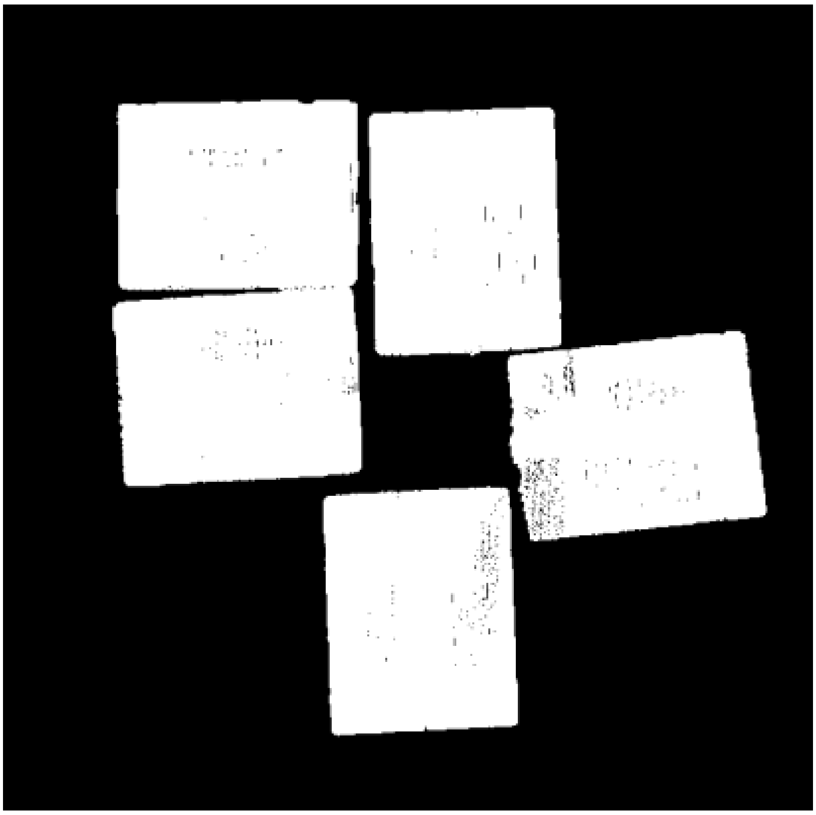


3. Get Highest Layer Regions in Depth Map obtains the highest area(s) in a depth map based on the masks and masks’ depth information output from Segment Depth Map and generates the mask areas of the valid region closest to the camera under proper **Layer Height**. There could be one or multiple areas that satisfy this condition.

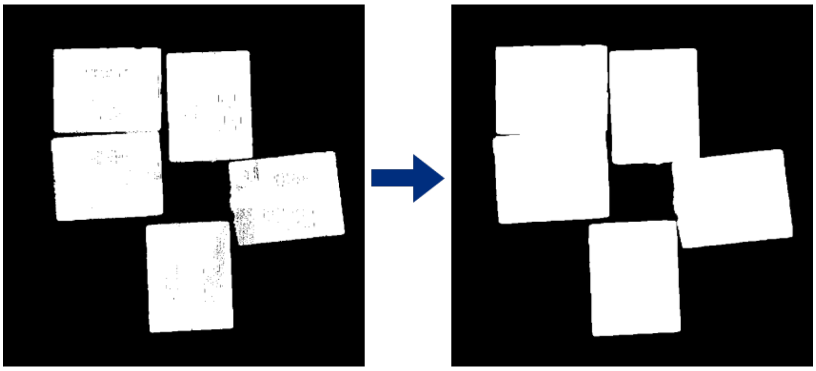


Morphological Transformations

1. Merge Mask Images merges the masks input from Get Highest Layer Regions in Depth Map if the input contains more than one mask to facilitate the further morphological transformation. The following image shows the mask of the highest layer after merging.

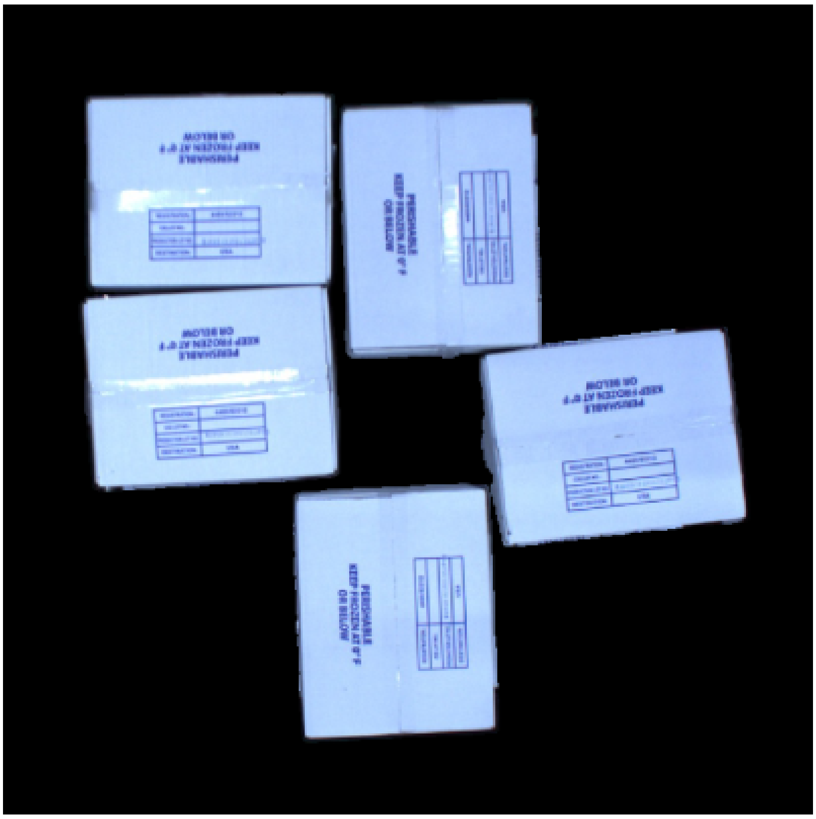


2. Morphological Transformations **dilates** the mask(s) to facilitate recognition or picking later (DilateOperator is selected).

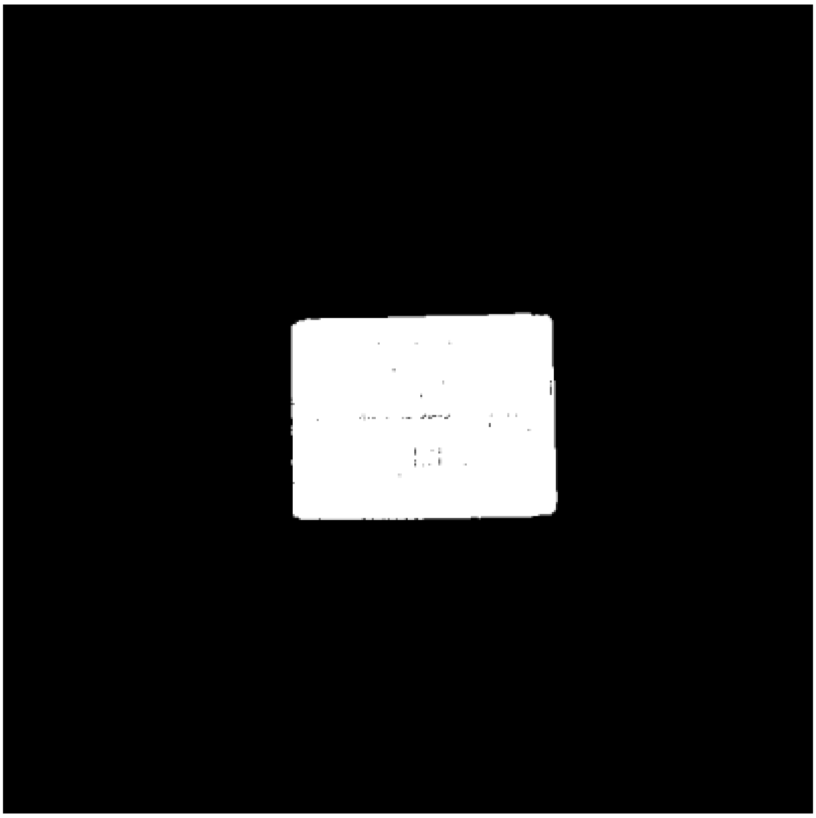


Get Color image Segment under the Mask

1. Apply Masks to Image obtains the color image segment(s) under the transformed mask(s), as shown below.



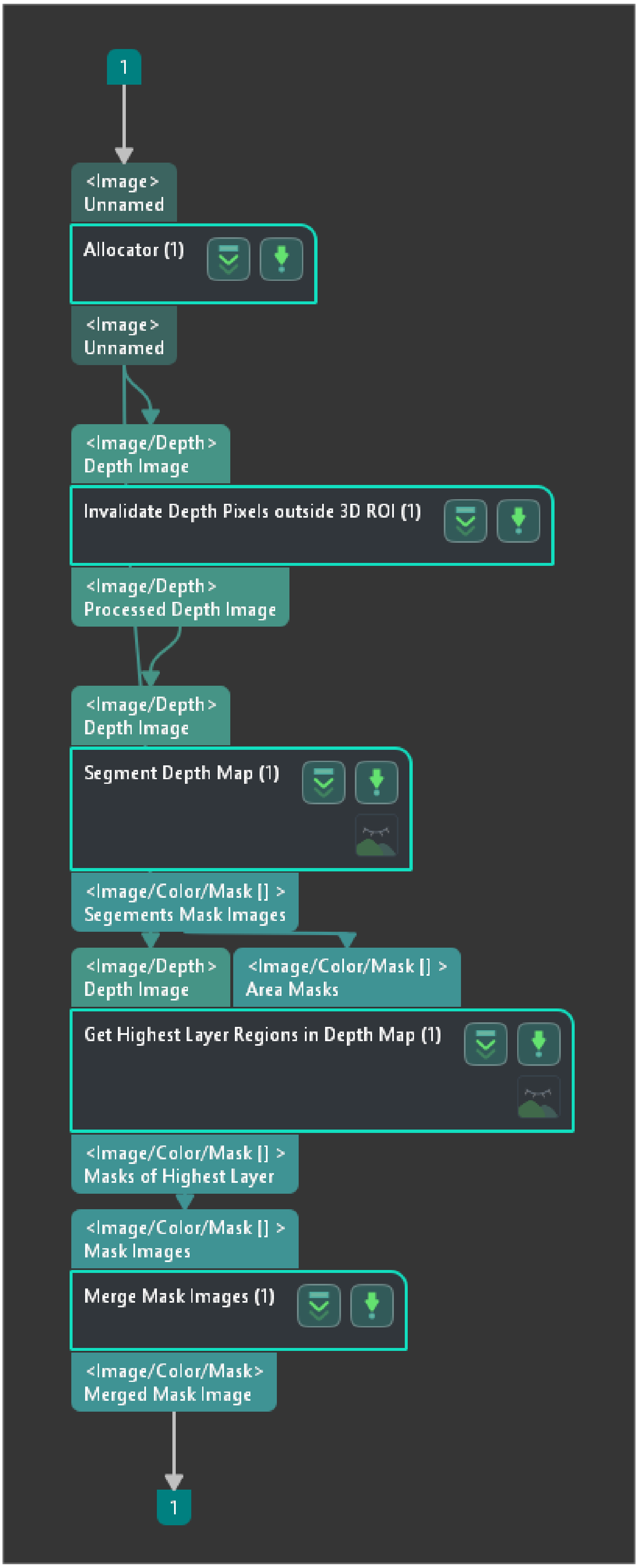
2. Unpack Data obtains the color image of the first index in the list of image segments under masks.



4.3.391. Get Mask for Highest Layer

Function

This Procedure is mainly used to get the mask for the highest-layer. The components of the Procedure are shown as below.

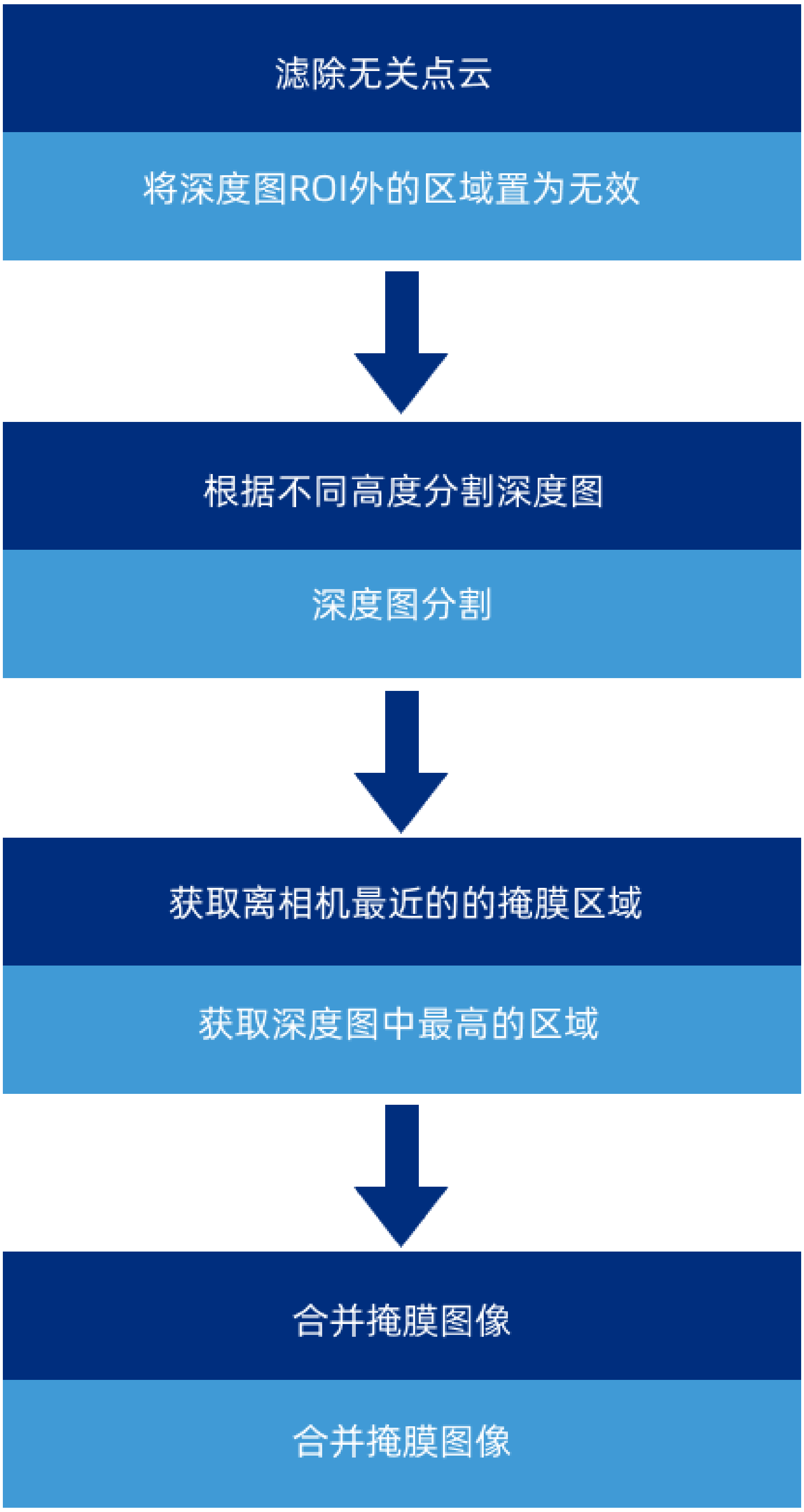

1
<Image>
Unnamed
Allocator (1)
<Image>
Unnamed
<Image/Depth>
Depth Image
Invalidate Depth Pixels outside 3D ROI (1)
<Image/Depth>
Processed Depth Image
<Image/Depth>
Depth Image
Segment Depth Map (1)
<Image/Color/Mask [] >
Segements Mask Images
<Image/Depth>
Depth Image
<Image/Color/Mask [] >
Area Masks
Get Highest Layer Regions in Depth Map (1)
<Image/Color/Mask [] >
Masks of Highest Layer
<Image/Color/Mask [] >
Mask Images
Merge Mask Images (1)
<Image/Color/Mask>
Merged Mask Image
1

Usage Scenario

Usually, this Procedure is applied when the depth situation is quite complex and should be separated into layers. It can remove invalid regions and stratify the depth map to get the mask for the highest layer.

Processing Workflow and Parameter Tuning

The workflow of this Procedure is shown below:



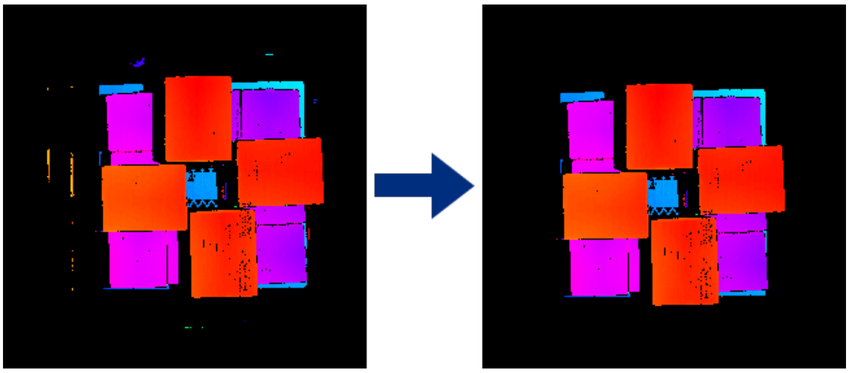
Filter Out Point Clouds That Exceed The Limit

Point clouds that exceed the limit should be filtered out for higher processing efficiency of subsequent Steps. Therefore, only needed areas of the depth map can be retained, which requires an ROI to be selected in the original depth map.

Double-click to enter this Procedure. Click the **[Set 3D ROI]** button in the Step Parameters panel of

Invalidate Depth Pixels Outside 3D ROI and select 3D ROI in the pop-up Set ROI window.

The following show the point clouds before and after 3D ROI selection. The image on the left is the original depth map, containing many irrelevant point clouds; the image on the right is the depth map after 3D ROI selection, which no longer contains point clouds of regions outside the selected ROI.



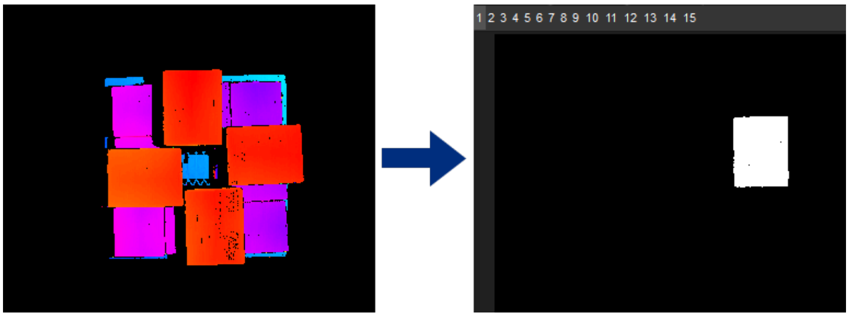
Segment Depth Map by Depth

Segment Depth Map segments the depth map within the ROI based on each pixel's depth and generates the corresponding mask(s).

In the Step Parameters Panel of Segment Depth Map, click the **[Set 2D ROI]** button and select 2D ROI in the pop-up Set ROI window.

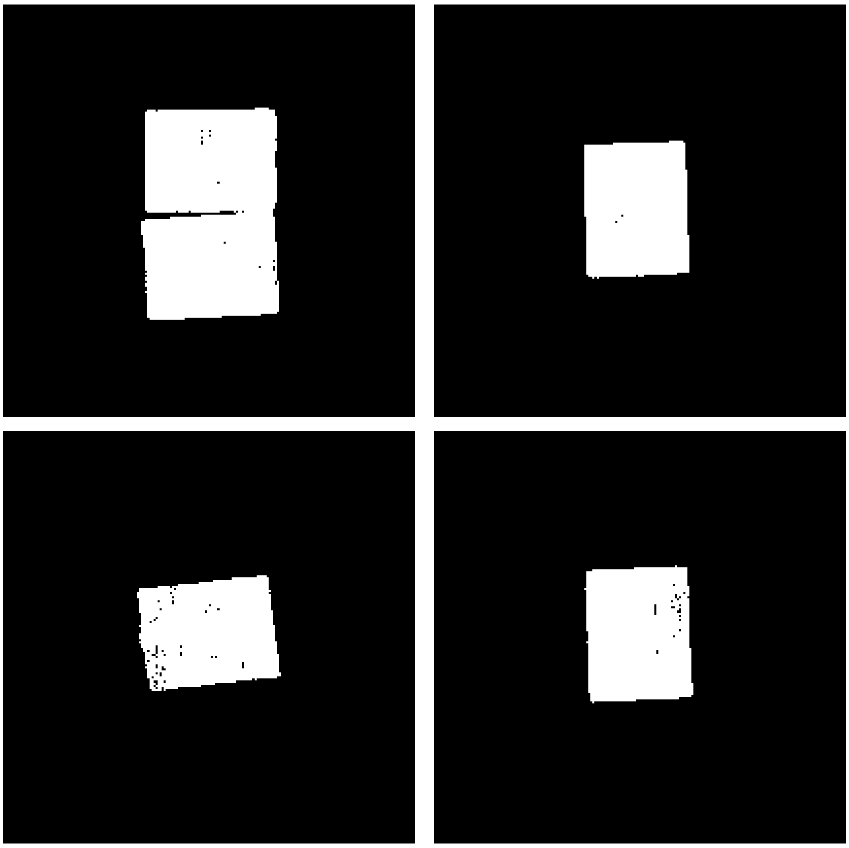
Then, set **Max Height Diff between Neighbouring Pixels** to determine whether the neighboring pixels belong to the same segment; and set **Min Segment Area** to determine the valid segment.

As shown below, the image on the left is the depth map before segmentation, and that on the right is the depth map after segmentation.



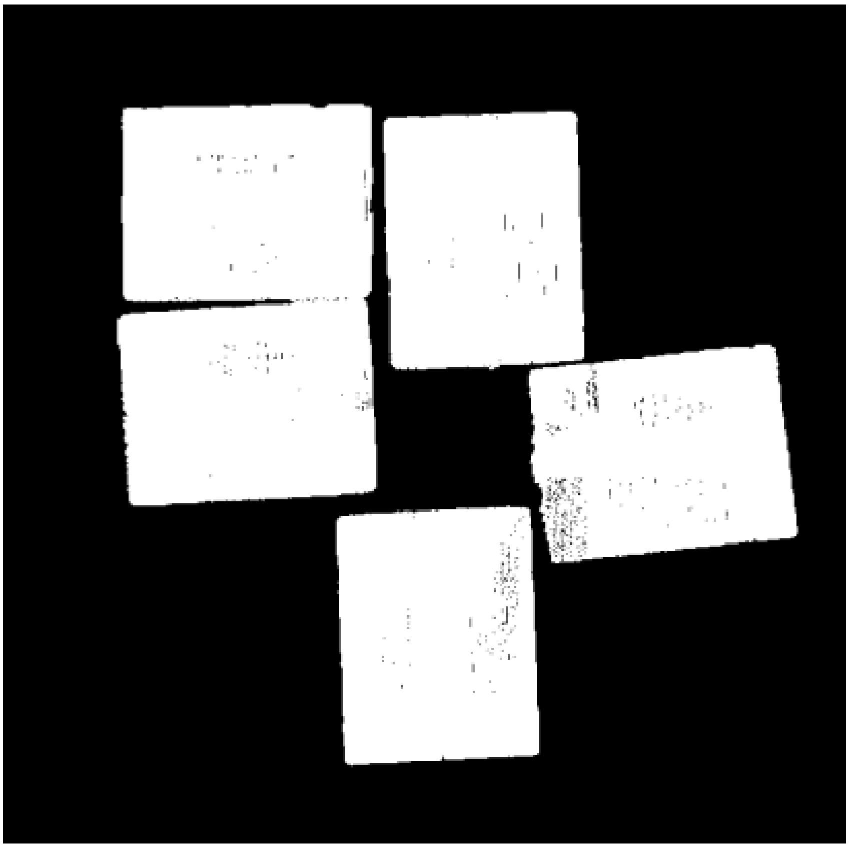
Get Highest Layer Regions in Depth Map

Get Highest Layer Regions in Depth Map obtains the highest area(s) in a depth map based on the masks and masks' depth information output from Segment Depth Map and generates the mask areas of the closest valid region to the camera under proper **Layer Height**. There could be one or multiple areas that satisfy this condition.



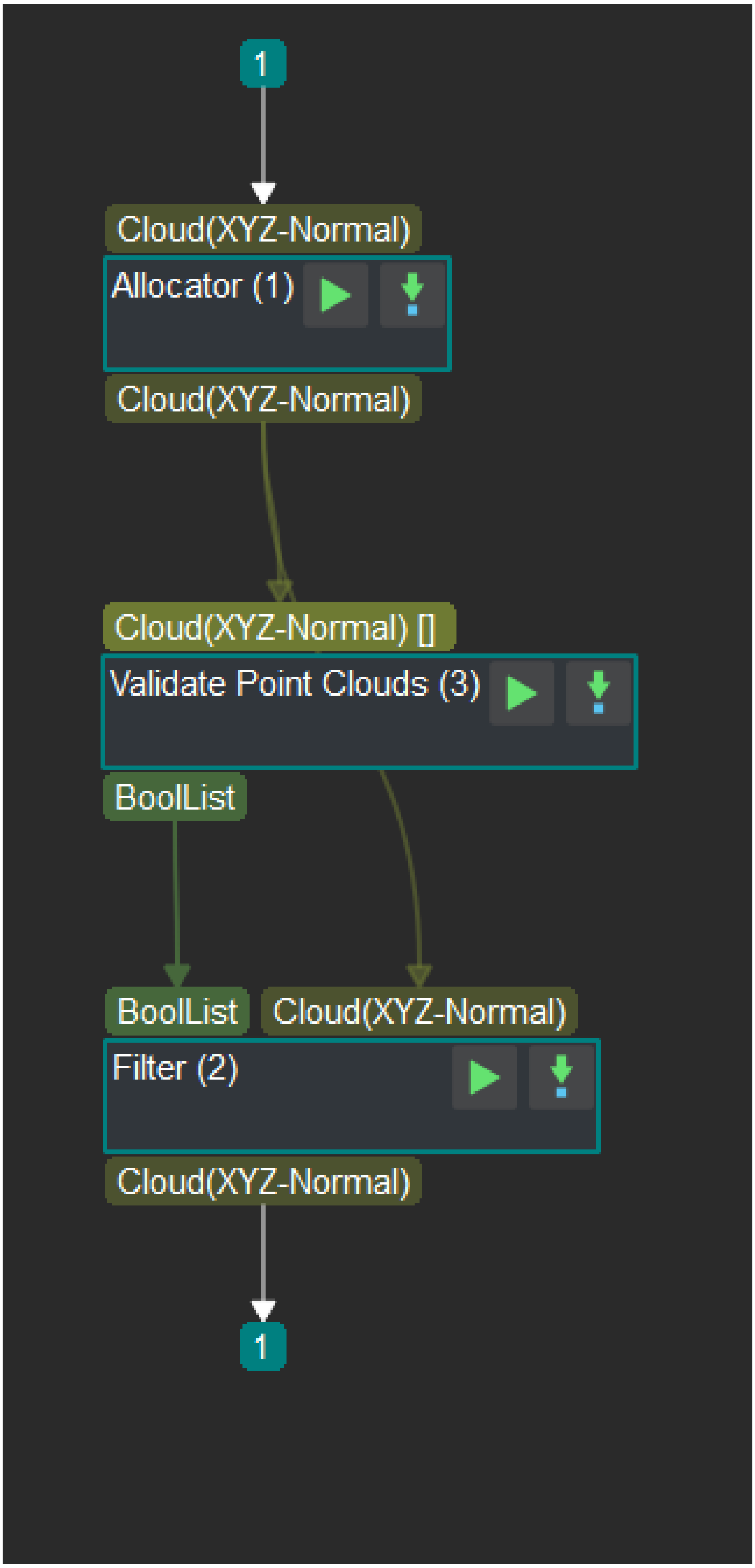
Merge Mask Images

Merge Mask Images merges the masks input from Get Highest Layer Regions in Depth Map if the input contains more than one mask to get the mask of the highest layer. The following image shows the mask of the highest layer after merging.



4.3.392. Filter Out Point Clouds That Exceed The Limit

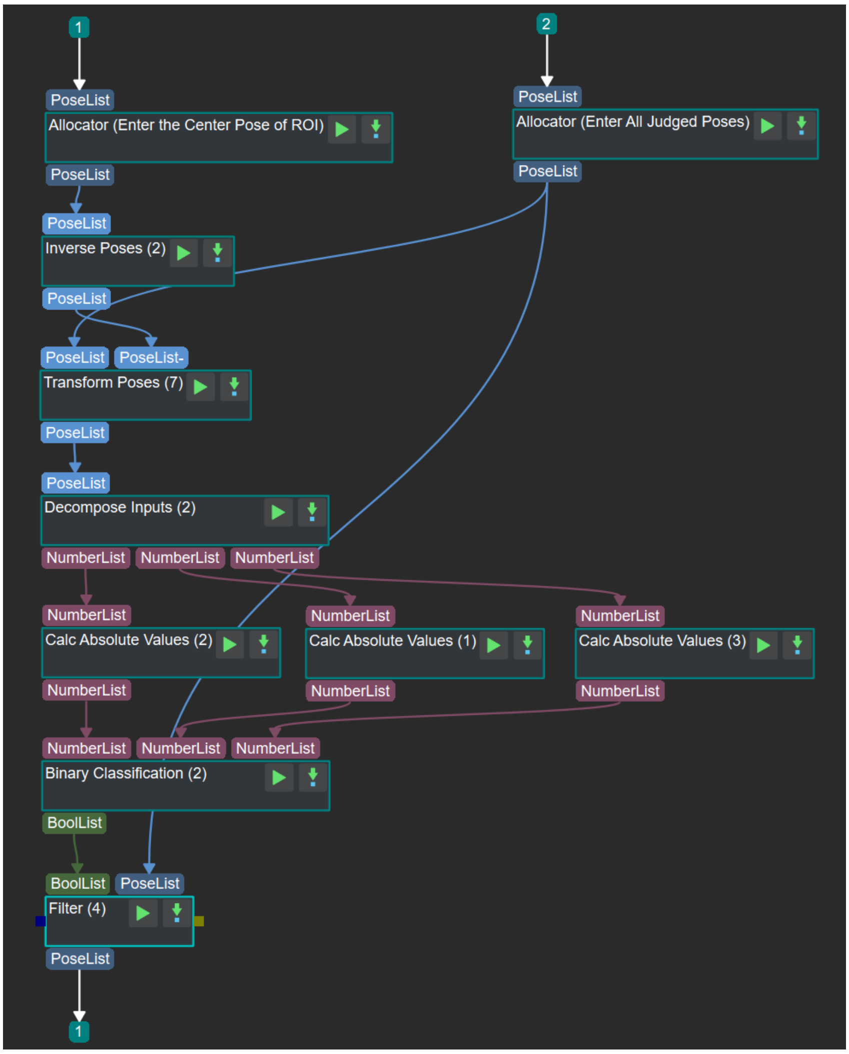
The main function of this step is to filter out point clouds whose number of points exceeds the limit, which can be set by the user. Thus, the project would run faster. The combination is as shown below.



Determine whether the point cloud is within the set range of point clouds by the Step Validate Point Clouds. Then, the output is input into the Step Filter to remove the point cloud that exceeds the size limit and output the filtered point cloud.

4.3.393. Filter Out Poses Out Of ROI

The main function of this step is to filter out poses out of the ROI to accelerate the subsequent process about poses. The combination is shown below.



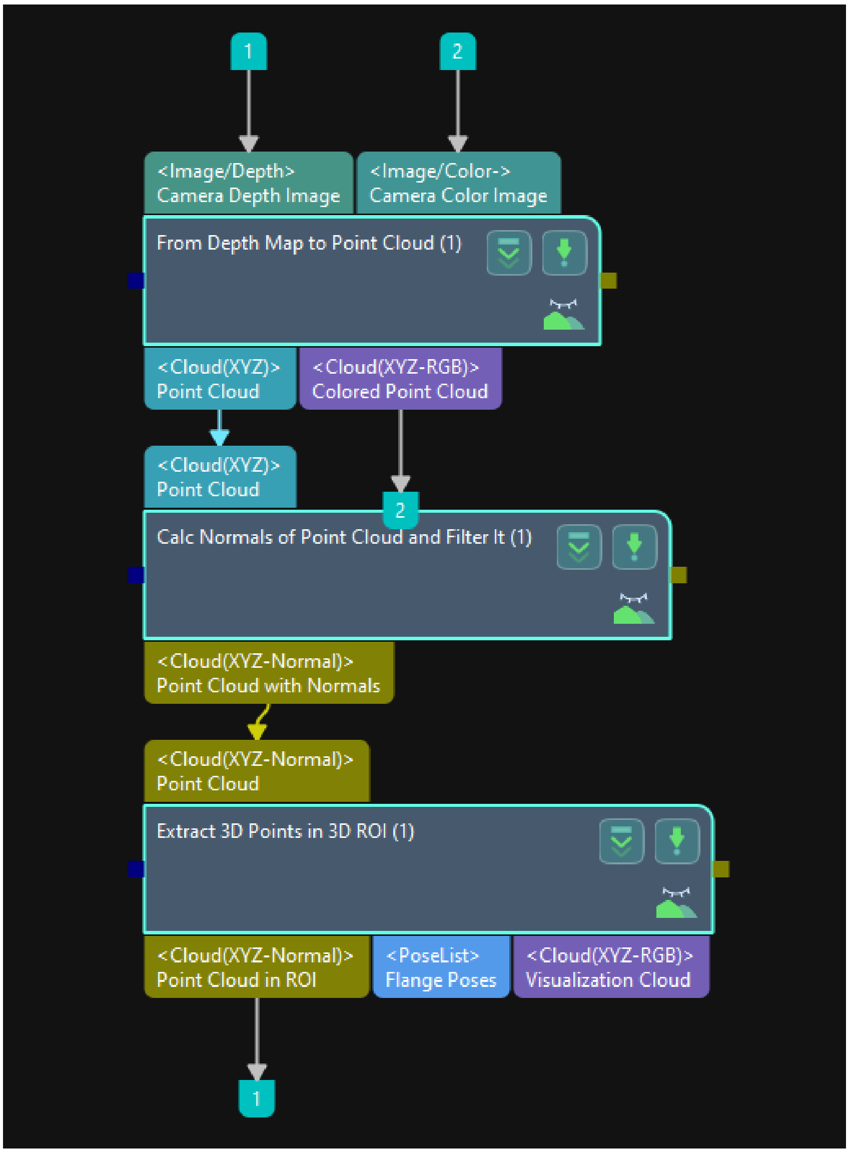
The introduction to each step is as follows:

1. Inverse Poses Calculate the inverse of the center pose of the ROI.
2. Transform Point Clouds Input the pose to be filtered (Input 1) and the center pose of the ROI (Input 2). This step would transform the relative coordinate of Input1 into the coordinate system of Input 2.
3. Decompose Object Dimensions and Calc Absolute Values Change the XYZ coordinate of the pose to absolute value to get ready for detecting whether the coordinate is in the ROI.
4. Dichotomize Values by Threshold Set the ROI and detect whether the XYZ coordinate(Absolute value) is in the ROI.
5. Filter Filter out poses out of the ROI.

4.3.394. Point Cloud Preprocessing

Function

The main function of this procedure is to generate the point cloud, remove noise, and then obtain the point cloud in the ROI, thus speeding up the process of subsequent Steps. The components of the Procedure are shown as below.


1
2
<Image/Depth> Camera Depth Image
<Image/Color-> Camera Color Image
From Depth Map to Point Cloud (1)
<Cloud(XYZ)> Point Cloud
<Cloud(XYZ-RGB)> Colored Point Cloud
<Cloud(XYZ)> Point Cloud
2
Calc Normals of Point Cloud and Filter It (1)
<Cloud(XYZ-Normal)> Point Cloud with Normals
<Cloud(XYZ-Normal)> Point Cloud
Extract 3D Points in 3D ROI (1)
<Cloud(XYZ-Normal)> Point Cloud in ROI
<PoseList> Flange Poses
<Cloud(XYZ-RGB)> Visualization Cloud
1

Usage Scenario

It usually follows the Step Capture Images from Camera to preprocess the original point cloud and thus accelerate the process of subsequent Steps.

Preprocessing Process

The Procedure includes the following Steps: From Depth Map to Point Cloud, Calc Normals of Point Cloud and Filter It, and Extract 3D Points in 3D ROI.

From Depth Map to Point Cloud

Input the original colored image and its depth image into the Step From Depth Map to Point Cloud, and the current scene point cloud is output.

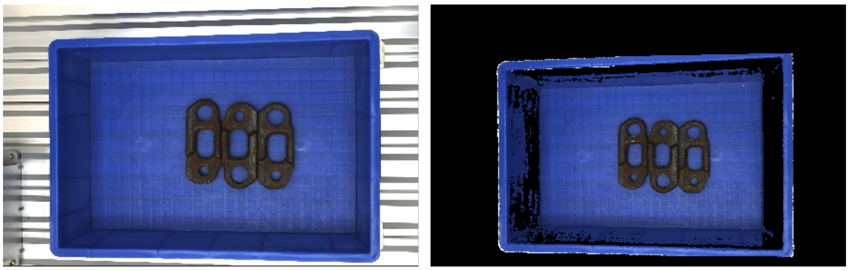
Calc Normals of Point Cloud and Filter It

Calculate the normal vectors and remove useless points of the point cloud obtained by inputting the scene point cloud to the Step Calc Normals of Point Cloud and Filter It, and the filtered point cloud with normal vectors is output.

Extract 3D Points in 3D ROI

Input the processed point cloud into the Step Extract 3D Points in 3D ROI and then extract part of the point cloud in the ROI.

The following figures show the results before and after point cloud preprocessing. The figure on the left is the result before point cloud preprocessing, and as no ROI is set, it contains the irrelevant point cloud in the scene. The figure on the right is that after point cloud preprocessing, and as a ROI is set, the useless point cloud is removed to facilitate the process of subsequent Steps.



Parameter Tuning

Extract 3D Points in 3D ROI

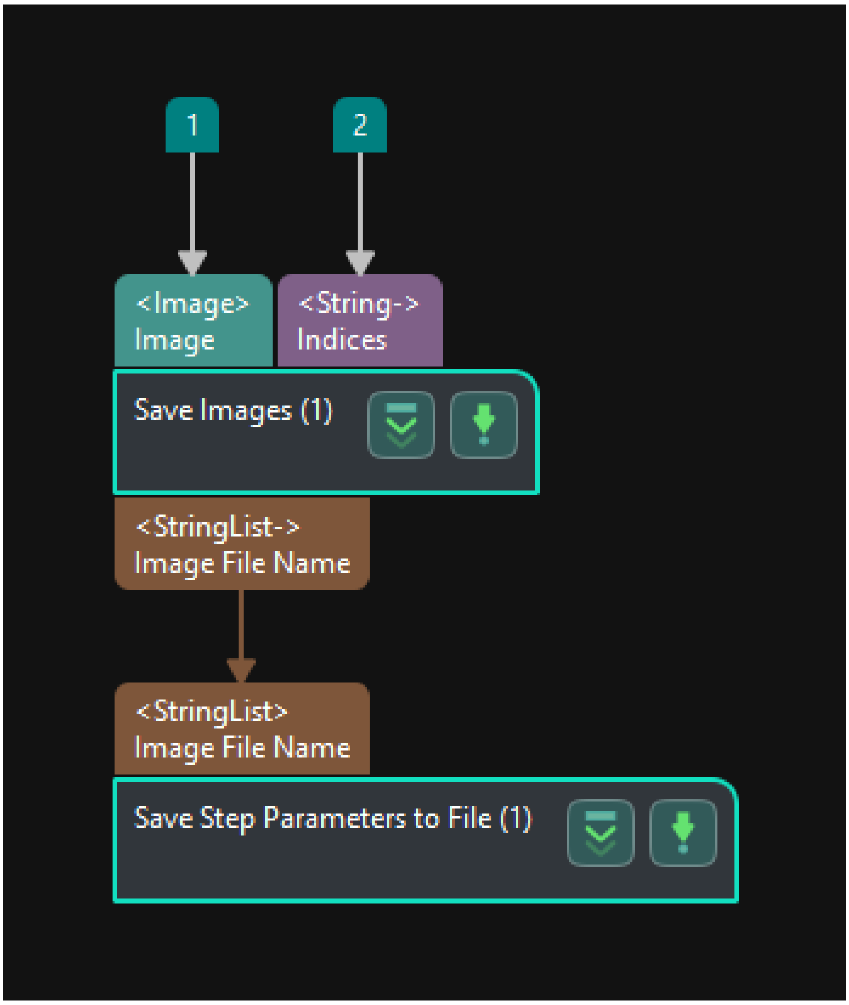
Set 3D ROI

Description: This parameter is used to set the 3D ROI.

Instruction: Click **[Set 3D ROI]**, and then set the 3D ROI in the pop-up window.

4.3.395. Save Images and Step Parameters

The main function of this Step is to save images under the specified directory and step parameters of the project. The combination is shown below.



The introduction to each step is as follows:

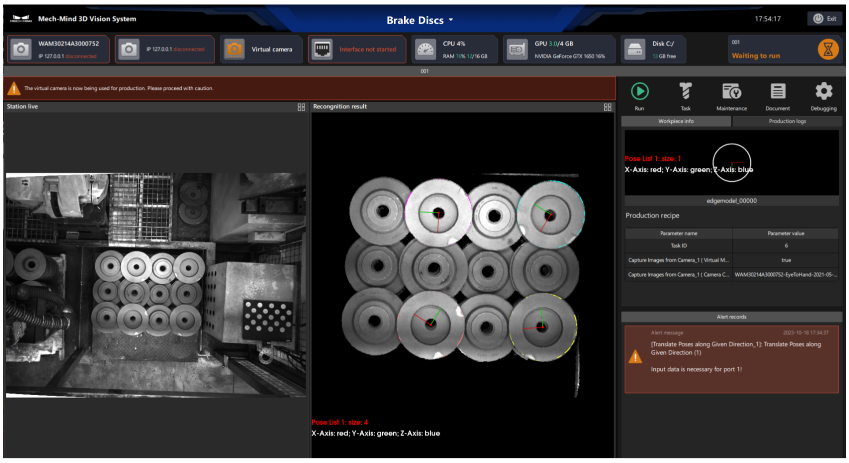
1. Input images to be saved, including colored and depth images and others, into the Step Save Images. Users can input multiple images together and add indexes to image names if needed.
2. The Step Save Step Parameters to File is optional, and users can remove it if not needed.

4.4. Guide to Production Interface

This chapter will guide you through configuring and using the production interface.

Introduction

As shown in the figure below, to configure the production interface in Mech-Vision, which is done by the engineer, the operator can quickly deploy and stabilize production of vision solutions based on a simple and user-friendly information panel.



The virtual camera is now being used for production. Please proceed with caution.

Parameter name	Parameter value
Task ID	6
Capture Images from Camera_1 (Virtual M...	true
Capture Images from Camera_1 (Camera C...	WAM30214A3000752-EyeToHand-2021-05-...

Alert message 2023-10-18 17:34:37
[Translate Poses along Given Direction_1]: Translate Poses along Given Direction (1)

Input data is necessary for port 1!

The main functions and description of the production interface are shown in the following table:

No.	Main Function	Description
1	Quick Facts of Production Status	The production interface provides visual information panel to help people get quick knowledge of production status and project execution status, which can help Operator identify and judge production abnormities.
2	Quick View of Production Results	The production interface provides various running views, including Station live, Recognition result, Deep learning result, and Picking result. Operator can quickly view the production results.
3	Add New Workpiece and Switch Workpiece Types	The production interface provides various ways to create workpiece models for different types and shapes of workpieces, and set production recipes to add new workpieces and switch workpiece types.
4	Maintenance and Troubleshooting	The production interface provides the role-based access control to effectively reduce the risk of data leakage. It also provides the information of troubleshooting, production log, and alert records to help the operator quickly troubleshoot and resolve faults to achieve stable production.

Reading Guide

If you are an engineer, please read the following section to learn about how to configure the production interface.

Configure the Production Interface

If you are an operator, please read the following section to learn about how to use the production interface.

Use the Production Interface

4.4.1. Configure the Production Interface

This chapter is intended to guide the engineer through configuring the production interface. You can click the button next to the **Production Interface** in the toolbar to open the Production Interface Configurator.

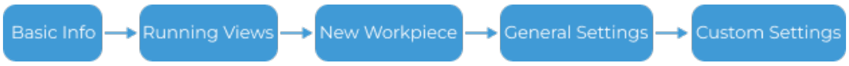
> The production interface cannot be configured if the solution is newly created and has not been saved to a directory.

▶ https://www.youtube.com/watch?v=ahV_Wk7iICw *(YouTube video)*

Video: Configure the Production Interface

Configuration Process

You should configure the content for display and define the functions you intend to use before applying the production interface. The overall configuration process is shown in the figure below.



1. Basic info: for configuring the solution name, production unit name, and unit function.
2. Running views: for selecting the associated project and specifying the content to be displayed in the running views.
3. New workpiece: for selecting the method to make the point cloud model for the new workpiece and setting the production recipe when there are new infeed workpieces.

   > The production recipe is the parameter recipe used in the dedicated project and is the important basis for switching and adding workpiece types.

4. General settings: for saving production data, setting up role-based access control, and creating the document.
5. Custom settings: for customizing alert rules, monitoring the disk space, etc.

Basic Info

Basic info contains the solution name, production unit name, and unit function. Please follow the steps below to configure.

1. Configure the solution name.

   The solution name will be displayed as the title name on the production interface. This setting will not change the name of the actual solution file (.msol).

2. Configure the production unit.

   a. Name the production unit.

      A production unit is the minimum running unit on the production interface. A maximum of 3 production units can be added for a production interface. Click the **[+]** button to add a production unit. Click **[-]** to delete the last added production unit.

   b. Select the Unit Function.

      The functions of the production unit are categorized into **Display running views** and **Display running views&Add new workpiece**.

      - Display running views: provide the functions of displaying running views and switching workpieces.

- Display running views&Add new workpiece: provide the functions of displaying running views, switching workpieces, and adding new workpieces. The “adding new workpieces” function requires the parameter recipes used in the project.

  If no parameter recipe is applied in any one of the projects in the solution, the **Display running views&Add new workpiece** function cannot be selected.

Click the **[Save]** button to save the configured content, and you can continue to configure the next time you open the Production Interface Configurator. The production interface cannot be started until all configurations have been completed in the Production Interface Configurator.

After naming the solution and production unit, click **[Next]** and you can start to configure the running views.

Running Views

In the Running views page, select a project to associate and configure the content to be displayed in the production unit. Please follow the steps below to configure.

1. Select the project to associate.

   The associated project determines which views of the production unit can be displayed and the ways to switch or add workpiece types. It is recommended to select the main recognition project under the solution.

   The associated project is related to the selected **Unit function**. If the unit function is **Display running views&Add new workpiece**, only projects containing a parameter recipe(s) can be associated.

2. Set the content for display in the running views.

   a. Select running views.

      The running views are used to display the production status. This visualization area can display four different running views for various purposes: Station live, Recognition result, Deep learning result, and Picking result.

      The “Station live” is the default option. You can click the **[+]** button to add other running views. Click **[-]** to delete the last added running view.

      - Only one running view for each type can be added, i.e., a maximum of four running views can be added.
      - At least one running view should be kept.

   b. Select the Step.

      After selecting the running view(s), you should select the Step corresponding to each view and click the **[Preview]** button to view the content. The Steps supporting visualization in each running view are listed in the table below.

      Please select the running views related to the project only.

      For example, if there is not any deep learning Step in the project, and a “Deep learning result” running view is added, no content can be displayed in this running view, and you cannot proceed with the configuration.

Running view option	Added by default	Setting procedures	Selectable Step
Station live	Yes	① Select method to capture images; ②Select a Step as the source of the view.	Use multiple cameras and stitch captured images: Merge Point Clouds.
			Use one camera to capture images: Camera 2D, Capture Images from Camera, Read Images V2.
Recognition result	No	Select a Step as the source of the view.	3D Workpiece Recognition, 3D Fine Matching, 3D Fine Matching Lite, 3D Fine Matching (Multiple Models), 3D Matching and Classification (Multiple Models).
Deep learning result	No	Select a Step as the source of the view.	Deep Learning Model Package Inference, Deep Learning Result Parser.
Picking result	No	① Set the scene point cloud; ②Select a Step as the source of the view.	The scene point cloud in the robot reference frame will be used by default. All Steps with an output port in the “PoseList” format can be selected.

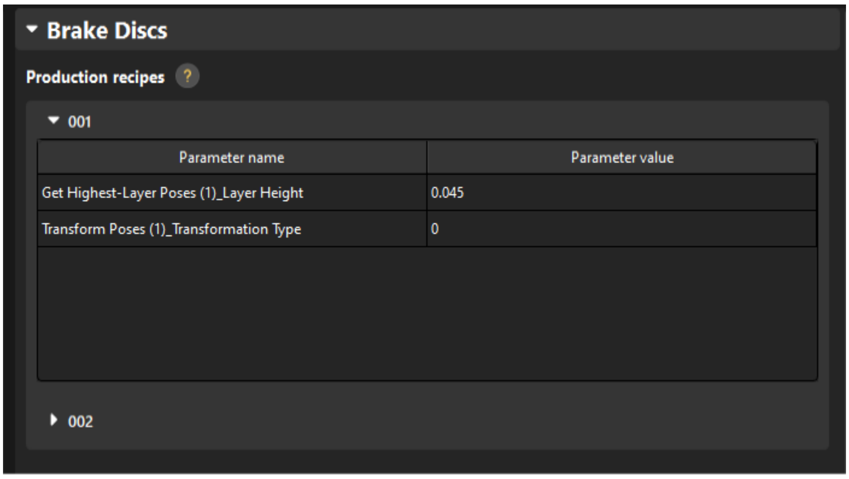
Now you have finished configuring the running views. Click **[Next]** to add new workpieces.

New Workpiece

Different unit functions (**Display running views** and **Display running views&Add new workpiece**) set in the Basic info page will lead to different content in the New workpiece page.

Display Running Views

If the unit function is set to **Display running views**, only Production recipes will be displayed in this page for basic workpiece switching, as shown in the figure below.



Brake Discs

Production recipes

001

Parameter name	Parameter value
Get Highest-Layer Poses (1)_Layer Height	0.045
Transform Poses (1)_Transformation Type	0

002

Display Running Views&Add New Workpiece

If the unit function is set to **Display running views&Add new workpiece**, Production recipes displayed in this page are used for basic workpiece switching. You can also set the method to add new workpieces.

1. Select the method to make the point cloud model of the workpiece.

   Three methods to make the point cloud model for the new type of workpieces are listed in the table below:

No.	Method to make the point cloud model	Description	Further operation
1	Model editor	Use the Matching Model and Pick Point Editor to make the point cloud model.	None.
2	Model-making project	Use a separate vision project to generate the point cloud model file.	You should select a vision project that can generate point cloud model of the workpiece.
3	Geometric model	Create geometric models to be used as point cloud models.	Select the geometric model corresponding to the workpiece.

2. Set the production recipe.

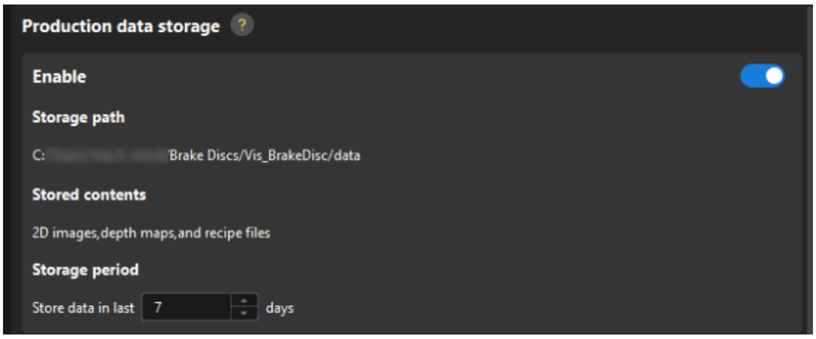
   Set the way to enter the production recipe parameters. The following ways are supported:

   - User input: You can manually enter the values of the recipe parameters for adding a new type of workpiece. The value is a variable.
   - System default: No user input is required and the default value is used. The value is a constant.

Now you have finished configuring the settings related to new workpieces. Click **[Next]** to proceed with the general settings.

General Settings

1. Save production data.



Production data storage ?

Enable

Storage path

C:[redacted]Brake Discs/Vis_BrakeDisc/data

Stored contents

2D images,depth maps,and recipe files

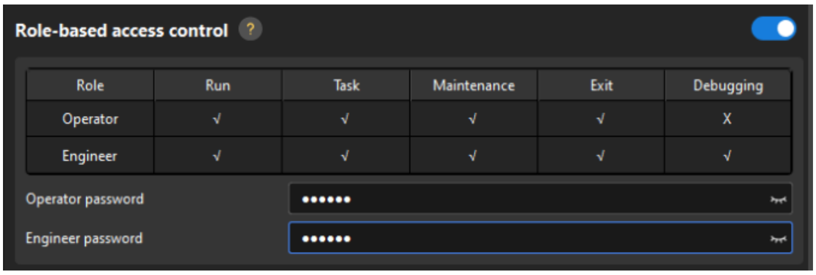
Storage period

Store data in last 7 days

   To save the production data, enable the **Production data storage** feature, set a **Storage period**, and the production data will be saved to the specified path.

- Production data will be automatically saved to the same storage path as the Data Storage feature in the Project Assistant.
- Only image data captured by the “Capture Images from Camera” Step with the “Camera Type” set to “Mech-Eye” can be saved. Currently, image data captured by cameras of other types are not supported.

2. Role-based access control.



Role-based access control ?

Role	Run	Task	Maintenance	Exit	Debugging
Operator	√	√	√	√	X
Engineer	√	√	√	√	√

Operator password ••••••

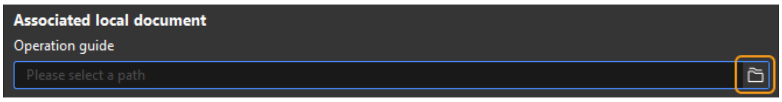
Engineer password ••••••

When this function is not enabled, all operations on the production interface are permitted for all users. After it is enabled, users can have permission to perform corresponding operations after entering the correct password. The operations available for each feature are described below.

- Run: Operation to run the project. You can manually trigger the project to run once to view the production result.
- Task: Production management operations, including adding workpiece types and switching workpiece types.
- Maintenance: Basic maintenance operations, such as file backup and troubleshooting.
- Exit: Production stop operation. You will exit the production interface and close the software.
- Debugging: Project debugging operations. You will enter the Standard mode to modify the project.

3. Document generator.

   a. Click the folder icon under “Associated local document” to select the PDF file. Then you can open the operation guide by clicking the Document icon directly on the production interface.



Associated local document

Operation guide

Please select a path

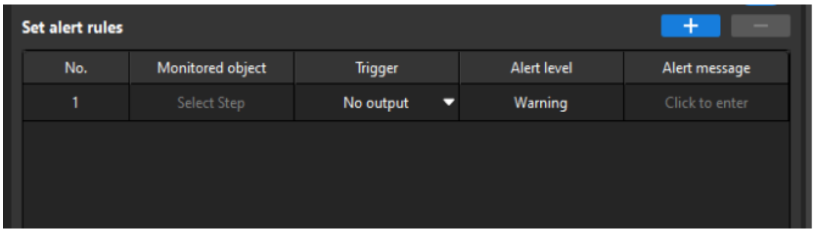
Now you have finished configuring the general settings. Click **[Next]** to proceed with the custom settings.

Custom Settings

The custom settings include **Custom alert**, **Disk space monitoring**, etc. Please follow the steps below to configure.

1. Custom alert

   Create custom alert rules for the project. When the conditions of an alert rule are met, custom alert messages will appear in the alert area, but the project will not be terminated. Click the **[+]** or **[-]** button to add or delete an alert rule.



Set alert rules

No.	Monitored object	Trigger	Alert level	Alert message
1	Select Step	No output	Warning	Click to enter

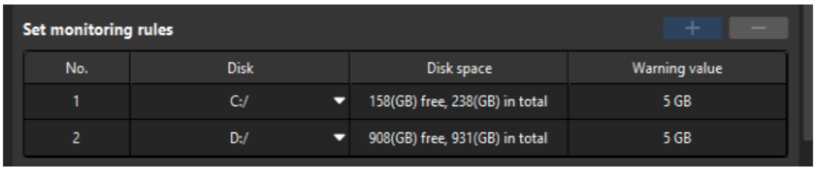
Common usage cases:

- Engineers should create custom alerts for key Steps. These custom alerts can help operators quickly locate the problem when any abnormality occurs in the Steps.
- Engineers can also create custom alert messages for special processes (such as error-prevention Steps) to help operators analyze and locate the causes of the production interruption.

> If the engineer does not create custom alerts for all key Steps in the project, no alerts will be reported when any abnormality occurs in these Steps, which hinders operators in efficiently troubleshooting issues.

2. Disk space monitoring

Add rules to monitor the free space of disks. Insufficient free disk space will affect the normal running of the system.



Set monitoring rules

No.	Disk	Disk space	Warning value
1	C:/	158(GB) free, 238(GB) in total	5 GB
2	D:/	908(GB) free, 931(GB) in total	5 GB

Click [+] or [-] to add or delete a monitoring rule.

> - Only the system disk(s) will be monitored by default.
> - Up to two disks can be monitored.

3. Brand display

Display the brand and product information you select.

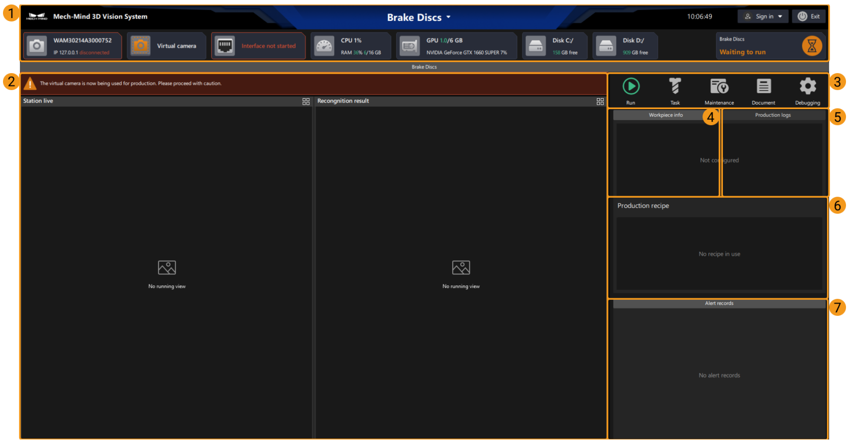
> The Mech-Mind brand logo will be displayed by default.

After completing the Custom settings, you have finished all configurations of the production interface. Then you can use the production interface in real production.

4.4.2. Use the Production Interface

This chapter will guide you through using the production interface.

After the engineer finishes the configuration of the production interface, the operator can start to use it. You can refer to the following for the introduction and instruction of the production interface.



The virtual camera is now being used for production. Please proceed with caution.

The following table describes the functions of each area in the figure above.

No.	Area	Function	Instruction
1	Status area	Displays the brand name, solution name, camera connection status, communication status, and system monitoring status.	• Enter the Production Interface • Sign In / Sign Out • Reset Password • Check the Production Status • View the Running Status of the Project • Exit the Production Interface
2	View area	Displays different types of running views, i.e., Station live, Recognition result, Deep learning result, and Picking result.	• View the Production Results
3	Operation area	For running the project, adding new workpieces, switching the workpiece types, maintaining the solution, and referring to the operation guide.	• Switch Workpiece Types • Add New Workpiece • Backup Files Management • Troubleshooting • View the Operation Guide • Return to Standard Mode
4	Workpiece info area	Displays the information of the workpiece model.	• Display Workpiece Information
5	Log area	Displays Mech-Vision log, Console log, and Operation log.	• Check Production Logs
6	Production recipe area	Displays the recipe in use.	
7	Alert area	Displays the alert messages when running the project.	• Check Alert Records

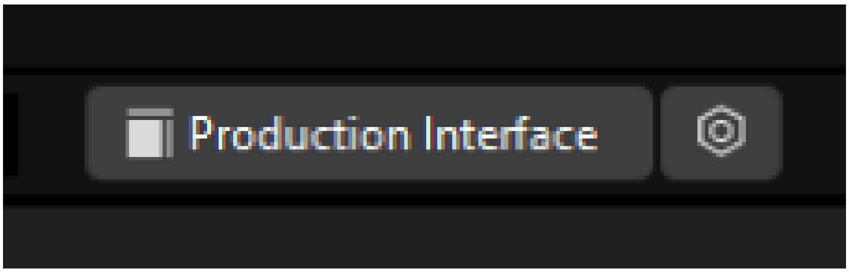
- When using the production interface, the resolution and scaling of the screen should follow the correspondence in the table below. If the resolution and scaling of your screen do not match the those in the table below, some display problems may occur.

Resolution	Scaling
1920×1080 (16:9)	100%
2560×1440 (16:9)	125%
3840×2160 (16:9)	150%, 175%

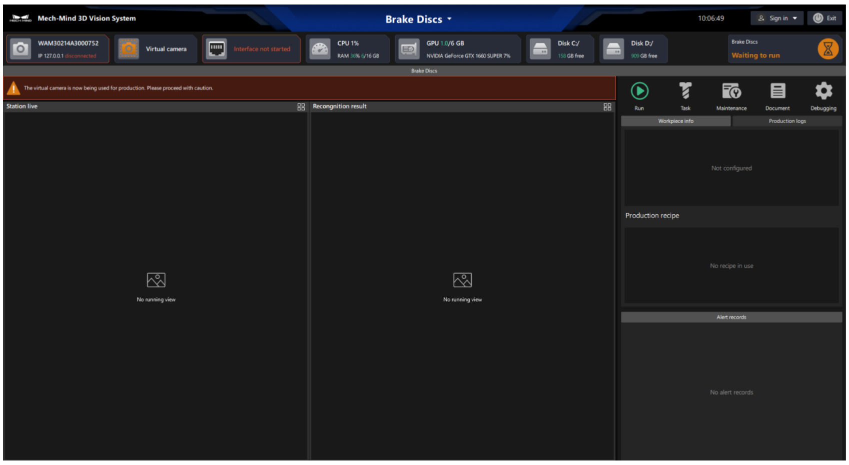
- Dual monitors are not supported at present.

4.4.2.1. Enter the Production Interface

After completing the settings in Production Interface Configurator, click the **[Production Interface]** button on the Mech-Vision toolbar to access the production interface.



The production interface is shown in the figure below.



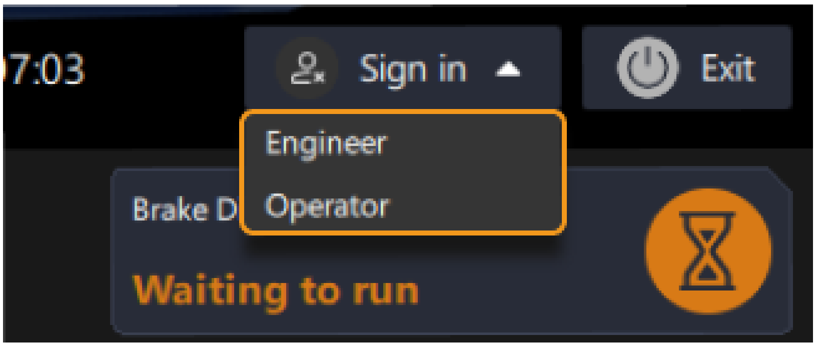
The virtual camera is now being used for production. Please proceed with caution.

4.4.2.2. Sign In / Sign Out

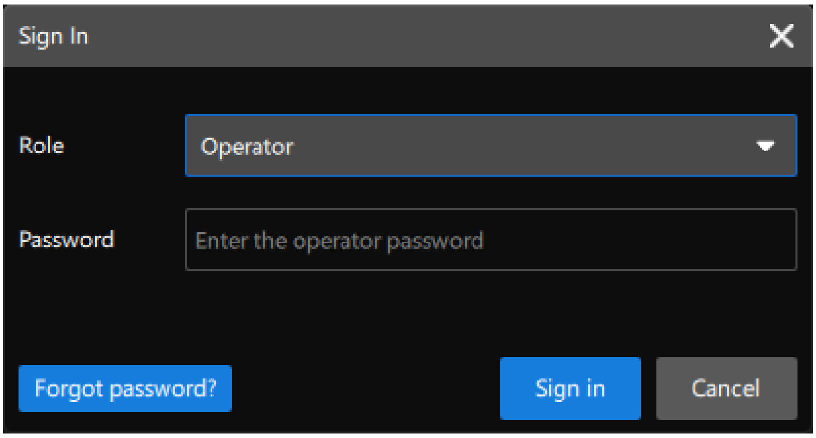
After the **Role-based access control** in General settings is enabled in the Production Interface Configurator, you can follow the steps below to sign in or sign out the account.

Sign In

1. Click the **[Sign in]** button in the upper-right corner of the production interface, and then select the account in the drop-down menu.



2. Enter the password in the pop-up window and then click **[Sign in]**. After successfully logging in, the current account avatar will be displayed and the relevant function buttons will be activated.


Sign In

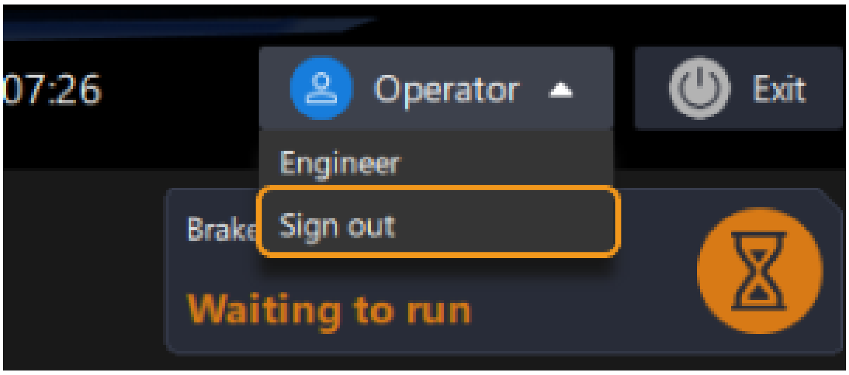
Role: Operator

Password: Enter the operator password

If you forget your password, you can refer to Reset Password.

Sign Out

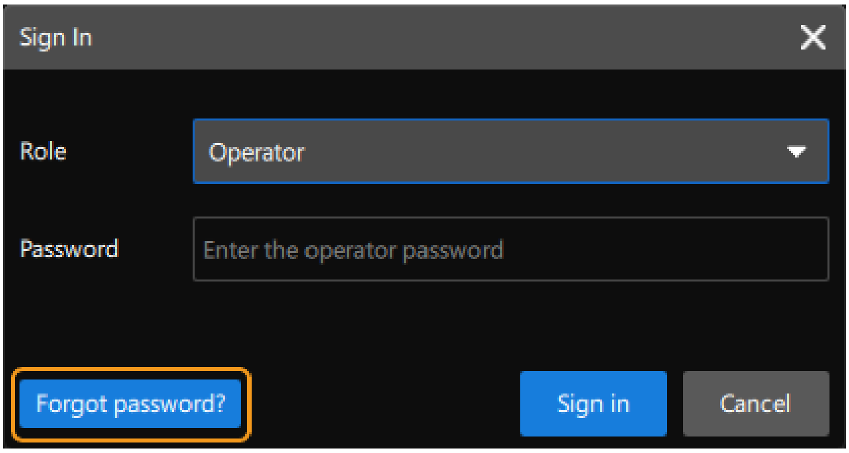
To sign out, you can click the account avatar in the upper-right corner of the production interface, and then select **Sign out** in the drop-down menu.



4.4.2.3. Reset Password

If engineers or operators forget the password when signing in, they can reset the password as follows.

Click **[Forgot password?]** in the Sign In window.

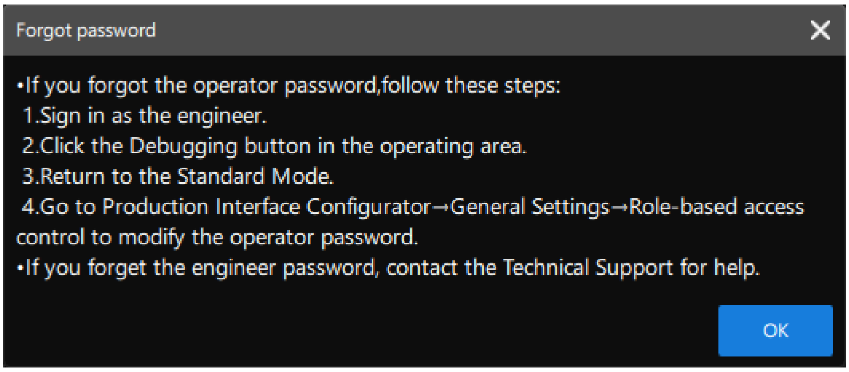


Sign In

Role: Operator

Password: Enter the operator password

Then follow the prompts to reset the engineer or operator password.

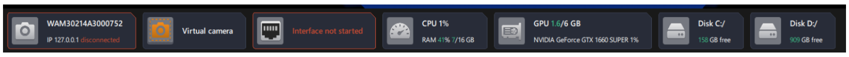


Forgot password

•If you forgot the operator password,follow these steps:
1.Sign in as the engineer.
2.Click the Debugging button in the operating area.
3.Return to the Standard Mode.
4.Go to Production Interface Configurator→General Settings→Role-based access control to modify the operator password.
•If you forget the engineer password, contact the Technical Support for help.

4.4.2.4. Check the Production Status

While using the production interface, you can go to the status area at the top to check the status of all aspects, which is shown below.



WAM30214A3000752
IP 127.0.0.1 disconnected

Virtual camera

Interface not started

CPU 1%
RAM 41% 7/16 GB

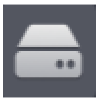
GPU 1.6/6 GB
NVIDIA GeForce GTX 1660 SUPER 1%

Disk C:/
158 GB free

Disk D:/
909 GB free

The specific description of production status is shown in the following table.

Icon	Description
📷	Camera connection
🖧	Communication connection
⏲	CPU and memory (the percentage after “CPU” represents CPU utilization)
🖥	GPU (the percentage represents GPU utilization)

Icon	Description
	Hard disk

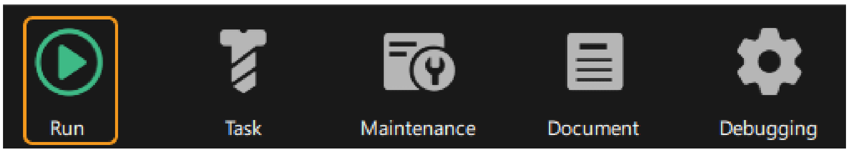
- The communication connection status is updated every 30 seconds.
- The statuses of CPU and memory, GPU, and hard disk are updated every 5 seconds.

4.4.2.5. View the Production Results

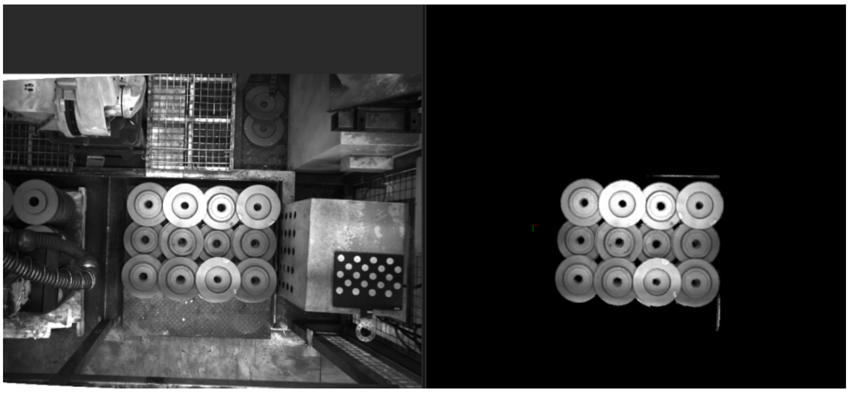
In the production process, the operator can check the production results via the running views and can adjust the display mode of the running views as well.

View the Production Results via Running Views

After entering the production interface, you can click the ▶ button in the operation area to run the project for a single time, thus viewing the production results.



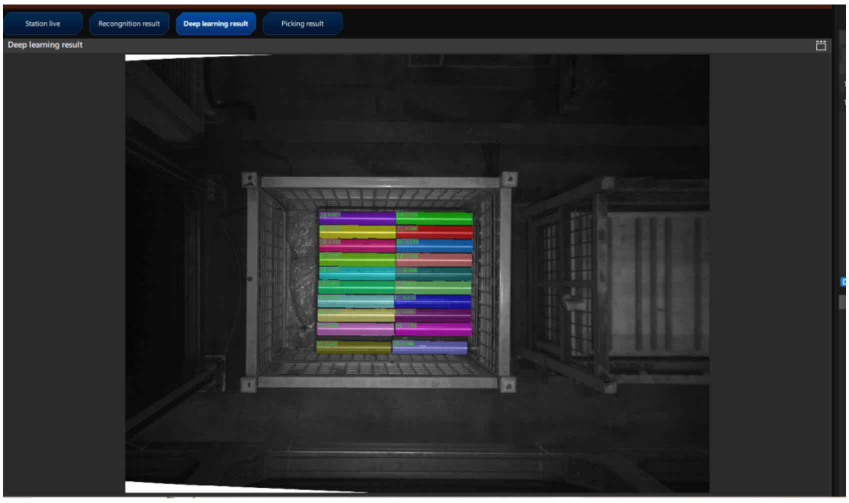
Then you can check the production results in the default grid view as shown below.



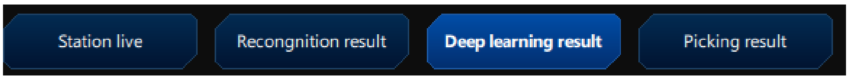
- If there is no operation for 30 minutes, the running view will automatically switch from the grid view to the full-screen view, displaying the Station live of the workstation.
- The point cloud displayed in the “Picking result” view has been downsampled, with a sampling interval of 5 mm.
- The number of displayed running views is determined by the number of running views added in Production Interface Configurator.

Switch the Display Mode of Running View

The above figure shows the grid view. To display one of the running views in full screen (as shown in the figure below), click the ⊞ button in the upper right corner.

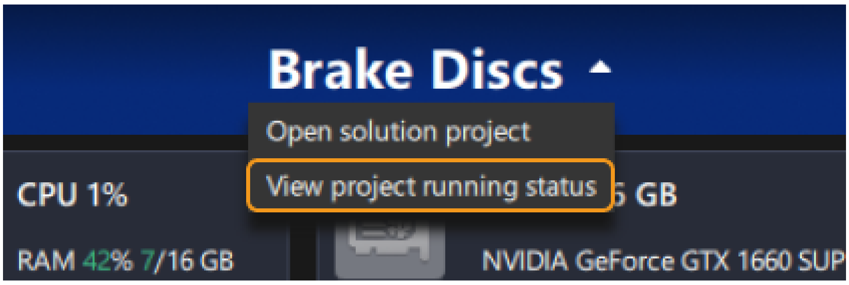


In the full screen mode, you can select the name of the running view to switch between different views.

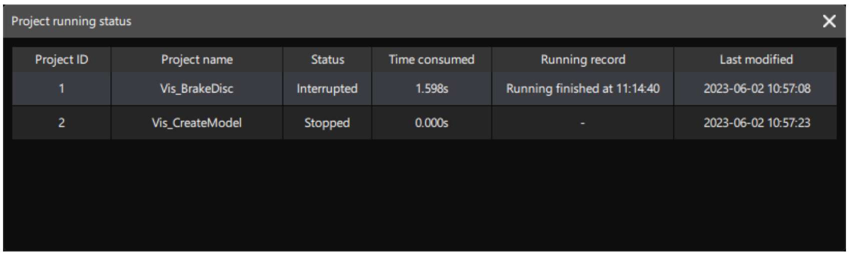


4.4.2.6. View the Running Status of the Project

To view the running status of the project, you can click the ▼ button on the right side of the solution name which is at the top of the production interface, and then select **View project running status** in the drop-down menu.



The Project running status window is shown in the following figure, where you can check the Status, Time consumed, Running record and Last modified time of the project.



Project running status

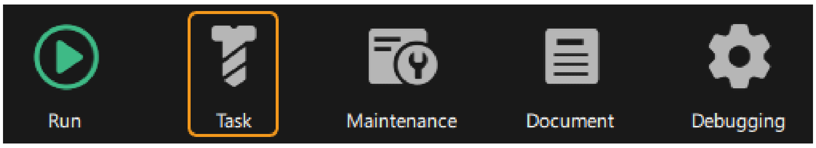
Project ID	Project name	Status	Time consumed	Running record	Last modified
1	Vis_BrakeDisc	Interrupted	1.598s	Running finished at 11:14:40	2023-06-02 10:57:08
2	Vis_CreateModel	Stopped	0.000s	-	2023-06-02 10:57:23

4.4.2.7. Switch Workpiece Types

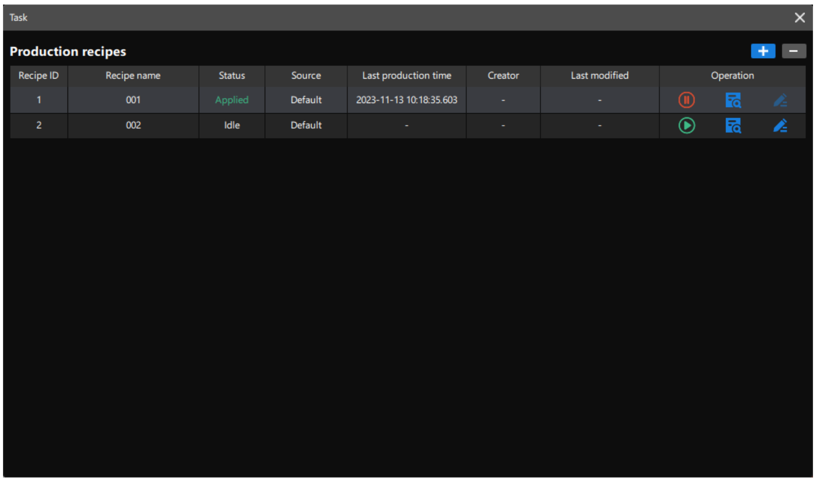
For example, when the infeed workpiece changes from workpiece A to workpiece B and a production recipe for the recognition of workpiece B already exists, you can switch the recipe on the Production Interface for the recognition of workpiece B. Please follow the steps below to operate.

1. Open the Production recipes.

   Click the button in the operation area.



   The Production recipes window is shown below.



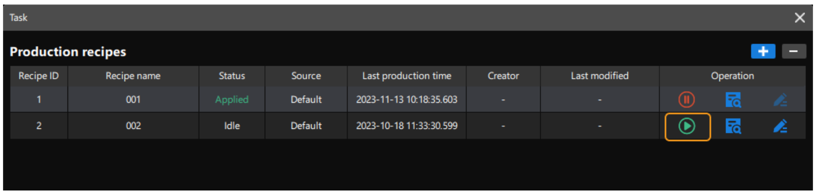
Production recipes

Recipe ID	Recipe name	Status	Source	Last production time	Creator	Last modified	Operation
1	001	Applied	Default	2023-11-13 10:18:35.603	-	-	
2	002	Idle	Default	-	-	-	

2. Switch the production recipe.

   As shown in the figure above, there are two production recipes in the Production recipes. According to the status, Production Recipe 1 is applied in the current production, while Production Recipe 2 is idle.

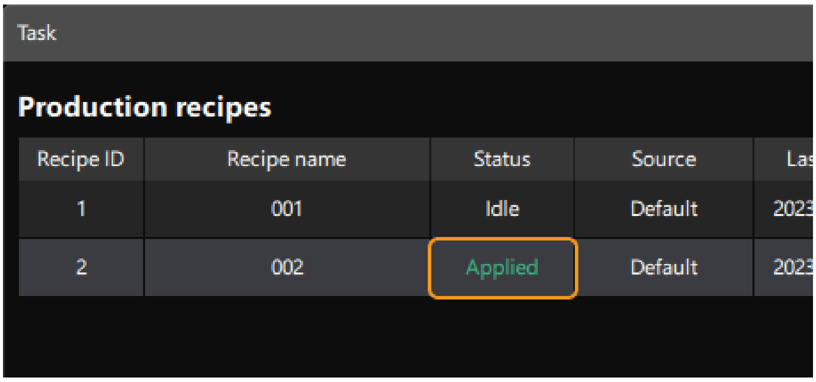
   You can click the button on the right side of Production Recipe 2 to switch the current production recipe.



Production recipes

Recipe ID	Recipe name	Status	Source	Last production time	Creator	Last modified	Operation
1	001	Applied	Default	2023-11-13 10:18:35.603	-	-	
2	002	Idle	Default	2023-10-18 11:33:30.599	-	-	

   When the status of new production recipe is **Applied**, it means that the production recipe has been switched successfully. After switching the production recipe, the new one will be displayed in the Production recipe area.



Task

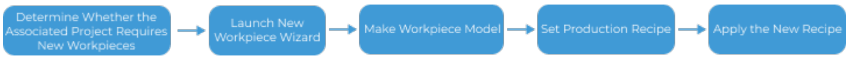
Production recipes

Recipe ID	Recipe name	Status	Source	Las
1	001	Idle	Default	2023
2	002	Applied	Default	2023

- If you want to stop applying the current production recipe, click ⏸ on the right of the production recipe.
- To check the parameters in the production recipes, you can click the 🔍 button on the right side of the production recipe, and check them in the pop-up window.

4.4.2.8. Add New Workpiece

For example, when the infeed workpiece changes from workpiece A to workpiece B and there is not an available production recipe for the new workpiece, you can make a point cloud model of the new workpiece on the production interface and add a new production recipe for the recognition of workpiece B. Please follow the steps below to operate.



Determine Whether the Associated Project Requires New Workpieces → Launch New Workpiece Wizard → Make Workpiece Model → Set Production Recipe → Apply the New Recipe

1. Determine whether new workpieces will be added in the current associated project, and confirm that the Associated project selected in Production Interface Configurator is the correct one.
2. Open the Production Interface Configurator and click New workpiece. Then make a new workpiece model and set the production recipe.
3. Make the point cloud model for the new workpiece.
4. Set the production recipe for the new workpiece.
5. Apply the new production recipe to the project for production.

Determine Whether the Associated Project Requires New Workpieces

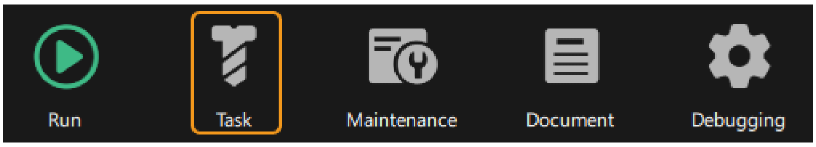
Please check whether the current associated project requires new workpieces first. If no new workpieces will be added to the current associated project, please contact the engineer to modify the associated project in the Production Interface Configurator.

Launch New Workpiece Wizard

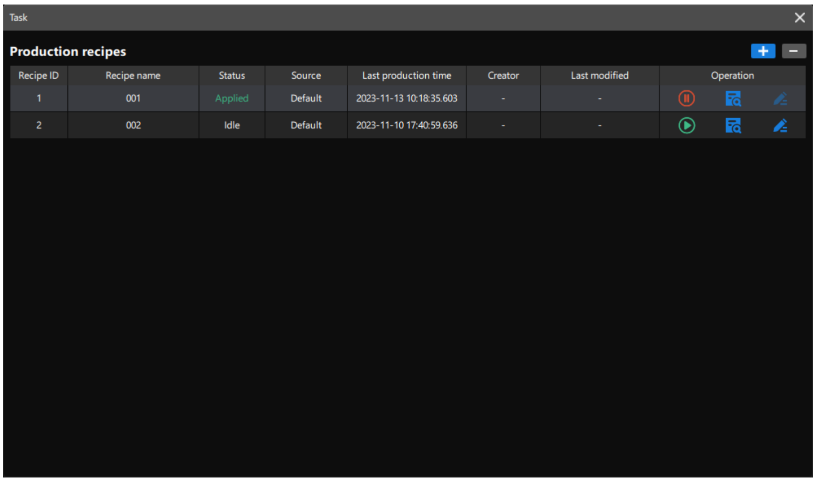
The New Workpiece Wizard can be opened in the Production recipe window. The detailed steps are as follow.

1. Open the Production recipe window.

   Click the button in the operation area.



The production recipe window is shown below.

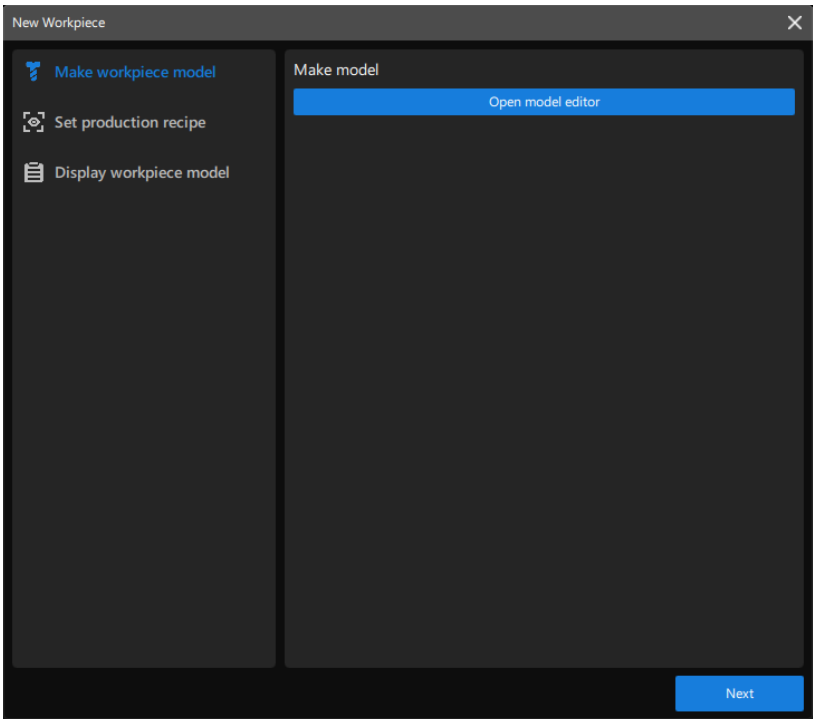

Task

Production recipes

Recipe ID	Recipe name	Status	Source	Last production time	Creator	Last modified	Operation
1	001	Applied	Default	2023-11-13 10:18:35.603	-	-	⏸
2	002	Idle	Default	2023-11-10 17:40:59.636	-	-	▶

2. Launch the New Workpiece wizard.

   Click the [+] button in the upper right corner to launch the New Workpiece wizard. The interface of the New Workpiece wizard is shown below.



As shown in the figure above, the New workpiece configuration can mainly divided into two processes, i.e., **make workpiece model** and **set production recipe**.

Make Workpiece Model

When new workpieces are added, models for the new workpieces should be make. The method for making workpiece models is specified in the Production Interface Configurator. The three methods are described as follows.

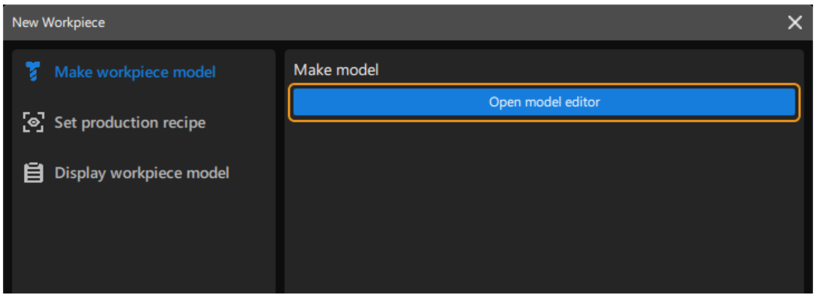
> The workpiece model contains the point cloud model and pick points.

Use Matching Model and Pick Point Editor

When the method for making workpiece model is set to **Model editor** in the Production Interface Configurator, you should use the Matching Model and Pick Point Editor to make the point cloud model for the new workpiece. Please follow the steps below to operate.

1. Open the Model Editor.

   Click **[Open model editor]** in the New Workpiece wizard to open the tool.

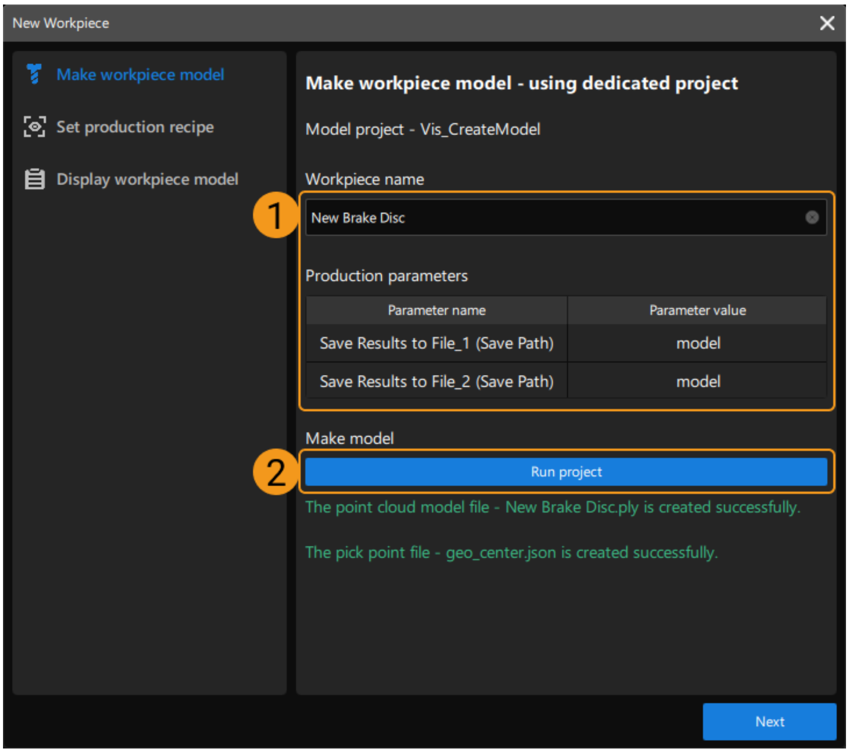


2. Make the point cloud model.

   For detailed instructions on how to use the tool to make the point cloud model for the workpiece, please refer to Generate Point Cloud Model.

Use Dedicated Project to Make Workpiece Model

When the method for making workpiece model is set to **Model-making project** in the Production Interface Configurator, you should name the new workpiece and set the production parameters in the New Workpiece wizard, and then click the **[Run project]** button. A message **“The point cloud model file - XXX.ply is created successfully”** will appear if the model is created successfully.



Make workpiece model - using dedicated project

Model project - Vis_CreateModel

Workpiece name

New Brake Disc

Production parameters

Parameter name	Parameter value
Save Results to File_1 (Save Path)	model
Save Results to File_2 (Save Path)	model

Make model

Run project

The point cloud model file - New Brake Disc.ply is created successfully.

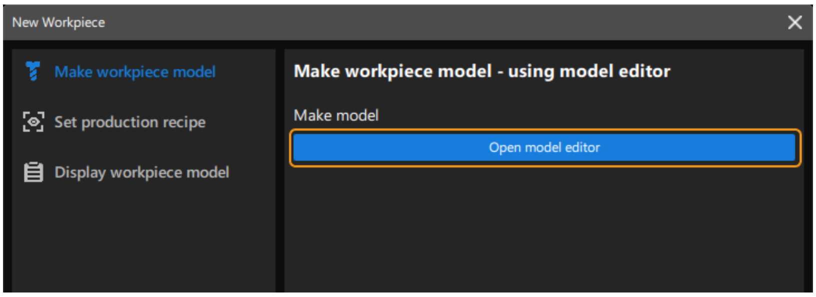
The pick point file - geo_center.json is created successfully.

Use Geometric Model

When the method for making workpiece model is set to **Geometric model** in the Production Interface Configurator, you should use the **Create Geometric Model** tool to make the point cloud model for the new workpiece. Please follow the steps below to operate.

1. Open the Model Editor.

   Click **[Open model editor]** in the New Workpiece wizard, and the **Create Geometric Model** window will pop up.



Make workpiece model - using model editor

Make model

2. Use a geometric model as the point cloud model.

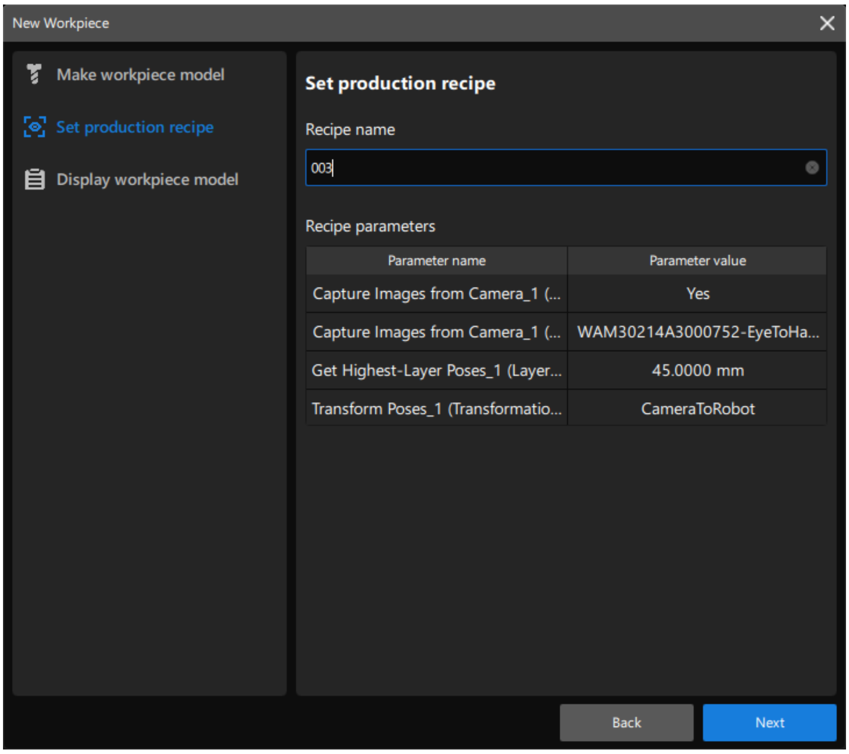
   For detailed instructions on how to use the tool to make the geometric model for the workpiece, please refer to Create Geometric Model as Point Cloud Model.

Set Production Recipe

After making the point cloud model for the new workpiece, click **[Next]** to set the production recipe.

As shown in the figure below, enter the **Recipe name** and set the **Recipe parameters**, and then click

[Next].



New Workpiece

- Make workpiece model
- Set production recipe
- Display workpiece model

Set production recipe

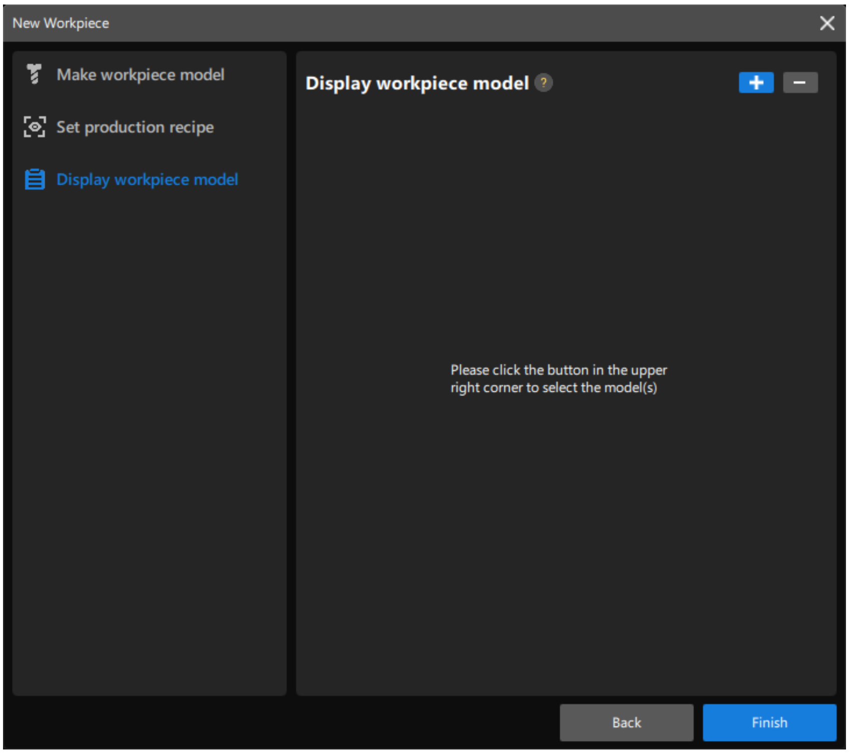
Recipe name: 003

Recipe parameters

Parameter name	Parameter value
Capture Images from Camera_1 (...	Yes
Capture Images from Camera_1 (...	WAM30214A3000752-EyeToHa...
Get Highest-Layer Poses_1 (Layer...	45.0000 mm
Transform Poses_1 (Transformatio...	CameraToRobot

Display Workpiece Model

After setting the production parameter, if you want to display the workpiece model in the Workpiece info area on the production interface, click the [+] button in the upper right corner in the “Display workpiece model” window.



New Workpiece

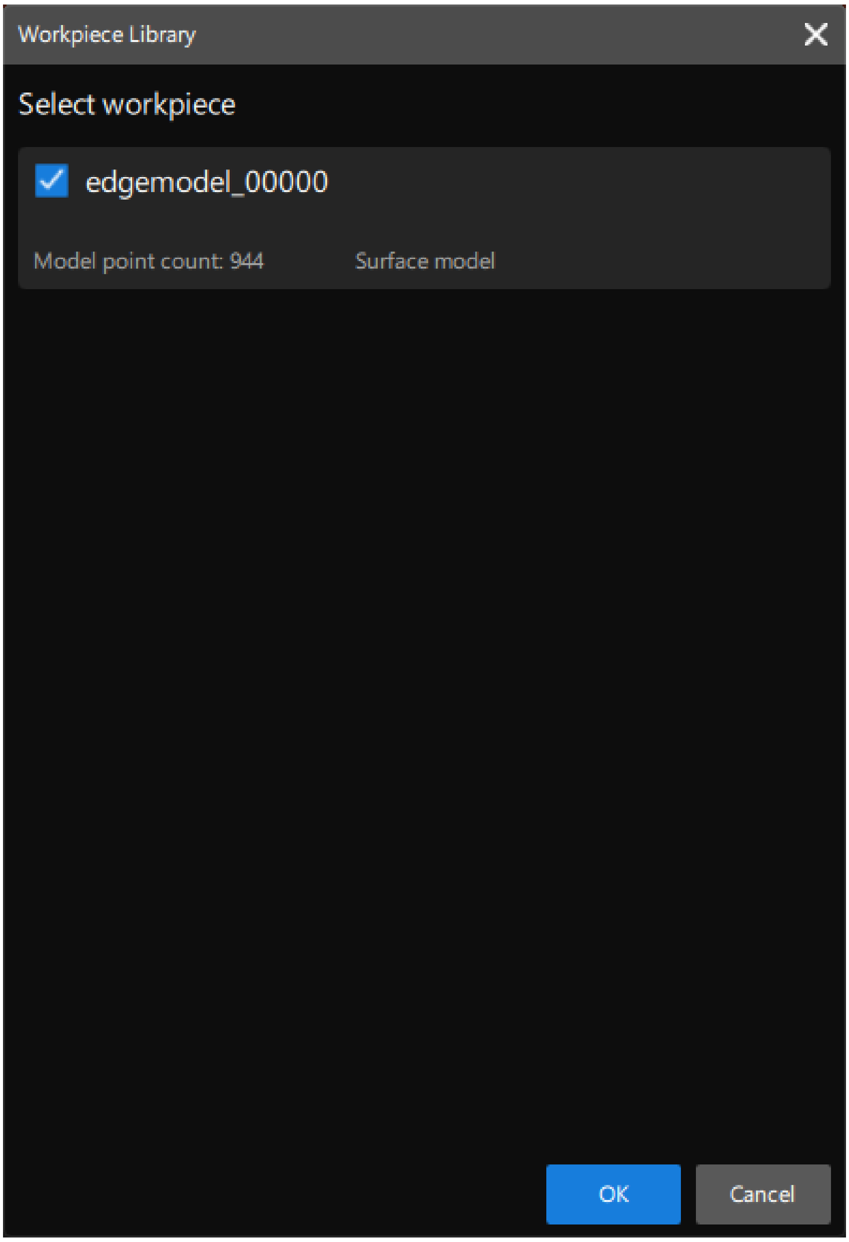
- Make workpiece model
- Set production recipe
- Display workpiece model

Display workpiece model

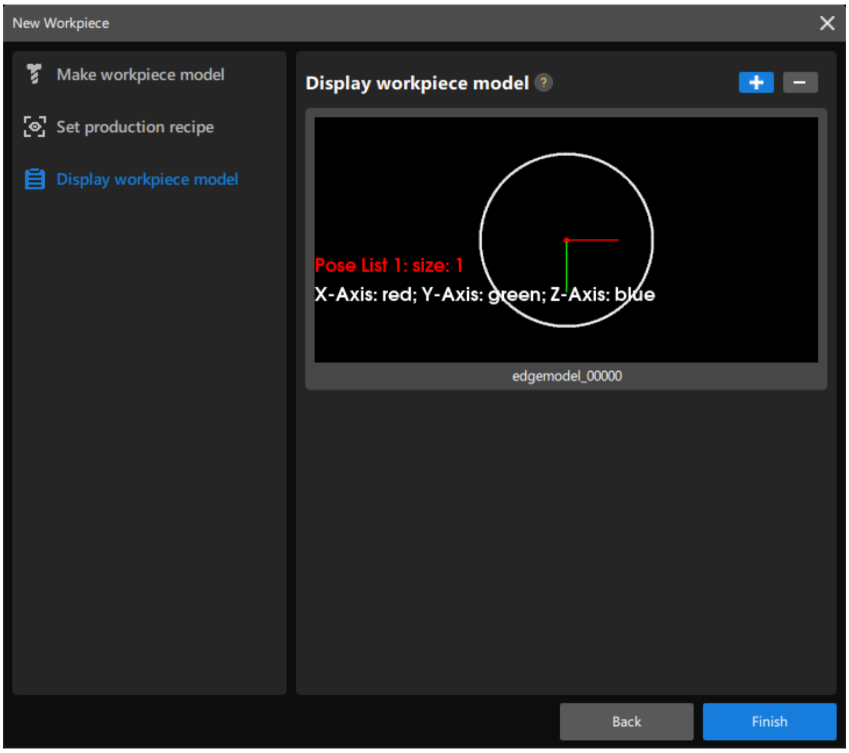
Please click the button in the upper right corner to select the model(s)

Up to two workpiece models can be added for display.

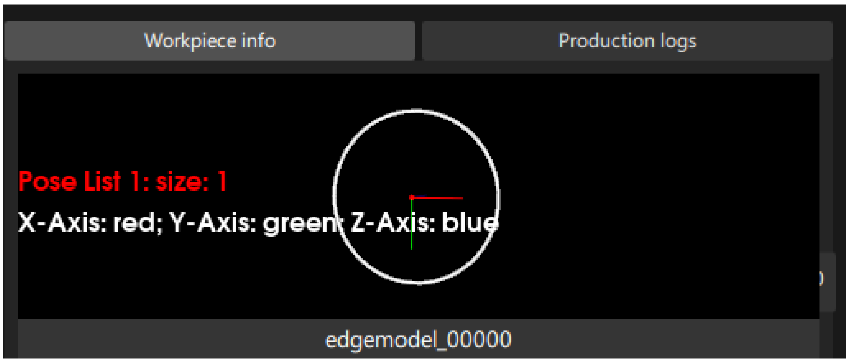
Select the workpiece model in the pop-up window, and click [OK].



After adding the workpiece model for display, click [Finish].



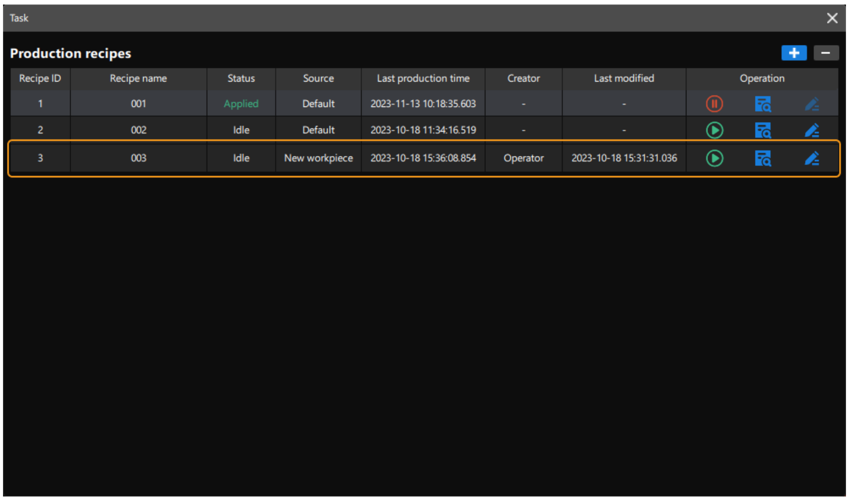
Now, the workpiece point cloud model you made will be displayed in the Workpiece info area on the production interface.



If the point cloud model you made is not stored in the 3d_matching folder under `project folder\resource`, the point cloud model cannot be displayed in the Workpiece info area.

Apply the New Recipe

The production recipe configured in the New Workpiece wizard will be added to the Production recipe list, as shown below.

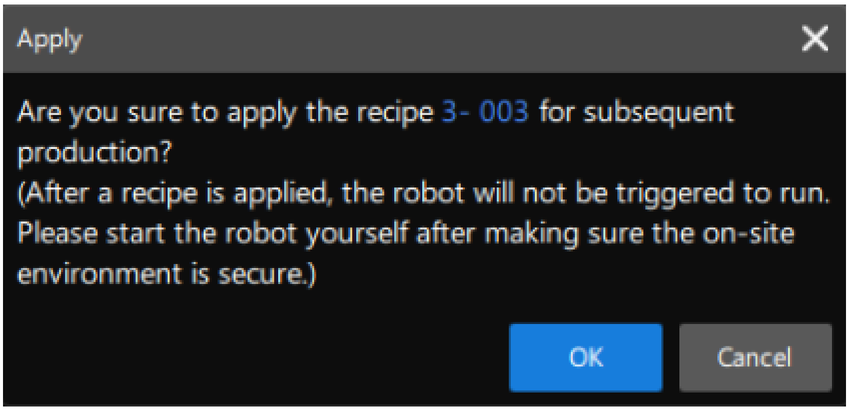


Task

Production recipes

Recipe ID	Recipe name	Status	Source	Last production time	Creator	Last modified	Operation
1	001	Applied	Default	2023-11-13 10:18:35.603	-	-	⏸ 🔍 ✎
2	002	Idle	Default	2023-10-18 11:34:16.519	-	-	▶ 🔍 ✎
3	003	Idle	New workpiece	2023-10-18 15:36:08.854	Operator	2023-10-18 15:31:31.036	▶ 🔍 ✎

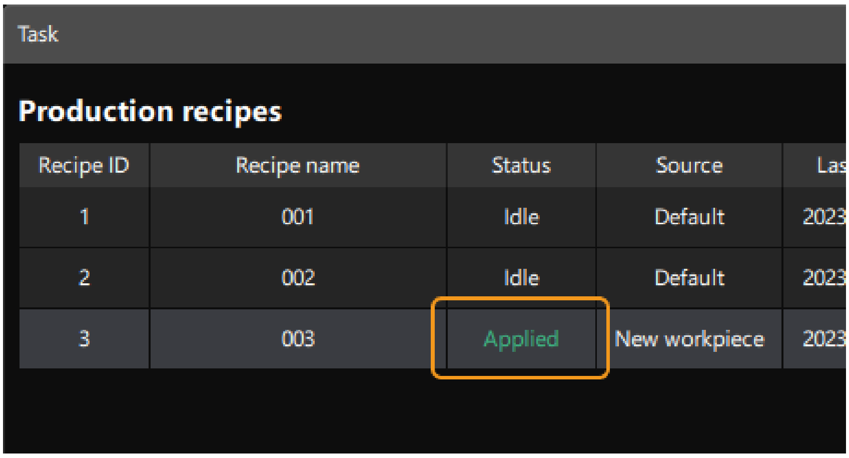
Click the ▶ button next to the new recipe to apply it to the project. If the on-site safety can be guaranteed, click [OK] in the pop-up window.



Apply

Are you sure to apply the recipe 3- 003 for subsequent production?
(After a recipe is applied, the robot will not be triggered to run. Please start the robot yourself after making sure the on-site environment is secure.)

If **Applied** appears in the Status column, the production recipe for the new workpiece is applied successfully.



Task

Production recipes

Recipe ID	Recipe name	Status	Source	Las
1	001	Idle	Default	2023
2	002	Idle	Default	2023
3	003	Applied	New workpiece	2023

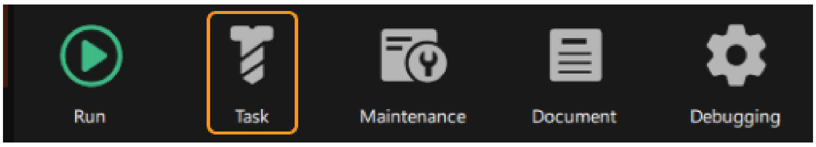
4.4.2.9. Display Workpiece Information

The workpiece models can be displayed in the workpiece info area.

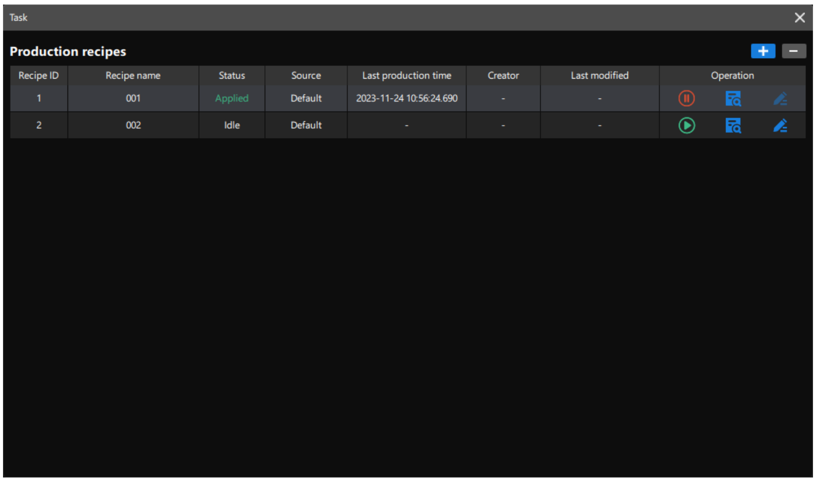
After entering the production interface, if the production recipe contains workpiece models, the workpiece info will be displayed. If the production recipe does not contain workpiece models, but you want to display the workpiece information, you need to modify the production recipe.

1. Open the Production recipes.

   Click the button in the operation area.



   The Production recipes window is shown below.

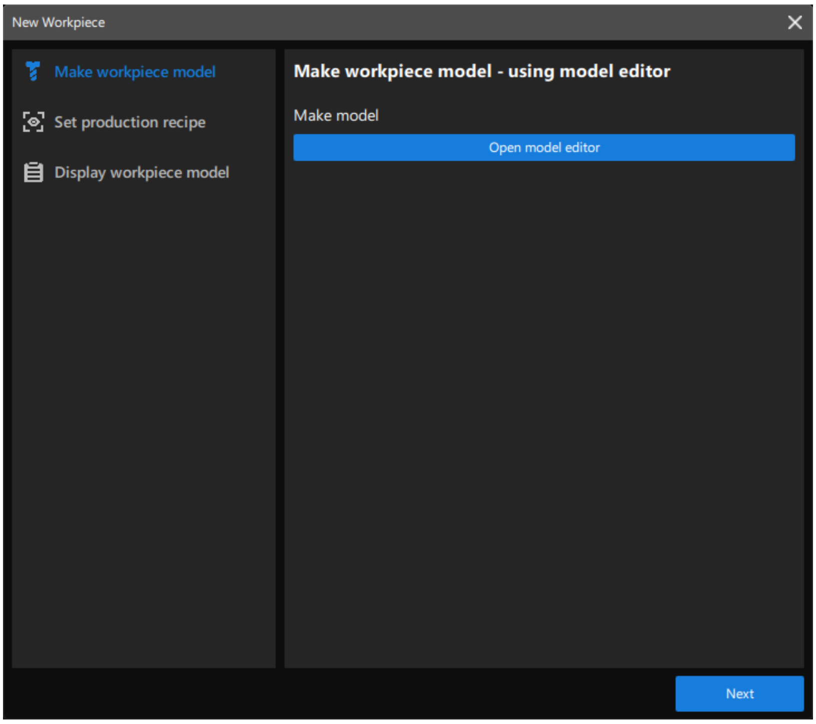


Production recipes

Recipe ID	Recipe name	Status	Source	Last production time	Creator	Last modified	Operation
1	001	Applied	Default	2023-11-24 10:56:24.690	-	-	
2	002	Idle	Default	-	-	-	

2. Modify the production recipe.

   Click button on the right side of production recipes to modify the production recipes.



Make workpiece model - using model editor

Make model

The production recipe can be modified when it is in “Idle” status.

3. Make workpiece model.

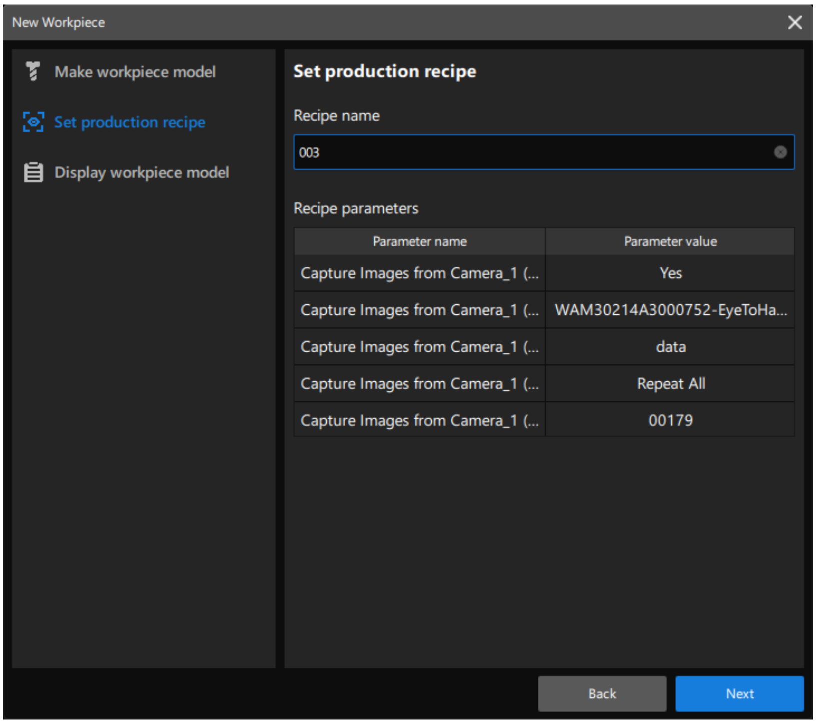
   If there is point cloud model in the project, you can directly click **[Next]**.

   If there is not point cloud model in the project, you need to make models for the workpieces. You can refer to Make Workpiece Model.

4. Set production recipe.

   After making the point cloud model for the new workpiece, click **[Next]** to set the production recipe.

   As shown in the figure below, enter the **Recipe name**, and then click **[Next]**.



Set production recipe

Recipe name

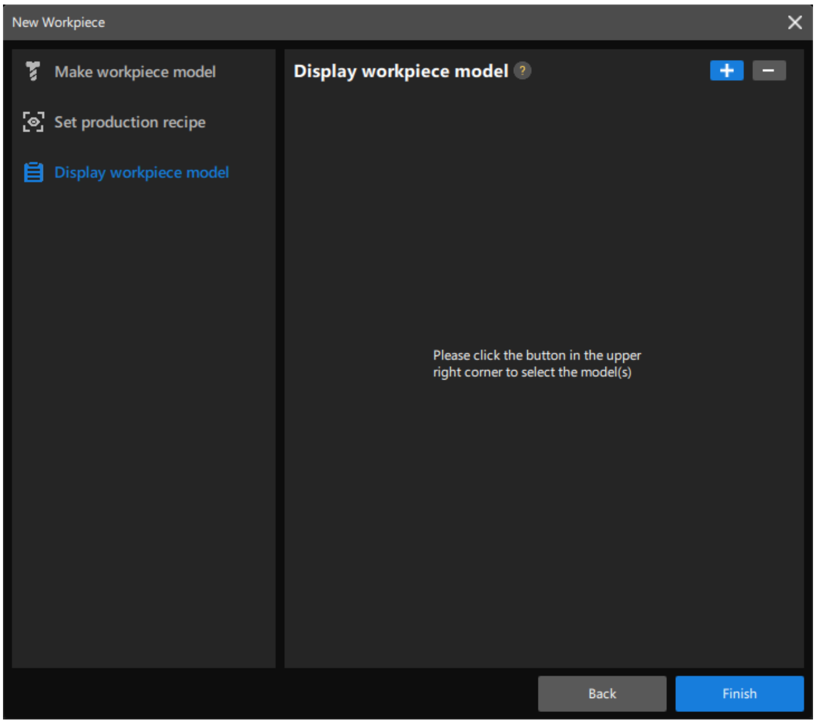
003

Recipe parameters

Parameter name	Parameter value
Capture Images from Camera_1 (...	Yes
Capture Images from Camera_1 (...	WAM30214A3000752-EyeToHa...
Capture Images from Camera_1 (...	data
Capture Images from Camera_1 (...	Repeat All
Capture Images from Camera_1 (...	00179

5. Display workpiece model.

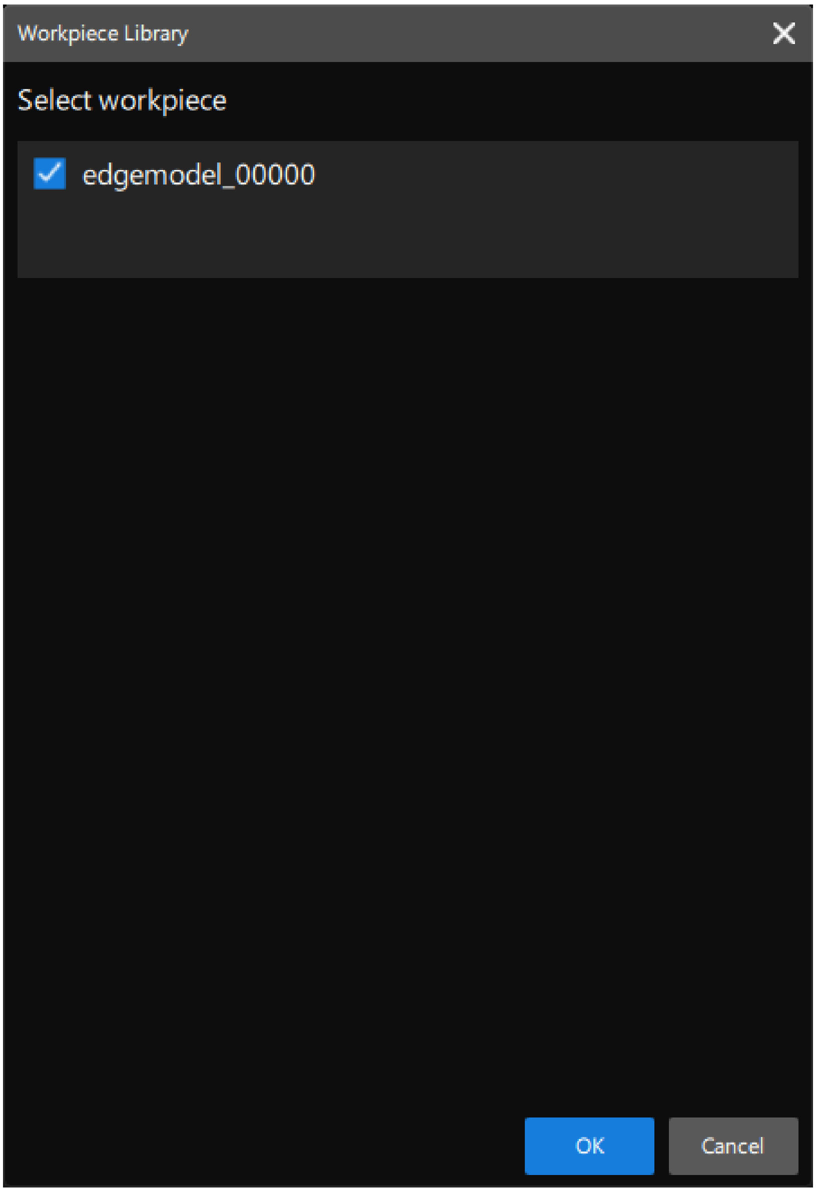
   After setting up the production recipe, click **[+]** button on the upper-right corner of the “Display workpiece model” tab.



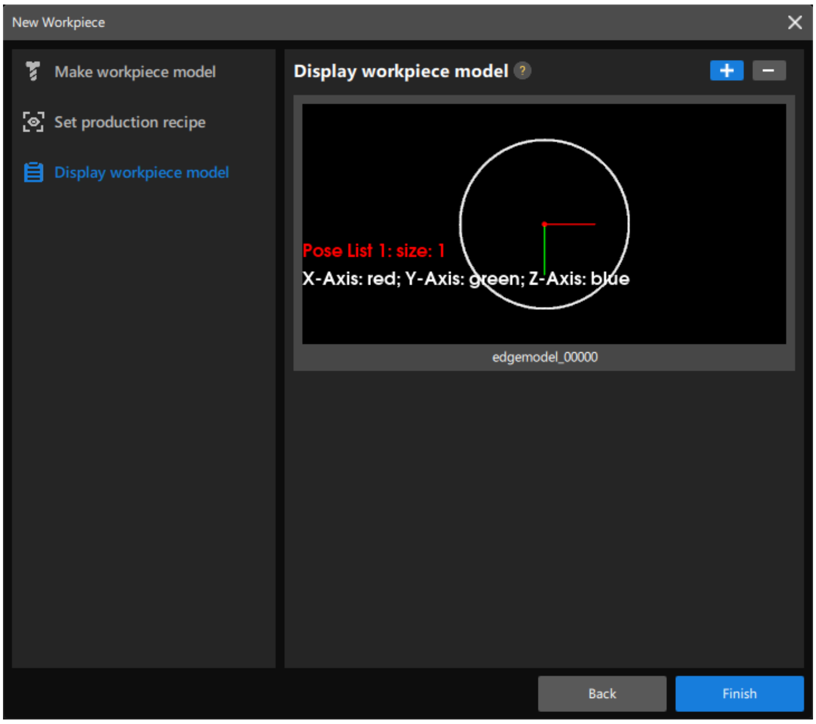
Please click the button in the upper right corner to select the model(s)

Up to two workpiece models can be added for display.

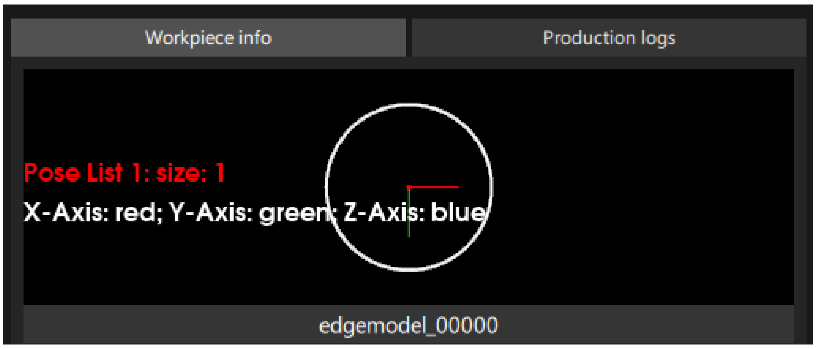
Select the workpiece model in the pop-up window, and click [OK].



After adding the workpiece model for display, click [**Finish**].



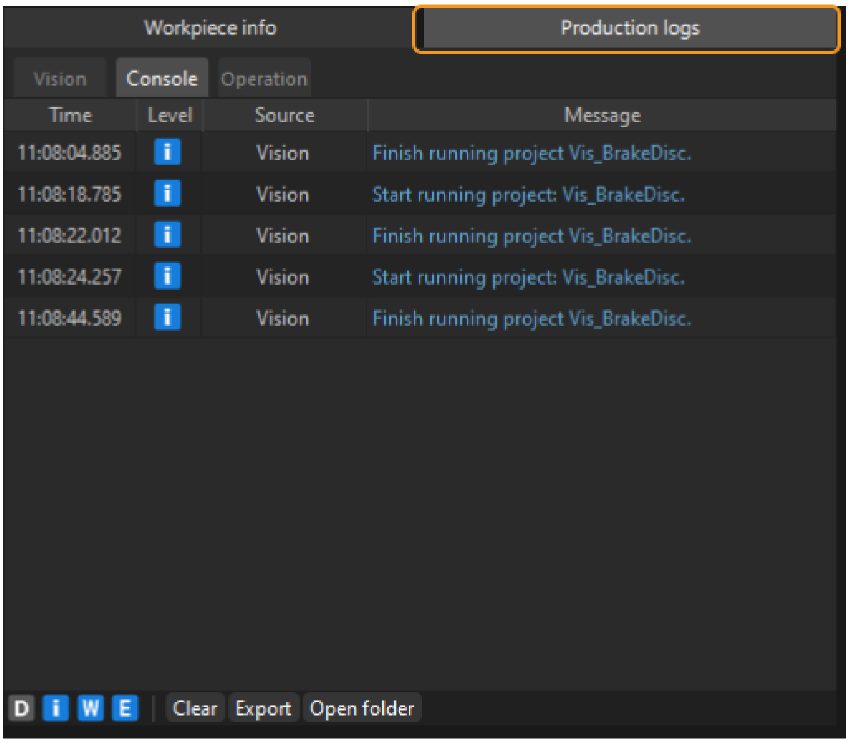
Now, the workpiece point cloud model you made will be displayed in the workpiece info area on the production interface.



- If there is no operation for 30 minutes, the “Workpiece info” panel will automatically switch to “Production logs.”
- If the point cloud model you made is not stored in the 3d_matching folder under `project folder\resource`, the point cloud model cannot be displayed in the workpiece info area.

4.4.2.10. Check Production Logs

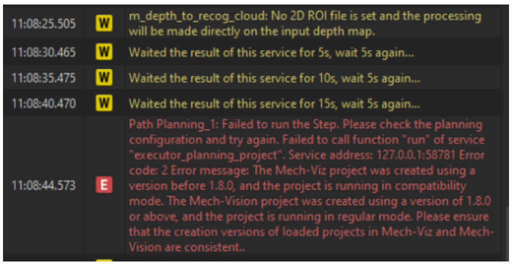
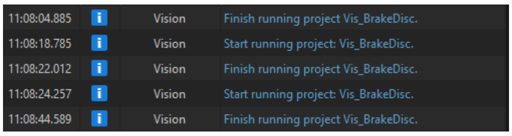
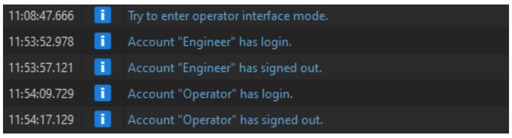
In the process of Production, you can check the relevant alerts when the project is running in the Alert area, which is shown in the following figure.



```
Time          Level  Source  Message
11:08:04.885  i      Vision  Finish running project Vis_BrakeDisc.
11:08:18.785  i      Vision  Start running project: Vis_BrakeDisc.
11:08:22.012  i      Vision  Finish running project Vis_BrakeDisc.
11:08:24.257  i      Vision  Start running project: Vis_BrakeDisc.
11:08:44.589  i      Vision  Finish running project Vis_BrakeDisc.
```

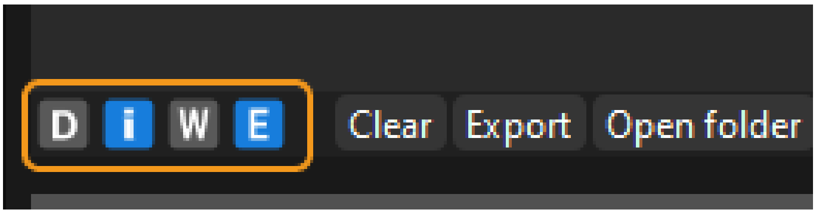
Production Log Introduction

Production logs include Vision log, Console log, and Operation log. The detailed description is shown in the following table.

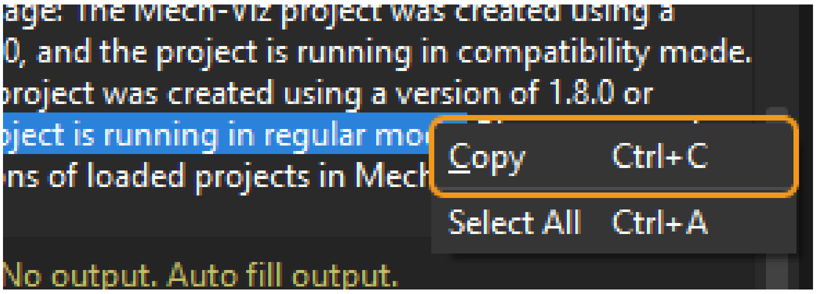
No.	Log	Description	Image
1	Vision	Logs when running the Vision project, including project debugging information, project error information, etc.	 11:08:25.505 W m_depth_to_recog_cloud: No 2D ROI file is set and the processing will be made directly on the input depth map. 11:08:30.465 W Waited the result of this service for 5s, wait 5s again... 11:08:35.475 W Waited the result of this service for 10s, wait 5s again... 11:08:40.470 W Waited the result of this service for 15s, wait 5s again... 11:08:44.573 E Path Planning_1: Failed to run the Step. Please check the planning configuration and try again. Failed to call function "run" of service "executor_planning_project". Service address: 127.0.0.1:58781 Error code: 2 Error message: The Mech-Viz project was created using a version before 1.8.0, and the project is running in compatibility mode. The Mech-Vision project was created using a version of 1.8.0 or above, and the project is running in regular mode. Please ensure that the creation versions of loaded projects in Mech-Viz and Mech-Vision are consistent..
2	Console	The control information related to the Vision project.	 11:08:04.885 i Vision Finish running project Vis_BrakeDisc. 11:08:18.785 i Vision Start running project: Vis_BrakeDisc. 11:08:22.012 i Vision Finish running project Vis_BrakeDisc. 11:08:24.257 i Vision Start running project: Vis_BrakeDisc. 11:08:44.589 i Vision Finish running project Vis_BrakeDisc.
3	Operation	Operation records while using the production interface.	 11:08:47.666 i Try to enter operator interface mode. 11:53:52.978 i Account "Engineer" has login. 11:53:57.121 i Account "Engineer" has signed out. 11:54:09.729 i Account "Operator" has login. 11:54:17.129 i Account "Operator" has signed out.

Production Log Instruction

- The letters in the lower-left corner of production log area are log level icons that represent different log levels. You can click corresponding icons to display or hide the log levels. An icon in blue indicates that log information of this level will be displayed. When the icon is in gray, the log information of this level will not be displayed.



- Double-click a single line in the log, select the content, and right-click and select **Copy** in the pop-up menu to copy the selected content.

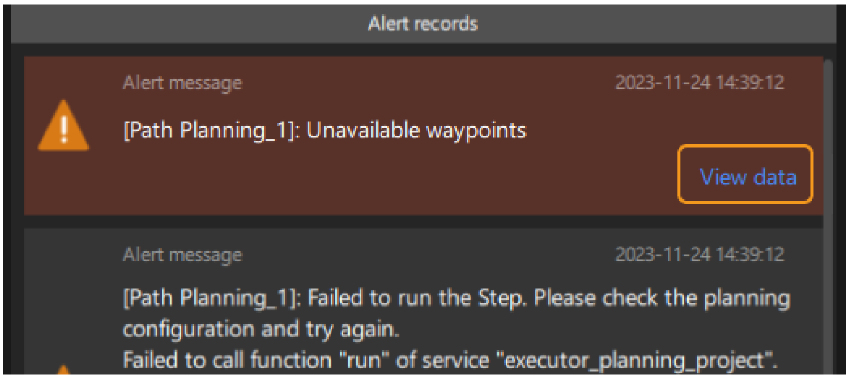


```
age: The Mech-Viz project was created using a
0, and the project is running in compatibility mode.
project was created using a version of 1.8.0 or
ject is running in regular mo
ns of loaded projects in Mech

No output. Auto fill output.
```

- Click **[Clear]** button at the bottom of production log area to clear something in one of the tabs of this area.
- Click **[Export]** button at the bottom of production log area to export something in one of the tabs of this area. The exported logs (.html files) will be saved to `logs` folder in the Mech-Vision installation path.
- Click **[Open folder]** button at the bottom of production log area to check production logs (.log files).

4.4.2.11. Check Alert Records

In the process of production, you can check the alerts relative to the project execution in the Alert area, which is shown in the following figure.



Alert records

Alert message 2023-11-24 14:39:12

[Path Planning_1]: Unavailable waypoints

Alert message 2023-11-24 14:39:12

[Path Planning_1]: Failed to run the Step. Please check the planning configuration and try again.
Failed to call function "run" of service "executor_planning_project".

You can click **[View data]**, and then view the exception data in the pop-up folder.

The following prerequisites are required to view the exception data in the pop-up folder:

- The camera must be a **real camera**.
- **Production data storage** feature is enabled in the General settings of the Production Interface Configurator.

4.4.2.12. Backup Files Management

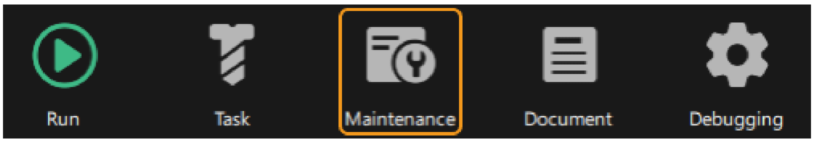
In the process of production, you can back up the files relevant to the project as follows.

Create Backups

This part introduces how to create a backup.

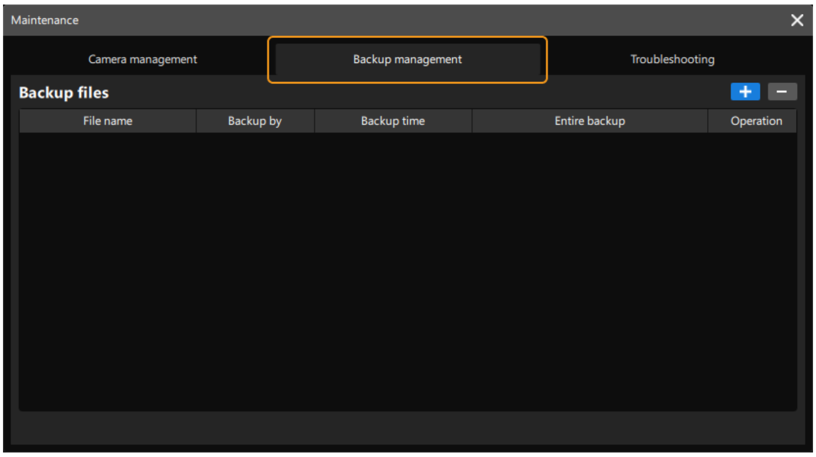
1. Open the Maintenance window.

   Click the button in the operation area.



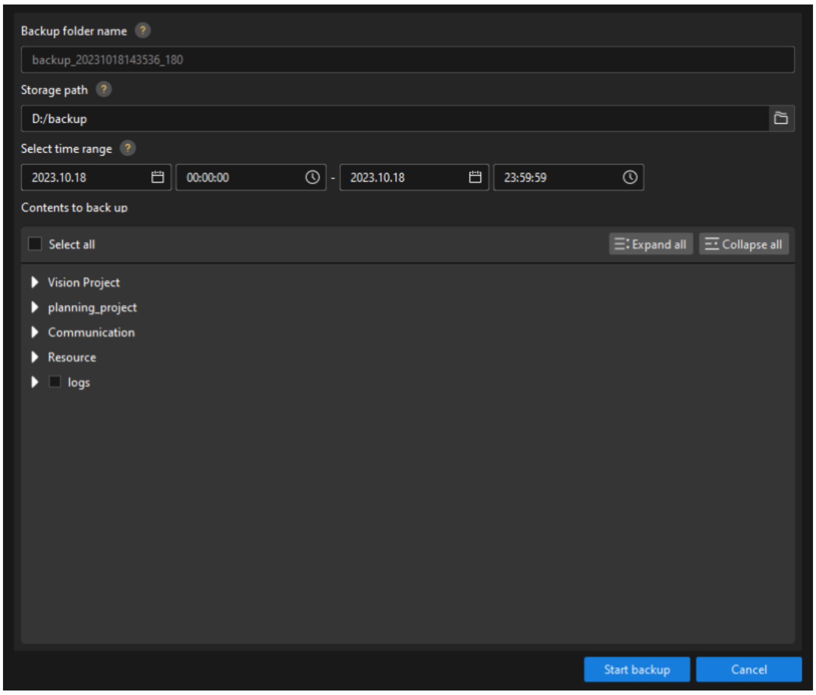
2. Click the Backup management tab.

   Click **Backup management** in the Maintenance window.



3. Configure the backup solutions.

   Click the [+] button in the Backup management tab, and then configure the backup solution in the pop-up Backup Solution window, including Backup folder name, Storage path, Select time range, and Contents to back up. After the configuration, click **[Start backup]** on the bottom, and then click **[OK]** in the pop-up window of “Backup succeeded.”



Backup folder name ?

backup_20231018143536_180

Storage path ?

D:/backup

Select time range ?

2023.10.18 00:00:00 - 2023.10.18 23:59:59

Contents to back up

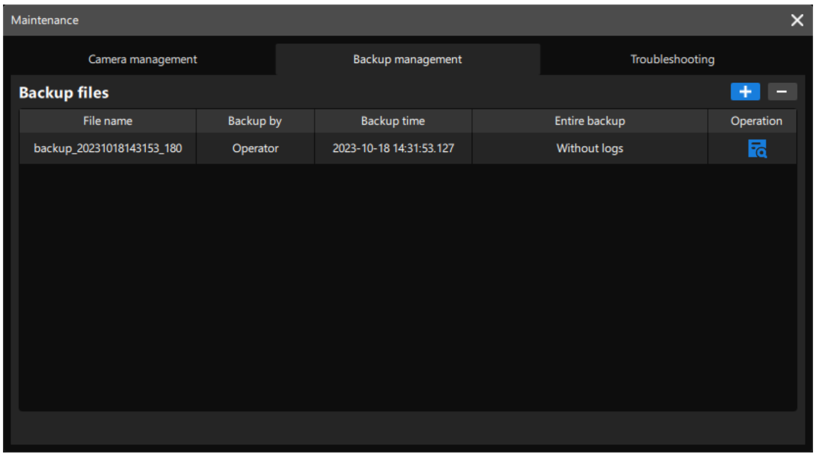
□ Select all

- Vision Project
- planning_project
- Communication
- Resource
- □ logs

The name of backup solution is the same as the name of the folder where the backup files are stored. The modification of its folder name is not supported. The folder is named as “backup_system time_version number” by default.

4. Check backup records.

   After the backup, you can check the backup records in the Backup management tab.



Backup files

File name	Backup by	Backup time	Entire backup	Operation
backup_20231018143153_180	Operator	2023-10-18 14:31:53.127	Without logs	🔍

Check Backup Content

To check the backup content, you can click 🔍 on the right side of the corresponding backup record, and then you can check its content in the pop-up window.

Delete Backup Content

To delete the backup content, you can click the **[-]** button on the right side of the corresponding backup record, and then click **[OK]** in the pop-up window.

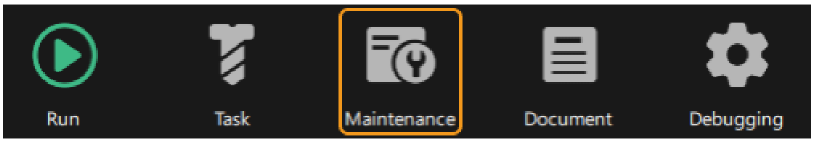
After deleting the backup content, the selected backup records and relevant backup files will be deleted.

4.4.2.13. Troubleshooting

When encountering problems in the process of production, you can use the Maintenance function in the operation area of the production interface for troubleshooting. Please follow the steps below to operate.

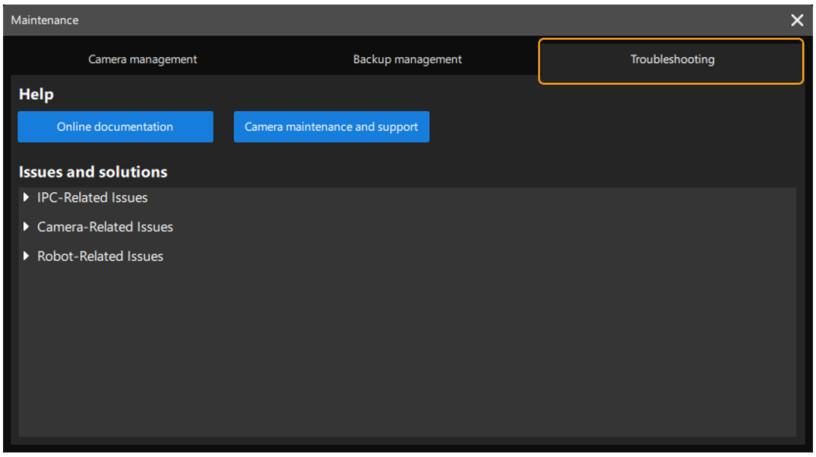
1. Enter the Maintenance window.

   Click the button in the operation area.



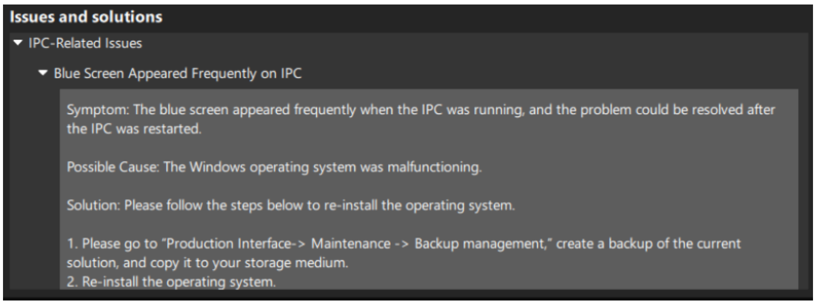
2. Enter the Troubleshooting tab.

   Click **Troubleshooting** in the Maintenance window.



3. Troubleshoot in line with issues listed there.

   Some issues are listed in the Troubleshooting tab. Click ▶ button to check the issues, possible reasons and solutions.



Issues and solutions

IPC-Related Issues

Blue Screen Appeared Frequently on IPC

Symptom: The blue screen appeared frequently when the IPC was running, and the problem could be resolved after the IPC was restarted.

Possible Cause: The Windows operating system was malfunctioning.

Solution: Please follow the steps below to re-install the operating system.

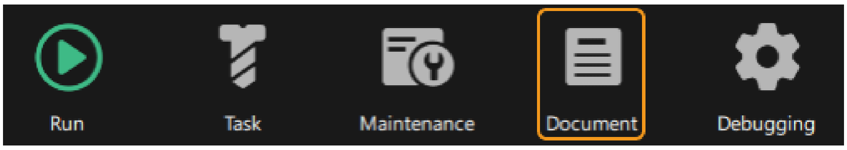
1. Please go to "Production Interface-> Maintenance -> Backup management," create a backup of the current solution, and copy it to your storage medium.
2. Re-install the operating system.

> You can also click [**Vision system service manual**] and [**Camera maintenance and support**] to check the online documents for troubleshooting.

4.4.2.14. View the Operation Guide

The associated operation guide set in the General settings in the Production Interface Configurator can be opened in the production interface.

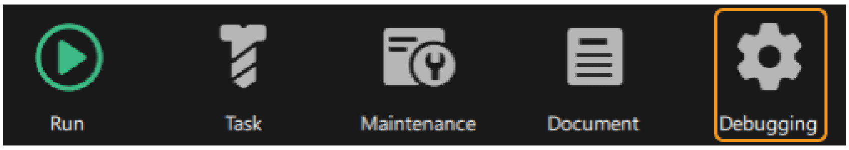
To view the operation guide in the production interface, you can click the ▤ button in the operation area.



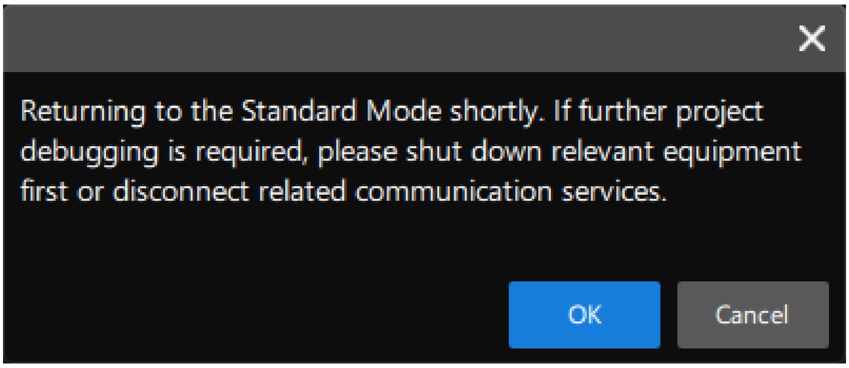
The associated operation guide can then be opened.

4.4.2.15. Return to Standard Mode

To return to the standard mode to modify the project, you can click the ⚙ button in the operation area.



Then click [OK] in the pop-up window.

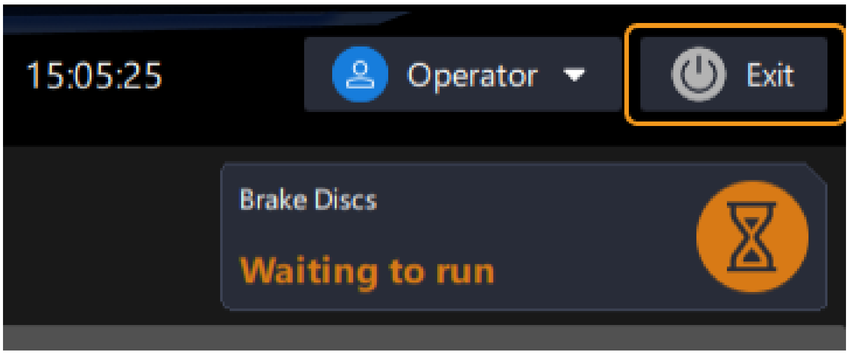


Returning to the Standard Mode shortly. If further project debugging is required, please shut down relevant equipment first or disconnect related communication services.

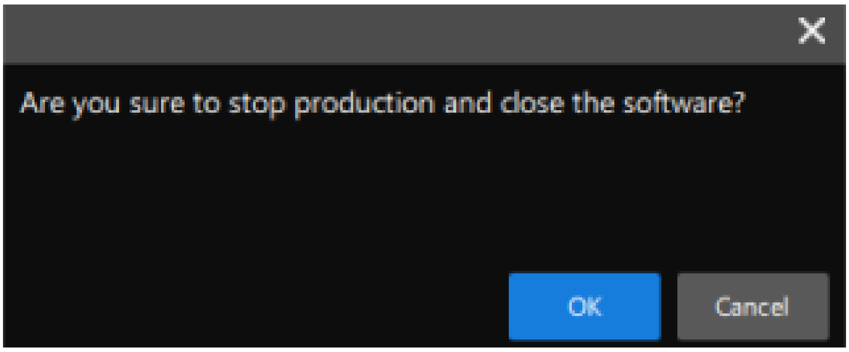
This function is subject to account permissions control, and it is disabled when the account role is Operator.

4.4.2.16. Exit the Production Interface

To end the production, exit the production interface and close the software, you can click the [Exit] button in the upper-right corner of the production interface.



Then click [OK] in the pop-up window to exit the production interface. The Mech-Vision home interface will be closed with the exit of the production interface.


Are you sure to stop production and close the software?

- When the project is running, the production interface cannot be closed.
- This function is subject to account permissions control, and it is disabled when the account role is Operator.

4.5. Hand-Eye Calibration Guide

This chapter guides you to complete the hand-eye calibration for robots.

Read the following section to select the proper hand-eye calibration process.

Select Proper Hand-Eye Calibration Process

Please follow the operation guide for different hand-eye calibration processes.

- Hand-Eye Calibration in the Eye-to-Hand Setup
  - Automatic Calibration in the Eye-to-Hand Setup (Six-Axis Robots)
  - Automatic Calibration in the Eye-to-Hand Setup (Four-Axis Robots)
  - Manual Calibration in the Eye-to-Hand Setup (Six-Axis Robots - Using Multiple Random Calibration Board Poses Method)
  - Manual Calibration in the Eye-to-Hand Setup (Four-Axis Robots - Using Multiple Random Calibration Board Poses Method)
  - Manual Calibration in the Eye-to-Hand Setup (Six-Axis Robots - Using TCP Touch Method)
  - Manual Calibration in the Eye-to-Hand Setup (Four-Axis Robots - Using TCP Touch Method)
- Hand-Eye Calibration in the Eye-in-Hand Setup
  - Automatic Calibration in the Eye-in-Hand Setup (Six-Axis Robots)
  - Automatic Calibration in the Eye-in-Hand Setup (Four-Axis Robots)
  - Manual Calibration in the Eye-in-Hand Setup (Six-Axis Robots - Using Multiple Random Calibration Board Poses Method)
  - Manual Calibration in the Eye-in-Hand Setup (Four-Axis Robots - Using Multiple Random Calibration Board Poses Method)
  - Manual Calibration in the Eye-in-Hand Setup (Six-Axis Robots - Using TCP Touch Method)
  - Manual Calibration in the Eye-in-Hand Setup (Four-Axis Robots - Using TCP Touch Method)

- Hand-Eye Calibration for Truss/Gantry Robots
- Hand-Eye Calibration in the Eye-to-Eye Setup

Read the following sections to learn more about hand-eye calibration.

Calibration-related Concepts

Calibration Principles

Explanation of the Calibration for Truss/Gantry Robots

Calibration Results Check and Analysis

Calibration Troubleshooting

4.5.1. Select Proper Hand-Eye Calibration Process

Hand-Eye Calibration establishes **the transformation relationship between the camera and robot reference frames**. With this relationship, the object pose determined by the vision system can be transformed into that in the robot reference frame, which guides the robot to perform its tasks. The accuracy of hand-eye calibration results affects the accuracy of robot picking.

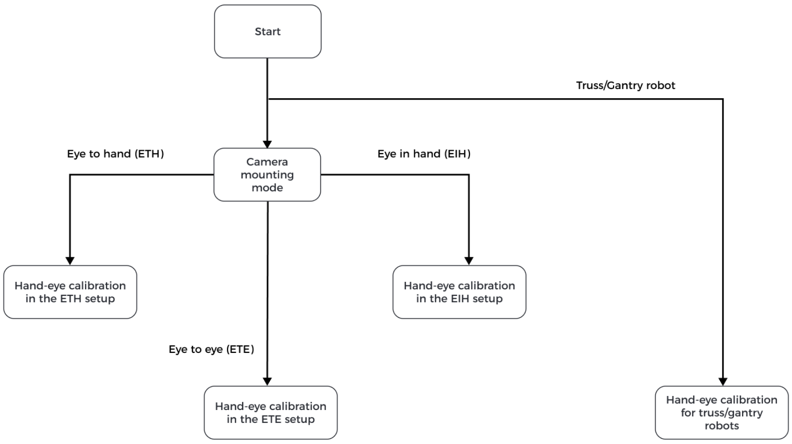
After the Mech-Mind Vision System is constructed, you can start the hand-eye calibration. In addition, you need to perform hand-eye calibration again in any of the following conditions:

- The camera moves in reference to the robot base (in the eye-to-hand setup) or flange (in the eye-in-hand setup).
- The camera is replaced.
- The robot is replaced.

Select the Proper Calibration Process

The calibration procedures are affected by robot communication modes, camera mounting modes, calibration modes, and calibration data collection methods. Mech-Vision provides standardized calibration procedures for various scenarios, helping you calibrate the camera’s intrinsic and extrinsic parameters quickly and accurately.

For explanations of the robot communication modes, camera mounting modes, calibration modes, and calibration data collection methods, refer to the section Calibration-related Concepts.


Start
Truss/Gantry robot
Eye to hand (ETH)
Camera mounting mode
Eye in hand (EIH)
Hand-eye calibration in the ETH setup
Hand-eye calibration in the EIH setup
Eye to eye (ETE)
Hand-eye calibration in the ETE setup
Hand-eye calibration for truss/gantry robots

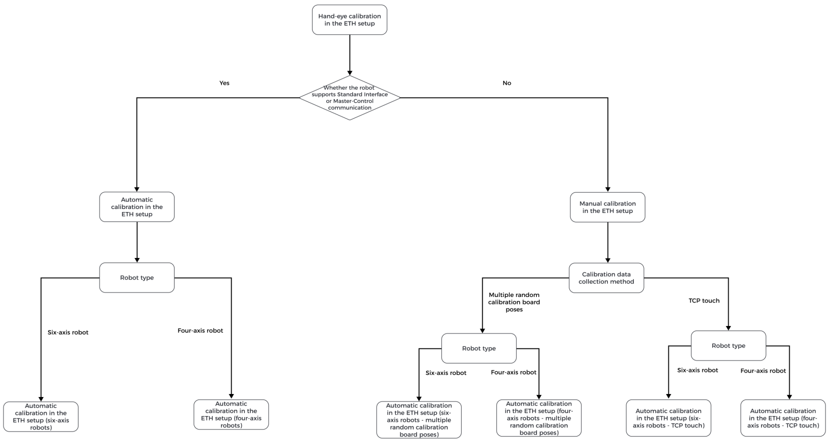
Please select the calibration processes according to the camera mounting modes.

- If the camera mounting mode is eye to hand (ETH), go to Hand-Eye Calibration in the ETH Setup for further selection.
- If the camera mounting mode is eye to hand (ETH), go to Hand-Eye Calibration in the EIH Setup for further selection.
- If the camera mounting mode is eye to eye (ETE), follow the process of Hand-Eye Calibration in the ETE Setup.

If a truss/gantry robot is used in a project, follow the process of Hand-Eye Calibration for Truss/Gantry Robots.

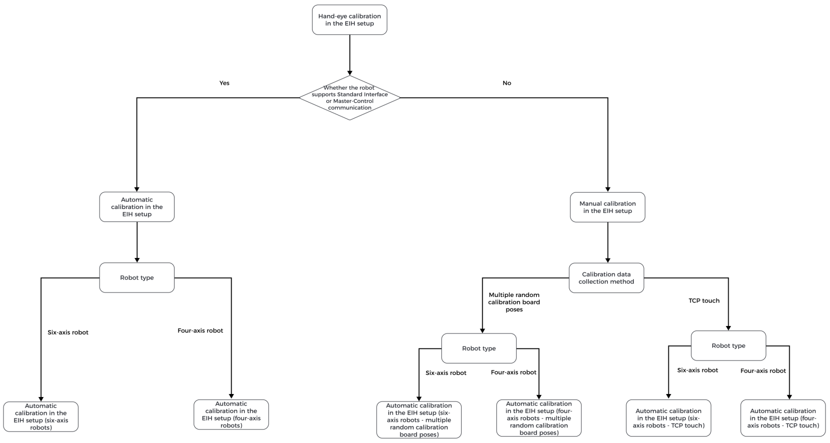
Hand-Eye Calibration in the ETH Setup

If the camera mounting mode is ETH, further select a specific calibration process in this section.


Hand-eye calibration in the ETH setup
Yes
Whether the robot supports Standard Interface or Master-Control communication
No
Automatic calibration in the ETH setup
Manual calibration in the ETH setup
Robot type
Calibration data collection method
Six-axis robot
Four-axis robot
Multiple random calibration board poses
TCP touch
Robot type
Robot type
Six-axis robot
Four-axis robot
Six-axis robot
Four-axis robot
Automatic calibration in the ETH setup (six-axis robots)
Automatic calibration in the ETH setup (four-axis robots)
Automatic calibration in the ETH setup (six-axis robots - multiple random calibration board poses)
Automatic calibration in the ETH setup (four-axis robots - multiple random calibration board poses)
Automatic calibration in the ETH setup (six-axis robots - TCP touch)
Automatic calibration in the ETH setup (four-axis robots - TCP touch)

Hand-Eye Calibration in the EIH Setup

If the camera mounting mode is EIH, further select a specific calibration process in this section.


Hand-eye calibration in the EIH setup
Whether the robot supports Standard Interface or Master-Control communication
Yes
No
Automatic calibration in the EIH setup
Robot type
Six-axis robot
Four-axis robot
Automatic calibration in the EIH setup (six-axis robots)
Automatic calibration in the EIH setup (four-axis robots)
Manual calibration in the EIH setup
Calibration data collection method
Multiple random calibration board poses
TCP touch
Robot type
Six-axis robot
Four-axis robot
Automatic calibration in the EIH setup (six-axis robots - multiple random calibration board poses)
Automatic calibration in the EIH setup (four-axis robots - multiple random calibration board poses)
Robot type
Six-axis robot
Four-axis robot
Automatic calibration in the EIH setup (six-axis robots - TCP touch)
Automatic calibration in the EIH setup (four-axis robots - TCP touch)

4.5.2. Hand-Eye Calibration in the Eye-to-Hand Setup

You can select a proper calibration process according to the calibration mode (automatic or manual), robot type (six-axis robots or four-axis robots) and calibration data collection method (multiple random calibration board poses or TCP touch), and follow the operation guide for the selected calibration process.

- Automatic Calibration in the Eye-to-Hand Setup (Six-Axis Robots)
- Automatic Calibration in the Eye-to-Hand Setup (Four-Axis Robots)
- Manual Calibration in the Eye-to-Hand Setup (Six-Axis Robots - Using Multiple Random Calibration Board Poses Method)
- Manual Calibration in the Eye-to-Hand Setup (Four-Axis Robots - Using Multiple Random Calibration Board Poses Method)
- Manual Calibration in the Eye-to-Hand Setup (Six-Axis Robots - Using TCP Touch Method)
- Manual Calibration in the Eye-to-Hand Setup (Four-Axis Robots - Using TCP Touch Method)

4.5.2.1. Automatic Calibration in the Eye-to-Hand Setup (Six-Axis Robots)

This how-to guide introduces how to complete the automatic calibration for six-axis robots in the eye-to-hand (ETH) setup.

Preparation before Calibration

Before hand-eye calibration, you need to finish the following preparations:

- Construct the Mech-Mind Vision System.
- Prepare the materials required for calibration.
- Adjust the effect of calibration board’s point cloud.
- Complete the robot communication configuration.

Construct the Vision System

Construct the Mech-Mind Vision System by referring to the section Vision System Hardware Setup.

You need to use Mech-Eye Viewer, Mech-Vision and Mech-Viz during hand-eye calibration. Please ensure that they have been installed and are running the latest versions.

Prepare the Materials Required for Calibration

The automatic calibration in the ETH setup needs to use the calibration board.

Please prepare the calibration board according to the following requirements:

- Ensure that the circles of the calibration board are clearly visible and without obvious scratches, and the board does not suffer from deformations.
- In the ETH setup, mount the robot-specific bracket for calibration board onto the robot flange, and then mount the calibration board onto the bracket. Make sure that the calibration board is attached securely, and that the board is parallel to the XY plane of the robot flange frame.

> If an undetachable gripper is connected to the robot flange, you can attach the calibration board directly to the gripper.

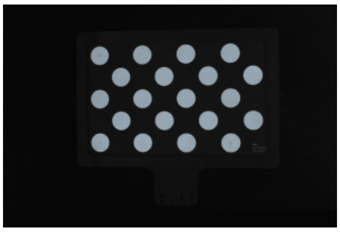
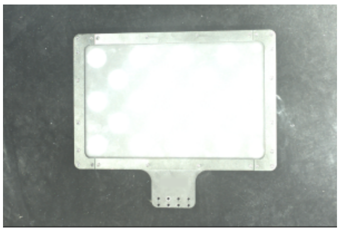
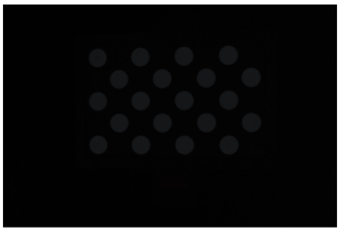
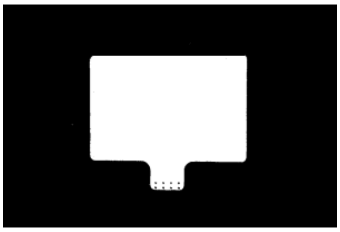
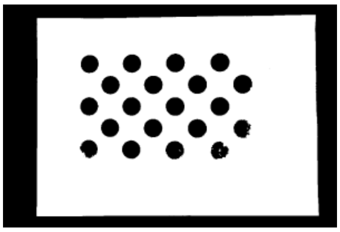
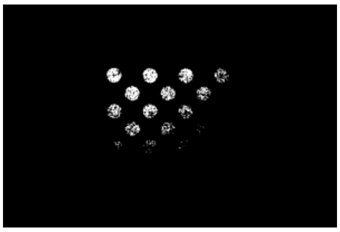
In addition, before calibration, move the robot to the starting point for calibration. For the automatic calibration in the ETH setup, the starting point is the bottom center of the camera's field of view (the robot will move the calibration board upwards during calibration).

Adjust the Effect of Calibration Board's Point Cloud

1. Open Mech-Eye Viewer to adjust camera parameters.
2. Adjust the 2D parameters to make sure that the 2D image of the calibration board is clear and neither overexposed nor underexposed.
3. Adjust 3D parameters to make sure that the point clouds of the circles on the calibration board are complete and have clear contours. It is recommended to set **Surface Smoothing** and **Outlier Removal** to **Normal** in the **Point Cloud Processing** section to reduce the point cloud fluctuation range.

> If the on-site ambient lights are not ideal and affect the quality of 2D images and point clouds, you can use shading or supplemental light to improve the lighting conditions.

4. Make sure that the images and point cloud of the calibration board are up to standard after completing the preceding steps.

	Normal	Overexposed	Underexposed
2D image			
Point cloud			

Complete the Robot Communication Configuration

If the robot uses the Standard Interface to communicate with the Vision System, please set up the Standard Interface communication with the robot. For details, refer to the section Standard Interface Communication.

If the robot uses the Master-Control to communicate with the Vision System, please set up the Master-Control communication with the robot. For details, refer to the section Master-Control Communication.

Pre-calibration Configuration

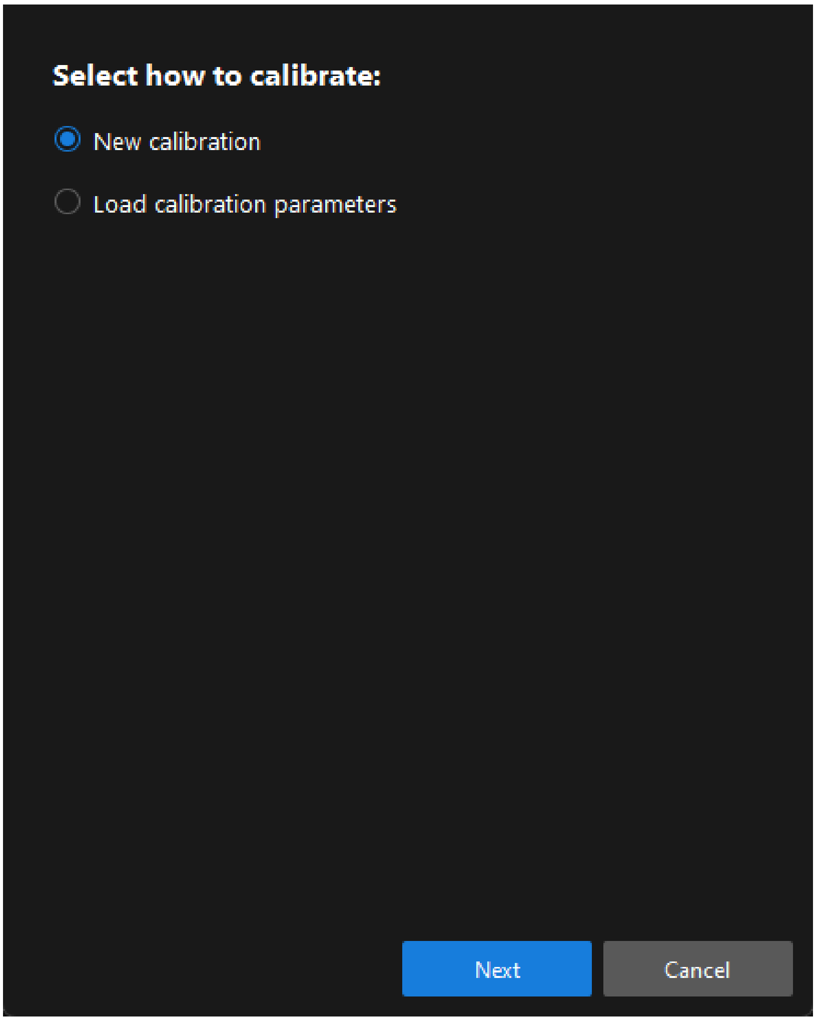
If the robot uses the Standard Interface to communicate with the Vision System, complete Pre-calibration Configuration (Standard Interface).

If the robot uses the Master-Control to communicate with the Vision System, complete Pre-calibration Configuration (Master-Control).

Pre-calibration Configuration (Standard Interface)

If the Standard Interface is used to communicate with the robot, follow these steps:

1. Open Mech-Vision, and click the **[Camera Calibration (Standard)]** button in the toolbar.The **Configuration before Calibration** window will be prompted.
2. In the **Select how to calibrate** window, select the **New calibration** radio button, and then click the **[Next]** button.

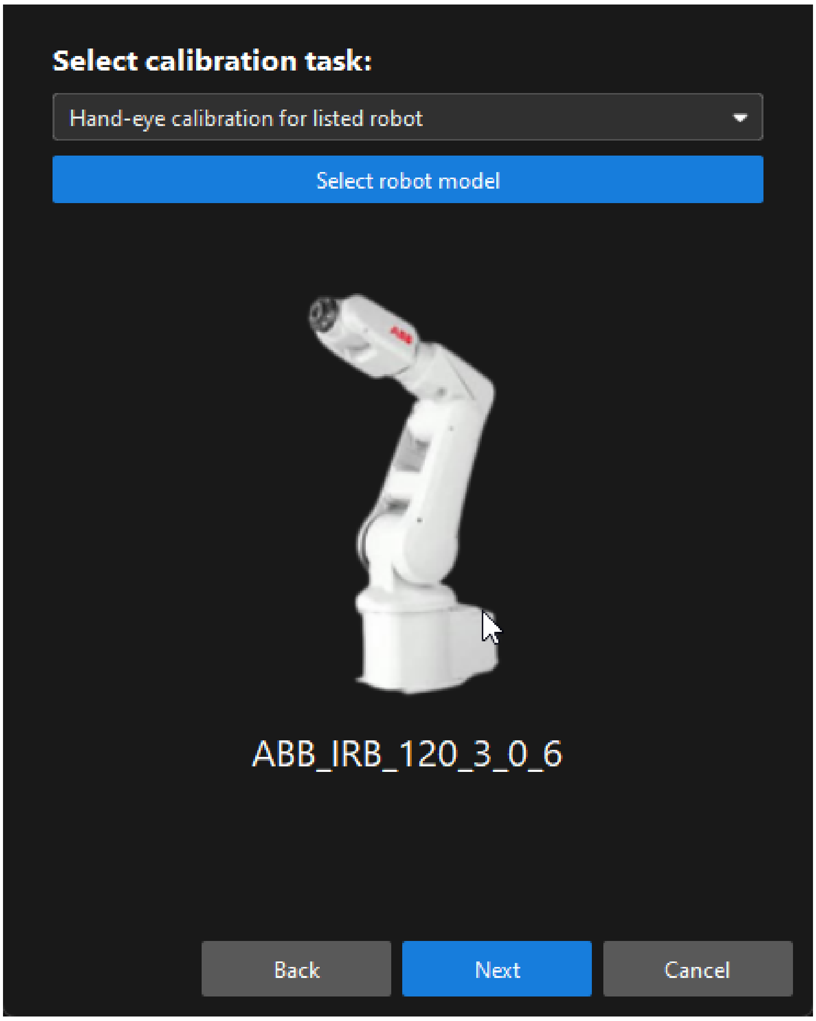


Select how to calibrate:

● New calibration

○ Load calibration parameters

3. In the **Select calibration task** window, select **Hand-eye calibration for listed robot** from the drop-down list box, click the **[Select robot model]** button to select the robot model used by the project, and then click the **[Next]** button.

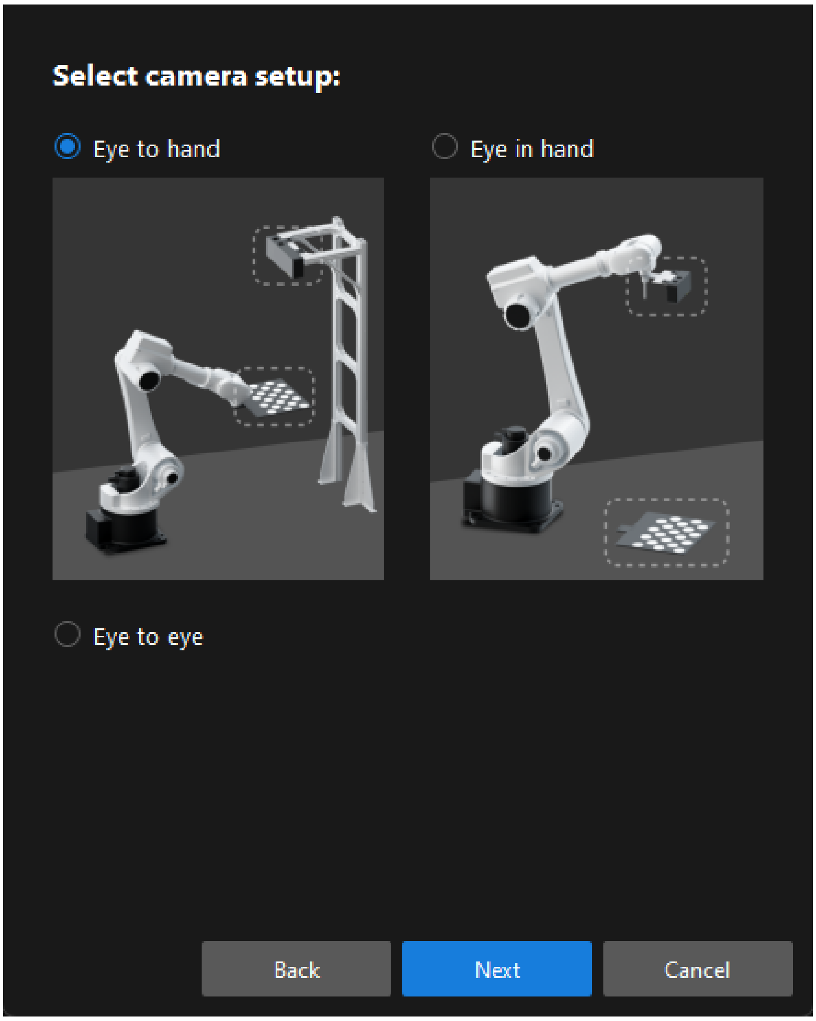


Select calibration task:

Hand-eye calibration for listed robot

ABB_IRB_120_3_0_6

4. In the **Select camera setup** window, select the **Eye to hand** radio button, and then click the **[Next]** button.



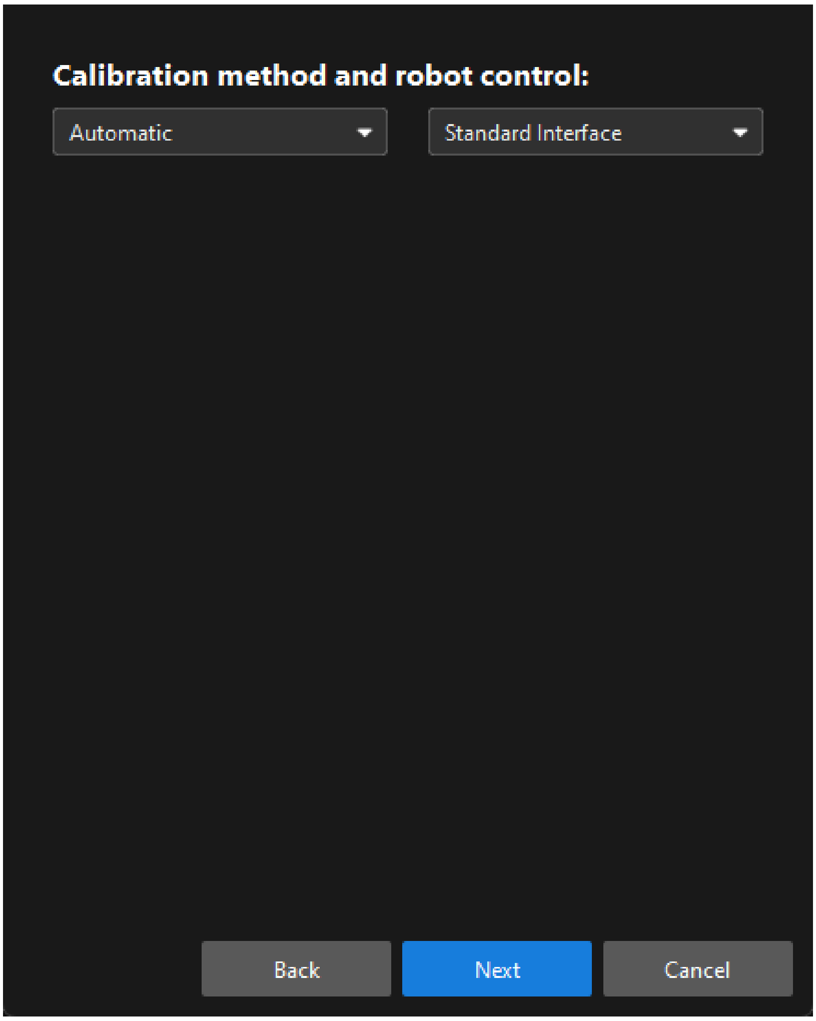
Select camera setup:

● Eye to hand

○ Eye in hand

○ Eye to eye

5. In the **Calibration method and robot control** window, select **Automatic** and **Standard Interface**, and then click the **[Next]** button.

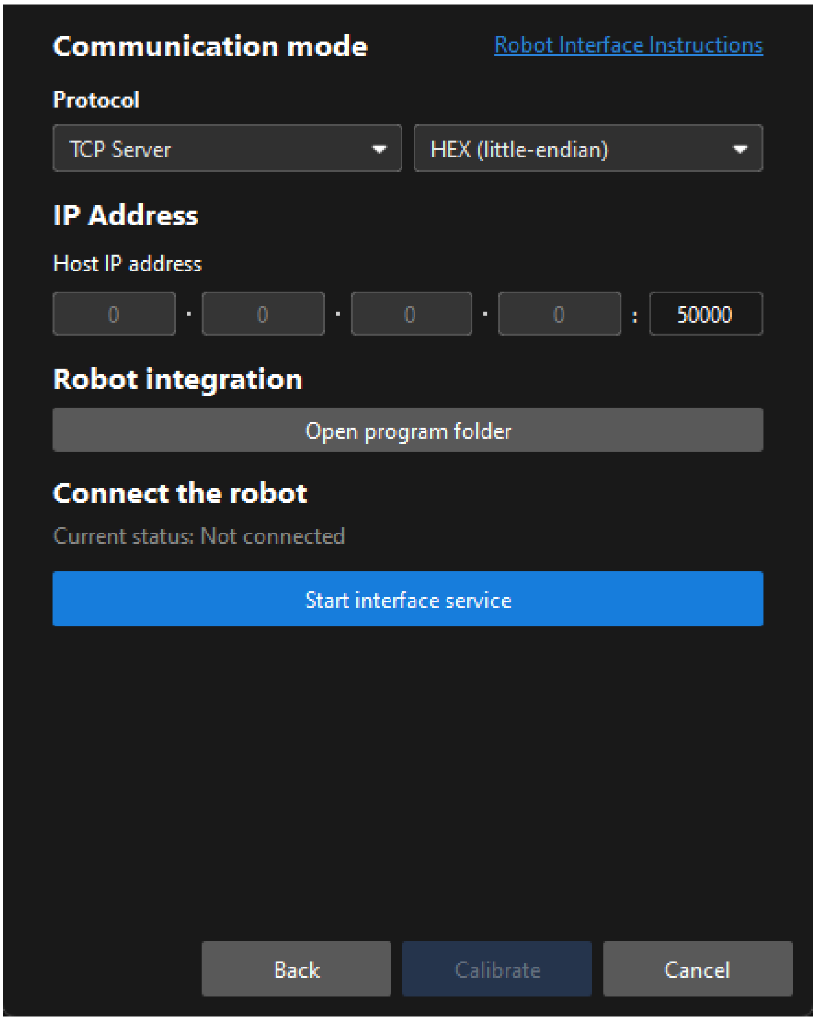


Calibration method and robot control:

Automatic

Standard Interface

6. In the new window, set parameters **Protocol** and **Host IP address**, and then click the **[Start interface service]** button. The button will turn into **Waiting for the robot to connect...**



Communication mode

Protocol

TCP Server

HEX (little-endian)

IP Address

Host IP address

0 . 0 . 0 . 0 : 50000

Robot integration

Connect the robot

Current status: Not connected

- In the preceding figure, **TCP Server** and **Hex (Little-endian)** are only the **Protocol** setting example of the ABB robot. The protocol settings of other robot brands are different. The software will automatically set this parameter and you just need to keep the default setting.
- The port number for the **Host IP address** parameter is **50000** by default. If the port number determined in the communication solution is not 50000, please change it as required.

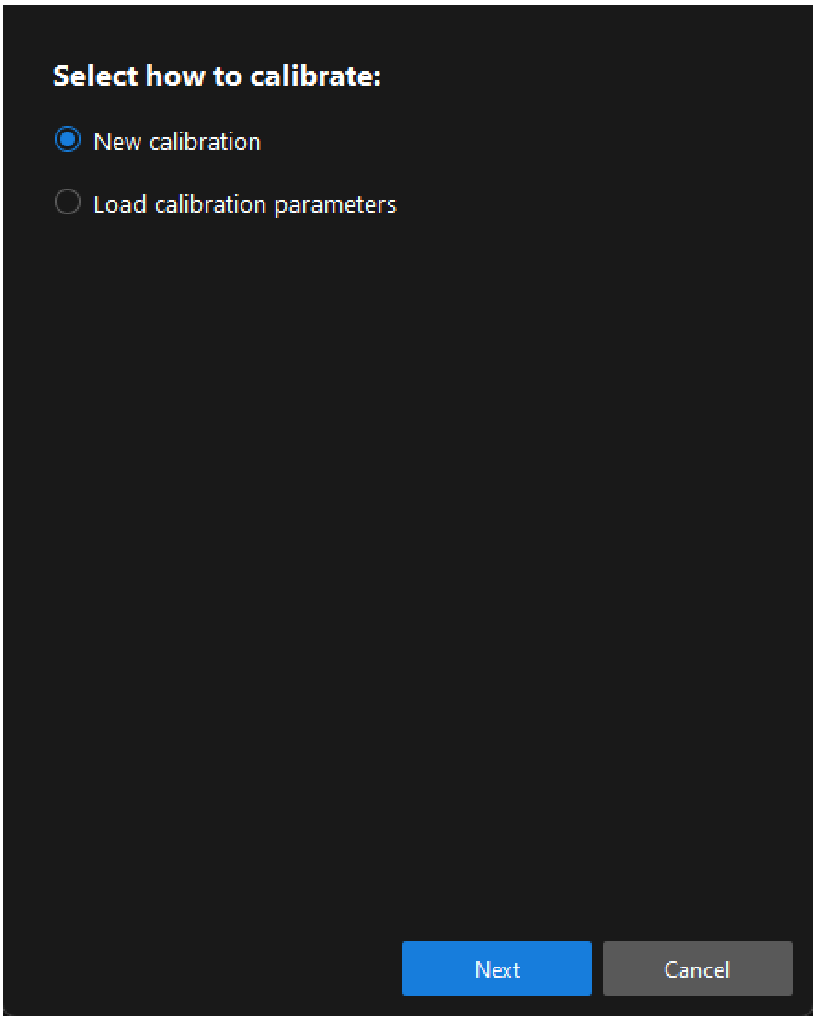
7. On the robot teach pendant, **select the automatic calibration program, teach the start point and run the program**. For details on the calibration programs of different robots, refer to the section Standard Interface Communication. After the program is started, the log message "Entering the calibration process, please start the calibration in Mech-Vision" will be printed in the Console log panel.
8. Return to Mech-Vision, confirm that "Connected" status message is displayed in the **Connect the robot** area, and click the **[Calibrate]** button. The **Calibration (Eye to hand)** window will be prompted.

Pre-calibration Configuration (Master-Control)

If the Master-Control is used to communicate with the robot, follow these steps:

1. Open Mech-Vision, and click the **[Camera Calibration (Standard)]** button in the toolbar. The **Configuration before Calibration** window will be prompted.

2. In the **Select how to calibrate** window, select the **New calibration** radio button, and then click the **[Next]** button.

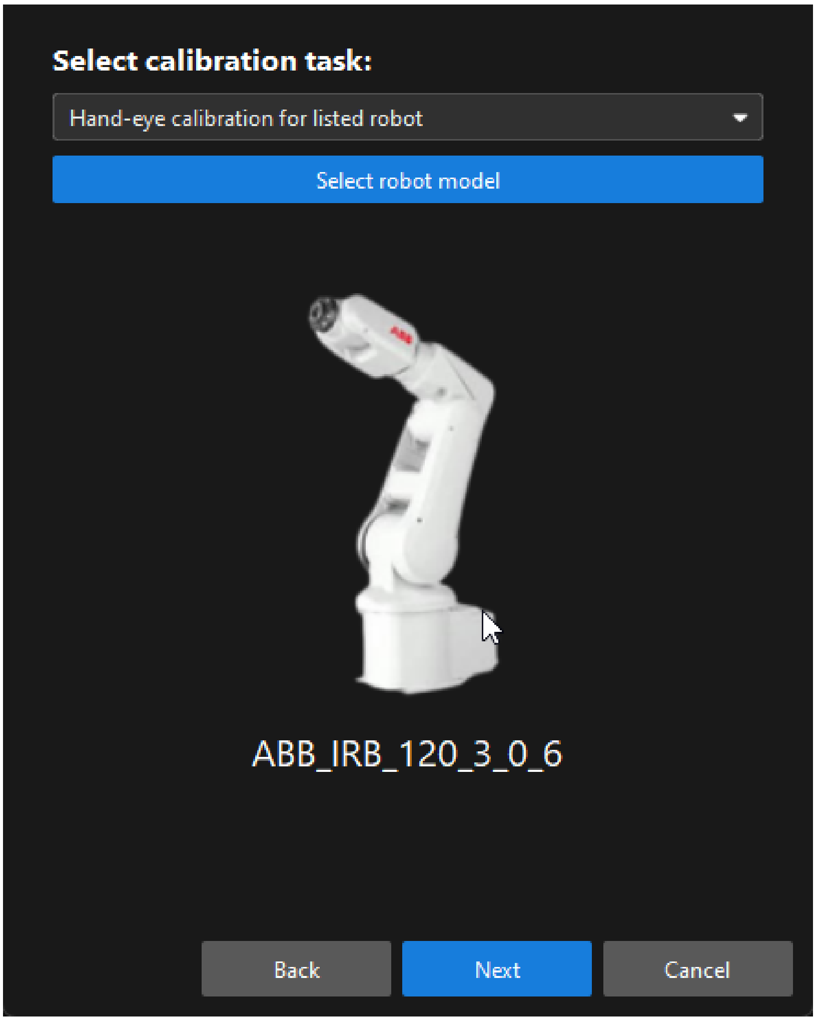


Select how to calibrate:

● New calibration

○ Load calibration parameters

3. In the **Select calibration task** window, select **Hand-eye calibration for listed robot** from the drop-down list box, click the **[Select robot model]** button to select the robot model used by the project, and then click the **[Next]** button.

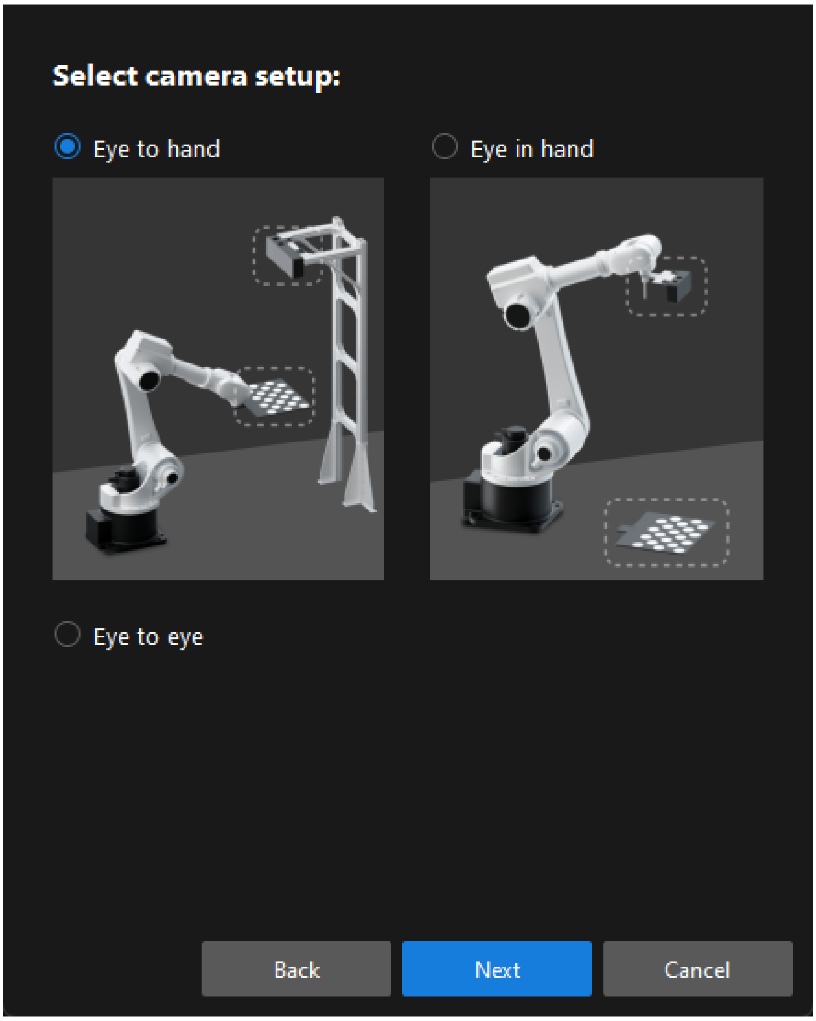


Select calibration task:

Hand-eye calibration for listed robot

ABB_IRB_120_3_0_6

4. In the **Select camera setup** window, select the **Eye to hand** radio button, and then click the **[Next]** button.



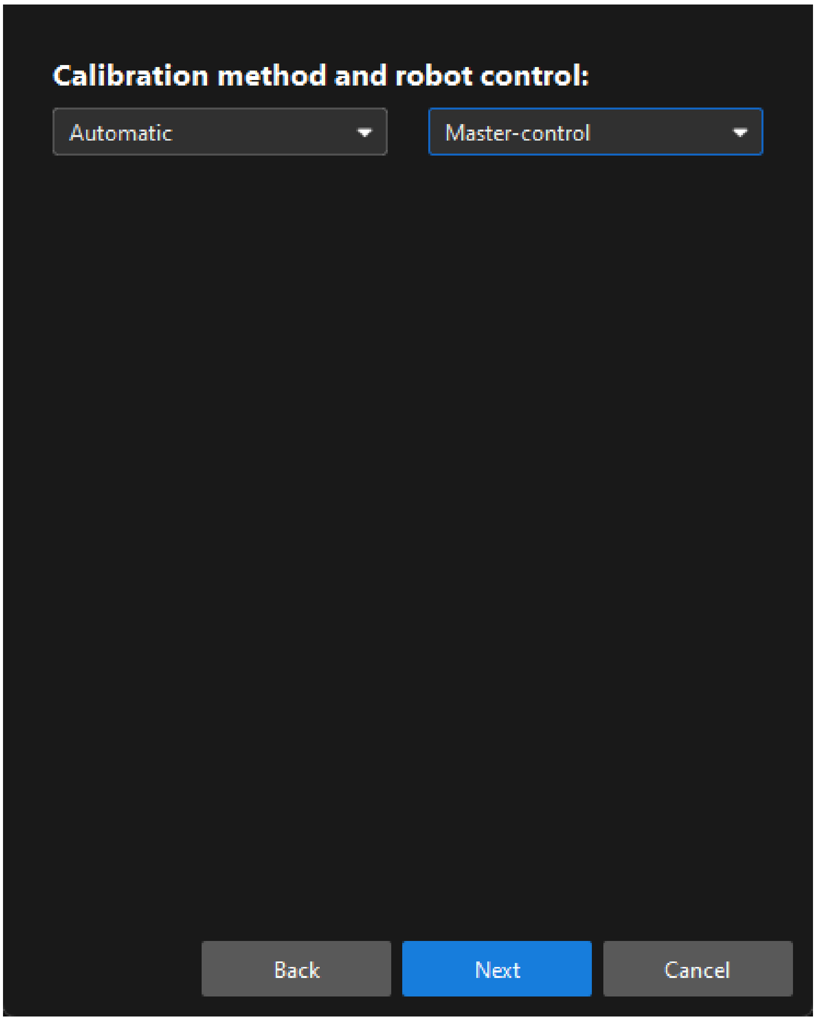
Select camera setup:

● Eye to hand

○ Eye in hand

○ Eye to eye

5. In the **Calibration method and robot control** window, select **Automatic** and **Master-control**, and then click the [**Next**] button.

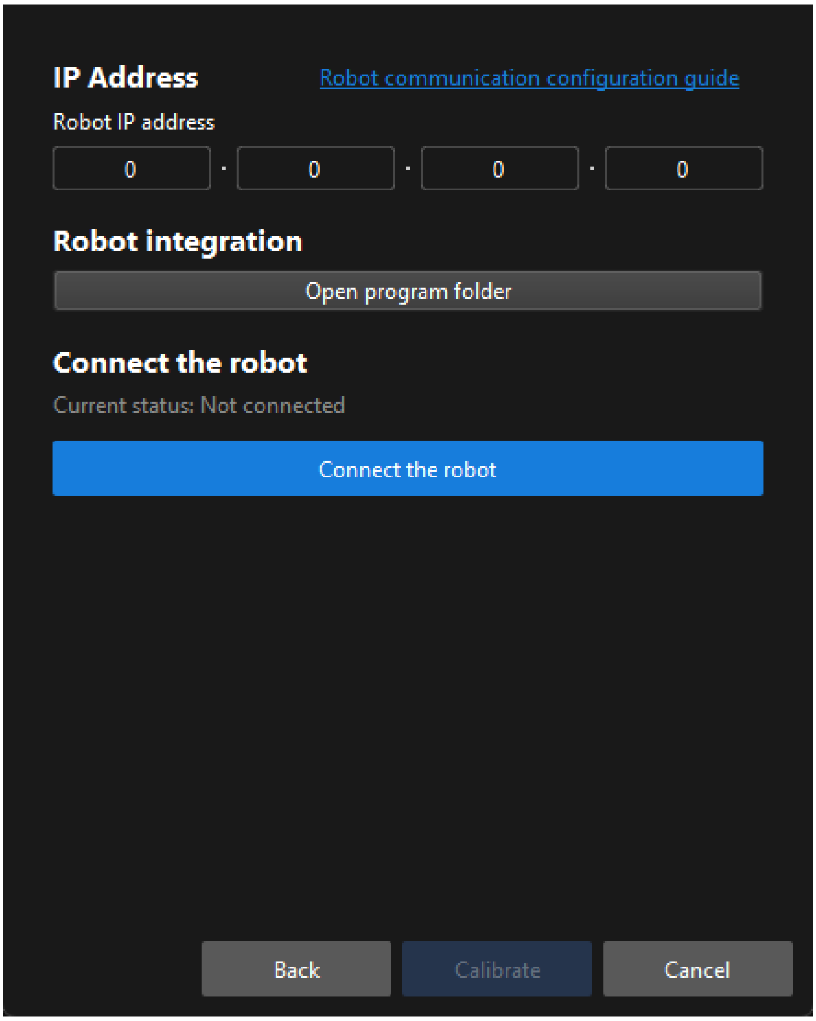


Calibration method and robot control:

Automatic

Master-control

6. In the prompted window, set the **Robot IP address** parameter.



IP Address

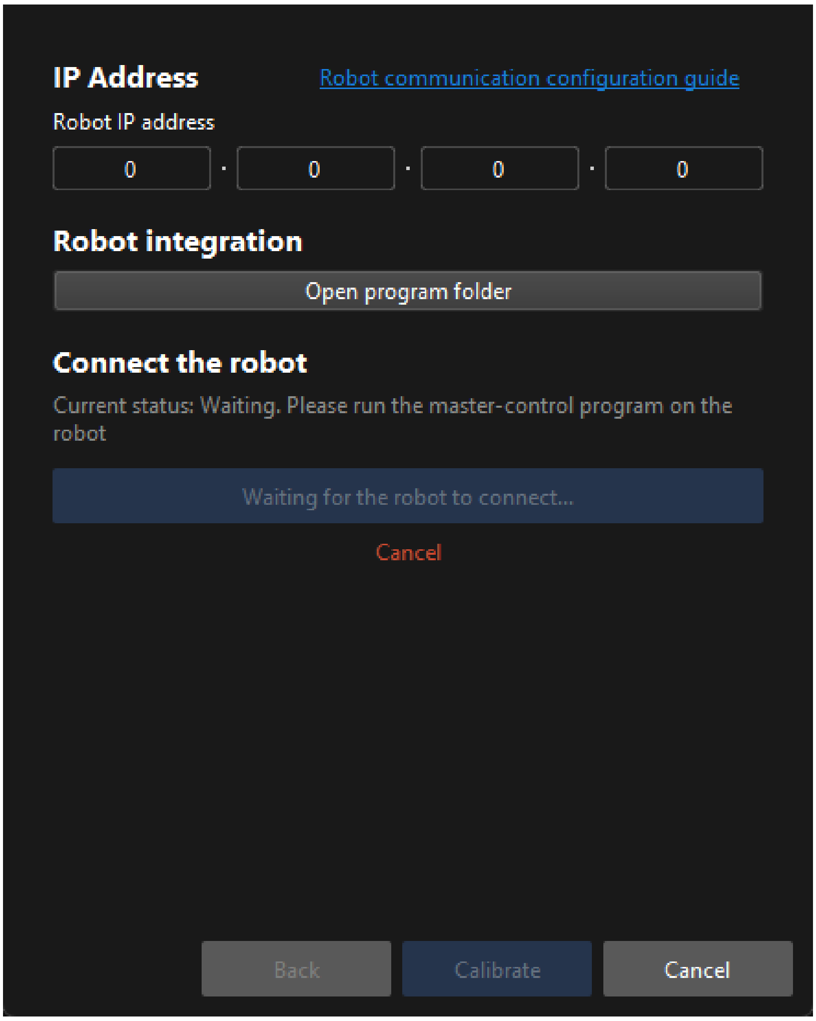
Robot IP address

0 . 0 . 0 . 0

Robot integration

Current status: Not connected

7. On the robot teach pendant, **select and run the Master-control program**. For the instructions of different robots, refer to the section Master-Control Communication.
8. Return to Mech-Vision, and click the **[Connect the robot]** button in the **Connect the robot** area. The button will turn into **Waiting for the robot to connect...**



IP Address

Robot IP address

0 . 0 . 0 . 0

Robot integration

Connect the robot

Current status: Waiting. Please run the master-control program on the robot

Robot communication configuration guide
Open program folder
Waiting for the robot to connect...
Cancel
Back
Calibrate
Cancel

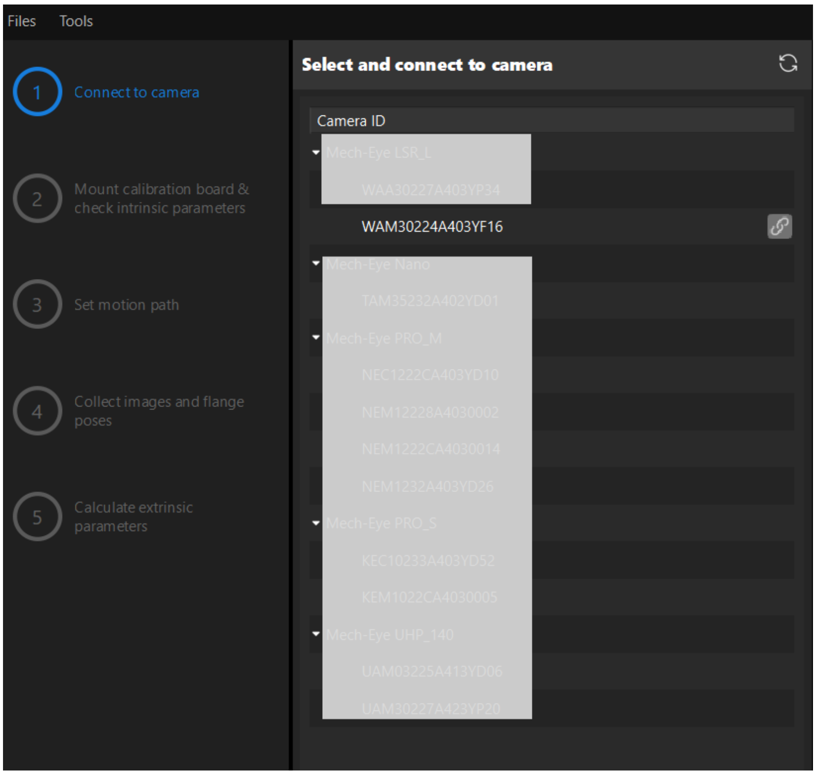
9. Wait until the "Connected" status message is displayed in the **Connect the robot** area, and then click the **[Calibrate]** button.The **Calibration (Eye to hand)** window will be prompted.

Till now, you have completed the pre-calibration configuration and can start the calibration procedure.

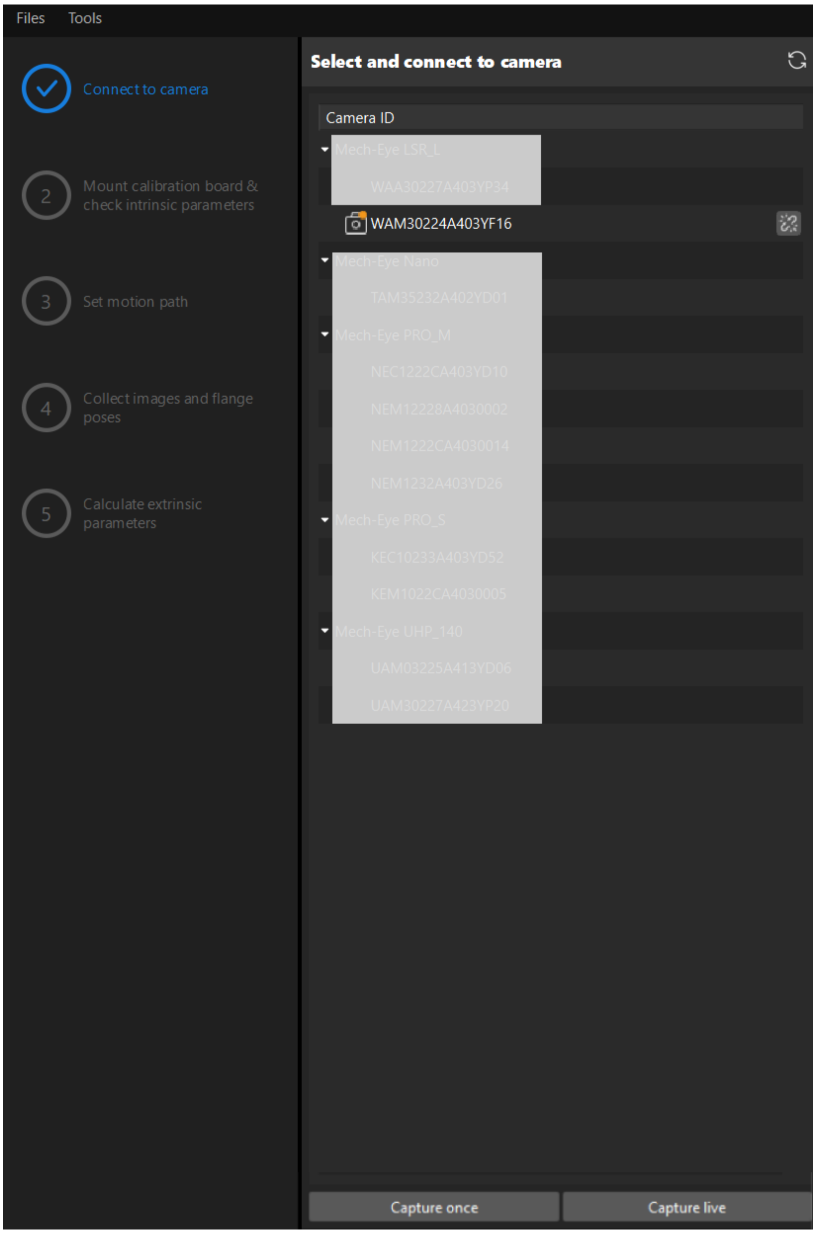
Calibration Procedure

Connect to the Camera

1. In the **Connect to camera** step, select the camera to connect in the **Camera ID** list, and then click the 🔗 button or double-click the camera entry to connect to it.



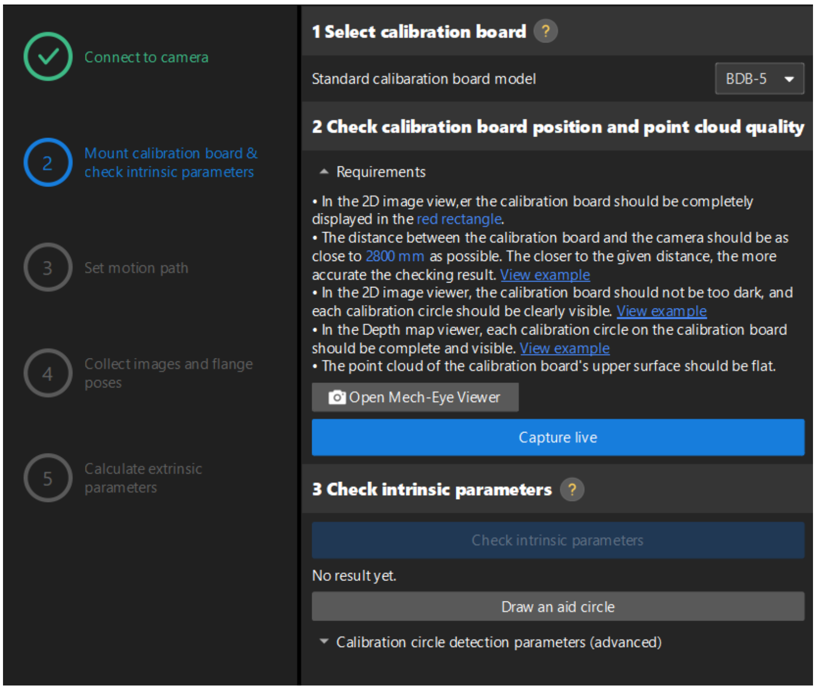
2. After the camera is connected, click the **[Capture once]** or **[Capture live]** button.



3. In the **Image viewer** panel, ensure that the camera can capture images normally and click the **[Next]** button on the bottom bar.

Mount Calibration Board & Check Intrinsic Parameters

1. In the **Mount calibration board & check intrinsic parameters** step, select the calibration board model for the **Standard calibration board model** parameter according to its model nameplate in the **1 Select calibration board** area.



Connect to camera

2 Mount calibration board & check intrinsic parameters

3 Set motion path

4 Collect images and flange poses

5 Calculate extrinsic parameters

1 Select calibration board ?

Standard calibaration board model BDB-5

2 Check calibration board position and point cloud quality

Requirements

- In the 2D image view,er the calibration board should be completely displayed in the red rectangle.
- The distance between the calibration board and the camera should be as close to 2800 mm as possible. The closer to the given distance, the more accurate the checking result. View example
- In the 2D image viewer, the calibration board should not be too dark, and each calibration circle should be clearly visible. View example
- In the Depth map viewer, each calibration circle on the calibration board should be complete and visible. View example
- The point cloud of the calibration board's upper surface should be flat.

Open Mech-Eye Viewer

Capture live

3 Check intrinsic parameters ?

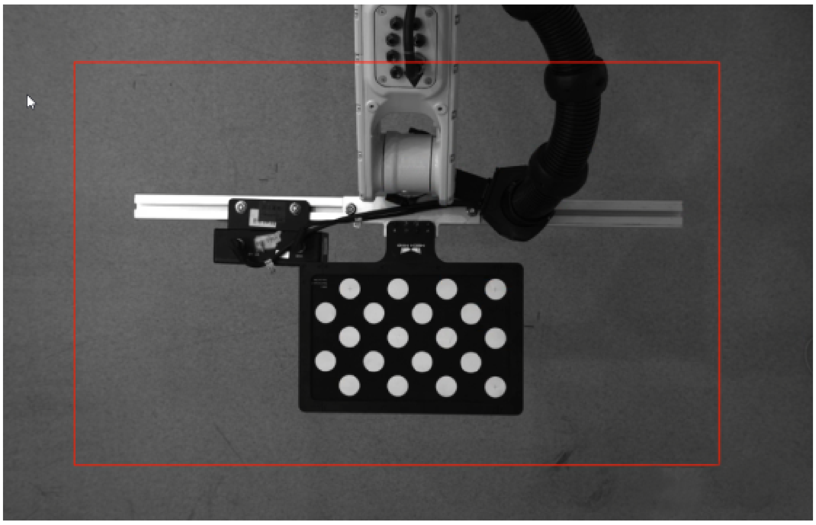
Check intrinsic parameters

No result yet.

Draw an aid circle

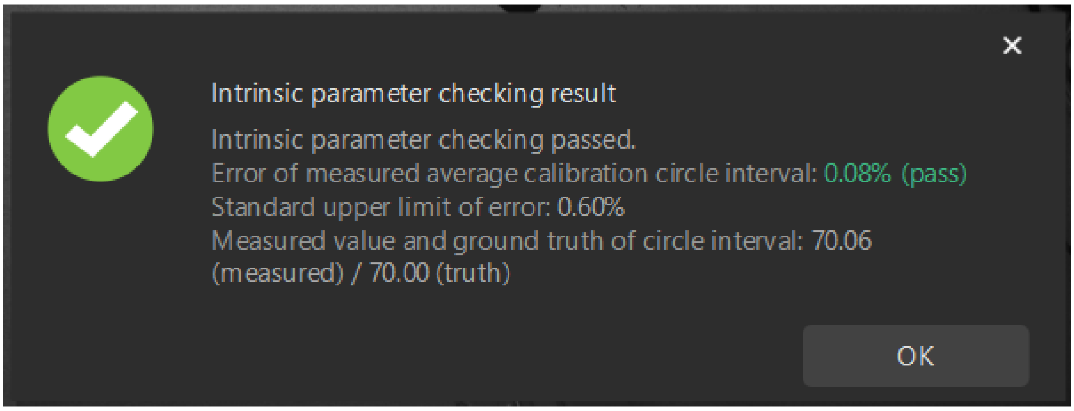
Calibration circle detection parameters (advanced)

2. In the **2 Check calibration board position and point cloud quality** area, read carefully the requirements on the calibration board position and point cloud quality, and then click the **[Capture live]** button.The **[Capture live]** button will turn into **[Stop capturing and detect position]**.
3. Control the robot to move the calibration board to the proper position, and ensure that the 2D image and depth map of the calibration board meet the requirements, and then click the **[Stop capturing and detect position]** button.



If the captured images do not meet the requirements, click **[Open Mech-Eye Viewer]** button to open the Mech-Eye Viewer software, adjust the 2D and 3D exposure parameters and re-capture images. Please note that you need to change the **Parameter group** parameter to “calib” first.

4. In the **3 Check intrinsic parameters** area, click the **[Check intrinsic parameters]** button.
5. Confirm the results of the camera intrinsic parameter check.
   - If the camera intrinsic parameter check passes, click the **[OK]** button in the prompted window, and then click the **[Next]** button on the bottom bar.



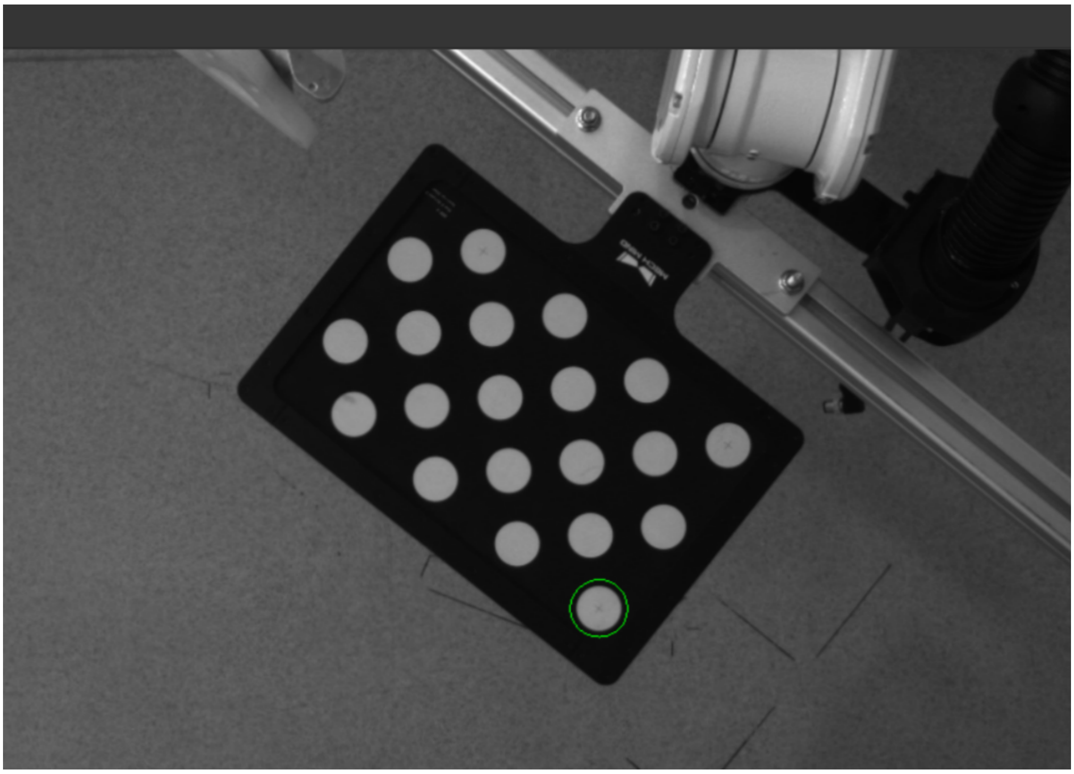
Intrinsic parameter checking result

Intrinsic parameter checking passed.
Error of measured average calibration circle interval: 0.08% (pass)
Standard upper limit of error: 0.60%
Measured value and ground truth of circle interval: 70.06 (measured) / 70.00 (truth)

- If the camera intrinsic parameter check fails, adjust detection parameters by drawing aid circles or manually adjusting the calibration circle detection parameters, and then click the **[Recheck intrinsic parameters]** button.

Draw an Aid Circle

1. To draw an aid circle, click the **[Draw an aid circle]** button.
2. In the right **Image viewer** panel, right-click the calibration board image, clear the **Fit to window** checkbox, press the `Ctrl` key and drag the roller to adjust the image to a suitable size.
3. Move the mouse pointer to the cross center point of the calibration circle, press the left mouse button and make the aid circle completely include the calibration circle and then release it.



4. Click the **[Recheck intrinsic parameters]** button, and confirm that the camera intrinsic parameter check passes.

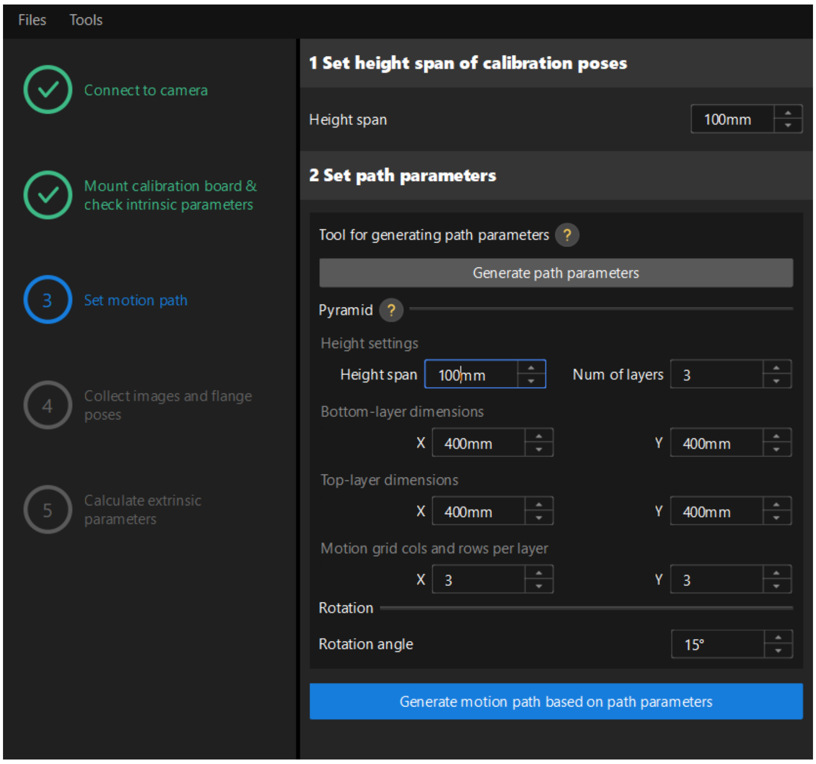
Manually Adjust Detection Parameters

To manually adjust detection parameters, click **[Calibration circle detection parameters (advanced)]**, adjust parameter settings and click the **[Recheck intrinsic parameters]** to confirm whether the camera intrinsic parameter check passes.

If the calibration circles are still not detected, please adjust the camera settings according to the on-site operating conditions.

Set Motion Path

1. In the **Set motion path** step, set the **Height span** parameter in the **1 Set height span of calibration poses** area.


Files
Tools
Connect to camera
Mount calibration board & check intrinsic parameters
3
Set motion path
4
Collect images and flange poses
5
Calculate extrinsic parameters
Generate path parameters
Generate motion path based on path parameters

1 Set height span of calibration poses

Height span: 100mm

2 Set path parameters

Tool for generating path parameters

Pyramid

Height settings

Height span: 100mm; Num of layers: 3

Bottom-layer dimensions

X: 400mm; Y: 400mm

Top-layer dimensions

X: 400mm; Y: 400mm

Motion grid cols and rows per layer

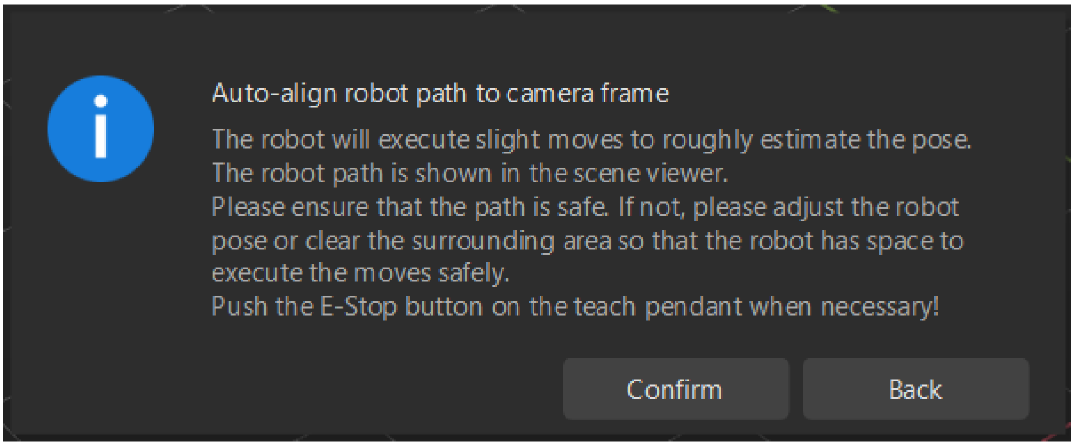
X: 3; Y: 3

Rotation

Rotation angle: 15°

The **Height span** parameter should be set according to the recommended working distance range of the camera and the size of the robot’s working space.

2. Click the **[Generate path parameters]** button, and click the **[OK]** button in the prompted window.

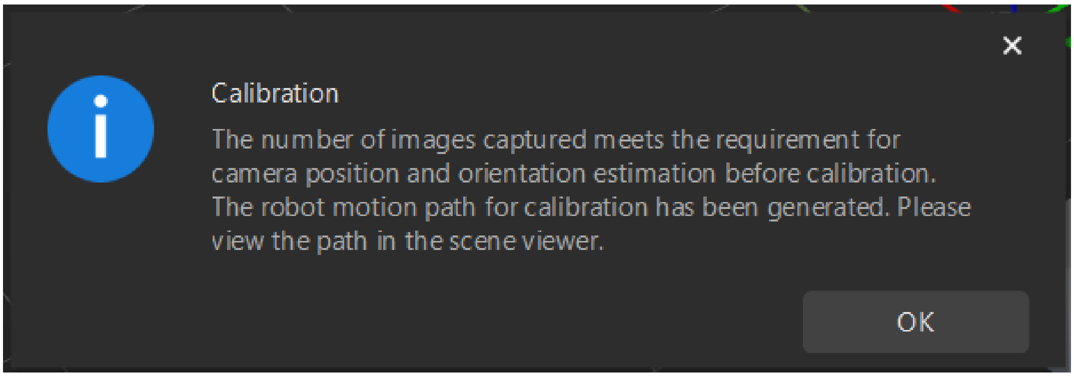


Auto-align robot path to camera frame

The robot will execute slight moves to roughly estimate the pose.
The robot path is shown in the scene viewer.
Please ensure that the path is safe. If not, please adjust the robot pose or clear the surrounding area so that the robot has space to execute the moves safely.
Push the E-Stop button on the teach pendant when necessary!

This operation will slightly move the robot and capture images. The whole process may take 10 to 15 seconds. Please ensure that the motion path is safe. In case of emergency, please tap the emergency stop button on the robot teach pendant to stop the robot immediately.

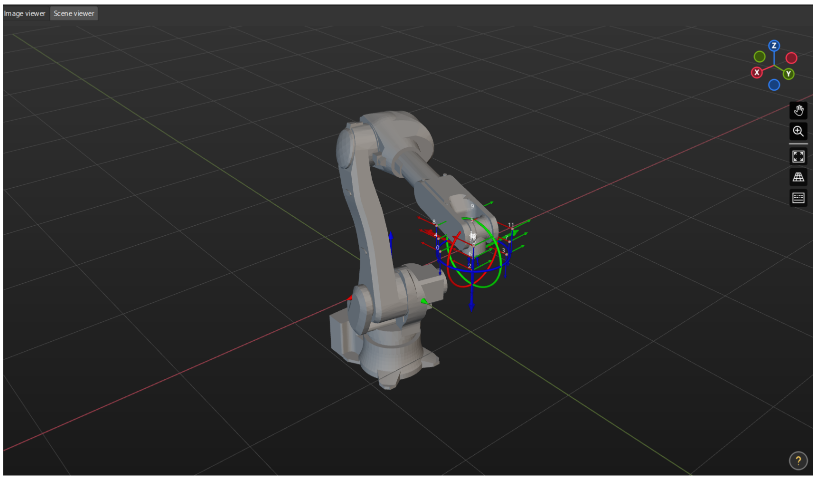
3. In the prompted **Calibrate** window, click the **[OK]** button.



Calibration

The number of images captured meets the requirement for camera position and orientation estimation before calibration. The robot motion path for calibration has been generated. Please view the path in the scene viewer.

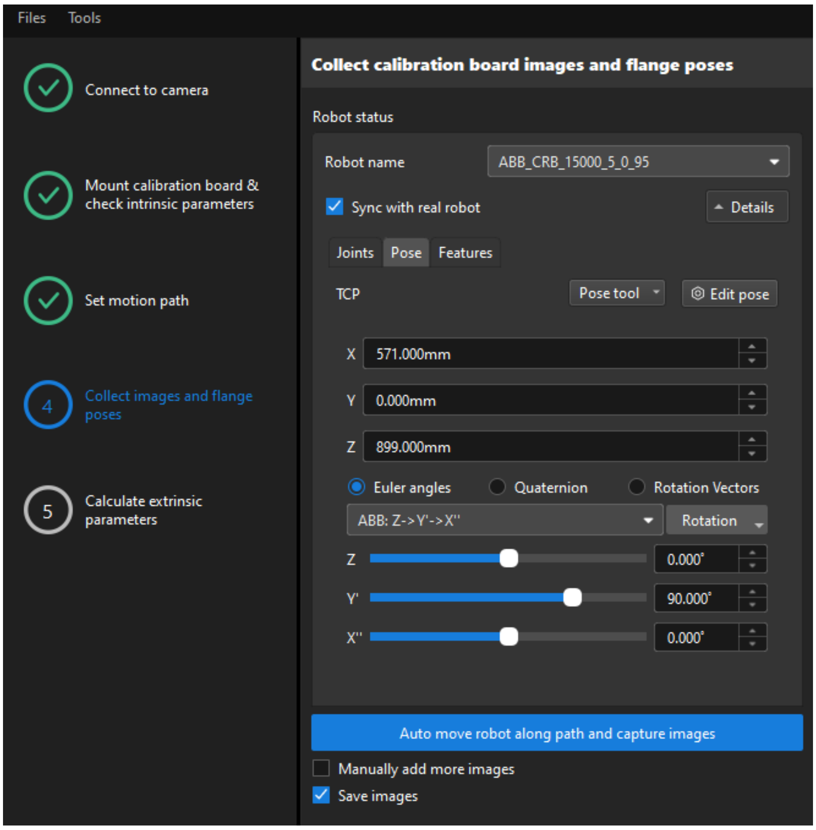
4. In the right **Scene viewer** panel, confirm that waypoints are reasonable and will not collide with obstacles.



5. Click the **[Generate motion path based on path parameters]** button and then click the **[Next]** button on the bottom bar.

Collect Images and Flange Poses

1. In the **Collect images and flange poses** step, select the **Save images** checkbox.


Files
Tools
Connect to camera
Mount calibration board & check intrinsic parameters
Set motion path
4
Collect images and flange poses
5
Calculate extrinsic parameters
Details
Joints
Pose
Features
Pose tool
Edit pose
Rotation
Auto move robot along path and capture images

Collect calibration board images and flange poses

Robot status

Robot name: ABB_CRB_15000_5_0_95

☑ Sync with real robot

TCP

X 571.000mm

Y 0.000mm

Z 899.000mm

● Euler angles ○ Quaternion ○ Rotation Vectors

ABB: Z->Y'->X''

Z 0.000°

Y' 90.000°

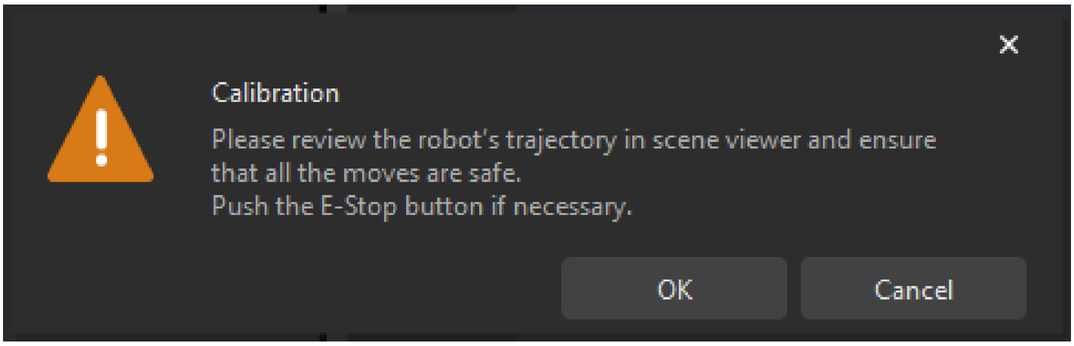
X'' 0.000°

☐ Manually add more images

☑ Save images

2. If the robot uses the Master-Control to communicate with the Vision System, select the **Sync with real robot** checkbox (selected by default, recommended), and click **Details** to confirm that the robot's JPs and pose data are the same as those on the teach pendant. Clear the checkbox if you do not need to synchronize the data with the real robot.

3. Click the **[Auto move robot along path and capture images]** button, read the safety window carefully, and then click the **[OK]** button.



Calibration

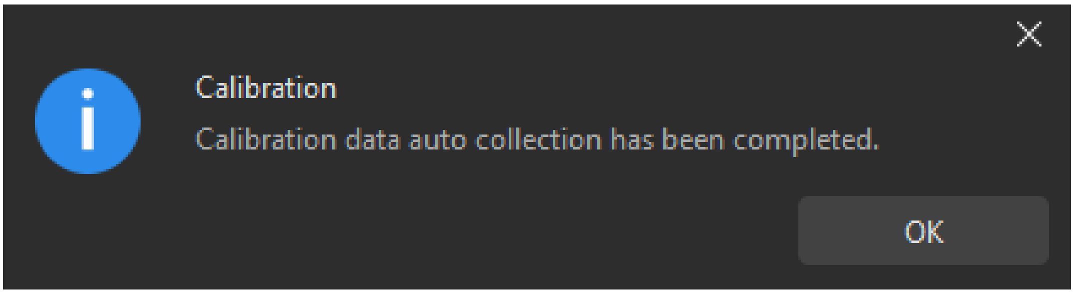
Please review the robot's trajectory in scene viewer and ensure that all the moves are safe.
Push the E-Stop button if necessary.

4. Wait until the robot finishes moving along the preset path and the camera finishes capturing images on all waypoints. In the right **Image viewer** panel, all the captured images are displayed.

- Please stay away from the robot working area to keep safe.
- Clicking the **[Stop robot]** button can exit the calibration process. But the robot will not stop until it finishes the current waypoint. In case of emergency, please tap the emergency stop button on the robot teach pendant to stop the robot immediately

(the robot needs to be reconnected after tapping the emergency stop button).

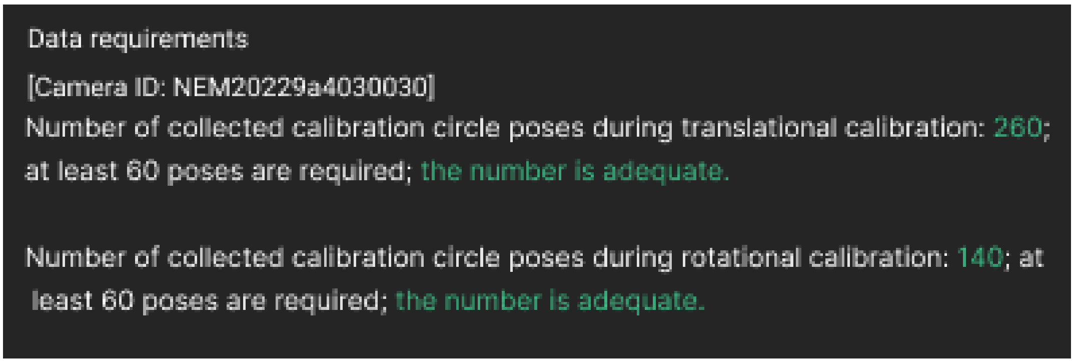
5. After all images are captured, click the [OK] button in the prompted window.



Calibration

Calibration data auto collection has been completed.

6. Confirm that the collected calibration data meets the data requirements, and then click the [Next] button on the bottom bar.



Data requirements

[Camera ID: NEM20229a4030030]

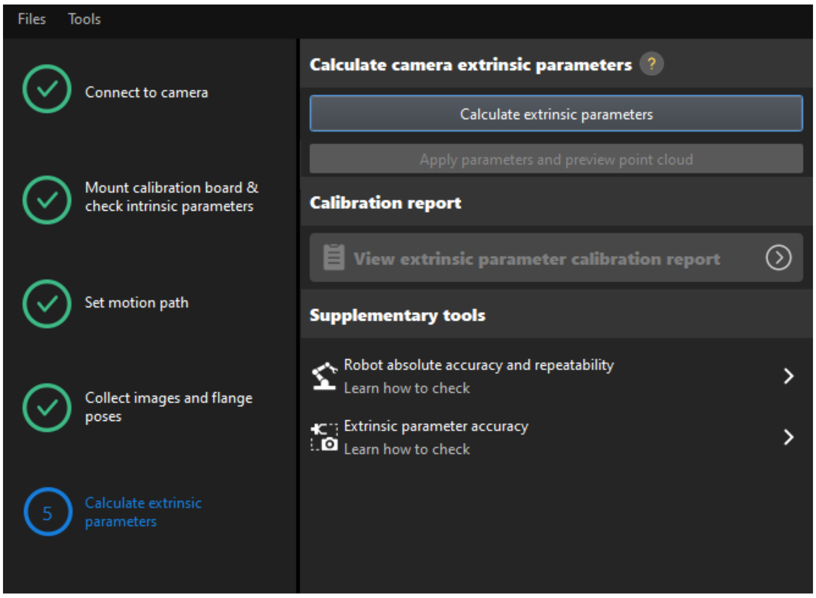
Number of collected calibration circle poses during translational calibration: 260; at least 60 poses are required; the number is adequate.

Number of collected calibration circle poses during rotational calibration: 140; at least 60 poses are required; the number is adequate.

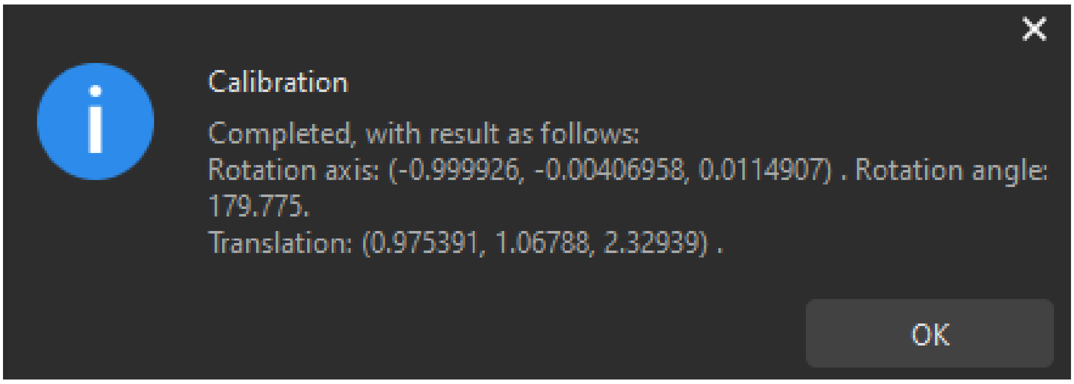
If the requirements are not met, please manually move the robot (through the teach pendant or Mech-Viz), select the [Manually add more images] checkbox, and click the [Add an image and record flange pose] button to add the calibration board image and enter the robot flange pose.

Calculate Extrinsic Parameters

1. In the **Calculate extrinsic parameters** step, click the [Calculate extrinsic parameters] button.


Files
Tools
Connect to camera
Mount calibration board & check intrinsic parameters
Set motion path
Collect images and flange poses
5
Calculate extrinsic parameters
Calculate camera extrinsic parameters
Calculate extrinsic parameters
Apply parameters and preview point cloud
Calibration report
View extrinsic parameter calibration report
Supplementary tools
Robot absolute accuracy and repeatability
Learn how to check
Extrinsic parameter accuracy
Learn how to check

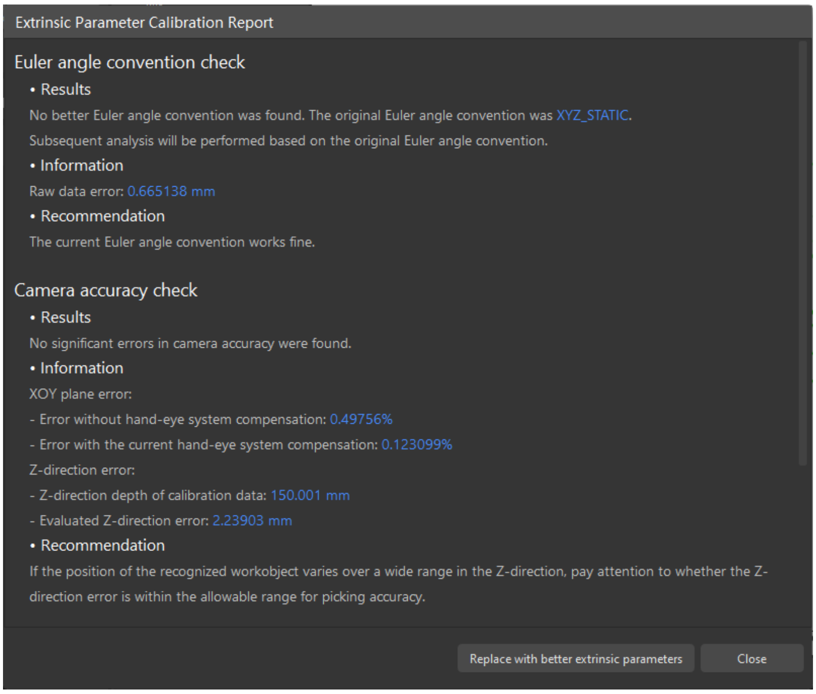
2. In the prompted window indicating calibration success, click the **OK** button.



Calibration

Completed, with result as follows:
Rotation axis: (-0.999926, -0.00406958, 0.0114907) . Rotation angle: 179.775.
Translation: (0.975391, 1.06788, 2.32939) .

3. Click the **[View extrinsic parameter calibration report]** button. The extrinsic parameter calibration report will be automatically displayed after it is generated.



Extrinsic Parameter Calibration Report

Euler angle convention check

- Results

No better Euler angle convention was found. The original Euler angle convention was XYZ_STATIC.
Subsequent analysis will be performed based on the original Euler angle convention.

- Information

Raw data error: 0.665138 mm

- Recommendation

The current Euler angle convention works fine.

Camera accuracy check

- Results

No significant errors in camera accuracy were found.

- Information

XOY plane error:
- Error without hand-eye system compensation: 0.49756%
- Error with the current hand-eye system compensation: 0.123099%

Z-direction error:
- Z-direction depth of calibration data: 150.001 mm
- Evaluated Z-direction error: 2.23903 mm

- Recommendation

If the position of the recognized workobject varies over a wide range in the Z-direction, pay attention to whether the Z-direction error is within the allowable range for picking accuracy.

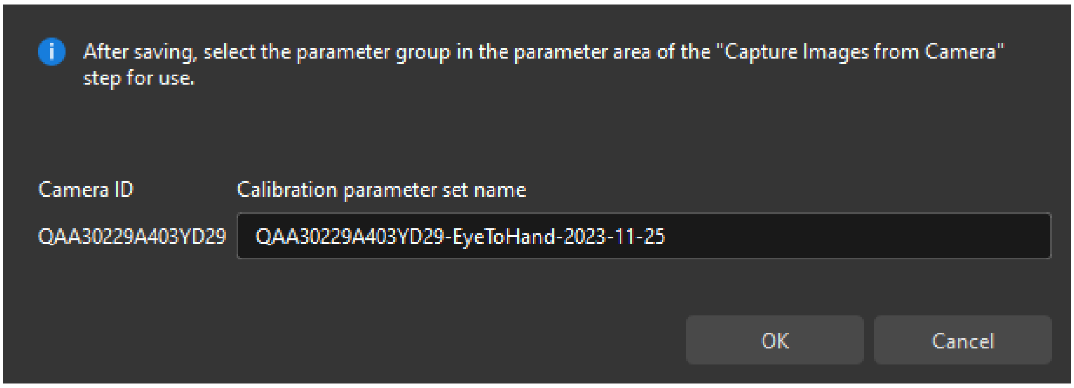
The extrinsic parameter calibration report displays the results of the Euler angle convention check, camera accuracy check, and robot absolute accuracy check.

4. Confirm the results of the Euler angle convention check, camera accuracy check, and robot absolute accuracy check in the report, and then click the **[Close]** button to close the report.

   - If the camera accuracy in the report does not meet the requirements, please follow the recommendation to troubleshoot the camera accuracy errors. If you want to enhance the calibration accuracy, please refer to the section Calibration Result Check and Analysis.
   - To evaluate the robot absolute accuracy and the accuracy of calculated extrinsic parameters, you can click **[Robot absolute accuracy and repeatability]** and **[Extrinsic parameter accuracy]** in the “Calculate extrinsic parameters” window and use the error analysis tool to perform robot absolute accuracy check and analyze extrinsic parameter error in ETH setup.

5. Click the **[Save]** button on the bottom bar. In the prompted **Save Calibration Files** dialog box, click the **[OK]** button. The camera calibration result will be automatically saved in the

"calibration" directory of the project.



After saving, select the parameter group in the parameter area of the "Capture Images from Camera" step for use.

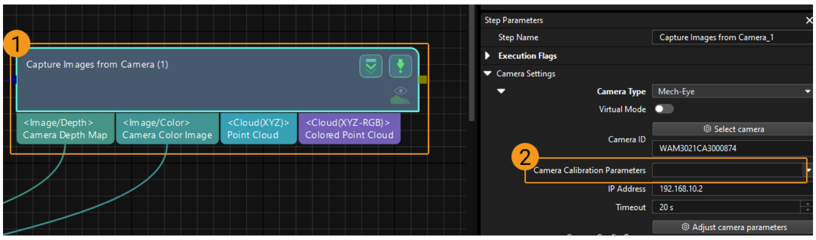
Camera ID	Calibration parameter set name
QAA30229A403YD29	QAA30229A403YD29-EyeToHand-2023-11-25

Till now, the calibration process is completed.

Change the Extrinsic Parameter File

If the camera's extrinsic parameter file has been used by a Mech-Vision project, you need to change the extrinsic parameter file used by the Mech-Vision project to a new one after you finish the hand-eye calibration.

1. Select the **Capture Images from Camera** Step.



1
Capture Images from Camera (1)
<Image/Depth> Camera Depth Map
<Image/Color> Camera Color Image
<Cloud(XYZ)> Point Cloud
<Cloud(XYZ-RGB)> Colored Point Cloud
2
Execution Flags
Camera Settings
Select camera
Adjust camera parameters

Step Parameters

Step Name	Capture Images from Camera_1
Camera Type	Mech-Eye
Virtual Mode	
Camera ID	WAM3021CA3000874
Camera Calibration Parameters	
IP Address	192.168.10.2
Timeout	20 s

2. In the **Step Parameters** panel, click ▾ of the **Camera Calibration Parameters** parameter, and select the newly calibrated extrinsic parameter file.

Calibration-Related Status Codes

If the robot uses the Standard Interface to communicate with the Vision System during hand-eye calibration, the Vision System will return status codes in response data to indicate the execution status of the commands. The status codes indicate both the normal execution results and execution failures. You can perform further troubleshooting according to the status codes.

Calibration Error Codes

Error code	Description
7001	Calibration: Parameter error
7002	Calibration: Calibration flange pose not provided by Mech-Vision
7003	Calibration: Calibration joint positions not provided by Mech-Vision
7004	Calibration: Robot failed to reach calibration point

Calibration Normal Status Codes

Normal Status Code	Description
7100	Calibration: Robot reaches calibration point successfully
7101	Calibration: Pose received from Mech-Vision successfully

4.5.2.2. Automatic Calibration in the Eye-to-Hand Setup (Four-Axis Robots)

This how-to guide introduces how to complete the automatic calibration for four-axis robots in the eye-to-hand (ETH) setup.

Preparation before Calibration

Before hand-eye calibration, you need to finish the following preparations:

- Construct the Mech-Mind Vision System.
- Prepare the materials required for calibration.
- Adjust the effect of calibration board’s point cloud.
- Complete the robot communication configuration.

Construct the Vision System

Construct the Mech-Mind Vision System by referring to the section Vision System Hardware Setup.

You need to use Mech-Eye Viewer, Mech-Vision and Mech-Viz during hand-eye calibration. Please ensure that they have been installed and are running the latest versions.

Prepare the Materials Required for Calibration

The automatic calibration in the ETH setup needs to use the calibration board.

Please prepare the calibration board according to the following requirements:

- Ensure that the circles of the calibration board are clearly visible and without obvious scratches, and the board does not suffer from deformations.
- In the ETH setup, mount the robot-specific bracket for calibration board onto the robot flange, and then mount the calibration board onto the bracket. Make sure that the calibration board is attached securely, and that the board is parallel to the XY plane of the robot flange frame.

> If an undetachable gripper is connected to the robot flange, you can attach the calibration board directly to the gripper.

In addition, before calibration, move the robot to the starting point for calibration. For the automatic calibration in the ETH setup, the starting point is the bottom center of the camera’s field of view (the robot will move the calibration board upwards during calibration).

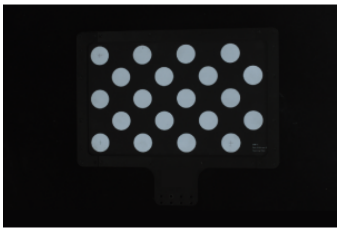
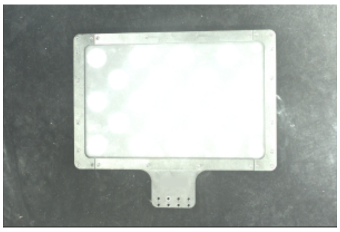
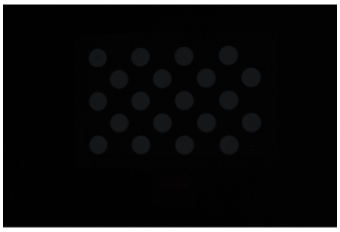
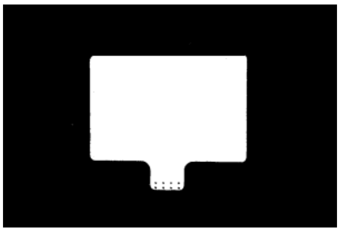
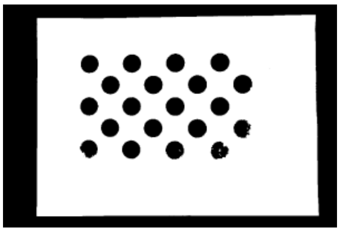
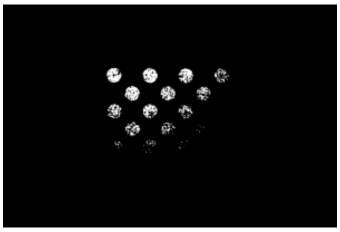
Adjust the Effect of Calibration Board’s Point Cloud

1. Open Mech-Eye Viewer to adjust camera parameters.
2. Adjust the 2D parameters to make sure that the 2D image of the calibration board is clear and neither overexposed nor underexposed.

3. Adjust 3D parameters to make sure that the point clouds of the circles on the calibration board are complete and have clear contours. It is recommended to set **Surface Smoothing** and **Outlier Removal** to **Normal** in the **Point Cloud Processing** section to reduce the point cloud fluctuation range.

   > If the on-site ambient lights are not ideal and affect the quality of 2D images and point clouds, you can use shading or supplemental light to improve the lighting conditions.

4. Make sure that the images and point cloud of the calibration board are up to standard after completing the preceding steps.

	Normal	Overexposed	Underexposed
2D image			
Point cloud			

Complete the Robot Communication Configuration

If the robot uses the Standard Interface to communicate with the Vision System, please set up the Standard Interface communication with the robot. For details, refer to the section Standard Interface Communication.

If the robot uses the Master-Control to communicate with the Vision System, please set up the Master-Control communication with the robot. For details, refer to the section Master-Control Communication.

Pre-calibration Configuration

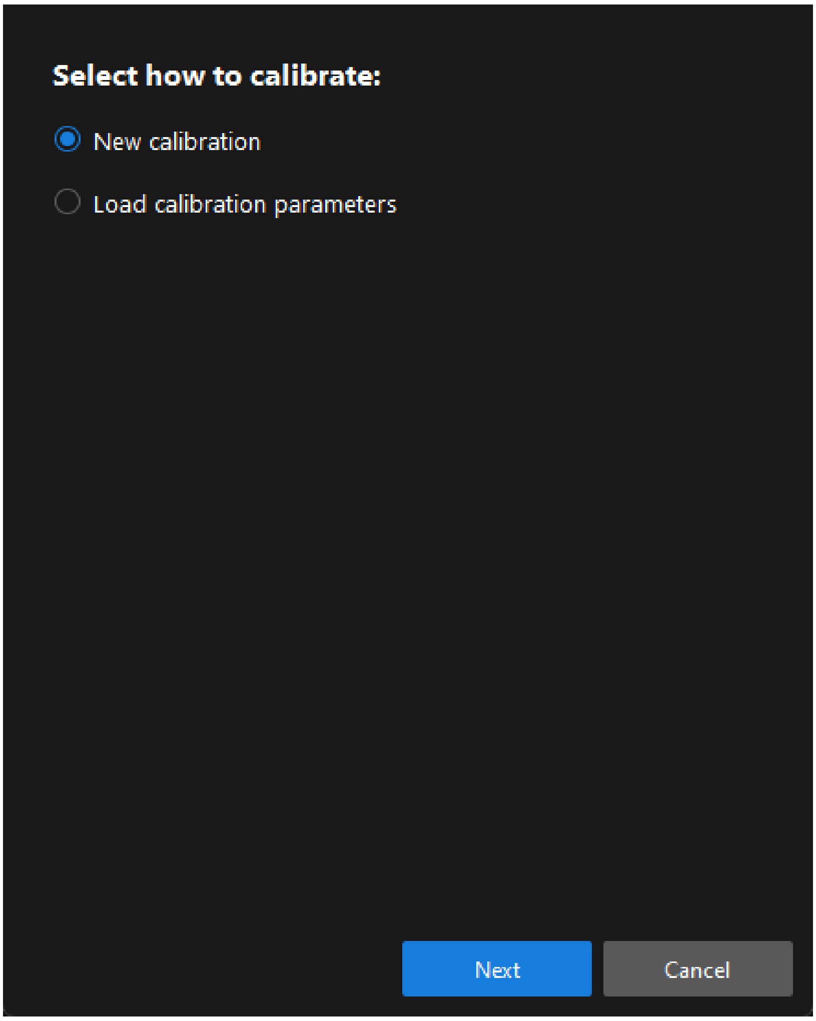
If the robot uses the Standard Interface to communicate with the Vision System, complete Pre-calibration Configuration (Standard Interface).

If the robot uses the Master-Control to communicate with the Vision System, complete Pre-calibration Configuration (Master-Control).

Pre-calibration Configuration (Standard Interface)

If the Standard Interface is used to communicate with the robot, follow these steps:

1. Open Mech-Vision, and click the **[Camera Calibration (Standard)]** button in the toolbar.The **Configuration before Calibration** window will be prompted.
2. In the **Select how to calibrate** window, select the **New calibration** radio button, and then click the **[Next]** button.

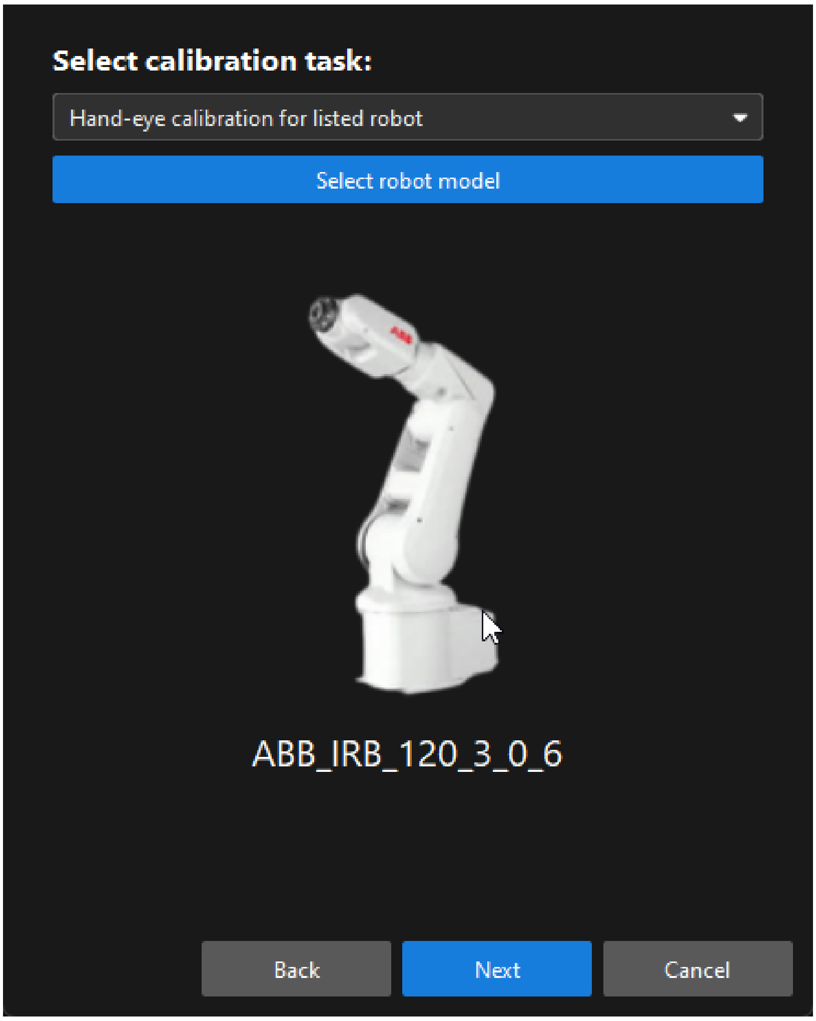


Select how to calibrate:

● New calibration

○ Load calibration parameters

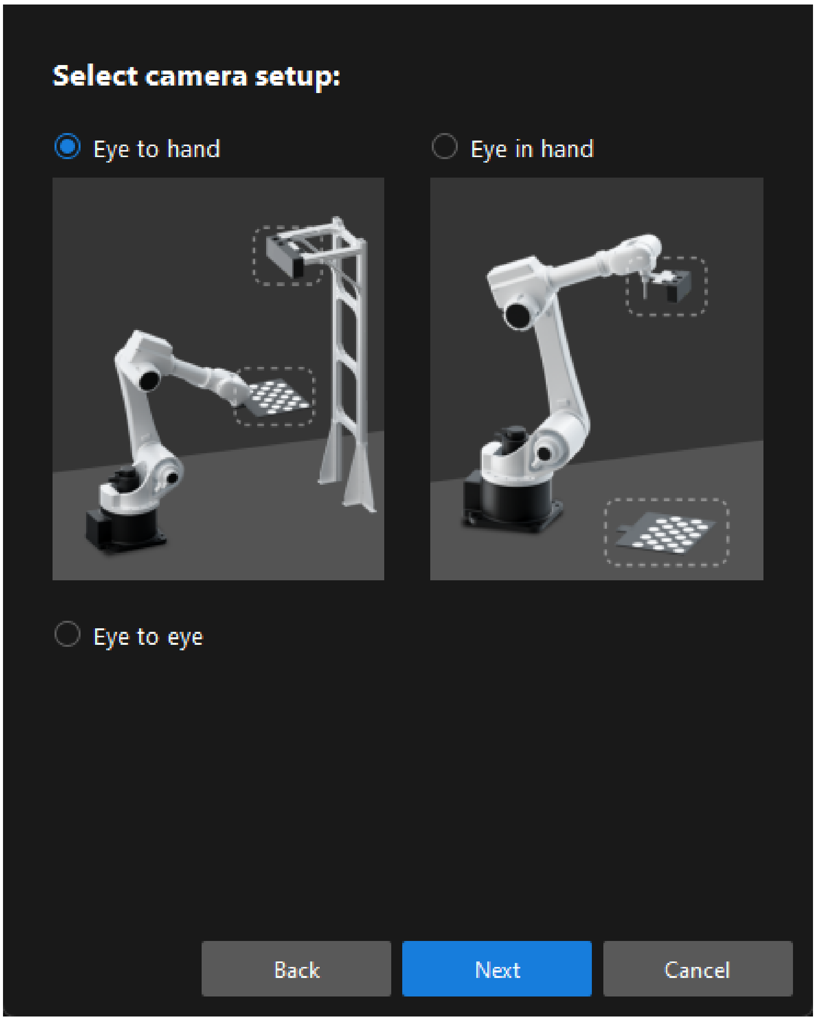
3. In the **Select calibration task** window, select **Hand-eye calibration for listed robot** from the drop-down list box, click the **[Select robot model]** button to select the robot model used by the project, and then click the **[Next]** button.



Select calibration task:

Hand-eye calibration for listed robot

4. In the **Select camera setup** window, select the **Eye to hand** radio button, and then click the **[Next]** button.



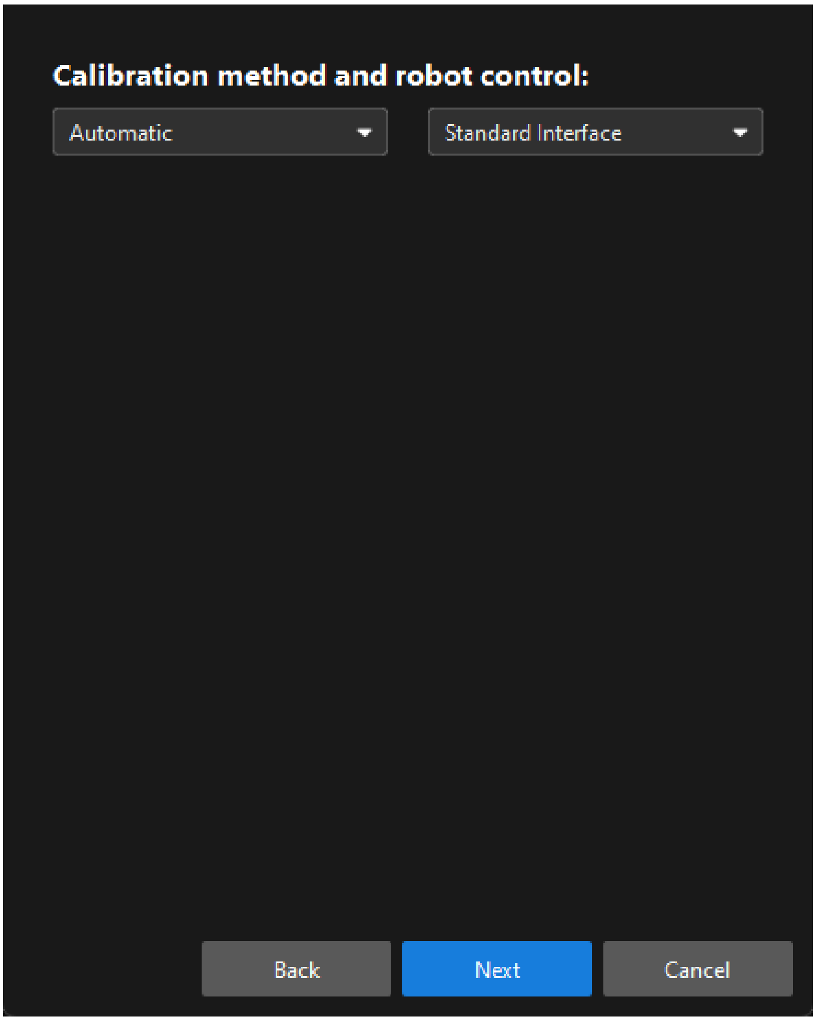
Select camera setup:

● Eye to hand

○ Eye in hand

○ Eye to eye

5. In the **Calibration method and robot control** window, select **Automatic** and **Standard Interface**, and then click the **[Next]** button.

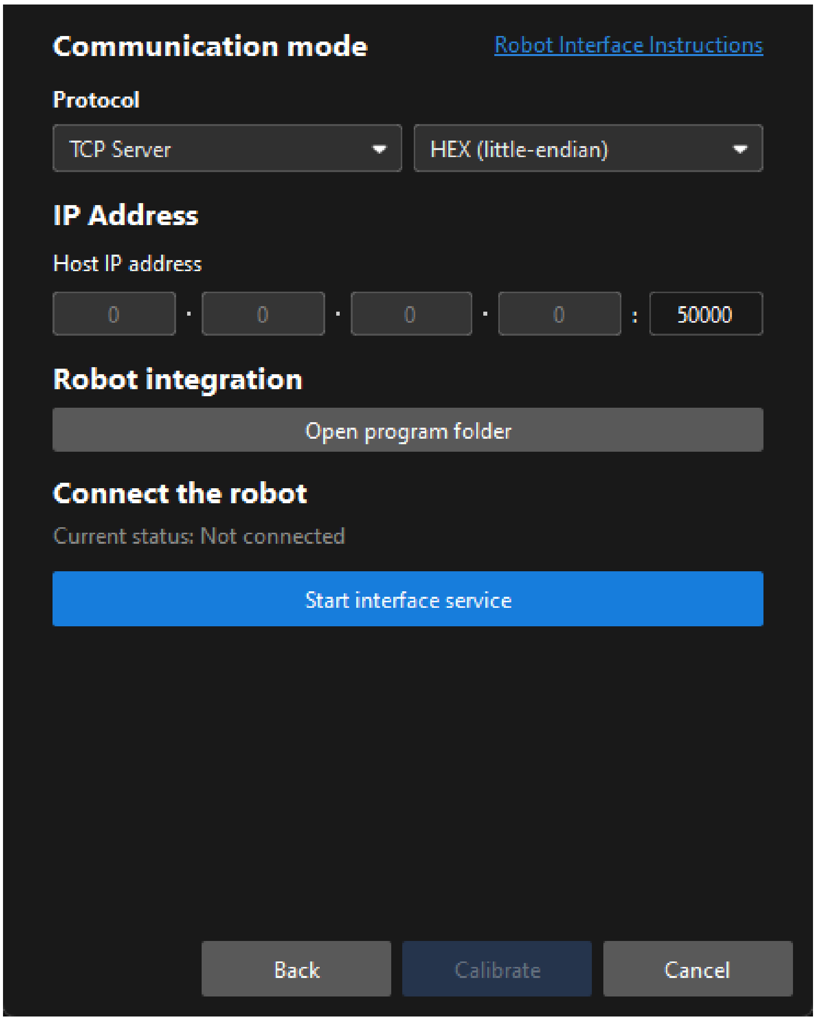


Calibration method and robot control:

Automatic

Standard Interface

6. In the new window, set parameters **Protocol** and **Host IP address**, and then click the **[Start interface service]** button. The button will turn into **Waiting for the robot to connect...**



Communication mode

Protocol

TCP Server

HEX (little-endian)

IP Address

Host IP address

0 . 0 . 0 . 0 : 50000

Robot integration

Connect the robot

Current status: Not connected

- In the preceding figure, **TCP Server** and **Hex (Little-endian)** are only the **Protocol** setting example of the ABB robot. The protocol settings of other robot brands are different. The software will automatically set this parameter and you just need to keep the default setting.
- The port number for the **Host IP address** parameter is **50000** by default. If the port number determined in the communication solution is not 50000, please change it as required.

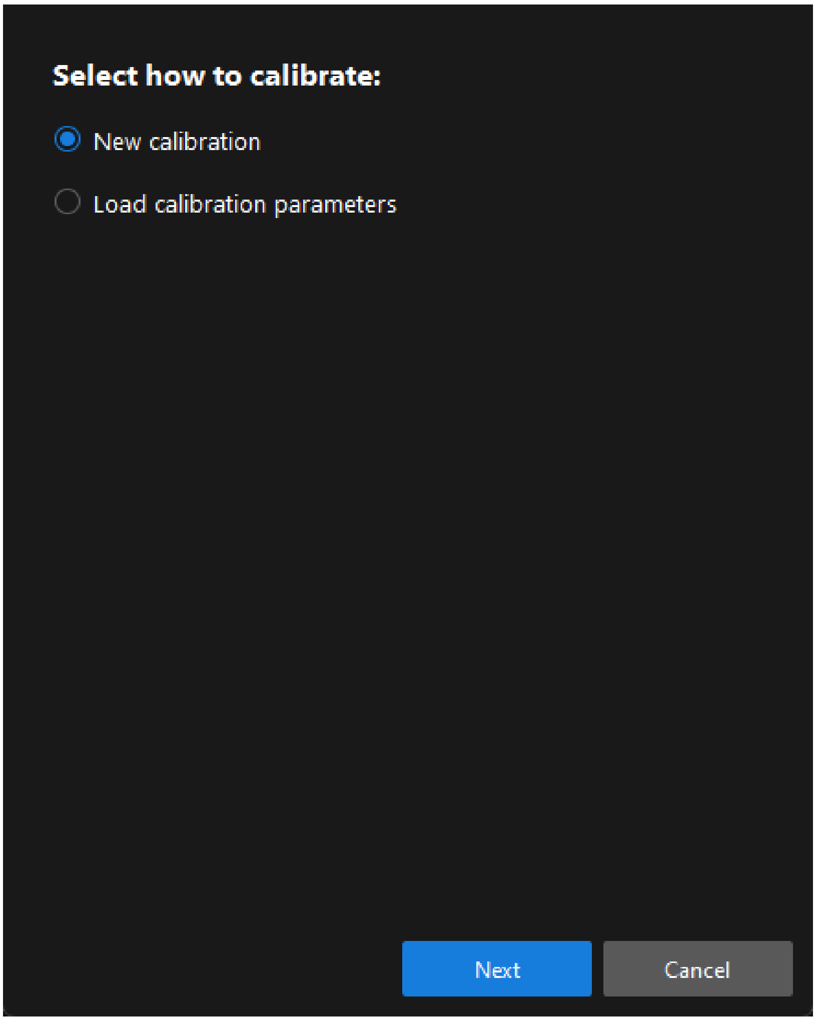
7. On the robot teach pendant, **select the automatic calibration program, teach the start point and run the program**. For details on the calibration programs of different robots, refer to the section Standard Interface Communication. After the program is started, the log message "Entering the calibration process, please start the calibration in Mech-Vision" will be printed in the Console log panel.
8. Return to Mech-Vision, confirm that "Connected" status message is displayed in the **Connect the robot** area, and click the **[Calibrate]** button. The **Calibration (Eye to hand)** window will be prompted.

Pre-calibration Configuration (Master-Control)

If the Master-Control is used to communicate with the robot, follow these steps:

1. Open Mech-Vision, and click the **[Camera Calibration (Standard)]** button in the toolbar. The **Configuration before Calibration** window will be prompted.

2. In the **Select how to calibrate** window, select the **New calibration** radio button, and then click the **[Next]** button.

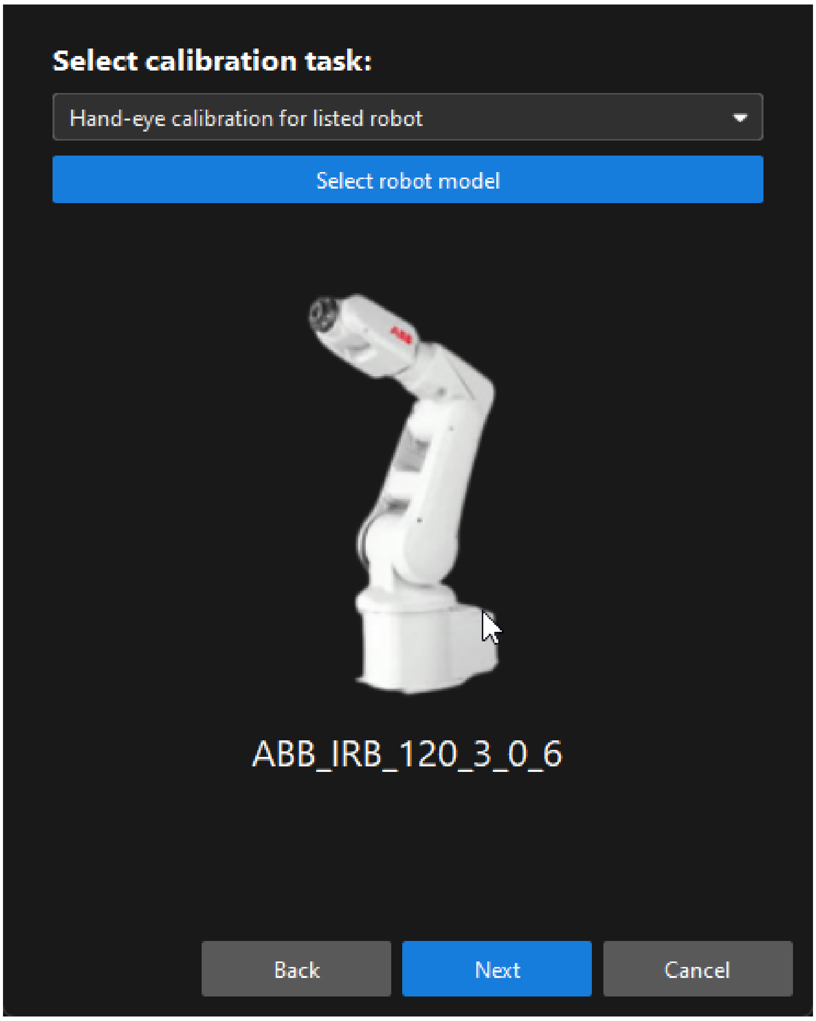


Select how to calibrate:

● New calibration

○ Load calibration parameters

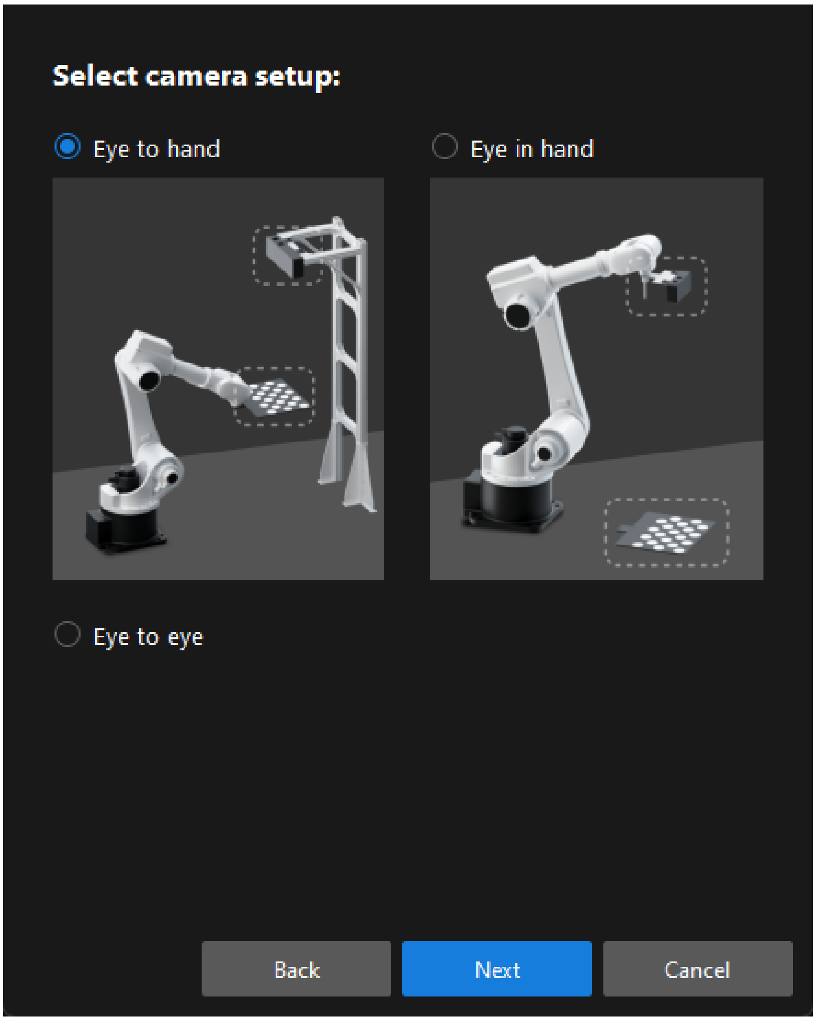
3. In the **Select calibration task** window, select **Hand-eye calibration for listed robot** from the drop-down list box, click the **[Select robot model]** button to select the robot model used by the project, and then click the **[Next]** button.



Select calibration task:

Hand-eye calibration for listed robot

4. In the **Select camera setup** window, select the **Eye to hand** radio button, and then click the **[Next]** button.



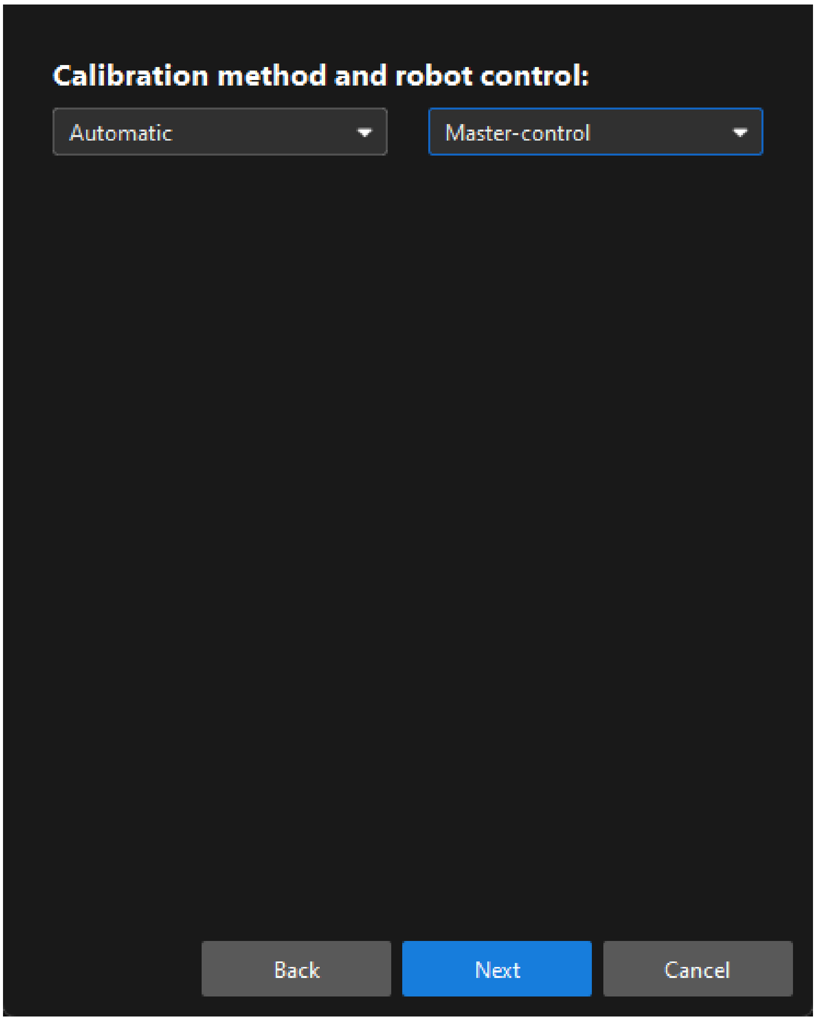
Select camera setup:

● Eye to hand

○ Eye in hand

○ Eye to eye

5. In the **Calibration method and robot control** window, select **Automatic** and **Master-control**, and then click the **[Next]** button.

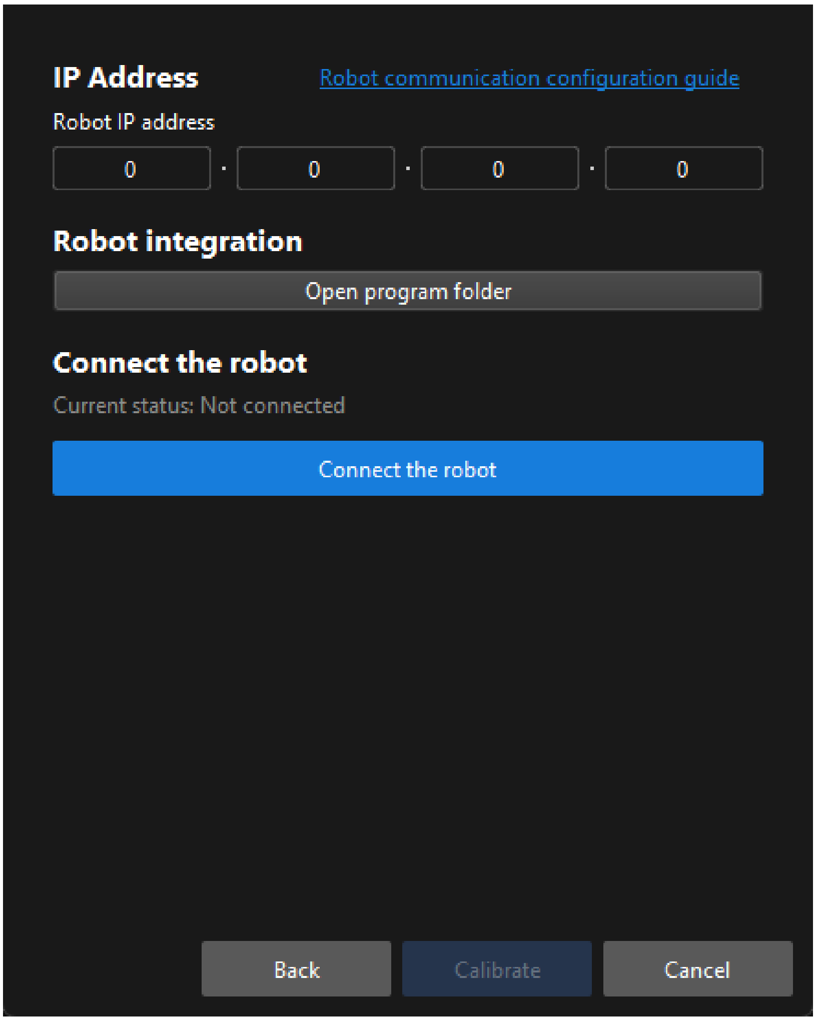


Calibration method and robot control:

Automatic

Master-control

6. In the prompted window, set the **Robot IP address** parameter.



IP Address

Robot IP address

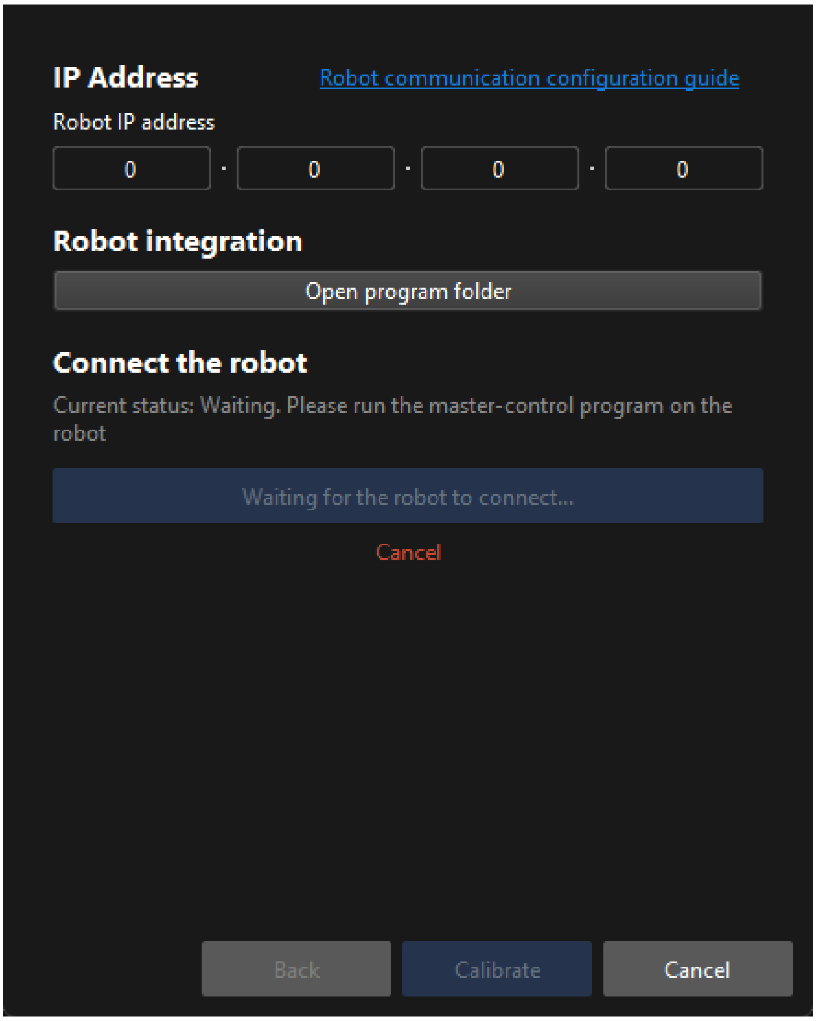
0 . 0 . 0 . 0

Robot integration

Current status: Not connected

7. On the robot teach pendant, **select and run the Master-control program**. For the instructions of different robots, refer to the section Master-Control Communication.

8. Return to Mech-Vision, and click the **[Connect the robot]** button in the **Connect the robot** area. The button will turn into **Waiting for the robot to connect...**



IP Address

Robot communication configuration guide

Robot IP address

0 . 0 . 0 . 0

Robot integration

Open program folder

Connect the robot

Current status: Waiting. Please run the master-control program on the robot

Waiting for the robot to connect...

Cancel

Back Calibrate Cancel

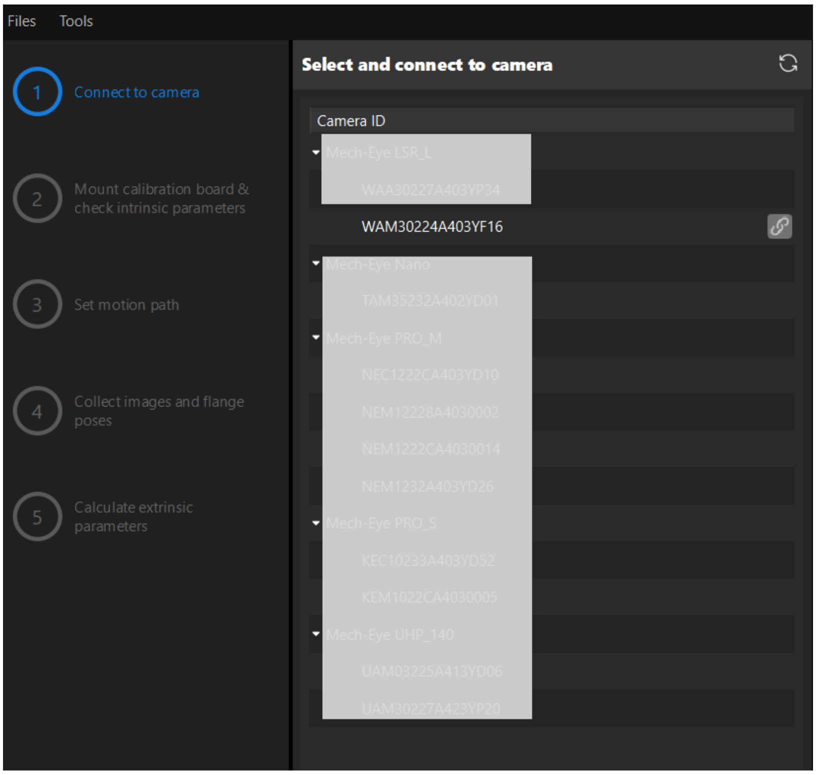
9. Wait until the "Connected" status message is displayed in the **Connect the robot** area, and then click the **[Calibrate]** button.The **Calibration (Eye to hand)** window will be prompted.

Till now, you have completed the pre-calibration configuration and can start the calibration procedure.

Calibration Procedure

Connect to the Camera

1. In the **Connect to camera** step, select the camera to connect in the **Camera ID** list, and then click the 🔗 button or double-click the camera entry to connect to it.



Files Tools

1 Connect to camera

2 Mount calibration board & check intrinsic parameters

3 Set motion path

4 Collect images and flange poses

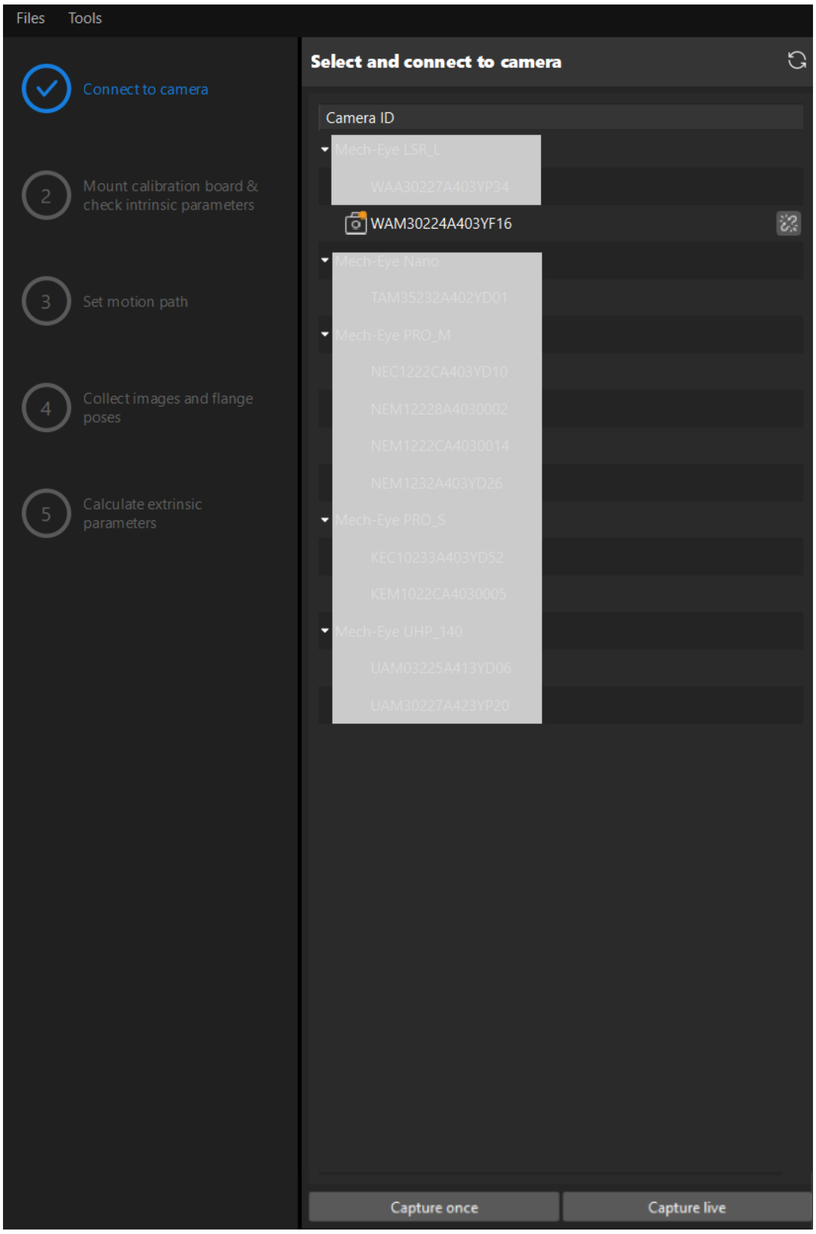
5 Calculate extrinsic parameters

Select and connect to camera

Camera ID

- Mech-Eye LSR_L
  - WAA30227A403YP34
  - WAM30224A403YF16
- Mech-Eye Nano
  - TAM35232A402YD01
- Mech-Eye PRO_M
  - NEC1222CA403YD10
  - NEM12228A4030002
  - NEM1222CA4030014
  - NEM1232A403YD26
- Mech-Eye PRO_S
  - KEC10233A403YD52
  - KEM1022CA4030005
- Mech-Eye UHP_140
  - UAM03225A413YD06
  - UAM30227A423YP20

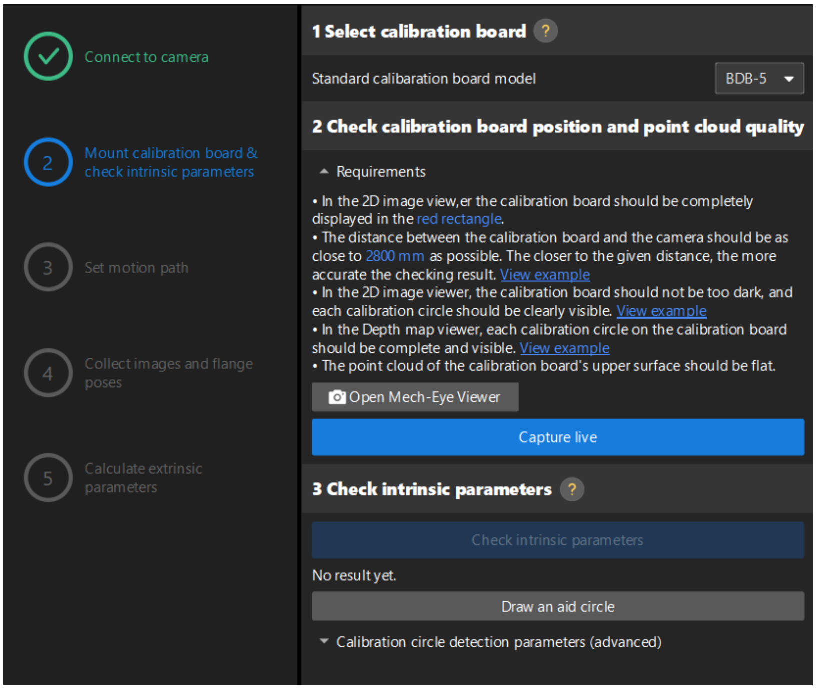
2. After the camera is connected, click the [**Capture once**] or [**Capture live**] button.


Files
Tools
Connect to camera
2
Mount calibration board & check intrinsic parameters
3
Set motion path
4
Collect images and flange poses
5
Calculate extrinsic parameters
Select and connect to camera
Camera ID
[redacted]
WAM30224A403YF16
[redacted]
Capture once
Capture live

3. In the **Image viewer** panel, ensure that the camera can capture images normally and click the **[Next]** button on the bottom bar.

Mount Calibration Board & Check Intrinsic Parameters

1. In the **Mount calibration board & check intrinsic parameters** step, select the calibration board model for the **Standard calibration board model** parameter according to its model nameplate in the **1 Select calibration board** area.



Connect to camera

2 Mount calibration board & check intrinsic parameters

3 Set motion path

4 Collect images and flange poses

5 Calculate extrinsic parameters

1 Select calibration board ?

Standard calibration board model: BDB-5

2 Check calibration board position and point cloud quality

Requirements

• In the 2D image view,er the calibration board should be completely displayed in the red rectangle.
• The distance between the calibration board and the camera should be as close to 2800 mm as possible. The closer to the given distance, the more accurate the checking result. View example
• In the 2D image viewer, the calibration board should not be too dark, and each calibration circle should be clearly visible. View example
• In the Depth map viewer, each calibration circle on the calibration board should be complete and visible. View example
• The point cloud of the calibration board's upper surface should be flat.

Open Mech-Eye Viewer

Capture live

3 Check intrinsic parameters ?

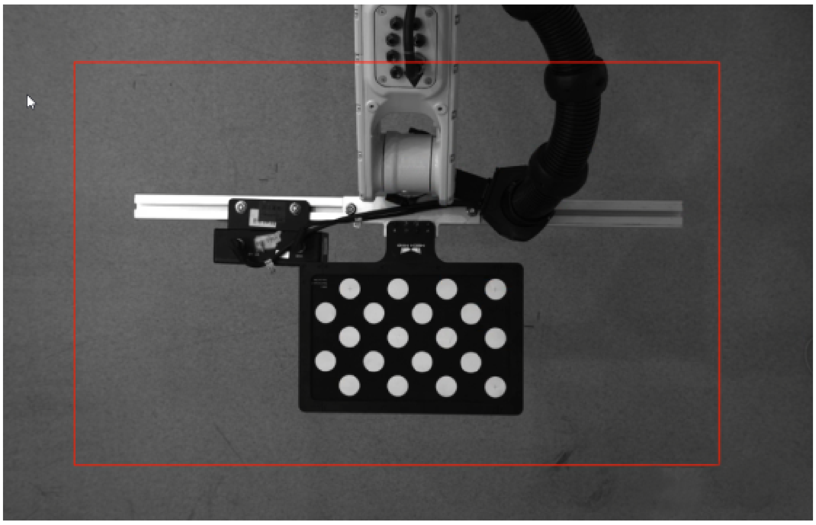
Check intrinsic parameters

No result yet.

Draw an aid circle

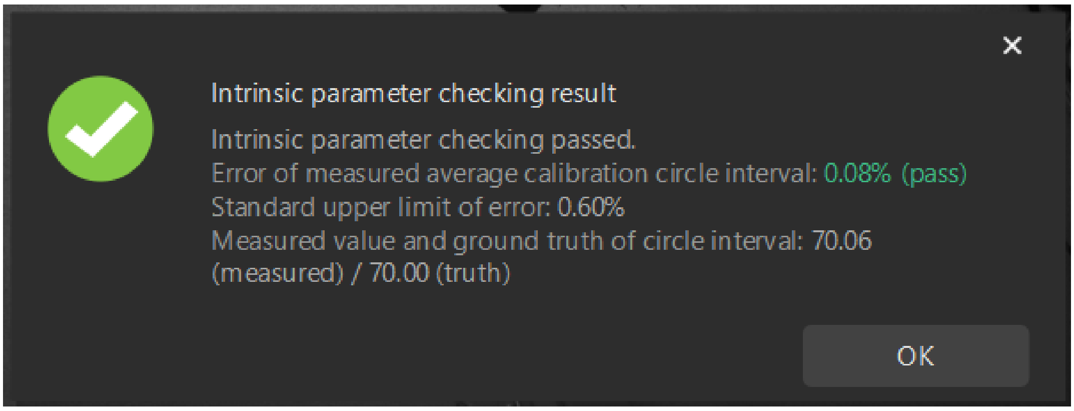
Calibration circle detection parameters (advanced)

2. In the **2 Check calibration board position and point cloud quality** area, read carefully the requirements on the calibration board position and point cloud quality, and then click the **[Capture live]** button.The **[Capture live]** button will turn into **[Stop capturing and detect position]**.
3. Control the robot to move the calibration board to the proper position, and ensure that the 2D image and depth map of the calibration board meet the requirements, and then click the **[Stop capturing and detect position]** button.



If the captured images do not meet the requirements, click **[Open Mech-Eye Viewer]** button to open the Mech-Eye Viewer software, adjust the 2D and 3D exposure parameters and re-capture images. Please note that you need to change the **Parameter group** parameter to “calib” first.

4. In the **3 Check intrinsic parameters** area, click the **[Check intrinsic parameters]** button.
5. Confirm the results of the camera intrinsic parameter check.
   - If the camera intrinsic parameter check passes, click the **[OK]** button in the prompted window, and then click the **[Next]** button on the bottom bar.



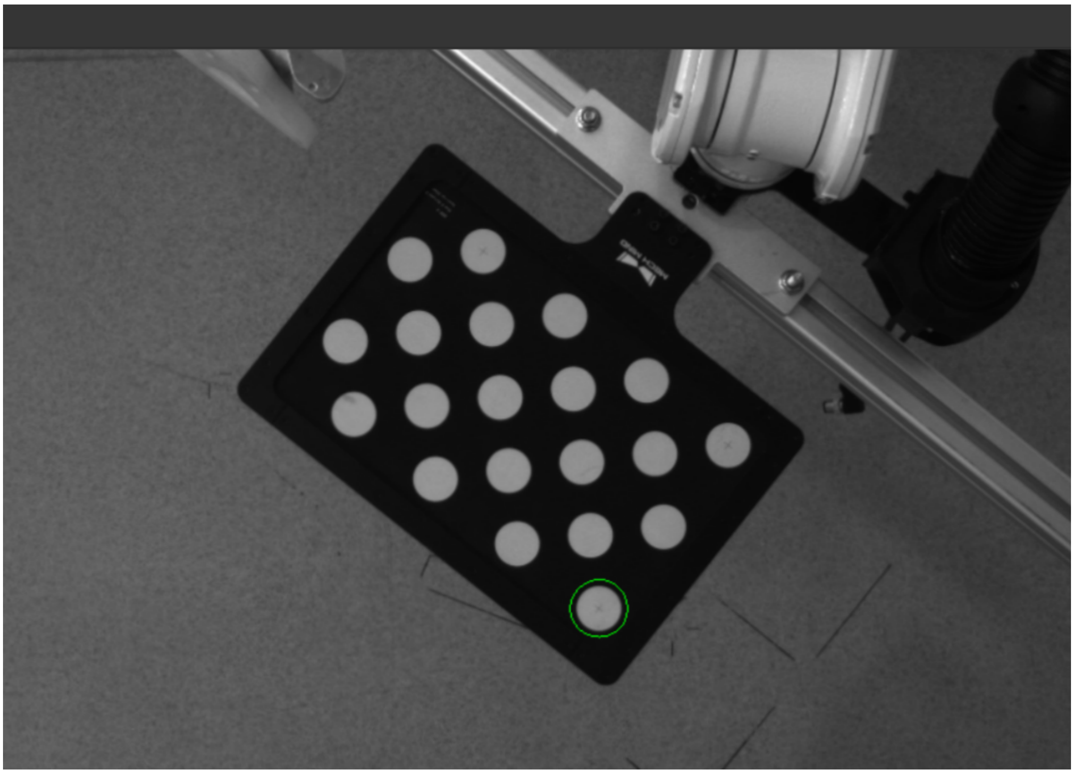
Intrinsic parameter checking result

Intrinsic parameter checking passed.
Error of measured average calibration circle interval: 0.08% (pass)
Standard upper limit of error: 0.60%
Measured value and ground truth of circle interval: 70.06 (measured) / 70.00 (truth)

- If the camera intrinsic parameter check fails, adjust detection parameters by drawing aid circles or manually adjusting the calibration circle detection parameters, and then click the **[Recheck intrinsic parameters]** button.

Draw an Aid Circle

1. To draw an aid circle, click the **[Draw an aid circle]** button.
2. In the right **Image viewer** panel, right-click the calibration board image, clear the **Fit to window** checkbox, press the `Ctrl` key and drag the roller to adjust the image to a suitable size.
3. Move the mouse pointer to the cross center point of the calibration circle, press the left mouse button and make the aid circle completely include the calibration circle and then release it.



4. Click the **[Recheck intrinsic parameters]** button, and confirm that the camera intrinsic parameter check passes.

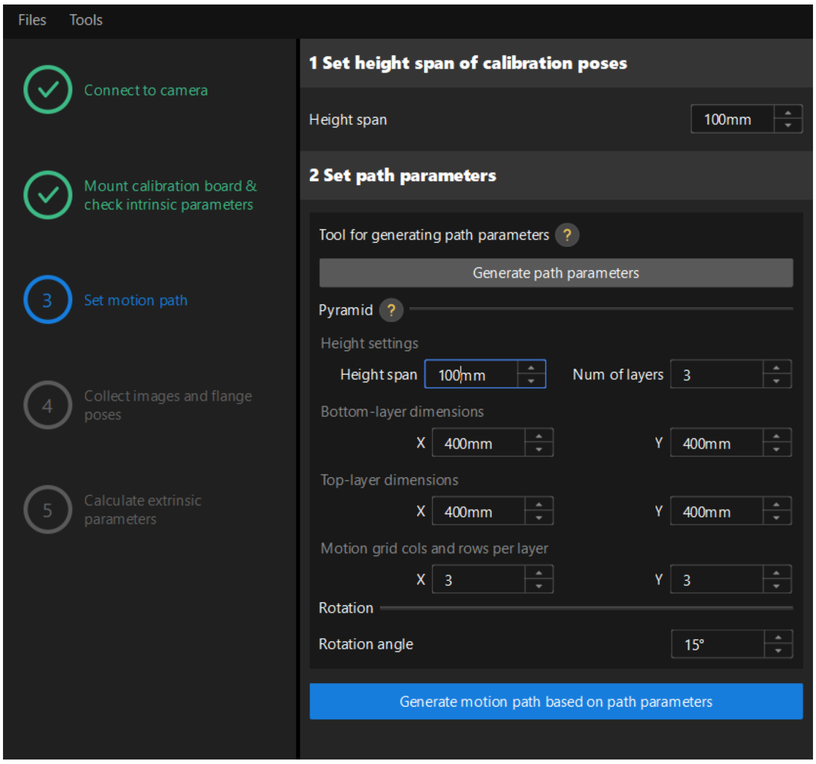
Manually Adjust Detection Parameters

To manually adjust detection parameters, click **[Calibration circle detection parameters (advanced)]**, adjust parameter settings and click the **[Recheck intrinsic parameters]** to confirm whether the camera intrinsic parameter check passes.

If the calibration circles are still not detected, please adjust the camera settings according to the on-site operating conditions.

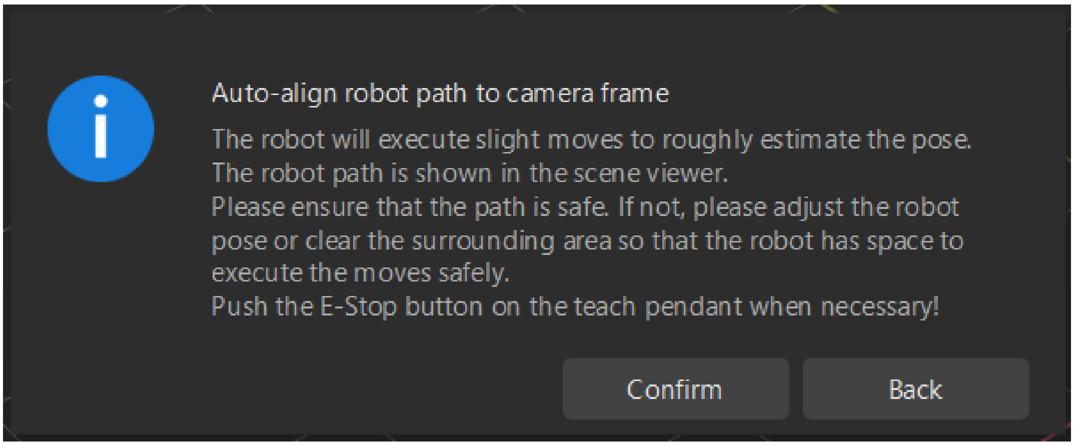
Set Motion Path

1. In the **Set motion path** step, set the **Height span** parameter in the **1 Set height span of calibration poses** area.


Files
Tools
Connect to camera
Mount calibration board & check intrinsic parameters
3
Set motion path
4
Collect images and flange poses
5
Calculate extrinsic parameters
1 Set height span of calibration poses
Height span
100mm
2 Set path parameters
Tool for generating path parameters
Generate path parameters
Pyramid
Height settings
Height span
100mm
Num of layers
3
Bottom-layer dimensions
X
400mm
Y
400mm
Top-layer dimensions
X
400mm
Y
400mm
Motion grid cols and rows per layer
X
3
Y
3
Rotation
Rotation angle
15°
Generate motion path based on path parameters

The **Height span** parameter should be set according to the recommended working distance range of the camera and the size of the robot’s working space.

2. Click the **[Generate path parameters]** button, and click the **[OK]** button in the prompted window.

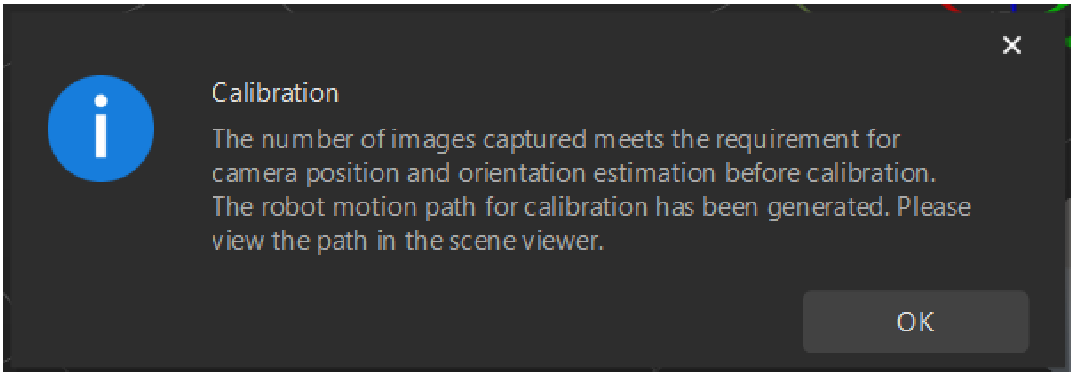


Auto-align robot path to camera frame

The robot will execute slight moves to roughly estimate the pose.
The robot path is shown in the scene viewer.
Please ensure that the path is safe. If not, please adjust the robot pose or clear the surrounding area so that the robot has space to execute the moves safely.
Push the E-Stop button on the teach pendant when necessary!

This operation will slightly move the robot and capture images. The whole process may take 10 to 15 seconds. Please ensure that the motion path is safe. In case of emergency, please tap the emergency stop button on the robot teach pendant to stop the robot immediately.

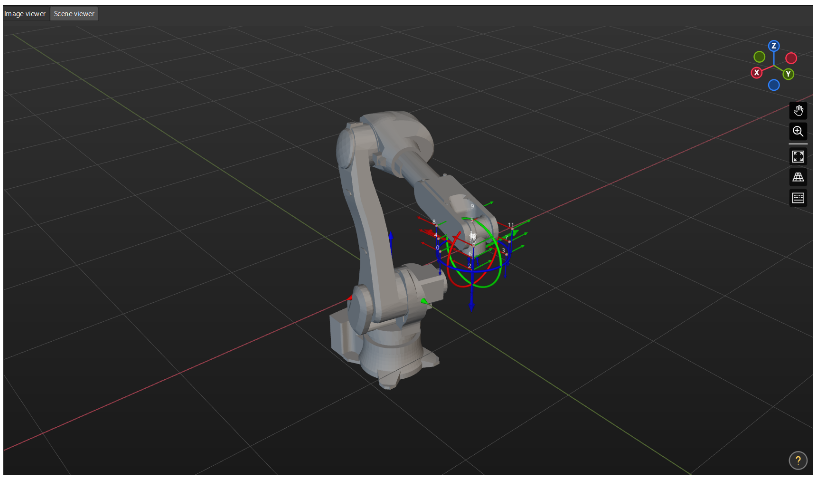
3. In the prompted **Calibrate** window, click the **[OK]** button.



Calibration

The number of images captured meets the requirement for camera position and orientation estimation before calibration.
The robot motion path for calibration has been generated. Please view the path in the scene viewer.

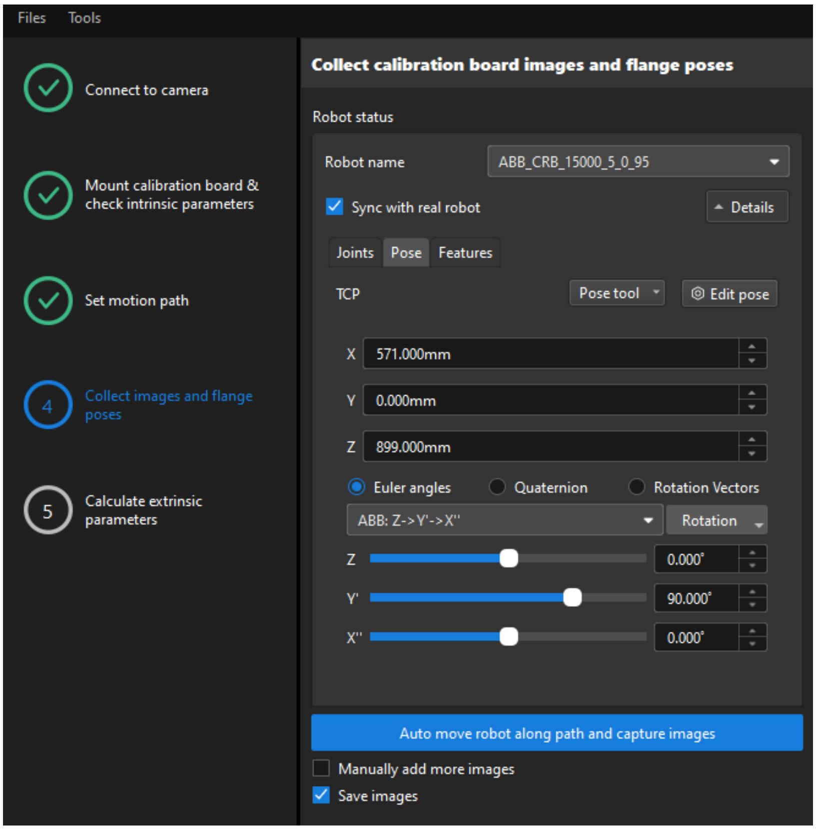
4. In the right **Scene viewer** panel, confirm that waypoints are reasonable and will not collide with obstacles.



5. Click the **[Generate motion path based on path parameters]** button and then click the **[Next]** button on the bottom bar.

Collect Images and Flange Poses

1. In the **Collect images and flange poses** step, select the **Save images** checkbox.


Files
Tools
Connect to camera
Mount calibration board & check intrinsic parameters
Set motion path
4
Collect images and flange poses
5
Calculate extrinsic parameters
Collect calibration board images and flange poses
Robot status
Details
Joints
Pose
Features
TCP
Pose tool
Edit pose
Auto move robot along path and capture images

Robot name: ABB_CRB_15000_5_0_95

☑ Sync with real robot

X 571.000mm

Y 0.000mm

Z 899.000mm

● Euler angles ○ Quaternion ○ Rotation Vectors

ABB: Z->Y'->X'' | Rotation

Z 0.000°

Y' 90.000°

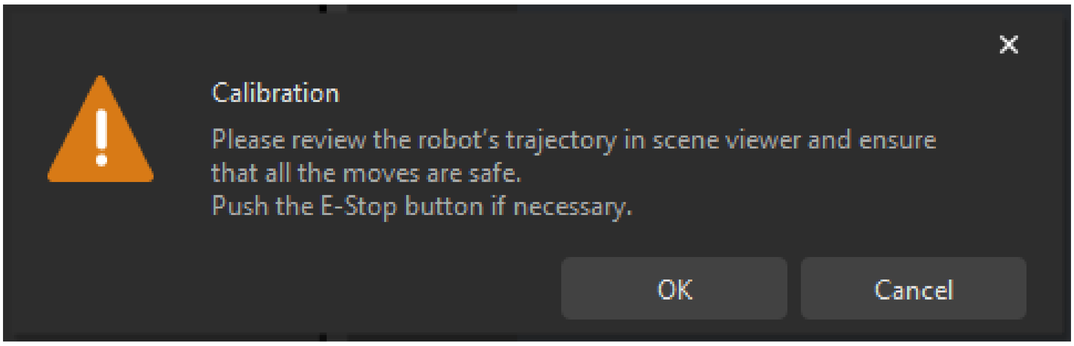
X'' 0.000°

□ Manually add more images

☑ Save images

2. If the robot uses the Master-Control to communicate with the Vision System, select the **Sync with real robot** checkbox (selected by default, recommended), and click **Details** to confirm that the robot's JPs and pose data are the same as those on the teach pendant. Clear the checkbox if you do not need to synchronize the data with the real robot.

3. Click the **[Auto move robot along path and capture images]** button, read the safety window carefully, and then click the **[OK]** button.



Calibration

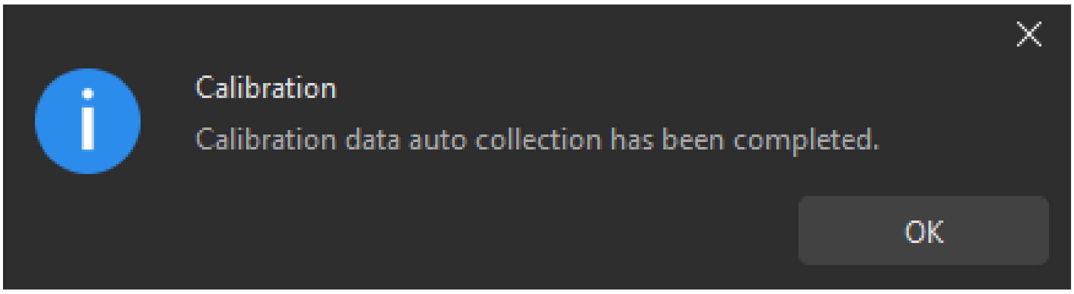
Please review the robot's trajectory in scene viewer and ensure that all the moves are safe.
Push the E-Stop button if necessary.

4. Wait until the robot finishes moving along the preset path and the camera finishes capturing images on all waypoints. In the right **Image viewer** panel, all the captured images are displayed.

- Please stay away from the robot working area to keep safe.
- Clicking the **[Stop robot]** button can exit the calibration process. But the robot will not stop until it finishes the current waypoint. In case of emergency, please tap the emergency stop button on the robot teach pendant to stop the robot immediately

(the robot needs to be reconnected after tapping the emergency stop button).

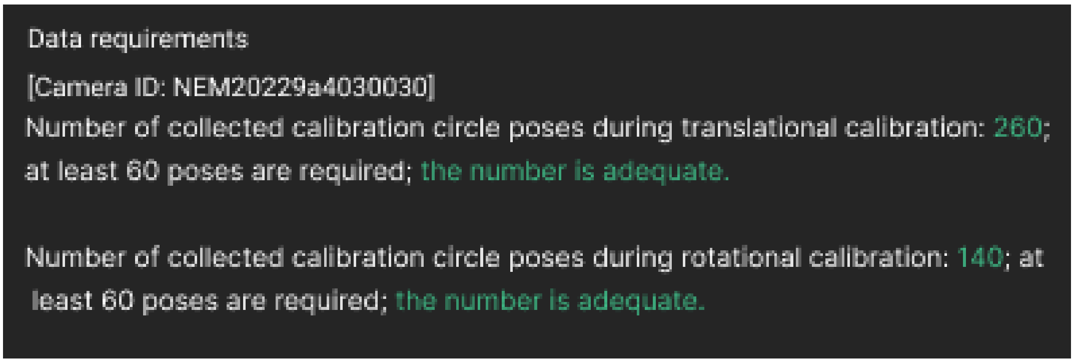
5. After all images are captured, click the [OK] button in the prompted window.



Calibration

Calibration data auto collection has been completed.

6. Confirm that the collected calibration data meets the data requirements, and then click the [Next] button on the bottom bar.



Data requirements

[Camera ID: NEM20229a4030030]

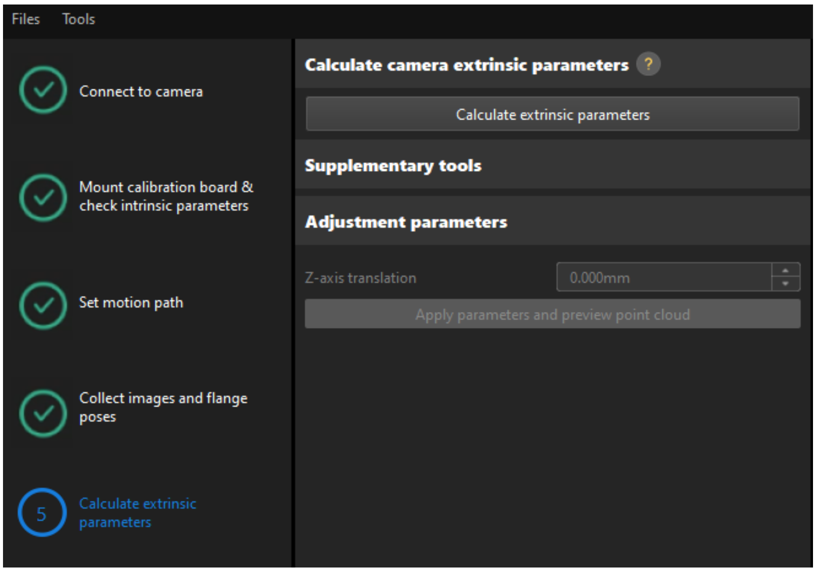
Number of collected calibration circle poses during translational calibration: 260; at least 60 poses are required; the number is adequate.

Number of collected calibration circle poses during rotational calibration: 140; at least 60 poses are required; the number is adequate.

If the requirements are not met, please manually move the robot (through the teach pendant or Mech-Viz), select the [Manually add more images] checkbox, and click the [Add an image and record flange pose] button to add the calibration board image and enter the robot flange pose.

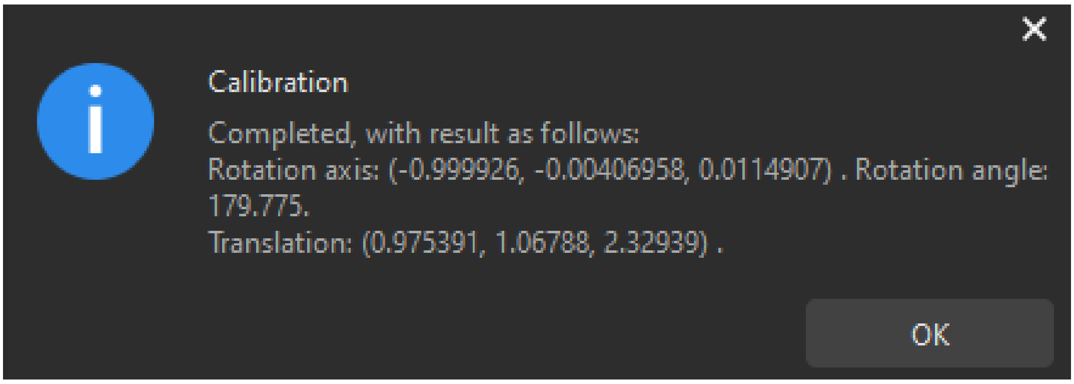
Calculate Extrinsic Parameters

1. In the **Calculate extrinsic parameters** step, click the [**Calculate extrinsic parameters**] button.



Files
Tools
Connect to camera
Mount calibration board & check intrinsic parameters
Set motion path
Collect images and flange poses
5
Calculate extrinsic parameters
Calculate camera extrinsic parameters
?
Calculate extrinsic parameters
Supplementary tools
Adjustment parameters
Z-axis translation
0.000mm
Apply parameters and preview point cloud

1. In the prompted window indicating calibration success, click the **OK** button.



Calibration

Completed, with result as follows:
Rotation axis: (-0.999926, -0.00406958, 0.0114907) . Rotation angle: 179.775.
Translation: (0.975391, 1.06788, 2.32939) .

To evaluate the robot absolute accuracy and the accuracy of calculated extrinsic parameters, you can click **[Robot absolute accuracy and repeatability]** and **[Extrinsic parameter accuracy]** in the “Calculate extrinsic parameters” window and use the error analysis tool to perform robot absolute accuracy check and analyze extrinsic parameter error in ETH setup.

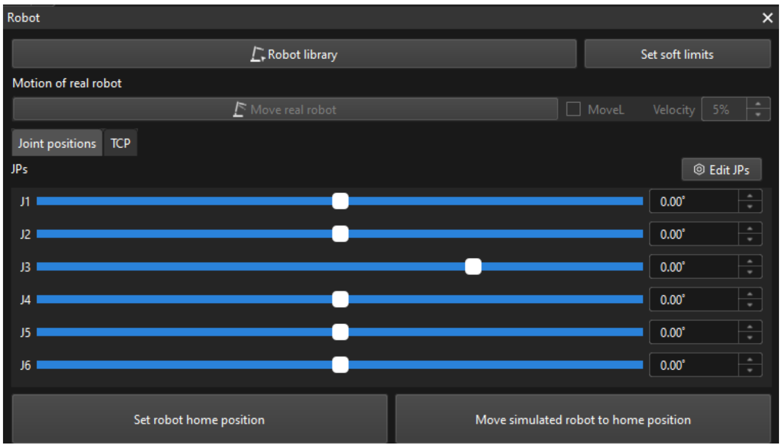
2. Adjust the Z-axis value of the extrinsic parameters.

   Due to the lack of **degrees of freedom in rotation** in four-axis robots, the captured images during calibration may lack the **images for calibration board rotation**. After calibration, you need to manually adjust the Z-axis value of the extrinsic parameters.

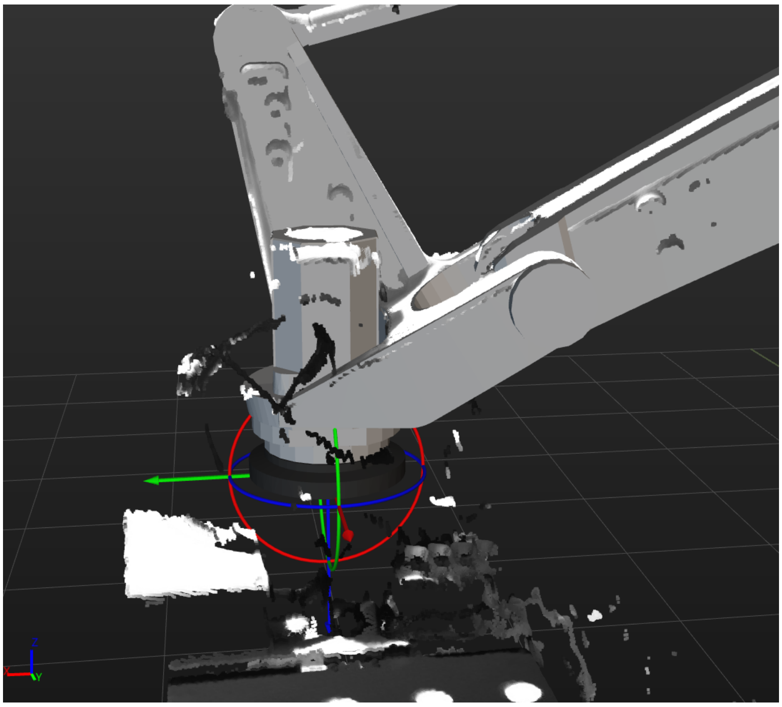
   a. Control the robot to move to a position in the camera’s field of view, and then click the **[Apply parameters and preview point cloud]** button in the **Adjust parameters** area.

   b. In Mech-Viz, confirm whether the robot’s actual point cloud coincides with the robot model.

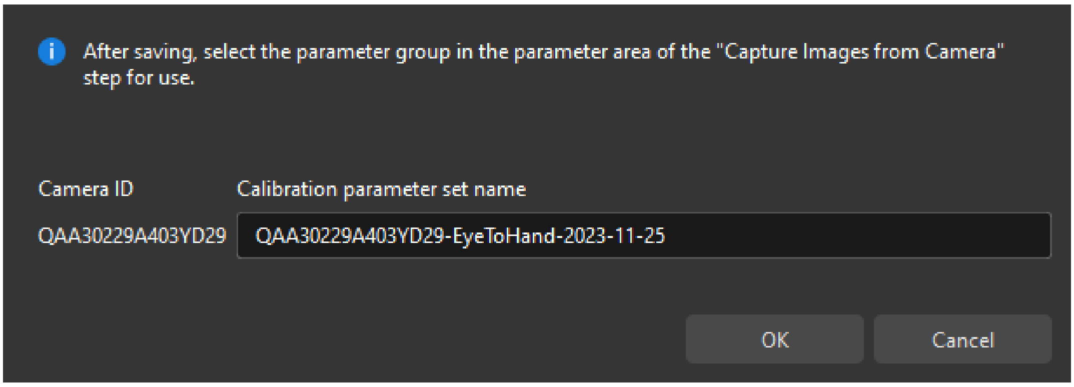
   c. Check the robot’s current pose in the teach pendant, and enter the robot’s current JPs in the **Robot** panel of Mech-Viz.



   d. In the 3D Simulation Area of Mech-Viz, confirm whether the robot’s actual point cloud coincides with the robot model. If yes, proceed to the next step. Otherwise modify the value of the **Z-axis translation** parameter in Mech-Vision, and click the **[Apply parameters and preview point cloud]** button. Repeat this operation until the robot’s actual point cloud coincides with the robot model.



3. Click the [Save] button on the bottom bar. In the prompted **Save Calibration Files** dialog box, click the [OK] button. The camera calibration result will be automatically saved in the "calibration" directory of the project.



After saving, select the parameter group in the parameter area of the "Capture Images from Camera" step for use.

Camera ID: QAA30229A403YD29

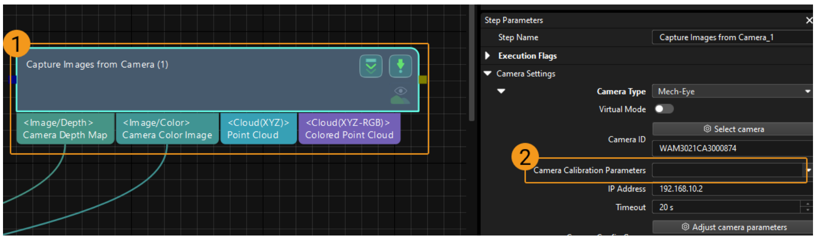
Calibration parameter set name: QAA30229A403YD29-EyeToHand-2023-11-25

Till now, the calibration process is completed.

Change the Extrinsic Parameter File

If the camera's extrinsic parameter file has been used by a Mech-Vision project, you need to change the extrinsic parameter file used by the Mech-Vision project to a new one after you finish the hand-eye calibration.

1. Select the **Capture Images from Camera** Step.



Step Parameters

Step Name: Capture Images from Camera_1

Execution Flags

Camera Settings

Camera Type: Mech-Eye

Virtual Mode

Camera ID: Select camera; WAM3021CA3000874

Camera Calibration Parameters

IP Address: 192.168.10.2

Timeout: 20 s

Adjust camera parameters

2. In the **Step Parameters** panel, click ▼ of the **Camera Calibration Parameters** parameter, and select the newly calibrated extrinsic parameter file.

Calibration-Related Status Codes

If the robot uses the Standard Interface to communicate with the Vision System during hand-eye calibration, the Vision System will return status codes in response data to indicate the execution status of the commands. The status codes indicate both the normal execution results and execution failures. You can perform further troubleshooting according to the status codes.

Calibration Error Codes

Error code	Description
7001	Calibration: Parameter error
7002	Calibration: Calibration flange pose not provided by Mech-Vision
7003	Calibration: Calibration joint positions not provided by Mech-Vision
7004	Calibration: Robot failed to reach calibration point

Calibration Normal Status Codes

Normal Status Code	Description
7100	Calibration: Robot reaches calibration point successfully
7101	Calibration: Pose received from Mech-Vision successfully

4.5.2.3. Manual Calibration in the Eye-to-Hand Setup (Six-Axis Robots - Using Multiple Random Calibration Board Poses Method)

This how-to guide introduces how to complete the manual calibration for six-axis robots using the multiple random calibration board poses method in the eye-to-hand (ETH) setup.

Preparation before Calibration

Before hand-eye calibration, you need to finish the following preparations:

- Construct the Mech-Mind Vision System.
- Prepare the materials required for calibration.
- Adjust the effect of calibration board’s point cloud.

Construct the Vision System

Construct the Mech-Mind Vision System by referring to the section Vision System Hardware Setup.

You need to use Mech-Eye Viewer, Mech-Vision and Mech-Viz during hand-eye calibration. Please ensure that they have been installed and are running the latest versions.

Prepare the Materials Required for Calibration

The hand-eye calibration in the ETH setup needs to use the calibration board.

Please prepare the calibration board according to the following requirements:

- Ensure that the circles of the calibration board are clearly visible and without obvious scratches, and the board does not suffer from deformations.
- In the ETH setup, mount the robot-specific bracket for calibration board onto the robot flange, and then mount the calibration board onto the bracket. Make sure that the calibration board is attached securely, and that the board is parallel to the XY plane of the robot flange frame.

> If an undetachable gripper is connected to the robot flange, you can attach the calibration board directly to the gripper.

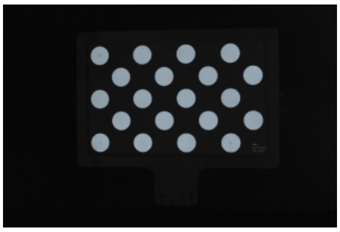
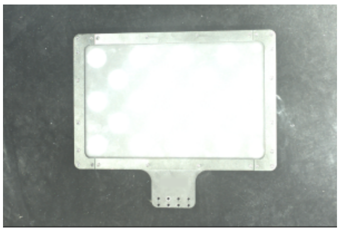
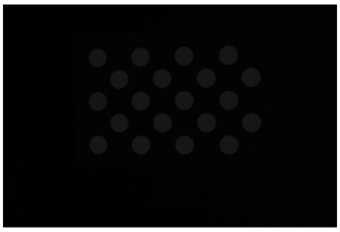
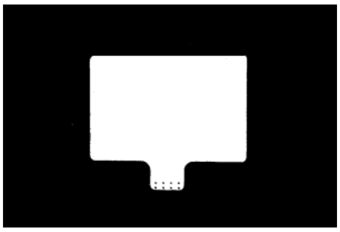
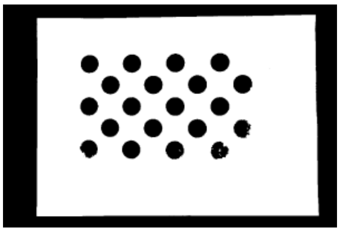
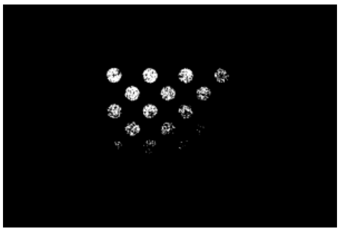
In addition, before calibration, move the robot to the starting point for calibration.

Adjust the Effect of Calibration Board's Point Cloud

1. Open Mech-Eye Viewer to adjust camera parameters.
2. Adjust the 2D parameters to make sure that the 2D image of the calibration board is clear and neither overexposed nor underexposed.
3. Adjust 3D parameters to make sure that the point clouds of the circles on the calibration board are complete and have clear contours. It is recommended to set **Surface Smoothing** and **Outlier Removal** to **Normal** in the **Point Cloud Processing** section to reduce the point cloud fluctuation range.

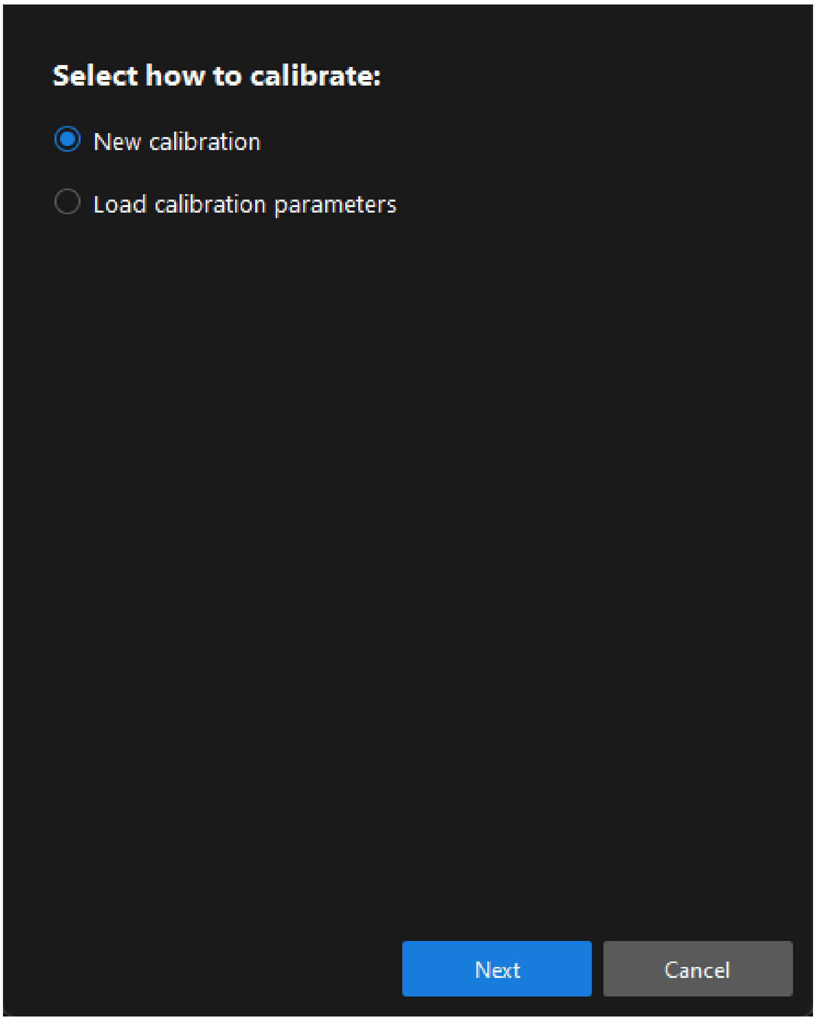
   > If the on-site ambient lights are not ideal and affect the quality of 2D images and point clouds, you can use shading or supplemental light to improve the lighting conditions.

4. Make sure that the images and point cloud of the calibration board are up to standard after completing the preceding steps.

	Normal	Overexposed	Underexposed
2D image			
Point cloud			

Pre-calibration Configuration

1. Open Mech-Vision, and click the **[Camera Calibration (Standard)]** button in the toolbar.The **Configuration before Calibration** window will be prompted.
2. In the **Select how to calibrate** window, select the **New calibration** radio button, and then click the **[Next]** button.

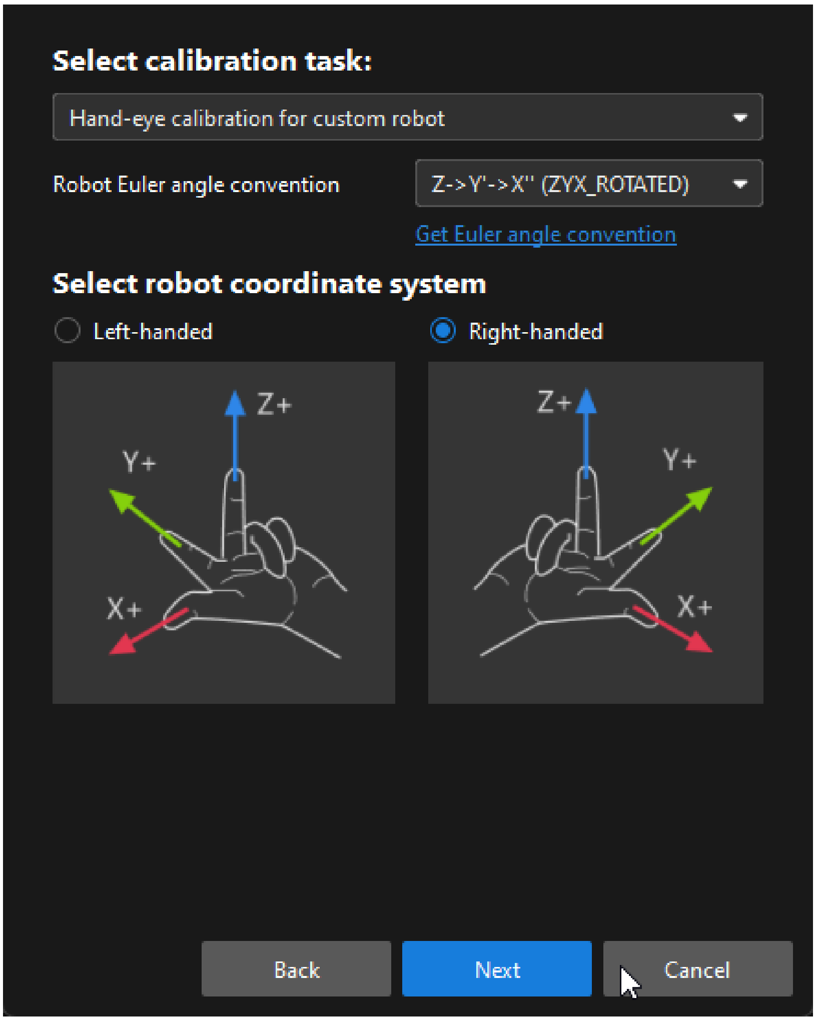


Select how to calibrate:

● New calibration

○ Load calibration parameters

3. In the **Select calibration task** window, select **Hand-eye calibration for custom robot** from the drop-down list box, specify the **Robot Euler angle convention** parameter, select the robot coordinate system type, and then click the **[Next]** button.



Select calibration task:

Hand-eye calibration for custom robot

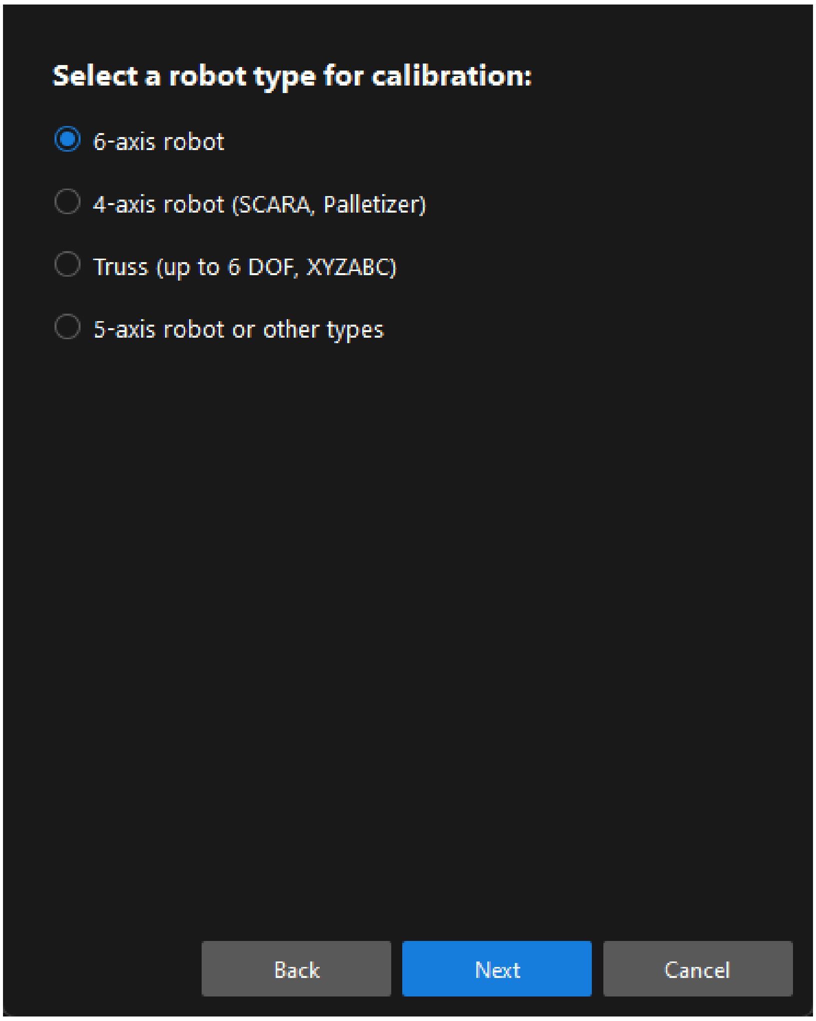
Robot Euler angle convention: Z->Y'->X'' (ZYX_ROTATED)

Get Euler angle convention

Select robot coordinate system

○ Left-handed ● Right-handed

4. In the **Select a robot type for calibration** window, select the **6-axis robot** radio button, and then click the **[Next]** button.



Select a robot type for calibration:

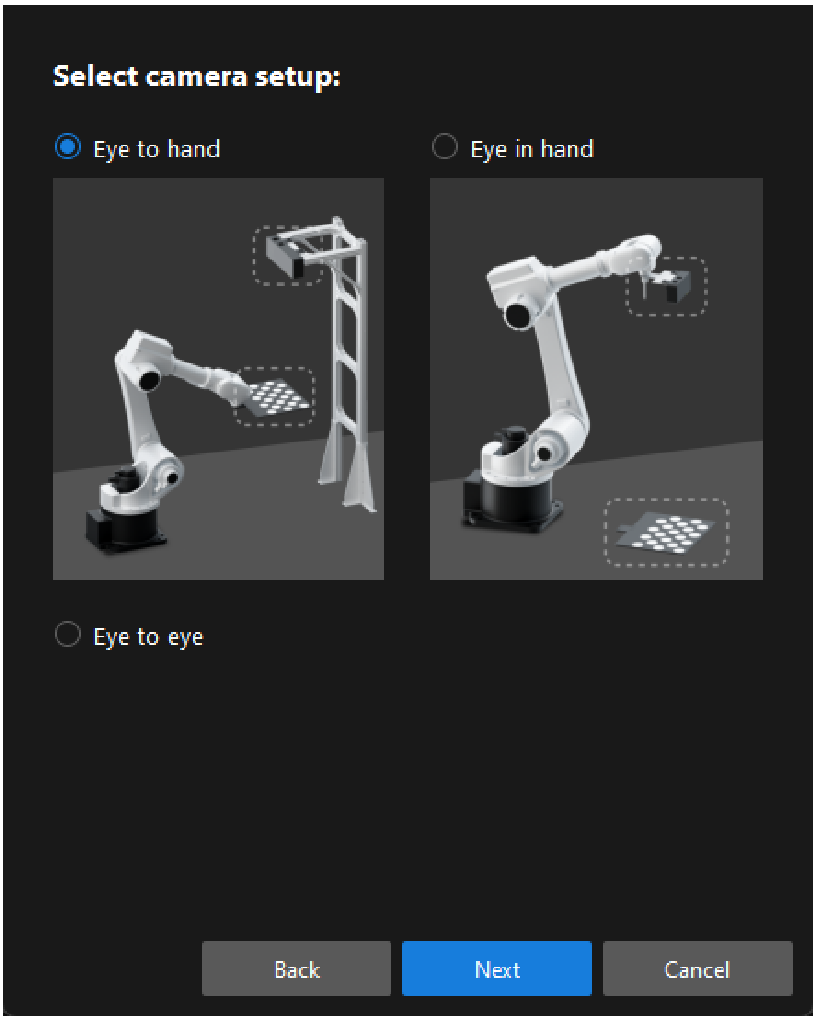
● 6-axis robot

○ 4-axis robot (SCARA, Palletizer)

○ Truss (up to 6 DOF, XYZABC)

○ 5-axis robot or other types

5. In the **Select camera setup** window, select the **Eye to hand** radio button, and then click the **[Next]** button.

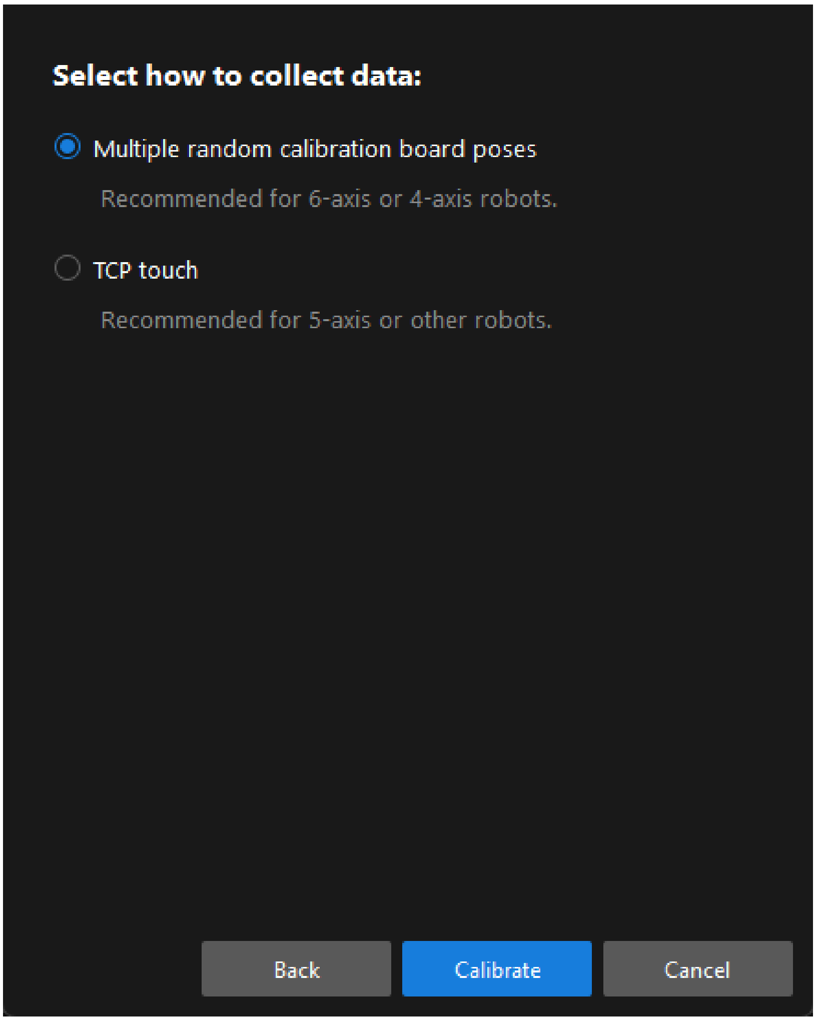


Select camera setup:

● Eye to hand ○ Eye in hand

○ Eye to eye

6. In the **Select how to collect data** window, select the **Multiple random calibration board poses**, and then click the **[Calibrate]** button.The **Calibration (Eye to hand)** window will be prompted.



Select how to collect data:

● Multiple random calibration board poses

Recommended for 6-axis or 4-axis robots.

○ TCP touch

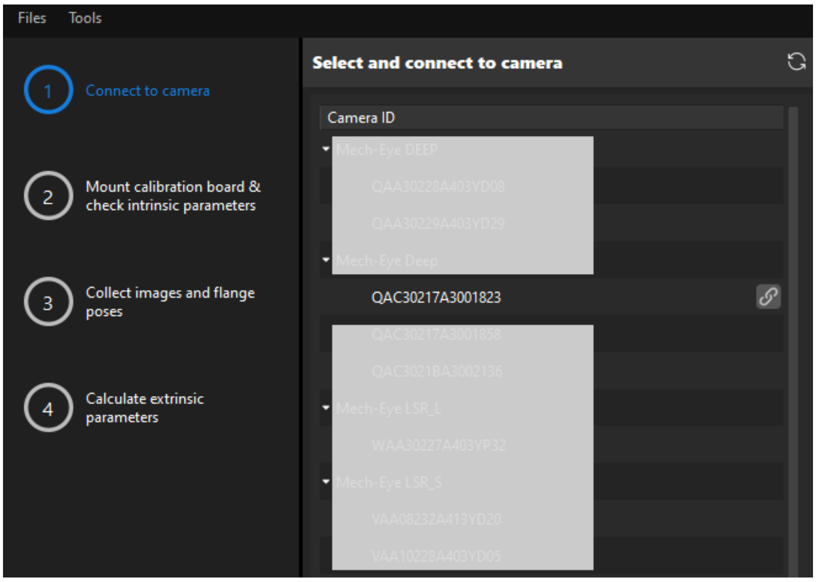
Recommended for 5-axis or other robots.

Till now, you have completed the pre-calibration configuration and can start the calibration procedure.

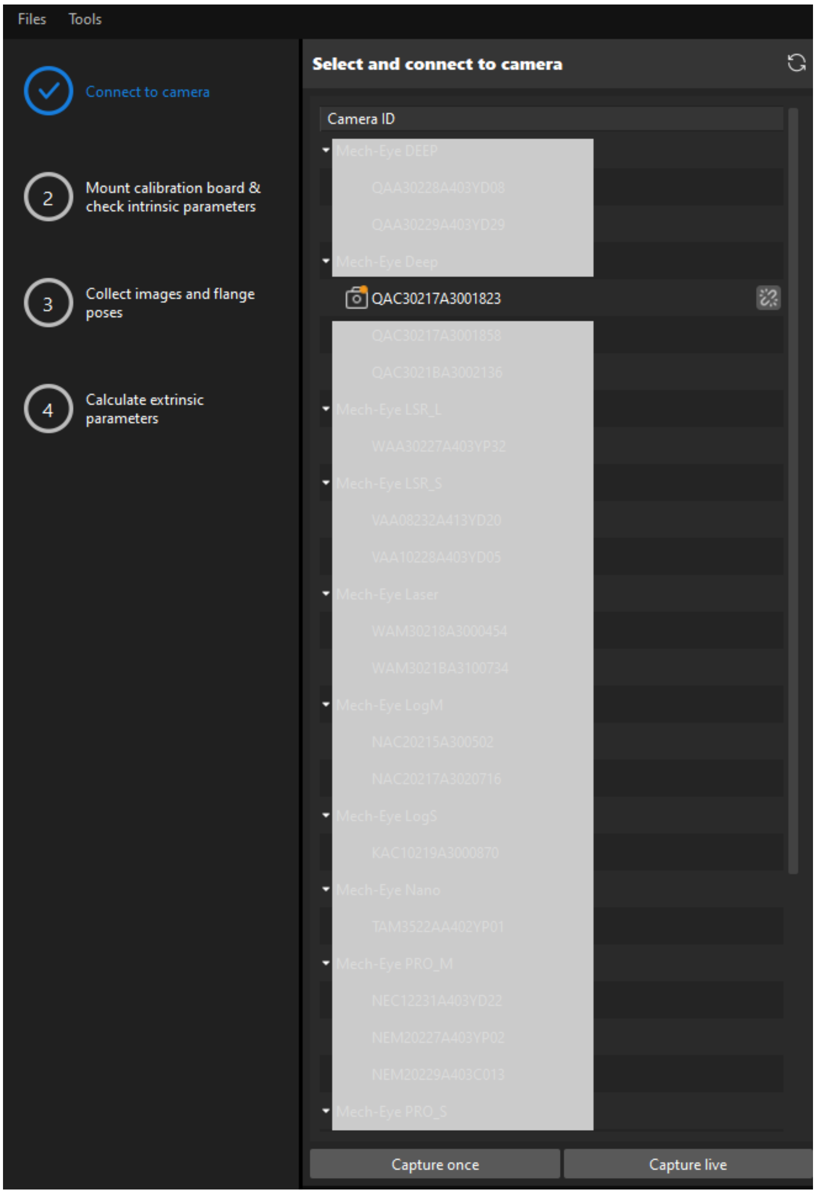
Calibration Procedure

Connect to the Camera

1. In the **Connect to camera** step, select the camera to connect in the Camera ID list, and then click the 🔗 button or double-click the camera entry to connect to it.


Files
Tools
1
Connect to camera
2
Mount calibration board & check intrinsic parameters
3
Collect images and flange poses
4
Calculate extrinsic parameters
Select and connect to camera
Camera ID
[redacted]
QAC30217A3001823
[redacted]

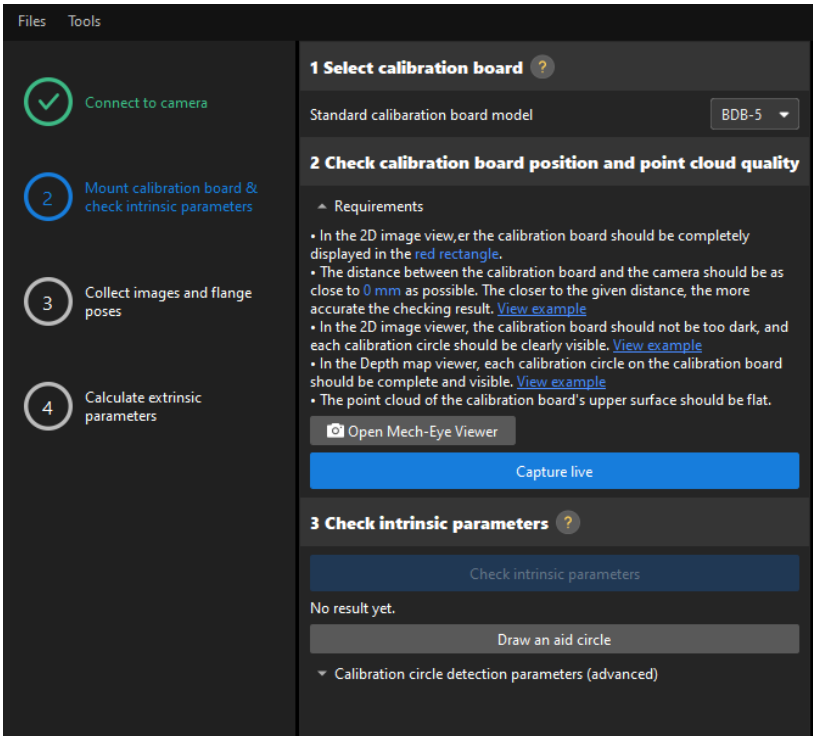
2. After the camera is connected, click the [Capture once] or [Capture live] button.


Files
Tools
Connect to camera
2
Mount calibration board & check intrinsic parameters
3
Collect images and flange poses
4
Calculate extrinsic parameters
Select and connect to camera
Camera ID
[redacted]
QAC30217A3001823
[redacted]
Capture once
Capture live

3. In the **Image viewer** panel, ensure that the camera can capture images normally and click the **[Next]** button on the bottom bar.

Mount Calibration Board & Check Intrinsic Parameters

1. In the **Mount calibration board & check intrinsic parameters** step, select the calibration board model for the **Standard calibration board model** parameter according to its model nameplate in the **1 Select calibration board** area.


Files
Tools
Connect to camera
2
Mount calibration board & check intrinsic parameters
3
Collect images and flange poses
4
Calculate extrinsic parameters
?
Open Mech-Eye Viewer
Capture live
Check intrinsic parameters
Draw an aid circle

1 Select calibration board

Standard calibration board model: BDB-5

2 Check calibration board position and point cloud quality

Requirements

- In the 2D image view,er the calibration board should be completely displayed in the red rectangle.
- The distance between the calibration board and the camera should be as close to 0 mm as possible. The closer to the given distance, the more accurate the checking result. View example
- In the 2D image viewer, the calibration board should not be too dark, and each calibration circle should be clearly visible. View example
- In the Depth map viewer, each calibration circle on the calibration board should be complete and visible. View example
- The point cloud of the calibration board's upper surface should be flat.

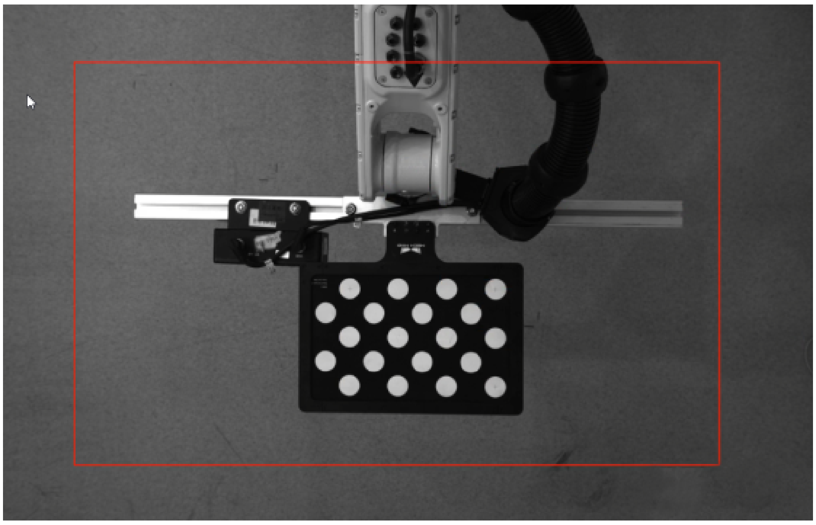
3 Check intrinsic parameters

No result yet.

Calibration circle detection parameters (advanced)

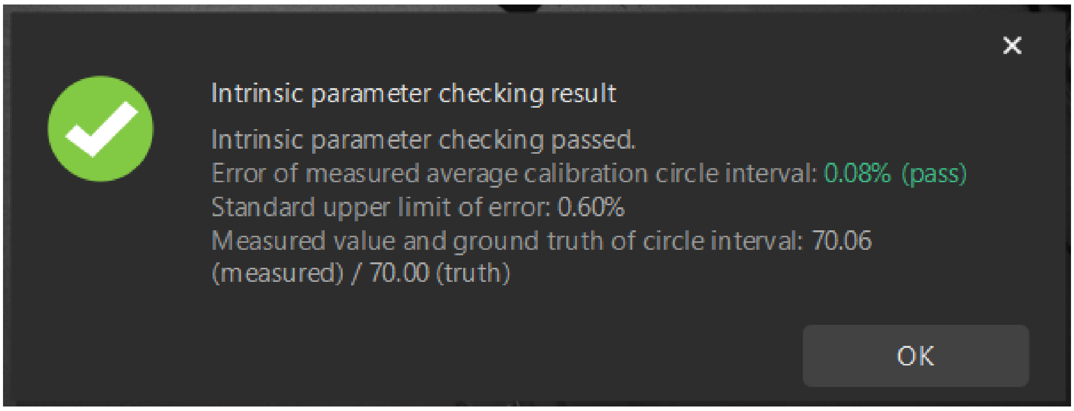
2. In the **2 Check calibration board position and point cloud quality** area, read carefully the requirements on the calibration board position and point cloud quality, and then click the **[Capture live]** button.The **[Capture live]** button will turn into **[Stop capturing and detect position]**.

3. Control the robot to move the calibration board to the proper position (in the red rectangle), and ensure that the 2D image and depth map of the calibration board meet the requirements, and then click the **[Stop capturing and detect position]** button.



If the captured images do not meet the requirements, click [Open Mech-Eye Viewer] button to open the Mech-Eye Viewer software, adjust the 2D and 3D exposure parameters and re-capture images. Please note that you need to change the **Parameter group** parameter to “calib” first.

4. In the **3 Check intrinsic parameters** area, click the [**Check intrinsic parameters**] button.
5. Confirm the results of the camera intrinsic parameter check.
   - If the camera intrinsic parameter check passes, click the [**OK**] button in the prompted window, and then click the [**Next**] button on the bottom bar.



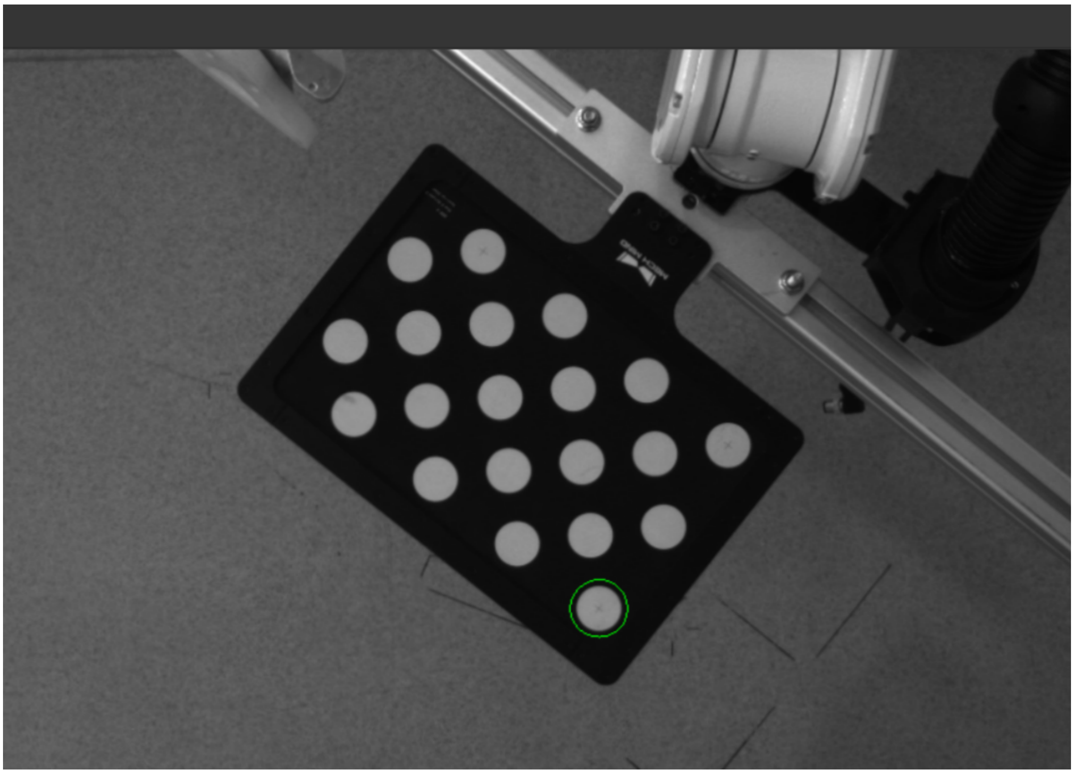
Intrinsic parameter checking result

Intrinsic parameter checking passed.
Error of measured average calibration circle interval: 0.08% (pass)
Standard upper limit of error: 0.60%
Measured value and ground truth of circle interval: 70.06 (measured) / 70.00 (truth)

   - If the camera intrinsic parameter check fails, adjust detection parameters by drawing aid circles or manually adjusting the calibration circle detection parameters, and then click the [**Recheck intrinsic parameters**] button.

Draw an Aid Circle

1. To draw an aid circle, click the [**Draw an aid circle**] button.
2. In the right **Image viewer** panel, right-click the calibration board image and clear the **Fit to window** checkbox, and press the `Ctrl` key and drag the roller to adjust the image to a suitable size.
3. Move the mouse pointer to the cross center point of the calibration circle, press the left mouse button and make the aid circle completely include the calibration circle and then release it.



4. Click the **[Recheck intrinsic parameters]** button, and confirm that the camera intrinsic parameter check passes.

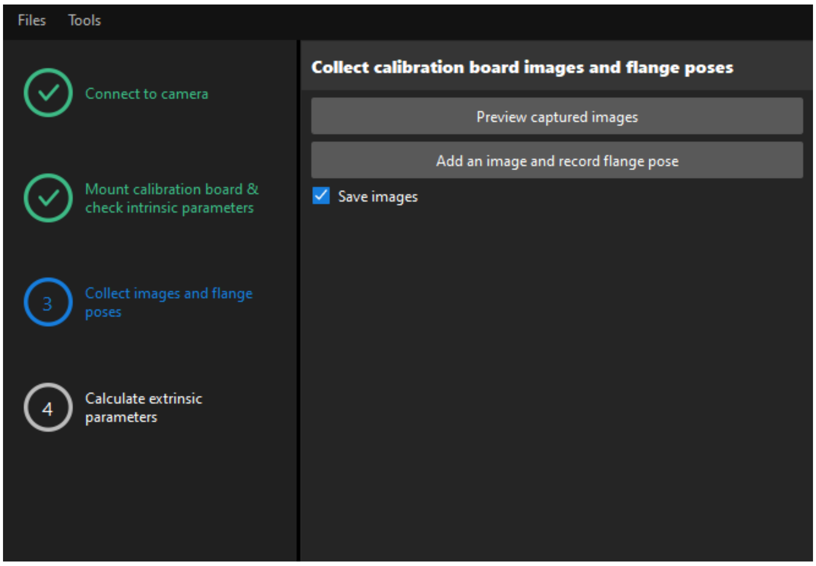
Manually Adjust Detection Parameters

To manually adjust detection parameters, click **[Calibration circle detection parameters (advanced)]**, adjust parameter settings and click the **[Recheck intrinsic parameters]** to confirm whether the camera intrinsic parameter check passes.

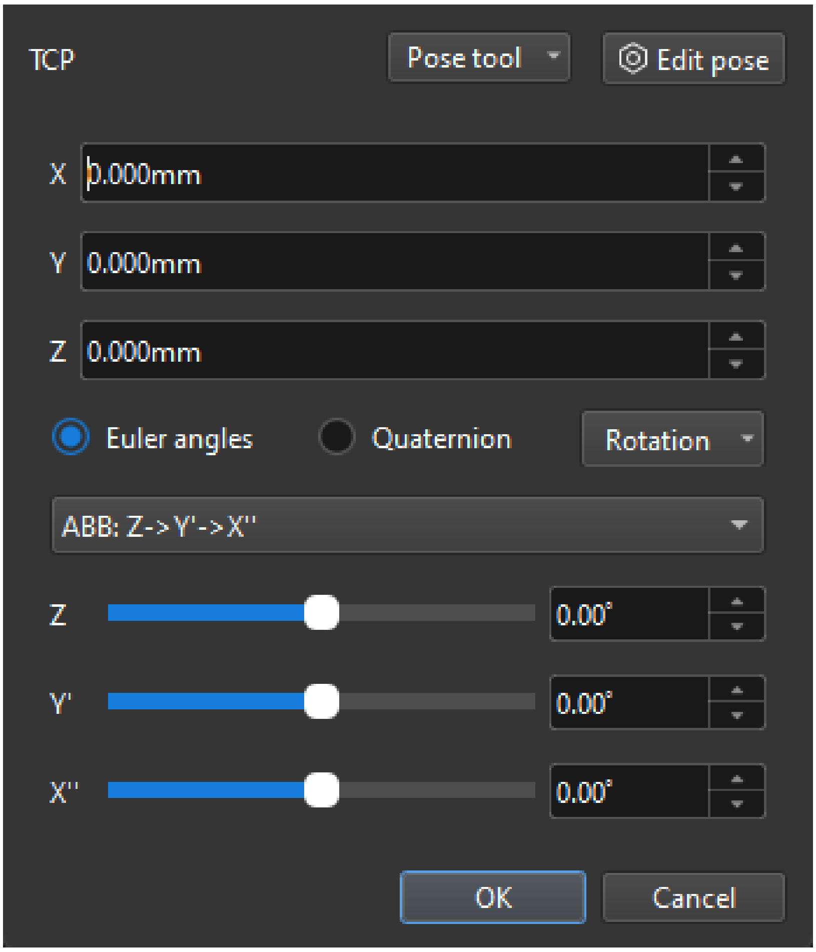
If the calibration circles are still not detected, please adjust the camera settings according to the on-site operating conditions.

Collect Images and Flange Poses

1. In the **Collect images and flange poses** step, select the **[Save images]** checkbox (selected by default, recommended). If you do not need to save images of calibration boards, clear this checkbox.


Files
Tools
Connect to camera
Mount calibration board & check intrinsic parameters
3
Collect images and flange poses
4
Calculate extrinsic parameters
Collect calibration board images and flange poses
Preview captured images
Add an image and record flange pose
☑ Save images

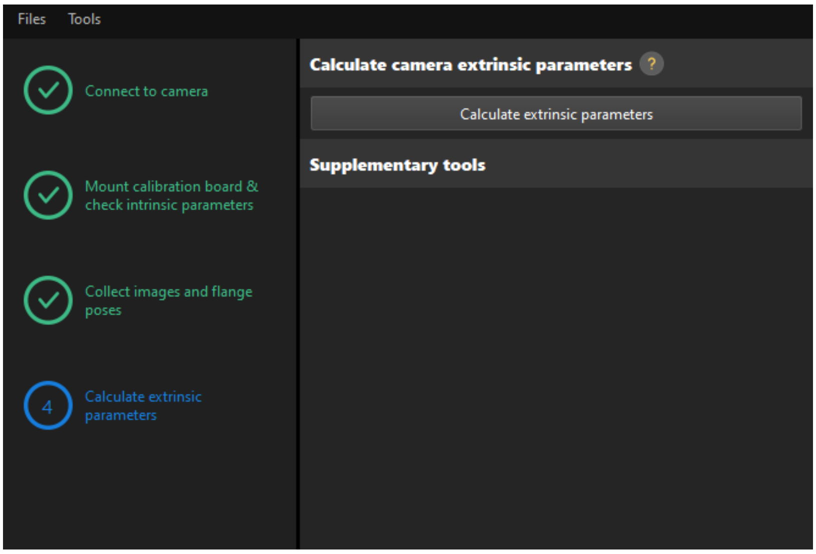
2. Control the robot to move to different calibration points and then click the [**Add an image and record flange pose**] button.
3. In the prompted window, enter the robot flange pose.



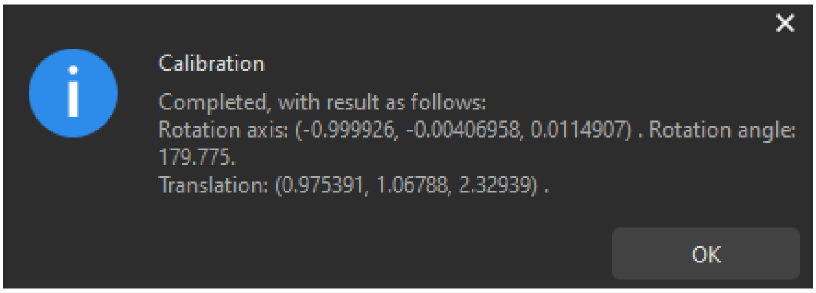
4. Confirm that the number of the currently recognized calibration circles meets the requirements, and then click the [**Next**] button on the bottom bar.

Calculate Extrinsic Parameters

1. In the **Calculate extrinsic parameters** step, click the **[Calculate extrinsic parameters]** button.


Files
Tools
Connect to camera
Mount calibration board & check intrinsic parameters
Collect images and flange poses
4
Calculate extrinsic parameters
Calculate camera extrinsic parameters
?
Calculate extrinsic parameters
Supplementary tools

2. In the prompted window indicating calibration success, click the **[OK]** button.

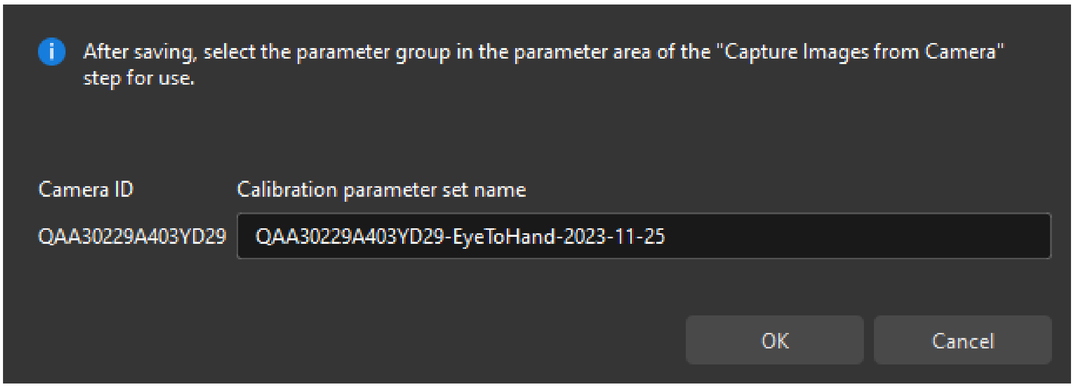


Calibration

Completed, with result as follows:
Rotation axis: (-0.999926, -0.00406958, 0.0114907) . Rotation angle: 179.775.
Translation: (0.975391, 1.06788, 2.32939) .

- If you want to enhance the calibration accuracy, please refer to the section Calibration Result Check and Analysis.
- To evaluate the robot absolute accuracy and the accuracy of calculated extrinsic parameters, you can click **[Robot absolute accuracy and repeatability]** and **[Extrinsic parameter accuracy]** in the “Calculate extrinsic parameters” window and use the error analysis tool to perform robot absolute accuracy check and analyze extrinsic parameter error in ETH setup.

3. Click the **[Save]** button on the bottom bar. In the prompted **Save Calibration Files** dialog box, click the **[OK]** button. The camera calibration result will be automatically saved in the “calibration” directory of the project.



After saving, select the parameter group in the parameter area of the "Capture Images from Camera" step for use.

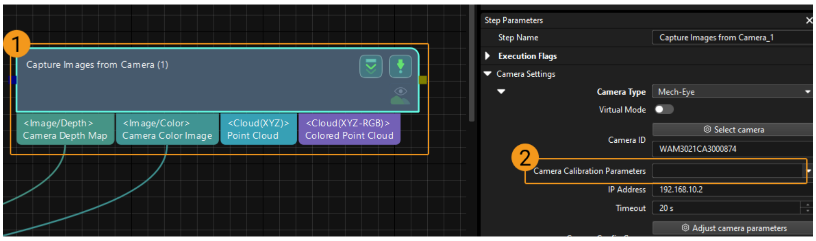
Camera ID	Calibration parameter set name
QAA30229A403YD29	QAA30229A403YD29-EyeToHand-2023-11-25

Till now, the calibration process is completed.

Change the Extrinsic Parameter File

If the camera's extrinsic parameter file has been used by a Mech-Vision project, you need to change the extrinsic parameter file used by the Mech-Vision project to a new one after you finish the hand-eye calibration.

1. Select the **Capture Images from Camera** Step.


1
Capture Images from Camera (1)
<Image/Depth> Camera Depth Map
<Image/Color> Camera Color Image
<Cloud(XYZ)> Point Cloud
<Cloud(XYZ-RGB)> Colored Point Cloud
Step Parameters
Execution Flags
Camera Settings
Select camera
2
Adjust camera parameters

Step Name: Capture Images from Camera_1

Camera Type: Mech-Eye

Virtual Mode

Camera ID: WAM3021CA3000874

Camera Calibration Parameters:

IP Address: 192.168.10.2

Timeout: 20 s

2. In the **Step Parameters** panel, click ▾ of the **Camera Calibration Parameters** parameter, and select the newly calibrated extrinsic parameter file.

4.5.2.4. Manual Calibration in the Eye-to-Hand Setup (Four-Axis Robots - Using Multiple Random Calibration Board Poses Method)

This how-to guide introduces how to complete the manual calibration for four-axis robots using the multiple random calibration board poses method in the eye-to-hand (ETH) setup.

Preparation before Calibration

Before hand-eye calibration, you need to finish the following preparations:

- Construct the Mech-Mind Vision System.
- Prepare the materials required for calibration.
- Adjust the effect of calibration board's point cloud.

Construct the Vision System

Construct the Mech-Mind Vision System by referring to the section Vision System Hardware Setup.

You need to use Mech-Eye Viewer, Mech-Vision and Mech-Viz during hand-eye calibration. Please ensure that they have been installed and are running the latest versions.

Prepare the Materials Required for Calibration

The hand-eye calibration in the ETH setup needs to use the calibration board.

Please prepare the calibration board according to the following requirements:

- Ensure that the circles of the calibration board are clearly visible and without obvious scratches, and the board does not suffer from deformations.
- In the ETH setup, mount the robot-specific bracket for calibration board onto the robot flange, and then mount the calibration board onto the bracket. Make sure that the calibration board is attached securely, and that the board is parallel to the XY plane of the robot flange frame.

> If an undetachable gripper is connected to the robot flange, you can attach the calibration board directly to the gripper.

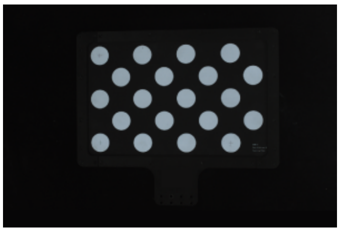
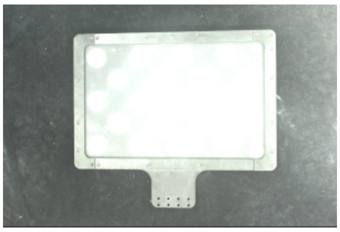
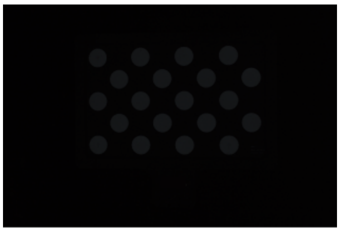
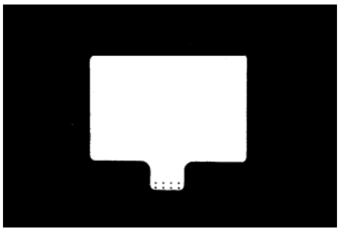
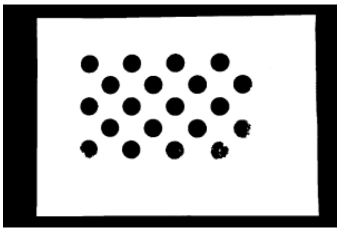
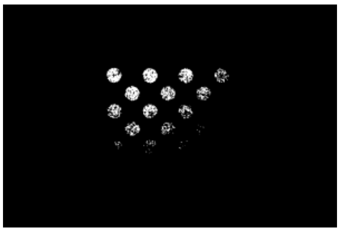
In addition, before calibration, move the robot to the starting point for calibration.

Adjust the Effect of Calibration Board's Point Cloud

1. Open Mech-Eye Viewer to adjust camera parameters.
2. Adjust the 2D parameters to make sure that the 2D image of the calibration board is clear and neither overexposed nor underexposed.
3. Adjust 3D parameters to make sure that the point clouds of the circles on the calibration board are complete and have clear contours. It is recommended to set **Surface Smoothing** and **Outlier Removal** to **Normal** in the **Point Cloud Processing** section to reduce the point cloud fluctuation range.

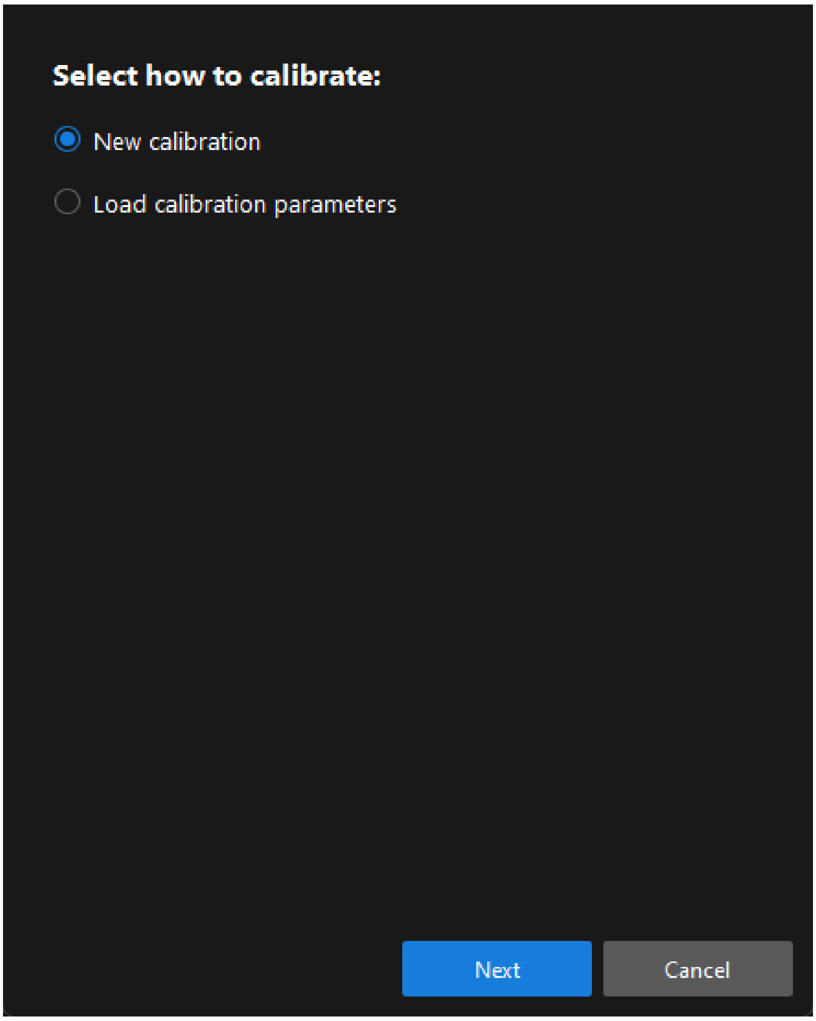
   > If the on-site ambient lights are not ideal and affect the quality of 2D images and point clouds, you can use shading or supplemental light to improve the lighting conditions.

4. Make sure that the images and point cloud of the calibration board are up to standard after completing the preceding steps.

	Normal	Overexposed	Underexposed
2D image			
Point cloud			

Pre-calibration Configuration

1. Open Mech-Vision, and click the **[Camera Calibration (Standard)]** button in the toolbar.The **Configuration before Calibration** window will be prompted.
2. In the **Select how to calibrate** window, select the **New calibration** radio button, and then click the **[Next]** button.

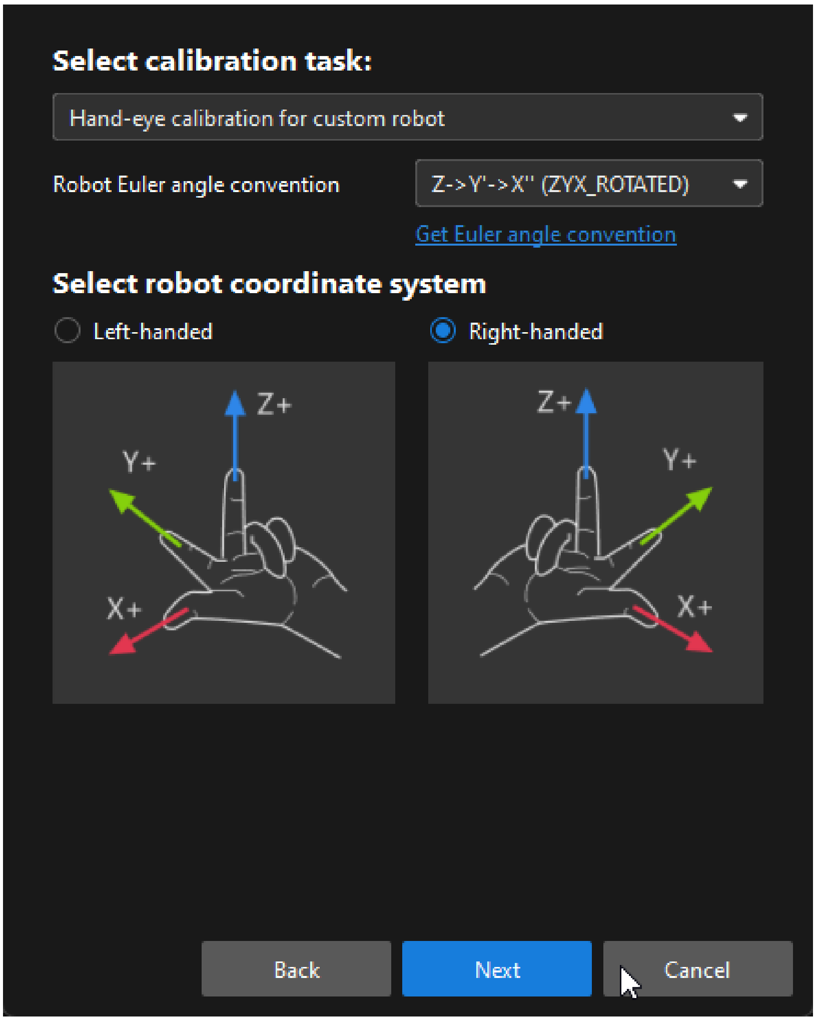


Select how to calibrate:

● New calibration

○ Load calibration parameters

3. In the **Select calibration task** window, select **Hand-eye calibration for custom robot** from the drop-down list box, specify the **Robot Euler angle convention** parameter, select the robot coordinate system type, and then click the **[Next]** button.



Select calibration task:

Hand-eye calibration for custom robot

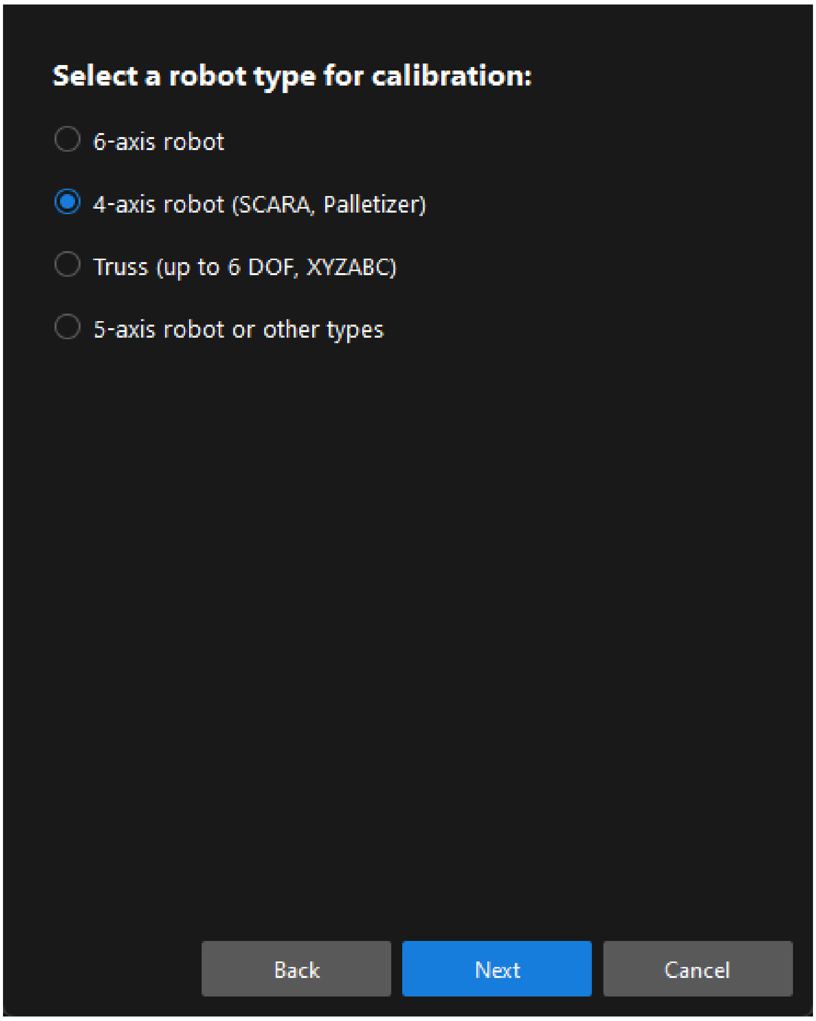
Robot Euler angle convention: Z->Y'->X'' (ZYX_ROTATED)

Get Euler angle convention

Select robot coordinate system

○ Left-handed ● Right-handed

4. In **Select a robot type for calibration** window, select the **4-axis robot (SCARA, Palletizer)** radio button, and then click the **[Next]** button.



Select a robot type for calibration:

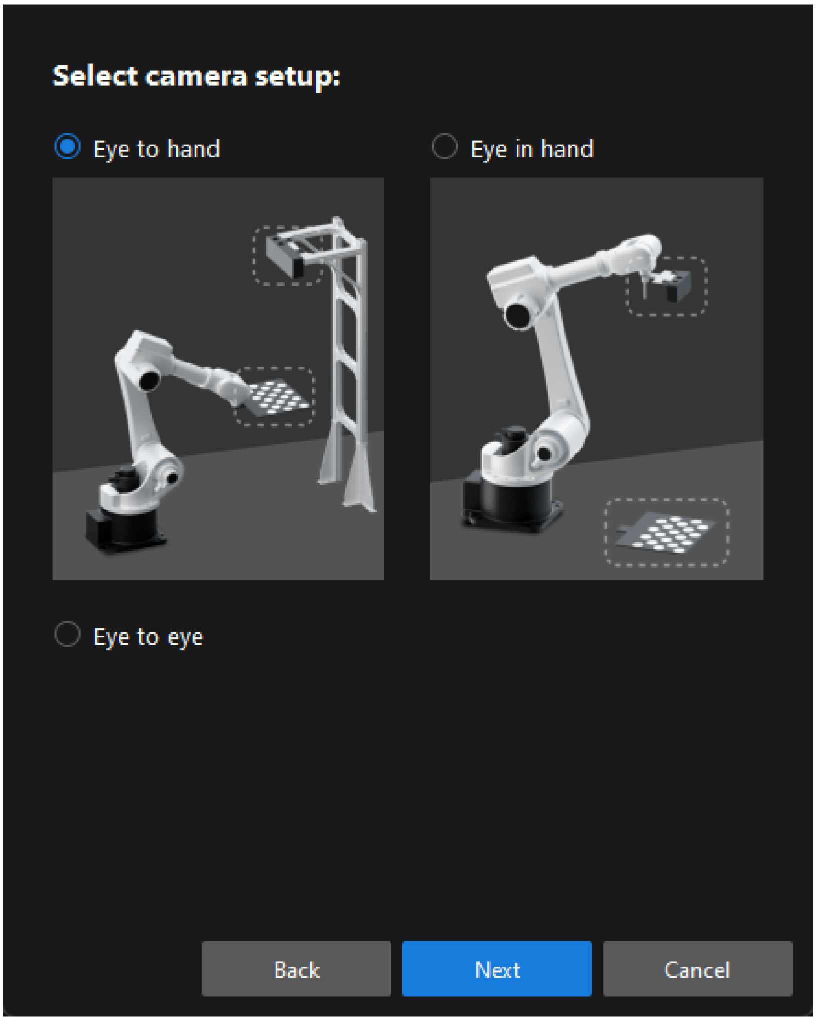
○ 6-axis robot

● 4-axis robot (SCARA, Palletizer)

○ Truss (up to 6 DOF, XYZABC)

○ 5-axis robot or other types

5. In the **Select camera setup** window, select the **Eye to hand** radio button, and then click the **[Next]** button.



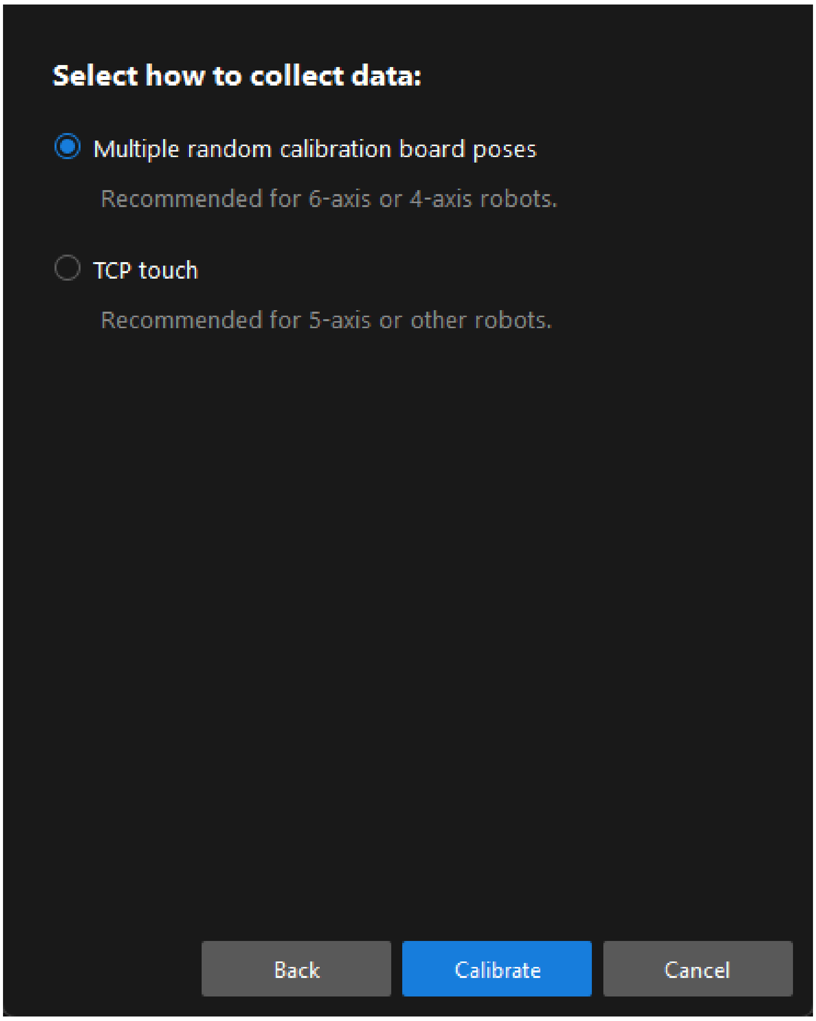
Select camera setup:

● Eye to hand

○ Eye in hand

○ Eye to eye

6. In the **Select how to collect data** window, select the **Multiple random calibration board poses**, and then click the [**Calibrate**] button.The **Calibration (Eye to hand)** window will be prompted.



Select how to collect data:

● Multiple random calibration board poses

Recommended for 6-axis or 4-axis robots.

○ TCP touch

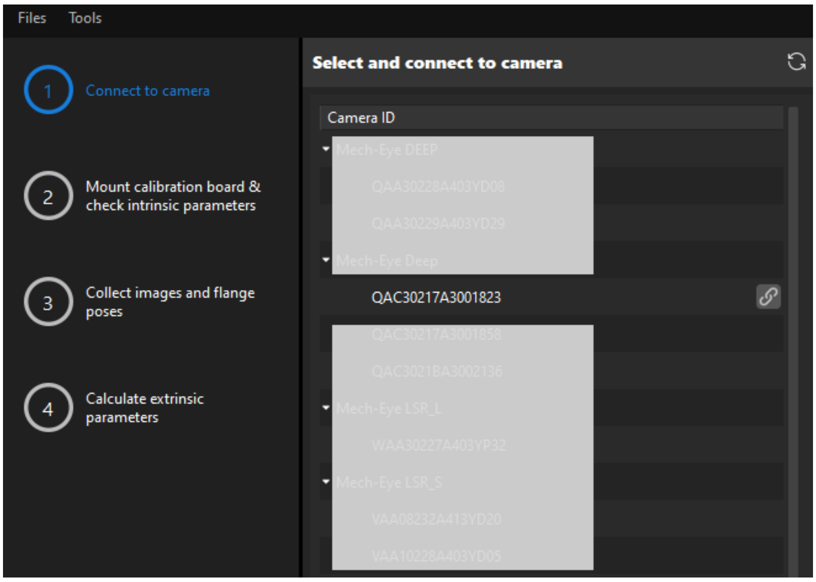
Recommended for 5-axis or other robots.

Till now, you have completed the pre-calibration configuration and can start the calibration procedure.

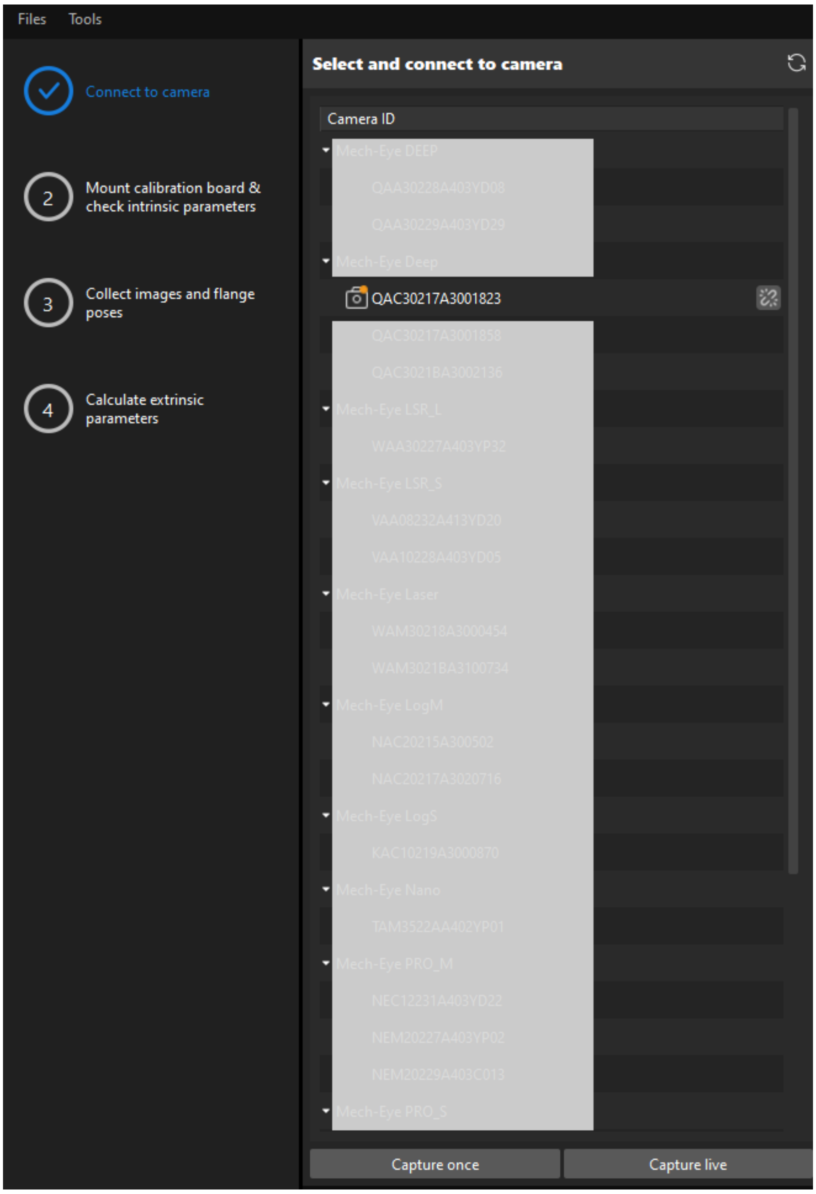
Calibration Procedure

Connect to the Camera

1. In the **Connect to camera** step, select the camera to connect in the Camera ID list, and then click the 🔗 button or double-click the camera entry to connect to it.


Files
Tools
1
Connect to camera
2
Mount calibration board & check intrinsic parameters
3
Collect images and flange poses
4
Calculate extrinsic parameters
Select and connect to camera
Camera ID
[redacted]
QAC30217A3001823
[redacted]

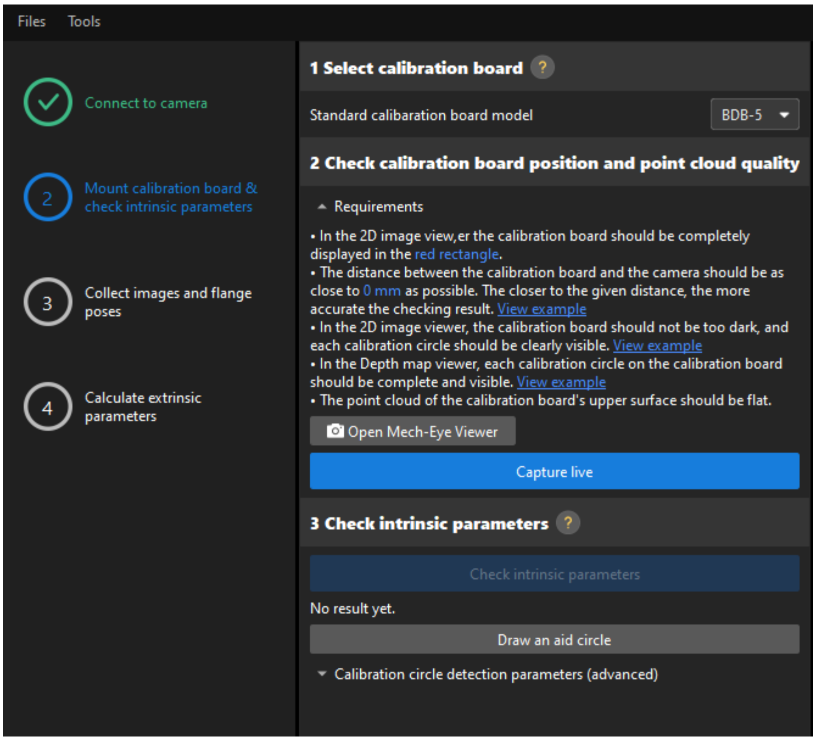
2. After the camera is connected, click the [Capture once] or [Capture live] button.


Files
Tools
Connect to camera
2
Mount calibration board & check intrinsic parameters
3
Collect images and flange poses
4
Calculate extrinsic parameters
Select and connect to camera
Camera ID
[redacted]
QAC30217A3001823
[redacted]
Capture once
Capture live

3. In the **Image viewer** panel, ensure that the camera can capture images normally and click the **[Next]** button on the bottom bar.

Mount Calibration Board & Check Intrinsic Parameters

1. In the **Mount calibration board & check intrinsic parameters** step, select the calibration board model for the **Standard calibration board model** parameter according to its model nameplate in the **1 Select calibration board** area.


Files
Tools
Connect to camera
2
Mount calibration board & check intrinsic parameters
3
Collect images and flange poses
4
Calculate extrinsic parameters
Open Mech-Eye Viewer
Capture live
Check intrinsic parameters
Draw an aid circle
Calibration circle detection parameters (advanced)

1 Select calibration board

Standard calibaration board model: BDB-5

2 Check calibration board position and point cloud quality

Requirements

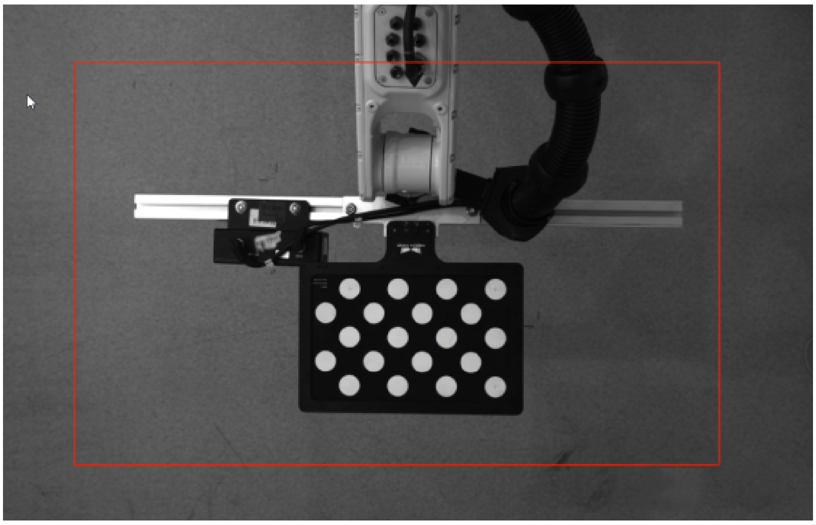
• In the 2D image view,er the calibration board should be completely displayed in the red rectangle.
• The distance between the calibration board and the camera should be as close to 0 mm as possible. The closer to the given distance, the more accurate the checking result. View example
• In the 2D image viewer, the calibration board should not be too dark, and each calibration circle should be clearly visible. View example
• In the Depth map viewer, each calibration circle on the calibration board should be complete and visible. View example
• The point cloud of the calibration board's upper surface should be flat.

3 Check intrinsic parameters

No result yet.

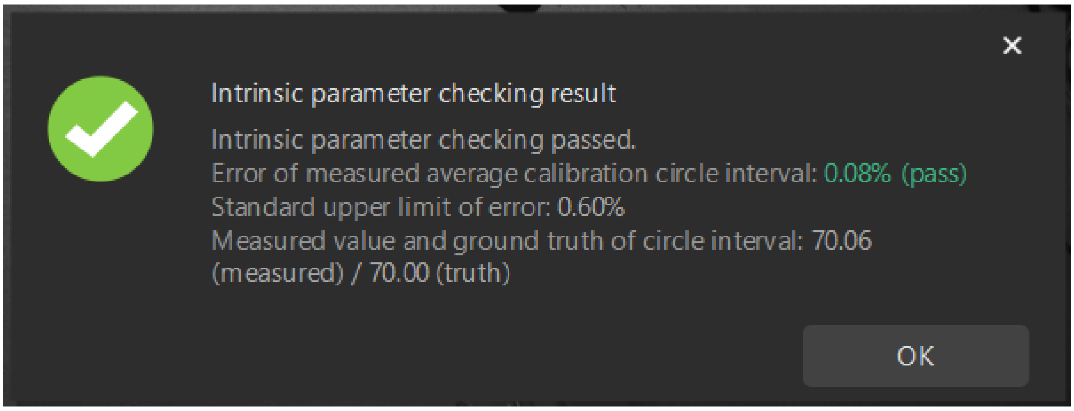
2. In the **2 Check calibration board position and point cloud quality** area, read carefully the requirements on the calibration board position and point cloud quality, and then click the **[Capture live]** button.The **[Capture live]** button will turn into **[Stop capturing and detect position]**.

3. Control the robot to move the calibration board to the proper position (in the red rectangle), and ensure that the 2D image and depth map of the calibration board meet the requirements, and then click the **[Stop capturing and detect position]** button.



If the captured images do not meet the requirements, click [Open Mech-Eye Viewer] button to open the Mech-Eye Viewer software, adjust the 2D and 3D exposure parameters and re-capture images. Please note that you need to change the **Parameter group** parameter to “calib” first.

4. In the **3 Check intrinsic parameters** area, click the [**Check intrinsic parameters**] button.
5. Confirm the results of the camera intrinsic parameter check.
   - If the camera intrinsic parameter check passes, click the [**OK**] button in the prompted window, and then click the [**Next**] button on the bottom bar.



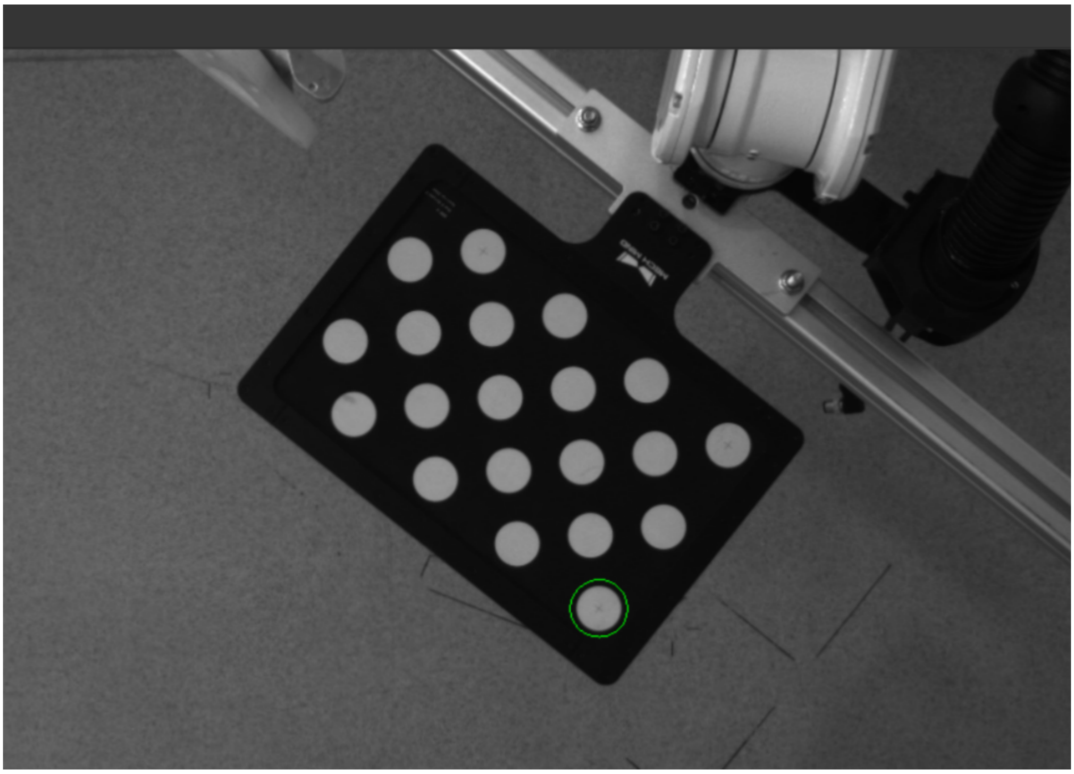
Intrinsic parameter checking result

Intrinsic parameter checking passed.
Error of measured average calibration circle interval: 0.08% (pass)
Standard upper limit of error: 0.60%
Measured value and ground truth of circle interval: 70.06 (measured) / 70.00 (truth)

   - If the camera intrinsic parameter check fails, adjust detection parameters by drawing aid circles or manually adjusting the calibration circle detection parameters, and then click the [**Recheck intrinsic parameters**] button.

Draw an Aid Circle

1. To draw an aid circle, click the [**Draw an aid circle**] button.
2. In the right **Image viewer** panel, right-click the calibration board image and clear the **Fit to window** checkbox, and press the `Ctrl` key and drag the roller to adjust the image to a suitable size.
3. Move the mouse pointer to the cross center point of the calibration circle, press the left mouse button and make the aid circle completely include the calibration circle and then release it.



4. Click the [**Recheck intrinsic parameters**] button, and confirm that the camera intrinsic parameter check passes.

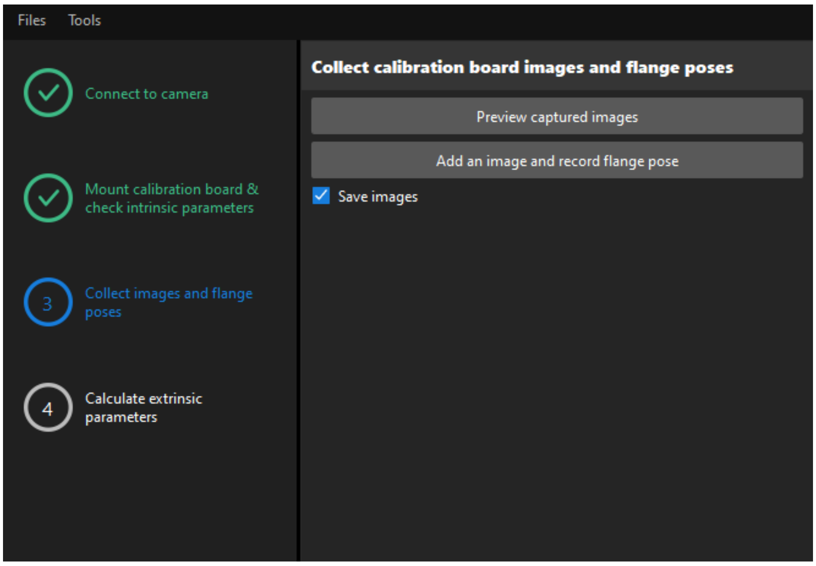
Manually Adjust Detection Parameters

To manually adjust detection parameters, click [**Calibration circle detection parameters (advanced)**], adjust parameter settings and click the [**Recheck intrinsic parameters**] to confirm whether the camera intrinsic parameter check passes.

If the calibration circles are still not detected, please adjust the camera settings according to the on-site operating conditions.

Collect Images and Flange Poses

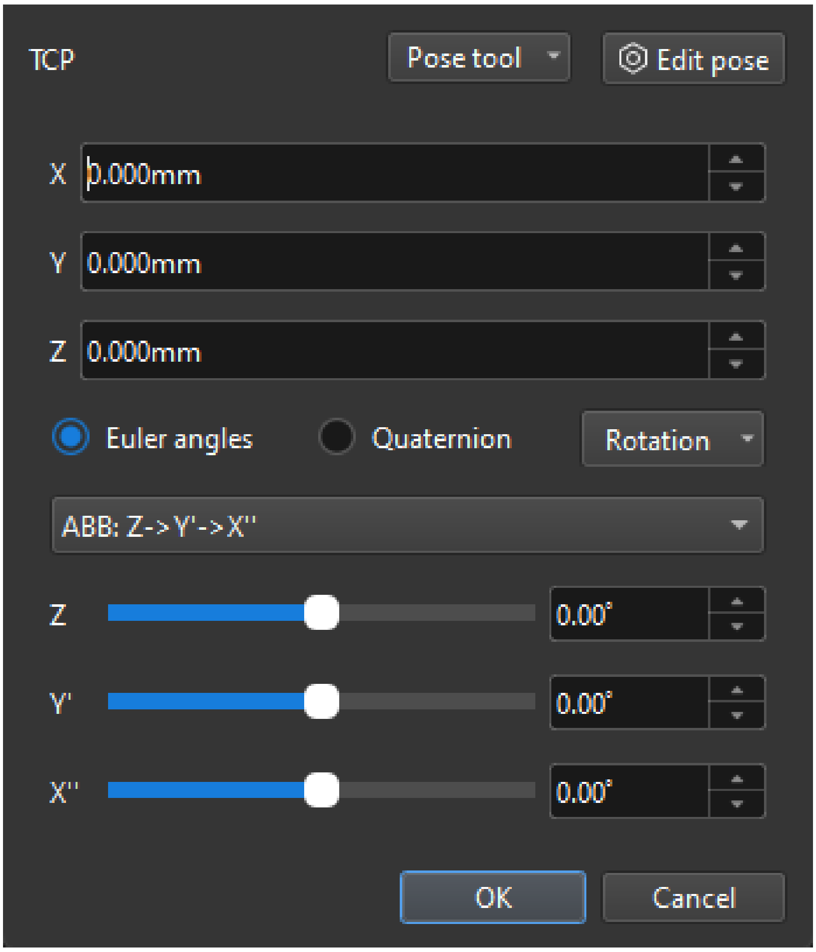
1. In the **Collect images and flange poses** step, select the [**Save images**] checkbox (selected by default, recommended). If you do not need to save images of calibration boards, clear this checkbox.


Files
Tools
Connect to camera
Mount calibration board & check intrinsic parameters
3
Collect images and flange poses
4
Calculate extrinsic parameters
Preview captured images
Add an image and record flange pose

Collect calibration board images and flange poses

☑ Save images

2. Control the robot to move to different calibration points and then click the [**Add an image and record flange pose**] button.
3. In the prompted window, enter the robot flange pose.



X 0.000mm

Y 0.000mm

Z 0.000mm

● Euler angles ○ Quaternion

ABB: Z->Y'->X''

Z 0.00°

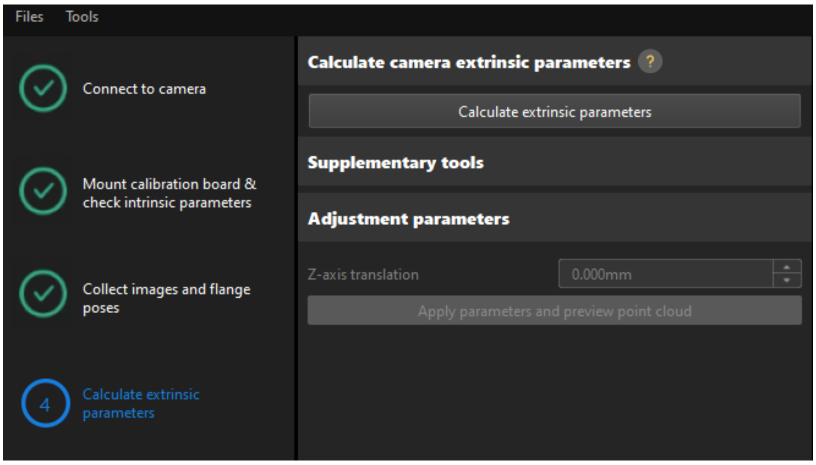
Y' 0.00°

X'' 0.00°

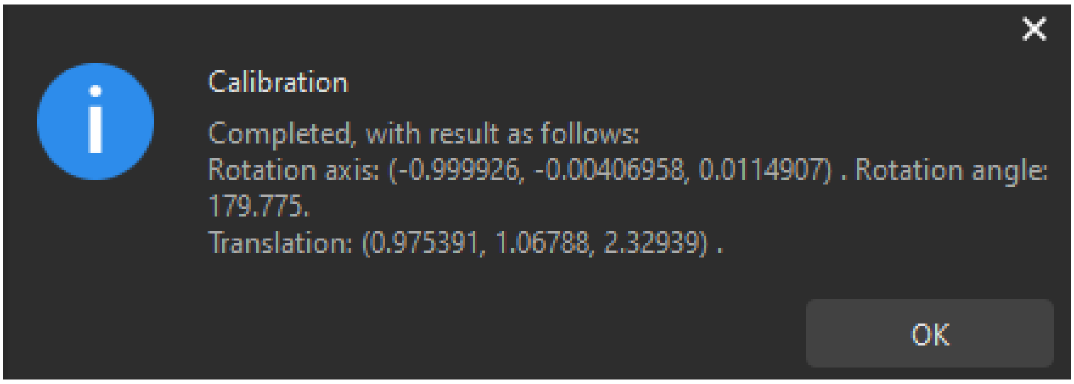
4. Confirm that the number of the currently recognized calibration circles meets the requirements, and then click the [**Next**] button on the bottom bar.

Calculate Extrinsic Parameters

1. In the **Calculate extrinsic parameters** step, click the **[Calculate extrinsic parameters]** button.


Files
Tools
Connect to camera
Mount calibration board & check intrinsic parameters
Collect images and flange poses
4
Calculate extrinsic parameters
Calculate camera extrinsic parameters
?
Calculate extrinsic parameters
Supplementary tools
Adjustment parameters
Z-axis translation
0.000mm
Apply parameters and preview point cloud

2. In the prompted window indicating calibration success, click the **OK** button.


Calibration

Completed, with result as follows:
Rotation axis: (-0.999926, -0.00406958, 0.0114907) . Rotation angle: 179.775.
Translation: (0.975391, 1.06788, 2.32939) .

To evaluate the robot absolute accuracy and the accuracy of calculated extrinsic parameters, you can click **[Robot absolute accuracy and repeatability]** and **[Extrinsic parameter accuracy]** in the “Calculate extrinsic parameters” window and use the error analysis tool to perform robot absolute accuracy check and analyze extrinsic parameter error in ETH setup.

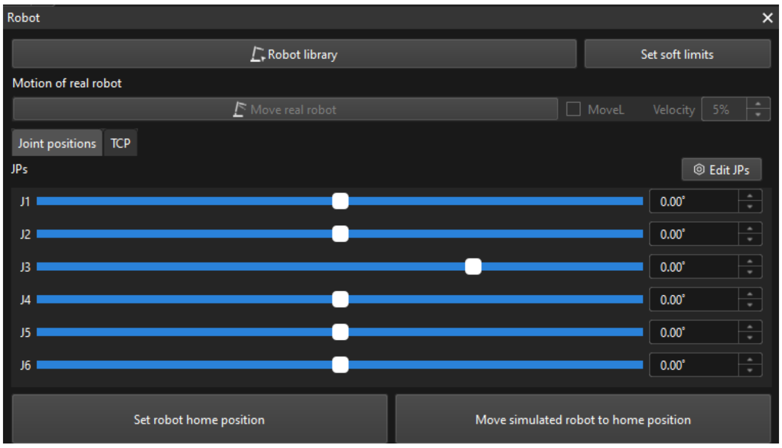
3. Adjust the Z-axis value of the extrinsic parameters.

Due to the lack of **degrees of freedom in rotation** in four-axis robots, the captured images during calibration may lack the **images for calibration board rotation**. After calibration, you need to manually adjust the Z-axis value of the extrinsic parameters.

a. Control the robot to move to a position in the camera’s field of view, and then click the **[Apply parameters and preview point cloud]** button in the **Adjust parameters** area.

b. In Mech-Viz, confirm whether the robot’s actual point cloud coincides with the robot model.

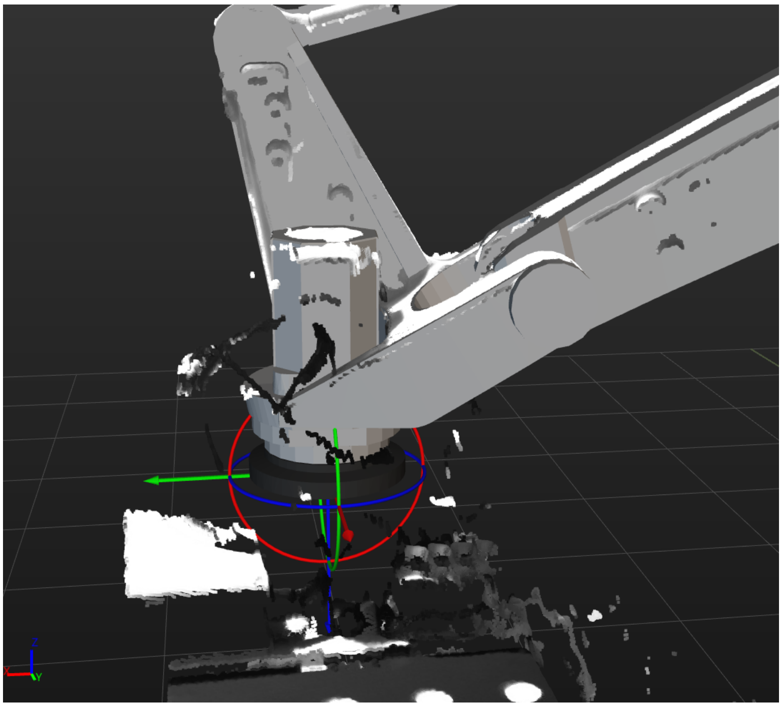
c. Check the robot’s current pose in the teach pendant, and enter the robot’s current JPs in the **Robot** panel of Mech-Viz.



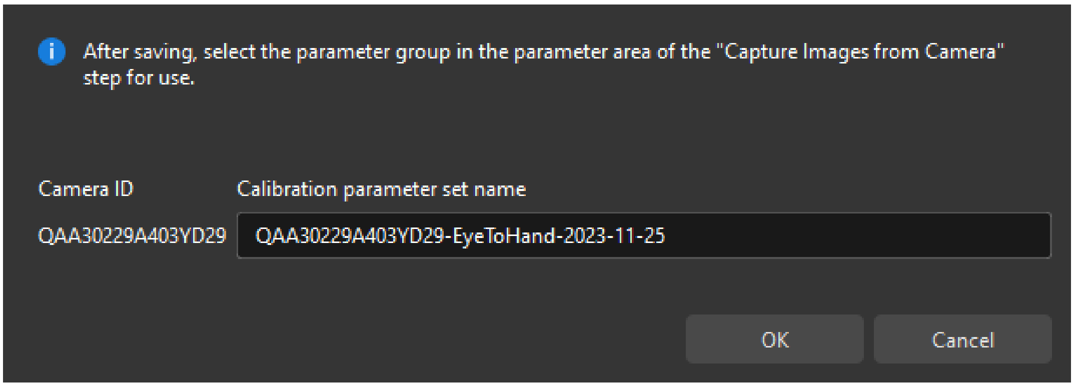
Velocity 5%

J1 0.00°
J2 0.00°
J3 0.00°
J4 0.00°
J5 0.00°
J6 0.00°

d. In the 3D Simulation Area of Mech-Viz, confirm whether the robot’s actual point cloud coincides with the robot model. If yes, proceed to the next step. Otherwise modify the value of the **Z-axis translation** parameter in Mech-Vision, and click the **[Apply parameters and preview point cloud]** button. Repeat this operation until the robot’s actual point cloud coincides with the robot model.



4. Click the **[Save]** button on the bottom bar. In the prompted **Save Calibration Files** dialog box, click the **[OK]** button. The camera calibration result will be automatically saved in the “calibration” directory of the project.



After saving, select the parameter group in the parameter area of the "Capture Images from Camera" step for use.

Camera ID: QAA30229A403YD29

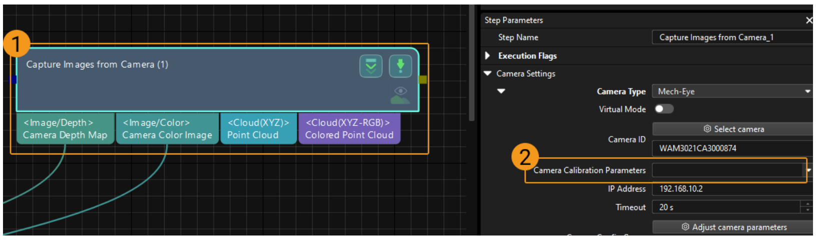
Calibration parameter set name: QAA30229A403YD29-EyeToHand-2023-11-25

Till now, the calibration process is completed.

Change the Extrinsic Parameter File

If the camera’s extrinsic parameter file has been used by a Mech-Vision project, you need to change the extrinsic parameter file used by the Mech-Vision project to a new one after you finish the hand-eye calibration.

1. Select the **Capture Images from Camera** Step.



1
Capture Images from Camera (1)
<Image/Depth> Camera Depth Map
<Image/Color> Camera Color Image
<Cloud(XYZ)> Point Cloud
<Cloud(XYZ-RGB)> Colored Point Cloud
Step Parameters
Execution Flags
Camera Settings
Select camera
2
Adjust camera parameters

Step Name: Capture Images from Camera_1

Camera Type: Mech-Eye

Virtual Mode

Camera ID: WAM3021CA3000874

Camera Calibration Parameters

IP Address: 192.168.10.2

Timeout: 20 s

2. In the **Step Parameters** panel, click ▼ of the **Camera Calibration Parameters** parameter, and select the newly calibrated extrinsic parameter file.

Calibration-Related Status Codes

If the robot uses the Standard Interface to communicate with the Vision System during hand-eye calibration, the Vision System will return status codes in response data to indicate the execution status of the commands. The status codes indicate both the normal execution results and execution failures. You can perform further troubleshooting according to the status codes.

Calibration Error Codes

Error code	Description
7001	Calibration: Parameter error
7002	Calibration: Calibration flange pose not provided by Mech-Vision
7003	Calibration: Calibration joint positions not provided by Mech-Vision
7004	Calibration: Robot failed to reach calibration point

Calibration Normal Status Codes

Normal Status Code	Description
7100	Calibration: Robot reaches calibration point successfully
7101	Calibration: Pose received from Mech-Vision successfully

4.5.2.5. Manual Calibration in the Eye-to-Hand Setup (Six-Axis Robots - Using TCP Touch Method)

This how-to guide introduces how to complete the manual calibration for six-axis robots using the TCP touch method in the eye-to-hand (ETH) setup.

Preparation before Calibration

Before hand-eye calibration, you need to finish the following preparations:

- Construct the Mech-Mind Vision System.
- Prepare the materials required for calibration.
- Adjust the effect of calibration board’s point cloud.

Construct the Vision System

Construct the Mech-Mind Vision System by referring to the section Vision System Hardware Setup.

You need to use Mech-Eye Viewer, Mech-Vision and Mech-Viz during hand-eye calibration. Please ensure that they have been installed and are running the latest versions.

Prepare the Materials Required for Calibration

The manual calibration in the ETH setup needs to use the calibration board and sharp tip.

Please prepare the calibration board and the sharp tip according to the following requirements:

- Ensure that the circles of the calibration board are clearly visible and without obvious scratches, and the board does not suffer from deformations.
- When the TCP touch method is used to collect calibration data in the ETH setup, please place the calibration board in the center of the working plane, where target objects are to be placed.
- The sharp tip should be mounted on the center of the robot flange. If an undetachable gripper is connected to the robot flange, you can attach the sharp point directly to the gripper.

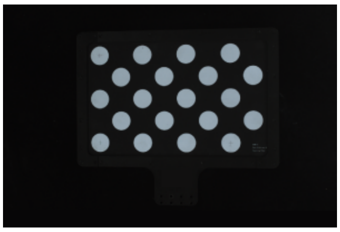
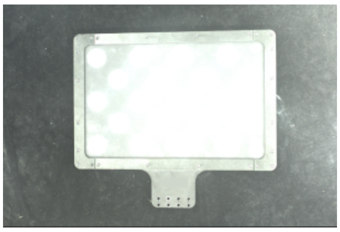
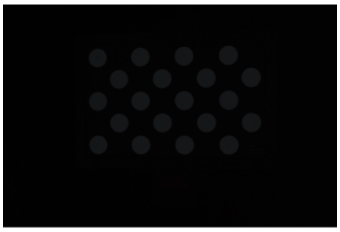
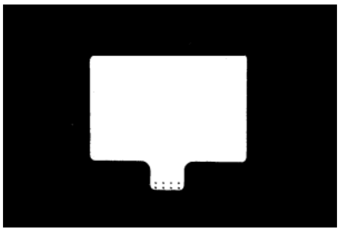
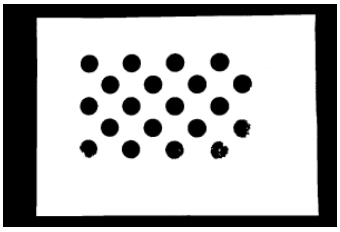
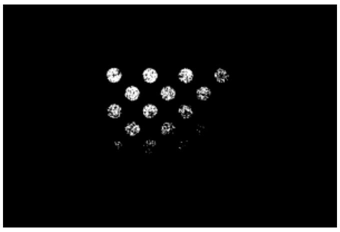
In addition, before calibration, move the robot to the starting point for calibration to ensure that the robot will not block the camera from collecting images of the calibration board.

Adjust the Effect of Calibration Board’s Point Cloud

1. Open Mech-Eye Viewer to adjust camera parameters.
2. Adjust the 2D parameters to make sure that the 2D image of the calibration board is clear and neither overexposed nor underexposed.
3. Adjust 3D parameters to make sure that the point clouds of the circles on the calibration board are complete and have clear contours. It is recommended to set **Surface Smoothing** and **Outlier Removal** to **Normal** in the **Point Cloud Processing** section to reduce the point cloud fluctuation range.

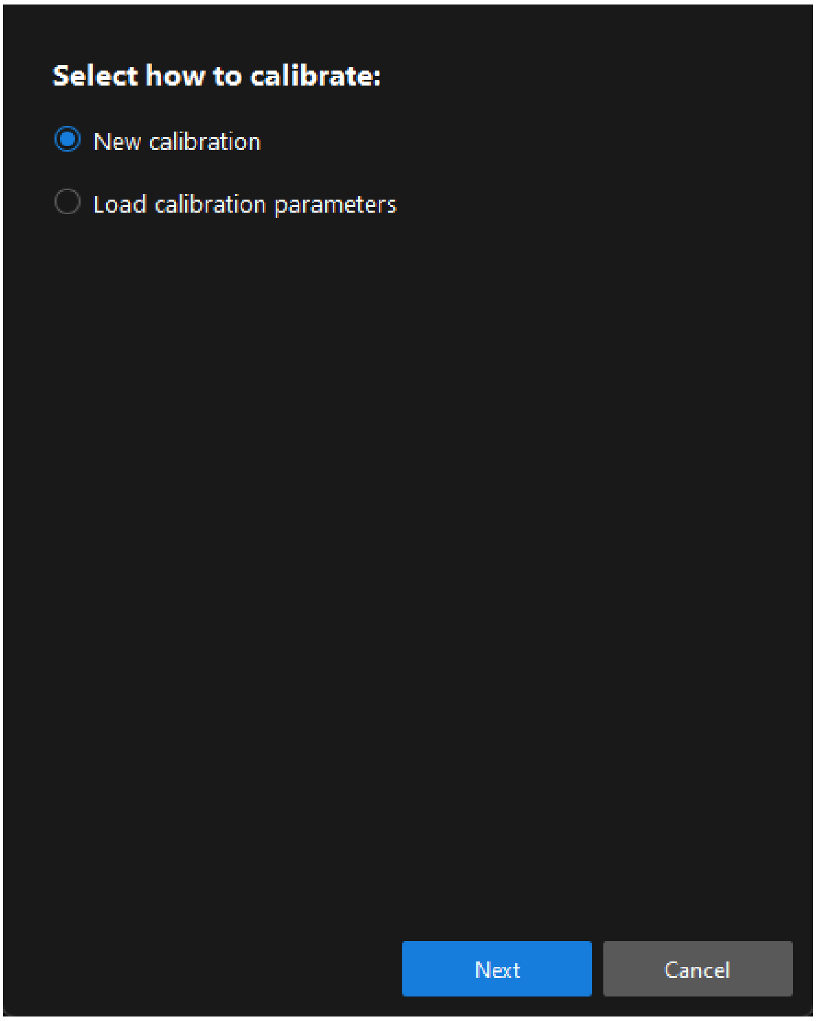
If the on-site ambient lights are not ideal and affect the quality of 2D images and point clouds, you can use shading or supplemental light to improve the lighting conditions.

4. Make sure that the images and point cloud of the calibration board are up to standard after completing the preceding steps.

	Normal	Overexposed	Underexposed
2D image			
Point cloud			

Pre-calibration Configuration

1. Open Mech-Vision, and click the **[Camera Calibration (Standard)]** button in the toolbar. The **Configuration before Calibration** window will be prompted.
2. In the **Select how to calibrate** window, select the **New calibration** radio button, and then click the **[Next]** button.

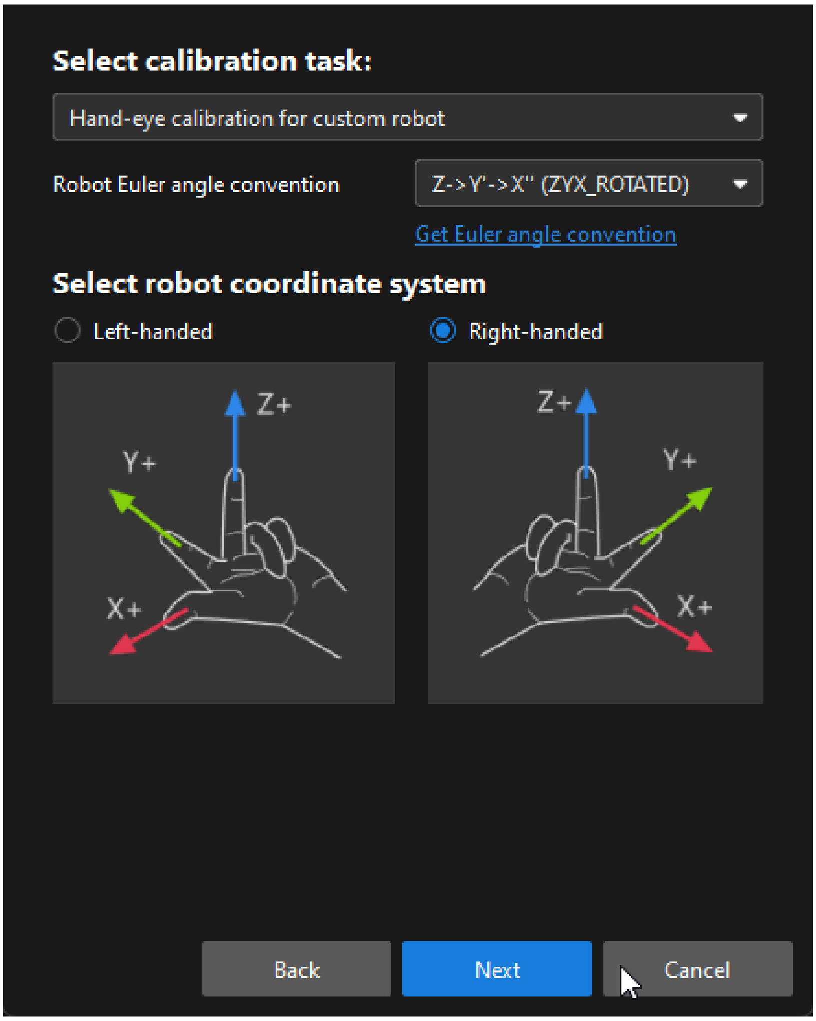


Select how to calibrate:

● New calibration

○ Load calibration parameters

3. In the **Select calibration task** window, select **Hand-eye calibration for custom robot** from the drop-down list box, specify the **Robot Euler angle convention** parameter, select the robot coordinate system type, and then click the **[Next]** button.



Select calibration task:

Hand-eye calibration for custom robot

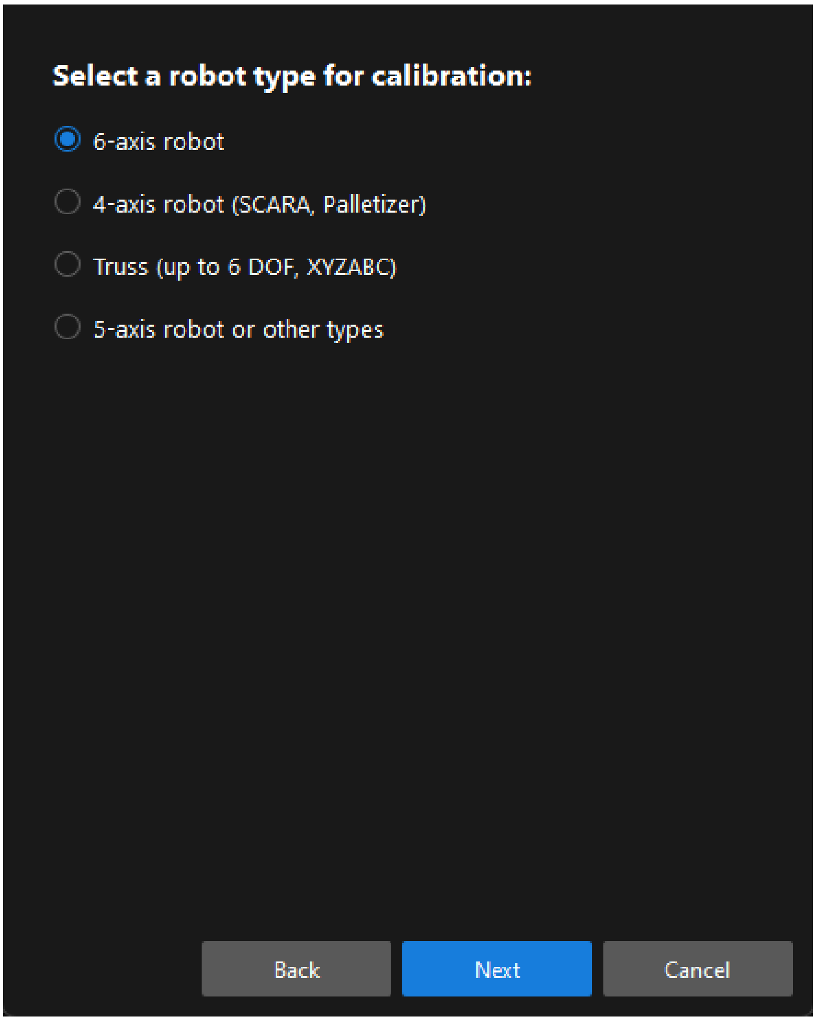
Robot Euler angle convention: Z->Y'->X'' (ZYX_ROTATED)

Get Euler angle convention

Select robot coordinate system

○ Left-handed ● Right-handed

4. In the **Select a robot type for calibration** window, select the **6-axis robot** radio button, and then click the **[Next]** button.



Select a robot type for calibration:

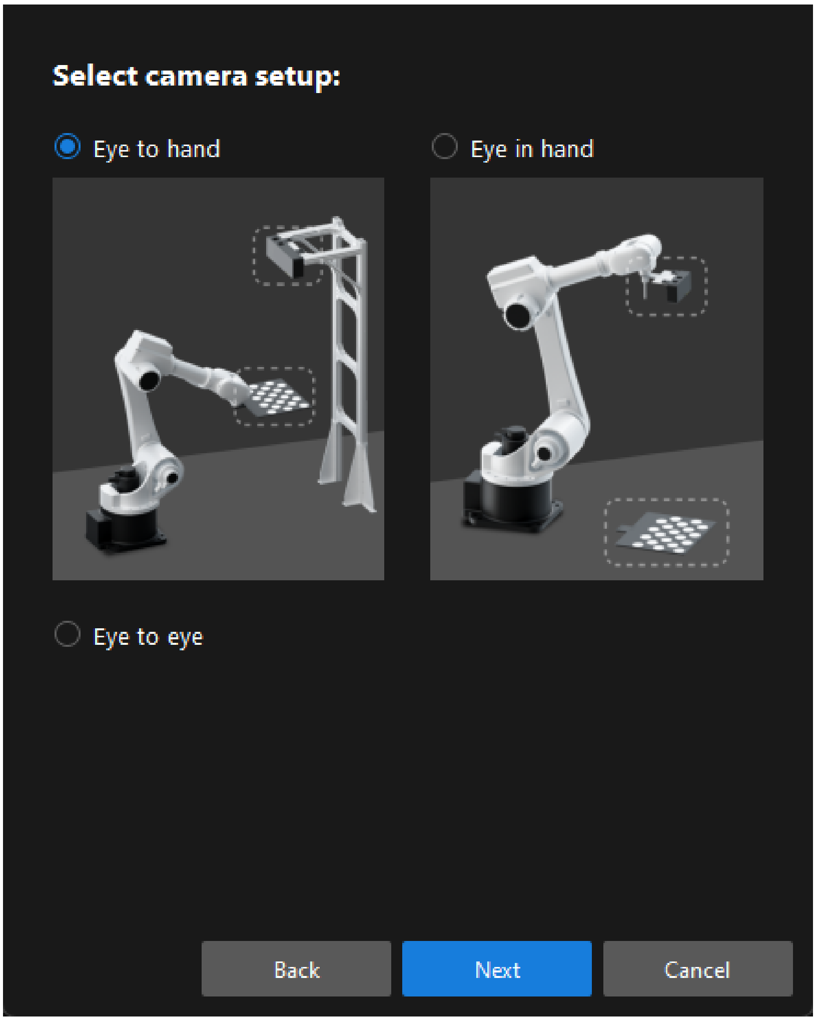
● 6-axis robot

○ 4-axis robot (SCARA, Palletizer)

○ Truss (up to 6 DOF, XYZABC)

○ 5-axis robot or other types

5. In the **Select camera setup** window, select the **Eye to hand** radio button, and then click the [Next] button.



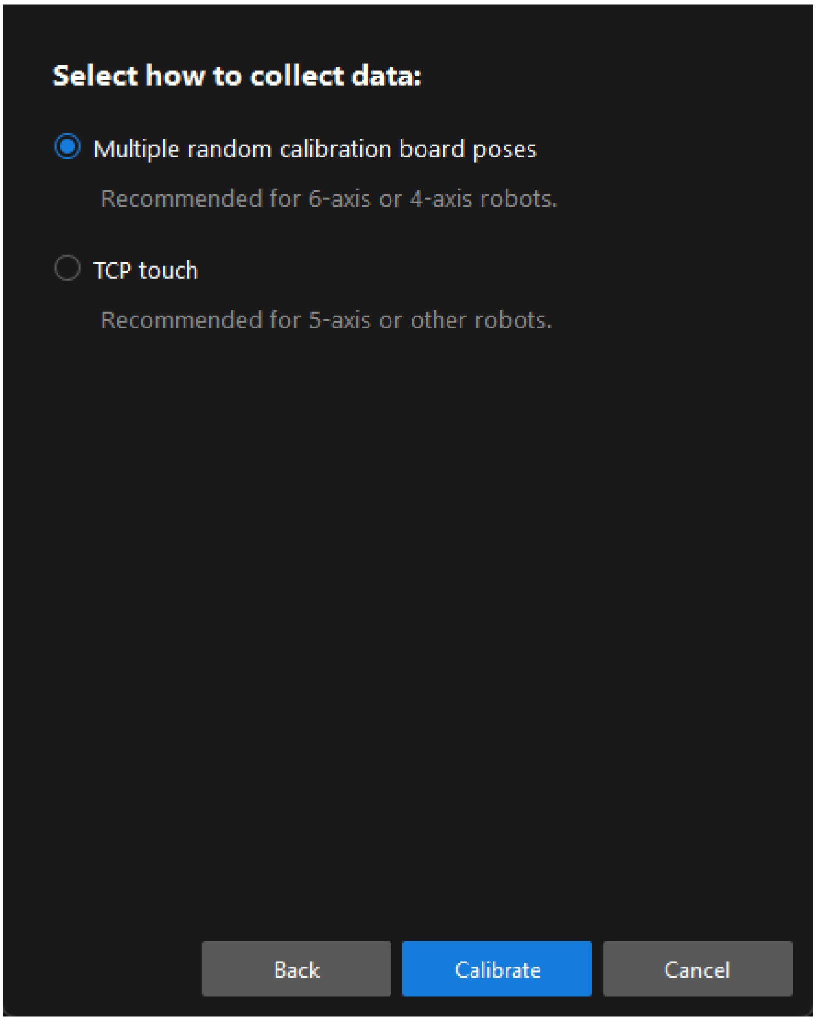
Select camera setup:

● Eye to hand

○ Eye in hand

○ Eye to eye

6. In the **Select how to collect data** window, select the **TCP touch** radio button, and then click the **[Calibrate]** button.The **Calibration (Eye to hand)** window will be prompted.


Select how to collect data:

● Multiple random calibration board poses

Recommended for 6-axis or 4-axis robots.

○ TCP touch

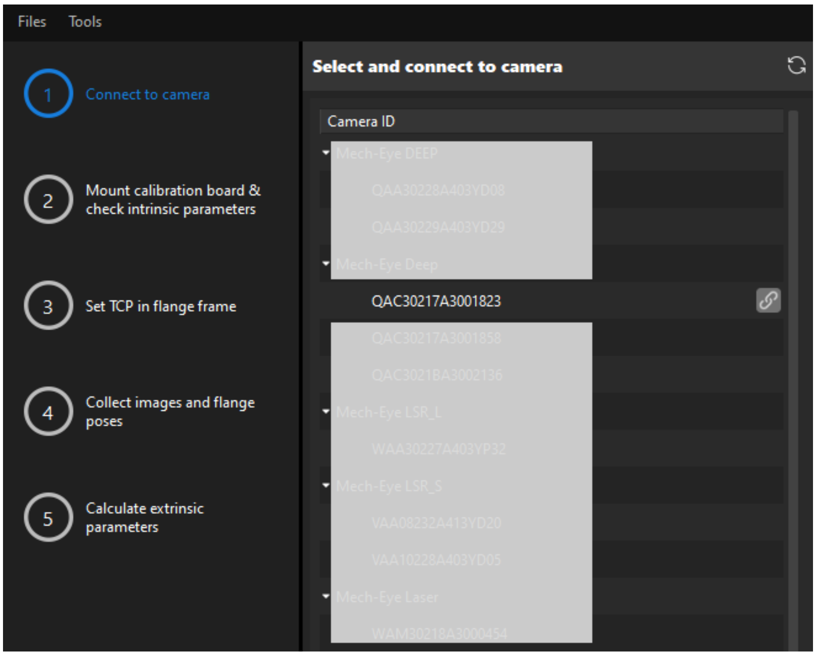
Recommended for 5-axis or other robots.

Till now, you have completed the pre-calibration configuration and can start the calibration procedure.

Calibration Procedure

Connect to the Camera

1. In the **Connect to camera** step, select the camera to connect in the Camera ID list, and then click the 🔗 button or double-click the camera entry to connect to it.



1 Connect to camera

2 Mount calibration board & check intrinsic parameters

3 Set TCP in flange frame

4 Collect images and flange poses

5 Calculate extrinsic parameters

Select and connect to camera

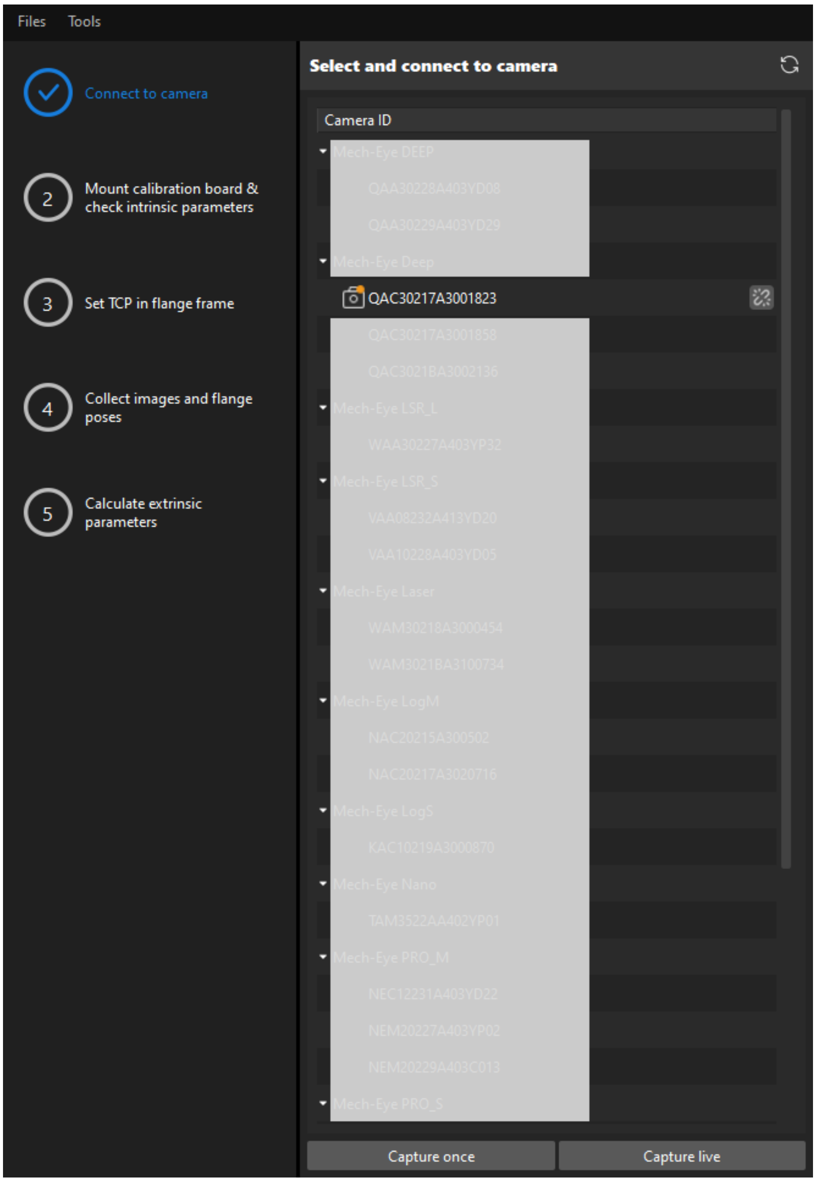
Camera ID

[redacted]

QAC30217A3001823

[redacted]

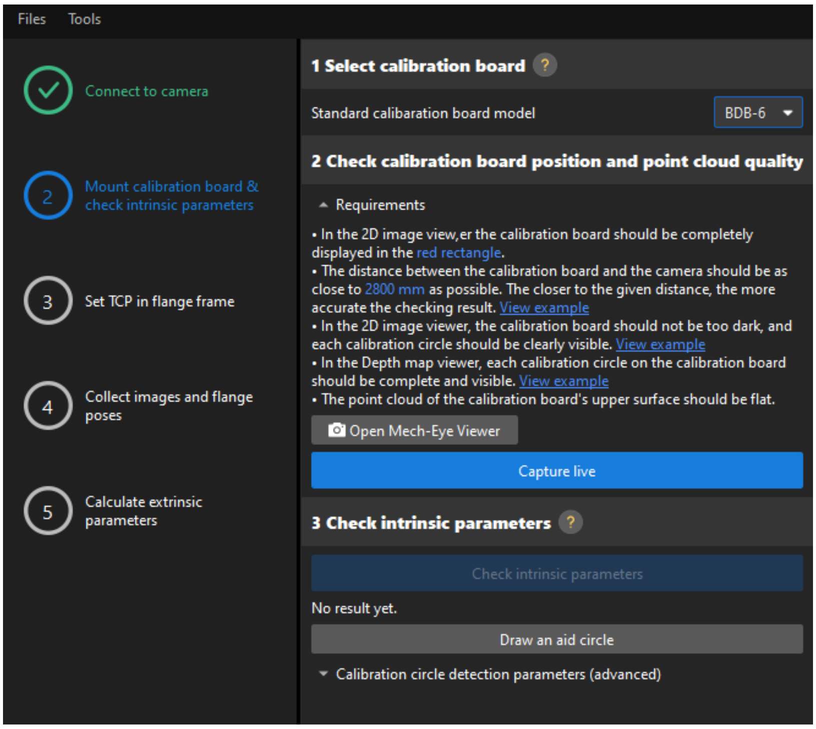
2. After the camera is connected, click the **[Capture once]** or **[Capture live]** button.



3. In the **Image viewer** panel, ensure that the camera can capture images normally and click the **[Next]** button on the bottom bar.

Mount Calibration Board & Check Intrinsic Parameters

1. In the **Mount calibration board & check intrinsic parameters** step, select the calibration board model for the **Standard calibration board model** parameter according to its model nameplate in the **1 Select calibration board** area.


Files
Tools
Connect to camera
2
Mount calibration board & check intrinsic parameters
3
Set TCP in flange frame
4
Collect images and flange poses
5
Calculate extrinsic parameters
Open Mech-Eye Viewer
Capture live
Check intrinsic parameters
Draw an aid circle

1 Select calibration board ?

Standard calibaration board model: BDB-6

2 Check calibration board position and point cloud quality

Requirements

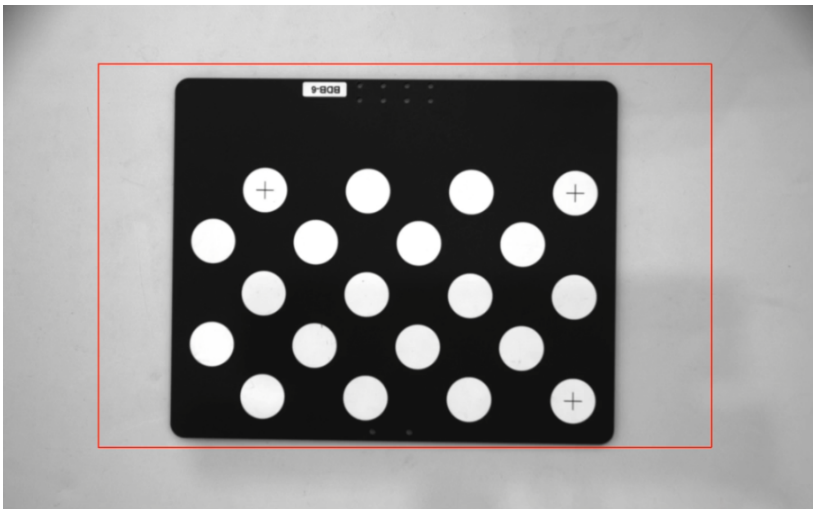
- In the 2D image view,er the calibration board should be completely displayed in the red rectangle.
- The distance between the calibration board and the camera should be as close to 2800 mm as possible. The closer to the given distance, the more accurate the checking result. View example
- In the 2D image viewer, the calibration board should not be too dark, and each calibration circle should be clearly visible. View example
- In the Depth map viewer, each calibration circle on the calibration board should be complete and visible. View example
- The point cloud of the calibration board's upper surface should be flat.

3 Check intrinsic parameters ?

No result yet.

Calibration circle detection parameters (advanced)

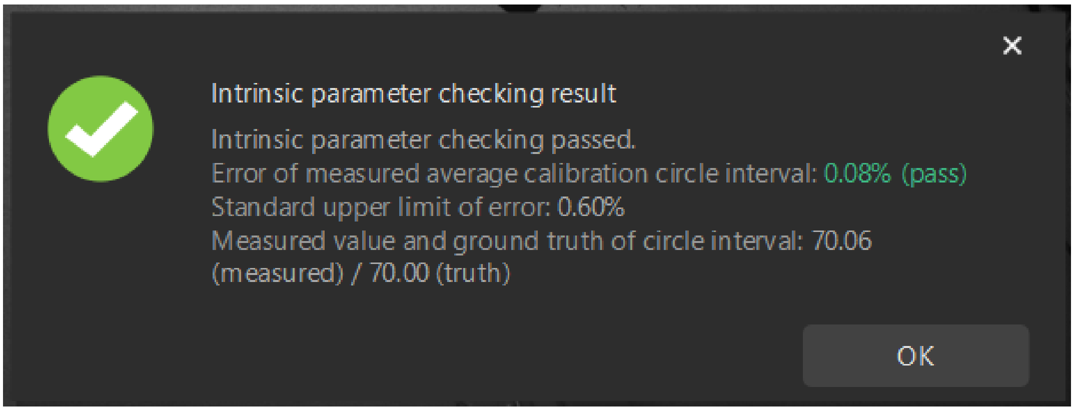
2. In the **2 Check calibration board position and point cloud quality** area, read carefully the requirements on the calibration board position and point cloud quality, and then click the [**Capture live**] button.The [**Capture live**] button will turn into [**Stop capturing and detect position**].
3. Manually move the calibration board to the proper position (in the red rectangle), and ensure that the 2D image and depth map of the calibration board meet the requirements, and then click the [**Stop capturing and detect position**] button.



> If the captured images do not meet the requirements, click the [**Open Mech-Eye Viewer**] button to open the Mech-Eye Viewer software, adjust the 2D and 3D exposure parameters and re-capture images. Please note that you need to change the **Parameter group** parameter to “calib” first.

4. In the **3 Check intrinsic parameters** area, click the [**Check intrinsic parameters**] button.
5. Confirm the results of the camera intrinsic parameter check.
   - If the camera intrinsic parameter check passes, click the [**OK**] button in the prompted

window, and then click the [**Next**] button on the bottom bar.



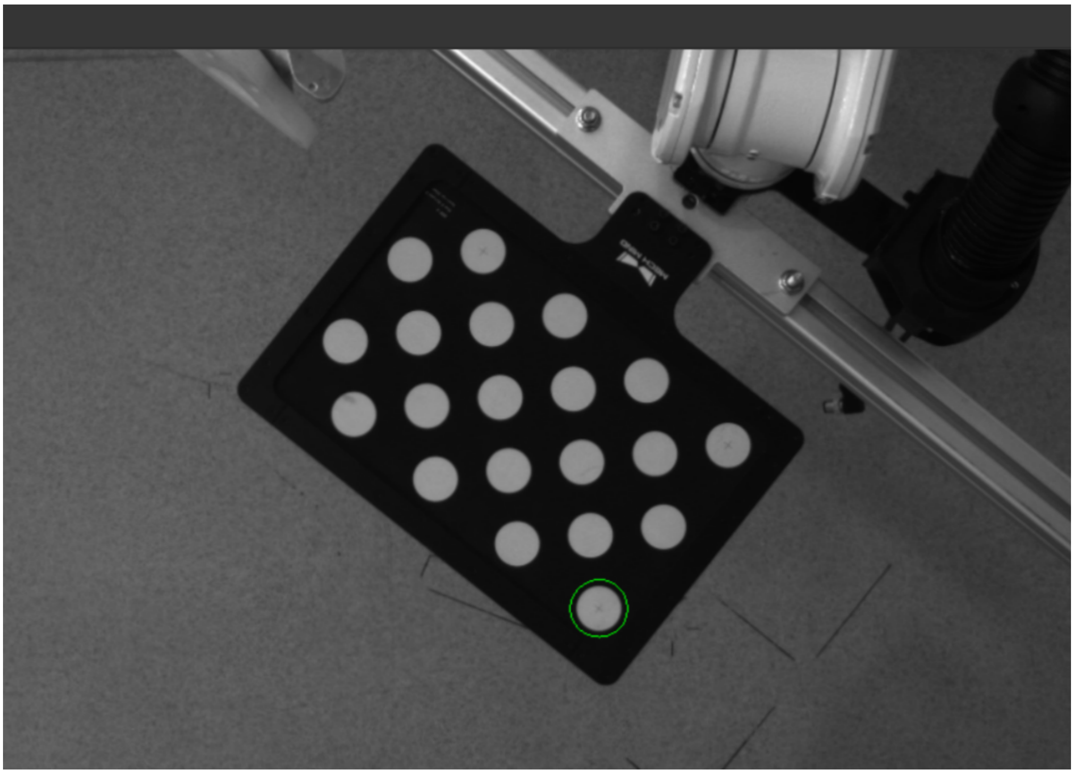
Intrinsic parameter checking result

Intrinsic parameter checking passed.
Error of measured average calibration circle interval: 0.08% (pass)
Standard upper limit of error: 0.60%
Measured value and ground truth of circle interval: 70.06 (measured) / 70.00 (truth)

- If the camera intrinsic parameter check fails, adjust detection parameters by drawing aid circles or manually adjusting the calibration circle detection parameters, and then click the [**Recheck intrinsic parameters**] button.

Draw an Aid Circle

1. To draw an aid circle, click the [**Draw an aid circle**] button.
2. In the right **Image viewer** panel, right-click the calibration board image and clear the **Fit to window** checkbox, and press the `Ctrl` key and drag the roller to adjust the image to a suitable size.
3. Move the mouse pointer to the cross center point of the calibration circle, press the left mouse button and make the aid circle completely include the calibration circle and then release it.



4. Click the [**Recheck intrinsic parameters**] button, and confirm that the camera intrinsic parameter check passes.

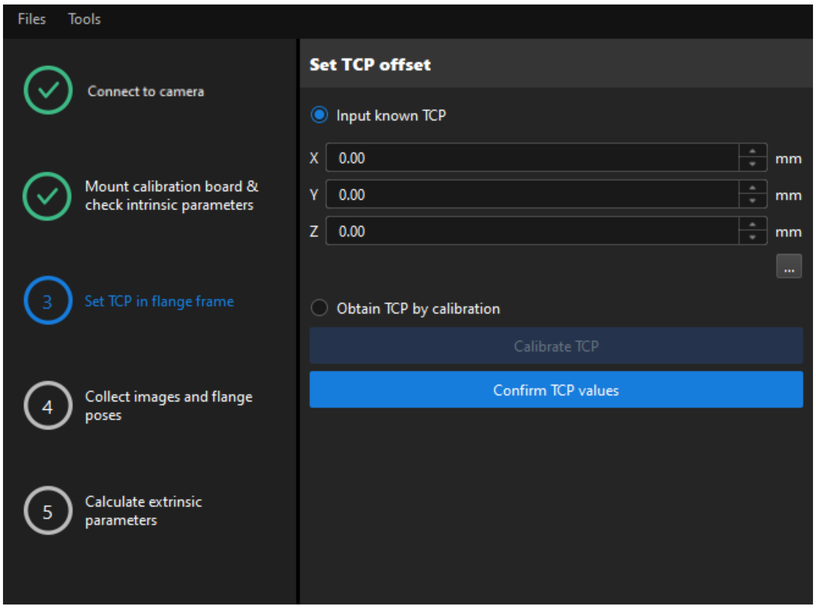
Manually Adjust Detection Parameters

To manually adjust detection parameters, click [**Calibration circle detection parameters (advanced)**], adjust parameter settings and click the [**Recheck intrinsic parameters**] button to confirm whether the camera intrinsic parameter check passes.

If the calibration circles are still not detected, please adjust the camera settings according to the on-site operating conditions.

Set TCP in Flange Frame

1. In the **Set TCP in flange frame** step, select the **Input known TCP** radio button, and enter the TCP values.


Files
Tools
Connect to camera
Mount calibration board & check intrinsic parameters
3
Set TCP in flange frame
4
Collect images and flange poses
5
Calculate extrinsic parameters
Set TCP offset
...
Calibrate TCP
Confirm TCP values

● Input known TCP

X 0.00 mm

Y 0.00 mm

Z 0.00 mm

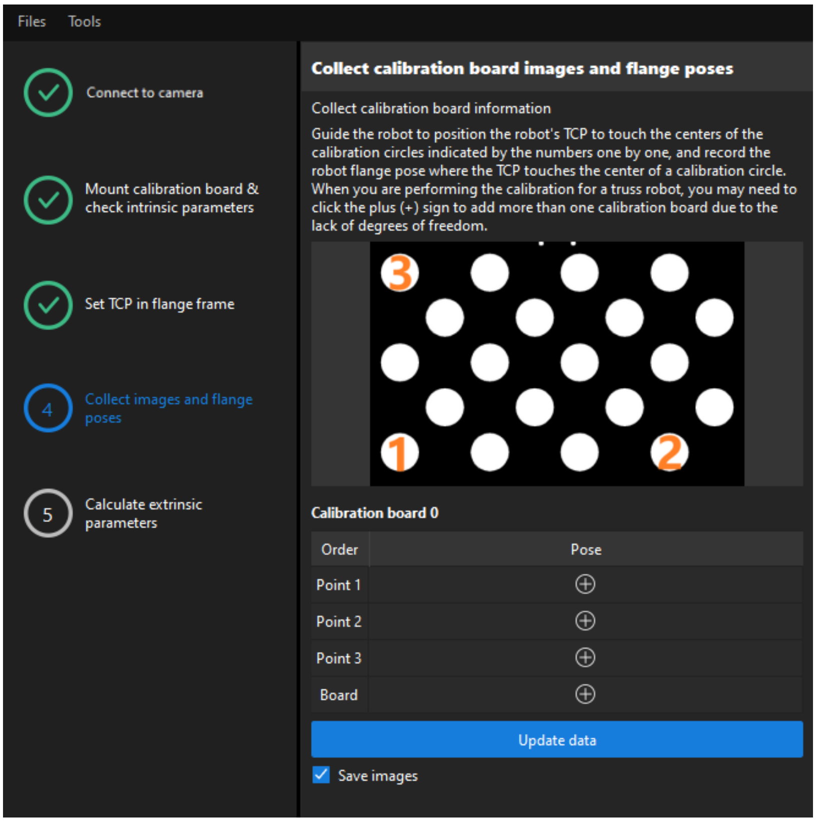
○ Obtain TCP by calibration

2. Click the [**Confirm TCP values**] button.
3. Click the [**Next**] button on the bottom bar.

> If the TCP values are unknown, select the **Obtain TCP by calibration** radio button, and then click the [**Calibrate TCP**] button. You can use the **TCP calibration** tool to calculate the TCP values.

Collect Images and Flange Poses

1. In the **Collect images and flange poses** step, select the **Save images** checkbox.



Files
Tools
Connect to camera
Mount calibration board & check intrinsic parameters
Set TCP in flange frame
4
Collect images and flange poses
5
Calculate extrinsic parameters
3
1
2
Update data

Collect calibration board images and flange poses

Collect calibration board information

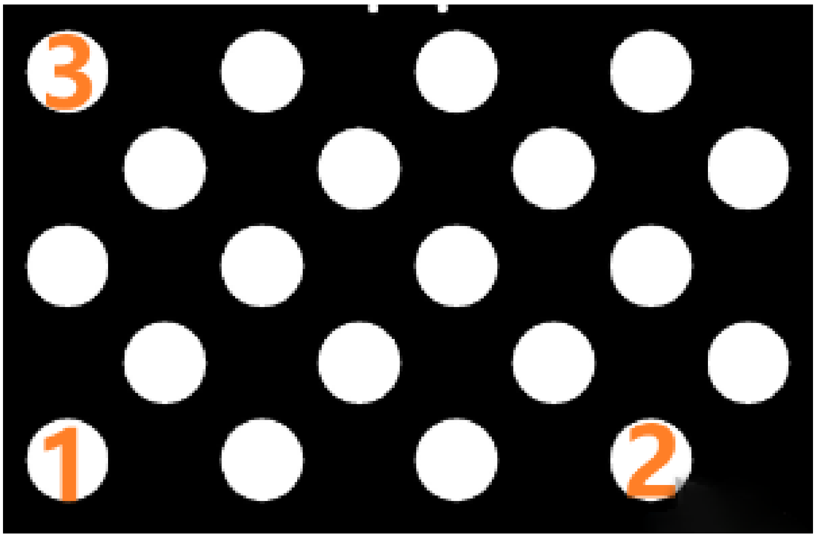
Guide the robot to position the robot's TCP to touch the centers of the calibration circles indicated by the numbers one by one, and record the robot flange pose where the TCP touches the center of a calibration circle. When you are performing the calibration for a truss robot, you may need to click the plus (+) sign to add more than one calibration board due to the lack of degrees of freedom.

Calibration board 0

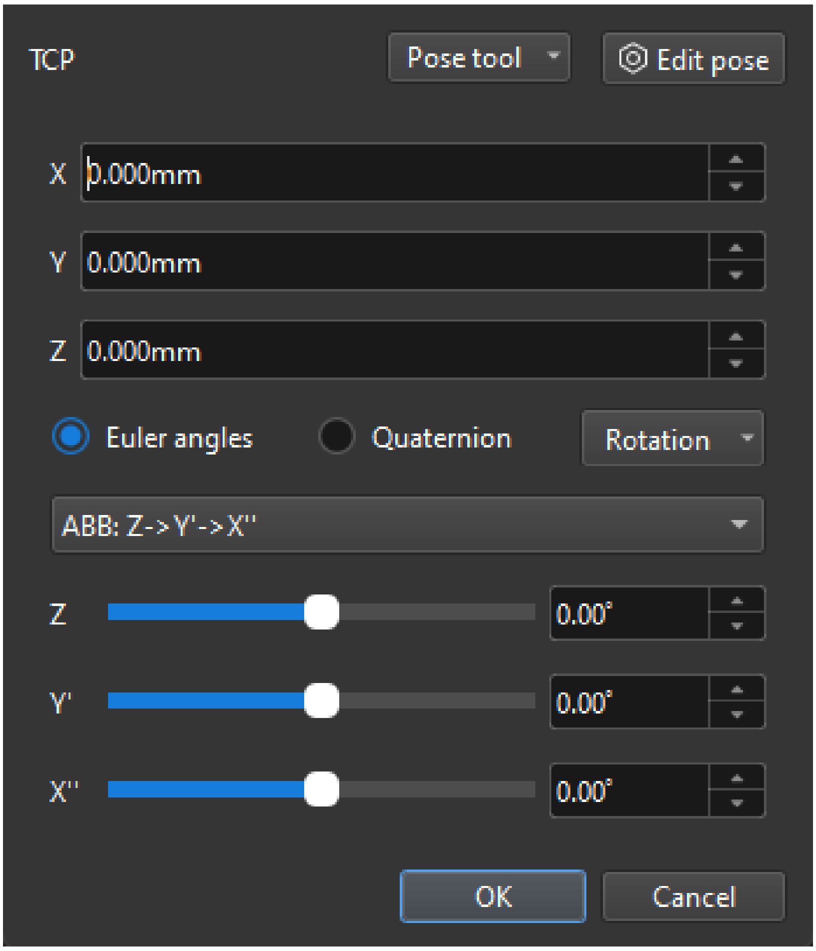
Order	Pose
Point 1	⊕
Point 2	⊕
Point 3	⊕
Board	⊕

☑ Save images

2. Control the robot to let its end tip touch the cross center point of calibration circle “point 1” on the calibration board, and record the robot flange pose in the teach pendant.



3. Click the ⊕ button of point 1, and enter the robot flange pose in the prompted **Input Flange Pose of Robot** dialog box, and then click the [OK] button.



X 0.000mm

Y 0.000mm

Z 0.000mm

● Euler angles ○ Quaternion Rotation

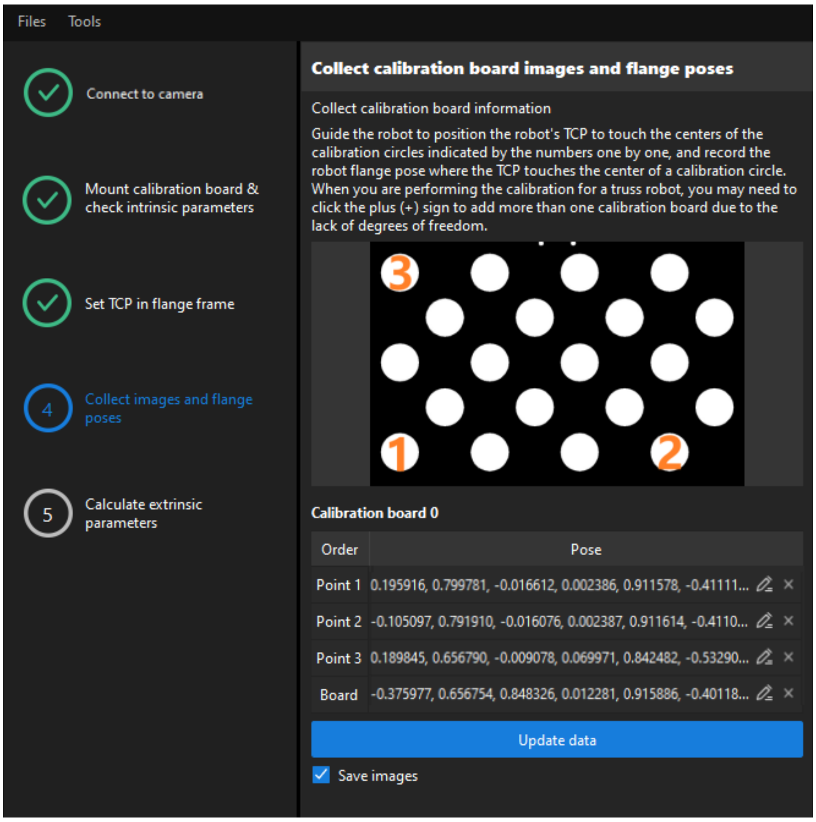
ABB: Z->Y'->X''

Z 0.00°

Y' 0.00°

X'' 0.00°

4. Repeat the preceding steps, and let the end tip touch the cross center points of point 2 and point 3, and enter robot flange poses.
5. Click the ⊕ button of the current calibration board to capture the images of it.
6. Click the **[Update data]** button and then click the **[Next]** button on the bottom bar.



Files
Tools
Connect to camera
Mount calibration board & check intrinsic parameters
Set TCP in flange frame
4
Collect images and flange poses
5
Calculate extrinsic parameters
3
1
2

Collect calibration board images and flange poses

Collect calibration board information

Guide the robot to position the robot's TCP to touch the centers of the calibration circles indicated by the numbers one by one, and record the robot flange pose where the TCP touches the center of a calibration circle. When you are performing the calibration for a truss robot, you may need to click the plus (+) sign to add more than one calibration board due to the lack of degrees of freedom.

Calibration board 0

Order	Pose
Point 1	0.195916, 0.799781, -0.016612, 0.002386, 0.911578, -0.41111...
Point 2	-0.105097, 0.791910, -0.016076, 0.002387, 0.911614, -0.4110...
Point 3	0.189845, 0.656790, -0.009078, 0.069971, 0.842482, -0.53290...
Board	-0.375977, 0.656754, 0.848326, 0.012281, 0.915886, -0.40118...

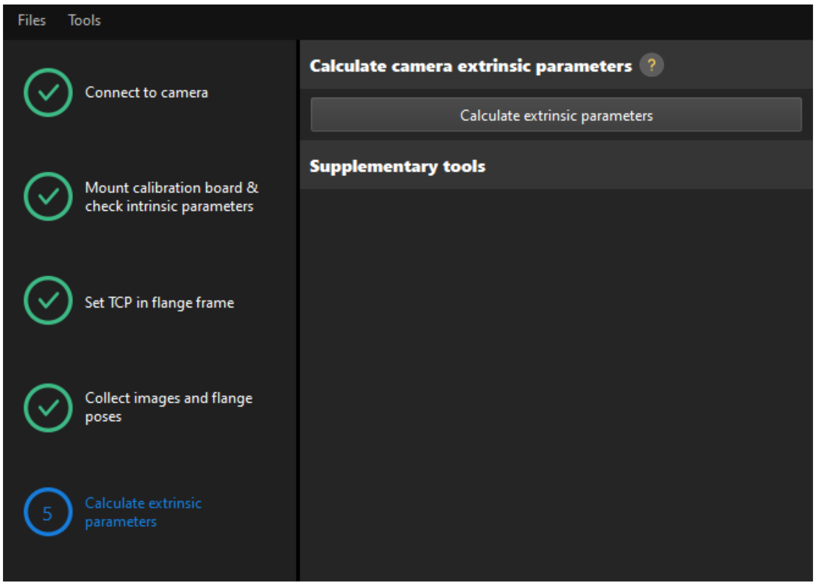
Update data

☑ Save images

The calibration program requires that at least three points not in a line should be touched for calculating extrinsic parameters.

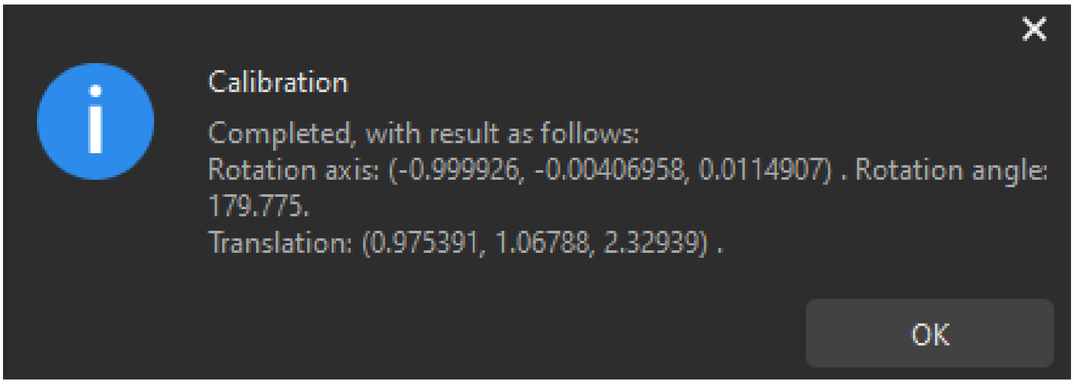
Calculate Extrinsic Parameters

1. In the **Calculate extrinsic parameters** step, click the **[Calculate extrinsic parameters]** button.



Files
Tools
Connect to camera
Mount calibration board & check intrinsic parameters
Set TCP in flange frame
Collect images and flange poses
5
Calculate extrinsic parameters
Calculate camera extrinsic parameters
?
Calculate extrinsic parameters
Supplementary tools

2. In the prompted window indicating calibration success, click the **[OK]** button.

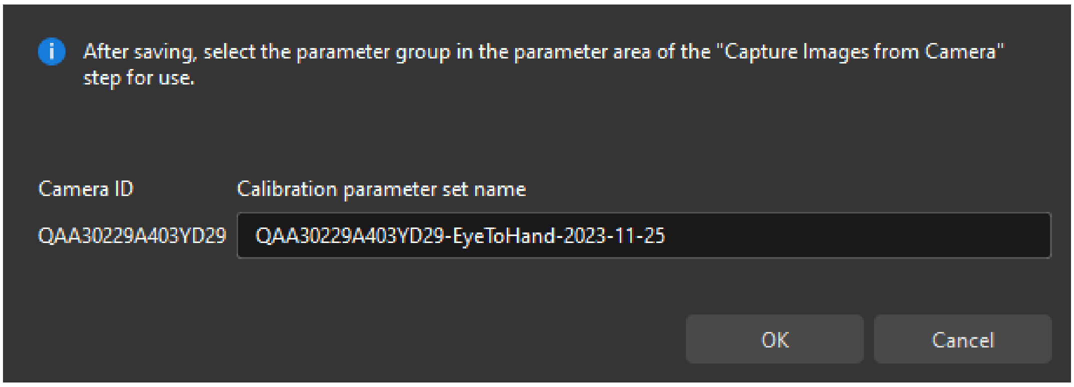


Calibration

Completed, with result as follows:
Rotation axis: (-0.999926, -0.00406958, 0.0114907) . Rotation angle: 179.775.
Translation: (0.975391, 1.06788, 2.32939) .

- If you want to enhance the calibration accuracy, please refer to the section Calibration Result Check and Analysis.
- To evaluate the robot absolute accuracy and the accuracy of calculated extrinsic parameters, you can click **[Robot absolute accuracy and repeatability]** and **[Extrinsic parameter accuracy]** in the "Calculate extrinsic parameters" window and use the error analysis tool to perform robot absolute accuracy check and analyze extrinsic parameter error in ETH setup.

3. Click the **[Save]** button on the bottom bar. In the prompted **Save Calibration Files** dialog box, click the **[OK]** button. The camera calibration result will be automatically saved in the "calibration" directory of the project.



After saving, select the parameter group in the parameter area of the "Capture Images from Camera" step for use.

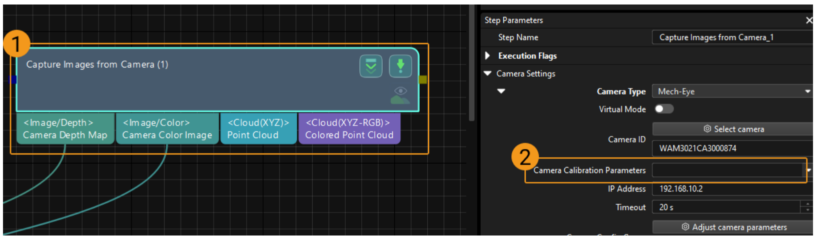
Camera ID	Calibration parameter set name
QAA30229A403YD29	QAA30229A403YD29-EyeToHand-2023-11-25

Till now, the calibration process is completed.

Change the Extrinsic Parameter File

If the camera's extrinsic parameter file has been used by a Mech-Vision project, you need to change the extrinsic parameter file used by the Mech-Vision project to a new one after you finish the hand-eye calibration.

1. Select the **Capture Images from Camera** Step.



1
Capture Images from Camera (1)
<Image/Depth> Camera Depth Map
<Image/Color> Camera Color Image
<Cloud(XYZ)> Point Cloud
<Cloud(XYZ-RGB)> Colored Point Cloud
2

Step Parameters
Step Name: Capture Images from Camera_1
Execution Flags
Camera Settings
Camera Type: Mech-Eye
Virtual Mode
Select camera
Camera ID: WAM3021CA3000874
Camera Calibration Parameters
IP Address: 192.168.10.2
Timeout: 20 s
Adjust camera parameters

2. In the **Step Parameters** panel, click ▾ of the **Camera Calibration Parameters** parameter, and select the newly calibrated extrinsic parameter file.

4.5.2.6. Manual Calibration in the Eye-to-Hand Setup (Four-Axis Robots - Using TCP Touch Method)

This how-to guide introduces how to complete the manual calibration for four-axis robots using the TCP touch method in the eye-to-hand (ETH) setup.

Preparation before Calibration

Before hand-eye calibration, you need to finish the following preparations:

- Construct the Mech-Mind Vision System.
- Prepare the materials required for calibration.
- Adjust the effect of calibration board's point cloud.

Construct the Vision System

Construct the Mech-Mind Vision System by referring to the section Vision System Hardware Setup.

You need to use Mech-Eye Viewer, Mech-Vision and Mech-Viz during hand-eye calibration. Please ensure that they have been installed and are running the latest versions.

Prepare the Materials Required for Calibration

The manual calibration in the ETH setup needs to use the calibration board and sharp tip.

Please prepare the calibration board and the sharp tip according to the following requirements:

- Ensure that the circles of the calibration board are clearly visible and without obvious scratches, and the board does not suffer from deformations.
- When the TCP touch method is used to collect calibration data in the ETH setup, please place the calibration board in the center of the working plane, where target objects are to be placed.
- The sharp tip should be mounted on the center of the robot flange. If an undetachable gripper is connected to the robot flange, you can attach the sharp point directly to the gripper.

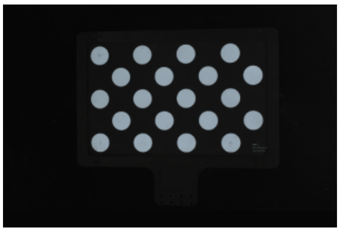
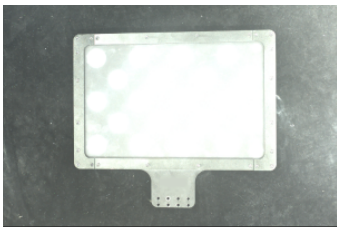
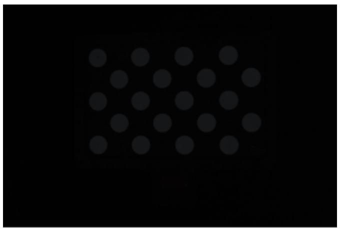
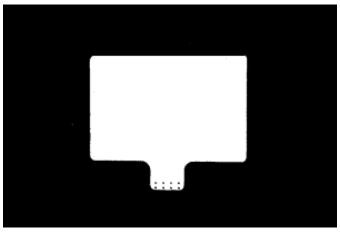
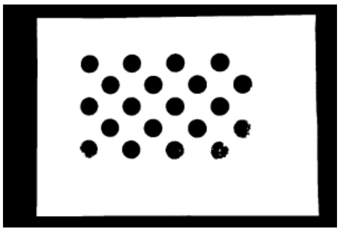
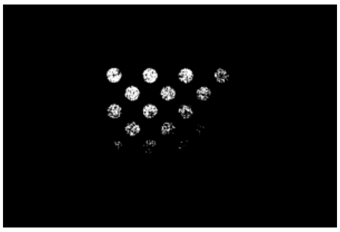
In addition, before calibration, move the robot to the starting point for calibration to ensure that the robot will not block the camera from collecting images of the calibration board.

Adjust the Effect of Calibration Board's Point Cloud

1. Open Mech-Eye Viewer to adjust camera parameters.
2. Adjust the 2D parameters to make sure that the 2D image of the calibration board is clear and neither overexposed nor underexposed.
3. Adjust 3D parameters to make sure that the point clouds of the circles on the calibration board are complete and have clear contours. It is recommended to set **Surface Smoothing** and **Outlier Removal** to **Normal** in the **Point Cloud Processing** section to reduce the point cloud fluctuation range.

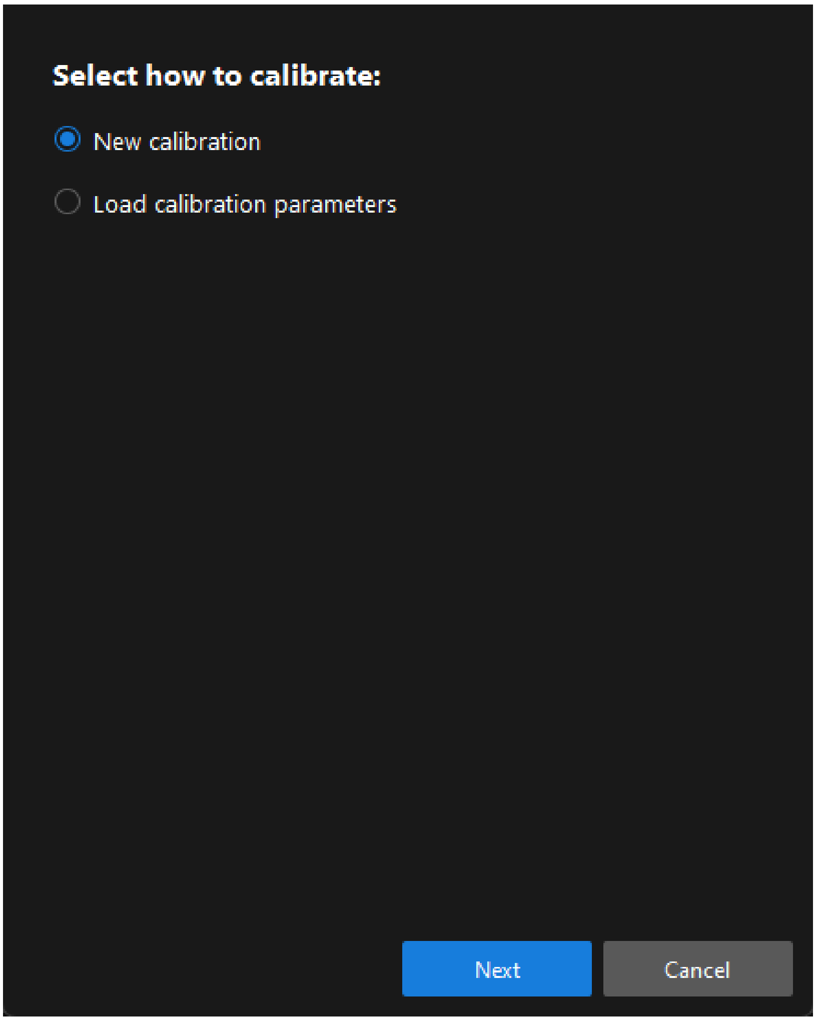
   If the on-site ambient lights are not ideal and affect the quality of 2D images and point clouds, you can use shading or supplemental light to improve the lighting conditions.

4. Make sure that the images and point cloud of the calibration board are up to standard after completing the preceding steps.

	Normal	Overexposed	Underexposed
2D image			
Point cloud			

Pre-calibration Configuration

1. Open Mech-Vision, and click the [**Camera Calibration (Standard)**] button in the toolbar. The **Configuration before Calibration** window will be prompted.
2. In the **Select how to calibrate** window, select the **New calibration** radio button, and then click the [**Next**] button.

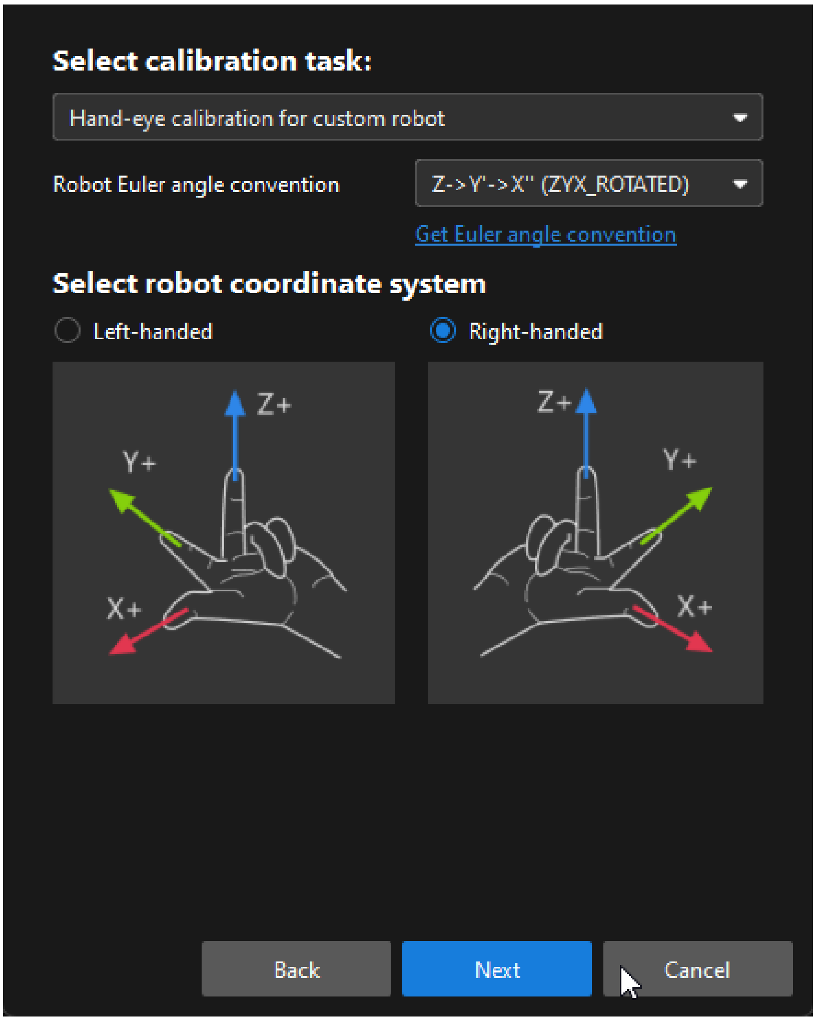


Select how to calibrate:

● New calibration

○ Load calibration parameters

3. In the **Select calibration task** window, select **Hand-eye calibration for custom robot** from the drop-down list box, specify the **Robot Euler angle convention** parameter, select the robot coordinate system type, and then click the **[Next]** button.



Select calibration task:

Hand-eye calibration for custom robot

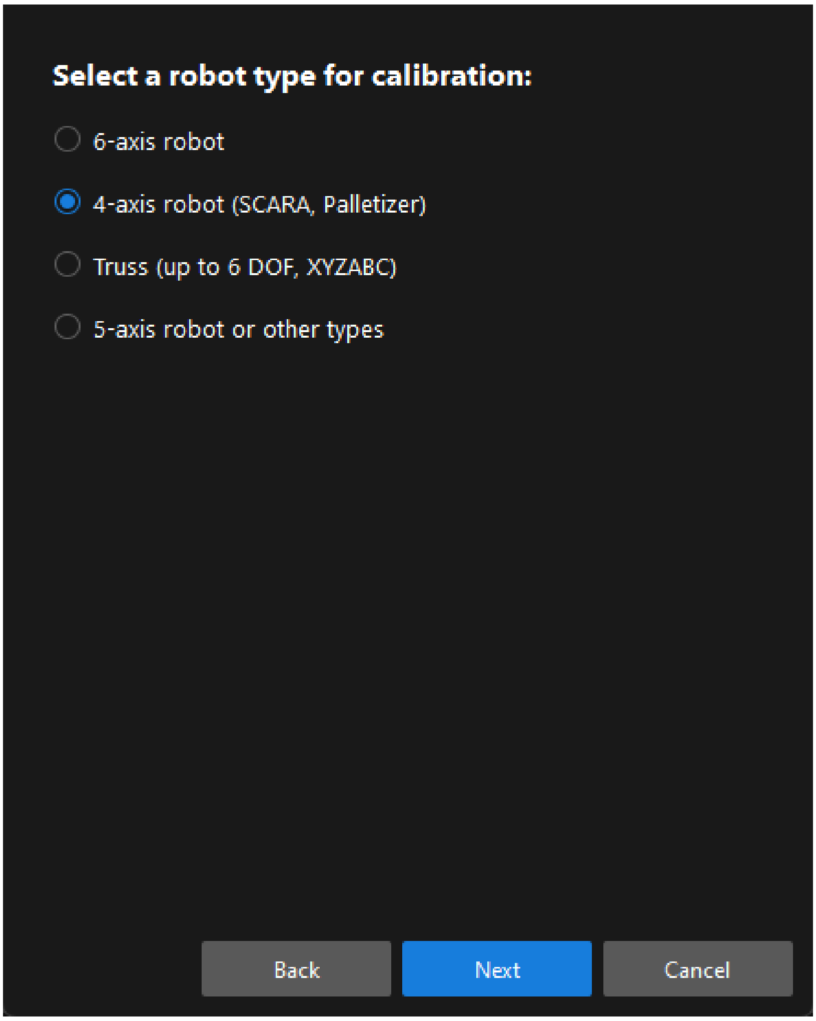
Robot Euler angle convention: Z->Y'->X'' (ZYX_ROTATED)

Get Euler angle convention

Select robot coordinate system

○ Left-handed ● Right-handed

4. In the **Select a robot type for calibration** window, select the **4-axis robot (SCARA, Palletizer)** radio button, and then click the **[Next]** button.



Select a robot type for calibration:

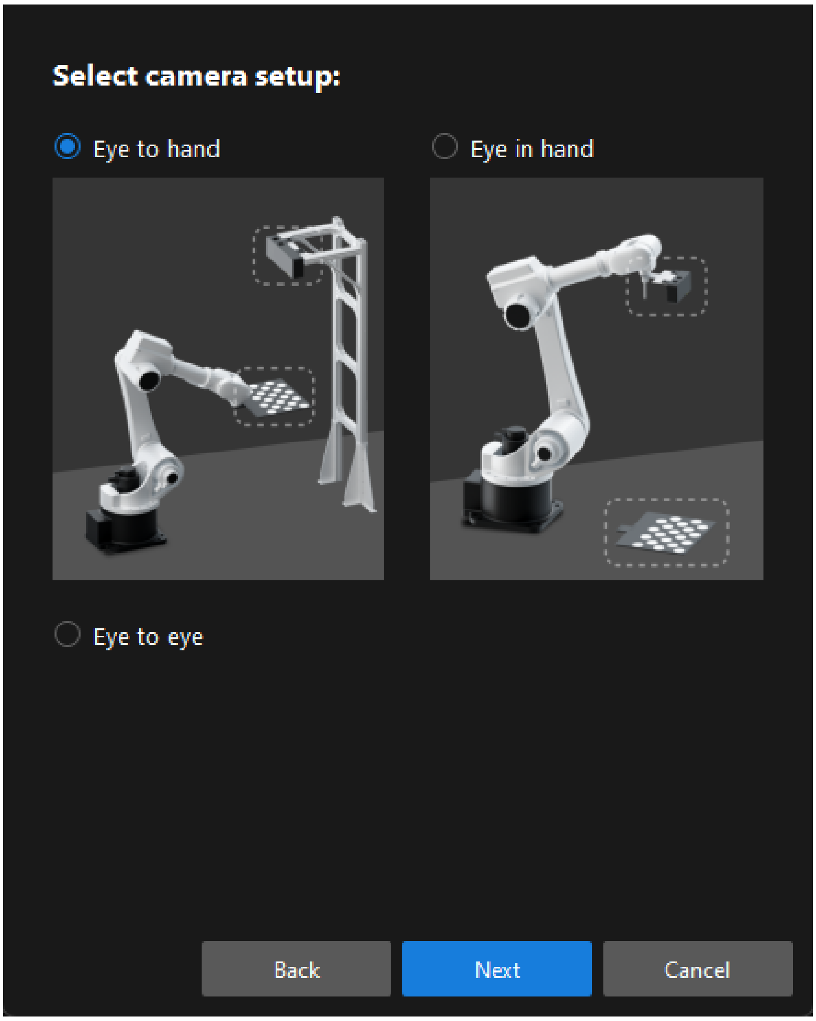
○ 6-axis robot

● 4-axis robot (SCARA, Palletizer)

○ Truss (up to 6 DOF, XYZABC)

○ 5-axis robot or other types

5. In the **Select camera setup** window, select the **Eye to hand** radio button, and then click the **[Next]** button.



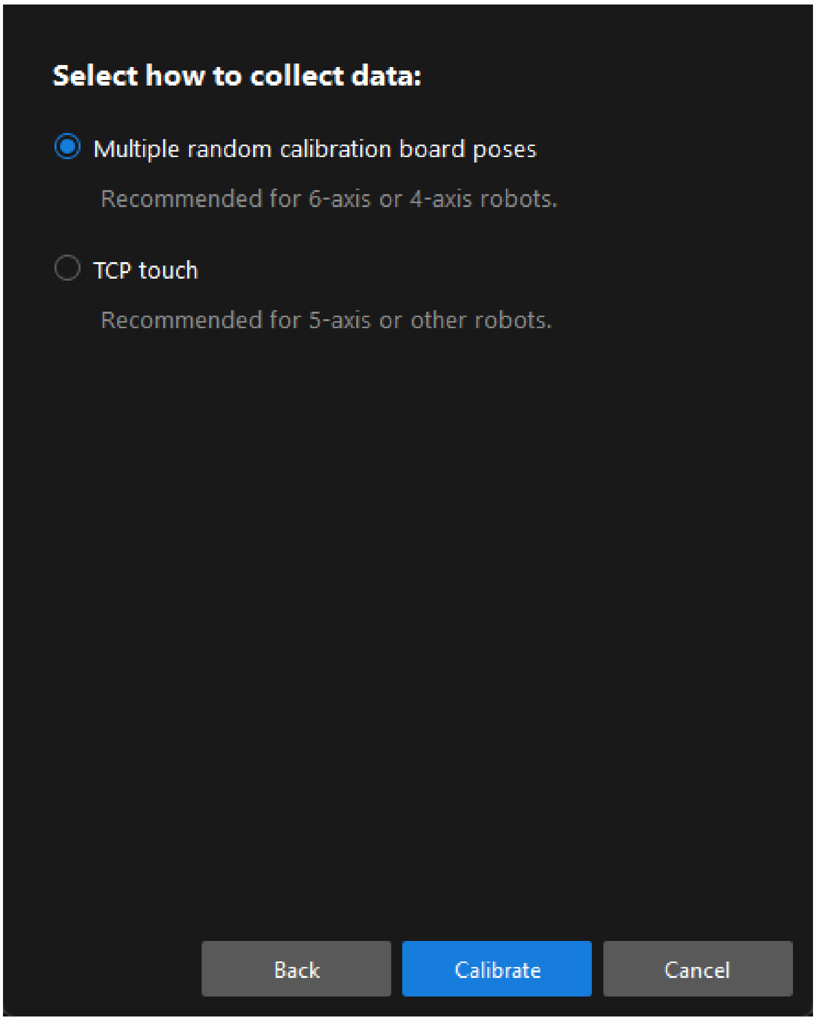
Select camera setup:

● Eye to hand

○ Eye in hand

○ Eye to eye

6. In the **Select how to collect data** window, select the **TCP touch** radio button, and then click the **[Calibrate]** button.The **Calibration (Eye to hand)** window will be prompted.



Select how to collect data:

● Multiple random calibration board poses

Recommended for 6-axis or 4-axis robots.

○ TCP touch

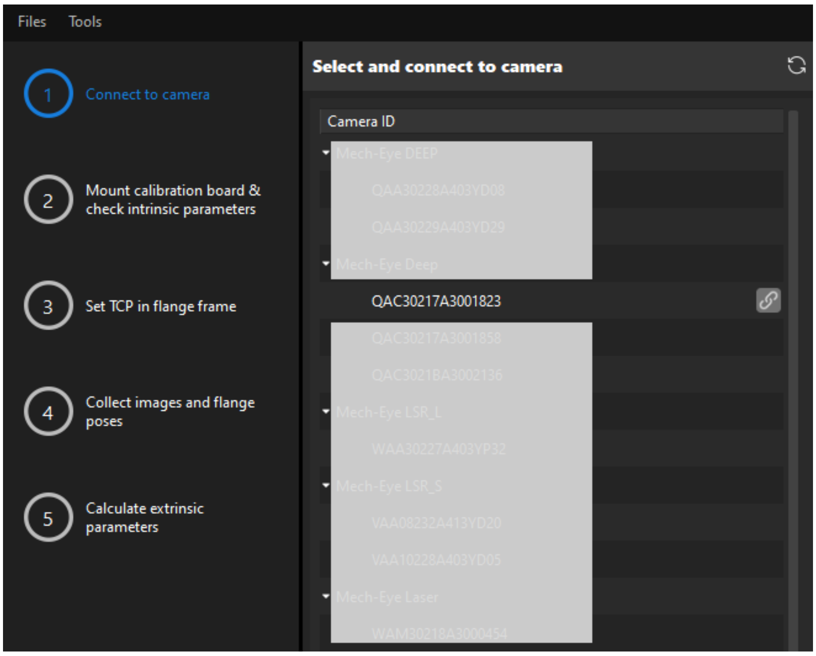
Recommended for 5-axis or other robots.

Till now, you have completed the pre-calibration configuration and can start the calibration procedure.

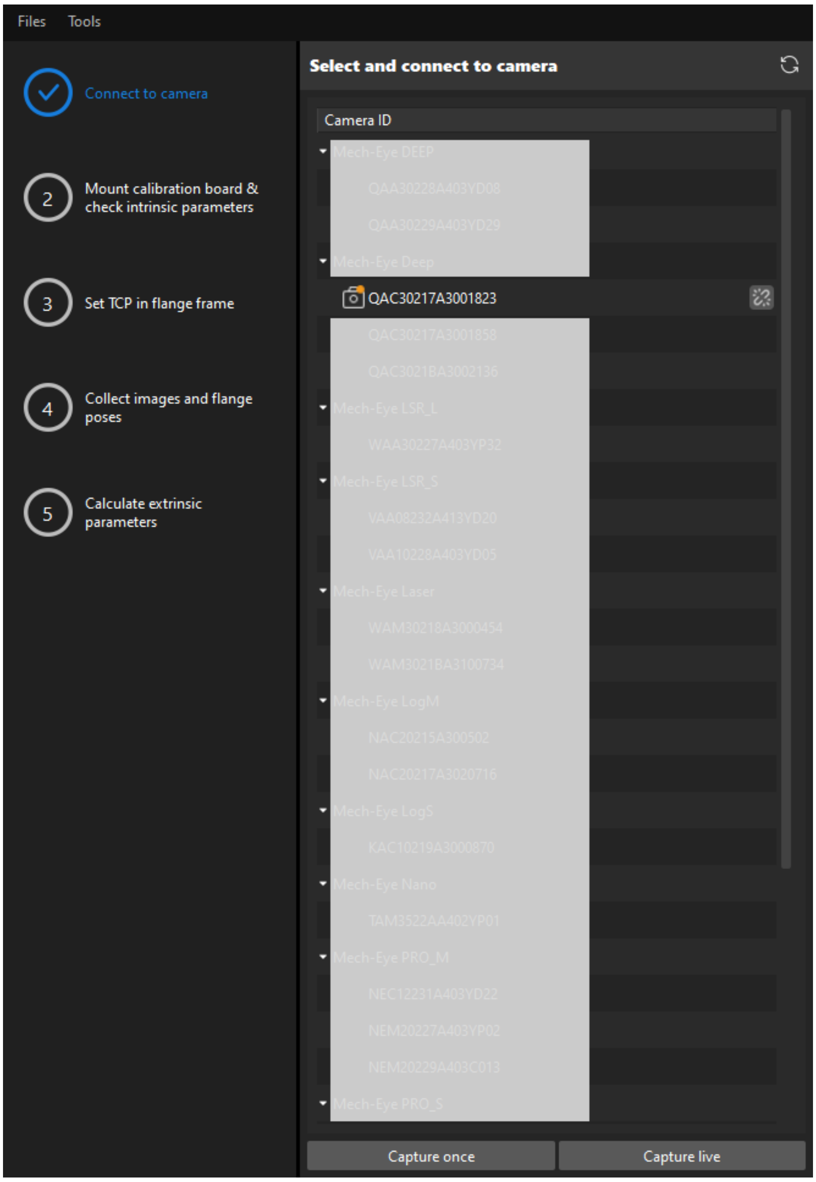
Calibration Procedure

Connect to the Camera

1. In the **Connect to camera** step, select the camera to connect in the Camera ID list, and then click the 🔗 button or double-click the camera entry to connect to it.


Files
Tools
1
Connect to camera
2
Mount calibration board & check intrinsic parameters
3
Set TCP in flange frame
4
Collect images and flange poses
5
Calculate extrinsic parameters
Select and connect to camera
Camera ID
QAC30217A3001823

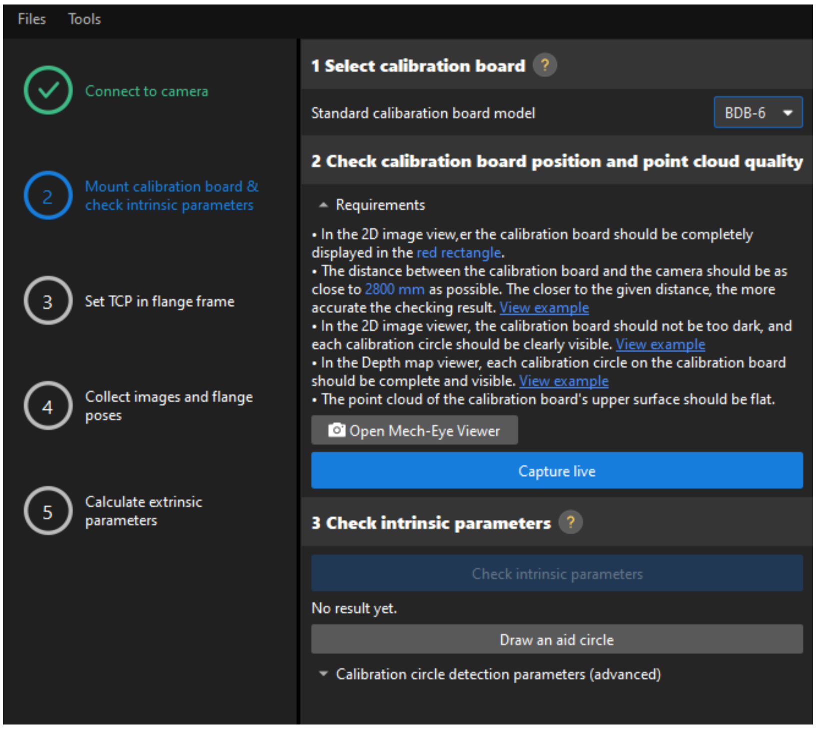
2. After the camera is connected, click the **[Capture once]** or **[Capture live]** button.


Files
Tools
Connect to camera
2 Mount calibration board & check intrinsic parameters
3 Set TCP in flange frame
4 Collect images and flange poses
5 Calculate extrinsic parameters
Select and connect to camera
Camera ID
[redacted]
QAC30217A3001823
[redacted]
Capture once
Capture live

3. In the **Image viewer** panel, ensure that the camera can capture images normally and click the **[Next]** button on the bottom bar.

Mount Calibration Board & Check Intrinsic Parameters

1. In the **Mount calibration board & check intrinsic parameters** step, select the calibration board model for the **Standard calibration board model** parameter according to its model nameplate in the **1 Select calibration board** area.


Files
Tools
Connect to camera
2
Mount calibration board & check intrinsic parameters
3
Set TCP in flange frame
4
Collect images and flange poses
5
Calculate extrinsic parameters
Open Mech-Eye Viewer
Capture live
Check intrinsic parameters
Draw an aid circle

1 Select calibration board ?

Standard calibaration board model: BDB-6

2 Check calibration board position and point cloud quality

Requirements

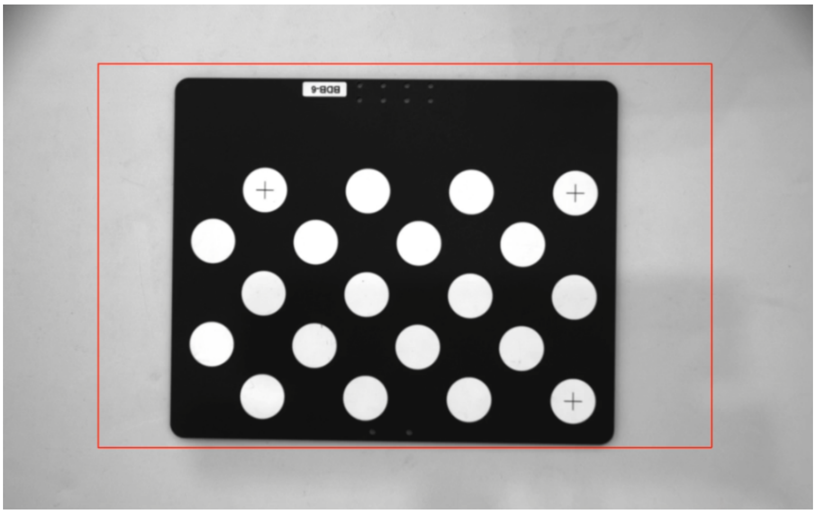
- In the 2D image view,er the calibration board should be completely displayed in the red rectangle.
- The distance between the calibration board and the camera should be as close to 2800 mm as possible. The closer to the given distance, the more accurate the checking result. View example
- In the 2D image viewer, the calibration board should not be too dark, and each calibration circle should be clearly visible. View example
- In the Depth map viewer, each calibration circle on the calibration board should be complete and visible. View example
- The point cloud of the calibration board's upper surface should be flat.

3 Check intrinsic parameters ?

No result yet.

Calibration circle detection parameters (advanced)

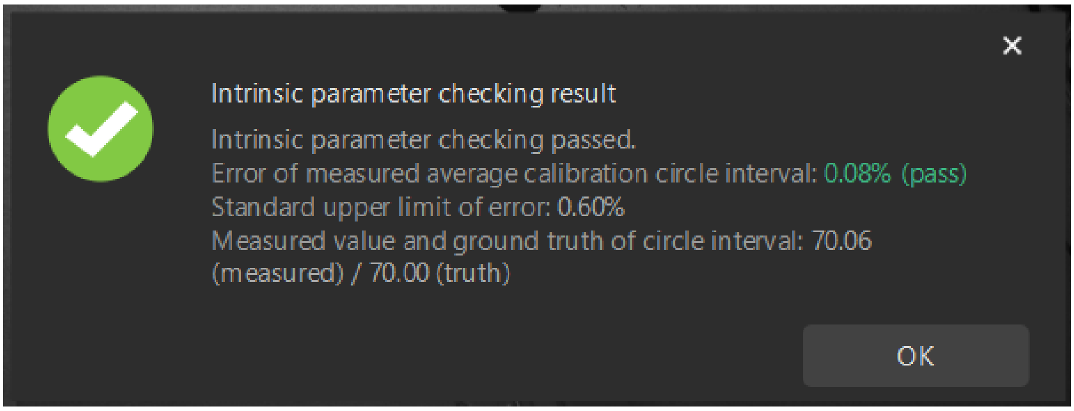
2. In the **2 Check calibration board position and point cloud quality** area, read carefully the requirements on the calibration board position and point cloud quality, and then click the [**Capture live**] button.The [**Capture live**] button will turn into [**Stop capturing and detect position**].
3. Manually move the calibration board to the proper position (in the red rectangle), and ensure that the 2D image and depth map of the calibration board meet the requirements, and then click the [**Stop capturing and detect position**] button.



If the captured images do not meet the requirements, click the [**Open Mech-Eye Viewer**] button to open the Mech-Eye Viewer software, adjust the 2D and 3D exposure parameters and re-capture images. Please note that you need to change the **Parameter group** parameter to "calib" first.

4. In the **3 Check intrinsic parameters** area, click the [**Check intrinsic parameters**] button.
5. Confirm the results of the camera intrinsic parameter check.
   - If the camera intrinsic parameter check passes, click the [**OK**] button in the prompted

window, and then click the [**Next**] button on the bottom bar.



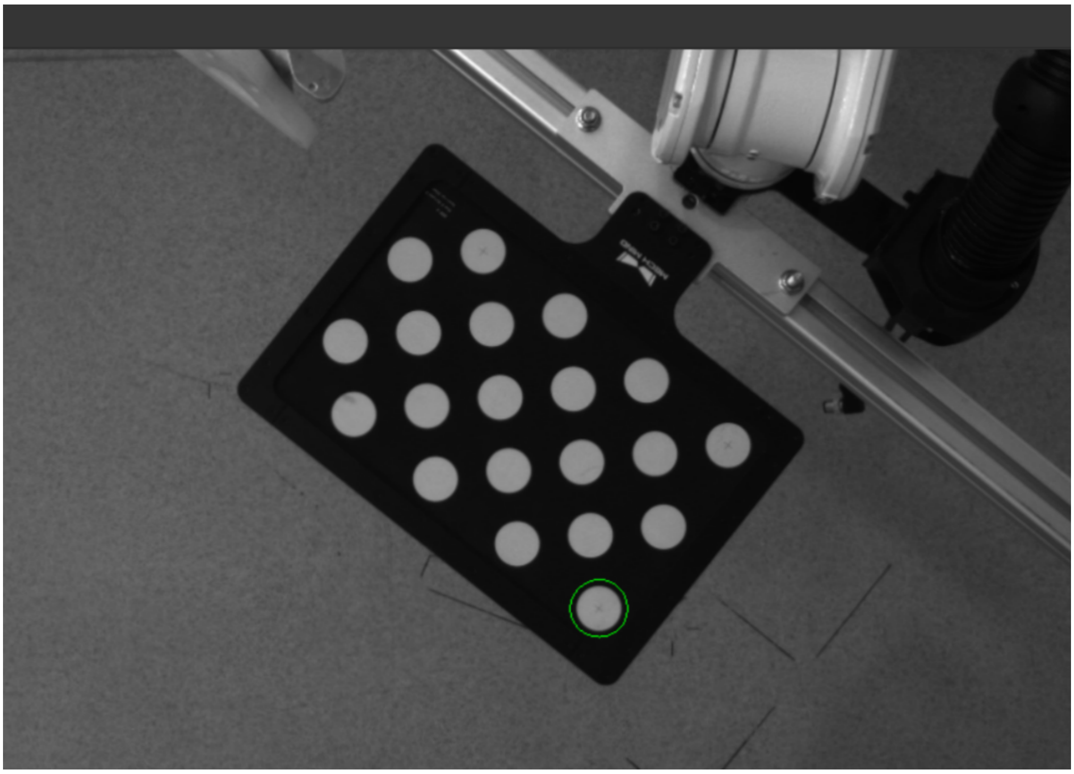
Intrinsic parameter checking result

Intrinsic parameter checking passed.
Error of measured average calibration circle interval: 0.08% (pass)
Standard upper limit of error: 0.60%
Measured value and ground truth of circle interval: 70.06 (measured) / 70.00 (truth)

- If the camera intrinsic parameter check fails, adjust detection parameters by drawing aid circles or manually adjusting the calibration circle detection parameters, and then click the [**Recheck intrinsic parameters**] button.

Draw an Aid Circle

1. To draw an aid circle, click the [**Draw an aid circle**] button.
2. In the right **Image viewer** panel, right-click the calibration board image and clear the **Fit to window** checkbox, and press the `Ctrl` key and drag the roller to adjust the image to a suitable size.
3. Move the mouse pointer to the cross center point of the calibration circle, press the left mouse button and make the aid circle completely include the calibration circle and then release it.



4. Click the [**Recheck intrinsic parameters**] button, and confirm that the camera intrinsic parameter check passes.

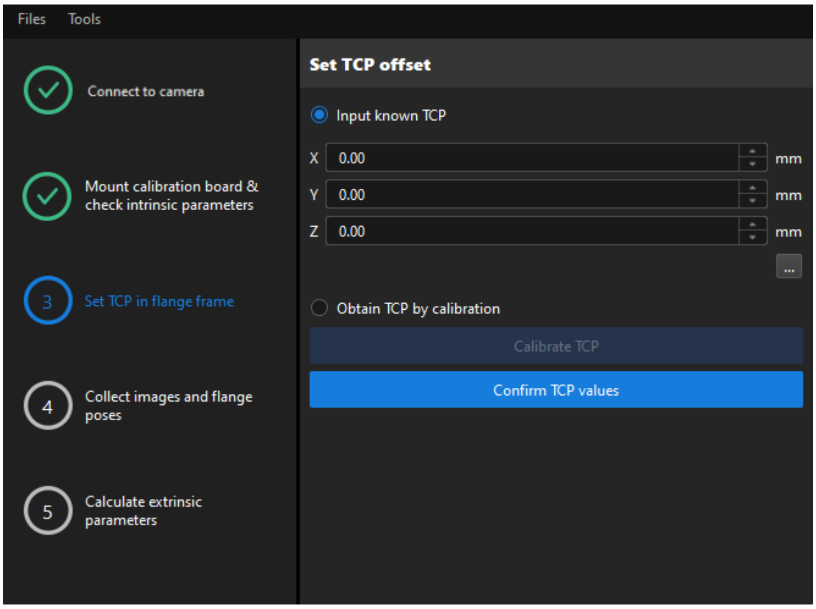
Manually Adjust Detection Parameters

To manually adjust detection parameters, click **[Calibration circle detection parameters (advanced)]**, adjust parameter settings and click the **[Recheck intrinsic parameters]** button to confirm whether the camera intrinsic parameter check passes.

If the calibration circles are still not detected, please adjust the camera settings according to the on-site operating conditions.

Set TCP in Flange Frame

1. In the **Set TCP in flange frame** step, select the **Input known TCP** radio button, and enter the TCP values.

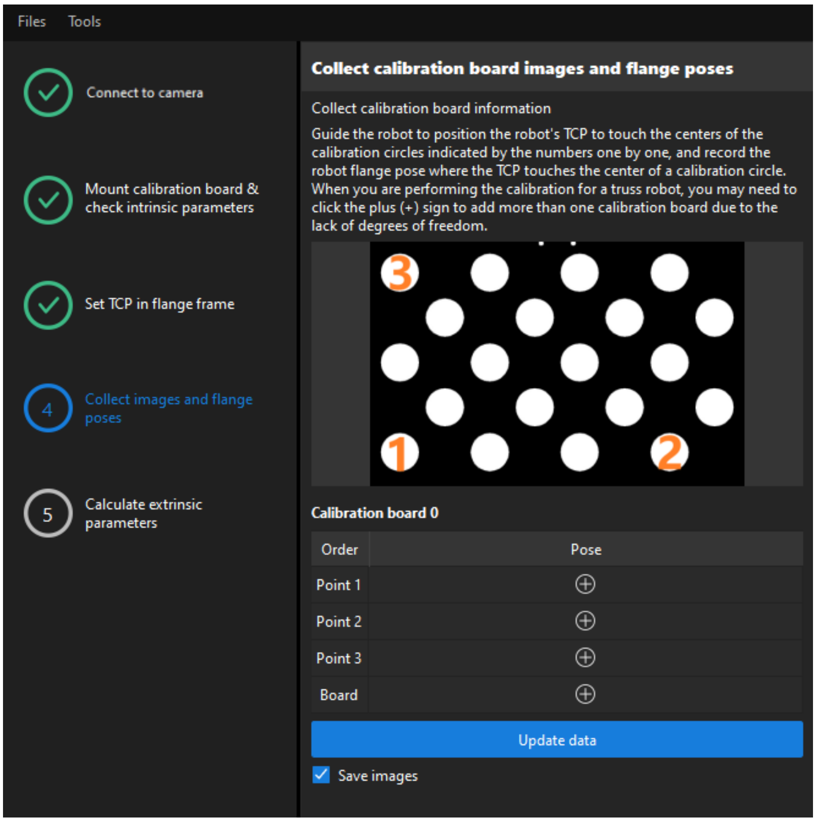


2. Click the **[Confirm TCP values]** button.
3. Click the **[Next]** button on the bottom bar.

> If the TCP values are unknown, select the **Obtain TCP by calibration** radio button, and then click the **[Calibrate TCP]** button. You can use the **TCP calibration** tool to calculate the TCP values.

Collect Images and Flange Poses

1. In the **Collect images and flange poses** step, select the **Save images** checkbox.



Files Tools

- Connect to camera
- Mount calibration board & check intrinsic parameters
- Set TCP in flange frame
- 4 Collect images and flange poses
- 5 Calculate extrinsic parameters

Collect calibration board images and flange poses

Collect calibration board information

Guide the robot to position the robot's TCP to touch the centers of the calibration circles indicated by the numbers one by one, and record the robot flange pose where the TCP touches the center of a calibration circle. When you are performing the calibration for a truss robot, you may need to click the plus (+) sign to add more than one calibration board due to the lack of degrees of freedom.

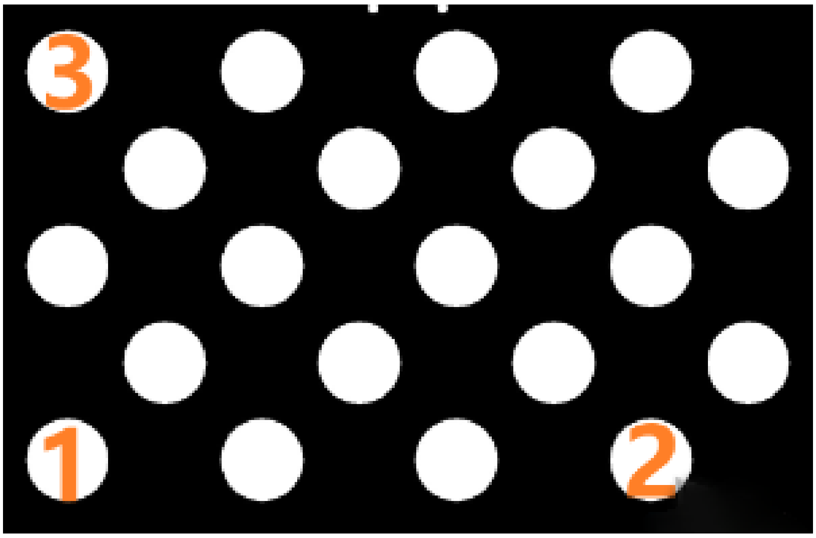
Calibration board 0

Order	Pose
Point 1	⊕
Point 2	⊕
Point 3	⊕
Board	⊕

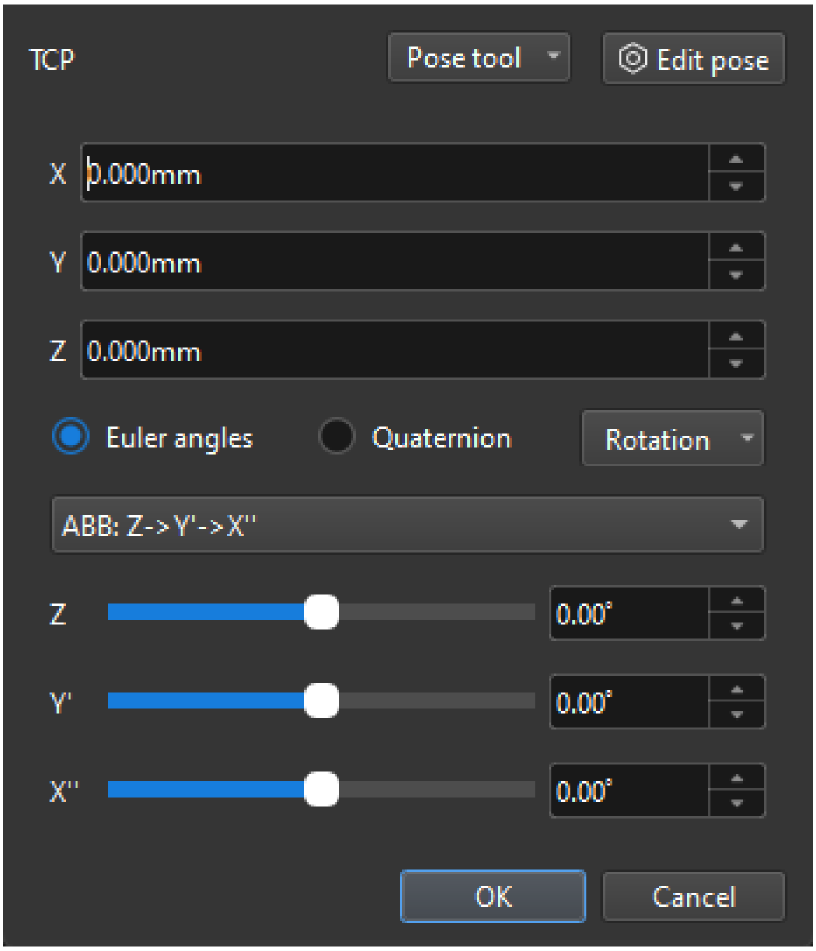
Update data

☑ Save images

2. Control the robot to let its end tip touch the cross center point of calibration circle “point 1” on the calibration board, and record the robot flange pose in the teach pendant.



3. Click the ⊕ button of point 1, and enter the robot flange pose in the prompted **Input Flange Pose of Robot** dialog box, and then click the [OK] button.



TCP

X 0.000mm

Y 0.000mm

Z 0.000mm

● Euler angles ○ Quaternion

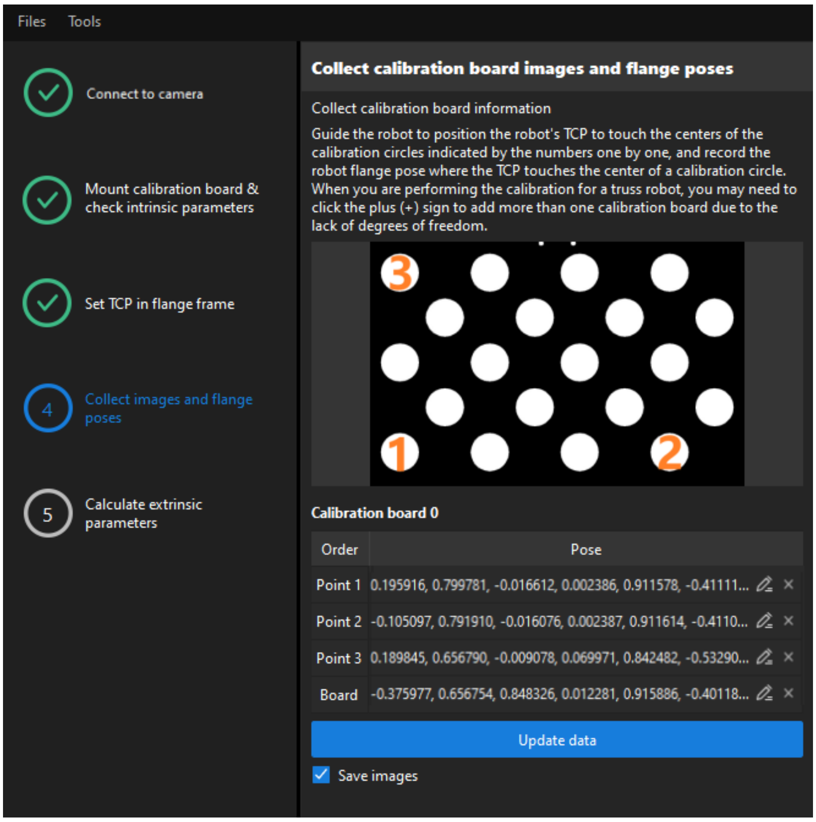
ABB: Z->Y'->X''

Z 0.00°

Y' 0.00°

X'' 0.00°

4. Repeat the preceding steps, and let the end tip touch the cross center points of point 2 and point 3, and enter robot flange poses.
5. Click the ⊕ button of the current calibration board to capture the images of it.
6. Click the [**Update data**] button and then click the [**Next**] button on the bottom bar.


Files
Tools
Connect to camera
Mount calibration board & check intrinsic parameters
Set TCP in flange frame
4
Collect images and flange poses
5
Calculate extrinsic parameters
3
1
2
Update data
☑ Save images

Collect calibration board images and flange poses

Collect calibration board information

Guide the robot to position the robot's TCP to touch the centers of the calibration circles indicated by the numbers one by one, and record the robot flange pose where the TCP touches the center of a calibration circle. When you are performing the calibration for a truss robot, you may need to click the plus (+) sign to add more than one calibration board due to the lack of degrees of freedom.

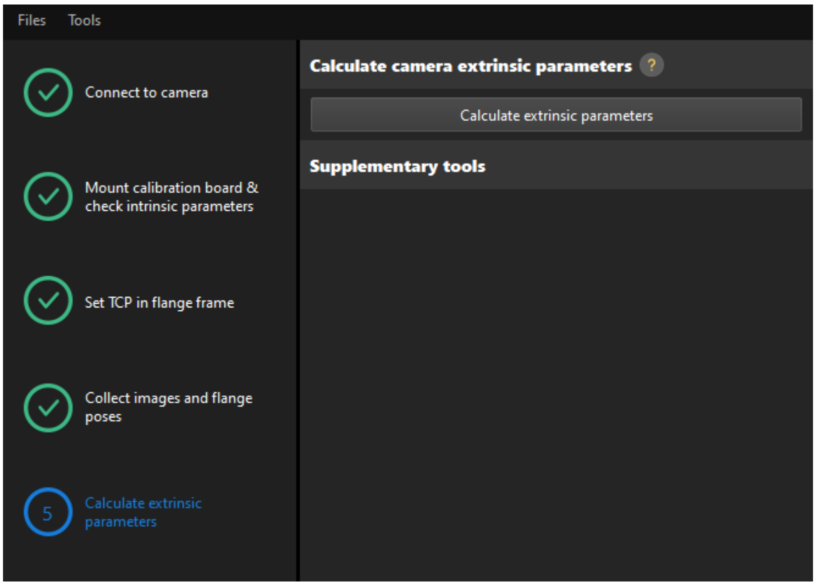
Calibration board 0

Order	Pose
Point 1	0.195916, 0.799781, -0.016612, 0.002386, 0.911578, -0.41111...
Point 2	-0.105097, 0.791910, -0.016076, 0.002387, 0.911614, -0.4110...
Point 3	0.189845, 0.656790, -0.009078, 0.069971, 0.842482, -0.53290...
Board	-0.375977, 0.656754, 0.848326, 0.012281, 0.915886, -0.40118...

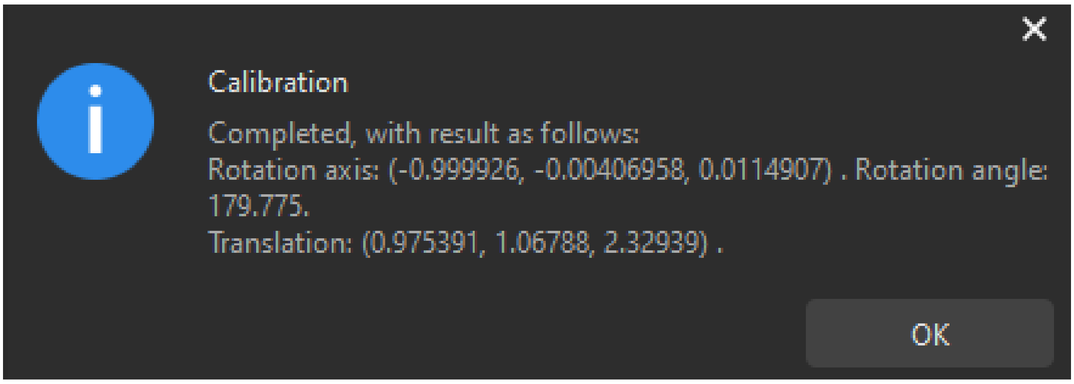
The calibration program requires that at least three points not in a line should be touched for calculating extrinsic parameters.

Calculate Extrinsic Parameters

1. In the **Calculate extrinsic parameters** step, click the **[Calculate extrinsic parameters]** button.


Files
Tools
Connect to camera
Mount calibration board & check intrinsic parameters
Set TCP in flange frame
Collect images and flange poses
5
Calculate extrinsic parameters
Calculate camera extrinsic parameters
?
Calculate extrinsic parameters
Supplementary tools

2. In the prompted window indicating calibration success, click the **[OK]** button.

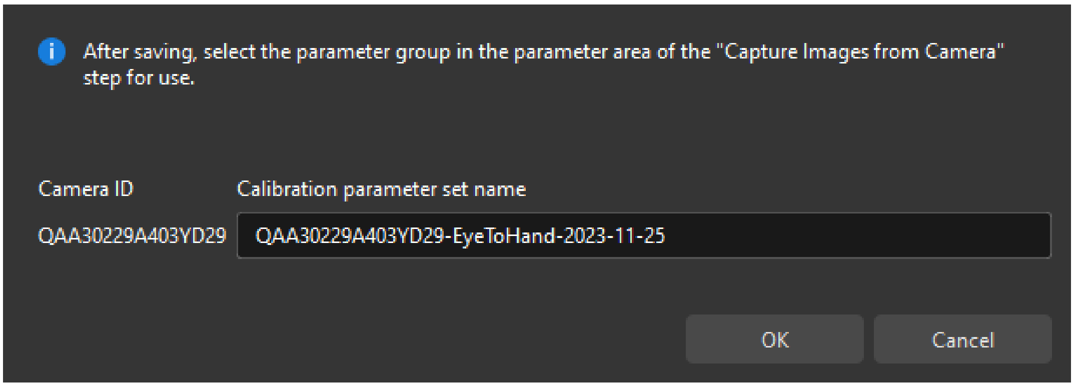


Calibration

Completed, with result as follows:
Rotation axis: (-0.999926, -0.00406958, 0.0114907) . Rotation angle: 179.775.
Translation: (0.975391, 1.06788, 2.32939) .

- If you want to enhance the calibration accuracy, please refer to the section Calibration Result Check and Analysis.
- To evaluate the robot absolute accuracy and the accuracy of calculated extrinsic parameters, you can click **[Robot absolute accuracy and repeatability]** and **[Extrinsic parameter accuracy]** in the “Calculate extrinsic parameters” window and use the error analysis tool to perform robot absolute accuracy check and analyze extrinsic parameter error in ETH setup.

3. Click the **[Save]** button on the bottom bar. In the prompted **Save Calibration Files** dialog box, click the **[OK]** button. The camera calibration result will be automatically saved in the “calibration” directory of the project.



After saving, select the parameter group in the parameter area of the "Capture Images from Camera" step for use.

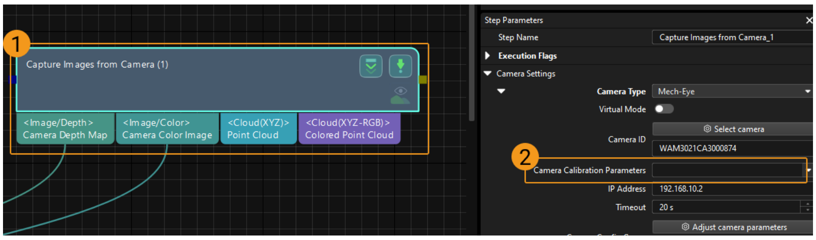
Camera ID	Calibration parameter set name
QAA30229A403YD29	QAA30229A403YD29-EyeToHand-2023-11-25

Till now, the calibration process is completed.

Change the Extrinsic Parameter File

If the camera’s extrinsic parameter file has been used by a Mech-Vision project, you need to change the extrinsic parameter file used by the Mech-Vision project to a new one after you finish the hand-eye calibration.

1. Select the **Capture Images from Camera** Step.



Step Parameters

Step Name: Capture Images from Camera_1

Execution Flags

Camera Settings

Camera Type: Mech-Eye

Virtual Mode

Select camera

Camera ID: WAM3021CA3000874

Camera Calibration Parameters

IP Address: 192.168.10.2

Timeout: 20 s

Adjust camera parameters

2. In the **Step Parameters** panel, click ▾ of the **Camera Calibration Parameters** parameter, and select the newly calibrated extrinsic parameter file.

Calibration-Related Status Codes

If the robot uses the Standard Interface to communicate with the Vision System during hand-eye calibration, the Vision System will return status codes in response data to indicate the execution status of the commands. The status codes indicate both the normal execution results and execution failures. You can perform further troubleshooting according to the status codes.

Calibration Error Codes

Error code	Description
7001	Calibration: Parameter error
7002	Calibration: Calibration flange pose not provided by Mech-Vision
7003	Calibration: Calibration joint positions not provided by Mech-Vision
7004	Calibration: Robot failed to reach calibration point

Calibration Normal Status Codes

Normal Status Code	Description
7100	Calibration: Robot reaches calibration point successfully
7101	Calibration: Pose received from Mech-Vision successfully

4.5.3. Hand-Eye Calibration in the Eye-in-Hand Setup

You can select a proper calibration process according to the calibration mode (automatic or manual), robot type (six-axis robots or four-axis robots) and calibration data collection method (multiple random calibration board poses or TCP touch), and follow the operation guide for the selected calibration process.

- Automatic Calibration in the Eye-in-Hand Setup (Six-Axis Robots)
- Automatic Calibration in the Eye-in-Hand Setup (Four-Axis Robots)
- Manual Calibration in the Eye-in-Hand Setup (Six-Axis Robots - Using Multiple Random Calibration Board Poses Method)
- Manual Calibration in the Eye-in-Hand Setup (Four-Axis Robots - Using Multiple Random Calibration Board Poses Method)
- Manual Calibration in the Eye-in-Hand Setup (Six-Axis Robots - Using TCP Touch Method)
- Manual Calibration in the Eye-in-Hand Setup (Four-Axis Robots - Using TCP Touch Method)

4.5.3.1. Automatic Calibration in the Eye-in-Hand Setup (Six-Axis Robots)

This how-to guide introduces how to complete the automatic calibration for six-axis robots in the eye-in-hand (EIH) setup.

If you use the Standard Interface communication, you can refer to this guide and the following video to calibrate.

▶ https://www.youtube.com/watch?v=KYa_6rDsD84 *(YouTube video)*

Video: Hand-Eye Calibration in the Eye-in-Hand Setup (Standard Interface Communication)

Preparation before Calibration

Before hand-eye calibration, you need to finish the following preparations:

- Construct the Mech-Mind Vision System.
- Prepare the materials required for calibration.
- Adjust the effect of calibration board's point cloud.
- Complete the robot communication configuration.

Construct the Vision System

Construct the Mech-Mind Vision System by referring to the section Vision System Hardware Setup.

You need to use Mech-Eye Viewer, Mech-Vision and Mech-Viz during hand-eye calibration. Please ensure that they have been installed and are running the latest versions.

Prepare the Materials Required for Calibration

The automatic calibration in the EIH setup needs to use the calibration board.

Please prepare the calibration board according to the following requirements:

- Ensure that the circles of the calibration board are clearly visible and without obvious scratches, and the board does not suffer from deformations.
- In the EIH setup, please place the calibration board in the center of the working plane, where target objects are to be placed.

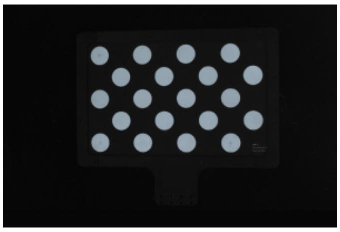
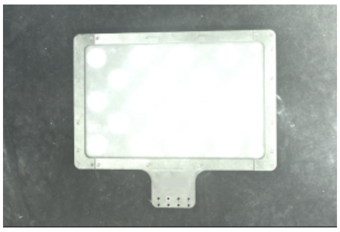
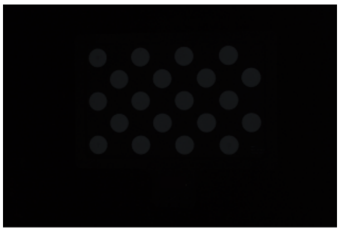
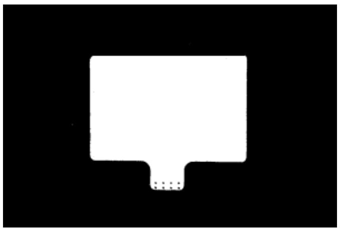
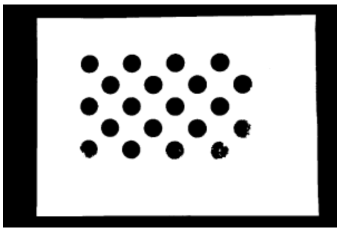
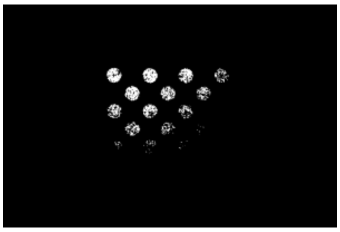
In addition, before calibration, move the robot to the starting point for calibration. For the automatic calibration in the EIH setup, the starting point is the camera's image-capturing position (the robot will move the camera upwards during calibration).

Adjust the Effect of Calibration Board's Point Cloud

1. Open Mech-Eye Viewer to adjust camera parameters.
2. Adjust the 2D parameters to make sure that the 2D image of the calibration board is clear and neither overexposed nor underexposed.
3. Adjust 3D parameters to make sure that the point clouds of the circles on the calibration board are complete and have clear contours. It is recommended to set **Surface Smoothing** and **Outlier Removal** to **Normal** in the **Point Cloud Processing** section to reduce the point cloud fluctuation range.

   > If the on-site ambient lights are not ideal and affect the quality of 2D images and point clouds, you can use shading or supplemental light to improve the lighting conditions.

4. Make sure that the images and point cloud of the calibration board are up to standard after completing preceding steps.

	Normal	Overexposed	Underexposed
2D image			
Point cloud			

Complete the Robot Communication Configuration

If the robot uses the Standard Interface to communicate with the Vision System, please set up the Standard Interface communication with the robot. For details, refer to the section Standard Interface Communication.

If the robot uses the Master-Control to communicate with the Vision System, please set up the Master-Control communication with the robot. For details, refer to the section Master-control Communication.

Pre-calibration Configuration

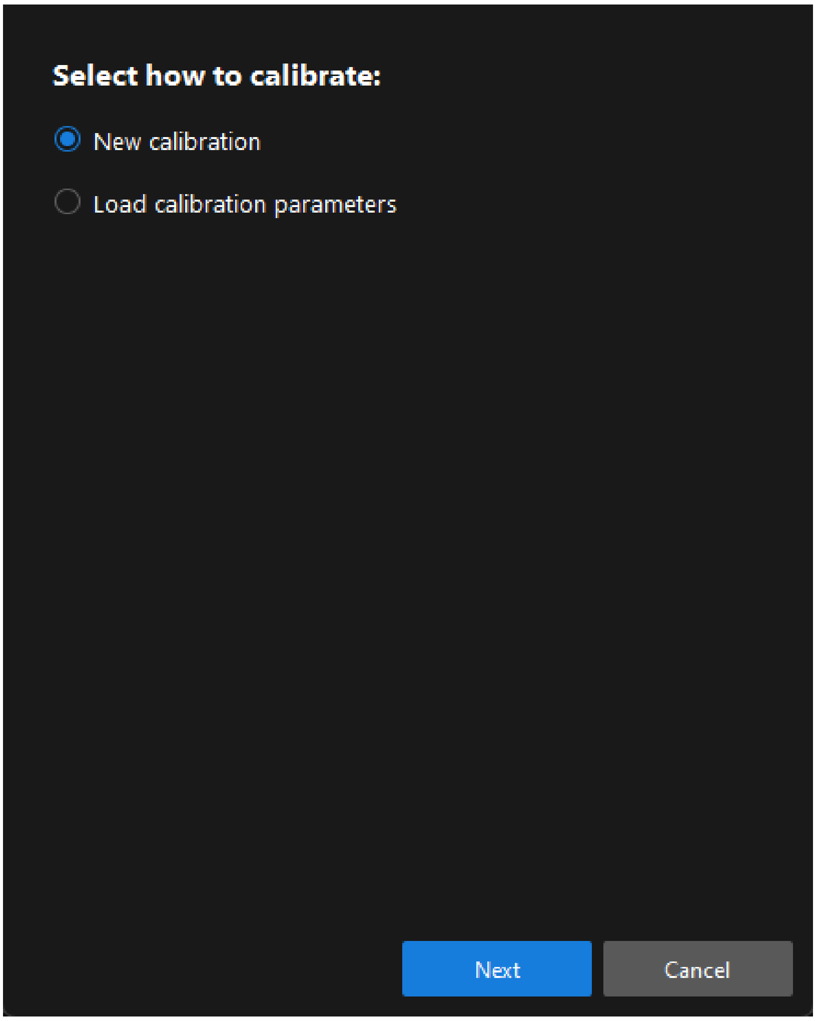
If the robot uses the Standard Interface to communicate with the Vision System, complete Pre-calibration Configuration (Standard Interface).

If the robot uses the Master-Control to communicate with the Vision System, complete Pre-calibration Configuration (Master-control).

Pre-calibration Configuration (Standard Interface)

If the Standard Interface is used to communicate with the robot, follow these steps:

1. Open Mech-Vision, and click the **[Camera Calibration (Standard)]** button in the toolbar.**The Configuration before Calibration window** will be prompted.
2. In the **Select how to calibrate** window, select the **New calibration** radio button, and then click the **[Next]** button.

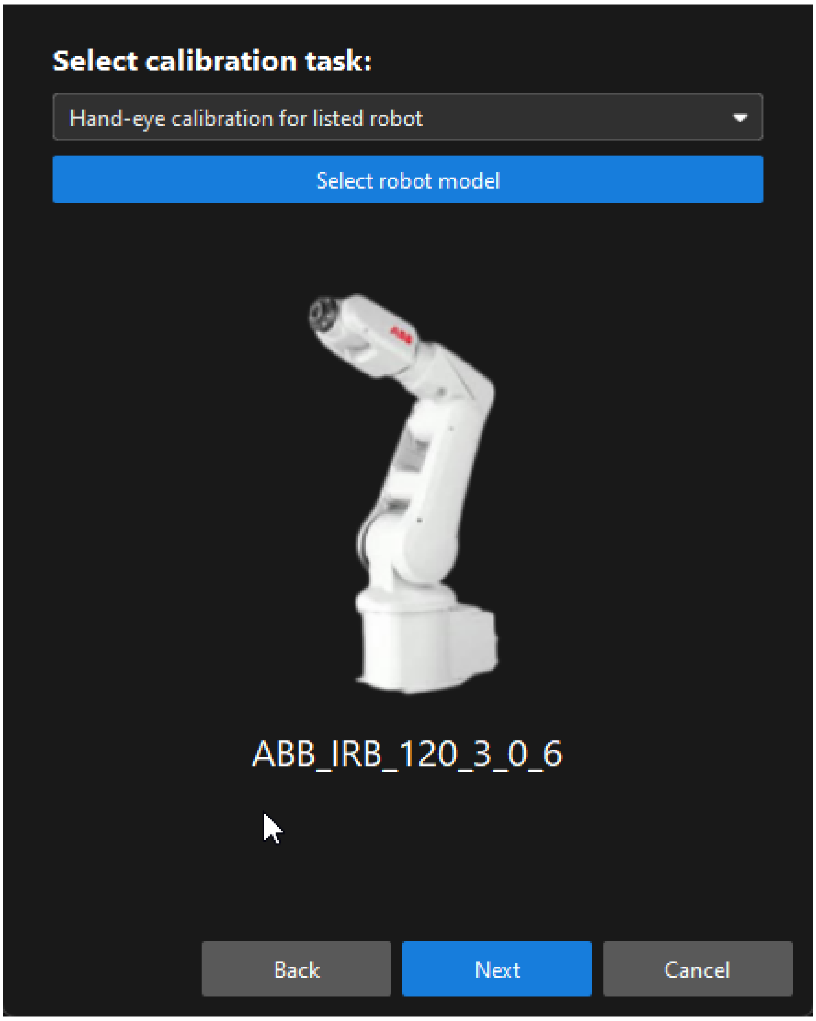


Select how to calibrate:

● New calibration

○ Load calibration parameters

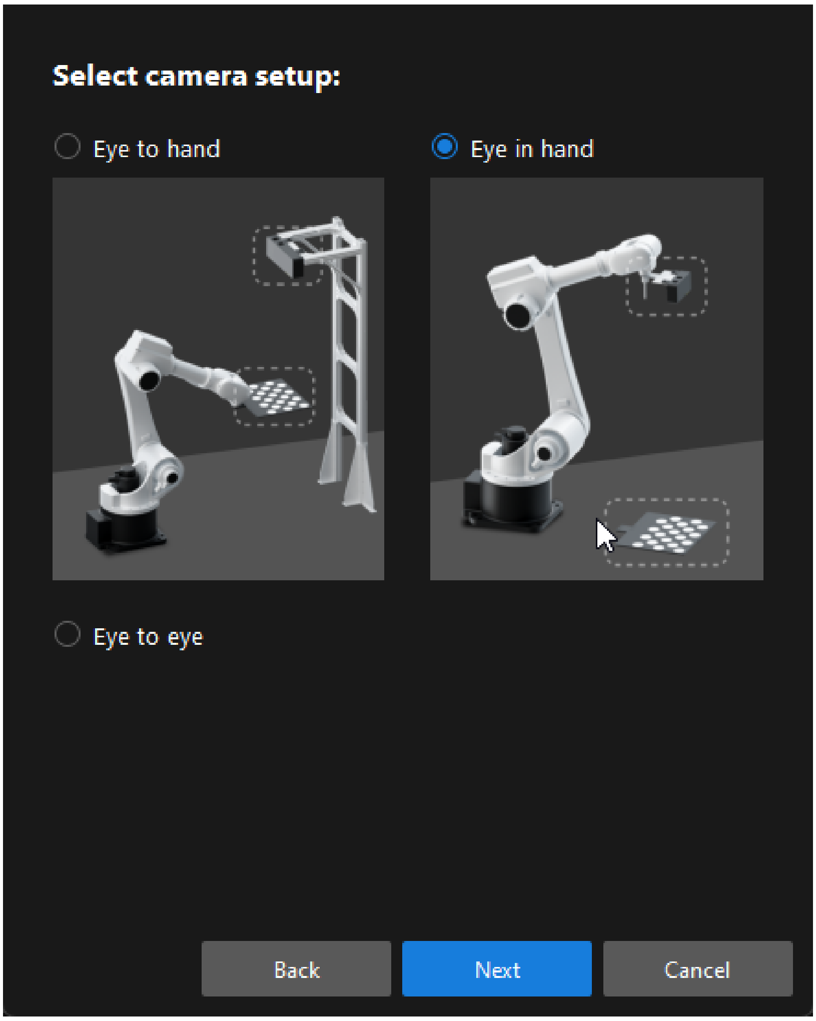
3. In the **Select calibration task** window, select **Hand-eye calibration for listed robot** from the drop-down list box, click the **[Select robot model]** button to select the robot model used by the project, and then click the **[Next]** button.



Select calibration task:

Hand-eye calibration for listed robot

4. In the **Select camera setup** window, select the **Eye in hand** radio button, and then click the **[Next]** button.



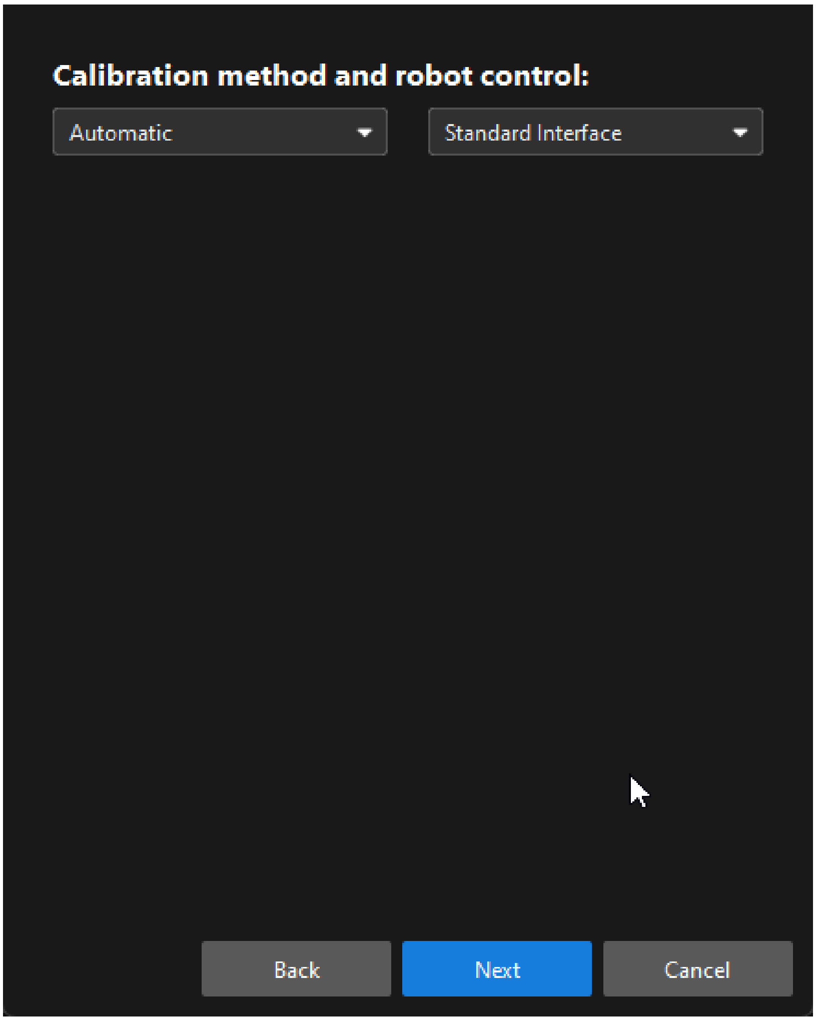
Select camera setup:

○ Eye to hand

● Eye in hand

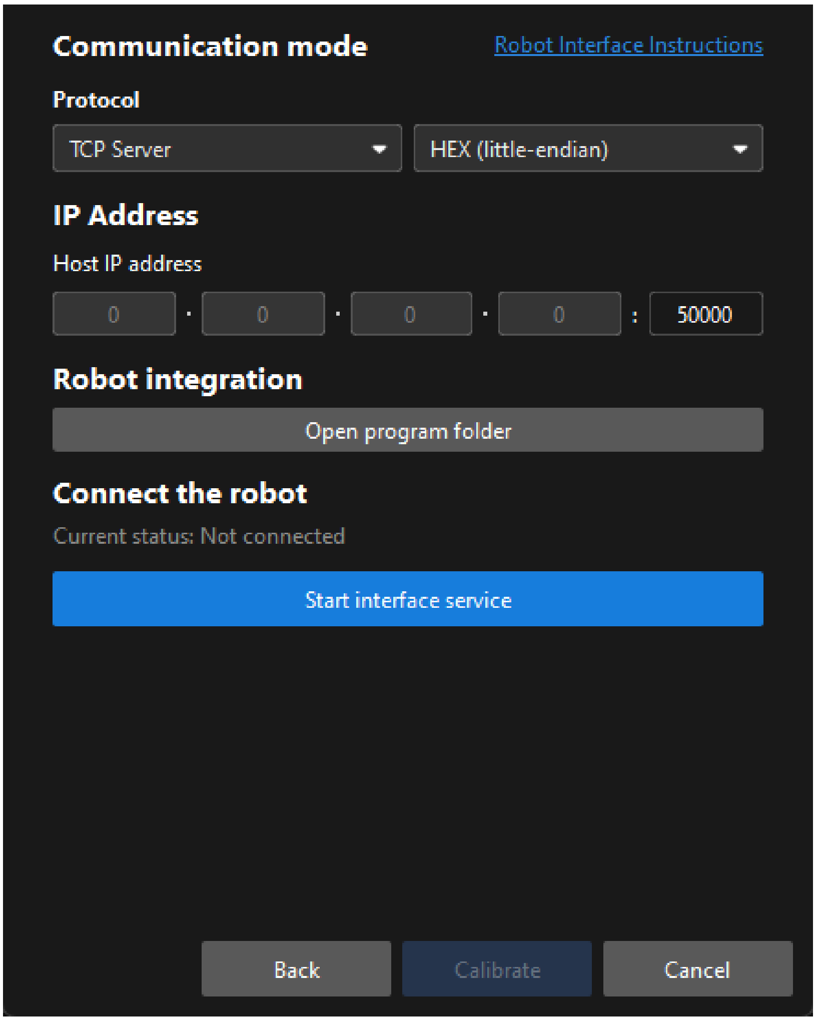
○ Eye to eye

5. In the **Calibration method and robot control** window, select **Automatic** and **Standard Interface**, and then click the [**Next**] button.



Calibration method and robot control:

6. In the new window, set parameters **Protocol** and **Host IP address**, and then click the **[Start interface service]** button. The button will turn into **Waiting for the robot to connect...**



Communication mode

Protocol: TCP Server | HEX (little-endian)

IP Address

Host IP address: 0 . 0 . 0 . 0 : 50000

Robot integration

Connect the robot

Current status: Not connected

- In the preceding figure, **TCP Server** and **Hex (Little-endian)** are only the **Protocol** setting example of the ABB robot. The protocol settings of other robot brands are different. The software will automatically set this parameter and you just need to keep the default setting.
- The port number for the **Host IP address** parameter is **50000** by default. If the port number determined in the communication solution is not 50000, please change it as required.

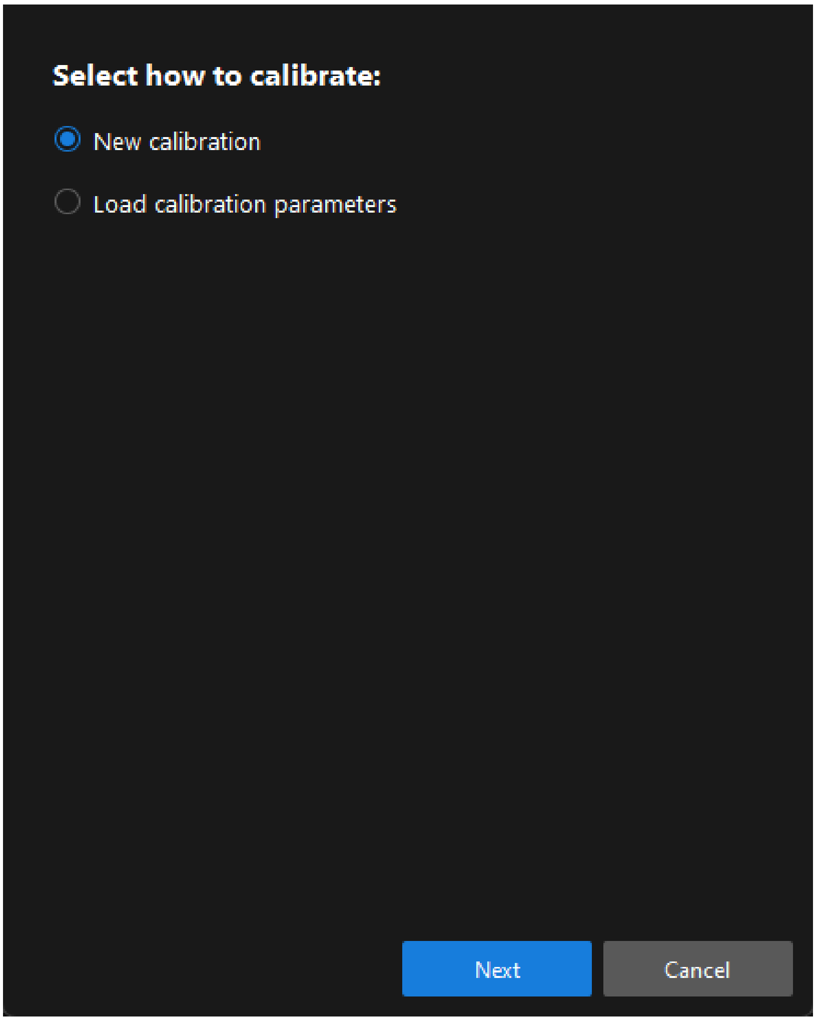
7. On the robot teach pendant, **select the automatic calibration program, teach the start point and run the program**. For details on the calibration programs of different robots, refer to the section Standard Interface Communication. After the program is started, the log message "Entering the calibration process, please start the calibration in Mech-Vision" will be printed in the Console log panel.
8. Return to Mech-Vision, confirm that "Connected" status message is displayed in the **Connect the robot** area, and click the **[Calibrate]** button.The **Calibration (Eye in Hand)** window will be prompted.

Pre-calibration Configuration (Master-Control)

If the Master-Control is used to communicate with the robot, follow these steps:

1. Open Mech-Vision, and click the **[Camera Calibration (Standard)]** button in the toolbar.**The Configuration before Calibration window** will be prompted.

2. In the **Select how to calibrate** window, select the **New calibration** radio button, and then click the **[Next]** button.

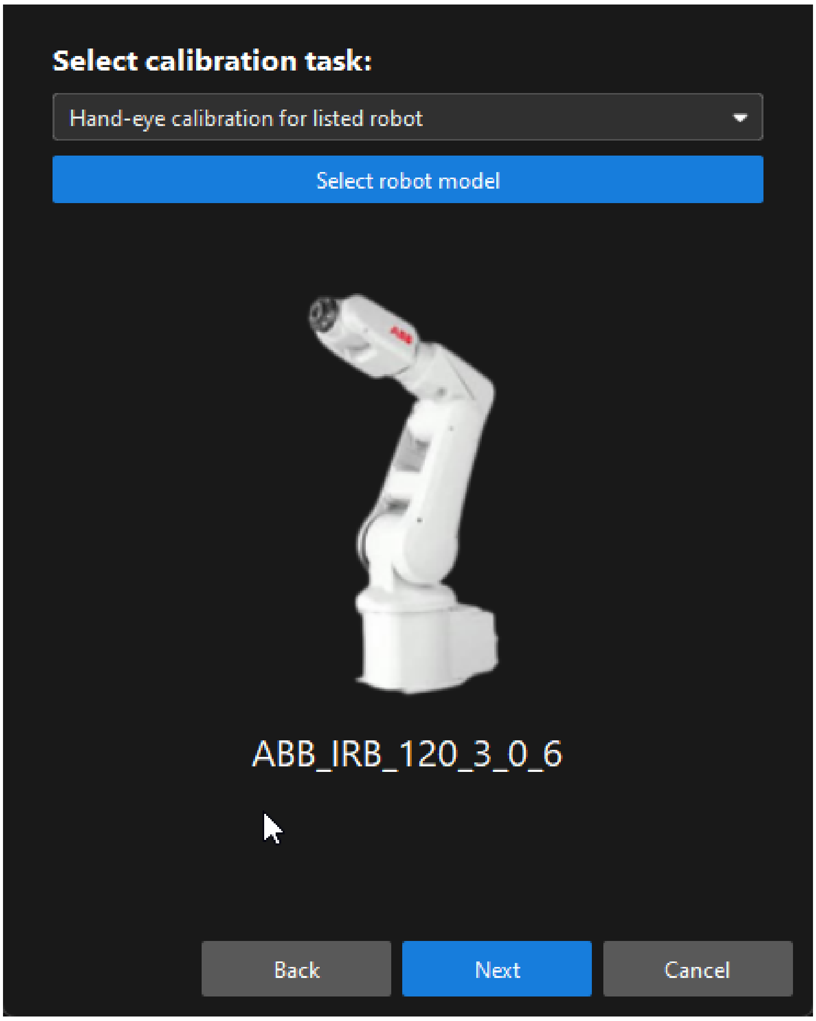


Select how to calibrate:

● New calibration

○ Load calibration parameters

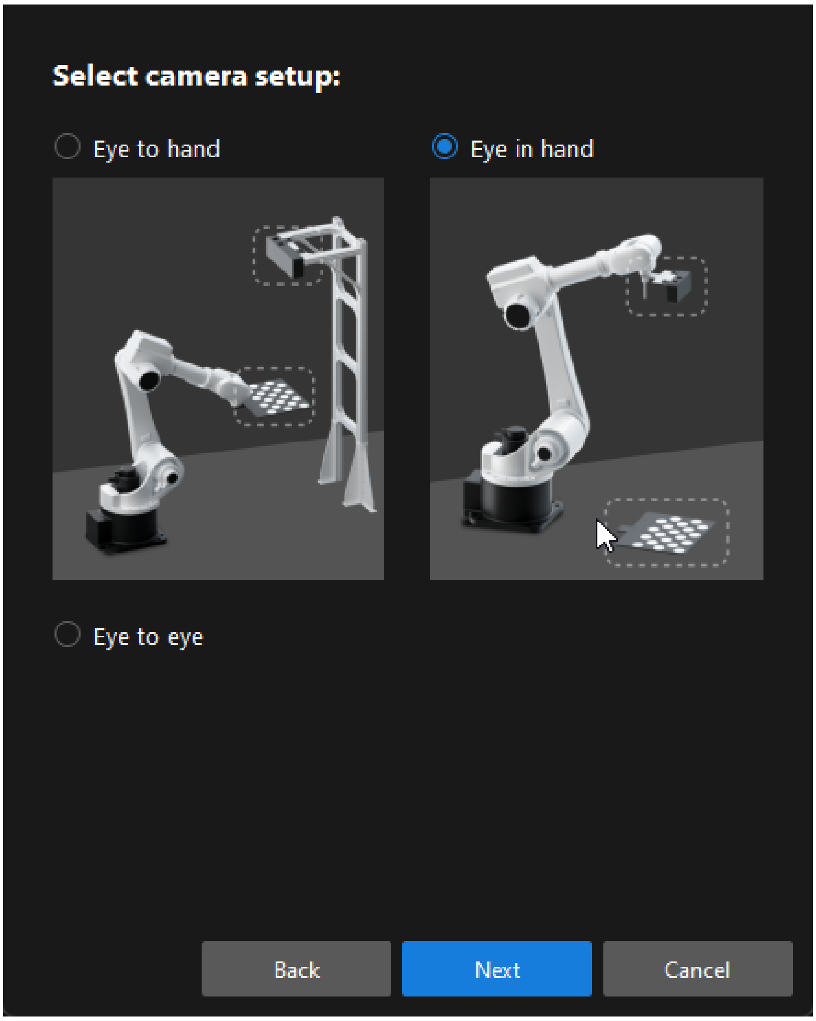
3. In the **Select calibration task** window, select **Hand-eye calibration for listed robot** from the drop-down list box, click the **[Select robot model]** button to select the robot model used by the project, and then click the **[Next]** button.



Select calibration task:

Hand-eye calibration for listed robot

4. In the **Select camera setup** window, select the **Eye in hand** radio button, and then click the [Next] button.



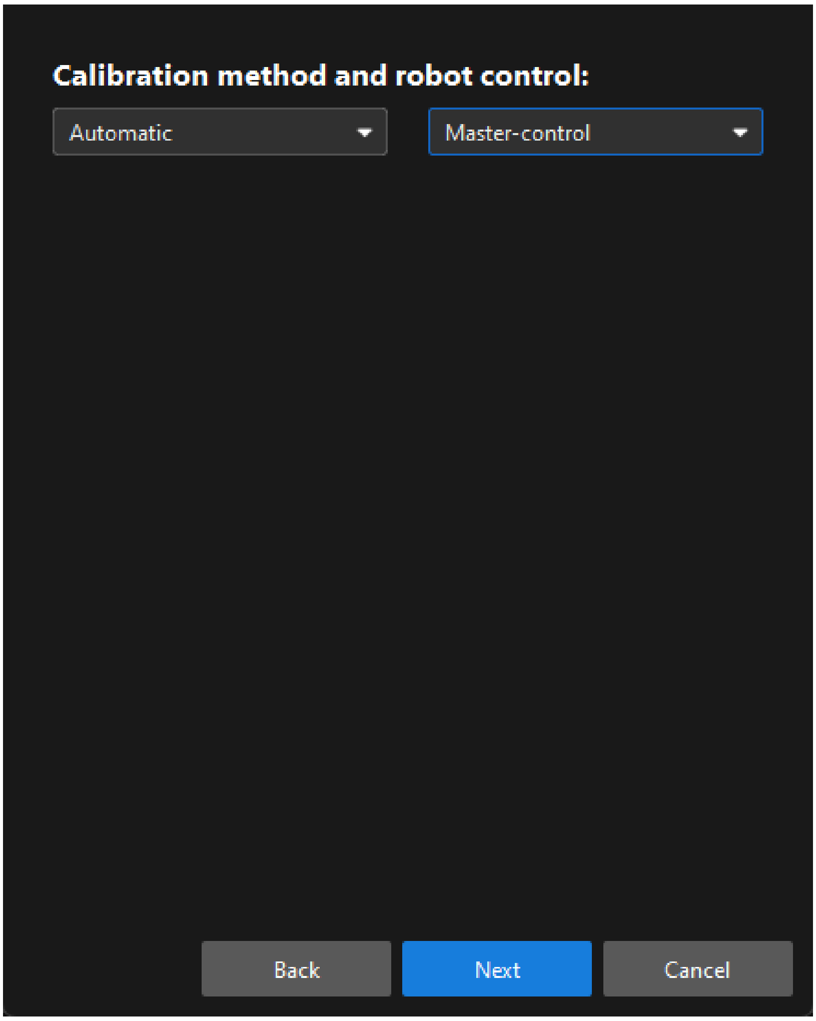
Select camera setup:

○ Eye to hand

● Eye in hand

○ Eye to eye

5. In the **Calibration method and robot control** window, select **Automatic** and **Master-control**, and then click the **[Next]** button.

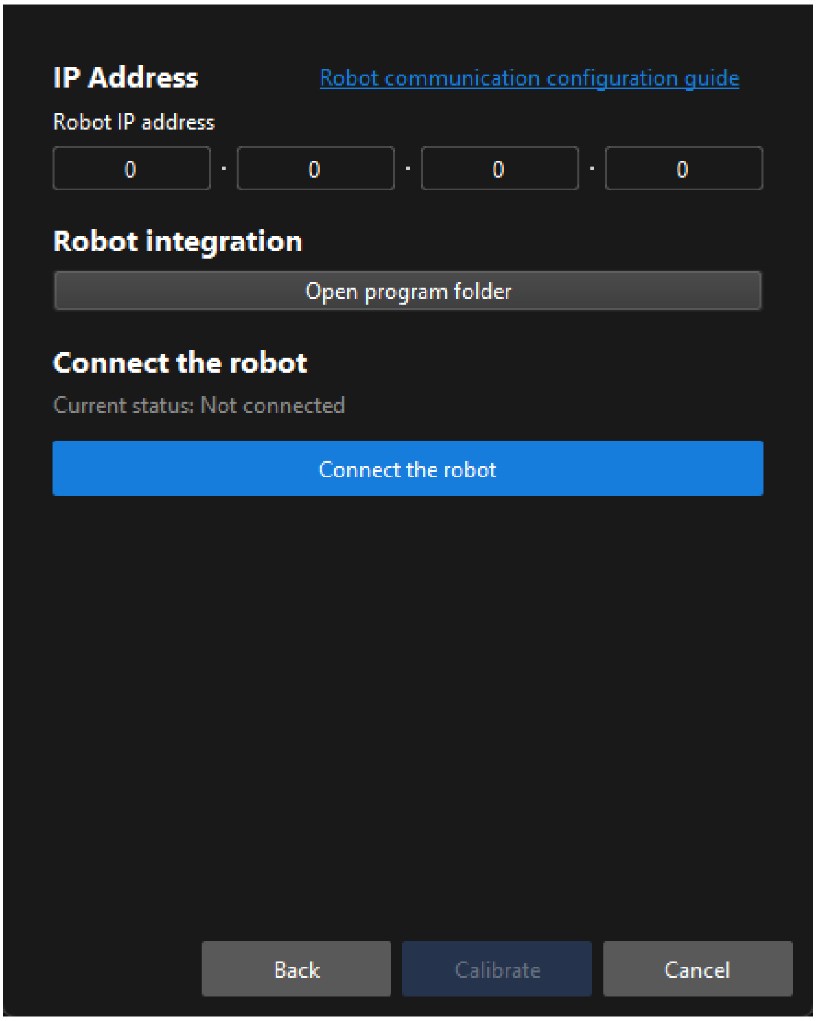


Calibration method and robot control:

Automatic

Master-control

6. In the prompted window, set the **Robot IP address** parameter.



IP Address

Robot communication configuration guide

Robot IP address

0 . 0 . 0 . 0

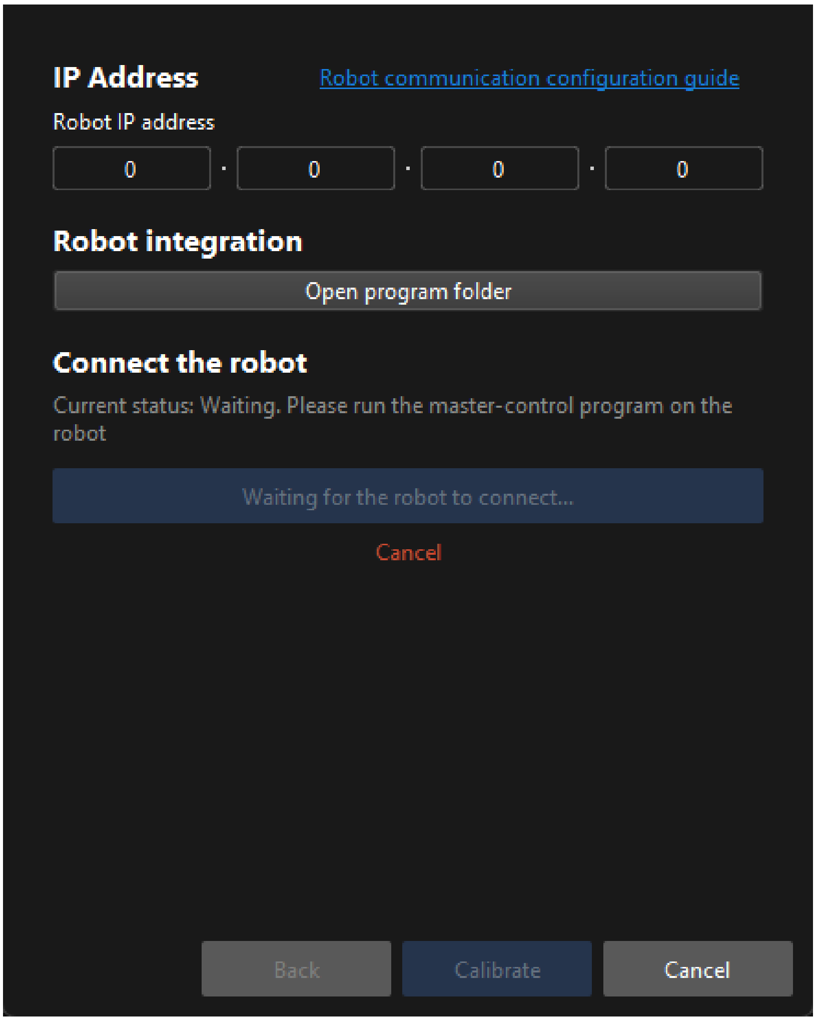
Robot integration

Open program folder

Connect the robot

Current status: Not connected

7. On the robot teach pendant, **select and run the Master-Control program**. For the instructions of different robots, refer to the section Master-control Communication.
8. Return to Mech-Vision, and click the **[Connect the robot]** button in the **Connect the robot** area. The button will turn into **Waiting for the robot to connect...**



IP Address

Robot communication configuration guide

Robot IP address

0 . 0 . 0 . 0

Robot integration

Open program folder

Connect the robot

Current status: Waiting. Please run the master-control program on the robot

Waiting for the robot to connect...

Cancel

Back Calibrate Cancel

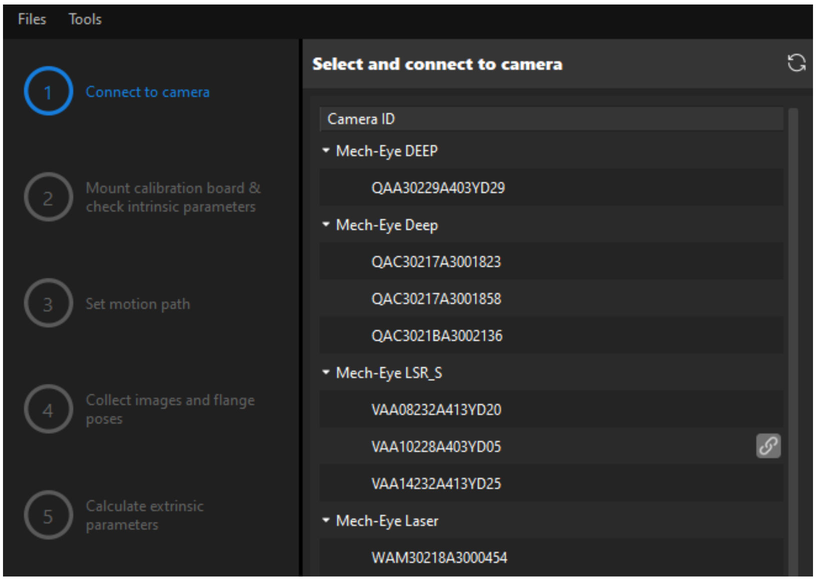
9. Wait until the "Connected" status message is displayed in the **Connect the robot** area, and then click the **[Calibrate]** button.The **Calibration (Eye in Hand)** window will be prompted.

Till now, you have completed the pre-calibration configuration and can start the calibration procedure.

Calibration Procedure

Connect to the Camera

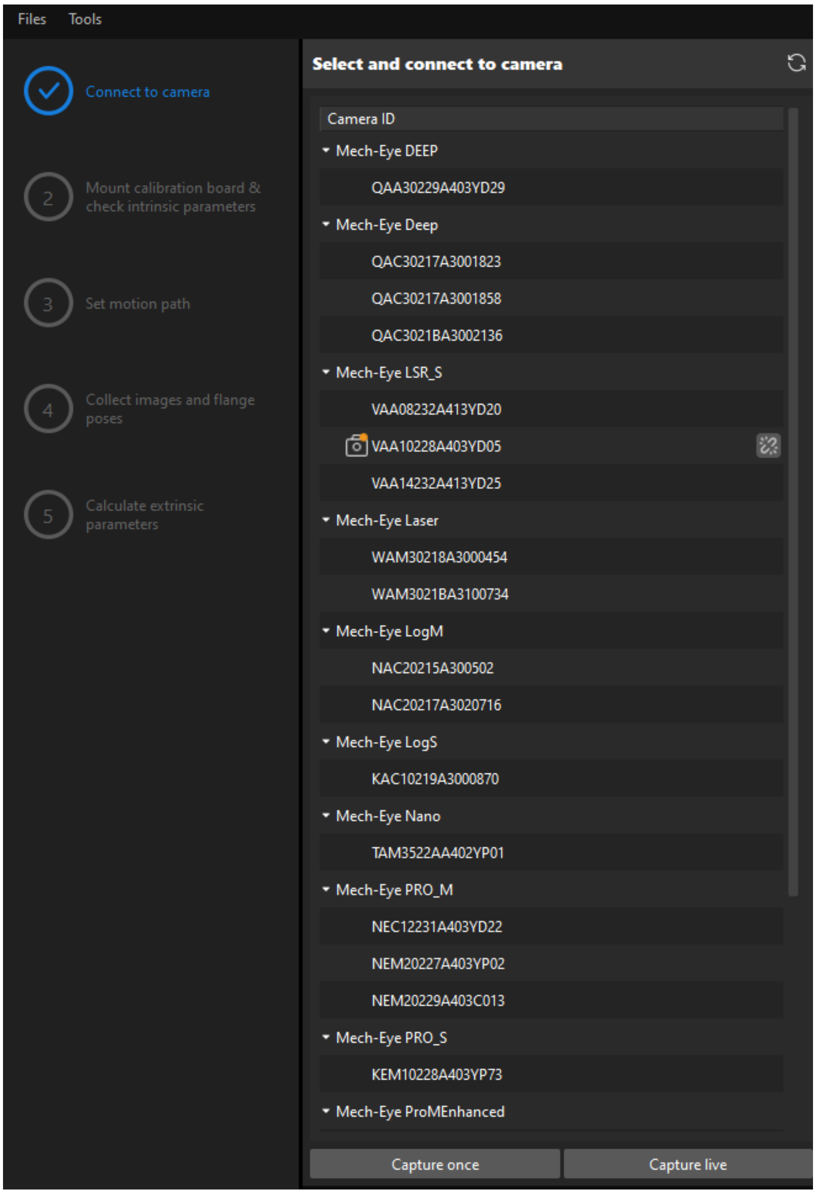
1. In the **Connect to camera** step, select the camera to connect in the Camera ID list, and then click the 🔗 button or double-click the camera entry to connect to it.


Files
Tools
1 Connect to camera
2 Mount calibration board & check intrinsic parameters
3 Set motion path
4 Collect images and flange poses
5 Calculate extrinsic parameters

Select and connect to camera

Camera ID
- Mech-Eye DEEP
  - QAA30229A403YD29
- Mech-Eye Deep
  - QAC30217A3001823
  - QAC30217A3001858
  - QAC3021BA3002136
- Mech-Eye LSR_S
  - VAA08232A413YD20
  - VAA10228A403YD05
  - VAA14232A413YD25
- Mech-Eye Laser
  - WAM30218A3000454

2. After the camera is connected, click the **[Capture once]** or **[Capture live]** button.


Files
Tools
Connect to camera
2 Mount calibration board & check intrinsic parameters
3 Set motion path
4 Collect images and flange poses
5 Calculate extrinsic parameters
Capture once
Capture live

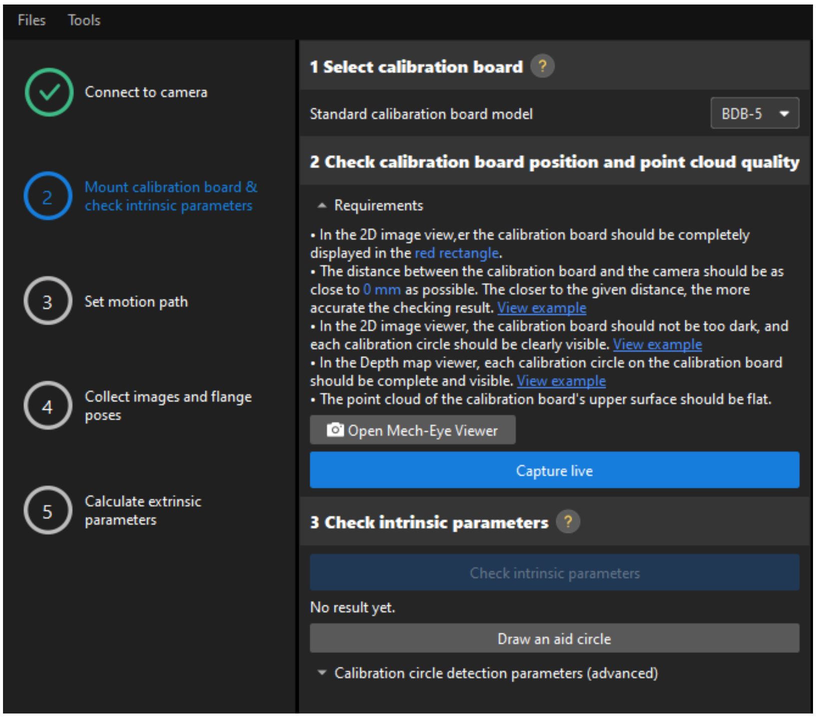
Select and connect to camera

Camera ID
- Mech-Eye DEEP
  - QAA30229A403YD29
- Mech-Eye Deep
  - QAC30217A3001823
  - QAC30217A3001858
  - QAC3021BA3002136
- Mech-Eye LSR_S
  - VAA08232A413YD20
  - VAA10228A403YD05
  - VAA14232A413YD25
- Mech-Eye Laser
  - WAM30218A3000454
  - WAM3021BA3100734
- Mech-Eye LogM
  - NAC20215A300502
  - NAC20217A3020716
- Mech-Eye LogS
  - KAC10219A3000870
- Mech-Eye Nano
  - TAM3522AA402YP01
- Mech-Eye PRO_M
  - NEC12231A403YD22
  - NEM20227A403YP02
  - NEM20229A403C013
- Mech-Eye PRO_S
  - KEM10228A403YP73
- Mech-Eye ProMEnhanced

3. In the **Image viewer** panel, ensure that the camera can capture images normally and click the **[Next]** button on the bottom bar.

Mount Calibration Board & Check Intrinsic Parameters

1. In the **Mount calibration board & check intrinsic parameters** step, select the calibration board model for the **Standard calibration board model** parameter according to its model nameplate in the **1 Select calibration board** area.


Files
Tools
Connect to camera
2 Mount calibration board & check intrinsic parameters
3 Set motion path
4 Collect images and flange poses
5 Calculate extrinsic parameters

1 Select calibration board

Standard calibration board model: BDB-5

2 Check calibration board position and point cloud quality

Requirements

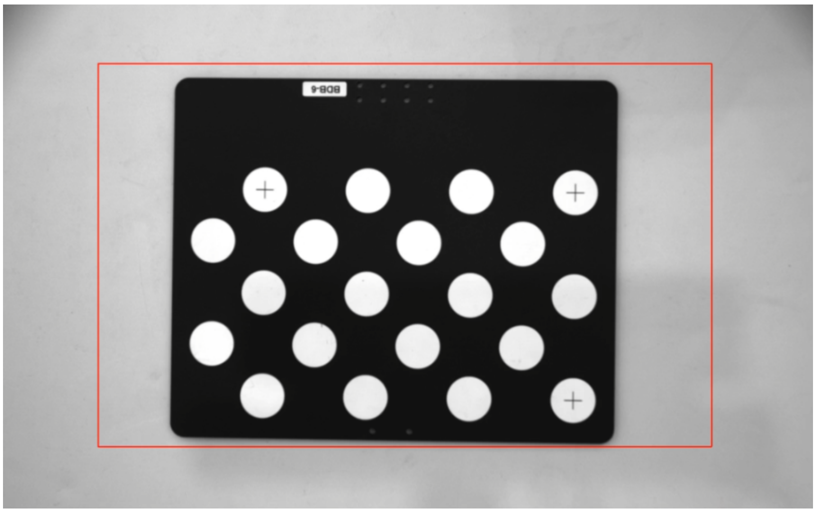
- In the 2D image view,er the calibration board should be completely displayed in the red rectangle.
- The distance between the calibration board and the camera should be as close to 0 mm as possible. The closer to the given distance, the more accurate the checking result. View example
- In the 2D image viewer, the calibration board should not be too dark, and each calibration circle should be clearly visible. View example
- In the Depth map viewer, each calibration circle on the calibration board should be complete and visible. View example
- The point cloud of the calibration board's upper surface should be flat.

3 Check intrinsic parameters

No result yet.

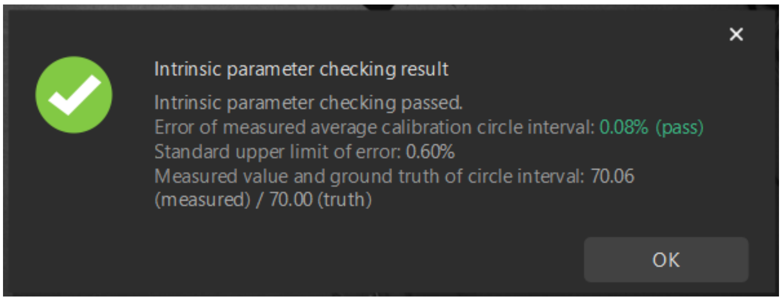
2. In the **2 Check calibration board position and point cloud quality** area, read carefully the requirements on the calibration board position and point cloud quality, and then click the **[Capture live]** button.The **[Capture live]** button will turn into **[Stop capturing and detect position]**.

3. Manually move the calibration board to the proper position, and ensure that the 2D image and depth map of the calibration board meet the requirements, and then click the **[Stop capturing and detect position]** button.



If the captured images do not meet the requirements, click the [**Open Mech-Eye Viewer**] button to open the Mech-Eye Viewer software, adjust the 2D and 3D exposure parameters and re-capture images. Please note that you need to change the **Parameter group** parameter to "calib" first.

4. In the **3 Check intrinsic parameters** area, click the [**Check intrinsic parameters**] button.
5. Confirm the results of the camera intrinsic parameter check.
   - If the camera intrinsic parameter check passes, click the [**OK**] button in the prompted window, and then click the [**Next**] button on the bottom bar.



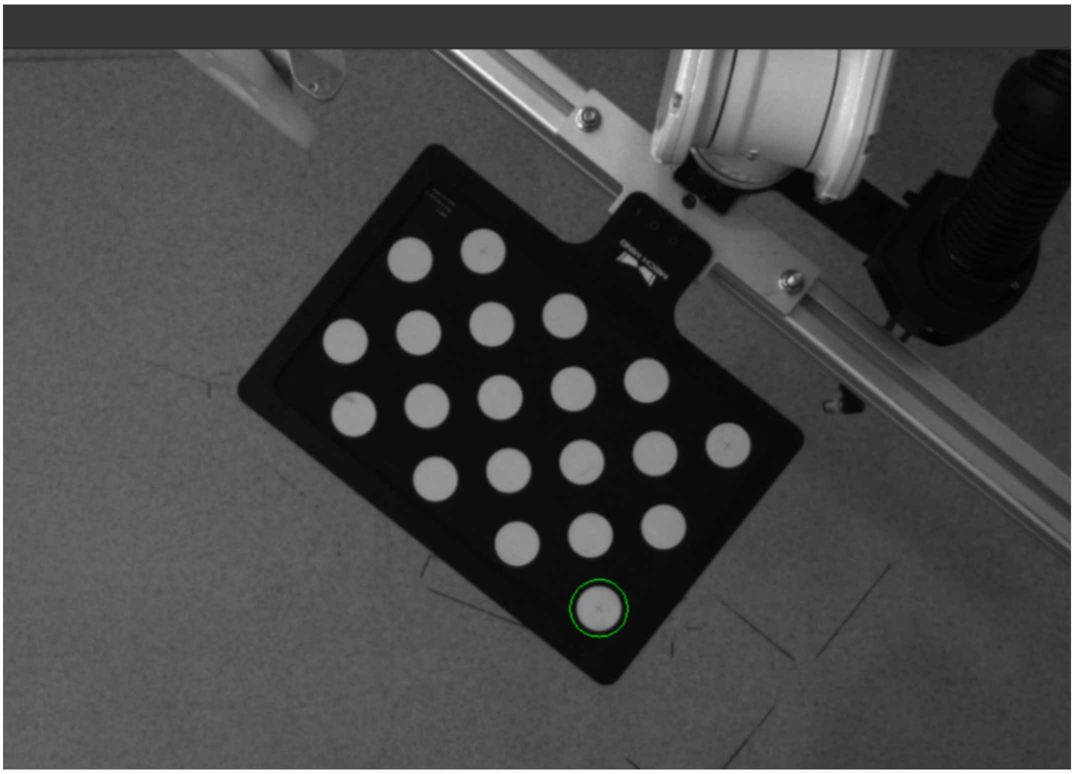
Intrinsic parameter checking result

Intrinsic parameter checking passed.
Error of measured average calibration circle interval: 0.08% (pass)
Standard upper limit of error: 0.60%
Measured value and ground truth of circle interval: 70.06 (measured) / 70.00 (truth)

   - If the camera intrinsic parameter check fails, adjust detection parameters by drawing aid circles or manually adjusting the calibration circle detection parameters, and then click the [**Recheck intrinsic parameters**] button.

Draw an Aid Circle

1. To draw an aid circle, click the [**Draw an aid circle**] button.
2. In the right **Image viewer** panel, right-click the calibration board image and clear the **Fit to window** checkbox, and press the `Ctrl` key and drag the roller to adjust the image to a suitable size.
3. Move the mouse pointer to the cross center point of the calibration circle, press the left mouse button and make the aid circle completely include the calibration circle and then release it.



4. Click the **[Recheck intrinsic parameters]** button, and confirm that the camera intrinsic parameter check passes.

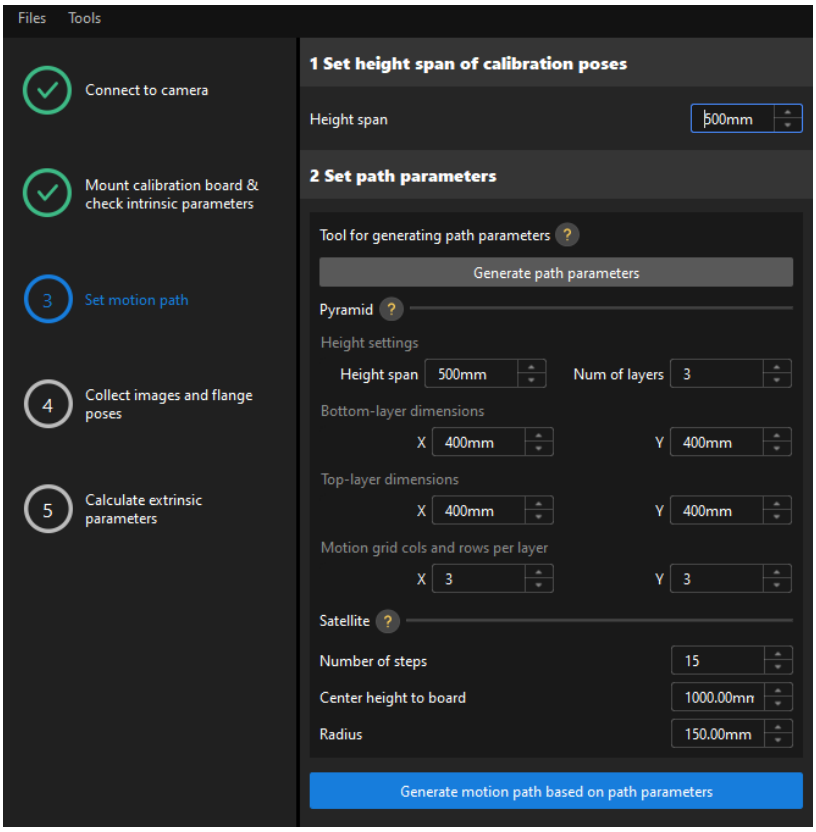
Manually Adjust Detection Parameters

To manually adjust detection parameters, click **[Calibration circle detection parameters (advanced)]**, adjust parameter settings and click the **[Recheck intrinsic parameters]** button to confirm whether the camera intrinsic parameter check passes.

If the calibration circles are still not detected, please adjust the camera settings according to the on-site operating conditions.

Set Motion Path

1. In the **Set motion path** step, set the **Height span** parameter in the **1 Set height span of calibration poses** area.


Files
Tools
Connect to camera
Mount calibration board & check intrinsic parameters
3 Set motion path
4 Collect images and flange poses
5 Calculate extrinsic parameters
Generate path parameters
Generate motion path based on path parameters

1 Set height span of calibration poses

Height span: 500mm

2 Set path parameters

Tool for generating path parameters ?

Pyramid ?

Height settings

Height span: 500mm — Num of layers: 3

Bottom-layer dimensions

X: 400mm — Y: 400mm

Top-layer dimensions

X: 400mm — Y: 400mm

Motion grid cols and rows per layer

X: 3 — Y: 3

Satellite ?

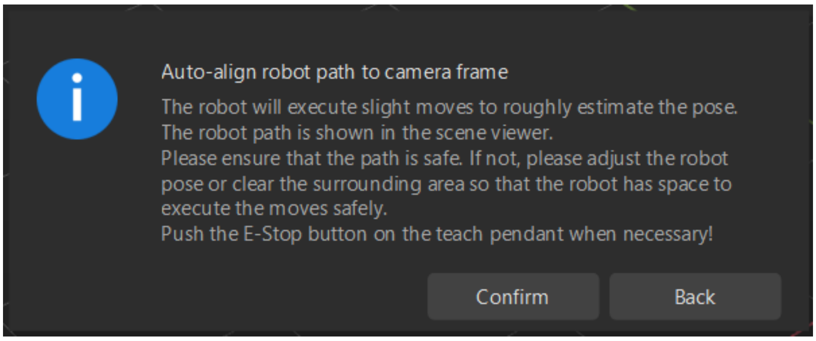
Number of steps: 15

Center height to board: 1000.00mn

Radius: 150.00mm

The **Height span** parameter should be set according to the recommended working distance range of the camera and the size of the robot's working space.

2. Click the [**Generate path parameters**] button, and click the [**OK**] button in the prompted window.

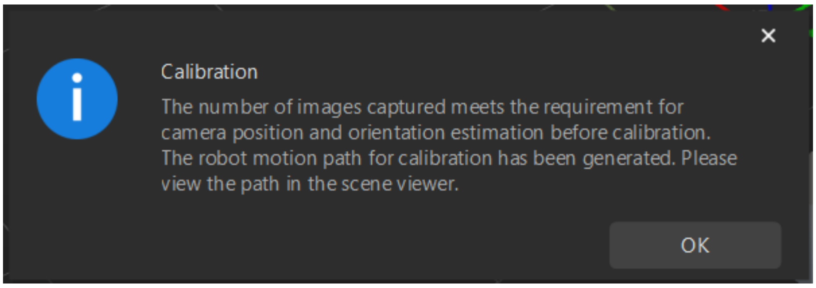


Auto-align robot path to camera frame

The robot will execute slight moves to roughly estimate the pose.
The robot path is shown in the scene viewer.
Please ensure that the path is safe. If not, please adjust the robot pose or clear the surrounding area so that the robot has space to execute the moves safely.
Push the E-Stop button on the teach pendant when necessary!

This operation will slightly move the robot and capture images. The whole process may take 10 to 15 seconds. Please ensure that the motion path is safe. In case of emergency, please tap the emergency stop button on the robot teach pendant to stop the robot immediately.

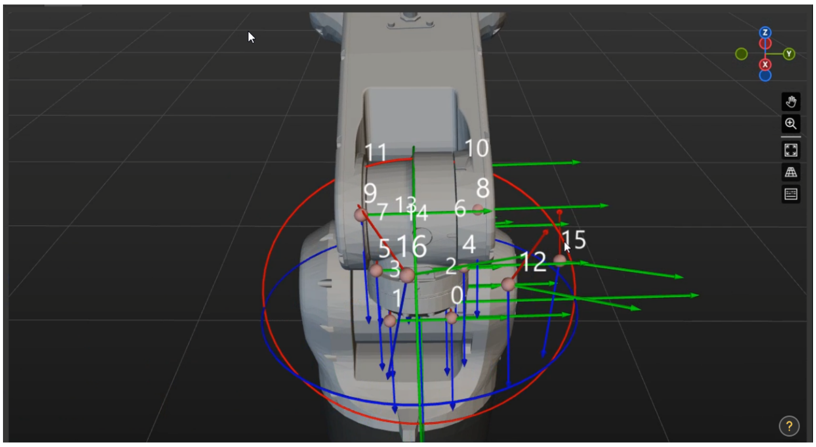
3. In the prompted **Calibrate** window, click the [**OK**] button.



Calibration

The number of images captured meets the requirement for camera position and orientation estimation before calibration. The robot motion path for calibration has been generated. Please view the path in the scene viewer.

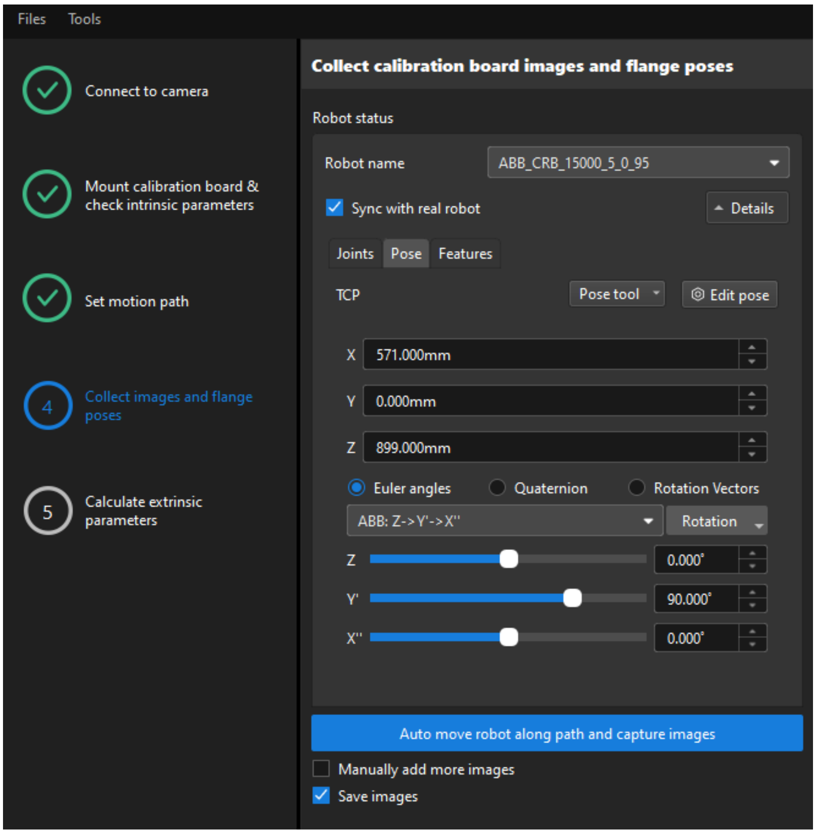
4. Confirm that waypoints are reasonable and will not collide with obstacles in the right scene viewer panel.



5. Click the **[Generate motion path based on path parameters]** button and then click the **[Next]** button on the bottom bar.

Collect Images and Flange Poses

1. In the **Collect images and flange poses** step, select the **Save images** checkbox.


Files
Tools
Connect to camera
Mount calibration board & check intrinsic parameters
Set motion path
4
Collect images and flange poses
5
Calculate extrinsic parameters
Collect calibration board images and flange poses
Robot status
Details
Joints
Pose
Features
TCP
Pose tool
Edit pose
Rotation
Auto move robot along path and capture images

Robot name: ABB_CRB_15000_5_0_95

☑ Sync with real robot

X 571.000mm

Y 0.000mm

Z 899.000mm

● Euler angles ○ Quaternion ○ Rotation Vectors

ABB: Z->Y'->X''

Z 0.000°

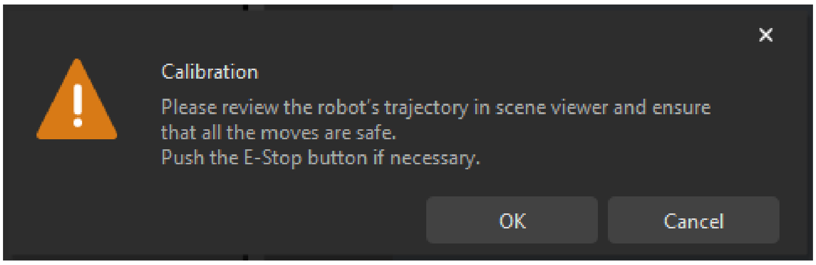
Y' 90.000°

X'' 0.000°

□ Manually add more images

☑ Save images

2. If the robot uses the Master-Control to communicate with the Vision System, select the **Sync with real robot** checkbox (selected by default, recommended), and click **Details** to confirm that the robot’s JPs and pose data are the same as those on the teach pendant. Clear the checkbox if you do not need to synchronize the data with the real robot.
3. Click the **[Auto move robot along path and capture images]** button, read the safety window carefully, and then click the **[OK]** button.



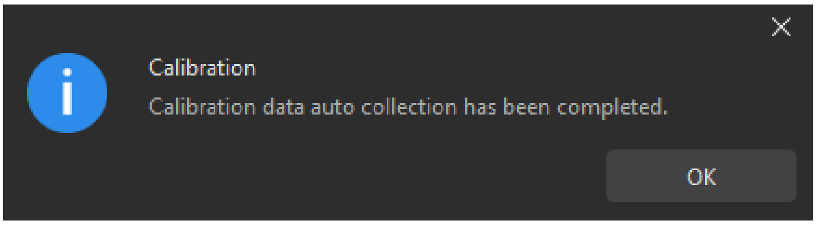
Calibration

Please review the robot’s trajectory in scene viewer and ensure that all the moves are safe.
Push the E-Stop button if necessary.

4. Wait until the robot finishes moving along the preset path and the camera finishes capturing images on all waypoints. In the right **Image viewer** panel, all the captured images are displayed.

- Please stay away from the robot working area to keep safe.
- Clicking the Stop Robot button can exit the calibration process. But the robot will not stop until it finishes the current waypoint. In case of emergency, please tap the emergency stop button on the robot teach pendant to stop the robot immediately (the robot needs to be reconnected after tapping the emergency stop button).

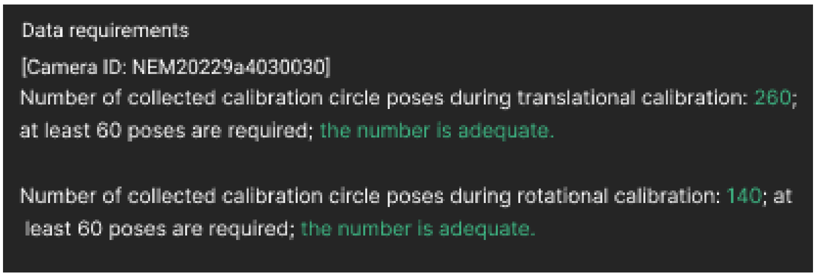
5. After all images are captured, click the **[OK]** button in the prompted window.



Calibration

Calibration data auto collection has been completed.

6. Confirm that the collected calibration data meets the data requirements, and then click the **[Next]** button on the bottom bar.



Data requirements

[Camera ID: NEM20229a4030030]

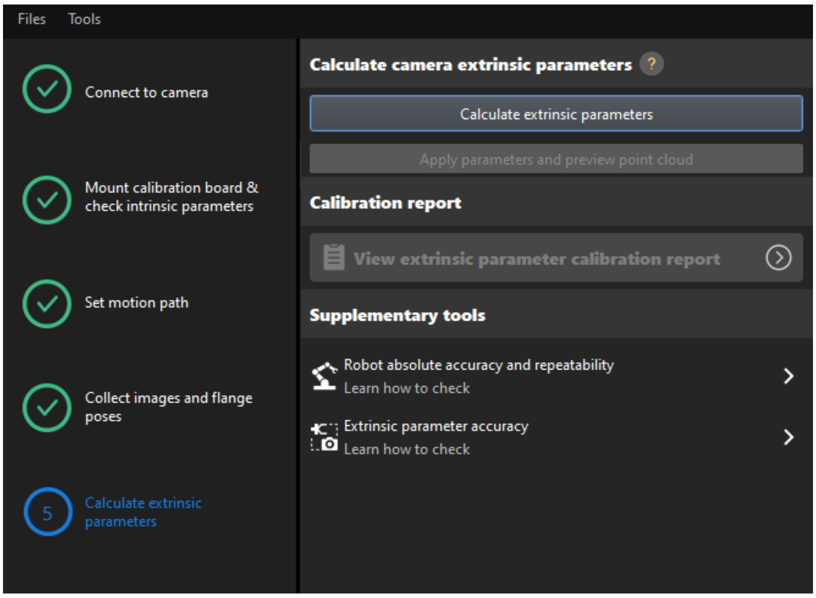
Number of collected calibration circle poses during translational calibration: 260; at least 60 poses are required; the number is adequate.

Number of collected calibration circle poses during rotational calibration: 140; at least 60 poses are required; the number is adequate.

If the requirements are not met, please manually move the robot (through the teach pendant or Mech-Viz), select the **Manually add more images** checkbox, and click the **[Add an image and record flange pose]** button to add the calibration board image and enter the robot flange pose.

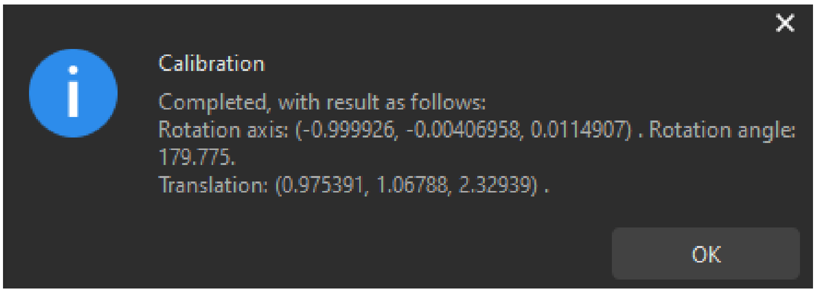
Calculate Extrinsic Parameters

1. In the **Calculate extrinsic parameters** step, click the **[Calculate extrinsic parameters]** button.



Files
Tools
Connect to camera
Mount calibration board & check intrinsic parameters
Set motion path
Collect images and flange poses
5
Calculate extrinsic parameters
Calculate camera extrinsic parameters
?
Calculate extrinsic parameters
Apply parameters and preview point cloud
Calibration report
View extrinsic parameter calibration report
Supplementary tools
Robot absolute accuracy and repeatability
Learn how to check
Extrinsic parameter accuracy
Learn how to check

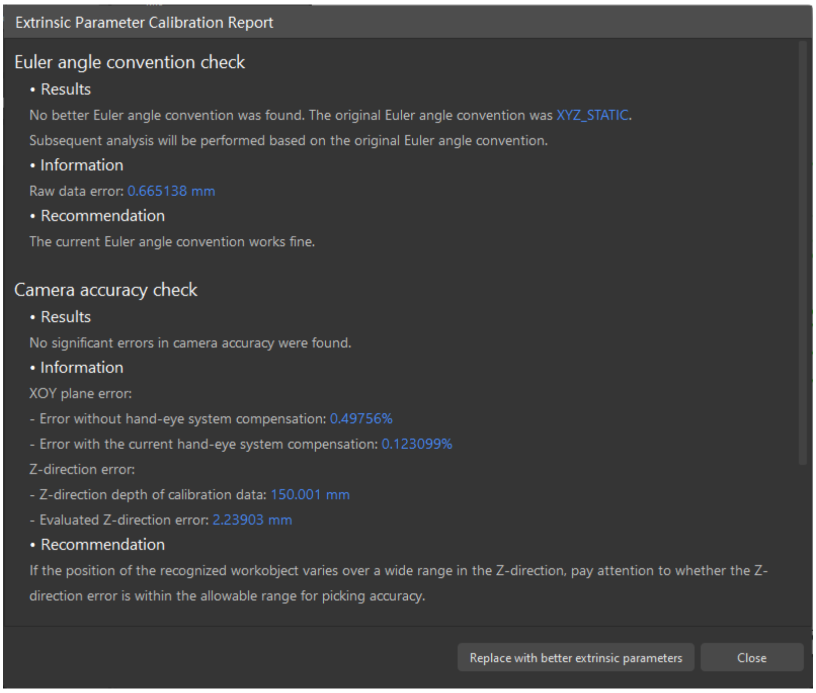
2. In the prompted window indicating calibration success, click the **OK** button.



Calibration

Completed, with result as follows:
Rotation axis: (-0.999926, -0.00406958, 0.0114907) . Rotation angle: 179.775.
Translation: (0.975391, 1.06788, 2.32939) .

3. Click the [**View extrinsic parameter calibration report**] button. The extrinsic parameter calibration report will be automatically displayed after it is generated.



Extrinsic Parameter Calibration Report

Euler angle convention check

- Results

No better Euler angle convention was found. The original Euler angle convention was XYZ_STATIC.
Subsequent analysis will be performed based on the original Euler angle convention.

- Information

Raw data error: 0.665138 mm

- Recommendation

The current Euler angle convention works fine.

Camera accuracy check

- Results

No significant errors in camera accuracy were found.

- Information

XOY plane error:
- Error without hand-eye system compensation: 0.49756%
- Error with the current hand-eye system compensation: 0.123099%
Z-direction error:
- Z-direction depth of calibration data: 150.001 mm
- Evaluated Z-direction error: 2.23903 mm

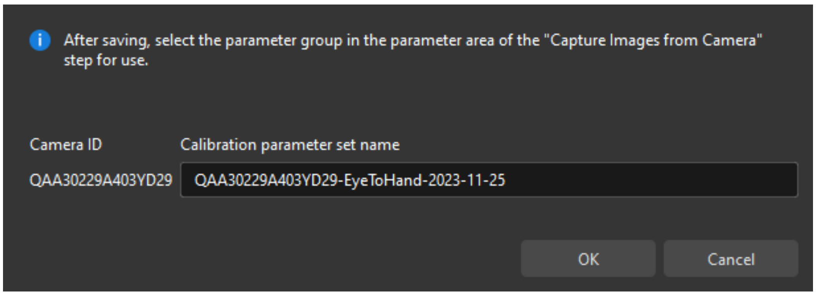
- Recommendation

If the position of the recognized workobject varies over a wide range in the Z-direction, pay attention to whether the Z-direction error is within the allowable range for picking accuracy.

- The extrinsic parameter calibration report displays the results of the Euler angle convention check, camera accuracy check, and robot absolute accuracy check.
- To evaluate the robot absolute accuracy and the accuracy of calculated extrinsic parameters, you can click [**Robot absolute accuracy and repeatability**] and [**Extrinsic parameter accuracy**] in the “Calculate extrinsic parameters” window and use the error analysis tool to perform robot absolute accuracy check and analyze extrinsic parameter error in EIH setup.

4. Confirm the results of the Euler angle convention check, camera accuracy check, and robot absolute accuracy check in the report, and then click the [**Close**] button to close the report.

5. Click the [**Save**] button on the bottom bar. In the prompted **Save Calibration Files** dialog box, click the [**OK**] button. The camera calibration result will be automatically saved in the “calibration” directory of the project.



After saving, select the parameter group in the parameter area of the "Capture Images from Camera" step for use.

Camera ID: QAA30229A403YD29

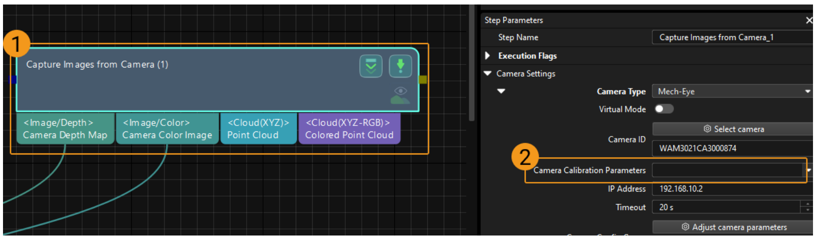
Calibration parameter set name: QAA30229A403YD29-EyeToHand-2023-11-25

Till now, the calibration process is completed.

Change the Extrinsic Parameter File

If the camera's extrinsic parameter file has been used by a Mech-Vision project, you need to change the extrinsic parameter file used by the Mech-Vision project to a new one after you finish the hand-eye calibration.

1. Select the **Capture Images from Camera** Step.



Step Parameters

Step Name: Capture Images from Camera_1

Execution Flags

Camera Settings

Camera Type: Mech-Eye

Virtual Mode

Camera ID: WAM3021CA3000874

Camera Calibration Parameters

IP Address: 192.168.10.2

Timeout: 20 s

2. In the **Step Parameters** panel, click ▼ of the **Camera Calibration Parameters** parameter, and select the newly calibrated extrinsic parameter file.

Calibration-Related Status Codes

If the robot uses the Standard Interface to communicate with the Vision System during hand-eye calibration, the Vision System will return status codes in response data to indicate the execution status of the commands. The status codes indicate both the normal execution results and execution failures. You can perform further troubleshooting according to the status codes.

Calibration Error Codes

Error code	Description
7001	Calibration: Parameter error
7002	Calibration: Calibration flange pose not provided by Mech-Vision
7003	Calibration: Calibration joint positions not provided by Mech-Vision
7004	Calibration: Robot failed to reach calibration point

Calibration Normal Status Codes

Normal Status Code	Description
7100	Calibration: Robot reaches calibration point successfully

Normal Status Code	Description
7101	Calibration: Pose received from Mech-Vision successfully

4.5.3.2. Automatic Calibration in the Eye-in-Hand Setup (Four-Axis Robots)

This how-to guide introduces how to complete the automatic calibration for four-axis robots in the eye-in-hand (EIH) setup.

Preparation before Calibration

Before hand-eye calibration, you need to finish the following preparations:

- Construct the Mech-Mind Vision System.
- Prepare the materials required for calibration.
- Adjust the effect of calibration board's point cloud.
- Complete the robot communication configuration.

Construct the Vision System

Construct the Mech-Mind Vision System by referring to the section Vision System Hardware Setup.

You need to use Mech-Eye Viewer, Mech-Vision and Mech-Viz during hand-eye calibration. Please ensure that they have been installed and are running the latest versions.

Prepare the Materials Required for Calibration

The automatic calibration in the EIH setup needs to use the calibration board.

Please prepare the calibration board according to the following requirements:

- Ensure that the circles of the calibration board are clearly visible and without obvious scratches, and the board does not suffer from deformations.
- In the EIH setup, please place the calibration board in the center of the working plane, where target objects are to be placed.

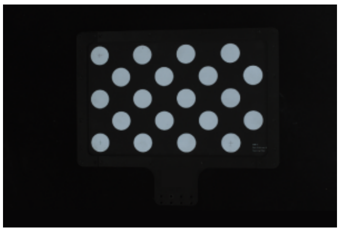
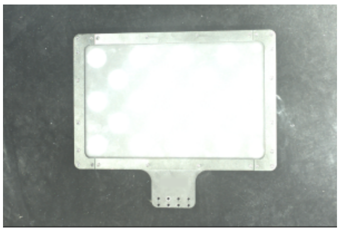
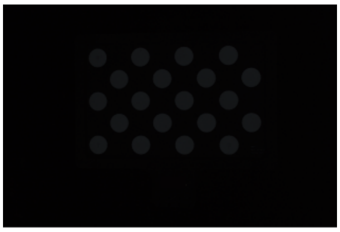
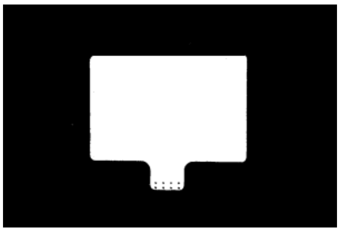
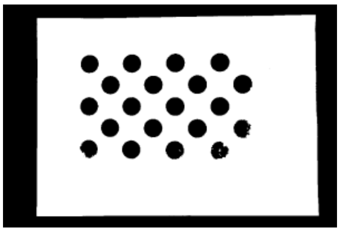
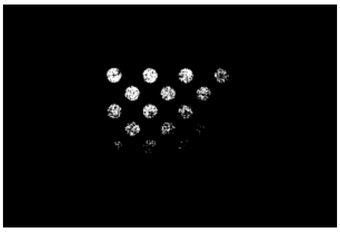
In addition, before calibration, move the robot to the starting point for calibration. For the automatic calibration in the EIH setup, the starting point is the camera's image-capturing position (the robot will move the camera upwards during calibration).

Adjust the Effect of Calibration Board's Point Cloud

1. Open Mech-Eye Viewer to adjust camera parameters.
2. Adjust the 2D parameters to make sure that the 2D image of the calibration board is clear and neither overexposed nor underexposed.
3. Adjust 3D parameters to make sure that the point clouds of the circles on the calibration board are complete and have clear contours. It is recommended to set **Surface Smoothing** and **Outlier Removal** to **Normal** in the **Point Cloud Processing** section to reduce the point cloud fluctuation range.

If the on-site ambient lights are not ideal and affect the quality of 2D images and point clouds, you can use shading or supplemental light to improve the lighting conditions.

4. Make sure that the images and point cloud of the calibration board are up to standard after completing preceding steps.

	Normal	Overexposed	Underexposed
2D image			
Point cloud			

Complete the Robot Communication Configuration

If the robot uses the Standard Interface to communicate with the Vision System, please set up the Standard Interface communication with the robot. For details, refer to the section Standard Interface Communication.

If the robot uses the Master-Control to communicate with the Vision System, please set up the Master-Control communication with the robot. For details, refer to the section Master-control Communication.

Pre-calibration Configuration

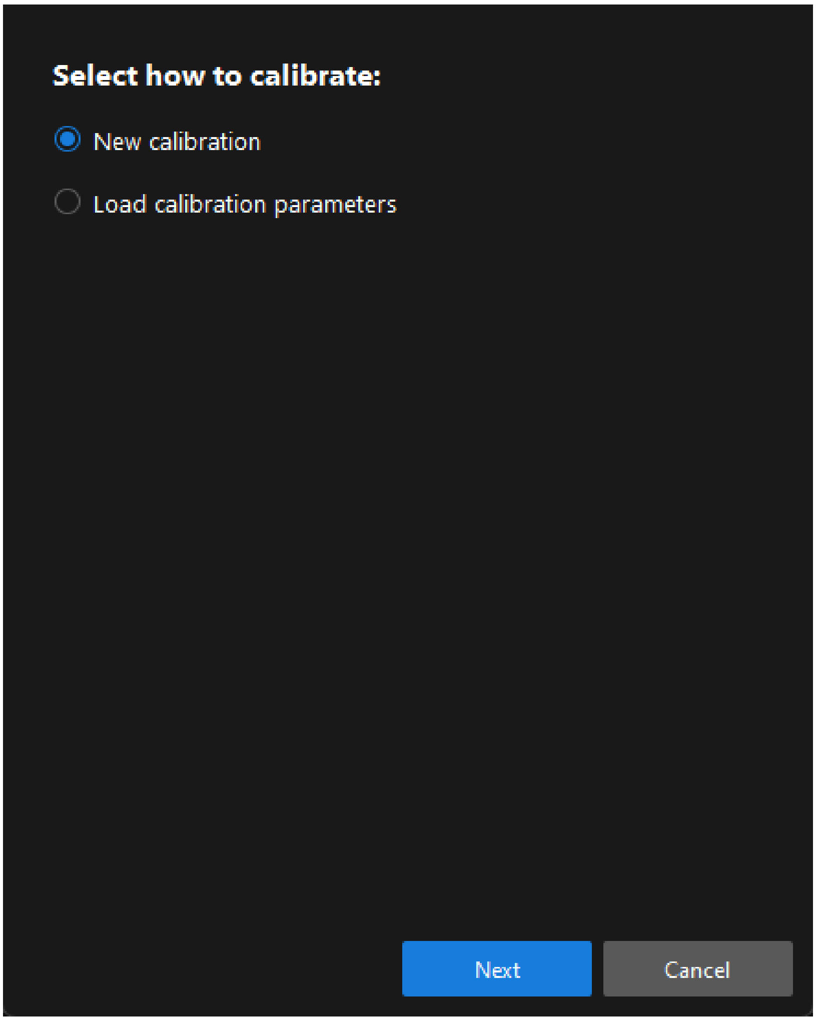
If the robot uses the Standard Interface to communicate with the Vision System, complete Pre-calibration Configuration (Standard Interface).

If the robot uses the Master-Control to communicate with the Vision System, complete Pre-calibration Configuration (Master-control).

Pre-calibration Configuration (Standard Interface)

If the Standard Interface is used to communicate with the robot, follow these steps:

1. Open Mech-Vision, and click the **[Camera Calibration (Standard)]** button in the toolbar.**The Configuration before Calibration window** will be prompted.
2. In the **Select how to calibrate** window, select the **New calibration** radio button, and then click the **[Next]** button.

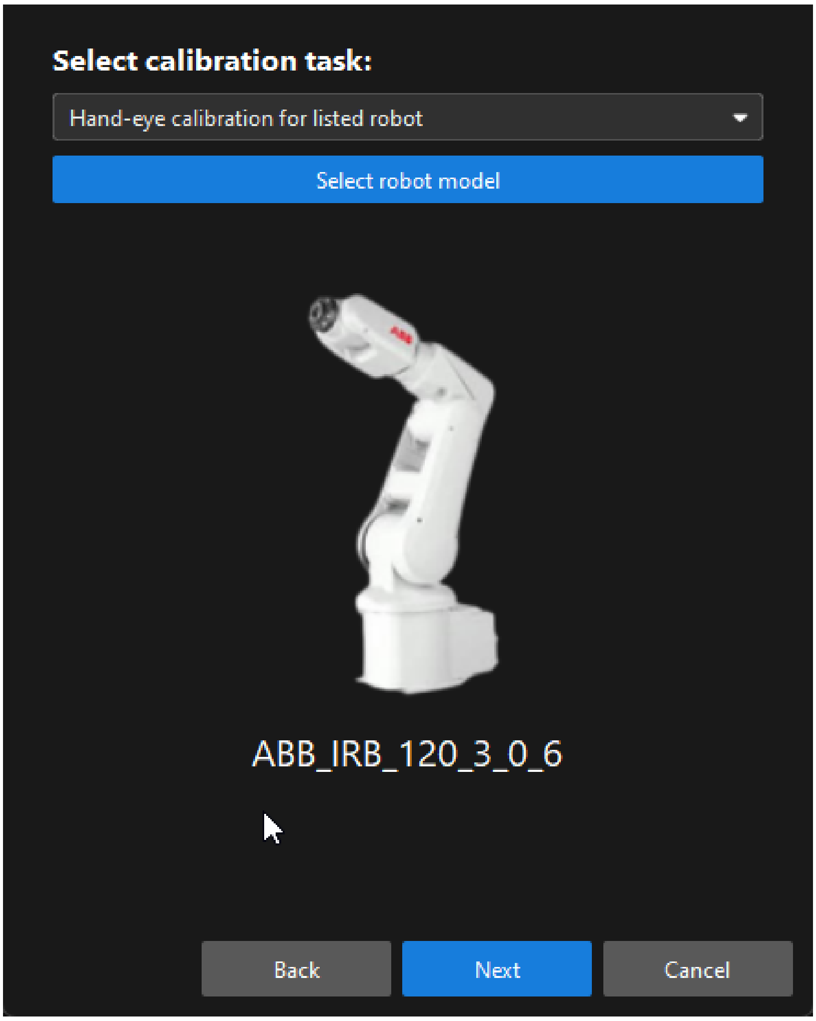


Select how to calibrate:

● New calibration

○ Load calibration parameters

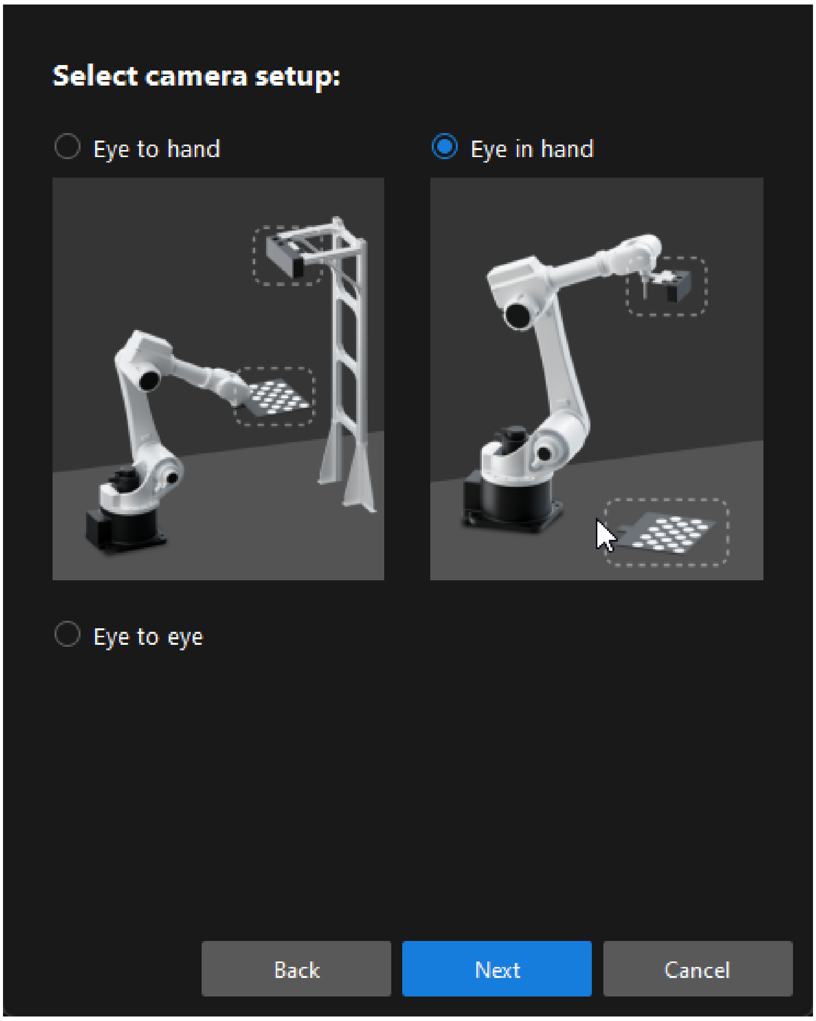
3. In the **Select calibration task** window, select **Hand-eye calibration for listed robot** from the drop-down list box, click the **[Select robot model]** button to select the robot model used by the project, and then click the **[Next]** button.



Select calibration task:

Hand-eye calibration for listed robot

4. In the **Select camera setup** window, select the **Eye in hand** radio button, and then click the **[Next]** button.



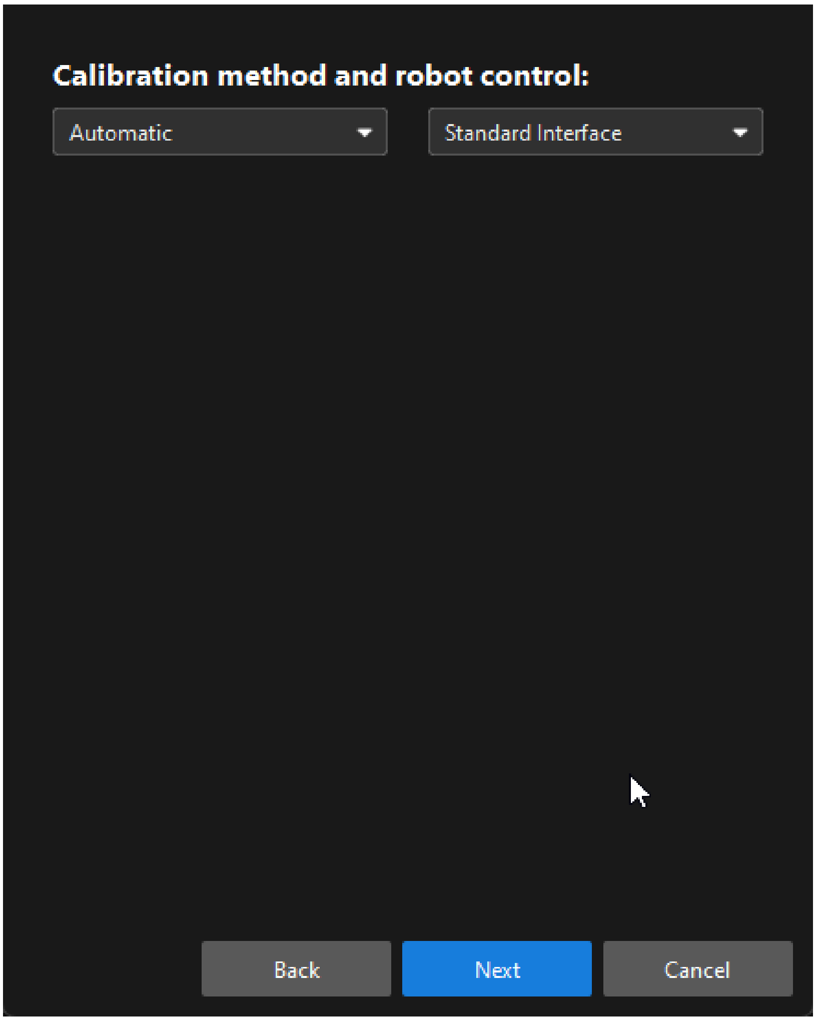
Select camera setup:

○ Eye to hand

● Eye in hand

○ Eye to eye

5. In the **Calibration method and robot control** window, select **Automatic** and **Standard Interface**, and then click the **[Next]** button.

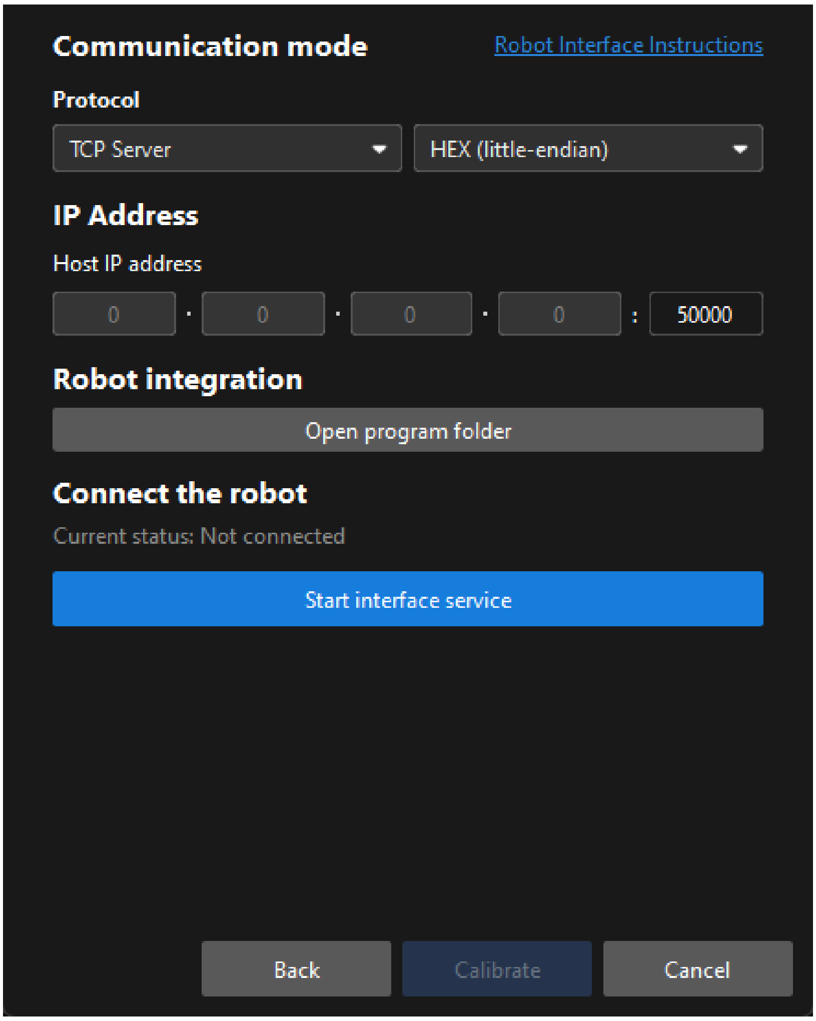


Calibration method and robot control:

Automatic

Standard Interface

6. In the new window, set parameters **Protocol** and **Host IP address**, and then click the **[Start interface service]** button. The button will turn into **Waiting for the robot to connect...**



Communication mode

Robot Interface Instructions

Protocol

TCP Server ▾ HEX (little-endian) ▾

IP Address

Host IP address

0 . 0 . 0 . 0 : 50000

Robot integration

Open program folder

Connect the robot

Current status: Not connected

Start interface service

- In the preceding figure, **TCP Server** and **Hex (Little-endian)** are only the **Protocol** setting example of the ABB robot. The protocol settings of other robot brands are different. The software will automatically set this parameter and you just need to keep the default setting.
- The port number for the **Host IP address** parameter is **50000** by default. If the port number determined in the communication solution is not 50000, please change it as required.

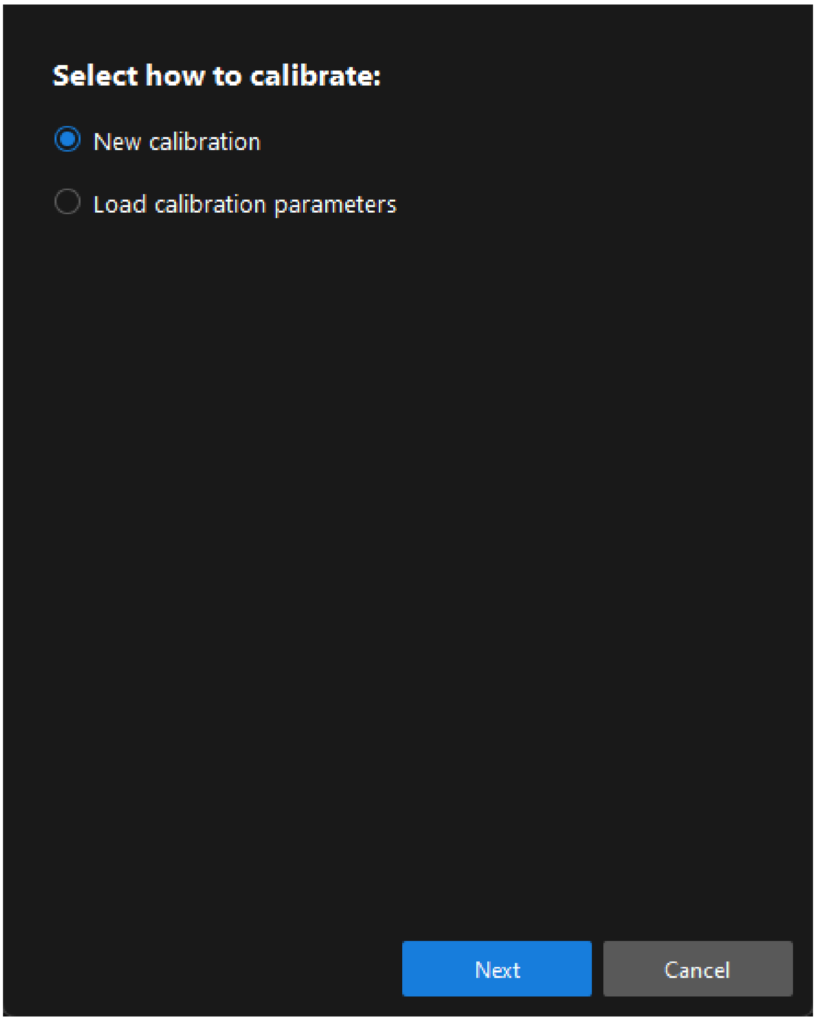
7. On the robot teach pendant, **select the automatic calibration program, teach the start point and run the program**. For details on the calibration programs of different robots, refer to the section Standard Interface Communication. After the program is started, the log message "Entering the calibration process, please start the calibration in Mech-Vision" will be printed in the Console log panel.
8. Return to Mech-Vision, confirm that "Connected" status message is displayed in the **Connect the robot** area, and click the **[Calibrate]** button.The **Calibration (Eye in Hand)** window will be prompted.

Pre-calibration Configuration (Master-Control)

If the Master-Control is used to communicate with the robot, follow these steps:

1. Open Mech-Vision, and click the **[Camera Calibration (Standard)]** button in the toolbar.**The Configuration before Calibration window** will be prompted.

2. In the **Select how to calibrate** window, select the **New calibration** radio button, and then click the **[Next]** button.

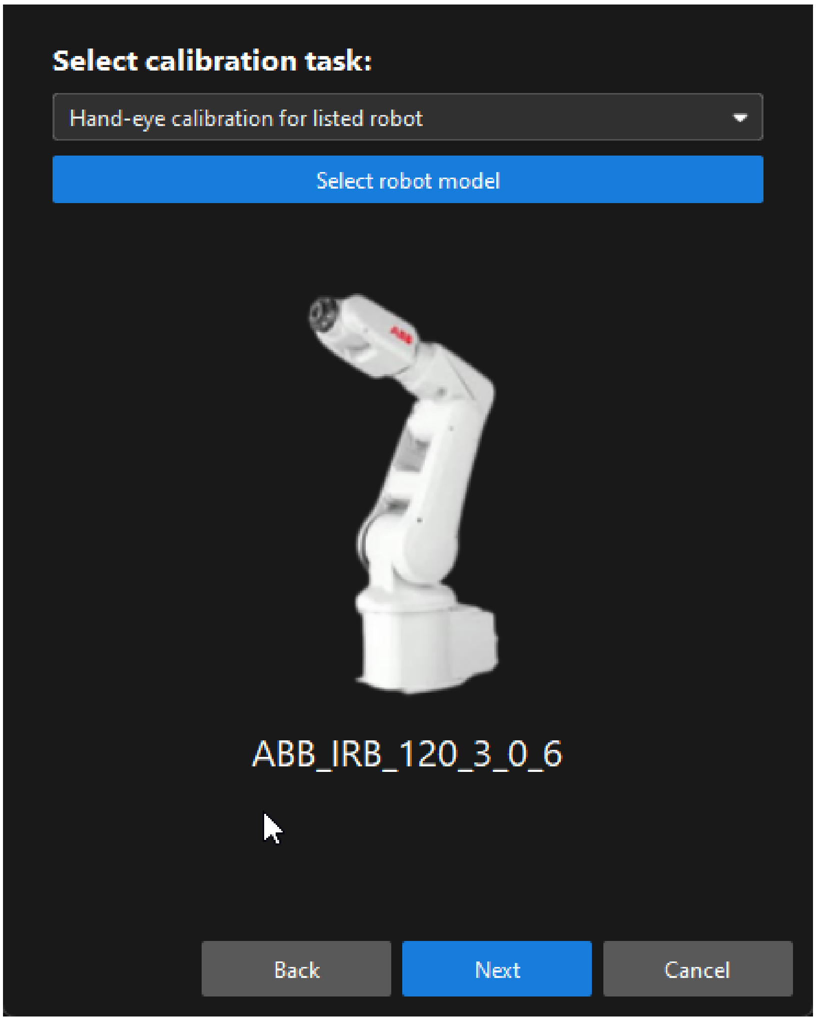


Select how to calibrate:

● New calibration

○ Load calibration parameters

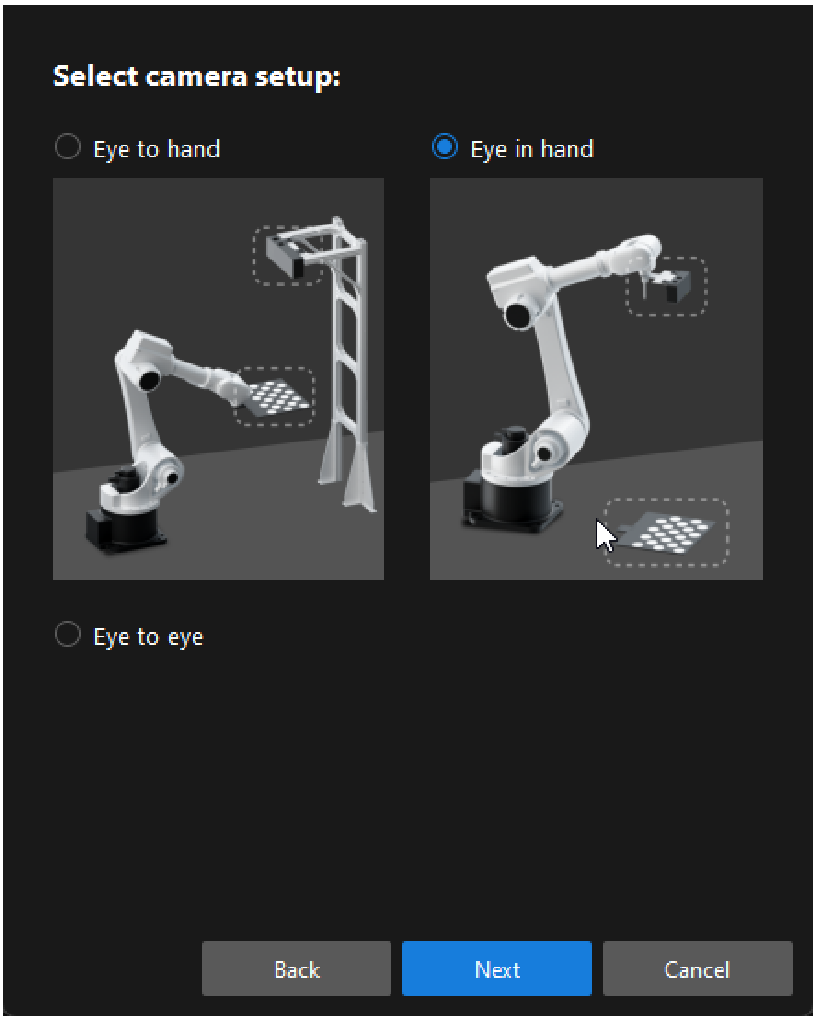
3. In the **Select calibration task** window, select **Hand-eye calibration for listed robot** from the drop-down list box, click the **[Select robot model]** button to select the robot model used by the project, and then click the **[Next]** button.



Select calibration task:

Hand-eye calibration for listed robot

4. In the **Select camera setup** window, select the **Eye in hand** radio button, and then click the **[Next]** button.



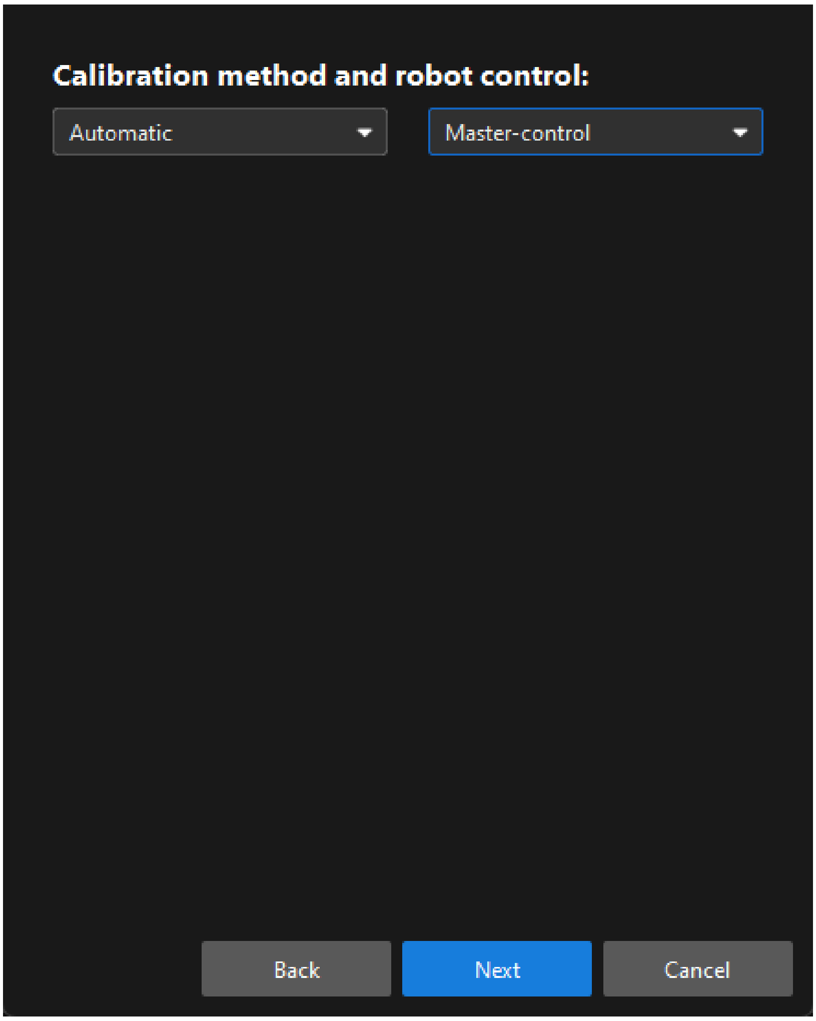
Select camera setup:

○ Eye to hand

● Eye in hand

○ Eye to eye

5. In the **Calibration method and robot control** window, select **Automatic** and **Master-control**, and then click the **[Next]** button.

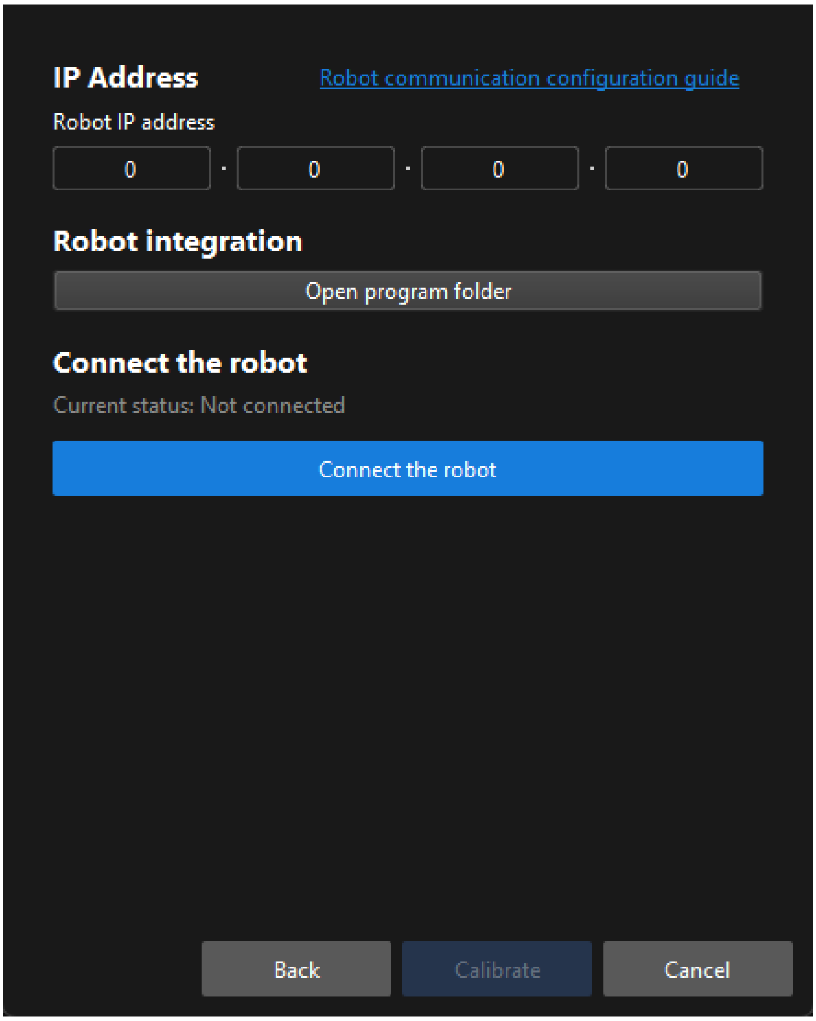


Calibration method and robot control:

Automatic

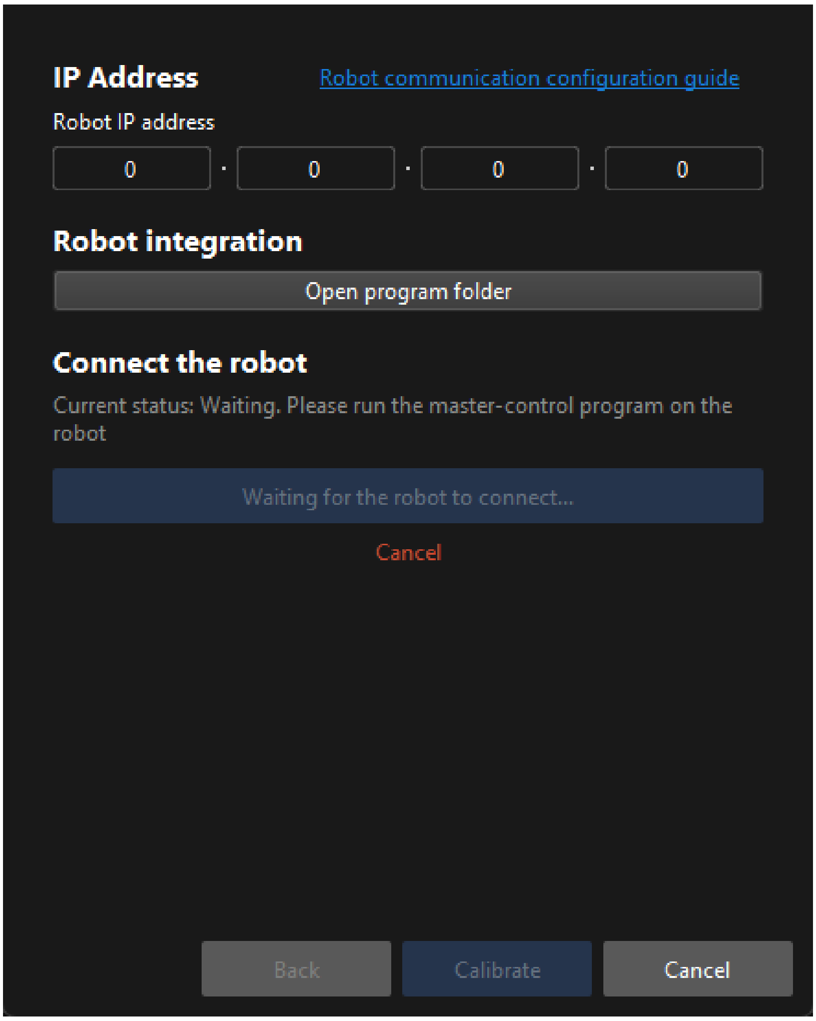
Master-control

6. In the prompted window, set the **Robot IP address** parameter.



7. On the robot teach pendant, **select and run the Master-Control program**. For the instructions of different robots, refer to the section Master-control Communication.

8. Return to Mech-Vision, and click the **[Connect the robot]** button in the **Connect the robot** area. The button will turn into **Waiting for the robot to connect...**



IP Address

Robot communication configuration guide

Robot IP address

0 . 0 . 0 . 0

Robot integration

Open program folder

Connect the robot

Current status: Waiting. Please run the master-control program on the robot

Waiting for the robot to connect...

Cancel

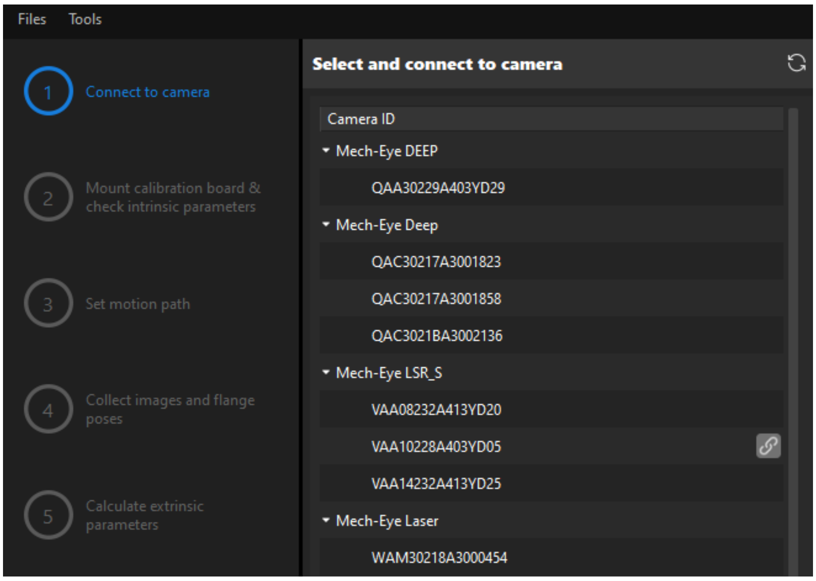
9. Wait until the "Connected" status message is displayed in the **Connect the robot** area, and then click the **[Calibrate]** button.The **Calibration (Eye in Hand)** window will be prompted.

Till now, you have completed the pre-calibration configuration and can start the calibration procedure.

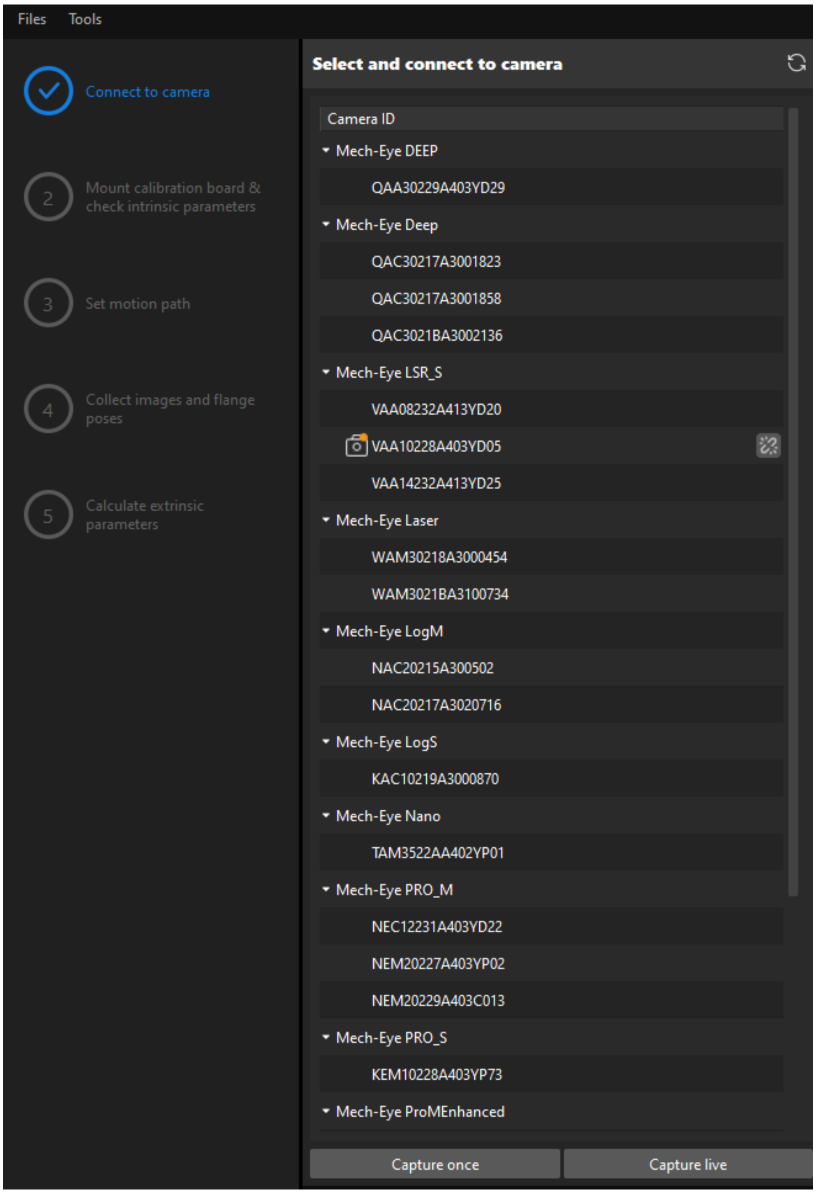
Calibration Procedure

Connect to the Camera

1. In the **Connect to camera** step, select the camera to connect in the Camera ID list, and then click the 🔗 button or double-click the camera entry to connect to it.



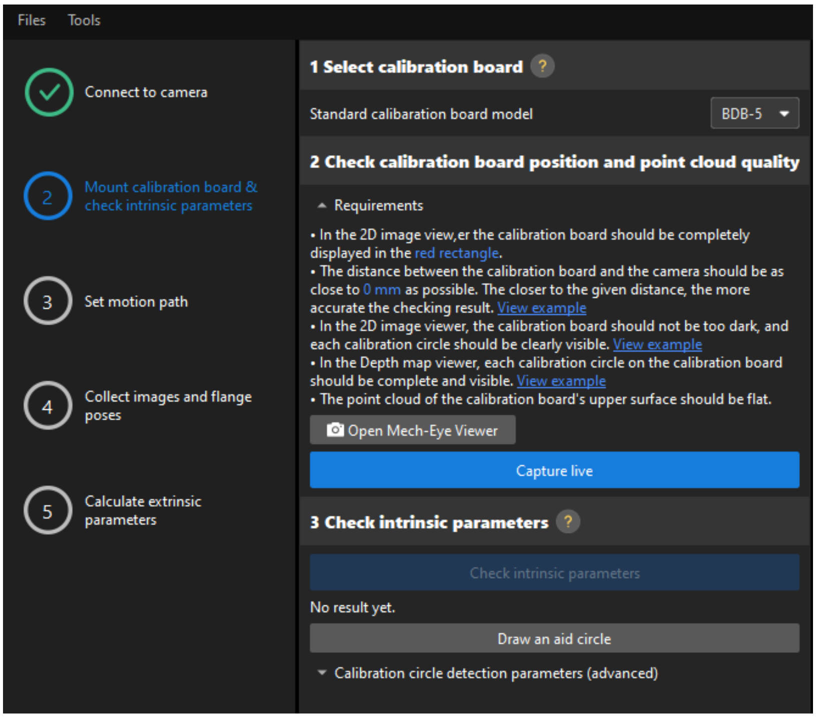
2. After the camera is connected, click the [Capture once] or [Capture live] button.



3. In the **Image viewer** panel, ensure that the camera can capture images normally and click the **[Next]** button on the bottom bar.

Mount Calibration Board & Check Intrinsic Parameters

1. In the **Mount calibration board & check intrinsic parameters** step, select the calibration board model for the **Standard calibration board model** parameter according to its model nameplate in the **1 Select calibration board** area.



Files Tools

Connect to camera

2 Mount calibration board & check intrinsic parameters

3 Set motion path

4 Collect images and flange poses

5 Calculate extrinsic parameters

1 Select calibration board ?

Standard calibration board model: BDB-5

2 Check calibration board position and point cloud quality

Requirements

- In the 2D image view,er the calibration board should be completely displayed in the red rectangle.
- The distance between the calibration board and the camera should be as close to 0 mm as possible. The closer to the given distance, the more accurate the checking result. View example
- In the 2D image viewer, the calibration board should not be too dark, and each calibration circle should be clearly visible. View example
- In the Depth map viewer, each calibration circle on the calibration board should be complete and visible. View example
- The point cloud of the calibration board's upper surface should be flat.

Open Mech-Eye Viewer

Capture live

3 Check intrinsic parameters ?

Check intrinsic parameters

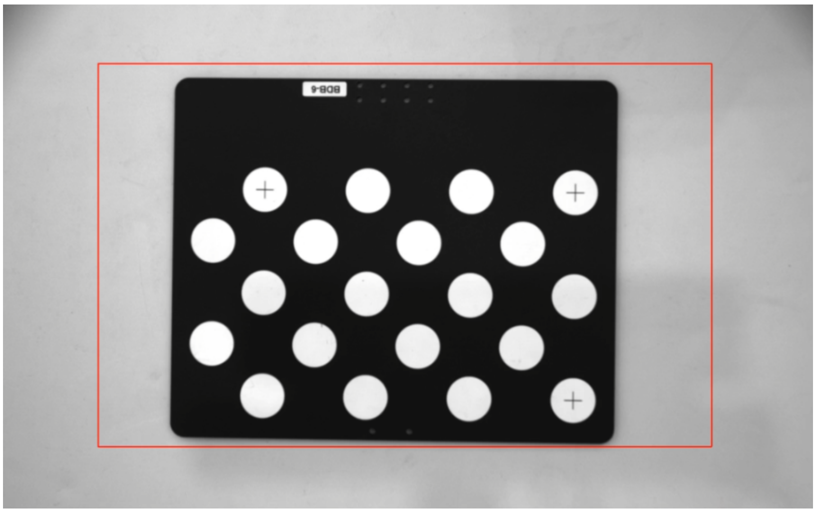
No result yet.

Draw an aid circle

Calibration circle detection parameters (advanced)

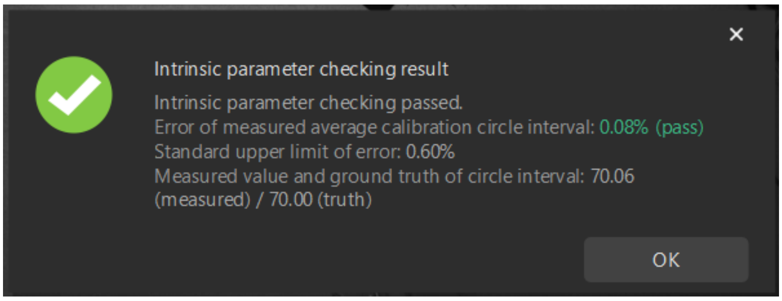
2. In the **2 Check calibration board position and point cloud quality** area, read carefully the requirements on the calibration board position and point cloud quality, and then click the **[Capture live]** button.The **[Capture live]** button will turn into **[Stop capturing and detect position]**.

3. Manually move the calibration board to the proper position, and ensure that the 2D image and depth map of the calibration board meet the requirements, and then click the **[Stop capturing and detect position]** button.



If the captured images do not meet the requirements, click the [**Open Mech-Eye Viewer**] button to open the Mech-Eye Viewer software, adjust the 2D and 3D exposure parameters and re-capture images. Please note that you need to change the **Parameter group** parameter to "calib" first.

4. In the **3 Check intrinsic parameters** area, click the [**Check intrinsic parameters**] button.
5. Confirm the results of the camera intrinsic parameter check.
   - If the camera intrinsic parameter check passes, click the [**OK**] button in the prompted window, and then click the [**Next**] button on the bottom bar.



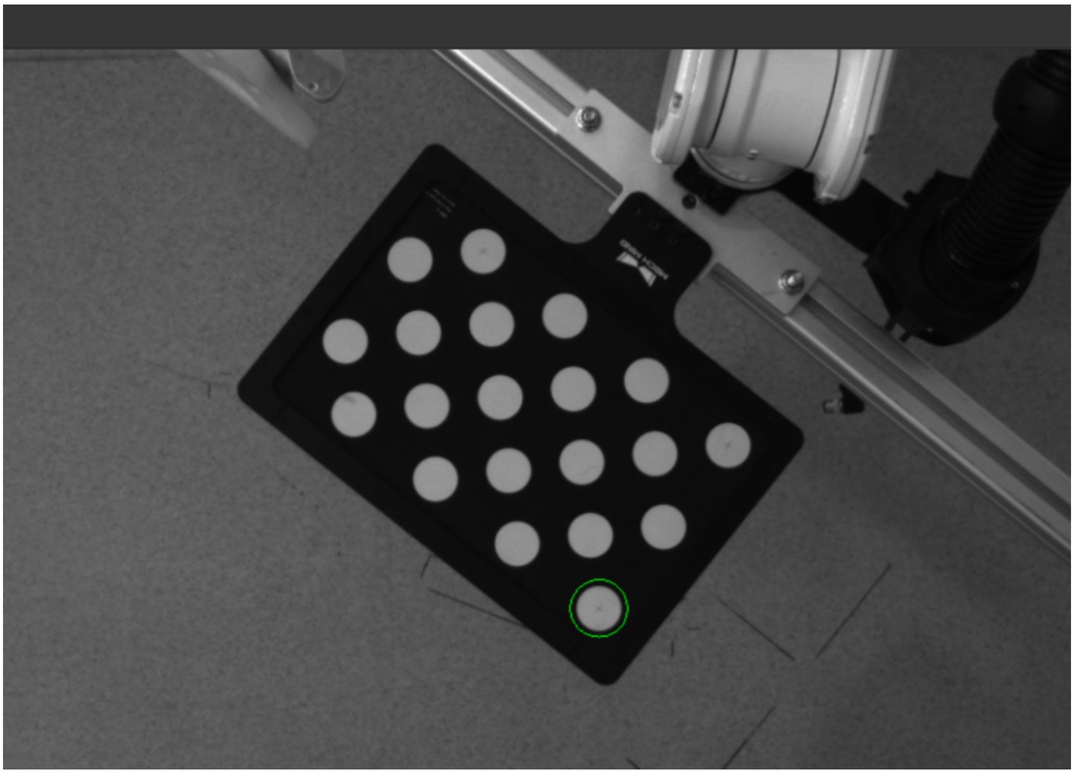
Intrinsic parameter checking result

Intrinsic parameter checking passed.
Error of measured average calibration circle interval: 0.08% (pass)
Standard upper limit of error: 0.60%
Measured value and ground truth of circle interval: 70.06 (measured) / 70.00 (truth)

   - If the camera intrinsic parameter check fails, adjust detection parameters by drawing aid circles or manually adjusting the calibration circle detection parameters, and then click the [**Recheck intrinsic parameters**] button.

Draw an Aid Circle

1. To draw an aid circle, click the [**Draw an aid circle**] button.
2. In the right **Image viewer** panel, right-click the calibration board image and clear the **Fit to window** checkbox, and press the `Ctrl` key and drag the roller to adjust the image to a suitable size.
3. Move the mouse pointer to the cross center point of the calibration circle, press the left mouse button and make the aid circle completely include the calibration circle and then release it.



4. Click the **[Recheck intrinsic parameters]** button, and confirm that the camera intrinsic parameter check passes.

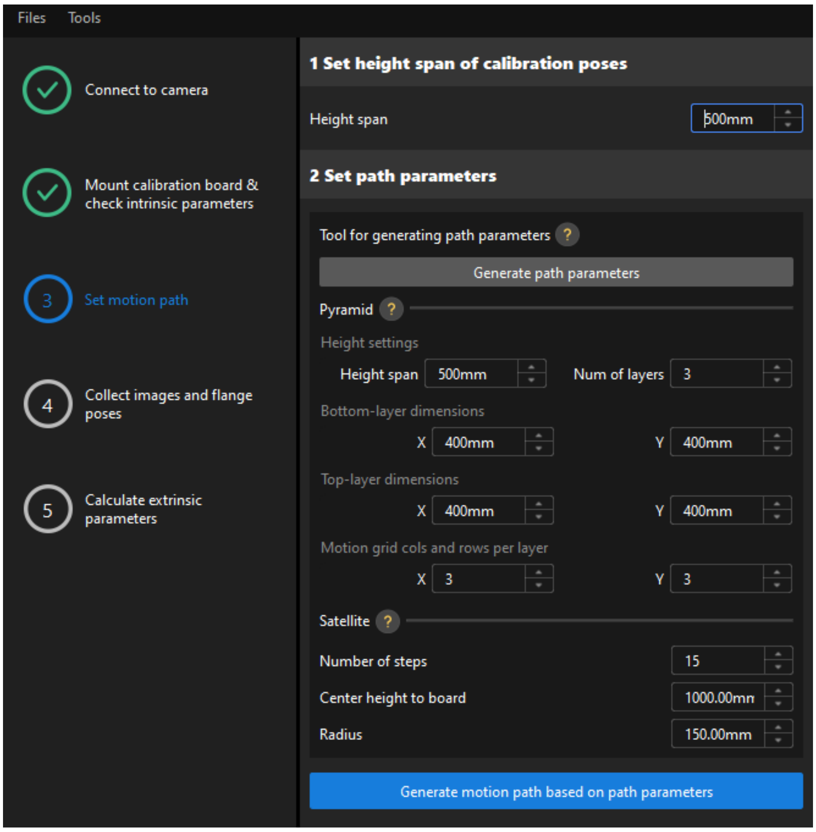
Manually Adjust Detection Parameters

To manually adjust detection parameters, click **[Calibration circle detection parameters (advanced)]**, adjust parameter settings and click the **[Recheck intrinsic parameters]** button to confirm whether the camera intrinsic parameter check passes.

If the calibration circles are still not detected, please adjust the camera settings according to the on-site operating conditions.

Set Motion Path

1. In the **Set motion path** step, set the **Height span** parameter in the **1 Set height span of calibration poses** area.


Files
Tools
Connect to camera
Mount calibration board & check intrinsic parameters
3
Set motion path
4
Collect images and flange poses
5
Calculate extrinsic parameters
Generate path parameters
Generate motion path based on path parameters

1 Set height span of calibration poses

Height span: 500mm

2 Set path parameters

Tool for generating path parameters ?

Pyramid ?

Height settings

Height span: 500mm Num of layers: 3

Bottom-layer dimensions

X: 400mm Y: 400mm

Top-layer dimensions

X: 400mm Y: 400mm

Motion grid cols and rows per layer

X: 3 Y: 3

Satellite ?

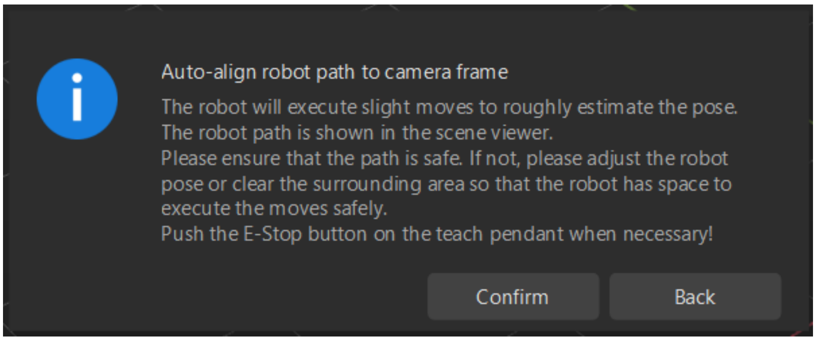
Number of steps: 15

Center height to board: 1000.00mn

Radius: 150.00mm

The **Height span** parameter should be set according to the recommended working distance range of the camera and the size of the robot’s working space.

2. Click the [**Generate path parameters**] button, and click the [**OK**] button in the prompted window.

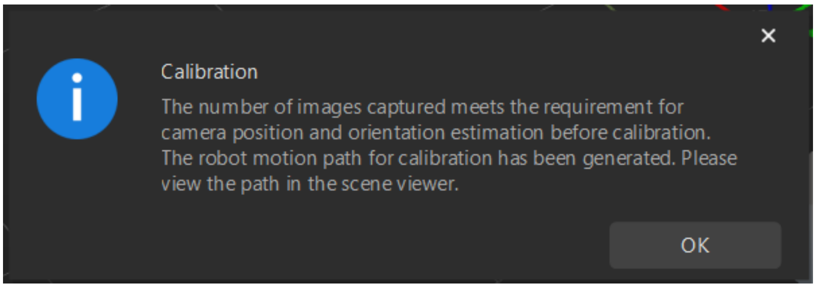


Auto-align robot path to camera frame

The robot will execute slight moves to roughly estimate the pose.
The robot path is shown in the scene viewer.
Please ensure that the path is safe. If not, please adjust the robot pose or clear the surrounding area so that the robot has space to execute the moves safely.
Push the E-Stop button on the teach pendant when necessary!

This operation will slightly move the robot and capture images. The whole process may take 10 to 15 seconds. Please ensure that the motion path is safe. In case of emergency, please tap the emergency stop button on the robot teach pendant to stop the robot immediately.

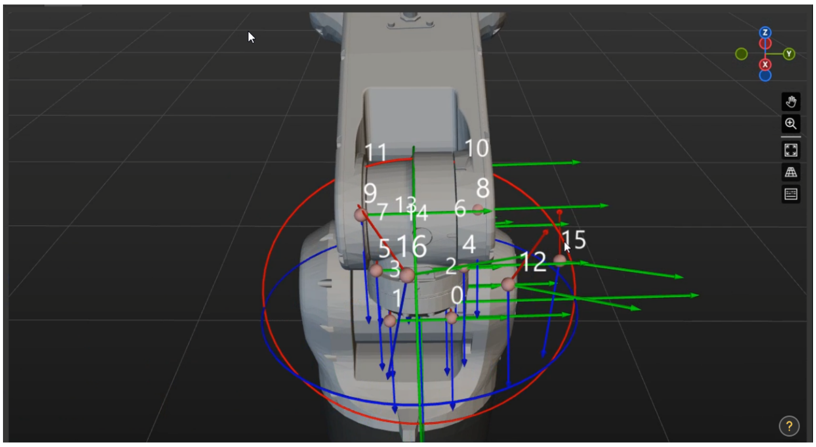
3. In the prompted **Calibrate** window, click the [**OK**] button.



Calibration

The number of images captured meets the requirement for camera position and orientation estimation before calibration. The robot motion path for calibration has been generated. Please view the path in the scene viewer.

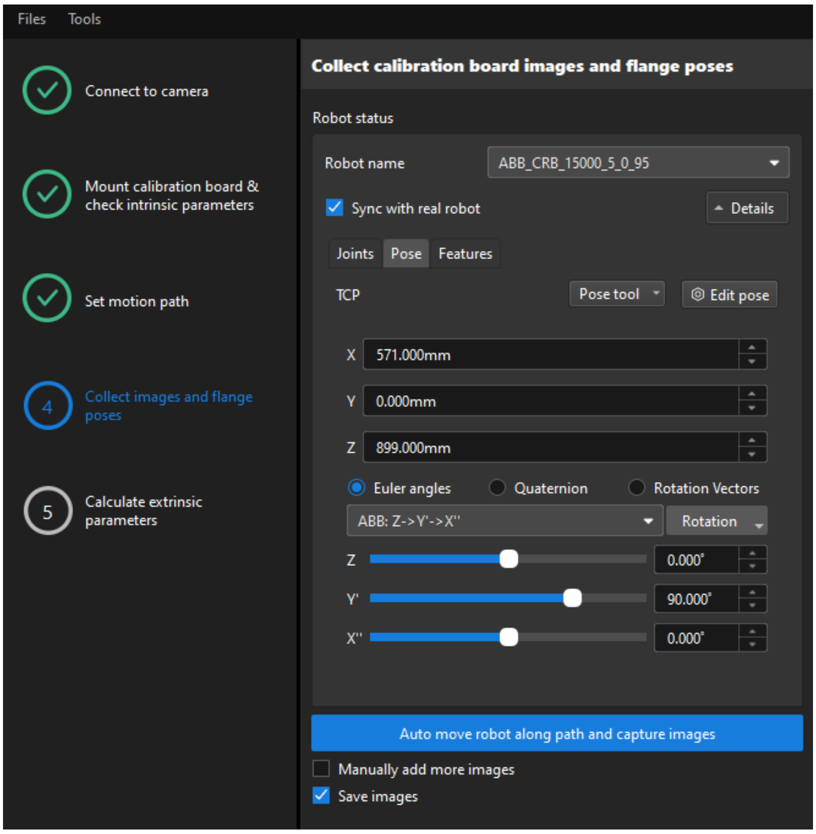
4. Confirm that waypoints are reasonable and will not collide with obstacles in the right scene viewer panel.



5. Click the **[Generate motion path based on path parameters]** button and then click the **[Next]** button on the bottom bar.

Collect Images and Flange Poses

1. In the **Collect images and flange poses** step, select the **Save images** checkbox.


Files Tools

Connect to camera

Mount calibration board & check intrinsic parameters

Set motion path

4 Collect images and flange poses

5 Calculate extrinsic parameters

Collect calibration board images and flange poses

Robot status

Robot name: ABB_CRB_15000_5_0_95

☑ Sync with real robot

Details

Joints | Pose | Features

TCP

Pose tool

Edit pose

X: 571.000mm

Y: 0.000mm

Z: 899.000mm

● Euler angles ○ Quaternion ○ Rotation Vectors

ABB: Z->Y'->X''

Rotation

Z: 0.000°

Y': 90.000°

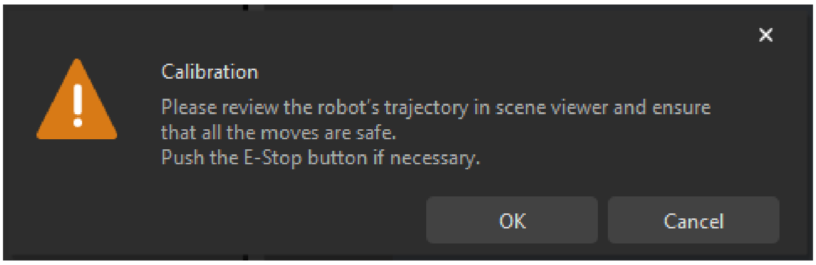
X'': 0.000°

Auto move robot along path and capture images

□ Manually add more images

☑ Save images

2. If the robot uses the Master-Control to communicate with the Vision System, select the **Sync with real robot** checkbox (selected by default, recommended), and click **Details** to confirm that the robot’s JPs and pose data are the same as those on the teach pendant. Clear the checkbox if you do not need to synchronize the data with the real robot.
3. Click the **[Auto move robot along path and capture images]** button, read the safety window carefully, and then click the **[OK]** button.


Calibration

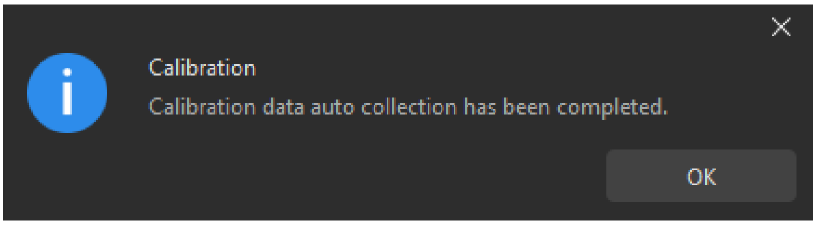
Please review the robot’s trajectory in scene viewer and ensure that all the moves are safe.
Push the E-Stop button if necessary.

OK Cancel

4. Wait until the robot finishes moving along the preset path and the camera finishes capturing images on all waypoints. In the right **Image viewer** panel, all the captured images are displayed.

- Please stay away from the robot working area to keep safe.
- Clicking the Stop Robot button can exit the calibration process. But the robot will not stop until it finishes the current waypoint. In case of emergency, please tap the emergency stop button on the robot teach pendant to stop the robot immediately (the robot needs to be reconnected after tapping the emergency stop button).

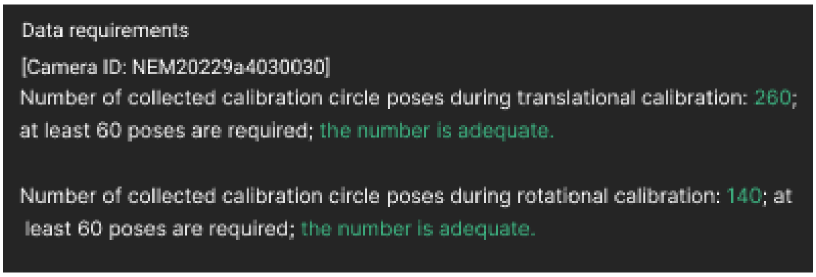
5. After all images are captured, click the **[OK]** button in the prompted window.



Calibration

Calibration data auto collection has been completed.

6. Confirm that the collected calibration data meets the data requirements, and then click the [Next] button on the bottom bar.



Data requirements

[Camera ID: NEM20229a4030030]

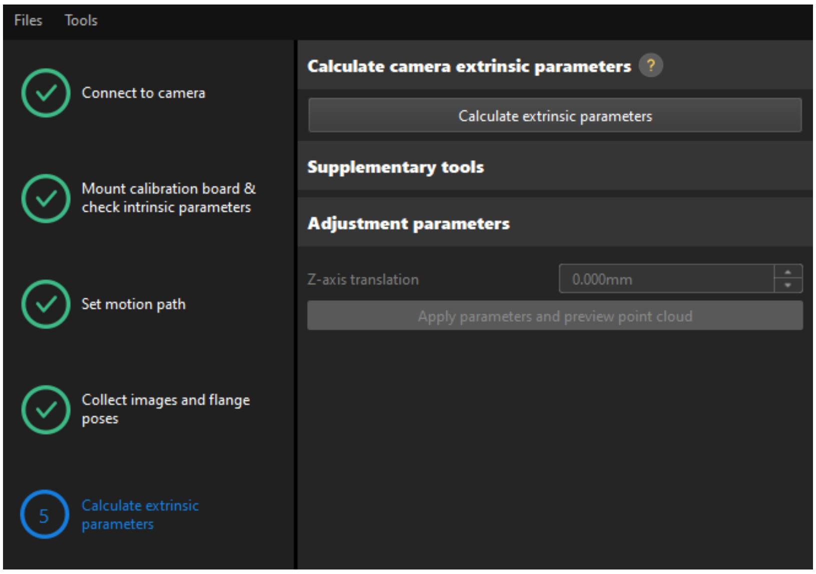
Number of collected calibration circle poses during translational calibration: 260; at least 60 poses are required; the number is adequate.

Number of collected calibration circle poses during rotational calibration: 140; at least 60 poses are required; the number is adequate.

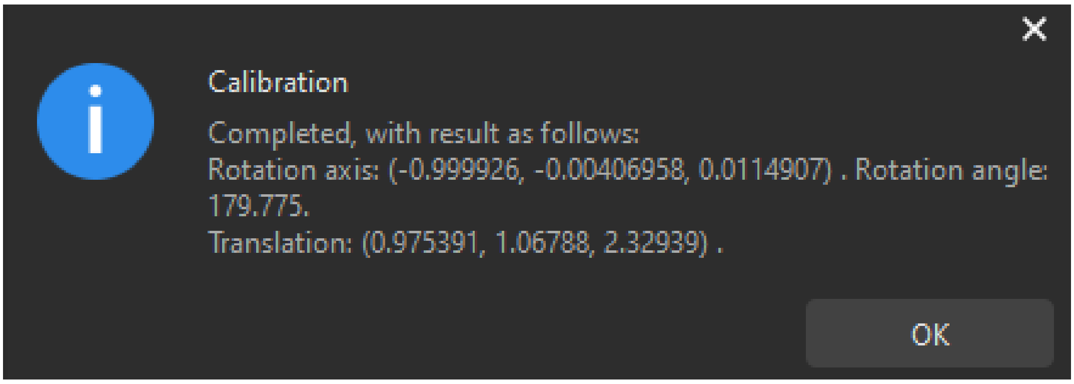
If the requirements are not met, please manually move the robot (through the teach pendant or Mech-Viz), select the **Manually add more images** checkbox, and click the **[Add an image and record flange pose]** button to add the calibration board image and enter the robot flange pose.

Calculate Extrinsic Parameters

1. In the **Calculate extrinsic parameters** step, click the **[Calculate extrinsic parameters]** button.


Files
Tools
Connect to camera
Mount calibration board & check intrinsic parameters
Set motion path
Collect images and flange poses
5
Calculate extrinsic parameters
Calculate camera extrinsic parameters
?
Calculate extrinsic parameters
Supplementary tools
Adjustment parameters
Z-axis translation
0.000mm
Apply parameters and preview point cloud

2. In the prompted window indicating calibration success, click the **OK** button.



Calibration

Completed, with result as follows:
Rotation axis: (-0.999926, -0.00406958, 0.0114907) . Rotation angle: 179.775.
Translation: (0.975391, 1.06788, 2.32939) .

To evaluate the robot absolute accuracy and the accuracy of calculated extrinsic parameters, you can click **[Robot absolute accuracy and repeatability]** and **[Extrinsic parameter accuracy]** in the "Calculate extrinsic parameters" window and use the error analysis tool to perform robot absolute accuracy check and analyze extrinsic parameter error in EIH setup.

3. Calculate the Z-axis translation of the extrinsic parameters.

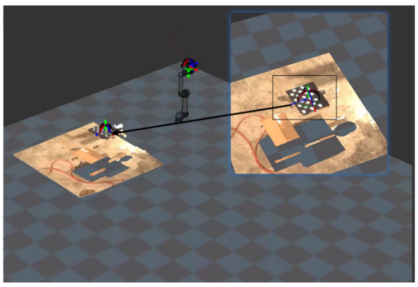
Due to the lack of **degrees of freedom in rotation** in four-axis robots, the captured images during calibration may lack the **images for calibration board rotation**. After calibration, you need to manually adjust the Z-axis value of the extrinsic parameters.

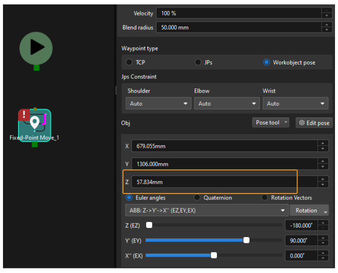
a. Mount a tip on the proper position of the robot gripper, and set the tool center point (TCP).

b. Control the robot to let its end tip touch the cross center point of a calibration circle on the calibration board, and record the tool pose (X0, Y0, Z0) in the teach pendant.

c. Move the robot to the image-capturing position to capture the image of the calibration board, click the **[Apply parameters and preview point cloud]** button in the **Adjust parameters** area to send the point cloud of the calibration board to Mech-Viz.

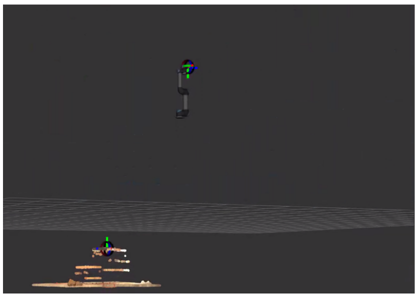
d. In Mech-Viz, add a "Fixed-Point Move" Step, and enter the tool pose (X0, Y0, Z0) in the previous step as the target waypoint. Adjust the Z-axis value of the tool pose to make sure that the target waypoint has moved to the surface of the calibration board's point cloud, and record the new Z-axis value (Z1). You can get the Z-axis value error ΔZ by subtracting Z1 from Z0.

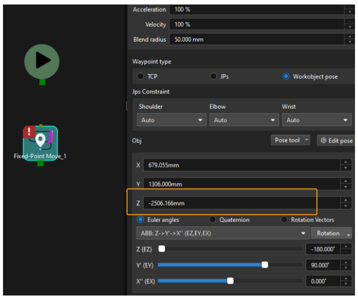




Velocity 100 %
Blend radius 50.000 mm
Waypoint type
○ TCP ○ JPs ● Workobject pose
Jps Constraint
Shoulder Auto
Elbow Auto
Wrist Auto
Obj
X 679.055mm
Y 1306.000mm
Z 57.834mm
● Euler angles ○ Quaternion ○ Rotation Vectors
ABB: Z->Y'->X'' (EZ,EY,EX)
Z (EZ) -180.000°
Y' (EY) 90.000°
X'' (EX) 0.000°

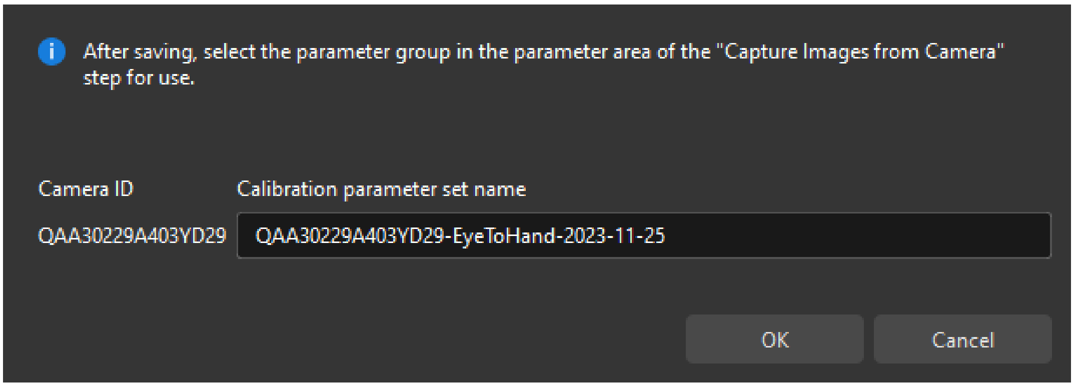
4. Back to the **Adjust parameters** area of the **Calculate extrinsic parameters** steps, manually change the **Z-axis translation** parameter to the error ΔZ, and then click the **[Apply parameters and preview point cloud]** button.
5. In the 3D Simulation Area of Mech-Viz, confirm that the pose (X0, Y0, Z0) is on the surface of the calibration board's point cloud.





Acceleration 100 %
Velocity 100 %
Blend radius 50.000 mm
Waypoint type
○ TCP ○ JPs ● Workobject pose
Jps Constraint
Shoulder Auto
Elbow Auto
Wrist Auto
Obj
X 679.055mm
Y 1306.000mm
Z -2506.166mm
● Euler angles ○ Quaternion ○ Rotation Vectors
ABB: Z->Y'->X'' (EZ,EY,EX)
Z (EZ) -180.000°
Y' (EY) 90.000°
X'' (EX) 0.000°

6. Click the **[Save]** button on the bottom bar. In the prompted **Save Calibration Files** dialog box, click the **[OK]** button. The camera calibration result will be automatically saved in the "calibration" directory of the project.



After saving, select the parameter group in the parameter area of the "Capture Images from Camera" step for use.

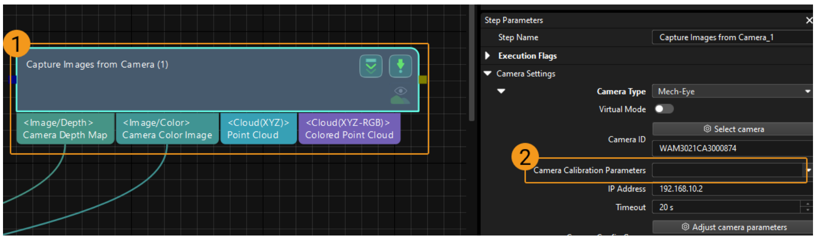
Camera ID: QAA30229A403YD29
Calibration parameter set name: QAA30229A403YD29-EyeToHand-2023-11-25

Till now, the calibration process is completed.

Change the Extrinsic Parameter File

If the camera's extrinsic parameter file has been used by a Mech-Vision project, you need to change the extrinsic parameter file used by the Mech-Vision project to a new one after you finish the hand-eye calibration.

1. Select the **Capture Images from Camera** Step.



Step Parameters

Step Name: Capture Images from Camera_1

Execution Flags

Camera Settings

Camera Type: Mech-Eye

Virtual Mode

Camera ID: WAM3021CA3000874

Camera Calibration Parameters

IP Address: 192.168.10.2

Timeout: 20 s

2. In the **Step Parameters** panel, click ▾ of the **Camera Calibration Parameters** parameter, and select the newly calibrated extrinsic parameter file.

Calibration-Related Status Codes

If the robot uses the Standard Interface to communicate with the Vision System during hand-eye calibration, the Vision System will return status codes in response data to indicate the execution status of the commands. The status codes indicate both the normal execution results and execution failures. You can perform further troubleshooting according to the status codes.

Calibration Error Codes

Error code	Description
7001	Calibration: Parameter error
7002	Calibration: Calibration flange pose not provided by Mech-Vision
7003	Calibration: Calibration joint positions not provided by Mech-Vision
7004	Calibration: Robot failed to reach calibration point

Calibration Normal Status Codes

Normal Status Code	Description
7100	Calibration: Robot reaches calibration point successfully
7101	Calibration: Pose received from Mech-Vision successfully

4.5.3.3. Manual Calibration in the Eye-in-Hand Setup (Six-Axis Robots - Using Multiple Random Calibration Board Poses Method)

This how-to guide introduces how to complete the manual calibration for six-axis robots using the multiple random calibration board poses method in the eye-in-hand (EIH) setup.

Preparation before Calibration

Before hand-eye calibration, you need to finish the following preparations:

- Construct the Mech-Mind Vision System.
- Prepare the materials required for calibration.
- Adjust the effect of calibration board’s point cloud.

Construct the Vision System

Construct the Mech-Mind Vision System by referring to the section Vision System Hardware Setup.

You need to use Mech-Eye Viewer, Mech-Vision and Mech-Viz during hand-eye calibration. Please ensure that they have been installed and are running the latest versions.

Prepare the Materials Required for Calibration

The manual calibration in the EIH setup needs to use the calibration board.

Please prepare the calibration board according to the following requirements:

- Ensure that the circles of the calibration board are clearly visible and without obvious scratches, and the board does not suffer from deformations.
- In the EIH setup, please place the calibration board in the center of the working plane, where target objects are to be placed.

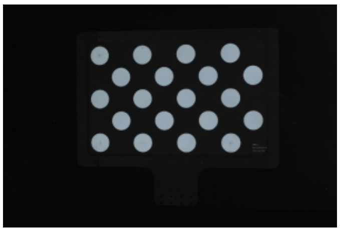
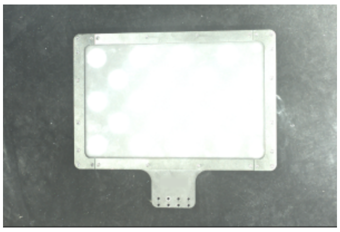
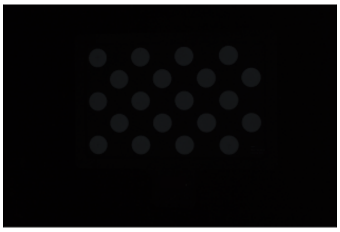
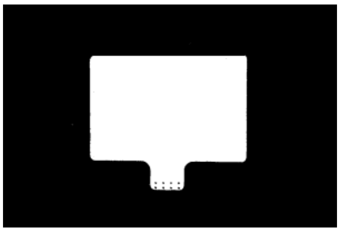
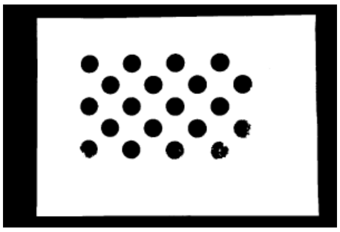
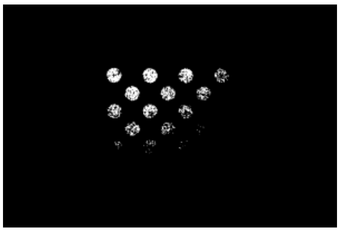
In addition, before calibration, move the robot to the starting point for calibration.

Adjust the Effect of Calibration Board's Point Cloud

1. Open Mech-Eye Viewer to adjust camera parameters.
2. Adjust the 2D parameters to make sure that the 2D image of the calibration board is clear and neither overexposed nor underexposed.
3. Adjust 3D parameters to make sure that the point clouds of the circles on the calibration board are complete and have clear contours. It is recommended to set **Surface Smoothing** and **Outlier Removal** to **Normal** in the **Point Cloud Processing** section to reduce the point cloud fluctuation range.

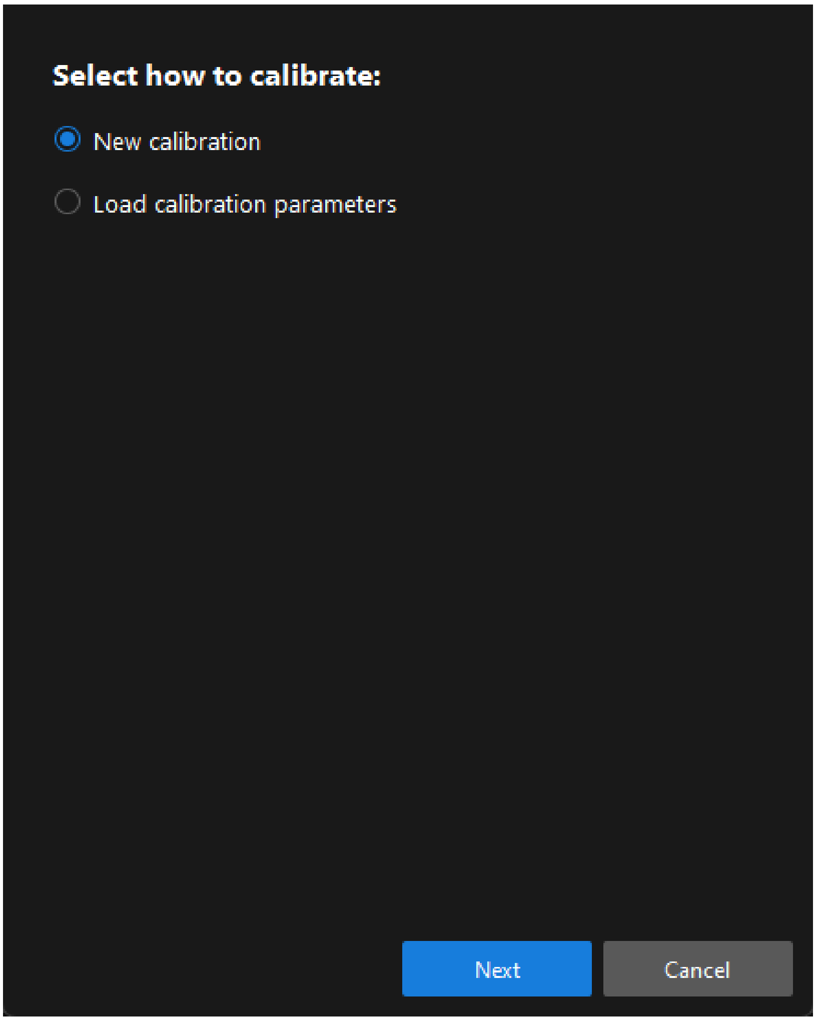
   If the on-site ambient lights are not ideal and affect the quality of 2D images and point clouds, you can use shading or supplemental light to improve the lighting conditions.

4. Make sure that the images and point cloud of the calibration board are up to standard after completing preceding steps.

	Normal	Overexposed	Underexposed
2D image			
Point cloud			

Pre-calibration Configuration

1. Open Mech-Vision, and click the **[Camera Calibration (Standard)]** button in the toolbar.The **Configuration before Calibration** window will be prompted.
2. In the **Select how to calibrate** window, select the **New calibration** radio button, and then click the **[Next]** button.

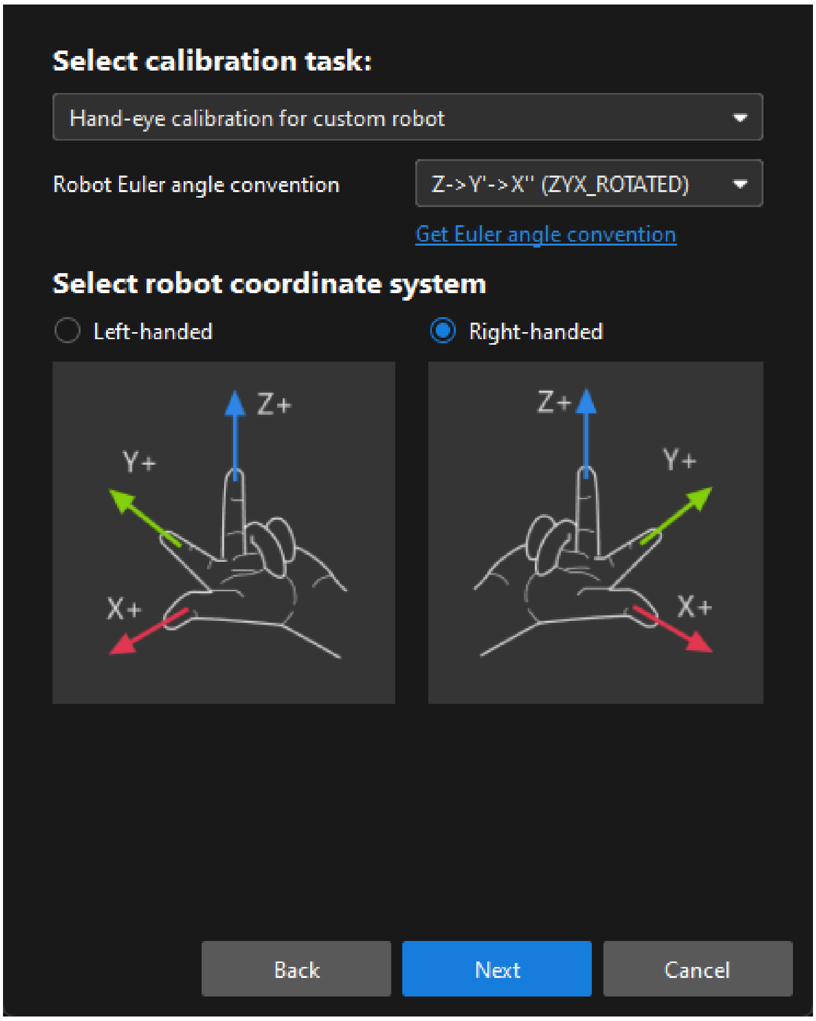


Select how to calibrate:

● New calibration

○ Load calibration parameters

3. In the **Select calibration task** window, select **Hand-eye calibration for custom robot** from the drop-down list box, specify the **Robot Euler angle convention** parameter, select the robot coordinate system type, and then click the **[Next]** button.



Select calibration task:

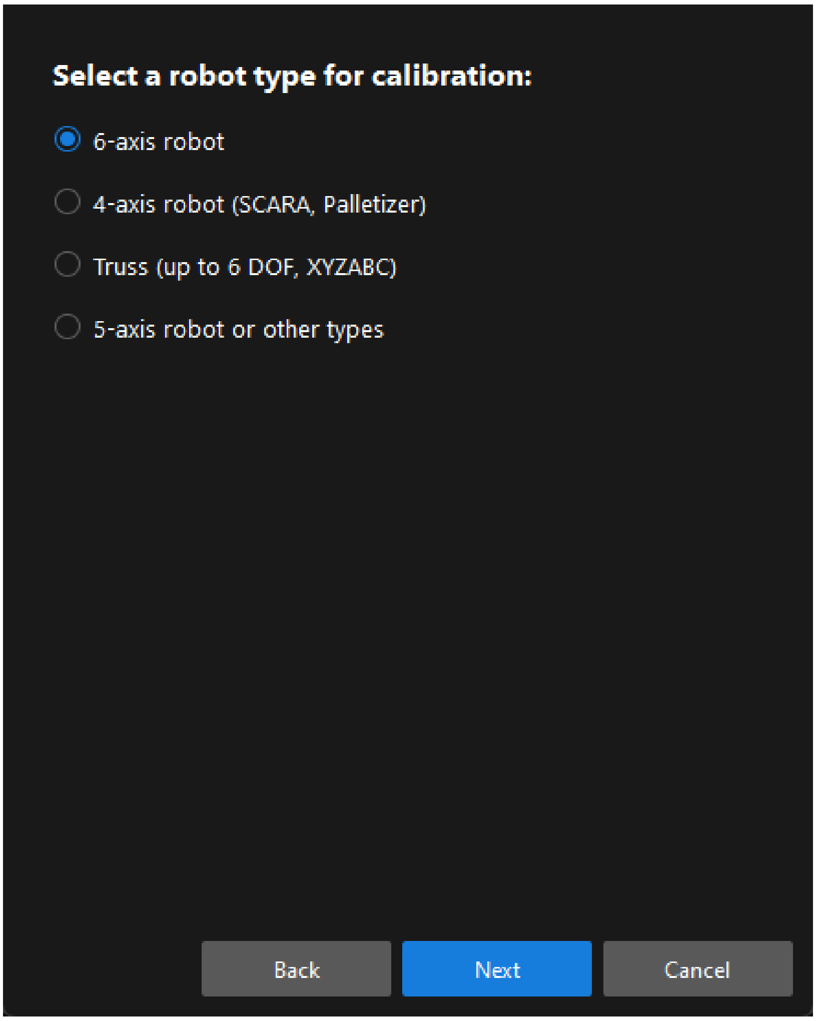
Hand-eye calibration for custom robot

Robot Euler angle convention: Z->Y'->X'' (ZYX_ROTATED)

Select robot coordinate system

○ Left-handed ● Right-handed

4. In the **Select a robot type for calibration** window, select the **6-axis robot** radio button, and then click the **[Next]** button.



Select a robot type for calibration:

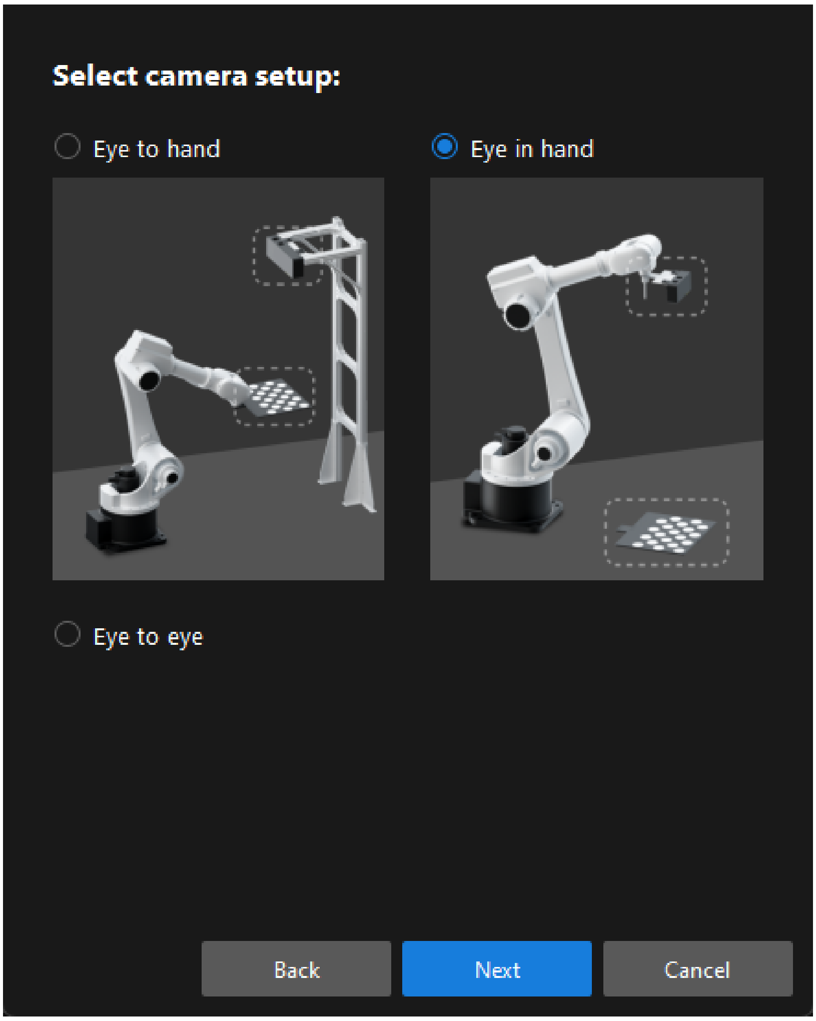
● 6-axis robot

○ 4-axis robot (SCARA, Palletizer)

○ Truss (up to 6 DOF, XYZABC)

○ 5-axis robot or other types

5. In the **Select camera setup** window, select the **Eye in hand** radio button, and then click the **[Next]** button.



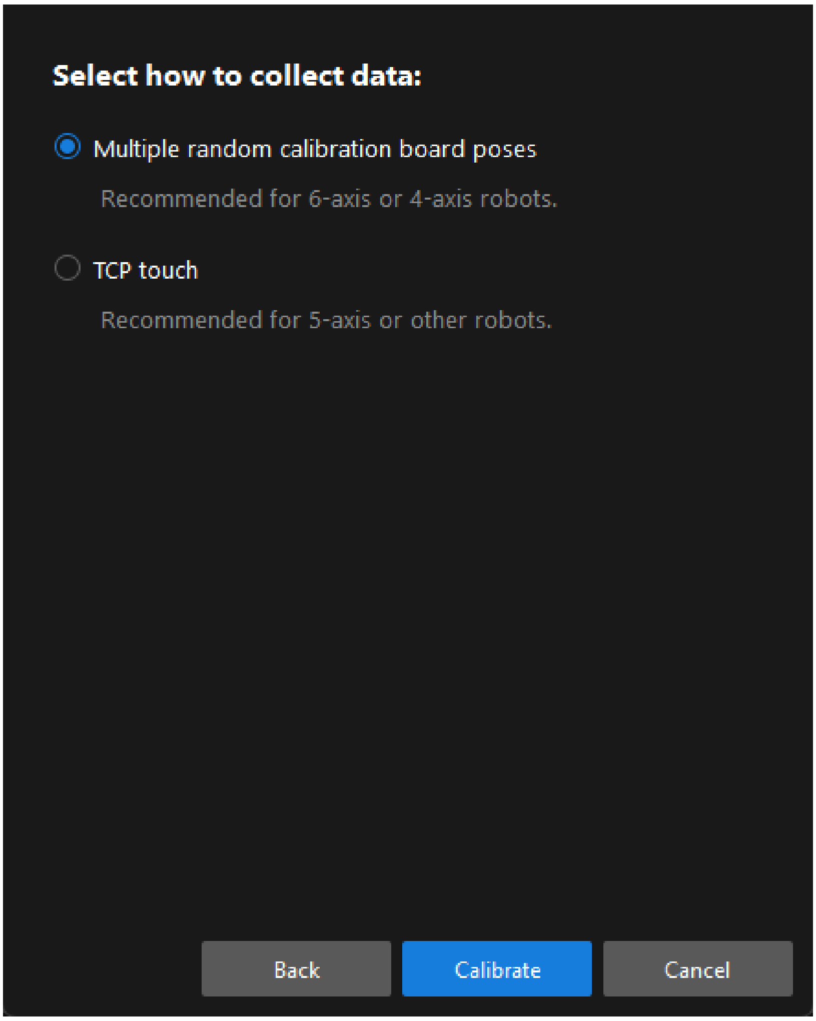
Select camera setup:

○ Eye to hand

● Eye in hand

○ Eye to eye

6. In the **Select how to collect data** window, select the **Multiple random calibration board poses**, and then click the **[Calibrate]** button.The **Calibration (Eye in Hand)** window will be prompted.



Select how to collect data:

● Multiple random calibration board poses

Recommended for 6-axis or 4-axis robots.

○ TCP touch

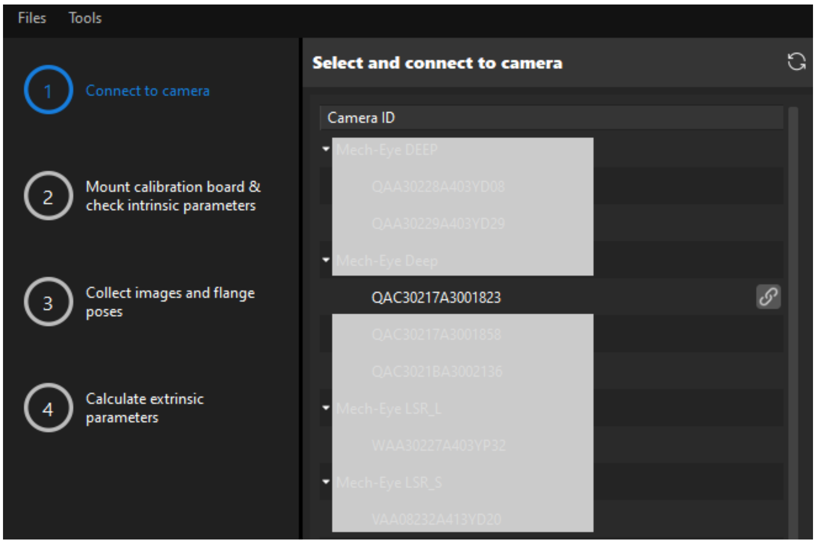
Recommended for 5-axis or other robots.

Till now, you have completed the pre-calibration configuration and can start the calibration procedure.

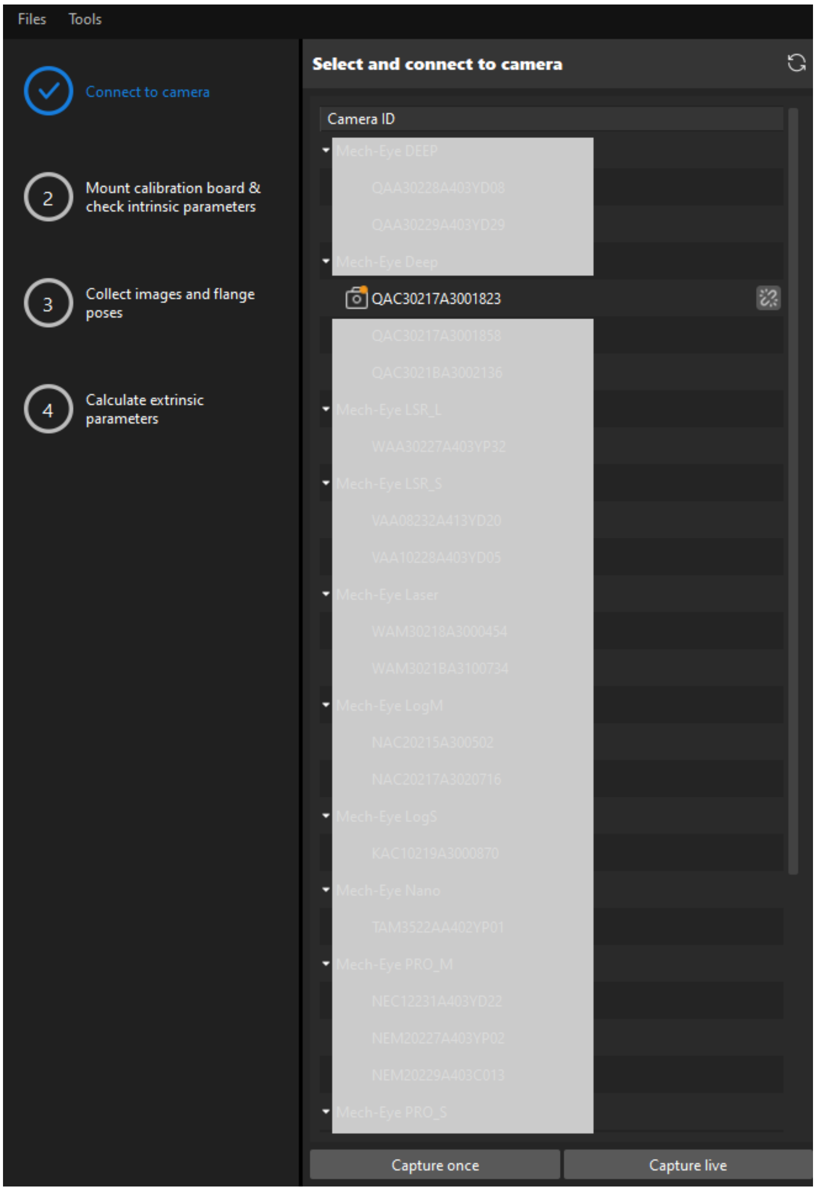
Calibration Procedure

Connect to the Camera

1. In the **Connect to camera** step, select the camera to connect in the Camera ID list, and then click the 🔗 button or double-click the camera entry to connect to it.


Files
Tools
1 Connect to camera
2 Mount calibration board & check intrinsic parameters
3 Collect images and flange poses
4 Calculate extrinsic parameters
Select and connect to camera
Camera ID
[redacted]
QAC30217A3001823
[redacted]

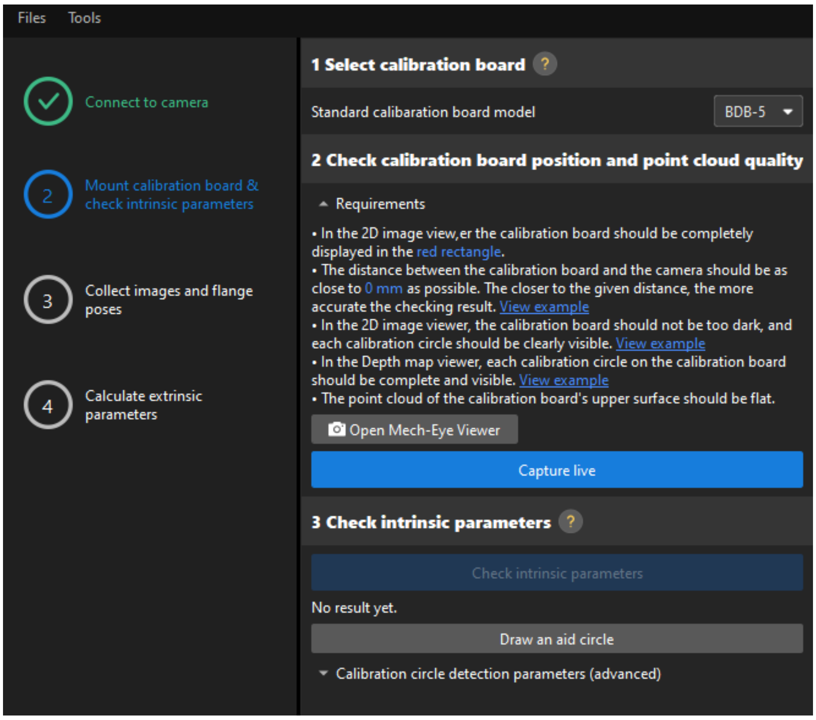
2. After the camera is connected, click the [Capture once] or [Capture live] button.


Files
Tools
Connect to camera
2 Mount calibration board & check intrinsic parameters
3 Collect images and flange poses
4 Calculate extrinsic parameters
Select and connect to camera
Camera ID
[redacted]
QAC30217A3001823
[redacted]
Capture once
Capture live

3. In the **Image viewer** panel, ensure that the camera can capture images normally and click the **[Next]** button on the bottom bar.

Mount Calibration Board & Check Intrinsic Parameters

1. In the **Mount calibration board & check intrinsic parameters** step, select the calibration board model for the **Standard calibration board model** parameter according to its model nameplate in the **1 Select calibration board** area.



Files
Tools
Connect to camera
2
Mount calibration board & check intrinsic parameters
3
Collect images and flange poses
4
Calculate extrinsic parameters

1 Select calibration board ?

Standard calibaration board model: BDB-5

2 Check calibration board position and point cloud quality

Requirements

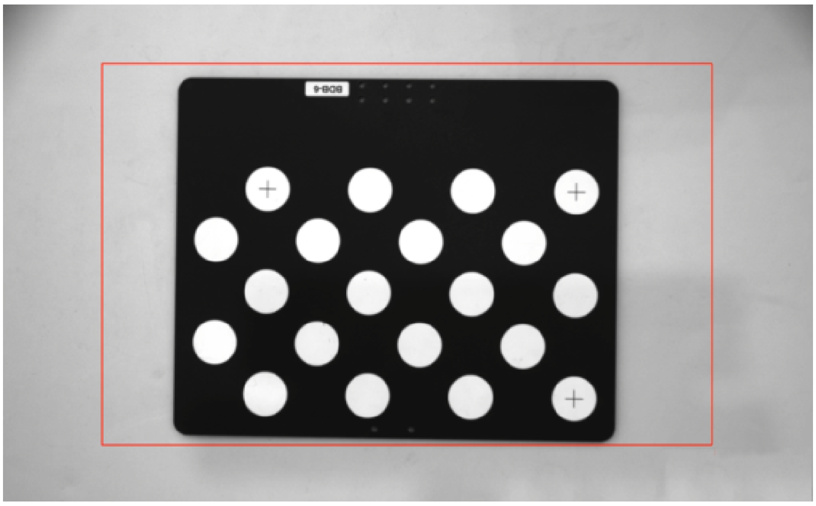
- In the 2D image view,er the calibration board should be completely displayed in the red rectangle.
- The distance between the calibration board and the camera should be as close to 0 mm as possible. The closer to the given distance, the more accurate the checking result. View example
- In the 2D image viewer, the calibration board should not be too dark, and each calibration circle should be clearly visible. View example
- In the Depth map viewer, each calibration circle on the calibration board should be complete and visible. View example
- The point cloud of the calibration board's upper surface should be flat.

3 Check intrinsic parameters ?

No result yet.

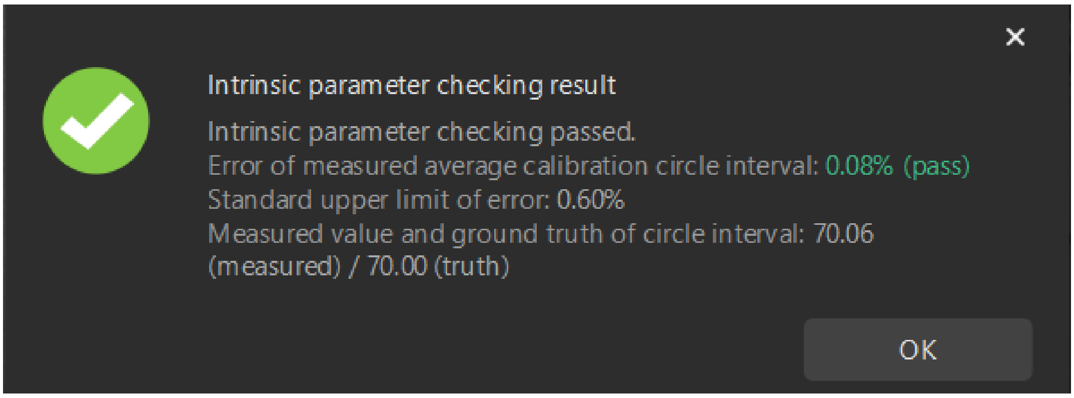
2. In the **2 Check calibration board position and point cloud quality** area, read carefully the requirements on the calibration board position and point cloud quality, and then click the **[Capture live]** button.The **[Capture live]** button will turn into **[Stop capturing and detect position]**.

3. Manually move the calibration board to the proper position (in the red rectangle), and ensure that the 2D image and depth map of the calibration board meet the requirements, and then click the **[Stop capturing and detect position]** button.



> If the captured images do not meet the requirements, click the [**Open Mech-Eye Viewer**] button to open the Mech-Eye Viewer software, adjust the 2D and 3D exposure parameters and re-capture images. Please note that you need to change the **Parameter group** parameter to “calib” first.

4. In the **3 Check intrinsic parameters** area, click the [**Check intrinsic parameters**] button.
5. Confirm the results of the camera intrinsic parameter check.
   - If the camera intrinsic parameter check passes, click the [**OK**] button in the prompted window, and then click the [**Next**] button on the bottom bar.



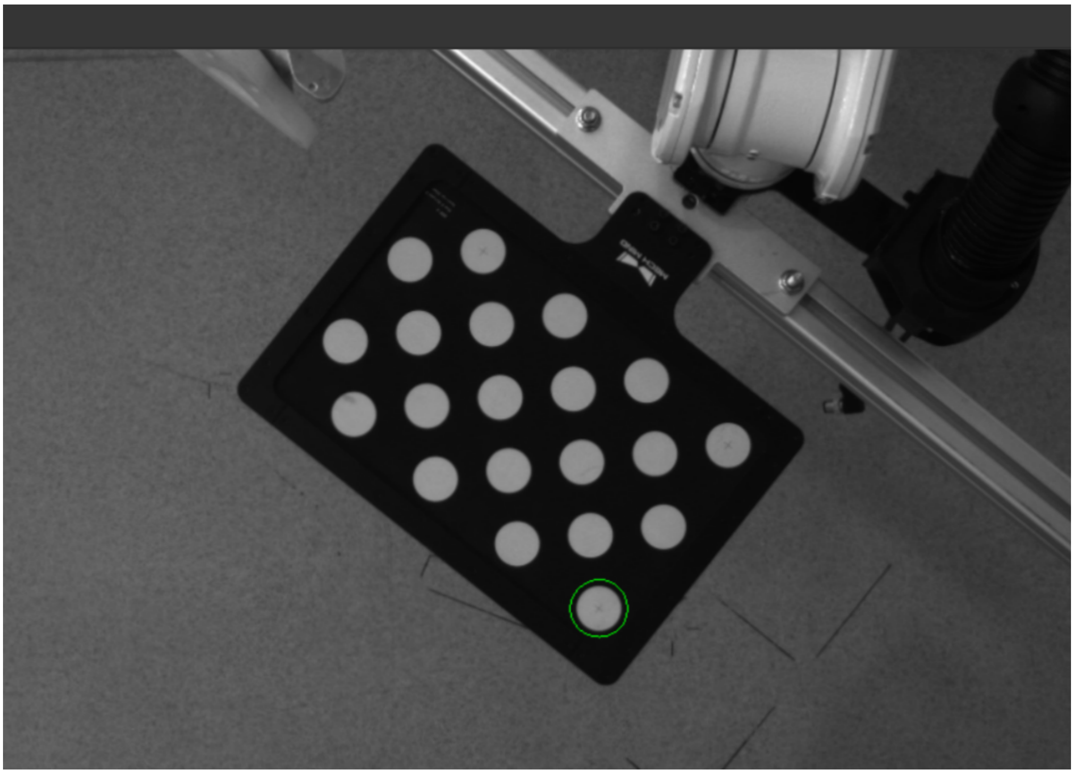
Intrinsic parameter checking result

Intrinsic parameter checking passed.
Error of measured average calibration circle interval: 0.08% (pass)
Standard upper limit of error: 0.60%
Measured value and ground truth of circle interval: 70.06 (measured) / 70.00 (truth)

   - If the camera intrinsic parameter check fails, adjust detection parameters by drawing aid circles or manually adjusting the calibration circle detection parameters, and then click the [**Recheck intrinsic parameters**] button.

Draw an Aid Circle

1. To draw an aid circle, click the [**Draw an aid circle**] button.
2. In the right **Image viewer** panel, right-click the calibration board image and clear the **Fit to window** checkbox, and press the `Ctrl` key and drag the roller to adjust the image to a suitable size.
3. Move the mouse pointer to the cross center point of the calibration circle, press the left mouse button and make the aid circle completely include the calibration circle and then release it.



4. Click the [**Recheck intrinsic parameters**] button, and confirm that the camera intrinsic parameter check passes.

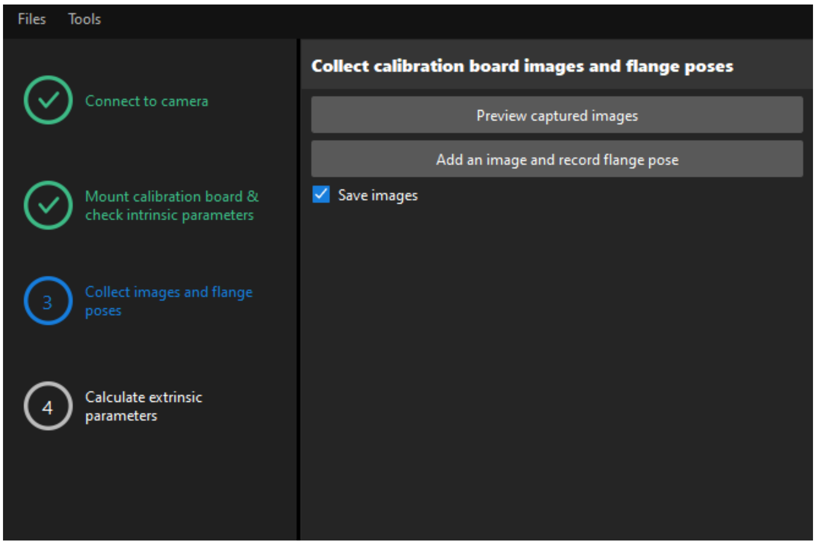
Manually Adjust Detection Parameters

To manually adjust detection parameters, click [**Calibration circle detection parameters (advanced)**], adjust parameter settings and click the [**Recheck intrinsic parameters**] button to confirm whether the camera intrinsic parameter check passes.

If the calibration circles are still not detected, please adjust the camera settings according to the on-site operating conditions.

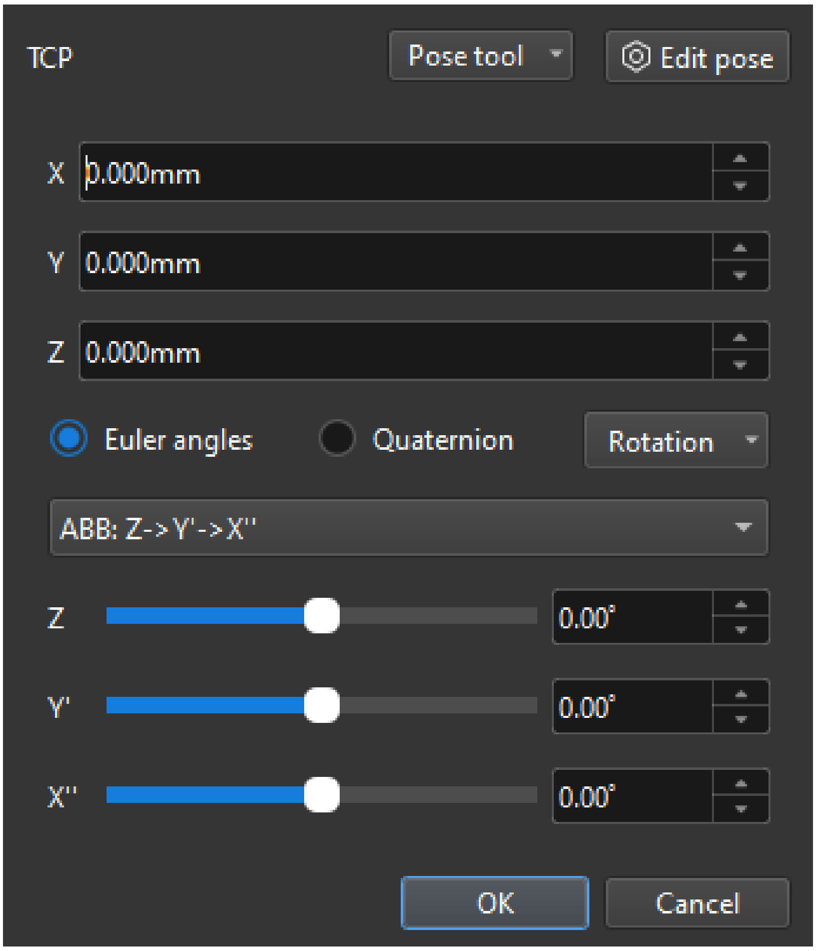
Collect Images and Flange Poses

1. In the **Collect images and flange poses** step, select the [**Save images**] checkbox (selected by default, recommended). If you do not need to save images of calibration boards, clear this checkbox.


Files
Tools
Connect to camera
Mount calibration board & check intrinsic parameters
3
Collect images and flange poses
4
Calculate extrinsic parameters
Collect calibration board images and flange poses
Preview captured images
Add an image and record flange pose
☑ Save images

2. Control the robot to move different calibration points and then click the [**Add an image and record flange pose**] button.

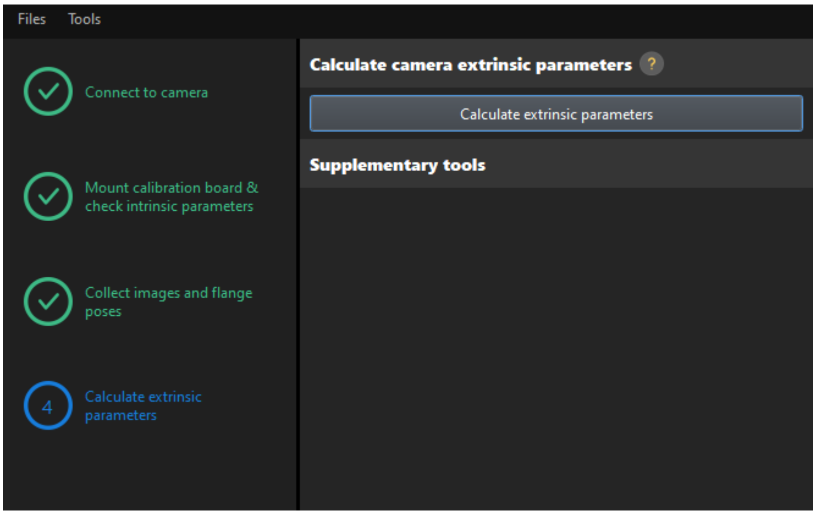
3. In the prompted window, enter the robot flange pose.



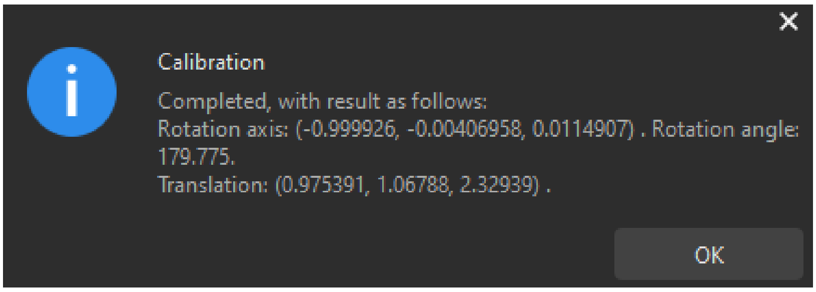
4. Confirm that the number of the currently recognized calibration circles meets the requirements, and then click the [**Next**] button on the bottom bar.

Calculate Extrinsic Parameters

1. In the **Calculate extrinsic parameters** step, click the **[Calculate extrinsic parameters]** button.


Files
Tools
Connect to camera
Mount calibration board & check intrinsic parameters
Collect images and flange poses
4
Calculate extrinsic parameters
Calculate camera extrinsic parameters
?
Calculate extrinsic parameters
Supplementary tools

2. In the prompted window indicating calibration success, click the **[OK]** button.

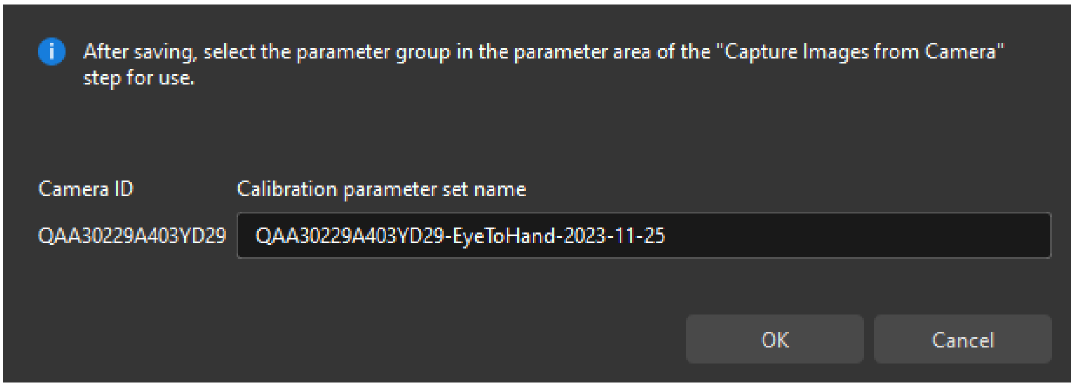


Calibration

Completed, with result as follows:
Rotation axis: (-0.999926, -0.00406958, 0.0114907) . Rotation angle: 179.775.
Translation: (0.975391, 1.06788, 2.32939) .

- If you want to enhance the calibration accuracy, please refer to the section Calibration Result Check and Analysis.
- To evaluate the robot absolute accuracy and the accuracy of calculated extrinsic parameters, you can click **[Robot absolute accuracy and repeatability]** and **[Extrinsic parameter accuracy]** in the "Calculate extrinsic parameters" window and use the error analysis tool to perform robot absolute accuracy check and analyze extrinsic parameter error in EIH setup.

3. Click the **[Save]** button on the bottom bar. In the prompted **Save Calibration Files** dialog box, click the **[OK]** button. The camera calibration result will be automatically saved in the "calibration" directory of the project.



After saving, select the parameter group in the parameter area of the "Capture Images from Camera" step for use.

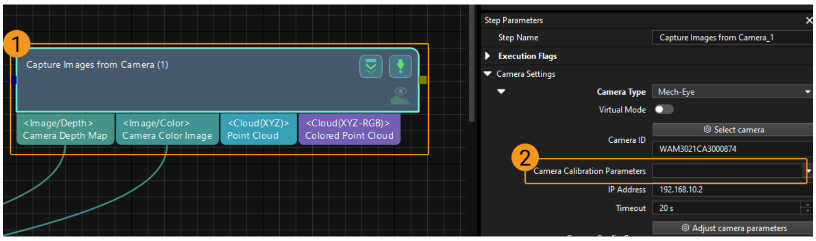
Camera ID	Calibration parameter set name
QAA30229A403YD29	QAA30229A403YD29-EyeToHand-2023-11-25

Till now, the calibration process is completed.

Change the Extrinsic Parameter File

If the camera's extrinsic parameter file has been used by a Mech-Vision project, you need to change the extrinsic parameter file used by the Mech-Vision project to a new one after you finish the hand-eye calibration.

1. Select the **Capture Images from Camera** Step.



Step Parameters
Step Name: Capture Images from Camera_1
Camera Type: Mech-Eye
Camera ID: WAM3021CA3000874
Camera Calibration Parameters:
IP Address: 192.168.10.2
Timeout: 20 s

1
Capture Images from Camera (1)
<Image/Depth> Camera Depth Map
<Image/Color> Camera Color Image
<Cloud(XYZ)> Point Cloud
<Cloud(XYZ-RGB)> Colored Point Cloud
Execution Flags
Camera Settings
Virtual Mode
Select camera
2
Adjust camera parameters

2. In the **Step Parameters** panel, click ▼ of the **Camera Calibration Parameters** parameter, and select the newly calibrated extrinsic parameter file.

4.5.3.4. Manual Calibration in the Eye-in-Hand Setup (Four-Axis Robots - Using Multiple Random Calibration Board Poses Method)

This how-to guide introduces how to complete the manual calibration for four-axis robots using the multiple random calibration board poses method in the eye-in-hand (EIH) setup.

Preparation before Calibration

Before hand-eye calibration, you need to finish the following preparations:

- Construct the Mech-Mind Vision System.
- Prepare the materials required for calibration.
- Adjust the effect of calibration board's point cloud.

Construct the Vision System

Construct the Mech-Mind Vision System by referring to the section Vision System Hardware Setup.

You need to use Mech-Eye Viewer, Mech-Vision and Mech-Viz during hand-eye calibration. Please ensure that they have been installed and are running the latest versions.

Prepare the Materials Required for Calibration

The manual calibration in the EIH setup needs to use the calibration board.

Please prepare the calibration board according to the following requirements:

- Ensure that the circles of the calibration board are clearly visible and without obvious scratches, and the board does not suffer from deformations.
- In the EIH setup, please place the calibration board in the center of the working plane, where target objects are to be placed.

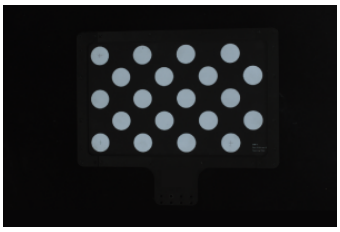
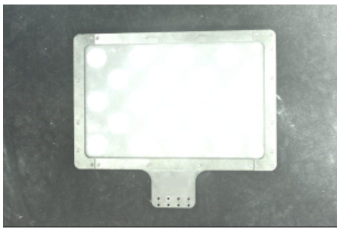
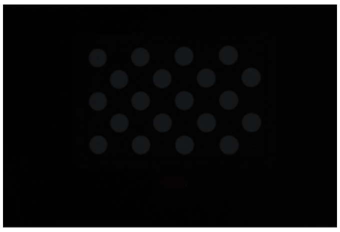
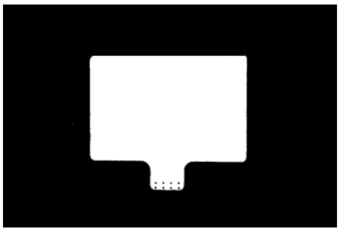
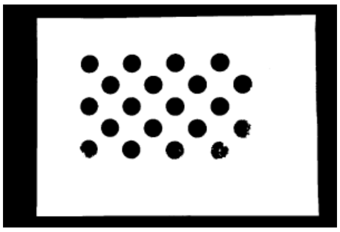
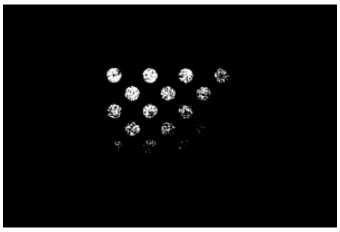
In addition, before calibration, move the robot to the starting point for calibration.

Adjust the Effect of Calibration Board’s Point Cloud

1. Open Mech-Eye Viewer to adjust camera parameters.
2. Adjust the 2D parameters to make sure that the 2D image of the calibration board is clear and neither overexposed nor underexposed.
3. Adjust 3D parameters to make sure that the point clouds of the circles on the calibration board are complete and have clear contours. It is recommended to set **Surface Smoothing** and **Outlier Removal** to **Normal** in the **Point Cloud Processing** section to reduce the point cloud fluctuation range.

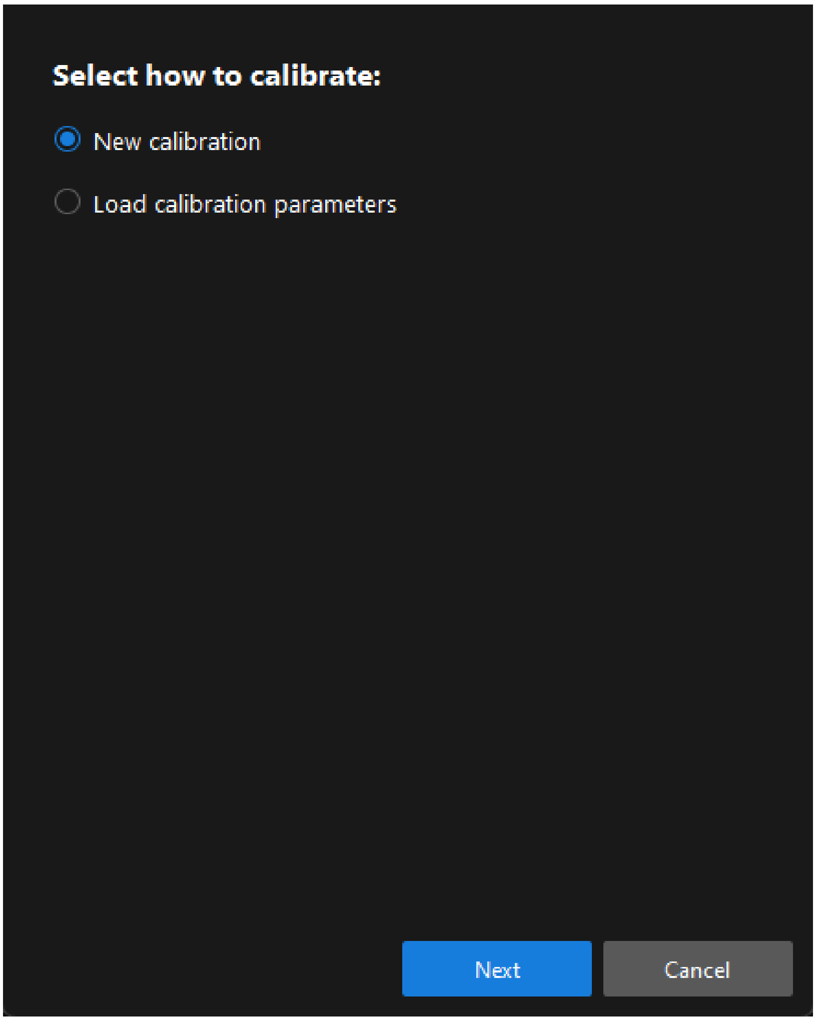
> If the on-site ambient lights are not ideal and affect the quality of 2D images and point clouds, you can use shading or supplemental light to improve the lighting conditions.

4. Make sure that the images and point cloud of the calibration board are up to standard after completing preceding steps.

	Normal	Overexposed	Underexposed
2D image			
Point cloud			

Pre-calibration Configuration

1. Open Mech-Vision, and click the **[Camera Calibration (Standard)]** button in the toolbar.The **Configuration before Calibration** window will be prompted.
2. In the **Select how to calibrate** window, select the **New calibration** radio button, and then click the **[Next]** button.

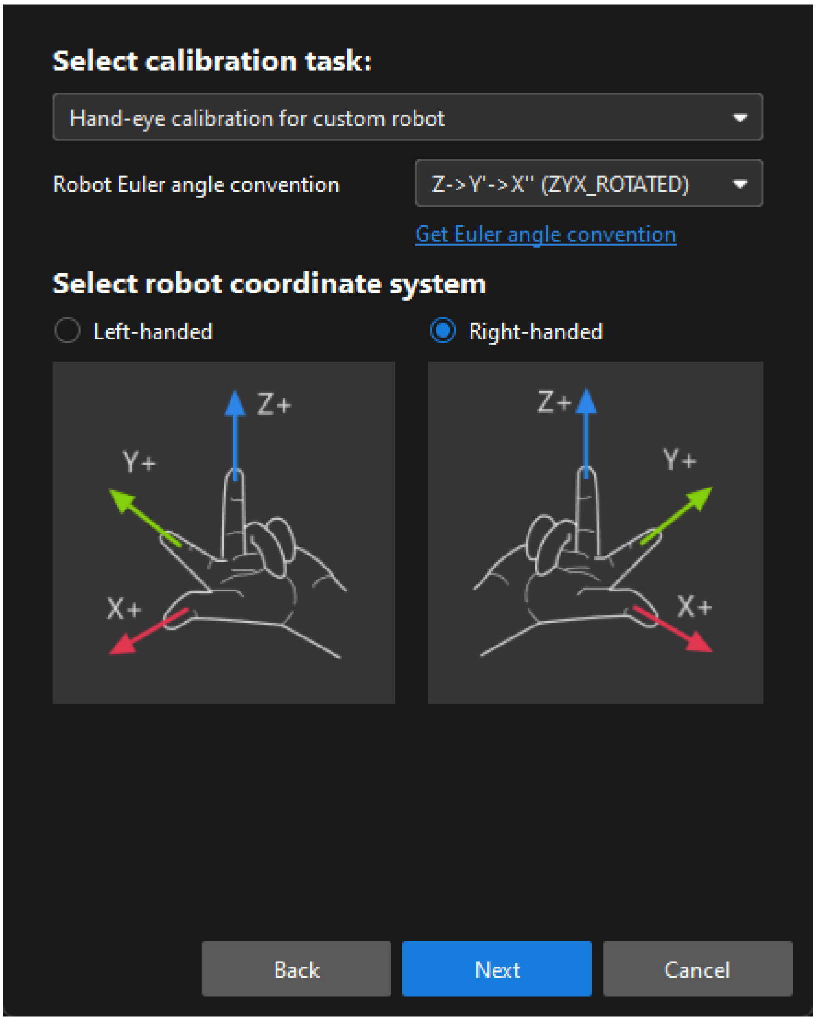


Select how to calibrate:

● New calibration

○ Load calibration parameters

3. In the **Select calibration task** window, select **Hand-eye calibration for custom robot** from the drop-down list box, specify the **Robot Euler angle convention** parameter, select the robot coordinate system type, and then click the **[Next]** button.



Select calibration task:

Hand-eye calibration for custom robot

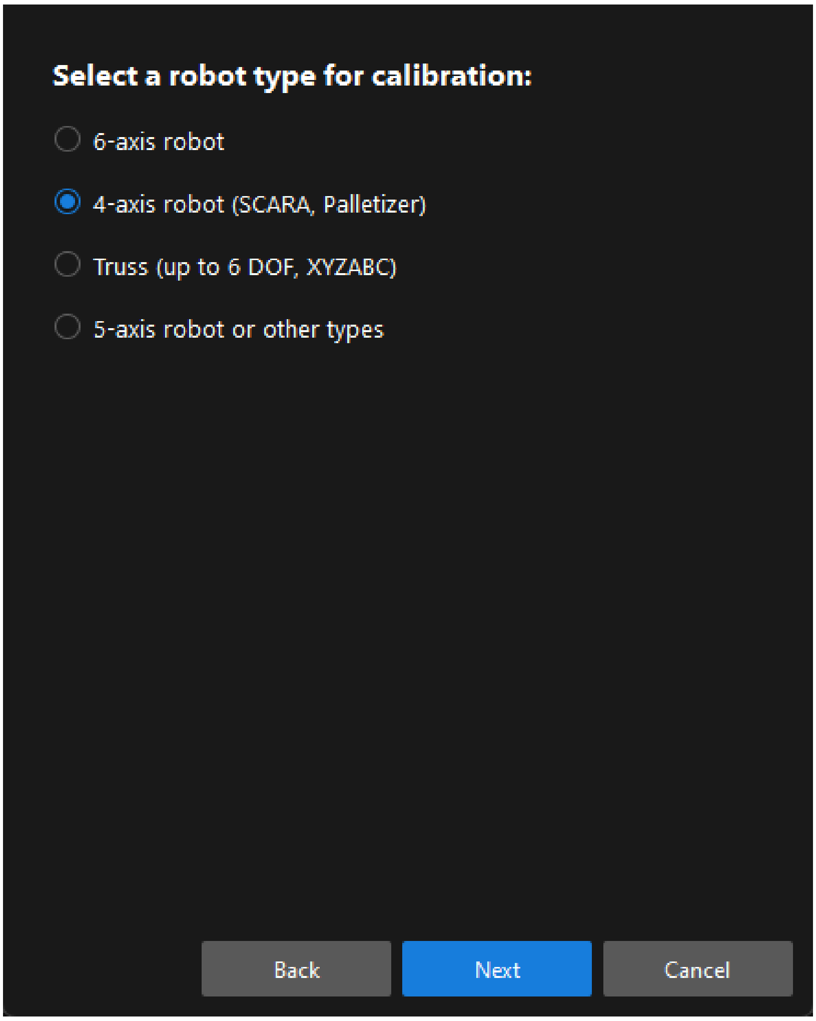
Robot Euler angle convention: Z->Y'->X'' (ZYX_ROTATED)

Get Euler angle convention

Select robot coordinate system

○ Left-handed ● Right-handed

4. In **Select a robot type for calibration** window, select the **4-axis robot (SCARA, Palletizer)** radio button, and then click the **[Next]** button.



Select a robot type for calibration:

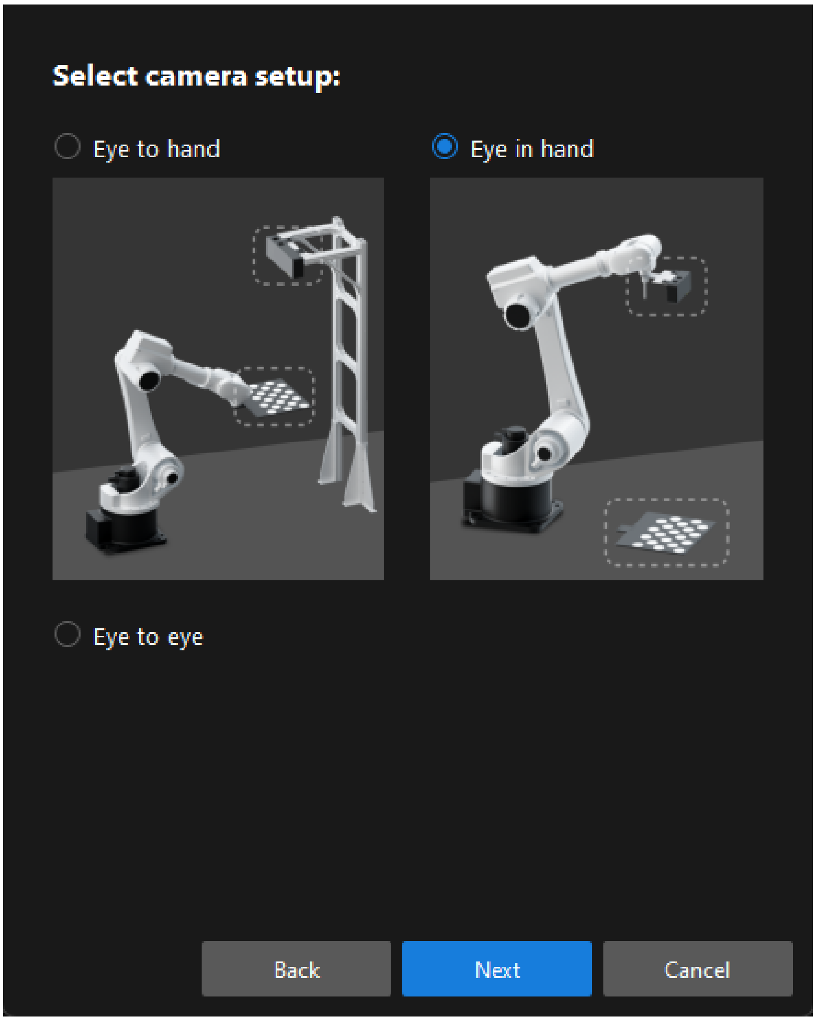
○ 6-axis robot

● 4-axis robot (SCARA, Palletizer)

○ Truss (up to 6 DOF, XYZABC)

○ 5-axis robot or other types

5. In the **Select camera setup** window, select the **Eye in hand** radio button, and then click the **[Next]** button.



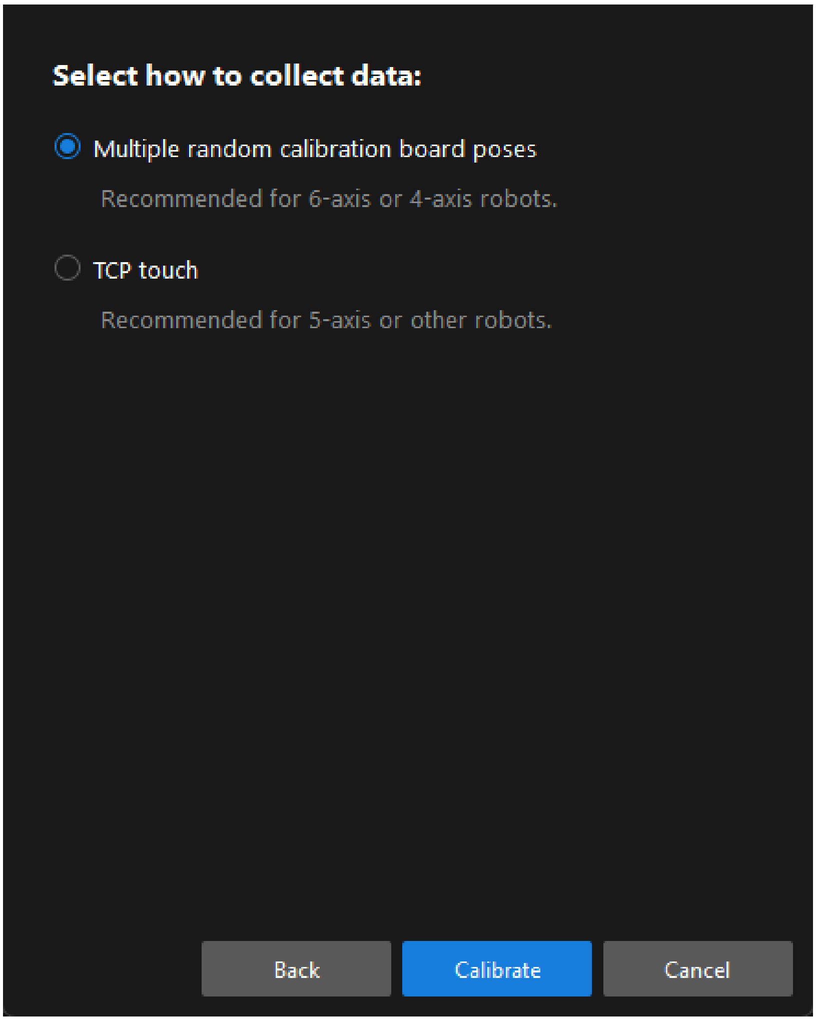
Select camera setup:

○ Eye to hand

● Eye in hand

○ Eye to eye

6. In the **Select how to collect data** window, select the **Multiple random calibration board poses**, and then click the **[Calibrate]** button.The **Calibration (Eye in Hand)** window will be prompted.



Select how to collect data:

● Multiple random calibration board poses

Recommended for 6-axis or 4-axis robots.

○ TCP touch

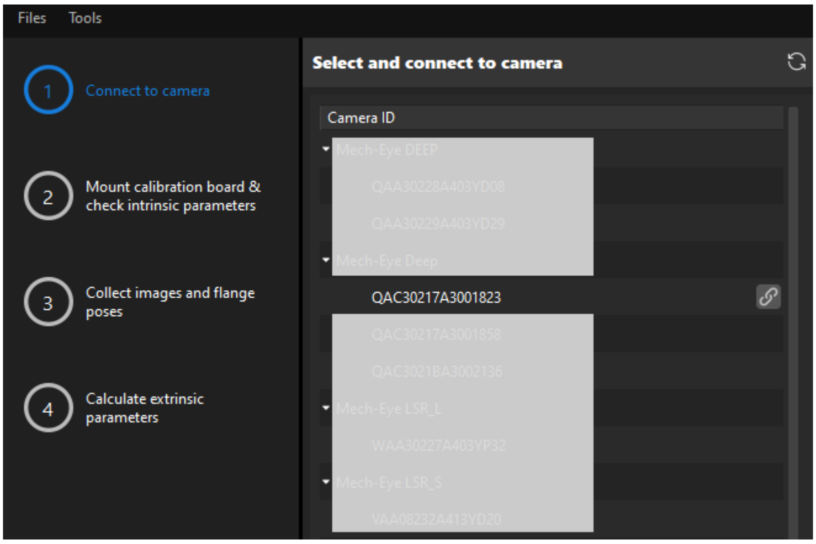
Recommended for 5-axis or other robots.

Till now, you have completed the pre-calibration configuration and can start the calibration procedure.

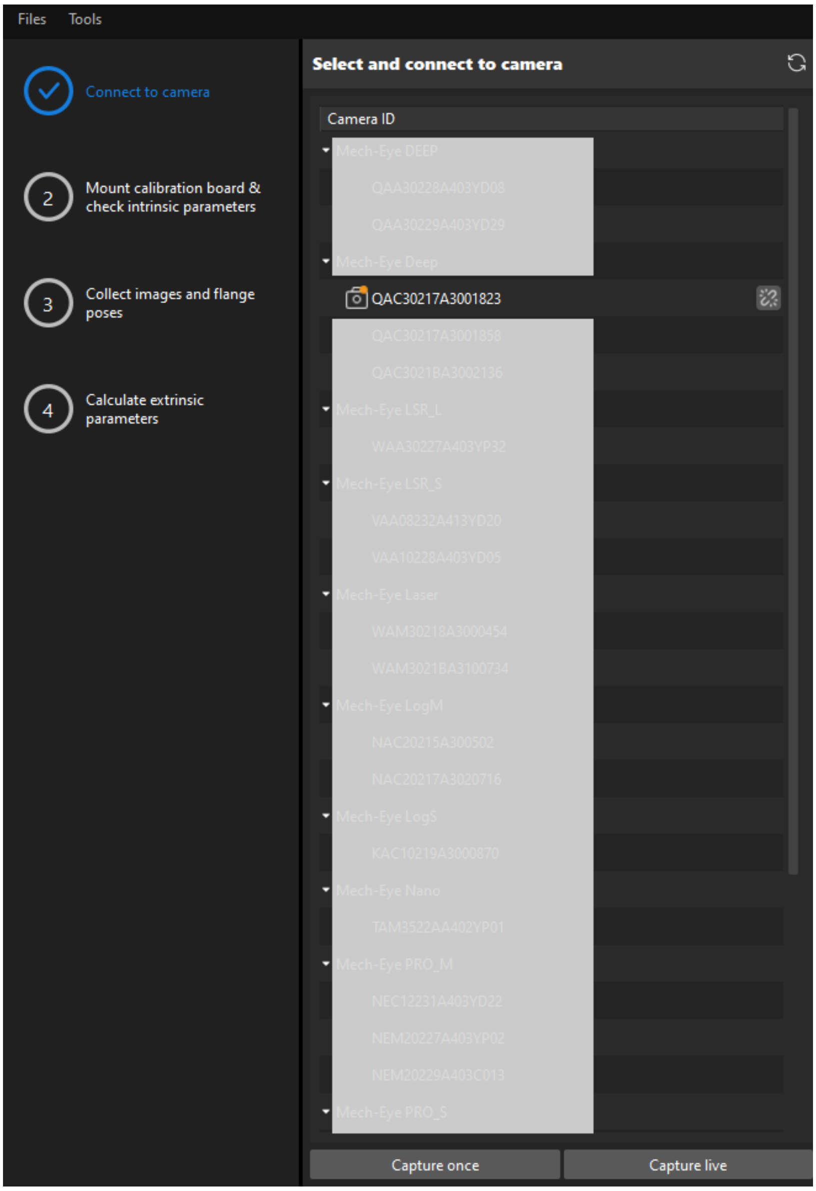
Calibration Procedure

Connect to the Camera

1. In the **Connect to camera** step, select the camera to connect in the Camera ID list, and then click the 🔗 button or double-click the camera entry to connect to it.


Files
Tools
1
Connect to camera
2
Mount calibration board & check intrinsic parameters
3
Collect images and flange poses
4
Calculate extrinsic parameters
Select and connect to camera
Camera ID
QAC30217A3001823

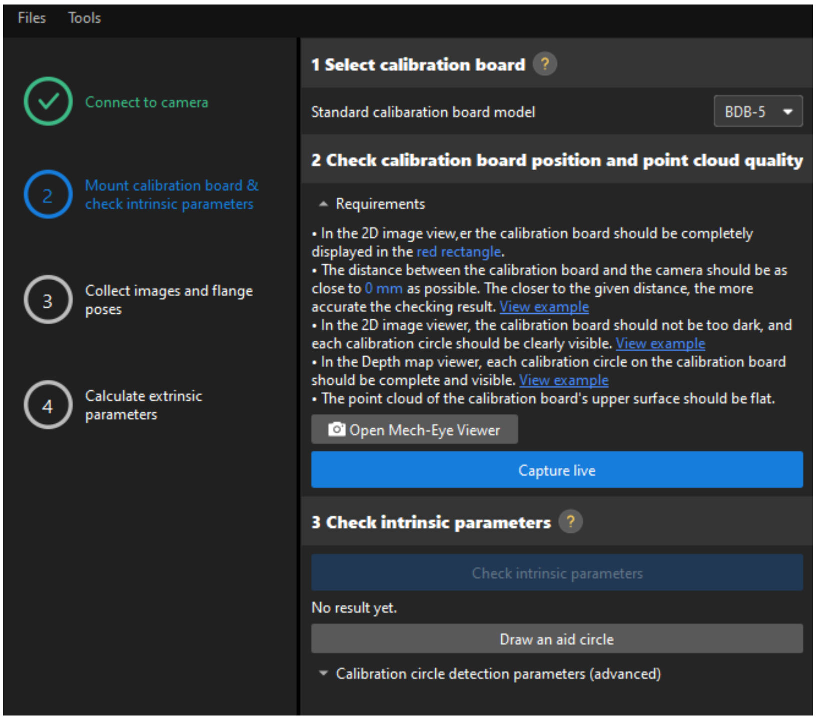
2. After the camera is connected, click the [Capture once] or [Capture live] button.


Files
Tools
Connect to camera
2
Mount calibration board & check intrinsic parameters
3
Collect images and flange poses
4
Calculate extrinsic parameters
Select and connect to camera
Camera ID
QAC30217A3001823
Capture once
Capture live

3. In the **Image viewer** panel, ensure that the camera can capture images normally and click the **[Next]** button on the bottom bar.

Mount Calibration Board & Check Intrinsic Parameters

1. In the **Mount calibration board & check intrinsic parameters** step, select the calibration board model for the **Standard calibration board model** parameter according to its model nameplate in the **1 Select calibration board** area.


Files
Tools
Connect to camera
2
Mount calibration board & check intrinsic parameters
3
Collect images and flange poses
4
Calculate extrinsic parameters
?
Open Mech-Eye Viewer
Capture live
?
Check intrinsic parameters
Draw an aid circle

1 Select calibration board

Standard calibaration board model: BDB-5

2 Check calibration board position and point cloud quality

Requirements

- In the 2D image view,er the calibration board should be completely displayed in the red rectangle.
- The distance between the calibration board and the camera should be as close to 0 mm as possible. The closer to the given distance, the more accurate the checking result. View example
- In the 2D image viewer, the calibration board should not be too dark, and each calibration circle should be clearly visible. View example
- In the Depth map viewer, each calibration circle on the calibration board should be complete and visible. View example
- The point cloud of the calibration board's upper surface should be flat.

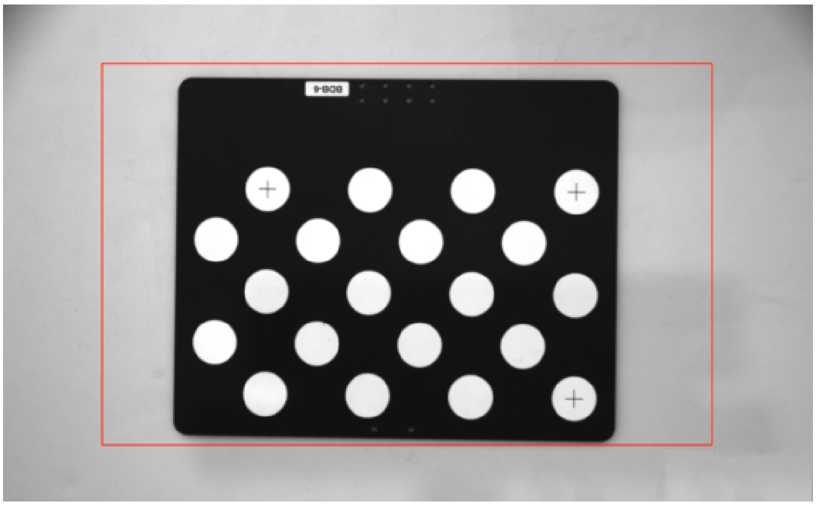
3 Check intrinsic parameters

No result yet.

Calibration circle detection parameters (advanced)

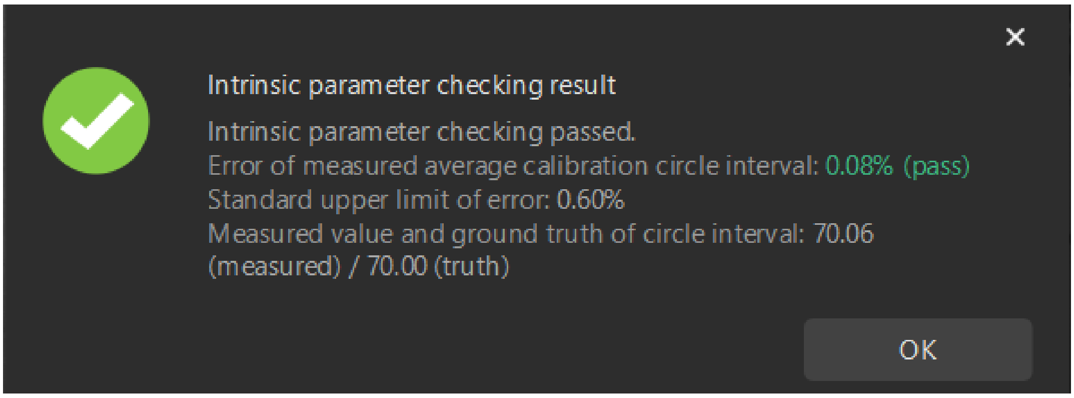
2. In the **2 Check calibration board position and point cloud quality** area, read carefully the requirements on the calibration board position and point cloud quality, and then click the **[Capture live]** button.The **[Capture live]** button will turn into **[Stop capturing and detect position]**.

3. Manually move the calibration board to the proper position (in the red rectangle), and ensure that the 2D image and depth map of the calibration board meet the requirements, and then click the **[Stop capturing and detect position]** button.



If the captured images do not meet the requirements, click the [**Open Mech-Eye Viewer**] button to open the Mech-Eye Viewer software, adjust the 2D and 3D exposure parameters and re-capture images. Please note that you need to change the **Parameter group** parameter to “calib” first.

4. In the **3 Check intrinsic parameters** area, click the [**Check intrinsic parameters**] button.
5. Confirm the results of the camera intrinsic parameter check.
   - If the camera intrinsic parameter check passes, click the [**OK**] button in the prompted window, and then click the [**Next**] button on the bottom bar.



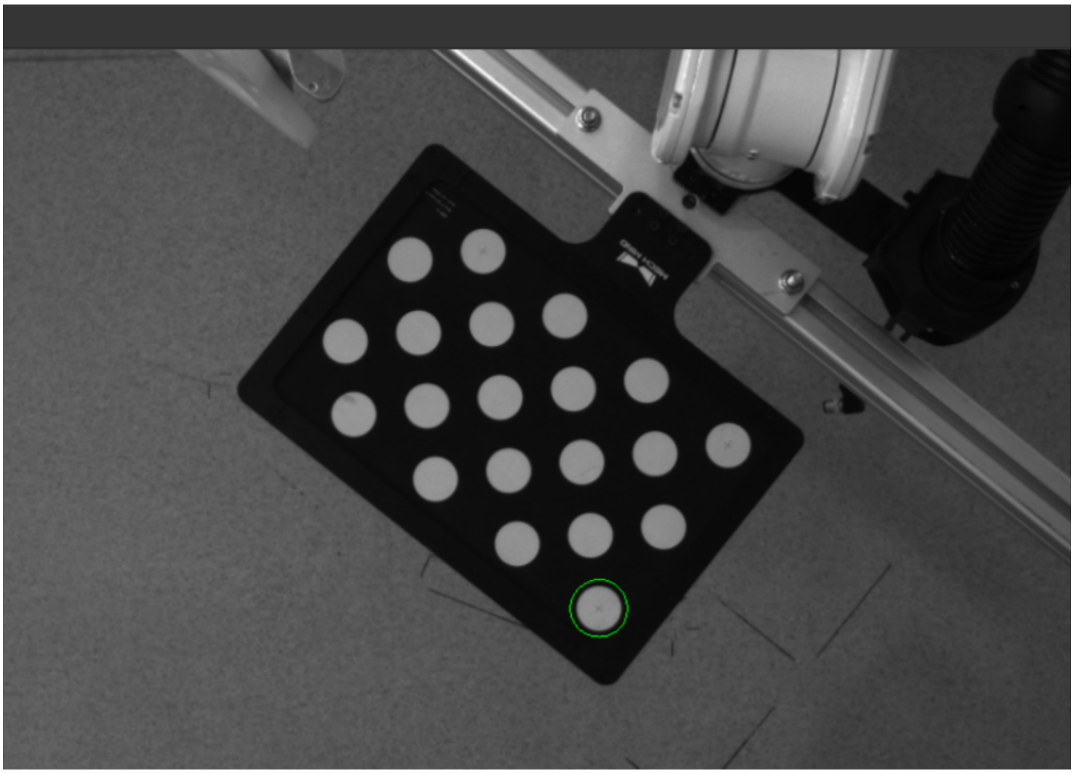
Intrinsic parameter checking result

Intrinsic parameter checking passed.
Error of measured average calibration circle interval: 0.08% (pass)
Standard upper limit of error: 0.60%
Measured value and ground truth of circle interval: 70.06 (measured) / 70.00 (truth)

   - If the camera intrinsic parameter check fails, adjust detection parameters by drawing aid circles or manually adjusting the calibration circle detection parameters, and then click the [**Recheck intrinsic parameters**] button.

Draw an Aid Circle

1. To draw an aid circle, click the [**Draw an aid circle**] button.
2. In the right **Image viewer** panel, right-click the calibration board image and clear the **Fit to window** checkbox, and press the `Ctrl` key and drag the roller to adjust the image to a suitable size.
3. Move the mouse pointer to the cross center point of the calibration circle, press the left mouse button and make the aid circle completely include the calibration circle and then release it.



4. Click the [**Recheck intrinsic parameters**] button, and confirm that the camera intrinsic parameter check passes.

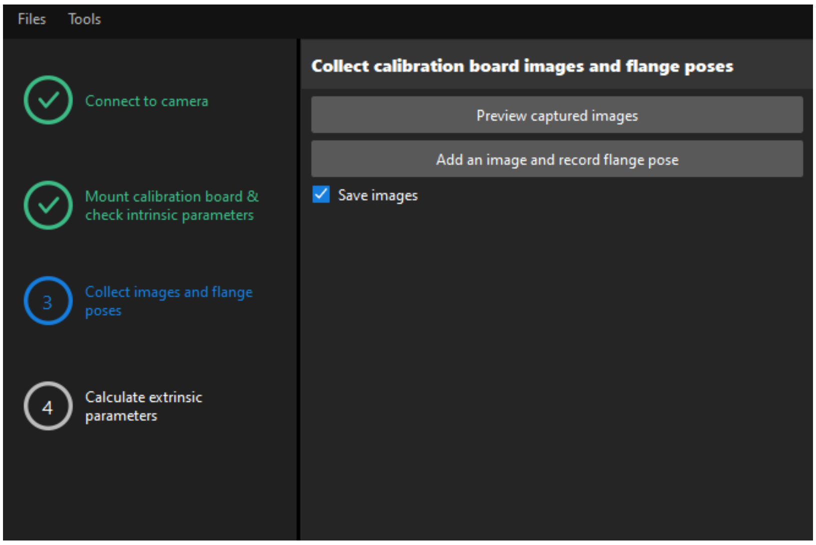
Manually Adjust Detection Parameters

To manually adjust detection parameters, click [**Calibration circle detection parameters (advanced)**], adjust parameter settings and click the [**Recheck intrinsic parameters**] button to confirm whether the camera intrinsic parameter check passes.

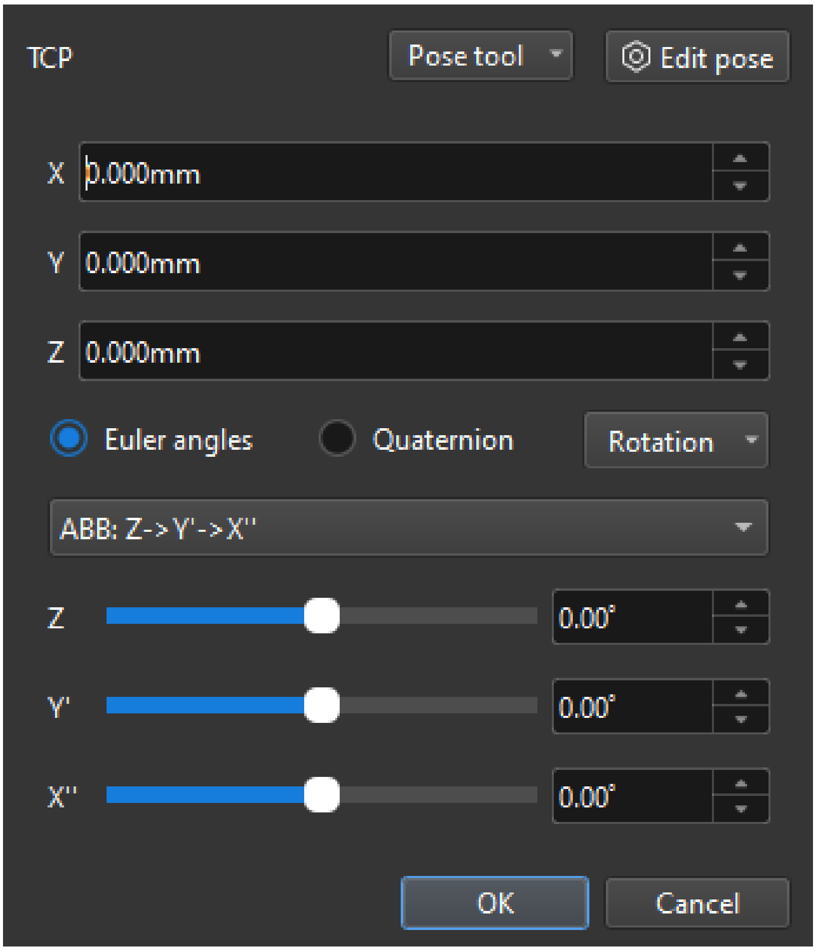
If the calibration circles are still not detected, please adjust the camera settings according to the on-site operating conditions.

Collect Images and Flange Poses

1. In the **Collect images and flange poses** step, select the [**Save images**] checkbox (selected by default, recommended). If you do not need to save images of calibration boards, clear this checkbox.


Files
Tools
Connect to camera
Mount calibration board & check intrinsic parameters
3
Collect images and flange poses
4
Calculate extrinsic parameters
Collect calibration board images and flange poses
Preview captured images
Add an image and record flange pose
☑ Save images

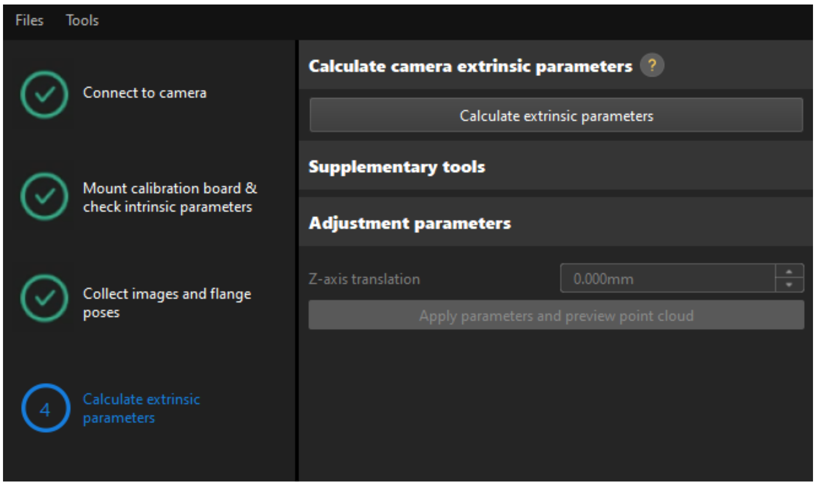
2. Control the robot to move different calibration points and then click the [**Add an image and record flange pose**] button.
3. In the prompted window, enter the robot flange pose.



4. Confirm that the number of the currently recognized calibration circles meets the requirements, and then click the [**Next**] button on the bottom bar.

Calculate Extrinsic Parameters

1. In the **Calculate extrinsic parameters** step, click the **[Calculate extrinsic parameters]** button.

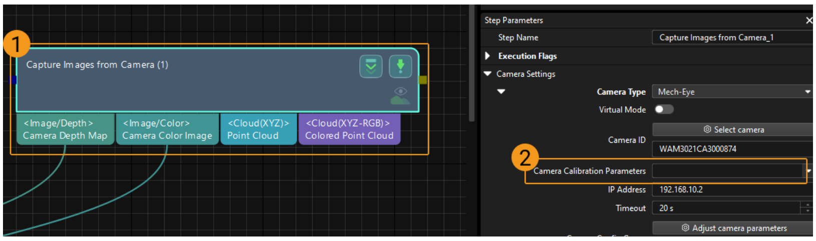

Files
Tools
Connect to camera
Mount calibration board & check intrinsic parameters
Collect images and flange poses
4
Calculate extrinsic parameters
Calculate camera extrinsic parameters
?
Calculate extrinsic parameters
Supplementary tools
Adjustment parameters
Apply parameters and preview point cloud

Z-axis translation: 0.000mm

Change the Extrinsic Parameter File

If the camera's extrinsic parameter file has been used by a Mech-Vision project, you need to change the extrinsic parameter file used by the Mech-Vision project to a new one after you finish the hand-eye calibration.

1. Select the **Capture Images from Camera** Step.



Step Name: Capture Images from Camera_1
Camera Type: Mech-Eye
Camera ID: WAM3021CA3000874
IP Address: 192.168.10.2
Timeout: 20 s

2. In the **Step Parameters** panel, click ▼ of the **Camera Calibration Parameters** parameter, and select the newly calibrated extrinsic parameter file.

Calibration-Related Status Codes

If the robot uses the Standard Interface to communicate with the Vision System during hand-eye calibration, the Vision System will return status codes in response data to indicate the execution status of the commands. The status codes indicate both the normal execution results and execution failures. You can perform further troubleshooting according to the status codes.

Calibration Error Codes

Error code	Description
7001	Calibration: Parameter error
7002	Calibration: Calibration flange pose not provided by Mech-Vision
7003	Calibration: Calibration joint positions not provided by Mech-Vision
7004	Calibration: Robot failed to reach calibration point

Calibration Normal Status Codes

Normal Status Code	Description
7100	Calibration: Robot reaches calibration point successfully
7101	Calibration: Pose received from Mech-Vision successfully

4.5.3.5. Manual Calibration in the Eye-in-Hand Setup (Six-Axis Robots - Using TCP Touch Method)

This how-to guide introduces how to complete the manual calibration for six-axis robots using the TCP touch method in the eye-in-hand (EIH) setup.

Preparation before Calibration

Before hand-eye calibration, you need to finish the following preparations:

- Construct the Mech-Mind Vision System.
- Prepare the materials required for calibration.
- Adjust the effect of calibration board's point cloud.

Construct the Vision System

Construct the Mech-Mind Vision System by referring to the section Vision System Hardware Setup.

You need to use Mech-Eye Viewer, Mech-Vision and Mech-Viz during hand-eye calibration. Please ensure that they have been installed and are running the latest versions.

Prepare the Materials Required for Calibration

The manual calibration in the EIH setup needs to use the calibration board and sharp tip.

Please prepare the calibration board according to the following requirements:

- Ensure that the circles of the calibration board are clearly visible and without obvious scratches, and the board does not suffer from deformations.
- In the EIH setup, please place the calibration board in the center of the working plane, where target objects are to be placed.

The sharp tip should be mounted on the center of the robot flange. If an undetachable gripper is connected to the robot flange, you can attach the sharp point directly to the gripper.

In addition, before calibration, move the robot to the starting point for calibration.

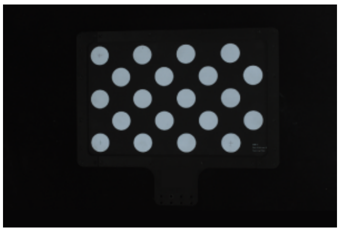
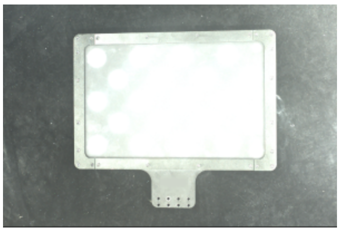
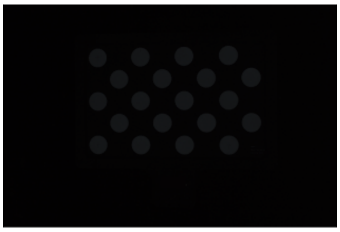
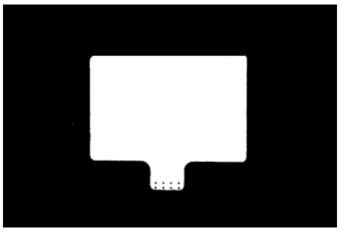
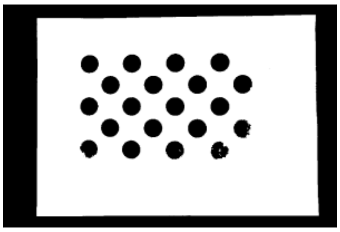
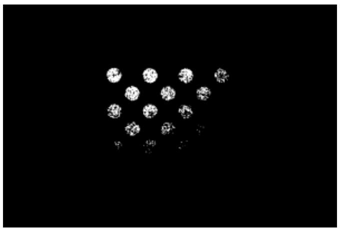
Adjust the Effect of Calibration Board's Point Cloud

1. Open Mech-Eye Viewer to adjust camera parameters.
2. Adjust the 2D parameters to make sure that the 2D image of the calibration board is clear and neither overexposed nor underexposed.
3. Adjust 3D parameters to make sure that the point clouds of the circles on the calibration board are complete and have clear contours. It is recommended to set **Surface Smoothing** and

Outlier Removal to **Normal** in the **Point Cloud Processing** section to reduce the point cloud fluctuation range.

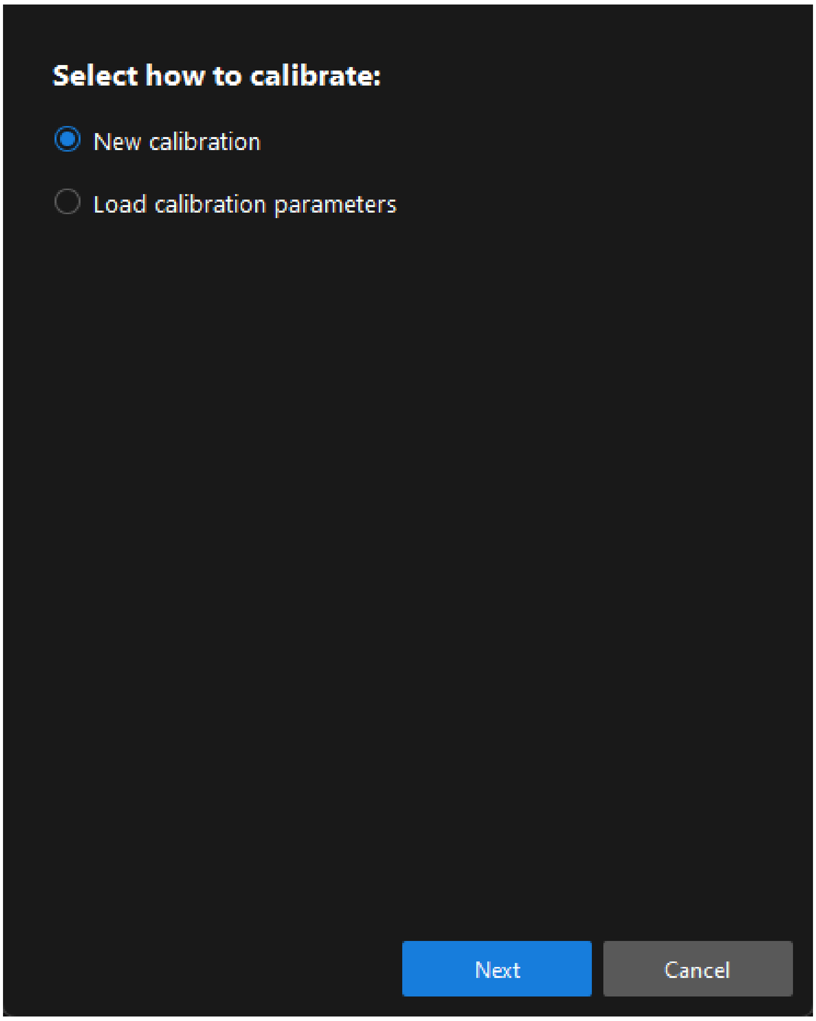
> If the on-site ambient lights are not ideal and affect the quality of 2D images and point clouds, you can use shading or supplemental light to improve the lighting conditions.

4. Make sure that the images and point cloud of the calibration board are up to standard after completing preceding steps.

	Normal	Overexposed	Underexposed
2D image			
Point cloud			

Pre-calibration Configuration

1. Open Mech-Vision, and click the **[Camera Calibration (Standard)]** button in the toolbar.The **Configuration before Calibration** window will be prompted.
2. In the **Select how to calibrate** window, select the **New calibration** radio button, and then click the **[Next]** button.

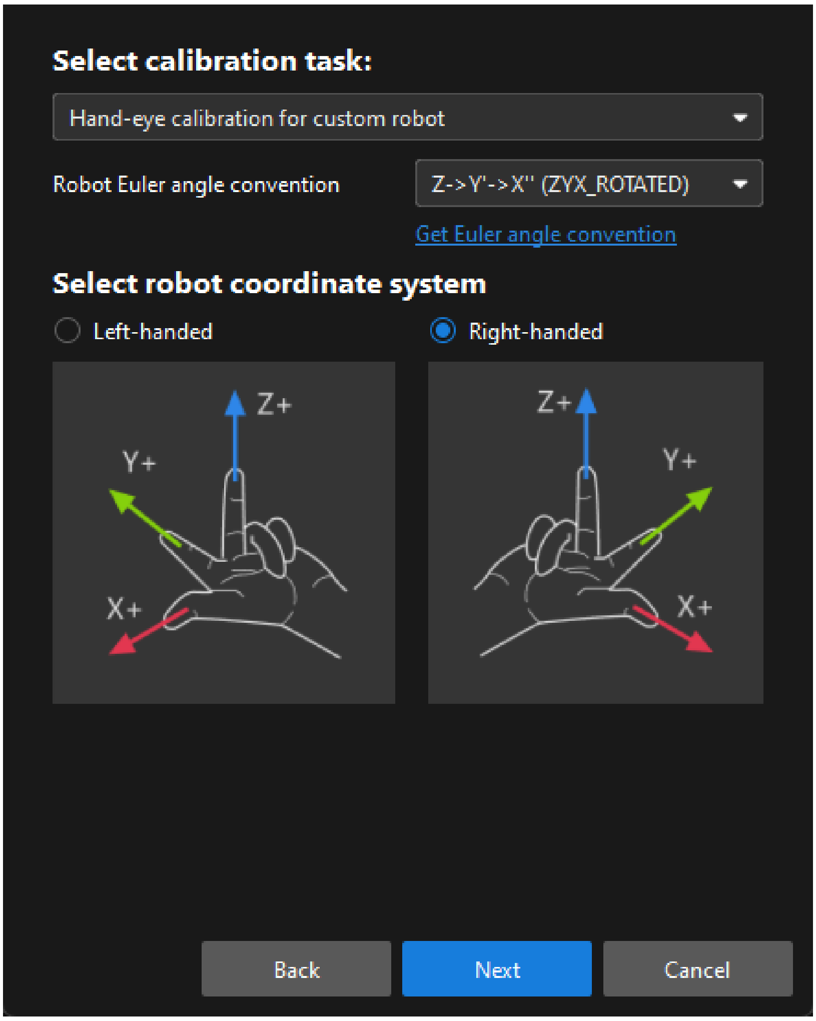


Select how to calibrate:

● New calibration

○ Load calibration parameters

3. In the **Select calibration task** window, select **Hand-eye calibration for custom robot** from the drop-down list box, specify the **Robot Euler angle convention** parameter, select the robot coordinate system type, and then click the **[Next]** button.



Select calibration task:

Hand-eye calibration for custom robot

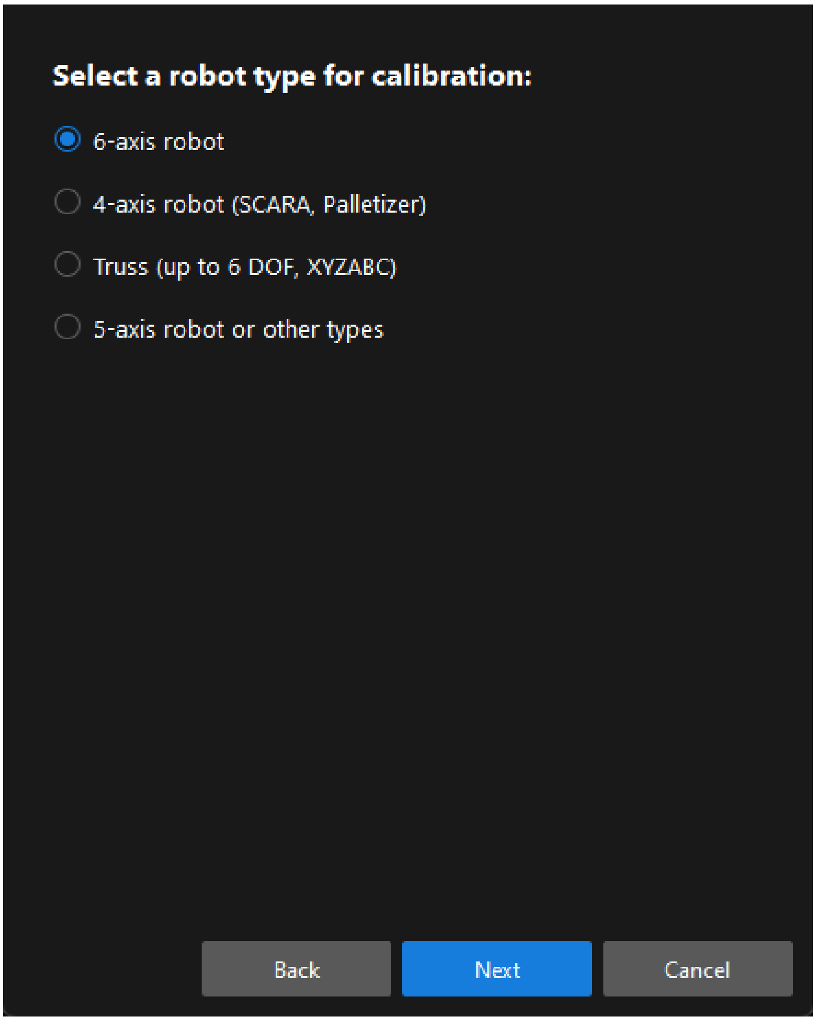
Robot Euler angle convention: Z->Y'->X'' (ZYX_ROTATED)

Get Euler angle convention

Select robot coordinate system

○ Left-handed ● Right-handed

4. In the **Select a robot type for calibration** window, select the **6-axis robot** radio button, and then click the **[Next]** button.



Select a robot type for calibration:

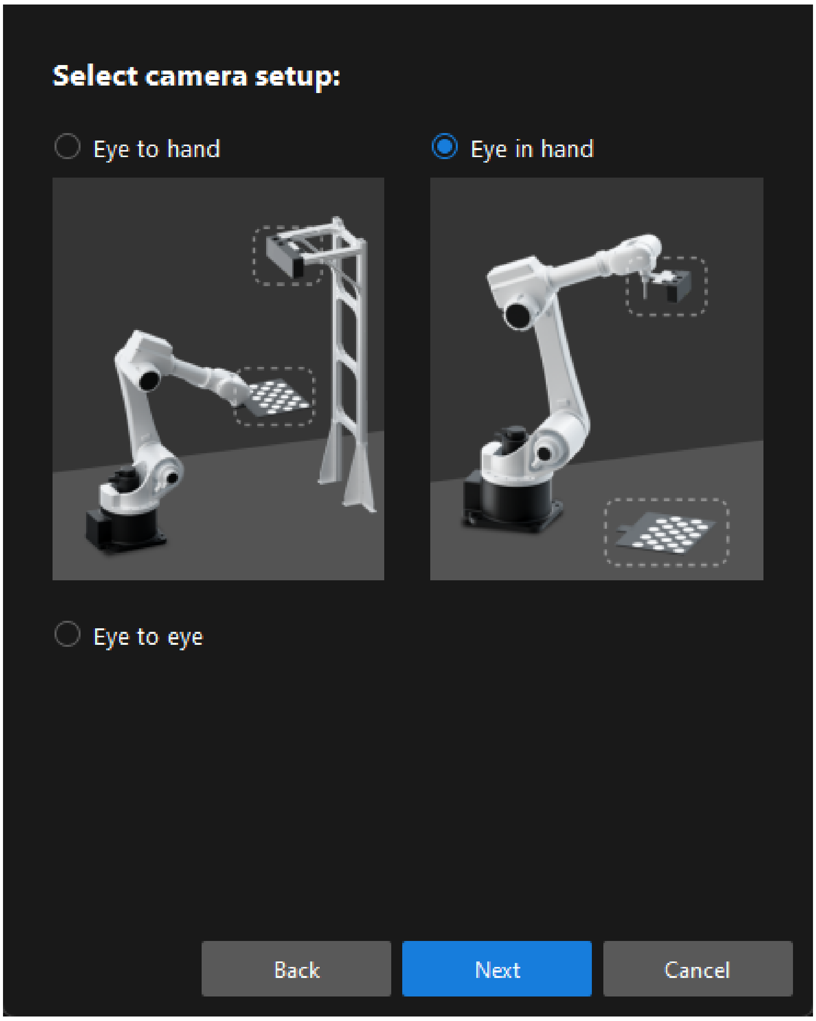
● 6-axis robot

○ 4-axis robot (SCARA, Palletizer)

○ Truss (up to 6 DOF, XYZABC)

○ 5-axis robot or other types

5. In the **Select camera setup** window, select the **Eye in hand** radio button, and then click the **[Next]** button.



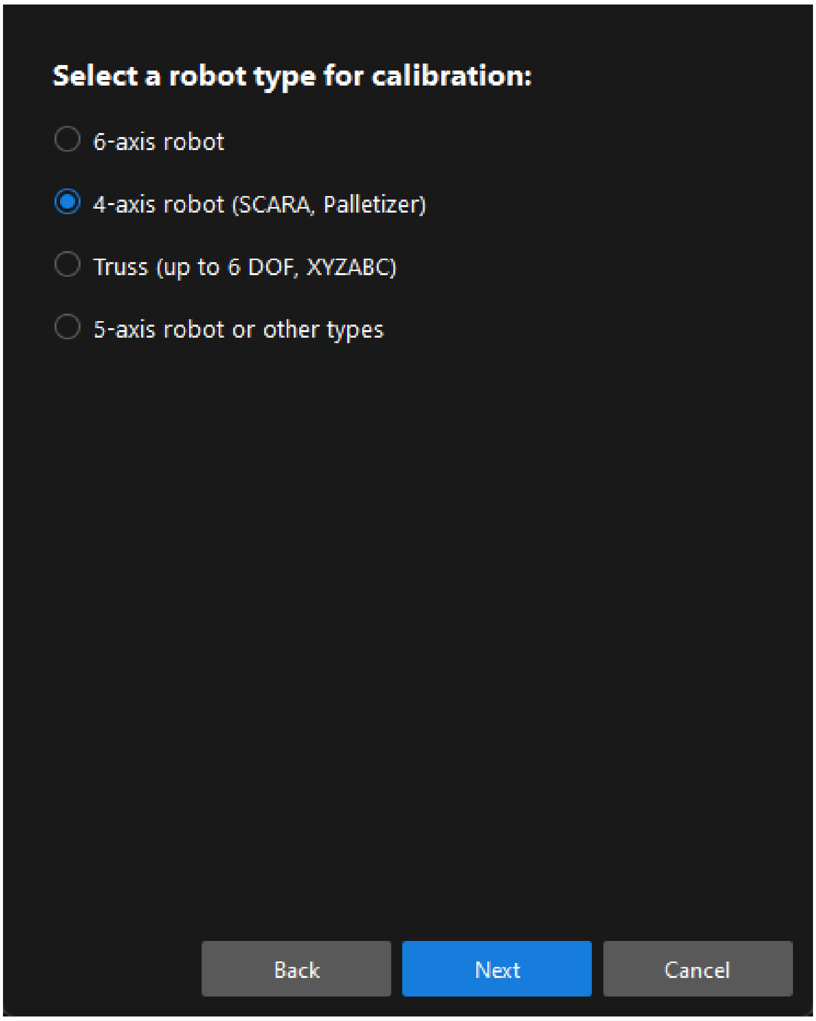
Select camera setup:

○ Eye to hand

● Eye in hand

○ Eye to eye

6. In the **Select how to collect data** window, select the **TCP touch** radio button, and then click the **[Calibrate]** button.The **Calibration (Eye in Hand) window** will be prompted.



Select a robot type for calibration:

○ 6-axis robot

● 4-axis robot (SCARA, Palletizer)

○ Truss (up to 6 DOF, XYZABC)

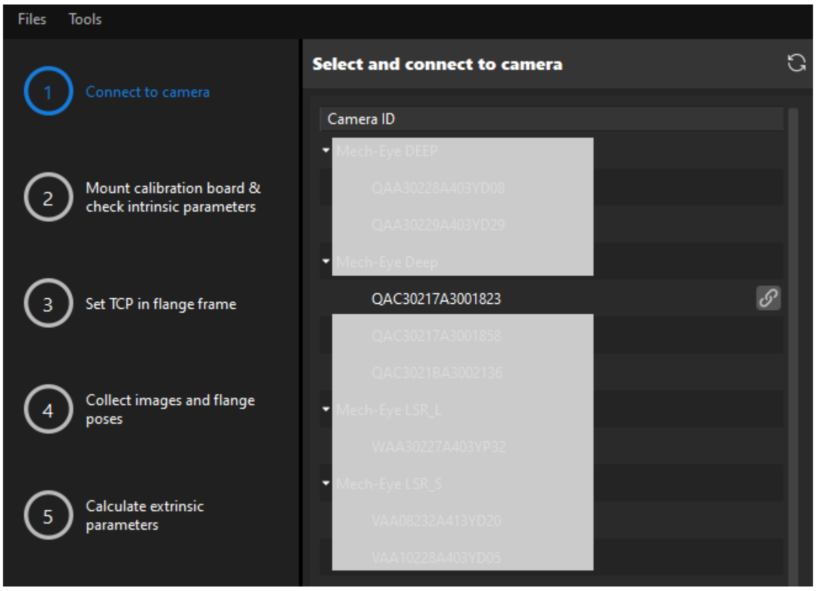
○ 5-axis robot or other types

Till now, you have completed the pre-calibration configuration and can start the calibration procedure.

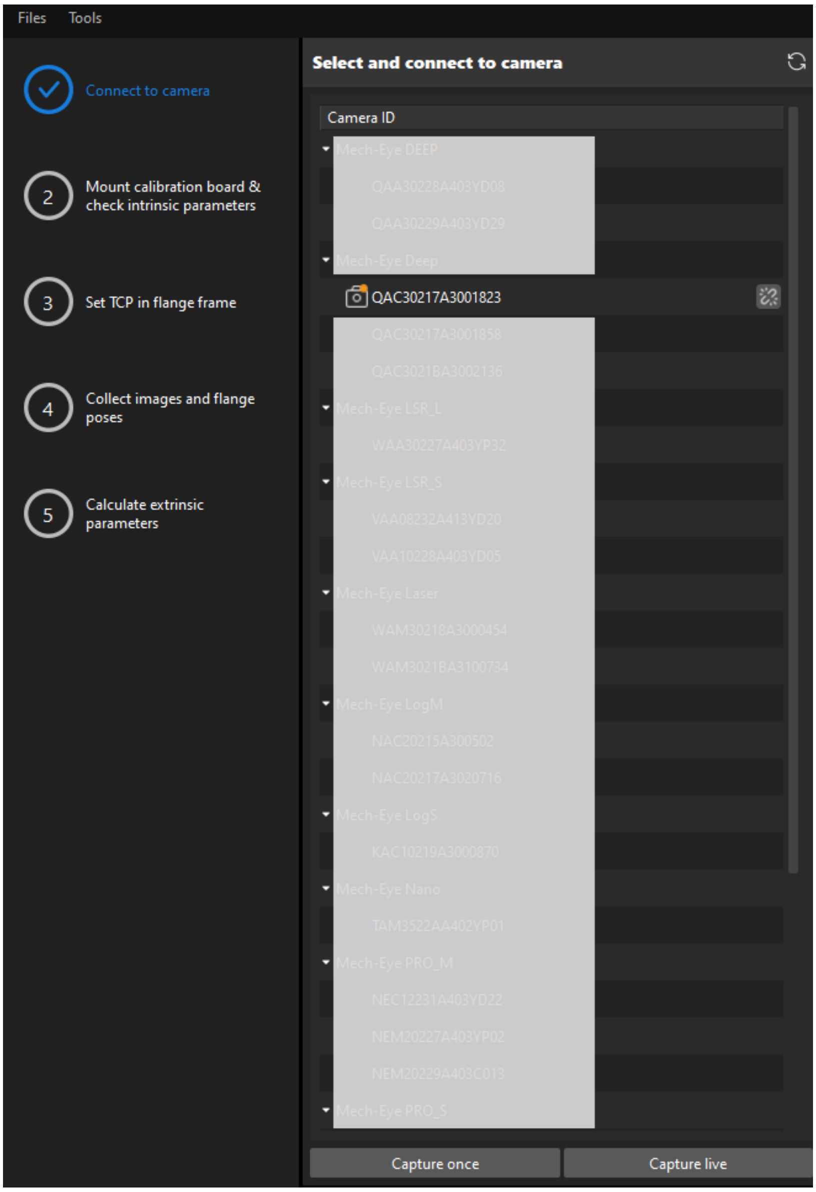
Calibration Procedure

Connect to the Camera

1. In the **Connect to camera** step, select the camera to connect in the Camera ID list, and then click the 🔗 button or double-click the camera entry to connect to it.


Files
Tools
1
Connect to camera
2
Mount calibration board & check intrinsic parameters
3
Set TCP in flange frame
4
Collect images and flange poses
5
Calculate extrinsic parameters
Select and connect to camera
Camera ID
[redacted]
QAC30217A3001823
[redacted]

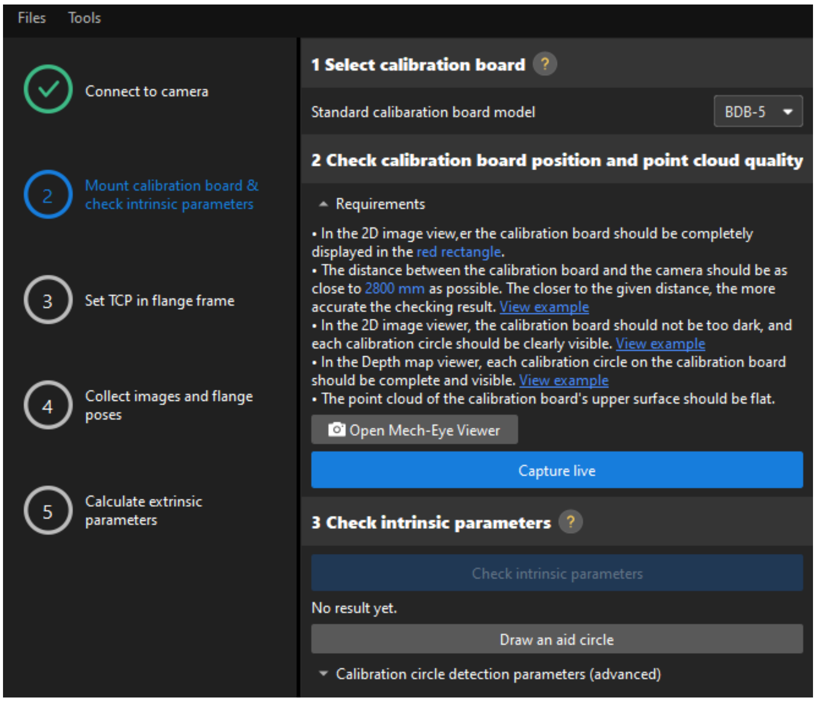
2. After the camera is connected, click the [Capture once] or [Capture live] button.


Files
Tools
Connect to camera
2
Mount calibration board & check intrinsic parameters
3
Set TCP in flange frame
4
Collect images and flange poses
5
Calculate extrinsic parameters
Select and connect to camera
Camera ID
[redacted]
QAC30217A3001823
[redacted]
Capture once
Capture live

3. In the **Image viewer** panel, ensure that the camera can capture images normally and click the **[Next]** button on the bottom bar.

Mount Calibration Board & Check Intrinsic Parameters

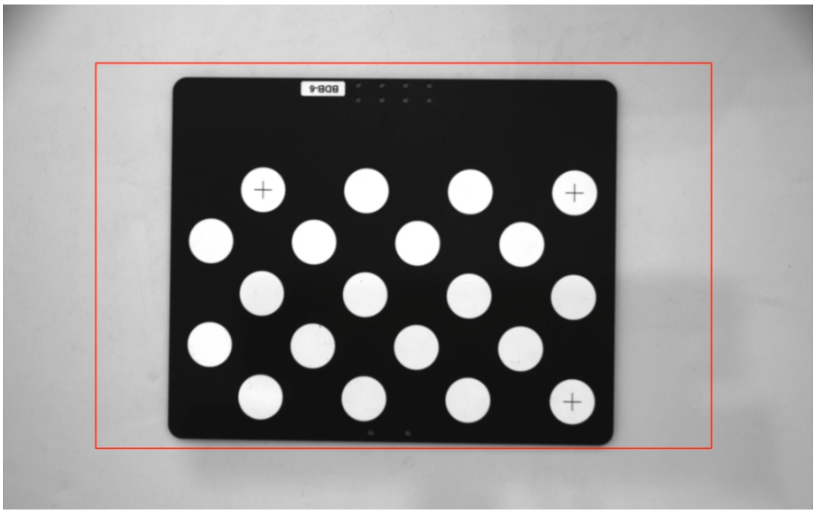
1. In the **Mount calibration board & check intrinsic parameters** step, select the calibration board model for the **Standard calibration board model** parameter according to its model nameplate in the **1 Select calibration board** area.


Files
Tools
Connect to camera
2
Mount calibration board & check intrinsic parameters
3
Set TCP in flange frame
4
Collect images and flange poses
5
Calculate extrinsic parameters
1 Select calibration board ?
Standard calibaration board model
BDB-5
2 Check calibration board position and point cloud quality
Requirements
Open Mech-Eye Viewer
Capture live
3 Check intrinsic parameters ?
Check intrinsic parameters
Draw an aid circle
Calibration circle detection parameters (advanced)

• In the 2D image view,er the calibration board should be completely displayed in the red rectangle.
• The distance between the calibration board and the camera should be as close to 2800 mm as possible. The closer to the given distance, the more accurate the checking result. View example
• In the 2D image viewer, the calibration board should not be too dark, and each calibration circle should be clearly visible. View example
• In the Depth map viewer, each calibration circle on the calibration board should be complete and visible. View example
• The point cloud of the calibration board's upper surface should be flat.

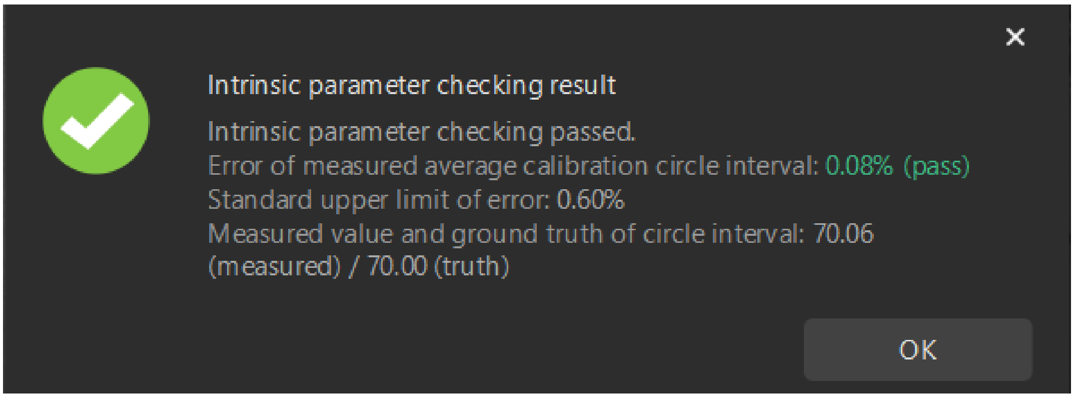
No result yet.

2. In the **2 Check calibration board position and point cloud quality** area, read carefully the requirements on the calibration board position and point cloud quality, and then click the **[Capture live]** button.The **[Capture live]** button will turn into **[Stop capturing and detect position]**.
3. Manually move the calibration board to the proper position (in the red rectangle), and ensure that the 2D image and depth map of the calibration board meet the requirements, and then click the **[Stop capturing and detect position]** button.



> If the captured images do not meet the requirements, click the [**Open Mech-Eye Viewer**] button to open the Mech-Eye Viewer software, adjust the 2D and 3D exposure parameters and re-capture images. Please note that you need to change the **Parameter group** parameter to "calib" first.

4. In the **3 Check intrinsic parameters** area, click the [**Check intrinsic parameters**] button.
5. Confirm the results of the camera intrinsic parameter check.
   - If the camera intrinsic parameter check passes, click the [**OK**] button in the prompted window, and then click the [**Next**] button on the bottom bar.



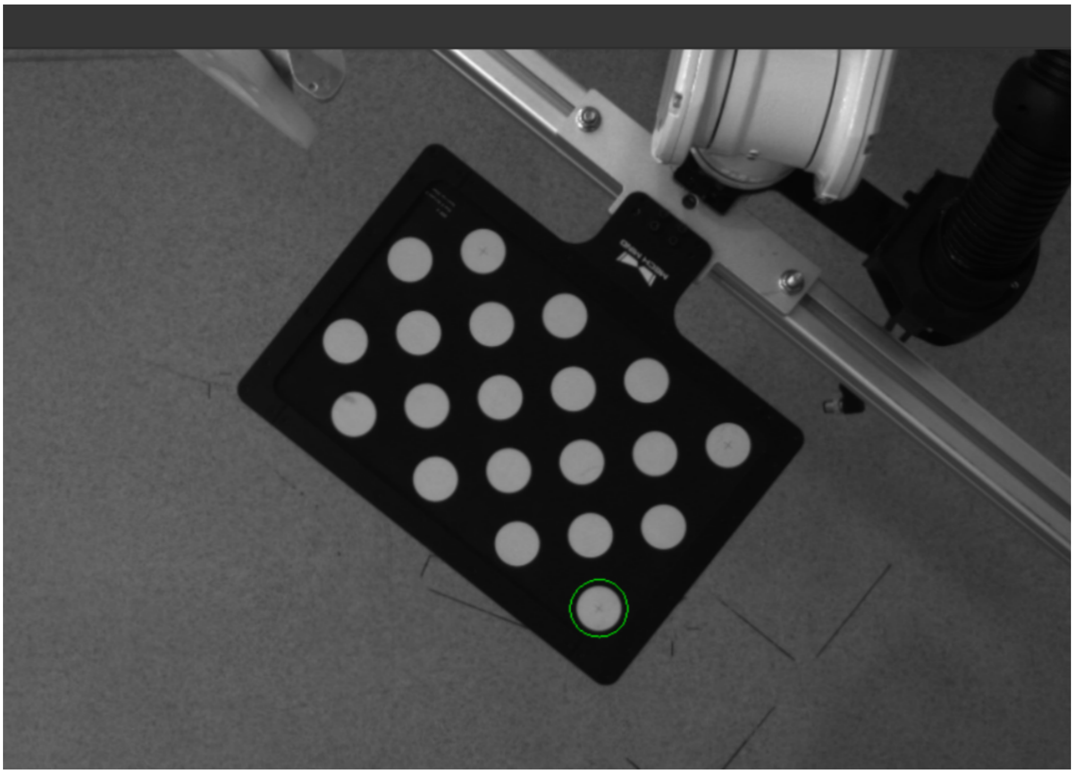
Intrinsic parameter checking result

Intrinsic parameter checking passed.
Error of measured average calibration circle interval: 0.08% (pass)
Standard upper limit of error: 0.60%
Measured value and ground truth of circle interval: 70.06 (measured) / 70.00 (truth)

   - If the camera intrinsic parameter check fails, adjust detection parameters by drawing aid circles or manually adjusting the calibration circle detection parameters, and then click the [**Recheck intrinsic parameters**] button.

Draw an Aid Circle

1. To draw an aid circle, click the [**Draw an aid circle**] button.
2. In the right **Image viewer** panel, right-click the calibration board image and clear the **Fit to window** checkbox, and press the `Ctrl` key and drag the roller to adjust the image to a suitable size.
3. Move the mouse pointer to the cross center point of the calibration circle, press the left mouse button and make the aid circle completely include the calibration circle and then release it.



4. Click the [**Recheck intrinsic parameters**] button, and confirm that the camera intrinsic parameter check passes.

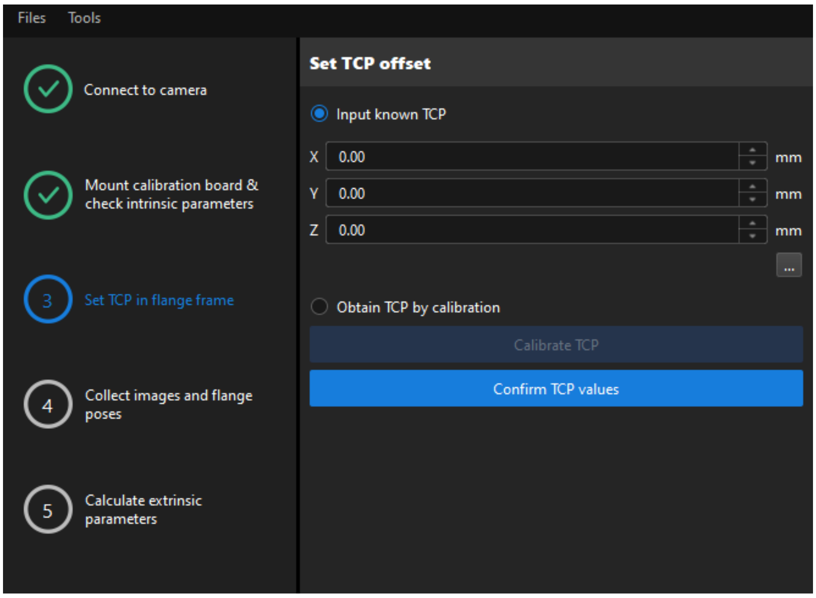
Manually Adjust Detection Parameters

To manually adjust detection parameters, click [**Calibration circle detection parameters (advanced)**], adjust parameter settings and click the [**Recheck intrinsic parameters**] to confirm whether the camera intrinsic parameter check passes.

If the calibration circles are still not detected, please adjust the camera settings according to the on-site operating conditions.

Set TCP in Flange Frame

1. In the **Set TCP in flange frame** step, select the **Input known TCP** radio button, and enter the TCP values.



Files Tools

Connect to camera

Mount calibration board & check intrinsic parameters

3 Set TCP in flange frame

4 Collect images and flange poses

5 Calculate extrinsic parameters

Set TCP offset

● Input known TCP

X 0.00 mm

Y 0.00 mm

Z 0.00 mm

○ Obtain TCP by calibration

Calibrate TCP

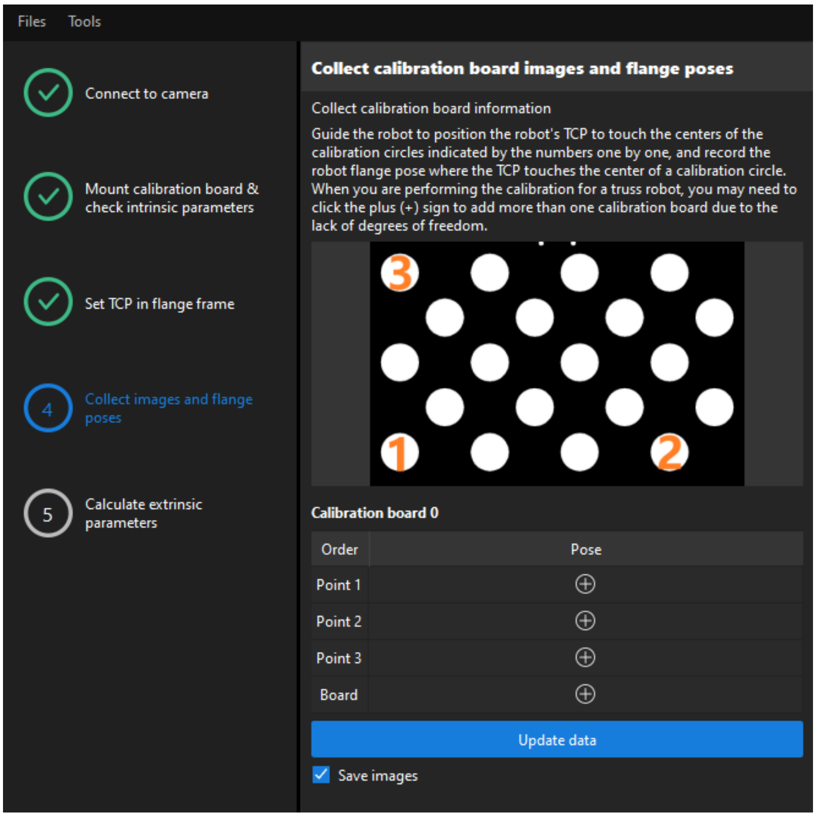
Confirm TCP values

2. Click the [Confirm TCP values] button.
3. Click the [Next] button on the bottom bar.

> If the TCP values are unknown, select the **Obtain TCP by calibration** radio button, and then click the [Calibrate TCP] button. You can use the **TCP calibration** tool to calculate the TCP values.

Collect Images and Flange Poses

1. In the **Collect images and flange poses** step, select the **Save images** checkbox.



Files Tools

Connect to camera

Mount calibration board & check intrinsic parameters

Set TCP in flange frame

4 Collect images and flange poses

5 Calculate extrinsic parameters

Collect calibration board images and flange poses

Collect calibration board information

Guide the robot to position the robot's TCP to touch the centers of the calibration circles indicated by the numbers one by one, and record the robot flange pose where the TCP touches the center of a calibration circle. When you are performing the calibration for a truss robot, you may need to click the plus (+) sign to add more than one calibration board due to the lack of degrees of freedom.

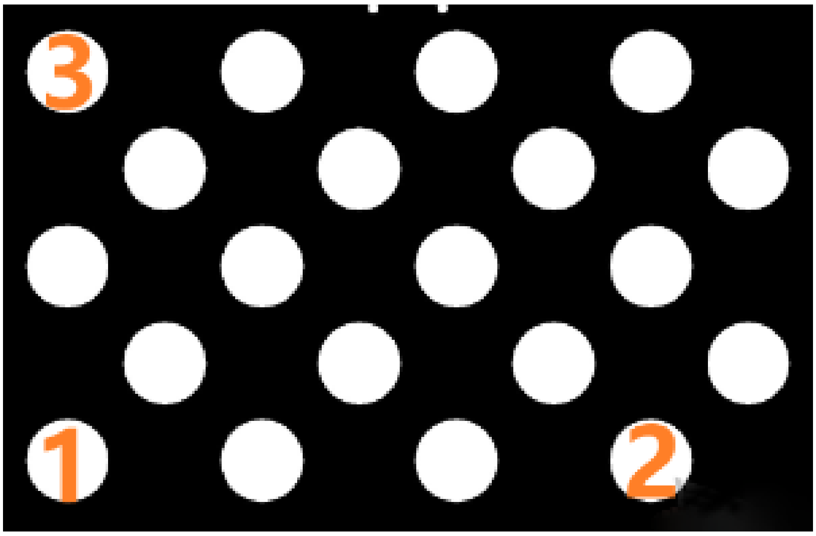
Calibration board 0

Order	Pose
Point 1	⊕
Point 2	⊕
Point 3	⊕
Board	⊕

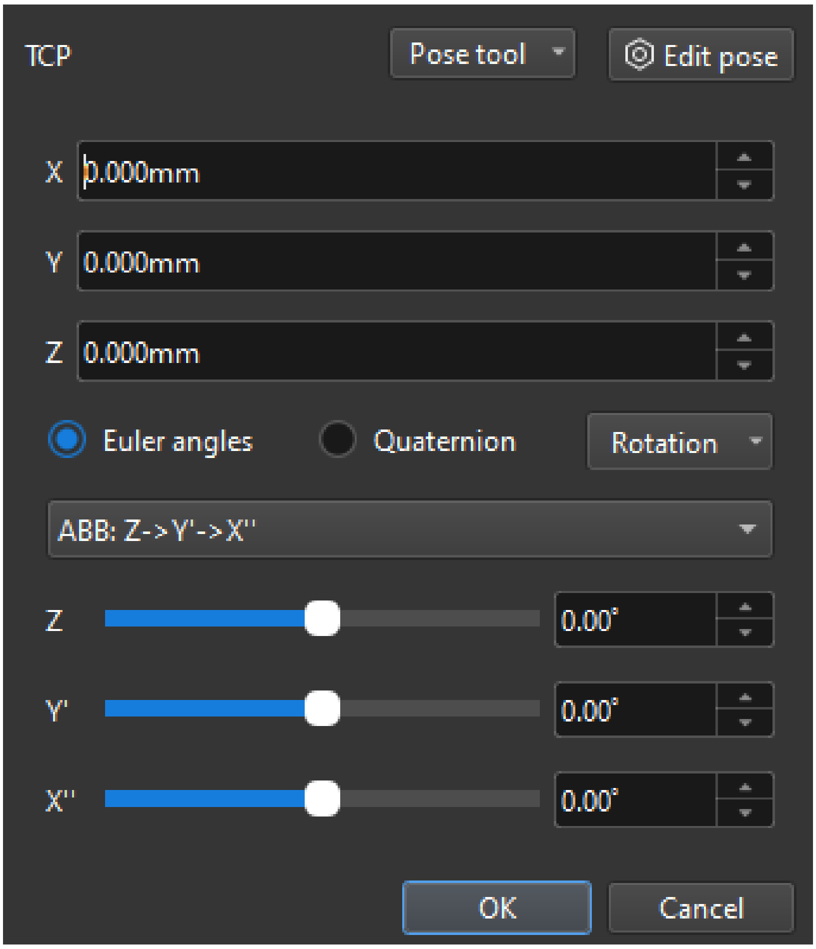
Update data

☑ Save images

2. Control the robot to let its end tip touch the cross center point of calibration circle “point 1” on the calibration board, and record the robot flange pose in the teach pendant.



3. Click the ⊕ button of point 1, and enter the robot flange pose in the prompted **Input Flange Pose of Robot** dialog box, and then click the [OK] button.



X 0.000mm

Y 0.000mm

Z 0.000mm

● Euler angles ○ Quaternion Rotation

ABB: Z->Y'->X''

Z 0.00°

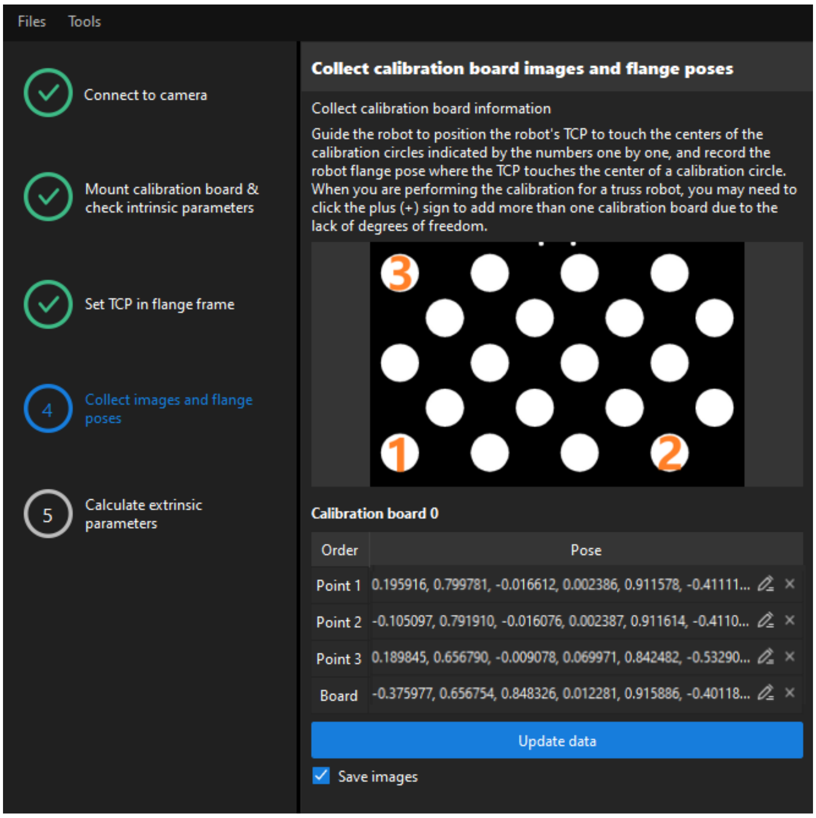
Y' 0.00°

X'' 0.00°

4. Repeat the preceding steps, and let the end tip touch the cross center points of point 2 and point 3, and enter robot flange poses.

5. Click the ⊕ button of the current calibration board to capture the images of it.

6. Click the **[Update data]** button and then click the **[Next]** button on the bottom bar.



Files
Tools
Connect to camera
Mount calibration board & check intrinsic parameters
Set TCP in flange frame
4
Collect images and flange poses
5
Calculate extrinsic parameters
3
1
2

Collect calibration board images and flange poses

Collect calibration board information

Guide the robot to position the robot's TCP to touch the centers of the calibration circles indicated by the numbers one by one, and record the robot flange pose where the TCP touches the center of a calibration circle. When you are performing the calibration for a truss robot, you may need to click the plus (+) sign to add more than one calibration board due to the lack of degrees of freedom.

Calibration board 0

Order	Pose
Point 1	0.195916, 0.799781, -0.016612, 0.002386, 0.911578, -0.41111...
Point 2	-0.105097, 0.791910, -0.016076, 0.002387, 0.911614, -0.4110...
Point 3	0.189845, 0.656790, -0.009078, 0.069971, 0.842482, -0.53290...
Board	-0.375977, 0.656754, 0.848326, 0.012281, 0.915886, -0.40118...

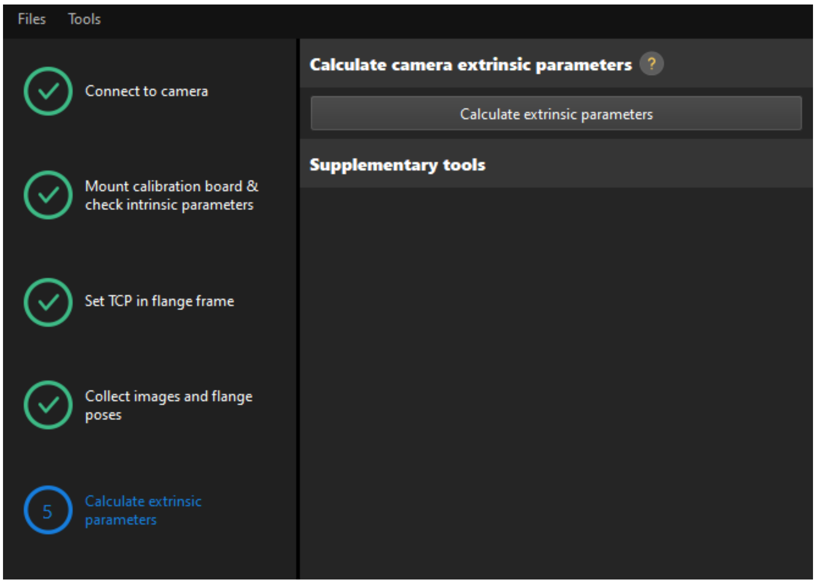
Update data

☑ Save images

The calibration program requires that at least three points not in a line should be touched for calculating extrinsic parameters.

Calculate Extrinsic Parameters

1. In the **Calculate extrinsic parameters** step, click the **[Calculate extrinsic parameters]** button.

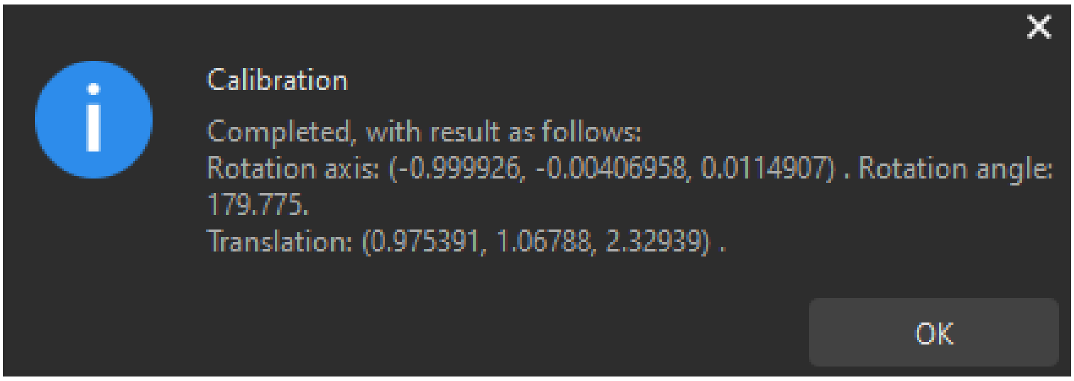


Files
Tools
Connect to camera
Mount calibration board & check intrinsic parameters
Set TCP in flange frame
Collect images and flange poses
5
Calculate extrinsic parameters
?
Calculate extrinsic parameters

Calculate camera extrinsic parameters

Supplementary tools

2. In the prompted window indicating calibration success, click the **[OK]** button.

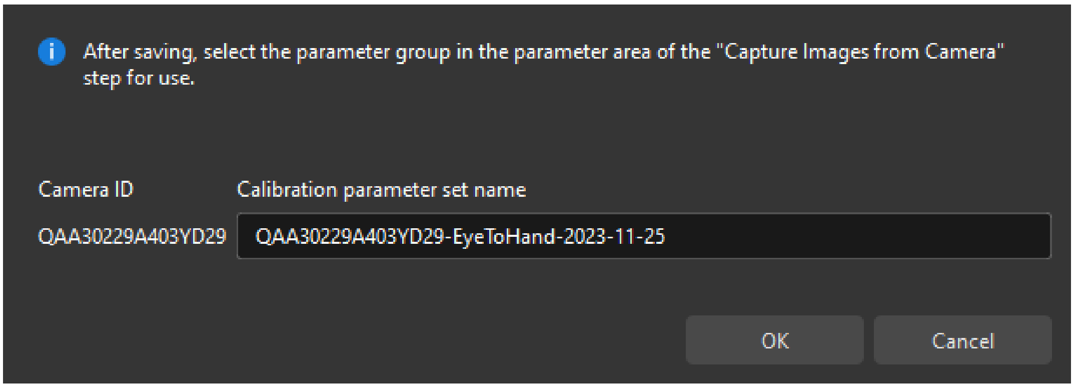


Calibration

Completed, with result as follows:
Rotation axis: (-0.999926, -0.00406958, 0.0114907) . Rotation angle: 179.775.
Translation: (0.975391, 1.06788, 2.32939) .

- If you want to enhance the calibration accuracy, please refer to the section Calibration Result Check and Analysis.
- To evaluate the robot absolute accuracy and the accuracy of calculated extrinsic parameters, you can click **[Robot absolute accuracy and repeatability]** and **[Extrinsic parameter accuracy]** in the “Calculate extrinsic parameters” window and use the error analysis tool to perform robot absolute accuracy check and analyze extrinsic parameter error in EIH setup.

3. Click the **[Save]** button on the bottom bar. In the prompted **Save Calibration Files** dialog box, click the **[OK]** button. The camera calibration result will be automatically saved in the “calibration” directory of the project.



After saving, select the parameter group in the parameter area of the "Capture Images from Camera" step for use.

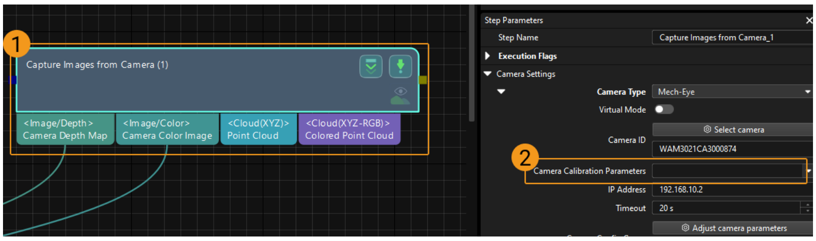
Camera ID	Calibration parameter set name
QAA30229A403YD29	QAA30229A403YD29-EyeToHand-2023-11-25

Till now, the calibration process is completed.

Change the Extrinsic Parameter File

If the camera’s extrinsic parameter file has been used by a Mech-Vision project, you need to change the extrinsic parameter file used by the Mech-Vision project to a new one after you finish the hand-eye calibration.

1. Select the **Capture Images from Camera** Step.



1
Capture Images from Camera (1)
<Image/Depth> Camera Depth Map
<Image/Color> Camera Color Image
<Cloud(XYZ)> Point Cloud
<Cloud(XYZ-RGB)> Colored Point Cloud
2
Step Parameters
Execution Flags
Camera Settings
Select camera
Adjust camera parameters

Step Name: Capture Images from Camera_1
Camera Type: Mech-Eye
Virtual Mode
Camera ID: WAM3021CA3000874
Camera Calibration Parameters
IP Address: 192.168.10.2
Timeout: 20 s

2. In the **Step Parameters** panel, click ▼ of the **Camera Calibration Parameters** parameter, and select the newly calibrated extrinsic parameter file.

4.5.3.6. Manual Calibration in the Eye-in-Hand Setup (Four-Axis Robots - Using TCP Touch Method)

This how-to guide introduces how to complete the manual calibration for four-axis robots using the TCP touch method in the eye-in-hand (EIH) setup.

Preparation before Calibration

Before hand-eye calibration, you need to finish the following preparations:

- Construct the Mech-Mind Vision System.
- Prepare the materials required for calibration.
- Adjust the effect of calibration board's point cloud.

Construct the Vision System

Construct the Mech-Mind Vision System by referring to the section Vision System Hardware Setup.

You need to use Mech-Eye Viewer, Mech-Vision and Mech-Viz during hand-eye calibration. Please ensure that they have been installed and are running the latest versions.

Prepare the Materials Required for Calibration

The manual calibration in the EIH setup needs to use the calibration board and sharp tip.

Please prepare the calibration board according to the following requirements:

- Ensure that the circles of the calibration board are clearly visible and without obvious scratches, and the board does not suffer from deformations.
- In the EIH setup, please place the calibration board in the center of the working plane, where target objects are to be placed.

The sharp tip should be mounted on the center of the robot flange. If an undetachable gripper is connected to the robot flange, you can attach the sharp point directly to the gripper.

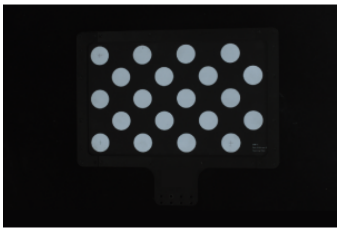
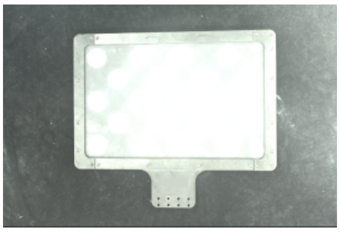
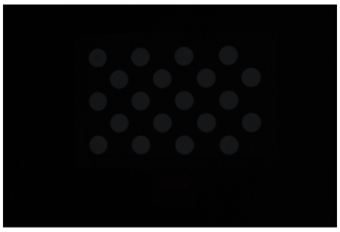
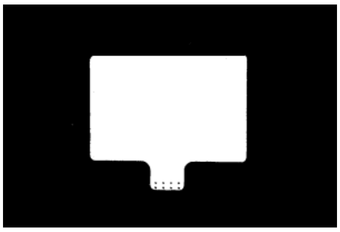
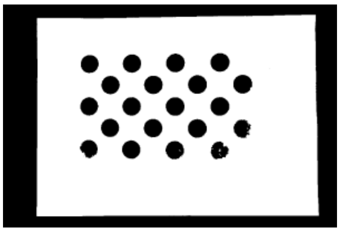
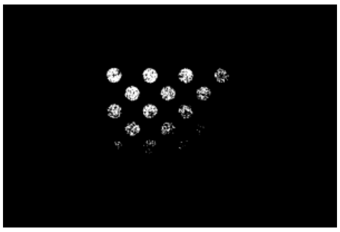
In addition, before calibration, move the robot to the starting point for calibration.

Adjust the Effect of Calibration Board's Point Cloud

1. Open Mech-Eye Viewer to adjust camera parameters.
2. Adjust the 2D parameters to make sure that the 2D image of the calibration board is clear and neither overexposed nor underexposed.
3. Adjust 3D parameters to make sure that the point clouds of the circles on the calibration board are complete and have clear contours. It is recommended to set **Surface Smoothing** and **Outlier Removal** to **Normal** in the **Point Cloud Processing** section to reduce the point cloud fluctuation range.

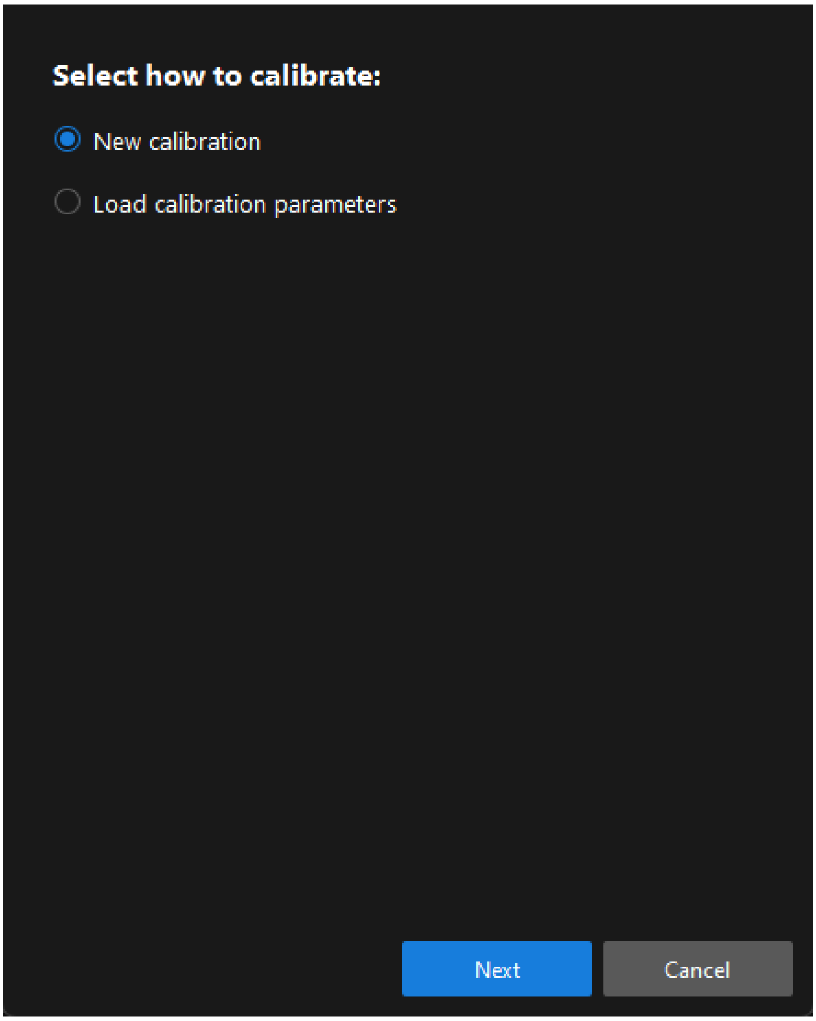
   > If the on-site ambient lights are not ideal and affect the quality of 2D images and point clouds, you can use shading or supplemental light to improve the lighting conditions.

4. Make sure that the images and point cloud of the calibration board are up to standard after completing preceding steps.

	Normal	Overexposed	Underexposed
2D image			
Point cloud			

Pre-calibration Configuration

1. Open Mech-Vision, and click the [**Camera Calibration (Standard)**] button in the toolbar.The **Configuration before Calibration** window will be prompted.
2. In the **Select how to calibrate** window, select the **New calibration** radio button, and then click the [**Next**] button.

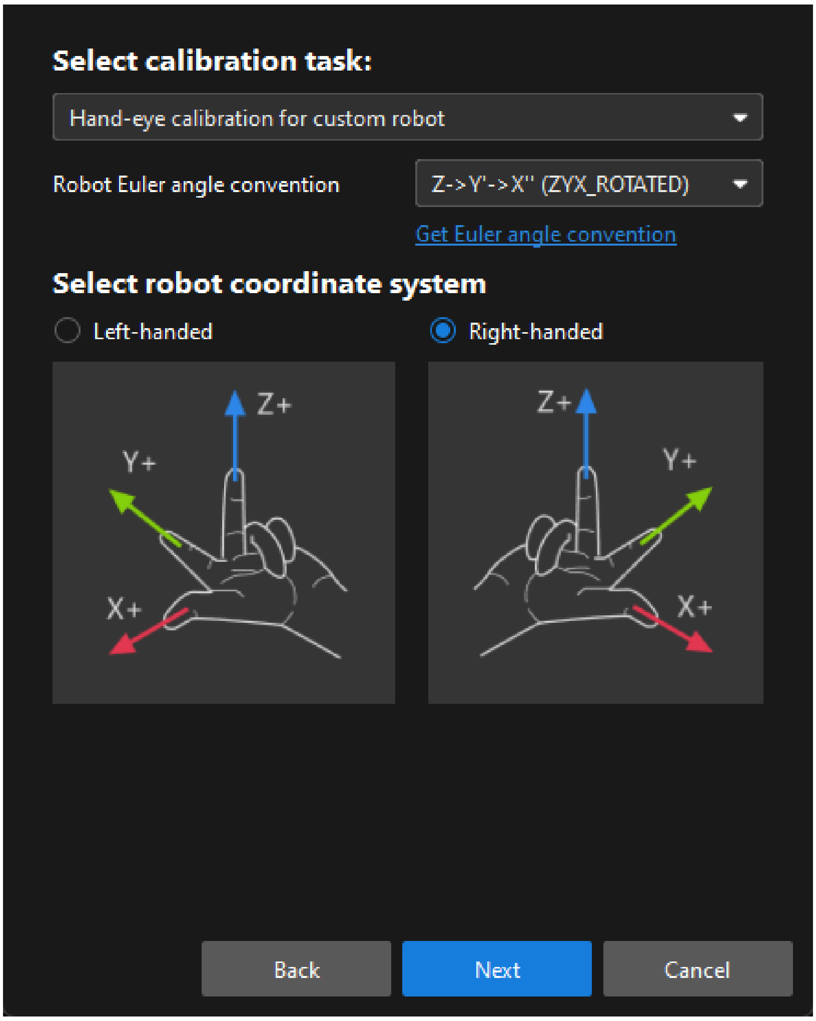


Select how to calibrate:

● New calibration

○ Load calibration parameters

3. In the **Select calibration task** window, select **Hand-eye calibration for custom robot** from the drop-down list box, specify the **Robot Euler angle convention** parameter, select the robot coordinate system type, and then click the **[Next]** button.



Select calibration task:

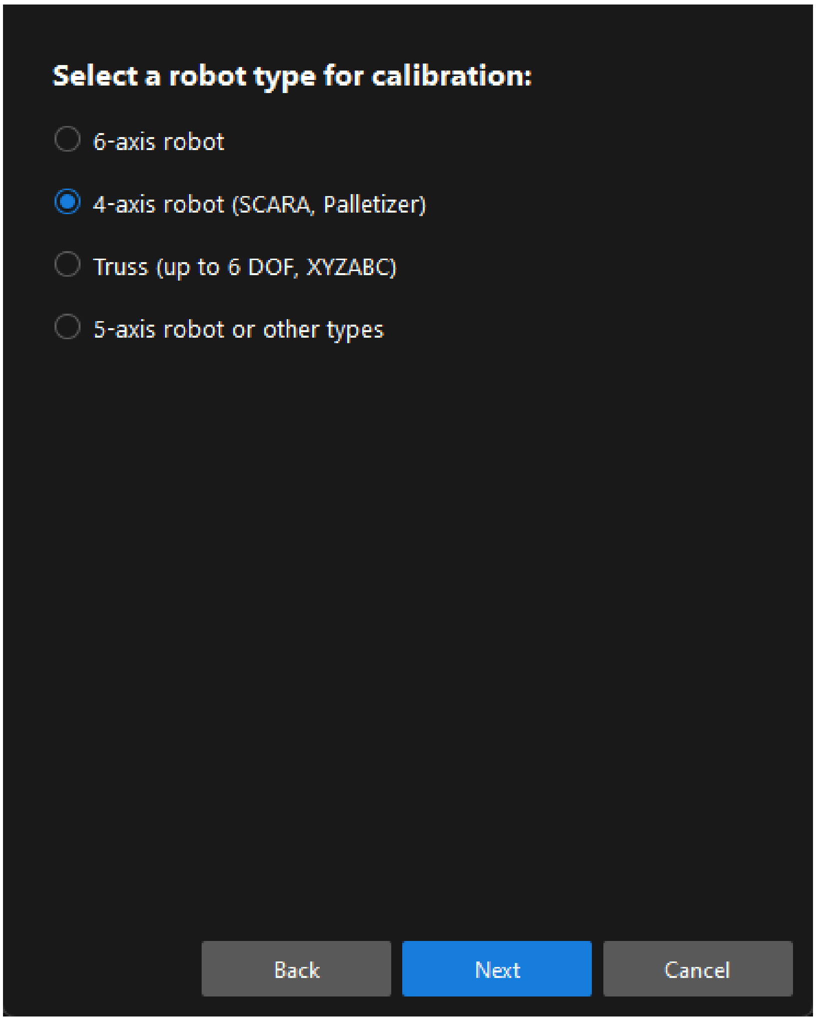
Hand-eye calibration for custom robot

Robot Euler angle convention: Z->Y'->X'' (ZYX_ROTATED)

Select robot coordinate system

○ Left-handed ● Right-handed

4. In **Select a robot type for calibration** window, select the **4-axis robot (SCARA, Palletizer)** radio button, and then click the **[Next]** button.



Select a robot type for calibration:

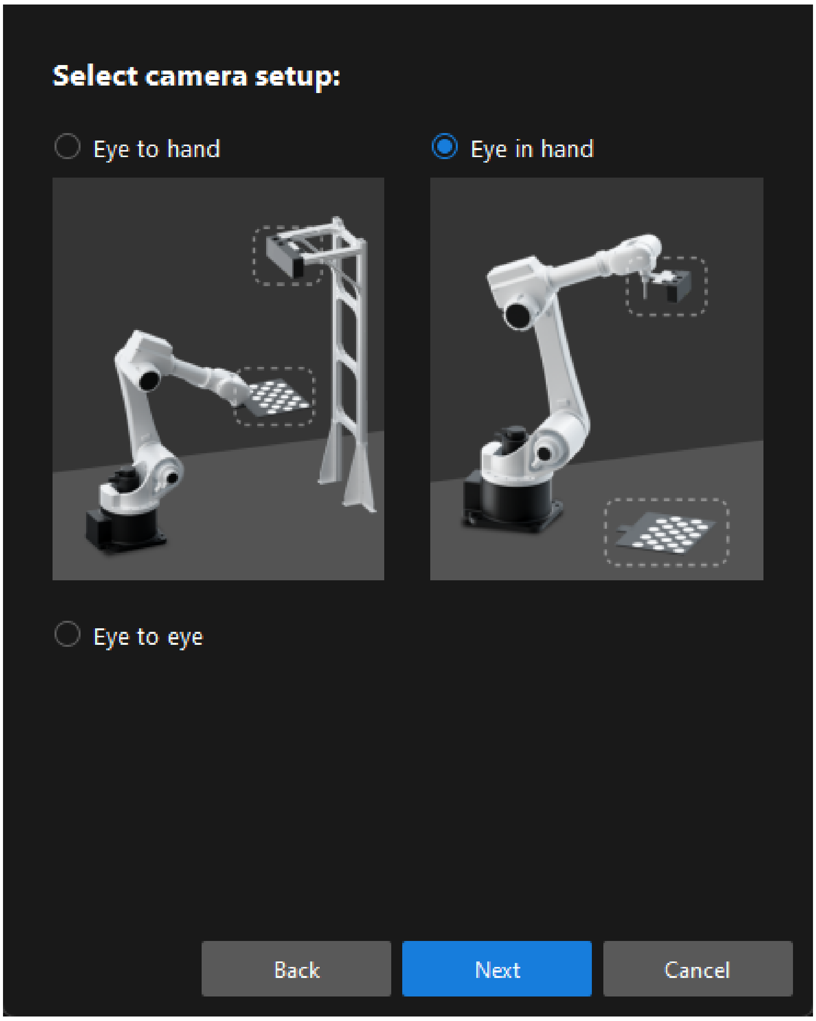
○ 6-axis robot

● 4-axis robot (SCARA, Palletizer)

○ Truss (up to 6 DOF, XYZABC)

○ 5-axis robot or other types

5. In the **Select camera setup** window, select the **Eye in hand** radio button, and then click the **[Next]** button.



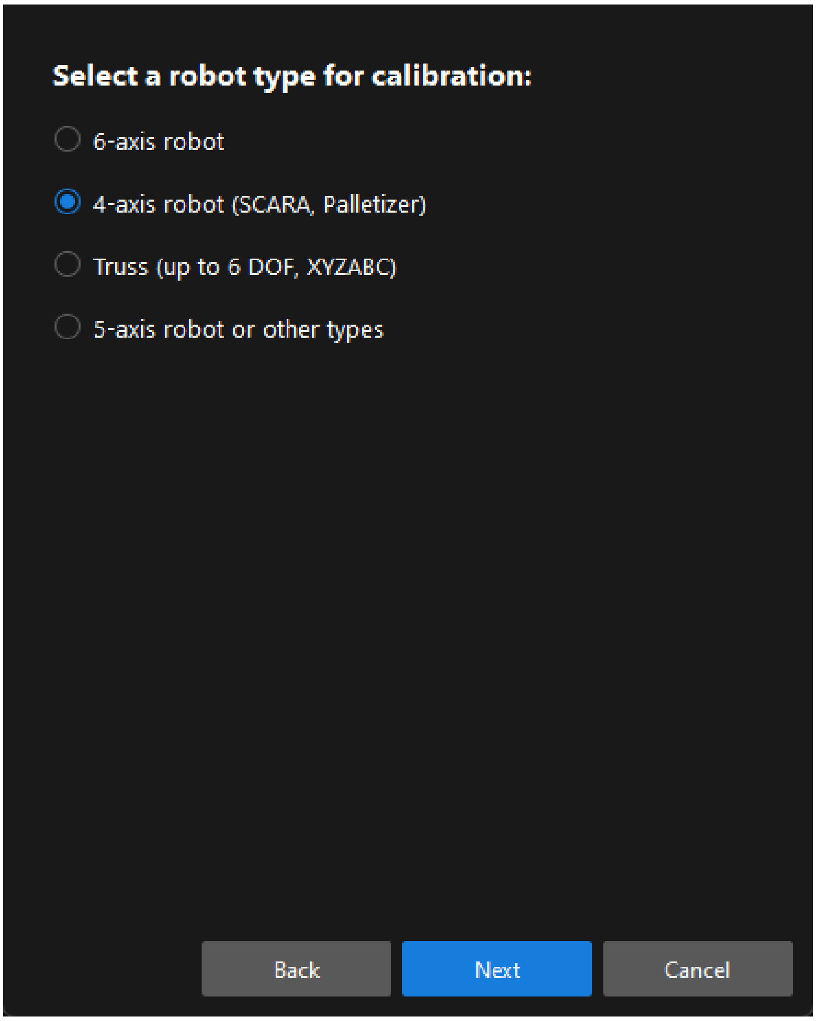
Select camera setup:

○ Eye to hand

● Eye in hand

○ Eye to eye

6. In the **Select how to collect data** window, select the **TCP touch** radio button, and then click the **[Calibrate]** button.The **Calibration (Eye in Hand) window** will be prompted.



Select a robot type for calibration:

○ 6-axis robot

● 4-axis robot (SCARA, Palletizer)

○ Truss (up to 6 DOF, XYZABC)

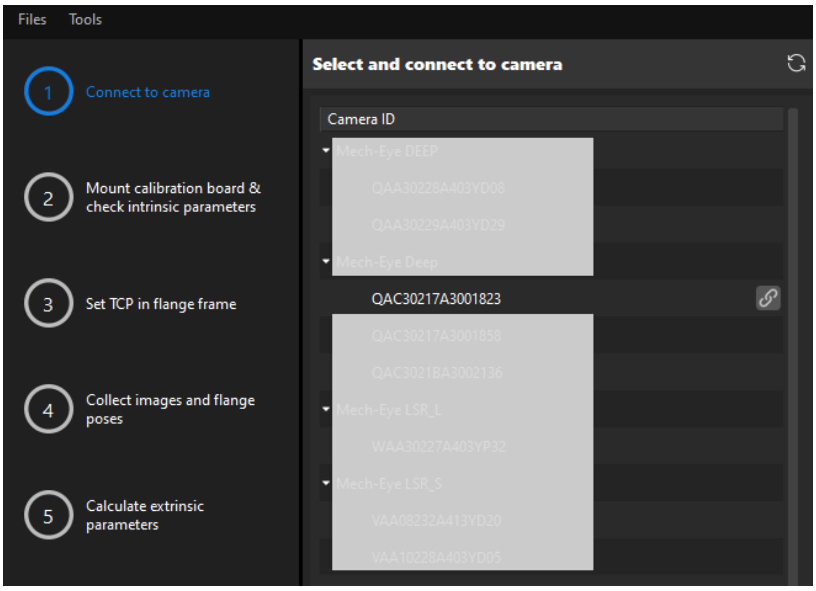
○ 5-axis robot or other types

Till now, you have completed the pre-calibration configuration and can start the calibration procedure.

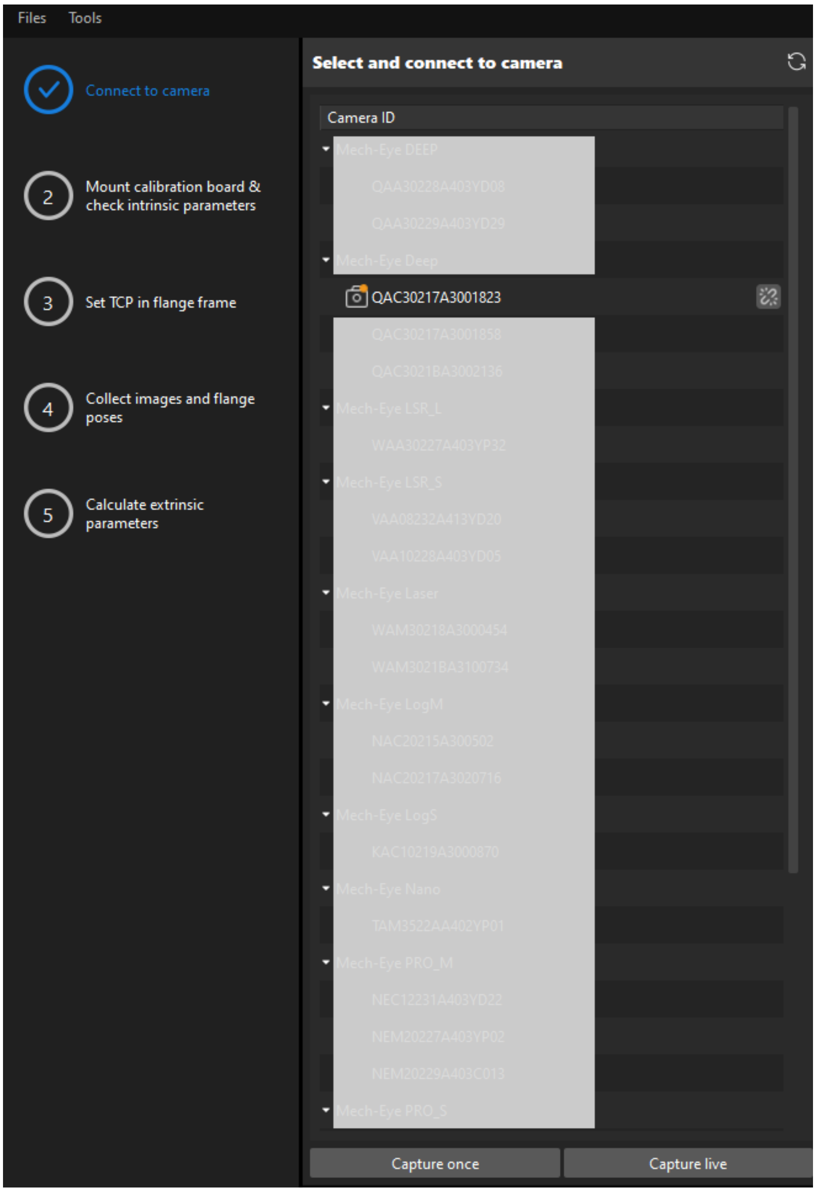
Calibration Procedure

Connect to the Camera

1. In the **Connect to camera** step, select the camera to connect in the Camera ID list, and then click the 🔗 button or double-click the camera entry to connect to it.


Files
Tools
1
Connect to camera
2
Mount calibration board & check intrinsic parameters
3
Set TCP in flange frame
4
Collect images and flange poses
5
Calculate extrinsic parameters
Select and connect to camera
Camera ID
[redacted]
QAC30217A3001823
[redacted]

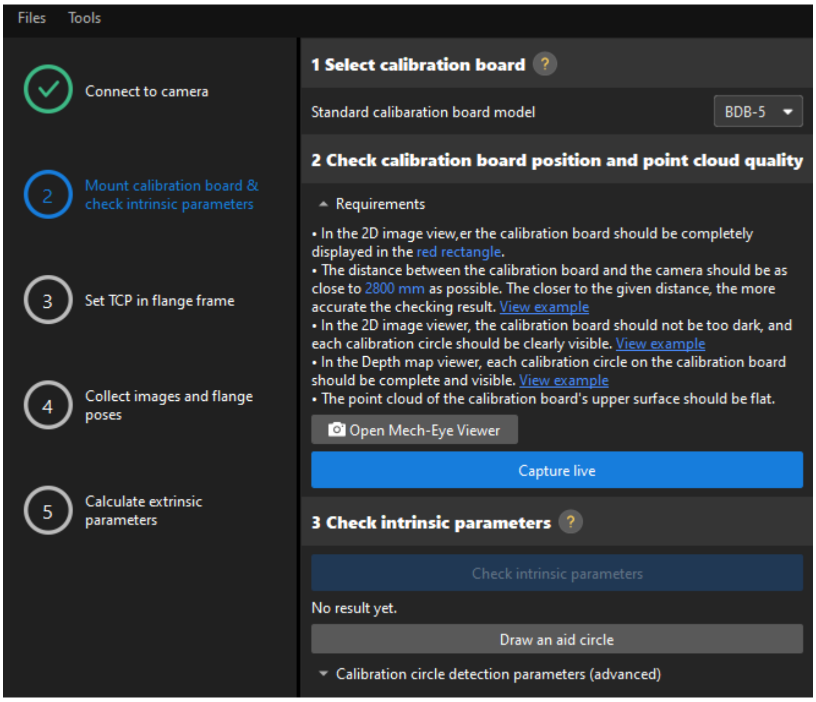
2. After the camera is connected, click the [Capture once] or [Capture live] button.


Files
Tools
Connect to camera
2
Mount calibration board & check intrinsic parameters
3
Set TCP in flange frame
4
Collect images and flange poses
5
Calculate extrinsic parameters
Select and connect to camera
Camera ID
[redacted]
QAC30217A3001823
[redacted]
Capture once
Capture live

3. In the **Image viewer** panel, ensure that the camera can capture images normally and click the **[Next]** button on the bottom bar.

Mount Calibration Board & Check Intrinsic Parameters

1. In the **Mount calibration board & check intrinsic parameters** step, select the calibration board model for the **Standard calibration board model** parameter according to its model nameplate in the **1 Select calibration board** area.



Files
Tools
Connect to camera
2
Mount calibration board & check intrinsic parameters
3
Set TCP in flange frame
4
Collect images and flange poses
5
Calculate extrinsic parameters
?
Open Mech-Eye Viewer
Capture live
?
Check intrinsic parameters
Draw an aid circle

1 Select calibration board

Standard calibaration board model: BDB-5

2 Check calibration board position and point cloud quality

Requirements

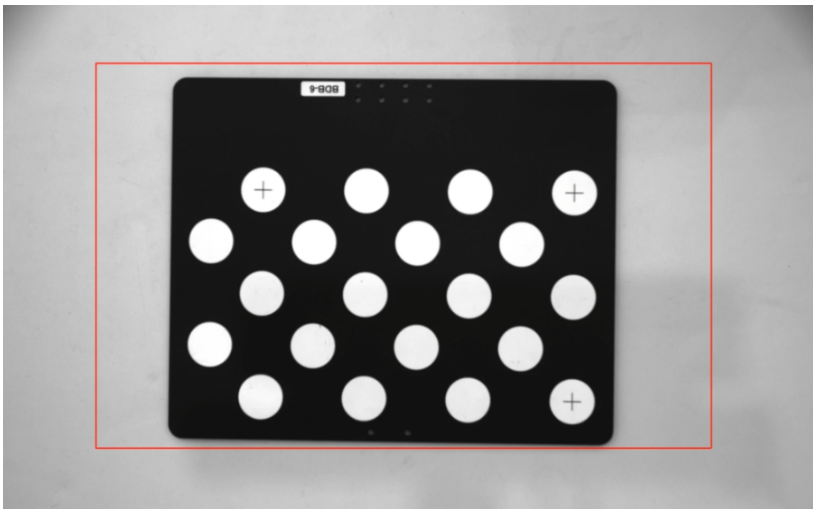
• In the 2D image view,er the calibration board should be completely displayed in the red rectangle.
• The distance between the calibration board and the camera should be as close to 2800 mm as possible. The closer to the given distance, the more accurate the checking result. View example
• In the 2D image viewer, the calibration board should not be too dark, and each calibration circle should be clearly visible. View example
• In the Depth map viewer, each calibration circle on the calibration board should be complete and visible. View example
• The point cloud of the calibration board's upper surface should be flat.

3 Check intrinsic parameters

No result yet.

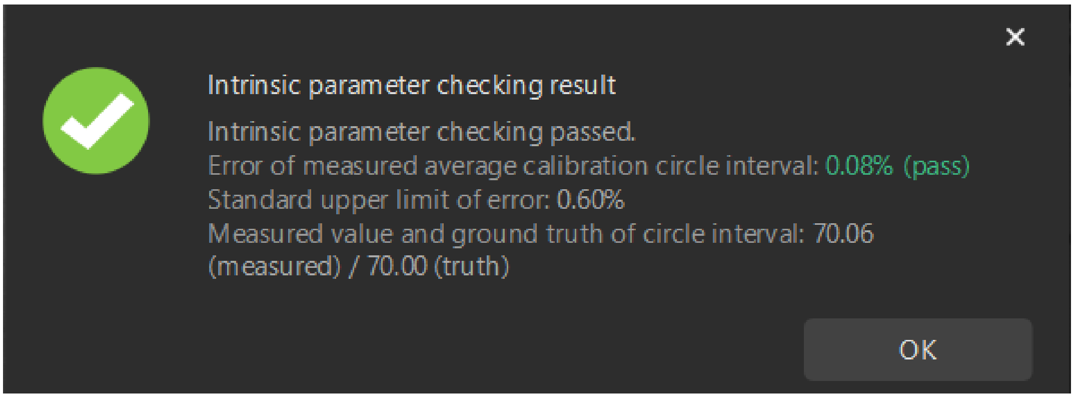
Calibration circle detection parameters (advanced)

2. In the **2 Check calibration board position and point cloud quality** area, read carefully the requirements on the calibration board position and point cloud quality, and then click the **[Capture live]** button.The **[Capture live]** button will turn into **[Stop capturing and detect position]**.
3. Manually move the calibration board to the proper position (in the red rectangle), and ensure that the 2D image and depth map of the calibration board meet the requirements, and then click the **[Stop capturing and detect position]** button.



> If the captured images do not meet the requirements, click the [**Open Mech-Eye Viewer**] button to open the Mech-Eye Viewer software, adjust the 2D and 3D exposure parameters and re-capture images. Please note that you need to change the **Parameter group** parameter to “calib” first.

4. In the **3 Check intrinsic parameters** area, click the [**Check intrinsic parameters**] button.
5. Confirm the results of the camera intrinsic parameter check.
   - If the camera intrinsic parameter check passes, click the [**OK**] button in the prompted window, and then click the [**Next**] button on the bottom bar.



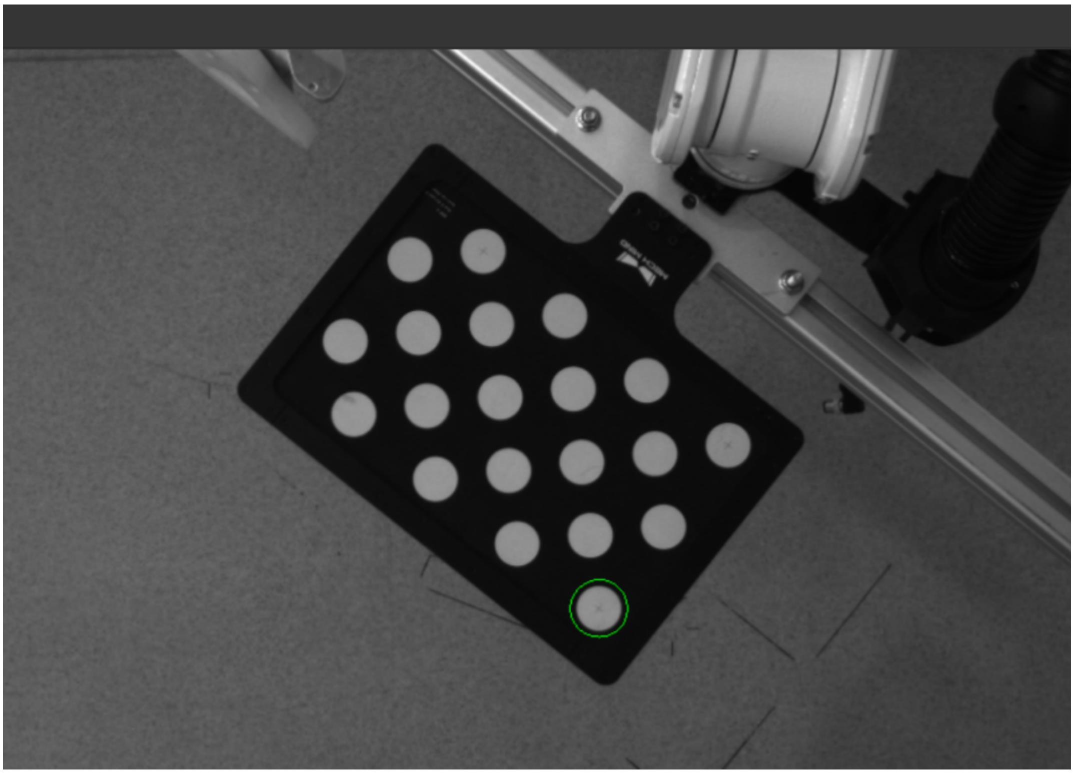
Intrinsic parameter checking result

Intrinsic parameter checking passed.
Error of measured average calibration circle interval: 0.08% (pass)
Standard upper limit of error: 0.60%
Measured value and ground truth of circle interval: 70.06 (measured) / 70.00 (truth)

   - If the camera intrinsic parameter check fails, adjust detection parameters by drawing aid circles or manually adjusting the calibration circle detection parameters, and then click the [**Recheck intrinsic parameters**] button.

Draw an Aid Circle

1. To draw an aid circle, click the [**Draw an aid circle**] button.
2. In the right **Image viewer** panel, right-click the calibration board image and clear the **Fit to window** checkbox, and press the `Ctrl` key and drag the roller to adjust the image to a suitable size.
3. Move the mouse pointer to the cross center point of the calibration circle, press the left mouse button and make the aid circle completely include the calibration circle and then release it.



4. Click the **[Recheck intrinsic parameters]** button, and confirm that the camera intrinsic parameter check passes.

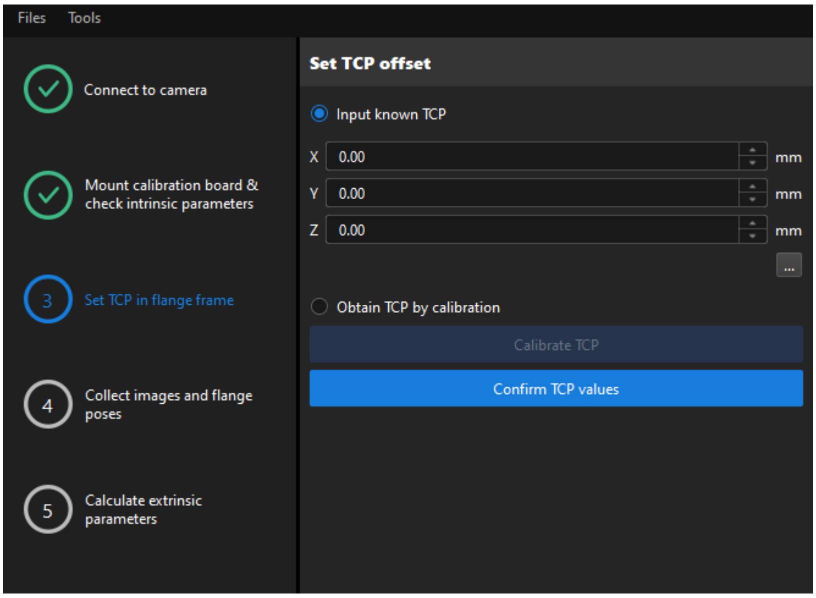
Manually Adjust Detection Parameters

To manually adjust detection parameters, click **[Calibration circle detection parameters (advanced)]**, adjust parameter settings and click the **[Recheck intrinsic parameters]** to confirm whether the camera intrinsic parameter check passes.

If the calibration circles are still not detected, please adjust the camera settings according to the on-site operating conditions.

Set TCP in Flange Frame

1. In the **Set TCP in flange frame** step, select the **Input known TCP** radio button, and enter the TCP values.


Files
Tools
Connect to camera
Mount calibration board & check intrinsic parameters
3
Set TCP in flange frame
4
Collect images and flange poses
5
Calculate extrinsic parameters
...
Calibrate TCP
Confirm TCP values

Set TCP offset

● Input known TCP

X 0.00 mm

Y 0.00 mm

Z 0.00 mm

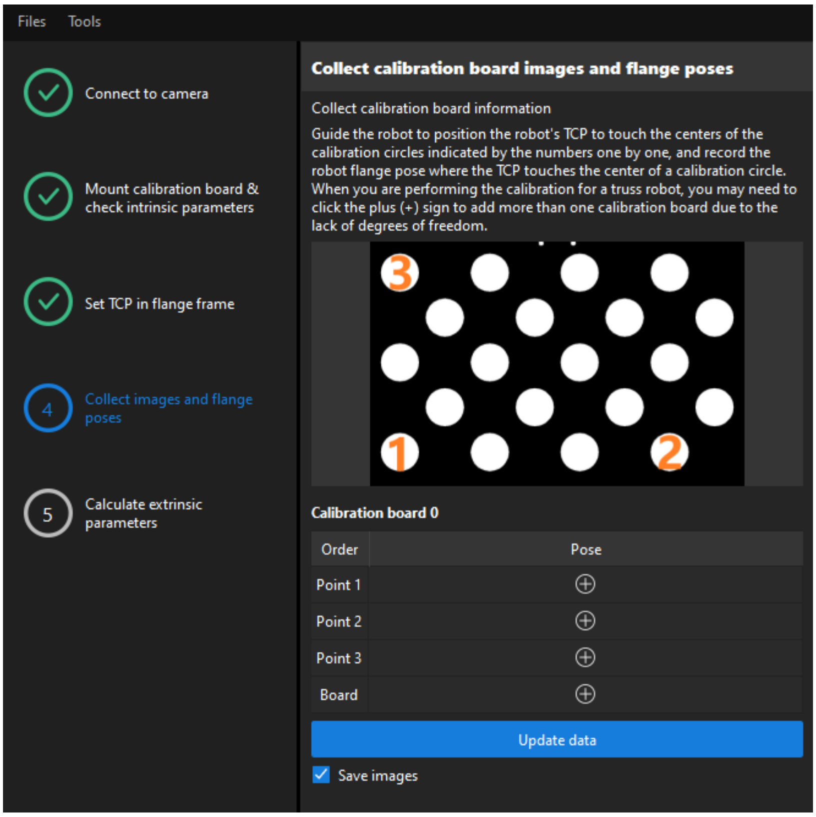
○ Obtain TCP by calibration

2. Click the [**Confirm TCP values**] button.
3. Click the [**Next**] button on the bottom bar.

> If the TCP values are unknown, select the **Obtain TCP by calibration** radio button, and then click the [**Calibrate TCP**] button. You can use the **TCP calibration** tool to calculate the TCP values.

Collect Images and Flange Poses

1. In the **Collect images and flange poses** step, select the **Save images** checkbox.


Files
Tools
Connect to camera
Mount calibration board & check intrinsic parameters
Set TCP in flange frame
4
Collect images and flange poses
5
Calculate extrinsic parameters
3
1
2
Update data

Collect calibration board images and flange poses

Collect calibration board information

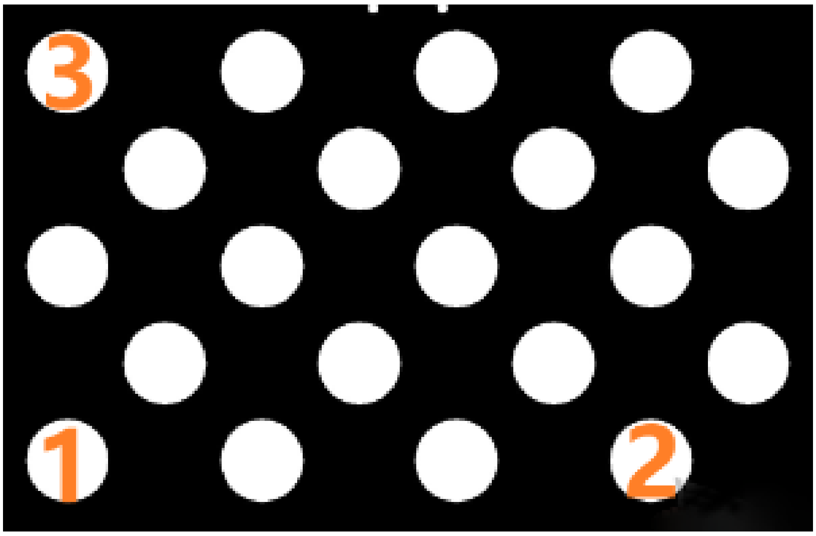
Guide the robot to position the robot's TCP to touch the centers of the calibration circles indicated by the numbers one by one, and record the robot flange pose where the TCP touches the center of a calibration circle. When you are performing the calibration for a truss robot, you may need to click the plus (+) sign to add more than one calibration board due to the lack of degrees of freedom.

Calibration board 0

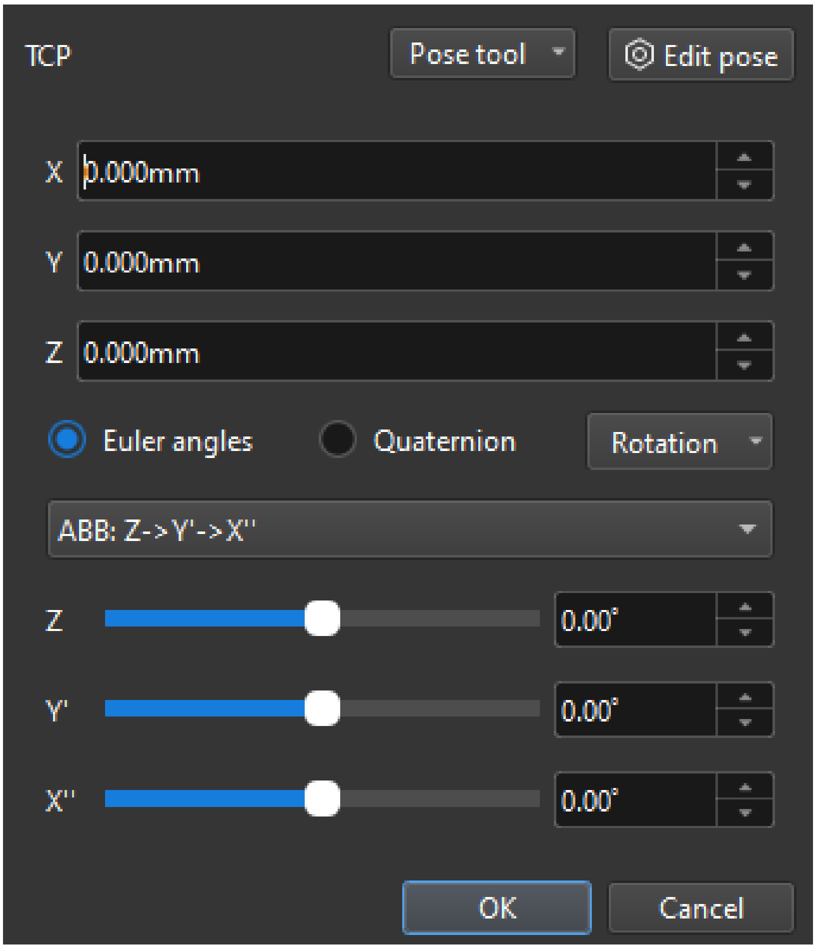
Order	Pose
Point 1	⊕
Point 2	⊕
Point 3	⊕
Board	⊕

☑ Save images

2. Control the robot to let its end tip touch the cross center point of calibration circle “point 1” on the calibration board, and record the robot flange pose in the teach pendant.



3. Click the ⊕ button of point 1, and enter the robot flange pose in the prompted **Input Flange Pose of Robot** dialog box, and then click the [OK] button.



X 0.000mm

Y 0.000mm

Z 0.000mm

● Euler angles ○ Quaternion

ABB: Z->Y'->X''

Z 0.00°

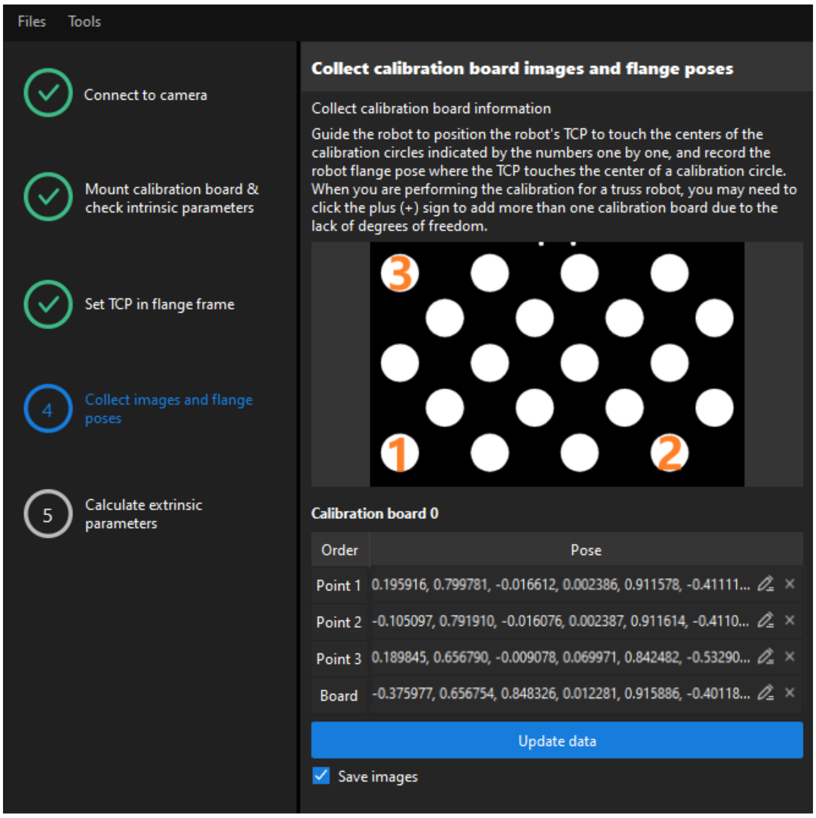
Y' 0.00°

X'' 0.00°

4. Repeat the preceding steps, and let the end tip touch the cross center points of point 2 and point 3, and enter robot flange poses.

5. Click the ⊕ button of the current calibration board to capture the images of it.

6. Click the [**Update data**] button and then click the [**Next**] button on the bottom bar.


Files
Tools
Connect to camera
Mount calibration board & check intrinsic parameters
Set TCP in flange frame
4
Collect images and flange poses
5
Calculate extrinsic parameters
3
1
2
Update data

Collect calibration board images and flange poses

Collect calibration board information

Guide the robot to position the robot's TCP to touch the centers of the calibration circles indicated by the numbers one by one, and record the robot flange pose where the TCP touches the center of a calibration circle. When you are performing the calibration for a truss robot, you may need to click the plus (+) sign to add more than one calibration board due to the lack of degrees of freedom.

Calibration board 0

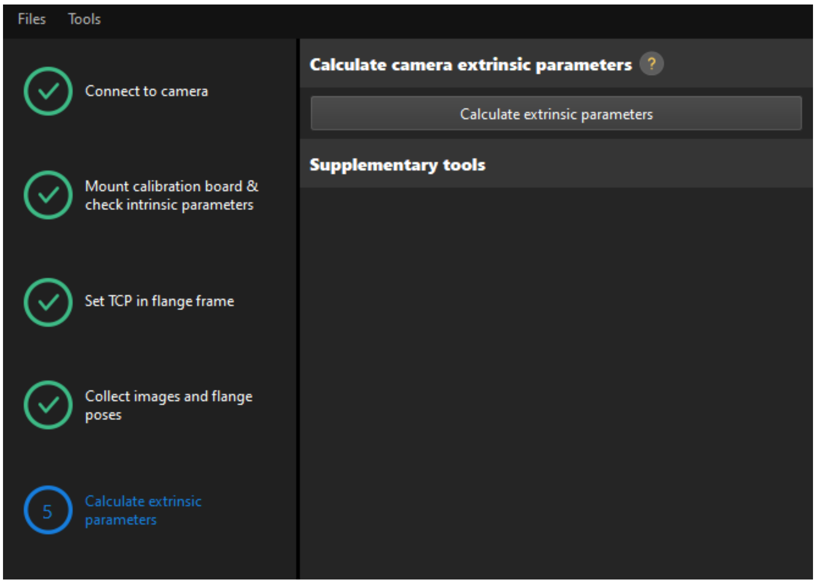
Order	Pose
Point 1	0.195916, 0.799781, -0.016612, 0.002386, 0.911578, -0.41111...
Point 2	-0.105097, 0.791910, -0.016076, 0.002387, 0.911614, -0.4110...
Point 3	0.189845, 0.656790, -0.009078, 0.069971, 0.842482, -0.53290...
Board	-0.375977, 0.656754, 0.848326, 0.012281, 0.915886, -0.40118...

☑ Save images

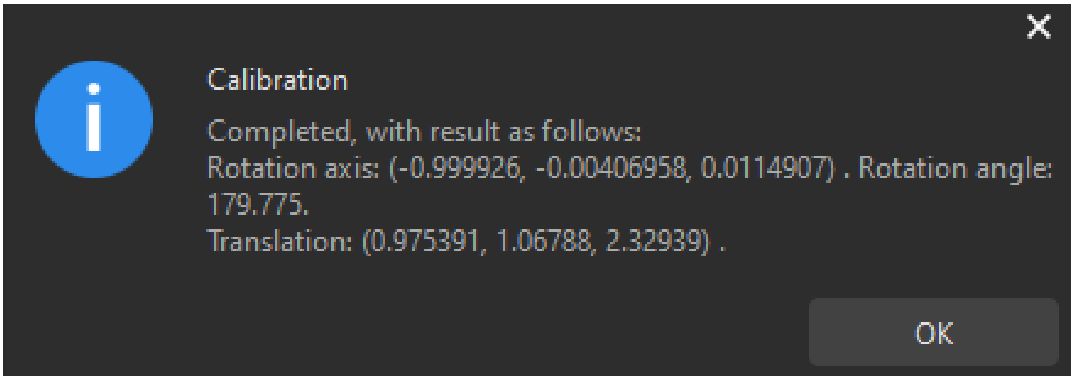
The calibration program requires that at least three points not in a line should be touched for calculating extrinsic parameters.

Calculate Extrinsic Parameters

1. In the **Calculate extrinsic parameters** step, click the [**Calculate extrinsic parameters**] button.


Files
Tools
Connect to camera
Mount calibration board & check intrinsic parameters
Set TCP in flange frame
Collect images and flange poses
5
Calculate extrinsic parameters
Calculate camera extrinsic parameters
?
Calculate extrinsic parameters
Supplementary tools

2. In the prompted window indicating calibration success, click the [**OK**] button.

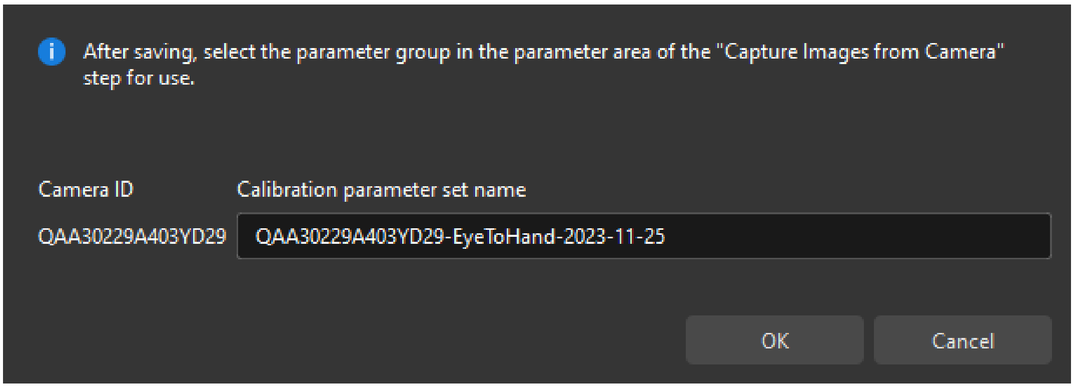


Calibration

Completed, with result as follows:
Rotation axis: (-0.999926, -0.00406958, 0.0114907) . Rotation angle: 179.775.
Translation: (0.975391, 1.06788, 2.32939) .

- If you want to enhance the calibration accuracy, please refer to the section Calibration Result Check and Analysis.
- To evaluate the robot absolute accuracy and the accuracy of calculated extrinsic parameters, you can click **[Robot absolute accuracy and repeatability]** and **[Extrinsic parameter accuracy]** in the “Calculate extrinsic parameters” window and use the error analysis tool to perform robot absolute accuracy check and analyze extrinsic parameter error in EIH setup.

3. Click the **[Save]** button on the bottom bar. In the prompted **Save Calibration Files** dialog box, click the **[OK]** button. The camera calibration result will be automatically saved in the “calibration” directory of the project.



After saving, select the parameter group in the parameter area of the "Capture Images from Camera" step for use.

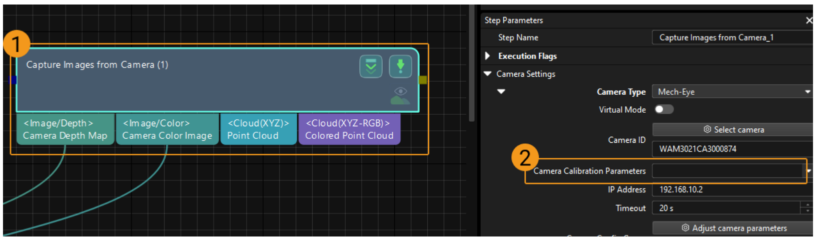
Camera ID	Calibration parameter set name
QAA30229A403YD29	QAA30229A403YD29-EyeToHand-2023-11-25

Till now, the calibration process is completed.

Change the Extrinsic Parameter File

If the camera’s extrinsic parameter file has been used by a Mech-Vision project, you need to change the extrinsic parameter file used by the Mech-Vision project to a new one after you finish the hand-eye calibration.

1. Select the **Capture Images from Camera** Step.



Step Name: Capture Images from Camera_1
Camera Type: Mech-Eye
Virtual Mode
Camera ID: WAM3021CA3000874
Camera Calibration Parameters
IP Address: 192.168.10.2
Timeout: 20 s

2. In the **Step Parameters** panel, click ▼ of the **Camera Calibration Parameters** parameter, and select the newly calibrated extrinsic parameter file.

Calibration-Related Status Codes

If the robot uses the Standard Interface to communicate with the Vision System during hand-eye calibration, the Vision System will return status codes in response data to indicate the execution status of the commands. The status codes indicate both the normal execution results and execution failures. You can perform further troubleshooting according to the status codes.

Calibration Error Codes

Error code	Description
7001	Calibration: Parameter error
7002	Calibration: Calibration flange pose not provided by Mech-Vision
7003	Calibration: Calibration joint positions not provided by Mech-Vision
7004	Calibration: Robot failed to reach calibration point

Calibration Normal Status Codes

Normal Status Code	Description
7100	Calibration: Robot reaches calibration point successfully
7101	Calibration: Pose received from Mech-Vision successfully

4.5.4. Hand-Eye Calibration for Truss/Gantry Robots

This how-to guide introduces how to complete the hand-eye calibration for truss robots (also called gantry robots).

For the concepts related to the hand-eye calibration for truss robots and the usage of the extrinsic parameter file, please refer to the section Explanation of the Calibration for Truss Robots.

Preparation before Calibration

Before hand-eye calibration, you need to finish the following preparations:

- Construct the Mech-Mind Vision System.
- Prepare the materials required for calibration.
- Adjust the effect of calibration board’s point cloud.

Construct the Vision System

Construct the Mech-Mind Vision System by referring to the section Vision System Hardware Setup.

You need to use Mech-Eye Viewer, Mech-Vision and Mech-Viz during hand-eye calibration. Please ensure that they have been installed and are running the latest versions.

Prepare the Materials Required for Calibration

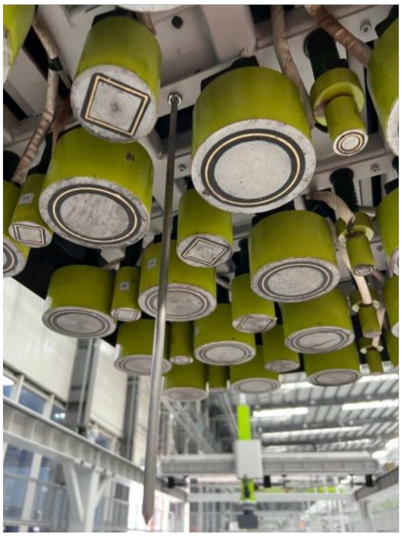
The hand-eye calibration for truss robots needs to use the following materials:

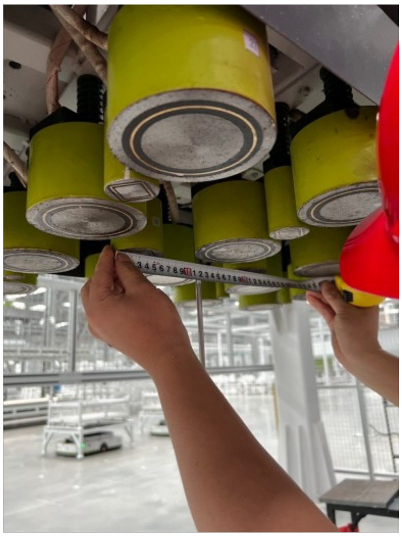
- Calibration board
- Sharp tip
- Tape measure

Please prepare the calibration board according to the following requirements:

- Ensure that the circles of the calibration board are clearly visible and without obvious scratches, and the board does not suffer from deformations.
- In the calibration for truss robot scenario, please place the calibration board in the center of the object plane, where target objects are to be placed.

The sharp tip should be mounted on the center of the truss robot's gripper. The tape measure is used to measure the deviation from the sharp point to the center of the robot flange in X, Y, and Z directions.



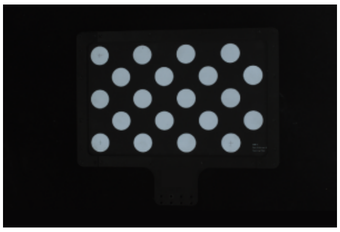
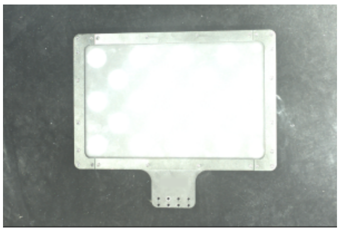
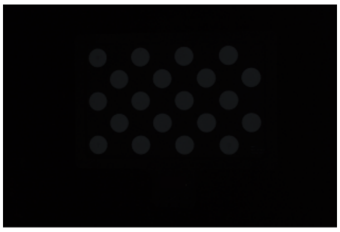
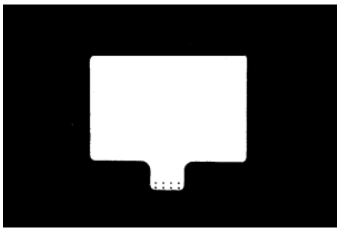
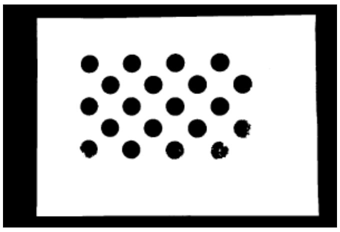
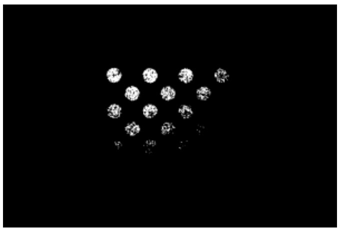


Adjust the Effect of Calibration Board's Point Cloud

1. Open Mech-Eye Viewer to adjust camera parameters.
2. Adjust the 2D parameters to make sure that the 2D image of the calibration board is clear and neither overexposed nor underexposed.
3. Adjust 3D parameters to make sure that the point clouds of the circles on the calibration board are complete and have clear contours. It is recommended to set **Surface Smoothing** and **Outlier Removal** to **Normal** in the **Point Cloud Processing** section to reduce the point cloud fluctuation range.

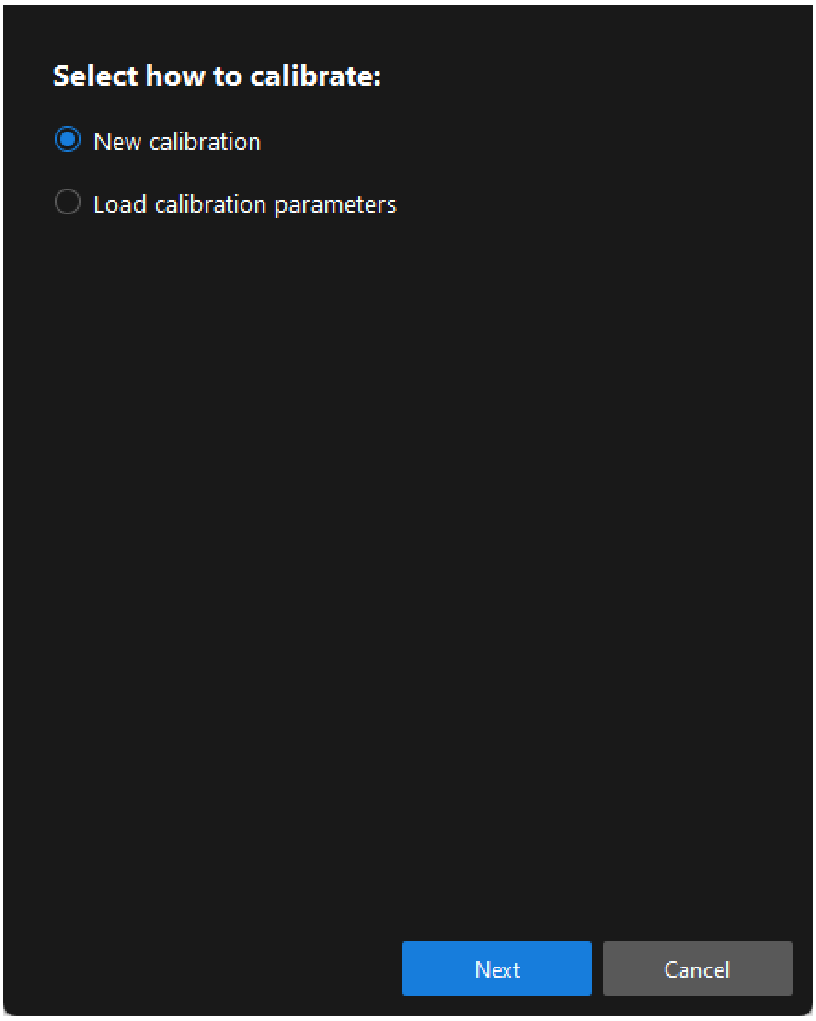
   > If the on-site ambient lights are not ideal and affect the quality of 2D images and point clouds, you can use shading or supplemental light to improve the lighting conditions.

4. Make sure that the images and point cloud of the calibration board are up to standard after completing preceding steps.

	Normal	Overexposed	Underexposed
2D image			
Point cloud			

Pre-calibration Configuration

1. Open Mech-Vision, and click the [**Camera Calibration (Standard)**] button in the toolbar. The **Configuration before Calibration** window will be prompted.
2. In the **Select how to calibrate** window, select the **New calibration** radio button, and then click the [**Next**] button.

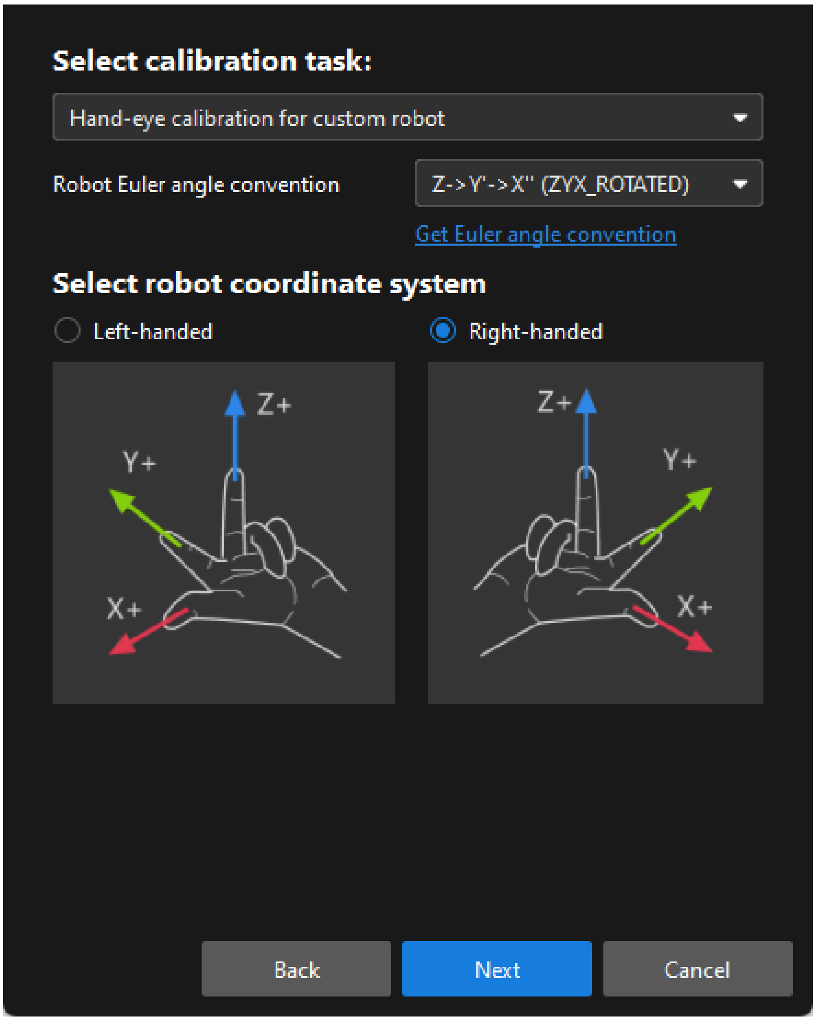


Select how to calibrate:

● New calibration

○ Load calibration parameters

3. In the **Select calibration task** window, select **Hand-eye calibration for custom robot** from the drop-down list box, specify the **Robot Euler angle convention** parameter, select the robot coordinate system type, and then click the **[Next]** button.



Select calibration task:

Hand-eye calibration for custom robot

Robot Euler angle convention: Z->Y'->X'' (ZYX_ROTATED)

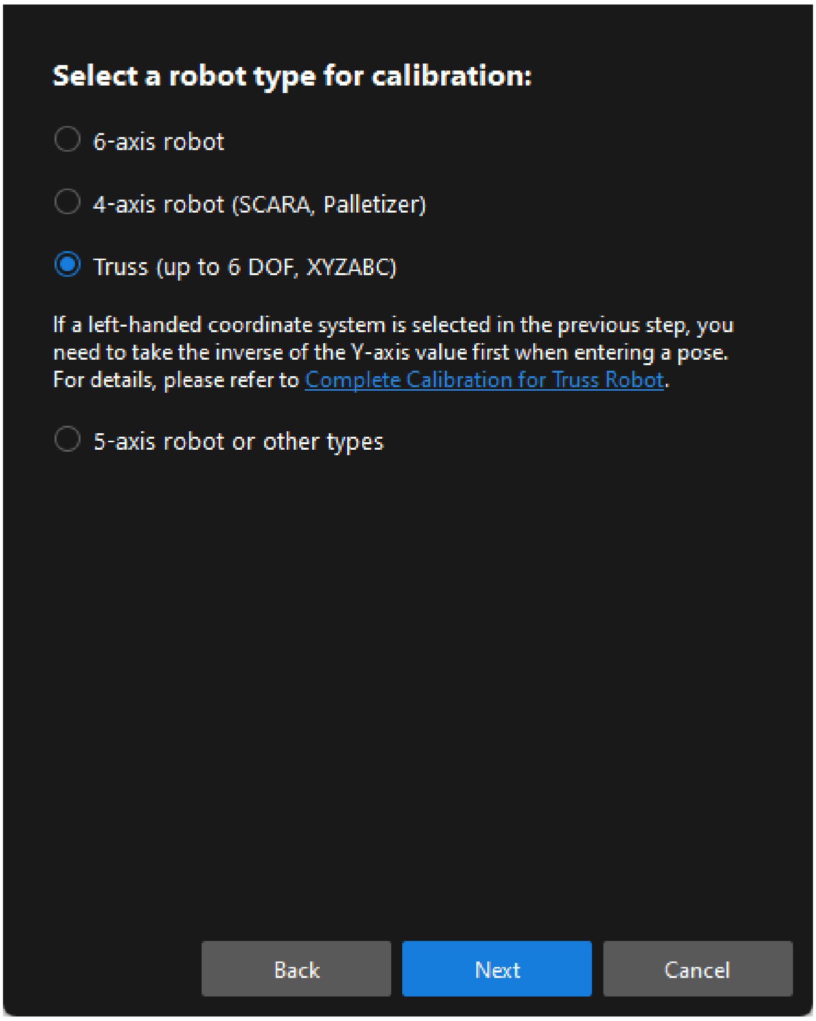
Get Euler angle convention

Select robot coordinate system

○ Left-handed ● Right-handed

- You can determine robot coordinate system type according to the section Left- and Right-handed Coordinate Systems. If the truss robot uses the left-handed coordinate system, you just need to select Left-handed here.
- When the vision system communicates with the robot, the communication module will automatically transform the coordinate system for the sent and received poses. When the communication module receives the robot pose from the robot side, it transforms the robot pose in the left-handed coordinate system to the one in the right-handed coordinate system; when it sends the pose output by the software to the robot side, it transforms the robot pose in the right-handed coordinate system to the one in the left-handed coordinate system.
- When the robot flange pose is required to be input during the calibration procedure, you need to take the inverse of the Y-axis value first.

4. In the **Select a robot type for calibration** window, select the **Truss (up to 6 DOF, XYZABC)**, and then click the **[Next]** button.



Select a robot type for calibration:

○ 6-axis robot

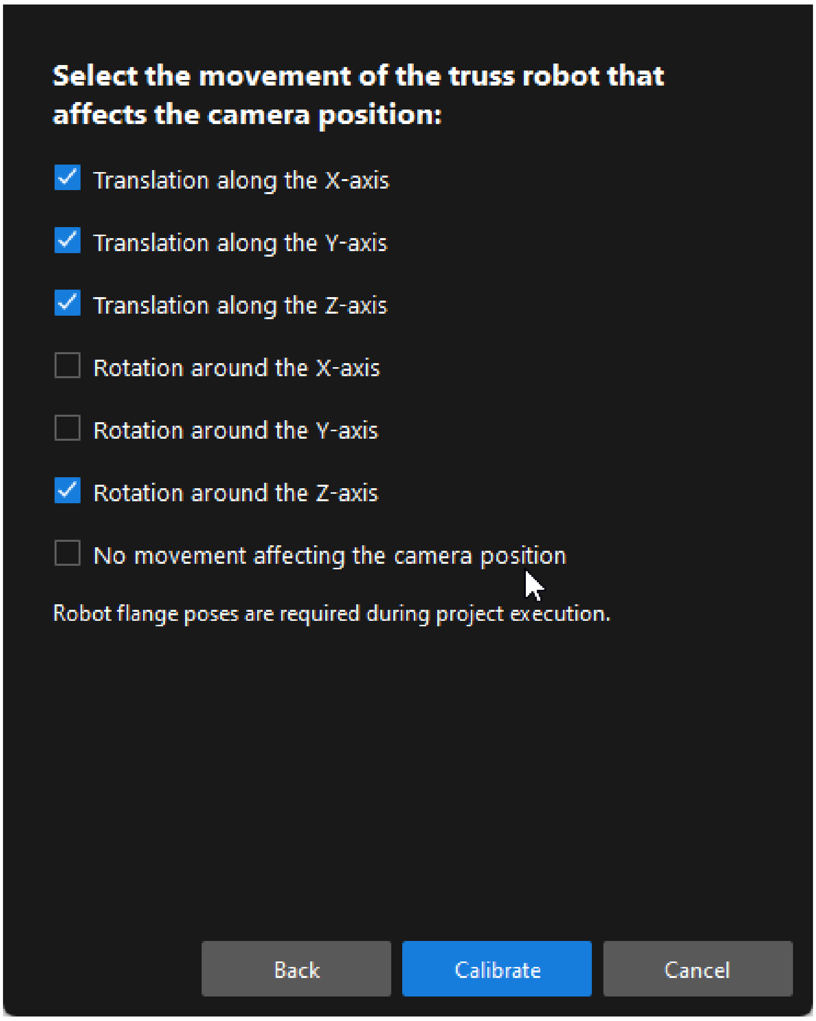
○ 4-axis robot (SCARA, Palletizer)

● Truss (up to 6 DOF, XYZABC)

If a left-handed coordinate system is selected in the previous step, you need to take the inverse of the Y-axis value first when entering a pose. For details, please refer to Complete Calibration for Truss Robot.

○ 5-axis robot or other types

5. In the **Select axes affecting camera** window, select the axes as required (such as **X-axis**, **Y-axis**, **Z-axis**, and **Rz**), and then click the **[Calibrate]** button.



Select the movement of the truss robot that affects the camera position:

☑ Translation along the X-axis

☑ Translation along the Y-axis

☑ Translation along the Z-axis

□ Rotation around the X-axis

□ Rotation around the Y-axis

☑ Rotation around the Z-axis

□ No movement affecting the camera position

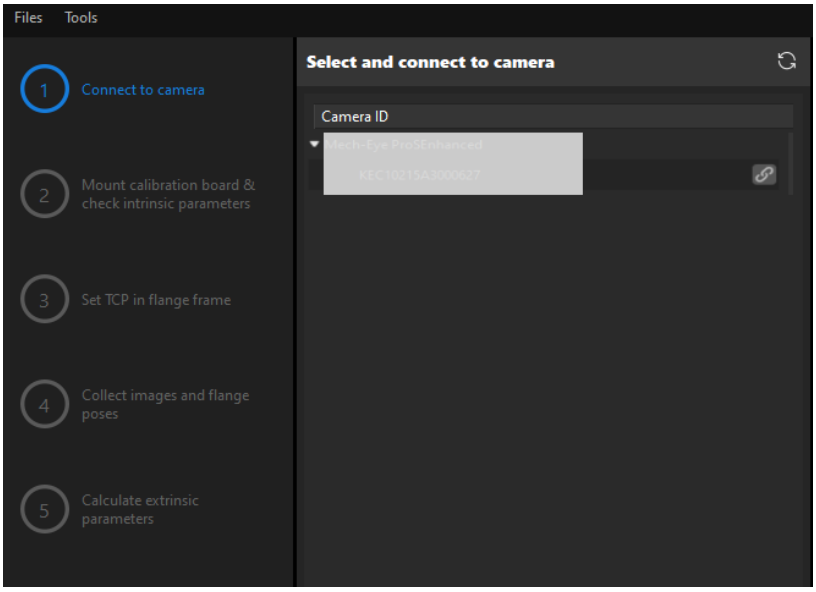
Robot flange poses are required during project execution.

Till now, you have completed the pre-calibration configuration and can start the calibration procedure.

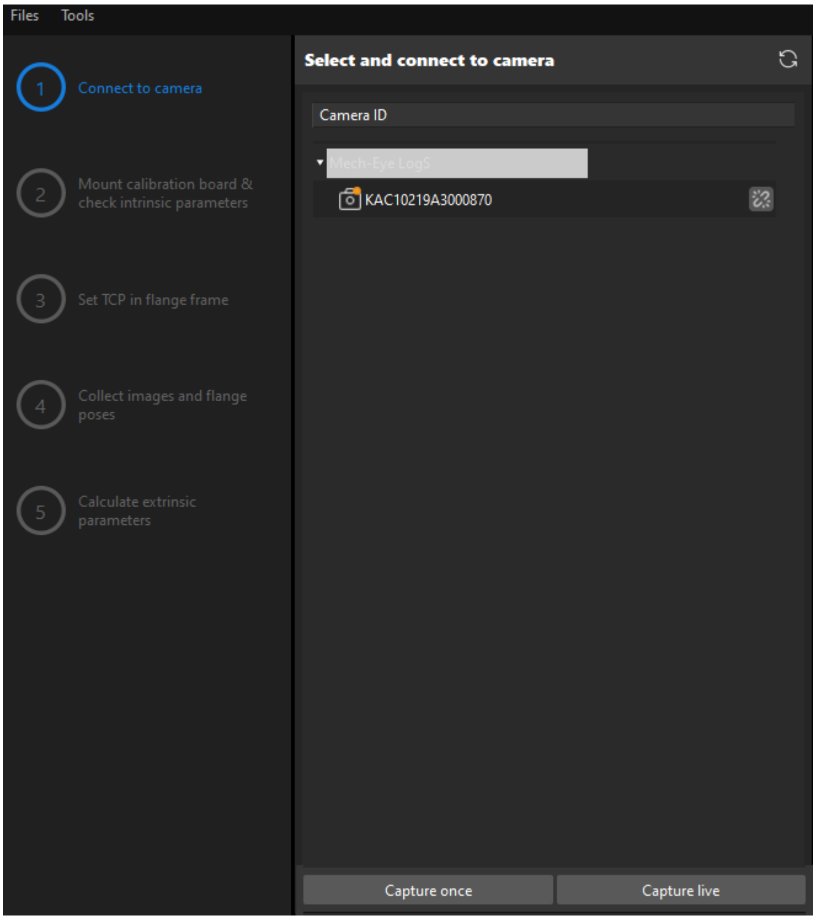
Calibration Procedure

Connect to the Camera

1. In the **Connect to camera** step, select the camera to connect in the Camera ID list, and then click the 🔗 button or double-click the camera entry to connect to it.


Files
Tools
1 Connect to camera
2 Mount calibration board & check intrinsic parameters
3 Set TCP in flange frame
4 Collect images and flange poses
5 Calculate extrinsic parameters
Select and connect to camera
Camera ID
[redacted]

2. After the camera is connected, click the [Capture once] or [Capture live] button.

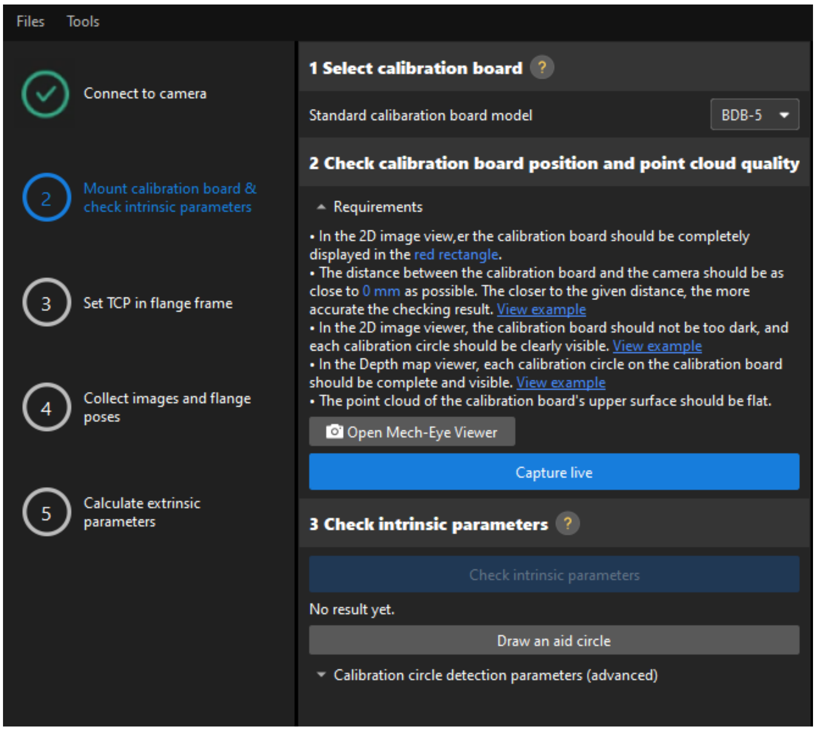

Files
Tools
1 Connect to camera
2 Mount calibration board & check intrinsic parameters
3 Set TCP in flange frame
4 Collect images and flange poses
5 Calculate extrinsic parameters
Select and connect to camera
Camera ID
[redacted]
KAC10219A3000870
Capture once
Capture live

3. In the **Image viewer** panel, ensure that the camera can capture images normally and click the [Next] button on the bottom bar.

Mount Calibration Board & Check Intrinsic Parameters

1. In the **Mount calibration board & check intrinsic parameters** step, select the calibration board model for the **Standard calibration board model** parameter according to its model nameplate

in the **1 Select calibration board** area.



Files Tools

Connect to camera

2 Mount calibration board & check intrinsic parameters

3 Set TCP in flange frame

4 Collect images and flange poses

5 Calculate extrinsic parameters

1 Select calibration board ?

Standard calibaration board model BDB-5

2 Check calibration board position and point cloud quality

Requirements

• In the 2D image view,er the calibration board should be completely displayed in the red rectangle.
• The distance between the calibration board and the camera should be as close to 0 mm as possible. The closer to the given distance, the more accurate the checking result. View example
• In the 2D image viewer, the calibration board should not be too dark, and each calibration circle should be clearly visible. View example
• In the Depth map viewer, each calibration circle on the calibration board should be complete and visible. View example
• The point cloud of the calibration board's upper surface should be flat.

Open Mech-Eye Viewer

Capture live

3 Check intrinsic parameters ?

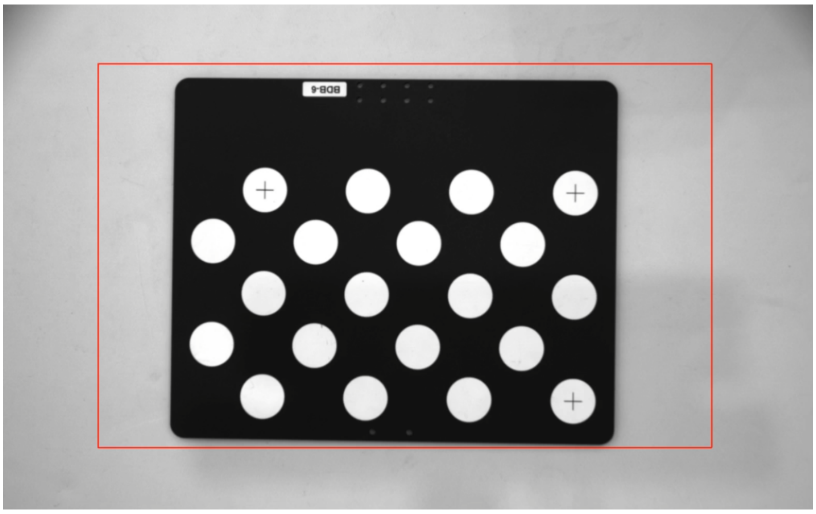
Check intrinsic parameters

No result yet.

Draw an aid circle

Calibration circle detection parameters (advanced)

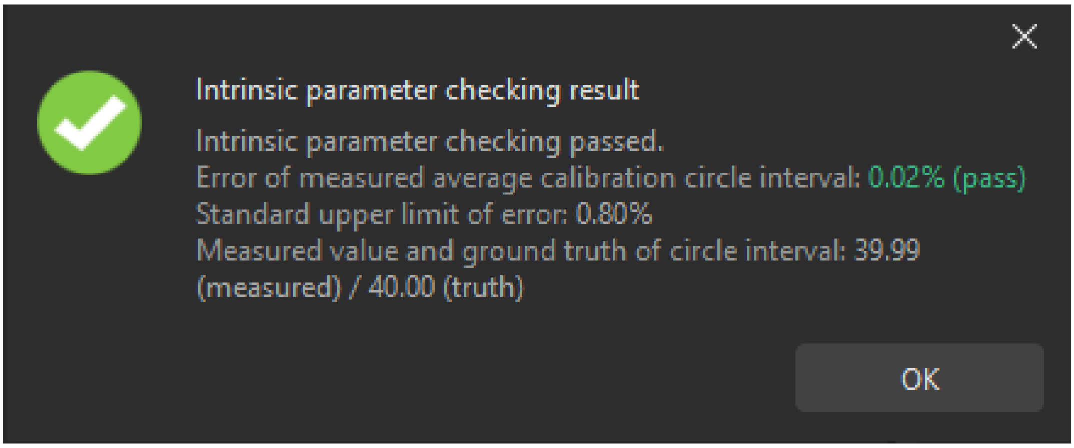
2. In the **2 Check calibration board position and point cloud quality** area, read carefully the requirements on the calibration board position and point cloud quality, and then click the **[Capture live]** button. The **[Capture live]** button will turn into **[Stop capturing and detect position]**.
3. Manually move the calibration board to the proper position (in the red rectangle), and ensure that the 2D image and depth map of the calibration board meet the requirements, and then click the **[Stop capturing and detect position]** button.



If the captured images do not meet the requirements, click the **[Open Mech-Eye Viewer]** button to open the Mech-Eye Viewer software, adjust the 2D and 3D exposure parameters and re-capture images. Please note that you need to change the **Parameter group** parameter to “calib” first.

4. In the **3 Check intrinsic parameters** area, click the **[Check intrinsic parameters]** button.

5. Confirm the results of the camera intrinsic parameter check.
   - If the camera intrinsic parameter check passes, click the [OK] button in the prompted window, and then click the [Next] button on the bottom bar.



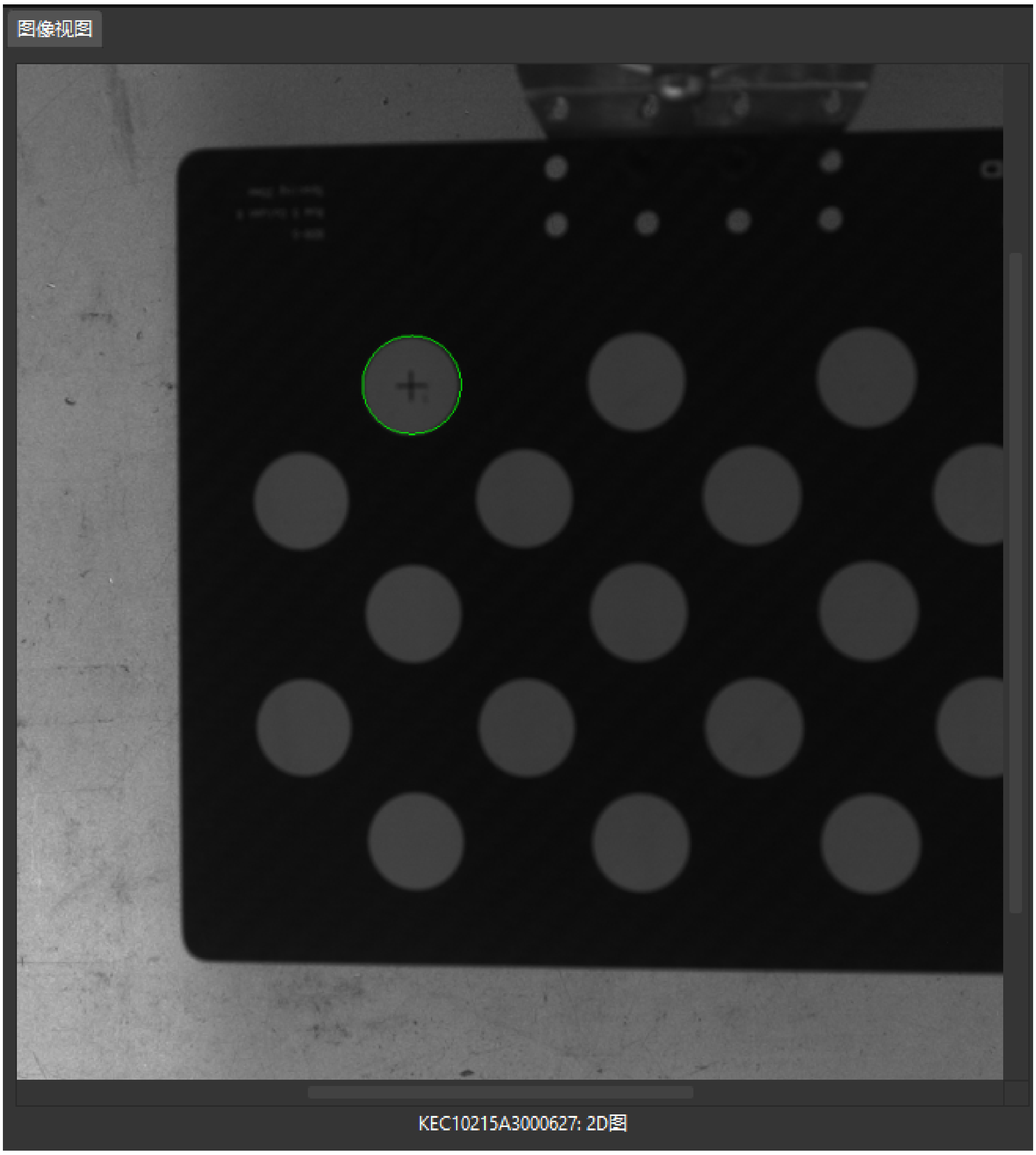
Intrinsic parameter checking result

Intrinsic parameter checking passed.
Error of measured average calibration circle interval: 0.02% (pass)
Standard upper limit of error: 0.80%
Measured value and ground truth of circle interval: 39.99 (measured) / 40.00 (truth)

   - If the camera intrinsic parameter check fails, adjust detection parameters by drawing aid circles or manually adjusting the calibration circle detection parameters, and then click the [**Recheck intrinsic parameters**] button.

Draw an Aid Circle

1. To draw an aid circle, click the [**Draw an aid circle**] button.
2. In the right **Image viewer** panel, right-click the calibration board image and clear the **Fit to window** checkbox, and press the `Ctrl` key and drag the roller to adjust the image to a suitable size.
3. Move the mouse pointer to the cross center point of the calibration circle, press the left mouse button and make the aid circle completely include the calibration circle and then release it.



4. Click the [**Recheck intrinsic parameters**] button, and confirm that the camera intrinsic parameter check passes.

Manually Adjust Detection Parameters

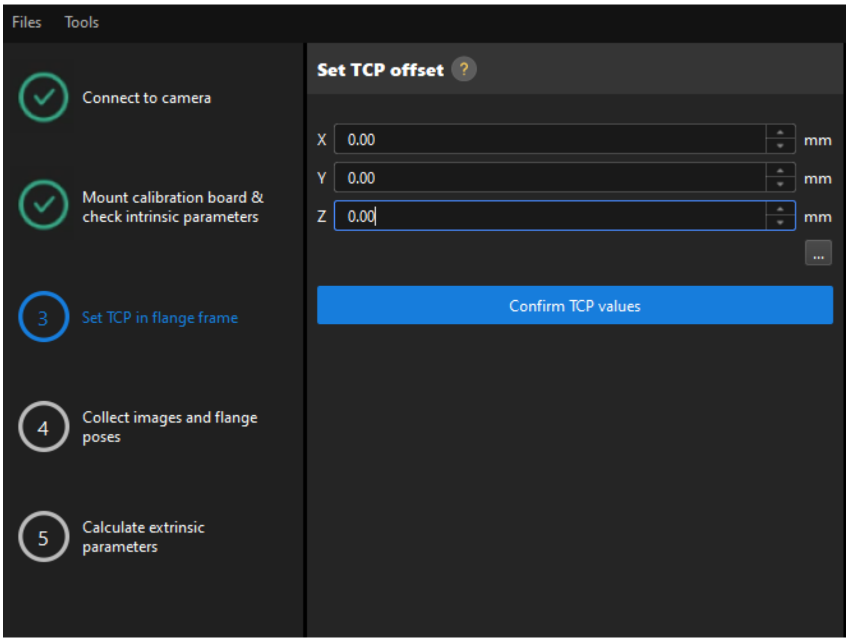
To manually adjust detection parameters, click [**Calibration circle detection parameters (advanced)**], adjust parameter settings and click the [**Recheck intrinsic parameters**] button to confirm whether the camera intrinsic parameter check passes.

If the calibration circles are still not detected, please adjust the camera settings according to the on-site operating conditions.

Set TCP in Flange Frame

Before this step, use the tape measure to measure the deviation (in mm) from the sharp point to the center of the flange in X, Y, and Z directions.

In the **Set TCP in flange frame** step, enter the deviation values in X, Y, and Z directions in the X, Y and Z text boxes, click the [**Confirm TCP values**] button and then click the [**Next**] button on the bottom bar.



✓ Connect to camera

✓ Mount calibration board & check intrinsic parameters

3 Set TCP in flange frame

4 Collect images and flange poses

5 Calculate extrinsic parameters

Set TCP offset ?

X 0.00 mm

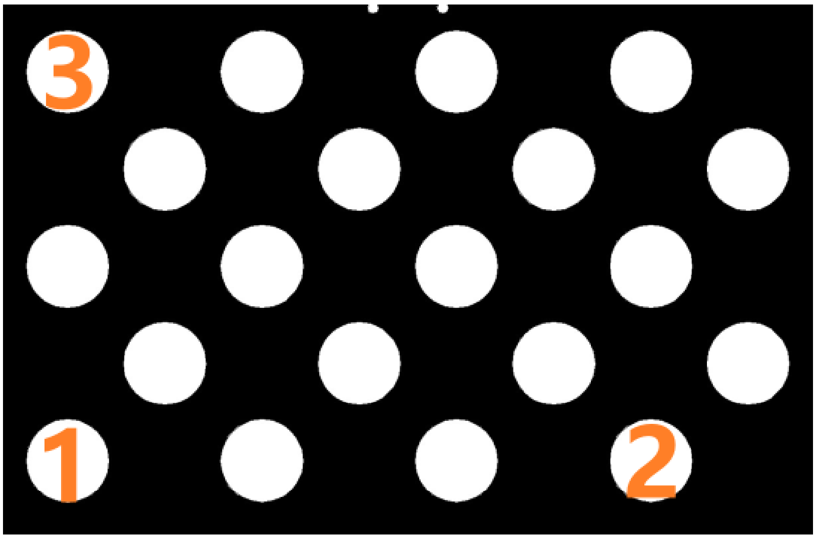
Y 0.00 mm

Z 0.00 mm

> If the truss robot uses the left-handed coordinate system, you need to take the inverse of the deviation value in the Y direction before entering it.

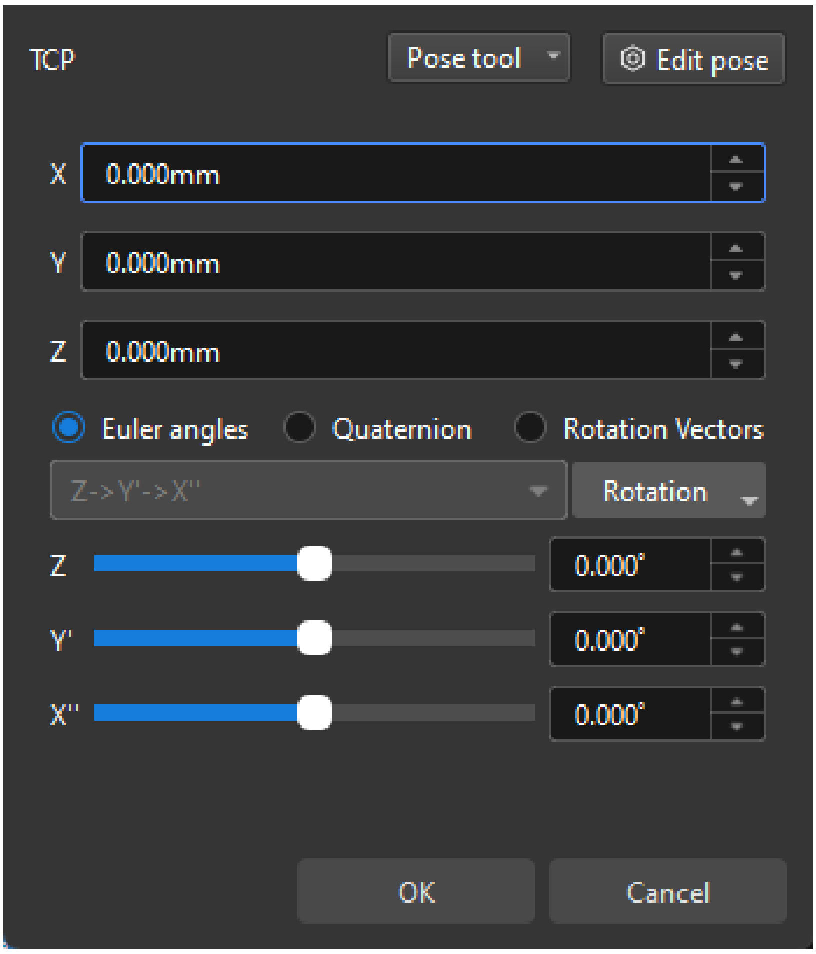
Collect Images and Flange Poses

1. Control the truss robot to let its end tip touch the cross center point of the calibration circle “point 1” on the calibration board, and record the robot flange pose in the teach pendant.



2. In the **Collect images and flange poses** step, click the ⊕ button of point 1, and enter the robot flange pose in the prompted **Input Flange Pose of Robot** dialog box, and then click the [OK] button.

   > If the truss robot uses the left-handed coordinate system, you need to take the inverse of the Y-axis value before entering the robot flange pose.



TCP

X 0.000mm

Y 0.000mm

Z 0.000mm

● Euler angles ○ Quaternion ○ Rotation Vectors

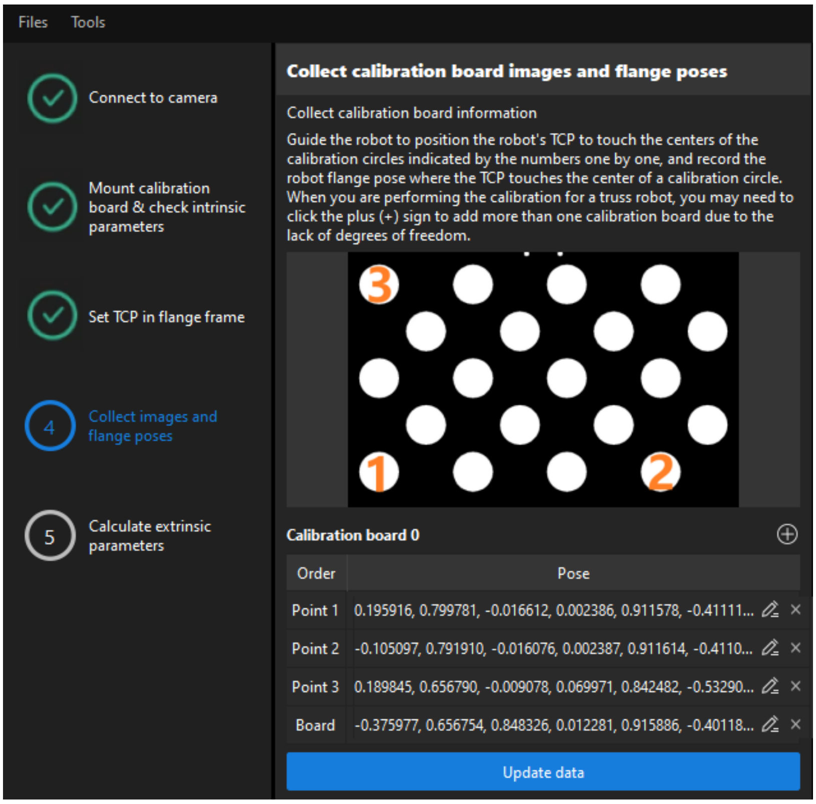
Z->Y'->X'' Rotation

Z 0.000°

Y' 0.000°

X'' 0.000°

3. Repeat the preceding steps, and let the end tip touch the cross center points of point 2 and point 3, and enter robot flange poses.
4. Click the ⊕ button of the current calibration board to capture the images of it.
5. Click the **[Update data]** button and then click the **[Next]** button on the bottom bar.



Files
Tools
Connect to camera
Mount calibration board & check intrinsic parameters
Set TCP in flange frame
4
Collect images and flange poses
5
Calculate extrinsic parameters
3
1
2

Collect calibration board images and flange poses

Collect calibration board information

Guide the robot to position the robot's TCP to touch the centers of the calibration circles indicated by the numbers one by one, and record the robot flange pose where the TCP touches the center of a calibration circle. When you are performing the calibration for a truss robot, you may need to click the plus (+) sign to add more than one calibration board due to the lack of degrees of freedom.

Calibration board 0 ⊕

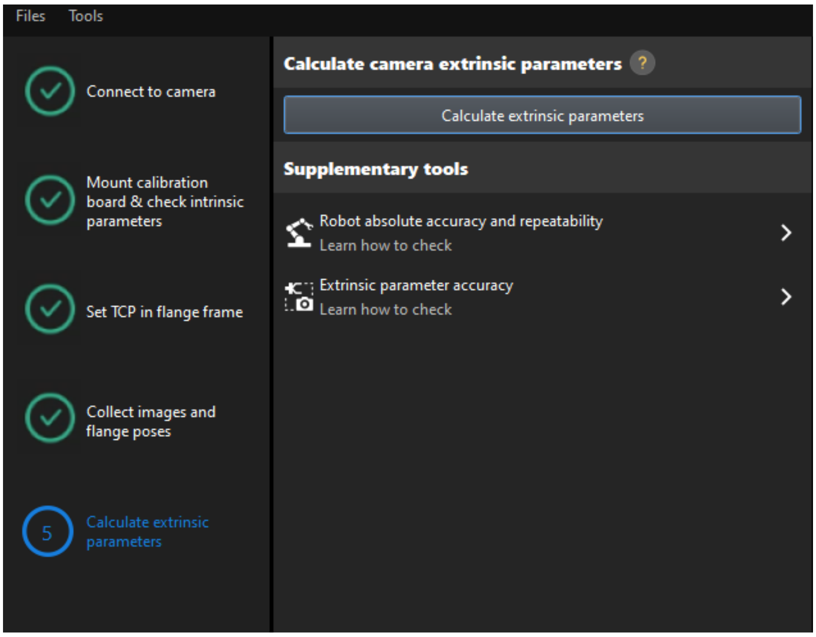
Order	Pose
Point 1	0.195916, 0.799781, -0.016612, 0.002386, 0.911578, -0.41111...
Point 2	-0.105097, 0.791910, -0.016076, 0.002387, 0.911614, -0.4110...
Point 3	0.189845, 0.656790, -0.009078, 0.069971, 0.842482, -0.53290...
Board	-0.375977, 0.656754, 0.848326, 0.012281, 0.915886, -0.40118...

Update data

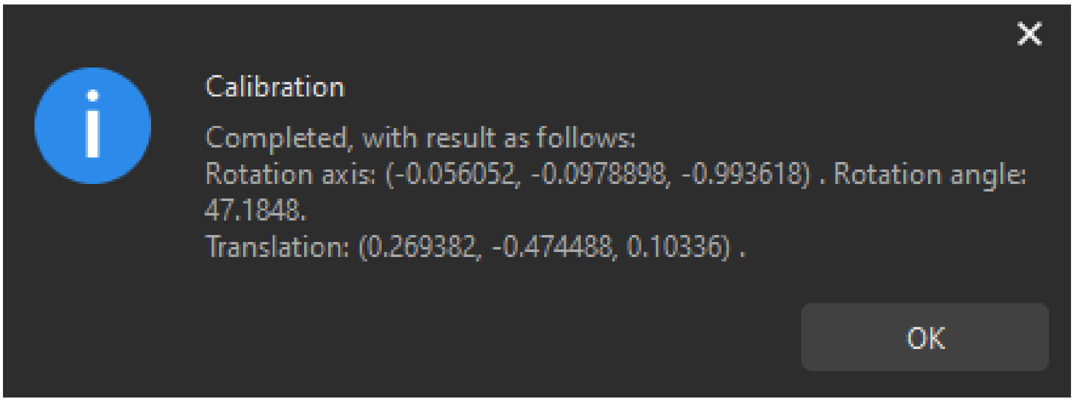
- The calibration program requires that at least three points not in a line should be touched for calculating extrinsic parameters.
- If the truss robot lacks a degree of freedom in a certain direction (for example, in the Y direction), you can add the poses of multiple calibration boards. First, adjust the direction of the calibration board so that the sharp point can touch two points on the calibration board, and at the same time input the pose of the robot flange and collect the image of the calibration board. Then raise (or lower) the calibration board in the Z direction, and again touch the points on the calibration board along the X direction, and at the same time input the robot flange pose and collect the image of the calibration board. Make sure that the new position of the calibration board is significantly different from the previous one.

Calculate Extrinsic Parameters

1. In the **Calculate extrinsic parameters** step, click the **[Calculate extrinsic parameters]** button.


Files
Tools
Connect to camera
Mount calibration board & check intrinsic parameters
Set TCP in flange frame
Collect images and flange poses
5
Calculate extrinsic parameters
Calculate camera extrinsic parameters ?
Calculate extrinsic parameters
Supplementary tools
Robot absolute accuracy and repeatability
Learn how to check
Extrinsic parameter accuracy
Learn how to check

2. In the prompted window indicating calibration success, click the [OK] button.


Calibration

Completed, with result as follows:
Rotation axis: (-0.056052, -0.0978898, -0.993618) . Rotation angle: 47.1848.
Translation: (0.269382, -0.474488, 0.10336) .

If you want to enhance the calibration accuracy, please refer to the section Calibration Result Check and Analysis.

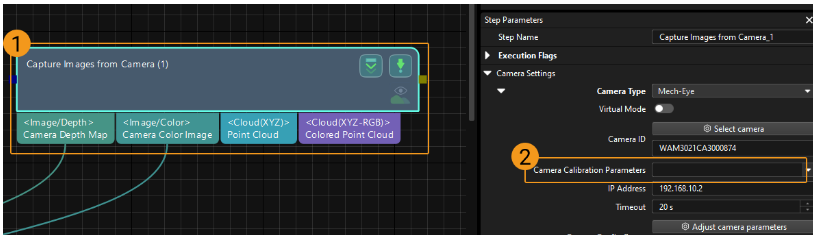
3. Click the [Save] button on the bottom bar. In the prompted **Save Calibration Files** dialog box, click the [OK] button. The camera calibration result will be automatically saved in the “calibration” directory of the project.

Till now, the calibration process is completed.

Change the Extrinsic Parameter File

If the camera’s extrinsic parameter file has been used by a Mech-Vision project, you need to change the extrinsic parameter file used by the Mech-Vision project to a new one after you finish the hand-eye calibration.

1. Select the **Capture Images from Camera** Step.


1
Capture Images from Camera (1)
<Image/Depth> Camera Depth Map
<Image/Color> Camera Color Image
<Cloud(XYZ)> Point Cloud
<Cloud(XYZ-RGB)> Colored Point Cloud
2
Select camera
Adjust camera parameters

Step Parameters

Step Name: Capture Images from Camera_1

Execution Flags

Camera Settings

Camera Type: Mech-Eye

Virtual Mode

Camera ID: WAM3021CA3000874

Camera Calibration Parameters

IP Address: 192.168.10.2

Timeout: 20 s

2. In the **Step Parameters** panel, click ▼ of the **Camera Calibration Parameters** parameter, and select the newly calibrated extrinsic parameter file.

Calibration-Related Status Codes

If the robot uses the Standard Interface to communicate with the Vision System during hand-eye calibration, the Vision System will return status codes in response data to indicate the execution status of the commands. The status codes indicate both the normal execution results and execution failures. You can perform further troubleshooting according to the status codes.

Calibration Error Codes

Error code	Description
7001	Calibration: Parameter error
7002	Calibration: Calibration flange pose not provided by Mech-Vision
7003	Calibration: Calibration joint positions not provided by Mech-Vision
7004	Calibration: Robot failed to reach calibration point

Calibration Normal Status Codes

Normal Status Code	Description
7100	Calibration: Robot reaches calibration point successfully
7101	Calibration: Pose received from Mech-Vision successfully

4.5.5. Hand-Eye Calibration in the Eye-to-Eye Setup

This how-to guide introduces how to complete the hand-eye calibration in the eye-to-eye (ETE) setup.

For more information about calibration in the ETE setup, please refer to Hand-Eye Calibration in the ETE Setup.

Preparation before Calibration

Before hand-eye calibration, you need to finish the following preparations:

- Construct the Mech-Mind Vision System.
- Prepare the materials required for calibration.
- Adjust the effect of calibration board’s point cloud.
- Complete the robot communication configuration.

Construct the Vision System

Construct the Mech-Mind Vision System by referring to the section Vision System Hardware Setup.

You need to use Mech-Eye Viewer, Mech-Vision and Mech-Viz during hand-eye calibration. Please ensure that they have been installed and are running the latest versions.

Prepare the Materials Required for Calibration

The hand-eye calibration in the ETE setup needs to use the calibration board. Please prepare the calibration board according to the following requirements:

- Ensure that the circles of the calibration board are clearly visible and without obvious scratches, and the board does not suffer from deformations.
- In the ETE setup, mount the robot-specific bracket for calibration board onto the robot flange, and then mount the calibration board onto the bracket. Make sure that the calibration board is attached securely, and that the board is parallel to the XY plane of the robot flange frame.

> If an undetachable gripper is connected to the robot flange, you can attach the calibration board directly to the gripper.

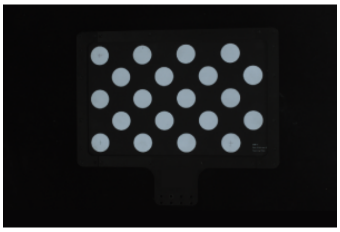
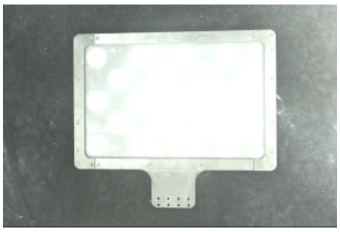
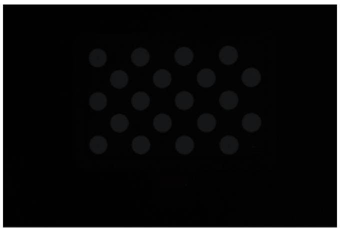
In addition, before calibration, move the robot to the starting point for calibration, that is, the bottom middle of the overlapped area of the two cameras' field of view.

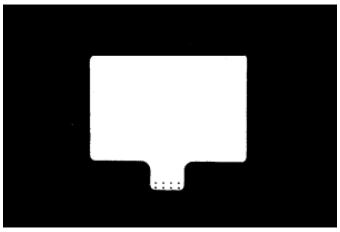
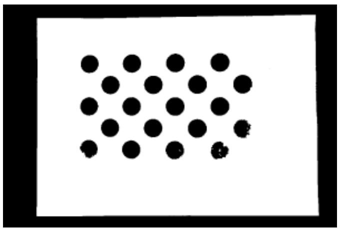
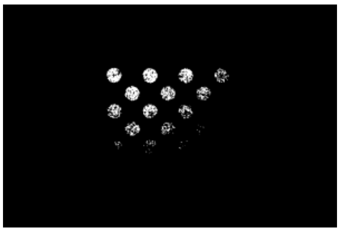
Adjust the Effect of Calibration Board's Point Cloud

1. Open Mech-Eye Viewer to adjust camera parameters.
2. Adjust the 2D parameters to make sure that the 2D image of the calibration board is clear and neither overexposed nor underexposed.
3. Adjust 3D parameters to make sure that the point clouds of the circles on the calibration board are complete and have clear contours. It is recommended to set **Surface Smoothing** and **Outlier Removal** to **Normal** in the **Point Cloud Processing** section to reduce the point cloud fluctuation range.

> If the on-site ambient lights are not ideal and affect the quality of 2D images and point clouds, you can use shading or supplemental light to improve the lighting conditions.

4. Make sure that the images and point cloud of the calibration board are up to standard after completing the preceding steps.

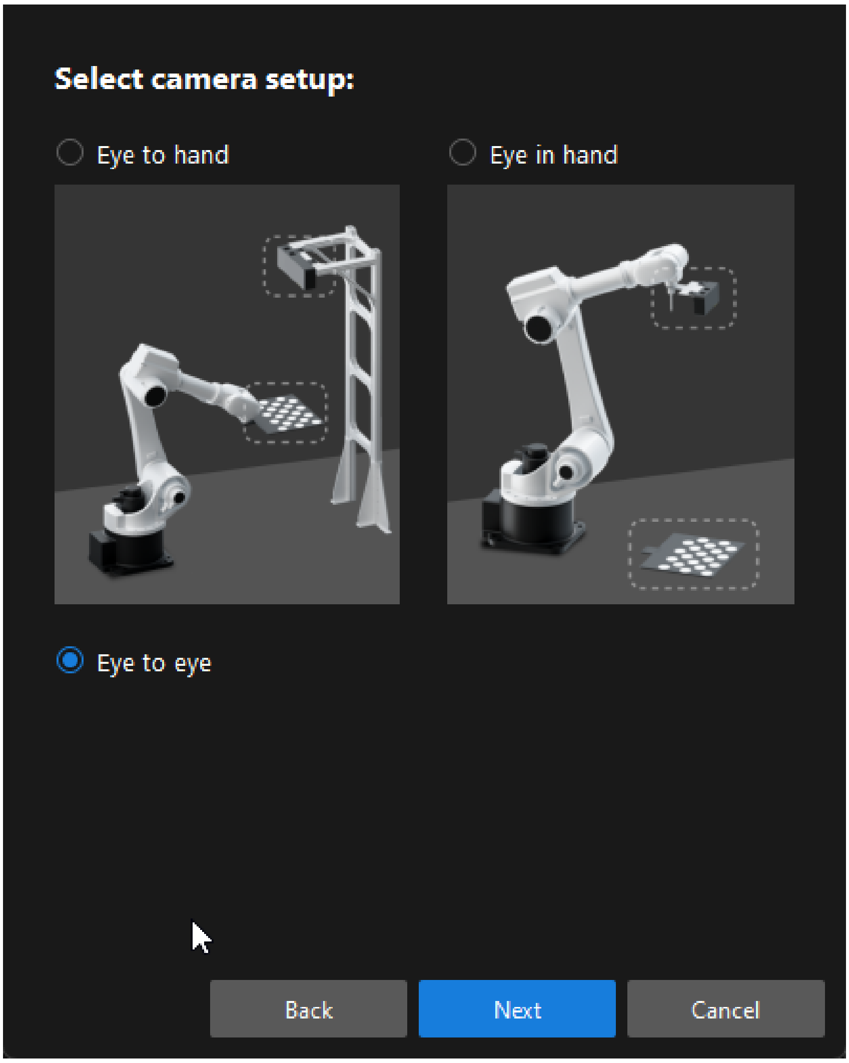
	Normal	Overexposed	Underexposed
2D image			

	Normal	Overexposed	Underexposed
Point cloud			

Pre-calibration Configuration

If the robot supports the Standard Interface or Master-Control communication mode, please complete pre-calibration configuration according to the section Pre-calibration Configuration for automatic calibration in the ETH setup.

Compared with the automatic calibration in the ETH setup, the following operation is different. During the pre-calibration configuration in the ETE setup, you need to select the **Eye to eye** radio button in the **Select camera setup** window.



Select camera setup:

○ Eye to hand

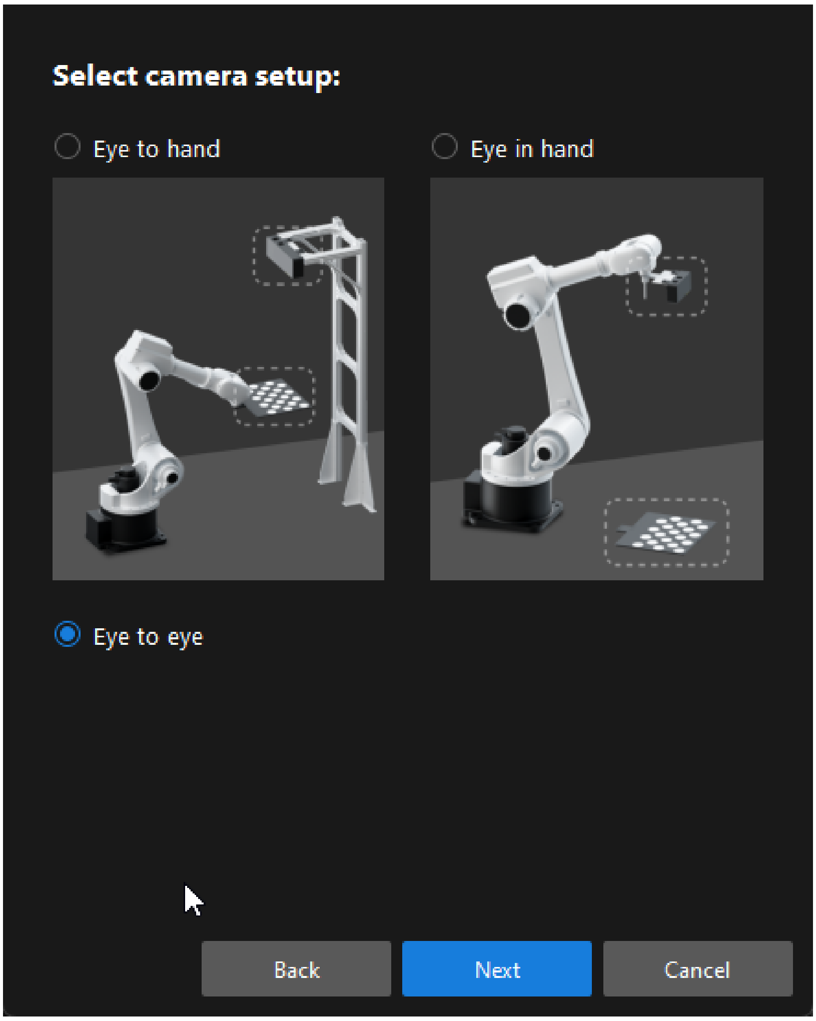
○ Eye in hand

● Eye to eye

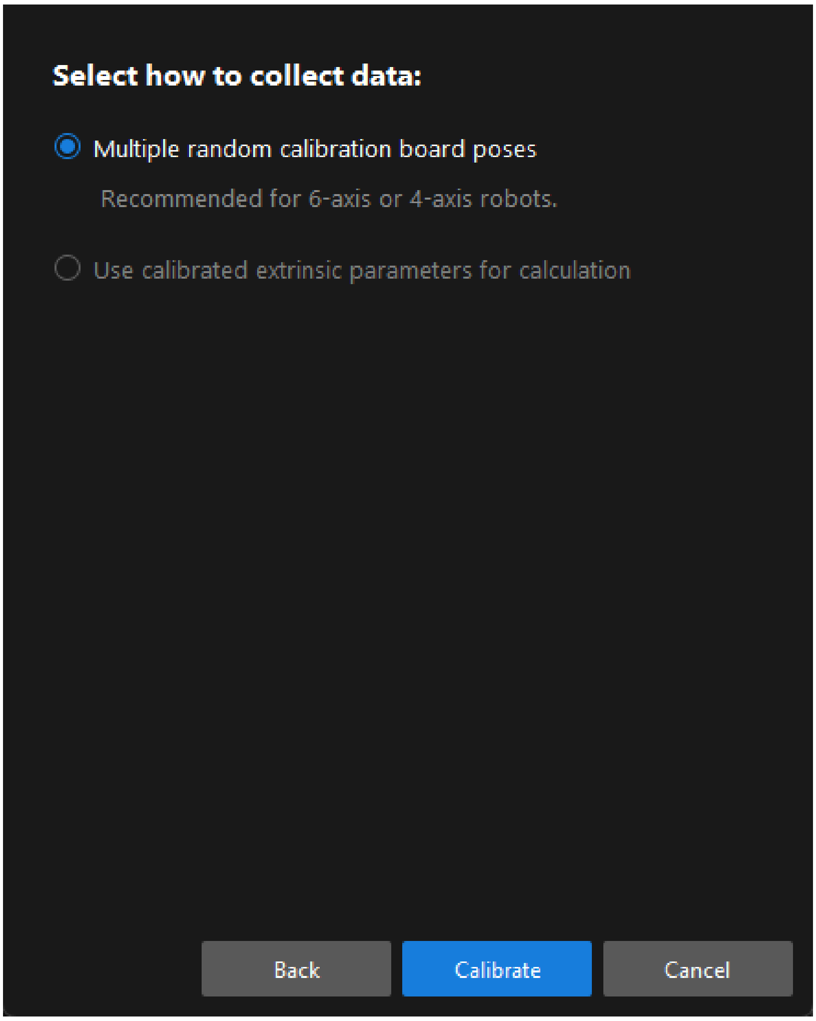
If the robot does not support the Standard Interface or Master-Control communication mode, please complete pre-calibration configuration according to the section Pre-calibration Configuration for manual calibration using the multiple random calibration board poses method in the ETH setup.

Compared with the preceding calibration process, the following operations are different during the pre-calibration configuration in the ETE setup:

1. In the **Select camera setup** window, select the **Eye to eye** radio button.



2. In the **Select how to collect data** window, select the **Multiple random calibration board poses** radio button.



Select how to collect data:

● Multiple random calibration board poses

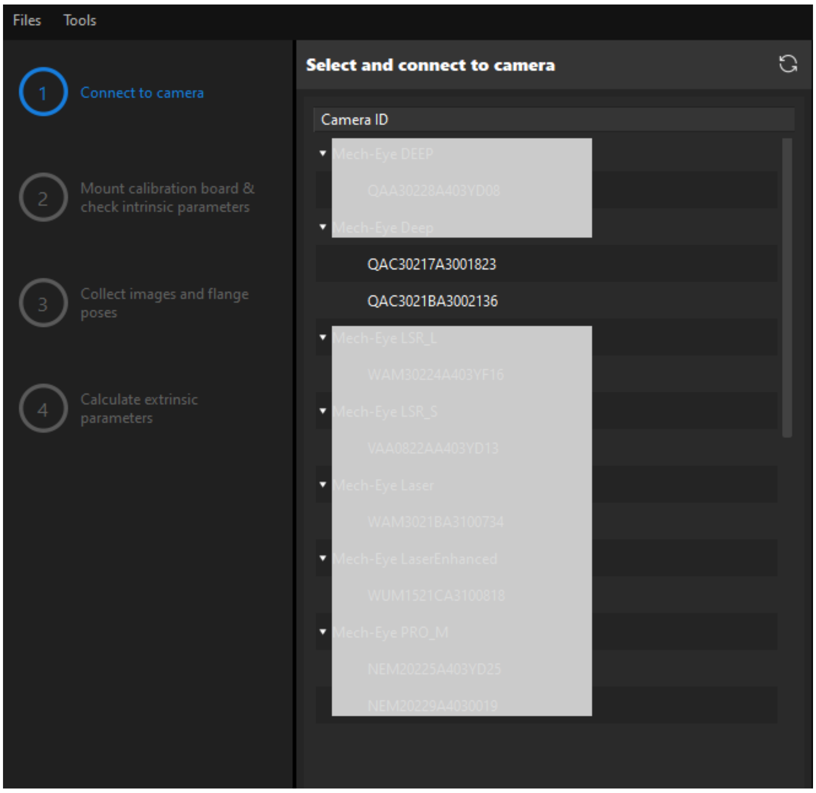
Recommended for 6-axis or 4-axis robots.

○ Use calibrated extrinsic parameters for calculation

Calibration Procedure

Connect to the Camera

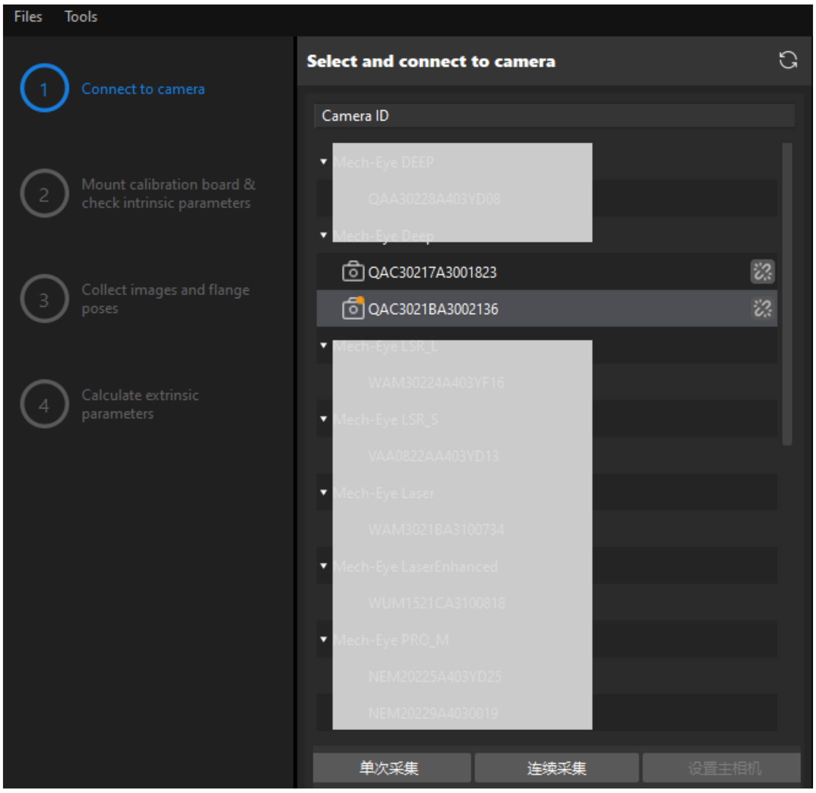
1. In the **Connect to camera** step, select the sub-camera to connect in the **Camera ID** list, and then click the 🔗 button or double-click the camera entry to connect to it.


Files
Tools
1 Connect to camera
2 Mount calibration board & check intrinsic parameters
3 Collect images and flange poses
4 Calculate extrinsic parameters
Select and connect to camera
Camera ID
[redacted]
QAC30217A3001823
QAC3021BA3002136
[redacted]

2. Repeat the preceding step to connect to the main camera. After the main camera is connected, the 📷 icon will be displayed before the camera ID.

> To switch the main camera, select a camera, and then click the [Set as main] button.

3. After the camera is connected, click the [Capture once] or [Capture live] button.


Files
Tools
1 Connect to camera
2 Mount calibration board & check intrinsic parameters
3 Collect images and flange poses
4 Calculate extrinsic parameters
Select and connect to camera
Camera ID
[redacted]
QAC30217A3001823
QAC3021BA3002136
[redacted]
单次采集
连续采集
设置主相机

4. In the **Image viewer** panel, ensure that the camera can capture images normally and click the **[Next]** button on the bottom bar.

> In this step, the 2D image and depth map are captured only for the main camera. To confirm the effects of the images captured by the sub-camera, you can switch it to the main camera, and switch the original main camera back to the main camera after confirming.

Other Steps

If the robot supports the Standard Interface or Master-Control communication mode, please complete remaining steps of the calibration procedure according to the section Calibration Procedure for automatic calibration in the ETH setup.

If the robot does not support Standard Interface or Master-Control communication mode, please complete remaining steps of the calibration procedure according to the section Calibration Procedure for manual calibration using the multiple random calibration board poses method in the ETH setup.

> - In the **Collect images and poses** step, the software will capture calibration images for the two cameras at the same time and collect the pose data. You can switch the camera in the **Calibration image and pose list** panel to view the calibration board images and calibration circle data collected for each camera.
> - In the **Calculate extrinsic parameters** step, the software will calculate the extrinsic parameters for both cameras, and the pose relationship between the two cameras.

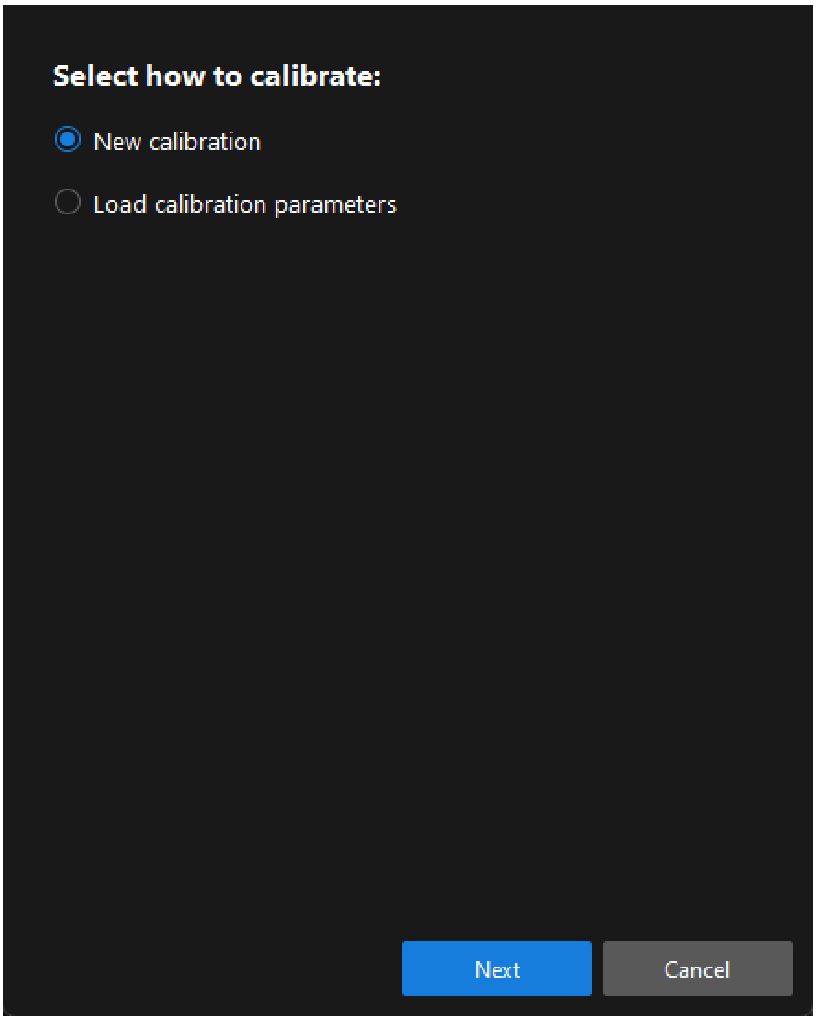
Till now, the calibration process is completed.

Calculate Pose Relationship between Cameras from the Calibrated Extrinsic Parameters Directly

After calibrating the extrinsic parameters of the two cameras separately, you can use the ETE calibration procedure to directly calculate the pose relationship between the two cameras.

To do so, follow these steps:

1. Open Mech-Vision, and click the **[Camera Calibration (Standard)]** button in the toolbar. The **Configuration before Calibration** window will be prompted.
2. In the **Select how to calibrate** window, select the **New calibration** radio button, and then click the **[Next]** button.

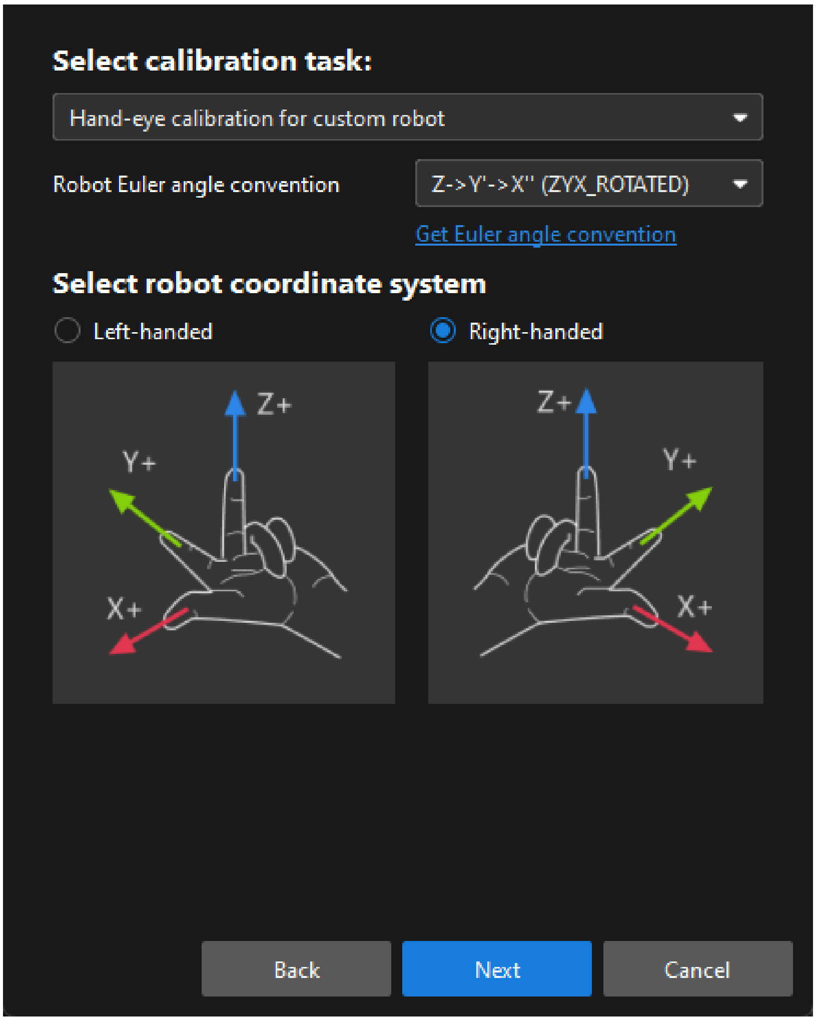


Select how to calibrate:

● New calibration

○ Load calibration parameters

3. In the **Select calibration task** window, select **Hand-eye calibration for custom robot** from the drop-down list box, specify the **Robot Euler angle convention** parameter, select the robot coordinate system type, and then click the **[Next]** button.



Select calibration task:

Hand-eye calibration for custom robot

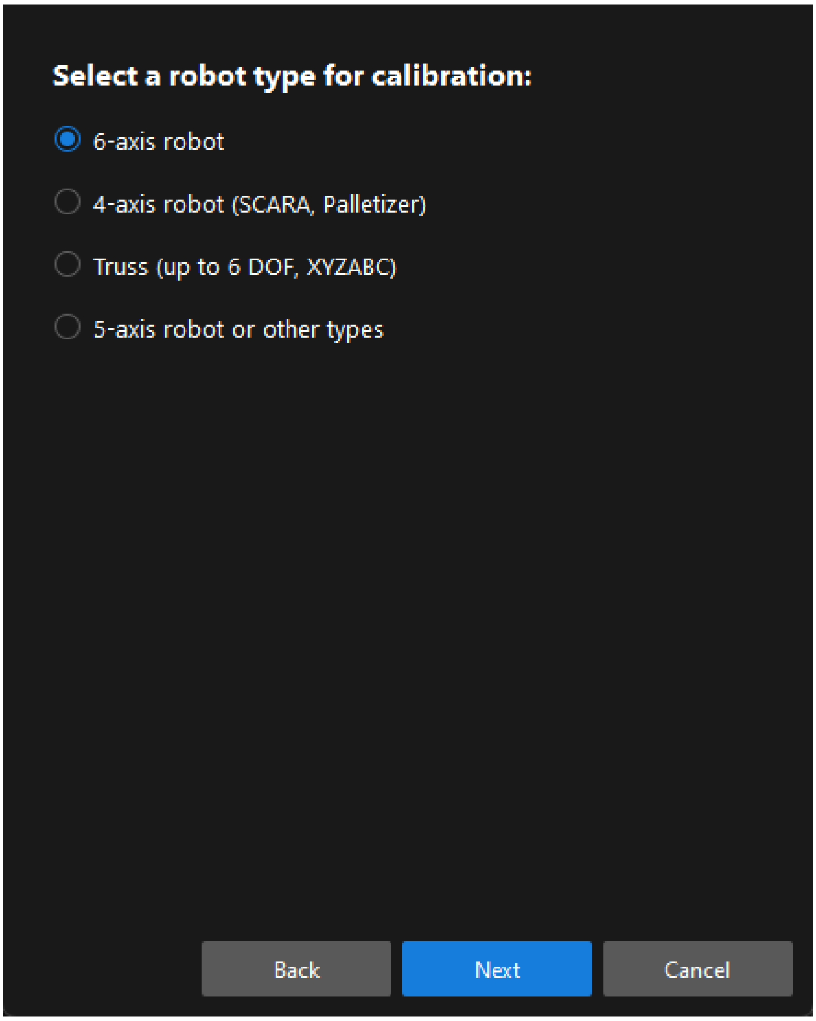
Robot Euler angle convention: Z->Y'->X'' (ZYX_ROTATED)

Get Euler angle convention

Select robot coordinate system

○ Left-handed ● Right-handed

4. In **Select a robot type for calibration** window, select a robot type among **6-axis robot**, **4-axis robot (SCARA, Palletizer)** or **5-axis robot or other types** radio button, and then click the **[Next]** button.



Select a robot type for calibration:

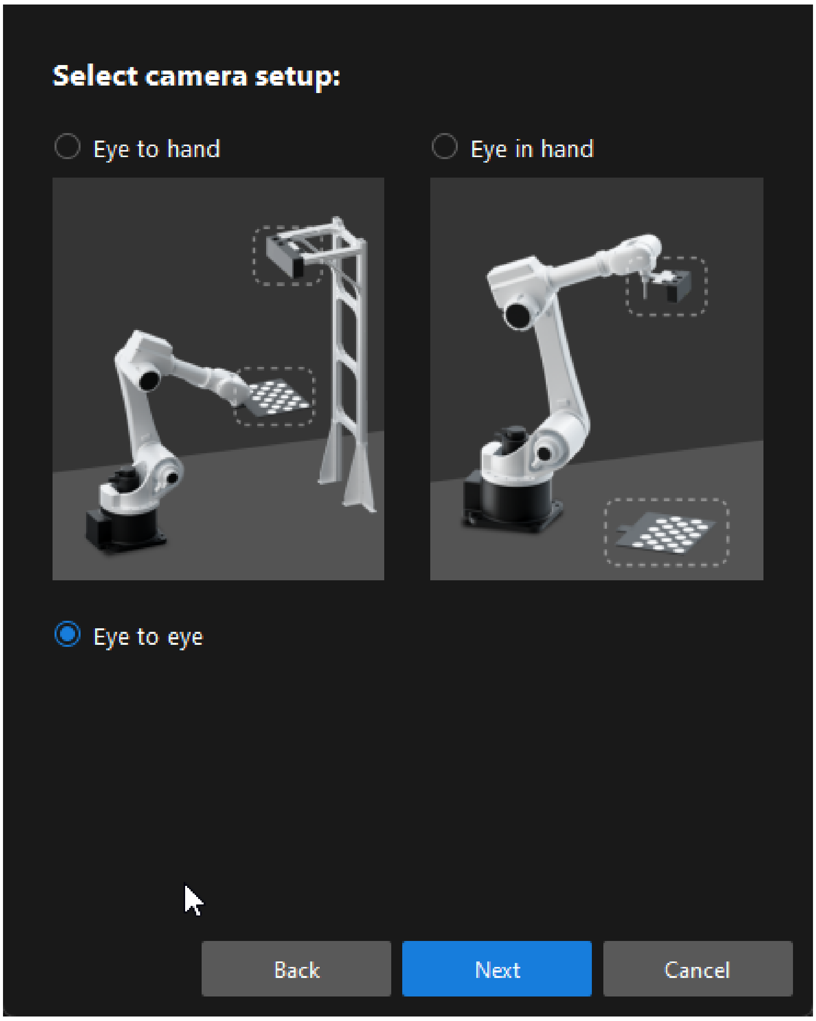
● 6-axis robot

○ 4-axis robot (SCARA, Palletizer)

○ Truss (up to 6 DOF, XYZABC)

○ 5-axis robot or other types

5. In the **Select camera setup** window, select the **Eye to eye** radio button, and then click the **[Next]** button.



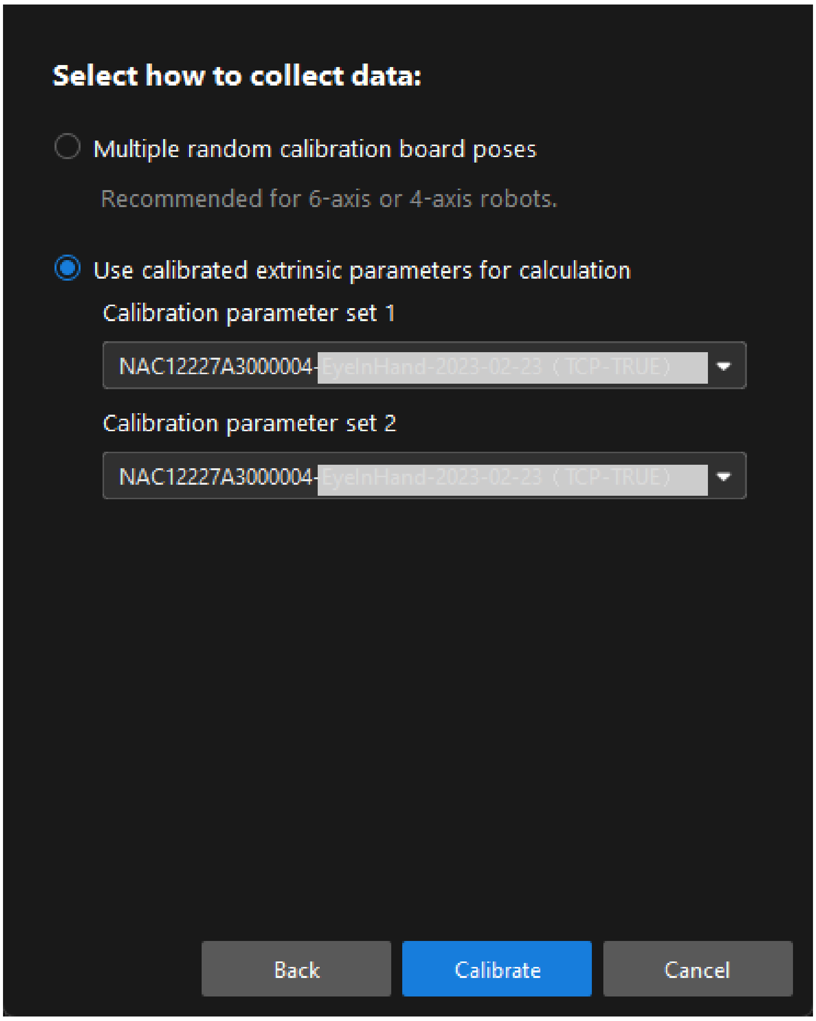
Select camera setup:

○ Eye to hand

○ Eye in hand

● Eye to eye

6. In the **Select how to collect data** window, select the **Use calibrated extrinsic parameters for calculation**, choose two camera's extrinsic parameter files, and then click the [**Calibrate**] button. The **Calibration (Eye to Eye)** window will be prompted.



Select how to collect data:

○ Multiple random calibration board poses

Recommended for 6-axis or 4-axis robots.

● Use calibrated extrinsic parameters for calculation

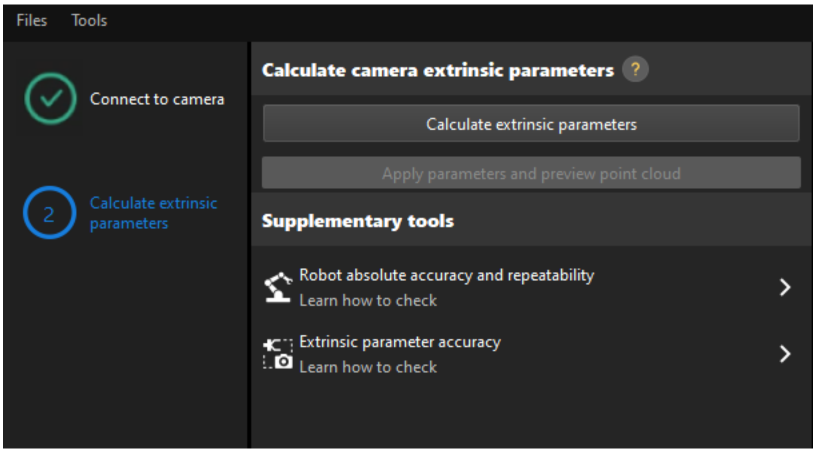
Calibration parameter set 1

NAC12227A3000004-[redacted]

Calibration parameter set 2

NAC12227A3000004-[redacted]

7. In the **Connect to camera** step, select the camera to connect in the Camera ID list, and then click the 🔗 button to connect to it.
8. Repeat the preceding step to connect to the second camera, and then click the **[Next]** button on the bottom bar.
9. In the **Calculate extrinsic parameters** step, click the **[Calculate extrinsic parameters]** button.



Files
Tools
Connect to camera
2
Calculate extrinsic parameters

Calculate camera extrinsic parameters ?

Calculate extrinsic parameters
Apply parameters and preview point cloud

Supplementary tools

Robot absolute accuracy and repeatability
Learn how to check

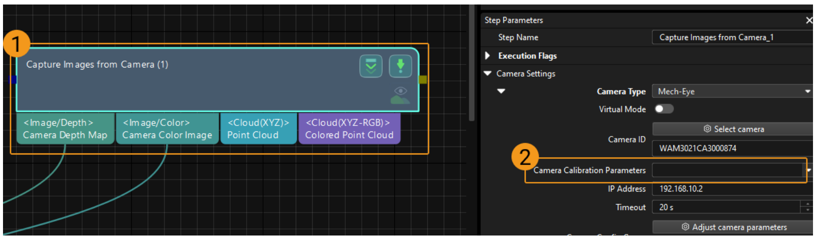
Extrinsic parameter accuracy
Learn how to check

After the camera extrinsic parameters have been calculated, you can view the fused point cloud in the **Point cloud viewer** panel.

Change the Extrinsic Parameter File

If the camera’s extrinsic parameter file has been used by a Mech-Vision project, you need to change the extrinsic parameter file used by the Mech-Vision project to a new one after you finish the hand-eye calibration.

1. Select the **Capture Images from Camera** Step.


1
Capture Images from Camera (1)
<Image/Depth> Camera Depth Map
<Image/Color> Camera Color Image
<Cloud(XYZ)> Point Cloud
<Cloud(XYZ-RGB)> Colored Point Cloud
2
Execution Flags
Camera Settings
Select camera
Adjust camera parameters

Step Parameters
Step Name: Capture Images from Camera_1
Camera Type: Mech-Eye
Virtual Mode
Camera ID: WAM3021CA3000874
Camera Calibration Parameters
IP Address: 192.168.10.2
Timeout: 20 s

2. In the **Step Parameters** panel, click ▾ of the **Camera Calibration Parameters** parameter, and select the newly calibrated extrinsic parameter file.

Calibration-Related Status Codes

If the robot uses the Standard Interface to communicate with the Vision System during hand-eye calibration, the Vision System will return status codes in response data to indicate the execution status of the commands. The status codes indicate both the normal execution results and execution failures. You can perform further troubleshooting according to the status codes.

Calibration Error Codes

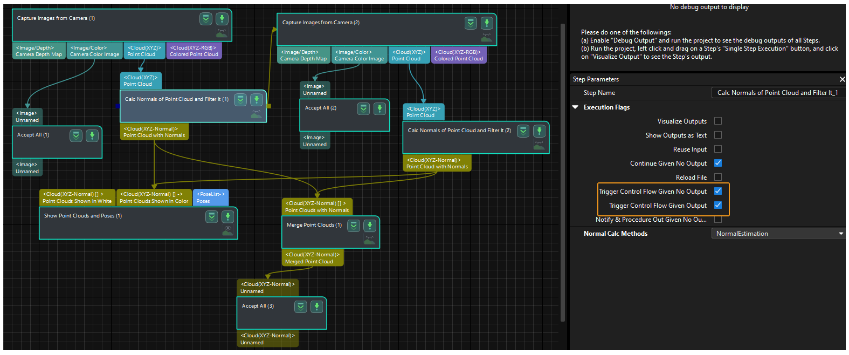
Error code	Description
7001	Calibration: Parameter error
7002	Calibration: Calibration flange pose not provided by Mech-Vision
7003	Calibration: Calibration joint positions not provided by Mech-Vision
7004	Calibration: Robot failed to reach calibration point

Calibration Normal Status Codes

Normal Status Code	Description
7100	Calibration: Robot reaches calibration point successfully
7101	Calibration: Pose received from Mech-Vision successfully

Check the Fusion Effect of the Calibration in the ETE Setup Using a Mech-Vision Project

You can build a project as shown in the following figure. Note that the "Trigger Control Flow Given No Output" and "Trigger Control Flow Given Output" options in Execution Flags area should be selected.


Capture Images from Camera (1)
<Image/Depth> Camera Depth Map
<Image/Color> Camera Color Image
<Cloud(XYZ)> Point Cloud
<Cloud(XYZ-RGB)> Colored Point Cloud
<Cloud(XYZ)> Point Cloud
Calc Normals of Point Cloud and Filter It (1)
<Cloud(XYZ-Normal)> Point Cloud with Normals
<Image> Unnamed
Accept All (1)
<Image> Unnamed
<Cloud(XYZ-Normal) [] > Point Clouds Shown in White
<Cloud(XYZ-Normal) [] -> Point Clouds Shown in Color
<PoseList-> Poses
Show Point Clouds and Poses (1)
Capture Images from Camera (2)
<Image/Depth> Camera Depth Map
<Image/Color> Camera Color Image
<Cloud(XYZ)> Point Cloud
<Cloud(XYZ-RGB)> Colored Point Cloud
<Image> Unnamed
Accept All (2)
<Image> Unnamed
<Cloud(XYZ)> Point Cloud
Calc Normals of Point Cloud and Filter It (2)
<Cloud(XYZ-Normal)> Point Cloud with Normals
<Cloud(XYZ-Normal) [] > Point Clouds with Normals
Merge Point Clouds (1)
<Cloud(XYZ-Normal)> Merged Point Cloud
<Cloud(XYZ-Normal)> Unnamed
Accept All (3)
<Cloud(XYZ-Normal)> Unnamed

No debug output to display

Please do one of the followings:
(a) Enable "Debug Output" and run the project to see the debug outputs of all Steps.
(b) Run the project, left click and drag on a Step's "Single Step Execution" button, and click on "Visualize Output" to see the Step's output.

Step Parameters

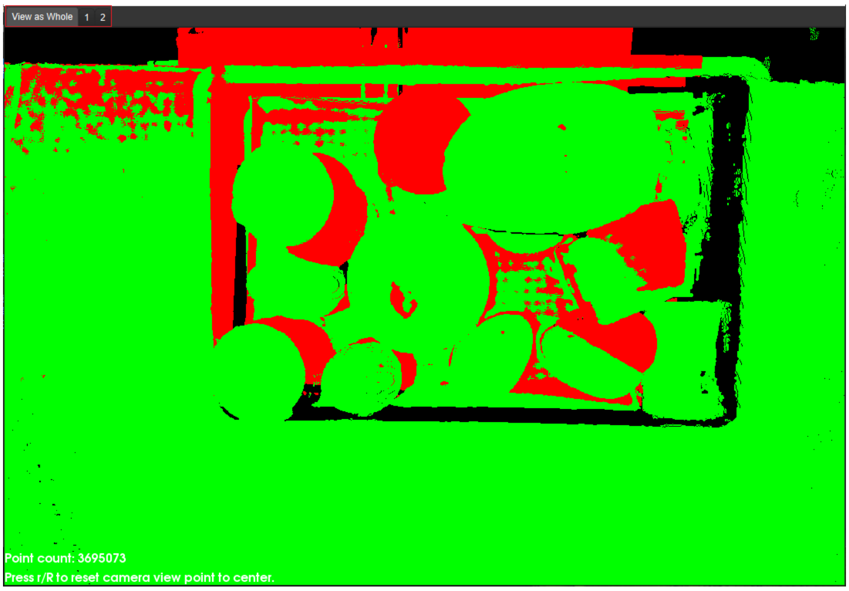
Step Name: Calc Normals of Point Cloud and Filter It_1

Execution Flags

Visualize Outputs □
Show Outputs as Text □
Reuse Input □
Continue Given No Output ☑
Reload File □
Trigger Control Flow Given No Output ☑
Trigger Control Flow Given Output ☑
Notify & Procedure Out Given No Ou... □

Normal Calc Methods: NormalEstimation

Execute the single Step "Merge Point Clouds" to display the fused point cloud. The point cloud output after merging is the fused whole point cloud, as shown in the figure below. You can click the upper left corner **View as Whole| 1 | 2** to switch point clouds.



Point count: 3695073
Press r/R to reset camera view point to center.

4.5.6. Reference

This section provides reference information related to hand-eye calibration.

Calibration-related Concepts

Calibration Principles

Explanation of the Calibration for Truss Robots

Calibration Results Check and Analysis

Calibration Troubleshooting

4.5.6.1. Calibration-related Concepts

Robot communication mode

It refers to how the robot communicates with the vision system. The Mech-Mind Vision System provides three communication modes, namely, Standard Interface, Adapter and Master-Control. For details, refer to the section Communication Overview.

Camera mounting mode

It refers to the way that the camera is mounted in the working unit. Commonly seen mounting modes are eye to hand (ETH) and eye in hand (EIH). In addition, to expand the camera's field of view and improve the point cloud quality, a project may have two cameras installed for one station, which is called eye to eye (ETE).

Calibration mode

It refers whether the calibration board images and poses are collected automatically. It can be divided into automatic calibration and manual calibration. The operations of manual calibration are relatively complicated. Therefore, automatic calibration is recommended whenever possible.

Automatic calibration (recommended)

During calibration, the robot will be connected, and Mech-Vision automatically plans the calibration path, and controls the robot to move along the planned path. It collects calibration board images and robot flange poses at each waypoint.

Manual calibration

During calibration, the robot is not connected, and you need to control the robot manually, move the robot along the path you planned or to touch the calibration circles, enter the robot flange poses, and trigger the software to capture calibration board images.

Calibration data collect method

It refers to how the calibration data is collected. Mech-Vision supports two calibration data collection methods, namely multiple random calibration board poses and TCP touch.

Multiple random calibration board poses (recommended)

It lets the robot move along the waypoints on the robot path automatically generated or you planned. This method captures images of calibration boards, detects calibration circles and collects robot flange poses at each waypoint on the paths in order to calculate the correct relationship between the calibration board, camera, and robot. It is easy to perform and has a high accuracy. It is recommended for 6-axis or 4-axis robots.

TCP touch

This method first determines the pose of the calibration board through the three-point touching method, and then establishes the spatial relationship among the calibration board, camera, and robot. This method suits situations where the robot is installed in a limited space and where the calibration board cannot be installed. It is recommended for 5-axis or other robots.

Camera intrinsic parameters

Intrinsic parameters are internal to a camera, including the focal length, the lens distortion, etc. These parameters are usually calibrated and stored in the camera before the camera leaves the factory.

Camera extrinsic parameters

Extrinsic parameters describe the pose transformation between the robot reference frame and the camera reference frame. The calibration of extrinsic parameters is also called the hand-eye calibration, where the camera is considered as the eye, and the robot the hand. As the spatial relationship between the robot and the camera changes from application to application, the hand-eye calibration needs to be conducted on-site to guarantee its accuracy.

Calibration point

It refers to the robot pose when the camera captures images of the calibration board during calibration. When the multiple random calibration board poses method is used to collect calibration data, the calibration points are the waypoints on the calibration path. When the TCP touch method is used to collect calibration data, the image of the calibration board only needs to be captured once after you touch three calibration circles that are not in a line.

Calibration circle

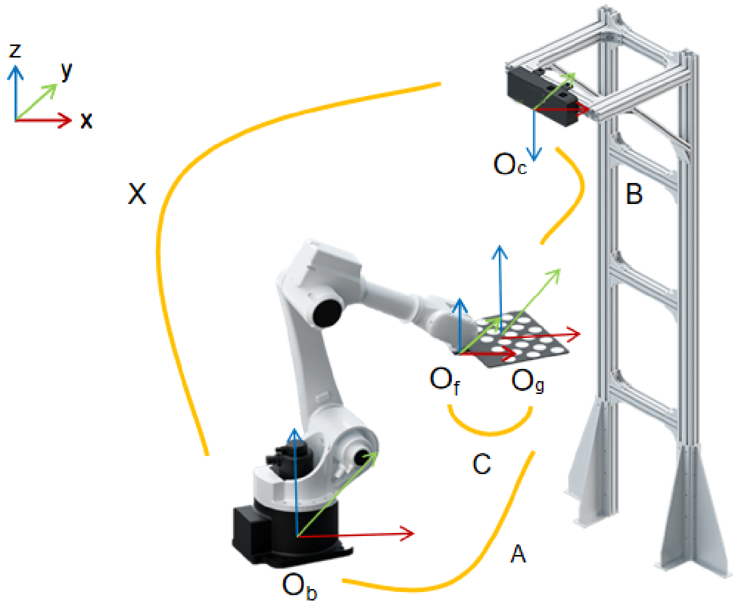
It refers to the circle-shaped feature point on the calibration board. During calibration, the software calculates the pixel coordinates of the center points of the calibration circles based on the 2D image and depth map of the calibration board, and the coordinates of the center points relative to the camera frame based on the depth image. It then calculates camera extrinsic parameters based on collected calibration circle data.

4.5.6.2. Calibration Principles

This topic describes the working principles of calibrations in different scenarios.

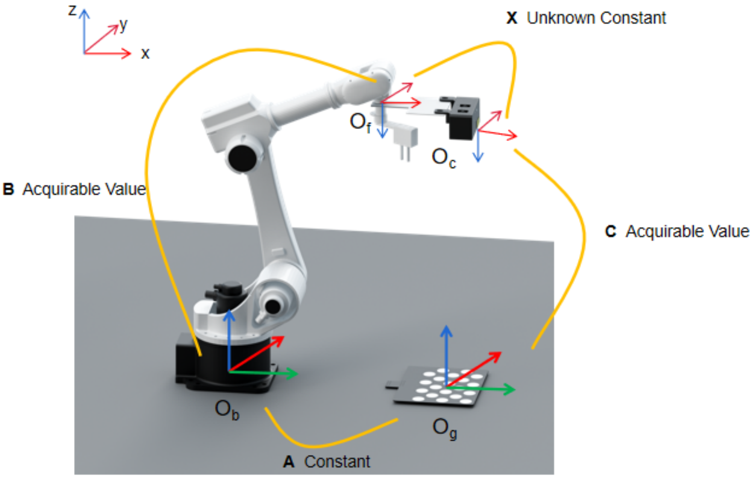
Hand-Eye Calibration in the ETH Setup (Multiple Random Calibration Board Poses)

A calibration board of known dimensions is installed on the robot flange, and the coordinates of each circle on the calibration board, denoted by A, in the robot base reference frame, can be obtained. By capturing images of the calibration board, the coordinates of the camera's optical center relative to each calibration circle, denoted by B, can be obtained. The transformation relationship between the reference frames of the camera's optical center and the robot base, denoted by X, is the unknown variable. Thus, A, B and X form a closed loop, and the value of X can be calculated through the equations generated from the loop. The spatial relationship between the calibration board and the robot flange, denoted by C, is unknown. Through movements of the calibration board during the calibration process, C can be obtained, which can be used to calculate A. By moving the robot, multiple poses of the calibration board can form a set of equations, which give an optimal value of X by numerical fitting and optimization algorithm. This is shown in the following figure.



Hand-Eye Calibration in the EIH Setup (Multiple Random Calibration Board Poses)

In this scenario, the camera is mounted on a flange that is fixed at the end of the robot. The spatial relationship between the center of the flange and the camera’s optical center, denoted by X, is an unknown constant. The pose of the flange in the robot base reference frame denoted by B, is a known variable. The spatial relationship between the camera’s optical center and each circle on the calibration board, denoted by C, can be calculated by capturing images of the calibration board with the camera. The calibration board is placed within the camera’s field of view. The pose of the calibration board in the robot base reference frame, denoted by A, is a measurable constant. Thus, A, B, C, and X form a closed loop. In the equations, since A is a constant, when the first two equations are combined, X will be the only unknown variable. As the robot moves to different poses, the camera will capture images from different angles and produce multiple sets of A, B, and C. Then, these values can be used to perform numerical fitting and obtain an optimal solution of X.



Equations for calculating the transformation relationship between the camera and robot flange reference frames are shown in the following figure.

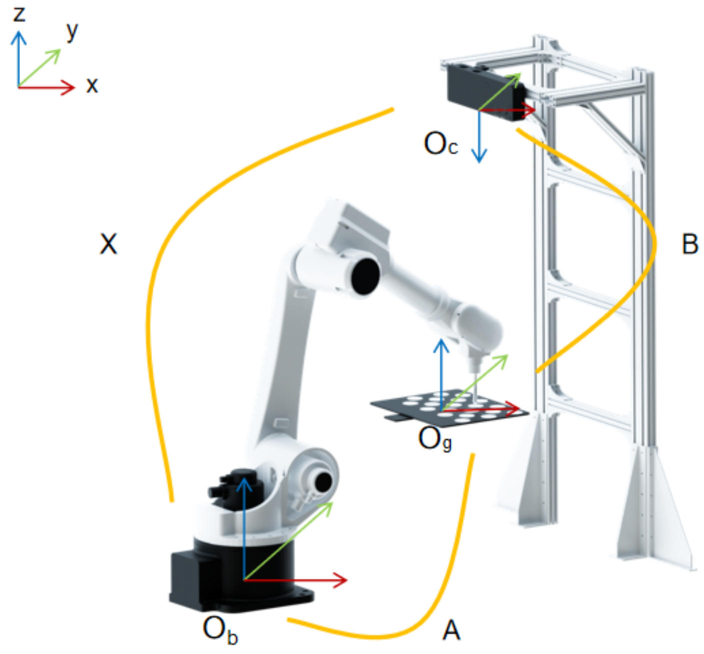
$$B_1 \bullet X \bullet C_1 = A$$

$$B_2 \bullet X \bullet C_2 = A$$

$$B_1 \bullet X \bullet C_1 = B_2 \bullet X \bullet C_2$$

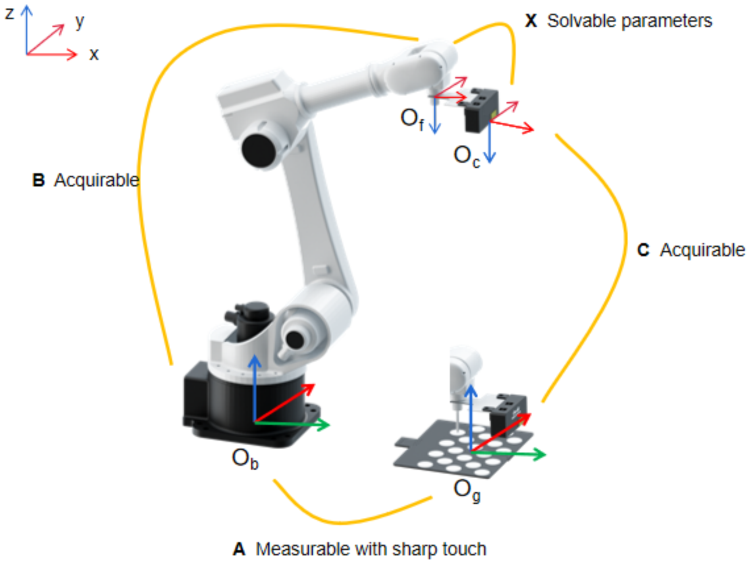
Hand-Eye Calibration in the ETH Setup (TCP Touch)

When the TCP touch method is used, the calibration board is placed on the object plane and a sharp-tipped tool with known dimensions is installed on the robot flange. Then, the robot will be moved to let the center point of the tool touch the cross center points of the calibration circles. As shown in the following figure, the values of A and B are known, and the value of X can be obtained. When the calibration board is not fixed on the flange, the value of A can be calculated by touching the cross center points of the calibration circles on the calibration board using a sharp-tipped tool with known TCP coordinates.



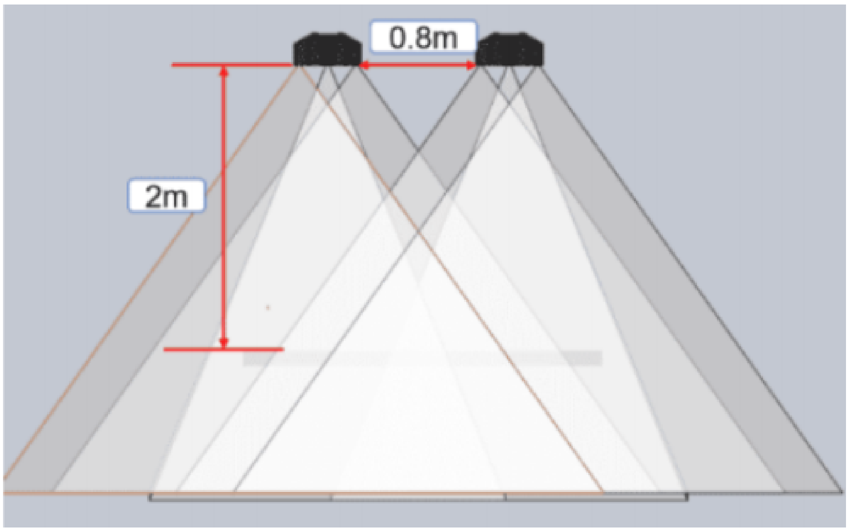
Hand-Eye Calibration in the EIH Setup (TCP Touch)

When the TCP touch method is used, the calibration board is placed on the object plane and a sharp-tipped tool with known dimensions is installed on the robot flange. Then, the robot will be moved to let the center point of the tool touch the cross center points of the calibration circles. As shown in the following figure, since the values of A, B and C are known, the value of X can also be obtained.



Hand-Eye Calibration in the ETE Setup

Using double cameras can expand the camera field of view and improve the point cloud quality of the two cameras' overlapping part, as shown in the following figure.



In the eye-to-eye (ETE) setup, two cameras, one as main camera and the other as sub-camera, are mounted on a stand independent of the robot. Eye to Eye calibration will not only calibrate the extrinsic parameters for the two cameras, but also calibrate the pose relationship between the two cameras. For the ETE setup, Mech-Vision provides a standard ETE calibration procedure.

> The two cameras used should have the same resolution and the overlapping area of their field of view (2D&3D) should cover the entire working area.

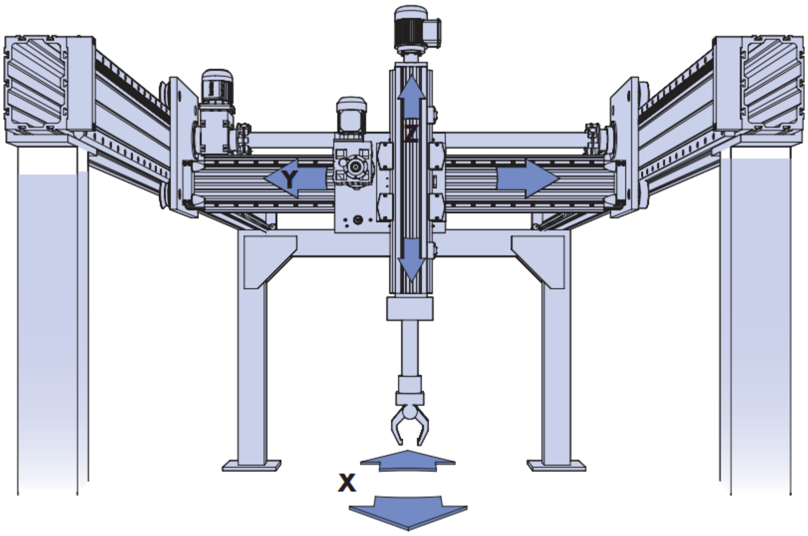
Eye to Eye calibration supports only the multiple random calibration board poses method for calibration data collection, and supports both automatic and manual calibration modes.

4.5.6.3. Explanation of the Calibration for Truss Robots

Related Concepts

Truss Robot

The truss robot (also called gantry robot) is an automation device that can realize automatic control. It is reprogrammable based on the XYZ Cartesian coordinate system in space, has multiple degrees of freedom, and is suitable for different tasks.



Degrees of Freedom of Truss Robot

The truss robot supports multi-degree-of-freedom movement, and the space angle between each degree of freedom is a right angle.

Truss robots usually support 2 to 6 degrees of freedom, for example:

- XYZ axes can move linearly (three degrees of freedom)
- XYZ axes can move linearly, and the Z axis (called Rz or C axis) can rotate (four degrees of freedom)
- XZ axes can move linearly (two degrees of freedom)
- XZ axes can move linearly, and the Z axis can rotate (three degrees of freedom)

The camera is usually mounted on a certain axis of the truss (such as the Z or C axis), and the camera’s scanning position will be affected by the movement of the axis. Therefore, in the pre-calibration configuration, you need to specify which axis movements will affect the camera’s position.

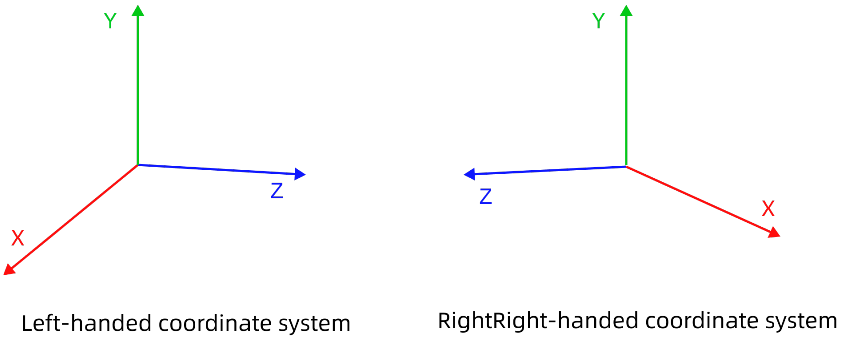
Left- and Right-handed Coordinate Systems

Mech-Mind Vision System uses the flange pose of the robot in the right-handed coordinate system to calculate the extrinsic parameters. Therefore, when using a truss robot, you need to determine whether the truss coordinate system is a right-handed coordinate system.

The left-handed and right-handed coordinate systems can be determined as follows:

- Let the left thumb point to the positive direction of the X-axis, and the index finger point to the positive direction of the Y-axis. If the middle finger can point to the positive direction of the Z-axis, this coordinate system is called a left-handed coordinate system (see the left figure below).

- Let the thumb of the right hand point to the positive direction of the X-axis, and the index finger point to the positive direction of the Y-axis. If the middle finger can point to the positive direction of the Z-axis, this coordinate system is called the right-hand coordinate system (see the right picture below).


Y
Z
X
Left-handed coordinate system
Y
Z
X
RightRight-handed coordinate system

When the truss robot is a left-handed coordinate system, it is recommended to change it to a right-handed coordinate system to facilitate subsequent calibration and vision processing. Typically, you need to invert the positive direction of the encoder motion for either XYZ axis on a truss robot. Generally, you just need to invert the axis installed on the truss base, and then select the right-handed coordinate system in Mech-Vision for calibration.

If you can make this change on the truss robot, you only need to select the left-handed coordinate system in Mech-Vision, and the software will automatically convert the input and output pose data.

Calibration Principles

For a truss robot, the camera is usually mounted on a certain axis of the robot, such as the Z axis, that is, the camera is mounted in Eye in Hand mode. The calibration of the truss robot aims to determine the relative relationship between the camera reference frame and the robot tool reference frame.

Considering the limited freedom and activity space of the truss robot, the Mech-Vision software uses the TCP touch method to collect calibration data and establish the relationship between the calibration board, camera and robot. In addition, the calibration process of the truss robot supports adding the poses of multiple calibration boards. If the truss robot cannot touch three points on the calibration board due to the limited degrees of freedom, you can use multiple calibration boards and control the truss robot to touch three points (not in a line) on multiple calibration boards.

For the principles of the hand-eye calibration for truss robots, please refer to the section Hand-Eye Calibration in the EIH Setup (TCP Touch).

Using the Extrinsic Parameter File

The calibration process of the truss robot generates an extrinsic parameter file for the calibration point, and Mech-Vision will calculate the dynamic extrinsic parameters in real time according to the pose of the truss robot. To calculate the dynamic extrinsic parameters in Mech-Vision projects, Steps Transform Point Clouds for Truss and Transform Poses for Truss need to be used to convert point clouds/poses in camera reference frame to those in the truss robot reference frame.

When the project is running, please ensure that the truss robot can communicate normally with the Mech-Mind Vision System. In addition, the robot program should provide the robot flange pose when triggering the Mech-Vision project to run, so that the "Capture Images from Camera" Step can successfully obtain the pose of the robot flange when capturing images.

> If the truss robot uses the left-handed coordinate system, the communication component automatically converts the robot flange poses sent by the robot program to those in the right-handed coordinate system, and then sends them to Mech-Vision for vision processing. When the vision result is returned, the communication component first automatically converts the poses in the truss robot reference frame output by Mech-Vision to the poses in the left-handed coordinate system, and then returns them to the robot program.

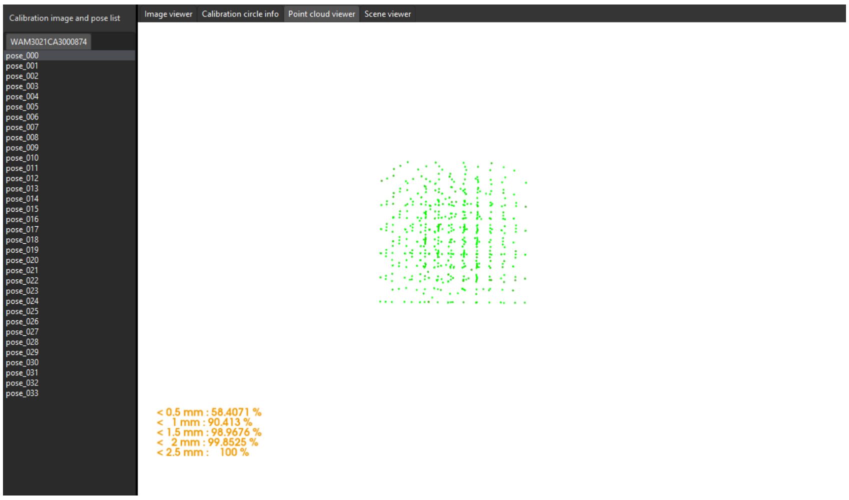
4.5.6.4. Calibration Results Check and Analysis

For the calibrated extrinsic parameters, you need to check whether the calibration accuracy meets the requirements. If the error falls outside the normal range, you should diagnose and solve the problems that are responsible for the error before recalibrating to obtain qualified extrinsic parameters.

Verify Calibration Results

Check Error Point Cloud in Point Cloud Viewer

In the **Calculate extrinsic parameters** step, the **Point cloud viewer** panel will display the point cloud showing the calibration error, as shown in the following figure. The error point cloud shows the deviation between the calculated value and the actual value of the circles on the calibration board.

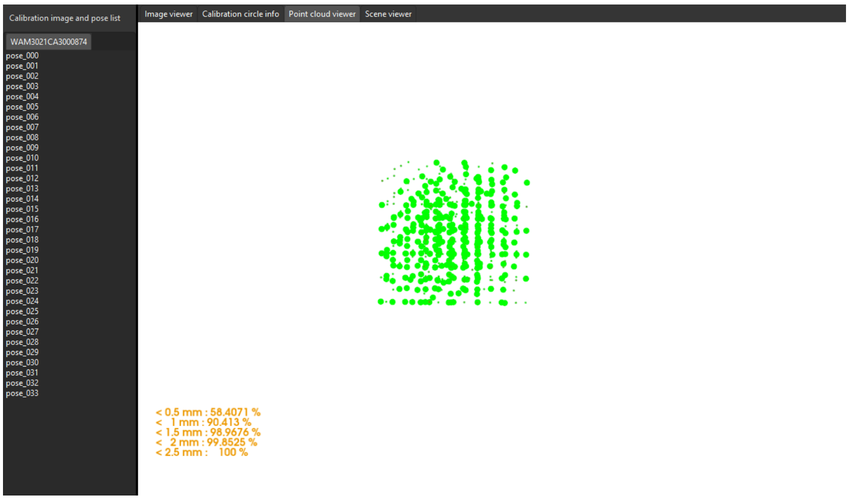


```
< 0.5 mm : 58.4071 %
<   1 mm : 90.413 %
< 1.5 mm : 98.9676 %
<   2 mm : 99.8525 %
< 2.5 mm :    100 %
```

The color of the point represents the error level (one error level per 0.5 mm of deviation). The darker the color is, the greater the error of the point.

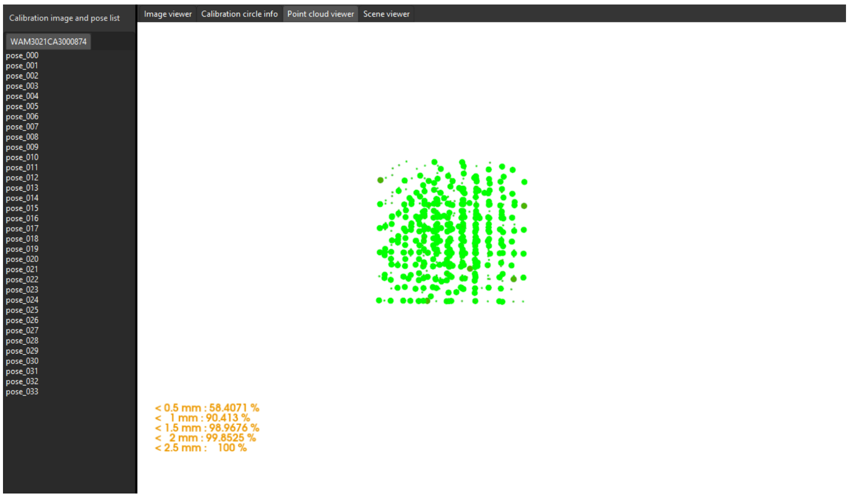
To view the points at each error level, in the **Point cloud viewer** panel, press the number keys 0 to 9. The software will highlight the points in the corresponding error level (0 corresponds to points with errors less than ±0.5 mm, 1 corresponds to points with errors between ±0.5 mm and ±1 mm, and so on and so forth).

The following figure displays all points with an error less than ±0.5 mm (by pressing the number key 0).



```
< 0.5 mm : 58.4071 %
<   1 mm : 90.413 %
< 1.5 mm : 98.9676 %
<   2 mm : 99.8525 %
< 2.5 mm :    100 %
```

The following figure displays all points with error between ±2.0 mm and ±2.5 mm (by pressing the number key 4).



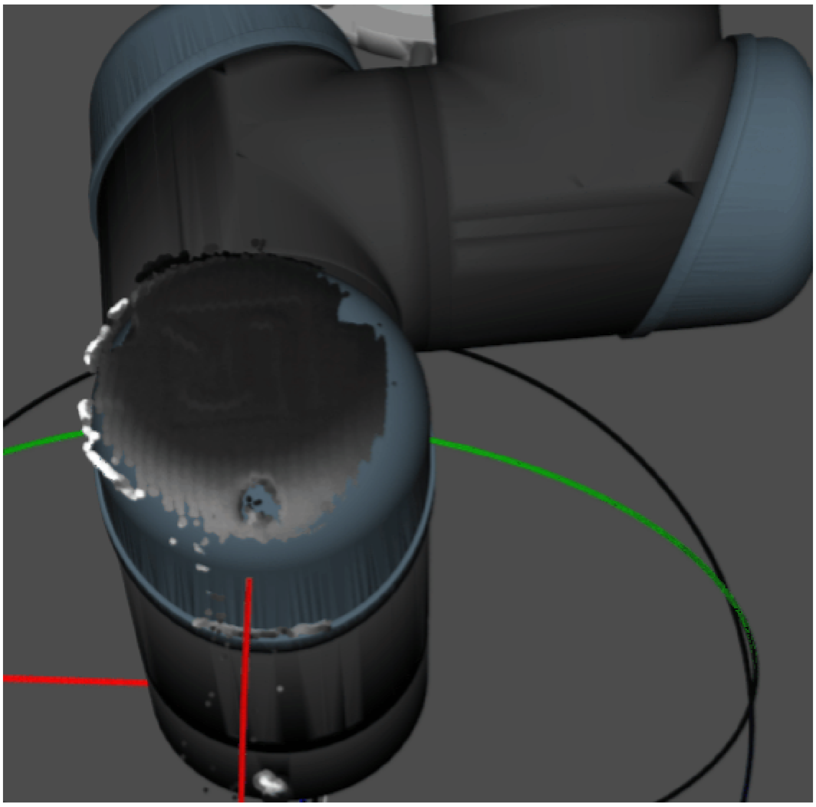
```
< 0.5 mm : 58.4071 %
<   1 mm : 90.413 %
< 1.5 mm : 98.9676 %
<   2 mm : 99.8525 %
< 2.5 mm :    100 %
```

Roughly Check Coincidence Degree between the Point Cloud of the Robot and the Robot Model in Scene Viewer (Suitable for the Eye-to-Hand Setup)

You can check calibration results roughly by inspecting the coincidence degree between the point cloud of the robot and the robot model in Scene Viewer. To do so, follow these steps:

1. After calibration, move the robot arms into the field of view of the camera.
2. In the **Calculate extrinsic parameters** step, click the **[Re-calculate extrinsic parameters]** button. This operation triggers the camera to capture images.
3. Click **Scene viewer** to check the coincidence degree between the point cloud of the robot and the robot model.

   If the point cloud of the robot roughly coincides with the robot model, the calibration is successful.



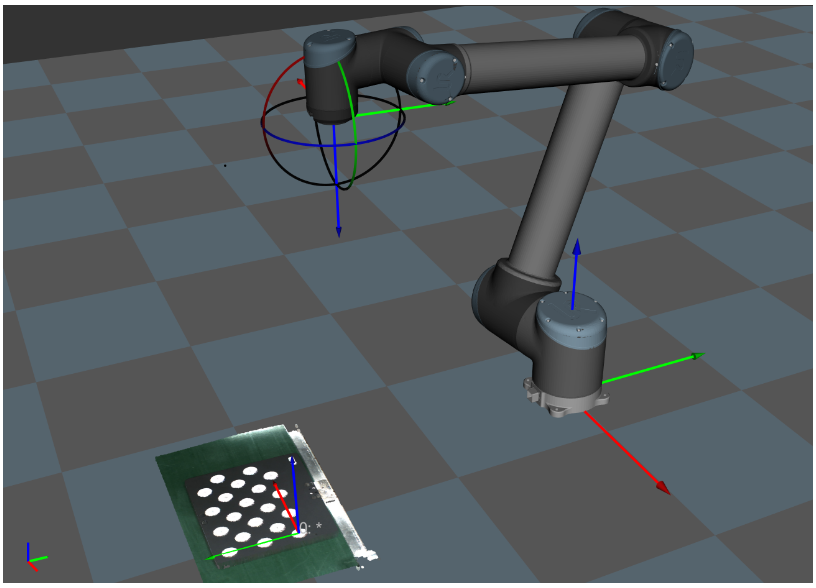
1. Be aware that the robot simulation cannot match the real robot completely, and thus cannot be used to fine-tune the extrinsic parameters.
2. Besides **Scene viewer**, you can also check the coincidence degree between the point cloud of the robot and the robot model in Mech-Viz.

Check the Shift of the Calibration Board's Point Cloud in Reference to a Fixed Point in Scene Viewer (Suitable for the Eye-in-Hand Setup)

You can check calibration results roughly by inspecting the shift of the calibration board's point cloud (to be specific, the crosshair of a calibration circle on the board) in reference to a fixed point in Scene Viewer. To do so, follow these steps:

1. Place the calibration board on a fixed position.
2. Open Mech-Viz, add a fixed point, and make sure that this point coincides with the crosshair of a calibration circle on the calibration board.
   a. Find the "Fixed-Point Move" Step from the Step library and then drag it to the graphical programming workspace.
   b. Select this Step, set the **Waypoint type** parameter to "Workobject pose" on the parameter panel, adjust the X, Y and Z values of the pose to make this point coincide with the crosshair of a calibration circle on the calibration board.
3. Control the robot to change the pose of the camera for several times, and click the **[Recalculate extrinsic parameters]** button in the **Calculate camera parameters** step of the calibration process. This operation will trigger the camera to capture images.
4. Check whether the calibration board's point cloud has a significant shift in reference to the fixed point in Scene viewer.

   If the calibration board's point cloud does not have a significant shift, the calibration is successful.



1. In the EIH setup, robot poses cannot be obtained in real time if the robot is not connected. You need to enter the robot pose where the camera captures images before checking the point cloud in Scene Viewer or Mech-Viz.
2. Besides **Scene viewer**, you can also check whether the calibration board's point cloud has a significant shift in reference to the fixed point in Mech-Viz.

Analyze Calibration Results

Calibration Result Evaluation Standard

For the calibrated extrinsic parameters, you need to check whether the calibration accuracy meets the requirements. In general, the standards for acceptable calibration error in the point cloud are as follows:

- For common projects, all data points should have errors lower than 3 mm (< 3 mm: 100%).
- For high-precision projects, all data points should have errors lower than 2 mm (< 2 mm: 100%).
- For palletizing and depalletizing projects, all data points should have errors lower than 5 mm (< 5 mm: 100%).

The above standards are just for reference. Please follow the specific accuracy requirement of the real application.

Check Extrinsic Parameter Calibration Report

For the automatic calibration and the manual calibration using the multiple random calibration board poses method for six-axis robots, Mech-Vision's camera calibration tool has provided the "extrinsic parameter calibration report" function.

In the **Calculate extrinsic parameters** step of the calibration process, after you have got the extrinsic parameters by clicking the **[Calculate extrinsic parameters]** button, you can click the **[View extrinsic parameter calibration report]** button.

The Mech-Vision's camera calibration tool will check the extrinsic parameter calibration results based on the collected calibration data, and generate the extrinsic parameter calibration report.

This report displays the results of the following checks and improvement recommendations:

- Euler angle convention check
- Camera accuracy check
- Robot absolute accuracy check

If the camera accuracy in the report does not meet the requirements, please follow the recommendation to troubleshoot the camera accuracy errors.

Improve the Accuracy of Extrinsic Parameters

If the calibration accuracy is below standard, you need to follow the following instructions to inspect the calibration data and find the root cause for the errors.

Analyze Extrinsic Parameter Errors Using the Error Analysis Tool

Mech-Vision has provided the error analysis tools to analyze extrinsic parameter errors in the eye-to-hand setup and eye-in-hand setup. The error analysis covers the check of the extrinsic parameter errors and the robot absolute accuracy errors.

Please follow the instructions in the tools to analyze errors and improve accuracy according to the recommendations in the analysis result.

You can open the error analysis tools for extrinsic parameters in any of the following ways:

- In the menu bar, Toolkit › Error Analysis Tool › Extrinsic parameters › Analyze extrinsic parameter error in EIH/ETH setup.
- In the **Supplementary tools** area of the **Calculate extrinsic parameters** in the calibration process, click **Extrinsic parameter accuracy**.

For details, please refer to the section Analyze extrinsic parameter errors.

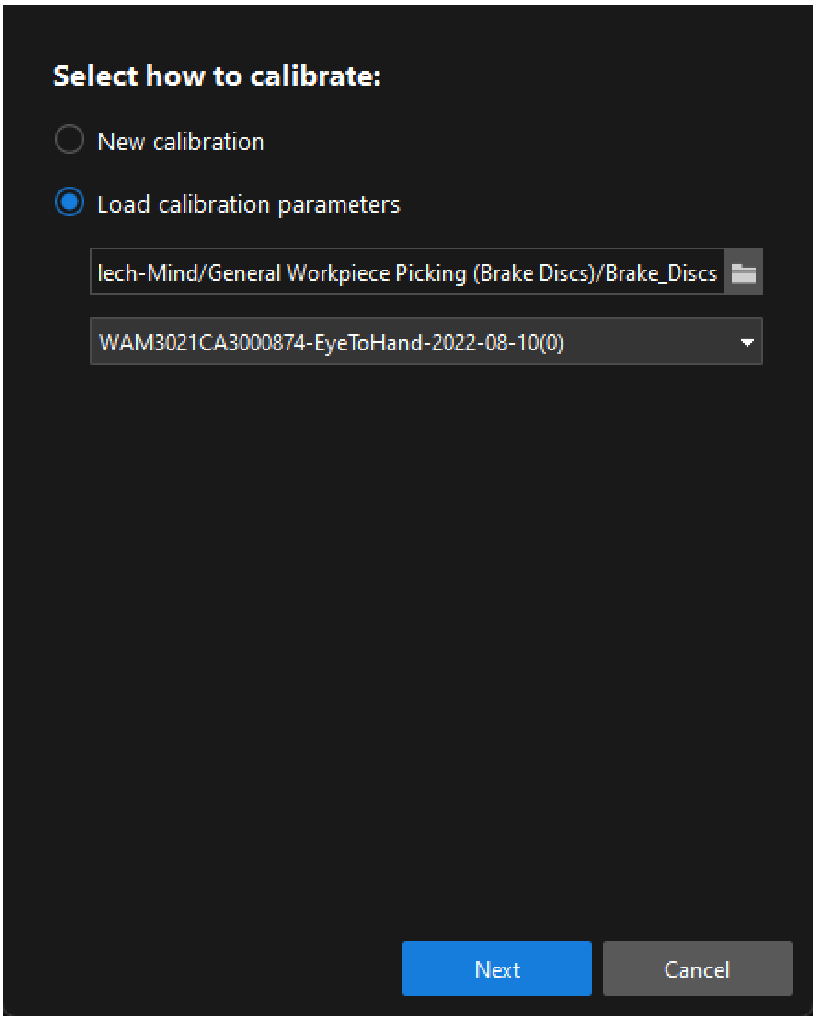
Inspect Calibration Data

Please check the calibration data in the following order to locate the root cause.

Calibration data refers to the calibration point data generated during the calibration process and is stored in calib_data.json. This file records data such as the flange poses at calibration points and calibration circle data.

To inspect calibration data, follow these steps:

1. Open Mech-Vision, and click the **[Camera Calibration (Standard)]** button in the toolbar. The **Configuration before Calibration** window will be prompted.
2. In the **Select how to calibrate** window, select the **Load calibration parameters** radio button, choose the calibration parameter file, and then click the **[Next]** button.



Select how to calibrate:

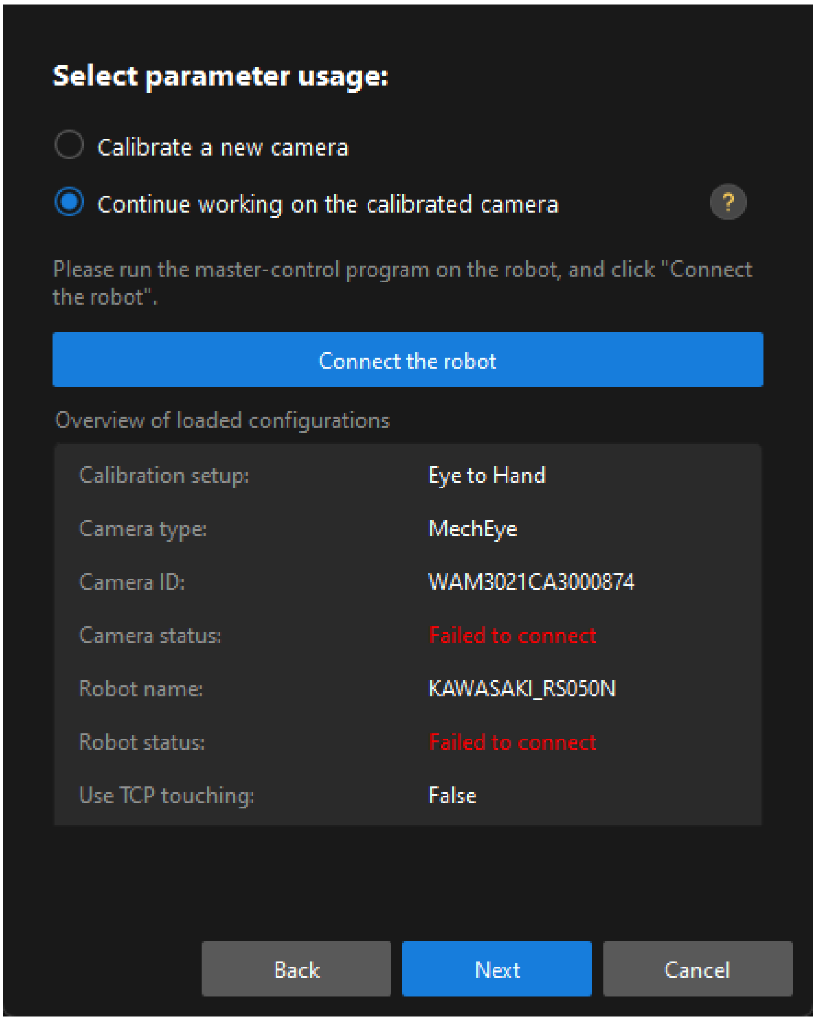
○ New calibration

● Load calibration parameters

lech-Mind/General Workpiece Picking (Brake Discs)/Brake_Discs

WAM3021CA3000874-EyeToHand-2022-08-10(0)

3. In the **Select parameter usage** window, select the **Continue working on the calibrated camera** radio button, and then click the **[Next]** button.



Select parameter usage:

○ Calibrate a new camera

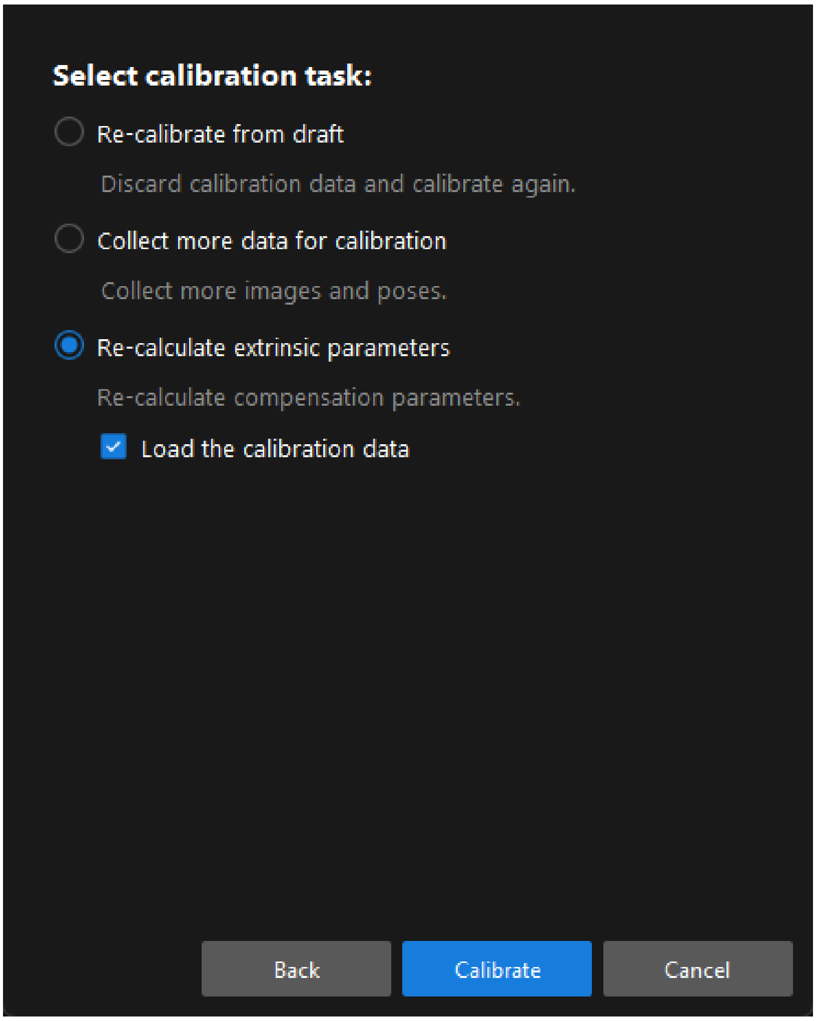
● Continue working on the calibrated camera

Please run the master-control program on the robot, and click "Connect the robot".

Overview of loaded configurations

Calibration setup:	Eye to Hand
Camera type:	MechEye
Camera ID:	WAM3021CA3000874
Camera status:	Failed to connect
Robot name:	KAWASAKI_RS050N
Robot status:	Failed to connect
Use TCP touching:	False

4. In the **Select calibration task** window, select the **Re-calculate extrinsic parameters** radio button, select the **Load the calibration data** checkbox, and then click the [**Calibrate button**].



Select calibration task:

○ Re-calibrate from draft

Discard calibration data and calibrate again.

○ Collect more data for calibration

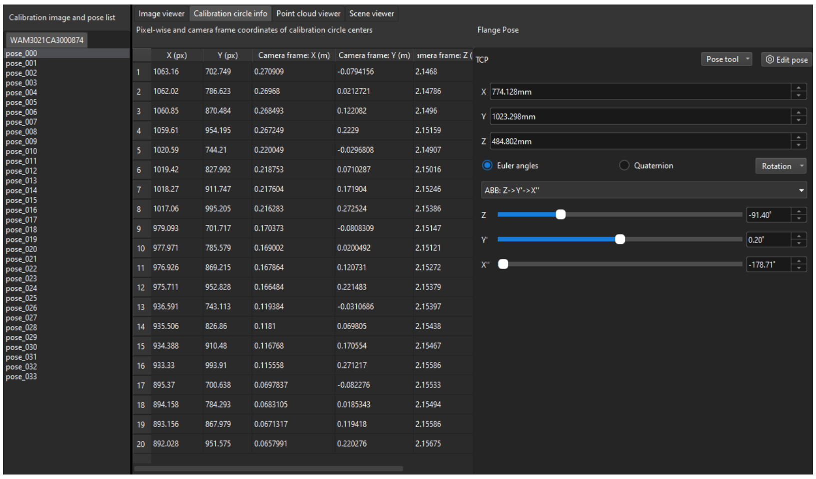
Collect more images and poses.

● Re-calculate extrinsic parameters

Re-calculate compensation parameters.

☑ Load the calibration data

5. In the **Calibrate extrinsic parameters** step of the calibration process, click a calibration point in the **Calibration image and pose list** panel (such as pose_000), and click the **Calibration circle info** tab to view the robot flange pose and the calibration circle data at this calibration point.



pose_000, pose_001, pose_002, pose_003, pose_004, pose_005, pose_006, pose_007, pose_008, pose_009, pose_010, pose_011, pose_012, pose_013, pose_014, pose_015, pose_016, pose_017, pose_018, pose_019, pose_020, pose_021, pose_022, pose_023, pose_024, pose_025, pose_026, pose_027, pose_028, pose_029, pose_030, pose_031, pose_032, pose_033

Pixel-wise and camera frame coordinates of calibration circle centers

	X (px)	Y (px)	Camera frame: X (m)	Camera frame: Y (m)	amera frame: Z (
1	1063.16	702.749	0.270909	-0.0794156	2.1468
2	1062.02	786.623	0.26968	0.0212721	2.14786
3	1060.85	870.484	0.268493	0.122082	2.1496
4	1059.61	954.195	0.267249	0.2229	2.15159
5	1020.59	744.21	0.220049	-0.0296808	2.14907
6	1019.42	827.992	0.218753	0.0710287	2.15016
7	1018.27	911.747	0.217604	0.171904	2.15246
8	1017.06	995.205	0.216283	0.272524	2.15386
9	979.093	701.717	0.170373	-0.0808309	2.15147
10	977.971	785.579	0.169002	0.0200492	2.15121
11	976.926	869.215	0.167864	0.120731	2.15272
12	975.711	952.828	0.166484	0.221483	2.15379
13	936.591	743.113	0.119384	-0.0310686	2.15397
14	935.506	826.86	0.1181	0.069805	2.15438
15	934.388	910.48	0.116768	0.170554	2.15467
16	933.33	993.91	0.115558	0.271217	2.15586
17	895.37	700.638	0.0697837	-0.082276	2.15533
18	894.158	784.293	0.0683105	0.0185343	2.15494
19	893.156	867.979	0.0671317	0.119418	2.15586
20	892.028	951.575	0.0657991	0.220276	2.15675

Flange Pose

TCP

X 774.128mm

Y 1023.298mm

Z 484.802mm

● Euler angles ○ Quaternion

ABB: Z->Y'->X''

Z -91.40°

Y' 0.20°

X'' -178.71°

In the figure, X (pixel) and Y (pixel) columns represent the pixel coordinates of the center points of the calibration circles in the current 2D image; Camera frame: X(m)/Y(m)/Z(m) columns represent the coordinates of the center points relative to the camera reference frame in the current depth image.

Please inspect the calibration data from the following aspects:

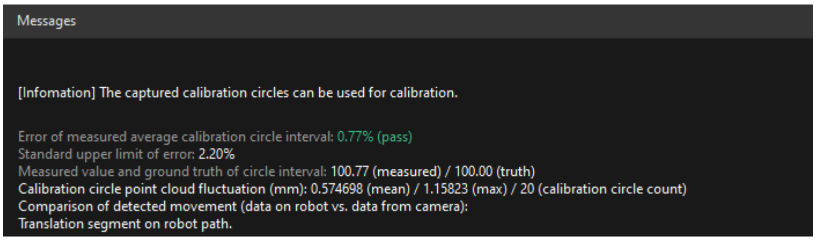
1. Check if the flange poses change between different sets of calibration points in the pyramid calibration path.

   > Only calibration points in the pyramid calibration path are checked, excluding the rotating sets of calibration points. This inspection is applied to calibration scenarios using the multiple random calibration board poses method to collect calibration data. In the **Calibration image and pose list** panel, select the calibration points one by one (such as pose_000), and then click the **Calibration circle info** tab to view the robot flange pose at the calibration point. When the robot moves along the pyramid calibration path, it will move along its base coordinate or flange without rotating. Therefore, the Euler angles will keep consistent throughout the entire procedure.
   >
   > Depending on the precision level of the robot, the Euler angles might fluctuate differently. If the fluctuation exceeds 1 degree, the robot might have lost its zero position or have bad precision.
   >
   > **Solution**: in this case, hand-eye calibration should not be continued. You should check the robot's zero position and fix the precision issue before continuing to calibrate.

2. Check if the "Error of measured average calibration circle interval" of the calibration points exceeds the standard value.

   In the **Calibration image and pose list** panel, select the calibration point, and view Error of measured average calibration circle interval in the Messages panel.

   > If it exceeds the standard value, the calibration point will turn to yellow for a warning. As part of the intrinsic parameters check result, Error of measured average calibration circle interval can partly reflect the current error in the camera intrinsic parameters.

   Messages

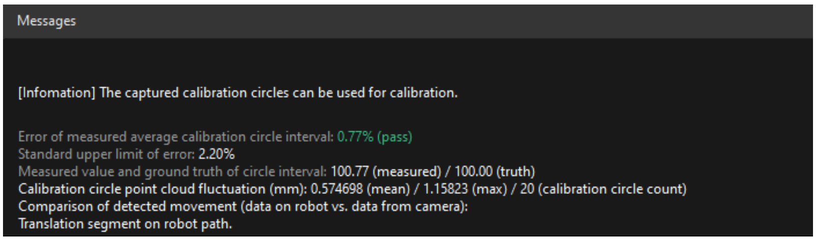
   ```
   [Infomation] The captured calibration circles can be used for calibration.

   Error of measured average calibration circle interval: 0.77% (pass)
   Standard upper limit of error: 2.20%
   Measured value and ground truth of circle interval: 100.77 (measured) / 100.00 (truth)
   Calibration circle point cloud fluctuation (mm): 0.574698 (mean) / 1.15823 (max) / 20 (calibration circle count)
   Comparison of detected movement (data on robot vs. data from camera):
   Translation segment on robot path.
   ```

   Solution: for projects with low precision requirements, intrinsic parameters with errors that slightly exceed the standard may still be used. For situations where high precision is required or the errors greatly exceed the standard, it is recommended to re-calibrate the intrinsic parameters or replace the camera.

   > Errors in intrinsic parameters will affect the calibration result, but if it is within the normal range, it is generally acceptable.

3. Check if the max value of the "Calibration circle point cloud fluctuation" exceeds 3 mm.

   In the **Calibration image and pose list** panel, select the calibration point, and view **Calibration circle point cloud fluctuation** in the Messages panel. **Calibration circle point cloud fluctuation** represents the overall fluctuation of the plane containing the coordinates of all the circles' center points. This value directly influences the accuracy of the extrinsic parameters.



```
[Infomation] The captured calibration circles can be used for calibration.

Error of measured average calibration circle interval: 0.77% (pass)
Standard upper limit of error: 2.20%
Measured value and ground truth of circle interval: 100.77 (measured) / 100.00 (truth)
Calibration circle point cloud fluctuation (mm): 0.574698 (mean) / 1.15823 (max) / 20 (calibration circle count)
Comparison of detected movement (data on robot vs. data from camera):
Translation segment on robot path.
```

The higher a project’s precision requirements are, the lower the fluctuation value should be. Generally, the max acceptable value for the fluctuation is 3 mm. If more than three calibration points have point cloud fluctuation greater than 3 mm, the whole accuracy is not acceptable. You should troubleshoot the issue and perform re-calibration.

Potential factors that might cause point cloud fluctuation are as follows:

Reason 1: The 3D camera exposure parameters are not optimized. It is likely that the **Surface Smoothing** and **Outliers Removal** options are disabled or that **Gain** is enabled. **Solution**: Change **Surface Smoothing** and **Outliers Removal** to "Normal", set **Gain** to zero, and adjust the 3D exposure parameters accordingly. If the point cloud of the calibration circles is still fragmented, on-site shading is required. **Reason 2**: When automatic calibration is performed in the ETH setup, the calibration board is not securely mounted on the robot flange. When the robot is too fast, the calibration board might vibrate during the calibration process, thus causing excessive point cloud fluctuation. **Solution**: Lower the speed limit on the robot, install the calibration board firmly, and extend the wait time before capturing a new image at each waypoint.

Re-calculate the Compensation Parameters

Compensation parameters refer to the parameter values calculated by dividing the calibration area into blocks. Generally, these parameter values will be calculated according to the actual maximum working range of the camera when the camera leaves the factory.

Due to the inconsistency between the actual on-site calibration conditions and those where the factory compensation parameters are calculated, the factory compensation parameters may not be applicable to all cases.

If you troubleshoot the calibration data according to the preceding sections, you need to re-calculate compensation parameters if the calibration results still do not meet project requirements.

- If you need to re-calculate compensation parameters, please contact Mech-Mind Technical Support.
- Before re-calculating compensation parameters, please back up a copy of the extri_param.json file and rename it to factory compensation parameter.

4.5.6.5. Calibration Troubleshooting and FAQs

Non-six-axis Robot Calibration

Four-axis Robot

Four-axis robots include truss/gantry robots, SCARA robots, and palletizing robots, etc.

Mech-Viz is only adapted to a small number of SCARA and palletizing robots (more robots will be supported in future).

- Tip fixing issue when using the TCP touch method

Mount the sharp tip on the end of the robot and check whether the tip is securely fixed at one point and installed in the center of the flange by rotating the fourth axis. Since the TCP of the four-axis robot cannot be calibrated using the tool, you need to manually measure the XYZ values of the TCP.

- Z-axis values of extrinsic parameters need to be adjusted for four-axis robots

When using the multiple random calibration board poses method, the lack of rotational degree of freedom will result in the lack of the rotational values during the calibration process. After the calibration is completed, you need to manually adjust the Z-axis value of the extrinsic parameters.

For details, refer to the how-to guide of the hand-eye calibration for four-axis robots.

- Calibration instructions for truss/gantry robots

For the instructions on the hand-eye calibration for truss/gantry robots, refer to the section Hand-Eye Calibration for Truss/Gantry Robots.

Calibration for Seven-axis Robots/Six-axis Robots with Slide Rail/Five-axis Robots

> The six-axis robot with slide rail mentioned here means that the slide rail is integrated into the robot teach pendant. Thus, it is equivalent to a seven-axis robot.

The TCP touch method is recommended for the above three types of robots.

- Requirements for calibration using the multiple random calibration board poses method for seven-axis robots

When there is no suitable sharp tip on site or the sharp tip cannot be mounted, the multiple random calibration board poses method can be used to calibrate the seven-axis robot.

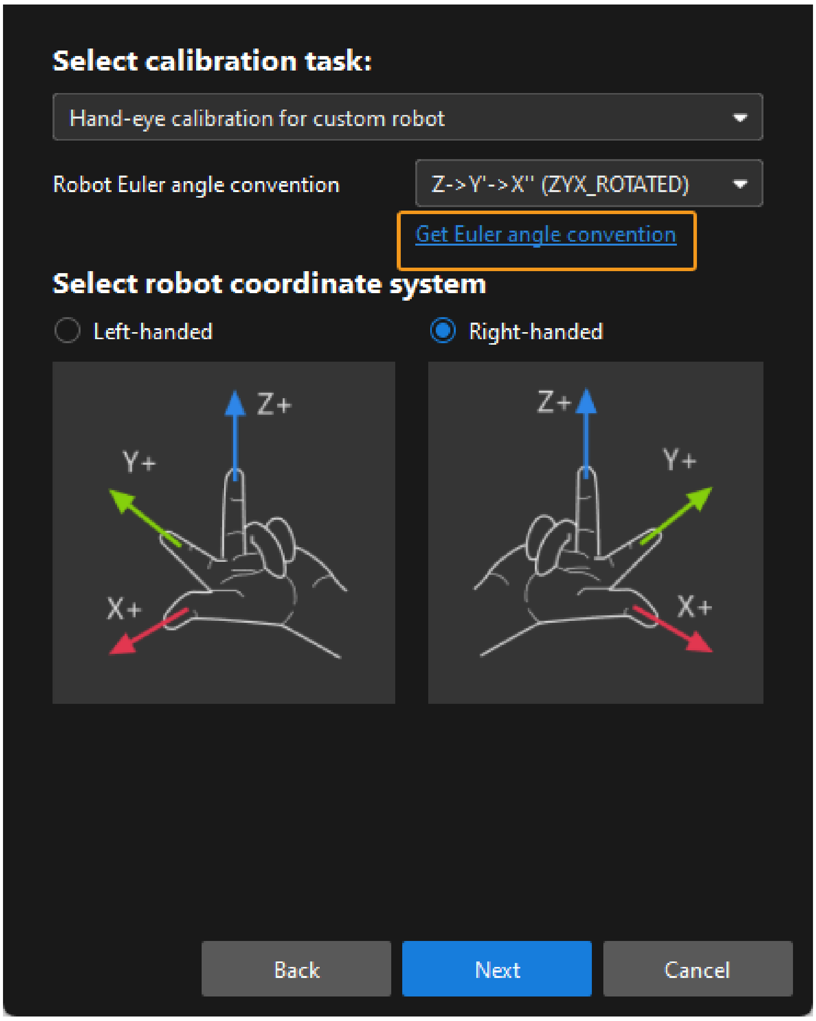
During the calibration process, you need to restrict the movement of one of the axes and idealize it as a six-axis robot. The rest of the operations are roughly the same as the calibration of the six-axis robot.

How to Calibrate When the Euler Angle Convention of the Robot is Uncertain

When the Euler angle convention of the robot is unknown, use the "Get Euler angle convention" tool on the calibration interface during the pre-calibration configuration.

The detailed instructions are as follows:

1. Open Mech-Vision, and click the **[Camera Calibration (Standard)]** button in the toolbar. The **Configuration before Calibration** window will be prompted.
2. In the **Select how to calibrate** window, select the **New calibration** radio button, and then click the **[Next]** button.
3. In the **Select calibration task** window, select **Hand-eye calibration for custom robot** from the drop-down list box, and click the **Get Euler angle convention**.



Select calibration task:

Hand-eye calibration for custom robot

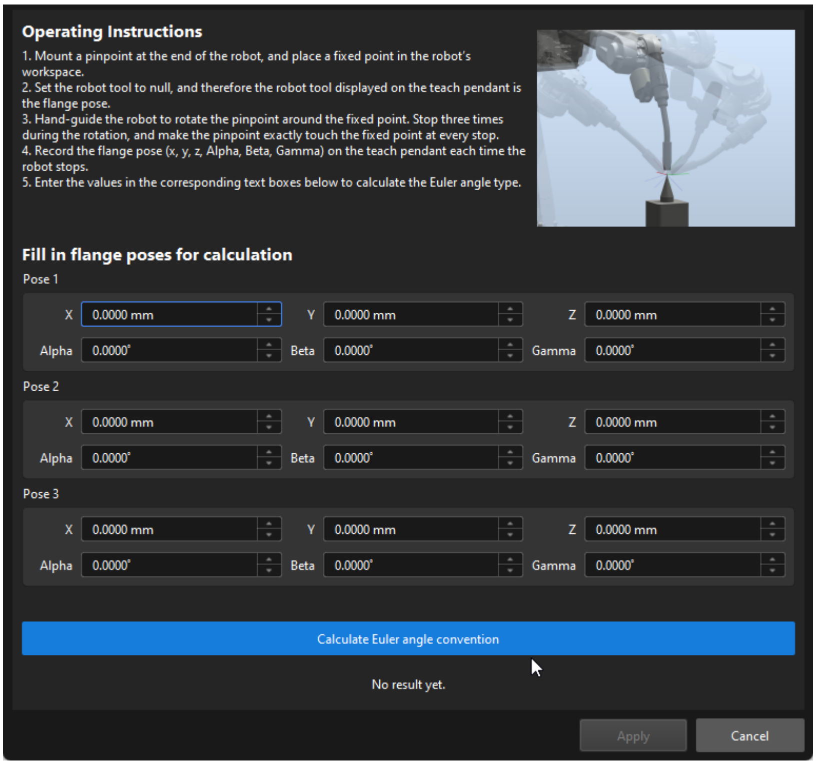
Robot Euler angle convention: Z->Y'->X'' (ZYX_ROTATED)

Get Euler angle convention

Select robot coordinate system

○ Left-handed ● Right-handed

4. Follow the operating instructions of this tool to record and enter three robot flange poses, and then click **[Calculate Euler angle convention]** to get the Euler angle convention of the robot.



Operating Instructions

1. Mount a pinpoint at the end of the robot, and place a fixed point in the robot's workspace.
2. Set the robot tool to null, and therefore the robot tool displayed on the teach pendant is the flange pose.
3. Hand-guide the robot to rotate the pinpoint around the fixed point. Stop three times during the rotation, and make the pinpoint exactly touch the fixed point at every stop.
4. Record the flange pose (x, y, z, Alpha, Beta, Gamma) on the teach pendant each time the robot stops.
5. Enter the values in the corresponding text boxes below to calculate the Euler angle type.

Fill in flange poses for calculation

Pose 1

X 0.0000 mm Y 0.0000 mm Z 0.0000 mm
Alpha 0.0000° Beta 0.0000° Gamma 0.0000°

Pose 2

X 0.0000 mm Y 0.0000 mm Z 0.0000 mm
Alpha 0.0000° Beta 0.0000° Gamma 0.0000°

Pose 3

X 0.0000 mm Y 0.0000 mm Z 0.0000 mm
Alpha 0.0000° Beta 0.0000° Gamma 0.0000°

No result yet.

How to Calibrate If There Is No Suitable Point on the Working Plane

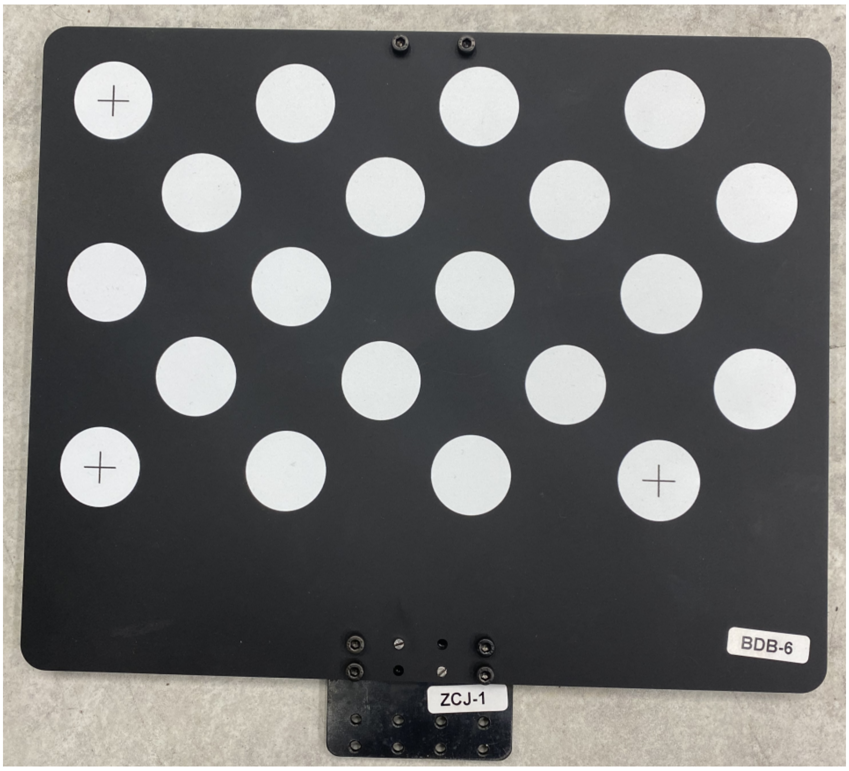
Since the accurate Euler angle convention of the robot is unknown, the Euler angle cannot be inputted correctly when the TCP touch method is used for calibration.

In this case, switch the pose type on the robot teach pendant to TCP to obtain the pose of the tip.

Let the tool tip touch the cross center points of three calibration circles in turn, read the values of XYZ on the teach pendant and input them. Choose any Euler angle convention and enter any constant value. Make sure that the Euler angles of the three input poses are the same.

Intolerable Point Cloud Fluctuation on Multiple Calibration Points, Which Cannot be Improved by Adjusting Camera Parameters

Check the calibration board to see if there are crosses which may cause large point cloud fluctuations, as shown in the following figure:



Cut out a few circles from A4 paper and cover them over the center of each calibration circle with a cross (do not cover the edge of the circles). The crosses' effect on point cloud fluctuation should be reduced after it is done.

If the above method still does not solve the issue, try to manually add more poses and delete the poses with severe point cloud fluctuations.

Misunderstandings about Calibration Path

- **The more calibration points, the better**

Too many calibration points may introduce abnormal points, leading to an increase in the overall error ratio.

When using the factory compensation parameters to calculate extrinsic parameters, determine the number of points on each layer according to the camera focal length, the size of the calibration board, etc:

- When the focal length is 300 to 2000 mm, it is recommended to use a 2*2 configuration, with four calibration points per layer. The number of layers is generally three. Use four layers for high stacks.
- When the focal length is 2000 to 3500 mm, it is recommended to use a 3*3 configuration, with nine calibration points per layer. The number of layers is generally three. Use four to five layers for high stacks.
- **The calibration range has to cover the entire working area**

When the factory compensation parameters are used, calibration should be carried out in layers around the center of focus and around the focal length of the camera.

When the factory compensation parameters are not used or when it is impossible to calibrate around the camera’s focal length, it is recommended to calibrate in the entire working area.

- **The extrinsic parameters in the calibrated area are accurate, while those in the excluded area are inaccurate**

The error point cloud generated after calibration shows the extrinsic parameter error in the calibrated area. However, it does not mean that the extrinsic parameters of the excluded area must be inaccurate. On the contrary, when the factory compensation parameters are used, the extrinsic parameters of the excluded area are also accurate.

4.6. Measurement Mode Guide

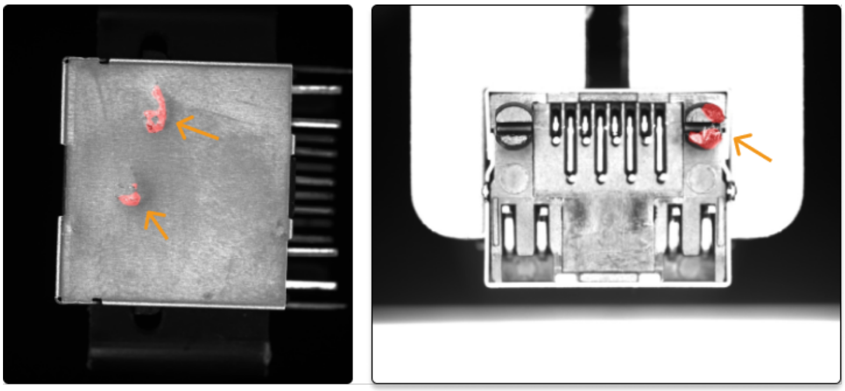
Quick Facts about Measurement Mode

Measurement, detection, alignment, and recognition are common applications in industrial industry.

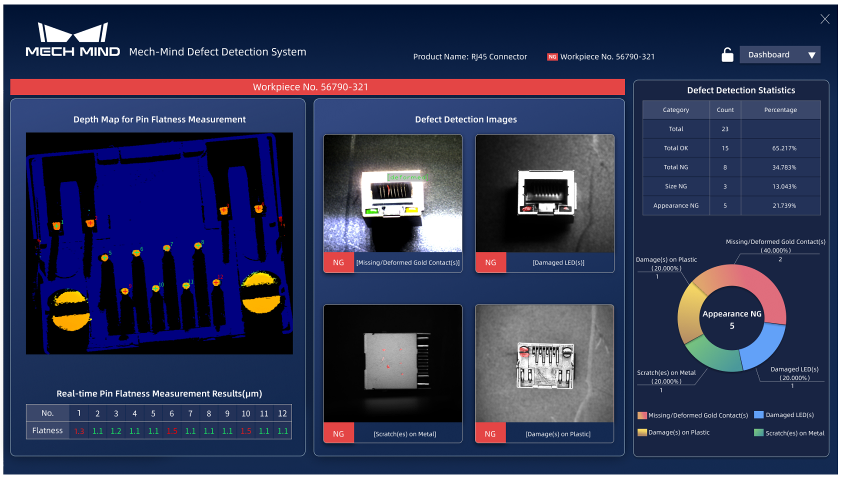
Therefore, Mech-Mind provides a measurement mode applicable to industries as electronic manufacturing, lithium-ion battery, automotive, home appliance, etc.

The Measurement Mode is integrated with various measurement algorithms. A sketchpad area is introduced for visualization and manual annotation. It also has a custom operator interface, which enables to configure **high precision measurement** and **defect detection** according to actual needs.

Defect detection



Custom operator interface


Mech-Mind Defect Detection System
Product Name: RJ45 Connector
NG Workpiece No. 56790-321
Dashboard
Workpiece No. 56790-321
Depth Map for Pin Flatness Measurement
Defect Detection Images
NG [Missing/Deformed Gold Contact(s)]
NG [Damaged LED(s)]
NG [Scratch(es) on Metal]
NG [Damage(s) on Plastic]
Missing/Deformed Gold Contact(s) (40.000%) 2
Damage(s) on Plastic (20.000%) 1
Appearance NG 5
Scratch(es) on Metal (20.000%) 1
Damaged LED(s) (20.000%) 1
🟥 Missing/Deformed Gold Contact(s)
🟦 Damaged LED(s)
🟨 Damage(s) on Plastic
🟩 Scratch(es) on Metal

Real-time Pin Flatness Measurement Results(μm)

No.	1	2	3	4	5	6	7	8	9	10	11	12
Flatness	1.3	1.1	1.2	1.1	1.1	1.5	1.1	1.1	1.1	1.5	1.1	1.1

Defect Detection Statistics

Category	Count	Percentage
Total	23	
Total OK	15	65.217%
Total NG	8	34.783%
Size NG	3	13.043%
Appearance NG	5	21.739%

Check the section below to **getting started with measurement mode** and complete your first application.

Getting Started with Measurement Mode

Check the section below to learn about the **Main Algorithms** in measurement mode.

Main Algorithms

4.6.1. Getting Started with Measurement Mode

Please prepare some objects with circular parts first. In this application, you can utilize the measurement mode to check if the radius of the circle conforms to the specification or not.

Create a New Project and Switch to Measurement Mode

1. Go to Settings › Options › Advanced, check Measurement project in Measurement Settings, and then **restart the software**;
2. Select File › New Project to create a new project;
3. Click Measurement Mode in the toolbar to switch to measurement mode.

Input

The source data can be input via the following Steps:

- Capture Images from Camera
- Laser Profiler
- Read Point Cloud V2
- Read Images V2

The Steps Laser Profiler and Read Point Cloud V2 should be connected with Orthographic Projection when utilizing Steps that belong to 2.5D Measuring algorithm.

Capture Images from Camera is used to input source data in the application below.

1. Locate Capture Images from Camera in Step Library and drag it into the graphical programming workspace.

   > You can click **[>]** in the lower right corner of **Step Library** to switch the display format of the Steps.

2. Please refer to Capture Images from Camera to configure the parameters in this Step.

Processing

This example aims to check if the circular part of the object conforms to the specification or not, so the Step Measure Circles is used.

1. Locate the Step:

   Go to Measuring ¬ 2D Measuring in the Step Library, and drag the Step Measure Circles into the graphical programming workspace.

2. Connect Steps:

   Select Measure Circles, and then select **Capture Images from Camera_1_Camera Color Image** as Input 1 (Color Image) in the Step Input Source Selection panel.

   > The type of input data should be consistent with that of the input port. If the types of data and port do not match, an alert will pop up.

3. Run the project:

   Click **[Run]** in the top middle of the graphical programming workspace or click ⏬ of the Step Capture Images from Camera to run the project. Then the color image will appear in the sketchpad area.

4. Select the circular part for detection:

   Click Measure Circles first. The select frame is in the upper left corner of the image. It is recommended to zoom out the image first and then adjust the frame to a proper size. Then you can drag it to the circular part in the image and therefore select an ROI.

   The sketchpad is used for **visualization**, **selecting an ROI**, and **adding points**:

   a. You can switch the image display type between color image, gray image, and binary image by clicking the buttons in the upper left corner.

   b. You can click the buttons in the upper right corner of the sketchpad to zoom in or out the image, restore the image scale to 100%, and adjust the Pen Width, Background Brightness, Background Contrast and Font Size.

   > You can also scroll upwards or downwards to zoom in or out the image.

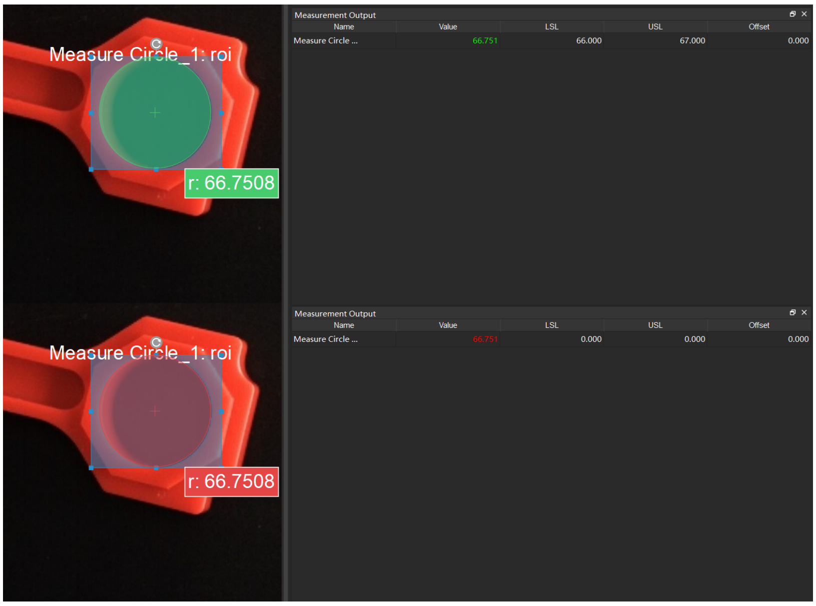
Output

1. Read the measurement:

  - The radius of the circle in the ROI will be displayed in the sketchpad area in real-time;
  - The value will be displayed in the **Measurement Output** panel as well.

2. Determine whether the measurement conforms to the specification or not:

   The measurement result is displayed in red, which represents that the result failed to meet the requirement. In this example project, it is because that the LSL (lower specification limit) and USL (upper specification limit) have not been set:



Measurement Output

Name	Value	LSL	USL	Offset
Measure Circle ...	66.751	66.000	67.000	0.000

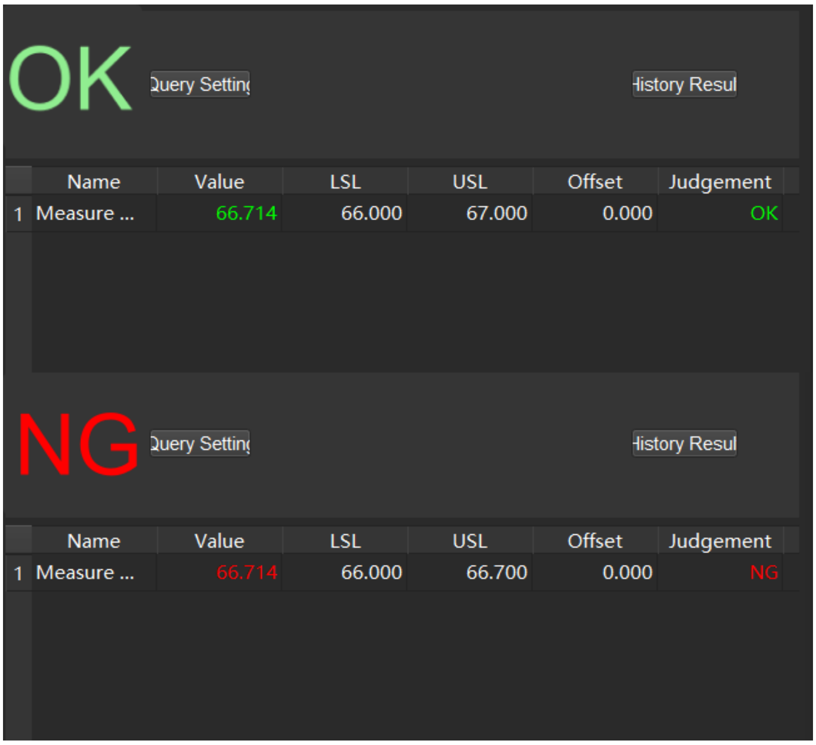
Measurement Output

Name	Value	LSL	USL	Offset
Measure Circle ...	66.751	0.000	0.000	0.000

   After setting the LSL and USL according to actual situation, if the measurement falls within the lower and upper limits, the value will be displayed in green, as shown below.

3. Result View:

   The **Result View** in Measurement Mode provides a more user-friendly display of the measurement result. Click + ▾, and select **[Result View]**. The result will be displayed in the upper left corner: OK represents that the result conforms to the specification, while NG represents the opposite.



	Name	Value	LSL	USL	Offset	Judgement
1	Measure ...	66.714	66.000	67.000	0.000	OK

	Name	Value	LSL	USL	Offset	Judgement
1	Measure ...	66.714	66.000	66.700	0.000	NG

4. Operator Interface:

   Click **[Operator Interface (Custom)]** in the toolbar to open the Operator Interface. If you need a custom development of this interface, please contact Mech-Mind Technical Support.

Now you have completed your first application using Measurement Mode!

General Workflow in Measurement Mode

By completing the first application, the general workflow in measurement mode can be summarized as follows.

1. Create a New Project and switch to measurement mode.
2. Connect the Steps to program and configure parameters in Steps. The main algorithms in measurement mode will be introduced in the next section.
3. Select an ROI and add base points or detect points in the sketchpad area.
4. Open the Result View, select the output Step and set the LSL and USL.
5. Check the real-time product image, measurement and statistics in the custom operator interface.

4.6.2. Main Algorithms

Different Steps are integrated with various algorithms.

Main Algorithms:

1. 2D measuring algorithm which is used for measurement, positioning and other applications based on 2D images.

Relevant Steps: Measure Circles , Measure Distances Circles to Circles , Measure Distances Circles to Segments , Measure Distances Points to Circles , Measure Distances Points to Points , Measure Distances Points to Segments , Measure Distances Segments to Segments , Measure Longest Line Segment , Measure Angles Segments to Segments , etc.

2. 2.5D measuring algorithm which is used to measure height difference based on depth maps.

   Relevant Steps: Measure Height Difference Point to Point , Measure Height Differences Points to Baseline , Measure Height Differences Points to Plane , etc.

3. 3D measuring algorithm which is used to measure distance based on point clouds.

   Relevant Steps: Calc Diagonal Lengths , etc.

4. Global dimension and tolerance algorithm which is used to measure the flatness of a surface.

   Relevant Steps: Calc Flatness Error , Calc Parallelism Error , etc.

5. Deep Learning algorithm which is used for defect detection and other applications.

4.7. Supplementary Tools

This chapter will introduce the tools in Mech-Vision. These tools are used in some special application scenarios in order to improve the efficiency and accuracy of vision solutions.

- camera-viewer.pdf
- Set Static Background for Project
- Set ROI
- Set the Scene Point Cloud
- Deep Learning Server
- Matching Model and Pick Point Editor
- Deep Learning Model Package Management Tool
- Parameter Recipe
- Save Data
- Log

4.7.1. Error Analysis Tool

During the actual production process, various issues can arise, such as inaccuracies in camera intrinsic parameters and imprecise picking. These problems often stem from a multitude of factors that can be traced to different stages of production.

This section covers how to use the Error Analysis Tool to troubleshoot the above issues.

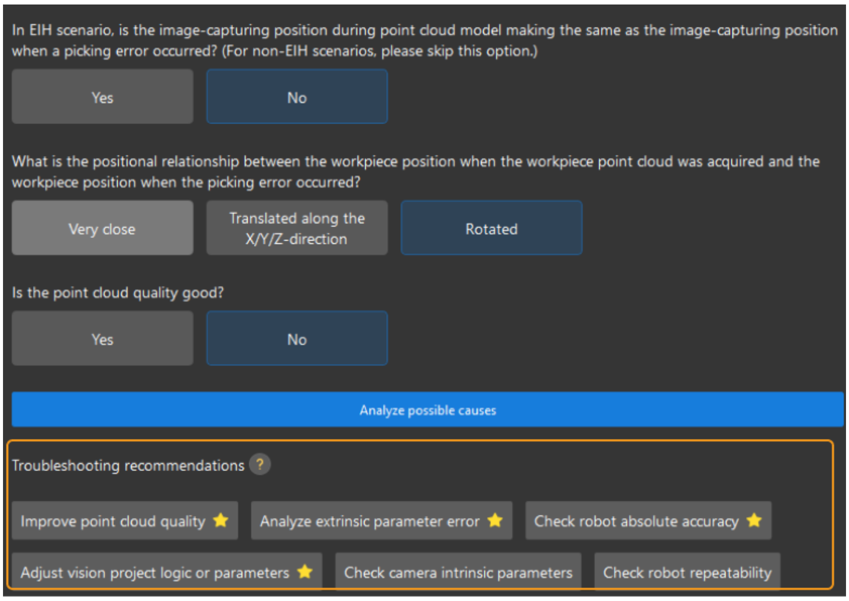
Introduction

When an error occurs during recognition and positioning, this tool can be used for troubleshooting. Based on the on-site situation you described, the tool will recommend specific items to check, including camera accuracy, robot absolute accuracy and repeatability, and the result of extrinsic parameters after calibration. This tool cannot be used to check the project logic or analyze the matching accuracy.

After opening the tool, you will answer the following questions to analyze the possible causes.

1. In EIH scenario, is the image-capturing position during point cloud model making the same as the image-capturing position when a picking error occurred?
2. What is the positional relationship between the workpiece position when the workpiece point cloud was acquired and the workpiece position when the picking error occurred?
3. Is the point cloud quality good?

After completing the questions, click the **[Analyze possible causes]** button, and the tool will provide some **troubleshooting recommendations** for reference.



In EIH scenario, is the image-capturing position during point cloud model making the same as the image-capturing position when a picking error occurred? (For non-EIH scenarios, please skip this option.)

Yes [✓ No]

What is the positional relationship between the workpiece position when the workpiece point cloud was acquired and the workpiece position when the picking error occurred?

Very close | Translated along the X/Y/Z-direction | [✓ Rotated]

Is the point cloud quality good?

Yes [✓ No]

Analyze possible causes
Troubleshooting recommendations ?
Improve point cloud quality ★
Analyze extrinsic parameter error ★
Check robot absolute accuracy ★
Adjust vision project logic or parameters ★
Check camera intrinsic parameters
Check robot repeatability

The item with a star icon is most likely to cause the error, so please check it carefully.

Main Functions

This tool can be used to analyze the causes of the errors related to cameras, robots, and extrinsic parameters.

For camera error analysis, please read the section below.

Camera

For robot error analysis, please read the section below.

Robot

For extrinsic parameter error analysis, please read the section below.

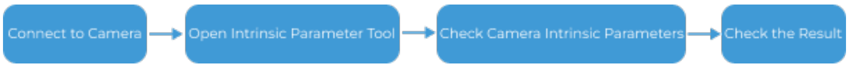
Extrinsic parameters

4.7.1.1. Camera

This section describes how to check the intrinsic parameters of the camera.

The accuracy of the camera’s intrinsic parameters significantly impacts extrinsic parameter accuracy and positioning accuracy. If the project requires high accuracy, please select the high-accuracy mode in Mech-Eye Viewer to check the intrinsic parameters, which guarantees optimal positioning performance.

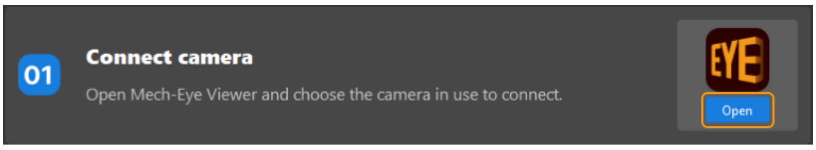
The following figure shows the checking process.


Connect to Camera
Open Intrinsic Parameter Tool
Check Camera Intrinsic Parameters
Check the Result

1. Open Mech-Eye Viewer and choose the camera in use to connect.
2. Click Tools in the menu bar of Mech-Eye Viewer and then select “Intrinsic Parameter Tool” to open the tool.
3. Follow the steps suggested in “Intrinsic Parameter Tool” of Mech-Eye Viewer to check intrinsic parameters.
4. Improve the accuracy according to the check result and suggestions.

Connect to Camera

1. Click the **[Open]** button on the right to open Mech-Eye Viewer.



01

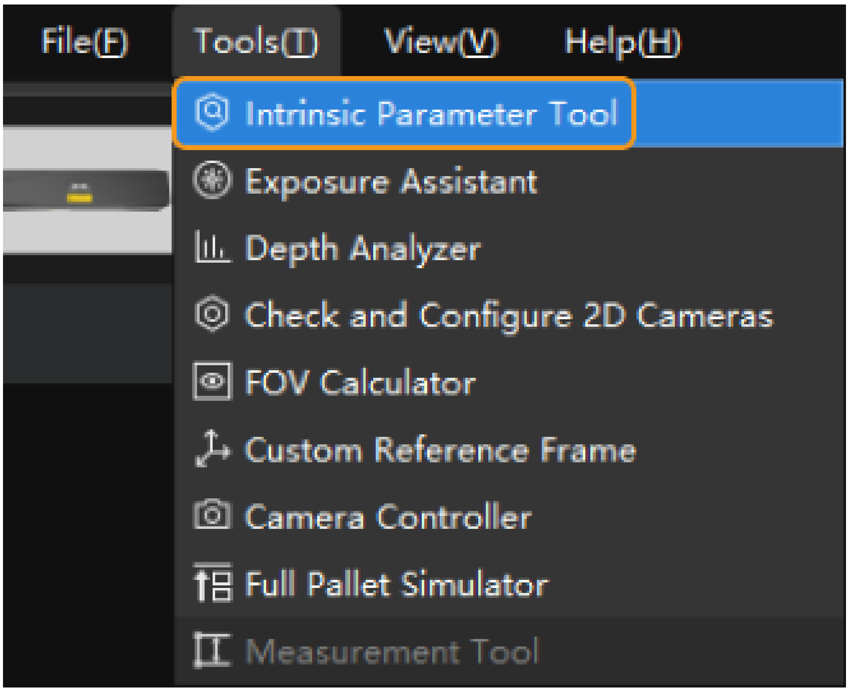
Connect camera

Open Mech-Eye Viewer and choose the camera in use to connect.

2. Find the camera you want to connect in Mech-Eye Viewer and click **[Connect]**.

Open Intrinsic Parameter Tool

Click Tools in the menu bar of Mech-Eye Viewer and then select “Intrinsic Parameter Tool” to open the tool.



Check Camera Intrinsic Parameters

You can refer to Intrinsic Parameter Tool for more instructions.

Check the Result

You can improve the accuracy according to the check result and suggestions.

4.7.1.2. Robot

- Check robot absolute accuracy
  - The absolute accuracy of the robot can greatly affect the result of extrinsic parameter calibration.
  - In picking tasks with a relatively large working range, such as picking on planes and picking in deep bins, the robot absolute accuracy will also result in varying degrees of picking errors.
- Check robot repeatability
  - In EIH scenarios, the image-capturing position is usually fixed, and therefore the robot repeatability significantly impacts the extrinsic parameter accuracy.
  - In tasks with a relatively small working range, such as assembly and conveyor belt unloading, attention should be paid to whether the robot repeatability meets the requirement.

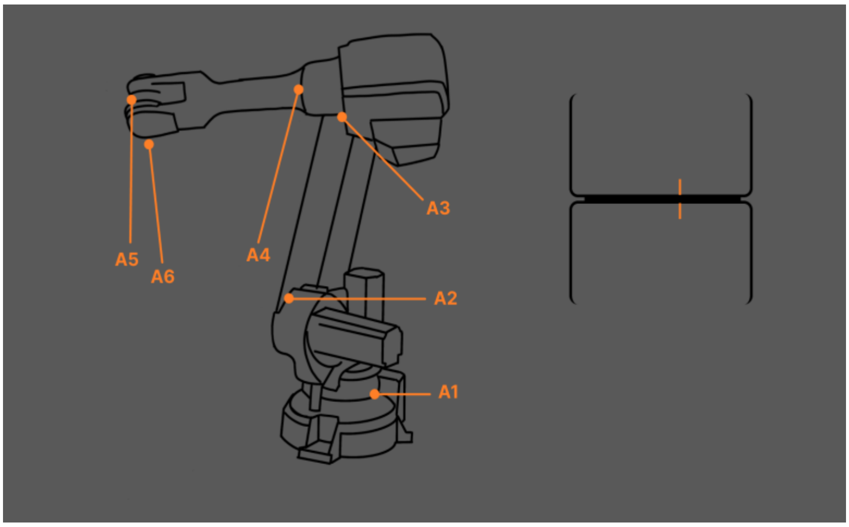
4.7.1.2.1. Check Robot Absolute Accuracy

This section describes how to check the absolute accuracy of the robot.

Ways to check the absolute accuracy are as follows.

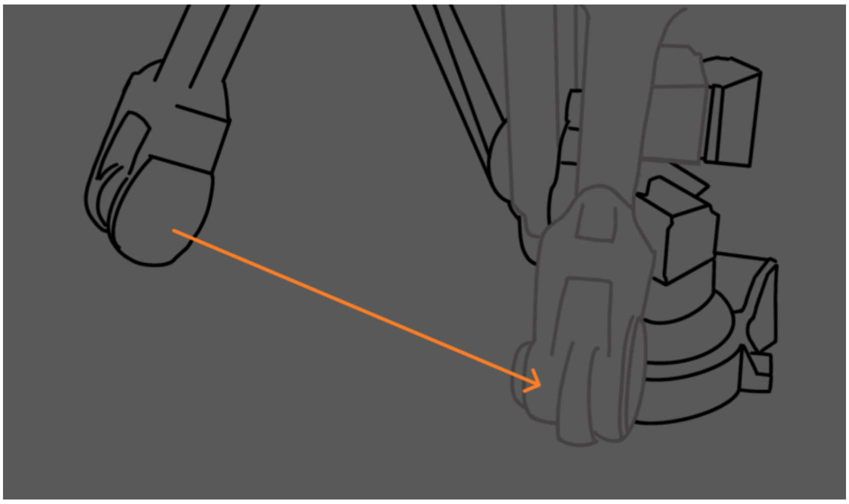
Check the Zero Position of Each Axis

Refer to the manufacturer’s instructions for checking zero positions on the specific robot in use.



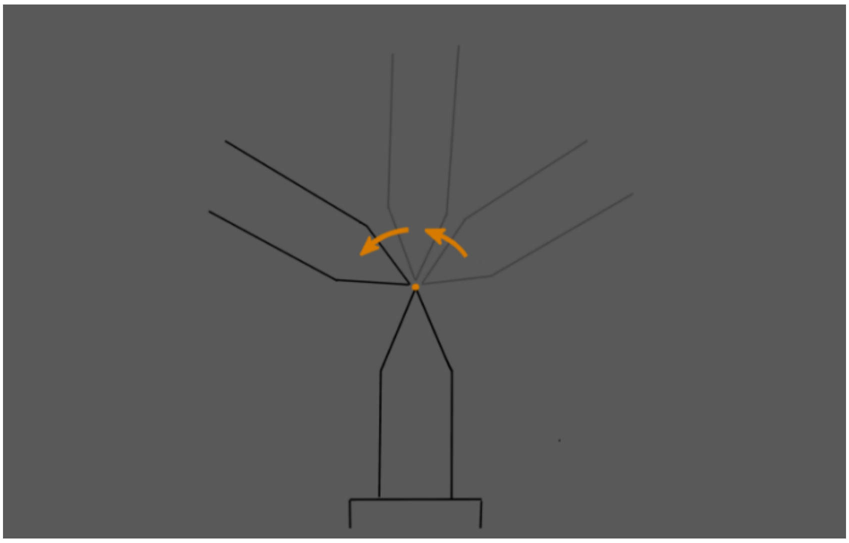
Measure the Error in the Robot’s Movement Distance

Use the teach pendant to control the robot to move a certain distance along a fixed direction (e.g., X or Y-direction) in the workspace. After the movement, compare the theoretical distance with the actual distance the robot moved and measure the difference.



For example, the movement distance displayed on the teach pendant is 1000 mm. However, the measured distance the robot actually moved is 998 mm. Then the error of the robot’s movement distance is 2 mm.

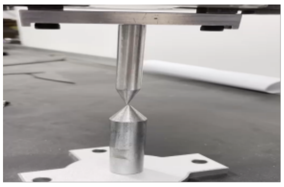
Check By Rotating around a Pinpoint

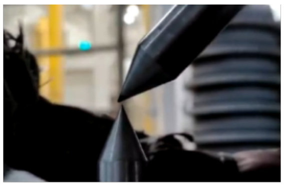


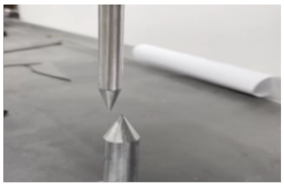
As shown in the figure above, the absolute accuracy of the robot can be checked by rotating the TCP around a pinpoint. The instructions are as follows.

1. Mount a pinpoint at the end flange of the robot.
2. Enter the coordinates of the current pinpoint on the teach pendant to set it as the TCP of the robot.
3. Mount and fasten another pinpoint in the robot workspace.
4. Move the robot using the teach pendant and make the robot TCP exactly touch the pinpoint in the workspace.
5. Rotate the robot end around the pinpoint, check if the two pinpoints are still in contact, and record the distance fluctuation range if the two pinpoints depart.

As shown in the figure below, in Figure 1, the pinpoint mounted at the robot flange coincides with another pinpoint, indicating good absolute accuracy. In Figure 2, the pinpoint mounted at the robot flange slightly departs from another pinpoint, indicating poor absolute accuracy. In Figure 3, the pinpoint mounted at the robot flange significantly departs from another pinpoint, indicating very poor absolute accuracy.


1


2


3

Other Checking Suggestions

Absolute accuracy is the primary factor influencing the accuracy of recognition and picking. If the robot’s absolute accuracy, as determined by the previously outlined steps, cannot meet the requirements, please contact the robot manufacturer for accuracy correction first.

If the robot manufacturer is unable to rectify the accuracy or the corrected accuracy still cannot meet the requirements, consider creating different point cloud models at different positions or jogging the robot to record different pick points to compensate for the robot’s absolute accuracy.

4.7.1.2.2. Check Robot Repeatability

This section describes how to check the robot repeatability.

Check with Image-Capturing Position in EIH Setup

This feature is used to check the robot repeatability at the image-capturing position in EIH setup. In the picking process, if the robot has poor repeatability at the image-capturing position, the accuracy of the extrinsic parameters will be affected.

Build Project to Obtain Calibration Board Poses

1. Add the “Capture Images from Camera” Step.

   Create a new project and add the “Capture Images from Camera” Step, or select the Step in an opened project. Then click the **[Select camera]** button in the Step Parameters panel to select and connect the camera.

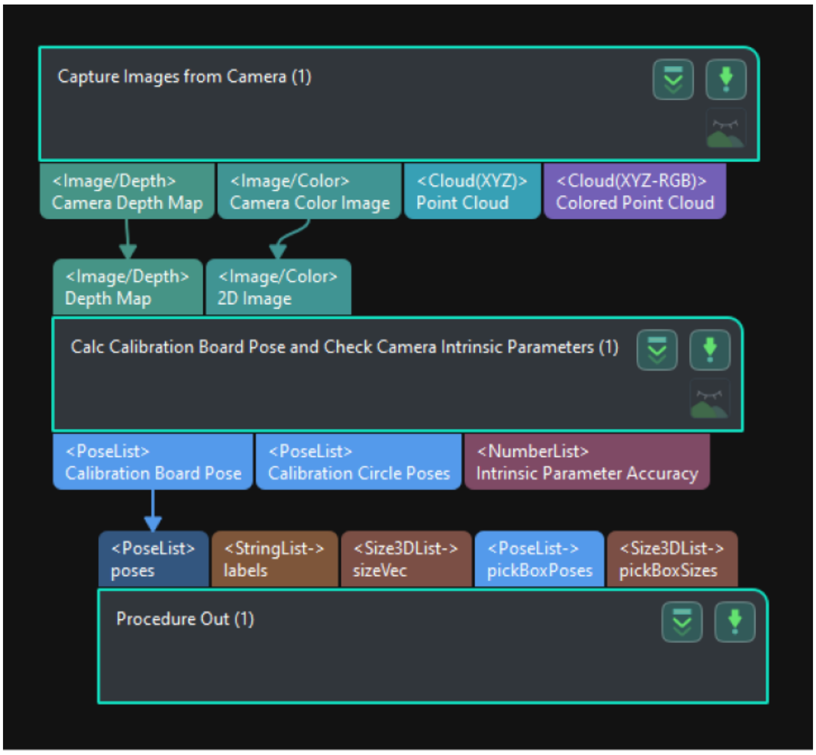
2. Add the “Calc Calibration Board Pose and Check Camera Intrinsic Parameters” Step.

   Add the “Calc Calibration Board Pose and Check Camera Intrinsic Parameters” Step and connect it with the “Capture Images from Camera” Step.

3. Add the “Procedure Out” Step.

   Add the “Procedure Out” Step, set Port Type to “Predefined (vision result),” and connect the “Calc Calibration Board Pose and Check Camera Intrinsic Parameters” Step with the “Procedure Out” Step to output the calculated pose of the calibration board.

   The built project is shown as below.

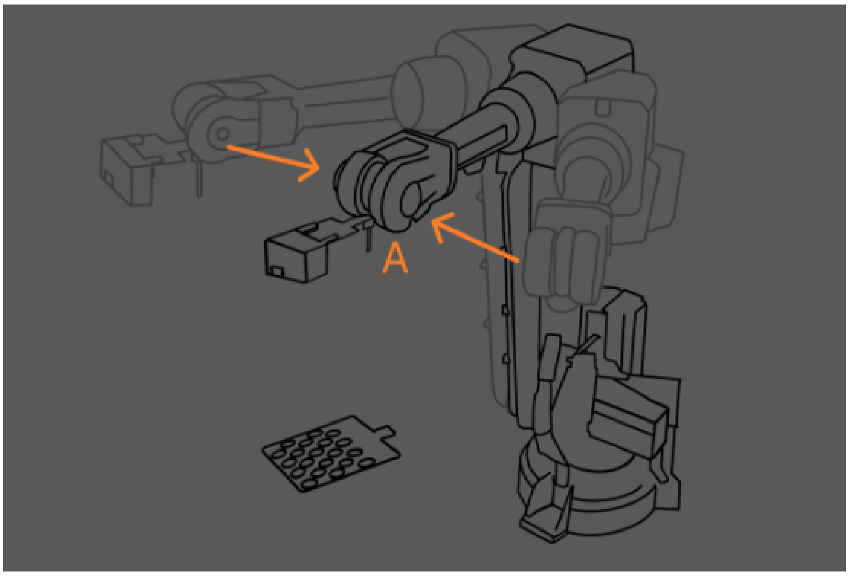

Capture Images from Camera (1)
<Image/Depth> Camera Depth Map
<Image/Color> Camera Color Image
<Cloud(XYZ)> Point Cloud
<Cloud(XYZ-RGB)> Colored Point Cloud
<Image/Depth> Depth Map
<Image/Color> 2D Image
Calc Calibration Board Pose and Check Camera Intrinsic Parameters (1)
<PoseList> Calibration Board Pose
<PoseList> Calibration Circle Poses
<NumberList> Intrinsic Parameter Accuracy
<PoseList> poses
<StringList-> labels
<Size3DList-> sizeVec
<PoseList-> pickBoxPoses
<Size3DList-> pickBoxSizes
Procedure Out (1)

4. Run the project and check the result.

   Run the project and check the calculation result of the calibration board pose in the Log panel at the bottom.

Robot Carries the Camera to Capture Images of the Calibration Board from Different Positions

To obtain the calibration board poses from different positions, the robot should carry the camera to capture images of the calibration board from different positions, as shown below.



1. Record the position where the camera captures the image for the first time as point A and run the project at that point. Then obtain and record the calibration board pose in the camera reference frame.

2. Use the teach pendant to move the robot to another position next to the image-capturing point in the robot picking path. It is recommended to include both translation and rotation in the movement.
3. Use the teach pendant to move the robot to point A. Then obtain and record the calibration board pose.
4. Repeat steps 2 and 3 to obtain and record more calibration board poses. The poses obtained with the steps above will be used to check the robot repeatability.

Add Poses and Analyze Error

1. Click **[+]** to add a pose.
2. Click **[Analyze error]** to view the result.

   If the error is too large, please add fixed waypoints near positions that require high accuracy, and then capture the images of the calibration board and analyze the error again. For example, add a waypoint near the image-capturing position or pick point. In this way, when the robot reaches that waypoint from a distance, the picking error due to poor repeatability can be reduced.

Check with Fixed Pick Point in ETH Setup

This feature is used to check the robot repeatability at the fixed pick point in ETH setup. In ETH scenarios, the robot needs to reach a fixed pick point from different waypoints, and poor robot repeatability will lead to poor picking accuracy.

Build Project to Obtain Calibration Board Poses

1. Add the “Capture Images from Camera” Step.

   Create a new project and add the “Capture Images from Camera” Step, or select the Step in an opened project. Then click the **[Select camera]** button in the Step Parameters panel to select and connect the camera.

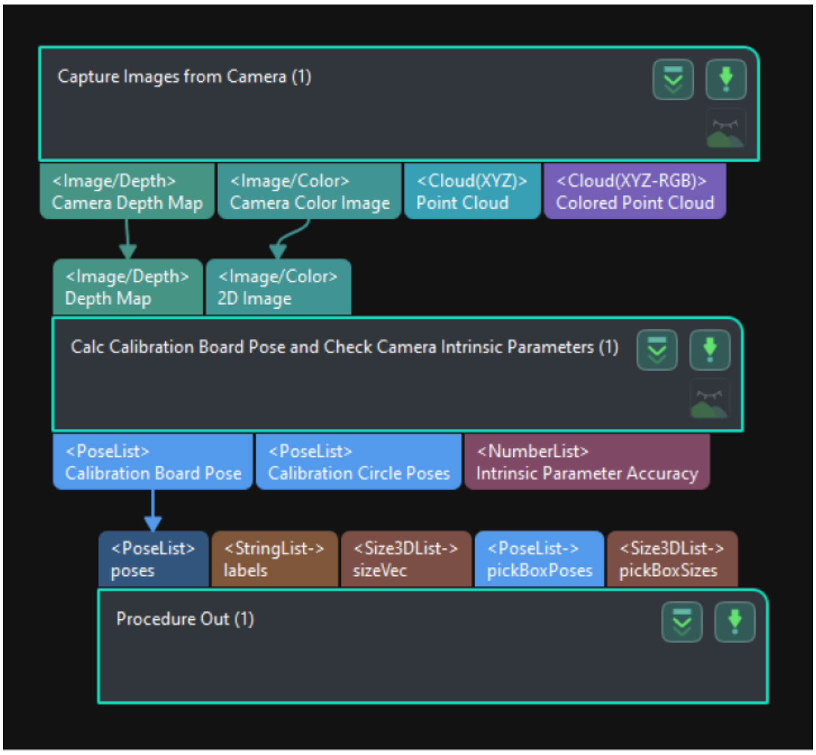
2. Add the “Calc Calibration Board Pose and Check Camera Intrinsic Parameters” Step.

   Add the “Calc Calibration Board Pose and Check Camera Intrinsic Parameters” Step and connect it with the “Capture Images from Camera” Step.

3. Add the “Procedure Out” Step.

   Add the “Procedure Out” Step, set Port Type to “Predefined (vision result),” and connect the “Calc Calibration Board Pose and Check Camera Intrinsic Parameters” Step with the “Procedure Out” Step to output the calculated pose of the calibration board.

   The built project is shown as below.

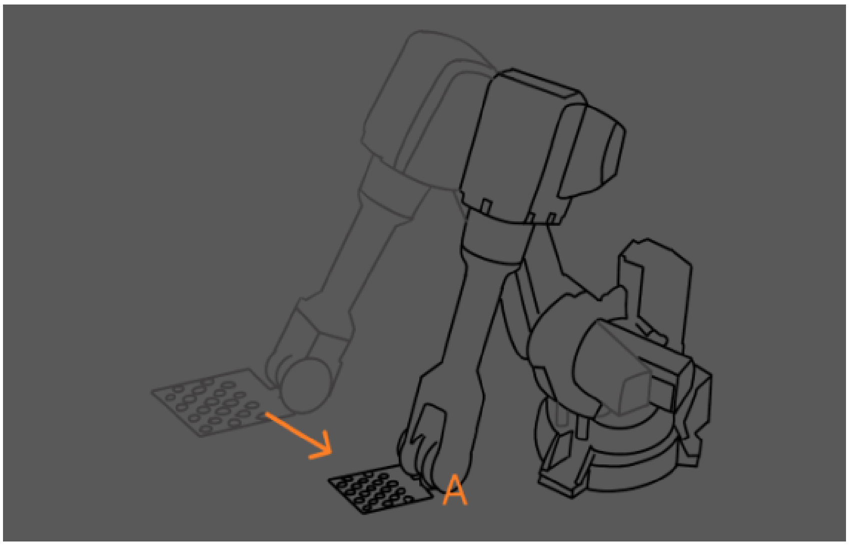

Capture Images from Camera (1)
<Image/Depth> Camera Depth Map
<Image/Color> Camera Color Image
<Cloud(XYZ)> Point Cloud
<Cloud(XYZ-RGB)> Colored Point Cloud
<Image/Depth> Depth Map
<Image/Color> 2D Image
Calc Calibration Board Pose and Check Camera Intrinsic Parameters (1)
<PoseList> Calibration Board Pose
<PoseList> Calibration Circle Poses
<NumberList> Intrinsic Parameter Accuracy
<PoseList> poses
<StringList-> labels
<Size3DList-> sizeVec
<PoseList-> pickBoxPoses
<Size3DList-> pickBoxSizes
Procedure Out (1)

4. Run the project and check the result.

   Run the project and check the calculation result of the calibration board pose in the Log panel at the bottom.

Robot Carries the Calibration Board and Moves to the Fixed Pick Point from Different Poses

To check the robot repeatability at the pick point, the robot should carry the calibration board and move to the fixed pick point from different poses, as shown below.



1. Record the position where the robot performs picking for the first time as point A and run the project at that point. Then obtain and record the calibration board pose in the camera reference frame.
2. Use the teach pendant to move the robot to another position next to the image-capturing point in the robot picking path. It is recommended to include both translation and rotation in the

movement.

3. Use the teach pendant to move the robot to point A. Then obtain and record the calibration board pose.
4. Repeat steps 2 and 3 to obtain and record more calibration board poses. The poses obtained with the steps above will be used to check the robot repeatability.

Add Poses and Analyze Error

1. Click **[+]** to add more poses.
2. Click **[Analyze error]** to view the result.

   If the error is too large, please add fixed waypoints near positions that require high accuracy, and then capture the images of the calibration board and analyze the error again. For example, add a waypoint near the image-capturing position or pick point. In this way, when the robot reaches that waypoint from a distance, the picking error due to poor repeatability can be reduced.

4.7.1.3. Extrinsic Parameters

- Analyze extrinsic parameter error in EIH setup

Analyze the extrinsic parameter error in the EIH scenario by examining calibration board poses that are acquired at different image-capturing positions.This analysis result is a comprehensive one that is affected by the extrinsic parameter calibration, robot accuracy, and robot repeatability.

- Analyze extrinsic parameter error in ETH setup

Analyze the extrinsic parameter error in the ETH scenario by adding a calibration board point cloud in the scene and checking the position of the TCP relative to the circle on the calibration at different positions. This analysis result is a comprehensive one that is affected by the extrinsic parameter calibration, robot accuracy, and robot repeatability.

4.7.1.3.1. Analyze Extrinsic Parameter Error in EIH Setup

Build Project to Obtain Calibration Board Poses

1. Add the “Capture Images from Camera” Step.

   Create a new project and add the “Capture Images from Camera” Step, or select the Step in an opened project. Click the **[Select camera]** button in the Step Parameters panel to select and connect the camera. Then select the Camera Calibration Parameters to be checked.

2. Add the “Calc Calibration Board Pose and Check Camera Intrinsic Parameters” Step.

   Add the “Calc Calibration Board Pose and Check Camera Intrinsic Parameters” Step and connect it with the “Capture Images from Camera” Step.

3. Add the “Transform Poses” Step.

   Add the “Transform Poses” Step and set Transformation Type to “CameraToRobot.” Then connect the “Calc Calibration Board Pose and Check Camera Intrinsic Parameters” Step with this Step.

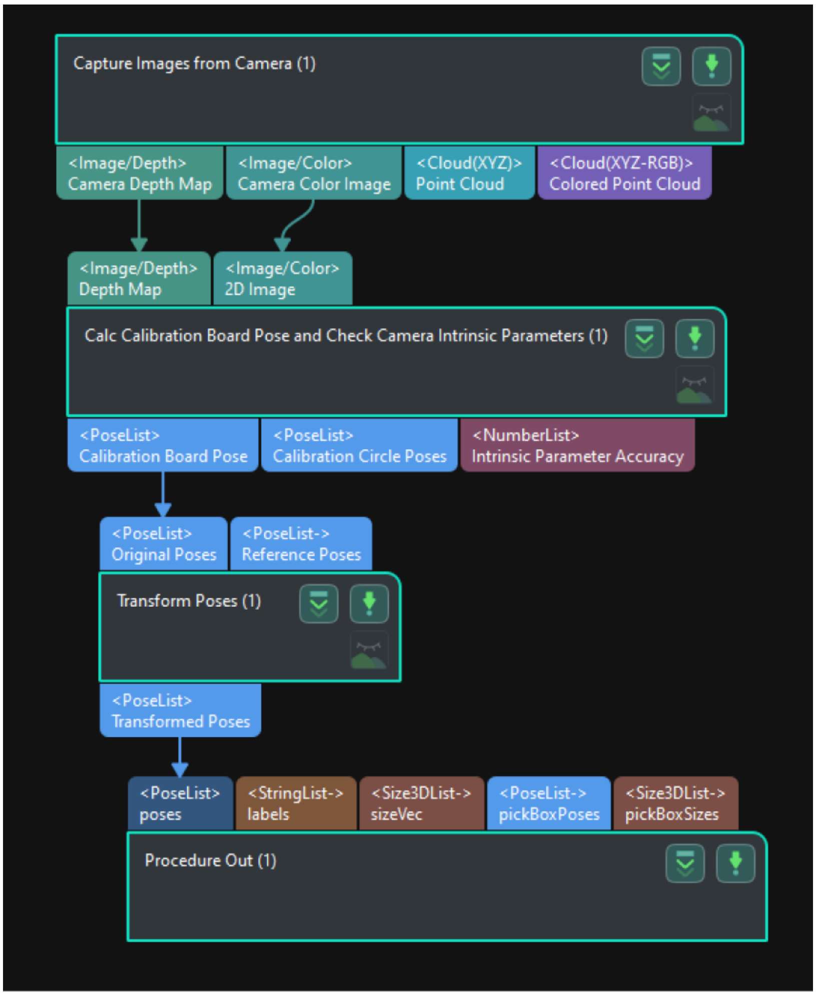
   If you are using a gantry robot, please replace the “Transform Poses” Step with the

"Transform Poses for Truss" Step.

4. Add the "Procedure Out" Step.

   Add the "Procedure Out" Step, set Port Type to "Predefined (vision result)," and connect the "Transform Poses" Step with the "Procedure Out" Step to output the calculated pose of the calibration board.

   The built project is shown as below.

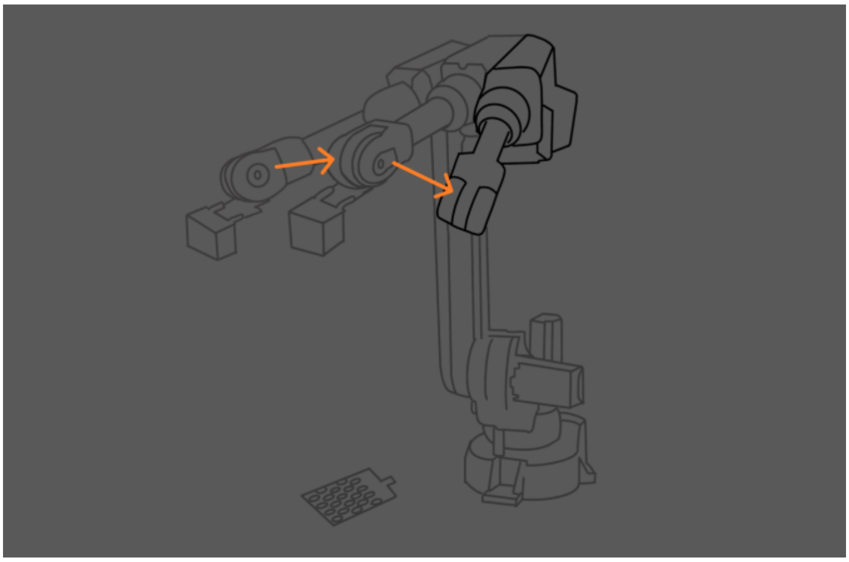

Capture Images from Camera (1)
<Image/Depth> Camera Depth Map
<Image/Color> Camera Color Image
<Cloud(XYZ)> Point Cloud
<Cloud(XYZ-RGB)> Colored Point Cloud
<Image/Depth> Depth Map
<Image/Color> 2D Image
Calc Calibration Board Pose and Check Camera Intrinsic Parameters (1)
<PoseList> Calibration Board Pose
<PoseList> Calibration Circle Poses
<NumberList> Intrinsic Parameter Accuracy
<PoseList> Original Poses
<PoseList-> Reference Poses
Transform Poses (1)
<PoseList> Transformed Poses
<PoseList> poses
<StringList-> labels
<Size3DList-> sizeVec
<PoseList-> pickBoxPoses
<Size3DList-> pickBoxSizes
Procedure Out (1)

5. Run the project and check the result.

   Run the project and check the calculation result of the calibration board pose in the Log panel at the bottom.

Robot Carries the Camera to Capture Images of the Calibration Board from Different Positions

The robot carries the camera and moves to different positions in the workspace. It is recommended to include both translation and rotation in the movement. The camera will capture images after the project is run, and then the pose data of the calibration board will be recorded and used to analyze the extrinsic parameter error.



Add Poses and Analyze Error

1. Click **[+]** to add more poses.
2. Click **[Analyze error]** to view the result.

   If the error is too large, please add fixed waypoints near positions that require high accuracy, and then capture the images of the calibration board and analyze the error again. For example, add a waypoint near the image-capturing position or pick point. In this way, when the robot reaches that waypoint from a distance, the picking error due to poor repeatability can be reduced.

4.7.1.3.2. Analyze Extrinsic Parameter Error in ETH Setup

Ensure the Calibration Board Has Been Mounted

To obtain the pose of the calibration board and the scene point cloud containing calibration board for the ETH extrinsic parameter error evaluation, it is necessary to check the calibration board before proceeding. Ensure that the calibration board is mounted on the robot. For more instructions, please refer to Calibration Board Mounting Method.

Build Project to Obtain Calibration Board Poses

1. Add the “Capture Images from Camera” Step.

   Create a new project and add the “Capture Images from Camera” Step, or select the Step in an opened project. Click the **[Select camera]** button in the Step Parameters panel to select and connect the camera. Then select the Camera Calibration Parameters to be checked.

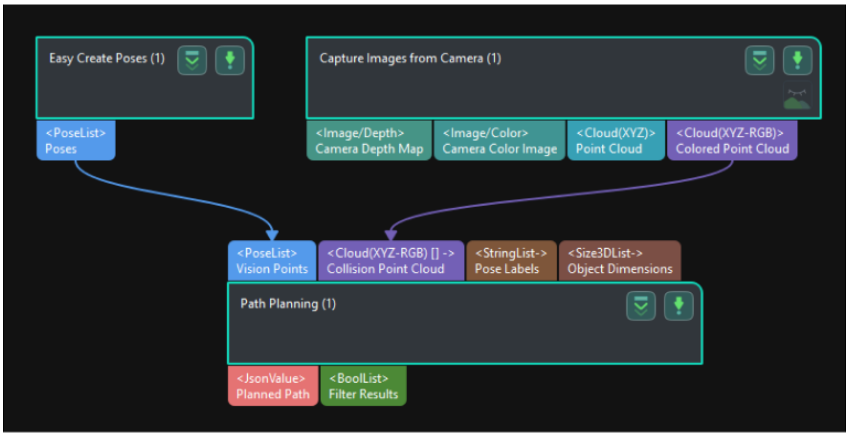
2. Add the “Path Planning” Step.

   Add the “Path Planning” Step and connect the Colored Point Cloud output port of the “Capture Images from Camera” Step with the Collision Point Cloud input port of this Step.

3. Add the “Easy Create Poses” Step.

   Add the “Easy Create Poses” Step and connect the Poses output port with the Vision Points input port of the “Path Planning” Step.

The built project is shown as below.


Easy Create Poses (1)
<PoseList> Poses
Capture Images from Camera (1)
<Image/Depth> Camera Depth Map
<Image/Color> Camera Color Image
<Cloud(XYZ)> Point Cloud
<Cloud(XYZ-RGB)> Colored Point Cloud
<PoseList> Vision Points
<Cloud(XYZ-RGB) [] -> Collision Point Cloud
<StringList-> Pose Labels
<Size3DList-> Object Dimensions
Path Planning (1)
<JsonValue> Planned Path
<BoolList> Filter Results

1. Enter the simulation interface.

   Click **[Open the editor]** in Step Parameters panel of the “Path Planning” Step to enter the 3D simulation area.

Synchronize the Pose of the Real Robot and Obtain the Scene Point Cloud Containing the Calibration Board Point Cloud

1. Click **[Open the editor]** in Step Parameters panel of the “Path Planning” Step to enter the 3D simulation area.
2. Synchronize the pose of the real robot with that of the simulated robot in the “Path Planning” Step, so that the poses of the simulated robot in the 3D simulation area and the real robot will be the same.
3. Click the **[Simulate]** button to visualize the scene point cloud containing the calibration board. (Please ignore the error message after clicking **[Simulate]**.)

Create Virtual TCP on the Calibration Board

1. Add an end tool to the resources in the path planning tool.
2. Check the visualization of the end tool in the 3D simulation area, and adjust the parameters to make the TCP coincide with a circle center on the calibration board (the pose axis should coincide with the cross in the circle and the XOY plane should be closely aligned with the calibration plane).
3. Enter a tool name and click OK to finish creating the virtual TCP.

Check Extrinsic Parameters at Other Positions in the Workspace

1. Move the real robot using the teach pendant to other positions in the workspace.
2. Synchronize the pose of the real robot with that of the simulated robot in the “Path Planning” Step, so that the poses of the simulated robot in the 3D simulation area and the real robot will be the same.
3. Click the “Simulate” button to obtain the new scene point cloud containing the calibration board.
4. Check if the virtual TCP added previously coincides with the circle center on the calibration board.

Other Checking Suggestions

If there is a significant error in the extrinsic parameters, please refer to the following suggestions:

- The accuracy of the extrinsic parameters is mainly influenced by the absolute accuracy of the robot. Please check the absolute accuracy of the robot.
- You can try creating different point cloud models at different positions or jogging the robot to record different pick points to compensate for the robot’s absolute accuracy.

4.7.2. Matching Model and Pick Point Editor

Matching Model and Pick Point Editor is often used in projects that require recognition of workpieces, such as machine tending. You can use Matching Model And Pick Point Editor to create point cloud models of the target objects, edit these models, and add pick points.

To use the Matching Model and Pick Point Editor, you will need to use the data and configurations in the project. Therefore, please open the corresponding project or create a new project before using this tool. In the opened project, click the Matching Model and Pick Point Editor button on the toolbar to open the tool.

There are two main ways for you to add pick points in Matching Model and Pick Point Editor. Before you configure the point cloud model and pick point, it is recommended to read Interface Introduction to learn about the user interface and functions first.

Adjust the robot pose manually through the teach pendant and obtain the pick point

- Suitable scenario: the robot picking pose, TCP, and object pose in the camera reference frame are almost unchanged; the TCP is difficult to measure.
- Procedure: Obtain the pick point → Generate a point cloud model → Edit the point cloud model
  - Add Pick Point by Teaching
  - Generate Point Cloud Model
  - Edit Point Cloud Model

Directly add and adjust a pick point in Matching Model and Pick Point Editor

- Suitable scenario: requirement on picking accuracy not very high.
- Procedure: Generate a point cloud model → Edit the point cloud model → Add the pick point
  - Generate Point Cloud Model
  - Edit Point Cloud Model
  - Add Pick Point by Adding a Pose

Besides the above two ways, you can import an available pick point file. Click the icon on the toolbar, and select the pick point file in the pop-up file selection window to add a pick point quickly.

> The Steps which may require files generated from Matching Model and Pick Point Editor are as follows.
>
> - 3D Coarse Matching, 3D Coarse Matching (Multiple Models), 3D Fine Matching, and 3D Fine Matching (Multiple Models) in which the point cloud model file is required

- Map to Multiple Pick Points in which the pick point file is required

4.7.3. Deep Learning Server

This section introduces how to use the deep learning server.

Introduction to the Tool

Deep learning server is a tool provided by Mech-Vision to manage deep learning services. You can click Deep Learning › Deep Learning Server in the menu bar of the main interface to open the deep learning server.

Usually, you do not need to do anything in the deep learning server window. Once a Mech-Vision project is opened, the deep learning server will be started automatically according to the **Server IP** and **Server Port (1–65535)** parameters set in the deep learning Steps.

Compatibility

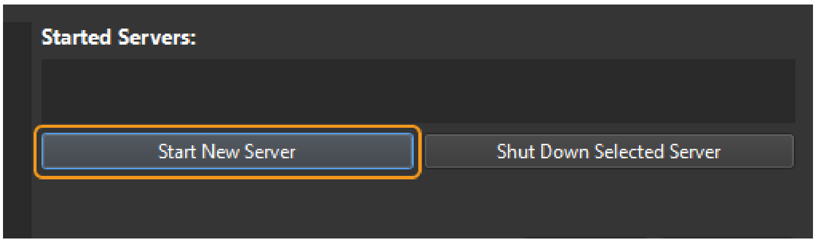
Please refer to Compatibilities of Deep Learning Steps to check whether you should start the deep learning server when a deep learning Step of a specific version is used.

Basic Procedures

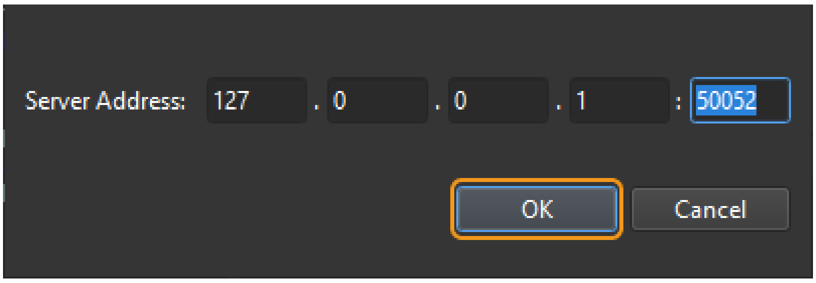
Please read on to learn about basic procedures for using the deep learning server.

Start a New Deep Learning Server

1. Click **[Start New Server]** in the Deep Learning Server window.

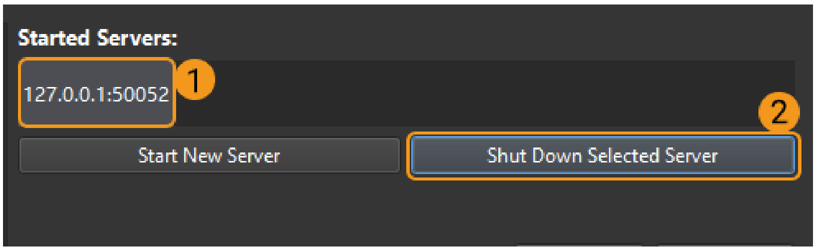


2. Enter the server address and port number in the pop-up window, and click **[OK]** to start a new deep learning server.

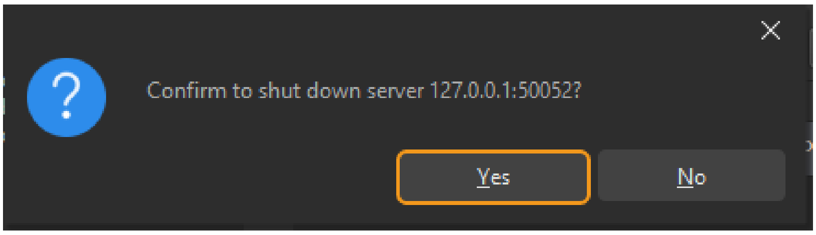

Server Address: 127 . 0 . 0 . 1 : 50052

Close the Selected Deep Learning Server

1. When multiple deep learning servers have been started, select the deep learning server you want to shut down in the Started Servers list and click **[Shut Down Selected Server]**.



2. Select [Yes] in the pop-up window to shut down the selected deep learning server.



Confirm to shut down server 127.0.0.1:50052?

Restart the Deep Learning Server

To restart the deep learning server, shut down the server first and then start a new server with the same server address and port number.

4.7.4. Deep Learning Model Package Management Tool

This section offers a guide on using the deep learning model package management tool and some notes worth your attention.

Introduction

Deep learning model package management tool is designed to manage all deep learning model packages in Mech-Vision. It can be used to optimize single model packages or cascaded model packages exported by Mech-DLK 2.2.0 or above and manage and monitor the operation mode, hardware type, model efficiency, and model package status. Besides, this tool can be used to monitor the GPU memory usage of the IPC.

If deep learning Steps are used in the project, you can import the model package to the deep learning model package management tool first and then use the models in the relative Steps. Importing the model package to the tool facilitates optimizing the model package in advance.

> From Mech-DLK 2.4.1, model packages can be divided into single model packages and cascaded model packages.
>
> - Single model package: There is only one deep learning model in the model package, such as an “Instance Segmentation” model.
> - Cascaded model package: Multiple models are cascaded in the model package, and the output result of the previous model is input to the next model. For example, there are two models, “Object Detection” and “Instance Segmentation” in the model package and the inference sequence is menu:Object Detection [Instance Segmentation]. The output of “Object Detection” is input to the “Instance Segmentation” model.

Interface Introduction

You can open the tool in either of the following two ways:

- After creating or opening a project, select Deep Learning › Deep Learning Model Package

Management Tool in the menu bar.

- In an opened project, select the "Deep Learning Model Package Inference" Step, and click the **[Model Package Management Tool]** button in the Step Parameters panel.

The options in the deep learning model package management tool are shown in the table below.

Option	Description
Available model package	The names of imported model packages
Project name	The Mech-Vision project in which the corresponding model package is used
Model package type	The type of the model package, such as "Object Detection" (single model package) and "Object Detection + Defect Segmentation" (cascaded model package)
Operation mode	The operation mode of the model package, including "Sharing mode" and "Performance mode"
Hardware type	The hardware type used for model package inference. If you are using a GPU model, you can modify the hardware type, i.e., GPU (default), GPU (optimization), and CPU
Model efficiency	The inference efficiency of the model package
Model package status	The status of the model package, such as "Loading and optimizing", "Loading completed", and "Optimization failed"

Operation mode

- Sharing mode: Once this option is selected, when multiple Steps use the same model package, the inferences will be done one by one in sequence and less memory will be used.
- Performance mode: Once this option is selected, when multiple Steps use the same model package, the inferences will be done simultaneously, and the inference speed will be relatively fast. However, more memory will be used in this mode.

Hardware type

- CPU: Use CPU for model package inference. Compared to GPU, the inference time is longer and the recognition accuracy is lower.
- GPU (default): Once this option is selected, the model package will not be optimized according to the hardware type and the deep learning inference will not be speeded up.
- GPU (optimization): Once this option is selected, the model package will be optimized according to the hardware type and the one-time optimization process takes about 5 to 15 minutes. When an optimized model package is used, the inference time will be reduced.

Usage

Follow the steps below to learn about common procedures for using the deep learning model package management tool.

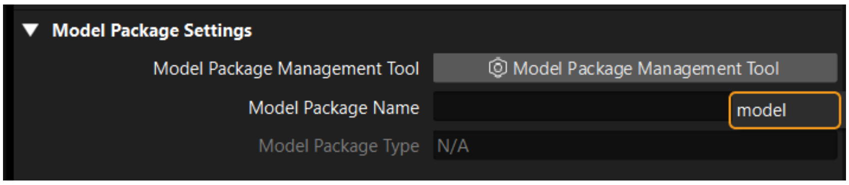
Import the Deep Learning Model Package

1. Open the deep learning model package management tool, and click **[Import]** in the upper left corner.
2. Select the model package you want to import in the pop-up Select File window and click **[Open]**. Then the deep learning model package will appear in the list in the window, which suggests that the model package is imported successfully.

> To import a model package successfully, the minimum version requirement for the graphics driver is 472.50, and the minimum requirement for the CPU is a 6th-generation Intel Core processor. It is not recommended to use a graphics driver above version 500, which may cause fluctuations in the execution time of deep learning Steps. If the hardware cannot meet the requirement, the deep learning model package cannot be imported successfully.

Select the Deep Learning Model Package in the Step

After importing the model package into the tool, if you want to use the model package in the Deep Learning Model Package Inference Step, you can select the model package in the drop-down list of the **Model Package Name** parameter in the Step Parameters panel.

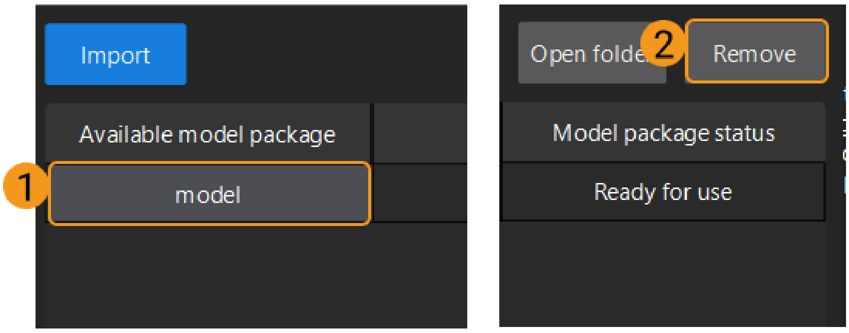


▼ Model Package Settings

Model Package Management Tool	Model Package Management Tool
Model Package Name	model
Model Package Type	N/A

Remove the Imported Deep Learning Model Package

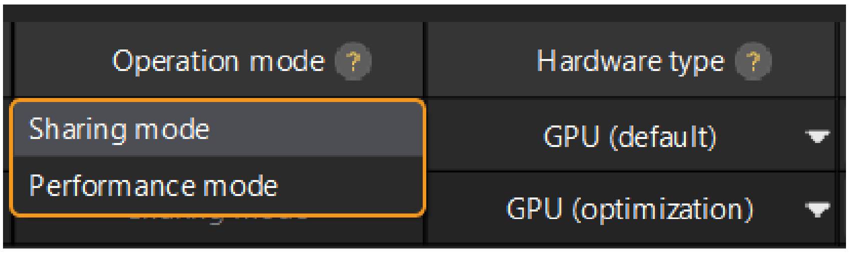
If you want to remove an imported deep learning model package, select the model package first, and click the **[Remove]** button in the upper right corner.



> When the deep learning model package is **Loading and optimizing** or the project using the deep learning model package is **running**, the model package cannot be removed.

Switch the Operation Mode

If you want to switch the **Operation mode** for deep learning model package inference, you can click ▾ in the Operation mode column in the deep learning model package management tool, and select **Sharing mode** or **Performance mode**.

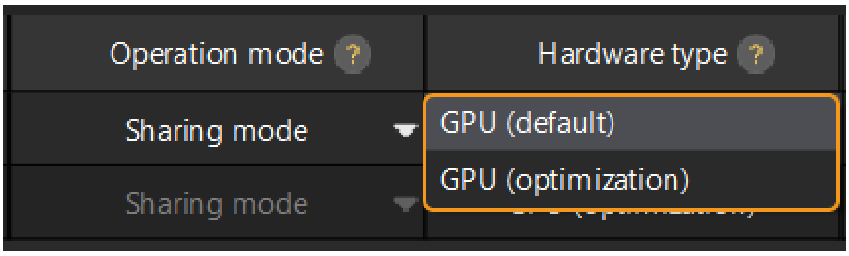


- When the deep learning model package is **Loading and optimizing** or **In use**(i.e., the project using the model package is running), the **Operation mode** cannot be changed.
- When the operation mode of the model package is **Sharing mode**, the **GPU ID** in the Deep Learning Model Package Inference Step cannot be changed.

Switch the Hardware Type

You can change the hardware type for deep learning model package inference to GPU (default), GPU (optimization), or CPU.

Click ▼ in the **Hardware type** column in the deep learning model package management tool, and select **GPU (default)**, **GPU (optimization)**, or **CPU**.



- Only model packages exported by Mech-DLK 2.4.1 or above support switching between CPU and GPU.
- You cannot use the **GPU (default)** for either of the following types of model packages:
  - Instance segmentation model package and instance segmentation super model package exported by Mech-DLK 2.2.0.
  - Model package that only contains the DLKMT file exported by Mech-DLK.
- When the deep learning model package is **Loading and optimizing** or **In use**(i.e., the project using the model package is running), the **Hardware type** cannot be changed.

Configure the Model Efficiency

When a model package exported by Mech-DLK 2.4.1 or above is used, the model efficiency can be configured. The procedures for configuring the model efficiency are as follows.

1. Import the deep learning model package.
2. Click the **[Configure]** button in the Model efficiency column, a Model Efficiency configuration window will pop up, and you can configure the “Batch size” and “Precision”.

   The “Batch size” and “Precision” parameters can affect the model efficiency configuration.

   - Batch size: the number of images that will be passed through the neural network at once during inference, ranging from 1 to 128. Increasing the value will increase the model’s inference speed, but more video memory will be used. If the value is not set properly, the inference speed will be slowed down. The instance segmentation models do not support configuring “Batch size”, and the default “Batch size” must be set to 1.

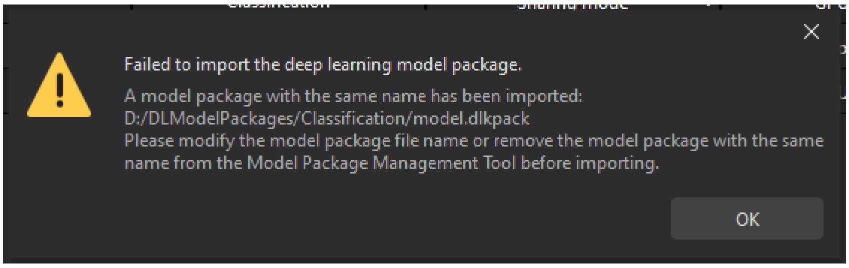
- Precision (Only available when the "Hardware type" is "GPU optimization"): FP32: high-precision model with lower inference speed. FP16: low-precision model with higher inference speed.

> It is recommended to set the "Batch size" the same as the actual number of images that are passed through the neural network.
>
> If the set "Batch size" is much larger than the actual number of images that are passed through the neural network, part of the resources will be wasted, resulting in slower inference.
>
> For example, if the number of images is 26, and the "Batch size" is set to 20, two separate inferences will be performed. In the first inference, 20 images will be fed to the neural network, while in the second inference, 6 images will be fed to the neural network. For the second inference, the set "Batch size" is much larger than the actual number of images that are passed through the neural network, part of the resources will be wasted, resulting in slower inference. Therefore, please set the "Batch size" to a proper value to ensure the efficient use of the resources.

Notes

Fail to Import a Deep Learning Model Package

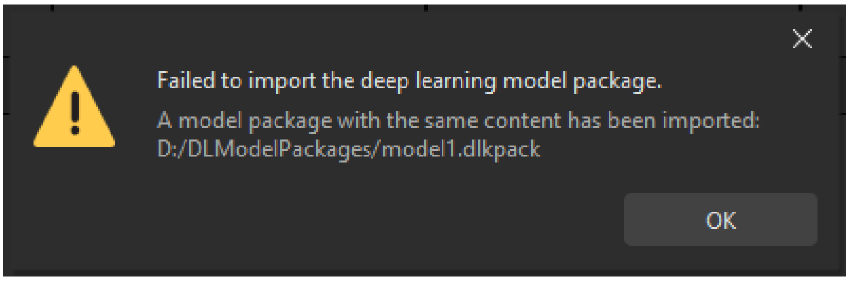
- When there is already an imported deep learning model package, and you want to import another one with the same name, please **modify the name of the new model package** or **remove the model package with the same name**, and then start to import.



Failed to import the deep learning model package.

A model package with the same name has been imported:
D:/DLModelPackages/Classification/model.dlkpack
Please modify the model package file name or remove the model package with the same name from the Model Package Management Tool before importing.

- When you have imported a model package, and start to import another one with a different name but the same content, an error message **"Failed to import the deep learning model package."** will pop up.



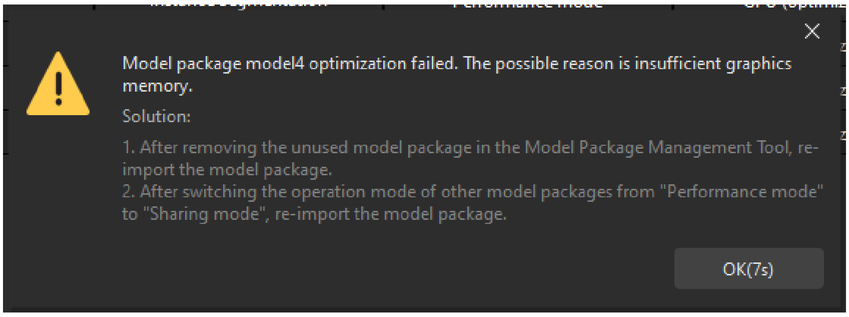
Failed to import the deep learning model package.

A model package with the same content has been imported:
D:/DLModelPackages/model1.dlkpack

- When the software and hardware cannot meet the minimum requirement, the deep learning model package cannot be imported successfully. The minimum required version for the graphics driver is 472.50, and the minimum required CPU version is 6th generation Intel Core.

Fail to Optimize a Deep Learning Model Package

When there is not enough memory, the deep learning model package may not be able to be optimized successfully. A pop-up window as shown below will appear. You can fix the problem according to the solutions in the window.



Model package model4 optimization failed. The possible reason is insufficient graphics memory.

Solution:

1. After removing the unused model package in the Model Package Management Tool, re-import the model package.
2. After switching the operation mode of other model packages from "Performance mode" to "Sharing mode", re-import the model package.

Notes for Compatibilities

- Deep learning model packages exported by Mech-DLK 2.4.1 can be used in Mech-Vision 1.7.1. However, there may be some compatibility issues. It is recommended to use deep learning model packages exported by Mech-DLK 2.4.1 or above with Mech-Vision 1.7.2 or above.
  - Cascaded model packages cannot be used in Mech-Vision.
  - The model efficiency cannot be configured.
  - The performance of image classification may be diminished.
  - Model packages cannot be used on CPU devices.
- If model packages optimized with Mech-Vision 1.7.1 are used in Mech-Vision 1.7.2, the execution time may be longer when the optimized model package is used in the "Deep Learning Model Package Inference" Step for the first time.
- Please pay attention to the following issue when the **Hardware type** for model package inference is set to **GPU (optimization)**.

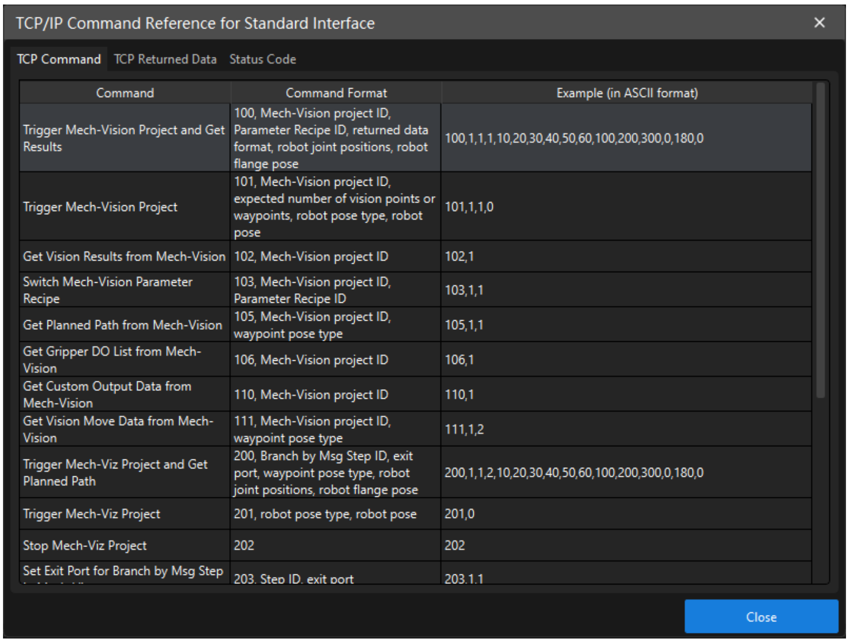
  If the model package is not optimized in Mech-Vision 1.7.X, and it is optimized in Mech-Vision 1.8.0 or later versions, the model package cannot function properly in Mech-Vision 1.7.X. You should click **[Open folder]** in the Deep Learning Model Package Management Tool, delete the cache folder corresponding to the model package, and optimize the model package again.

> You can check the cache folder corresponding to the model package in the model_config.json file.

4.7.5. TCP/IP Command Reference for Standard Interface

Mech-Vision provides a built-in TCP/IP command reference for Standard Interface, making it easy for engineers to quickly check Standard Interface TCP/IP commands. Click Robot and Communication › TCP/IP Command Reference for Standard Interface in the menu bar to view the table.

For instructions on using the tool, please refer to How to use "TCP/IP Command Reference for Standard Interface".



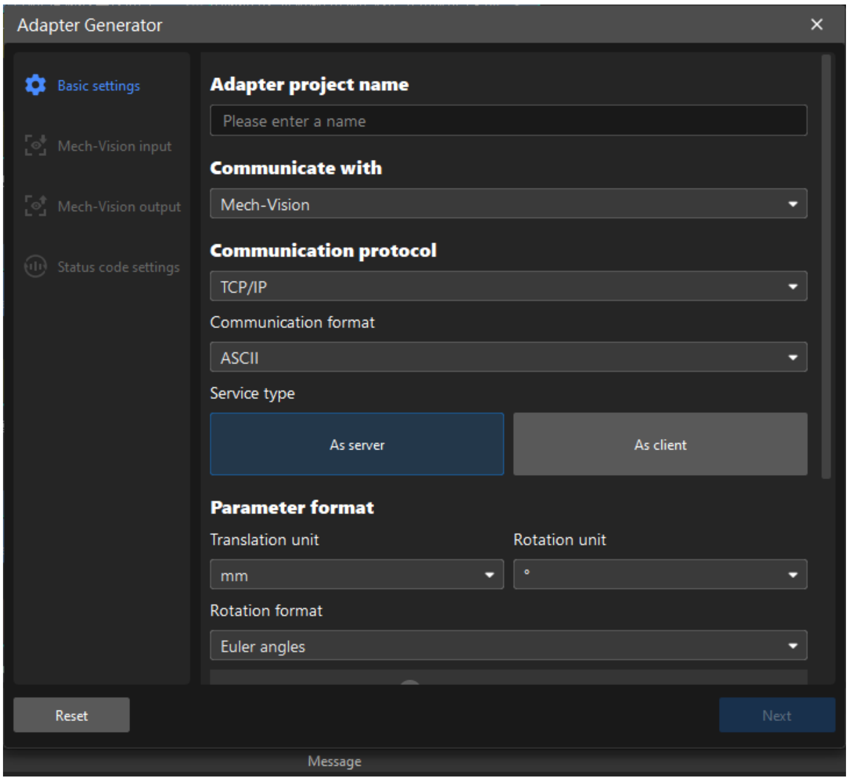
TCP/IP Command Reference for Standard Interface

Command	Command Format	Example (in ASCII format)
Trigger Mech-Vision Project and Get Results	100, Mech-Vision project ID, Parameter Recipe ID, returned data format, robot joint positions, robot flange pose	100,1,1,1,10,20,30,40,50,60,100,200,300,0,180,0
Trigger Mech-Vision Project	101, Mech-Vision project ID, expected number of vision points or waypoints, robot pose type, robot pose	101,1,1,0
Get Vision Results from Mech-Vision	102, Mech-Vision project ID	102,1
Switch Mech-Vision Parameter Recipe	103, Mech-Vision project ID, Parameter Recipe ID	103,1,1
Get Planned Path from Mech-Vision	105, Mech-Vision project ID, waypoint pose type	105,1,1
Get Gripper DO List from Mech-Vision	106, Mech-Vision project ID	106,1
Get Custom Output Data from Mech-Vision	110, Mech-Vision project ID	110,1
Get Vision Move Data from Mech-Vision	111, Mech-Vision project ID, waypoint pose type	111,1,2
Trigger Mech-Viz Project and Get Planned Path	200, Branch by Msg Step ID, exit port, waypoint pose type, robot joint positions, robot flange pose	200,1,1,2,10,20,30,40,50,60,100,200,300,0,180,0
Trigger Mech-Viz Project	201, robot pose type, robot pose	201,0
Stop Mech-Viz Project	202	202
Set Exit Port for Branch by Msg Step	203, Step ID, exit port	203,1,1

4.7.6. Adapter Generator

Adapter Generator is a tool for conveniently generating an Adapter program. User can generate an Adapter program simply by configuring parameters on a graphical interface. User can also customize the program files to implement more advanced features that go beyond the capabilities of the Standard Interface communication. Click Robot and Communication › Adapter Generator in the menu bar to open the tool.

For instructions on using the Adapter Generator, please refer to Adapter Generator Guide.



Adapter Generator

Adapter project name

Please enter a name

Communicate with

Mech-Vision

Communication protocol

TCP/IP

Communication format: ASCII

Service type: As server | As client

Parameter format

Translation unit: mm

Rotation unit: °

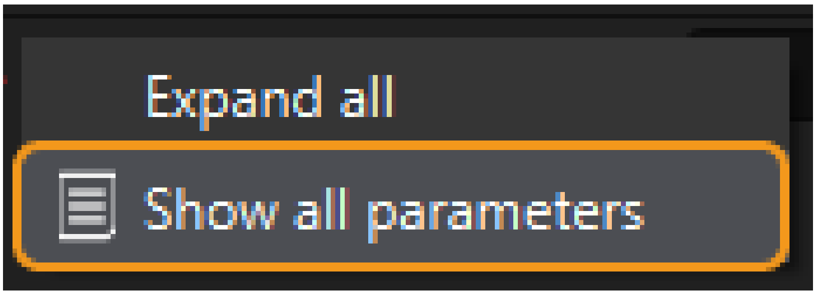
Rotation format: Euler angles

4.7.7. Set Static Background for Project

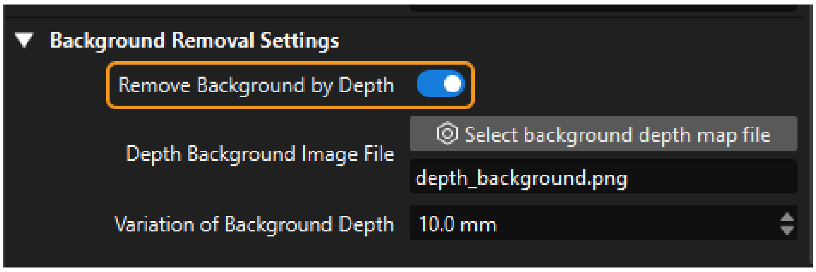
Setting a static background and obtaining a depth map of it can avoid the interference of the background in subsequent Steps and facilitate the calculation of the object height.

Instructions for Setting Static Background

1. Open Mech-Vision project and select the Step Capture Images from Camera. Then right-click on the empty space in the Step Parameters section and select **[Show all parameters]**, as shown below.

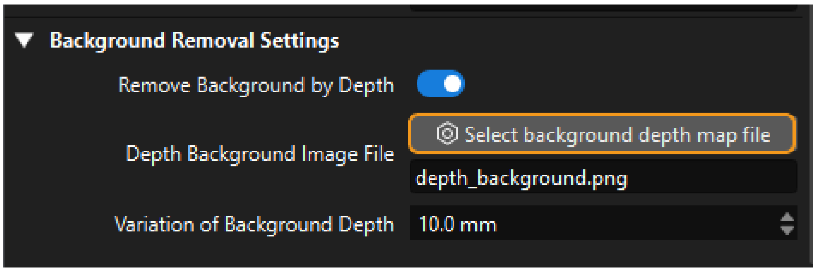


2. Select **[Remove Background by Depth]** in Background Removal Settings.



Background Removal Settings
Remove Background by Depth
Depth Background Image File: depth_background.png
Variation of Background Depth: 10.0 mm

3. Click **[Select background depth map file]** to open the background setting tool.



Background Removal Settings
Remove Background by Depth
Depth Background Image File: depth_background.png
Variation of Background Depth: 10.0 mm

4. The Set Background tool includes four sections: Camera Connection, Image Viewer, Log Messages, and Image Saving, as shown below.

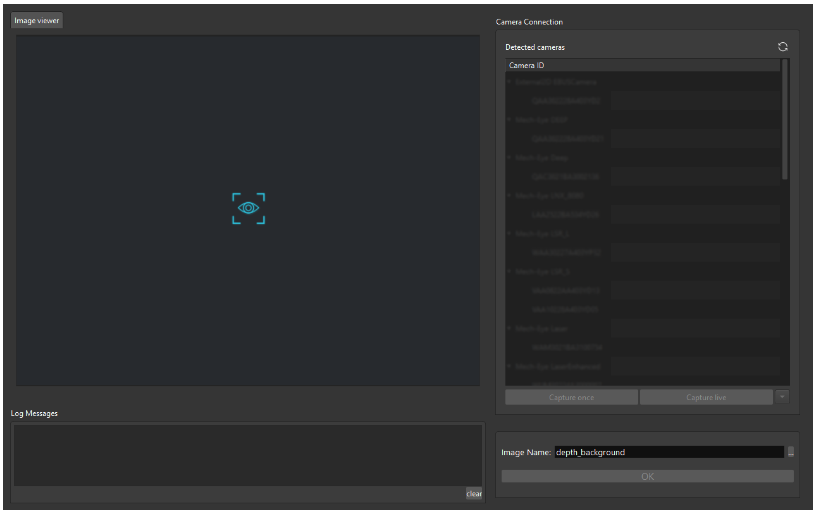
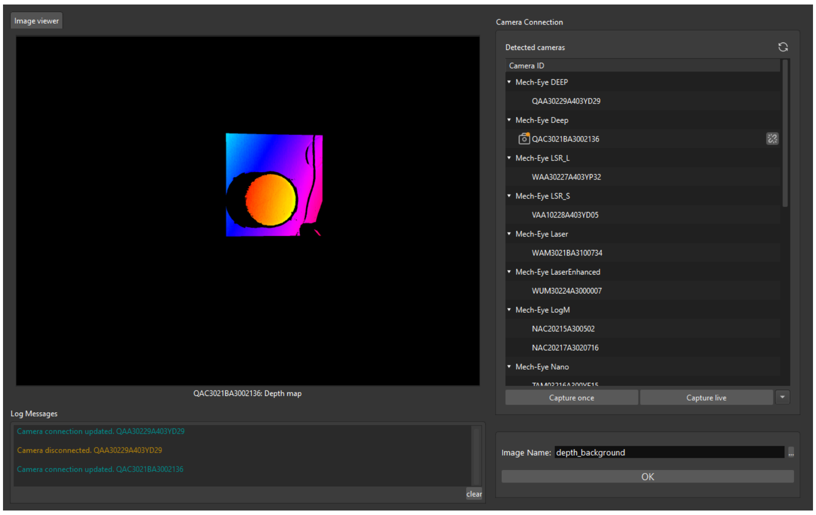


Image Name: depth_background

5. You will need to connect a camera before setting the background. Double-click the camera number or click to connect the camera. After the camera is connected successfully, click [Capture once] or [Capture live] to capture an image. Click [OK] to save the image you need.



QAC3021BA3002136: Depth map

- Mech-Eye DEEP
  - QAA30229A403YD29
- Mech-Eye Deep
  - QAC3021BA3002136
- Mech-Eye LSR_L
  - WAA30227A403YP32
- Mech-Eye LSR_S
  - VAA10228A403YD05
- Mech-Eye Laser
  - WAM3021BA3100734
- Mech-Eye LaserEnhanced
  - WUM30224A3000007
- Mech-Eye LogM
  - NAC20215A300502
  - NAC20217A3020716
- Mech-Eye Nano

```
Camera connection updated. QAA30229A403YD29
Camera disconnected. QAA30229A403YD29
Camera connection updated. QAC3021BA3002136
```

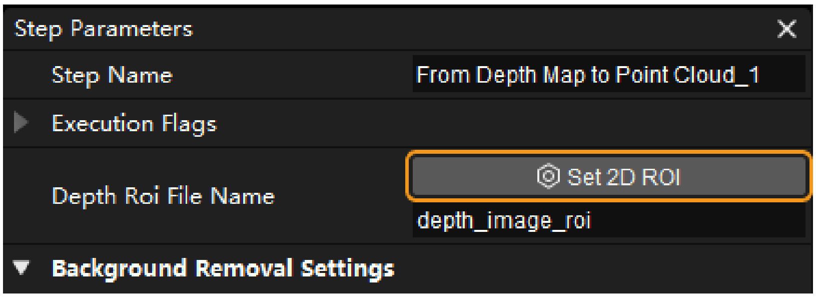
Image Name: depth_background

4.7.8. Set ROI

By setting a 2D or 3D ROI, the unwanted pixels in the image or points in the point cloud can be filtered in succeeding Step, and thus the processing speed of the entire Mech-Vision project can be increased.

Instructions for Setting 2D ROI

1. Select the Step in which you want to set a 2D ROI, click [Set 2D ROI] button in the Step Parameters panel, and then a Set ROI window will pop up.


Step Parameters

Step Name	From Depth Map to Point Cloud_1
Execution Flags	
Depth Roi File Name	Set 2D ROI depth_image_roi
Background Removal Settings	

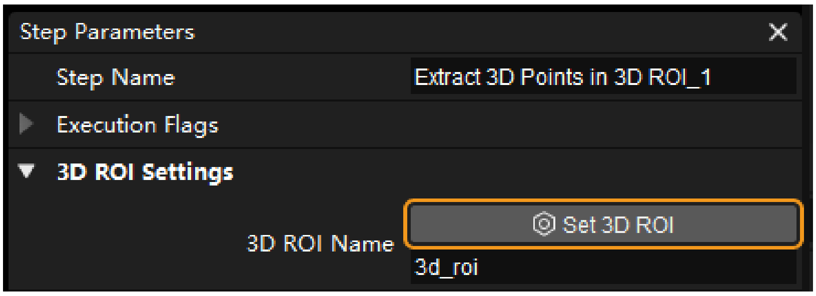
2. Hold down the left mouse button and drag to select an ROI on the displayed 2D image or depth map, and then click the left mouse button again to confirm. The select frame will be displayed in red if the ROI is set successfully. If you need to re-select the ROI, please click the left mouse button and drag again.
3. The coordinates of the selected ROI will be displayed in the ROI Properties section. Click [OK] to save and exit.

Steps related to 2D ROI include: Scale Image in 2D ROI, Segment Depth Map, From Depth Map to Point Cloud.

Instructions for Setting 3D ROI

Before setting a 3D ROI, please go to **Project Assistant** and select a scene point cloud for reference, or else an alert window will pop up and the setting cannot continue.

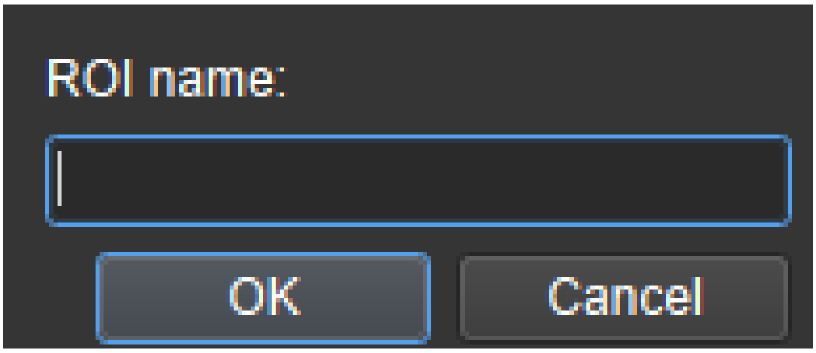
1. Select the Step in which you want to set a 3D ROI, click [Set 3D ROI] button in the Step Parameters panel, and then a Set ROI window will pop up.


Step Parameters

Step Name	Extract 3D Points in 3D ROI_1
Execution Flags	
3D ROI Settings	
3D ROI Name	Set 3D ROI 3d_roi

2. The Set 3D ROI window consists of 4 parts: Point Cloud Display, 3D ROI Parameters, Display, and Tips.
   - Point Cloud Display area includes robot base coordinate system, camera reference frame, 3D ROI, and target cloud.
   - 3D ROI Parameters includes the name, size, center, and orientation of the 3D ROI.
   - Display Settings area is used for adjusting the display status of ground and robot base coordinate system.
   - Tips provides tips for 3D ROI setting.
3. Hold down the `Ctrl` key and drag the faces of the selection tool to adjust the position of the 3D ROI. Hold down the `Ctrl` key and drag the handles of the selection tool to adjust the dimensions of the 3D ROI. You can also configure the size, center, and orientation to adjust the 3D ROI in the 3D ROI Parameters section.
4. In addition to editing the default 3D ROI directly, you can create a new 3D ROI. Click + next to the ROI Name, and an Add ROI pop-up window will appear, as shown below. Enter the name of the new 3D ROI, and click [OK] to finish. Then you can adjust the 3D ROI in the Point Cloud

Display area. To switch between different 3D ROIs, please click on the down arrow icon next to the ROI Name and select the 3D ROI you want to edit in the drop-down list.



Steps related to 3D ROI include: Extract 3D Points in 3D ROI, Collect Poses in 3D ROI, Invalidate Depth Pixels outside 3D ROI, Validate Existence of Poses in 3D ROI, Read 3D ROI Center, Save Regions around Poses as 3D ROIs, Extract Empty Regions in Depth Map within 3D ROI.

Please pay attention to the following issues when setting the 2D ROI or 3D ROI:

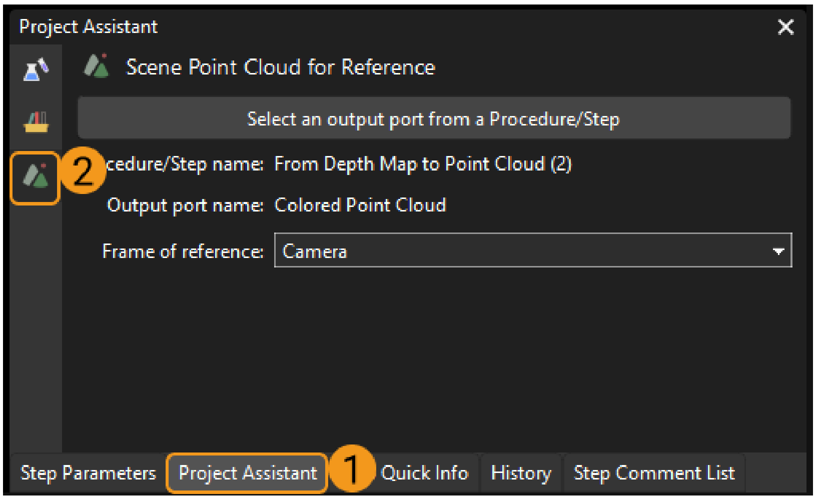
- If different 3D ROIs are created, please make sure that the one used in the Step corresponds to the appropriate 3D ROI.
- The set ROI should cover the bin where workpieces are placed, which can reduce the picking deviation due to the worn bin or shaking during picking. In addition,the ROI should be slightly larger than the bin used on site.

4.7.9. Set the Scene Point Cloud

The scene point cloud consists of all point clouds in the camera’s field of view, and can be used as a reference for setting the ROI in relevant Steps. Therefore, a scene point cloud should be selected before setting the ROI.

This section describes how to use the tool to select a scene point cloud.

1. Open [**Project Assistant**] and click on icon to open the window for configuration.

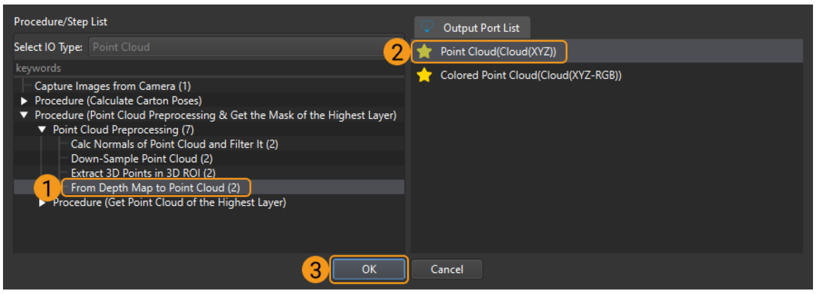


cedure/Step name: From Depth Map to Point Cloud (2)

Output port name: Colored Point Cloud

Frame of reference: Camera

2. Click[**Select an output port from a Procedure/Step**], and select the Step which outputs the scene point cloud on the left side, and select the output port on the right side. Then click[**OK**].


Procedure/Step List
keywords
Capture Images from Camera (1)
Procedure (Calculate Carton Poses)
Procedure (Point Cloud Preprocessing & Get the Mask of the Highest Layer)
Point Cloud Preprocessing (7)
Calc Normals of Point Cloud and Filter It (2)
Down-Sample Point Cloud (2)
Extract 3D Points in 3D ROI (2)
From Depth Map to Point Cloud (2)
Procedure (Get Point Cloud of the Highest Layer)
Output Port List
Point Cloud(Cloud(XYZ))
Colored Point Cloud(Cloud(XYZ-RGB))
1
2
3
OK
Cancel

Select IO Type: Point Cloud

It is recommended to select the point cloud output by From Depth Map to Point Cloud as the scene point cloud to ensure the processing speed in the subsequent Steps. Usually, the Step Capture Images from Camera is not recommended to be selected here.

- Icon ⭐ represents that the output port is connected with an input port of a succeeding Step.
- Icon ◡ represents that the output port is not connected with a succeeding Step.

3. After selecting the scene point cloud, you can also choose the Frame of reference according to actual requirements.

After setting up the scene point cloud, you can refer to Setting ROI to learn more details about setting up ROI.

4.7.10. Parameter Recipe

This section introduces the application and configuration of **parameter recipes**.

Introduction

Parameter recipes are sets of parameter settings that need to be adjusted according to different situations for the same project. With parameter recipes, you do not need to build multiple projects with the same logic and configure their parameters differently to meet different on-site requirements. Instead, you will only need to switch between parameter recipes in one project to make it applicable to various scenarios and therefore the productivity can be increased.

Mech-Vision provides a tool to configure **Parameter recipes**.

Application Scenarios

Parameter recipes are applicable to scenarios where the parameters need to be set according to different situations. The typical application scenarios are summarized as follows:

- Use the same Mech-Vision project to process different workpieces or objects. In this scenario, you can configure different parameter recipes and switch parameter settings dynamically for different workpieces or objects.
- Use the same Mech-Vision project to process workpieces in different bins. In this scenario, you can configure different parameter recipes and switch parameter settings dynamically for different bins.
- Under different light conditions (e.g., in the daytime or in the evening), the Mech-Vision project should use different camera configuration parameter groups and other parameter settings. In this scenario, you can configure different parameter recipes and switch parameter settings dynamically for different light conditions.
- In Eye In Hand mode, the Mech-Vision project should capture images in different areas. In this scenario, you can configure different parameter recipes and switch parameter settings

dynamically for different image capturing areas.

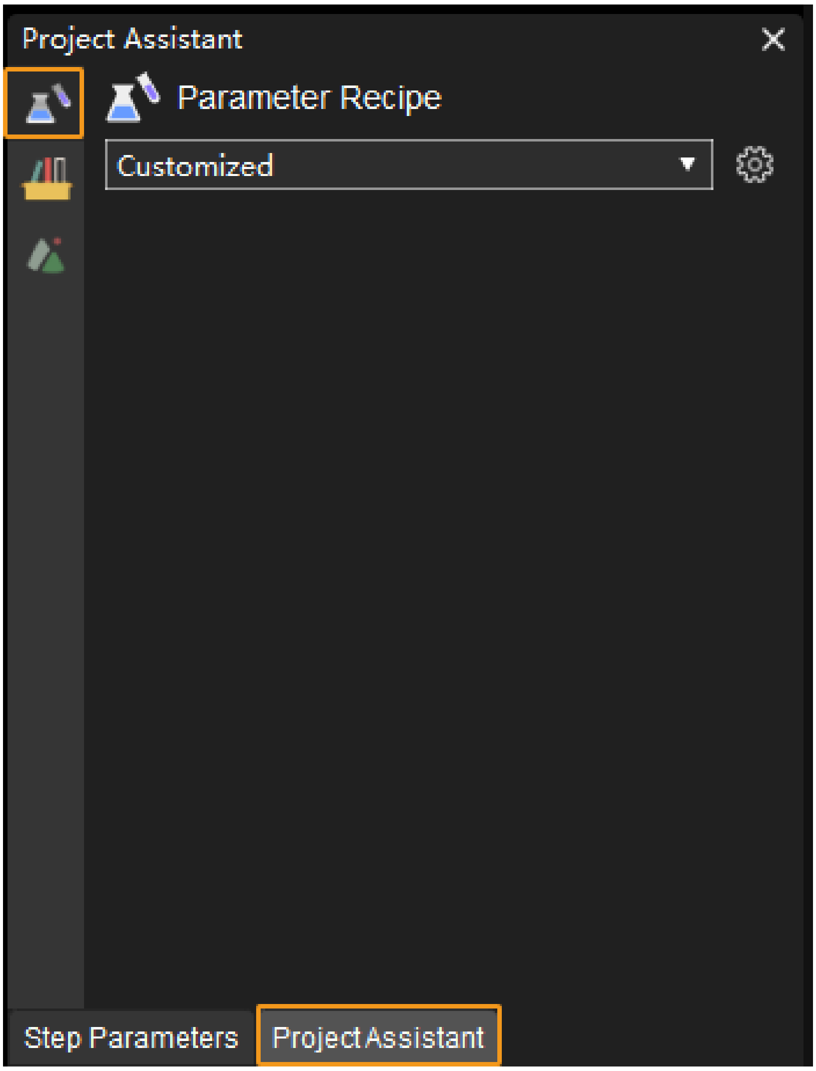
When master-control communication is used, you should switch the corresponding parameter recipe in a Mech-Vision project manually. When Standard Interfaces or Adapter is used for communication, you should run a recipe-switching command on the robot side, and Mech-Vision will switch the parameter recipe automatically according to the command it received.

Configure Parameter Recipes

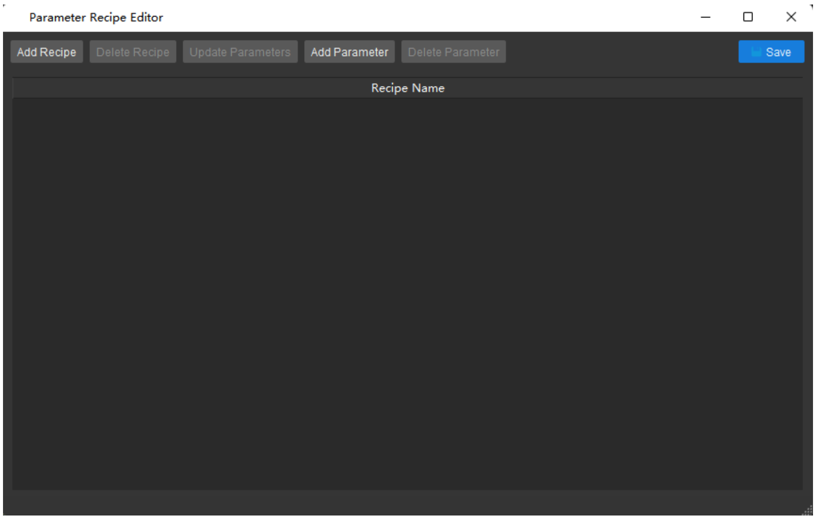
Add a Parameter Recipe

Follow these steps to add a parameter recipe:

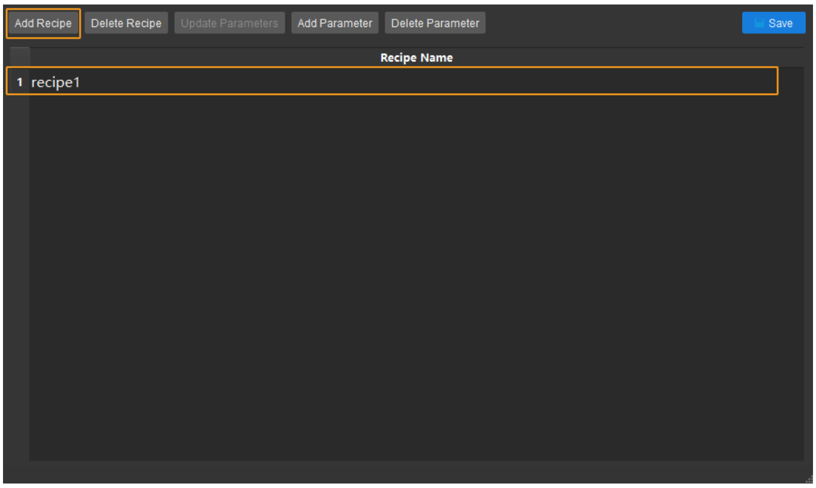
1. Open a project, click the **Project Assistant** tab, and click ⚗ to open the Parameter Recipe panel.



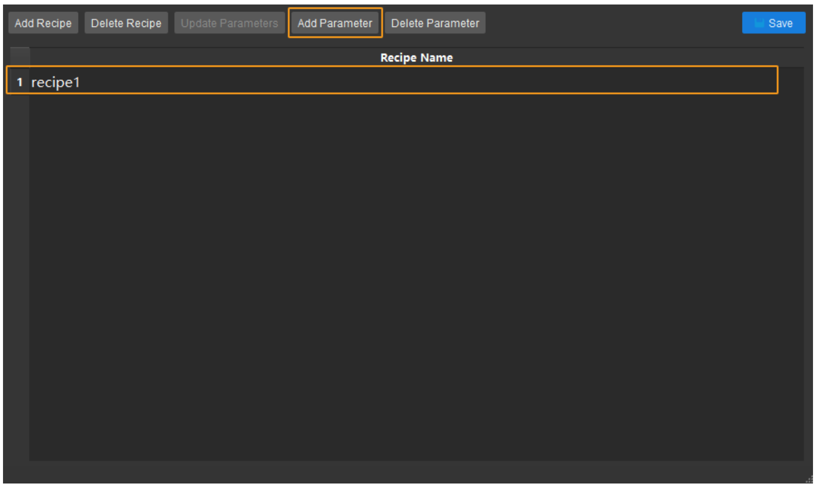
2. Click ⚙ in the Parameter Recipe panel.
3. Click the **[Add Recipe]** button in the pop-up **Parameter Recipe Editor** window.



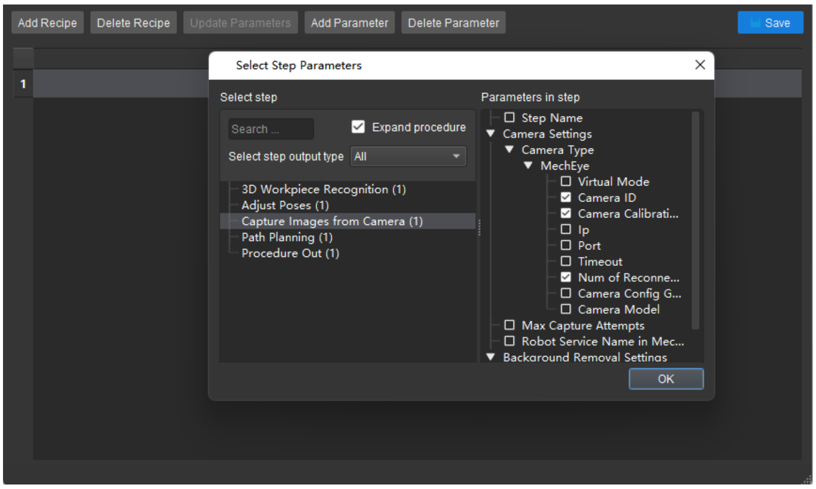
4. Double-click the newly added item in the parameter list and enter the recipe name.



5. Select the newly added recipe and then click the [Add Parameter] button, and a **Select Step Parameters** dialog box will pop up.



6. In the **Select Step Parameters** dialog box, select the Step you want to add in the left panel, and then select the parameters of the Step on the right.



Select Step Parameters

Select step

☑ Expand procedure

Select step output type: All

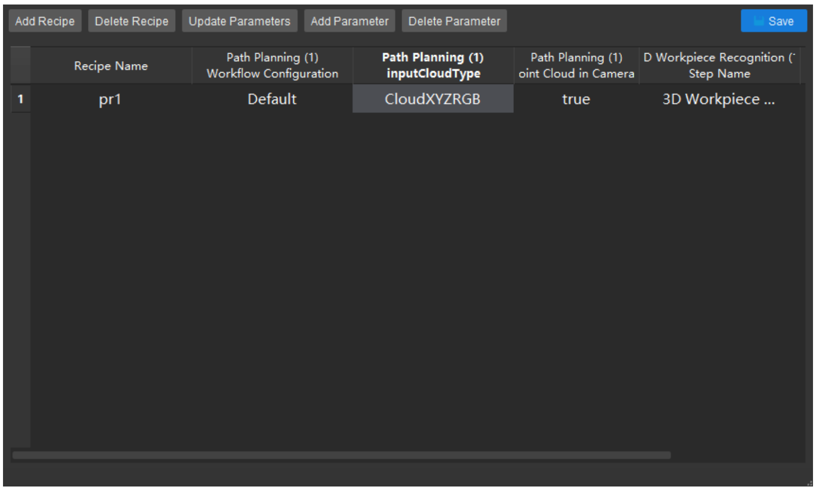
- 3D Workpiece Recognition (1)
- Adjust Poses (1)
- Capture Images from Camera (1)
- Path Planning (1)
- Procedure Out (1)

Parameters in step

- ☐ Step Name
- ▼ Camera Settings
  - ▼ Camera Type
    - ▼ MechEye
      - ☐ Virtual Mode
      - ☑ Camera ID
      - ☑ Camera Calibrati...
      - ☐ Ip
      - ☐ Port
      - ☐ Timeout
      - ☑ Num of Reconne...
      - ☐ Camera Config G...
      - ☐ Camera Model
  - ☐ Max Capture Attempts
  - ☐ Robot Service Name in Mec...
- ▼ Background Removal Settings

7. Add all the parameters of Steps you need, and then click **[OK]**.

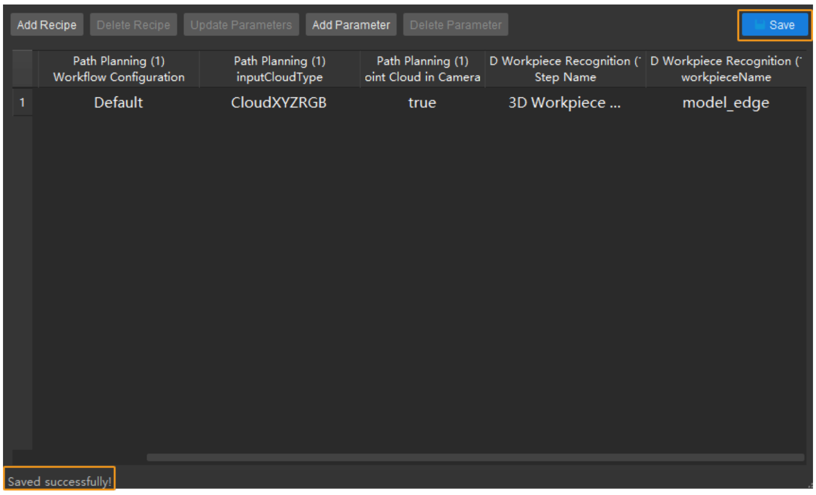
8. Double-click the parameters one by one to modify their settings in the parameter recipe list.



	Recipe Name	Path Planning (1) Workflow Configuration	Path Planning (1) inputCloudType	Path Planning (1) oint Cloud in Camera	D Workpiece Recognition (Step Name
1	pr1	Default	CloudXYZRGB	true	3D Workpiece ...

Please pay attention to the order to set parameters in the recipe. You may not be able to set a parameter successfully because of the setting order.

9. Click **[Save]** in the **Parameter Recipe Editor** dialog box. A Saved successfully message will appear in the lower left corner of the dialog box.



	Path Planning (1) Workflow Configuration	Path Planning (1) inputCloudType	Path Planning (1) oint Cloud in Camera	D Workpiece Recognition (Step Name	D Workpiece Recognition (workpieceName
1	Default	CloudXYZRGB	true	3D Workpiece ...	model_edge

Saved successfully!

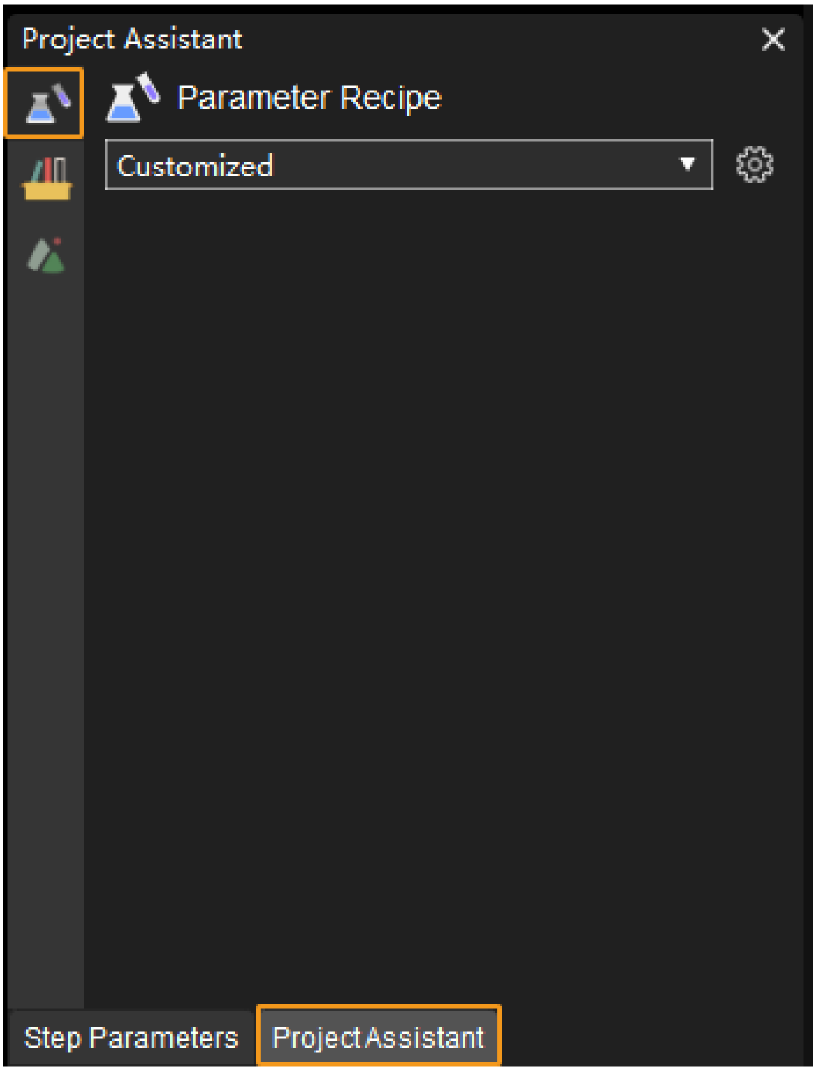
If you need multiple parameter recipes in the project, just follow the above instructions to add multiple recipes. The parameter settings of the newly added parameter recipe will be the same as those in the current project by default.

Apply or Switch the Parameter Recipe

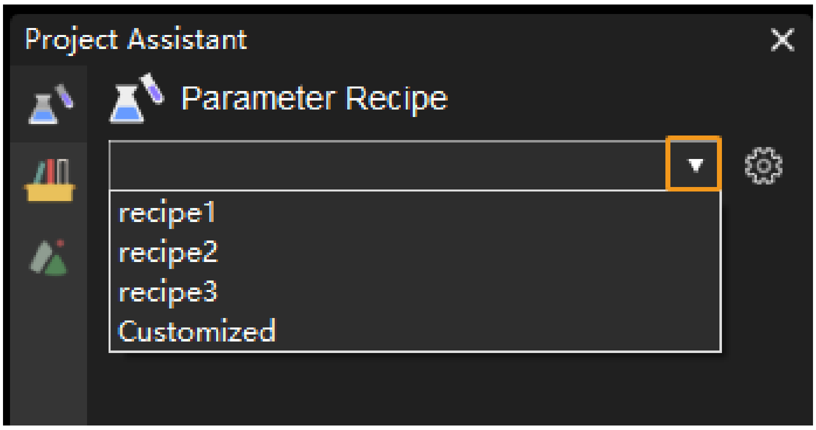
> When Standard Interfaces or Adapter is used for communication, you should run a command on the robot side to switch the parameter recipe successfully.

Follow these steps to apply or switch the parameter recipe manually:

1. Click the **Project Assistant** tab, and click the icon to open the Parameter Recipe panel.



2. In the Parameter Recipe panel, select the parameter recipe corresponding to the workpiece in the drop-down list. The selected parameter recipe will take effect when the project runs.



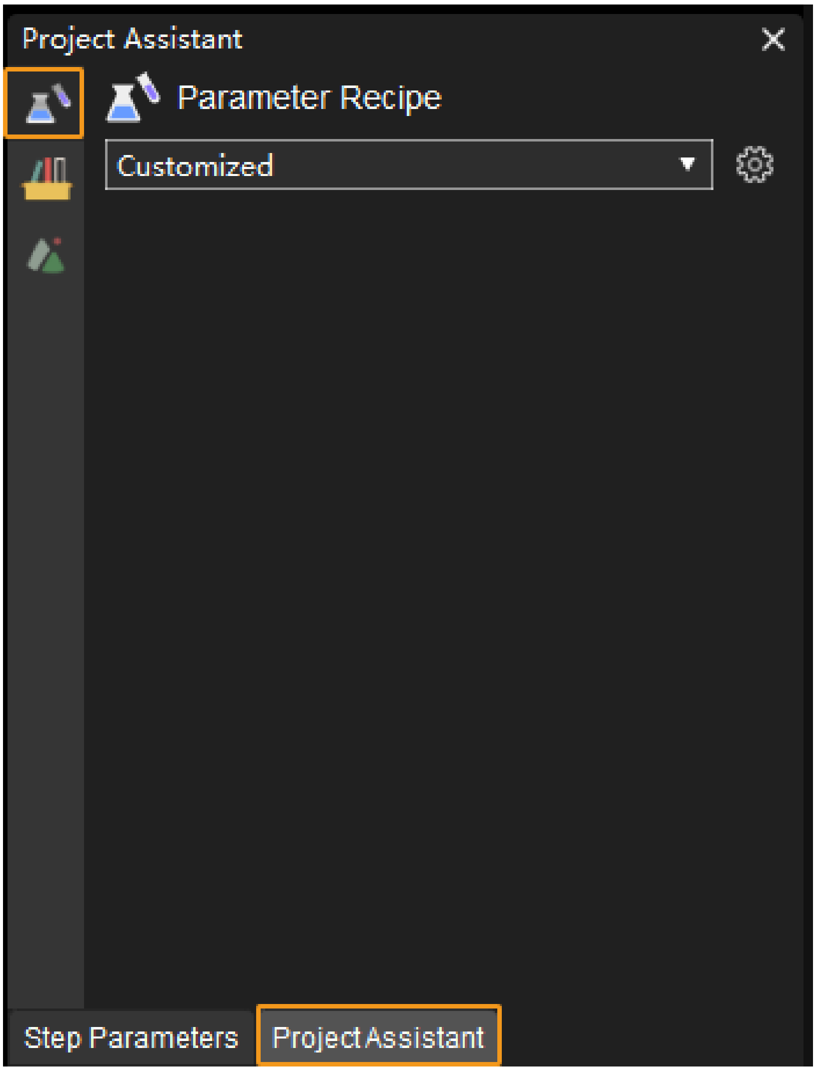
After running the project with the selected parameter recipe, if you want to modify the parameter settings in the recipe, please refer to Modify Specific Parameter Setting in the Parameter Recipe. If you modify the parameter settings in the project directly, the parameter recipe will be switched to Customized when you run the project again.

View the Parameter Recipe ID

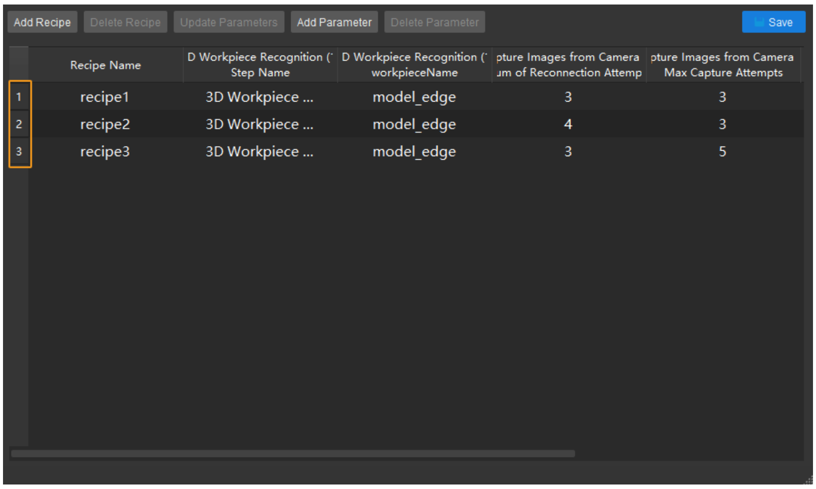
When Standard Interfaces or Adapter is used for communication, you should specify the parameter recipe ID instead of the recipe name in the command sent from the robot end.

Follow these steps to view the ID of the parameter recipe:

1. Open a project, click the **Project Assistant** tab, and click to open the Parameter Recipe panel.



2. Click ⚙ in the Parameter Recipe panel.
3. Check the ID of the recipe you need in the leftmost column of the table.

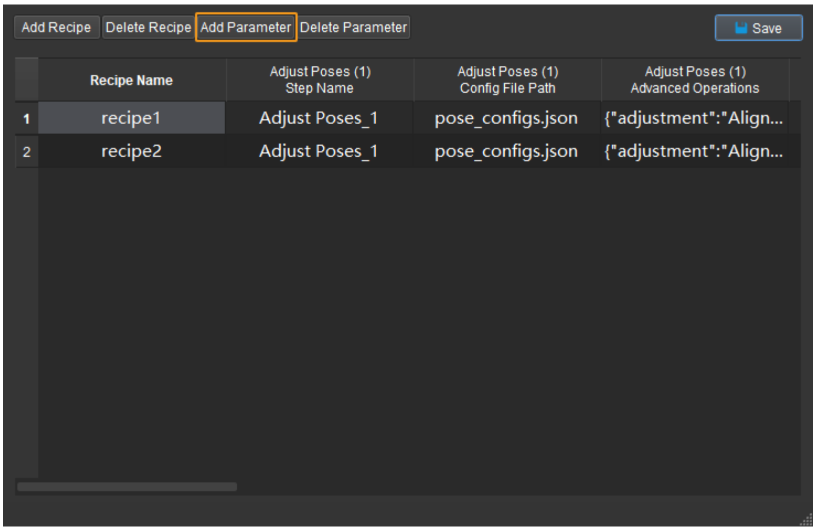


	Recipe Name	D Workpiece Recognition (Step Name	D Workpiece Recognition (workpieceName	pture Images from Camera ım of Reconnection Attemp	pture Images from Camera Max Capture Attempts
1	recipe1	3D Workpiece ...	model_edge	3	3
2	recipe2	3D Workpiece ...	model_edge	4	3
3	recipe3	3D Workpiece ...	model_edge	3	5

Add Parameters to the Parameter Recipe

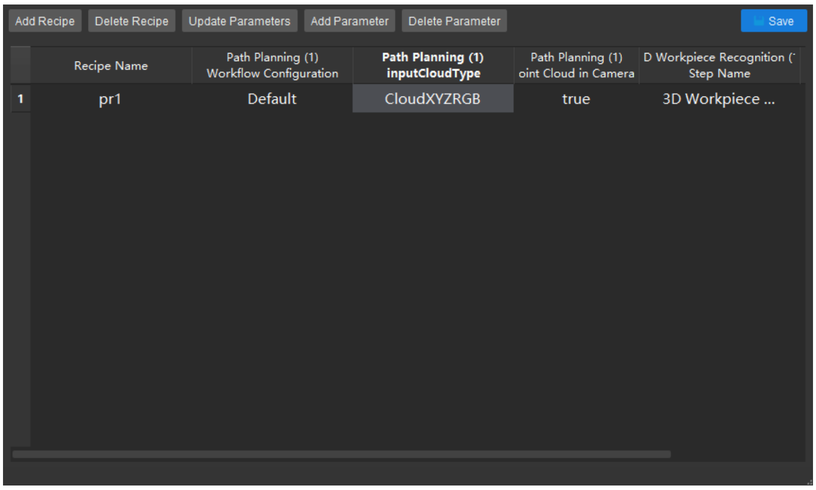
Follow these steps to add parameters to the recipe:

1. Select the recipe you want to modify in the **Parameter Recipe Editor** dialog box, and click the **[Add Parameter]** button.



	Recipe Name	Adjust Poses (1) Step Name	Adjust Poses (1) Config File Path	Adjust Poses (1) Advanced Operations
1	recipe1	Adjust Poses_1	pose_configs.json	{"adjustment":"Align...
2	recipe2	Adjust Poses_1	pose_configs.json	{"adjustment":"Align...

2. Select new Steps and parameters in the **Select Step Parameters** dialog box, and click **[OK]** to confirm.
3. Double-click the new parameter settings in the parameter recipe list to modify.



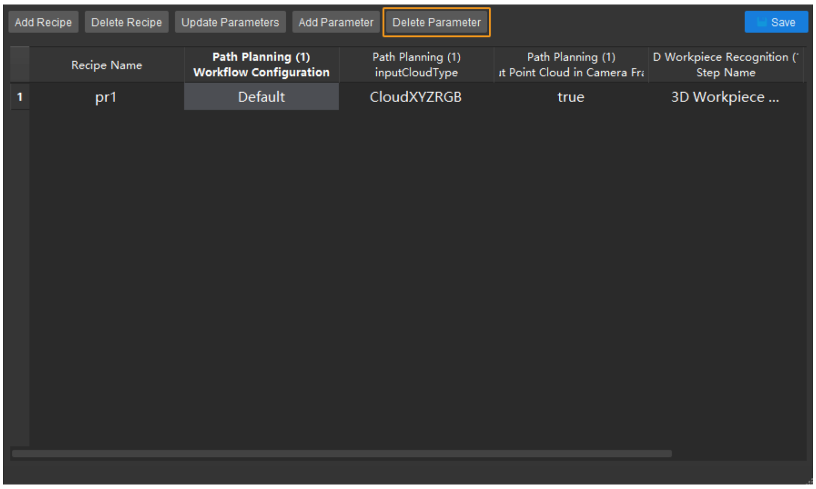
	Recipe Name	Path Planning (1) Workflow Configuration	Path Planning (1) inputCloudType	Path Planning (1) oint Cloud in Camera	D Workpiece Recognition (' Step Name
1	pr1	Default	CloudXYZRGB	true	3D Workpiece ...

4. Click **[Save]** in the **Parameter Recipe Editor** dialog box.

Delete Parameters from the Parameter Recipe

Follow these steps to delete parameters in the recipe:

1. Select the parameter you want to delete in the **Parameter Recipe Editor** dialog box, and click the **[Delete Parameter]** button.



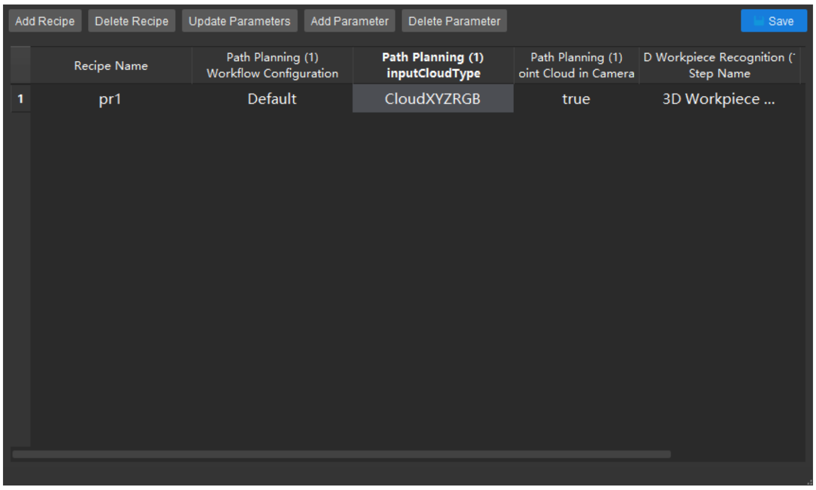
	Recipe Name	Path Planning (1) Workflow Configuration	Path Planning (1) inputCloudType	Path Planning (1) ıt Point Cloud in Camera Fra	D Workpiece Recognition (Step Name
1	pr1	Default	CloudXYZRGB	true	3D Workpiece ...

2. Click [Save] in the Parameter Recipe Editor dialog box.

Modify Specific Parameter Setting in the Parameter Recipe

Follow these steps to modify the settings of a specific parameters in the recipe:

1. Double-click the parameter settings you want to adjust in the parameter recipe entry to modify.



	Recipe Name	Path Planning (1) Workflow Configuration	Path Planning (1) inputCloudType	Path Planning (1) oint Cloud in Camera	D Workpiece Recognition (Step Name
1	pr1	Default	CloudXYZRGB	true	3D Workpiece ...

2. Click [Save] in the Parameter Recipe Editor dialog box.

Update Parameter Settings according to the Current Project Configuration

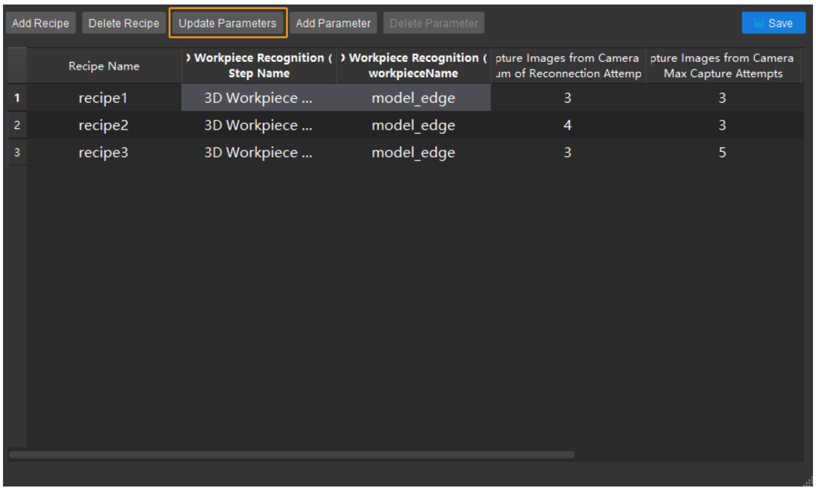
You can update the settings of specific parameters or all parameters in a recipe according to the configuration of the current project.

Follow these steps to update the settings of specific parameters in a recipe according to the configuration of the current project:

1. Select the parameter(s) of a specific recipe you want to update in the **Parameter Recipe Editor** dialog box.

   Press the `Ctrl` key to select multiple parameter settings.

2. Click the **Update Parameter** button on the toolbar.



	Recipe Name	› Workpiece Recognition (Step Name	› Workpiece Recognition (workpieceName	pture Images from Camera um of Reconnection Attemp	pture Images from Camera Max Capture Attempts
1	recipe1	3D Workpiece ...	model_edge	3	3
2	recipe2	3D Workpiece ...	model_edge	4	3
3	recipe3	3D Workpiece ...	model_edge	3	5

Follow these steps to update the settings of all parameters in a recipe according to the configuration of the current project:

1. Click the recipe ID in the **Parameter Recipe Editor** dialog box to select the entire recipe item.
2. Click the **Update Parameter** button on the toolbar.

Synchronize Parameter Settings Between Two Parameter Recipes

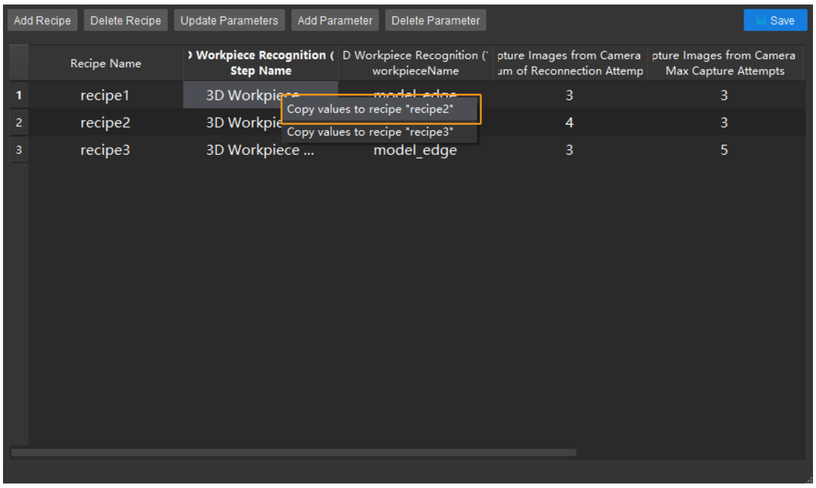
You can synchronize settings of specific parameters or all parameters between two recipes.

Follow these steps to synchronize specific parameter settings between two recipes:

1. Select the parameters of the recipe you want to synchronize in the **Parameter Recipe Editor** dialog box.

   Press the `Ctrl` key to select multiple parameter settings.

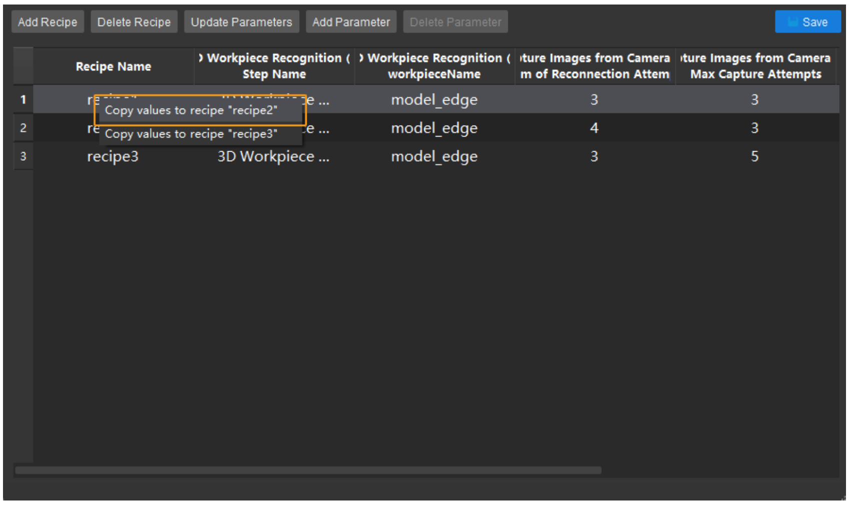
2. Right-click and select **Copy values to recipe “xx”** in the context menu.



	Recipe Name	› Workpiece Recognition (Step Name	D Workpiece Recognition (workpieceName	pture Images from Camera um of Reconnection Attemp	pture Images from Camera Max Capture Attempts
1	recipe1	3D Workpiece	model_edge	3	3
2	recipe2	3D Workpie		4	3
3	recipe3	3D Workpiece ...	model_edge	3	5

Follow these steps to synchronize all parameter settings between two recipes:

1. Click the recipe ID in the **Parameter Recipe Editor** dialog box to select the entire recipe item.
2. Right-click the recipe name and select **Copy values to recipe “xx”** in the context menu.



	Recipe Name	❯ Workpiece Recognition (Step Name	❯ Workpiece Recognition (workpieceName	ıture Images from Camera m of Reconnection Attem	ıture Images from Camera Max Capture Attempts
1	re	e ...	model_edge	3	3
2	re	e ...	model_edge	4	3
3	recipe3	3D Workpiece ...	model_edge	3	5

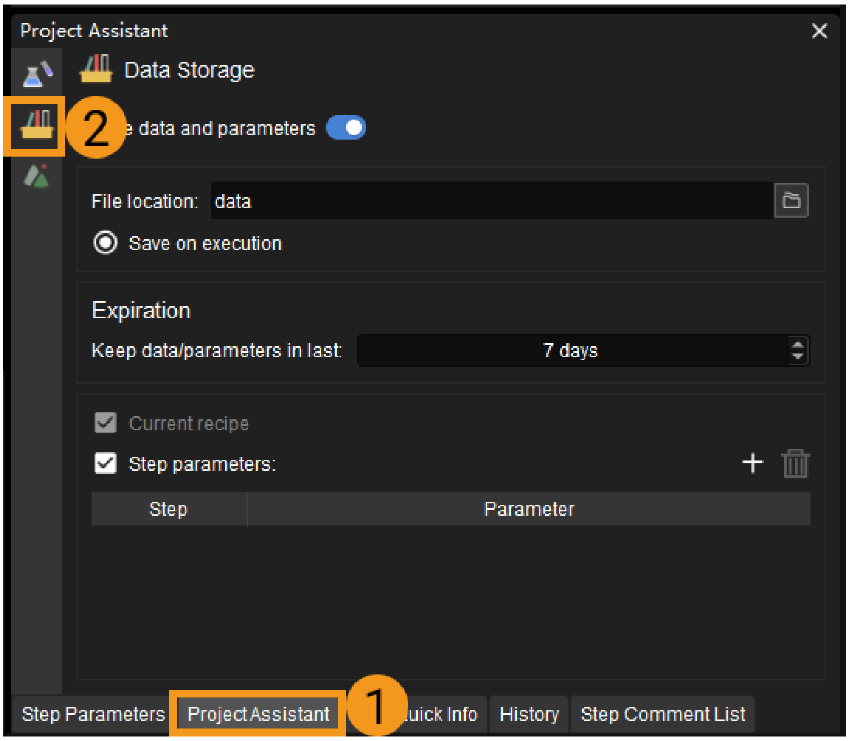
Copy values to recipe "recipe2"
Copy values to recipe "recipe3"

4.7.11. Save Data

Function

This function is used to save on-site data for efficient troubleshooting and regression testing.

Click **[Project Assistant]** tab in the lower right corner of the interface, and then click 📊 to open the data storage panel. The Data Storage interface is shown below.



e data and parameters ●

File location: data

○ Save on execution

Expiration

Keep data/parameters in last: 7 days

☑ Current recipe

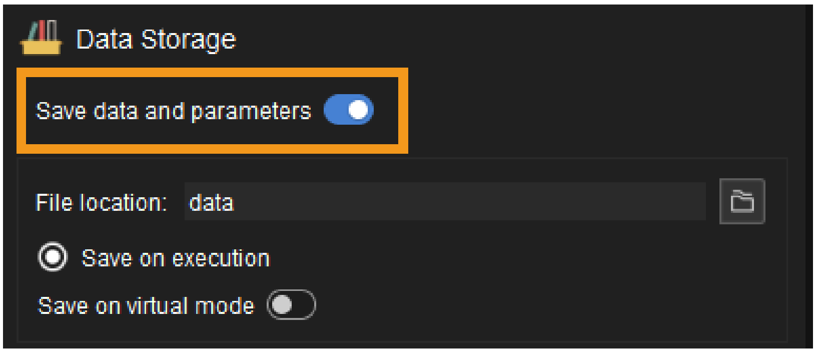
☑ Step parameters: + 🗑

Step	Parameter

> If you cannot find the Project Assistant tab in the interface, please go to **View** in the menu bar and check whether the checkbox before **Project Assistant** has been selected.

Instructions

1. Open the **[Save images and parameters]** switch, and the data will be automatically stored in the **data** folder in the project folder by default.



Data Storage

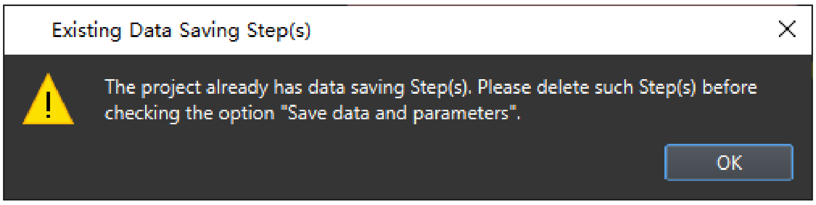
Save data and parameters ●

File location: data

◉ Save on execution

Save on virtual mode ○

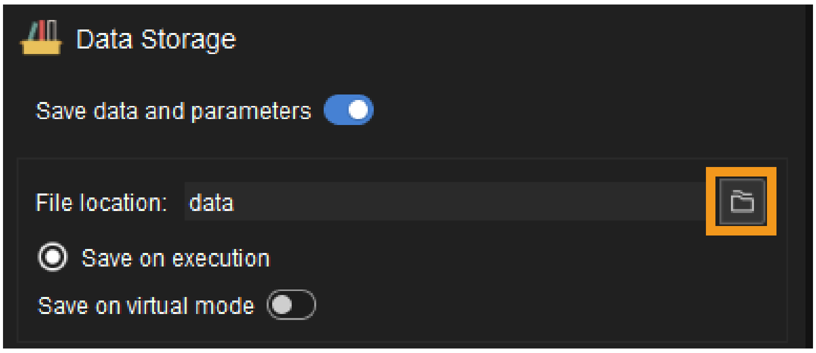
After opening the **[Save images and parameters]** switch, the software will check if the Procedure “Save Images and Step Parameters” exists in the project, and it is recommended to delete the Procedure and use the Data Storage feature instead.



Existing Data Saving Step(s)

The project already has data saving Step(s). Please delete such Step(s) before checking the option "Save data and parameters".

You can also click on 🗀 to specify another directory to store the data.



Data Storage

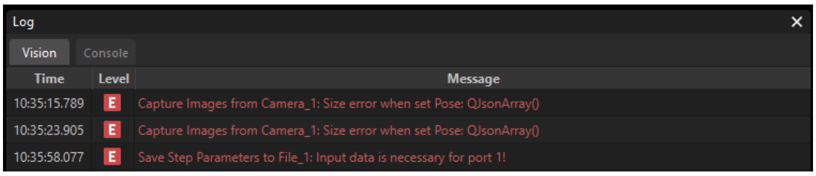
Save data and parameters ●

File location: data

◉ Save on execution

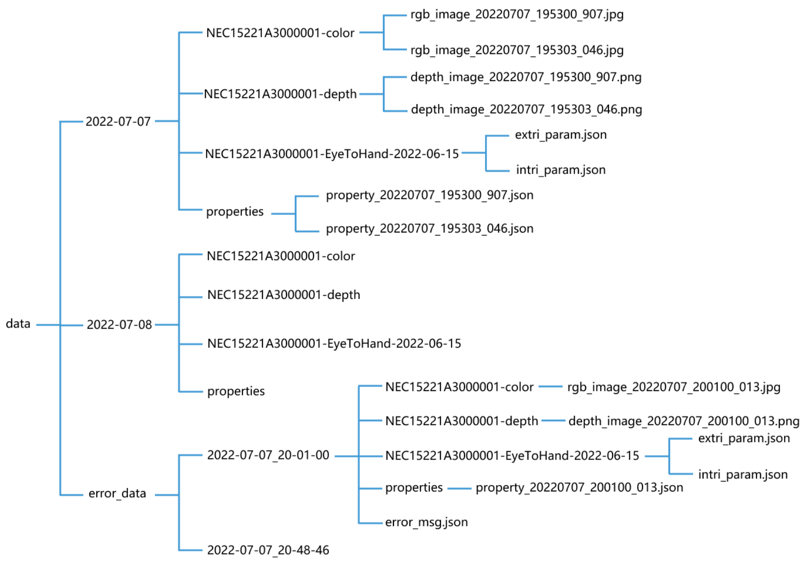
Save on virtual mode ○

When errors of the **E** level occur in the project, the abnormal data will be stored in the **error_data** folder in the data folder.

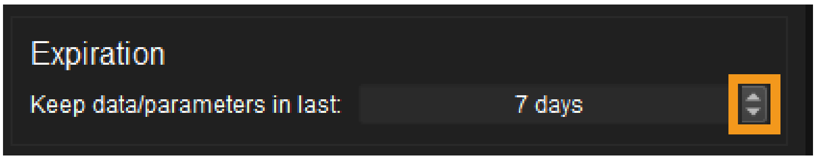


```
Time          Level  Message
10:35:15.789  E      Capture Images from Camera_1: Size error when set Pose: QJsonArray()
10:35:23.905  E      Capture Images from Camera_1: Size error when set Pose: QJsonArray()
10:35:58.077  E      Save Step Parameters to File_1: Input data is necessary for port 1!
```

For example, the figure below shows the directory structure and the names of saved files in the data folder. In this example project, depth map and color image data has been obtained from “Capture Images from Camera”. The names of image files are added with timestamps, which indicate the date and time of the day when the images are captured.


data
2022-07-07
NEC15221A3000001-color
rgb_image_20220707_195300_907.jpg
rgb_image_20220707_195303_046.jpg
NEC15221A3000001-depth
depth_image_20220707_195300_907.png
depth_image_20220707_195303_046.png
NEC15221A3000001-EyeToHand-2022-06-15
extri_param.json
intri_param.json
properties
property_20220707_195300_907.json
property_20220707_195303_046.json
2022-07-08
NEC15221A3000001-color
NEC15221A3000001-depth
NEC15221A3000001-EyeToHand-2022-06-15
properties
error_data
2022-07-07_20-01-00
NEC15221A3000001-color
rgb_image_20220707_200100_013.jpg
NEC15221A3000001-depth
depth_image_20220707_200100_013.png
NEC15221A3000001-EyeToHand-2022-06-15
extri_param.json
intri_param.json
properties
property_20220707_200100_013.json
error_msg.json
2022-07-07_20-48-46

2. Click ⬍ in the **Expiration** panel to adjust the maximum days to save the data. The data will be overwritten after this period.



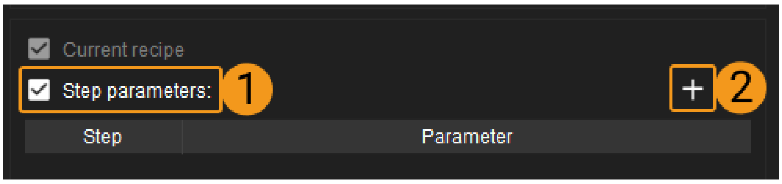
Expiration

Keep data/parameters in last: 7 days ⬍

3. Others

   - Current recipe: select to save the parameter recipes used in the project.
   - Step parameters: select to save the parameters of certain Steps so that the Step can be guaranteed to run successfully in data playback mode.

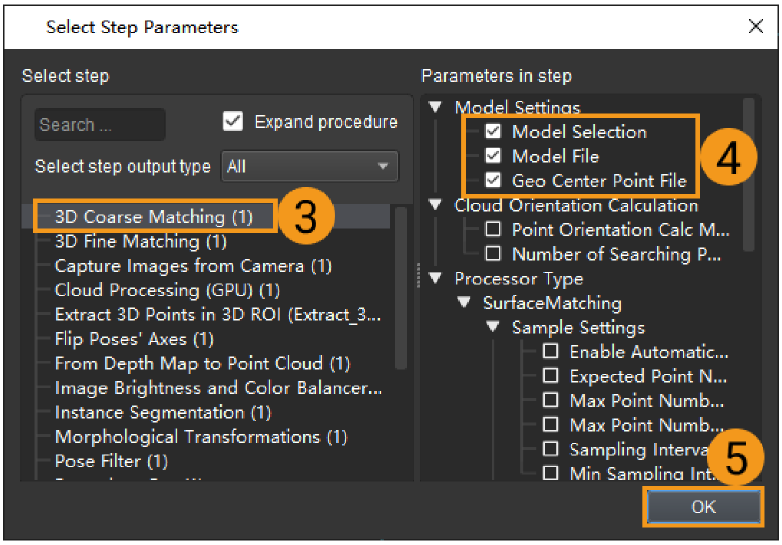
     Select **Step parameters**, click +. Select the Step and relevant parameters you want to save in the pop-up **Select Step Parameters** window, and then click **[OK]**.



☑ Current recipe

☑ Step parameters:

Step	Parameter



Select Step Parameters

Select step

Search ...

☑ Expand procedure

Select step output type All

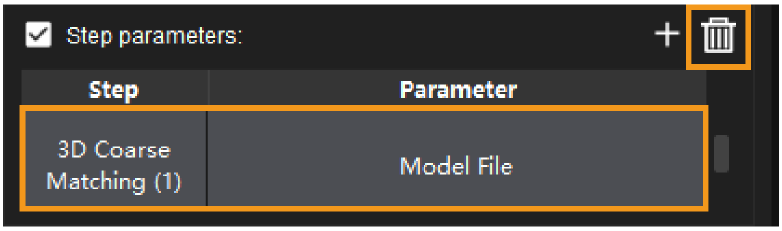
- 3D Coarse Matching (1)
- 3D Fine Matching (1)
- Capture Images from Camera (1)
- Cloud Processing (GPU) (1)
- Extract 3D Points in 3D ROI (Extract_3...
- Flip Poses' Axes (1)
- From Depth Map to Point Cloud (1)
- Image Brightness and Color Balancer...
- Instance Segmentation (1)
- Morphological Transformations (1)
- Pose Filter (1)

Parameters in step

- ▼ Model Settings
  - ☑ Model Selection
  - ☑ Model File
  - ☑ Geo Center Point File
- ▼ Cloud Orientation Calculation
  - □ Point Orientation Calc M...
  - □ Number of Searching P...
- ▼ Processor Type
  - ▼ SurfaceMatching
    - ▼ Sample Settings
      - □ Enable Automatic...
      - □ Expected Point N...
      - □ Max Point Numb...
      - □ Max Point Numb...
      - □ Sampling Interva
      - □ Min Sampling Int...

OK

To delete a selected parameter of a Step, select the parameter in the Step parameter list and then click 🗑.



☑ Step parameters: + 🗑

Step	Parameter
3D Coarse Matching (1)	Model File

- For data playback in virtual mode, if you want to use the saved parameters instead of current parameters, please select **Reuse parameters for virtual camera** at the bottom of the panel.

> The **Data Storage** tool enables to save data when multiple cameras are used. If there are multiple “Capture Images from Camera” Steps in the project, the data derived from all the cameras will be automatically saved.

4.7.12. Log

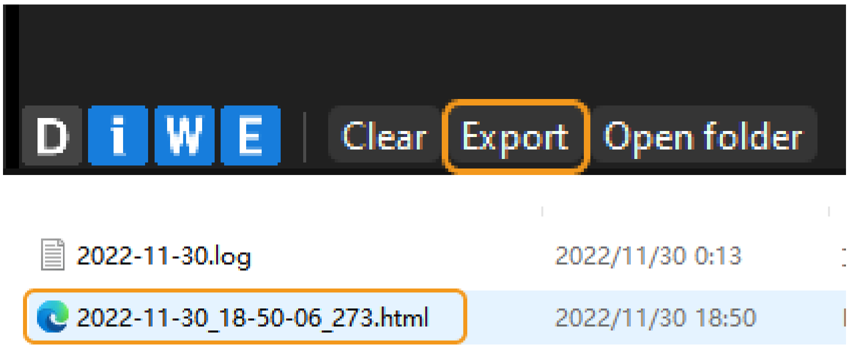
The log panel displays the log in real time while running a project/solution for users to find the execution record of a certain moment in this panel.

You can check logs that belong to two modules, i.e., **Vision** and **Console**. **Vision** logs provide information about the execution status of the vision solution/project, and **Console** logs provide information about the console.

The relative operations are described as follows.

Export Logs

Click the **[Export]** button at the bottom of the panel, and you can find the log file in HTML format in the pop-up window.

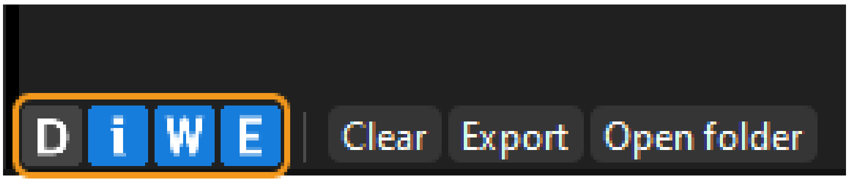


Filter Logs

There are four levels of log, as shown in the table below.

Level	Description
D	Debug information
i	Information
W	Warning Information
E	Error information

If you want to check the logs of a certain level, click the level tag in the lower left corner of the log panel. An tag in blue indicates that log information of this level will be displayed. When the tab is in gray, the log information of this level will not be displayed.



Other Operations

Clear Logs

Click the **[Clear]** button in the log panel to clear the current information.

Copy Logs

- **Copy part of the log of a certain line:** Double-click the log in the Message column, select the content you want to copy, and then right-click and select **Copy** to copy the selected content.
- **Copy the entire log of the line:** Double-click the log in the Message column, right-click and select **Select All** to copy the entire log in the current line. Then right-click again and select **Copy** to copy the entire log of the line.

4.7.13. Path Planning Tool

The usage and some of parameters of the path planning tool differs from that of Mech-Viz. To avoid confusion between the two, please refer to their respective manual when using the tools/software.

This section introduces the path planning tool and how to use it.

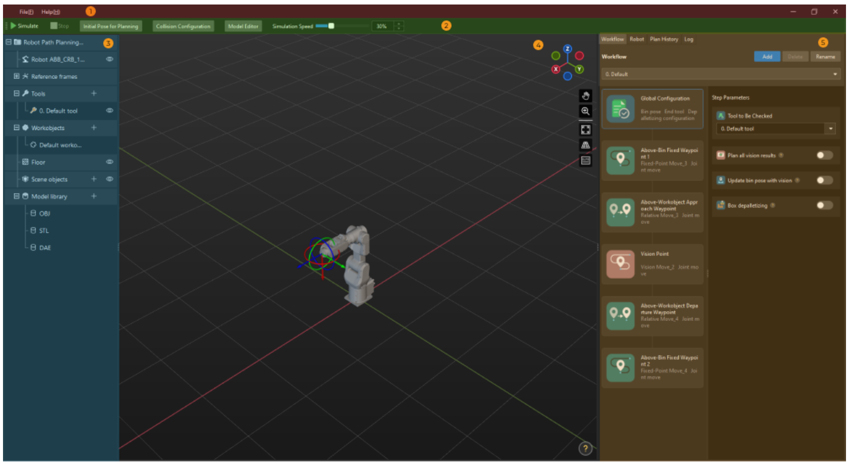
Function

The path planning tool is used to plan a collision-free motion path near the workobject for the robot.

The robot motion path is essentially the motion path of the TCP when the robot completes a task. When the robot moves, the tool may collide with the workobject or scene object. Therefore, to provide various resources and data required for the collision detection and path planning, you need to build a scene that is close to the real scenario, configure the robot tool, and input the vision points and scene point cloud. With the path planning tool, you can input these resources and data to enable the collision detection and path planning. The planned path will be output via the "Path Planning" Step, and therefore the robot can be guided to complete tasks such as picking.

Introduction to Interface

The interface of the path planning tool, as shown below, consists of five parts.



1. Menu bar	For resetting the settings, saving the settings, and exiting the path planning tool.
2. Toolbar	For project simulation, setting the initial robot pose, collision detection, and opening the model editor.
3. Project resource tree	For managing the resources used in the project, including tools, workobjects (i.e., the target objects to be picked or processed), and scene objects.
4. 3D simulation area	The area where the robot motion path, vision poses, point cloud, etc., are visualized during the simulation.
5. Functional panels	For configuring the workflow and robot pose and checking the plan history and log.

Basic Procedures

Before using the path planning tool, please connect the input and output ports of the **Path Planning Step** with others according to the port type.

After you connect the Steps properly, click **[Open the editor]** in the Step Parameters panel of the

Path Planning Step, select the robot model according to the actual robot model in use in the Robot Model Library window, and then you can enter the interface of the path planning tool.

The basic procedures for using the path planning tool are as follows. Click the link below to get started with the path planning tool.

1. Configure project resources	Configure the resources required in the project, including tools, workobjects, scene objects, etc.
2. Configure workflow	Create the robot motion control program in the form of a flowchart.
3. Configure collision detection	Detect possible collisions as the robot moves along the path.
4. Simulate and debug	Run and debug the project by using a simulated robot.

4.7.13.1. Interface Introduction

This section introduces the user interface of the path planning tool, including menu bar, toolbar, project resource tree, 3D simulation area, and functional panels.

Menu Bar

Option	Description	Shortcut
Reset All	Reset all settings in the path planning tool	Ctrl + N
Save	Save all current settings	Ctrl + S
Exit	Exit the path planning tool	Ctrl + W
User Manual	Open the user manual	N/A

Toolbar

Option	Description
Simulate	Run the simulated robot
Stop	Stop running the simulated robot
Initial Pose for Planning	Set the initial pose of the robot for the simulation
Collision Configuration	Configure the collision detection settings
Model Editor	Open the model editor
Simulation Speed	Set the running speed of the simulated robot

Project Resource Tree

The Resources panel provides the interfaces for the robot, tools, workobjects (i.e., the target objects to be picked or processed), scene objects, etc.

Robot

This entry shows the robot model used in the current project.

Right-click the robot name, and the following option will appear.

Option	Description
Open Robot File Directory	Open the folder where the robot model file is stored.
Reload Robot	Usually used after the parameter file of the local robot model is modified. Click this option to update the robot model and its parameters in the software.

Click 👁 to make the displayed robot semitransparent, and click 👁 to restore to the original display setting.

Reference Frames

This entry provides the interface for displaying all reference frames in the 3D simulation area.

Click 👁 to display the corresponding reference frame, and click 👁 to hide the reference frame. Click ... to enable or disable the “translation” or “rotation” of the manipulator.

Tools

This entry provides the interface for configuring the tools in the project. Please refer to Tools for how to configure the tools.

Click **[+]** to open the Tool Configuration window, where you can add tools.

The default tool is used to avoid errors due to unconfigured end tools.

Right-click the name of a configured tool, and the following options will appear.

Option	Description	Shortcut
Copy	Copy the current tool.	Ctrl + C
Paste	Paste the current tool.	Ctrl + V
Delete	Delete the current tool.	Delete
Rename	Rename the current tool.	None
Tool Configuration	Open the Tool Configuration window of the current tool.	None
Set As Active Tool	Set the current tool as the active tool.	None

Click 👁 to make the tool model semitransparent, and click 👁 to restore to the original display setting.

Workobjects

This entry provides the interface for configuring the workobjects in the project. Please refer to Workobjects for how to configure the workobjects.

Click **[+]** to open a Workobject Configuration window, where you can add workobjects.

“Default workobject” is the workobject configuration used for vision results without labels.

Right-click the name of a configured workobject, and the following options will appear.

Option	Description	Shortcut
Delete	Delete the current workobject.	Delete
Rename	Rename the current workobject.	None
Workobject Configuration	Open the Workobject Configuration window of the current workobject.	None

Floor

This entry provides the interface for modifying the height of the floor and floor display setting.

Right-click the Floor and you can modify the height of it in the pop-up window.

Click 👁 to hide the floor and click ◡ to restore.

Scene Objects

This entry provides the interface for configuring the scene objects in the project. Please refer to Scene Objects for how to configure the scene objects.

Click **[+]** to open a Scene Object Configuration window, where you can add scene objects.

Right-click the name of a configured scene object, and the following options will appear.

Option	Description	Shortcut
Copy	Copy the current scene object.	Ctrl + C
Paste	Paste the scene object.	Ctrl + V
Cut	Cut the current scene object.	Ctrl + X
Delete	Delete the current scene object.	Delete
Rename	Rename the current scene object.	None
Scene Object Configuration	Open the Scene Object Configuration window of the current workobject.	None

Click 👁 to make the scene object semitransparent; click 👁 to make the scene object transparent; and ◡ to restore to the original display setting.

You can also drag and drop the scene object name in the Resources panel to make the scene object model a child model or an individual model.

Model Library

This entry provides the interface for importing custom models to the project. All the custom models imported will be displayed in the Model Library, and the models will be categorized according to their formats.

Click **[+]** to open the file selection window to import an custom model.

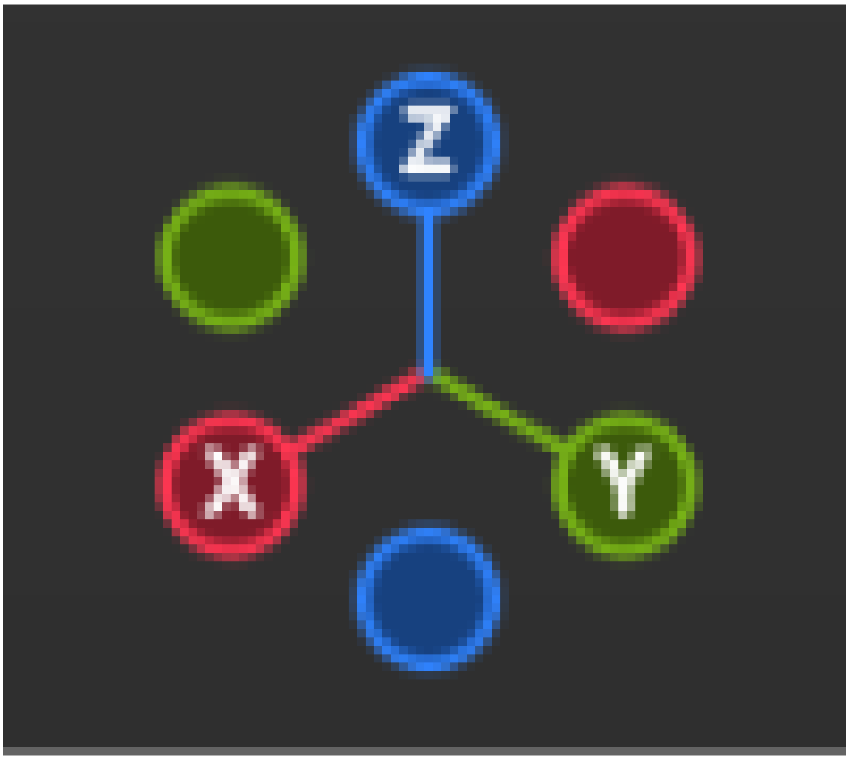
Right-click the imported custom model, and the following options will appear.

Option	Description	Shortcut
Copy	Copy the current model.	Ctrl + C
Paste	Paste the copied model.	Ctrl + V
Delete	Delete the current model.	Delete
Model Configuration	Open the Model Configuration window of the current model.	None
Model Editor	Open the model editor and the current model will be used as the reference model.	None
Open Folder	Open the folder where the model file is stored.	None

3D Simulation Area

The robot, scene objects, tools, workobject models, robot motion path, pick points, and collisions will be displayed in the 3D Simulation Area. Also in this area, you can drag and drop the models of scene objects.

The navigation gizmo shows the current view in the workspace. Click the icon of each axis (X, Y, Z, -X, -Y, -Z) to align the view with the axis. Press and hold the left mouse button and drag the navigation gizmo to rotate the view.



Rotate the view	Press and hold the **scroll wheel** and drag in any direction, or press and hold the **left mouse button** and drag the navigation gizmo in any direction.
Pan the view	Press and hold ✋ and drag in any direction, or press and hold **Shift + scroll wheel** and drag in any direction.
Zoom in/out	Press and hold 🔍 to zoom in by dragging upwards and zoom out by dragging downwards, or **scroll the mouse wheel**.

Fit the view	Click ⛶.
Switch to perspective/orthographic view	Click ⛰.
Show pressed keyboard keys and mouse buttons	Once ⌨ is selected (highlighted in blue), the keyboard keys and mouse buttons you pressed will be displayed in the lower right corner in the 3D Editing Workspace.

Functional Panels

The functional panels include Workflow, Robot, Plan History, and Log.

Workflow

You can configure the workflow in this panel.

- Click **[Add]** to create a new workflow.
- Click **[Delete]** to delete the workflow.
- Click **[Rename]** to edit the name of the current workflow.
- Click ▾ to switch the workflow.
- Select a step, and its parameters will be displayed on the right panel.
- Move the mouse cursor to the line connecting the Steps and then click ⊕ to insert a Step.
- Double-click the Step name to modify it. (The name of Global Configuration cannot be modified.)

Robot

You can set the joint positions, TCP, and robot home position in this panel.

Plan History

On this panel, you can view the plan history of Mech-Viz in detail. You can also use the plan history to troubleshoot problems, and optimize projects. For details, please refer to Plan History.

You can configure the related settings at the bottom of this functional panel.

Option	Description
Animation duration	When the plan history record is selected, 3D simulation area will display the duration of path animation.
Expire after	The expired time of the plan history. After the expired time, the previous plan history will be overwritten by the current plan history.
Load plan history	Load the saved plan history.
🧹	Clear all plan history in the panel.
?	A brief introduction of the plan history.

Right-click on an item in the plan history, you can see the following options.

Option	Description	Shortcut
Copy	Copy the selected content.	Ctrl + C
Search	After clicking **[Search]**, you can see two filter buttons at the bottom of the panel. You can search for the planning item you need according to the plan result type and the collided link index.	Ctrl+F
Collapse All	Collapse all sub-nodes in the panel and only displays the parent nodes.	N/A
Delete Selected Plan History	Delete the selected plan history item.	N/A

Log

On the log panel, you can view the software log in detail and manage the log settings at the bottom of the panel. Click one log entry and press `Ctrl` + `C`, and the entry will be copied to your clipboard.

Option	Description
D: Debug	Screen out the logs to be displayed according to their levels. Logs under selected levels will be displayed. You can select multiple levels.
i: Info	
W: Warning	
E: Error	
Clear	Clear all log messages on the page.
Export	Export the log messages to the HTML file. The exported files are located in the logs folder in the Mech-Viz installation directory.
Open folder	Click to open the logs folder in Mech-Viz's installation directory. The logs folder contains log files recorded by date.

4.7.13.2. Configure Project Resources

Resources include the robot, tools, workobjects, and scene objects, which are fundamental to a project.

In the Resources panel, you can do the following:

- Import tool models and scene object models which will be used for the collision detection.
- Set the symmetry of the tool, symmetry of the workobject, and picking relaxation, which may lead to more alternative waypoints and a more flexible path planning.

Tools

Tools, such as grippers and suction cups, are specially designed mechanical devices attached to the flange of the robot to perform various tasks.

Please refer to Tools for detailed instructions.

Workobjects

Workobjects refer to the objects on which the tool performs tasks, such as cartons, metal parts, and parts to be glued or welded.

Please refer to Workobjects for detailed instructions.

Floor

The simulated floor based on the real scenario.

The default floor plane is at the same level where the bottom of the robot base is located. You should adjust the height of the floor according to the actual situation.

Right-click **[Floor]** in the project resource tree on the left and modify the value or drag the slider bar in the pop-up window to adjust the height of the floor.

Scene Objects

Scene objects, usually including the safety fence, picking bin, tray, camera, and camera stand, refer to objects in the real scenarios where the robot performs various tasks.

Please refer to Scene Objects for detailed instructions.

4.7.13.2.1. Tools

This section introduces tools and corresponding configurations.

Introduction

Tools, such as grippers and suction cups, are mechanical devices attached to the end of the robotic arm to perform various tasks.

Model of the Tool

In order to visualize the tool in the 3D simulation area and make the tool involve the collision detection, you need to create a visualization model and a collision model of the tool and import them to the model library.

The following table shows the formats of the supported visualization model and collision model of the tool.

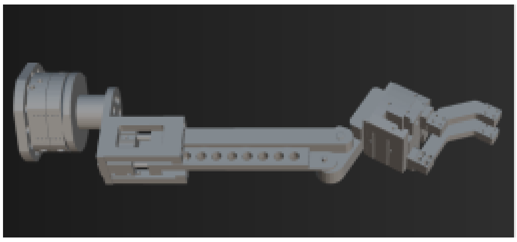
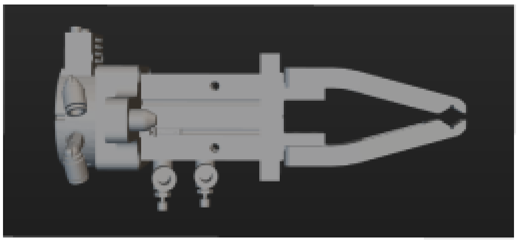
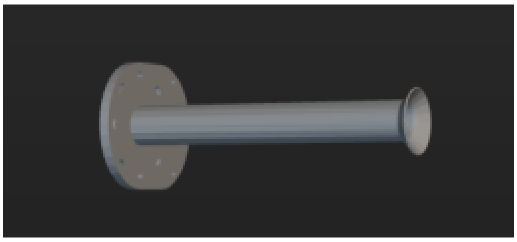
Format	STL	OBJ	DAE	Binvox
Visualization Model	√	√	√	×
Collision model	×	√	×	×

- The collision models of tools used in the software should be OBJ models that are composed entirely of convex hulls. You can use the Model Editor to create convex hulls on the original tool model and convert the model file to OBJ format.
- The visualization model can only be used to visualize the tool in the 3D simulation area and cannot be used in collision detection; while the collision model can only be used in collision detection and cannot be displayed in the 3D simulation area.

Rotational Symmetry

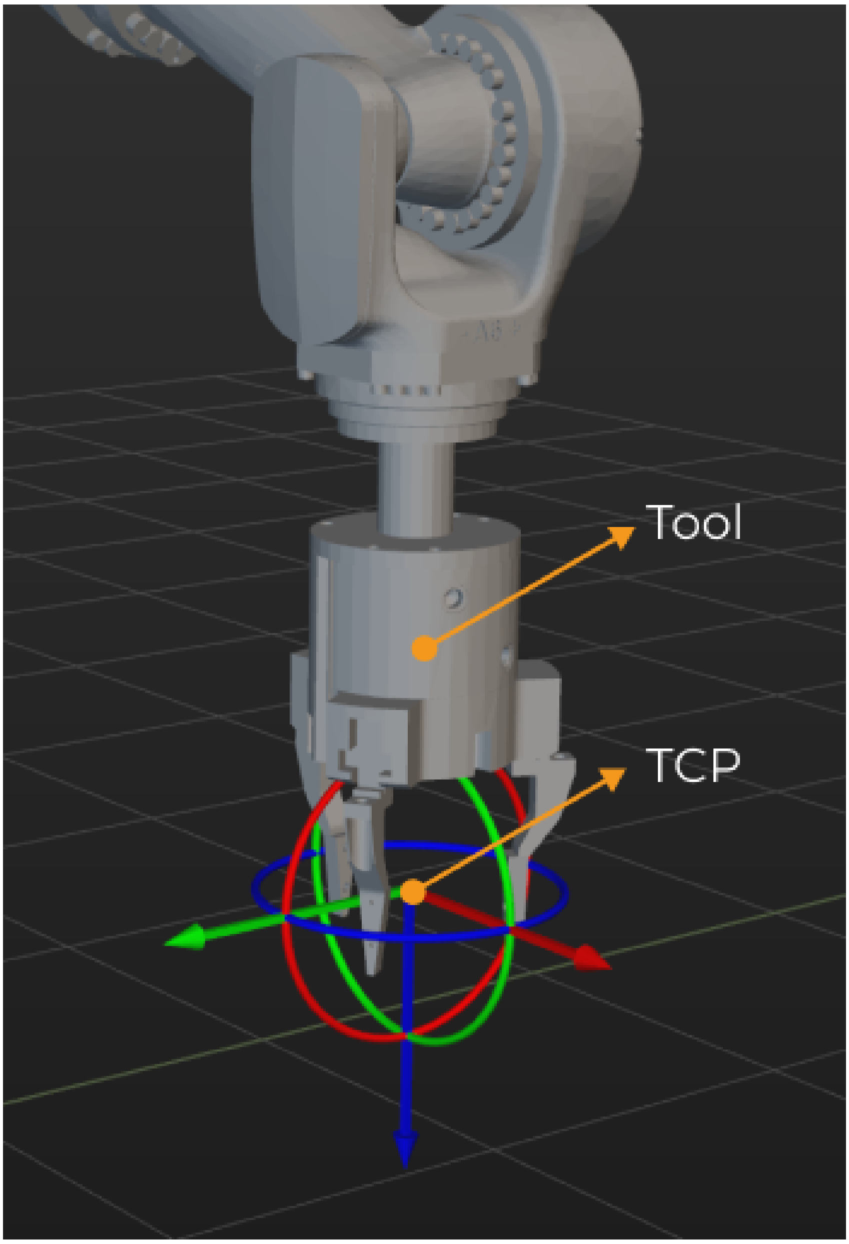
A rotational symmetric tool will coincide with itself after rotating around its axis of symmetry.

Configuring the symmetry of the tool can prevent the end of the robot from rotating meaninglessly when picking and placing, and therefore increase the picking success rate and reduce the planning time consumption. Therefore, the robot can move more smoothly and swiftly.

No Rotational Symmetry	
180° Rotational Symmetry	
Rotational Symmetry at Any Angle (Circular)	

Tool Center Point (TCP)

The tool is used to pick and place the object. When you move the robot to a specified point to perform pick-or-place task, you are actually moving the TCP (Tool Center Point) to that point. The default TCP is located at the end of the robot, but sometimes you need to adjust the TCP to ensure that the TCP is at the right place for the actual pick-and-place process.



Configure the Tool

Import the Collision Model and Visualization Model

Click the [+] next to Resources › Model Library, and select the collision model and visualization model in the pop-up window, and then click **[Open]**.

Add the Tool

Go to Resources › Tools and click **[+]** to open the Tool Configuration window.

1. Enter a **Tool name**.
2. Select the **Tool type** according to the actual situation. When a **depalletizing vacuum gripper** is used, please configure depalletizing vacuum gripper.
3. Select the **Collision model** in the drop-down list.

4. Select the **Visualization model** for the visualization in the 3D simulation area. When the position or size of the visualization model is not correct, please refer to Adjust the Position or Size of the Tool Model to adjust the model.
5. Set the **Rotational Symmetry** to **None**, **N-fold**, or **Circular** according to the actual situation. If you select **N-fold**, you also need to set the **Number of folds** parameter.
6. Follow either of the methods below to set the TCP.

Modify the TCP parameter
Modify the Euler angles or quaternions directly.
Use precise TCP data
If you have already obtained the accurate TCP data by other methods, you can click **Edit pose** and paste the data in the pop-up window.

7. Click **[OK]** in the end.

If you need to add more tools, please repeat the above steps.

Delete the Tool

Follow either of the following steps to delete the tool.

- Go to Resources › Tools, select the tool name and press the `Delete` key.
- Go to Resources › Tools, right-click the tool name and select **[Delete]** in the context menu.

Modify the Tool

1. Follow either of the following steps to open the Tool Configuration window.
   - Go to Resources › Tools, double-click the tool name.
   - Go to Resources › Tools, right-click the tool name and select **[Tool Configuration]** in the context menu.
2. Modify the parameters according to the actual requirement.
3. Click **[OK]**.

Set as Active Tool

If you only add one tool, the current tool is the active tool.

If you add multiple tools, the first added tool will be used as the active tool by default.

To set a tool as active tool, go to Resources › Tools, right-click the tool you want to use and select **[Set as Active Tool]**.

Adjust the Position or Size of the Tool Model

Follow the steps below to adjust the tool model when its position is not correct.

1. Double-click the tool model under Resources › Model Library.
2. Adjust the pose parameters in the pop-up Edit Model Transform window to adjust the model's position in the 3D simulation area.

Follow the steps below to adjust the model when its size is not correct.

1. Double-click the tool model under Resources › Model Library.
2. In the pop-up Edit Model Transform window:
   - Adjust the **Scale** if you want to adjust the scale of the entire model.
   - If you want to adjust the scale on the individual X, Y, or Z direction, clear the **Keep ratio for X,Y,Z** checkbox, and adjust the scale on each direction respectively.

4.7.13.2.2. Workobjects

This section introduces workobjects and how to configure them.

Introduction

Workobjects refer to the objects on which the tool performs tasks, such as cartons, metal parts, parts to be glued or welded, etc.

Rotational Symmetry

A rotational symmetric workobject will coincide with itself after rotating around its axis of symmetry. Configuring the rotational symmetry of the workobject can prevent the end of the robot from rotating meaninglessly when picking and placing, and therefore the picking success rate can be increased and the planning time can be reduced. Therefore, the robot can move more smoothly and swiftly.

Please refer to Rotational Symmetry of Workobjects for detailed information.

Picking Relaxation

When the robot picks certain workobjects, the picking pose of the robot is allowed to be changed flexibly within a specific angle range, which is the picking relaxation.

Please refer to Picking Relaxation for detailed information.

Solution Selection Strategy

- **Minimum tool rotation**: When this strategy is selected, during the whole pick-and-place process, the workobject pick point that causes the minimum rotation degree of the tool's Z-axis will be selected as the optimal pick point. This strategy can prevent the tool from rotating in vain after picking the object, and avoid falling the picked workobject.
- **Minimum difference between tool and vision pose**: When this strategy is selected, the workobject pick point that has the minimum angle of deviation with the workobject pose sent by Mech-Vision will be selected as the optimal pick point.
- **Minimum collision between tool and point cloud**: When this strategy is selected, the workobject pick point that leads to the minimum volume of collision between the tool and the workobject point clouds will be selected as the optimal pick point.

Workobject Configuration

Add the Workobject

1. Enter a **Workobject name**.

2. According to the actual situation of the workobject, select **Around Z-axis**, **Around Y-axis**, or **Around X-axis** as the **Symmetry type** (you can select multiple options), and then set the **Number of folds** and **Attempt range**. You do not need to set this parameter if the workobject is not rotational symmetric.
3. According to the actual situation of the workobject, select **Around Y-axis** or **Around X-axis** for the **Picking relaxation** parameter (you can select multiple options), and then set the **Attempt range** and **Step**. You do not need to set this parameter if a picking relaxation is not applicable.
4. Select a **Solution selection strategy** according to the actual requirement.
5. Click **[OK]**.

If you need to add more workobjects, please repeat the above steps.

Delete the Workobject

Follow either of the following steps to delete a workobject.

- Go to Project resource tree › Workobjects, select the workobject, and press the `Delete` key.
- Right-click the workobject under Project resource tree › Workobjects and select **[Delete]** in the context menu.

Modify the Workobject

1. Follow either of the following steps to open the Workobject Configuration window.
   - Go to Project resource tree › Workobjects and double-click the workobject.
   - Right-click the workobject under Project resource tree › Workobjects and select **[Workobject Configuration]** in the context menu.
2. Modify the parameters according to the actual requirement.
3. Click **[OK]**.

4.7.13.2.3. Scene Objects

This section introduces the scene objects and corresponding configurations.

Introduction

Scene objects, usually including the safety fence, picking bin, tray, camera, and camera stand, refer to objects in the real scenarios where the robot performs various tasks.

Scene Object Model

Adding scene object models makes the scene in the software closer to the real scenario, which facilitates the collision detection and path planning.

You can create models of a cuboid, bin, or cylinder. You can also use a custom model imported from outside the software. The supported format of the visualization model and collision model are shown in the table below.

Format	STL	OBJ	DAE	Binvox
Visualization Model	√	√	√	×

Collision model	√	√	×	×

The Parent-Child Relationship of the Scene Objects

You can establish a parent-child relationship between scene objects. Once the relationship is established, the child model will be moved as well if the parent model is moved, and the child model will be deleted as well if the parent model is deleted. However, the parent model will not be affected if you move or delete the child model individually.

Configure the Scene Object

You can add, modify, or delete the scene objects in the project resource tree.

Add a Scene Object

Please follow the steps below to add a scene object model of the **Cuboid**, **Cylinder**, or **Bin** type.

1. Go to Project resource tree › Scene objects, and click [+] to open the Scene Object Configuration window.
2. In the **Object Settings** tab, select the **Scene model** type in the drop-down list.
   - Set the **X, Y, Z (length, width, height)** for a **Cuboid** model.
   - Set the **Radius and Height** for a **Cylinder** model.
   - Set the **X, Y, Z (length, width, height), and Thickness** for a **Bin** model.
3. Enter a custom name for the scene object model in the **Object name** text box.
4. By default, the newly added scene object is located at the origin of the robot base reference frame. You can click the [Object Pose] tab in the Scene Object Configuration window to switch the configuration interface, and modify the parameters to adjust the pose of the scene object model.
5. Click [OK].

Please follow the steps below to add a **Custom model**.

1. Click the [+] next to Project resource tree › Model library, and select the collision model and visualization model in the pop-up window, and then click [Open].
2. Go to Project resource tree › Scene objects, and click [+] to open the Scene Object Configuration window.
3. In the **Object Settings** tab, select **Custom model** as the **Scene model**.
4. Set the scene object model used for the collision detection as the **Collision model**.
5. Set the scene object model used for visualization in the 3D simulation area as the **Visualization model**.
6. Enter a custom name for the scene object model in the **Object name** text box.
7. By default, the newly added scene object is located at the origin of the robot base reference frame. You can click the [Object Pose] tab in the Scene Object Configuration window to switch the configuration interface, and modify the parameters to adjust the pose of the scene object model.
8. Click [OK].

If you need to add more scene object models, please repeat the above steps.

> If you select an added model in the project resource tree and then click **[+]**, a child model of the current model will be added, and the default pose of the child model will be the same as that of the parent model.
>
> You can drag the scene object in the project resource tree to change it into a child model or individual model.

Delete the Scene Object

Follow either of the following steps to delete a scene object.

- Select the scene model in the 3D simulation area or the model name under Project resource tree › Scene objects and press the `Delete` key.
- Right-click the scene object in the 3D simulation area or the model name under Project resource tree › Scene objects and select **[Delete]** in the context menu.

Modify the Scene Object

1. Follow either of the following steps to open the Scene Object Configuration window.
   - Double-click the object model in the 3D simulation area.
   - Go to Project resource tree › Scene objects and double-click the scene object name.
   - Right-click the name of the scene object under Project resource tree › Scene objects and select **[Scene Object Configuration]** in the context menu.
2. Modify the parameters according to the actual requirement.
3. Click **[OK]**.

> You can also select the scene object in the 3D simulation area, and adjust the model pose with the manipulator.

4.7.13.3. Workflow Configuration

You can build a workflow, i.e. a robot motion control program in the form of a flowchart, in the Workflow tab.

Configure the Default Workflow

You can modify the default workflow to plan a feasible robot motion path for picking.

1. Global Configuration
   - **Tool to Be Set** is a mandatory parameter that specifies the end tool used in the current project.

     Click the Step and click ▾ in the **Step Parameters** panel to select the corresponding tool configured in the Project Resources panel.

   - You can enable the options below and configure the parameters according to the actual requirements.

Plan all vision results	This option enables to loop the current workflow until all received vision results are used for the path planning.

Update bin pose with vision	This option dynamically updates the position of the bin.
Box depalletizing	This option is used for depalletizing scenarios.

2. Above-Bin Fixed Waypoint 1

   This Step sets a waypoint where the robot reaches before it approaches the workobject.

   Please refer to Fixed-Point Move for setup instructions and parameter description.

3. Above-Workobject Approach Waypoint

   This Step sets a waypoint above the workobject when the robot approaches the workobject.

   Please refer to Relative Move for setup instructions and parameter description.

4. Vision Point

   This Step specifies the vision point.

   Please refer to Vision Point for setup instructions and parameter description.

5. Above-Workobject Departure Waypoint

   This Step sets a waypoint above the workobject when the robot leaves the workobject.

   Please refer to Relative Move for setup instructions and parameter description.

6. Above-Bin Fixed Waypoint 2

   This Step sets a waypoint where the robot reaches after it leaves the workobject.

   Please refer to Fixed-Point Move for setup instructions and parameter description.

Add or Delete Steps in the Workflow

If the above waypoints cannot meet the actual requirement, you can add waypoints by adding “Fixed-Point Move” or “Relative Move” Steps to the workflow.

Operation	Instruction
Add Steps	Move the mouse cursor to the line connecting the Steps, and click ⊕, and then select the Step you want to add.
Delete Steps	Right-click the Step and then click **[Delete]**. You can also delete it by clicking the Step and then press the `Delete` key.

Add a New Workflow

The same resources can be used in different workflows which result in different planned paths.

In scenarios where the project involves different workobjects or bins, you may need to add multiple workflows and change the workflow by switching the parameter recipe.

Operation	Instruction

Add a new workflow	Click [**Add**], enter a workflow name in the pop-up window, and then click [**OK**] to create a new workflow.
Switch the workflow	Click ▼ on the far right of the workflow name, and select another workflow in the drop-down list.
Rename the workflow	Click [**Rename**], enter the new name in the pop-up window, and click [**OK**].
Delete the workflow	When there are at least two workflows, click [**Delete**] to delete the current workflow.

4.7.13.4. Configure collision detection

Collision configuration is used to detect and avoid possible collisions. By default, collisions between any two of the robot, robot tool, and scene objects will be detected. Once **Enable collision detection with point cloud** is enabled, the collision between the input point cloud and the robot tool will be detected.

1. Click the [**Collision Configuration**] button in the toolbar and [**Enable collision detection with point cloud**].

   Set the following parameters based on the project's allowed level of collisions.

   - Point cloud cube side length

     In the collision detection, cubes with each point in the point cloud as their centers will be generated. Detecting the collision between the tool and the point cloud is actually detecting the collision between the tool and the cubes. The longer the cube's side length, the higher the possibility of the tool colliding with the point cloud, and the larger the collision volume.

   - Tool-point cloud collision volume threshold

     When the collision volume between the tool and the point cloud is above the set threshold, it will be assumed that a collision occurs; otherwise, it will be assumed that no collision occurs.

   - Point cloud collision record

     - Record: The plan history will record the positions and quantity of the point clouds where collisions occur. When the user views the planning history, the recorded point cloud will be highlighted in the 3D simulation area. Recording affects the execution speed.
     - Do not record: The user will not be able to trace which part of the point cloud was involved in collision(s) in the plan history. On the other side, the execution speed is higher when not recording the point cloud in collision.

2. After you finish the configuration, click [**OK**].

4.7.13.5. Simulate and Debug

After you finish the above configurations, you can control the simulated robot to test whether there is a collision or other error in the planned path.

- When you simulate the project for the first time, you should click [**Run**] in the main interface of Mech-Vision to input the poses and point cloud data to the path planning tool.

- In the path planning tool window, clicking [**Simulate**] will simulate the project based on the poses and point cloud data that result from the last run of the Mech-Vision project.

1. Click [**Simulate**] on the toolbar.
2. If an error occurs during the simulation, such as a collision or unreachable waypoint occurs, you can check the error messages in **Plan History** and **Log**, and therefore you can locate the error quickly. Then you can adjust the project resource configuration, workflow, and collision configuration according to the error messages to debug the project.
3. If no error occurs in the simulation, click File › Save to save the configurations.

After you finish using the path planning tool, click File › Exit to close the path planning tool. Then you can select the workflow you created in the drop-down list under **Workflow Configuration** in the Step Parameters panel of the **Path Planning** Step.

4.8. User Guide of Deep Learning

Introduction

This chapter describes how **deep learning** works in the 3D-vision guided robot picking scenario. After reading from **Basics to Getting Started**, you will learn how to use Mech-Mind deep learning algorithm modules to empower robot picking.

1. Basics: Introduces the basics of deep learning and how Mech-Mind deep learning algorithm modules can assist in 3D vision.
2. Getting Started: Introduces deep learning use scenarios, how to choose deep learning models according to the actual situation, use process, and finally practice with video tutorials.

Reading Guide

If you need to used deep learning in 3D-vision guided robot picking scenario, please refer to the following content to acquire the necessary knowledge.

- Read the following chapter to learn about deep learning.

  Learn about Deep Learning

- Read the following chapter to learn how to use deep learning.

  Getting Started with Deep Learning

4.8.1. Basics of Deep Learning

In some vision recognition processes, traditional matching and clustering methods may not achieve satisfactory results. For instance, when picking highly reflective workobjects, the quality of point clouds may be poor, leading to inaccurate results from 3D matching methods. Additionally, when picking workobjects that are closely placed against each other, the results from clustering methods may not be desirable, adversely affecting vision recognition and robot picking. Given these circumstances, you can consider using deep learning for assistance in recognition.

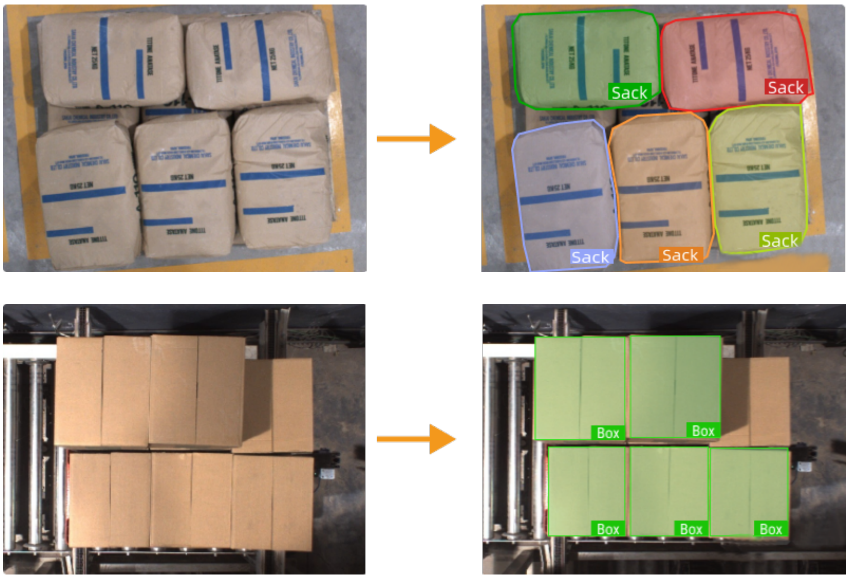
]

Deep learning falls within the realm of artificial intelligence and involves intricate neural network models.Once a large amount of data is input, deep learning techniques can simulate the human learning process, predict or identify patterns based on extensive datasets, extract data features, and subsequently perform relevant tasks.

Mech-Mind provides an instance segmentation algorithm module in the field of 3D vision-guided robot picking. When traditional methods fail to achieve the expected results, the instance segmentation method can be used to assist in recognition. The following sections will introduce the effects of the instance segmentation algorithm module in various picking scenarios.

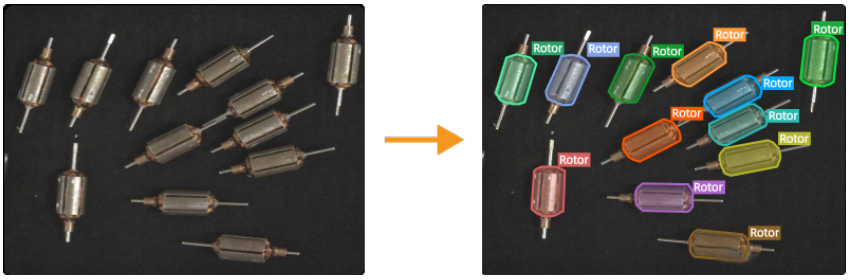
Palletizing and Depalletizing

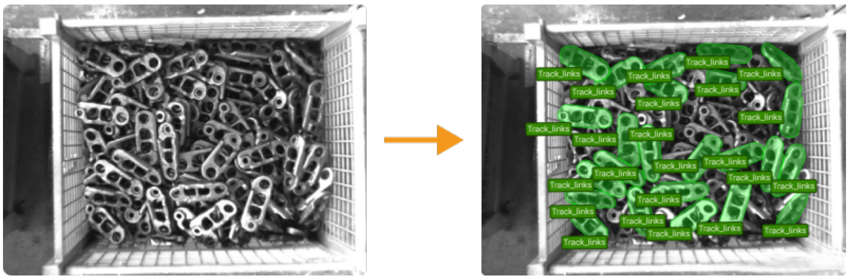
Palletizing and depalletizing refers to disassembling objects such as cartons, turnover boxes, and sacks from the pallet as required, and placing them on the pallet or related equipment (such as bag opening devices, conveyors, etc.). The instance segmentation algorithm module can segment cartons and sacks that are closely placed against each other, as shown in the figure below.



Loading and Unloading

Loading and unloading refers to the handling and picking of complex workobjects, metal components, and irregular parts in industries such as automotive, steel, and machinery. The instance segmentation algorithm module can segment the contour of each target object while simultaneously outputting class labels. For example, it can segment scattered and stacked rotors, track links, as shown in the figure below.



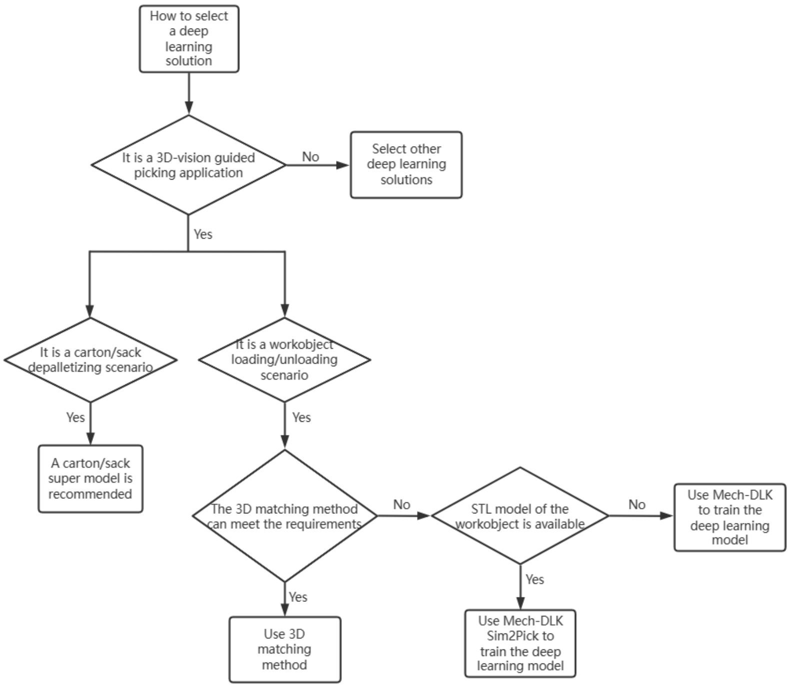


4.8.2. Getting Started with Deep Learning

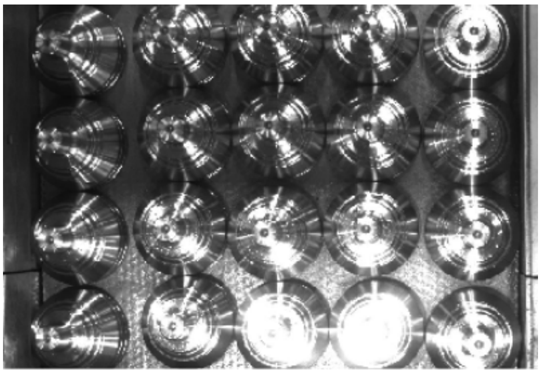
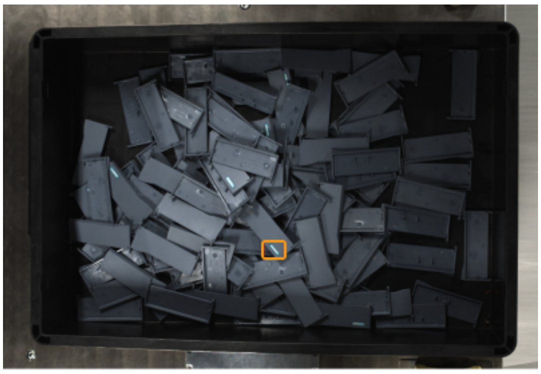
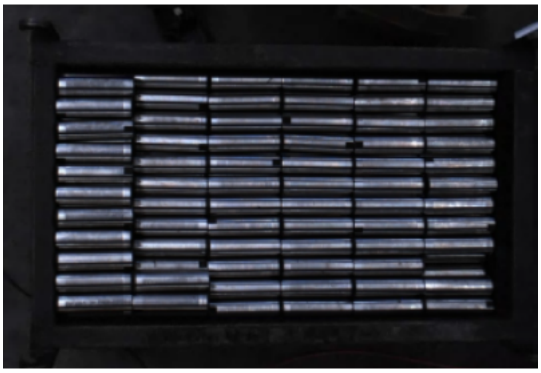
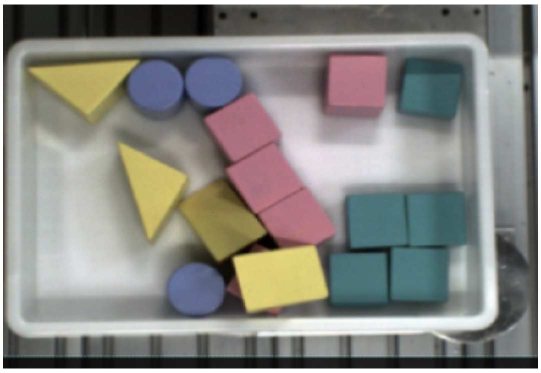
After learning about the basics of deep learning, this section will guide you to dive deeper into deep learning. You need to select a deep learning model according to your actual requirement, and then you can apply the model in relevant tasks.

Select Deep Learning Solution

When designing the vision solution, you should determine the use scenario (Loading and unloading or palletizing and depalletizing) of the solution first. Also, you need to assess whether the recognition performance of traditional 3D matching methods can meet the requirements. Then you can decide whether to employ deep learning and the specific deep learning solution can be determined. If you cannot select a deep learning solution based on the actual situation, please contact Mech-Mind Technical Support for assistance.


How to select a deep learning solution
It is a 3D-vision guided picking application
No
Select other deep learning solutions
Yes
It is a carton/sack depalletizing scenario
It is a workobject loading/unloading scenario
Yes
A carton/sack super model is recommended
Yes
The 3D matching method can meet the requirements
No
STL model of the workobject is available
No
Use Mech-DLK to train the deep learning model
Yes
Use 3D matching method
Yes
Use Mech-DLK Sim2Pick to train the deep learning model

If traditional methods fail to address the following issues during the vision recognition process, employing deep learning may provide a solution.

No.	Challenges to traditional methods	Illustration
1	The surfaces of the workobjects are highly reflective, and the quality of the point cloud is poor.	
2	The curve of the workobject point cloud is not obvious, and there are few feature points in the point cloud.	
3	When the workobjects are arranged neatly and placed closely against each other, the performance of point cloud clustering is poor.	
4	The workobject feature is only visible in the 2D image but not visible in the point cloud.	

No.	Challenges to traditional methods	Illustration
5	The project has a high requirement on the vision cycle time. When there is a large number of workobjects, the matching time using the point cloud model can be long.	

Apply Deep Learning

After the deep learning solution is determined, please refer to How to Apply Deep Learning to learn about the workflow of applying deep learning to assist the 3D-vision guided robot picking.

5. Mech-Viz

Welcome to Mech-Viz user manual!

Mech-Viz is a dynamic path planning software for industrial robots. It generates a collision-free robot motion path dynamically based on the vision result received from the vision service. With the intuitive graphical interface of the software, it is easy to get started with it even you do not have a background in programming. The software supports one-click simulation, and it is integrated with tools used for collision detection, carton combination, vacuum gripper smart offset, mixed case palletizing, and others, which can meet the requirements of various industrial scenarios.

This manual consists of the following contents:

No.	Chapter	Content
1	Brief Introduction of Mech-Viz	Overview of the basic concepts, and introduction of the user interface
2	Configuration of the Resources	Detailed instructions on configuring robot, tool, workobject, scene, and other resources in the project
3	Build a Workflow	Workflow of building a project and examples
4	Step Library	Step parameters details
5	Project Simulation and Improvement	Instructions for the use of collision detection, project simulation, plan history, and log
6	Built-in Tools	Instructions for the use of tools such as the model editor, vacuum gripper configurator, and pallet editor
7	Appendix	Create and import the robot model, option settings, and display settings

5.1. Mech-Viz Introduction

5.1.1. Fundamentals

Project

Projects refer to the robot path planning projects created in Mech-Viz.

Once you have completed the necessary setup of the project in Mech-Viz, you can use the project to plan a path and guide the robot to move. All the configurations of the project are stored in the folder with the same name as the project.

Project Resources

Project resources refer to various fundamental resources used in the project, including the robot, tools, workobjects, and scene objects.

Robot	In Mech-Viz, robots refer to multi-joint robotic arms or the gantry robot for industrial use.

Tools	Tools, such as grippers and suction cups, are specially designed mechanical devices attached to the flange of the robot to perform various tasks.
Workobjects	Workobjects refer to the objects on which the tool performs tasks, such as cartons, metal parts, parts to be glued or welded, etc.
Scene Objects	Scene objects, usually including the safety fence, picking bin, tray, camera, camera mounting frame, and so on, refer to objects in the real scenarios where the robot performs various tasks.

Workflow

The workflow refers to the robot motion control program built in the flowchart form in Mech-Viz.

Pose

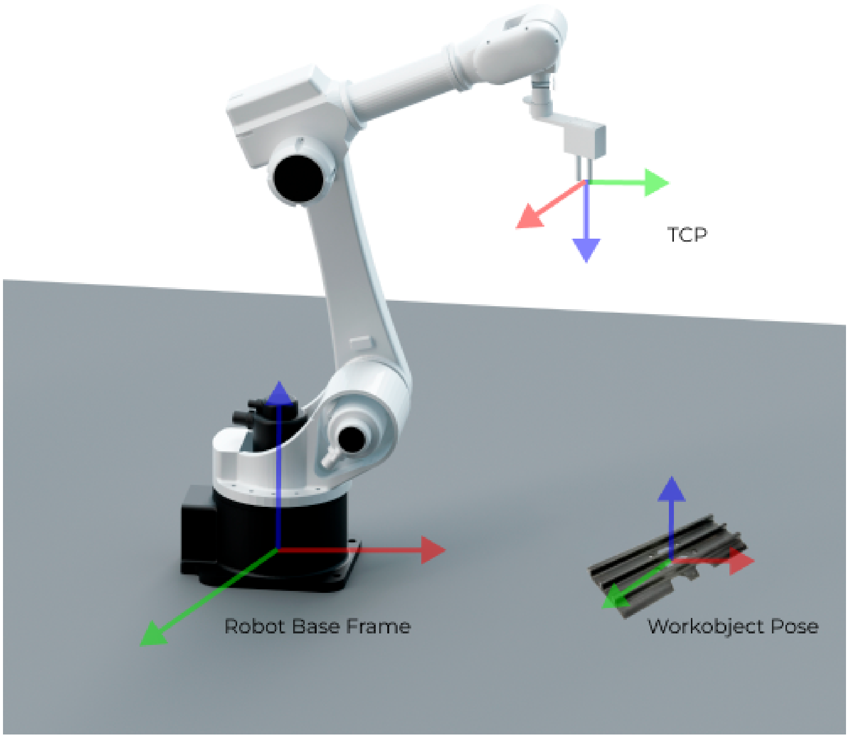
TCP (Tool Center Point)

TCP is the pose of a specific point on the tool or outside the tool relative to the flange at the robot end.

In order to complete tasks such as picking, we usually say that the robot should move to a specific point in space, which actually means its TCP should move to that point.

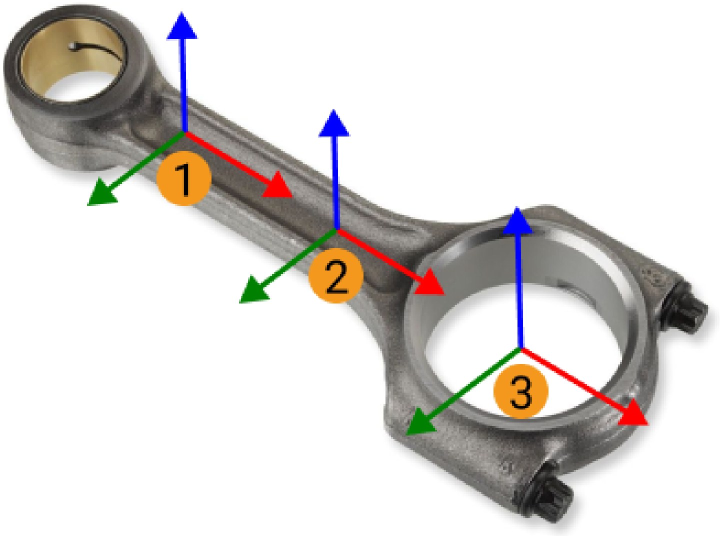
Workobject Pose

Workobject pose is the pose of a certain point on the workobject relative to the robot base frame. When placing the workobject, we usually make the workobject pose coincide with a certain target pose.



Workobject Pick Point

There may be several different pick points on one workobject, which are output by Mech-Vision. When picking the workobject, the robot will move its TCP to the location of the pick point so that the origins and X-axes of these two points coincide with each other, while the directions of their Y-axes and Z-axes are opposite.



Among them, ① and ② are the pick points of shafts, while ③ is the pick point of ring center.

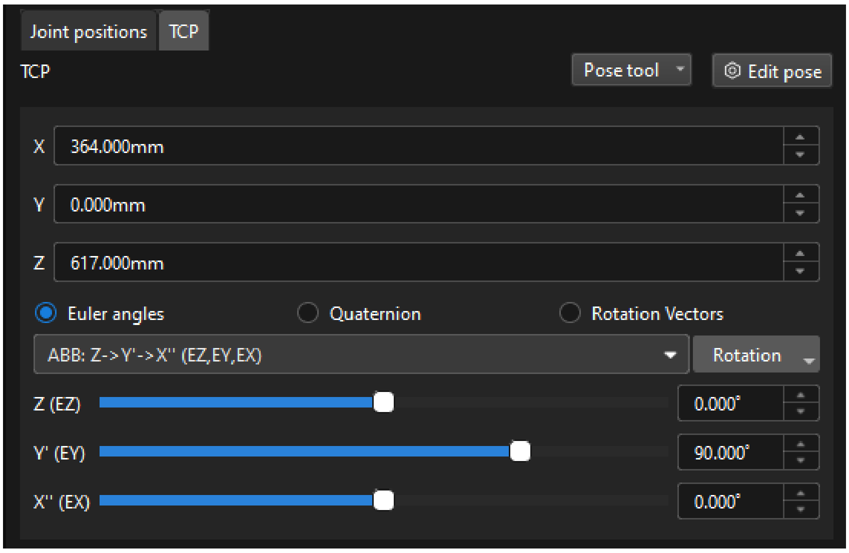
Robot Pose

The robot pose usually refers to the position and orientation of the robot in the 3D space. It is expressed in the way of tool pose or joint positions.

Tool Pose

The tool pose describes the position of the tool reference frame relative to the robot base reference frame.

The tool pose determines the position (x, y, z) and orientation (usually represented by Euler angles, rotation matrix, and quaternions) of the tool. It focuses on the positional relationship between the tool and the working space and describes the position and orientation of the tool, such as robotic claw, and welding torch, when the robot performs a task.



TCP

X 364.000mm

Y 0.000mm

Z 617.000mm

● Euler angles ○ Quaternion ○ Rotation Vectors

ABB: Z->Y'->X'' (EZ,EY,EX)

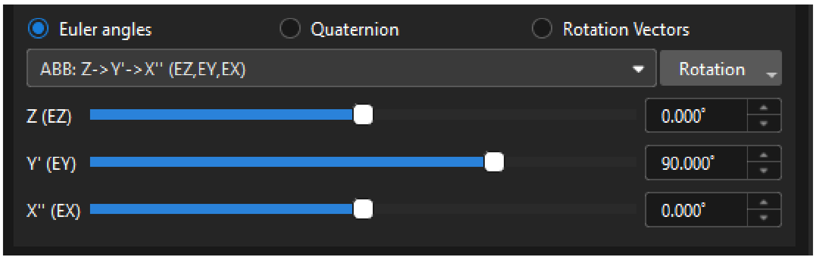
Z (EZ) 0.000°

Y' (EY) 90.000°

X'' (EX) 0.000°

- Euler Angles

  Euler Angels are used to describe the rotation of the object in the 3D space.



● Euler angles ○ Quaternion ○ Rotation Vectors

ABB: Z->Y'->X'' (EZ,EY,EX)

Z (EZ) 0.000°

Y' (EY) 90.000°

X'' (EX) 0.000°

It consists of three angles that correspond to the rotation of each of the three axes. These three axes usually rotate around the X, Y and Z-axis of the object's fixed reference frame respectively. The same orientation can be expressed in different ways using multiple Euler angles. To avoid ambiguity, a consistent definition of Euler angles is applied in the robots from different brands. For example:

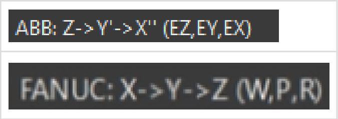
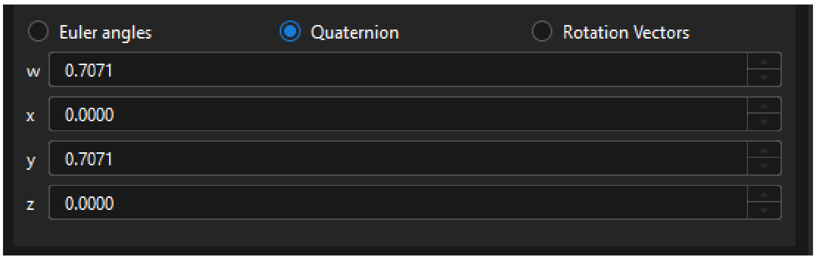


ABB: Z->Y'->X'' (EZ,EY,EX)

FANUC: X->Y->Z (W,P,R)

- Quaternions

  To avoid the gimbal lock problem of using Euler angles, some robot manufacturers use quaternions to represent the orientations in the space. A quaternion roughly refers to using three numbers to define the a spatial rotation axis and using the fourth number to define the rotation angle. The tool reference frame can reach the target pose from its initial state by rotating the angle around the spatial rotation axis.



○ Euler angles ● Quaternion ○ Rotation Vectors

w 0.7071

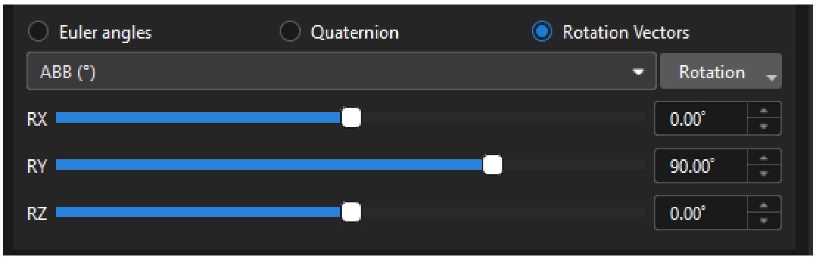
x 0.0000

y 0.7071

z 0.0000

- Rotation Vector

  The rotation vector, also known as the axial angle, is used to describe the rotation state of the tool. Currently, it is only used by UR robots.



○ Euler angles ○ Quaternion ● Rotation Vectors

ABB (°)

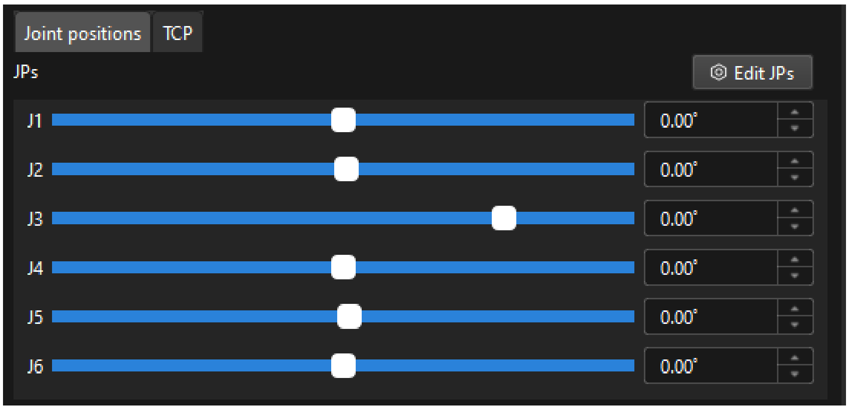
RX 0.00°

RY 90.00°

RZ 0.00°

Joint Positions

Joint positions describe the angle or position of each joint of the robot. This description can represent the state of each joint in line with the hardware structure of the robot.



J1 0.00°

J2 0.00°

J3 0.00°

J4 0.00°

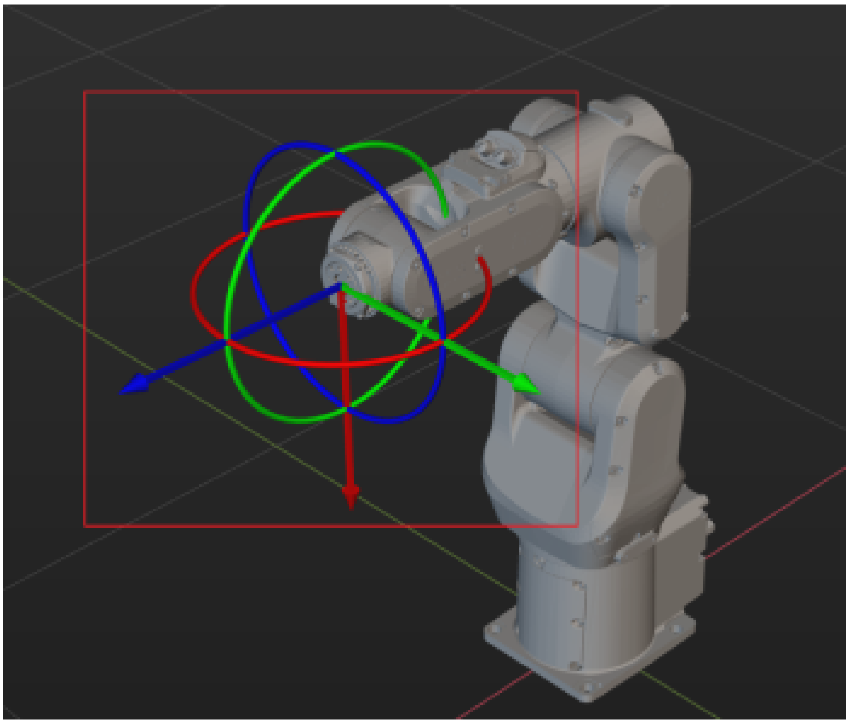
J5 0.00°

J6 0.00°

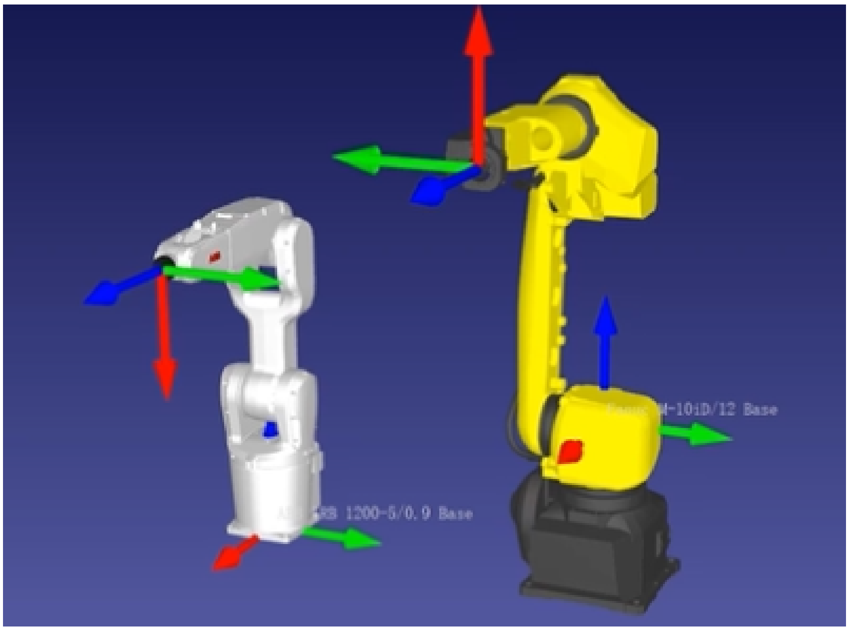
The joint positions are mapped to the tool pose by forward kinematics computation, while the tool pose is mapped to the joint positions by inverse kinematic computation.

Flange Pose

The flange pose is the pose of the robot flange frame relative to the robot reference frame, whose representation is the same as that of the tool pose. It can be seen as a special tool pose that the TCP is at the center of the flange without any rotation.



The default orientation of the flange reference frame defined by each robot manufacturer is different, resulting in different default angles for the flange pose. For example, the default X-axis orientation for ABB robots is downward, while that for FANUC robots is upward. In addition, most robots have a flange pose with the Z-axis perpendicular to the flange surface facing outward, but for the TURIN robots, it is the X-axis that is perpendicular to the flange surface facing outward.

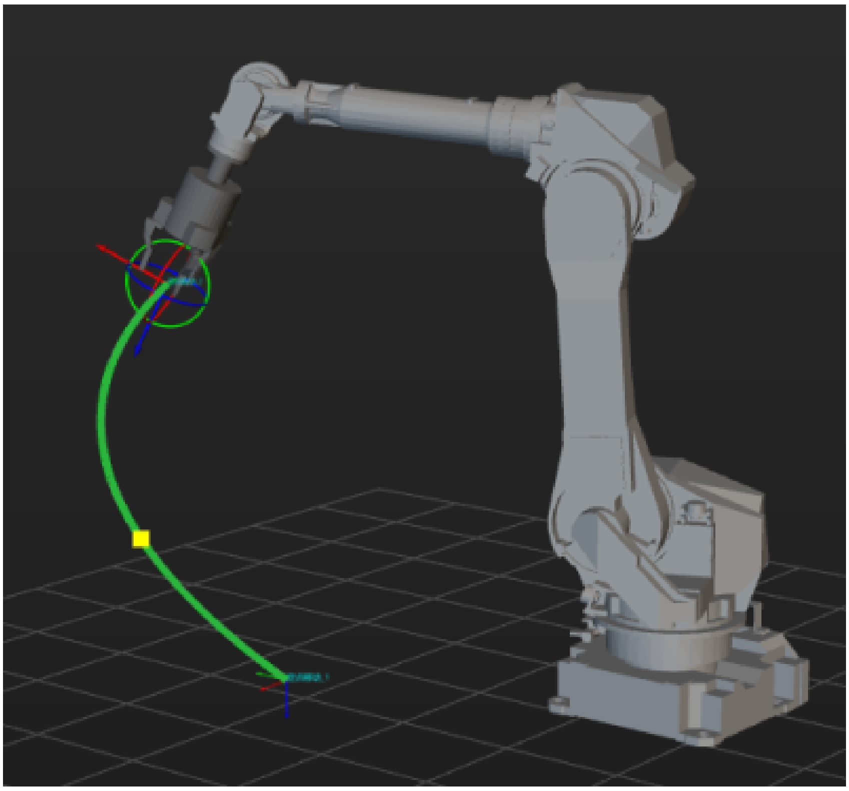


Motion Type

Robot motion usually refers to the process of moving the robot tool center point (TCP) from point A to point B in space. The joint motion and linear motion are two fundamental methods of robot motion.

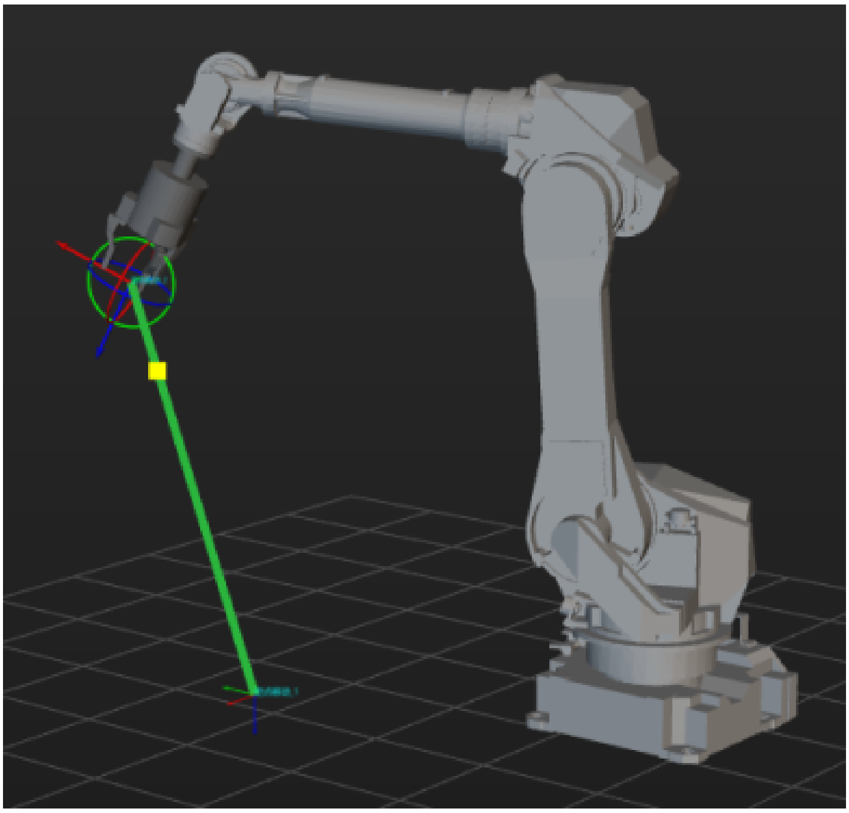
Joint Motion

Joint motion is the motion of a robot achieved by adjusting the angle or position of each joint. Each joint can rotate or move to get the entire robot tool to an expected position.



Linear Motion

The linear motion is the motion of the robot tool on a linear path. Unlike joint motion, linear motion is achieved by moving the robot tool along a linear path.



Rotational Symmetry and Picking Relaxation

Rotational Symmetry of the Tool

A rotational symmetric tool will coincide with itself after rotating around its axis of symmetry.

Please refer to Rotational Symmetry of the Tool for detailed description.

Rotational Symmetry of the Workobject

A rotational symmetric workobject will coincide with itself after rotating around its axis of symmetry.

Please refer to Rotational Symmetry of the Workobject for detailed description.

Workobject Picking Relaxation

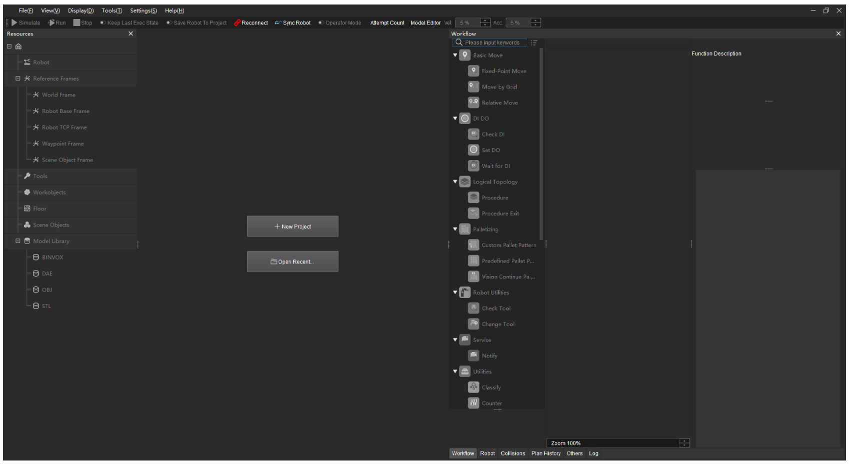
When the robot picks certain workobjects, the picking pose of the robot is allowed to be changed flexibly within a specific angle range, which is the picking relaxation.

Please refer to Workobject Picking Relaxation for detailed description.

5.1.2. User Interface

File

After Mech-Viz is open, the File page is displayed.



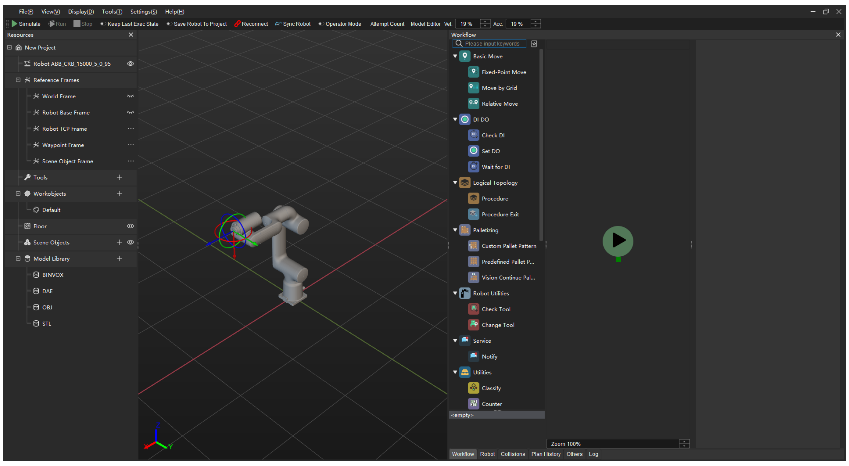
This interface consists of the following parts:

Option		Description	Shortcut
Home	New blank project	Create a new blank project.	Ctrl + N
	Recently used	Open a recently used project.	None
New	New blank project	Create a new blank project.	Ctrl + N
	Example projects	Built-in example projects. Please refer to Example Projects for how to use them.	None
Open		Select a project folder and open it.	Ctrl + O
Save		Save changes to the current project.	Ctrl + S
Save As		Save the project under a specified directory.	Ctrl + Shift + S
Save as JSON		Save the project, and save the VIZ file in the project folder as a JSON file.	None
Close Project		Close the current project.	Ctrl + W
Software Directory		Open the folder where the software is located.	None
Options		Open the options window.	None
Help		View the software support, version information, release note, and user manual.	None
Exit		Close Mech-Viz.	Ctrl + Q

- Click any option including File, View, Display, Tools, and Settings on the menu bar to exit from the File page.
- Mech-Viz 1.4.0 and above saves the project file in VIZ format. Previous versions save the project file in JSON format.

Home Software Interface

After creating a new blank project, select a robot model in the pop-up “Robot Model Library” window, and then enter the home software interface.



The main interface consists of 5 main sections, as listed in the table below.

1	Menu Bar	Includes File, View, Display, Tools, and Settings. Please refer to corresponding links for detailed information.
2	Toolbar	Used to simulate the project, control the real robot, adjust the robot speed, and so on. For more information, please refer to Toolbar.
3	Project Resource Tree	Used to manage the resources in the project, including tools, workobjects, scene objects, etc. For more information, please refer to Project Resources.
4	3D Simulation Area	The area where the robot motion path, collision detection result, vision pose, point cloud, etc. are displayed when the project is running or simulating. For more information, please refer to 3D Simulation Area.
5	Functional Panels	Include Workflow, Robot, Collisions, Plan history, Others, and Log. Please refer to the corresponding link to learn about the detailed information.

5.1.2.1. Menu Bar

This section introduces every option in the menu bar.

View

This menu is used to configure the display of software interface.

Option	Description
Resources	Display the project resource tree panel, selected by default.
Workflow	Display the Workflow panel, selected by default.
Robot	Display the Robot panel, selected by default.
Collisions	Display the Collisions panel, selected by default.
Plan history	Display the Plan history panel, selected by default.

Option	Description
Others	Display the Others panel, selected by default.
Log	Display the Log panel, selected by default.
Function Description	Display function description of selected Step in the Workflow tab.
Default Layout	Reset the layout of software interface to default.

Display

This menu is used to configure the display of the 3D simulation area.

Option	Description
Show Received Vision Poses	Display received vision poses, indices, and labels, selected by default.
Show Point Cloud	Display received point cloud, selected by default.
Show Picked Object	Display the model of the picked workobject, selected by default.
Show Collision While Planning	Display collisions detected during path planning, selected by default.
Show Octree	Display the octree structure of the object point cloud.
Show Received Carton Models	Display models of cartons in the 3D simulation area, selected by default.
Show Object Pose	Display the poses of picked workobjects in the 3D simulation area. Unselected by default.
Display Settings	More detailed display settings in the 3D simulation area.
Pose Status Colors	Pose status represented by the color code for vision pose indices and labels.

Tools

Option	Description
Robot Library Tools	Import robot model file; export current robot model to a designated directory.

Settings

Option	Description
Set Mech-Center Address	Set the IP address and ports of Mech-Center.
Lock/Unlock Project	When this option is selected, all modification to the project are prevented (you can still simulate and run the project).

5.1.2.2. Toolbar

This section introduces every option in the toolbar.

Option	Description
Simulate	Run the simulated robot with the vision service or vision records.
Run	Run a real robot.
Stop	Stop the execution of the current project.
Keep Last Exec State	Used for debugging in palletizing scenarios to allow a resumption of palletization from where robots left off.
Master-Control Robot	Establish connections with real robots.
Sync Robot	Synchronize the pose of the simulated robot with that of the real robot.
Operator Mode	Increase the efficiency of software operation if the operator mode is selected.
Attempt Count	The number of attempts per pose when picking
Model Editor	Open the model editor.
Vel.	Set the robot velocity.
Acc.	Set the robot acceleration.

> The efficiency of software operation can be increased mainly through the following two points in the operator mode:
>
> - Downsample the display of point clouds, and close the octree display to reduce the rendering load.
> - When the volume of collision with point cloud exceeds the threshold, skip the remaining calculations without recording it in the plan history.

5.1.2.3. Project Resource Tree

This section covers the project resource tree and the corresponding context menus.

Please refer to Project Resource Configuration to learn about how to configure various resources in the project.

Project

This entry provides the interfaces for the project name, project modification status, and the project autoload status.

Right-click the project name, and the following options will appear.

Option	Description
Open Project Folder	Open the folder where the project is stored.
Autoload Project	Select to automatically open the current project when you open the software next time.
Back Up Current Project	Back up the current project.

Vision services

This entry provides all vision services that can be called.

The list will be automatically updated every time you open the project, simulate, or run the project. You can also click ⟳ to update the list manually. By default, vision results from all vision services will be displayed. Right-click **Vision services** and you can check **Only display the vision result of the latest vision service**.

Vision records

The parent entry shows the name of the vision service, and the child entry shows the vision records.

The simulation result will be displayed in the 3D simulation area when the corresponding vision record is selected, and it will no longer be displayed after you deselect it.

Right-click each vision record, and the following options will appear in the context menu:

Option	Description
Number of Saved Records	Specify the maximum number of vision records that can be saved.
Open Vision Record Folder	Open the folder where the vision records are stored.
Delete File(s)	Delete the current vision record file.

Robot

This entry shows the robot model used in the current project.

Right-click the robot name, and the following option will appear.

Option	Description
Open Robot File Directory	Open the folder where the robot model file is stored.
Reload Robot	Usually used after the parameter file of the local robot model is modified. Click this option to update the robot model and its parameters in the software.

Click 👁 to make the displayed robot semitransparent, and click 👁 to restore to the original display setting.

Reference Frames

This entry provides the interface for displaying all reference frames in the 3D simulation area.

Click ⌣ to display the corresponding reference frame, and click 👁 to hide the reference frame. Click ... to enable or disable the "translation" or "rotation" of the manipulator.

Tools

This entry provides the interface for configuring the tools in the project. Please refer to Tools for how to configure the tools.

Click **[+]** to open the Tool Configuration window, where you can add tools.

The default tool is used to avoid errors due to unconfigured end tools.

Right-click the name of a configured tool, and the following options will appear.

Option	Description	Shortcut
Copy	Copy the current tool.	Ctrl + C
Paste	Paste the current tool.	Ctrl + V
Delete	Delete the current tool.	Delete
Rename	Rename the current tool.	None
Tool Configuration	Open the Tool Configuration window of the current tool.	None
Set As Active Tool	Set the current tool as the active tool.	None

Click 👁 to make the tool model semitransparent, and click 👁 to restore to the original display setting.

Workobjects

This entry provides the interface for configuring the workobjects in the project. Please refer to Workobjects for how to configure the workobjects.

Click **[+]** to open a Workobject Configuration window, where you can add workobjects.

“Default workobject” is the workobject configuration used for vision results without labels.

Right-click the name of a configured workobject, and the following options will appear.

Option	Description	Shortcut
Delete	Delete the current workobject.	Delete
Rename	Rename the current workobject.	None
Workobject Configuration	Open the Workobject Configuration window of the current workobject.	None

Floor

This entry provides the interface for modifying the height of the floor and floor display setting.

Right-click the Floor and you can modify the height of it in the pop-up window.

Click 👁 to hide the floor and click 👁 to restore.

Scene Objects

This entry provides the interface for configuring the scene objects in the project. Please refer to Scene Objects for how to configure the scene objects.

Click **[+]** to open a Scene Object Configuration window, where you can add scene objects.

Right-click the name of a configured scene object, and the following options will appear.

Option	Description	Shortcut
Copy	Copy the current scene object.	Ctrl + C
Paste	Paste the scene object.	Ctrl + V
Cut	Cut the current scene object.	Ctrl + X
Delete	Delete the current scene object.	Delete
Rename	Rename the current scene object.	None
Scene Object Configuration	Open the Scene Object Configuration window of the current workobject.	None

Click 👁 to make the scene object semitransparent; click 👁 to make the scene object transparent; and 👁 to restore to the original display setting.

You can also drag and drop the scene object name in the Resources panel to make the scene object model a child model or an individual model.

Model Library

This entry provides the interface for importing custom models to the project. All the custom models imported will be displayed in the Model Library, and the models will be categorized according to their formats.

Click **[+]** to open the file selection window to import an custom model.

Right-click the imported custom model, and the following options will appear.

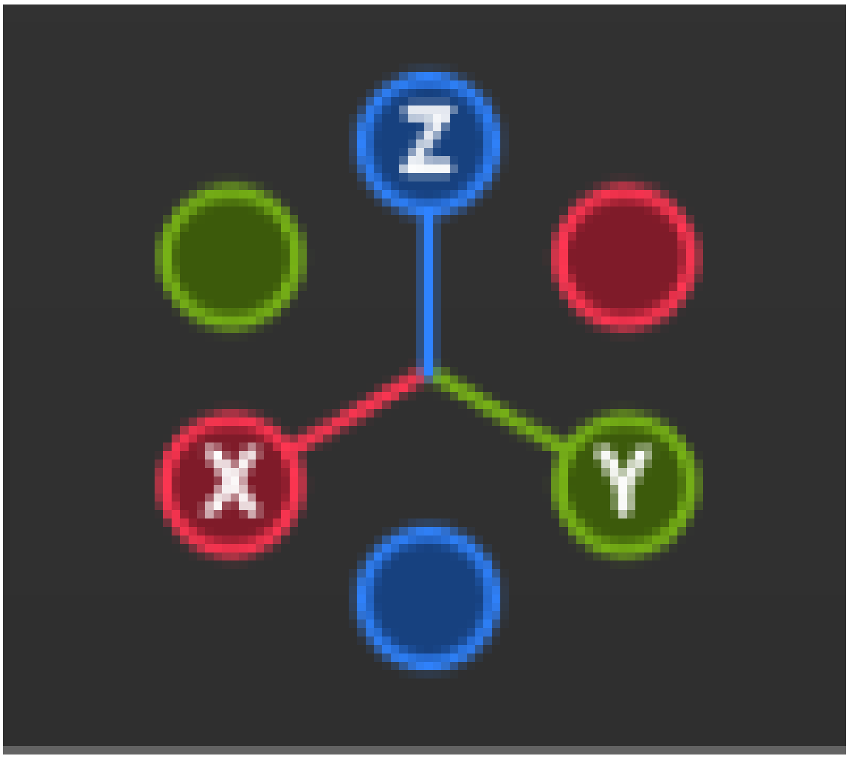
Option	Description	Shortcut
Copy	Copy the current model.	Ctrl + C
Paste	Paste the copied model.	Ctrl + V
Delete	Delete the current model.	Delete
Model Configuration	Open the Model Configuration window of the current model.	None
Model Editor	Open the model editor and the current model will be used as the reference model.	None
Open Folder	Open the folder where the model file is stored.	None

5.1.2.4. 3D Simulation Area

The robot, scene objects, tools, workobject models, robot motion path, pick points, and collisions will be displayed in the 3D Simulation Area. Also in this area, you can drag and drop the models of scene objects.

Mouse Actions

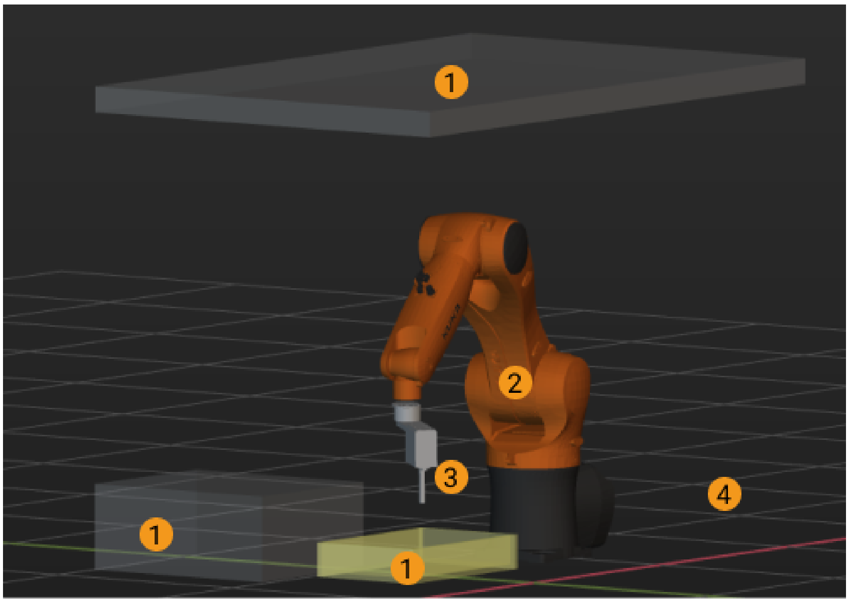
The navigation gizmo shows the current view in the workspace. Click the icon of each axis (X, Y, Z, -X, -Y, -Z) to align the view with the axis. Press and hold the left mouse button and drag the navigation gizmo to rotate the view.



Rotate the view	Press and hold the **scroll wheel** and drag in any direction, or press and hold the **left mouse button** and drag the navigation gizmo in any direction.
Pan the view	Press and hold ✋ and drag in any direction, or press and hold **Shift + scroll wheel** and drag in any direction.
Zoom in/out	Press and hold 🔍 to zoom in by dragging upwards and zoom out by dragging downwards, or **scroll the mouse wheel**.
Fit the view	Click ⛶.
Switch to perspective/orthographic view	Click ◭.
Show pressed keyboard keys and mouse buttons	Once ⌨ is selected (highlighted in blue), the keyboard keys and mouse buttons you pressed will be displayed in the lower right corner in the 3D Editing Workspace.

Display Scene Objects

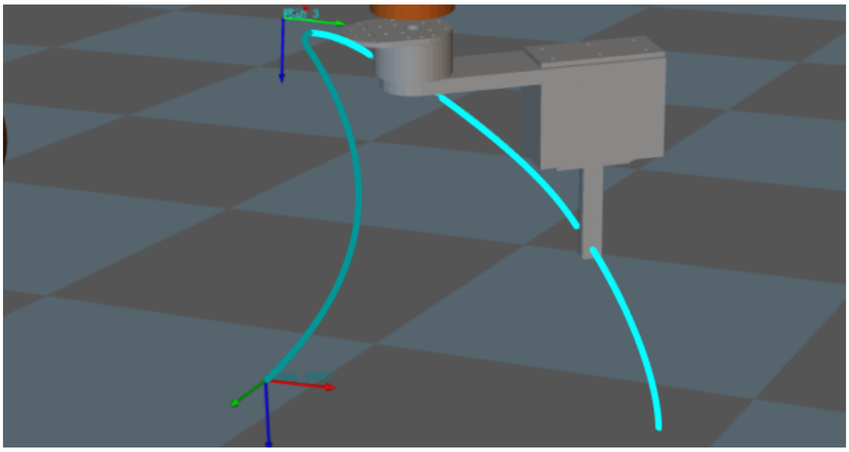
The 3D simulation area displays the following models in a project:



① Scene object model, ② Robot model, ③ Tool model, ④ Ground

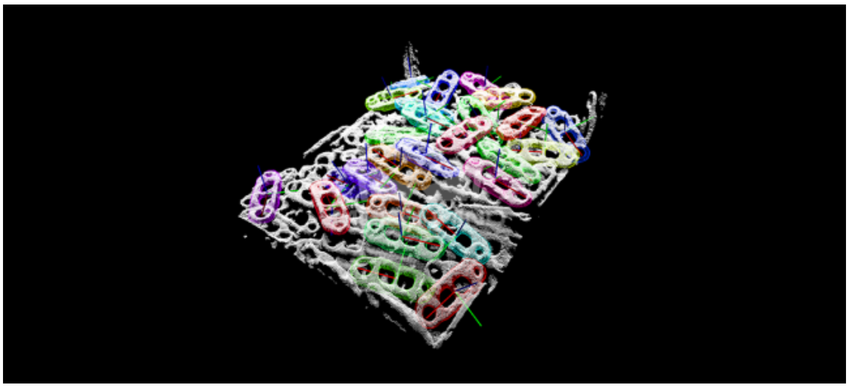
Display Planned Path

Whether for a real scenario or a simulated one, the 3D simulation area can display the planned path of the robot. You can use the displayed path as an aid for adjusting the motion of the robot.



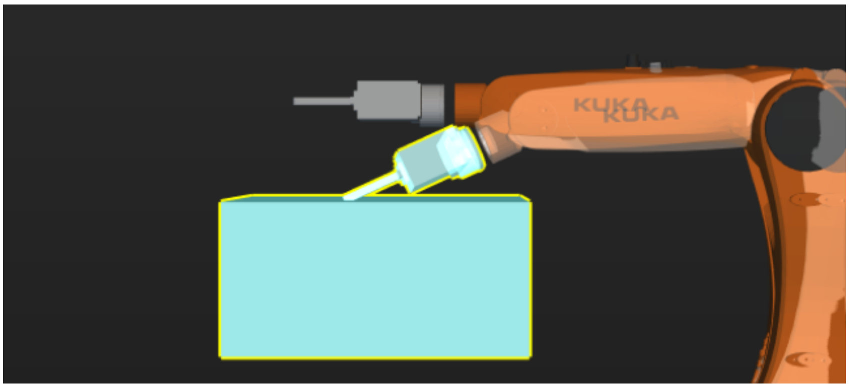
Display Received Vision Result

The vision result returned by the vision service will be displayed in the 3D simulation area. Generally speaking, a vision result includes: workobject point clouds, scene point clouds, workobject poses, workobject ID, and customized pose labels that indicate the characteristics of the workpieces.



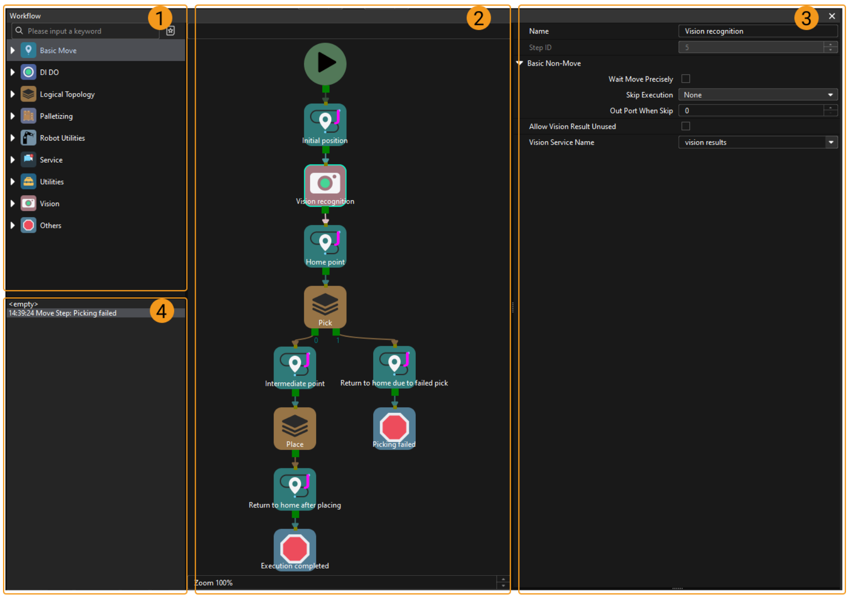
Display Detected Collisions

When planning a path for the robot, Mech-Viz detects whether the robot and/or its tool will collide with scene objects, workpieces, or workobject containers. When a possible collision is detected, the two objects involved in the collision will be displayed with blue highlight and yellow outlines in the 3D simulation area, as shown below.



5.1.2.5. Interface of Workflow

This functional panel consists of ① Step library, ② graphical programming workspace, ③ parameter panel, and ④ programming record area, as shown in the figure below.



<empty>
14:39:24 Move Step: Picking failed

Name: Vision recognition
Step ID: 5
Basic Non-Move
Wait Move Precisely ☐
Skip Execution: None
Out Port When Skip: 0
Allow Vision Result Unused ☐
Vision Service Name: vision results

Please refer to Workflow for detailed instructions on how to build a workflow with Steps.

Step Library

A Step is a graphical programming module used to enable a specific function. Steps are the fundamental modules of robot programming. All Steps are divided into various Step categories according to their functions. For example, “Basic Move” includes Steps related to basic motions of the robot, such as Fixed-Point Move, Relative Move, and Move by Grid. You can click the Step category to expand or collapse all Steps.

Steps can be categorized as “move-type Steps” and “non-move Steps”.

- Move-type Steps: All Steps that belong to “Basic Move” and “Palletizing” categories and the Step Vision Move.
- Non-move Steps: Steps that belong to DI DO, Logical Topology, Robot Utilities, Service, Utilities, Vision (except Vision Move), and Others

You can adjust the display settings of the Step Library.

- Right-click and select **Show all** in the context menu to expand all Step categories.
- Right-click and select **Collapse** in the context menu to collapse all Step categories.

In addition, you can enter a keyword in the search box above the Step Library to search the Step you need.

Graphical Programming Workspace

This area is used for graphical programming. In the process of graphical programming, you can drag all kinds of steps from the step library, each representing a specific function or operation. To define the logic and flow of the program, you need to set the parameters of these steps and then connect the steps.

Right-click on the blank area in the graphical programming workspace, and you can see the following options.

Option	Description	Shortcut
Navigate Up	Navigate to the previous level of the Procedure	PgUp
Navigate Down	Navigate to the next level of the Procedure (only effective when you right-click on the Procedure)	PgDown
Copy	Copy the selected Step	Ctrl + C
Cut	Cut the selected Step	Ctrl + X
Paste	Paste the copied Step	Ctrl + V
Delete	Delete the selected project	Delete
Select All	Select all Steps in the graphical programming workspace	Ctrl + A
Save Screenshot	Save the screenshot of the workflow to a specified path	None
Format	Automatically format all Steps in the workflow	Alt + Shift + F
Pack as Procedure	Pack multiple Steps as a Procedure	Ctrl + G

Unpack Procedure	Unpack the selected Procedure	Ctrl + Shift + G
Import or Export Procedure	Import the selected Procedure to another Procedure, or export the selected Procedure	None
Custom Description	Add a custom description to the selected Step	None
Reset DO	Reset the DO to 0, which is usually used to turn off the vacuum gripper	None

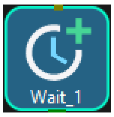
To search for Steps in the workflow, click the blank area in the graphical programming workspace and press `Ctrl` + `F`.

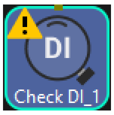
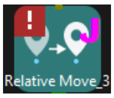
In addition, you can press and hold the `ctrl` key and scroll the mouse to zoom in or out.

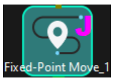
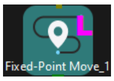
Step Mark

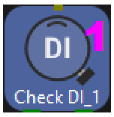
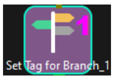
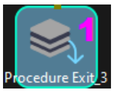
Some step marks and their meanings are listed in the following table.

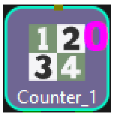
The marks are located on the corners of the step icons, and maybe change depending on the parameter settings, project running situation, and so on.

	This mark represents the step that is currently being planned. **"Planning" refers to the process when the software attempts to find a collision-free path that the robot can follow to reach the target position.** **The planning of the workflow is completed before actually running the project.**

	This mark indicates that there is a warning in the current step; hover the mouse cursor over the warning mark to view the specific warning message.
	This mark indicates that there is an error in the current step; hover the mouse cursor over the error mark to view the specific error message.

	J mark represents that the current motion type is "Joint motion."
	L mark represents that the current motion type is "Linear motion."

	This mark represents the number of DI port checked in the "Check DI" Step.
	This mark represents the "Tag value" set in the "Set Tag for Branch" Step.
	This mark represents the port number of exiting the procedure.

	This mark represents the current count of the counter.

Parameter Panel

Select a Step in the workflow and all its parameters will be displayed in the parameter panel. If you check Function Description in the View menu, a brief description of the selected Step will also be displayed at the top of the parameter panel.

Programming Record Area

All your actions, including adding a Step, deleting a Step, adding a connection, and deleting a connection will be recorded in the programming record area.

Click on a record to return to the workflow status at that time.

5.1.2.6. Robot

In this functional panel, you can change the robot model, set soft limits, move the real robot, adjust the robot pose, set robot home position, etc. Please refer to Robot Configuration for detailed information.

5.1.2.7. Collisions

In this functional panel, you can configure the collision detection.

Collision Detection Configuration: Click **[Collision detection configuration]** in the Collisions tab to open the Collision Detection Configuration window. You can configure the detection of collisions between the following objects such as point cloud, tool, robot links, and held workobjects. Please refer to Collisions for detailed information.

Collision Detection Table: The collision detection table shows the configuration of all collision detections. Click **[legend]** to see the meaning of the different colors in the table.

5.1.2.8. Plan History

On this panel, you can view the plan history of Mech-Viz in detail. You can also use the plan history to troubleshoot problems, and optimize projects. For details, please refer to Plan History.

You can configure the related settings at the bottom of this functional panel.

Option	Description
Animation duration	When the plan history record is selected, 3D simulation area will display the duration of path animation.
Expire after	The expired time of the plan history. After the expired time, the previous plan history will be overwritten by the current plan history.
Load plan history	Load the saved plan history.
🧹	Clear all plan history in the panel.
?	A brief introduction of the plan history.

Right-click on an item in the plan history, you can see the following options.

Option	Description	Shortcut
Copy	Copy the selected content.	Ctrl + C
Search	After clicking **[Search]**, you can see two filter buttons at the bottom of the panel. You can search for the planning item you need according to the plan result type and the collided link index.	Ctrl+F
Collapse All	Collapse all sub-nodes in the panel and only displays the parent nodes.	N/A
Delete Selected Plan History	Delete the selected plan history item.	N/A

5.1.2.9. Others

This functional panel contains execution settings, Step collecting settings, Singularity detection, and Global JPs constraints.

Others

- Skip waypoints approximate to others

  When two waypoints are close to each other, only one waypoint is sent to the robot.

- Only use default workobject configuration

  After this option is enabled, the default workobject configurations will be used even when an object label is received.

- Robot service timeout

  The timeout period for communication between Mech-Viz and the robot. This feature does not work for move-type Steps and the “Transfer Control” Step. The default timeout period is 1000 ms.

Singularity detection

Purpose of Singularity Detection

If the robot reaches the point of a singularity, it may not be able to execute certain linear motion commands, causing the robot’s alarm to activate and leading to a halt in production. The singularity detection function can provide early warnings and attempt to avoid singularities in the path planning process, ensuring smoother robot motion and reducing the occurrence of system alarms and anomalies, which helps maintain production continuity.

Singularity Detection Method

Once the Max velocity, Reduction ratio, and range of the JPs are specified under **Singularity detection** in the **Others** functional panel of Mech-Viz, the software can detect singularities automatically. There are four options for the Detection method.

1. Do not detect
2. Detect velocity of each joint
3. Detect angle of specified joint
4. Detect velocity of each joint and angle of specified joint

> The singularity detection only takes effect for move-type Steps in which the motion type is set to **Linear move**.

Do not detect

- Description

  Do not enable singularity detection in the software.

- Instruction

  Select **Do not detect** in the drop-down list of the **Detection method** parameter.

Detect velocity of each joint

- Description

  When the detected velocity of any joint exceeds the specified **Max velocity**, the velocity set in the move-type Step will be reduced according to the **Reduction ratio**, and the singularity detection will be performed again. If the reduced velocity is **less than** the set **Max velocity**, the robot will move at the reduced velocity in the corresponding Step. If the reduced velocity still **exceeds** the set **Max velocity**, the software will determine that a singularity will occur when the robot moves, and the robot cannot reach the desired position.

  > Reduced velocity = Original velocity × Reduction ratio

- Instruction

  1. Select **Detect velocity of each joint** in the drop-down list of the **Detection method** parameter.
  2. Set **Max velocity** and **Reduction ratio**.

Detect angle of specified joint

- Description

  When the robot is moving linearly, Mech-Viz will detect whether the angle of the specified joint falls within the set range specified by the Angle lower limit and Angle upper limit. If the joint angle falls within the set range, the software will determine that a singularity will occur when the robot moves, and the robot cannot reach the desired position.

- Instruction

  1. Select **Detect angle of specified joint** in the drop-down list of the **Detection method** parameter.
  2. Specify the joint to be detected.
  3. Set the **Angle lower limit** and **Angle upper limit**.

Detect velocity of each joint and angle of specified joint

- Description

  Detect whether the angular velocity of each joint exceeds the set “Max velocity” and whether the angle of the specified joint falls within the set angle range in a linear movement at the same time. Please refer to the description for the “Detect velocity of each joint” and “Detect angle of specified joint” above.

  When the robot does not meet any of the requirements of the above two options, the software will determine that a singularity will occur when the robot moves, and the robot cannot reach the desired position.

- Instruction

  1. Select **Detect velocity of each joint and angle of specified joint** in the drop-down list of the **Detection method** parameter.
  2. Refer to the description for the “Detect velocity of each joint” and “Detect angle of specified joint” options above, and set **Max velocity**, **Reduction ratio**, **Select joint**, **Angle lower limit**, and **Angle upper limit**.

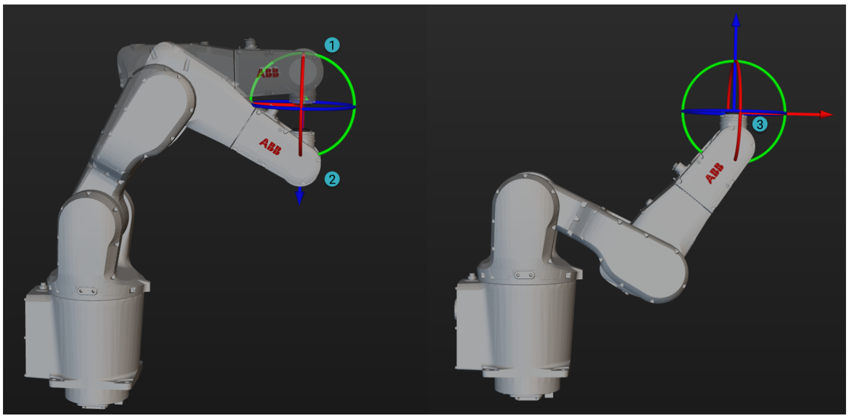
Global JPs Constraints

> For move-type Steps, if the JPs Constraint parameters are set to Auto, the settings of Mech-Viz will take effect. Otherwise, the settings in the move-type Steps will take effect.

- No shoulder flip/No elbow flip/No wrist Flip

  You can choose to reduce unnecessary rotation of the robot **While Holding** objects or during the **Entire Execution**.

However, setting the shoulder/elbow/wrist all to “no rotation” is not necessarily the best choice. An example is shown in the figure below.



When the robot needs to move from position 1 to position 2 and “No Joint Flip” is set, unnecessary motion is required to reach that point, and the robot may end in position 3. In this case, only No Shoulder Flip and No Elbow Flip need to be set.

5.1.2.10. Log

On the log panel, you can view the software log in detail and manage the log settings at the bottom of the panel. Click one log entry and press `Ctrl` + `C`, and the entry will be copied to your clipboard.

Option	Description
D: Debug	Screen out the logs to be displayed according to their levels. Logs under selected levels will be displayed. You can select multiple levels.
i: Info	
W: Warning	
E: Error	
Clear	Clear all log messages on the page.
Export	Export the log messages to the HTML file. The exported files are located in the logs folder in the Mech-Viz installation directory.
Open folder	Click to open the logs folder in Mech-Viz’s installation directory. The logs folder contains log files recorded by date.

5.1.3. Getting Started

Example Projects

A variety of built-in example projects are provided in Mech-Viz to help you get familiar with the software quickly. These projects cover common usage scenarios, and you can run the example projects to see the operation effect.

Taking **Basic Pick and Place** project under the category of **Simple handling** for example, this section will describe how to open and run the example project (The same operation method goes to other projects).

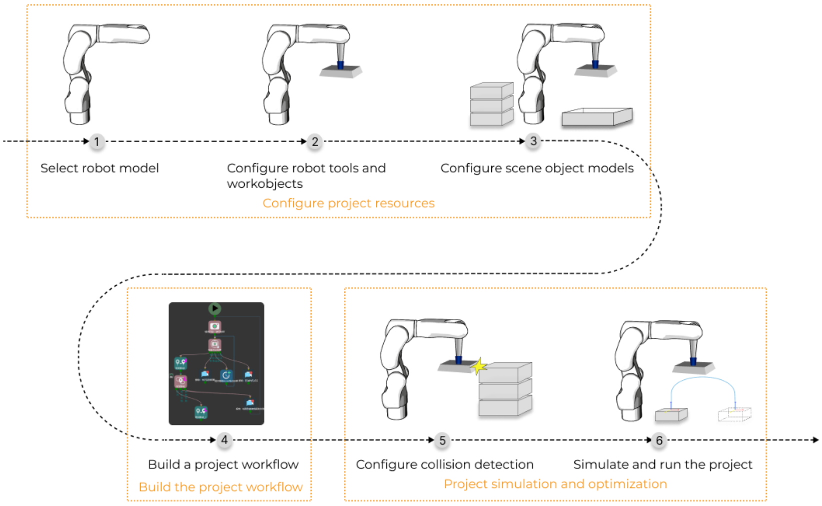
1. Go to File>New>Example projects, click the **Basic Pick and Place** project card, and then click **[Choose Project]** in the project description panel to open the project.
2. Click **[Simulate]** in the upper-left corner of the interface.
3. Click **[Start simulation]** in the pop-up **Select Vision Record to Use** window.

- Click the **[Pause]** button to pause simulation.
- Click the **[Stop]** button to stop simulation.
- To change the example project, select other example projects in the File>New>Example projects when the project in use stops.
- To reduce the speed of robot, modify **Vel.** and **Acc.**on the toolbar when the project in use stops.

Create a Project

The process of building and configuring a Mech-Viz project usually consists of the following steps:

1. Click File>New>**[New blank project]**.
2. Select a robot model in the Robot Model Library.
3. Configure the resources, including tool, workobject, scene, etc., in the project.
4. Build a workflow, i.e. robot control logic.
5. Simulate and improve the project.


1
Select robot model
2
Configure robot tools and workobjects
3
Configure scene object models
Configure project resources
4
Build a project workflow
Build the project workflow
5
Configure collision detection
6
Simulate and run the project
Project simulation and optimization

5.2. Resources

Resources include the robot, tools, workobjects, and scene objects, which are fundamental to a project.

Configuration of the Resources

Please read the following sections to learn about how to configure the resources in the project.

- Robot
- Tools
- Workobjects
- Floor
- Scene objects

By right-clicking the project name, you can open the project folder, set the project to autoload, and backup the current project.

Note of the Models

If you want to import models from outside the software for the project resources (robot, tools, workobject, scene object, etc), you should import both the **visualization model** and **collision model**.

Model type	Description
Visualization model	Only used for visualization in the 3D simulation area, and will not be used for collision detection.
Collision model	Only used for collision detection, and will not be displayed in the 3D simulation area.

The table below shows the supported model formats for different project resources.

		STL	OBJ	DAE	Binvox
Robot	Visualization model	√	√	√	×
	Collision model	√	√	×	×
Scene objects	Visualization model	√	√	√	×
	Collision model	√	√	×	×
Tools	Visualization model	√	√	√	×
	Collision model	×	√	×	×
Workobjects	Visualization model	√	×	×	×
	Collision model	×	×	×	√

5.2.1. Robot

In the Robot panel, you can change the robot model, set soft limits, adjust the pose of the simulated robot, set the robot home position, and move the real robot.

Change Robot Model

If you have already selected a robot model in the **Welcome interface** and want to change the robot model, please refer to Robot Model Library.

Set Soft Limits

Soft limits restricts the movement range of robot axes. A collision may occur if other devices are within the movement range of an axis of the robot. A non-standard gripper or robot dress packs may restrict the robot’s movement as well. In order to avoid collisions and unreachable poses in the path, you can set soft limits beforehand.

1. Click **[Set Soft Limits]**.
2. Modify the lower limit and upper limit of each axis according to the actual requirement.
3. Click **[OK]**.

The soft limits set in Mech-Viz will only take effect when Mech-Viz is used. It will not affect the soft limits set on the teach pendant.

Adjust the Pose of the Simulated Robot

Follow either of the steps below to adjust the joint positions of the simulated robot.

- Click the Joint Positions tab, and drag the slider or enter a value in the box on the right to move the simulated robot in real time.
- Click the **[Edit JPs]** button, enter the value in the text box and click **[OK]**.

Follow either of the steps below to adjust the TCP of the simulated robot.

- Click the TCP Pose tab, and modify the values in Euler angles or quaternions to move the simulated robot in real time.
- Click the **[Edit pose]** tab, enter the value in the text box and click **[OK]**.
- If a real robot has been connected, you can click **[Update TCP Pose]** to synchronize the TCP from the real robot.

Set Robot Home Position

A home position is the original pose of the robot set by the user.

1. Please refer to Adjust the Pose of the Simulated Robot to adjust the robot pose.
2. Click **[Move simulated robot to home position]**.

> When the simulated robot is in a different position other than the home position, you can click **[Move simulated robot to home position]** to adjust the robot pose quickly.

Move Real Robot

Once a real robot is connected, Move Real Robot can move the real robot to the current pose of the simulated robot.

1. Please refer to Adjust the Pose of the Simulated Robot to adjust the pose of the robot model.
2. Adjust the **Speed** to 5% or lower.
3. Click **[Move real robot]**.

5.2.2. Tools

This section introduces tools and corresponding configurations.

Introduction

Tools, such as grippers and suction cups, are mechanical devices attached to the end of the robotic arm to perform various tasks.

Model of the Tool

In order to visualize the tool in the 3D simulation area and make the tool involve the collision detection, you need to create a visualization model and a collision model of the tool and import them to the model library.

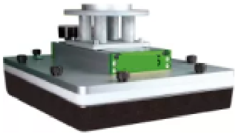
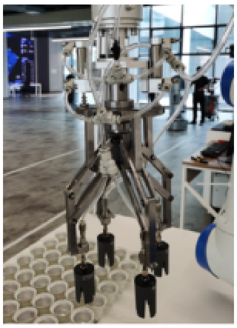
The following table shows the formats of the supported visualization model and collision model of the tool.

Format	STL	OBJ	DAE	Binvox
Visualization Model	√	√	√	×
Collision Model	×	√	×	×

- The collision models of tools used in the software should be OBJ models that are composed entirely of convex hulls. You can use the Model Editor to create convex hulls on the original tool model and convert the model file to OBJ format.
- The visualization model can only be used to visualize the tool in the 3D simulation area and cannot be used in collision detection; while the collision model can only be used in collision detection and cannot be displayed in the 3D simulation area.

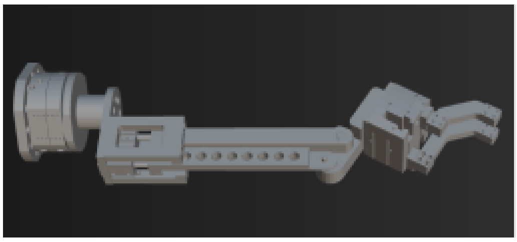
Type of the Tool

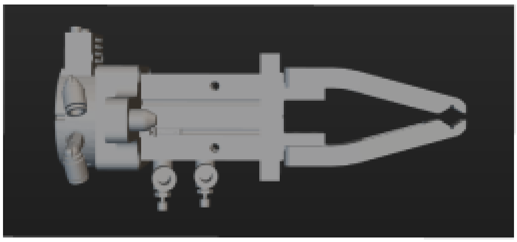
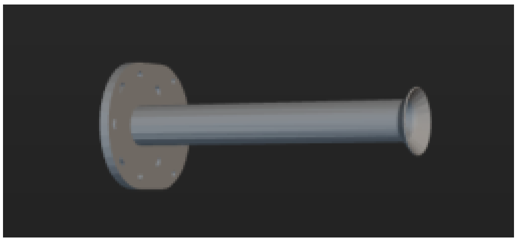
Mech-Viz supports the regular tool, depalletizing vacuum gripper, and array gripper.

Type of the Tool	Description	Illustration
Regular Tool	All tools except suction cups used for depalletizing and array grippers. For example, the pneumatic gripper and single suction cup.	
Depalletizing Vacuum Gripper	Rectangular suction cups used for depalletizing. Multi-block suction cups are supported.	
Array Gripper	An array of tools at the end of the robot that can be used to perform pick-or-place tasks concertedly.	

Rotational Symmetry

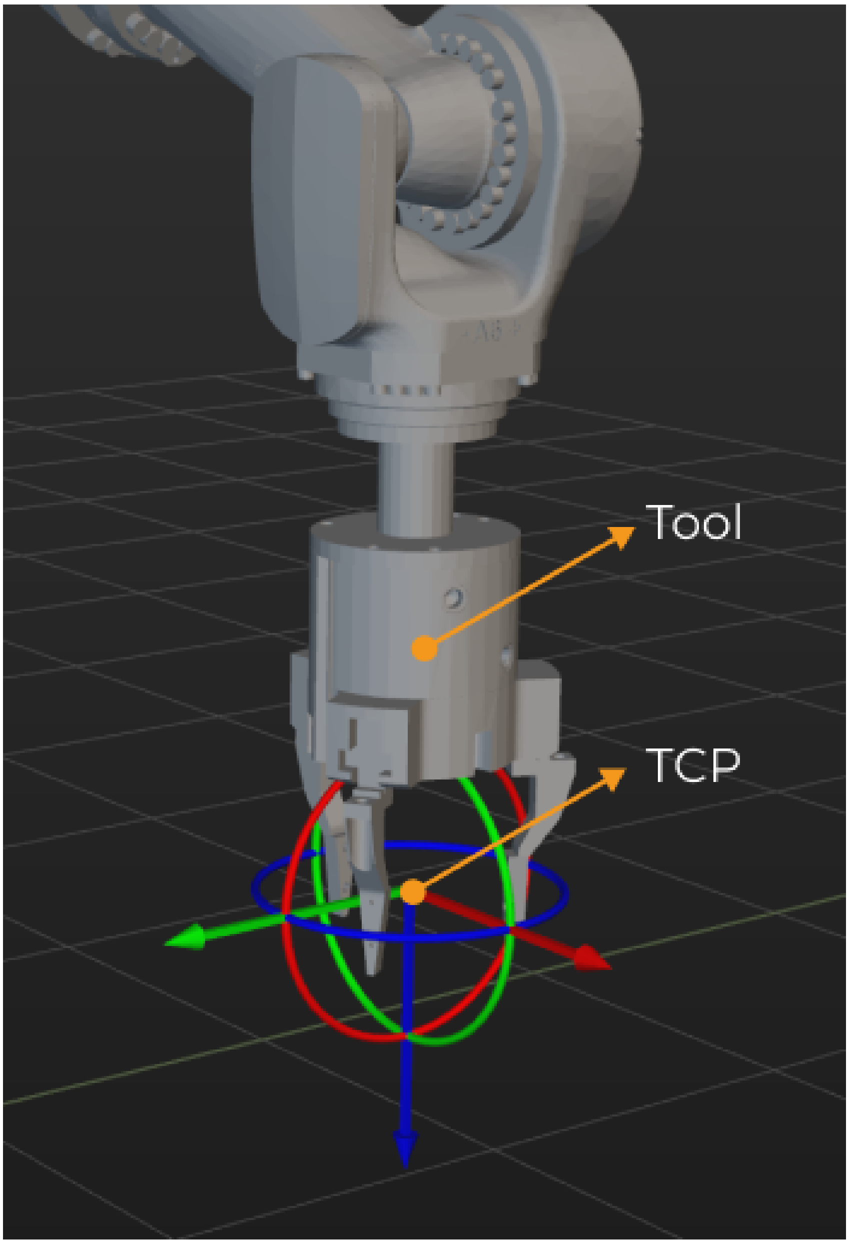
A rotational symmetric tool will coincide with itself after rotating around its axis of symmetry. Configuring the symmetry of the tool can prevent the end of the robot from rotating meaninglessly when picking and placing, and therefore increase the picking success rate and reduce the planning time consumption. Therefore, the robot can move more smoothly and swiftly.

No Rotational Symmetry	

180° Rotational Symmetry	
Rotational Symmetry at Any Angle (Circular)	

Tool Center Point (TCP)

The tool is used to pick and place the object. When you move the robot to a specified point to perform pick-or-place task, you are actually moving the TCP (Tool Center Point) to that point. The default TCP is located at the end of the robot, but sometimes you need to adjust the TCP to ensure that the TCP is at the right place for the actual pick-and-place process.



Configure the Tool

Import the Collision Model and Visualization Model

Click the [+] next to Project resource tree › Model library, and select the collision model and visualization model in the pop-up window, and then click [**Open**].

Add the Tool

Go to Project resource tree › Tools and click [+] to open the Tool Configuration window.

1. Enter a **Tool name**.
2. Select the **Tool type** according to the actual situation.
3. Select the **Collision model** in the drop-down list.
4. Select the **Visualization model** for the visualization in the 3D simulation area. When the

position or size of the visualization model is not correct, please refer to Adjust the Position or Size of the Tool Model to adjust the model.

5. Set the **Rotational Symmetry** to **None**, **N-fold**, or **Circular** according to the actual situation. If you select **N-fold**, also set the **Number of folds** parameter.
6. Follow either of the methods below to set the TCP.

Update TCP from robot	Click the button to synchronize the TCP of real robot to the software.
Perform TCP calibration	Control the robot to rotate around a fixed point and record multiple sets of flange poses to calculate the TCP.
Modify the TCP parameter	Modify the Euler angles or quaternion directly.
Use precise TCP data	If you have already obtained the accurate TCP data by other methods, you can click Edit pose and paste the data in the pop-up window.

7. If you are using a regular tool, you do not need to configure the control logic. If you are using a vacuum gripper or array gripper, please refer to Configure the Control Logic of the Tool to continue to configure the control logic.
8. Click **[OK]** in the end.

If you need to add more tools, please repeat the above steps.

Delete the Tool

Follow either of the following steps to delete the tool.

- Go to Project resource tree › Tools, select the tool name and press the `Delete` key.
- Go to Project resource tree › Tools, right-click the tool name and select 删除 in the context menu.

Modify the Tool

1. Follow either of the following steps to open the Tool Configuration window.
   - Go to Project resource tree › Tools, double-click the tool name.
   - Go to Project resource tree › Tools, right-click the tool name and select **[Tool Configuration]** in the context menu.
2. Modify the parameters according to the actual requirement.
3. Click **[OK]**.

Set as Active Tool

If you only add one tool, the current tool is the active tool. If you add multiple tools, the first added tool will be used as the active tool by default.

To set a tool as active tool, go to Project resource tree › Tools, right-click the tool you want to use and select **[Set as Active Tool]**.

Adjust the Position or Size of the Tool Model

Follow the steps below to adjust the tool model when its position is not correct.

1. Double-click the tool model under Project resource tree › Model library.
2. Adjust the pose parameters in the pop-up Edit Model Transform window to adjust the model's position in the 3D simulation area.

Follow the steps below to adjust the model when its size is not correct.

1. Double-click the tool model under Project resource tree › Model library.
2. In the pop-up Edit Model Transform window:
    - Adjust the **Scale** if you want to adjust the scale of the entire model.
    - If you want to adjust the scale on the individual X, Y, or Z direction, clear the **Keep ratio for X,Y,Z** checkbox, and adjust the scale on each direction respectively.

Configure the Control Logic of the Tool

If you are using a regular tool, you do not need to configure the control logic. If you are using a vacuum gripper or array gripper, please configure the control logic according to the actual situation. You can click the **[Configure depalletizing vacuum gripper]** or **[Configure array gripper]** button to open the corresponding configuration window.

Depalletizing vacuum gripper	Please refer to Depalletizing Vacuum Gripper for detailed instructions.
Array gripper	Please refer to Array Gripper Configurator for detailed instructions.

5.2.3. Workobjects

This section introduces workobjects and how to configure them.

Introduction

Workobjects refer to the objects on which the tool performs tasks, such as cartons, metal parts, parts to be glued or welded, etc.

Rotational Symmetry

A rotational symmetric workobject will coincide with itself after rotating around its axis of symmetry. Configuring the rotational symmetry of the workobject can prevent the end of the robot from rotating meaninglessly when picking and placing, and therefore the picking success rate can be increased and the planning time can be reduced. Therefore, the robot can move more smoothly and swiftly.

Please refer to Rotational Symmetry of Workobjects for detailed information.

Picking Relaxation

When the robot picks certain workobjects, the picking pose of the robot is allowed to be changed flexibly within a specific angle range, which is the picking relaxation.

Please refer to Picking Relaxation for detailed information.

Solution selection strategy

Option	Description
Minimum tool rotation	When this strategy is selected, during the whole pick-and-place process, the workobject pick point that causes the minimum rotation degree of the tool's Z-axis will be defined as the optimal pick point. This strategy can prevent the tool from rotating in vain after picking the object, and avoid falling the picked object.
Minimum difference between tool and vision pose	When this strategy is selected, the workobject pick point that has the minimum angle of deviation with the workobject pose will be defined as the optimal pick point.
Minimum collision between tool and point cloud	When this strategy is selected, the workobject pick point that leads to the minimum volume of collision between the tool and the workobject point clouds will be defined as the optimal pick point.

Workobject Configuration

Add the Workobject

1. Click the **[+]** next to Project resource tree › Workobjects to open the Workobject Configuration window.
2. Enter a **Workobject name**.

> ℹ The workobject name should be the same as the name of the “labels” field output by Mech-Vision. You can follow either of the steps below to check the field name set in Mech-Vision:
>
> - Check the detailed information of the “Procedure Out” Step in the **Log** panel, and the content displayed in "labels":["XXX"] is the workobject name.
> - Double-click the dataflow input to the **labels** input port of the “Procedure Out” Step, the content in quotes in the Debug Output panel on the right is the workobject name.

3. According to the actual situation of the workobject, select **Around Z-axis**, **Around Y-axis** or **Around X-axis** as the **Symmetry type** (can select multiple options), and then set the **Number of folds** and **Attempt range**. You do not need to set this parameter if the workobject is not rotational symmetric. You do not need to set this parameter if the workobject is not rotational symmetric.
4. According to the actual situation of the workobject, select **Around Y-axis** or **Around X-axis** for the **Picking relaxation** parameter (can select multiple options), and then set the **Attempt range** and **Step**. You do not need to set this parameter if a picking relaxation is not applicable.
5. Select a **Solution selection strategy** according to the actual requirement.
6. Click **[OK]**.

If you need to add more workobjects, please repeat the above steps.

Delete the Workobject

Follow either of the following steps to delete a workobject.

- Go to Resources › Workobjects, select the workobject, and press the `Delete` key.

- Right-click the workobject under Resources › Workobjects and select **[Delete]** in the context menu.

Modify the Workobject

1. Follow either of the following steps to open the Workobject Configuration window.
   - Go to Resources › Workobjects and double-click the workobject.
   - Right-click the workobject under Resources › Workobjects and select **[Workobject Configuration]** in the context menu.
2. Modify the parameters according to the actual requirement.
3. Click **[OK]**.

5.2.3.1. Rotational Symmetry of the Workobject

This section introduces the rotational symmetry of the workobject.

Introduction

The "symmetry" in this section refers to "rotational symmetry".

A rotational symmetric workobject will coincide with itself after rotating around its axis of symmetry. If a workobject has rotational symmetry, you can configure the rotational symmetry parameters according to the specific requirement in the Workobject Configuration window. Configuring the rotational symmetry of the workobject can prevent the end of the robot from rotating meaninglessly when picking and placing, and therefore the picking success rate can be increased and the planning time can be reduced. Therefore, the robot can move more smoothly and swiftly.

Rotational Symmetry

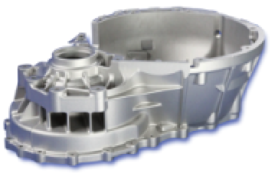
Determine the Axis of Symmetry

The axis of symmetry is one of the axes of the workobject's geometric center set in Matching Model and Pick Point Editor. The axis of symmetry is not unique. For the same workobject, in Mech-Vision, there can be multiple geometric centers due to different workobject positions and tool types, and therefore the determination of the axis of symmetry will be affected as well.

The table below shows the symmetry types of some workobjects that are placed in a common way. If you are not sure about the axis of symmetry of the workobject you use, please refer to Geometric Center Settings for more information.

Non-Symmetrical

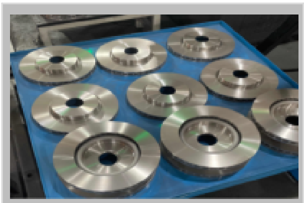
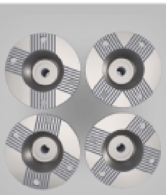
The workobjects in the table below are non-symmetrical workobjects.

Track pad of the bulldozer	Drive shaft flange yoke	Sheet metal of the car door	Transmission bell housing

Symmetrical about the Z-Axis

When the workobjects are placed in a normal way, workobjects as shown in the table below are symmetrical about the Z-axis.

			
Disc brake	Flange	Gear	Carton

Symmetrical about the X/Y-Axis

When the workobjects are placed in a normal way, workobjects as shown in the table below are symmetrical about the X/Y-axis.

Highly reflective steel rods that are arranged neatly	Steel rods that are stacked randomly	Oil pipe inserts	Bolts

Number of Folds

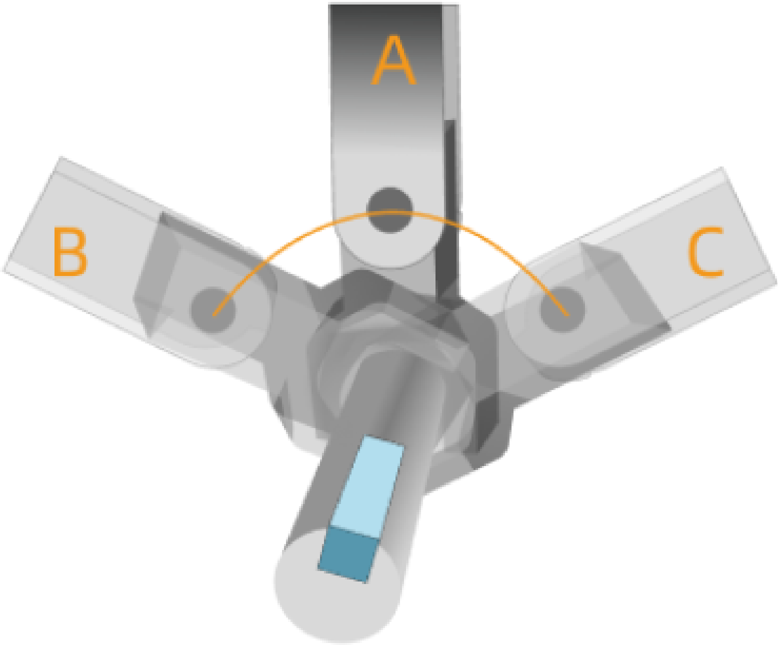
A rotational symmetric workobject will coincide with itself after rotating around its axis of symmetry by a certain degree (i.e., angle of rotation a°), and the number of folds N = 360°/a°.

For a common carton, N = 2; for a triangular prism, N = 3; for a cylinder, N = Infinity (i.e., circular symmetry in the software).

N = 2	N = 4	N = 9	Circular symmetry

Attempt Range

As shown in the figure below, the angle between B and C represents the attempt range.



"A" represents the picking pose when the tool is at the pick point, "B" and "C" represent the pose when the tool picks at the margin of the picking relaxation.

Please set the attempt range according to the workobject position, shape of the bin, tool type, takt time and other factors. An overly large attempt range may increase the planning time; while an overly small attempt range may result in missing the feasible pick point.

Number of Attempts

The number of attempts is calculated automatically from the number of folds and attempt range.

Assuming that the number of folds N = 10, and the attempt range is ±80°, then the angle of rotation can be calculated by 360°/10 = 36°. The number of attempts in one direction can be calculated by 80/36 = 2 and the remainder is 8, which is 2 attempts by the degree of 36° and 72°. The total number of attempts equals 1 + 2 * 2 = 5, which are by the degrees -72°, -36°, 0°, 36°, and 72°.

5.2.3.2. Picking Relaxation

This section introduces the picking relaxation.

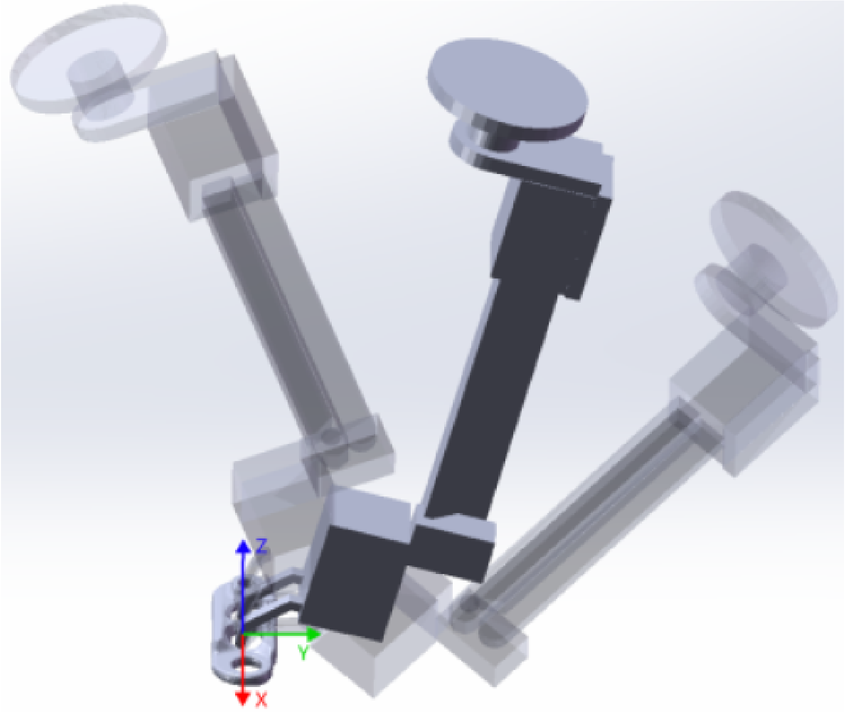
Introduction

When the robot picks certain workobjects, the picking pose of the robot is allowed to be changed flexibly within a specific angle range, which is the picking relaxation. Principally, picking relaxation is a tolerance based on the pick point. By setting the picking relaxation, the robot can utilize such tolerance flexibly to avoid problems such as collisions and singularities.

If the placing pose is set to the “workobject pose” and the robot picks the workobject with picking relaxation. The software will automatically record the angle deviation between the actual picking pose and the workobject pose. When the robot places the workobject, the software will adjust the tool pose according to the angle deviation and ensure that the workobject can be placed in a correct position.

Determine the Axis of Symmetry

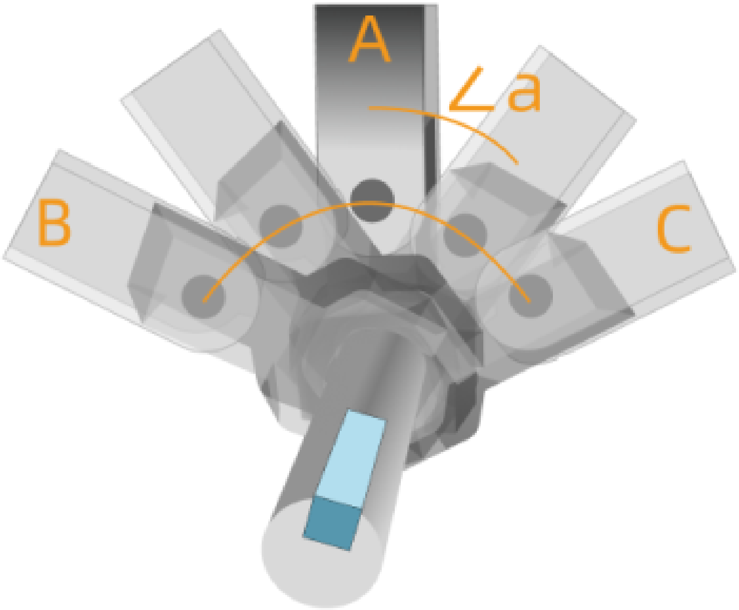
As shown in the figure below, the tool can pick the workobject flexibly within certain degrees around the X-axis of the workobject pose, and therefore the X-axis is the axis of symmetry.



Attempt Range and Step

The attempt range is the maximum allowable angle deviation between the picking pose and the vision pose. The angle between B and C represents the attempt range.

The Step specifies the degree interval between two picking attempts. The angle ∠ a in the figure below represents the step.



“A” represents the picking pose when the tool is at the pick point, “B” and “C” represent the pose when the tool picks at the margin of the picking relaxation.

Number of Attempts

The number of attempts is calculated automatically from the attempt range and step.

Assuming that the step is 5° and the attempt range is ±10°, then the number of attempts in one direction can be calculated by 10/5 = 2, which is 2 attempts by the degree of 5° and 10°. The total number of attempts = 1 + 2 * 2 = 5, which are by the degrees -10°, -5°, 0°, 5°, and 10°.

5.2.4. Floor

The simulated floor based on the real scenario. The default floor plane is at the same level where the bottom of the robot base is located. You should adjust the height of the floor according to the actual situation.

Adjust the Height of the Floor	Right-click Project resource tree › Floor, and modify the value or drag the slider bar in the pop-up window to adjust the height of the floor.
Modify the Style of the Floor	Select Settings › Options › Basic Settings › Ground Style, and select a style you like.

5.2.5. Scene Objects

This section introduces the scene objects and corresponding configurations.

Introduction

Scene objects, usually including the safety fence, picking bin, tray, camera, and camera stand, refer to objects in the real scenarios where the robot performs various tasks.

Scene Object Model

Adding scene object models makes the scene in the software closer to the real scenario, which facilitates the collision detection and path planning. You can create models of a cuboid, bin, or cylinder. You can also use a custom model imported from outside the software. The supported format of the visualization model and collision model are shown in the table below.

Format	STL	OBJ	DAE	Binvox
Visualization model	√	√	√	×
Collision model	√	√	×	×

The Parent-Child Relationship of the Scene Objects

You can establish a parent-child relationship between scene objects. Once the relationship is established, the child model will be moved as well if the parent model is moved, and the child model will be deleted as well if the parent model is deleted. However, the parent model will not be affected if you move or delete the child model individually.

Configure the Scene Object

You can add, modify, or delete the scene objects in the Resources panel.

Add a Scene Object

Please follow the steps below to add a **Cuboid**, **Cylinder**, or **Bin** type scene object.

1. Go to Project resource tree › Scene objects, and click **[+]** to open the Scene Object Configuration window.
2. In the **Object Settings** tab, select the **Scene Model** type in the drop-down list.
    - Set the **X, Y, Z (length, width, height)** for a **Cuboid** model.
    - Set the **Radius and Height** for a **Cylinder** model.
    - Set the **X, Y, Z (length, width, height), and Thickness** for a **Bin** model and then Set the Effective Range for the Vision Pose of the Bin.
3. Enter a custom name for the scene model in the **Object Name** text box.

> - If you do not want the currently added scene object to be involved in the collision detection, please unselect the **Involve in collision detection** parameter.
> - If you do not want the currently added scene object to be selected in the 3D simulation area, please unselect the **Model selectable** parameter.

4. By default, the newly added scene object is located at the origin of the robot base reference frame. You can click the **[Object Pose]** tab in the Scene Object Configuration window to switch the configuration interface, and modify the parameters to adjust the pose of the scene object.
5. Click **[OK]**.

Please follow the steps below to add a **Custom model**.

1. Click the **[+]** next to Project resource tree › Model library, and select the collision model and

visualization model in the pop-up window, and then click **[Open]**.

2. Go to Project resource tree › Scene objects, and click **[+]** to open the Scene Object Configuration window.
3. In the **Object Settings** tab, select **Custom model** as the **Scene Model**.
4. Set the model used for the collision detection as the **Collision model**.
5. Set the model used for visualization in the 3D simulation area as the **Visualization model**.
6. Enter a custom name for the scene model in the **Object Name** text box.

> ◦ If you do not want the currently added scene object to be involved in the collision detection, please unselect the **Involve in collision detection** parameter.
>
> ◦ If you do not want the currently added scene object to be selected in the 3D simulation area, please unselect the **Model selectable** parameter.

7. By default, the newly added scene object is located at the origin of the robot base reference frame. You can click the **[Object Pose]** tab in the Scene Object Configuration window to switch the configuration interface, and modify the parameters to adjust the pose of the scene object.
8. Click **[OK]**.

> If the “Collision model” of a “Custom model” is in STL format, it is recommended to create a cuboid or cylinder to cover the custom model for a higher accuracy of the collision detection.

If you need to add more tools, please repeat the above steps.

> - If you select an added model in the project resource tree and then click **[+]**, a child model of the current model will be added, and the default pose of the child model will be the same as that of the parent model.
> - You can drag the scene object in the project resource tree to change it into a child model or individual model.

Delete the Scene Object

Follow either of the following steps to delete a scene object.

- Select the scene model in the 3D simulation area or the model name under Project resource tree › Scene objects and press the **[Delete]** key.
- Right-click the scene object in the 3D simulation area or the model name under Project resource tree › Scene objects and select **[Delete]** in the context menu.

Modify the Scene Object

1. Follow either of the following steps to open the Scene Object Configuration window.
   - Double-click the object model in the 3D simulation area.
   - Go to Project resource tree › Scene objects and double-click the scene object name.
   - Right-click the name of the scene object under Project resource tree › Scene objects and select **[Scene Object Configuration]** in the context menu.
2. Modify the parameters according to the actual requirement.
3. Click **[OK]**.

> You can also select the scene object in the 3D simulation area, and adjust the model pose

> with the dragger.

Set the Effective Range for the Vision Pose of the Bin

When you add a bin as the scene model, you will need to configure the effective range for the vision pose in the **Bin Settings** tab.

- If the bin is only used as a scene model and it has no impact on the effective range for the vision pose, please unselect the **Set effective range for pose** parameter.
- If the vision pose must be in the bin, please select **In the bin**.
- If the vision pose may be above the bin top, please select **In the bin and above the bin top** and set the **Height range above bin** parameter.
- If you want to customize an effective range, please select **Custom effective range** and set parameters in **Effective Range of Pose**.

5.3. Build a Workflow

After configuring the project resources, you can start to build the workflow.

Introduction

The workflow of Mech-Viz is built by graphical programming. You can select and drag various Steps from the Workflow panel, and each Step refers to a function module of robot programming. You can create the robot control logic by setting the parameters of these Steps and connecting them together to define the logic and flow of the program.

Mech-Viz provides plenty of Steps to meet the requirements of various industrial scenarios, such as robot motion control, vision system connection, vision result processing, DI/DO control, logical topology, logical judgment, picking, depalletizing, and communication.

Basic Procedures for Using Steps

Objectives	Description
Search for a Step	Enter a keyword in the search box at the top of the Workflow panel or look in the relevant category for the Step you want.
Add a Step	Drag the Step from the Step library to the graphical programming workspace.
Delete a Step	Select the Step you want to delete and then press the `Delete` key; or right-click the Step you want to delete and select **[Delete]** in the context menu.
Connect Steps	Move the cursor onto the exit port of a Step and then drag the cursor to the entry port of another Step.
Delete a Connection between Steps	Click the connection you want to delete and then press the `Delete` key. Or you can select the connection, right-click it, and then select **[Delete]** in the context menu.
Customize Description of the Step	Right-click a Step and select **[Custom Description]** in the context menu.

Further Operations

1. Run Simulated Robot
2. Run Real Robot

5.3.1. Build a Workflow

After configuring the project resources, you can start to build the workflow.

Introduction

The workflow of Mech-Viz is built by graphical programming. You can select and drag various Steps from the Workflow panel, and each Step refers to a function module of robot programming. You can create the robot control logic by setting the parameters of these Steps and connecting them together to define the logic and flow of the program.

Mech-Viz provides plenty of Steps to meet the requirements of various industrial scenarios, such as robot motion control, vision system connection, vision result processing, DI/DO control, logical topology, logical judgment, picking, depalletizing, and communication.

Basic Procedures for Using Steps

Objectives	Description
Search for a Step	Enter a keyword in the search box at the top of the Workflow panel or look in the relevant category for the Step you want.
Add a Step	Drag the Step from the Step library to the graphical programming workspace.
Delete a Step	Select the Step you want to delete and then press the `Delete` key; or right-click the Step you want to delete and select **[Delete]** in the context menu.
Connect Steps	Move the cursor onto the exit port of a Step and then drag the cursor to the entry port of another Step.
Delete a Connection between Steps	Click the connection you want to delete and then press the `Delete` key. Or you can select the connection, right-click it, and then select **[Delete]** in the context menu.
Customize Description of the Step	Right-click a Step and select **[Custom Description]** in the context menu.

Further Operations

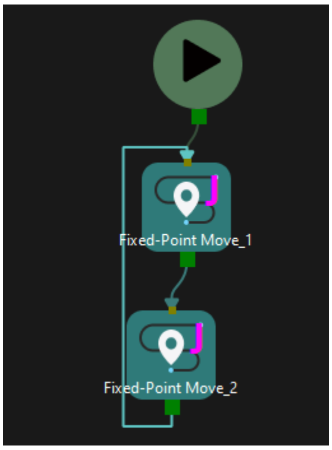
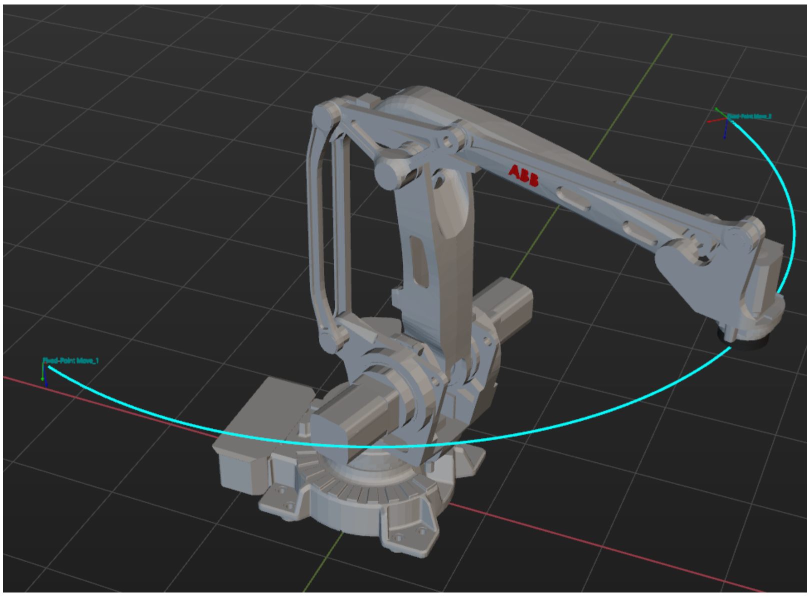
1. Run Simulated Robot
2. Run Real Robot

5.3.2. Run Simulated Robot

You need to plan a motion path for a simulated robot so that it can move by the path. A path consists of multiple waypoints. By adding waypoints, you can plan a path. Use the Fixed-Point Move Step to set the waypoints.

Instructions:

1. Open Mech-Viz, click **[New blank project]**.
2. Select any robot model in the Robot Model Library.
3. Click **[Workflow]** in the lower-right corner.
4. Find the Fixed-Point Move Step under the Basic Move category in the Workflow panel and then drag it to the right graphical programming workspace.
5. View the initial angles of robot joints under the JPs parameter on the right side. You can adjust the position of a joint by dragging the corresponding slider, and in the meanwhile, the pose of the robot changes with it, as can be seen in the 3D simulation area. The pose after adjustment can be taken as a waypoint.
6. Connect the start icon with Fixed-Point Move_1. Click **[Simulate]** button in the upper-left corner of the software interface. The robot will move to the waypoint you set before.
7. Add another waypoint. Similarly, drag the Fixed-Point Move Step in the Workflow panel to the graphical programming workspace. Then, adjust the angle of a joint. Connect the output port of Fixed-Point Move_1 with the input port of Fixed-Point Move_2 and then click the **[Simulate]** button. The robot will move from one waypoint to another.
8. Connect the output port of Fixed-Point Move_2 with the input port of Fixed-Point Move_1 and then click the **[Simulate]** button. The robot will move back and forth between the two waypoints. Click the **[Stop]** button to make it stop.

Workflow	Simulation
	

Now, you can plan a simple motion path for a simulated robot.

5.3.3. Run Real Robot

> The software communicates with the robot through Master-Control communication in this section.

1. Please refer to Master-Control Robot to establish the communication between the software and the robot, and then the robot can receive the commands from the software.
2. Drag the Fixed-Point Move Step in the Workflow panel to the graphical programming workspace. Then, change the JPs. It is recommended to **make a slight change of JPs within the safe working range**. The pose of the simulated robot at this point corresponds to a

waypoint of the real robot.

3. Connect the start icon with Fixed-Point Move_1.

> - Please lower the velocity of the real robot before making it move. You can **set the values of Acceleration and Velocity to 5%** in the Basic move settings.
> - When an emergency occurs, please press the emergency stop button on the robot teach pendant.

4. Click the **[Run]** button, and the robot will move to the waypoint set in Fixed-Point Move_1.

Now, you can use Mech-Viz to move a real robot.

5.4. Project Simulation and Improvement

After you complete configuring the project resources and building the workflow, you can simulate the project for testing.

Various errors may occur when simulating a newly created project. Please follow the guidelines in this section to configure relevant settings so that the project can meet the requirement in the actual application.

In scenarios where there are constraints, such as picking in a deep bin, it is vital to prevent the tool from colliding with the bin or other obstacles.Therefore, please enable the collision detection to detect possible collisions first. Then you can click **Simulate** in the upper left corner to start the simulation.

Collision Detection

If an error occurs during the simulation, such as a collision or unreachable waypoint occurs, you can check the error messages in Plan history and Log, and therefore you can locate the error quickly and then improve the project.

Plan history

Log

After improving the project, you can run the real robot to test the project.

Run a Real Robot

5.4.1. Collision Detection

This section introduces settings relevant to collision detection.

Overview

In applications such as machine tending and (de)palletizing, avoiding collisions between the robot and the bin and other obstacles is crucial to keep the project running continuously. Mech-Viz can detect possible collisions and plan the robot path accordingly, so that the resulting path is collision-free. If a collision cannot be avoided, Mech-Viz will stop the project and display the detected collision in the 3D simulation area, so that you can adjust the project accordingly.

Collision Detection Configuration

To configure the following collision detection, click **[Collision detection configuration]** in the **Collisions** panel to open the Collision Detection Configuration window.

Collision occurs between two objects. Different sets of collision models can make up different collision detection combinations. By default, Mech-Viz detects collisions between the following objects: robot links - robot links, robot links - scene objects, robot links - robot tool, scene objects - robot tool. Additionally, you can configure point clouds, held workobjects, and more. Before you start the configuration, please add and configure the corresponding collision models in the project resource tree.

- For adding and configuring the robot tool models, please refer to Tools.
- For adding and configuring the scene object models, please refer to Scene objects.
- The workobject models are automatically generated from the data provided by Mech-Vision. See below for detailed configuration instructions.

Collision between Point Cloud and Other Objects

Click the **Detect collision between point cloud and others** option in the **Configuration on point cloud** pane on the left to enable the detection of collisions between point cloud and the end tool. In addition, you can configure the detection of collisions between point cloud and robot links, and collisions between point cloud and held workobjects.

Mech-Viz only detects collisions with point cloud for the “Vision Move” Step and the “Relative Move” Steps that depend on the “Vision Move” Step, but not for the other move-type Steps.

- Point cloud cube edge length

  Point cloud cubes are the cubes centered at each point in the point cloud. If these cubes collide with other objects, the point cloud is considered to have collided with these objects.

  Adjust the **Point cloud cube edge length** parameter according to the requirements of collision detection accuracy and speed. If you want to improve detection accuracy, you can decrease this parameter, but it will result in slower detection speed. Conversely, increasing this parameter can speed up detection but may reduce accuracy.

- Point cloud collision calculation mode

  It is recommended to use Full calculation during project debugging, and use Minimal calculation after the project is stable.

Option	Description
Minimal calculation	In this mode, the calculation speed is fast, but the information is not recorded completely. The collision volume reported in the plan history is not the final value, and point cloud that causes the collision is not recorded. Applicable for the production phase after the project is stable.

Option	Description
Full calculation	In this mode, the calculation speed is lower, but the complete information can be recorded. The collision volume reported in the plan history is the final volume. You can click on the corresponding item to view the point cloud that causes the collision. Applicable for project debugging.

Point Cloud and Tools

On the Configuration on tool pane, Collision volume threshold refers to the allowable collision volume between the **point cloud cube** and **tool collision model**. If the collision volume exceeds this threshold, a collision between the robot tool and point cloud is considered to have occurred.

Please adjust the **Collision volume threshold** parameter according to the actual tolerance of collision.

Point Cloud and Robot Links

On the Configuration on robot links pane, Collision volume threshold refers to the allowable collision volume between the **point cloud cube** and **robot link model**. If the collision volume exceeds this threshold, a collision between the robot tool and point cloud is considered to have occurred.

Robot links are divided into wrist, forearm, upper arm and base. By default, collisions between the robot links and point cloud are not detected.

Please enable the collision detection for robot links according to the actual requirements and adjust the **Collision volume threshold** parameter according to the actual tolerance of collision.

Point Cloud and Held Workobjects

Enable **Detect collision between held workobject and others** option to detect collisions between the point cloud and held workobjects. Please follow the steps below to configure.

Collision between Held Workobject and Other Objects

Follow below steps to detect collisions between held workobjects and other objects. This option is mainly used for detecting collisions when moving the held workobject.

Please select Geometric solid or Custom model according to the model type of held workobject model.

Geometric Solid

Add Geometric Solid Models

Mech-Viz can generate collision models of cuboid and cylinder shapes based on the workobject dimensions provided by Mech-Vision. Therefore, please complete the following configurations in Mech-Vision to have the workobject dimensions output.

Configuration process:

1. In the Mech-Vision project, use either of the following Steps to obtain the workobject dimensions: Calc Poses and Dimensions from Planar Point Clouds: recommended for applications with multiple types of workobjects; Read Object Dimensions: recommended for

applications with a single type of workobject.

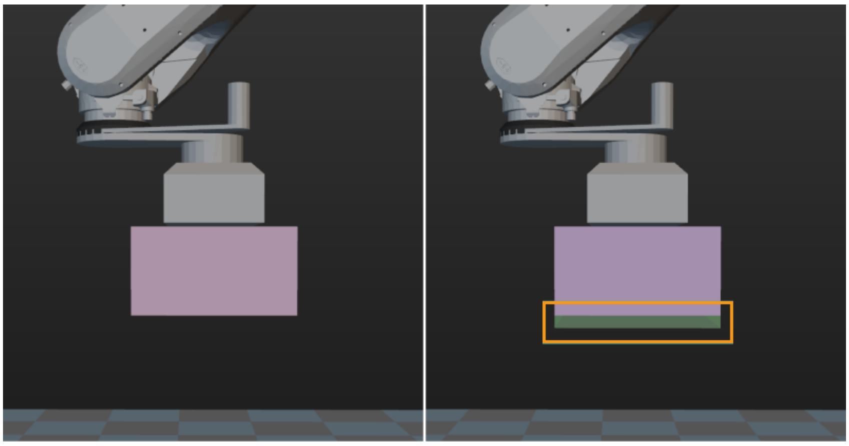
2. Output the workobject dimensions with the “Procedure Out” Step. Then Mech-Viz can generate the corresponding collision models.

> The number of output “pick point poses” and that of “object dimensions” should match.

Bottom Safety Distance in Mixed-Case Palletizing

For mixed-case palletizing applications, please set an appropriate cuboid bottom safety distance for the cuboid workobject models. The cuboid bottom safety distance adds a collision detection zone to the bottom of the workobject model. If other objects enter this zone, these objects are considered to have collided with the workobjects. Setting a safety distance at the bottom of the cuboid can effectively avoid collisions between the placed workobjects and the held workobject that the robot is moving.

To set a cuboid bottom safety distance, enable **Bottom safety distance in mixed-case palletizing**, and then adjust the value of **Cuboid bottom safety distance**.



Collision Volume Threshold

Collision volume threshold refers to the allowable collision volume between the **point cloud cube** and **held workobject model**. If the collision volume exceeds this threshold, a collision between the robot tool and point cloud is considered to have occurred. To adjust this parameter, enable the **Detect collision between point cloud and others** option.

Please adjust this parameter according to the actual tolerance of collision.

Custom Model

Add Custom Models

Configuration process:

1. Move the STL and BINVOX model files of the workobject to the `collision_models` folder under the Mech-Viz project folder. If there is no collision_models folder, please create one yourself.
2. Check the label name through the “Pose labels” port of the corresponding Step in Mech-Vision, and rename the model files to the label name. If you add a workobject configuration in the project resource tree of Mech-Viz, the workobject name must be the same as the label name.

3. In the “Send Point Cloud to External Service” Step of the corresponding Mech-Vision project, check the **Send Object Information** parameter and uncheck the **Input Point Cloud in Camera Frame** parameter. Connect all the input ports of this Step to the corresponding data flows.

   The output point cloud and pose must be in the robot reference frame.

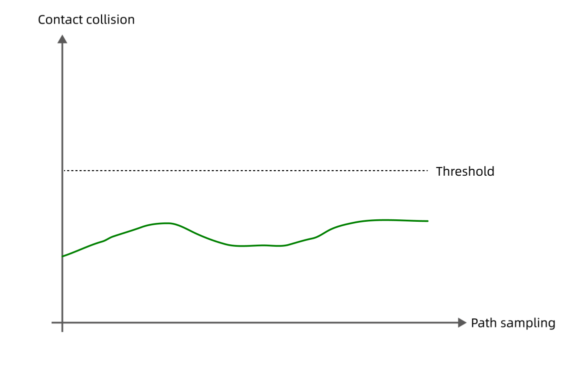
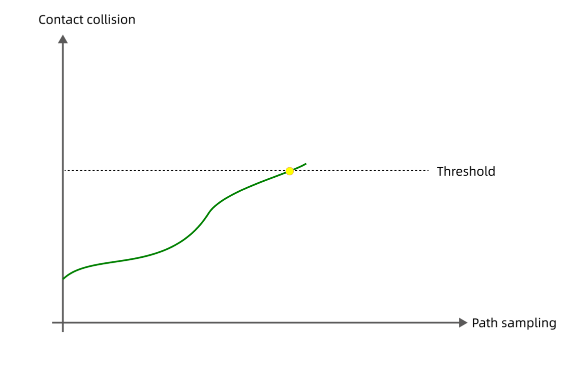
Collision Volume Threshold

Collision volume threshold refers to maximum allowable collision volume between the **point cloud cube** and **held workobject model**. If the collision volume between the held workobject model and the point cloud cubes exceeds this threshold, a collision will be detected. To adjust this parameter, enable the **Detect collision between point cloud and others** option.

Please adjust this parameter according to the actual tolerance of collision.

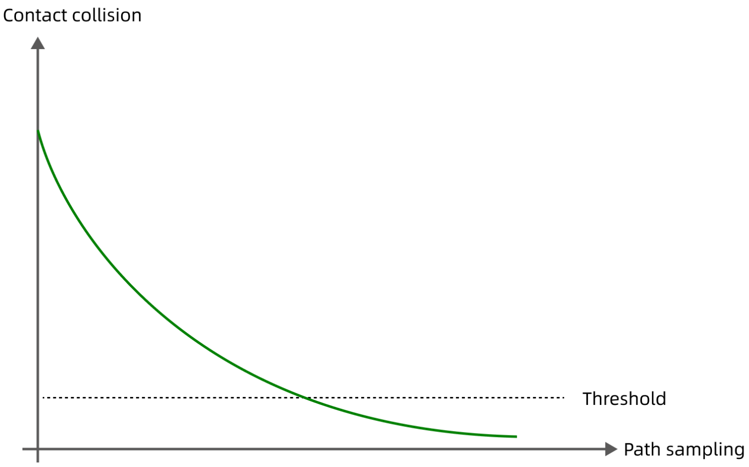
Collision Table Legend Description

- [Gray Block] Collision will not be detected or collision will not occur.
- [Blue Block] Consider collision detected when there is any contact.
- [Orange Block] Ignore collisions with contacts smaller than the configured volume threshold. Conversely, if the contact volume is larger than the configured volume threshold, a collision will be detected.

	
Contact volume remains below the configured volume threshold throughout the path sampling, no collision detected	Contact volume exceeds the configured volume threshold throughout the path sampling, collision detected

- [Green Block] Consider collision detected when there is any object within bottom safety distance below held workobject(s).
- [Yellow Block] Ignore collisions with contacts that are larger than the configured volume threshold at first but the collision volumes gradually decrease, and collisions with contacts smaller than the configured volume threshold.

  Although the contact volume **exceeds the configured volume threshold**, collisions whose contact volume shows **a decreasing trend** throughout the path sampling are allowed by the software.



5.4.2. Plan History

This section introduces the function of Plan history. The following topics are covered:

Overview

The overall plan history is laid out in a tree structure, which contains every node in the whole process of path planning from Mech-Viz.

Comparing to the log, plan history contains more details of every node (especially when there is an error) in the process.

- Some errors are serial errors, which means that if the error occurs at one of the nodes, the final plan result will fail. For example, a certain joint position is set wrong, and the robot cannot move to the planned waypoint.
- Some errors are parallel errors, which means that the final plan result will still succeed if the planning at one of the parallel nodes is successful. For example, when the robot attempts to pick with the pick points sent by Mech-Vision, if either one of the pick points works, the workobject can be picked successfully.

> You can focus on the nodes where errors occur when running the project.

Visualize Collision

Compute and Record Collision Contacts

During project debugging, in order to record the complete collision contact and save it to the plan history, before you simulate and run the project, please click Collision detection configuration and then set the **Point cloud collision calculation mode** to **Full calculation** in Configuration on point cloud panel.

Animation Prompts for Collisions in Plan History

Click any item in the plan history, and the animation corresponding to the current plan is displayed, including:

- The green solid line represents a successful path.

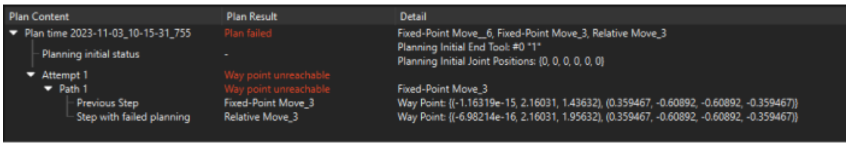
- The red solid line represents a failed path.
- The red dashed line represents an unreachable path.
- The yellow ball on the line shows the direction of the robot’s motion.
- The highlighted parts represent the areas where collisions occur.

Change the **Animation duration** parameter in the lower-left corner of the Plan history interface to adjust the duration of the animation display.

Causes of Common Failures and Fixes

The following lists several common failures and corresponding fixes.

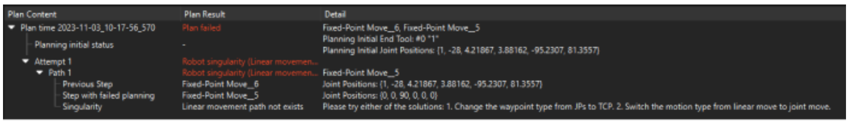
Waypoint unreachable



Plan Content	Plan Result	Detail
Plan time 2023-11-03_10-15-31_755	Plan failed	Fixed-Point Move__6, Fixed-Point Move_3, Relative Move_3
Planning initial status	-	Planning Initial End Tool: #0 "1" Planning Initial Joint Positions: {0, 0, 0, 0, 0, 0}
Attempt 1	Way point unreachable	
Path 1	Way point unreachable	Fixed-Point Move_3
Previous Step	Fixed-Point Move_3	Way Point: {(-1.16319e-15, 2.16031, 1.43632), (0.359467, -0.60892, -0.60892, -0.359467)}
Step with failed planning	Relative Move_3	Way Point: {(-6.98214e-16, 2.16031, 1.95632), (0.359467, -0.60892, -0.60892, -0.359467)}

Plan Result	The waypoint of Step “Relative Move_3” is out of the reachable range of the robot.
Fix	Adjust the waypoint of step “Relative Move_3” to make the robot reach the waypoint.

Robot singularity

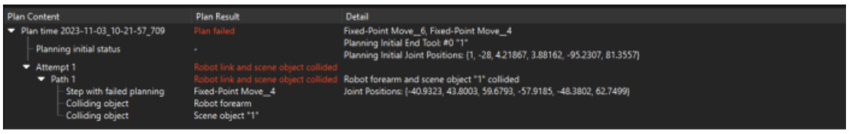


Plan Content	Plan Result	Detail
Plan time 2023-11-03_10-17-56_570	Plan failed	Fixed-Point Move__6, Fixed-Point Move__5
Planning initial status	-	Planning Initial End Tool: #0 "1" Planning Initial Joint Positions: {1, -28, 4.21867, 3.88162, -95.2307, 81.3557}
Attempt 1	Robot singularity (Linear movemen...	
Path 1	Robot singularity (Linear movemen...	Fixed-Point Move__5
Previous Step	Fixed-Point Move__6	Joint Positions: {1, -28, 4.21867, 3.88162, -95.2307, 81.3557}
Step with failed planning	Fixed-Point Move__5	Joint Positions: {0, 0, 90, 0, 0, 0}
Singularity	Linear movement path not exists	Please try either of the solutions: 1. Change the waypoint type from JPs to TCP. 2. Switch the motion type from linear move to joint move.

Plan Result	There is a singularity in the path from step “Fixed-Point Move_6” to Step “Fixed-Point Move_7”.
Fixes	Change the waypoint type of step “Fixed-Point Move_7” from JPs to TCP.
	Switch the motion type from linear move to joint move.
	Modify the parameters corresponding to singularity.

Collision

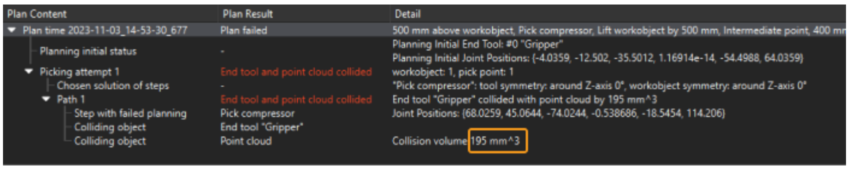
Robot link and scene object collided



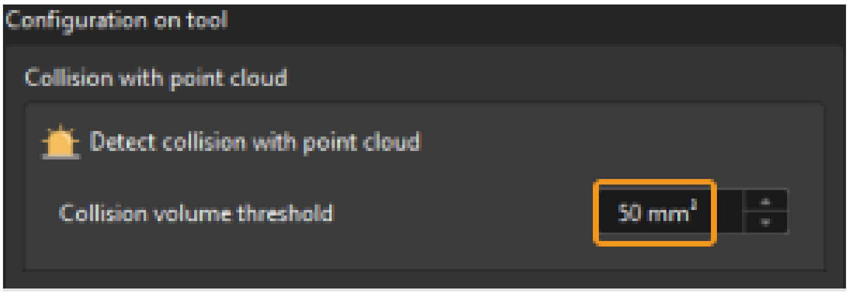
Plan Content	Plan Result	Detail
Plan time 2023-11-03_10-21-57_709	Plan failed	Fixed-Point Move__6, Fixed-Point Move__4
Planning initial status	-	Planning Initial End Tool: #0 "1" Planning Initial Joint Positions: {1, -28, 4.21867, 3.88162, -95.2307, 81.3557}
Attempt 1	Robot link and scene object collided	
Path 1	Robot link and scene object collided	Robot forearm and scene object "1" collided
Step with failed planning	Fixed-Point Move__4	Joint Positions: {-40.9323, 43.8003, 59.6793, -57.9185, -48.3802, 62.7499}
Colliding object	Robot forearm	
Colliding object	Scene object "1"	

Plan Result	Robot forearm and scene object 1 collided at the waypoint of Step “Fixed-Point Move_4”.
Fixes	Adjust the waypoint of Step “Fixed-Point Move_4” to avoid collision with scene object 1.
	Adjust the position of scene object 1 if the actual scenario allows.

End tool and point cloud collided



Plan Content	Plan Result	Detail
Plan time 2023-11-03_14-53-30_677	Plan failed	500 mm above workobject, Pick compressor, Lift workobject by 500 mm, Intermediate point, 400 mr
Planning initial status	-	Planning Initial End Tool: #0 "Gripper" Planning Initial Joint Positions: {-4.0359, -12.502, -35.5012, 1.16914e-14, -54.4988, 64.0359}
Picking attempt 1	End tool and point cloud collided	workobject: 1, pick point: 1
Chosen solution of steps	-	"Pick compressor": tool symmetry: around Z-axis 0°, workobject symmetry: around Z-axis 0°
Path 1	End tool and point cloud collided	End tool "Gripper" collided with point cloud by 195 mm^3
Step with failed planning	Pick compressor	Joint Positions: {68.0259, 45.0644, -74.0244, -0.538686, -18.5454, 114.206}
Colliding object	End tool "Gripper"	
Colliding object	Point cloud	Collision volume 195 mm^3



Configuration on tool

Collision with point cloud

Detect collision with point cloud

Collision volume threshold 50 mm³

Plan Result	The collision volume of the tool and the point cloud is **154 mm³**, which is greater than the set threshold **50 mm³**.
Fixes	You may increase the maximum collision volume threshold according to the actual tolerance for collision.
	Adjust the TCP, pick points, and pose of the collision model of the end tool according to the actual situation.
	Adjust the point cloud output through Mech-Vision.

5.4.3. Run a Real Robot

Please follow these steps to run a real robot.

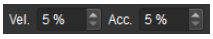
- Before you run a real robot, please ensure that you have lowered the robot speed to a proper value for safety concerns. Pay attention to avoid an accident.
- When an emergency occurs when the robot is moving, please press the emergency stop button on the robot teach pendant.

1. Connect to the Robot Properly

   Before you use Mech-Viz to guide a real robot to move, please ensure that the **communication between the robot and the software** is established successfully, or else the robot cannot be moved by Mech-Viz.

2. Reduce the Robot Speed

   Adjust the **Velocity** and **Acceleration** in the Mech-Viz toolbar to 5%.

   Vel. 5 % Acc. 5 %

3. Click **[Run]** in the Mech-Viz toolbar, and then the robot will follow the planned path to move.
4. Click **[Stop]** in the Mech-Viz toolbar to stop the project.

If an error occurs when running a real robot, you can locate the reason with the assistance of Plan History and Log.

5.5. Step Library

Please click the corresponding links for the detailed descriptions of Step parameters.

Basic Move	Adaptive Relative Move
	Dynamic Move
	Fixed-Point Move
	Move by Grid
	Move by List
	External Move
	Relative Move
DI DO	Check DI
	Set DO
Logical Topology	Branch by Tag
	Branch by Msg
	Procedure
	Procedure Exit
	Set Tag for Branch
Palletizing	Custom Pallet Pattern
	Mixed Pallet Pattern
	Predefined Pallet Pattern
	Vision Continue Palletizing
	Multi Pick Palletizing
Robot Utilities	Check Tool
	Set Payload
	Change Tool
	Set Holding State
	Get JPs
	Transfer Control
Service	Notify
Utilities	Classify
	Counter
	Check Completion
	Change Index
	Reset
	Wait
Path	Path Procedure

Vision	Check Look
	Update Pallet Pose
	Update Held Workobject
	Update Scene Objects
	Vision Place
	Is Vision Result Used Up
	Vision Look
	Vision Move
Others	Stop Workflow

5.5.1. Basic Move

5.5.1.1. Adaptive Relative Move

Function

This Step adaptively adjusts the distance to a reference point based on the height of the held workobject.

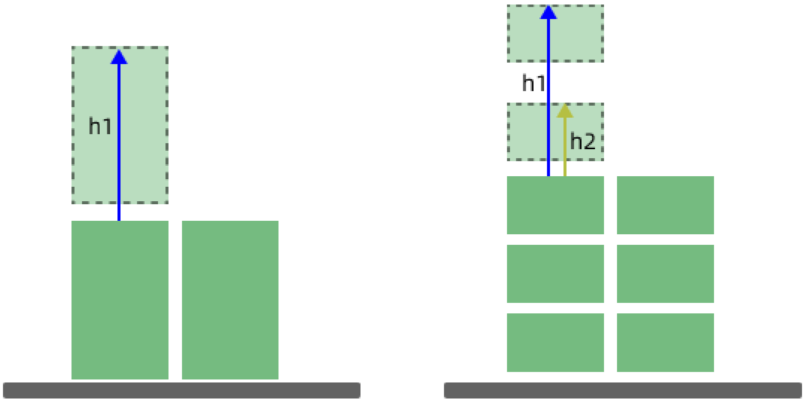
▶ https://docs.mech-mind.net/download/Mech-Viz/adaptive-relative-move-en.mp4 *(video)*

Usage Scenario

The lifting distance when the "Relative Move" Step is used is fixed, while the "Adaptive Relative Move" Step can adjust the lifting distance according to the box height. The box height is obtained from the vision result output by Mech-Vision.

Depalletizing Scenario

In depalletizing scenarios, to ensure that all boxes can be safely picked, the lifting distance should be set according to the dimensions of the largest box. As shown in the figure below, the lifting distance in the Z-direction is denoted by h1, i.e., boxes of all dimensions will be lifted by h1. When depalletizing smaller boxes as shown below, the box needs to be lifted by h2 ideally. However, the software will still guide the robot to lift the box by h1, which brings extra movement of the robot and affects the takt time and robot flexibility.

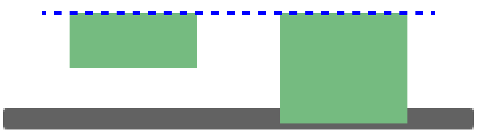
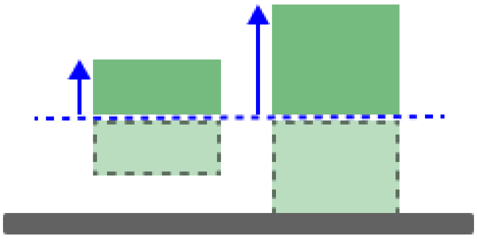


Once this Step is used, the lifting distance can be adjusted dynamically according to the box height, which reduces extra movement, upgrades the takt time of the robot motion, and prevents the robot from moving to unreachable waypoints.

Placement Scenario

After the boxes are depalletized, they may need to be placed on the conveyer belt. Since the pick points of the boxes are located on the upper surface of the boxes, the robot will release the held boxes at the same height and cannot adjust the placing positions according to the height of the held box, regardless of whether the waypoint type is TCP or workobject pose.

Once this Step is used, the box height will be added to the height for placing the box, and therefore the placing position can be adjusted.

	
Fixed placing positions	Placing positions after adding the box heights

Parameter Description

Move-Type Step Common Parameters

▼ *Send Waypoint*

Selected by default to send waypoint poses to the receiver, such as the robot. When this option is unselected, the waypoint pose will not be sent. However, the waypoint will remain in the planned path.

▼ *Try Continuously Running through Succeeding Non-Moves*

Unselected by default. When **non-move Steps**, such as Vision Look, Set DO, Check DI, etc., are connected between **move-type Steps**, the robot's path planning will be interrupted, and the actual robot will take a short pause, reducing the smoothness of running.

When this option is selected, the project will continue to run without waiting for the current move-type Step to complete execution, and therefore the robot can move in a smooth way without pauses. However, enabling this option may cause the execution of the Step to end prematurely.

Why will this option cause the execution of the Step to end prematurely?

Mech-Viz will send multiple poses simultaneously to the robot when the project is running. When the currently returned JPs of the robot correspond to the last pose sent by Mech-Viz, Mech-Viz will assume that the robot has moved to the last position.

For example, there are 10 move-type Steps in a path, and the pose of the 5th move-type Step is the same as that of the last move-type Step. When the robot moves at low speed, it sends JPs to Mech-Viz when it moves to the 5th move-type Step, Mech-Viz may mistakenly determine that the robot has finished the move-type Steps and prematurely ends the Steps since the poses of the 5th move-type Step and the last move-type Step are the same in the path.

▼ *Do Not Check Collision with Placed Workobject*

Unselected by default, namely that the collision with the already placed objects will not be detected. When this option is selected, the collisions between the robot, end tool, and placed objects will be detected.

In palletizing scenarios, the two possible cases of error are as follows:

1. When the robot is placing a carton, the robot may come into light contact with the placed cartons while no deformation will be caused. After Mech-Viz detects this collision in simulation, it will plan other positions for placing the carton, and therefore a full stack cannot be formed.
2. Usually, the TCP of a suction cup is inside the suction cup model instead of on the surface of it. Under this circumstance, the suction cup may be embedded in the model of the picked carton in the simulation of picking, while the software does not detect the collision between the end tool and the picked object. After the robot places the object and the carton model turns into an object model in the scene, a collision between the suction cup and the carton will be detected and the palletizing cannot be completed.

When this option is selected, no collision between the robot, end tool, and the placed object will be detected, and the above two cases of errors can be avoided.

▼ *Point Cloud Collision Detection Mode*

Select the proper mode according to the requirement of the on-site situation. Usually, the default setting **Auto** can be used. **Do not check collision** mode can be used in move-type Steps before the robot picks the object, and **Check collision** mode can be used after the robot picks the object.

Auto	Default setting. Collision with point cloud is checked only for the “Vision Move” Step and the “Relative Move” Step that depends on the “Vision Move” Step, but not for all move-type Steps.
Do not check collision	Point cloud collisions for all move-type Steps will not be detected.
Check collision	Point cloud collisions for all move-type Steps will be detected.

When Collisions › Collision detection configuration › Detect collision between point cloud and others is switched on, Mech-Viz will detect collisions between the robot model, end tool model, and point cloud when planning the path. Normally, detects whether the robot collides with other objects during picking and placing. When there are point cloud outliers, non-exiting collisions will be detected, which leads to errors in path planning.

▼ *Ignore Workobject Symmetry*

This parameter will only take effect when **Waypoint type** of the Step is set to **Workobject pose**.

None	Default setting, i.e., do not disable symmetry on any axis.
Around workobject frame Z axis	Only disable symmetry on Z-axis of the workobject reference frame.

Around workobject frame X&Y axis	Disable symmetry on X-axis and Y-axis of the workobject reference frame.
Around all axes	Once the object symmetry is disabled, the robot will place the objects strictly according to the workobject poses.

In some special cases, objects are not pickable due to their peculiar poses. Setting **Rotational symmetry** under Workobjects › Workobject Configuration in Resources may solve this problem. Based on the settings of Rotational symmetry, when the default object pose is not pickable, Mech-Viz will calculate candidate poses and determine whether the candidate poses are pickable. As the candidate poses calculated based on the settings of Rotational symmetry are different from the original poses output from Mech-Vision, the consistency of the objects' place poses cannot be guaranteed.

▼ *Plan Failure Out Port*

After selecting this parameter, this Step will add "Plan failure" out port.

During the planning process, planning is carried out along the branch after the "Success" out port; if the planning fails in the current Step, the branch process after the "Planning failure" out port will be executed.

Held Workobject Collision Detection Settings

▼ *Do Not Check Collision with Scene Object/Robot*

Unselected by default. Once this option is selected, the collisions between the **held workobject** with the **scene objects or robot** will not be detected, and therefore the calculation workload of collision detection will be reduced, the planning speed can be increased, and the cycle time can be shortened. It is usually enabled in the first one or two move-type Steps after the robot picks the object.

Please enable this option cautiously as there may be collision risks.

When **Detect collision between held workobject and others** under Collision detection configuration › Configuration on held workobject is enabled, the software will detect whether the model of the held object collides with the models of the scene objects and the robot.

In palletizing projects, the calculated carton dimensions have millimeter-level errors with the actual dimensions, and frictions between cartons may occur during picking but no collisions will occur. For some move-type Steps that will obviously not cause collisions, detecting such collisions only adds to the calculation workload and planning time, and consequently extending the cycle time. In palletizing projects, enabling Do Not Check Collision with Scene Objects does not affect the collision detection between the held carton and the placed cartons. This option can be enabled when there are scene objects under the stack to avoid failure of finding the palletizing solution.

▼ *Do Not Check Collision with Point Cloud*

Unselected by default. Once this option is selected, the collisions between the **held workobject** with the **point clouds in the scene** will not be detected, and therefore the calculation workload of collision detection will be reduced, the planning speed can be increased, and the cycle time can be shortened.

- When both **Detect collision between held workobject and others** under Collision detection configuration › Configuration on held workobject and **Configuration on point cloud ¬ Detect collision between point cloud and others** are enabled, the software will detect whether the model of the held workobject will collide with the point cloud in the scene.

- When Mech-Vision sends the point cloud and object model to Mech-Viz, the point cloud and the object model are fitted together. After the robot picks the object, the model moves along the planned path, and the collision between the model of the held workobject and the point cloud will occur.
- It is known that the model of the held workobject will have false collisions with the point cloud. Detecting such collisions unnecessarily adds to the calculation workload and extends the planning time.

Basic Move Settings

▼ *Motion type*

Joint move

Joint motion, which guides the robot to move in a curved path. It is less likely to reach singularities in the path for joint motion. This motion type is applicable to scenarios where the requirement of path accuracy is not strict and the robot moves in a large space.

Linear move

Linear motion, which guides the robot to move linearly. This motion type is applicable to scenarios where there is a strict requirement for path accuracy, such as welding, gluing, and certain types of picking.

▼ *Velocity & Acceleration*

Velocity and acceleration determine how fast the robot can move. Usually, the set acceleration should be lower than the velocity. When the set acceleration is higher than the velocity, the robot will move in a choppy way.

> ❗ The velocities of Vision Move and its prior and subsequent Steps should be relatively low to ensure that the objects can be picked steadily.

▼ *Blend radius*

Default value

50.00 mm

Notes

Usually, the default setting can be used.

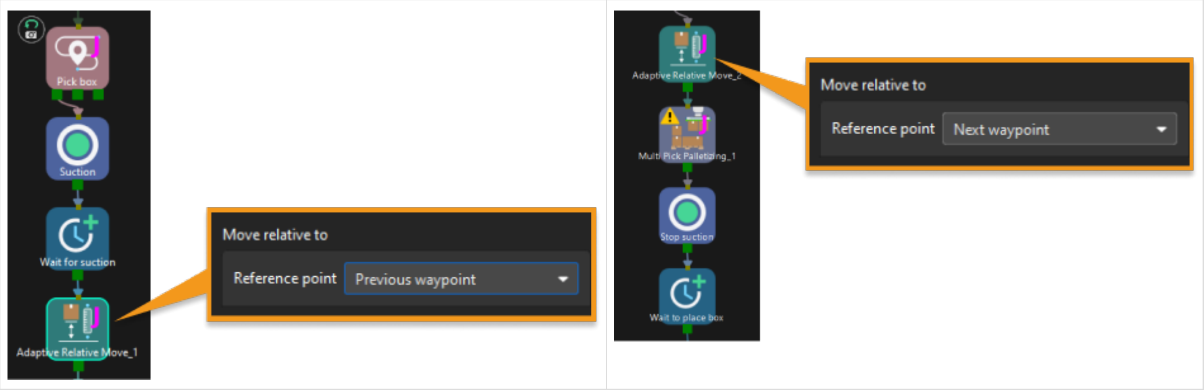
The blend radius refers to the distance between the target point and the point where the robot starts to turn. The larger the blend radius, the more smoother the robot motion transitions are. If the robot moves in a relatively small space, please set the blend radius to a smaller value.

If the robot moves in a relatively large space without obstacles and the distance between two consecutive path segments is long, please set the blend radius to a larger value.

Reference point

The reference point for the adaptive relative move, which will be used as the starting position in the calculation of the offset, and therefore the target waypoint of this Step can be generated.

Previous waypoint is usually selected after **picking**.	**Next waypoint** is usually selected after **placing**.



Move relative to

Reference point: Previous waypoint

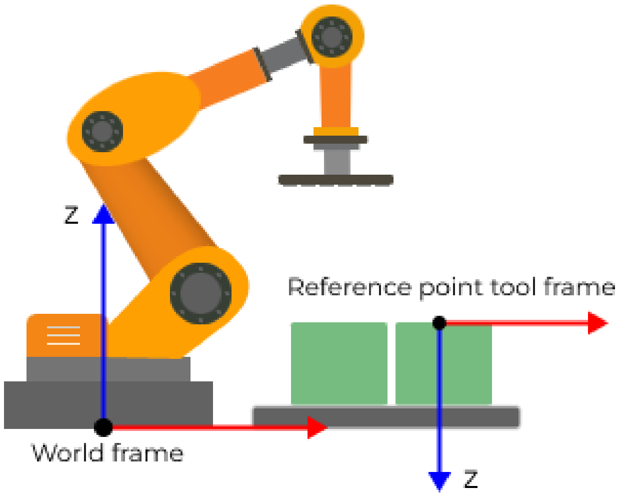
Move relative to

Reference point: Next waypoint

Relative move offset

You need to specify the **Offset direction** and **Fixed offset** to determine the offset here.

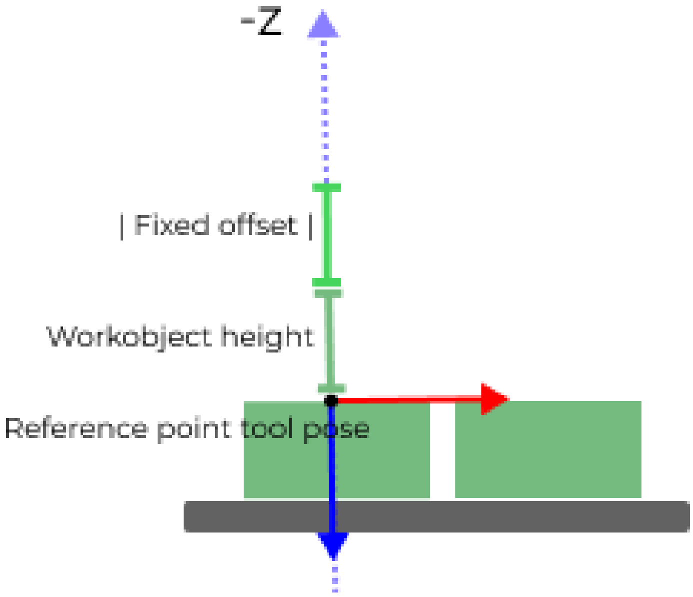
This Step supports setting the offset along the Z-direction of the reference point represented in the form of tool pose or along the Z-direction in the world frame.

> Illustration of the world frame and the tool frame of the reference point (assuming that the pick point is set to the reference point):
>
> 
>

- Tool pose Z-direction of reference point: Set the offset along the Z-direction of the reference point represented in the form of tool pose.

  Relative move offset = Fixed offset - Workobject height in the Z-direction

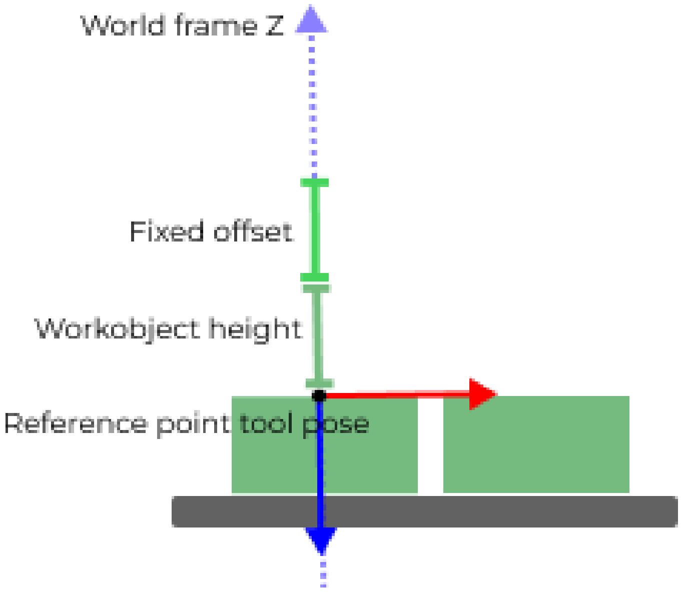
  Since the Z-axis of the tool pose usually points to the ground, it is recommended to set Fixed offset to a **negative value** to add a height and lift the box successfully. Then the robot end tool will offset along the **negative Z-direction** in the tool frame by the distance of |Fixed offset| plus the box height in the Z-direction.



- World frame Z-direction: Set the offset along the Z-direction in the world frame.

  Relative move rotation = Fixed offset + Workobject height in the Z-direction

  Since the Z-axis of the world frame usually points upwards, it is recommended to set Fixed offset to a **positive value**. Then the robot end tool will offset along the **positive Z-direction** in the world frame by the distance of Fixed offset plus the box height in the Z-direction.



Relative move rotation

The rotation offset relative to the tool pose of the reference point in the tool frame.

For example, if **Z(EZ)** under **Euler angles** is set to 90°, the suction cup will rotate 90° clockwise with the Z-axis as the rotation axis after picking up the object.

5.5.1.2. Dynamic Move

Function

This Step guides the robot to move according to the highest pose in the corresponding Vision Look Step, and therefore the camera can be at a proper distance from the workobject.

Usage Scenario

This Step is usually used in EIH mode.

Parameter Description

Move-Type Step Common Parameters

▼ *Send Waypoint*

Selected by default to send waypoint poses to the receiver, such as the robot. When this option is unselected, the waypoint pose will not be sent. However, the waypoint will remain in the planned path.

▼ *Try Continuously Running through Succeeding Non-Moves*

Unselected by default. When **non-move Steps**, such as Vision Look, Set DO, Check DI, etc., are connected between **move-type Steps**, the robot's path planning will be interrupted, and the actual robot will take a short pause, reducing the smoothness of running.

When this option is selected, the project will continue to run without waiting for the current move-type Step to complete execution, and therefore the robot can move in a smooth way without pauses. However, enabling this option may cause the execution of the Step to end prematurely.

Why will this option cause the execution of the Step to end prematurely?

Mech-Viz will send multiple poses simultaneously to the robot when the project is running. When the currently returned JPs of the robot correspond to the last pose sent by Mech-Viz, Mech-Viz will assume that the robot has moved to the last position.

For example, there are 10 move-type Steps in a path, and the pose of the 5th move-type Step is the same as that of the last move-type Step. When the robot moves at low speed, it sends JPs to Mech-Viz when it moves to the 5th move-type Step, Mech-Viz may mistakenly determine that the robot has finished the move-type Steps and prematurely ends the Steps since the poses of the 5th move-type Step and the last move-type Step are the same in the path.

▼ *Do Not Check Collision with Placed Workobject*

Unselected by default, namely that the collision with the already placed objects will not be detected. When this option is selected, the collisions between the robot, end tool, and placed objects will be detected.

In palletizing scenarios, the two possible cases of error are as follows:

1. When the robot is placing a carton, the robot may come into light contact with the placed cartons while no deformation will be caused. After Mech-Viz detects this collision in simulation, it will plan other positions for placing the carton, and therefore a full stack cannot be formed.
2. Usually, the TCP of a suction cup is inside the suction cup model instead of on the surface

of it. Under this circumstance, the suction cup may be embedded in the model of the picked carton in the simulation of picking, while the software does not detect the collision between the end tool and the picked object. After the robot places the object and the carton model turns into an object model in the scene, a collision between the suction cup and the carton will be detected and the palletizing cannot be completed.

When this option is selected, no collision between the robot, end tool, and the placed object will be detected, and the above two cases of errors can be avoided.

▼ *Point Cloud Collision Detection Mode*

Select the proper mode according to the requirement of the on-site situation. Usually, the default setting **Auto** can be used. **Do not check collision** mode can be used in move-type Steps before the robot picks the object, and **Check collision** mode can be used after the robot picks the object.

Auto	Default setting. Collision with point cloud is checked only for the “Vision Move” Step and the “Relative Move” Step that depends on the “Vision Move” Step, but not for all move-type Steps.
Do not check collision	Point cloud collisions for all move-type Steps will not be detected.
Check collision	Point cloud collisions for all move-type Steps will be detected.

When Collisions › Collision detection configuration › Detect collision between point cloud and others is switched on, Mech-Viz will detect collisions between the robot model, end tool model, and point cloud when planning the path. Normally, detects whether the robot collides with other objects during picking and placing. When there are point cloud outliers, non-exiting collisions will be detected, which leads to errors in path planning.

▼ *Ignore Workobject Symmetry*

This parameter will only take effect when **Waypoint type** of the Step is set to **Workobject pose**.

None	Default setting, i.e., do not disable symmetry on any axis.
Around workobject frame Z axis	Only disable symmetry on Z-axis of the workobject reference frame.
Around workobject frame X&Y axis	Disable symmetry on X-axis and Y-axis of the workobject reference frame.
Around all axes	Once the object symmetry is disabled, the robot will place the objects strictly according to the workobject poses.

In some special cases, objects are not pickable due to their peculiar poses. Setting **Rotational symmetry** under Workobjects › Workobject Configuration in Resources may solve this problem. Based on the settings of Rotational symmetry, when the default object pose is not pickable, Mech-Viz will calculate candidate poses and determine whether the candidate poses are pickable. As the candidate poses calculated based on the settings of Rotational symmetry are different from the original poses output from Mech-Vision, the consistency of the objects’ place poses cannot be guaranteed.

▼ *Plan Failure Out Port*

After selecting this parameter, this Step will add “Plan failure” out port.

During the planning process, planning is carried out along the branch after the “Success” out port; if the planning fails in the current Step, the branch process after the “Planning failure” out port will be executed.

Held Workobject Collision Detection Settings

▼ *Do Not Check Collision with Scene Object/Robot*

Unselected by default. Once this option is selected, the collisions between the **held workobject** with the **scene objects or robot** will not be detected, and therefore the calculation workload of collision detection will be reduced, the planning speed can be increased, and the cycle time can be shortened. It is usually enabled in the first one or two move-type Steps after the robot picks the object.

Please enable this option cautiously as there may be collision risks.

When **Detect collision between held workobject and others** under Collision detection configuration › Configuration on held workobject is enabled, the software will detect whether the model of the held object collides with the models of the scene objects and the robot.

In palletizing projects, the calculated carton dimensions have millimeter-level errors with the actual dimensions, and frictions between cartons may occur during picking but no collisions will occur. For some move-type Steps that will obviously not cause collisions, detecting such collisions only adds to the calculation workload and planning time, and consequently extending the cycle time. In palletizing projects, enabling Do Not Check Collision with Scene Objects does not affect the collision detection between the held carton and the placed cartons. This option can be enabled when there are scene objects under the stack to avoid failure of finding the palletizing solution.

▼ *Do Not Check Collision with Point Cloud*

Unselected by default. Once this option is selected, the collisions between the **held workobject** with the **point clouds in the scene** will not be detected, and therefore the calculation workload of collision detection will be reduced, the planning speed can be increased, and the cycle time can be shortened.

- When both **Detect collision between held workobject and others** under Collision detection configuration › Configuration on held workobject and **Configuration on point cloud ¬ Detect collision between point cloud and others** are enabled, the software will detect whether the model of the held workobject will collide with the point cloud in the scene.
- When Mech-Vision sends the point cloud and object model to Mech-Viz, the point cloud and the object model are fitted together. After the robot picks the object, the model moves along the planned path, and the collision between the model of the held workobject and the point cloud will occur.
- It is known that the model of the held workobject will have false collisions with the point cloud. Detecting such collisions unnecessarily adds to the calculation workload and extends the planning time.

Z Offset (Highest Workobject to Waypoint)

Default value	0

Instruction	When the camera is mounted in the EIH mode, the distance between the camera and the workobject should remain unchanged when the robot picks the workobject. Once this parameter is set, the waypoint Z coordinate corresponding to "Dynamic Move" will be adjusted according to the Z value of the highest pose from the Vision Look Step. (Waypoint Z coordinate = Z offset + highest workobject pose Z coordinate.)

Lower Limit of Robot Pose Z

Default value	−10 m
Instruction	The min value of the robot pose's Z that the real robot TCP is allowed to reach.

Basic Move Settings

Basic move settings of move-type Steps are used to set the **velocity** and **motion type** for the robot when it moves to the waypoint.

▼ *Pick or place*

Unspecified	Default value
Pick	Please select this option when the move-type Step is before "Vision Move".
Place	Please select this option when the move-type Step is after "Vision Move".

Setting this parameter facilitates checking the logic of the Mech-Viz project. Please select a proper option according to the on-site situation and ensure that the Step used to pick is placed before the Step used to place.

▼ *Motion type*

Joint move	Joint motion, which guides the robot to move in a curved path. It is less likely to reach singularities in the path for joint motion. This motion type is applicable to scenarios where the requirement of path accuracy is not strict and the robot moves in a large space.
Linear move	Linear motion, which guides the robot to move linearly. This motion type is applicable to scenarios where there is a strict requirement for path accuracy, such as welding, gluing, and certain types of picking.

▼ *Velocity & Acceleration*

Velocity and acceleration determine how fast the robot can move. Usually, the set acceleration should be lower than the velocity. When the set acceleration is higher than the velocity, the robot will move in a choppy way.

> ! The velocities of Vision Move and its prior and subsequent Steps should be relatively low to ensure that the objects can be picked steadily.

▼ *Blend radius*

Default value	50.00 mm

Notes

Usually, the default setting can be used.

The blend radius refers to the distance between the target point and the point where the robot starts to turn. The larger the blend radius, the more smoother the robot motion transitions are. If the robot moves in a relatively small space, please set the blend radius to a smaller value.

If the robot moves in a relatively large space without obstacles and the distance between two consecutive path segments is long, please set the blend radius to a larger value.

JPs Constraint

▼ *Definition of Terms*

Shoulder

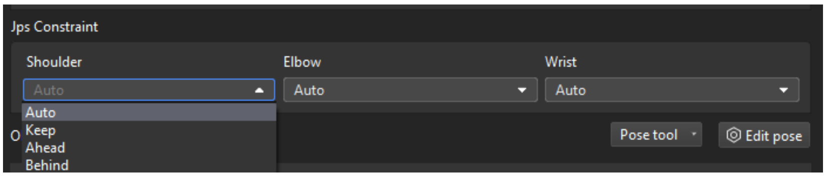
The relative positional relationship between the center of the wrist and Axis1. Axis1 refers to the rotation center axis of the robot axis 1.

Elbow

Relative positional relationship between the wrist and lower arm, which refers to the line connecting the rotation centers of the robot axis 2 and axis 3.

Wrist

Wrist refers to axis 5 of the robot. The negative wrist JPs suggests that the robot's wrist can rotate in the negative direction.



▼ *Value list*

Auto

There is no constraints on the motion of this joint. The optimal solution is the one in which each axis rotates the least.

Keep

Constrain the next JPs solution according to the current JPs status. For example, if the current JP of the axis 3 is positive, only solutions in which the JP of axis 3 is positive will be considered for the next waypoint.

Ahead

The wrist center is ahead of axis 1.

Behind

The wrist center is behind axis 1.

After clicking Show, a window showing all the JPs solutions to the current pose will pop up. After clicking one of the solutions, the corresponding pose of the solution will be displayed in the 3D simulation area. Therefore, you will know the possible solutions under different constraints, as shown below.

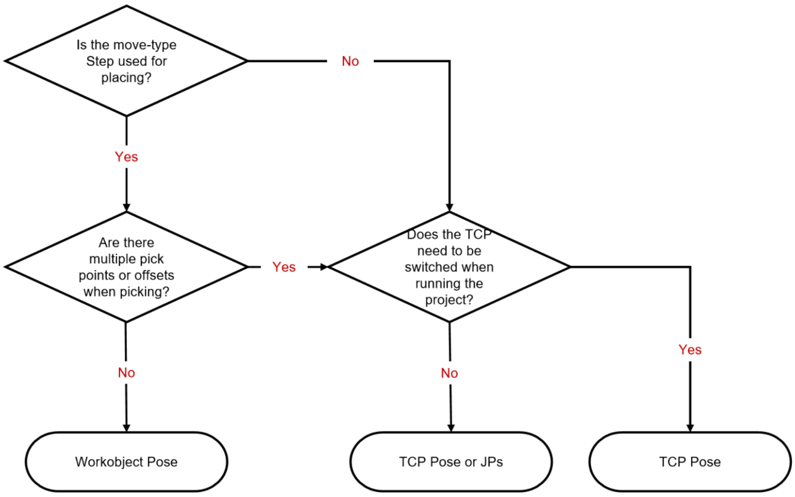
1. JPs constraint settings only apply to 6-axis robots. It is considered that 4-axis robots do not have abnormal shoulder, elbow, or wrist positions.
2. Relative Move, Custom Pallet Pattern, and Predefined Pallet Pattern do not support JPs constraint settings. There are default JPs constraint settings for these Steps, that is, the statuses of the shoulder, elbow, and wrist will not change, and the robot's motion does not cross solution systems (does not go through singularities).

Waypoint Type

▼ *Waypoint Type Description*

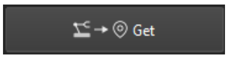
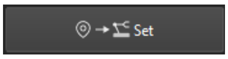
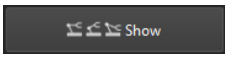
Tool	The waypoint will be represented by the X, Y, Z values, Euler angles, quaternions or rotation vectors in the tool reference frame. The tool pose is the pose of the tool, which is described as a relative pose of the tool reference frame to the robot reference frame. This description is in line with the task space, focusing on describing the position and orientation of the tool (such as truss arms, welding torch) when the robot is performing tasks.
JPs	The waypoint will be represented by the joint positions of the robot. Joint positions describe the angle or position of each joint of the robot. This description can represent the state of each joint in line with the hardware structure of the robot.
Workobject pose	The waypoint will be represented by the X, Y, Z values, Euler angles, quaternions or rotation vectors in the object reference frame. It focuses on describing the position and orientation of the actual workobjects that need to be operated or handled when the robot is performing tasks. Workobject pose is the pose of a certain point on the workobject relative to the robot reference frame.

The method to determine the waypoint type is shown in the figure below.


Is the move-type Step used for placing?
No
Yes
Are there multiple pick points or offsets when picking?
Yes
Does the TCP need to be switched when running the project?
Yes
No
No
Workobject Pose
TCP Pose or JPs
TCP Pose

▼ *Parameter Description*

Edit pose	Enter quaternions or Euler angles directly to adjust the pose. You can also copy and paste the pose.
Transform pose	Transform the current pose to a new one. It is suitable for fine-tuning.
Rectify pose	Set the coordinate of P1, P2, and P3 according to the instruction, and then rectify the object pose with the three-point method. It is suitable for scenarios where the objects are tilted and their poses cannot be easily determined. For instance, when a cuboid tilts, its pose is hard to determine. Rectify pose can be used here to calculate the pose of the cuboid and therefore the robot can move to the rectified pose.

Edit JPs	Similar to Edit Pose above; edit the JPs by copying and pasting or editing the JPs in either radians or degrees, which can be switched based on the actual requirement.
	Click to move the robot back to the specified waypoint.
	Get the pose of the simulated robot and set it as the waypoint.
	Show all the JPs solutions to the current waypoint.

5.5.1.3. Fixed-Point Move

Function

This Step specifies a waypoint in the robot motion path and the way to move to the waypoint.

Parameter Description

Move-Type Step Common Parameters

▼ *Send Waypoint*

Selected by default to send waypoint poses to the receiver, such as the robot. When this option is unselected, the waypoint pose will not be sent. However, the waypoint will remain in the planned path.

▼ *Try Continuously Running through Succeeding Non-Moves*

Unselected by default. When **non-move Steps**, such as Vision Look, Set DO, Check DI, etc., are connected between **move-type Steps**, the robot's path planning will be interrupted, and the actual robot will take a short pause, reducing the smoothness of running.

When this option is selected, the project will continue to run without waiting for the current move-type Step to complete execution, and therefore the robot can move in a smooth way without pauses. However, enabling this option may cause the execution of the Step to end prematurely.

Why will this option cause the execution of the Step to end prematurely?

Mech-Viz will send multiple poses simultaneously to the robot when the project is running. When the currently returned JPs of the robot correspond to the last pose sent by Mech-Viz, Mech-Viz will assume that the robot has moved to the last position.

For example, there are 10 move-type Steps in a path, and the pose of the 5th move-type Step is the same as that of the last move-type Step. When the robot moves at low speed, it sends JPs to Mech-Viz when it moves to the 5th move-type Step, Mech-Viz may mistakenly determine that the robot has finished the move-type Steps and prematurely ends the Steps since the poses of the 5th move-type Step and the last move-type Step are the same in the path.

▼ *Do Not Check Collision with Placed Workobject*

Unselected by default, namely that the collision with the already placed objects will not be detected. When this option is selected, the collisions between the robot, end tool, and placed objects will be detected.

In palletizing scenarios, the two possible cases of error are as follows:

1. When the robot is placing a carton, the robot may come into light contact with the placed cartons while no deformation will be caused. After Mech-Viz detects this collision in simulation, it will plan other positions for placing the carton, and therefore a full stack cannot be formed.
2. Usually, the TCP of a suction cup is inside the suction cup model instead of on the surface of it. Under this circumstance, the suction cup may be embedded in the model of the picked carton in the simulation of picking, while the software does not detect the collision between the end tool and the picked object. After the robot places the object and the carton model turns into an object model in the scene, a collision between the suction cup and the carton will be detected and the palletizing cannot be completed.

When this option is selected, no collision between the robot, end tool, and the placed object will be detected, and the above two cases of errors can be avoided.

▼ *Point Cloud Collision Detection Mode*

Select the proper mode according to the requirement of the on-site situation. Usually, the default setting **Auto** can be used. **Do not check collision** mode can be used in move-type Steps before the robot picks the object, and **Check collision** mode can be used after the robot picks the object.

Auto	Default setting. Collision with point cloud is checked only for the “Vision Move” Step and the “Relative Move” Step that depends on the “Vision Move” Step, but not for all move-type Steps.
Do not check collision	Point cloud collisions for all move-type Steps will not be detected.
Check collision	Point cloud collisions for all move-type Steps will be detected.

When Collisions › Collision detection configuration › Detect collision between point cloud and others is switched on, Mech-Viz will detect collisions between the robot model, end tool model, and point cloud when planning the path. Normally, detects whether the robot collides with other objects during picking and placing. When there are point cloud outliers, non-exiting collisions will be detected, which leads to errors in path planning.

▼ *Ignore Workobject Symmetry*

This parameter will only take effect when **Waypoint type** of the Step is set to **Workobject pose**.

None	Default setting, i.e., do not disable symmetry on any axis.
Around workobject frame Z axis	Only disable symmetry on Z-axis of the workobject reference frame.
Around workobject frame X&Y axis	Disable symmetry on X-axis and Y-axis of the workobject reference frame.
Around all axes	Once the object symmetry is disabled, the robot will place the objects strictly according to the workobject poses.

In some special cases, objects are not pickable due to their peculiar poses. Setting **Rotational symmetry** under Workobjects › Workobject Configuration in Resources may solve this problem. Based on the settings of Rotational symmetry, when the default object pose is not pickable, Mech-Viz will calculate candidate poses and determine whether the candidate poses are pickable. As the candidate poses calculated based on the settings of Rotational symmetry are different from the original poses output from Mech-Vision, the consistency of the objects’ place poses cannot be guaranteed.

▼ *Plan Failure Out Port*

After selecting this parameter, this Step will add “Plan failure” out port.

During the planning process, planning is carried out along the branch after the “Success” out port; if the planning fails in the current Step, the branch process after the “Planning failure” out port will be executed.

Held Workobject Collision Detection Settings

▼ *Do Not Check Collision with Scene Object/Robot*

Unselected by default. Once this option is selected, the collisions between the **held workobject** with the **scene objects or robot** will not be detected, and therefore the calculation workload of collision detection will be reduced, the planning speed can be increased, and the cycle time can be shortened. It is usually enabled in the first one or two move-type Steps after the robot picks the object.

Please enable this option cautiously as there may be collision risks.

When **Detect collision between held workobject and others** under Collision detection configuration › Configuration on held workobject is enabled, the software will detect whether the model of the held object collides with the models of the scene objects and the robot.

In palletizing projects, the calculated carton dimensions have millimeter-level errors with the actual dimensions, and frictions between cartons may occur during picking but no collisions will occur. For some move-type Steps that will obviously not cause collisions, detecting such collisions only adds to the calculation workload and planning time, and consequently extending the cycle time. In palletizing projects, enabling Do Not Check Collision with Scene Objects does not affect the collision detection between the held carton and the placed cartons. This option can be enabled when there are scene objects under the stack to avoid failure of finding the palletizing solution.

▼ *Do Not Check Collision with Point Cloud*

Unselected by default. Once this option is selected, the collisions between the **held workobject** with the **point clouds in the scene** will not be detected, and therefore the calculation workload of collision detection will be reduced, the planning speed can be increased, and the cycle time can be shortened.

- When both **Detect collision between held workobject and others** under Collision detection configuration › Configuration on held workobject and **Configuration on point cloud ¬ Detect collision between point cloud and others** are enabled, the software will detect whether the model of the held workobject will collide with the point cloud in the scene.
- When Mech-Vision sends the point cloud and object model to Mech-Viz, the point cloud and the object model are fitted together. After the robot picks the object, the model moves along the planned path, and the collision between the model of the held workobject and the point cloud will occur.
- It is known that the model of the held workobject will have false collisions with the point cloud. Detecting such collisions unnecessarily adds to the calculation workload and extends the planning time.

Basic Move Settings

Basic move settings of move-type Steps are used to set the **velocity** and **motion type** for the robot when it moves to the waypoint.

▼ *Pick or place*

Unspecified	Default value
Pick	Please select this option when the move-type Step is before "Vision Move".
Place	Please select this option when the move-type Step is after "Vision Move".

Setting this parameter facilitates checking the logic of the Mech-Viz project. Please select a proper option according to the on-site situation and ensure that the Step used to pick is placed before the Step used to place.

▼ *Motion type*

Joint move	Joint motion, which guides the robot to move in a curved path. It is less likely to reach singularities in the path for joint motion. This motion type is applicable to scenarios where the requirement of path accuracy is not strict and the robot moves in a large space.
Linear move	Linear motion, which guides the robot to move linearly. This motion type is applicable to scenarios where there is a strict requirement for path accuracy, such as welding, gluing, and certain types of picking.

▼ *Velocity & Acceleration*

Velocity and acceleration determine how fast the robot can move. Usually, the set acceleration should be lower than the velocity. When the set acceleration is higher than the velocity, the robot will move in a choppy way.

> ❗ The velocities of Vision Move and its prior and subsequent Steps should be relatively low to ensure that the objects can be picked steadily.

▼ *Blend radius*

Default value	50.00 mm
Notes	Usually, the default setting can be used. The blend radius refers to the distance between the target point and the point where the robot starts to turn. The larger the blend radius, the more smoother the robot motion transitions are. If the robot moves in a relatively small space, please set the blend radius to a smaller value. If the robot moves in a relatively large space without obstacles and the distance between two consecutive path segments is long, please set the blend radius to a larger value.

Smart Obstacle Avoidance

This function provides multiple alternative waypoints that are close to the waypoint of a move-type Step, which increases the number of feasible paths planned by Mech-Viz and the success rate of planning and reduces the difficulty of debugging.

Obstacle Avoidance Approach

You can select the **Obstacle Avoidance Approach** according to the requirements of the speed and success rate of path planning.

Disabled

Disable the Obstacle Avoidance Approach.

Speed first

Plan the path with speed priority. A minimum number of alternative waypoints will be generated.

Success rate first

Plan the path with success rate priority. A maximum number of alternative waypoints will be generated.

Trade-off

Balance the speed and success rate to plan the path. A moderate number of alternative waypoints will be generated.

Position Adjustment Range Radius

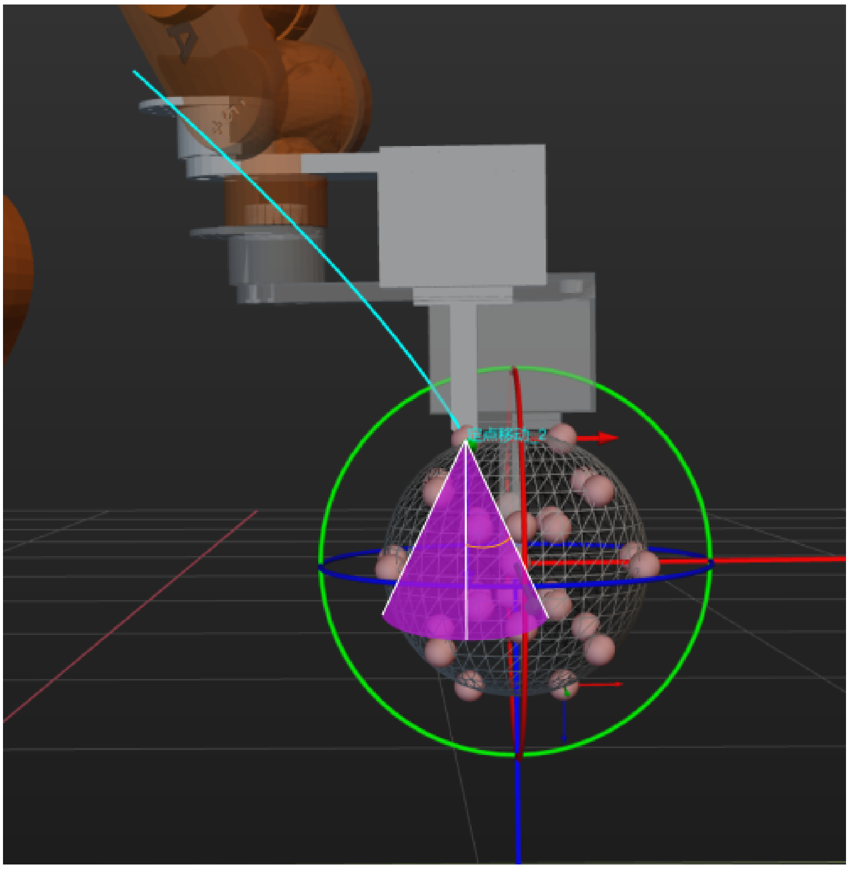
Default value: 5 mm

The alternative waypoints will be generated within the sphere with the set value as the radius.

Orientation Adjustment Angle Range

The default value is 0.00°, and the adjustment range is 0°–79°.

The TCP will be adjusted within the range of the alternative waypoints.

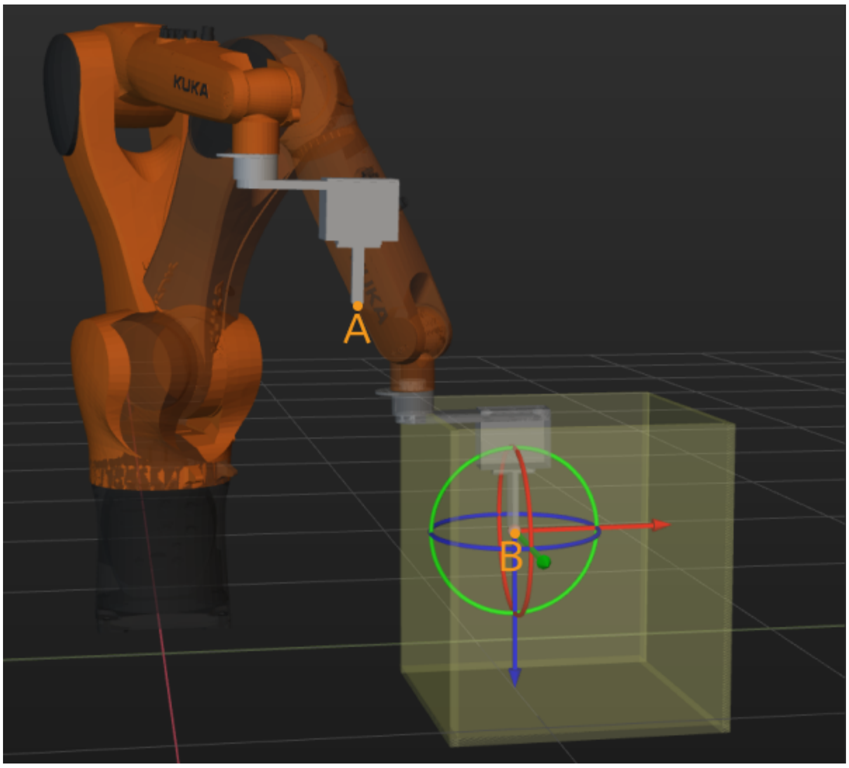


Show All Candidate Poses

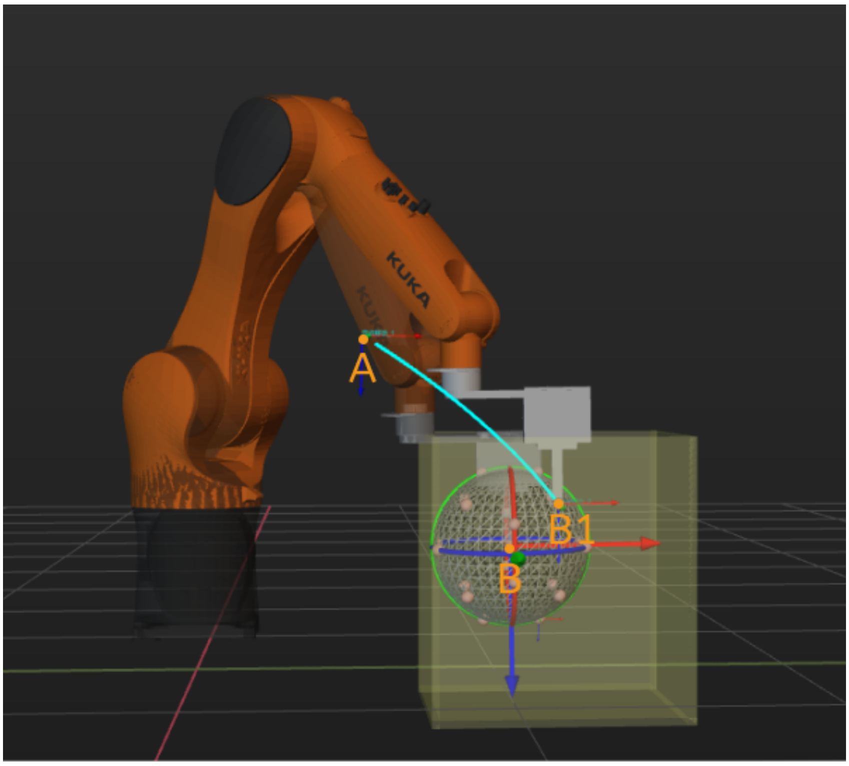
Selected by default to show all alternative waypoints. Unselect to hide all alternative waypoints.

Application Examples

As shown in the figure below, the robot tool will collide with the bin when the robot moves from Waypoint A to Waypoint B.



After enabling the Smart Obstacle Avoidance at Waypoint B, Mech-Viz will automatically select the alternative Waypoint B1 to avoid the collision, as shown below.



5.5.1.4. Move by Grid

Function

This Step is used to set multiple **waypoints** in the path according to the grid, specify the **way** to move to those waypoints, and guide the robot to move to the waypoints in sequence.

Parameter Description

Move-Type Step Common Parameters

▼ *Send Waypoint*

Selected by default to send waypoint poses to the receiver, such as the robot. When this option is unselected, the waypoint pose will not be sent. However, the waypoint will remain in the planned path.

▼ *Try Continuously Running through Succeeding Non-Moves*

Unselected by default. When **non-move Steps**, such as Vision Look, Set DO, Check DI, etc., are connected between **move-type Steps**, the robot's path planning will be interrupted, and the actual robot will take a short pause, reducing the smoothness of running.

When this option is selected, the project will continue to run without waiting for the current move-type Step to complete execution, and therefore the robot can move in a smooth way without pauses. However, enabling this option may cause the execution of the Step to end prematurely.

Why will this option cause the execution of the Step to end prematurely?

Mech-Viz will send multiple poses simultaneously to the robot when the project is running. When the currently returned JPs of the robot correspond to the last pose sent by Mech-Viz, Mech-Viz will assume that the robot has moved to the last position.

For example, there are 10 move-type Steps in a path, and the pose of the 5th move-type Step is the same as that of the last move-type Step. When the robot moves at low speed, it sends JPs to Mech-Viz when it moves to the 5th move-type Step, Mech-Viz may mistakenly determine that the robot has finished the move-type Steps and prematurely ends the Steps since the poses of the 5th move-type Step and the last move-type Step are the same in the path.

▼ *Do Not Check Collision with Placed Workobject*

Unselected by default, namely that the collision with the already placed objects will not be detected. When this option is selected, the collisions between the robot, end tool, and placed objects will be detected.

In palletizing scenarios, the two possible cases of error are as follows:

1. When the robot is placing a carton, the robot may come into light contact with the placed cartons while no deformation will be caused. After Mech-Viz detects this collision in simulation, it will plan other positions for placing the carton, and therefore a full stack cannot be formed.
2. Usually, the TCP of a suction cup is inside the suction cup model instead of on the surface of it. Under this circumstance, the suction cup may be embedded in the model of the picked carton in the simulation of picking, while the software does not detect the collision between the end tool and the picked object. After the robot places the object and the carton model turns into an object model in the scene, a collision between the suction cup and the carton

will be detected and the palletizing cannot be completed.

When this option is selected, no collision between the robot, end tool, and the placed object will be detected, and the above two cases of errors can be avoided.

▼ *Point Cloud Collision Detection Mode*

Select the proper mode according to the requirement of the on-site situation. Usually, the default setting **Auto** can be used. **Do not check collision** mode can be used in move-type Steps before the robot picks the object, and **Check collision** mode can be used after the robot picks the object.

Auto	Default setting. Collision with point cloud is checked only for the “Vision Move” Step and the “Relative Move” Step that depends on the “Vision Move” Step, but not for all move-type Steps.
Do not check collision	Point cloud collisions for all move-type Steps will not be detected.
Check collision	Point cloud collisions for all move-type Steps will be detected.

When Collisions › Collision detection configuration › Detect collision between point cloud and others is switched on, Mech-Viz will detect collisions between the robot model, end tool model, and point cloud when planning the path. Normally, detects whether the robot collides with other objects during picking and placing. When there are point cloud outliers, non-exiting collisions will be detected, which leads to errors in path planning.

▼ *Ignore Workobject Symmetry*

This parameter will only take effect when **Waypoint type** of the Step is set to **Workobject pose**.

None	Default setting, i.e., do not disable symmetry on any axis.
Around workobject frame Z axis	Only disable symmetry on Z-axis of the workobject reference frame.
Around workobject frame X&Y axis	Disable symmetry on X-axis and Y-axis of the workobject reference frame.
Around all axes	Once the object symmetry is disabled, the robot will place the objects strictly according to the workobject poses.

In some special cases, objects are not pickable due to their peculiar poses. Setting **Rotational symmetry** under Workobjects › Workobject Configuration in Resources may solve this problem. Based on the settings of Rotational symmetry, when the default object pose is not pickable, Mech-Viz will calculate candidate poses and determine whether the candidate poses are pickable. As the candidate poses calculated based on the settings of Rotational symmetry are different from the original poses output from Mech-Vision, the consistency of the objects’ place poses cannot be guaranteed.

▼ *Plan Failure Out Port*

After selecting this parameter, this Step will add “Plan failure” out port.

During the planning process, planning is carried out along the branch after the “Success” out port; if the planning fails in the current Step, the branch process after the “Planning failure” out port will be executed.

Held Workobject Collision Detection Settings

▼ *Do Not Check Collision with Scene Object/Robot*

Unselected by default. Once this option is selected, the collisions between the **held workobject** with the **scene objects or robot** will not be detected, and therefore the calculation workload of collision detection will be reduced, the planning speed can be increased, and the cycle time can be shortened. It is usually enabled in the first one or two move-type Steps after the robot picks the object.

Please enable this option cautiously as there may be collision risks.

When **Detect collision between held workobject and others** under Collision detection configuration › Configuration on held workobject is enabled, the software will detect whether the model of the held object collides with the models of the scene objects and the robot.

In palletizing projects, the calculated carton dimensions have millimeter-level errors with the actual dimensions, and frictions between cartons may occur during picking but no collisions will occur. For some move-type Steps that will obviously not cause collisions, detecting such collisions only adds to the calculation workload and planning time, and consequently extending the cycle time. In palletizing projects, enabling Do Not Check Collision with Scene Objects does not affect the collision detection between the held carton and the placed cartons. This option can be enabled when there are scene objects under the stack to avoid failure of finding the palletizing solution.

▼ *Do Not Check Collision with Point Cloud*

Unselected by default. Once this option is selected, the collisions between the **held workobject** with the **point clouds in the scene** will not be detected, and therefore the calculation workload of collision detection will be reduced, the planning speed can be increased, and the cycle time can be shortened.

- When both **Detect collision between held workobject and others** under Collision detection configuration › Configuration on held workobject and **Configuration on point cloud ¬ Detect collision between point cloud and others** are enabled, the software will detect whether the model of the held workobject will collide with the point cloud in the scene.
- When Mech-Vision sends the point cloud and object model to Mech-Viz, the point cloud and the object model are fitted together. After the robot picks the object, the model moves along the planned path, and the collision between the model of the held workobject and the point cloud will occur.
- It is known that the model of the held workobject will have false collisions with the point cloud. Detecting such collisions unnecessarily adds to the calculation workload and extends the planning time.

> - **Do Not Check Collision with Robot** is usually used in the first or second move-type Steps after the robot picks the object. It may increase the risk of collision. Please enable it with caution.
> - Enabling **Do Not Check Collision with Scene Objects** will not affect the collision detection between the held carton and the placed cartons. When there are scene objects under the pallet, this function can be used to avoid solution selection failure for palletizing.

Index

Starting Index

The index of the waypoint to be executed. Example: The index of the first waypoint is 0. If you want the robot to move from the third waypoint in the grid, please set the value to 2.

Current Index

The index of the waypoint currently being executed.

Grid

This parameter specifies the number and spacing of the current grid points in each direction.

X Count

The number of waypoints in the X direction of the grid.

X Space	The spacing between two waypoints in the X direction of the grid.
Y Count	The number of waypoints in the Y direction of the grid.
Y Space	The spacing between two waypoints in the Y direction of the grid.

Basic Move Settings

Basic move settings of move-type Steps are used to set the **velocity** and **motion type** for the robot when it moves to the waypoint.

▼ *Pick or place*

Unspecified	Default value
Pick	Please select this option when the move-type Step is before “Vision Move”.
Place	Please select this option when the move-type Step is after “Vision Move”.

Setting this parameter facilitates checking the logic of the Mech-Viz project. Please select a proper option according to the on-site situation and ensure that the Step used to pick is placed before the Step used to place.

▼ *Motion type*

Joint move	Joint motion, which guides the robot to move in a curved path. It is less likely to reach singularities in the path for joint motion. This motion type is applicable to scenarios where the requirement of path accuracy is not strict and the robot moves in a large space.
Linear move	Linear motion, which guides the robot to move linearly. This motion type is applicable to scenarios where there is a strict requirement for path accuracy, such as welding, gluing, and certain types of picking.

▼ *Velocity & Acceleration*

Velocity and acceleration determine how fast the robot can move. Usually, the set acceleration should be lower than the velocity. When the set acceleration is higher than the velocity, the robot will move in a choppy way.

> ❗ The velocities of Vision Move and its prior and subsequent Steps should be relatively low to ensure that the objects can be picked steadily.

▼ *Blend radius*

Default value	50.00 mm
Notes	Usually, the default setting can be used. The blend radius refers to the distance between the target point and the point where the robot starts to turn. The larger the blend radius, the more smoother the robot motion transitions are. If the robot moves in a relatively small space, please set the blend radius to a smaller value. If the robot moves in a relatively large space without obstacles and the distance between two consecutive path segments is long, please set the blend radius to a larger value.

Show Base

Once this parameter is selected, the grid waypoints and the corresponding index will be displayed in the 3D simulation area of Mech-Viz.

Show Target

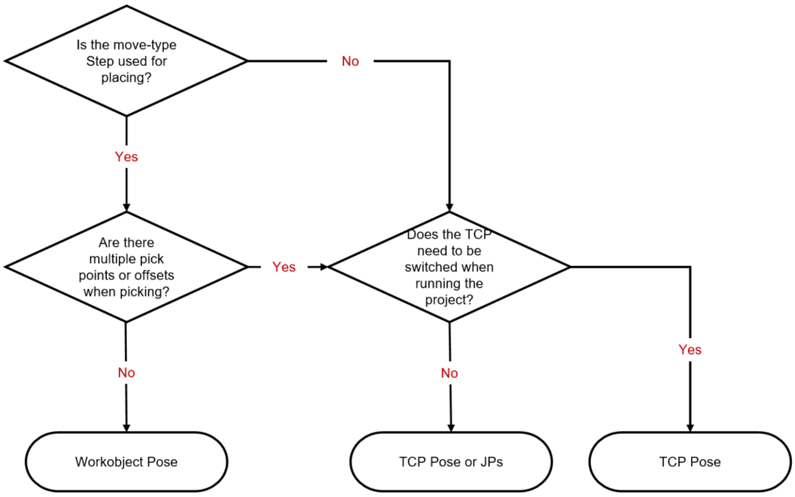
Once this parameter is selected, the robot and workobject pose will be displayed in the 3D simulation area of Mech-Viz, while the grid will not be displayed.

Waypoint Type

▼ *Waypoint Type Description*

Tool	The waypoint will be represented by the X, Y, Z values, Euler angles, quaternions or rotation vectors in the tool reference frame. The tool pose is the pose of the tool, which is described as a relative pose of the tool reference frame to the robot reference frame. This description is in line with the task space, focusing on describing the position and orientation of the tool (such as truss arms, welding torch) when the robot is performing tasks.
JPs	The waypoint will be represented by the joint positions of the robot. Joint positions describe the angle or position of each joint of the robot. This description can represent the state of each joint in line with the hardware structure of the robot.
Workobject pose	The waypoint will be represented by the X, Y, Z values, Euler angles, quaternions or rotation vectors in the object reference frame. It focuses on describing the position and orientation of the actual workobjects that need to be operated or handled when the robot is performing tasks. Workobject pose is the pose of a certain point on the workobject relative to the robot reference frame.

The method to determine the waypoint type is shown in the figure below.


Is the move-type Step used for placing?
No
Yes
Are there multiple pick points or offsets when picking?
Yes
Does the TCP need to be switched when running the project?
No
No
Yes
Workobject Pose
TCP Pose or JPs
TCP Pose

▼ *Parameter Description*

Edit pose	Enter quaternions or Euler angles directly to adjust the pose. You can also copy and paste the pose.
Transform pose	Transform the current pose to a new one. It is suitable for fine-tuning.
Rectify pose	Set the coordinate of P1, P2, and P3 according to the instruction, and then rectify the object pose with the three-point method. It is suitable for scenarios where the objects are tilted and their poses cannot be easily determined. For instance, when a cuboid tilts, its pose is hard to determine. Rectify pose can be used here to calculate the pose of the cuboid and therefore the robot can move to the rectified pose.
Edit JPs	Similar to Edit Pose above; edit the JPs by copying and pasting or editing the JPs in either radians or degrees, which can be switched based on the actual requirement.
Get	Click to move the robot back to the specified waypoint.
Set	Get the pose of the simulated robot and set it as the waypoint.
Show	Show all the JPs solutions to the current waypoint.

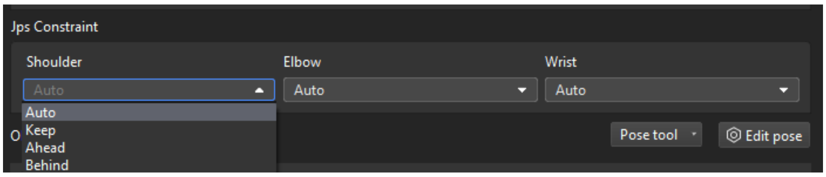
JPs Constraint

▼ *Definition of Terms*

Shoulder
The relative positional relationship between the center of the wrist and Axis1. Axis1 refers to the rotation center axis of the robot axis 1.

Elbow
Relative positional relationship between the wrist and lower arm, which refers to the line connecting the rotation centers of the robot axis 2 and axis 3.

Wrist
Wrist refers to axis 5 of the robot. The negative wrist JPs suggests that the robot’s wrist can rotate in the negative direction.



▼ *Value list*

Auto
There is no constraints on the motion of this joint. The optimal solution is the one in which each axis rotates the least.

Keep
Constrain the next JPs solution according to the current JPs status. For example, if the current JP of the axis 3 is positive, only solutions in which the JP of axis 3 is positive will be considered for the next waypoint.

Ahead
The wrist center is ahead of axis 1.

Behind
The wrist center is behind axis 1.

After clicking Show, a window showing all the JPs solutions to the current pose will pop up. After clicking one of the solutions, the corresponding pose of the solution will be displayed in the 3D simulation area. Therefore, you will know the possible solutions under different constraints, as shown below.

1. JPs constraint settings only apply to 6-axis robots. It is considered that 4-axis robots do not have abnormal shoulder, elbow, or wrist positions.
2. Relative Move, Custom Pallet Pattern, and Predefined Pallet Pattern do not support JPs constraint settings. There are default JPs constraint settings for these Steps, that is, the statuses of the shoulder, elbow, and wrist will not change, and the robot’s motion does not cross solution systems (does not go through singularities).

5.5.1.5. Move by List

Function

This Step is used to set multiple **waypoints** in the path according to the grid, specify the **way** to move to those waypoints, and guide the robot to move to the waypoints in sequence.

Parameter Description

Move-Type Step Common Parameters

▼ *Send Waypoint*

Selected by default to send waypoint poses to the receiver, such as the robot. When this option is unselected, the waypoint pose will not be sent. However, the waypoint will remain in the planned path.

▼ *Try Continuously Running through Succeeding Non-Moves*

Unselected by default. When **non-move Steps**, such as Vision Look, Set DO, Check DI, etc., are connected between **move-type Steps**, the robot’s path planning will be interrupted, and the actual robot will take a short pause, reducing the smoothness of running.

When this option is selected, the project will continue to run without waiting for the current move-type Step to complete execution, and therefore the robot can move in a smooth way without pauses. However, enabling this option may cause the execution of the Step to end prematurely.

Why will this option cause the execution of the Step to end prematurely?

Mech-Viz will send multiple poses simultaneously to the robot when the project is running. When the currently returned JPs of the robot correspond to the last pose sent by Mech-Viz, Mech-Viz will assume that the robot has moved to the last position.

For example, there are 10 move-type Steps in a path, and the pose of the 5th move-type Step is the same as that of the last move-type Step. When the robot moves at low speed, it sends JPs to Mech-Viz when it moves to the 5th move-type Step, Mech-Viz may mistakenly determine that the robot has finished the move-type Steps and prematurely ends the Steps since the poses of the 5th move-type Step and the last move-type Step are the same in the path.

▼ *Do Not Check Collision with Placed Workobject*

Unselected by default, namely that the collision with the already placed objects will not be detected. When this option is selected, the collisions between the robot, end tool, and placed objects will be detected.

In palletizing scenarios, the two possible cases of error are as follows:

1. When the robot is placing a carton, the robot may come into light contact with the placed cartons while no deformation will be caused. After Mech-Viz detects this collision in simulation, it will plan other positions for placing the carton, and therefore a full stack cannot be formed.
2. Usually, the TCP of a suction cup is inside the suction cup model instead of on the surface of it. Under this circumstance, the suction cup may be embedded in the model of the picked carton in the simulation of picking, while the software does not detect the collision between the end tool and the picked object. After the robot places the object and the carton model turns into an object model in the scene, a collision between the suction cup and the carton will be detected and the palletizing cannot be completed.

When this option is selected, no collision between the robot, end tool, and the placed object will be detected, and the above two cases of errors can be avoided.

▼ *Point Cloud Collision Detection Mode*

Select the proper mode according to the requirement of the on-site situation. Usually, the default setting **Auto** can be used. **Do not check collision** mode can be used in move-type Steps before the robot picks the object, and **Check collision** mode can be used after the robot picks the object.

Auto	Default setting. Collision with point cloud is checked only for the “Vision Move” Step and the “Relative Move” Step that depends on the “Vision Move” Step, but not for all move-type Steps.
Do not check collision	Point cloud collisions for all move-type Steps will not be detected.
Check collision	Point cloud collisions for all move-type Steps will be detected.

! When Collisions › Collision detection configuration › Detect collision between point cloud and others is switched on, Mech-Viz will detect collisions between the robot model, end tool model, and point cloud when planning the path. Normally, detects whether the robot collides with other objects during picking and placing. When there are point cloud outliers, non-exiting collisions will be detected, which leads to errors in path planning.

▼ *Ignore Workobject Symmetry*

This parameter will only take effect when **Waypoint type** of the Step is set to **Workobject pose**.

None	Default setting, i.e., do not disable symmetry on any axis.
Around workobject frame Z axis	Only disable symmetry on Z-axis of the workobject reference frame.
Around workobject frame X&Y axis	Disable symmetry on X-axis and Y-axis of the workobject reference frame.
Around all axes	Once the object symmetry is disabled, the robot will place the objects strictly according to the workobject poses.

In some special cases, objects are not pickable due to their peculiar poses. Setting **Rotational symmetry** under Workobjects › Workobject Configuration in Resources may solve this problem. Based on the settings of Rotational symmetry, when the default object pose is not pickable, Mech-Viz will calculate candidate poses and determine whether the candidate poses are pickable. As the candidate poses calculated based on the settings of Rotational symmetry are different from the original poses output from Mech-Vision, the consistency of the objects' place poses cannot be guaranteed.

▼ *Plan Failure Out Port*

After selecting this parameter, this Step will add “Plan failure” out port.

During the planning process, planning is carried out along the branch after the “Success” out port; if the planning fails in the current Step, the branch process after the “Planning failure” out port will be executed.

Held Workobject Collision Detection Settings

▼ *Do Not Check Collision with Scene Object/Robot*

Unselected by default. Once this option is selected, the collisions between the **held workobject** with the **scene objects or robot** will not be detected, and therefore the calculation workload of collision detection will be reduced, the planning speed can be increased, and the cycle time can be shortened. It is usually enabled in the first one or two move-type Steps after the robot picks the object.

Please enable this option cautiously as there may be collision risks.

When **Detect collision between held workobject and others** under Collision detection configuration › Configuration on held workobject is enabled, the software will detect whether the model of the held object collides with the models of the scene objects and the robot.

In palletizing projects, the calculated carton dimensions have millimeter-level errors with the actual dimensions, and frictions between cartons may occur during picking but no collisions will occur. For some move-type Steps that will obviously not cause collisions, detecting such collisions only adds to the calculation workload and planning time, and consequently extending the cycle time. In palletizing projects, enabling Do Not Check Collision with Scene Objects does not affect the collision detection between the held carton and the placed cartons. This option can be enabled when there are scene objects under the stack to avoid failure of finding the palletizing solution.

▼ *Do Not Check Collision with Point Cloud*

Unselected by default. Once this option is selected, the collisions between the **held workobject** with the **point clouds in the scene** will not be detected, and therefore the calculation workload of collision detection will be reduced, the planning speed can be increased, and the cycle time can be shortened.

- When both **Detect collision between held workobject and others** under Collision detection configuration › Configuration on held workobject and **Configuration on point cloud ¬ Detect collision between point cloud and others** are enabled, the software will detect whether the model of the held workobject will collide with the point cloud in the scene.
- When Mech-Vision sends the point cloud and object model to Mech-Viz, the point cloud and the object model are fitted together. After the robot picks the object, the model moves along the planned path, and the collision between the model of the held workobject and the point cloud will occur.
- It is known that the model of the held workobject will have false collisions with the point cloud. Detecting such collisions unnecessarily adds to the calculation workload and extends the planning time.

> **Do Not Check Collision with Robot** is usually used in the first or second move-type Steps after the robot picks the object. It may increase the risk of collision. Please enable it with caution. Enabling **Do Not Check Collision with Scene Objects** will not affect the collision detection between the held carton and the placed cartons. When there are scene objects under the pallet, this function can be used to avoid solution selection failure for palletizing.

Index

Starting Index

The index of the waypoint to be executed.

Example: The index of the first waypoint is 0. If you want the robot to move from the third waypoint in the grid, please set the value to 2.

Current Index

The index of the waypoint currently being executed.

All Waypoints in One Move

Once this parameter is selected, the **Index** will be disabled. When the Step **Move by List** is executed in a Mech-Viz project, the robot will move through all waypoints in the list in sequence at once.

Use Basic Move Settings at Global Scale

Once this parameter is selected, all waypoints in the move list will use the same **Basic Move** parameters.

Basic Move Settings

Basic move settings of move-type Steps are used to set the **velocity** and **motion type** for the robot when it moves to the waypoint.

▼ *Pick or place*

Unspecified Default value

Pick Please select this option when the move-type Step is before “Vision Move”.

Place Please select this option when the move-type Step is after “Vision Move”.

Setting this parameter facilitates checking the logic of the Mech-Viz project. Please select a proper option according to the on-site situation and ensure that the Step used to pick is placed before the Step used to place.

▼ *Motion type*

Joint move Joint motion, which guides the robot to move in a curved path. It is less likely to reach singularities in the path for joint motion. This motion type is applicable to scenarios where the requirement of path accuracy is not strict and the robot moves in a large space.

Linear move Linear motion, which guides the robot to move linearly. This motion type is applicable to scenarios where there is a strict requirement for path accuracy, such as welding, gluing, and certain types of picking.

▼ *Velocity & Acceleration*

Velocity and acceleration determine how fast the robot can move. Usually, the set acceleration should be lower than the velocity. When the set acceleration is higher than the velocity, the robot will move in a choppy way.

> ❗ The velocities of Vision Move and its prior and subsequent Steps should be relatively low to ensure that the objects can be picked steadily.

▼ *Blend radius*

Default value 50.00 mm

Notes Usually, the default setting can be used.

The blend radius refers to the distance between the target point and the point where the robot starts to turn. The larger the blend radius, the more smoother the robot motion transitions are. If the robot moves in a relatively small space, please set the blend radius to a smaller value.

If the robot moves in a relatively large space without obstacles and the distance between two consecutive path segments is long, please set the blend radius to a larger value.

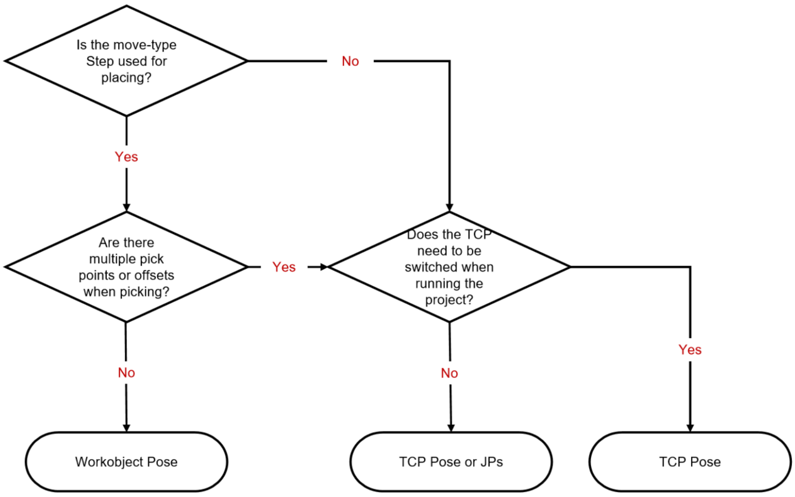
Waypoint Type

▼ *Waypoint Type Description*

Tool	The waypoint will be represented by the X, Y, Z values, Euler angles, quaternions or rotation vectors in the tool reference frame. The tool pose is the pose of the tool, which is described as a relative pose of the tool reference frame to the robot reference frame. This description is in line with the task space, focusing on describing the position and orientation of the tool (such as truss arms, welding torch) when the robot is performing tasks.

JPs	The waypoint will be represented by the joint positions of the robot. Joint positions describe the angle or position of each joint of the robot. This description can represent the state of each joint in line with the hardware structure of the robot.
Workobject pose	The waypoint will be represented by the X, Y, Z values, Euler angles, quaternions or rotation vectors in the object reference frame. It focuses on describing the position and orientation of the actual workobjects that need to be operated or handled when the robot is performing tasks. Workobject pose is the pose of a certain point on the workobject relative to the robot reference frame.

The method to determine the waypoint type is shown in the figure below.


Is the move-type Step used for placing?
No
Yes
Are there multiple pick points or offsets when picking?
Yes
Does the TCP need to be switched when running the project?
No
No
Yes
Workobject Pose
TCP Pose or JPs
TCP Pose

▼ *Parameter Description*

Edit pose	Enter quaternions or Euler angles directly to adjust the pose. You can also copy and paste the pose.
Transform pose	Transform the current pose to a new one. It is suitable for fine-tuning.
Rectify pose	Set the coordinate of P1, P2, and P3 according to the instruction, and then rectify the object pose with the three-point method. It is suitable for scenarios where the objects are tilted and their poses cannot be easily determined. For instance, when a cuboid tilts, its pose is hard to determine. Rectify pose can be used here to calculate the pose of the cuboid and therefore the robot can move to the rectified pose.
Edit JPs	Similar to Edit Pose above; edit the JPs by copying and pasting or editing the JPs in either radians or degrees, which can be switched based on the actual requirement.
Get	Click to move the robot back to the specified waypoint.
Set	Get the pose of the simulated robot and set it as the waypoint.
Show	Show all the JPs solutions to the current waypoint.

Move List

Add the waypoints in the **move list** in sequence manually and set corresponding parameters.

Click + to add a waypoint at the **end** of the list, and click - to delete the selected waypoint.

Add TCP list

You can add multiple waypoints as **TCP** in the list at once.

Input **TCP** as individual lines in the Add TCP list window.

5.5.1.6. External Move

Function

Mech-Viz obtains the tool pose from the external service, Adapter, and guides the robot to move to that pose.

Parameter Description

Move-Type Step Common Parameters

▼ *Send Waypoint*

Selected by default to send waypoint poses to the receiver, such as the robot. When this option is unselected, the waypoint pose will not be sent. However, the waypoint will remain in the planned path.

▼ *Try Continuously Running through Succeeding Non-Moves*

Unselected by default. When **non-move Steps**, such as Vision Look, Set DO, Check DI, etc., are connected between **move-type Steps**, the robot’s path planning will be interrupted, and the actual robot will take a short pause, reducing the smoothness of running.

When this option is selected, the project will continue to run without waiting for the current move-type Step to complete execution, and therefore the robot can move in a smooth way without pauses. However, enabling this option may cause the execution of the Step to end prematurely.

Why will this option cause the execution of the Step to end prematurely?

Mech-Viz will send multiple poses simultaneously to the robot when the project is running. When the currently returned JPs of the robot correspond to the last pose sent by Mech-Viz, Mech-Viz will assume that the robot has moved to the last position.

For example, there are 10 move-type Steps in a path, and the pose of the 5th move-type Step is the same as that of the last move-type Step. When the robot moves at low speed, it sends JPs to Mech-Viz when it moves to the 5th move-type Step, Mech-Viz may mistakenly determine that the robot has finished the move-type Steps and prematurely ends the Steps since the poses of the 5th move-type Step and the last move-type Step are the same in the path.

▼ *Do Not Check Collision with Placed Workobject*

Unselected by default, namely that the collision with the already placed objects will not be detected. When this option is selected, the collisions between the robot, end tool, and placed objects will be detected.

In palletizing scenarios, the two possible cases of error are as follows:

1. When the robot is placing a carton, the robot may come into light contact with the placed cartons while no deformation will be caused. After Mech-Viz detects this collision in simulation, it will plan other positions for placing the carton, and therefore a full stack cannot be formed.
2. Usually, the TCP of a suction cup is inside the suction cup model instead of on the surface of it. Under this circumstance, the suction cup may be embedded in the model of the picked carton in the simulation of picking, while the software does not detect the collision between the end tool and the picked object. After the robot places the object and the carton model turns into an object model in the scene, a collision between the suction cup and the carton will be detected and the palletizing cannot be completed.

When this option is selected, no collision between the robot, end tool, and the placed object will be detected, and the above two cases of errors can be avoided.

▼ *Point Cloud Collision Detection Mode*

Select the proper mode according to the requirement of the on-site situation. Usually, the default setting **Auto** can be used. **Do not check collision** mode can be used in move-type Steps before the robot picks the object, and **Check collision** mode can be used after the robot picks the object.

Auto	Default setting. Collision with point cloud is checked only for the “Vision Move” Step and the “Relative Move” Step that depends on the “Vision Move” Step, but not for all move-type Steps.
Do not check collision	Point cloud collisions for all move-type Steps will not be detected.
Check collision	Point cloud collisions for all move-type Steps will be detected.

❗ When Collisions › Collision detection configuration › Detect collision between point cloud and others is switched on, Mech-Viz will detect collisions between the robot model, end tool model, and point cloud when planning the path. Normally, detects whether the robot collides with other objects during picking and placing. When there are point cloud outliers, non-exiting collisions will be detected, which leads to errors in path planning.

▼ *Ignore Workobject Symmetry*

This parameter will only take effect when **Waypoint type** of the Step is set to **Workobject pose**.

None	Default setting, i.e., do not disable symmetry on any axis.
Around workobject frame Z axis	Only disable symmetry on Z-axis of the workobject reference frame.
Around workobject frame X&Y axis	Disable symmetry on X-axis and Y-axis of the workobject reference frame.
Around all axes	Once the object symmetry is disabled, the robot will place the objects strictly according to the workobject poses.

ℹ In some special cases, objects are not pickable due to their peculiar poses. Setting **Rotational symmetry** under Workobjects › Workobject Configuration in Resources may solve this problem. Based on the settings of Rotational symmetry, when the default object pose is not pickable, Mech-Viz will calculate candidate poses and determine whether the candidate poses are pickable. As the candidate poses calculated based on the settings of Rotational symmetry are different from the original poses output from Mech-Vision, the

> consistency of the objects' place poses cannot be guaranteed.

▼ *Plan Failure Out Port*

After selecting this parameter, this Step will add "Plan failure" out port.

During the planning process, planning is carried out along the branch after the "Success" out port; if the planning fails in the current Step, the branch process after the "Planning failure" out port will be executed.

Held Workobject Collision Detection Settings

▼ *Do Not Check Collision with Scene Object/Robot*

Unselected by default. Once this option is selected, the collisions between the **held workobject** with the **scene objects or robot** will not be detected, and therefore the calculation workload of collision detection will be reduced, the planning speed can be increased, and the cycle time can be shortened. It is usually enabled in the first one or two move-type Steps after the robot picks the object.

Please enable this option cautiously as there may be collision risks.

When **Detect collision between held workobject and others** under Collision detection configuration › Configuration on held workobject is enabled, the software will detect whether the model of the held object collides with the models of the scene objects and the robot.

In palletizing projects, the calculated carton dimensions have millimeter-level errors with the actual dimensions, and frictions between cartons may occur during picking but no collisions will occur. For some move-type Steps that will obviously not cause collisions, detecting such collisions only adds to the calculation workload and planning time, and consequently extending the cycle time. In palletizing projects, enabling Do Not Check Collision with Scene Objects does not affect the collision detection between the held carton and the placed cartons. This option can be enabled when there are scene objects under the stack to avoid failure of finding the palletizing solution.

▼ *Do Not Check Collision with Point Cloud*

Unselected by default. Once this option is selected, the collisions between the **held workobject** with the **point clouds in the scene** will not be detected, and therefore the calculation workload of collision detection will be reduced, the planning speed can be increased, and the cycle time can be shortened.

- When both **Detect collision between held workobject and others** under Collision detection configuration › Configuration on held workobject and **Configuration on point cloud ¬ Detect collision between point cloud and others** are enabled, the software will detect whether the model of the held workobject will collide with the point cloud in the scene.
- When Mech-Vision sends the point cloud and object model to Mech-Viz, the point cloud and the object model are fitted together. After the robot picks the object, the model moves along the planned path, and the collision between the model of the held workobject and the point cloud will occur.
- It is known that the model of the held workobject will have false collisions with the point cloud. Detecting such collisions unnecessarily adds to the calculation workload and extends the planning time.

Service Name

The Adapter’s external service name registered in Mech-Center, which is used to connect the interface of this Step and send waypoints. Please make sure that the service name is the same as the Adapter name.

Use JPs from External Service

Unselected by default. The software will use the previous waypoint as the initial position when planning the path. Once this option is selected, the initial position for path planning will be the joint positions obtained from external services. This function is usually used in the scenarios where Mech-Viz is not used for master-control of the robot.

5.5.1.7. Relative Move

Function

This Step guides the robot to move relative to a known waypoint of a move-type Step. You can set the relative move in different coordinate systems and set the rotational move in different directions.

Usage Scenario

It is usually used before or after the Vision Move Step to guide the robot to move along the Z direction of the vision point.

Parameter Description

Move-Type Step Common Parameters

▼ *Send Waypoint*

Selected by default to send waypoint poses to the receiver, such as the robot. When this option is unselected, the waypoint pose will not be sent. However, the waypoint will remain in the planned path.

▼ *Try Continuously Running through Succeeding Non-Moves*

Unselected by default. When **non-move Steps**, such as Vision Look, Set DO, Check DI, etc., are connected between **move-type Steps**, the robot’s path planning will be interrupted, and the actual robot will take a short pause, reducing the smoothness of running.

When this option is selected, the project will continue to run without waiting for the current move-type Step to complete execution, and therefore the robot can move in a smooth way without pauses. However, enabling this option may cause the execution of the Step to end prematurely.

Why will this option cause the execution of the Step to end prematurely?

Mech-Viz will send multiple poses simultaneously to the robot when the project is running. When the currently returned JPs of the robot correspond to the last pose sent by Mech-Viz, Mech-Viz will assume that the robot has moved to the last position.

For example, there are 10 move-type Steps in a path, and the pose of the 5th move-type Step is the same as that of the last move-type Step. When the robot moves at low speed, it sends JPs to Mech-Viz when it moves to the 5th move-type Step, Mech-Viz may mistakenly determine that the robot has finished the move-type Steps and prematurely ends the Steps since the poses of the 5th move-type Step and the last move-type Step are the same in the path.

▼ *Do Not Check Collision with Placed Workobject*

Unselected by default, namely that the collision with the already placed objects will not be detected. When this option is selected, the collisions between the robot, end tool, and placed objects will be detected.

In palletizing scenarios, the two possible cases of error are as follows:

1. When the robot is placing a carton, the robot may come into light contact with the placed cartons while no deformation will be caused. After Mech-Viz detects this collision in simulation, it will plan other positions for placing the carton, and therefore a full stack cannot be formed.
2. Usually, the TCP of a suction cup is inside the suction cup model instead of on the surface of it. Under this circumstance, the suction cup may be embedded in the model of the picked carton in the simulation of picking, while the software does not detect the collision between the end tool and the picked object. After the robot places the object and the carton model turns into an object model in the scene, a collision between the suction cup and the carton will be detected and the palletizing cannot be completed.

When this option is selected, no collision between the robot, end tool, and the placed object will be detected, and the above two cases of errors can be avoided.

▼ *Point Cloud Collision Detection Mode*

Select the proper mode according to the requirement of the on-site situation. Usually, the default setting **Auto** can be used. **Do not check collision** mode can be used in move-type Steps before the robot picks the object, and **Check collision** mode can be used after the robot picks the object.

Auto	Default setting. Collision with point cloud is checked only for the “Vision Move” Step and the “Relative Move” Step that depends on the “Vision Move” Step, but not for all move-type Steps.
Do not check collision	Point cloud collisions for all move-type Steps will not be detected.
Check collision	Point cloud collisions for all move-type Steps will be detected.

When Collisions › Collision detection configuration › Detect collision between point cloud and others is switched on, Mech-Viz will detect collisions between the robot model, end tool model, and point cloud when planning the path. Normally, detects whether the robot collides with other objects during picking and placing. When there are point cloud outliers, non-exiting collisions will be detected, which leads to errors in path planning.

▼ *Ignore Workobject Symmetry*

This parameter will only take effect when **Waypoint type** of the Step is set to **Workobject pose**.

None	Default setting, i.e., do not disable symmetry on any axis.
Around workobject frame Z axis	Only disable symmetry on Z-axis of the workobject reference frame.
Around workobject frame X&Y axis	Disable symmetry on X-axis and Y-axis of the workobject reference frame.

Around all axes	Once the object symmetry is disabled, the robot will place the objects strictly according to the workobject poses.

In some special cases, objects are not pickable due to their peculiar poses. Setting **Rotational symmetry** under Workobjects › Workobject Configuration in Resources may solve this problem. Based on the settings of Rotational symmetry, when the default object pose is not pickable, Mech-Viz will calculate candidate poses and determine whether the candidate poses are pickable. As the candidate poses calculated based on the settings of Rotational symmetry are different from the original poses output from Mech-Vision, the consistency of the objects' place poses cannot be guaranteed.

▼ *Plan Failure Out Port*

After selecting this parameter, this Step will add “Plan failure” out port.

During the planning process, planning is carried out along the branch after the “Success” out port; if the planning fails in the current Step, the branch process after the “Planning failure” out port will be executed.

Held Workobject Collision Detection Settings

▼ *Do Not Check Collision with Scene Object/Robot*

Unselected by default. Once this option is selected, the collisions between the **held workobject** with the **scene objects or robot** will not be detected, and therefore the calculation workload of collision detection will be reduced, the planning speed can be increased, and the cycle time can be shortened. It is usually enabled in the first one or two move-type Steps after the robot picks the object.

Please enable this option cautiously as there may be collision risks.

When **Detect collision between held workobject and others** under Collision detection configuration › Configuration on held workobject is enabled, the software will detect whether the model of the held object collides with the models of the scene objects and the robot.

In palletizing projects, the calculated carton dimensions have millimeter-level errors with the actual dimensions, and frictions between cartons may occur during picking but no collisions will occur. For some move-type Steps that will obviously not cause collisions, detecting such collisions only adds to the calculation workload and planning time, and consequently extending the cycle time. In palletizing projects, enabling Do Not Check Collision with Scene Objects does not affect the collision detection between the held carton and the placed cartons. This option can be enabled when there are scene objects under the stack to avoid failure of finding the palletizing solution.

▼ *Do Not Check Collision with Point Cloud*

Unselected by default. Once this option is selected, the collisions between the **held workobject** with the **point clouds in the scene** will not be detected, and therefore the calculation workload of collision detection will be reduced, the planning speed can be increased, and the cycle time can be shortened.

- When both **Detect collision between held workobject and others** under Collision detection configuration › Configuration on held workobject and **Configuration on point cloud ¬ Detect collision between point cloud and others** are enabled, the software will detect whether the model of the held workobject will collide with the point cloud in the scene.
- When Mech-Vision sends the point cloud and object model to Mech-Viz, the point cloud and the object model are fitted together. After the robot picks the object, the model moves along

the planned path, and the collision between the model of the held workobject and the point cloud will occur.

- It is known that the model of the held workobject will have false collisions with the point cloud. Detecting such collisions unnecessarily adds to the calculation workload and extends the planning time.

Basic Move Settings

Basic move settings of move-type Steps are used to set the **velocity** and **motion type** for the robot when it moves to the waypoint.

▼ *Pick or place*

Unspecified	Default value
Pick	Please select this option when the move-type Step is before “Vision Move”.
Place	Please select this option when the move-type Step is after “Vision Move”.

Setting this parameter facilitates checking the logic of the Mech-Viz project. Please select a proper option according to the on-site situation and ensure that the Step used to pick is placed before the Step used to place.

▼ *Motion type*

Joint move	Joint motion, which guides the robot to move in a curved path. It is less likely to reach singularities in the path for joint motion. This motion type is applicable to scenarios where the requirement of path accuracy is not strict and the robot moves in a large space.
Linear move	Linear motion, which guides the robot to move linearly. This motion type is applicable to scenarios where there is a strict requirement for path accuracy, such as welding, gluing, and certain types of picking.

▼ *Velocity & Acceleration*

Velocity and acceleration determine how fast the robot can move. Usually, the set acceleration should be lower than the velocity. When the set acceleration is higher than the velocity, the robot will move in a choppy way.

> ❗ The velocities of Vision Move and its prior and subsequent Steps should be relatively low to ensure that the objects can be picked steadily.

▼ *Blend radius*

Default value	50.00 mm
Notes	Usually, the default setting can be used. The blend radius refers to the distance between the target point and the point where the robot starts to turn. The larger the blend radius, the more smoother the robot motion transitions are. If the robot moves in a relatively small space, please set the blend radius to a smaller value. If the robot moves in a relatively large space without obstacles and the distance between two consecutive path segments is long, please set the blend radius to a larger value.

Relative Move Dependency

Value list: Previous waypoint, Next waypoint.

Instruction: When **Previous waypoint** is selected, the robot will move relative to the previous waypoint. When **Next waypoint** is selected, the robot will move relative to the next waypoint.

Waypoint type

Options: Tool, Robot, and Ref point.

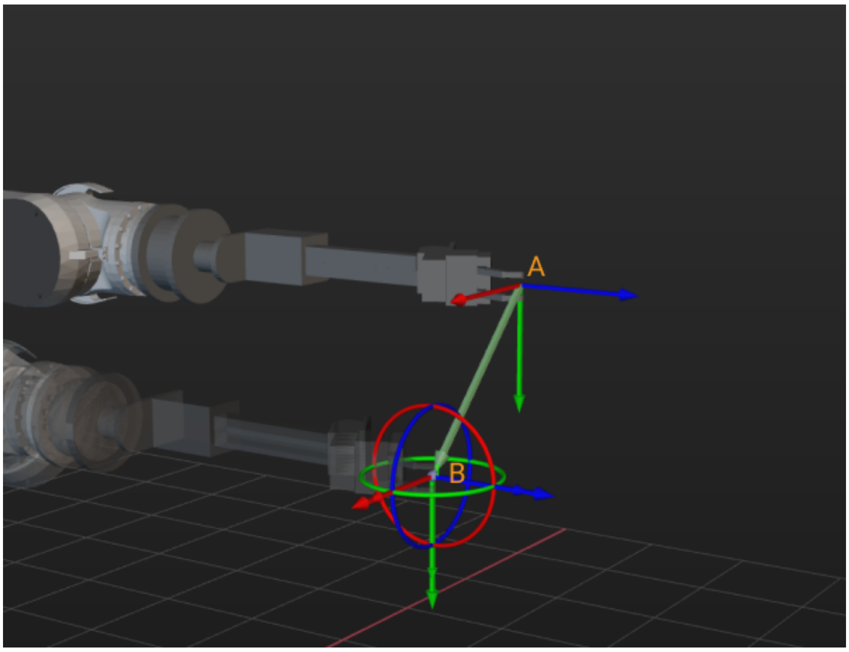
Tool

The robot will be offset on the basis of the TCP origin of Ref waypoint.

> Ref waypoint refers to the Previous waypoint or the Next waypoint.

Coordinates	Set the offset.

As shown below, A is the Ref waypoint, and B is the resulted waypoint when X and Y are set to 300. When this Step is executed, the robot will move from Waypoint A to Waypoint B.

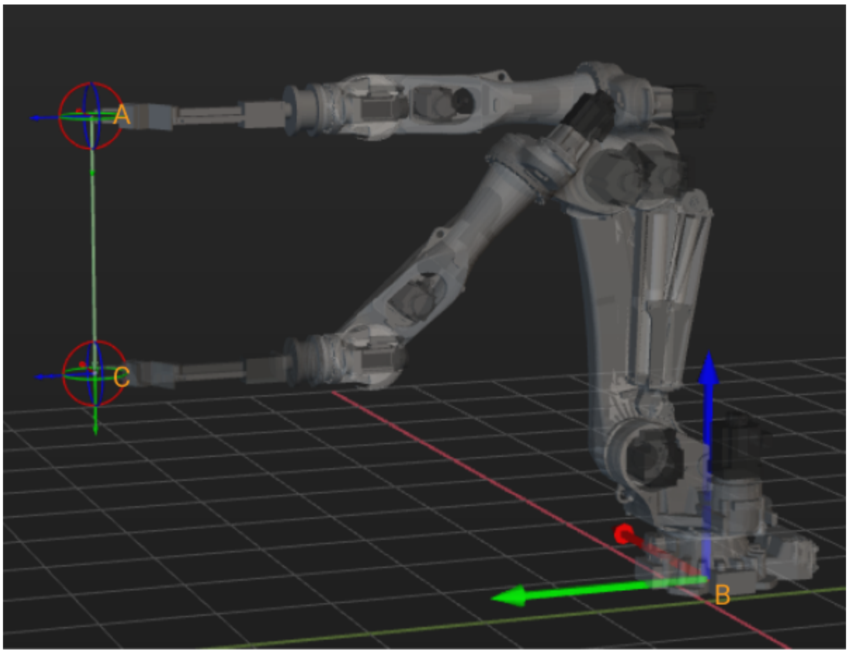


Robot

The robot will be offset on the basis of Ref waypoint, along the direction of the robot reference frame.

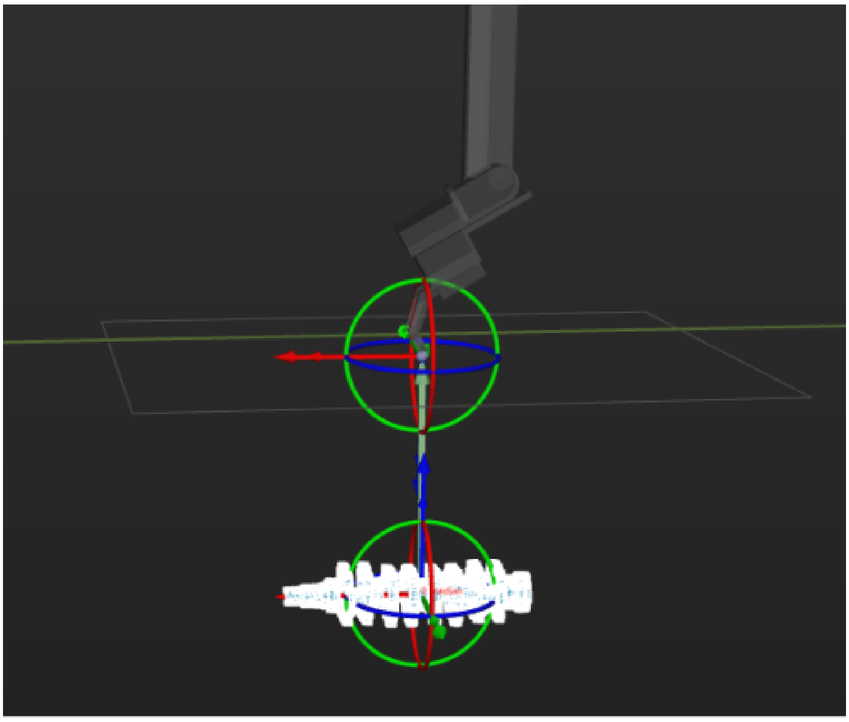
Coordinates	Set the offset.

As shown below, A is the Ref waypoint, and B is the origin of the robot reference frame. Waypoint C is the offset result when Z is set to −1000. When this Step is executed, the robot will move from Waypoint A to Waypoint C.



Lock X/Y/Z

The **Lock Z** check box is generally selected. When it is selected, the following plane will appear, as shown below. Lock Z is usually used with the Vision Move Step to set the Z value of the waypoint in the current Relative Move Step to a specified value. The robot will move to the specified height after each picking, as shown below.



Ref point

You can use a reference point to customize a reference direction. For example, if the object is placed in a deep bin, and picking in the vertical direction may lead to collisions with the surfaces of the bin, you can select the **Ref point** as the reference frame, and the robot will move in the direction toward the reference point. This Step is generally used in conjunction with the Step Vision Move.

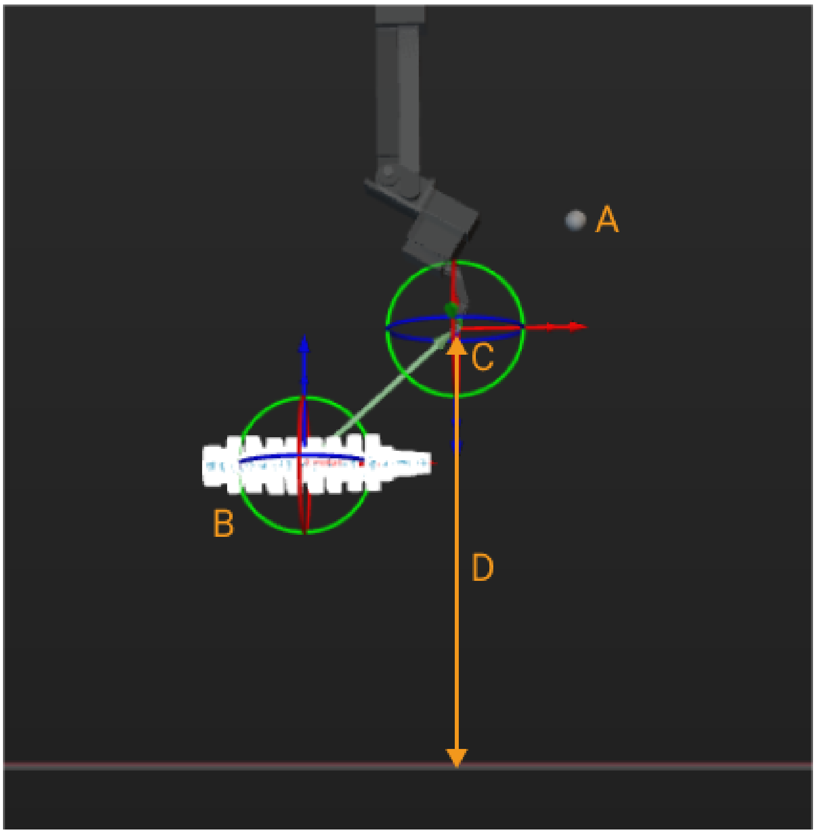
Ref point X/Y/Z

Set the position of the Ref point.

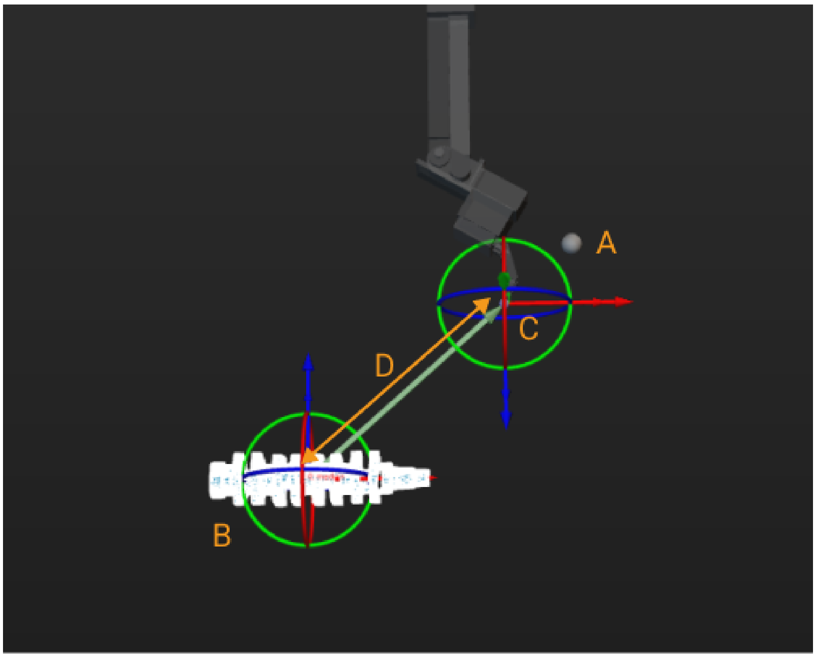
Waypoint position

Set the offset.

- By Z value in world frame: The vertical distance of the offset point from the XOY plane in the robot reference frame. As shown in the figure below, A is the Ref point, and B is the Ref waypoint, while C is the offset point. D represents the vertical distance of the offset point from the XOY plane in the robot reference frame.



- By distance from original pose: The distance that the offset point moves along the reference direction. In the following figure, A is the Ref point, B the Ref waypoint, C the offset point, and D is the distance that the offset point moves along the reference direction.



If **Move in XOY Plane** is selected, the offset point can only be set in the XOY plane of the Ref waypoint.

5.5.2. DI DO

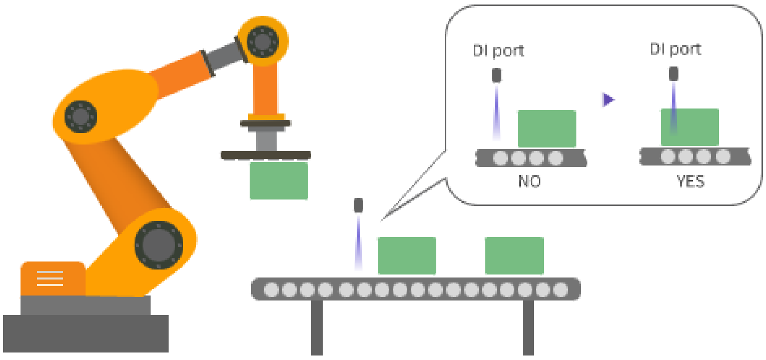
5.5.2.1. Check DI

From version 1.8.0, the "Check DI List" Step is merged with the "Check DI" Step to form the new "Check DI" Step.

Function

In robotics industry, DI usually stands for digital input. DIs are usually used to check and monitor the status of external devices or sensors.

This Step can be used to read the DI of a specified port and execute different branches (subworkflows) based on the DI.



Application Scenario

- Detect whether the box is dropped during depalletizing.

- Detect whether the vacuum gripper is attached to the box firmly during depalletizing.
- Detect specific DIs.

Parameter Description

General Parameters of Non-Move Steps

▼ *Skip Execution*

None	Default setting. Do not skip the current Step.
Simulation only	Skip the current Step during simulation. The exit port is specified by **Out Port When Skip**.
Always	Skip the current Step when the project is simulating or running. The exit port is specified by **Out Port When Skip**.
Instruction	When **Simulation only** or **Always** is selected, the current Step will be skipped and the subsequent Step will be executed when running the project. If this parameter is set to None in “Check DI” Step in the project, and there is no external input signal during simulation, the project will be stopped when executing to this Step. In this case, setting the parameter to Simulation only or Always enables the simulation to continue.

▼ *Out Port When Skip*

This parameter will take effect when **Skip Execution** is set to **Simulation only** or **Always**. It specifies the exit port when skipping a Step.

> ! If the Skip Execution parameter is set to None, a random DI value will be provided, and the exit port of this Step is random.

Pre Plan Out Port

> i The **Planning** refers to the process when the software attempts to find a collision-free path that the robot can follow to reach the target position.

For Steps that may interrupt the planning, the software cannot predict the actual exit port of these Steps during planning. In this case, an exit port can be specified for pre-planning so that the software can skip this Step and plan in a holistic approach.

The “Check DI” Step requires a real DI to be detected first until the branch can be executed, which will interrupt the planning. To run the whole workflow continuously and make the robot moves more smoothly, you can specify the exit port for pre-planning and therefore the branch connected with the port can be executed.

- When this parameter is set to −1 (default), this Step will be executed to check the current DI and interrupt the overall planning.
- When it is set to 0 or any integer larger than 0, the workflow will take the corresponding exit port to continue execution, and an overall planning can be achieved together with the Steps before and after the current Step. As a result, the overall motion of the robot will be much smoother.

Besides, if the actual exit port does not match the Pre Plan Out Port, the planning will be performed again based on the actual exit port.

Operation Mode

Source

The source of the DI can be set to either “Vision Move” or “Custom”.

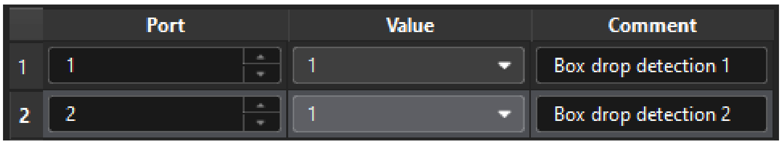
- Vision Move

  This option applies to box depalletizing scenarios only. According to the area covered by boxes on the bottom surface of the vacuum gripper, the software will automatically compute the DI port numbers that should be checked.

- Custom

  Set the DI ports and values to be checked manually. Please follow these steps to configure.

  1. Click **[+]** to add ports based on the number of ports to be checked.
  2. Set the port numbers and values to be checked, and you can comment on the function of the current port.



	Port	Value	Comment
1	1	1	Box drop detection 1
2	2	1	Box drop detection 2

- The port numbers cannot be duplicated. The configuration cannot be completed with duplicated port numbers.
- To clarify the function of each port, it is recommended to add a comment for each port.
- Select the line of a specific port and click **[-]** to delete the port configuration.

Logical Condition

- Any one satisfied: Successful if any one signal satisfied the condition, and the project will proceed along the corresponding exit port.
- All satisfied: Successful if all signals satisfied the condition, while failed if any one signal did not satisfy the condition.

Check Mode

- Check once

  Check once immediately when the Step is triggered.

- Check continuously

  Check continuously before a timeout occurs. If the signals satisfy the “Logical Condition,” the project will proceed along the corresponding exit port.

  You can set a “Timeout.” If the specified signal is not detected within the timeout, the branch connected with the “Timeout” exit port will be executed. If the “Set Timeout” parameter is not selected, the check will continue until the signal(s) satisfies the logical condition.

Examples

- When the Source is set to "Vision Move," please go to Tools in the resource tree, click "Configure depalletizing vacuum gripper" in the Tool Configuration window to configure the DI port and values.

Parameter Settings	Description
Any one satisfied + Check once	If any one signal satisfies the condition, the workflow will continue through the "Any one satisfied" exit port. If none of the signals satisfy the condition, the workflow will continue through the "None satisfied" exit port.
Any one satisfied + Check continuously	Check continuously before a timeout occurs. If any one signal satisfies the condition, the workflow will continue through the "Any one satisfied" exit port. If a timeout occurs, the workflow will continue through the "Timeout" exit port.
All satisfied + Check once	Only when all signals satisfy the condition, the workflow will continue through the "All satisfied" exit port, or else the workflow will continue through the "One or more not satisfied" exit port.
All satisfied + Check continuously	Check continuously before a timeout occurs. Only when all signals satisfy the condition, the workflow will continue through the "All satisfied" exit port. If a timeout occurs, the workflow will continue through the "Timeout" exit port.

- When the source is set to "Custom," an example of adding three ports: DI1, DI2, and DI3 is used for the illustration.

Parameter Settings	Description
Any one satisfied + Check once	If any one signal satisfies the condition, the workflow will continue through the corresponding exit port. If none of the signals satisfy the condition, the workflow will continue through the "None satisfied" exit port.
Any one satisfied + Check continuously	Check continuously before a timeout occurs. If any one signal satisfies the condition, the workflow will continue through the corresponding exit port. If a timeout occurs, the workflow will continue through the "Timeout" exit port.
All satisfied + Check once	Only when all signals satisfy the condition, the workflow will continue through the "All satisfied" exit port, or else the workflow will continue through the "One or more not satisfied" exit port.
All satisfied + Check continuously	Check continuously before a timeout occurs. Only when all signals satisfy the condition, the workflow will continue through the "All satisfied" exit port. If a timeout occurs, the workflow will continue through the "Timeout" exit port.

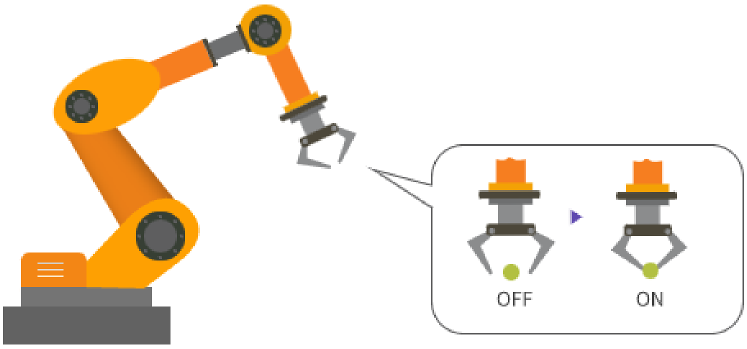
5.5.2.2. Set DO

From version 1.8.0, the "Set DO List" Step is merged with the "Set DO" Step to form the new "Set DO" Step.

Function

In robotics industry, DO usually stands for digital output. DOs are a type of electrical signal used to control devices or components in an automation system. These digital output signals are represented by the higher or the lower voltage level and are used to enable or disable connected devices or actuators. These devices include suction cups, motors, solenoid valves, relays, indicator lights, and other types of electrical components.

This Step enables external devices to perform specific actions by sending signals, such as opening and closing gripper fingers, turning suction cups on or off, and can also be used to control conveyor belts or trigger alarms, etc.



Application Scenario

- Control multi-section vacuum gripper.
- Send signals to specified DO port.

Parameter Description

General Parameters of Non-Move Steps

▼ *Skip Execution*

None	Default setting. Do not skip the current Step.
Simulation only	Skip the current Step during simulation. The exit port is specified by **Out Port When Skip**.
Always	Skip the current Step when the project is simulating or running. The exit port is specified by **Out Port When Skip**.
Instruction	When **Simulation only** or **Always** is selected, the current Step will be skipped and the subsequent Step will be executed when running the project. If this parameter is set to None in “Check DI” Step in the project, and there is no external input signal during simulation, the project will be stopped when executing to this Step. In this case, setting the parameter to Simulation only or Always enables the simulation to continue.

▼ *Out Port When Skip*

This parameter will take effect when **Skip Execution** is set to **Simulation only** or **Always**. It specifies the exit port when skipping a Step.

▼ *Wait Move Precisely*

Checked by default. This Step can only start executing when the robot reaches the specific waypoint corresponding to the move-type Step before the “Set DO” Step.

▼ *Receiver*

Please select the object that actually receives the DO, either the robot, Standard Interface or Adapter. If Adapter is selected, please enter the “Adapter Name.” In addition, there can be multiple receivers.

> If this parameter is not configured, the software cannot send the DO to the receiver.

Operation Mode

Source

The source of the DI can be set to either “Vision Move” or “Custom”.

Vision Move

This option applies to box depalletizing and multi-end tool application scenarios only. According to the depalletizing vacuum gripper or multi-end tool configuration together with the corresponding picking method, the software will automatically compute the DO port numbers of the tool ends that should be activated.

Custom

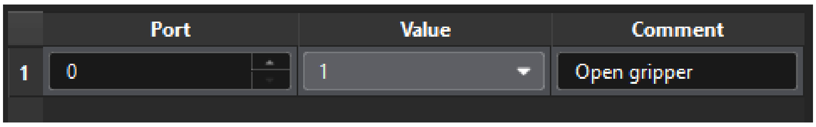
Specify the DO ports to be set manually.

- Please follow these steps to configure.
  1. Click [+] to add ports based on the number of ports to be set.
  2. Set the port numbers and values for controlling the external devices, and you can comment on the function of the current port.

  > - The port numbers cannot be duplicated. The configuration cannot be completed with duplicated port numbers.
  > - To clarify the function of each port, it is recommended to add a comment for each port.
  > - Select the line of a specific port and click [-] to delete the port configuration.

- Example

  For example, if Port 0 is used to control the gripper, and the gripper opens its fingers when the signal value is 1, the settings is shown below.



	Port	Value	Comment
1	0	1	Open gripper

5.5.3. Logical Topology

5.5.3.1. Branch by Tag

Function

This Step is usually used in combination with the “Set Tag for Branch” Step. When multiple logical branches contain the same Steps, the “Set Tag for Branch” and “Branch by Tag” Steps allow you to reuse the same Steps without duplication. Use the **Set Tag for Branch** Step to mark the branches and then connect the branches to the common part, followed by the **Branch by Tag** Step to connect the common part to different branches.

Parameter Description

General Parameters of Non-Move Steps

▼ *Skip Execution*

None
: Default setting. Do not skip the current Step.

Simulation only
: Skip the current Step during simulation. The exit port is specified by **Out Port When Skip**.

Always
: Skip the current Step when the project is simulating or running. The exit port is specified by **Out Port When Skip**.

Instruction
: When **Simulation only** or **Always** is selected, the current Step will be skipped and the subsequent Step will be executed when running the project. If this parameter is set to None in “Check DI” Step in the project, and there is no external input signal during simulation, the project will be stopped when executing to this Step. In this case, setting the parameter to Simulation only or Always enables the simulation to continue.

▼ *Out Port When Skip*

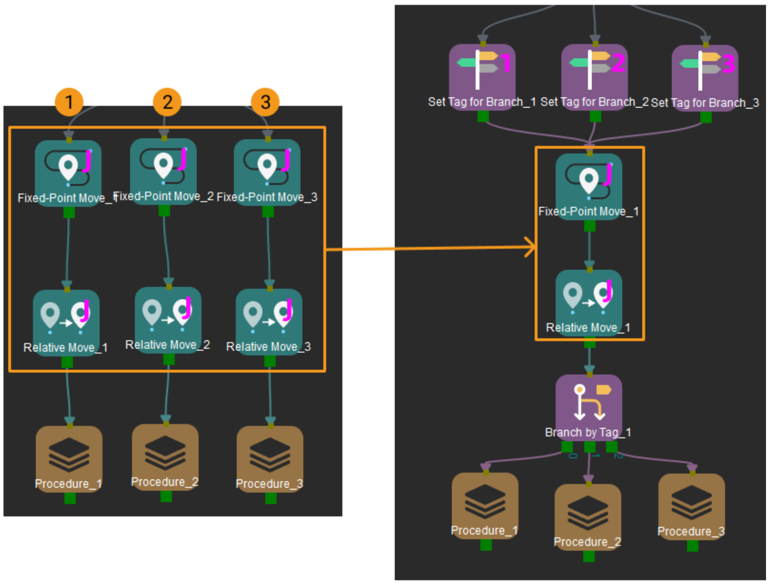
This parameter will take effect when **Skip Execution** is set to **Simulation only** or **Always**. It specifies the exit port when skipping a Step.

Out Port Count
: Used to set the number of exit ports for the Step. This parameter should be set in reference to the **Tag** parameters of Set Tag for Branch Steps. The number of exit ports should be the same as the number of tags. If two tags are set, then Out Port Count should be set to 2. Note that the branch tagged 1 corresponds to exit port 0 of the Branch by Tag Step, branch tagged 2 to exit port 1, and so on.

Application Examples

As shown in the following figure, three logical branches use the same Steps (“Fixed-Point Move” and “Relative Move”), and the parameters of these Steps are set to the same values.

1. Keep only one set of “Fixed-Point Move” and “Relative Move” Steps.
2. Add the **Set Tag for Branch** Step to each branch and set the tag for each branch (1, 2, and 3).
3. Once Mech-Viz proceeds to the **Branch by Tag** Step after the common Steps, it will proceed along the branch before the common Steps according to the tag of the branch.



5.5.3.2. Branch by Msg

Function

This Step is used to wait for the external commands and execute different branch flows according to the commands.

> The cache of the "Branch by Msg" Step forms a queue. For example, after the "Branch by Msg" Step receives command 0 and it has not executed the command, it receives command 1 afterwards. When the Step is run, command 0 will be executed first, and 1 will be executed in the next run. In addition, all cached commands will be cleared at the end of the run.

Usage Scenario

- Execute different branch flows according to the external commands.
- Pause the project execution and resume it until an external command is received.

Parameter Description

General Parameters of Non-Move Steps

▼ *Skip Execution*

None
: Default setting. Do not skip the current Step.

Simulation only
: Skip the current Step during simulation. The exit port is specified by **Out Port When Skip**.

Always
: Skip the current Step when the project is simulating or running. The exit port is specified by **Out Port When Skip**.

Instruction	When **Simulation only** or **Always** is selected, the current Step will be skipped and the subsequent Step will be executed when running the project. If this parameter is set to None in “Check DI” Step in the project, and there is no external input signal during simulation, the project will be stopped when executing to this Step. In this case, setting the parameter to Simulation only or Always enables the simulation to continue.

▼ *Out Port When Skip*

This parameter will take effect when **Skip Execution** is set to **Simulation only** or **Always**. It specifies the exit port when skipping a Step.

Out Port Count

This parameter is used to specify the out port count of this Step.

- To pause the project execution and resume upon the reception of an external command, set the out port count=1. In this way, the subsequent flow continues to be executed after the “Branch by Msg” receives an external command.
- To execute the corresponding branch flows according to different external commands, set the out port count to a value greater than 1. Please set the out port count based on the number of branches in actual need.

The parameters display differently according to the changes of out port count. Please choose to read the corresponding section as needed.

Out Port Count=1

The following parameters can be set when **Out Port Count** is 1.

It is generally used to interrupt the project execution. The subsequent flow continues to be executed after the “Branch by Msg” receives an external command.

Affect Subsequent Move (Will Interrupt Planning)

This Step will interrupt the planning by default, that is, the planning will end at this Step and the planning based on subsequent workflow will resume after an external command is received.

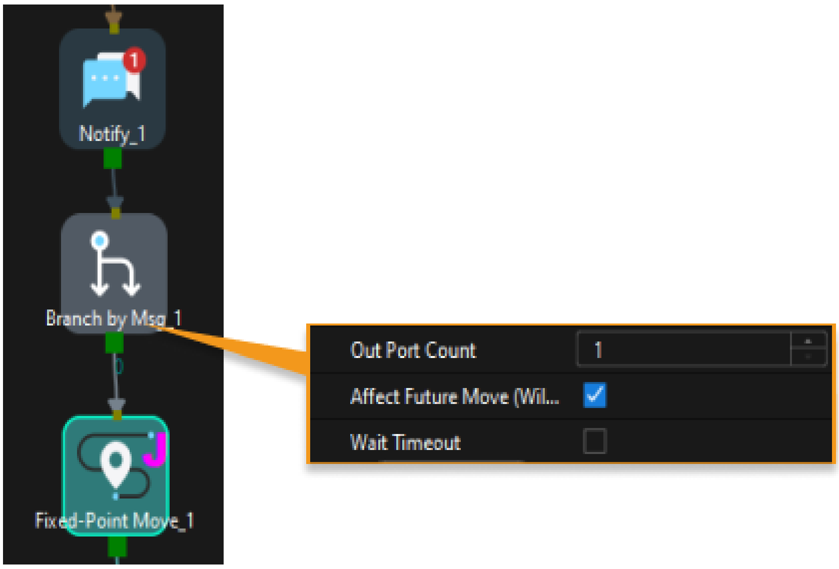
> Planning: The process when the software attempts to find a collision-free path that the robot can follow to reach the target position.

Wait Timeout

If selected, a timeout exit port is added, and you can set the **Wait Timeout** parameter.

When the wait time for the external command exceeds the set **Wait Timeout** value, the timeout exit port will be taken.

Application Examples



Out Port Count 1
Affect Future Move (Wil... ☑
Wait Timeout □

“Notification_1” sends messages to Adapter, which collects required information and calculates the result. Adapter then inputs the result to the corresponding Step, and informs “Branch by Msg_1” to continue to run.

The timing of collecting the underlying information used for the calculations is uncertain. Therefore, there is a need to ensure that all information is up to date. In addition, the Step that receives the result of the Adapter’s calculation involves a change in the state. So it is necessary to ensure that the Step receives the result before proceeding with planning. Therefore, “Branch by Msg” is needed to interrupt the planning.

Out Port Count>1

The following parameters can be set when **Out Port Count** is greater than 1.

It is generally used to execute the corresponding branch flows based on the external commands.

Pre Plan Out Port

To ensure the continuity of planning, it is necessary to specify the pre plan out port. The branch with the highest possibility is generally specified to reduce the probability of re-planning and reduce robot wait time.

When this parameter is set as non-out port with the default value being -1, it means that pre plan out port is not specified.

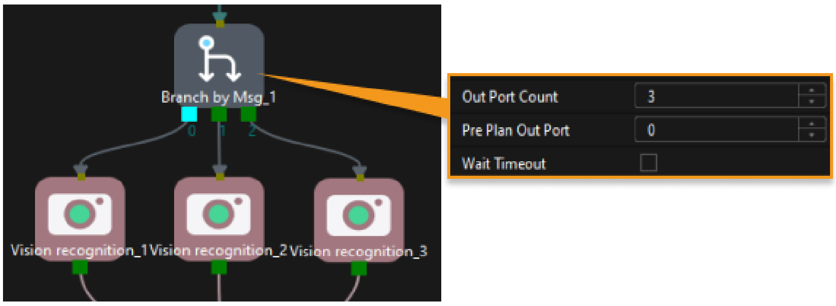
Wait Timeout

If selected, a timeout exit port is added, and you can set the **Wait Timeout** parameter.

When the wait time for the external command exceeds the **Wait Timeout** value, the timeout exit port will be taken.

Application Examples

In the Standard Interface communication, use **Branch by Msg** to call different vision projects.



Out Port Count 3

Pre Plan Out Port 0

Wait Timeout □

Take a TCP/IP command for example, if the Step ID of the “Branch by Msg” is 1, then the branch flow after exit port 0 will be executed when command 203,1,1 is sent; the branch flow after exit port 1 will be executed when command 203,1,2 is sent; and so on.

> For detailed information of Standard Interface commands, please refer to Command 203—Set Exit Port for Branch by Msg in Mech-Viz.

5.5.3.3. Procedure

Function

Multiple Steps can be grouped into a Procedure. It is also possible to nest Procedures within a procedure. If the workflow inside a procedure has multiple possible outcomes, use it with Procedure Exit.

Parameter Description

None.

Name

Customize the name of the Procedure.

5.5.3.4. Procedure Exit

Function

Designate the exit port to be taken when exiting a Procedure.

Usage Scenario

If the workflow inside a Procedure has multiple possible outcome, use **Procedure Exit** to select the exit port to be taken.

Parameter Description

General Parameters of Non-Move Steps

▼ *Skip Execution*

None
Default setting. Do not skip the current Step.

Simulation only
Skip the current Step during simulation. The exit port is specified by **Out Port When Skip**.

Always

Skip the current Step when the project is simulating or running. The exit port is specified by **Out Port When Skip**.

Instruction

When **Simulation only** or **Always** is selected, the current Step will be skipped and the subsequent Step will be executed when running the project. If this parameter is set to None in “Check DI” Step in the project, and there is no external input signal during simulation, the project will be stopped when executing to this Step. In this case, setting the parameter to Simulation only or Always enables the simulation to continue.

▼ *Out Port When Skip*

This parameter will take effect when **Skip Execution** is set to **Simulation only** or **Always**. It specifies the exit port when skipping a Step.

Exit Port

Default value

0

Instruction

Select the exit port of the Procedure to be taken. Each exit port is connected to a different succeeding logic.

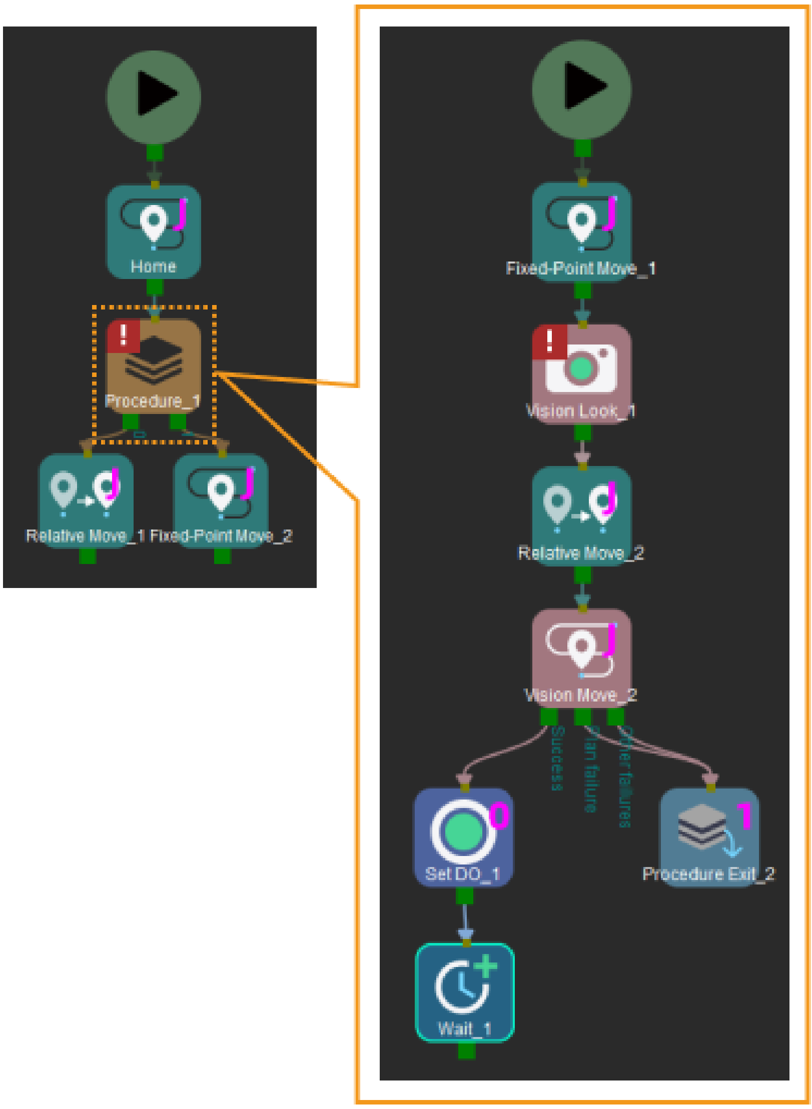
Exit Port should be set in sequence, starting from 1. Exit Port 0 corresponds to the branch not connected with **Procedure Exit** in the Procedure. See the following application examples for details.

Application Examples

The “Vision Move” Step in the Procedure has three possible outcomes: success, plan failure, and other failure. Two exit ports are needed on the Procedure to connect to succeeding logics for success and failure.

Connect the Plan failure and Other failure exit ports of the “Vision Move” Step to a **Procedure Exit** and set **Exit Port** of **Procedure Exit** to 1. An exit port 1 is added to the Procedure, the exit port taken when the “Vision Move” Step failed at planning a path.

The Success port of the “Vision Move” Step is not connected to a **Procedure Exit**, and when this branch is executed, exit port 0 will be taken to exit the Procedure.



5.5.3.5. Set Tag for Branch

Function

This Step is usually used in combination with the “Branch by Tag” Step. When multiple logical branches contain the same Steps, the “Set Tag for Branch” and “Branch by Tag” Steps allow you to reuse the same Steps without duplication. Use the **Set Tag for Branch** Step to mark the branches and then connect the branches to the common part, followed by the **Branch by Tag** Step to connect the common part to different branches.

Parameter Description

General Parameters of Non-Move Steps

▼ *Skip Execution*

None
Default setting. Do not skip the current Step.

Simulation only
Skip the current Step during simulation. The exit port is specified by **Out Port When Skip**.

Always

Skip the current Step when the project is simulating or running. The exit port is specified by **Out Port When Skip**.

Instruction

When **Simulation only** or **Always** is selected, the current Step will be skipped and the subsequent Step will be executed when running the project. If this parameter is set to None in "Check DI" Step in the project, and there is no external input signal during simulation, the project will be stopped when executing to this Step. In this case, setting the parameter to Simulation only or Always enables the simulation to continue.

▼ *Out Port When Skip*

This parameter will take effect when **Skip Execution** is set to **Simulation only** or **Always**. It specifies the exit port when skipping a Step.

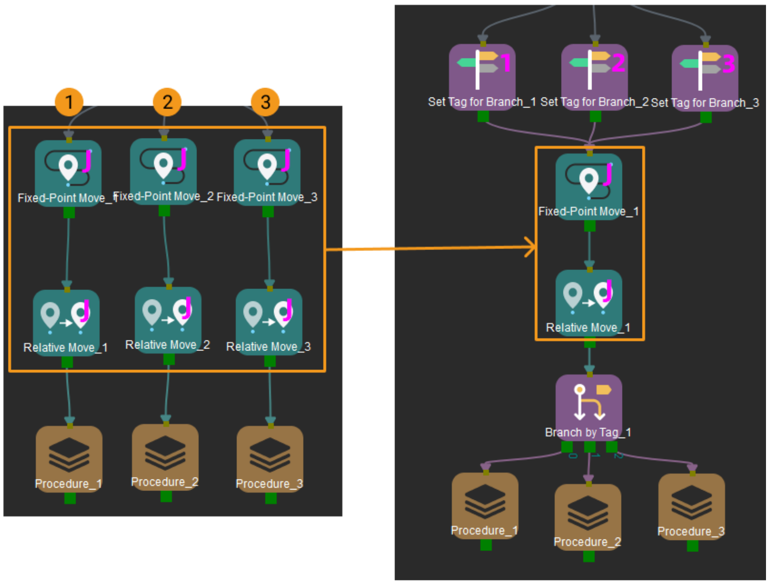
Tag

Used to set the tag of the current logical branch. This parameter should be set in reference to the **Out Port Count** parameters of the "Branch by Tag" Step. The number of exit ports should be the same as the number of tags. If two tags are set, then Out Port Count should be set to 2. Note that the branch tagged 1 corresponds to exit port 0 of the Branch by Tag Step, branch tagged 2 to exit port 1, and so on.

Application Examples

As shown in the following figure, three logical branches use the same Steps ("Fixed-Point Move" and "Relative Move"), and the parameters of these Steps are set to the same values.

1. Keep only one set of "Fixed-Point Move" and "Relative Move" Steps.
2. Add the **Set Tag for Branch** Step to each branch and set the tag for each branch (1, 2, and 3).
3. Once Mech-Viz proceeds to the **Branch by Tag** Step after the common Steps, it will proceed along the branch before the common Steps according to the tag of the branch.



5.5.4. Palletizing

5.5.4.1. Custom Pallet Pattern

Function

Use the pallet pattern customized in the Pallet Editor.

Usage Scenario

This Step applies to scenarios in which the pallet patterns are not symmetric and irregular.

Parameter Description

Move-Type Step Common Parameters

▼ *Send Waypoint*

Selected by default to send waypoint poses to the receiver, such as the robot. When this option is unselected, the waypoint pose will not be sent. However, the waypoint will remain in the planned path.

▼ *Try Continuously Running through Succeeding Non-Moves*

Unselected by default. When **non-move Steps**, such as Vision Look, Set DO, Check DI, etc., are connected between **move-type Steps**, the robot's path planning will be interrupted, and the actual robot will take a short pause, reducing the smoothness of running.

When this option is selected, the project will continue to run without waiting for the current move-type Step to complete execution, and therefore the robot can move in a smooth way without pauses. However, enabling this option may cause the execution of the Step to end prematurely.

Why will this option cause the execution of the Step to end prematurely?

Mech-Viz will send multiple poses simultaneously to the robot when the project is running. When the currently returned JPs of the robot correspond to the last pose sent by Mech-Viz, Mech-Viz will assume that the robot has moved to the last position.

For example, there are 10 move-type Steps in a path, and the pose of the 5th move-type Step is the same as that of the last move-type Step. When the robot moves at low speed, it sends JPs to Mech-Viz when it moves to the 5th move-type Step, Mech-Viz may mistakenly determine that the robot has finished the move-type Steps and prematurely ends the Steps since the poses of the 5th move-type Step and the last move-type Step are the same in the path.

▼ *Do Not Check Collision with Placed Workobject*

Unselected by default, namely that the collision with the already placed objects will not be detected. When this option is selected, the collisions between the robot, end tool, and placed objects will be detected.

In palletizing scenarios, the two possible cases of error are as follows:

1. When the robot is placing a carton, the robot may come into light contact with the placed cartons while no deformation will be caused. After Mech-Viz detects this collision in simulation, it will plan other positions for placing the carton, and therefore a full stack cannot be formed.

2. Usually, the TCP of a suction cup is inside the suction cup model instead of on the surface of it. Under this circumstance, the suction cup may be embedded in the model of the picked carton in the simulation of picking, while the software does not detect the collision between the end tool and the picked object. After the robot places the object and the carton model turns into an object model in the scene, a collision between the suction cup and the carton will be detected and the palletizing cannot be completed.

When this option is selected, no collision between the robot, end tool, and the placed object will be detected, and the above two cases of errors can be avoided.

▼ *Point Cloud Collision Detection Mode*

Select the proper mode according to the requirement of the on-site situation. Usually, the default setting **Auto** can be used. **Do not check collision** mode can be used in move-type Steps before the robot picks the object, and **Check collision** mode can be used after the robot picks the object.

Auto	Default setting. Collision with point cloud is checked only for the “Vision Move” Step and the “Relative Move” Step that depends on the “Vision Move” Step, but not for all move-type Steps.
Do not check collision	Point cloud collisions for all move-type Steps will not be detected.
Check collision	Point cloud collisions for all move-type Steps will be detected.

❗ When Collisions › Collision detection configuration › Detect collision between point cloud and others is switched on, Mech-Viz will detect collisions between the robot model, end tool model, and point cloud when planning the path. Normally, detects whether the robot collides with other objects during picking and placing. When there are point cloud outliers, non-exiting collisions will be detected, which leads to errors in path planning.

▼ *Ignore Workobject Symmetry*

This parameter will only take effect when **Waypoint type** of the Step is set to **Workobject pose**.

None	Default setting, i.e., do not disable symmetry on any axis.
Around workobject frame Z axis	Only disable symmetry on Z-axis of the workobject reference frame.
Around workobject frame X&Y axis	Disable symmetry on X-axis and Y-axis of the workobject reference frame.
Around all axes	Once the object symmetry is disabled, the robot will place the objects strictly according to the workobject poses.

ℹ In some special cases, objects are not pickable due to their peculiar poses. Setting **Rotational symmetry** under Workobjects › Workobject Configuration in Resources may solve this problem. Based on the settings of Rotational symmetry, when the default object pose is not pickable, Mech-Viz will calculate candidate poses and determine whether the candidate poses are pickable. As the candidate poses calculated based on the settings of Rotational symmetry are different from the original poses output from Mech-Vision, the consistency of the objects’ place poses cannot be guaranteed.

▼ *Plan Failure Out Port*

After selecting this parameter, this Step will add “Plan failure” out port.

During the planning process, planning is carried out along the branch after the “Success” out port; if the planning fails in the current Step, the branch process after the “Planning failure” out port will be executed.

Index

Starting Index

Description	The index of the box to be placed.
Value	Integers, and the default value is 0.
Usage	When there is not any box on the pallet, the value is **0**. If the palletizing process pauses and then continues, and N boxes have been palletized, the value should be set to N. Then the software will start with the N+1 box for palletizing.

Current Index

Description	Used to indicate the position of the current box. When the value is N, the box that is palletized in the *(N+1)*th attempt will be displayed.
Value	Integers that will be read automatically. It will be updated according to the external command if there is one.

Basic Pallet Settings

Hide Trajectory

Unselected by default, and the path along which the box is placed on the pallet will be displayed. Select to hide the path of the boxes.

Moves Count

Display the number of objects that can be placed on the pallet. Cannot be modified.

Motion Control

Auto Middle Point Force MoveJ

Selected by default, i.e., the robot moves in joint motion type before palletizing.

Entry Force MoveJ/Adjust Force MoveJ/Place Force MoveJ

Unselected by default, i.e., the robot does not need to move in joint motion type when it approaches the pallet, adjusts the pose, and places the box.

When the workspace on site is limited and the palletizing should be performed in linear motion, select the parameter. A limited workspace may lead to singularities. In this case, you can set the robot motion in any process (Entry, Adjust, Place) to joint motion.

Acc&Vel Scale Ratio

Value	0–100%, the default value is 100%.

Usage Scenario

This parameter can be configured when the robot’s speed differs between approaching the stack and actually placing the boxes.

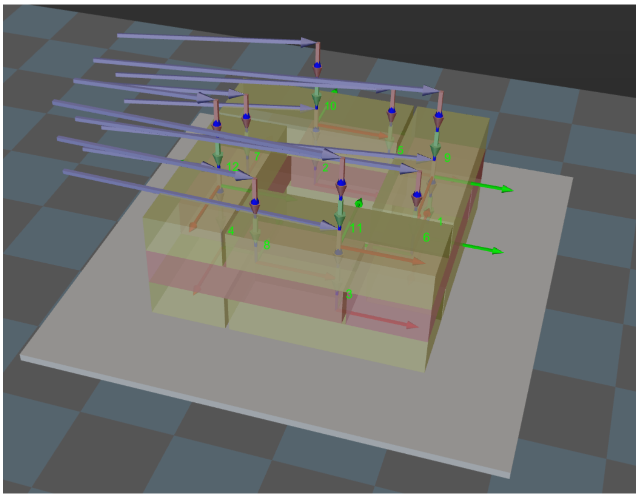
Description

This parameter is used to control the actual acceleration and velocity when the robot places the boxes. **Actual Acceleration/Velocity=Acceleration/Velocity×Acc&Vel Scale Ratio**.

The process of the robot entering the stack area can be divided into 3 segments:

Segment 1: indicated by purple arrows—approaching the stack
Segment 2: indicated by pink arrows—placing the box
Segment 3: indicated by green arrows—placing the box

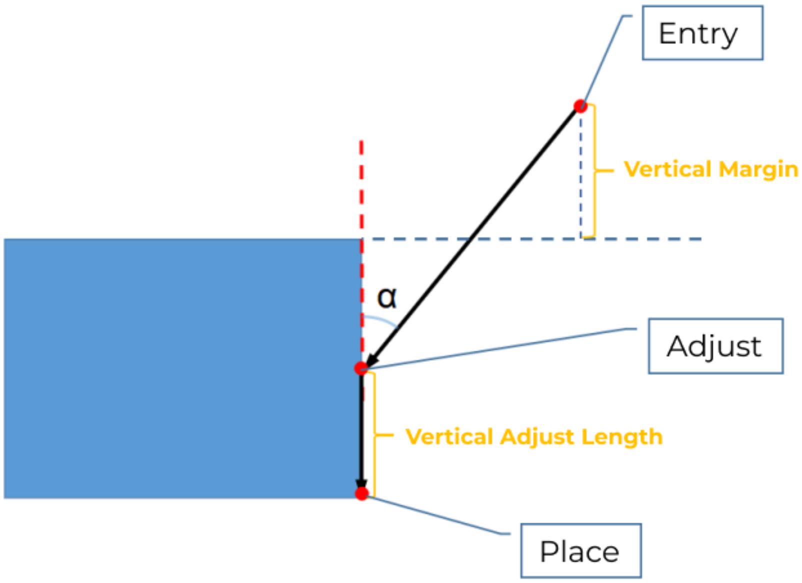
The acceleration and velocity when the robot approaches the stack (indicated by purple arrows) are specified in “Basic move settings”, and the acceleration/velocity for the last two segments are **Acceleration & Velocity × Acceleration & Velocity Scale Ratio**.



Entry Adjust

The three parameters together determines the path along which the robot enters the stack area. Adjust the entry path to make the held box approaches the palletized boxes at a specified angle, and then the held box will be placed vertically. If the box is directly placed on the stack in a vertical path, collisions may occur between the robot, held box, and the palletized boxes due to accuracy and other factors.

This parameter group determines three positions, as indicated by the red points in the figure, where the box will be moved to when it is placed on the stack. The figure below demonstrates the positions where the box will be moved (in the front view).



Vertical Adjust Length Ratio

Description	This parameter affects the “Adjust” position in the figure above. Vertical Adjust Length Ratio = Vertical Adjust Length / Height of the Box
Value range	0–1
Recommended value	0.5

Vertical Margin

Description	This parameter affects the “Entry” position in the figure above. The vertical margin is marked in the figure above.
Value range	0–∞; unit: mm. This parameter is used to control the distance above the boxes when the robot approaches the stack. The value should be adjusted according to the on-site situation.

Entry Angle Z

Description	This parameter determines the angle (°) between the path from the “Entry” point to the “Adjust” point and the vertical direction, indicated by α in the figure above.
Value range	-80°–80°
Recommended value	30°–45°

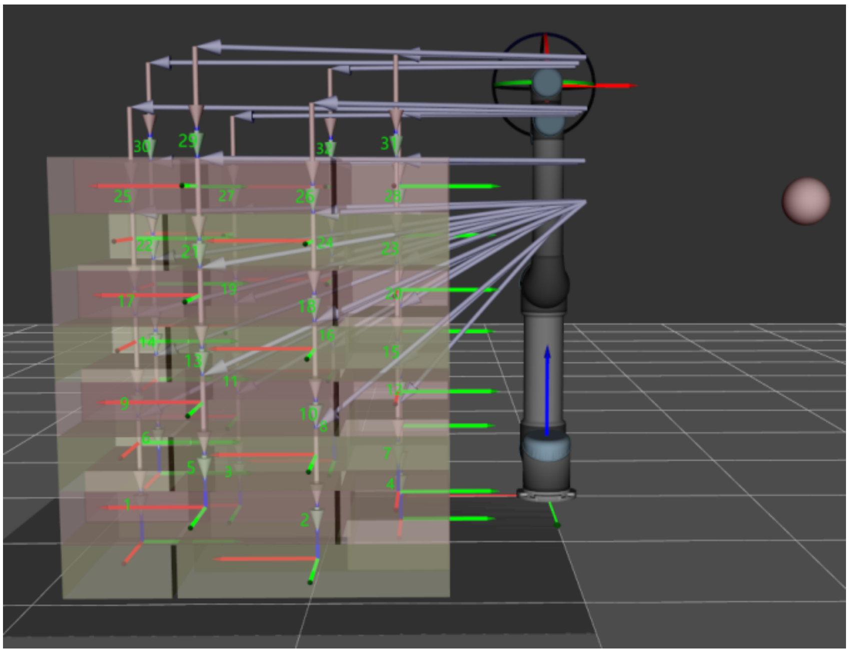
Auto Middle Point

X/Y

Set the position of the auto middle point (indicated by the pink sphere) in the robot reference frame. Based on this point, center coordinates of stacks with different heights will be calculated.

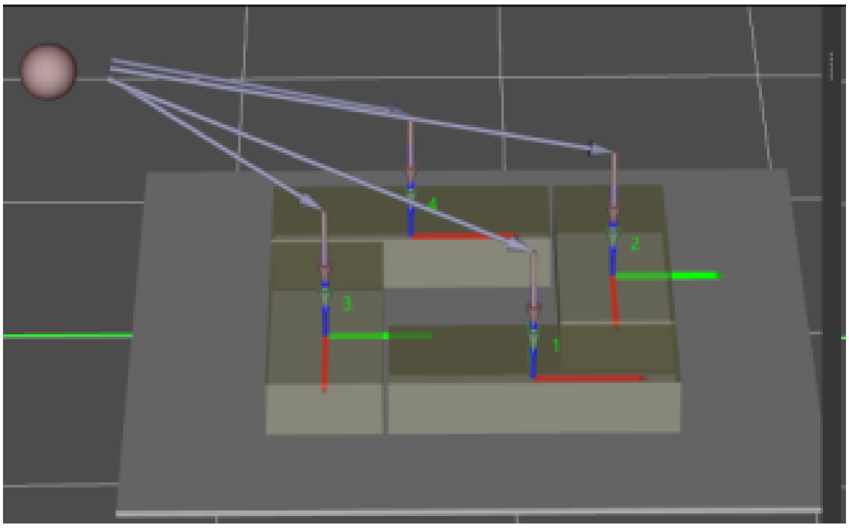
Min Z

When the robot enters the stack area (the path is indicated by the purple arrows), the minimum absolute distance in the Z-direction, i.e., the difference between the Z-direction height and the height of the current layer, as shown in the figure below.

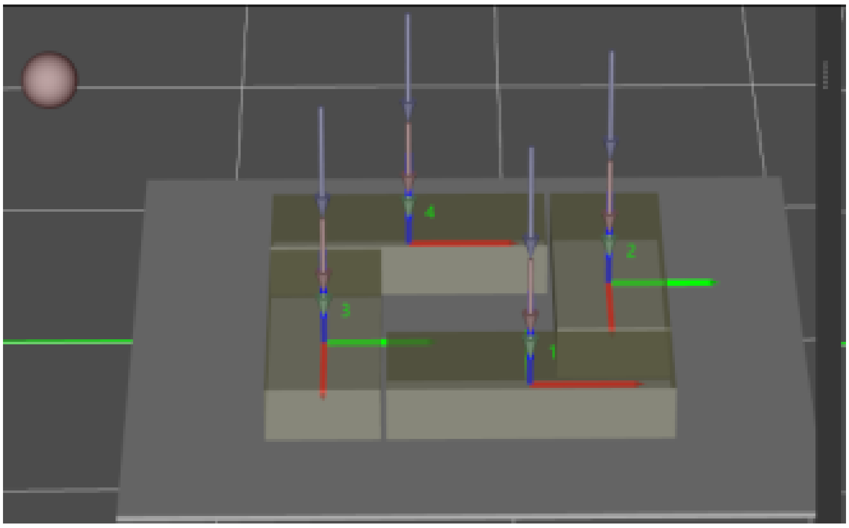


Is Middle Point Traj Vertical

Unselected by default, i.e., the robot will approach the stack from the auto middle point.



Once this option is selected, the robot will approach the stack from directly above each placing position.



Extended Distance For Entry

For some circumstances, the vacuum gripper is too large, and it may collide with the palletized boxes when approaching the stack. The Extended Distance For Entry can be set to avoid collisions in the approaching process and ensure safety.

> **Auto Middle Point** is introduced as a direction reference when the robot approaches the stack, and the point is not an actual waypoint where the robot will move. The auto middle point should be as far away from the pallet as possible. If the auto middle point is too close to the pallet, collisions may occur when palletizing.

Adjust Pallet via Vision

Adjust the position of the pallet dynamically via the “Vision Look” Step.

Need Adjust Pallet Pose

Unselected by default. Select to adjust the pallet position dynamically. When the workflow executes to this Step, the “Vision Look” Step will be used to recognize the position of the pallet.

Vision Name

Enter the name of the Mech-Vision project in which the “Vision Look” Step will be used to recognize the pallet position. When the workflow executes to this Step, the “Vision Look” Step in the Mech-Vision project will be used.

> The **parameters** can also be adjusted by the Adapter program.

For Depalletization

As Vision Service

It indicates that the pallet pattern is used for depalletization. Once this parameter is selected, you can select the “Predefined Pallet Pattern” Step as vision service in the “Vision Move” Step.

Dynamic Load

The Dynamic Load feature can only be used with a custom Adapter program. The set_task_property() function will be used to set the file paths.

Auto Entry Angle

Once this parameter is selected, the software will use the same algorithm as that of the “Auto Middle Point” in the “Predefined Pallet Pattern” Step to calculate the entry path.

Pallet Editor

1. Click **[Pallet Editor]**.
2. Please refer to Pallet Editor to edit the pallet pattern.
3. After editing and saving the pallet pattern, it will take effect.

Load Pattern

Load the edited pallet pattern.

1. Click **[Load Pattern]**.
2. Select the JSON files of the pallet pattern in the “Select a File” window.

5.5.4.2. Mixed Pallet Pattern

> Contents of this Step are under maintenance. If you need more information about this Step with urgency, please contact us at docs@mech-mind.net.

Function

This Step is used to palletize objects of different dimensions. A pallet pattern will be generated automatically based on the set parameters and dimensions of the target box recognized by the “Vision Look” Step.

Usage Scenario

For scenarios where boxes of different dimensions should be palletized.

Parameter Description

Move-Type Step Common Parameters

▼ *Send Waypoint*

Selected by default to send waypoint poses to the receiver, such as the robot. When this option is unselected, the waypoint pose will not be sent. However, the waypoint will remain in the planned path.

▼ *Try Continuously Running through Succeeding Non-Moves*

Unselected by default. When **non-move Steps**, such as Vision Look, Set DO, Check DI, etc., are connected between **move-type Steps**, the robot’s path planning will be interrupted, and the actual robot will take a short pause, reducing the smoothness of running.

When this option is selected, the project will continue to run without waiting for the current move-type Step to complete execution, and therefore the robot can move in a smooth way without pauses. However, enabling this option may cause the execution of the Step to end prematurely.

Why will this option cause the execution of the Step to end prematurely?

Mech-Viz will send multiple poses simultaneously to the robot when the project is running. When the currently returned JPs of the robot correspond to the last pose sent by Mech-Viz, Mech-Viz will assume that the robot has moved to the last position.

For example, there are 10 move-type Steps in a path, and the pose of the 5th move-type Step is the same as that of the last move-type Step. When the robot moves at low speed, it sends JPs to Mech-Viz when it moves to the 5th move-type Step, Mech-Viz may mistakenly determine that the robot has finished the move-type Steps and prematurely ends the Steps since the poses of the 5th move-type Step and the last move-type Step are the same in the path.

▼ *Do Not Check Collision with Placed Workobject*

Unselected by default, namely that the collision with the already placed objects will not be detected. When this option is selected, the collisions between the robot, end tool, and placed objects will be detected.

In palletizing scenarios, the two possible cases of error are as follows:

1. When the robot is placing a carton, the robot may come into light contact with the placed cartons while no deformation will be caused. After Mech-Viz detects this collision in simulation, it will plan other positions for placing the carton, and therefore a full stack cannot be formed.
2. Usually, the TCP of a suction cup is inside the suction cup model instead of on the surface of it. Under this circumstance, the suction cup may be embedded in the model of the picked carton in the simulation of picking, while the software does not detect the collision between the end tool and the picked object. After the robot places the object and the carton model turns into an object model in the scene, a collision between the suction cup and the carton will be detected and the palletizing cannot be completed.

When this option is selected, no collision between the robot, end tool, and the placed object will be detected, and the above two cases of errors can be avoided.

▼ *Point Cloud Collision Detection Mode*

Select the proper mode according to the requirement of the on-site situation. Usually, the default setting **Auto** can be used. **Do not check collision** mode can be used in move-type Steps before the robot picks the object, and **Check collision** mode can be used after the robot picks the object.

Auto	Default setting. Collision with point cloud is checked only for the “Vision Move” Step and the “Relative Move” Step that depends on the “Vision Move” Step, but not for all move-type Steps.
Do not check collision	Point cloud collisions for all move-type Steps will not be detected.
Check collision	Point cloud collisions for all move-type Steps will be detected.

When Collisions › Collision detection configuration › Detect collision between point cloud and others is switched on, Mech-Viz will detect collisions between the robot model, end tool model, and point cloud when planning the path. Normally, detects whether the robot collides with other objects during picking and placing. When there are point cloud outliers, non-exiting collisions will be detected, which leads to errors in path planning.

▼ *Ignore Workobject Symmetry*

This parameter will only take effect when **Waypoint type** of the Step is set to **Workobject pose**.

None	Default setting, i.e., do not disable symmetry on any axis.
Around workobject frame Z axis	Only disable symmetry on Z-axis of the workobject reference frame.
Around workobject frame X&Y axis	Disable symmetry on X-axis and Y-axis of the workobject reference frame.
Around all axes	Once the object symmetry is disabled, the robot will place the objects strictly according to the workobject poses.

ℹ In some special cases, objects are not pickable due to their peculiar poses. Setting **Rotational symmetry** under Workobjects › Workobject Configuration in Resources may solve this problem. Based on the settings of Rotational symmetry, when the default object pose is not pickable, Mech-Viz will calculate candidate poses and determine whether the candidate poses are pickable. As the candidate poses calculated based on the settings of Rotational symmetry are different from the original poses output from Mech-Vision, the consistency of the objects' place poses cannot be guaranteed.

▼ *Plan Failure Out Port*

After selecting this parameter, this Step will add “Plan failure” out port.

During the planning process, planning is carried out along the branch after the “Success” out port; if the planning fails in the current Step, the branch process after the “Planning failure” out port will be executed.

Index

Starting Index

Description The index of the box to be placed.

Value Integers, and the default value is 0.

Usage When there is not any box on the pallet, the value is **0**.

If the palletizing process pauses and then continues, and N boxes have been palletized, the value should be set to N. Then the software will start with the N+1 box for palletizing.

Current Index

Description Used to indicate the position of the current box. When the value is N, the box that is palletized in the *(N+1)*th attempt will be displayed.

Value Integers that will be read automatically. It will be updated according to the external command if there is one.

Basic Pallet Settings

Hide Trajectory

Unselected by default, and the path along which the box is placed on the pallet will be displayed. Select to hide the path of the boxes.

Moves Count

Display the number of objects that can be placed on the pallet. Cannot be modified.

Motion Control

Auto Middle Point Force MoveJ

Selected by default, i.e., the robot moves in joint motion type before palletizing.

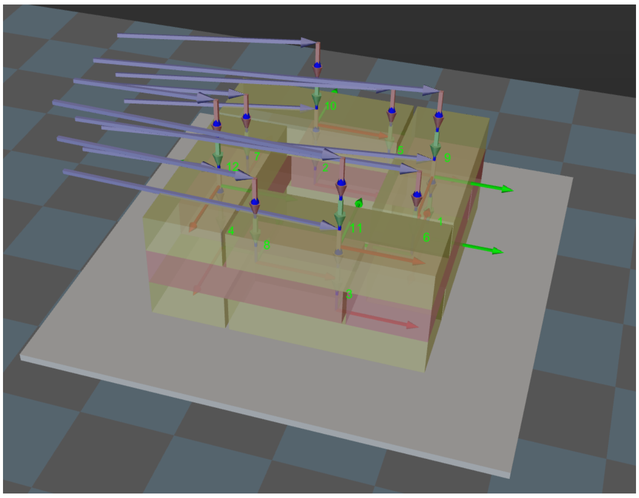
Entry Force MoveJ/Adjust Force MoveJ/Place Force MoveJ

Unselected by default, i.e., the robot does not need to move in joint motion type when it approaches the pallet, adjusts the pose, and places the box.

When the workspace on site is limited and the palletizing should be performed in linear motion, select the parameter. A limited workspace may lead to singularities. In this case, you can set the robot motion in any process (Entry, Adjust, Place) to joint motion.

Acc&Vel Scale Ratio

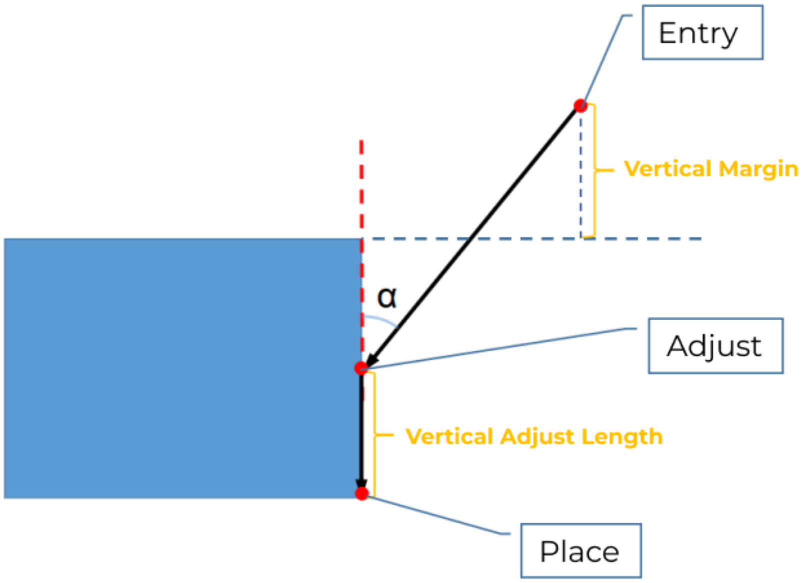
Value	0–100%, the default value is 100%.
Usage Scenario	This parameter can be configured when the robot’s speed differs between approaching the stack and actually placing the boxes.

Description	This parameter is used to control the actual acceleration and velocity when the robot places the boxes. **Actual Acceleration/Velocity=Acceleration/Velocity×Acc&Vel Scale Ratio**. The process of the robot entering the stack area can be divided into 3 segments: Segment 1: indicated by purple arrows—approaching the stack Segment 2: indicated by pink arrows—placing the box Segment 3: indicated by green arrows—placing the box The acceleration and velocity when the robot approaches the stack (indicated by purple arrows) are specified in “Basic move settings”, and the acceleration/velocity for the last two segments are **Acceleration & Velocity × Acceleration & Velocity Scale Ratio**. 

Entry Adjust

The three parameters together determines the path along which the robot enters the stack area. Adjust the entry path to make the held box approaches the palletized boxes at a specified angle, and then the held box will be placed vertically. If the box is directly placed on the stack in a vertical path, collisions may occur between the robot, held box, and the palletized boxes due to accuracy and other factors.

This parameter group determines three positions, as indicated by the red points in the figure, where the box will be moved to when it is placed on the stack. The figure below demonstrates the positions where the box will be moved (in the front view).



Vertical Adjust Length Ratio

Description	This parameter affects the “Adjust” position in the figure above. Vertical Adjust Length Ratio = Vertical Adjust Length / Height of the Box
Value range	0–1
Recommended value	0.5

Vertical Margin

Description	This parameter affects the “Entry” position in the figure above. The vertical margin is marked in the figure above.
Value range	0–∞; unit: mm. This parameter is used to control the distance above the boxes when the robot approaches the stack. The value should be adjusted according to the on-site situation.

Entry Angle Z

Description	This parameter determines the angle (°) between the path from the “Entry” point to the “Adjust” point and the vertical direction, indicated by α in the figure above.
Value range	-80°–80°
Recommended value	30°–45°

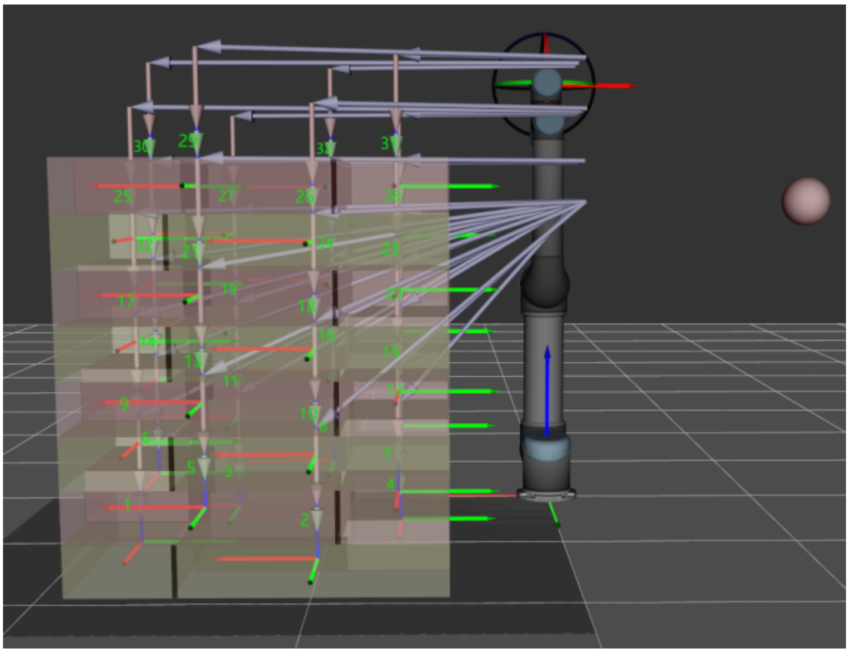
Auto Middle Point

X/Y

Set the position of the auto middle point (indicated by the pink sphere) in the robot reference frame. Based on this point, center coordinates of stacks with different heights will be calculated.

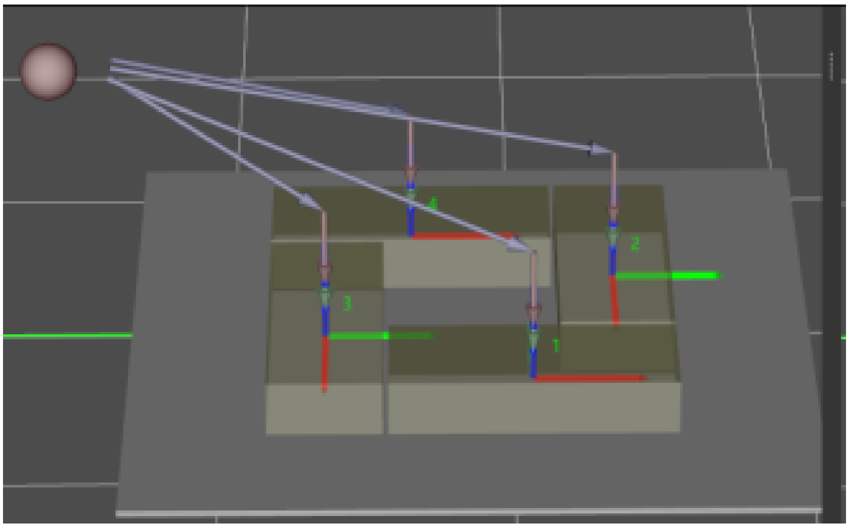
Min Z

When the robot enters the stack area (the path is indicated by the purple arrows), the minimum absolute distance in the Z-direction, i.e., the difference between the Z-direction height and the height of the current layer, as shown in the figure below.

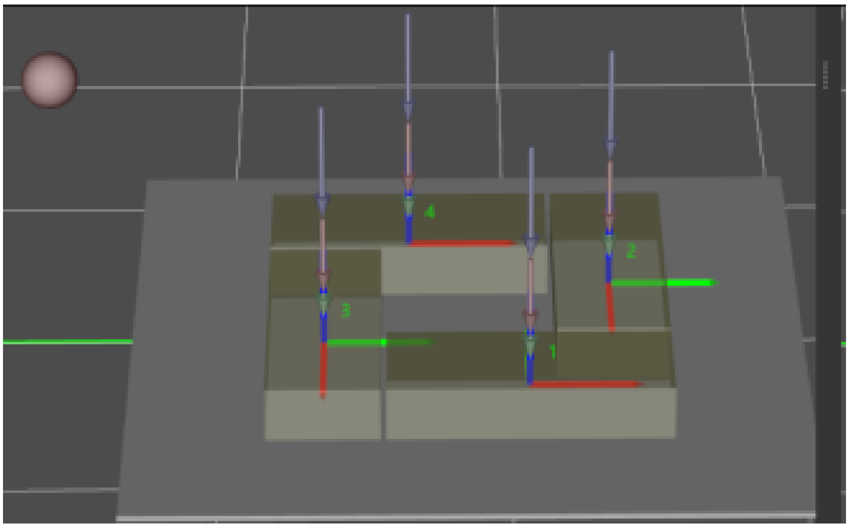


Is Middle Point Traj Vertical

Unselected by default, i.e., the robot will approach the stack from the auto middle point.



Once this option is selected, the robot will approach the stack from directly above each placing position.



Extended Distance For Entry

For some circumstances, the vacuum gripper is too large, and it may collide with the palletized boxes when approaching the stack. The Extended Distance For Entry can be set to avoid collisions in the approaching process and ensure safety.

> **Auto Middle Point** is introduced as a direction reference when the robot approaches the stack, and the point is not an actual waypoint where the robot will move. The auto middle point should be as far away from the pallet as possible. If the auto middle point is too close to the pallet, collisions may occur when palletizing.

Adjust Pallet via Vision

Adjust the position of the pallet dynamically via the "Vision Look" Step.

Need Adjust Pallet Pose

Unselected by default. Select to adjust the pallet position dynamically. When the workflow executes to this Step, the "Vision Look" Step will be used to recognize the position of the pallet.

Vision Name

Enter the name of the Mech-Vision project in which the "Vision Look" Step will be used to recognize the pallet position. When the workflow executes to this Step, the "Vision Look" Step in the Mech-Vision project will be used.

> The **parameters** can also be adjusted by the Adapter program.

Pallet Size

Pallet X Len
Set the length of the stack.

Pallet Y Len
Set the width of the stack.

Pallet Z Len
Set the height of the stack.

Pallet Z Overflow
The height of the actual stack can exceed the set height of the stack.

Example

The **Pallet Z Len** is 0.6 m, the **Pallet Z Overflow** is 0.1 m, and the actual allowable maximum height of the stack is 0.6 m + 0.1 m = 0.7 m. Assuming the height of the current stack is 0.45 m, and the height, the remaining height that can be stacked with cartons is 0.7 m - 0.45 m = 0.25 m.

- If the height of the carton is greater than 0.25 m, it cannot be placed on the stack.
- If the height of the carton is less than 0.25 m, it can be placed on the stack.

Pallet Type

Online
The dimensions of the incoming box are unknown, and the palletizing process of each box will be planned.

Offline
Dimensions of all boxes are known, and the palletizing process of all boxes will be planned at once.

> - This feature can be used to adjust the parameters of the pallet pattern file during debugging. The pallet pattern file in **JSON** format can be read and used to visualize the planned pallet pattern.
> - This feature cannot be enabled to move the real robot.

Online

Pallet Setting

Gap Width

The minimum width (mm) of the gap between boxes. To avoid collisions, the measured dimensions of the box should be smaller than the actual dimensions. Recommended value: 10 mm–20 mm

Prior Corner

Description: If a prior corner is selected, the box will be stacked in that corner first. The corner will change along with the pallet pose.

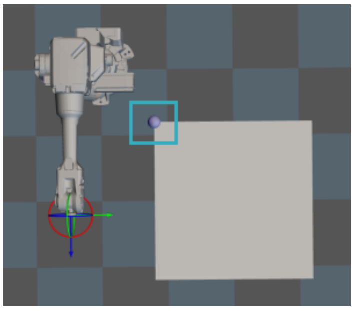
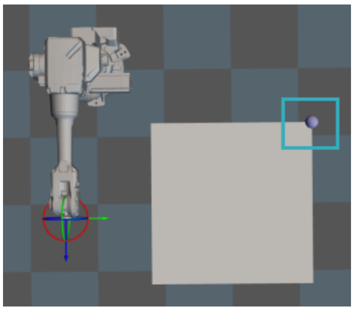
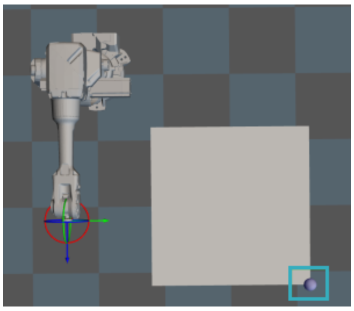
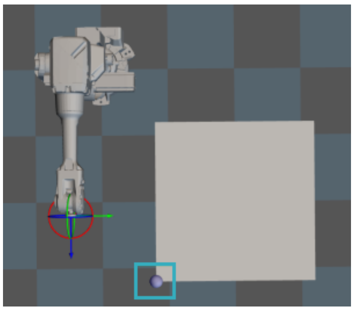
OO
The reference corner of the pallet. If the center of the pallet coincides with the origin of the robot reference frame, OO represents the corner in the negative X and negative Y direction of the robot reference frame.

OY
Based on the reference corner OO, the pallet corner that is in the positive Y direction of the robot reference frame.

XY
Based on the reference corner OO, the pallet corner that is in the positive Y and X directions of the robot reference frame.

XO
Based on the reference corner OO, the pallet corner that is in the positive X direction of the robot reference frame.

OO	OY

	
XY	XO
	

Enable Drop

Enable to place the held object by dropping.

Unselected by default, i.e., the object cannot be placed by dropping.

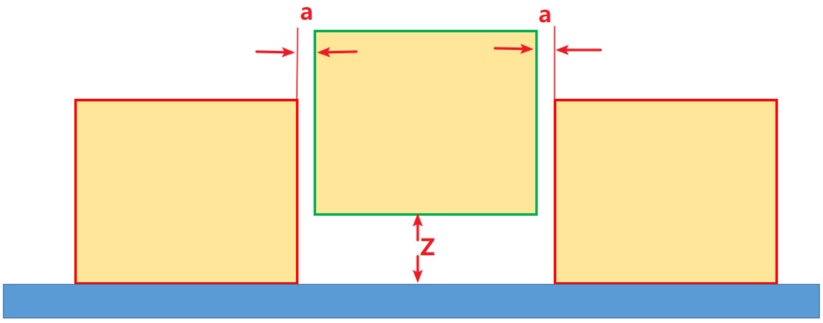
Once this option is selected, when the held carton is a certain distance away from the placing position in the Z-direction, it can be dropped directly.

Applicable Scenario When a shorter carton is placed between two taller ones or the boxes fit each other closely, the box can be placed by dropping to minimize the collision risk.

Drop Height Distance in the Z direction at which the object will be dropped.

Drop Safe Margin The minimum distance between the side of the carton to be placed and its adjacent cartons.

In the figure below, Z denotes the **Drop Height**, and a denotes the **Drop Safe Margin**.



Capture Twice to Update Carton Size

Unselected by default. When the dimensions of the entire object cannot be obtained by just one image-capturing, and a second one is needed, this parameter can be selected.

When the carton is picked via Vision Move and there is no height information of the carton, the height of the carton will be estimated for the planning of the mixed-case palletizing. When the workflow executes to this Step with the “Capture Twice to Update Carton Size” option selected, the height of the box can be obtained by capturing an image for the second time. Then the complete carton dimensions will be used for the planning of mixed-case palletizing.

> 💡 This feature can be used in conjunction with the “Update Held Workobject” Step.

Candidates Limit

Candidate Limit Counts

Default value	-1, which indicates that there is no limit on the number of candidate positions.
Description	Used to limit the total number of candidate positions of all boxes, which accelerates the subsequent path planning. If there are 20 candidate positions for planning, if collisions persist and there is no limit on the number of candidate positions, the planning will not stop until all 20 candidate positions are used up. If the Candidate Limit Counts is set to 5, the planning will stop at the 5th attempt if the collision persists.

Candidate Limit Per Box Counts

Default value	-1, which indicates that there is no limit on the number of candidate positions of each box.
Description	Used to limit the number of candidate positions for each box, which accelerates the subsequent path planning.

Boxes With Tags

Tag on Boundary

Unselected by default. This feature will not be used.

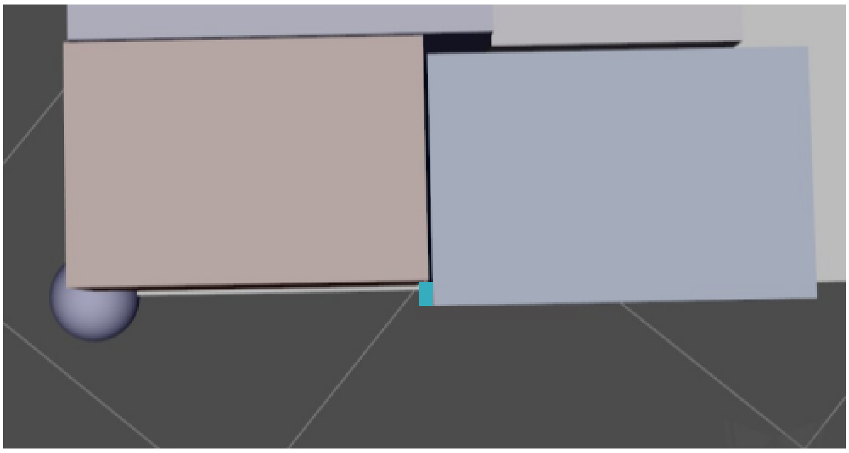
This feature applies to scenarios where tags are attached to the sides of the boxes and the tags must be exposed to the outside of the stack.

Max Dist From Tag To Pallet Boundary: The maximum distance between the box side where the tag is attached to the edge of the pallet.

Parameters For Find Candidates

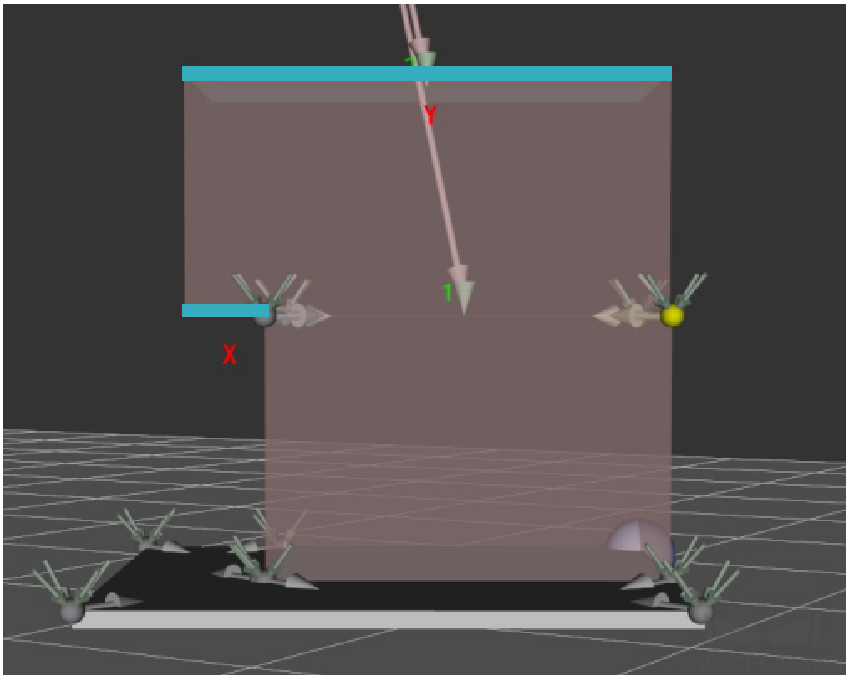
Allowed Excess

The allowable width that exceeds the edge of the stack. Recommended value: 0.02 m–0.05 m



Box Exceed Ratio

The maximum ratio of the upper box’s bottom surface area exceeding the surface under it to the total area of its bottom surface.



Layer Height Diff

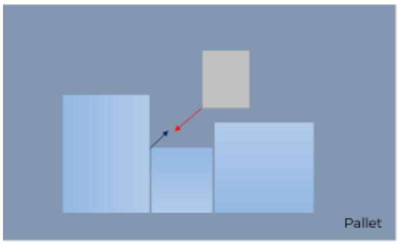
When the dimensions of the boxes on the top layer are larger than the boxes placed on the lower layer, the larger box is allowed to be placed on a plane with a height difference less than the value of this parameter.

Sampling Rate

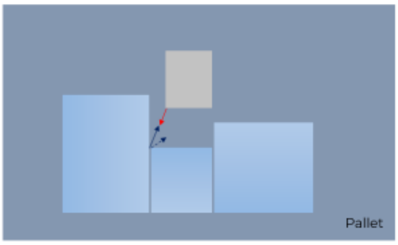
The sampling rate for box position planning (sample/meter). The higher the sampling rate, the more accurate the result. However, more calculation speed will be slower. Recommended value: 200, 500, 1000

Corner Free Angle

The angle between the box entry path and the side of an adjacent box on the pallet’s projection plane.



45°



25°

If the set value is too large, a U-shaped blank area may be left after the palletizing. If the set value is too small, collisions with adjacent boxes may occur.

Recommended value: 15°–30°

Enable Mid Point

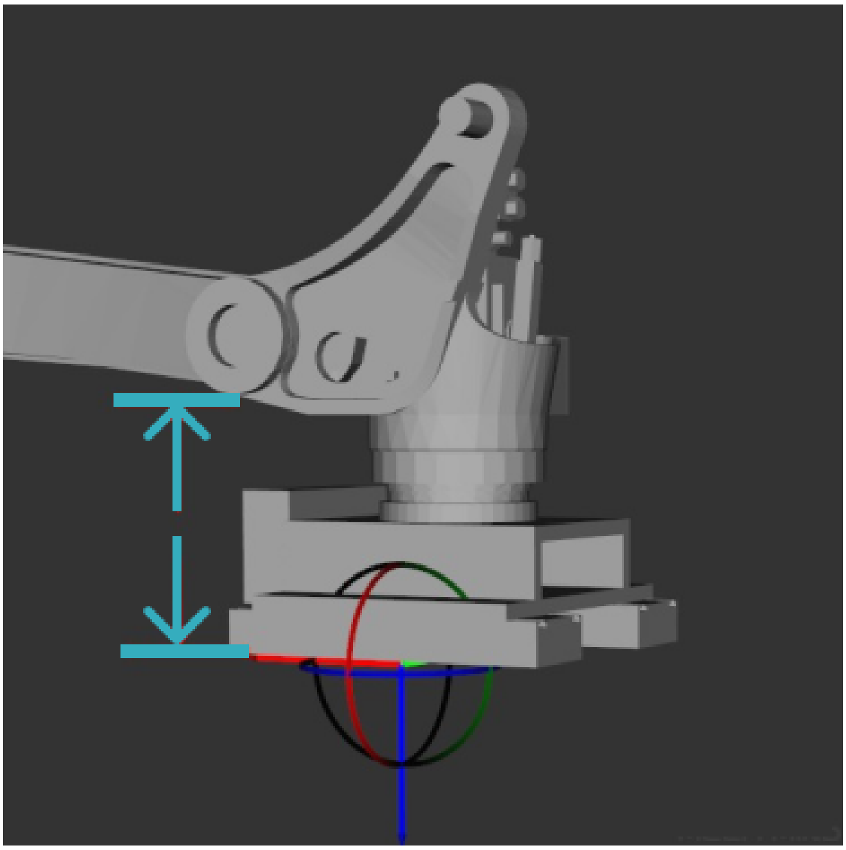
The mid point is the center point of the box side. When a hollow is formed in the stack, the boxes can be inserted vertically into the hollow.

Safe Radius

When finding the entry angle of a corner, check if a given angle is free of collision on the projected XOY plane within a radius of this value.

End Effector Dim Z

Height difference between vacuum gripper bottom surface and the bottom surface of the second joint from the bottom of the robot.



Mid Point Safe Gap

When the box is inserted to the candidate positions determined by the mid point, the minimum width of gap between adjacent boxes.

This parameter will only take effect after **Enable Mid Point** is selected.

If the robot motion error and box dimension error is small, this parameter can be set to a value that is smaller than the set Gap Width, which significantly enhances the flexibility or efficiency of stacking when operating in Online mode, especially when placing a box in the middle of the stack. When the Mid Point Safe Gap is greater than the originally set Gap Width, the planning becomes infeasible. The smaller the Mid Point Safe Gap, the more available placing poses.

Scoring Weight

The parameters in this group are all scoring weights that concertedly determine the position on the pallet where the box will be placed.

Adjacent Area Weight

The greater the value, the candidate positions where the box has more contact with the sides of adjacent boxes in the stack are more likely to be selected.

Support Area Weight

The greater the value, the candidate positions where the bottom surface of the box has more contact with the upper surface of the boxes under it (i.e., larger support area) are more likely to be selected.

Base Height Weight

The base height refers to the height difference between the bottom surface of the box to be placed and the pallet in the Z direction. The greater the value, the lower the target position, and the box tends to be placed on a lower plane.

Projected Dist to Corner Weight

The greater the value, the positions with a shorter distance projected on the pallet diagonal to the prior corner are more likely to be selected.

Support Box Num Weight

The greater the value, the positions where more boxes are under the box to be placed are more likely to be selected. The greater the value, the stabler the stack. However, the boxes in the stack may be less compact.

> It is recommended to use multiples as the adjustment intervals during parameter tuning. If you want the box to be placed closer to the prior corner, you can double the value of **Projected Dist to Corner Weight** and check the result.

Bottom Area Weight

The greater the value, the box with the larger bottom surface is more likely to be picked first.

Dist to Edge Weight

The greater the value, the closer the carton will be placed to the edge of the pallet.

Offline

Pallet Setting

Gap Width

The minimum width (mm) of the gap between boxes.

To avoid collisions, the measured dimensions of the box should be smaller than the actual dimensions.

Recommended value: 0.01 m–0.02 m

Prior Corner

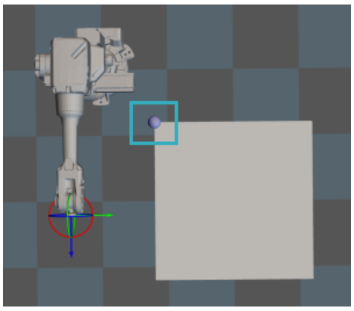
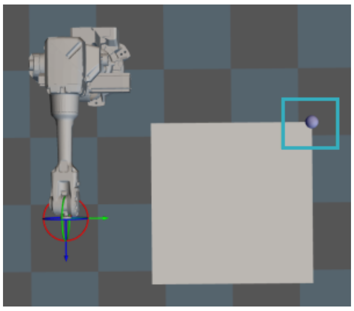
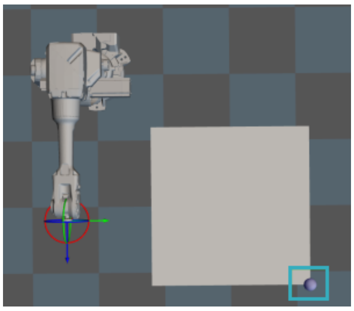
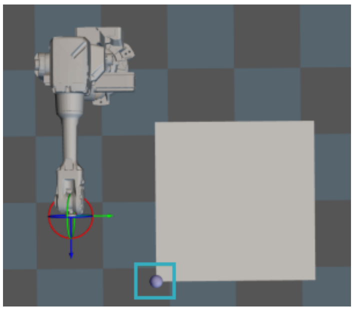
Description: If a prior corner is selected, the box will be stacked in that corner first. The corner will change along with the pallet pose.

OO
: The reference corner of the pallet. If the center of the pallet coincides with the origin of the robot reference frame, OO represents the corner in the negative X and negative Y direction of the robot reference frame.

OY
: Based on the reference corner OO, the pallet corner that is in the positive Y direction of the robot reference frame.

XY Based on the reference corner OO, the pallet corner that is in the positive Y and X directions of the robot reference frame.

XO Based on the reference corner OO, the pallet corner that is in the positive X direction of the robot reference frame.

OO	OY
	
XY	XO
	

Offline Method

Best Fit This method is applicable to scenarios where the box dimensions vary greatly or the Step is in the Online mode. The pallet pattern eventually planned is not as neat as those planned by using the other three methods.

ByStack Divide the boxes into different stacks by a specified rule (usually models or dimensions). The boxes that belong to the same stack can be sorted according to a specified rule. The nature of this method is 2-dimensional stacking.

ByLayer The concept of layer will be introduced. Boxes of the same dimensions will be placed on the same layer. If the layer cannot consist only of boxes of the same dimensions, the layer will be divided into four sections, and each type of box will be placed on an individual section.

In Batches The palletizing of one SKU can only begin after the palletizing of another SKU is completed. In AGV loading scenarios, this method can effectively reduce the number of times the AGV moves.

Offline Current Pallet Index

Description: Specify the index of the currently viewed pallet. Default value: -1

Min Box Length

An error message will pop up if the input box length is less than the value of this parameter. When the boxes are relatively small, the planning time for offline mixed-case palletizing will be longer. Setting this parameter can avoid an overly long computation time caused by the mistakenly input box length.

5.5.4.3. Multi Pick Palletizing

Function

This Step can automatically calculate multi-pick strategies based on the target pallet pattern and the arrangement of incoming boxes, aiming to improve palletizing efficiency. It is worth noting that this Step does not generate pallet patterns; instead, it requires selecting other palletizing Steps that can configure pallet patterns in the Step Parameters to obtain actual pallet pattern information. In the process of the whole workflow, this Step is mainly used to generate palletizing path, and calculate multi-pick strategies.

Usage Scenario

This Step applies to palletizing scenarios for most cases.

> The example project is provided to help you learn more about multi-pick palletizing. For the example project, you can go to File > New > Example projects > Box palletizing > Multi-Pick Palletizing in the software.

Parameter Description

Move-Type Step Common Parameters

▼ *Send Waypoint*

Selected by default to send waypoint poses to the receiver, such as the robot. When this option is unselected, the waypoint pose will not be sent. However, the waypoint will remain in the planned path.

▼ *Try Continuously Running through Succeeding Non-Moves*

Unselected by default. When **non-move Steps**, such as Vision Look, Set DO, Check DI, etc., are connected between **move-type Steps**, the robot’s path planning will be interrupted, and the actual robot will take a short pause, reducing the smoothness of running.

When this option is selected, the project will continue to run without waiting for the current move-type Step to complete execution, and therefore the robot can move in a smooth way without pauses. However, enabling this option may cause the execution of the Step to end prematurely.

Why will this option cause the execution of the Step to end prematurely?

Mech-Viz will send multiple poses simultaneously to the robot when the project is running. When the currently returned JPs of the robot correspond to the last pose sent by Mech-Viz, Mech-Viz will assume that the robot has moved to the last position.

For example, there are 10 move-type Steps in a path, and the pose of the 5th move-type Step is the same as that of the last move-type Step. When the robot moves at low speed, it sends JPs to Mech-Viz when it moves to the 5th move-type Step, Mech-Viz may mistakenly determine that the robot has finished the move-type Steps and prematurely ends the Steps since the poses of the 5th move-type Step and the last move-type Step are the same in the path.

▼ *Do Not Check Collision with Placed Workobject*

Unselected by default, namely that the collision with the already placed objects will not be detected. When this option is selected, the collisions between the robot, end tool, and placed objects will be detected.

In palletizing scenarios, the two possible cases of error are as follows:

1. When the robot is placing a carton, the robot may come into light contact with the placed cartons while no deformation will be caused. After Mech-Viz detects this collision in simulation, it will plan other positions for placing the carton, and therefore a full stack cannot be formed.
2. Usually, the TCP of a suction cup is inside the suction cup model instead of on the surface of it. Under this circumstance, the suction cup may be embedded in the model of the picked carton in the simulation of picking, while the software does not detect the collision between the end tool and the picked object. After the robot places the object and the carton model turns into an object model in the scene, a collision between the suction cup and the carton will be detected and the palletizing cannot be completed.

When this option is selected, no collision between the robot, end tool, and the placed object will be detected, and the above two cases of errors can be avoided.

▼ *Point Cloud Collision Detection Mode*

Select the proper mode according to the requirement of the on-site situation. Usually, the default setting **Auto** can be used. **Do not check collision** mode can be used in move-type Steps before the robot picks the object, and **Check collision** mode can be used after the robot picks the object.

Auto	Default setting. Collision with point cloud is checked only for the “Vision Move” Step and the “Relative Move” Step that depends on the “Vision Move” Step, but not for all move-type Steps.
Do not check collision	Point cloud collisions for all move-type Steps will not be detected.
Check collision	Point cloud collisions for all move-type Steps will be detected.

When Collisions › Collision detection configuration › Detect collision between point cloud and others is switched on, Mech-Viz will detect collisions between the robot model, end tool model, and point cloud when planning the path. Normally, detects whether the robot collides with other objects during picking and placing. When there are point cloud outliers, non-exiting collisions will be detected, which leads to errors in path planning.

▼ *Ignore Workobject Symmetry*

This parameter will only take effect when **Waypoint type** of the Step is set to **Workobject pose**.

None	Default setting, i.e., do not disable symmetry on any axis.
Around workobject frame Z axis	Only disable symmetry on Z-axis of the workobject reference frame.
Around workobject frame X&Y axis	Disable symmetry on X-axis and Y-axis of the workobject reference frame.

Around all axes	Once the object symmetry is disabled, the robot will place the objects strictly according to the workobject poses.

In some special cases, objects are not pickable due to their peculiar poses. Setting **Rotational symmetry** under Workobjects › Workobject Configuration in Resources may solve this problem. Based on the settings of Rotational symmetry, when the default object pose is not pickable, Mech-Viz will calculate candidate poses and determine whether the candidate poses are pickable. As the candidate poses calculated based on the settings of Rotational symmetry are different from the original poses output from Mech-Vision, the consistency of the objects' place poses cannot be guaranteed.

▼ *Plan Failure Out Port*

After selecting this parameter, this Step will add "Plan failure" out port.

During the planning process, planning is carried out along the branch after the "Success" out port; if the planning fails in the current Step, the branch process after the "Planning failure" out port will be executed.

Held Workobject Collision Detection Settings

▼ *Do Not Check Collision with Scene Object/Robot*

Unselected by default. Once this option is selected, the collisions between the **held workobject** with the **scene objects or robot** will not be detected, and therefore the calculation workload of collision detection will be reduced, the planning speed can be increased, and the cycle time can be shortened. It is usually enabled in the first one or two move-type Steps after the robot picks the object.

Please enable this option cautiously as there may be collision risks.

When **Detect collision between held workobject and others** under Collision detection configuration › Configuration on held workobject is enabled, the software will detect whether the model of the held object collides with the models of the scene objects and the robot.

In palletizing projects, the calculated carton dimensions have millimeter-level errors with the actual dimensions, and frictions between cartons may occur during picking but no collisions will occur. For some move-type Steps that will obviously not cause collisions, detecting such collisions only adds to the calculation workload and planning time, and consequently extending the cycle time. In palletizing projects, enabling Do Not Check Collision with Scene Objects does not affect the collision detection between the held carton and the placed cartons. This option can be enabled when there are scene objects under the stack to avoid failure of finding the palletizing solution.

▼ *Do Not Check Collision with Point Cloud*

Unselected by default. Once this option is selected, the collisions between the **held workobject** with the **point clouds in the scene** will not be detected, and therefore the calculation workload of collision detection will be reduced, the planning speed can be increased, and the cycle time can be shortened.

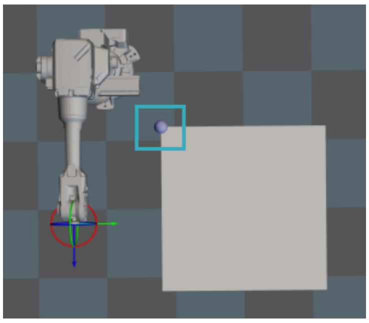
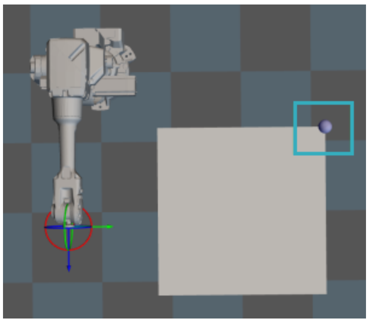
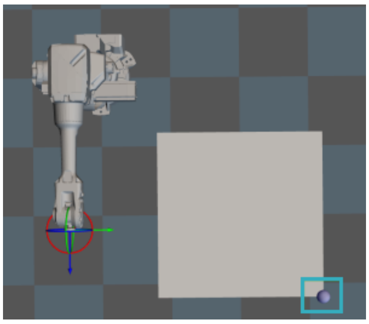
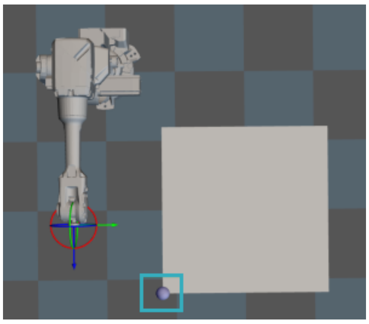
- When both **Detect collision between held workobject and others** under Collision detection configuration › Configuration on held workobject and **Configuration on point cloud ¬ Detect collision between point cloud and others** are enabled, the software will detect whether the model of the held workobject will collide with the point cloud in the scene.
- When Mech-Vision sends the point cloud and object model to Mech-Viz, the point cloud and the object model are fitted together. After the robot picks the object, the model moves along

the planned path, and the collision between the model of the held workobject and the point cloud will occur.

- It is known that the model of the held workobject will have false collisions with the point cloud. Detecting such collisions unnecessarily adds to the calculation workload and extends the planning time.

Prior Corner

This parameter specifies the corner of the pallet on which the materials are palletized first, and the position of this prior corner will be changed with the pallet pose.

Parameter	Description	Illustration
OO	The pallet corner closest to the origin of the robot reference frame.	
OY	Based on the reference corner OO, the pallet corner that is in the positive Y direction of the robot reference frame.	
XY	Based on the reference corner OO, the pallet corner that is in the positive Y and X directions of the robot reference frame.	
XO	Based on the reference corner OO, the pallet corner that is in the positive X direction of the robot reference frame.	

The position of this prior corner is determined when the pallet is not rotated.

No Symmetry

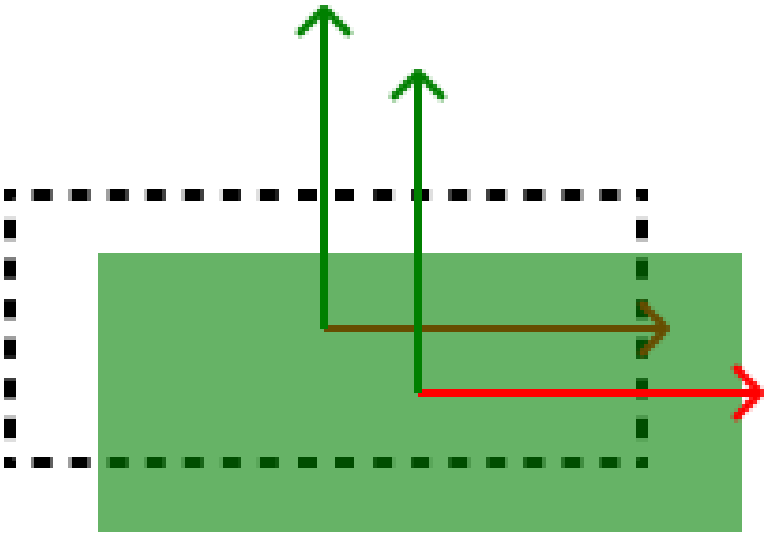
Once this parameter is selected, the rotational symmetry setting of the workobject will not take effect in the process of palletizing.

Revise to Ideal Pallet Pattern

Once this parameter is selected, the software will adjust the collision models of all the boxes after the palletizing is completed so that they can correspond to the desired box positions of the target pallet pattern.

Dist Error

When placing the box group, each placed box may deviate from the target position. This parameter specifies the maximum allowable deviation between the actual and expected position on the XY plane. When the deviation between the box and the target position exceeds this threshold, the software will determine that the match has failed. In short, it is used to determine the maximum error allowed in 2D coordinates.

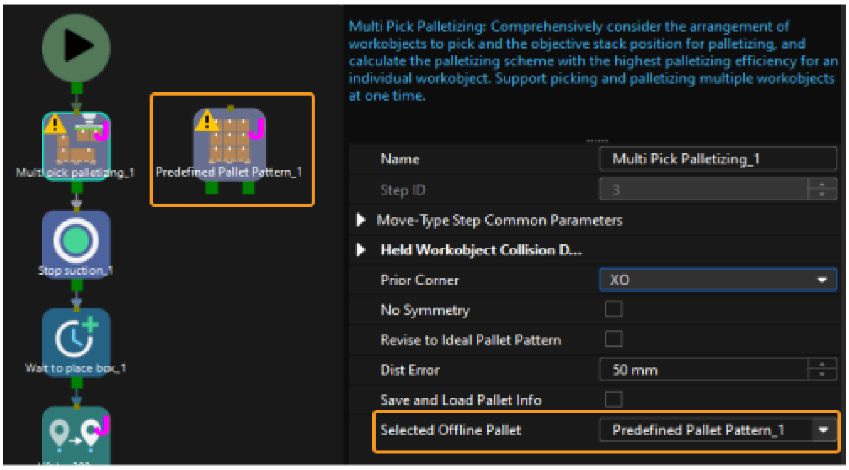


(Dashed lines indicate target positions, and the green ones indicate boxes)

Selected Offline Pallet

The Multi-Pick Palletizing Step cannot generate pallet pattern; instead, it works depending on the pallet pattern information provided by other palletizing Steps (For example, you can refer to Predefined Pallet Pattern、Custom Pallet Pattern、Mixed Pallet Pattern).

You can drag other palletizing Steps to the workflow and set the pallet pattern. Select the corresponding Steps in the right drop-down menu of this parameter, and the Multi-Pick Palletizing Step will use the pallet pattern information provided by this Step.



Multi Pick Palletizing: Comprehensively consider the arrangement of workobjects to pick and the objective stack position for palletizing, and calculate the palletizing scheme with the highest palletizing efficiency for an individual workobject. Support picking and palletizing multiple workobjects at one time.

Name	Multi Pick Palletizing_1
Step ID	3
▶ Move-Type Step Common Parameters	
▶ **Held Workobject Collision D...**	
Prior Corner	XO
No Symmetry	□
Revise to Ideal Pallet Pattern	□
Dist Error	50 mm
Save and Load Pallet Info	□
Selected Offline Pallet	Predefined Pallet Pattern_1

5.5.4.4. Predefined Pallet Pattern

Function

This Step is used to automatically generate the pallet pattern through setting the parameters. Compared to the “Custom Pallet Pattern” Step, this Step is easier to use and there is no need to manually edit the pallet pattern.

Parameter Description

Move-Type Step Common Parameters

▼ *Send Waypoint*

Selected by default to send waypoint poses to the receiver, such as the robot. When this option is unselected, the waypoint pose will not be sent. However, the waypoint will remain in the planned path.

▼ *Try Continuously Running through Succeeding Non-Moves*

Unselected by default. When **non-move Steps**, such as Vision Look, Set DO, Check DI, etc., are connected between **move-type Steps**, the robot’s path planning will be interrupted, and the actual robot will take a short pause, reducing the smoothness of running.

When this option is selected, the project will continue to run without waiting for the current move-type Step to complete execution, and therefore the robot can move in a smooth way without pauses. However, enabling this option may cause the execution of the Step to end prematurely.

Why will this option cause the execution of the Step to end prematurely?

Mech-Viz will send multiple poses simultaneously to the robot when the project is running. When the currently returned JPs of the robot correspond to the last pose sent by Mech-Viz, Mech-Viz will assume that the robot has moved to the last position.

For example, there are 10 move-type Steps in a path, and the pose of the 5th move-type Step is the same as that of the last move-type Step. When the robot moves at low speed, it sends JPs to Mech-Viz when it moves to the 5th move-type Step, Mech-Viz may mistakenly determine that the robot has finished the move-type Steps and prematurely ends the Steps since the poses of the 5th move-type Step and the last move-type Step are the same in the path.

▼ *Do Not Check Collision with Placed Workobject*

Unselected by default, namely that the collision with the already placed objects will not be detected. When this option is selected, the collisions between the robot, end tool, and placed objects will be detected.

In palletizing scenarios, the two possible cases of error are as follows:

1. When the robot is placing a carton, the robot may come into light contact with the placed cartons while no deformation will be caused. After Mech-Viz detects this collision in simulation, it will plan other positions for placing the carton, and therefore a full stack cannot be formed.
2. Usually, the TCP of a suction cup is inside the suction cup model instead of on the surface of it. Under this circumstance, the suction cup may be embedded in the model of the picked carton in the simulation of picking, while the software does not detect the collision between the end tool and the picked object. After the robot places the object and the carton model turns into an object model in the scene, a collision between the suction cup and the carton will be detected and the palletizing cannot be completed.

When this option is selected, no collision between the robot, end tool, and the placed object will be detected, and the above two cases of errors can be avoided.

▼ *Point Cloud Collision Detection Mode*

Select the proper mode according to the requirement of the on-site situation. Usually, the default setting **Auto** can be used. **Do not check collision** mode can be used in move-type Steps before the robot picks the object, and **Check collision** mode can be used after the robot picks the object.

Auto	Default setting. Collision with point cloud is checked only for the “Vision Move” Step and the “Relative Move” Step that depends on the “Vision Move” Step, but not for all move-type Steps.
Do not check collision	Point cloud collisions for all move-type Steps will not be detected.
Check collision	Point cloud collisions for all move-type Steps will be detected.

When Collisions › Collision detection configuration › Detect collision between point cloud and others is switched on, Mech-Viz will detect collisions between the robot model, end tool model, and point cloud when planning the path. Normally, detects whether the robot collides with other objects during picking and placing. When there are point cloud outliers, non-exiting collisions will be detected, which leads to errors in path planning.

▼ *Ignore Workobject Symmetry*

This parameter will only take effect when **Waypoint type** of the Step is set to **Workobject pose**.

None	Default setting, i.e., do not disable symmetry on any axis.
Around workobject frame Z axis	Only disable symmetry on Z-axis of the workobject reference frame.
Around workobject frame X&Y axis	Disable symmetry on X-axis and Y-axis of the workobject reference frame.
Around all axes	Once the object symmetry is disabled, the robot will place the objects strictly according to the workobject poses.

In some special cases, objects are not pickable due to their peculiar poses. Setting **Rotational symmetry** under Workobjects › Workobject Configuration in Resources may

> solve this problem. Based on the settings of Rotational symmetry, when the default object pose is not pickable, Mech-Viz will calculate candidate poses and determine whether the candidate poses are pickable. As the candidate poses calculated based on the settings of Rotational symmetry are different from the original poses output from Mech-Vision, the consistency of the objects' place poses cannot be guaranteed.

▼ *Plan Failure Out Port*

After selecting this parameter, this Step will add "Plan failure" out port.

During the planning process, planning is carried out along the branch after the "Success" out port; if the planning fails in the current Step, the branch process after the "Planning failure" out port will be executed.

Index

Starting Index

Description
: The index of the box to be placed.

Value
: Integers, and the default value is 0.

Usage
: When there is not any box on the pallet, the value is **0**.

: If the palletizing process pauses and then continues, and N boxes have been palletized, the value should be set to N. Then the software will start with the N+1 box for palletizing.

Current Index

Description
: Used to indicate the position of the current box. When the value is N, the box that is palletized in the *(N+1)*th attempt will be displayed.

Value
: Integers that will be read automatically. It will be updated according to the external command if there is one.

Basic Pallet Settings

Hide Trajectory

Unselected by default, and the path along which the box is placed on the pallet will be displayed. Select to hide the path of the boxes.

Moves Count

Display the number of objects that can be placed on the pallet. Cannot be modified.

Motion Control

Auto Middle Point Force MoveJ

Selected by default, i.e., the robot moves in joint motion type before palletizing.

Entry Force MoveJ/Adjust Force MoveJ/Place Force MoveJ

Unselected by default, i.e., the robot does not need to move in joint motion type when it approaches the pallet, adjusts the pose, and places the box.

When the workspace on site is limited and the palletizing should be performed in linear motion, select the parameter. A limited workspace may lead to singularities. In this case, you can set the robot motion in any process (Entry, Adjust, Place) to joint motion.

Acc&Vel Scale Ratio

Value

0–100%, the default value is 100%.

Usage Scenario

This parameter can be configured when the robot's speed differs between approaching the stack and actually placing the boxes.

Description

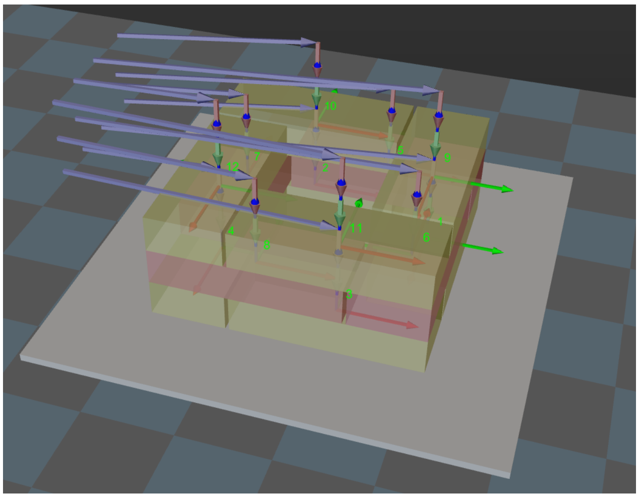
This parameter is used to control the actual acceleration and velocity when the robot places the boxes. **Actual Acceleration/Velocity=Acceleration/Velocity×Acc&Vel Scale Ratio**.

The process of the robot entering the stack area can be divided into 3 segments:

Segment 1: indicated by purple arrows—approaching the stack

Segment 2: indicated by pink arrows—placing the box
Segment 3: indicated by green arrows—placing the box

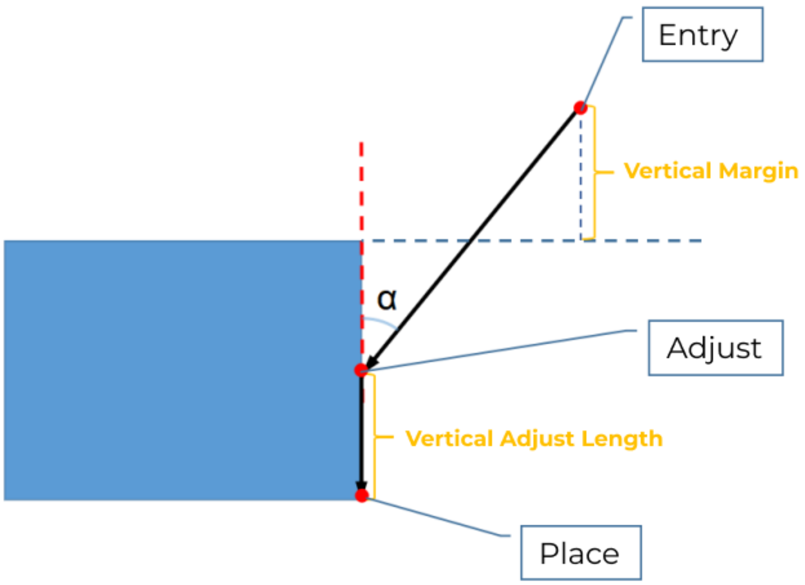
The acceleration and velocity when the robot approaches the stack (indicated by purple arrows) are specified in "Basic move settings", and the acceleration/velocity for the last two segments are **Acceleration & Velocity × Acceleration & Velocity Scale Ratio**.



Entry Adjust

The three parameters together determines the path along which the robot enters the stack area. Adjust the entry path to make the held box approaches the palletized boxes at a specified angle, and then the held box will be placed vertically. If the box is directly placed on the stack in a vertical path, collisions may occur between the robot, held box, and the palletized boxes due to accuracy and other factors.

This parameter group determines three positions, as indicated by the red points in the figure, where the box will be moved to when it is placed on the stack. The figure below demonstrates the positions where the box will be moved (in the front view).



Vertical Adjust Length Ratio

Description	This parameter affects the “Adjust” position in the figure above. Vertical Adjust Length Ratio = Vertical Adjust Length / Height of the Box
Value range	0–1
Recommended value	0.5

Vertical Margin

Description	This parameter affects the “Entry” position in the figure above. The vertical margin is marked in the figure above.
Value range	0–∞; unit: mm. This parameter is used to control the distance above the boxes when the robot approaches the stack. The value should be adjusted according to the on-site situation.

Entry Angle Z

Description	This parameter determines the angle (°) between the path from the “Entry” point to the “Adjust” point and the vertical direction, indicated by α in the figure above.
Value range	-80°–80°
Recommended value	30°–45°

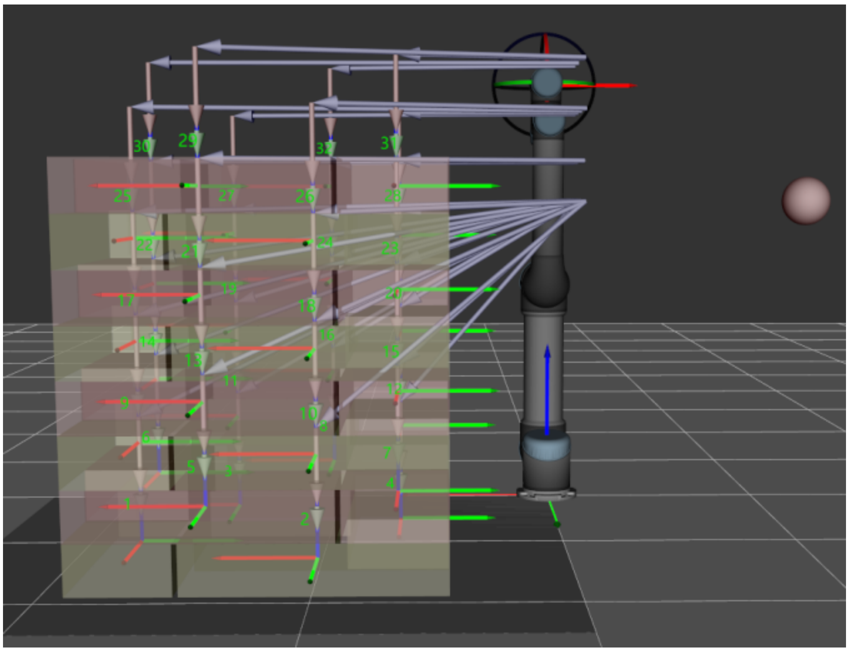
Auto Middle Point

X/Y

Set the position of the auto middle point (indicated by the pink sphere) in the robot reference frame. Based on this point, center coordinates of stacks with different heights will be calculated.

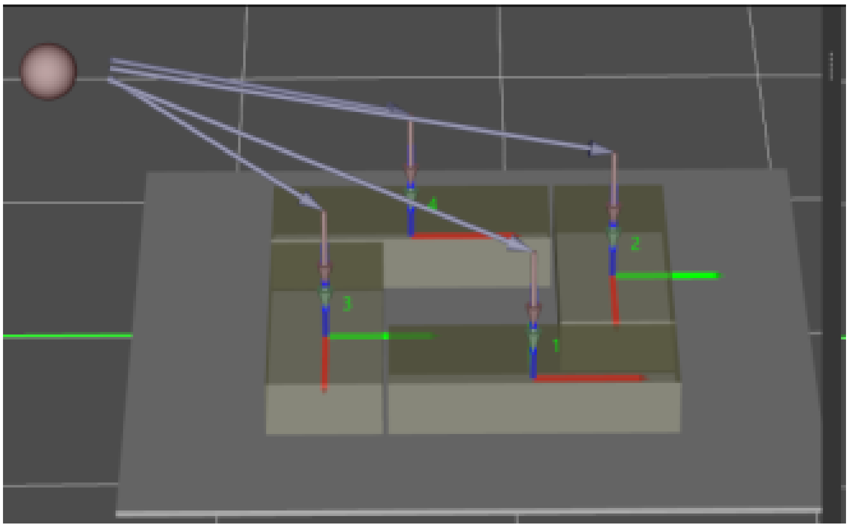
Min Z

When the robot enters the stack area (the path is indicated by the purple arrows), the minimum absolute distance in the Z-direction, i.e., the difference between the Z-direction height and the height of the current layer, as shown in the figure below.

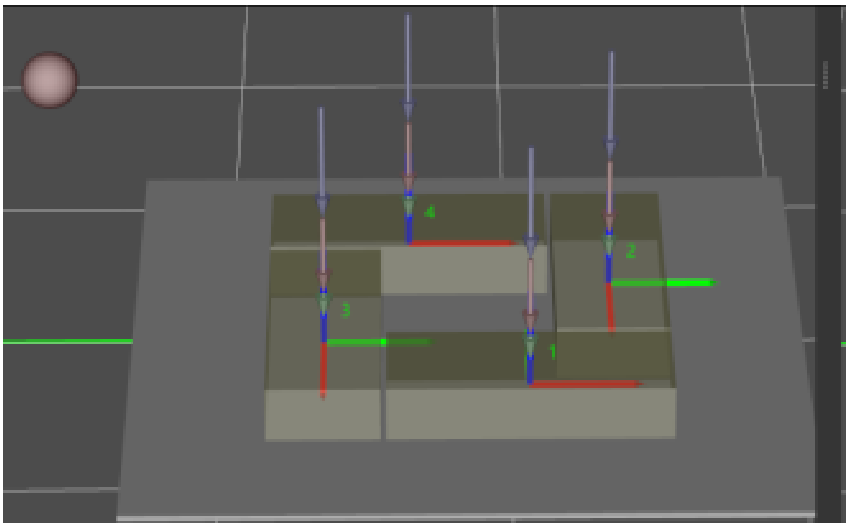


Is Middle Point Traj Vertical

Unselected by default, i.e., the robot will approach the stack from the auto middle point.



Once this option is selected, the robot will approach the stack from directly above each placing position.



Extended Distance For Entry

For some circumstances, the vacuum gripper is too large, and it may collide with the palletized boxes when approaching the stack. The Extended Distance For Entry can be set to avoid collisions in the approaching process and ensure safety.

> **Auto Middle Point** is introduced as a direction reference when the robot approaches the stack, and the point is not an actual waypoint where the robot will move. The auto middle point should be as far away from the pallet as possible. If the auto middle point is too close to the pallet, collisions may occur when palletizing.

Adjust Pallet via Vision

Adjust the position of the pallet dynamically via the "Vision Look" Step.

Need Adjust Pallet Pose

Unselected by default. Select to adjust the pallet position dynamically. When the workflow executes to this Step, the "Vision Look" Step will be used to recognize the position of the pallet.

Vision Name

Enter the name of the Mech-Vision project in which the "Vision Look" Step will be used to recognize the pallet position. When the workflow executes to this Step, the "Vision Look" Step in the Mech-Vision project will be used.

> The **parameters** can also be adjusted by the Adapter program.

Pallet Size

Pallet X Len	Set the pallet edge length (mm) along the X-direction. The pallet pattern will change with the pallet size in real-time.
Pallet Y Len	Set the pallet edge length (mm) along the Y-direction. The pallet pattern will change with the pallet size in real-time.

For Depalletization

As Vision Service

It indicates that the pallet pattern is used for depalletization. Once this parameter is selected, you can select the “Predefined Pallet Pattern” Step as vision service in the “Vision Move” Step.

Common Settings

Geometry Type

Currently supported: Sphere, Cuboid, Cylinder, Disc, Quadrangular prism, and Hexagonal prism. Please select the geometry type according to actual situation.

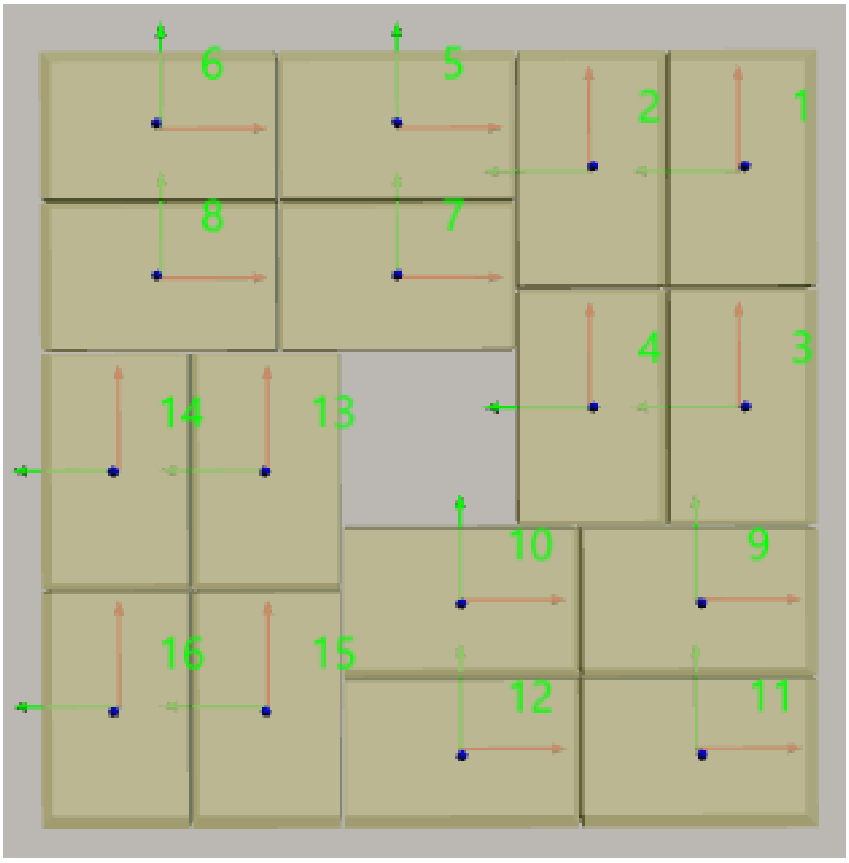
Pallet Type

Currently supported: Block, Pinwheel 1, Pinwheel 2, Automatic layout, and three special patterns. Please select the pallet type according to actual need.

The parameters that can be set are slightly different when the pallet type is different.

Pinwheel

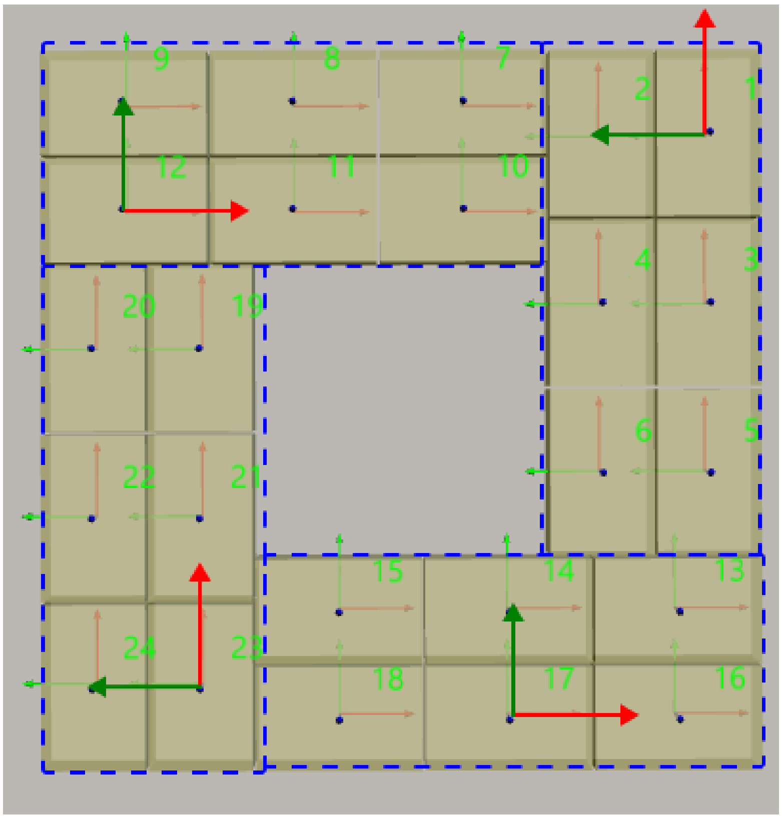
The pinwheel pallet pattern is shown in the following figure.



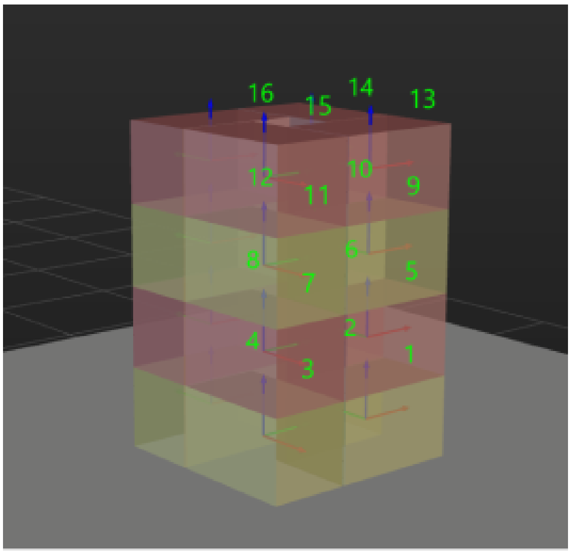
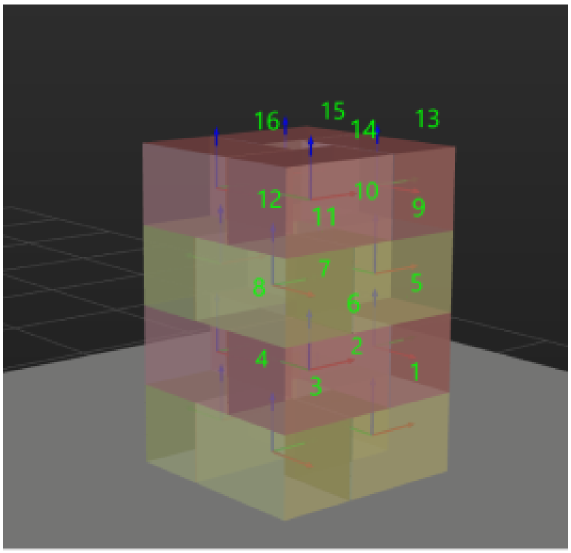
- **Rows/Unit**: the rows of the stacked boxes in each pallet pattern unit.
- **Cols/Unit**: the columns of the stacked boxes in each pallet pattern unit.

  In the following figure, the Rows/Unit is 3, and the Cols/Unit is 2:

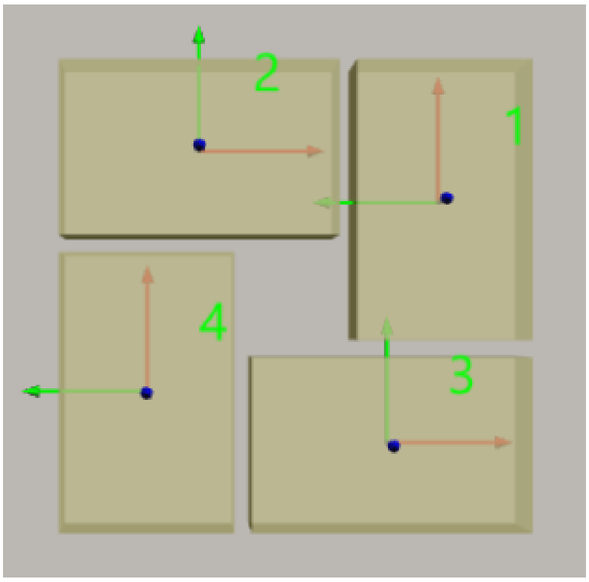
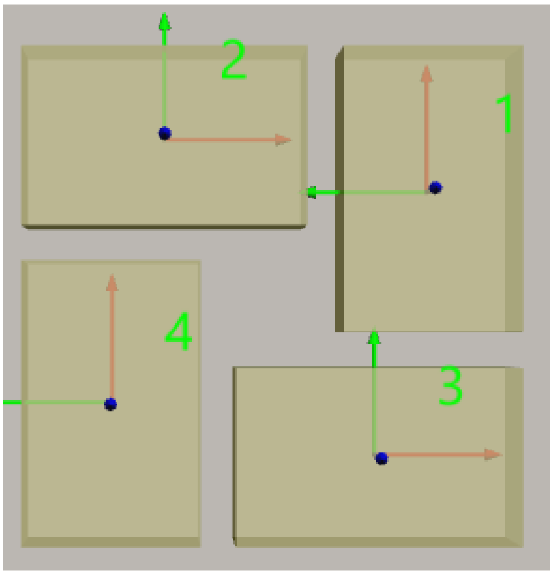
- A blue dashed box is a unit.
- In the cell, the X-direction (red arrow) of the box pose represents the column, and the Y-direction (green arrow) represents the row.



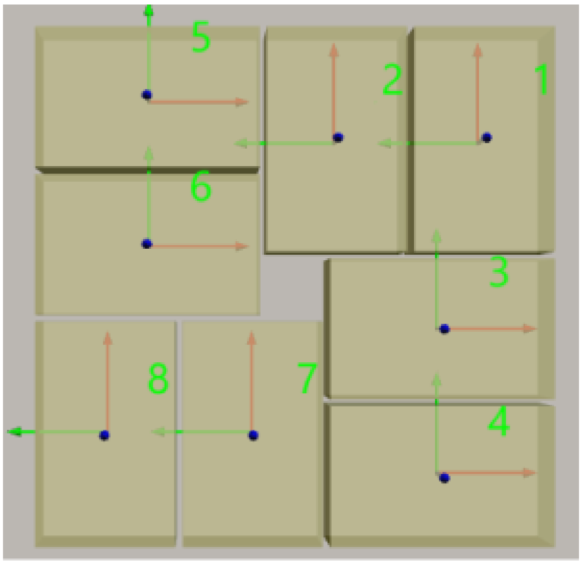
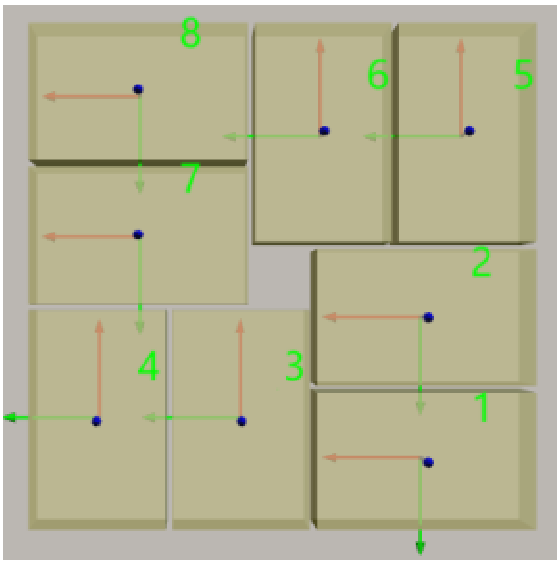
- **Is Antithetic**: Select to stack the boxes in different orientations on the odd and even layers, which makes the stack more stable.

 16 15 14 13 12 10 11 9 8 6 7 5 4 2 3 1	 16 15 14 13 12 10 11 9 8 7 6 5 4 2 3 1
Unselected	Selected

- **Gap Width**: Set the gap width of the boxes in the pallet pattern.

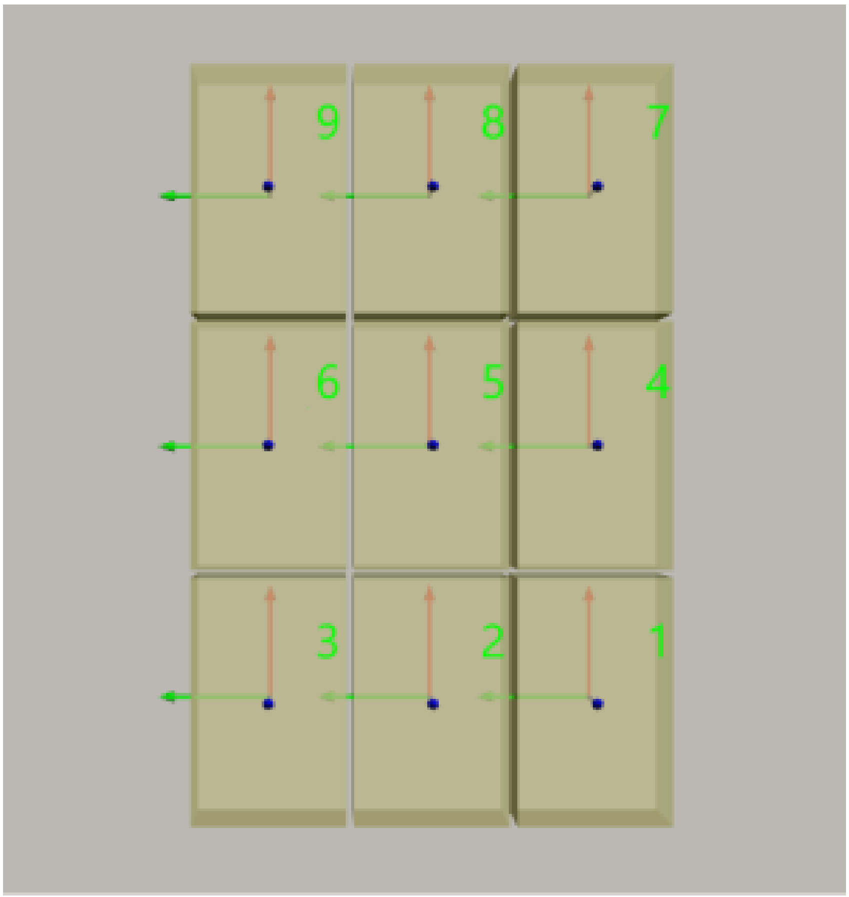
	
10mm	20mm

- **Rotate 90° Around Z-axis**: Rotate the whole pallet pattern 90°around Z-aixs.

	
Not rotated	Rotated

Block

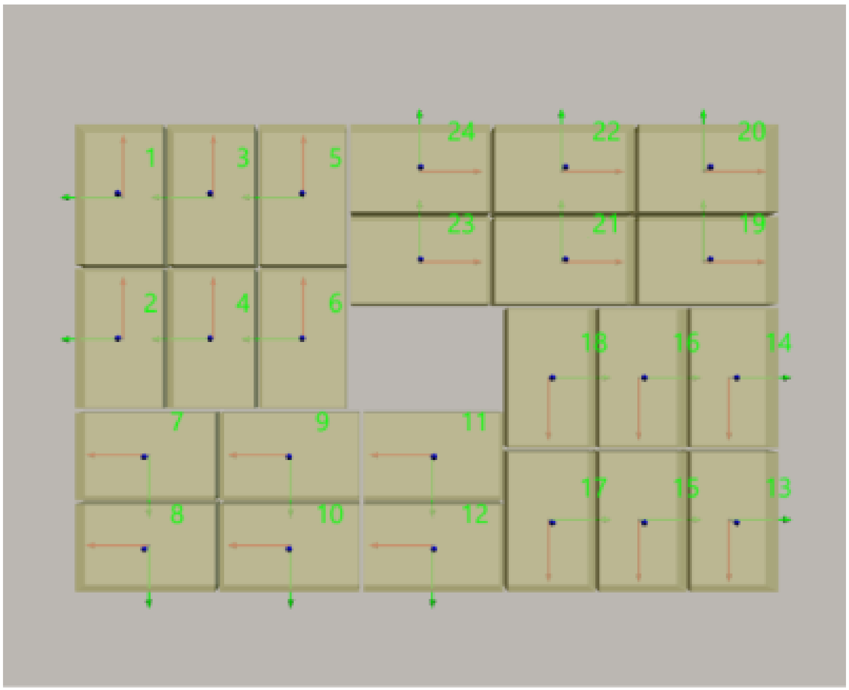
The block pallet pattern is shown in the following figure.



Rows/Unit	The rows of the stacked boxes per layer.
Cols/Unit	The columns of the stacked boxes per layer.
Transformation between Odd and Even Layers	Stacking the boxes in different orientations on the odd and even layers can make the stack more stable. Five transformation methods are provided in the right drop-down menu.
Gap Width	Set the gap width of the boxes in the pallet pattern.
Adjust to Square	Adjust the pallet pattern to square.
Rotate 90° Around Z-axis	Rotate the whole pallet pattern 90°around Z-axis.

Pinwheel 2

The pallet pattern is shown in the following figure.



Is Antithetic	Stacking the boxes in different orientations on the odd and even layers can make the stack more stable.
Gap Width	Set the gap width of the boxes in the pallet pattern.
Rotate 90° Around Z-axis	Rotate the whole pallet pattern 90°around Z-axis.
X-Oriented Boxes’ Num of Rows/Cols	Set the number of rows and columns of boxes in the X-orientation.
Y-Oriented Boxes’ Num of Rows/Cols	Set the number of rows and columns of boxes in the Y-orientation.

Automatic layout

Stack Generation Preference	Carton quantity first: Generate pallet pattern for as many boxes as possible, provided that other parameter constraints are met.
	Stack stability first: Generate more stable pallet pattern as much as possible, provided that other parameter constraints are met.
Gap Width	Set the gap width of the boxes in the pallet pattern.

Stack Height Settings

Specified Layer Count	Set the number of layers of the boxes in the pallet pattern.
Limit Stack Height	Once this parameter is selected, you can set the maximum height of the pallet pattern.
Limit Box Count	Once this parameter is selected, you can set the maximum number of the boxes for the pallet pattern.

Cuboid Settings

Carton Length	This group of parameters can be used to set the length, width and height of the boxes in the pallet pattern. Please set these parameters according to the actual situation.
Carton Width	
Carton Height	

5.5.4.5. Vision Continue Palletizing

Function

This Step is used to continue to palletize on the pallet that has already been stacked with boxes by using “Vision Look” to recognize vacant positions.

Usage Scenario

This Step is used in scenarios where there is a relatively high requirement for space utilization (such as the 3D dimensional storehouse). You can merge the unfilled pallets.

The relevant example project to the “Vision Continue Palletizing” Step is provided to help you learn more about the process of continuing the palletizing through vision. For the example project, you can go to File > New > Example projects > Box palletizing > Vision Continue Palletizing in the software.

Parameter Description

Move-Type Step Common Parameters

▼ *Send Waypoint*

Selected by default to send waypoint poses to the receiver, such as the robot. When this option is unselected, the waypoint pose will not be sent. However, the waypoint will remain in the planned path.

▼ *Try Continuously Running through Succeeding Non-Moves*

Unselected by default. When **non-move Steps**, such as Vision Look, Set DO, Check DI, etc., are connected between **move-type Steps**, the robot’s path planning will be interrupted, and the actual robot will take a short pause, reducing the smoothness of running.

When this option is selected, the project will continue to run without waiting for the current move-type Step to complete execution, and therefore the robot can move in a smooth way without pauses. However, enabling this option may cause the execution of the Step to end prematurely.

Why will this option cause the execution of the Step to end prematurely?

Mech-Viz will send multiple poses simultaneously to the robot when the project is running. When the currently returned JPs of the robot correspond to the last pose sent by Mech-Viz, Mech-Viz will assume that the robot has moved to the last position.

For example, there are 10 move-type Steps in a path, and the pose of the 5th move-type Step is the same as that of the last move-type Step. When the robot moves at low speed, it sends JPs to Mech-Viz when it moves to the 5th move-type Step, Mech-Viz may mistakenly determine that the robot has finished the move-type Steps and prematurely ends the Steps since the poses of the 5th move-type Step and the last move-type Step are the same in the path.

▼ *Do Not Check Collision with Placed Workobject*

Unselected by default, namely that the collision with the already placed objects will not be detected. When this option is selected, the collisions between the robot, end tool, and placed objects will be detected.

In palletizing scenarios, the two possible cases of error are as follows:

1. When the robot is placing a carton, the robot may come into light contact with the placed cartons while no deformation will be caused. After Mech-Viz detects this collision in simulation, it will plan other positions for placing the carton, and therefore a full stack cannot be formed.
2. Usually, the TCP of a suction cup is inside the suction cup model instead of on the surface of it. Under this circumstance, the suction cup may be embedded in the model of the picked carton in the simulation of picking, while the software does not detect the collision between the end tool and the picked object. After the robot places the object and the carton model turns into an object model in the scene, a collision between the suction cup and the carton will be detected and the palletizing cannot be completed.

When this option is selected, no collision between the robot, end tool, and the placed object will be detected, and the above two cases of errors can be avoided.

▼ *Point Cloud Collision Detection Mode*

Select the proper mode according to the requirement of the on-site situation. Usually, the default setting **Auto** can be used. **Do not check collision** mode can be used in move-type Steps before the robot picks the object, and **Check collision** mode can be used after the robot picks the object.

Auto	Default setting. Collision with point cloud is checked only for the “Vision Move” Step and the “Relative Move” Step that depends on the “Vision Move” Step, but not for all move-type Steps.
Do not check collision	Point cloud collisions for all move-type Steps will not be detected.
Check collision	Point cloud collisions for all move-type Steps will be detected.

When Collisions › Collision detection configuration › Detect collision between point cloud and others is switched on, Mech-Viz will detect collisions between the robot model, end tool model, and point cloud when planning the path. Normally, detects whether the robot collides with other objects during picking and placing. When there are point cloud outliers, non-exiting collisions will be detected, which leads to errors in path planning.

▼ *Ignore Workobject Symmetry*

This parameter will only take effect when **Waypoint type** of the Step is set to **Workobject pose**.

None	Default setting, i.e., do not disable symmetry on any axis.
Around workobject frame Z axis	Only disable symmetry on Z-axis of the workobject reference frame.
Around workobject frame X&Y axis	Disable symmetry on X-axis and Y-axis of the workobject reference frame.

Around all axes	Once the object symmetry is disabled, the robot will place the objects strictly according to the workobject poses.

In some special cases, objects are not pickable due to their peculiar poses. Setting **Rotational symmetry** under Workobjects › Workobject Configuration in Resources may solve this problem. Based on the settings of Rotational symmetry, when the default object pose is not pickable, Mech-Viz will calculate candidate poses and determine whether the candidate poses are pickable. As the candidate poses calculated based on the settings of Rotational symmetry are different from the original poses output from Mech-Vision, the consistency of the objects' place poses cannot be guaranteed.

▼ *Plan Failure Out Port*

After selecting this parameter, this Step will add "Plan failure" out port.

During the planning process, planning is carried out along the branch after the "Success" out port; if the planning fails in the current Step, the branch process after the "Planning failure" out port will be executed.

Held Workobject Collision Detection Settings

▼ *Do Not Check Collision with Scene Object/Robot*

Unselected by default. Once this option is selected, the collisions between the **held workobject** with the **scene objects or robot** will not be detected, and therefore the calculation workload of collision detection will be reduced, the planning speed can be increased, and the cycle time can be shortened. It is usually enabled in the first one or two move-type Steps after the robot picks the object.

Please enable this option cautiously as there may be collision risks.

When **Detect collision between held workobject and others** under Collision detection configuration › Configuration on held workobject is enabled, the software will detect whether the model of the held object collides with the models of the scene objects and the robot.

In palletizing projects, the calculated carton dimensions have millimeter-level errors with the actual dimensions, and frictions between cartons may occur during picking but no collisions will occur. For some move-type Steps that will obviously not cause collisions, detecting such collisions only adds to the calculation workload and planning time, and consequently extending the cycle time. In palletizing projects, enabling Do Not Check Collision with Scene Objects does not affect the collision detection between the held carton and the placed cartons. This option can be enabled when there are scene objects under the stack to avoid failure of finding the palletizing solution.

▼ *Do Not Check Collision with Point Cloud*

Unselected by default. Once this option is selected, the collisions between the **held workobject** with the **point clouds in the scene** will not be detected, and therefore the calculation workload of collision detection will be reduced, the planning speed can be increased, and the cycle time can be shortened.

- When both **Detect collision between held workobject and others** under Collision detection configuration › Configuration on held workobject and **Configuration on point cloud ¬ Detect collision between point cloud and others** are enabled, the software will detect whether the model of the held workobject will collide with the point cloud in the scene.
- When Mech-Vision sends the point cloud and object model to Mech-Viz, the point cloud and the object model are fitted together. After the robot picks the object, the model moves along

the planned path, and the collision between the model of the held workobject and the point cloud will occur.

- It is known that the model of the held workobject will have false collisions with the point cloud. Detecting such collisions unnecessarily adds to the calculation workload and extends the planning time.

Visual Match

Match Max Dist Error

Specify the maximum allowable distance error between the box pose inherited from the palletizing Step and the box pose recognized by the vision service in the X and Y-axis direction. The matching fails if the value exceeds the threshold.

Z Dist Tol

Specify the maximum allowable distance error between the box pose inherited from the palletizing Step and the box pose recognized by the vision service in the Z-axis direction. The matching fails if the value exceeds the threshold.

Max Rot Correction

Specify the maximum allowable rotation error between the box pose inherited from the palletizing Step and the box pose recognized by the vision service around the Z-axis. The matching fails if the value exceeds the threshold.

Match Symmetry

In actual application, the box orientation provided by the vision service and the box orientation inherited from the specified pallet pattern may differ, resulting in a matching failure. This feature can be used to fix this problem.

Option	Description
Auto	Use “Symmetry90Deg” when the difference between the sides of the boxes is less than 2 cm, and use “Symmetry180Deg” when the difference between the sides of the boxes is more than 2 cm.
NoSymmetry	No rotational symmetry will be applied.
Symmetry180Deg	Select this option when the XY-plane is a rectangle.
Symmetry90Deg	Select this option when the XY-plane is a square.

Correction

Correction Type

After the vision matching is completed, correct the pose of the pallet pattern based on the average error of the matched box poses.

Option	Description
Total	Correct both the translation error and rotation error.

Option	Description
TranslationOnly	Only correct the translation in the X-, Y- and Z-direction.
RotationOnly	Only correct the rotation around Z-axis.

J

Box Label of Interest

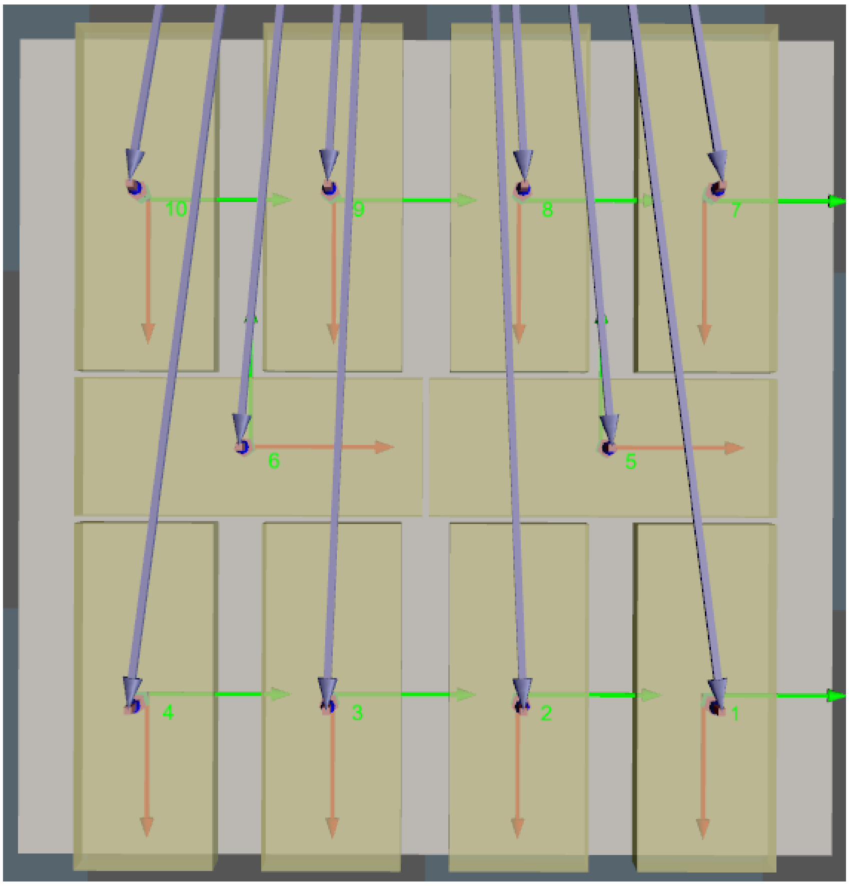
Boxes are filtered based on labels provided by vision projects, and only those with the specified labels will be used for visual matching. If no object label is set, all boxes recognized by vision projects will be used for visual matching.

Check Placed Boxes Order

Select to check the order of the palletized boxes in the project.

The requirements for continuing palletizing are that the box serial number must be uninterrupted and that serial number 1 must not be empty. If these requirements are not met, an error will be reported.

Take the following pallet pattern for example: There are 9 boxes on one layer, and the picking order will be calculated automatically.



For the following situations:

- If position indicated by 1 is not occupied with a box, and position 2 and 3 are occupied with boxes, an error will occur.

- If positions indicated by 1, 2, 3, 4, and 6 are occupied, while position 5 is not occupied, an error will occur.
- The order number of the palletized box starts from 1 and the subsequent numbers are continuous, the project runs normally.

Select Palletizing Step

This Step does not generate pallet patterns; instead, it requires selecting other palletizing Steps that can configure pallet patterns to obtain actual pallet pattern information.

Vision Service Name

The Vision Continue Palletizing Step recognizes and locates the boxes on the pallet by the vision project. This parameter can specify the vision project that is used to recognize and locate the boxes on the pallet.

> The vision result generally consists of the pose of the recognized box and the pose of the recognized pallet. The pallet pose is not mandatory data. When the vision result does not contain the pose of the pallet, the pallet pose of the specified pallet pattern will be inherited. If the actual pallet pose differs greatly from the set one, it is recommended to add the pallet pose to ensure the matching accuracy.

5.5.5. Robot Utilities

5.5.5.1. Check Tool

Function

This Step is used in scenarios where multiple robot tools are used. It is used to check if the current tool is the same as the selected tool. If the same, it will proceed along the first exit port, without executing the Steps for switching tools. If not the same, it will proceed along the second exit port and execute the Steps for switching tools.

Parameter Description

General Parameters of Non-Move Steps

▼ *Skip Execution*

None
: Default setting. Do not skip the current Step.

Simulation only
: Skip the current Step during simulation. The exit port is specified by **Out Port When Skip**.

Always
: Skip the current Step when the project is simulating or running. The exit port is specified by **Out Port When Skip**.

Instruction
: When **Simulation only** or **Always** is selected, the current Step will be skipped and the subsequent Step will be executed when running the project. If this parameter is set to None in “Check DI” Step in the project, and there is no external input signal during simulation, the project will be stopped when executing to this Step. In this case, setting the parameter to Simulation only or Always enables the simulation to continue.

▼ *Out Port When Skip*

This parameter will take effect when **Skip Execution** is set to **Simulation only** or **Always**. It specifies the exit port when skipping a Step.

Set tool as...

Select the model of the expected robot tool to check if the current tool is right.

5.5.5.2. Get JPs

Function

This Step is used to obtain the current joint positions of the connected robot and synchronize the joint positions to Mech-Viz.

Usage Scenario

If Mech-Viz does not fully control all movements of the robot, you can use this Step to obtain the current joint positions of the robot after it has moved according to its control program.

Parameter Description

General Parameters of Non-Move Steps

▼ *Skip Execution*

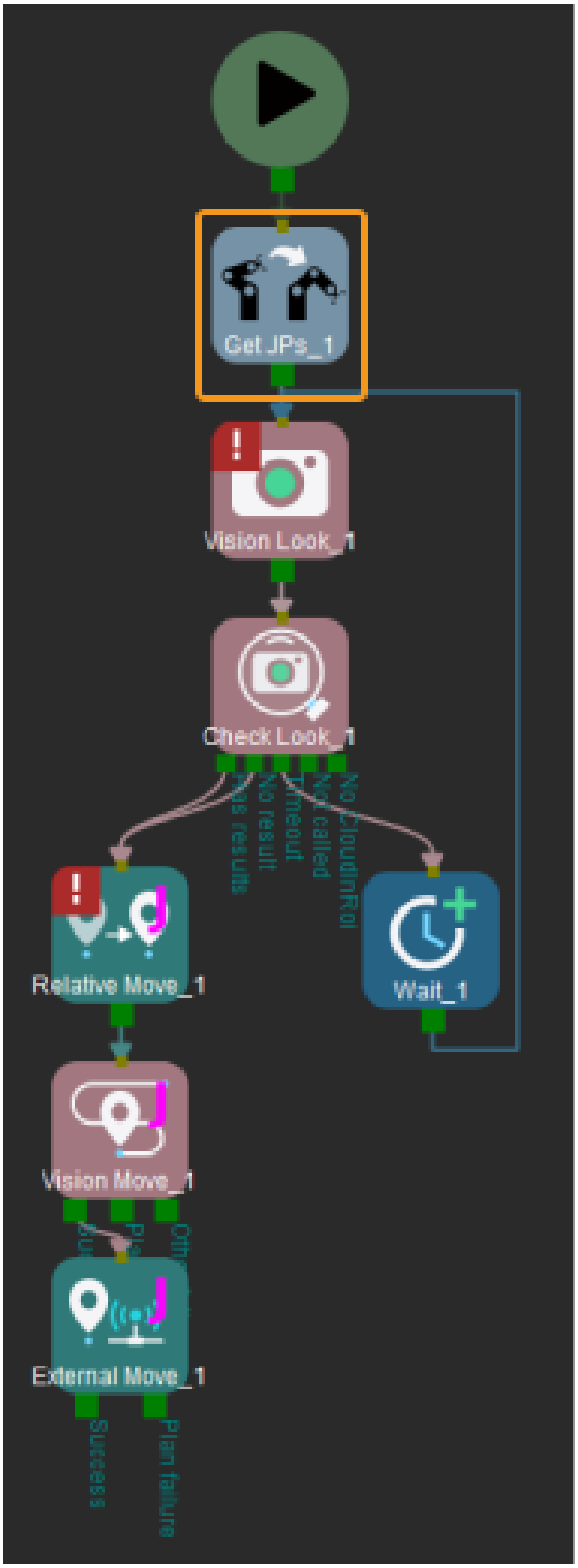
None	Default setting. Do not skip the current Step.
Simulation only	Skip the current Step during simulation. The exit port is specified by **Out Port When Skip**.
Always	Skip the current Step when the project is simulating or running. The exit port is specified by **Out Port When Skip**.
Instruction	When **Simulation only** or **Always** is selected, the current Step will be skipped and the subsequent Step will be executed when running the project. If this parameter is set to None in “Check DI” Step in the project, and there is no external input signal during simulation, the project will be stopped when executing to this Step. In this case, setting the parameter to Simulation only or Always enables the simulation to continue.

▼ *Out Port When Skip*

This parameter will take effect when **Skip Execution** is set to **Simulation only** or **Always**. It specifies the exit port when skipping a Step.

Application Examples

Before the **Vision Look** Step triggers the corresponding Mech-Vision project, Mech-Viz obtains the joint positions of the real robot through the **Get JPs** Step and uses the obtained joint positions as the starting position for path planning.


Get JPs_1
Vision Look_1
Check Look_1
Has results
No result
Timeout
Not called
NoCloudInRoI
Relative Move_1
Wait_1
Vision Move_1
External Move_1
Success
Plan failure

5.5.5.3. Set Payload

Function

This Step is used for setting the payload of robot.

Parameter Description

General Parameters of Non-Move Steps

▼ *Skip Execution*

None	Default setting. Do not skip the current Step.
Simulation only	Skip the current Step during simulation. The exit port is specified by **Out Port When Skip**.
Always	Skip the current Step when the project is simulating or running. The exit port is specified by **Out Port When Skip**.
Instruction	When **Simulation only** or **Always** is selected, the current Step will be skipped and the subsequent Step will be executed when running the project. If this parameter is set to None in “Check DI” Step in the project, and there is no external input signal during simulation, the project will be stopped when executing to this Step. In this case, setting the parameter to Simulation only or Always enables the simulation to continue.

▼ *Out Port When Skip*

This parameter will take effect when **Skip Execution** is set to **Simulation only** or **Always**. It specifies the exit port when skipping a Step.

Payload Id

If multiple payloads are used, the index of the payload in use should be specified. If only one payload is used, use the default value.

Payload

Set to the maximum payload (in kg) of the robot tool and workobject.

COG Position

X/Y/Z: Set the center of gravity of the payload in the flange coordinate system (only required by some robot brands).

5.5.5.4. Set Holding State

Function

This Step is used to set the holding state. It is generally used for updating the holding state after the robot task has failed.

Usage Scenario

This Step is used in conjunction with the Pick or Place parameter of move-type steps to update the holding state.

Parameter Description

General Parameters of Non-Move Steps

▼ *Skip Execution*

None	Default setting. Do not skip the current Step.
Simulation only	Skip the current Step during simulation. The exit port is specified by **Out Port When Skip**.
Always	Skip the current Step when the project is simulating or running. The exit port is specified by **Out Port When Skip**.
Instruction	When **Simulation only** or **Always** is selected, the current Step will be skipped and the subsequent Step will be executed when running the project. If this parameter is set to None in “Check DI” Step in the project, and there is no external input signal during simulation, the project will be stopped when executing to this Step. In this case, setting the parameter to Simulation only or Always enables the simulation to continue.

▼ *Out Port When Skip*

This parameter will take effect when **Skip Execution** is set to **Simulation only** or **Always**. It specifies the exit port when skipping a Step.

Pick State

Select Hold or None according to the actual workflow of the project.

5.5.5.5. Change Tool

Function

Switch TCP and tool models.

- Software side: Switch TCP and tool models.
- Robot side: Switch TCP. To switch tools on the real robot or change the status of the tools in actual situation, you should use this Step with the “Set DO” Step.

Usage Scenario

You need to switch tools in the process of picking.

Parameter Description

General Parameters of Non-Move Steps

▼ *Skip Execution*

None

Default setting. Do not skip the current Step.

Simulation only

Skip the current Step during simulation. The exit port is specified by **Out Port When Skip**.

Always

Skip the current Step when the project is simulating or running. The exit port is specified by **Out Port When Skip**.

Instruction

When **Simulation only** or **Always** is selected, the current Step will be skipped and the subsequent Step will be executed when running the project. If this parameter is set to None in “Check DI” Step in the project, and there is no external input signal during simulation, the project will be stopped when executing to this Step. In this case, setting the parameter to Simulation only or Always enables the simulation to continue.

▼ *Out Port When Skip*

This parameter will take effect when **Skip Execution** is set to **Simulation only** or **Always**. It specifies the exit port when skipping a Step.

Tool

Select the tool that you need to use currently.

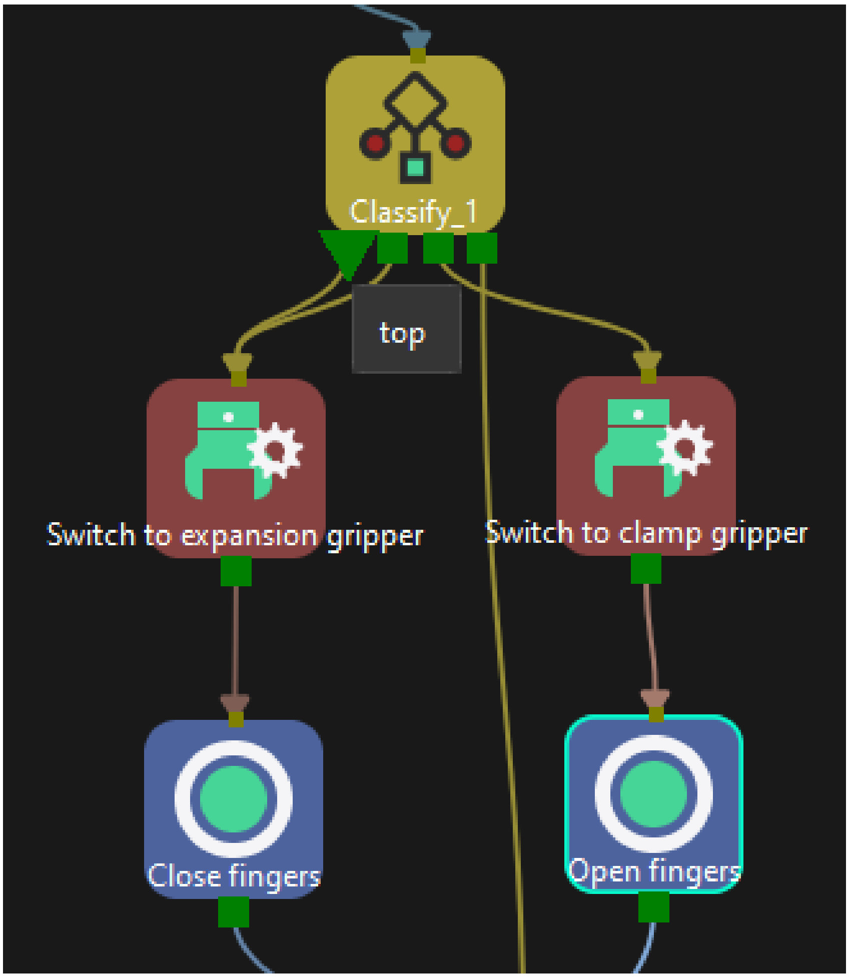
Before selecting the tool, you need to add and configure the corresponding tool in project resource tree—Tools.

Application Examples

Take the example project of “Switch End Tools According to Pick Point Labels” for example.

For the workpiece of bearing pedestal in this example project, you can use expansion gripper to grasp the workpiece inside its central hole, and use finger gripper to grasp it through the edge of the tray. According to pick point types, add **top**, **bottom**, and **edge** labels in the “Classify” Step:

- For workpieces with the **top** and **bottom** labels, close the fingers of the claw gripper, and then open the fingers to grasp the workpieces.
- For workpieces with the **edge** label, open the fingers of the claw gripper, and then close the fingers to grasp the workpieces.



5.5.5.6. Transfer Control

Function

This Step is used to temporarily transfer the control to the robot and wait for the control to be transferred back.

Parameter Description

General Parameters of Non-Move Steps

▼ *Skip Execution*

None	Default setting. Do not skip the current Step.
Simulation only	Skip the current Step during simulation. The exit port is specified by **Out Port When Skip**.
Always	Skip the current Step when the project is simulating or running. The exit port is specified by **Out Port When Skip**.

Instruction	When **Simulation only** or **Always** is selected, the current Step will be skipped and the subsequent Step will be executed when running the project. If this parameter is set to None in “Check DI” Step in the project, and there is no external input signal during simulation, the project will be stopped when executing to this Step. In this case, setting the parameter to Simulation only or Always enables the simulation to continue.

▼ *Out Port When Skip*

This parameter will take effect when **Skip Execution** is set to **Simulation only** or **Always**. It specifies the exit port when skipping a Step.

Size

Set the number of exit ports.

Check Jps

Select whether to check the joint positions of the robot before transferring control.

Jps Before Transfer

This option will only appear when the **Check Jps** option is checked. Please enter the joint positions required for checking.

5.5.6. Service

5.5.6.1. Notify

Function

This Step is used to send messages to external services.

Usage Scenario

This Step is usually used in scenarios where an external device is required for the communication.

Parameter Description

General Parameters of Non-Move Steps

▼ *Skip Execution*

None	Default setting. Do not skip the current Step.
Simulation only	Skip the current Step during simulation. The exit port is specified by **Out Port When Skip**.
Always	Skip the current Step when the project is simulating or running. The exit port is specified by **Out Port When Skip**.

Instruction	When **Simulation only** or **Always** is selected, the current Step will be skipped and the subsequent Step will be executed when running the project. If this parameter is set to None in “Check DI” Step in the project, and there is no external input signal during simulation, the project will be stopped when executing to this Step. In this case, setting the parameter to Simulation only or Always enables the simulation to continue.

▼ *Out Port When Skip*

This parameter will take effect when **Skip Execution** is set to **Simulation only** or **Always**. It specifies the exit port when skipping a Step.

Adapter Name

This parameter specifies the Adapter that will receive the message from this Step. If there are multiple **Notify** Steps in the project, the **Adapter Name** set in the Steps should be the same.

Message

The customized message. The Adapter executes different logic according to the content of the message. For example, you can set the message to **finish** if the branch operation ends.

Action When Failed

Description: This parameter specifies the action of the software when the notification message is not sent successfully.

Value:

- Warning: Send a warning message (default value).
- Retry: Try to re-send the message.
- Throw: Discard the notify Step.

Need Robot Stop

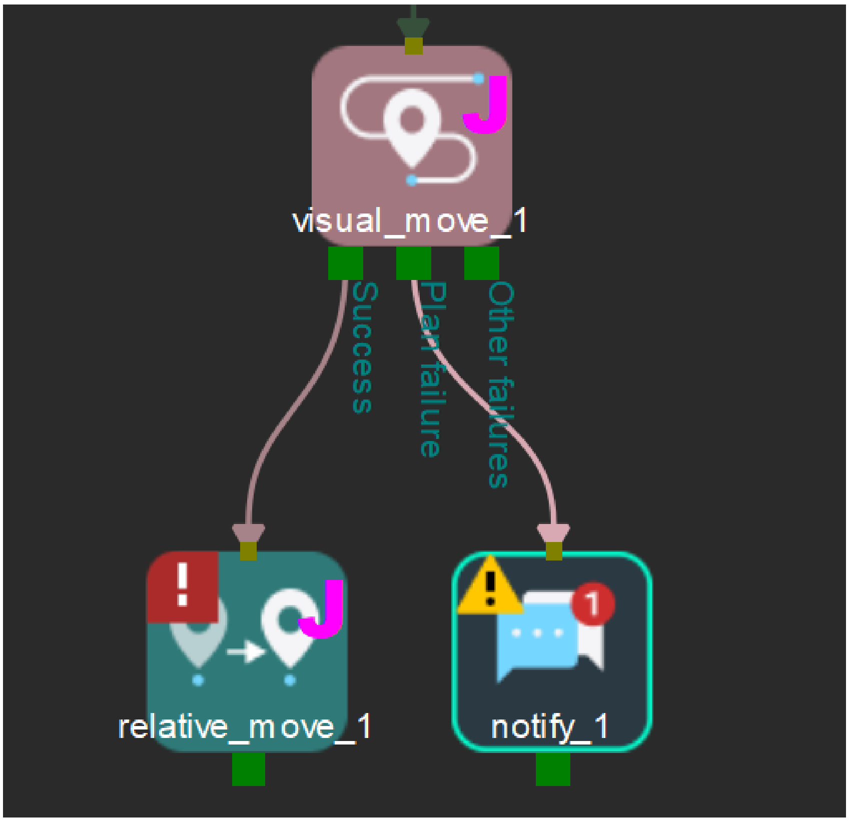
If this parameter is selected, the robot will pause when the project executes to this Step. If this parameter is not selected, the robot will keep moving while the Step sends the message.

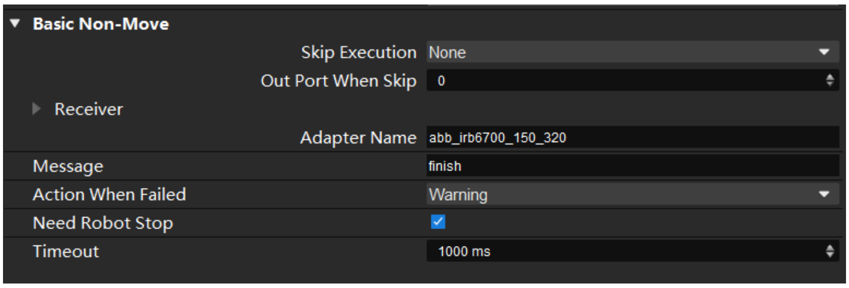
Timeout

If the message fails to be sent after the timeout period (unit: ms), the robot will perform the **Action When Failed**.

Application Examples

As shown in the figures below, an **Adapter Name** is set, and a **finish** message will be sent to the Adapter if the software fails to plan a path.





▼ **Basic Non-Move**

Skip Execution	None
Out Port When Skip	0
▸ Receiver	
Adapter Name	abb_irb6700_150_320
Message	finish
Action When Failed	Warning
Need Robot Stop	☑
Timeout	1000 ms

5.5.7. Utilities

5.5.7.1. Classify

Function

This Step is used before or after Vision Move to choose different branches according to the workobject labels identified by Mech-Vision.

Usage Scenario

This Step is used to classify the workobject labels identified by Mech-Vision.

- Before “Vision Move”: The default picking process: Identify the workobject > Select and operate on different branches according object’s labels > Pick workobjects with the label. Example: Change the gripper for different workobjects.
- After “Vision Move”: The default picking process: Identify the object > Pick the object > Operate differently according to the types of workobjects. Example: Use the same gripper to pick and place workobjects into different bins after classification.

Parameter Description

General Parameters of Non-Move Steps

▼ *Skip Execution*

None
Default setting. Do not skip the current Step.

Simulation only
Skip the current Step during simulation. The exit port is specified by **Out Port When Skip**.

Always
Skip the current Step when the project is simulating or running. The exit port is specified by **Out Port When Skip**.

Instruction
When **Simulation only** or **Always** is selected, the current Step will be skipped and the subsequent Step will be executed when running the project. If this parameter is set to None in "Check DI" Step in the project, and there is no external input signal during simulation, the project will be stopped when executing to this Step. In this case, setting the parameter to Simulation only or Always enables the simulation to continue.

▼ *Out Port When Skip*

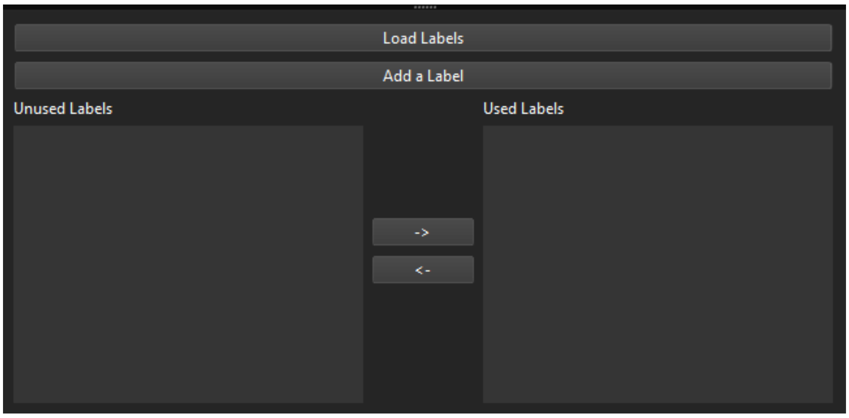
This parameter will take effect when **Skip Execution** is set to **Simulation only** or **Always**. It specifies the exit port when skipping a Step.

Get Label from "Update Held Workobject"

Selected
An Update Held Workobject Step should be used before this Step, and this Step can acquire the label information of the currently picked workobject and then perform operations according to the classification. Usage Scenario: The label information of the detected workobjects cannot be acquired directly due to the overlapped workpieces or other reasons; therefore, the held workpieces should be re-classified, and then corresponding labels can be output.

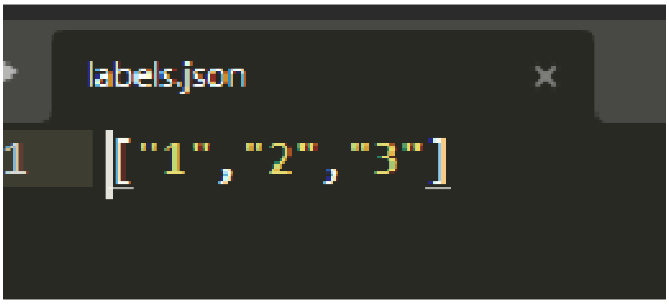
Unselected
The default setting. Use the object labels detected by Mech-Vision. Please select the corresponding Vision Move Step for **Classify before "Vision Move"**.

Load Labels & Add a label



Load Labels

Add multiple labels to **Unused Labels** at once. Click **Load Labels** and select a **JSON** file with multiple labels, as shown below. After that, all label files will be added to **Unused Labels**, and all the previous labels in both the Unused Labels and the Used Labels will be deleted.



```
["1","2","3"]
```

Add a label

Add a label named **unnamed_label** to **Unused Labels**. You can double-click the label to re-name it.

Label list

The label list includes **Unused Labels** and the **Used Labels**. You can shift the label's place between the two spaces by clicking -> or <-.

The **Used Labels** will appear at the exit port of this Step. There will be a label named **Other** to filter out labels that are not in the **Used Labels** space.

5.5.7.2. Counter

Function

Count the execution times of the current program or branch. You can set a count threshold and execute different branches when the count meets the threshold. This Step is usually used with Reset to reset the accumulated number.

Usage Scenario

This Step is used when the execution times need to be counted.

Parameter Description

General Parameters of Non-Move Steps

▼ *Skip Execution*

None

Default setting. Do not skip the current Step.

Simulation only

Skip the current Step during simulation. The exit port is specified by **Out Port When Skip**.

Always

Skip the current Step when the project is simulating or running. The exit port is specified by **Out Port When Skip**.

Instruction	When **Simulation only** or **Always** is selected, the current Step will be skipped and the subsequent Step will be executed when running the project. If this parameter is set to None in “Check DI” Step in the project, and there is no external input signal during simulation, the project will be stopped when executing to this Step. In this case, setting the parameter to Simulation only or Always enables the simulation to continue.

▼ *Out Port When Skip*

This parameter will take effect when **Skip Execution** is set to **Simulation only** or **Always**. It specifies the exit port when skipping a Step.

Count

The number of times to count. When the Step executes for more than the set value, the **Finished** exit port will be taken.

Counter Type

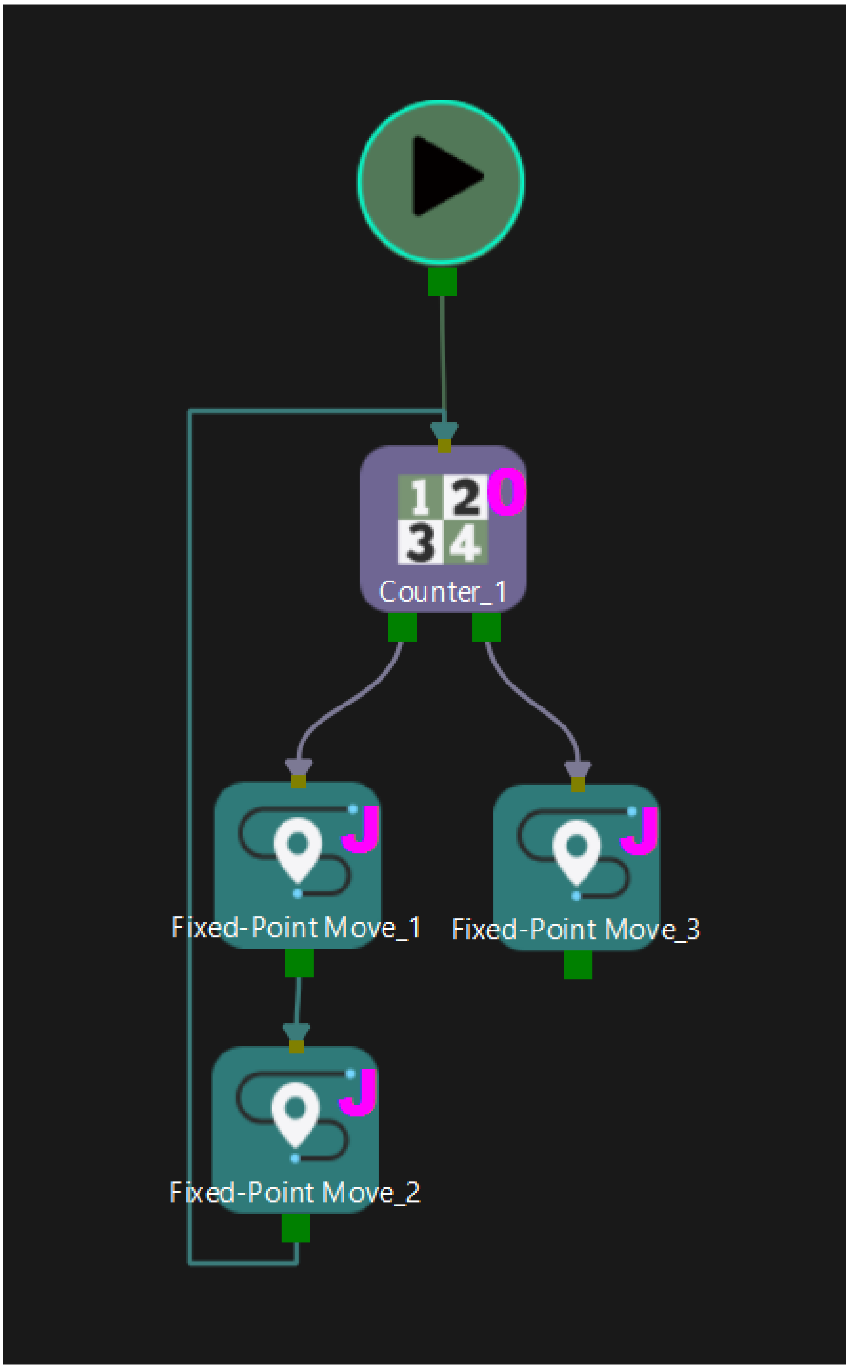
Execution: Count the number of times the “Counter” Step is executed. Picked Count: Count the number of times the Vision Move Step is executed.

Current Count

The number of times the Counter or the Vision Move Step is executed, which is related to **Counter Type**.

Application Examples

In a scenario where a path needs to be taken twice:



Given the **Count** parameter set to 2, if the Step is executed the third time, the **Finished** exit port will be taken.

5.5.7.3. Check Completion

Function

This Step is used to check whether all the Basic Move Steps with index (Move by Grid, Move by List, etc) are executed in a single cycle to avoid loops.

Usage Scenario

This Step is usually used together with the Move by Grid and Move by List Steps.

If the Basic Move Steps with index are executed in a single cycle, the **Finished** exit port will be taken, or else the **Unfinished** port will be taken.

Parameter Description

General Parameters of Non-Move Steps

▼ *Skip Execution*

None	Default setting. Do not skip the current Step.
Simulation only	Skip the current Step during simulation. The exit port is specified by **Out Port When Skip**.
Always	Skip the current Step when the project is simulating or running. The exit port is specified by **Out Port When Skip**.
Instruction	When **Simulation only** or **Always** is selected, the current Step will be skipped and the subsequent Step will be executed when running the project. If this parameter is set to None in "Check DI" Step in the project, and there is no external input signal during simulation, the project will be stopped when executing to this Step. In this case, setting the parameter to Simulation only or Always enables the simulation to continue.

▼ *Out Port When Skip*

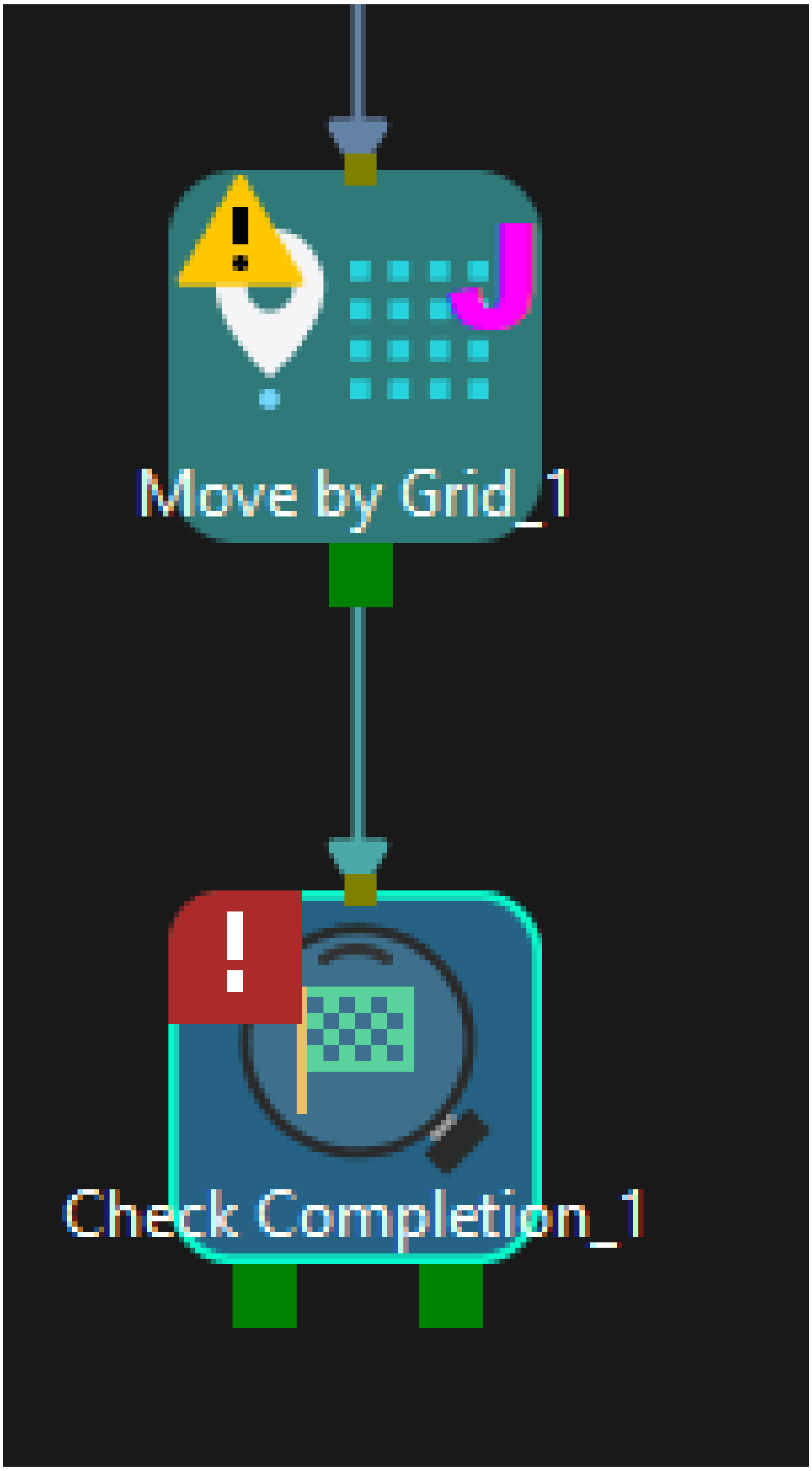
This parameter will take effect when **Skip Execution** is set to **Simulation only** or **Always**. It specifies the exit port when skipping a Step.

Select Move Step

Select the Basic Move Steps in the project.

Application Examples

Together with the Move by Grid Step, this Step determines whether the execution is completed successfully until the end of the loop.



5.5.7.4. Change Index

Function

This Step can be used with all the Basic Move Steps that have the **Current Index** parameter, or with the Step Counter.

Each time the Step executes, the **Current Index** in the Basic Move Steps or the **Current Count** in the

Counter Step will be changed according to the **Step**.

Usage Scenario

This Step is used with all the Basic Move Steps that have the **Current Index** parameter, or with the Step Counter.

Parameter Description

General Parameters of Non-Move Steps

▼ *Skip Execution*

None
Default setting. Do not skip the current Step.

Simulation only
Skip the current Step during simulation. The exit port is specified by **Out Port When Skip**.

Always
Skip the current Step when the project is simulating or running. The exit port is specified by **Out Port When Skip**.

Instruction
When **Simulation only** or **Always** is selected, the current Step will be skipped and the subsequent Step will be executed when running the project. If this parameter is set to None in “Check DI” Step in the project, and there is no external input signal during simulation, the project will be stopped when executing to this Step. In this case, setting the parameter to Simulation only or Always enables the simulation to continue.

▼ *Out Port When Skip*

This parameter will take effect when **Skip Execution** is set to **Simulation only** or **Always**. It specifies the exit port when skipping a Step.

Step

Default value: 0

> The **Current Index** of Basic Move Steps and the **Current Count** of the Step Counter will plus 1 for every execution according to their own rules.

Update the **Current Index** in the Basic Move Steps or the **Current Count** in the Counter Step according to the **Step** parameter.

Updated **Current Index** / **Current Count** = previous **Current Index** / **Current Count** + 1

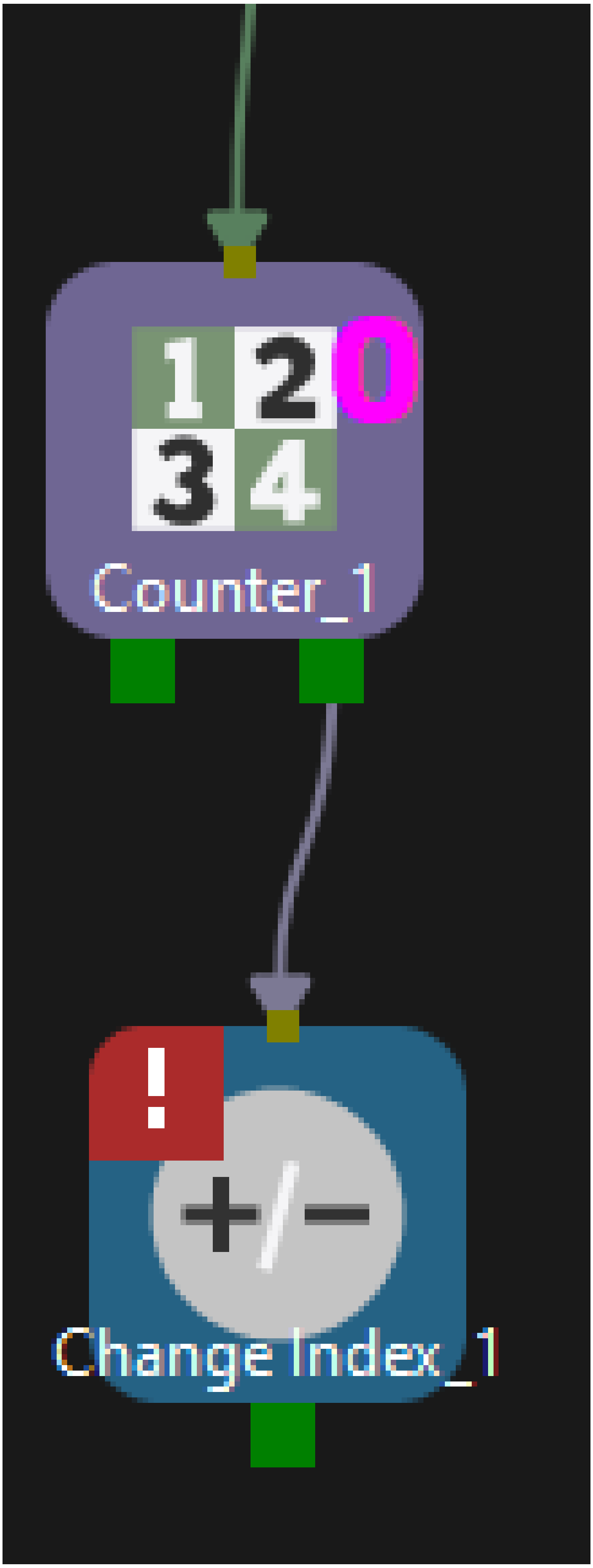
- If the Step parameter is 0, it will have no influence on the **Current Index** of Basic Move Steps and the **Current Count** of the Step Counter.
- If the Step parameter is −1, the **Current Index** of Basic Move Steps and the **Current Count** of the Step Counter will all minus 1.
- If the Step parameter is 1, the **Current Index** of Basic Move Steps and the **Current Count** of the Step Counter will all plus 1.

Select Move Step with Index or Counter

Select Basic Move Steps or the Counter Step according to your actual needs.

Application Examples

Connect this Step to the exit port of Counter and you can change the value of the **Current Count** parameter according to your need.


1 2 0
3 4
Counter_1
Change Index_1

If the Step parameter is −1, the **Current Count** of the Step Counter will minus 1 after every execution.

5.5.7.5. Reset

Function

Reset the selected Steps according to the setting.

Usage Scenario

This Steps is used for scenarios that require counting, determining picking and placing, re-using vision result, etc.

Parameter Description

General Parameters of Non-Move Steps

▼ *Skip Execution*

None	Default setting. Do not skip the current Step.
Simulation only	Skip the current Step during simulation. The exit port is specified by **Out Port When Skip**.
Always	Skip the current Step when the project is simulating or running. The exit port is specified by **Out Port When Skip**.
Instruction	When **Simulation only** or **Always** is selected, the current Step will be skipped and the subsequent Step will be executed when running the project. If this parameter is set to None in “Check DI” Step in the project, and there is no external input signal during simulation, the project will be stopped when executing to this Step. In this case, setting the parameter to Simulation only or Always enables the simulation to continue.

▼ *Out Port When Skip*

This parameter will take effect when **Skip Execution** is set to **Simulation only** or **Always**. It specifies the exit port when skipping a Step.

Partial Reset

Not selected by default. If this checkbox is not selected, reset will only apply to one Step that is executed before this Step, including Steps related to counting, determining picking and placing, and re-using vision result. Select the Step to be reset in **Select Step Name**.

If the checkbox is selected, and there are multiple Steps related to counting, determining picking and placing, and re-using vision result, you can select the Steps in **Select Step Name**, and the To Reset list will display all the Steps selected. Select the Steps to be reset according to your need.

Select Step Name

Select the Steps to be reset.

5.5.7.6. Wait

Function

Wait for the specified time (unit: ms) when the project executes to this Step.

Usage Scenario

This Step is used when a specific **Wait Time** is needed, usually after the Step Set DO or after the **No result** exit port of the Step Check Look.

Parameter Description

General Parameters of Non-Move Steps

▼ *Skip Execution*

None
Default setting. Do not skip the current Step.

Simulation only
Skip the current Step during simulation. The exit port is specified by **Out Port When Skip**.

Always
Skip the current Step when the project is simulating or running. The exit port is specified by **Out Port When Skip**.

Instruction
When **Simulation only** or **Always** is selected, the current Step will be skipped and the subsequent Step will be executed when running the project. If this parameter is set to None in “Check DI” Step in the project, and there is no external input signal during simulation, the project will be stopped when executing to this Step. In this case, setting the parameter to Simulation only or Always enables the simulation to continue.

▼ *Out Port When Skip*

This parameter will take effect when **Skip Execution** is set to **Simulation only** or **Always**. It specifies the exit port when skipping a Step.

Wait Time

Set the wait time according to your need. Default value: 1000 ms

5.5.8. Path

5.5.8.1. Path Procedure

Function

This Step guides the robot to move along dense points that are arranged in multiple continuous segments according to the template file of trajectory point. It need to be used with the Step Vision Move.

Parameter Description

None.

Traj Point Json Path

Pre-edited point trajectory file in JSON format. When the project is running, the number of points sent by the vision service must be consistent with the number of points in the template.

5.5.9. Vision

5.5.9.1. Check Look

Function

This Step is used to check the vision result and provide exit ports for different vision results.

There are 5 exit ports for this Step.

Exit Port	Description
Has results	Take this exit port if the “Vision Look” Step provides a result.
No result	Take this exit port if the “Vision Look” Step does not provide a result due to an error.
Timeout	If the vision result processing does not end after the set **Wait Time Out**, the project will take this exit port. This exit port can be connected with the entry port of the **Check Look** Step to run the Step again.
Not called	“Not called” indicates that the vision project specified by the “Vision Service Name” parameter has not been executed. This port is typically taken for projects with multiple “Vision Look” Steps. The project will take this exit port if there is not a “Vision Look” Step that provides vision service for the current Check Look Step in the previous workflow.
No CloudIn Roi	It usually indicates that all the workobjects to be picked have been picked successfully. This exit port is prioritized over “Has results” and “No results” exit ports.

The relevant example project to the “Check Look” Step is provided to help you learn more about the process of checking the vision results. For the example project, you can go to File > New > Example projects > Others > Check Vision Result in the software.

Parameter Description

General Parameters of Non-Move Steps

▼ *Skip Execution*

None
Default setting. Do not skip the current Step.

Simulation only
Skip the current Step during simulation. The exit port is specified by **Out Port When Skip**.

Always
Skip the current Step when the project is simulating or running. The exit port is specified by **Out Port When Skip**.

Instruction	When **Simulation only** or **Always** is selected, the current Step will be skipped and the subsequent Step will be executed when running the project. If this parameter is set to None in “Check DI” Step in the project, and there is no external input signal during simulation, the project will be stopped when executing to this Step. In this case, setting the parameter to Simulation only or Always enables the simulation to continue.

▼ *Out Port When Skip*

This parameter will take effect when **Skip Execution** is set to **Simulation only** or **Always**. It specifies the exit port when skipping a Step.

Check Custom Key

Unselected by default, and the custom key will not be checked. Select to check the custom key, and then you can customize the **Key to Check** to communicate with the Adapter.

Pre Plan Out Port

To make the path planning process in the actual application continuous, the software will execute a pre-planning beforehand on the backend. Specifying a pre-plan exit port along which the project is most likely to proceed can reduce the probability of re-planning and the wait time of the robot.

Parameter value	Description
-1	Do not specify an exit port for pre-planning.
0	The project will take the **Has results** exit port during pre-planning.

Wait Time Out

This parameter specifies the maximum time to wait for the vision service to respond (Unit: s; default value: 5s). This parameter is used in conjunction with the **Timeout** exit port. Please set the value according to the actual requirement.

Vision Service Name

Select the name of the Mech-Vision project that provides the vision service.

This parameter value should be the same as the “Vision Service Name” selected in the “Vision Look” Step.

5.5.9.2. Update Pallet Pose

Function

Update the pallet pose via the vision service.

Usage Scenario

After a palletizing task is completed, a new pallet will be in place, and the pallet pose should be updated. This Step updates the pallet pose in the specified palletizing Steps via the vision service.

Parameter Description

General Parameters of Non-Move Steps

▼ *Skip Execution*

None	Default setting. Do not skip the current Step.
Simulation only	Skip the current Step during simulation. The exit port is specified by **Out Port When Skip**.
Always	Skip the current Step when the project is simulating or running. The exit port is specified by **Out Port When Skip**.
Instruction	When **Simulation only** or **Always** is selected, the current Step will be skipped and the subsequent Step will be executed when running the project. If this parameter is set to None in “Check DI” Step in the project, and there is no external input signal during simulation, the project will be stopped when executing to this Step. In this case, setting the parameter to Simulation only or Always enables the simulation to continue.

▼ *Out Port When Skip*

This parameter will take effect when **Skip Execution** is set to **Simulation only** or **Always**. It specifies the exit port when skipping a Step.

Save Pallet Pose

Unselected by default, i.e., do not save the updated pallet pose. Select to save the updated pallet pose. You can click 📂 next to the **File Path to Save Pallet Pose** to specify a path to save the pallet pose file.

Load Pallet Pose

Unselected by default, i.e., do not load the pallet pose. Select to load the pallet pose. Click 📂 next to the **File Path to Load Pallet Pose** to select the pallet pose file to load.

Select Palletizing Step

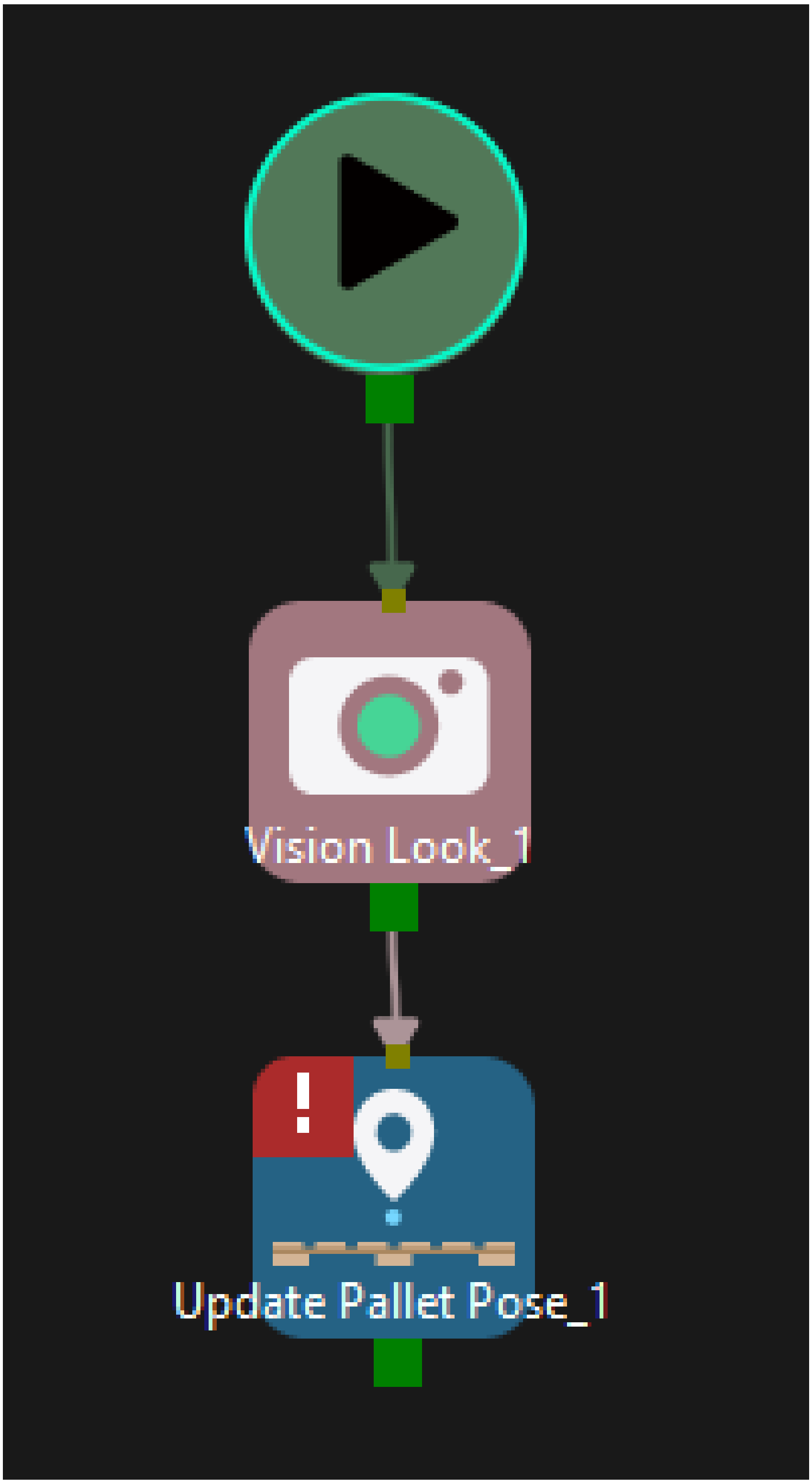
Select the palletizing Step in which the pallet pose should be updated.

Vision Name

Select the registered vision service from which the new pallet pose will be obtained.

Usage Example

After the robot completes a palletizing task, a new pallet will be used, and then the pallet pose should be updated. This Step should be used in conjunction with Vision Look.



5.5.9.3. Update Held Workobject

Function

Update the pose or size of the held workobject via the vision service to avoid collisions.

> The pose of the held workobject refers to the pose of the object from the time the robot picks it to the time it is put down.

Usage Scenario

This Step is usually used when the obtained object size or pose is inaccurate.

To avoid collisions, an accurate size and pose of the held workobject should be re-obtained, and you will need to re-select the collision model for the collision detection.

- The collision model of the held workobject supports cuboids and cylinders that are picked vertically or horizontally.
- This Step may interrupt the path planning of move-type Steps.

Description

General Parameters of Non-Move Steps

▼ *Skip Execution*

None	Default setting. Do not skip the current Step.
Simulation only	Skip the current Step during simulation. The exit port is specified by **Out Port When Skip**.
Always	Skip the current Step when the project is simulating or running. The exit port is specified by **Out Port When Skip**.
Instruction	When **Simulation only** or **Always** is selected, the current Step will be skipped and the subsequent Step will be executed when running the project. If this parameter is set to None in “Check DI” Step in the project, and there is no external input signal during simulation, the project will be stopped when executing to this Step. In this case, setting the parameter to Simulation only or Always enables the simulation to continue.

▼ *Out Port When Skip*

This parameter will take effect when **Skip Execution** is set to **Simulation only** or **Always**. It specifies the exit port when skipping a Step.

Update Approach

There are two Update Info Sources, i.e., **Property Value** and **Vision Service**.

Property Value

Please use this approach when there is no available size of the held workobject or the robot is not guided by the vision service. You can set a model of the object and the model will be attached to the end of the tool for collision detection.

Object Type	Cuboid, Vertical Cylinder, Horizontal Cylinder, and Disc.
Pose Needs Updating	Unselected by default, i.e., do not update the pose of the held workobject. Select to update the pose in TCP, and you can adjust the updated pose in **Pose Update**.

Size Needs Updating

Dimensions of the picked object. Unselected by default, i.e., do not update the dimensions of the held workobject. Select to update the dimensions of the held workobject, and you can adjust the updated dimensions in **Size Update**.

Disc Configuration File Path

Click 📂 to select the brake disc configuration file in **JSON** format. You can only adjust the size of the brake disc.

Vision Service

When the robot moves the object to a proper position, the camera can re-capture an image and the software can re-calculate the dimensions and pose in TCP of the held workobject.

Update Content

Pose Needs Updating

Selected by default, i.e., update the object pose in TCP to the project. Unselect to stop updating the pose of the held workobject.

Size Needs Updating

Selected by default, i.e., update the object dimensions to the project. Unselect to stop updating the dimensions of the held workobject.

Motion Pre-Plan

Pre Plan Out Port

To make the path planning process in the actual application continuous, the software will execute a pre-planning beforehand on the backend. Specifying a pre-plan exit port along which the project is most likely to proceed can reduce the probability of re-planning and the wait time of the robot.

- 0: Take the **No need to update** exit port when the deviation from the original vision result is below the **Updating Threshold**.
- 1: Take the **Update** exit port when the deviation from the original vision result is above the **Updating Threshold**.
- 2: Take the **Exception** exit port when an error of the vision service occurs.

Updating Threshold

Needs Accuracy Check

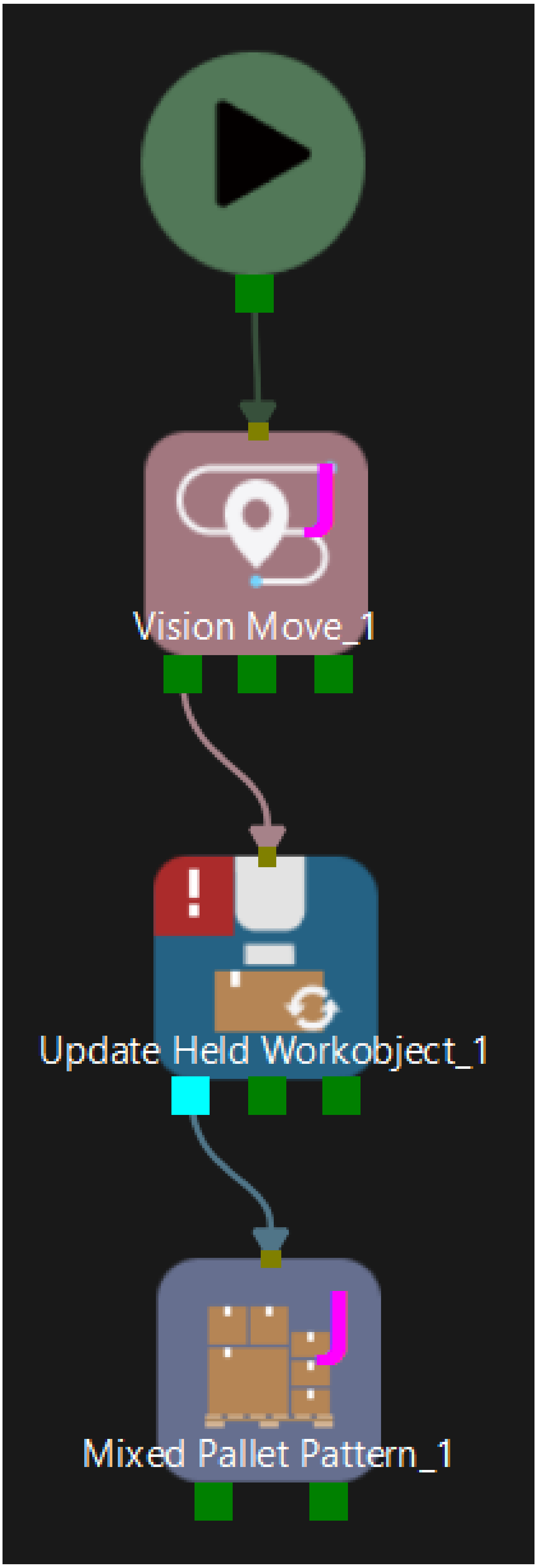
Select by default. You can adjust the **Size Threshold**, **Translation Threshold** and **Rotation Threshold** to specify the Updating Threshold. Unselect this parameter and no updating threshold can be set.

Usage Example

You can use this Step in conjunction with **Classify** and Mixed Pallet Pattern.

To use in conjunction with **Classify**, please select the **Get Label from "Update Held Workobject"** in Step parameters panel.

To use in conjunction with **Mixed Pallet Pattern**, please select **Capture Twice to Update Carton Size** under Pallet Type › Place By Drop in Step parameters panel of the Mixed Pallet Pattern Step. The fixed usage is shown in the figure below.


Vision Move_1
Update Held Workobject_1
Mixed Pallet Pattern_1

5.5.9.4. Update Scene Objects

Function

Update the pose and dimensions of the specified scene object via the vision service or parameter settings.

Usage Scenario

When running a Mech-Viz project, the position and dimensions of the scene object should be adjusted in order to avoid collisions in different situations.

Parameter Description

General Parameters of Non-Move Steps

▼ *Skip Execution*

None	Default setting. Do not skip the current Step.
Simulation only	Skip the current Step during simulation. The exit port is specified by **Out Port When Skip**.
Always	Skip the current Step when the project is simulating or running. The exit port is specified by **Out Port When Skip**.
Instruction	When **Simulation only** or **Always** is selected, the current Step will be skipped and the subsequent Step will be executed when running the project. If this parameter is set to None in “Check DI” Step in the project, and there is no external input signal during simulation, the project will be stopped when executing to this Step. In this case, setting the parameter to Simulation only or Always enables the simulation to continue.

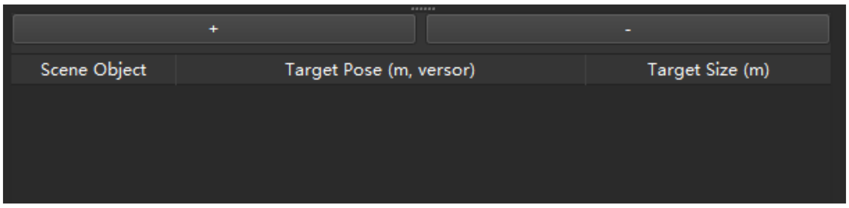
▼ *Out Port When Skip*

This parameter will take effect when **Skip Execution** is set to **Simulation only** or **Always**. It specifies the exit port when skipping a Step.

Update Info Source

This parameter specifies the information source to update the scene object’s pose and dimensions. The two supported information sources are **Config Value** and **Vision Service**.

Config Value



[+]: Add a new scene object.

[-]: Delete the selected scene object.

You can add or delete a scene object according to the actual requirement. The pose and dimensions of the scene object at the top of the Scene Object list will be updated first. For each item in the Scene Object list, you can select the scene object in the corresponding drop-down list and then specify the **Target Pose** and **Target Size**. The format of target pose should be XYZ (m) plus quaternions and the target size is the dimensions in X, Y, and Z-directions in meters.

For example, a target pose can be `1.00, 0.00, 0.00, 1.00, 0.00, 0.00, 0.00`, a target size of a cuboid can be `0.40, 0.40, 1.00` (length, width, height), and a target size of a vertical cylinder can be `0.50, 1.00, 0.00` (radius, height, 0).

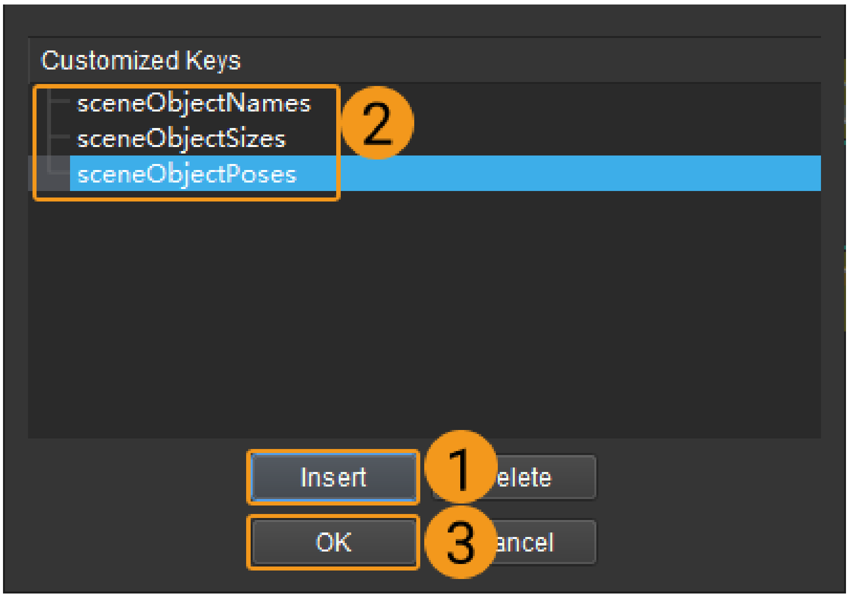
Vision Service

Vision Name

Select the Mech-Vision project that provides the vision service. The Mech-Vision project must be a registered one.

Once you selected a Mech-Vision project, please follow the steps below to configure the Procedure Out Step in the Mech-Vision project.

1. Go to the Step Parameters panel, and set the **Port Type** to **Custom**.
2. Click **[Set data type output to external communication]**, and the Procedure Out Setting Assistant window will pop up.
3. Click **[Insert]** to create 3 customized keys.
4. Double-click the customized keys to rename them to **sceneObjectNames**, **sceneObjectSizes**, and **sceneObjectPoses**.
5. Click **[OK]** to save the changes.
6. Connect the input port of Procedure Out with corresponding output ports respectively.

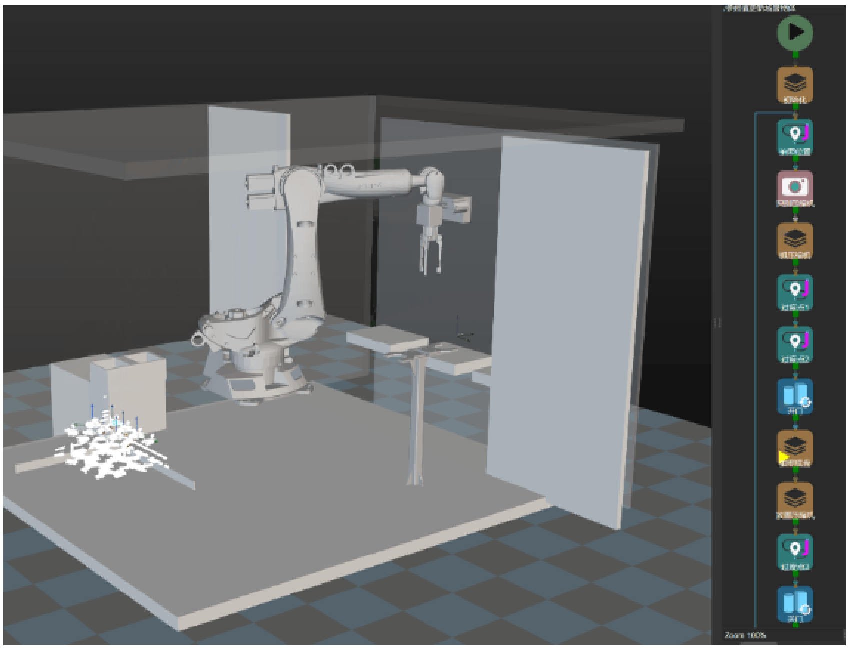


Customized Keys	Description
sceneObjectNames	The name should be consistent with that of the scene object in Mech-Viz.

Customized Keys	Description
sceneObjectSizes	Object dimensions in meters, (length, width, height) for cuboids and (radius, height, 0) for cylinders.
sceneObjectPoses	XYZ (m) plus quaternions.

Example

In a scenario where a sensor door is used, this Step is used to update the model of the door, thus collisions in different processes can be avoided. Config Value is used to update the scene object in this example.



5.5.9.5. Is Vision Result Used Up

Function

Check if the vision result in the corresponding Vision Move Step is used up.

> This Step only takes effect when the **Reuse Vision Result** parameter in Vision Move is selected.

Usage Scenario

When the vision result in Vision Move contains multiple pickable poses, the software will guide the robot with the poses to pick until the pickable poses are used up.

When the poses in the vision result are used up, the camera will capture an image again. For scenarios that have a high requirement on the takt time, reusing vision result can reduce the vision processing time and shorten the takt time.

Parameter Description

General Parameters of Non-Move Steps

▼ *Skip Execution*

None	Default setting. Do not skip the current Step.
Simulation only	Skip the current Step during simulation. The exit port is specified by **Out Port When Skip**.
Always	Skip the current Step when the project is simulating or running. The exit port is specified by **Out Port When Skip**.
Instruction	When **Simulation only** or **Always** is selected, the current Step will be skipped and the subsequent Step will be executed when running the project. If this parameter is set to None in “Check DI” Step in the project, and there is no external input signal during simulation, the project will be stopped when executing to this Step. In this case, setting the parameter to Simulation only or Always enables the simulation to continue.

▼ *Out Port When Skip*

This parameter will take effect when **Skip Execution** is set to **Simulation only** or **Always**. It specifies the exit port when skipping a Step.

Pre Plan Out Port

To make the path planning process in the actual application continuous, the software will execute a pre-planning beforehand on the backend. Specifying a pre-plan exit port along which the project is most likely to proceed can reduce the probability of re-planning and the wait time of the robot.

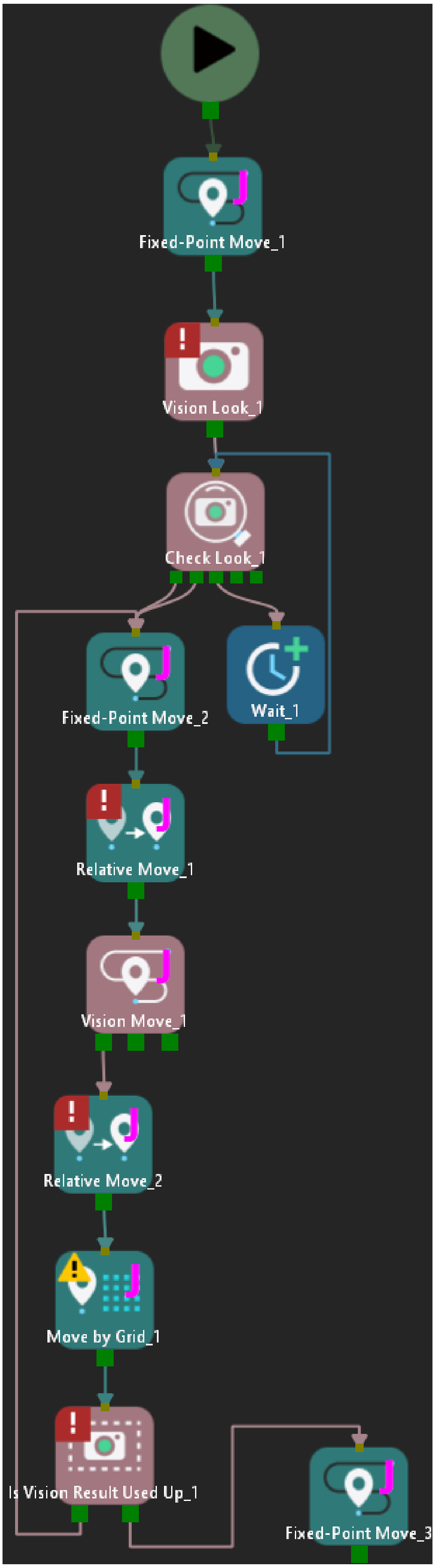
-1	Do not specify an exit port for pre-planning.
0	The project will take the **Not used up** exit port during pre-planning.
1	The project will take the **Used up** exit port during pre-planning.

Select “Vision Move”

Select the Vision Move Step in which the vision result should be checked.

Application Example

In a depalletizing task, the camera usually captures one image for one layer, and the corresponding vision result will be used to pick all the cartons on the same layer. **Is Vision Result Used Up** can be used to determine whether all the cartons on the same layer are picked. The example is shown below.


Fixed-Point Move_1
Vision Look_1
Check Look_1
Fixed-Point Move_2
Wait_1
Relative Move_1
Vision Move_1
Relative Move_2
Move by Grid_1
Is Vision Result Used Up_1
Fixed-Point Move_3

- If the vision result is **not used up**, the robot will pick repeatedly.
- If the vision result is **used up**, the **Used up** exit port will be taken.

5.5.9.6. Vision Look

Function

This Step starts the Mech-Vision project and inputs the vision result to Mech-Viz.

Usage Scenario

This Step is applicable to all projects, and it is usually used with Check Look, Relative Move and Vision Move.

Parameter Description

General Parameters of Non-Move Steps

▼ *Skip Execution*

None	Default setting. Do not skip the current Step.
Simulation only	Skip the current Step during simulation. The exit port is specified by **Out Port When Skip**.
Always	Skip the current Step when the project is simulating or running. The exit port is specified by **Out Port When Skip**.
Instruction	When **Simulation only** or **Always** is selected, the current Step will be skipped and the subsequent Step will be executed when running the project. If this parameter is set to None in “Check DI” Step in the project, and there is no external input signal during simulation, the project will be stopped when executing to this Step. In this case, setting the parameter to Simulation only or Always enables the simulation to continue.

▼ *Out Port When Skip*

This parameter will take effect when **Skip Execution** is set to **Simulation only** or **Always**. It specifies the exit port when skipping a Step.

Wait Move Precisely

Unselected by default, i.e., the robot does not necessarily need to move to a specified position before the project executes the current Step.

Allow Vision Result Unused

Unselected by default, i.e., use the obtained vision result. Once this parameter is selected, the obtained vision result cannot be used.

Vision Name

Select the Mech-Vision project that provides the vision service.

5.5.9.7. Vision Move

Function

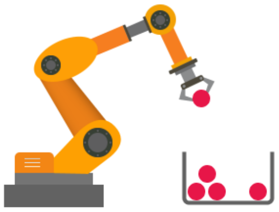
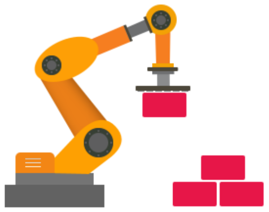
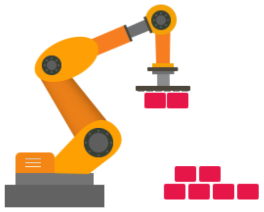
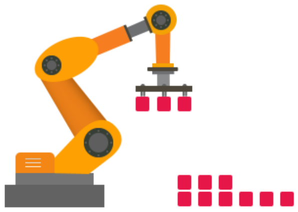
This Step plans the robot’s path and pose for picking the target workobject based on the vision result returned by the vision service.

Exit Port	Description
Success	The path was planned successfully.
Plan failure	The path planning failed.
Other failures	There are no available poses for planning. Possible causes: Mech-Vision did not output poses. The poses output by Mech-Vision cannot meet the requirement.

Usage Scenario

According to the actual usage scenarios and tool types, the **picking methods** are divided into **Regular picking**, **Box depalletizing** and **Picking with array**.

Please select the picking method according to the actual situation. In addition, the parameters for each picking method are different. See the table below for more information.

Picking Method		Description	Illustration
Regular Picking		The objects to be picked are industrial parts, delivery packages, etc. The end tool in use can be a single-end gripper, small suction cup, etc.	
Box depalletizing	Box Depalletizing—Multiple at Once	The objects to be picked are cartons, turnover boxes, or sacks, and only one object will be picked at once. The end tool in use is a depalletizing vacuum gripper.	
	Box Depalletizing—One by One	The objects to be picked are cartons, turnover boxes, or sacks, and multiple object can be picked at once. The end tool in use is a depalletizing vacuum gripper.	
Picking with array		The robot tool has multiple ends and each end picks one workobject specifically. At present, this method only supports multiple ends that are arranged in a single row.	

5.5.9.7.1. Regular Picking

When the **Picking Method** is set to **Regular picking**, you can configure the following parameters.

In addition, you should refer to Configure the Tool to configure the corresponding end tool.

Move-Type Step Common Parameters

▼ *Send Waypoint*

Selected by default to send waypoint poses to the receiver, such as the robot. When this option is unselected, the waypoint pose will not be sent. However, the waypoint will remain in the planned path.

▼ *Try Continuously Running through Succeeding Non-Moves*

Unselected by default. When **non-move Steps**, such as Vision Look, Set DO, Check DI, etc., are connected between **move-type Steps**, the robot’s path planning will be interrupted, and the actual robot will take a short pause, reducing the smoothness of running.

When this option is selected, the project will continue to run without waiting for the current move-type Step to complete execution, and therefore the robot can move in a smooth way without pauses. However, enabling this option may cause the execution of the Step to end prematurely.

Why will this option cause the execution of the Step to end prematurely?

Mech-Viz will send multiple poses simultaneously to the robot when the project is running. When the currently returned JPs of the robot correspond to the last pose sent by Mech-Viz, Mech-Viz will assume that the robot has moved to the last position.

For example, there are 10 move-type Steps in a path, and the pose of the 5th move-type Step is the same as that of the last move-type Step. When the robot moves at low speed, it sends JPs to Mech-Viz when it moves to the 5th move-type Step, Mech-Viz may mistakenly determine that the robot has finished the move-type Steps and prematurely ends the Steps since the poses of the 5th move-type Step and the last move-type Step are the same in the path.

▼ *Do Not Check Collision with Placed Workobject*

Unselected by default, namely that the collision with the already placed objects will not be detected. When this option is selected, the collisions between the robot, end tool, and placed objects will be detected.

In palletizing scenarios, the two possible cases of error are as follows:

1. When the robot is placing a carton, the robot may come into light contact with the placed cartons while no deformation will be caused. After Mech-Viz detects this collision in simulation, it will plan other positions for placing the carton, and therefore a full stack cannot be formed.
2. Usually, the TCP of a suction cup is inside the suction cup model instead of on the surface of it. Under this circumstance, the suction cup may be embedded in the model of the picked carton in the simulation of picking, while the software does not detect the collision between the end tool and the picked object. After the robot places the object and the carton model turns into an object model in the scene, a collision between the suction cup and the carton will be detected and the palletizing cannot be completed.

When this option is selected, no collision between the robot, end tool, and the placed object will be detected, and the above two cases of errors can be avoided.

▼ *Point Cloud Collision Detection Mode*

Select the proper mode according to the requirement of the on-site situation. Usually, the default setting **Auto** can be used. **Do not check collision** mode can be used in move-type Steps before the robot picks the object, and **Check collision** mode can be used after the robot picks the object.

Auto	Default setting. Collision with point cloud is checked only for the "Vision Move" Step and the "Relative Move" Step that depends on the "Vision Move" Step, but not for all move-type Steps.
Do not check collision	Point cloud collisions for all move-type Steps will not be detected.
Check collision	Point cloud collisions for all move-type Steps will be detected.

When Collisions › Collision detection configuration › Detect collision between point cloud and others is switched on, Mech-Viz will detect collisions between the robot model, end tool model, and point cloud when planning the path. Normally, detects whether the robot collides with other objects during picking and placing. When there are point cloud outliers, non-exiting collisions will be detected, which leads to errors in path planning.

▼ *Ignore Workobject Symmetry*

This parameter will only take effect when **Waypoint type** of the Step is set to **Workobject pose**.

None	Default setting, i.e., do not disable symmetry on any axis.
Around workobject frame Z axis	Only disable symmetry on Z-axis of the workobject reference frame.
Around workobject frame X&Y axis	Disable symmetry on X-axis and Y-axis of the workobject reference frame.
Around all axes	Once the object symmetry is disabled, the robot will place the objects strictly according to the workobject poses.

In some special cases, objects are not pickable due to their peculiar poses. Setting **Rotational symmetry** under Workobjects › Workobject Configuration in Resources may solve this problem. Based on the settings of Rotational symmetry, when the default object pose is not pickable, Mech-Viz will calculate candidate poses and determine whether the candidate poses are pickable. As the candidate poses calculated based on the settings of Rotational symmetry are different from the original poses output from Mech-Vision, the consistency of the objects' place poses cannot be guaranteed.

▼ *Plan Failure Out Port*

After selecting this parameter, this Step will add "Plan failure" out port.

During the planning process, planning is carried out along the branch after the "Success" out port; if the planning fails in the current Step, the branch process after the "Planning failure" out port will be executed.

Held Workobject Collision Detection Settings

▼ *Do Not Check Collision with Scene Object/Robot*

Unselected by default. Once this option is selected, the collisions between the **held workobject** with the **scene objects or robot** will not be detected, and therefore the calculation workload of collision detection will be reduced, the planning speed can be increased, and the cycle time can be shortened. It is usually enabled in the first one or two move-type Steps after the robot picks the object.

Please enable this option cautiously as there may be collision risks.

When **Detect collision between held workobject and others** under Collision detection configuration › Configuration on held workobject is enabled, the software will detect whether the model of the held object collides with the models of the scene objects and the robot.

In palletizing projects, the calculated carton dimensions have millimeter-level errors with the actual dimensions, and frictions between cartons may occur during picking but no collisions will occur. For some move-type Steps that will obviously not cause collisions, detecting such collisions only adds to the calculation workload and planning time, and consequently extending the cycle time. In palletizing projects, enabling Do Not Check Collision with Scene Objects does not affect the collision detection between the held carton and the placed cartons. This option can be enabled when there are scene objects under the stack to avoid failure of finding the palletizing solution.

▼ *Do Not Check Collision with Point Cloud*

Unselected by default. Once this option is selected, the collisions between the **held workobject** with the **point clouds in the scene** will not be detected, and therefore the calculation workload of collision detection will be reduced, the planning speed can be increased, and the cycle time can be shortened.

- When both **Detect collision between held workobject and others** under Collision detection configuration › Configuration on held workobject and **Configuration on point cloud ¬ Detect collision between point cloud and others** are enabled, the software will detect whether the model of the held workobject will collide with the point cloud in the scene.
- When Mech-Vision sends the point cloud and object model to Mech-Viz, the point cloud and the object model are fitted together. After the robot picks the object, the model moves along the planned path, and the collision between the model of the held workobject and the point cloud will occur.
- It is known that the model of the held workobject will have false collisions with the point cloud. Detecting such collisions unnecessarily adds to the calculation workload and extends the planning time.

Vision Result Global Usage

▼ *All Vision Poses in One Move*

Usually, Vision Move will only use one pose of the corresponding vision result. However, once this parameter is selected, the robot will move through all waypoints corresponding to the vision poses at one time.

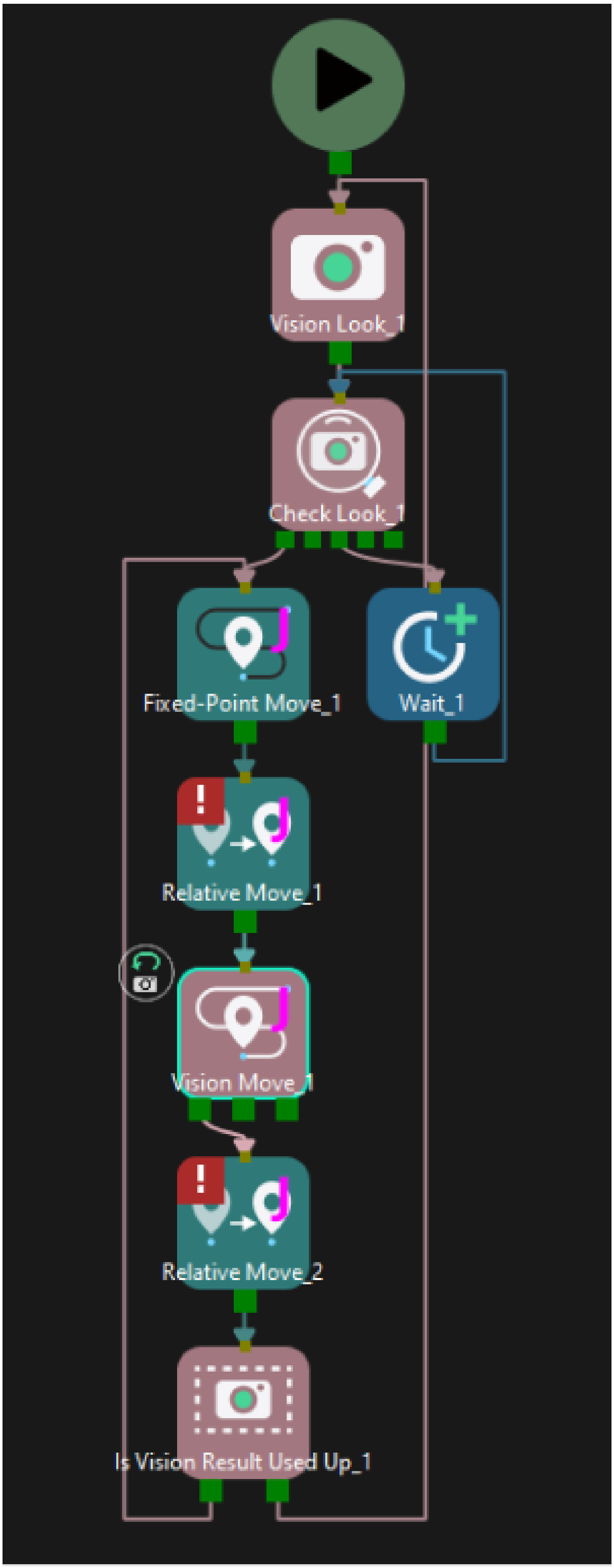
This parameter is usually used when the robot moves in a fixed path where no DO signals will be sent, such as the gluing application.

▼ *Reuse Vision Result*

When this feature is not enabled, each time “Vision Move” plans a robot picking path successfully, the rest unused vision results will be discarded. If this feature is enabled, the vision result that leads to a planning failure and the rest unused vision results will be retained and used for the next planning. This feature should be used with the “Is Vision Result Used Up” Step.

When there are multiple pickable objects in one vision result and **it can be guaranteed that picking one object will not affect the poses of other objects**, this option can be selected. The vision result will be reused and no new image will be captured until all vision results are used up.

Application Example


Vision Look_1
Check Look_1
Fixed-Point Move_1
Wait_1
Relative Move_1
Vision Move_1
Relative Move_2
Is Vision Result Used Up_1

▼ *Share Vision Result*

This feature enables multiple "Vision Move" Steps in which the same vision service is selected to share the vision result.

When one "Vision Move" Step plans successfully, the corresponding vision result will be used, and the unused vision result will be used by the next "Vision Move" Step with this feature enabled. After all "Vision Move" Steps that share the vision result complete planning, the rest results will be discarded.

In addition, this feature can be used in conjunction with the "Reuse Vision Result" feature. When "Reuse Vision Result" is selected, the rest vision results will not be discarded even after all "Vision Move" Steps have completed one round of planning.

Application Example

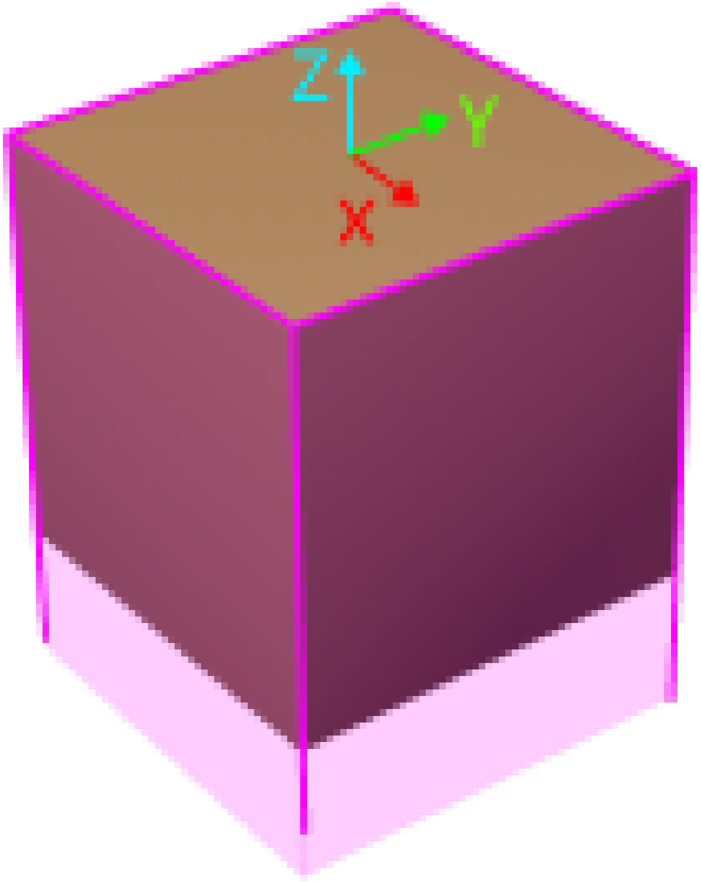
(To be supplemented)

Remove Point Cloud of Target Workobject

▼ *Remove Point Cloud of Target Workobject*

Select to remove the point cloud of the target workobject to be picked, and only collision between the robot end tool and non-target workobjects will be detected.

To avoid picking non-target boxes, the software will remove the point cloud of the target box and keep the point cloud of its adjacent boxes. During path planning, the software will discard a solution that may lead to a wrong picking due to the detected collision between the model of the vacuum gripper's buffer and the point cloud of adjacent boxes.

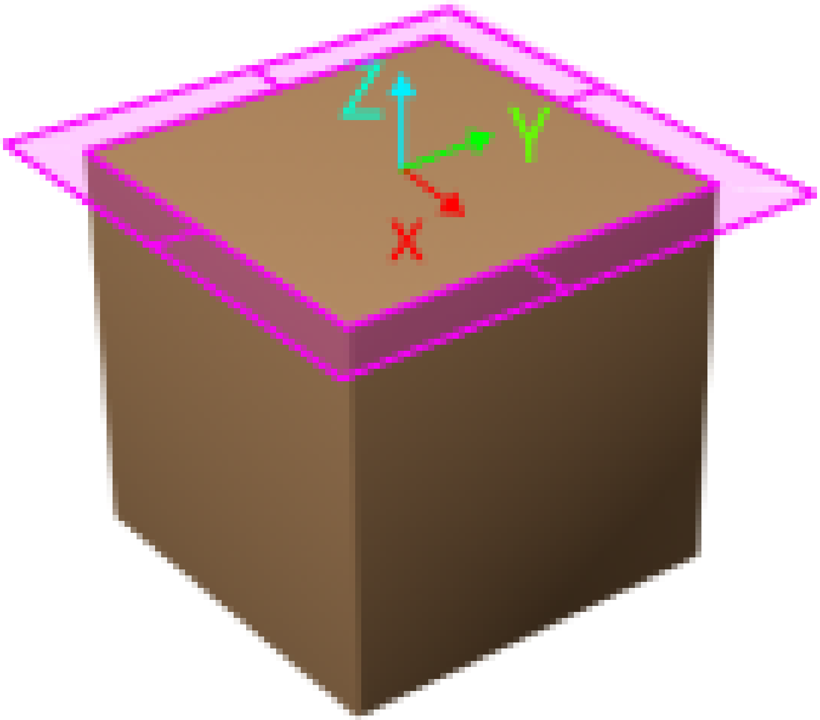


The default range for point cloud removal is a cuboid whose top surface is the top surface of the target box with an infinite height whose direction is the Z-direction of the box's vision pose.

You can adjust the "XY-Plane Point Cloud Removal Range Expansion" and "Z-Direction Point Cloud Removal Range Expansion" to expand the point cloud removal range for removing the point cloud near the target workobject.

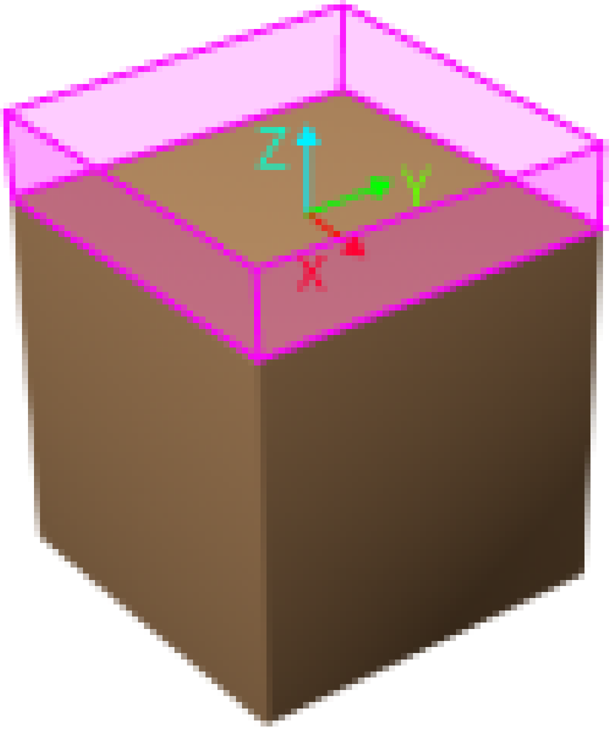
▼ *XY-Plane Point Cloud Removal Range Expansion*

Based on the original workobject dimensions, the XOY plane of the workobject pose is the top surface for the point cloud removal range. This parameter specifies the extended length from actual workobject edges on this surface.



▼ *Z-Direction Point Cloud Removal Range Expansion*

Based on the original workobject dimensions, the extended height in the positive Z-direction of the workobject pose for the point cloud removal range.



Supplementary Features

▼ *Update Bin Pose with Vision*

Once the “Update Bin Pose with Vision” parameter is selected, the bin will be recognized and located simultaneously while the camera captures the image for workobject recognition. Therefore, the bin pose in the simulation area can be updated dynamically, which facilitates the collision detection algorithm to effectively prevent the robot from colliding with the bin.

This feature will provide three fields, “scene_object_names,” “scene_object_sizes,” and “scene_object_poses,” in the vision result to define the name, dimensions, and position of the scene object to be updated.

▼ *Result Must Be in Specified Bin*

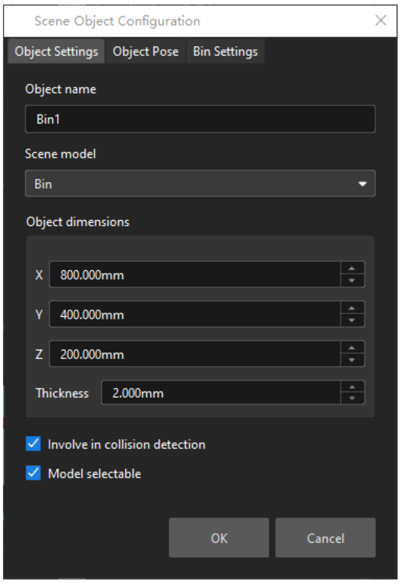
Result Must Be in Specified Bin

This feature limits that the vision result received by the current “Vision Move” Step must be in a specified bin. The vision result out of the effective bin range will not be used.

The effective range for each bin can be configured separately in the corresponding configuration window. Only bins with “Set effective range for pose” enabled can be selected in the drop-down menu.

Application Example

Add “Bin1”, and select the “Set effective range for pose” parameter for this bin.



Object name: Bin1

Scene model: Bin

Object dimensions

X 800.000mm

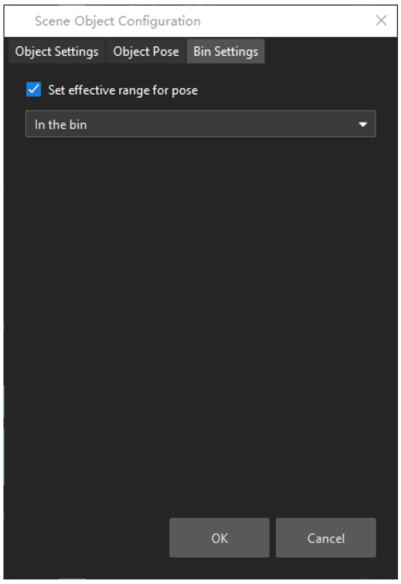
Y 400.000mm

Z 200.000mm

Thickness 2.000mm

☑ Involve in collision detection

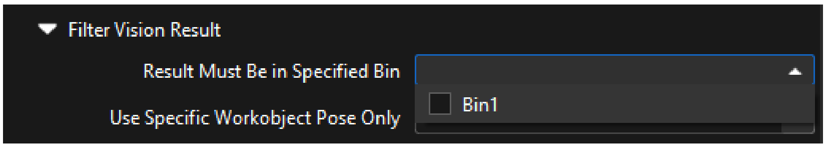
☑ Model selectable



☑ Set effective range for pose

In the bin

Select “Bin1” in the drop-down menu.



Result Must Be in Specified Bin

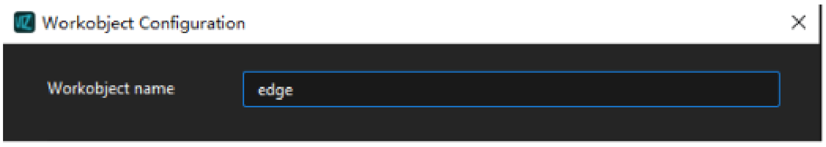
□ Bin1

Use Specific Workobject Pose Only

▼ *Use Specific Workobject Pose Only*

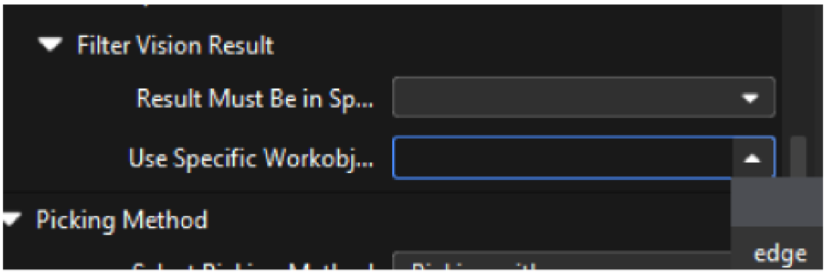
This parameter is used to filter the workobjects corresponding to the specified “labels” field in the vision results.

If you need to use this parameter, you need to add the corresponding workobjects in the resource tree whose name is the same as the "labels" field name, and then select the workobject in the dropdown menu on the right side of the parameter.



Workobject Configuration

Workobject name: edge



Filter Vision Result

Result Must Be in Sp...

Use Specific Workobj...: edge

Picking Method

▼ *Filter Out Unlikely Pick Point*

This parameter group is mainly used to avoid picking on the same pick point in scenarios where the object cannot be picked successfully.

Once the "Filter Out Unlikely Pick Point" parameter is selected, you can adjust the following parameters.

- Filter Subject
  - Pick point: only pick points with high failure rates will be deprioritized in the picking and discarded in some rounds.
  - Workobject: if any one pick point on the workobject is considered to have a high failure rate, all pick points on this workobject will be deprioritized in the picking and discarded in some rounds.
- Priority Downgrade Radius

  A sphere with this parameter value as its radius and the pick point successfully planned in the last round as its center will be introduced. If a pick point in the latest vision result falls within this sphere, the pick point will be put off for picking with a decreased picking priority.

- Discard Attempt Radius

  A sphere with this parameter value as its radius and the pick point successfully planned in the last round as its center will be introduced. If a pick point in the latest vision result falls within this sphere, the pick point will be discarded in this round of planning.

  For example, if the robot only moves the crankshaft but fails to pick it on the first attempt, there is a possibility of successful picking on the next attempt. Therefore, **Priority Downgrade Radius** can be set to downgrade the priority of the pick point, while the pick point will not be discarded. When the robot fails to move the crankshaft at all in the first attempt, it is highly unlikely that the crankshaft can be picked successfully in the next attempt, and therefore **Discard Attempt Radius** can be set to discard the pose directly.

- List Length

  A list of pick points with high failure rates. When the list length exceeds the upper limit, the pick point that is added to the list earliest will be removed from the list and can be used in

the next round of planning.

Count of Picked Workobjects

▼ *Limit Total Count*

Select Limit Total Count and you can set the maximum Planned Count and other parameters.

Vision Service Name

Select the corresponding Mech-Vision project.

5.5.9.7.2. Box Depalletizing—One by One

When the **Picking Method** is set to **Box depalletizing**, and the **Depalletizing Mode** is set to **One by one**, you can configure the following parameters.

In addition, you should refer to Configure the Tool to configure the corresponding end tool.

Move-Type Step Common Parameters

▼ *Send Waypoint*

Selected by default to send waypoint poses to the receiver, such as the robot. When this option is unselected, the waypoint pose will not be sent. However, the waypoint will remain in the planned path.

▼ *Try Continuously Running through Succeeding Non-Moves*

Unselected by default. When **non-move Steps**, such as Vision Look, Set DO, Check DI, etc., are connected between **move-type Steps**, the robot's path planning will be interrupted, and the actual robot will take a short pause, reducing the smoothness of running.

When this option is selected, the project will continue to run without waiting for the current move-type Step to complete execution, and therefore the robot can move in a smooth way without pauses. However, enabling this option may cause the execution of the Step to end prematurely.

Why will this option cause the execution of the Step to end prematurely?

Mech-Viz will send multiple poses simultaneously to the robot when the project is running. When the currently returned JPs of the robot correspond to the last pose sent by Mech-Viz, Mech-Viz will assume that the robot has moved to the last position.

For example, there are 10 move-type Steps in a path, and the pose of the 5th move-type Step is the same as that of the last move-type Step. When the robot moves at low speed, it sends JPs to Mech-Viz when it moves to the 5th move-type Step, Mech-Viz may mistakenly determine that the robot has finished the move-type Steps and prematurely ends the Steps since the poses of the 5th move-type Step and the last move-type Step are the same in the path.

▼ *Do Not Check Collision with Placed Workobject*

Unselected by default, namely that the collision with the already placed objects will not be detected. When this option is selected, the collisions between the robot, end tool, and placed objects will be detected.

In palletizing scenarios, the two possible cases of error are as follows:

1. When the robot is placing a carton, the robot may come into light contact with the placed cartons while no deformation will be caused. After Mech-Viz detects this collision in simulation, it will plan other positions for placing the carton, and therefore a full stack cannot be formed.
2. Usually, the TCP of a suction cup is inside the suction cup model instead of on the surface of it. Under this circumstance, the suction cup may be embedded in the model of the picked carton in the simulation of picking, while the software does not detect the collision between the end tool and the picked object. After the robot places the object and the carton model turns into an object model in the scene, a collision between the suction cup and the carton will be detected and the palletizing cannot be completed.

When this option is selected, no collision between the robot, end tool, and the placed object will be detected, and the above two cases of errors can be avoided.

▼ *Point Cloud Collision Detection Mode*

Select the proper mode according to the requirement of the on-site situation. Usually, the default setting **Auto** can be used. **Do not check collision** mode can be used in move-type Steps before the robot picks the object, and **Check collision** mode can be used after the robot picks the object.

Auto	Default setting. Collision with point cloud is checked only for the “Vision Move” Step and the “Relative Move” Step that depends on the “Vision Move” Step, but not for all move-type Steps.
Do not check collision	Point cloud collisions for all move-type Steps will not be detected.
Check collision	Point cloud collisions for all move-type Steps will be detected.

! When Collisions › Collision detection configuration › Detect collision between point cloud and others is switched on, Mech-Viz will detect collisions between the robot model, end tool model, and point cloud when planning the path. Normally, detects whether the robot collides with other objects during picking and placing. When there are point cloud outliers, non-exiting collisions will be detected, which leads to errors in path planning.

▼ *Ignore Workobject Symmetry*

This parameter will only take effect when **Waypoint type** of the Step is set to **Workobject pose**.

None	Default setting, i.e., do not disable symmetry on any axis.
Around workobject frame Z axis	Only disable symmetry on Z-axis of the workobject reference frame.
Around workobject frame X&Y axis	Disable symmetry on X-axis and Y-axis of the workobject reference frame.
Around all axes	Once the object symmetry is disabled, the robot will place the objects strictly according to the workobject poses.

i In some special cases, objects are not pickable due to their peculiar poses. Setting **Rotational symmetry** under Workobjects › Workobject Configuration in Resources may solve this problem. Based on the settings of Rotational symmetry, when the default object pose is not pickable, Mech-Viz will calculate candidate poses and determine whether the candidate poses are pickable. As the candidate poses calculated based on the settings of Rotational symmetry are different from the original poses output from Mech-Vision, the consistency of the objects’ place poses cannot be guaranteed.

▼ *Plan Failure Out Port*

After selecting this parameter, this Step will add “Plan failure” out port.

During the planning process, planning is carried out along the branch after the “Success” out port; if the planning fails in the current Step, the branch process after the “Planning failure” out port will be executed.

Held Workobject Collision Detection Settings

▼ *Do Not Check Collision with Scene Object/Robot*

Unselected by default. Once this option is selected, the collisions between the **held workobject** with the **scene objects or robot** will not be detected, and therefore the calculation workload of collision detection will be reduced, the planning speed can be increased, and the cycle time can be shortened. It is usually enabled in the first one or two move-type Steps after the robot picks the object.

Please enable this option cautiously as there may be collision risks.

When **Detect collision between held workobject and others** under Collision detection configuration › Configuration on held workobject is enabled, the software will detect whether the model of the held object collides with the models of the scene objects and the robot.

In palletizing projects, the calculated carton dimensions have millimeter-level errors with the actual dimensions, and frictions between cartons may occur during picking but no collisions will occur. For some move-type Steps that will obviously not cause collisions, detecting such collisions only adds to the calculation workload and planning time, and consequently extending the cycle time. In palletizing projects, enabling Do Not Check Collision with Scene Objects does not affect the collision detection between the held carton and the placed cartons. This option can be enabled when there are scene objects under the stack to avoid failure of finding the palletizing solution.

▼ *Do Not Check Collision with Point Cloud*

Unselected by default. Once this option is selected, the collisions between the **held workobject** with the **point clouds in the scene** will not be detected, and therefore the calculation workload of collision detection will be reduced, the planning speed can be increased, and the cycle time can be shortened.

- When both **Detect collision between held workobject and others** under Collision detection configuration › Configuration on held workobject and **Configuration on point cloud ¬ Detect collision between point cloud and others** are enabled, the software will detect whether the model of the held workobject will collide with the point cloud in the scene.
- When Mech-Vision sends the point cloud and object model to Mech-Viz, the point cloud and the object model are fitted together. After the robot picks the object, the model moves along the planned path, and the collision between the model of the held workobject and the point cloud will occur.
- It is known that the model of the held workobject will have false collisions with the point cloud. Detecting such collisions unnecessarily adds to the calculation workload and extends the planning time.

Vision Result Global Usage

▼ *All Vision Poses in One Move*

Usually, Vision Move will only use one pose of the corresponding vision result. However, once this parameter is selected, the robot will move through all waypoints corresponding to the vision poses at one time.

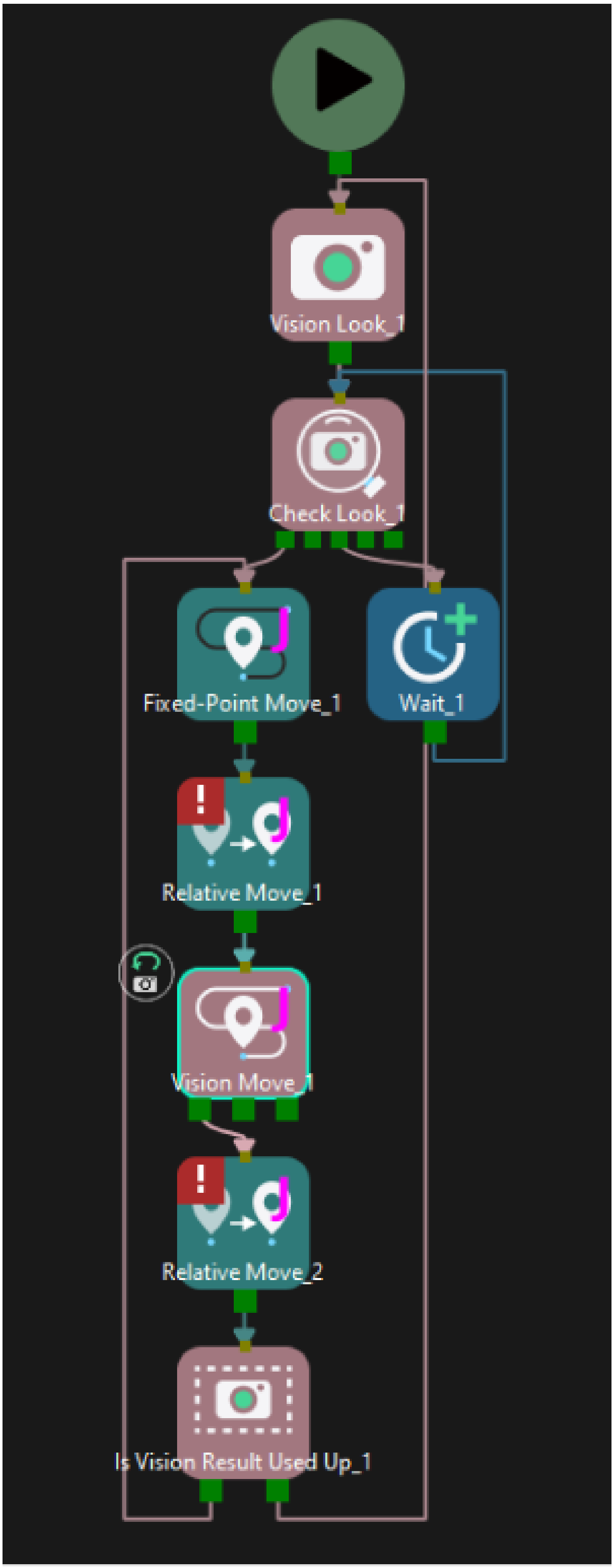
This parameter is usually used when the robot moves in a fixed path where no DO signals will be sent, such as the gluing application.

▼ *Reuse Vision Result*

When this feature is not enabled, each time “Vision Move” plans a robot picking path successfully, the rest unused vision results will be discarded. If this feature is enabled, the vision result that leads to a planning failure and the rest unused vision results will be retained and used for the next planning. This feature should be used with the “Is Vision Result Used Up” Step.

When there are multiple pickable objects in one vision result and **it can be guaranteed that picking one object will not affect the poses of other objects**, this option can be selected. The vision result will be reused and no new image will be captured until all vision results are used up.

Application Example


Vision Look_1
Check Look_1
Fixed-Point Move_1
Wait_1
Relative Move_1
Vision Move_1
Relative Move_2
Is Vision Result Used Up_1

▼ *Share Vision Result*

This feature enables multiple “Vision Move” Steps in which the same vision service is selected to share the vision result.

When one “Vision Move” Step plans successfully, the corresponding vision result will be used, and the unused vision result will be used by the next “Vision Move” Step with this feature enabled. After all “Vision Move” Steps that share the vision result complete planning, the rest results will be discarded.

In addition, this feature can be used in conjunction with the “Reuse Vision Result” feature. When “Reuse Vision Result” is selected, the rest vision results will not be discarded even after all “Vision Move” Steps have completed one round of planning.

Application Example

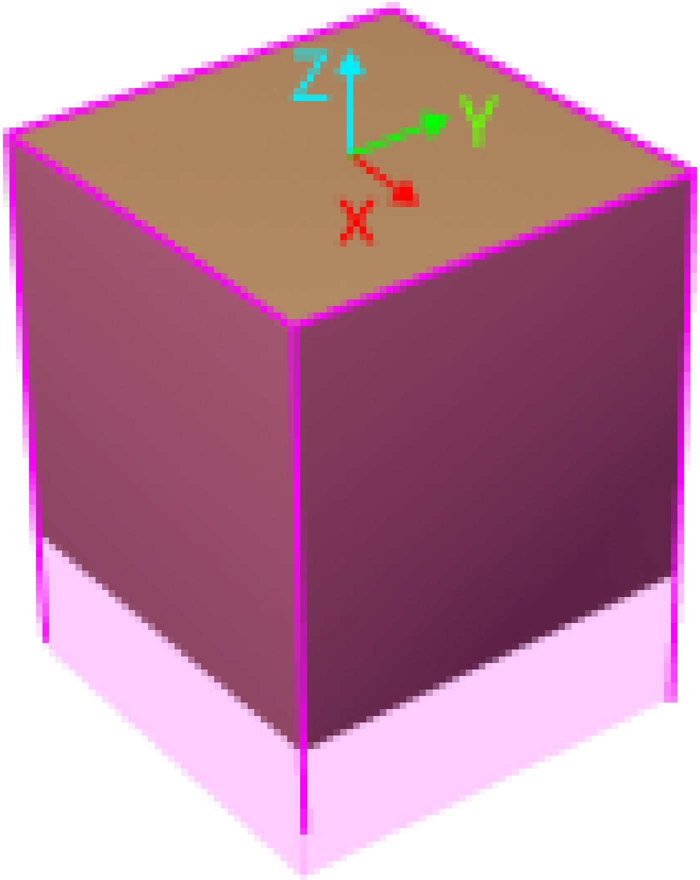
(To be supplemented)

Remove Point Cloud of Target Workobject

▼ *Remove Point Cloud of Target Workobject*

Select to remove the point cloud of the target workobject to be picked, and only collision between the robot end tool and non-target workobjects will be detected.

To avoid picking non-target boxes, the software will remove the point cloud of the target box and keep the point cloud of its adjacent boxes. During path planning, the software will discard a solution that may lead to a wrong picking due to the detected collision between the model of the vacuum gripper’s buffer and the point cloud of adjacent boxes.

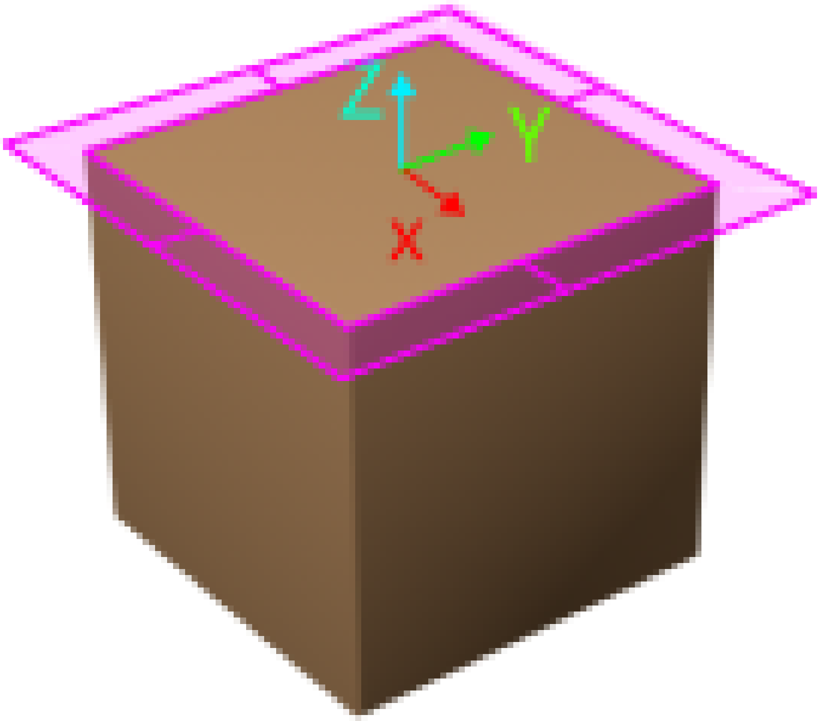


The default range for point cloud removal is a cuboid whose top surface is the top surface of the target box with an infinite height whose direction is the Z-direction of the box's vision pose.

You can adjust the "XY-Plane Point Cloud Removal Range Expansion" and "Z-Direction Point Cloud Removal Range Expansion" to expand the point cloud removal range for removing the point cloud near the target workobject.

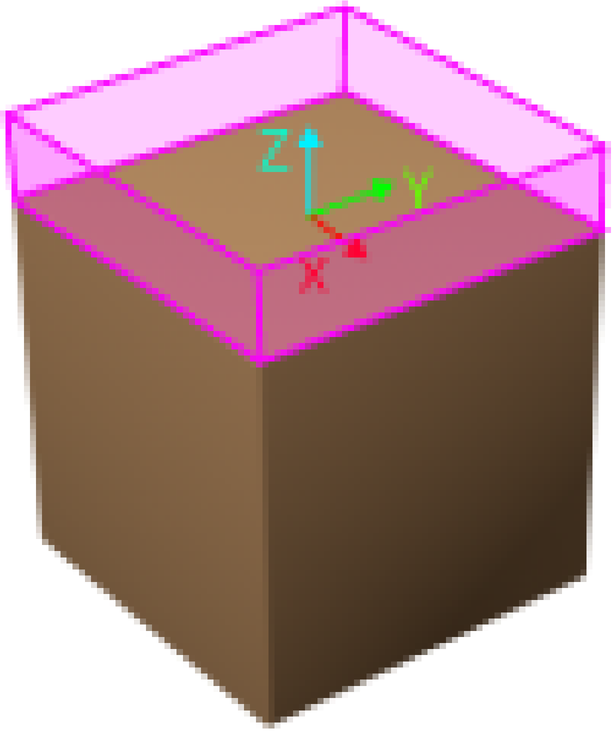
▼ *XY-Plane Point Cloud Removal Range Expansion*

Based on the original workobject dimensions, the XOY plane of the workobject pose is the top surface for the point cloud removal range. This parameter specifies the extended length from actual workobject edges on this surface.



▼ *Z-Direction Point Cloud Removal Range Expansion*

Based on the original workobject dimensions, the extended height in the positive Z-direction of the workobject pose for the point cloud removal range.



Supplementary Features

▼ *Update Bin Pose with Vision*

Once the “Update Bin Pose with Vision” parameter is selected, the bin will be recognized and located simultaneously while the camera captures the image for workobject recognition. Therefore, the bin pose in the simulation area can be updated dynamically, which facilitates the collision detection algorithm to effectively prevent the robot from colliding with the bin.

This feature will provide three fields, “scene_object_names,” “scene_object_sizes,” and “scene_object_poses,” in the vision result to define the name, dimensions, and position of the scene object to be updated.

▼ *Result Must Be in Specified Bin*

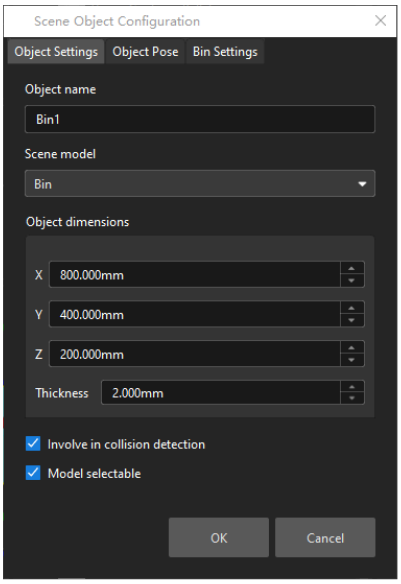
Result Must Be in Specified Bin

This feature limits that the vision result received by the current “Vision Move” Step must be in a specified bin. The vision result out of the effective bin range will not be used.

The effective range for each bin can be configured separately in the corresponding configuration window. Only bins with “Set effective range for pose” enabled can be selected in the drop-down menu.

Application Example

Add “Bin1”, and select the “Set effective range for pose” parameter for this bin.



Scene Object Configuration

Object Settings | Object Pose | Bin Settings

Object name: Bin1

Scene model: Bin

Object dimensions

X 800.000mm

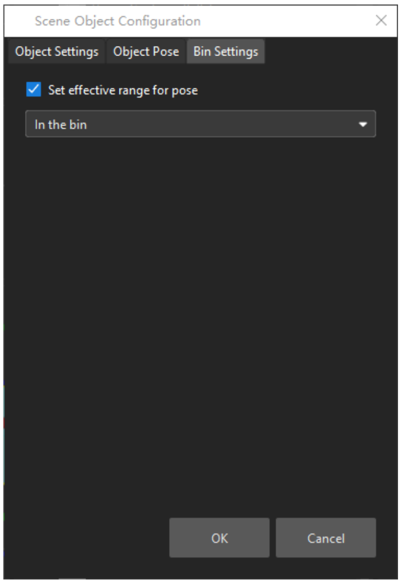
Y 400.000mm

Z 200.000mm

Thickness 2.000mm

☑ Involve in collision detection

☑ Model selectable



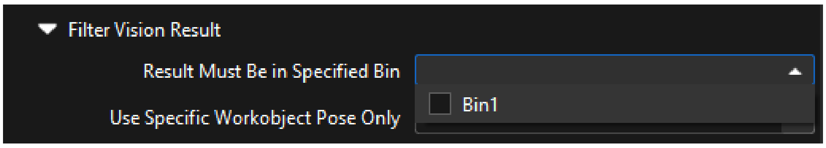
Scene Object Configuration

Object Settings | Object Pose | Bin Settings

☑ Set effective range for pose

In the bin

Select “Bin1” in the drop-down menu.



▼ Filter Vision Result

Result Must Be in Specified Bin

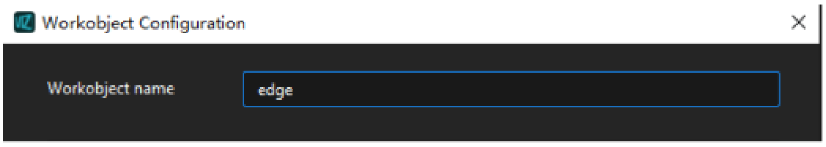
☐ Bin1

Use Specific Workobject Pose Only

▼ *Use Specific Workobject Pose Only*

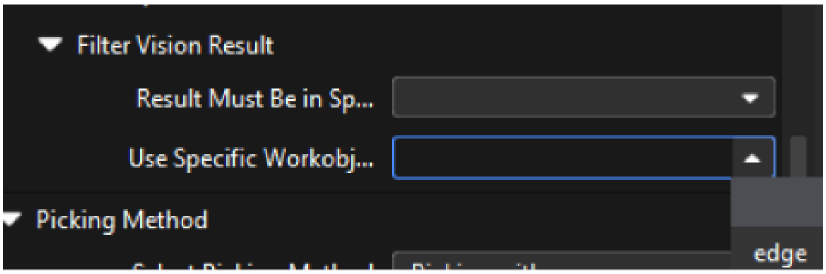
This parameter is used to filter the workobjects corresponding to the specified “labels” field in the vision results.

If you need to use this parameter, you need to add the corresponding workobjects in the resource tree whose name is the same as the “labels” field name, and then select the workobject in the dropdown menu on the right side of the parameter.



Workobject Configuration

Workobject name: edge



Filter Vision Result

Result Must Be in Sp...

Use Specific Workobj...: edge

Picking Method

▼ *Filter Out Unlikely Pick Point*

This parameter group is mainly used to avoid picking on the same pick point in scenarios where the object cannot be picked successfully.

Once the “Filter Out Unlikely Pick Point” parameter is selected, you can adjust the following parameters.

- Filter Subject
  - Pick point: only pick points with high failure rates will be deprioritized in the picking and discarded in some rounds.
  - Workobject: if any one pick point on the workobject is considered to have a high failure rate, all pick points on this workobject will be deprioritized in the picking and discarded in some rounds.
- Priority Downgrade Radius

  A sphere with this parameter value as its radius and the pick point successfully planned in the last round as its center will be introduced. If a pick point in the latest vision result falls within this sphere, the pick point will be put off for picking with a decreased picking priority.
- Discard Attempt Radius

  A sphere with this parameter value as its radius and the pick point successfully planned in the last round as its center will be introduced. If a pick point in the latest vision result falls within this sphere, the pick point will be discarded in this round of planning.

  For example, if the robot only moves the crankshaft but fails to pick it on the first attempt, there is a possibility of successful picking on the next attempt. Therefore, **Priority Downgrade Radius** can be set to downgrade the priority of the pick point, while the pick point will not be discarded. When the robot fails to move the crankshaft at all in the first attempt, it is highly unlikely that the crankshaft can be picked successfully in the next attempt, and therefore **Discard Attempt Radius** can be set to discard the pose directly.
- List Length

  A list of pick points with high failure rates. When the list length exceeds the upper limit, the pick point that is added to the list earliest will be removed from the list and can be used in

the next round of planning.

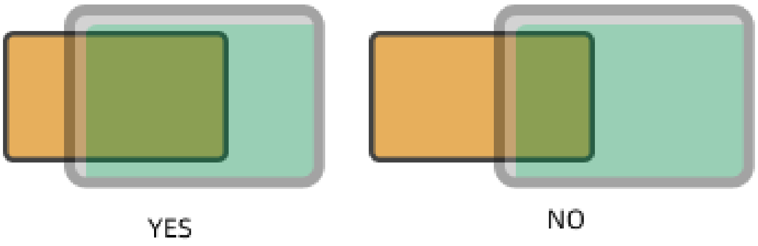
Suction Cup Offset

▼ *Box Surface Coverage Lower Limit*

This parameter specifies the minimum proportion of a box’s upper surface covered by the vacuum gripper when the vacuum gripper attempts to pick a box. If the coverage is below the threshold, it is considered that the vacuum gripper cannot attach to the box firmly and the picking is not allowed.

In addition, when the upper surface of a single box is larger than the entire vacuum gripper, and all sections of the vacuum gripper are activated, the coverage can reach 100%.

Supposing that this parameter is set to 50%, the picking feasibilities are shown as below.

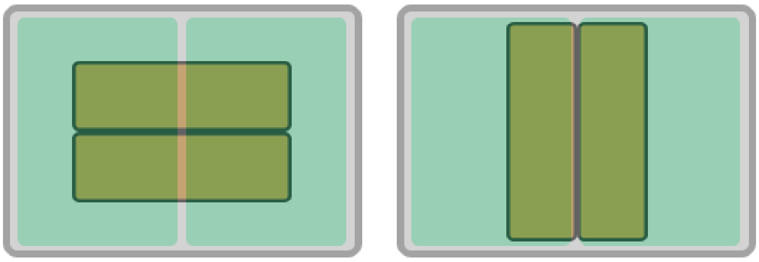


During the depalletizing process of the boxes, the fixed and non-replaceable suction cups are usually used. Due to various sizes of the boxes, the suction cups need to handle different boxes of various sizes. To avoid picking non-target boxes, the robot may not pick the target box with the center of the vacuum gripper. Instead, the corner or other parts of the vacuum gripper will be used. By setting the orientation of the suction cups and configuring the picking strategy, the problem of mis-suction can be effectively avoided, ensuring stable picking of boxes of various sizes.

▼ *Suction Cup Orientation*

This parameter specifies the orientation of the vacuum gripper relative to the box group during picking.

- Random orientation: The orientation of the suction cup is not specified.
- Custom: You can set the **Suction Cup X-axis Orientation** and **Orientation Reference**, and there are four relative positional relationships in total:
  - X-axes of objects are parallel to that of the vacuum gripper.
  - X-axes of objects are perpendicular to that of the vacuum gripper.
  - The X-axis of the vacuum gripper is parallel to the direction along which the boxes are grouped.
  - The X-axis of the vacuum gripper is perpendicular to the direction along which the boxes are grouped.



The figure on the left shows the effect when the X-axis of the vacuum gripper is parallel to the long edge of the box group, and the figure on the right shows the effect when the X-axis of the vacuum gripper is perpendicular to the long edge of the box group.

There are four offset strategies in total, and each strategy contains different parameters.

▼ *Default strategy*

The software sequentially applies three strategies—Center to center, Edge midpoint to edge midpoint, and Corner to corner—to attempt the vacuum gripper offset.

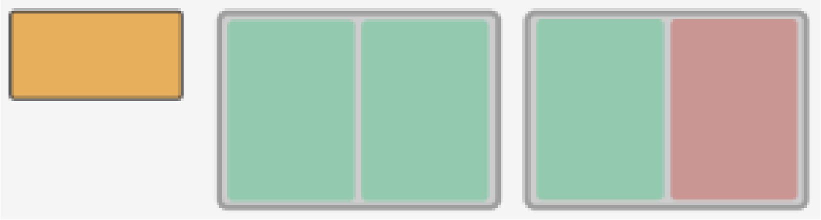
▼ *Custom strategy priority*

The software adheres to user-defined offset priorities to attempt the vacuum gripper offset.

There are three custom strategies in total, which are Center to center, Edge midpoint to edge midpoint, and Corner to corner. You can set the priority levels according to actual requirements.

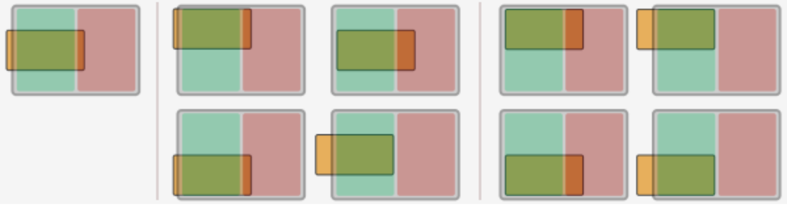
- Center to center: The center of the activated section of the vacuum gripper matches that of the box.
- Edge midpoint to edge midpoint: The edge midpoint of the activated section of the vacuum gripper matches that of the box.
- Corner to corner: The corner of the activated section of the vacuum gripper matches that of the box.

In the figure below, the orange rectangle represents the box, and the gray rectangle represents the vacuum gripper. The green part of the vacuum gripper indicates that the section is activated, while the red part indicates that the section is deactivated.



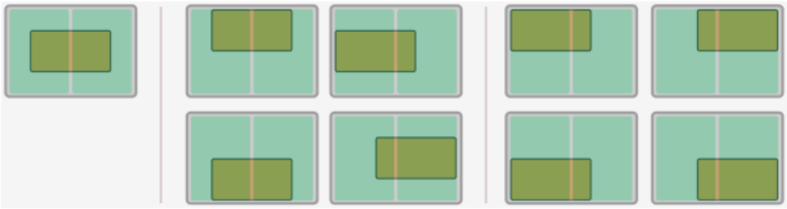
- When a single section is activated

  The figures below demonstrate the effects of using Center to center, Edge midpoint to edge midpoint, and Corner to corner from left to right.



- When both sections are activated

  The figures below demonstrate the effects of using Center to center, Edge midpoint to edge midpoint, and Corner to corner from left to right.

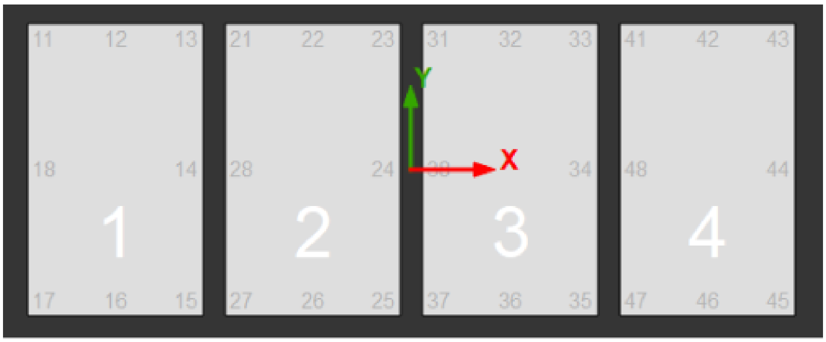


- The effect is similar when there are more than two activated sections of the vacuum gripper.

▼ *By edge/corner ID sequence*

The software attempts the offset according to the user-specified **Edge/Corner ID Sequence** (e.g., 11, 21, 31, 41).

The edge/corner IDs are generated automatically in the vacuum gripper configurator. The two-digit numbers at the edge of each section are edge/corner IDs, as shown below.



▼ *By distance to reference point*

You need to set a reference point to enable this feature. The offset strategies will be sorted according to the distance between the TCP and the reference point, and the offset strategy with a TCP closer to the reference point has a higher priority.

- Reference Point X/Y

  Specify the position of the reference point. The reference point will appear in the 3D simulation area.

- Keep Only High Priority Offset

  Only offset solutions with a shorter distance to the reference point will be kept.

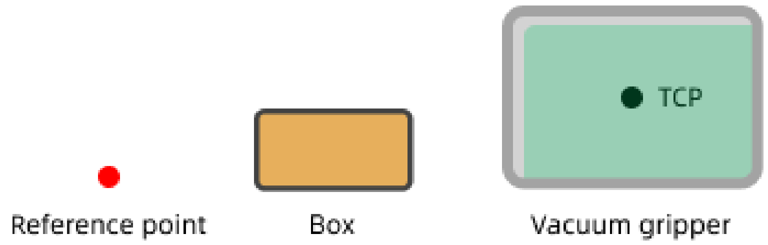
- Max Plan Result Count

  The maximum number of offset solutions that will be kept.

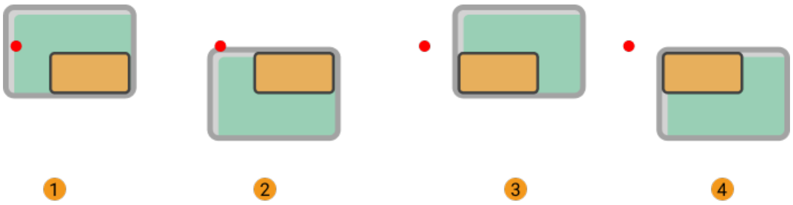
  If there are 4 possible solutions for the vacuum gripper offset in total, and the value of this parameter is set to 2, the 2 solutions in which the vacuum gripper TCPs are closer to the reference point will be kept.

  If there are 2 possible solutions for the vacuum gripper offset in total, and the value of this parameter is set to 4, only 2 solutions will be kept.

- Application Example



The software will attempt the offsets in the order shown below.



Carton Drop Detection

In actual box depalletizing projects, DI check points will be added to the working surface of the suction cups to monitor whether the boxes are dropped during picking and placing by comparing the sensor signal changes.

After the DI check points are added to the vacuum gripper configurator, the software will determine which sensors on the working surface of the vacuum gripper are covered by the picked boxes, and therefore the DI signals that should be detected can be confirmed.

▼ *Remove Edge DI*

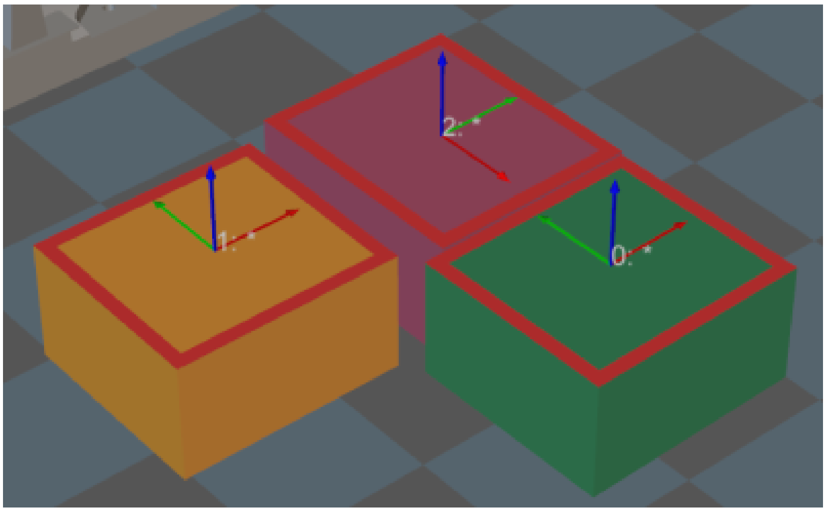
In actual depalletizing process, when the sensors attached to the vacuum gripper near the edge of the box, the software may mistakenly detect that the box is dropped due to large gap between boxes, loose suction at the edge, or other reasons.

To prevent these issues, you can select this parameter and set **Dist from Carton Edge To Remove DI**.

▼ *Dist from Carton Edge To Remove DI*

DI sensors that are within the removal range will not be taken into account in the box drop detection.

The width of the red frame shown in the figure below represents the Dist from Carton Edge To Remove DI. For actual application, please adjust this parameter according to the on-site requirement.



Count of Picked Workobjects

▼ *Limit Total Count*

Select Limit Total Count and you can set the maximum Planned Count and other parameters.

▼ *Limit Count per Picking*

- No limit: There is no limit to the number of workobjects that can be picked at a time.
- Maximum value: The maximum workobjects that can be picked at a time.
- Specified value: Set a fixed limit on the number of workobjects to be picked at a time.

5.5.9.7.3. Box Depalletizing—Multiple at Once

When the **Picking Method** is set to **Box depalletizing**, and the **Depalletizing Mode** is set to **Multiple at once**, you can configure the following parameters.

In addition, you should refer to Configure the Tool to configure the corresponding end tool.

Move-Type Step Common Parameters

▼ *Send Waypoint*

Selected by default to send waypoint poses to the receiver, such as the robot. When this option is unselected, the waypoint pose will not be sent. However, the waypoint will remain in the planned path.

▼ *Try Continuously Running through Succeeding Non-Moves*

Unselected by default. When **non-move Steps**, such as Vision Look, Set DO, Check DI, etc., are connected between **move-type Steps**, the robot's path planning will be interrupted, and the actual robot will take a short pause, reducing the smoothness of running.

When this option is selected, the project will continue to run without waiting for the current move-type Step to complete execution, and therefore the robot can move in a smooth way without pauses. However, enabling this option may cause the execution of the Step to end prematurely.

Why will this option cause the execution of the Step to end prematurely?

Mech-Viz will send multiple poses simultaneously to the robot when the project is running.

When the currently returned JPs of the robot correspond to the last pose sent by Mech-Viz, Mech-Viz will assume that the robot has moved to the last position.

For example, there are 10 move-type Steps in a path, and the pose of the 5th move-type Step is the same as that of the last move-type Step. When the robot moves at low speed, it sends JPs to Mech-Viz when it moves to the 5th move-type Step, Mech-Viz may mistakenly determine that the robot has finished the move-type Steps and prematurely ends the Steps since the poses of the 5th move-type Step and the last move-type Step are the same in the path.

▼ *Do Not Check Collision with Placed Workobject*

Unselected by default, namely that the collision with the already placed objects will not be detected. When this option is selected, the collisions between the robot, end tool, and placed objects will be detected.

In palletizing scenarios, the two possible cases of error are as follows:

1. When the robot is placing a carton, the robot may come into light contact with the placed cartons while no deformation will be caused. After Mech-Viz detects this collision in simulation, it will plan other positions for placing the carton, and therefore a full stack cannot be formed.
2. Usually, the TCP of a suction cup is inside the suction cup model instead of on the surface of it. Under this circumstance, the suction cup may be embedded in the model of the picked carton in the simulation of picking, while the software does not detect the collision between the end tool and the picked object. After the robot places the object and the carton model turns into an object model in the scene, a collision between the suction cup and the carton will be detected and the palletizing cannot be completed.

When this option is selected, no collision between the robot, end tool, and the placed object will be detected, and the above two cases of errors can be avoided.

▼ *Point Cloud Collision Detection Mode*

Select the proper mode according to the requirement of the on-site situation. Usually, the default setting **Auto** can be used. **Do not check collision** mode can be used in move-type Steps before the robot picks the object, and **Check collision** mode can be used after the robot picks the object.

Auto	Default setting. Collision with point cloud is checked only for the “Vision Move” Step and the “Relative Move” Step that depends on the “Vision Move” Step, but not for all move-type Steps.
Do not check collision	Point cloud collisions for all move-type Steps will not be detected.
Check collision	Point cloud collisions for all move-type Steps will be detected.

When Collisions › Collision detection configuration › Detect collision between point cloud and others is switched on, Mech-Viz will detect collisions between the robot model, end tool model, and point cloud when planning the path. Normally, detects whether the robot collides with other objects during picking and placing. When there are point cloud outliers, non-exiting collisions will be detected, which leads to errors in path planning.

▼ *Ignore Workobject Symmetry*

This parameter will only take effect when **Waypoint type** of the Step is set to **Workobject pose**.

None	Default setting, i.e., do not disable symmetry on any axis.

Around workobject frame Z axis	Only disable symmetry on Z-axis of the workobject reference frame.
Around workobject frame X&Y axis	Disable symmetry on X-axis and Y-axis of the workobject reference frame.
Around all axes	Once the object symmetry is disabled, the robot will place the objects strictly according to the workobject poses.

> In some special cases, objects are not pickable due to their peculiar poses. Setting **Rotational symmetry** under Workobjects › Workobject Configuration in Resources may solve this problem. Based on the settings of Rotational symmetry, when the default object pose is not pickable, Mech-Viz will calculate candidate poses and determine whether the candidate poses are pickable. As the candidate poses calculated based on the settings of Rotational symmetry are different from the original poses output from Mech-Vision, the consistency of the objects' place poses cannot be guaranteed.

▼ *Plan Failure Out Port*

After selecting this parameter, this Step will add "Plan failure" out port.

During the planning process, planning is carried out along the branch after the "Success" out port; if the planning fails in the current Step, the branch process after the "Planning failure" out port will be executed.

Held Workobject Collision Detection Settings

▼ *Do Not Check Collision with Scene Object/Robot*

Unselected by default. Once this option is selected, the collisions between the **held workobject** with the **scene objects or robot** will not be detected, and therefore the calculation workload of collision detection will be reduced, the planning speed can be increased, and the cycle time can be shortened. It is usually enabled in the first one or two move-type Steps after the robot picks the object.

Please enable this option cautiously as there may be collision risks.

When **Detect collision between held workobject and others** under Collision detection configuration › Configuration on held workobject is enabled, the software will detect whether the model of the held object collides with the models of the scene objects and the robot.

In palletizing projects, the calculated carton dimensions have millimeter-level errors with the actual dimensions, and frictions between cartons may occur during picking but no collisions will occur. For some move-type Steps that will obviously not cause collisions, detecting such collisions only adds to the calculation workload and planning time, and consequently extending the cycle time. In palletizing projects, enabling Do Not Check Collision with Scene Objects does not affect the collision detection between the held carton and the placed cartons. This option can be enabled when there are scene objects under the stack to avoid failure of finding the palletizing solution.

▼ *Do Not Check Collision with Point Cloud*

Unselected by default. Once this option is selected, the collisions between the **held workobject** with the **point clouds in the scene** will not be detected, and therefore the calculation workload of collision detection will be reduced, the planning speed can be increased, and the cycle time can be shortened.

- When both **Detect collision between held workobject and others** under Collision detection configuration › Configuration on held workobject and **Configuration on point cloud ¬ Detect collision between point cloud and others** are enabled, the software will detect whether the model of the held workobject will collide with the point cloud in the scene.

- When Mech-Vision sends the point cloud and object model to Mech-Viz, the point cloud and the object model are fitted together. After the robot picks the object, the model moves along the planned path, and the collision between the model of the held workobject and the point cloud will occur.
- It is known that the model of the held workobject will have false collisions with the point cloud. Detecting such collisions unnecessarily adds to the calculation workload and extends the planning time.

Vision Result Global Usage

▼ *All Vision Poses in One Move*

Usually, Vision Move will only use one pose of the corresponding vision result. However, once this parameter is selected, the robot will move through all waypoints corresponding to the vision poses at one time.

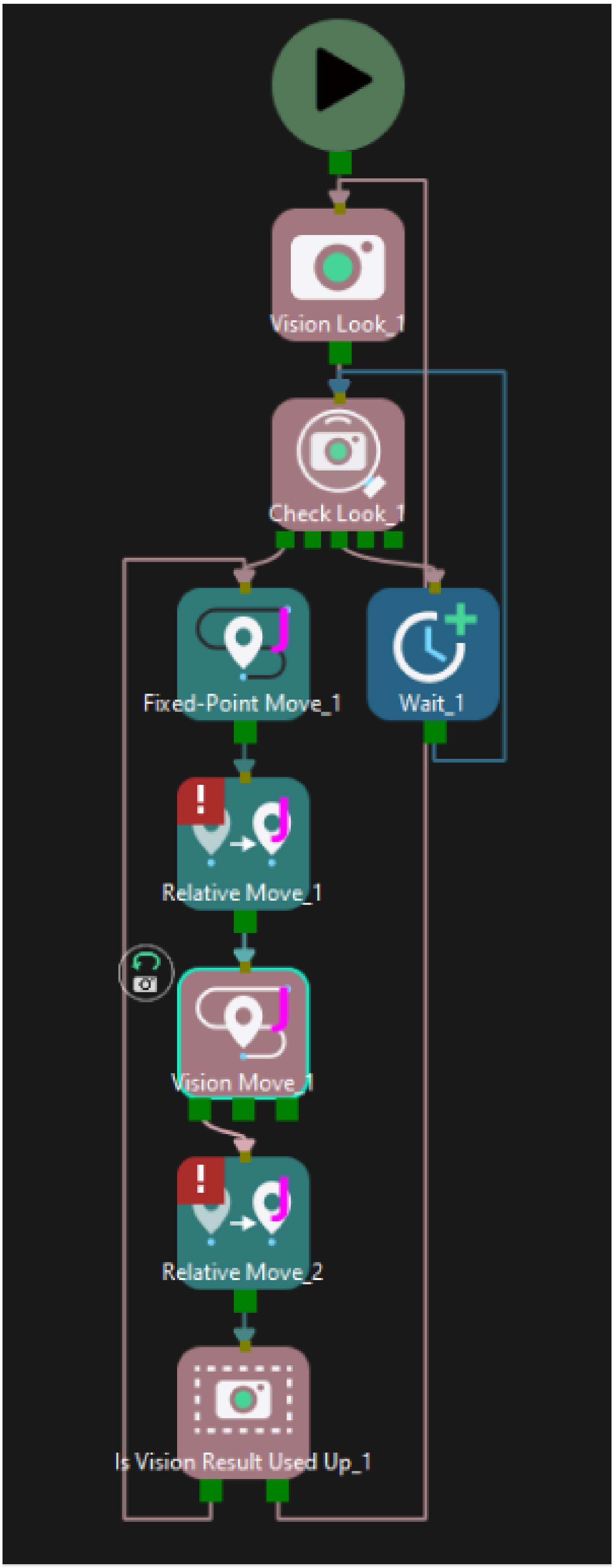
This parameter is usually used when the robot moves in a fixed path where no DO signals will be sent, such as the gluing application.

▼ *Reuse Vision Result*

When this feature is not enabled, each time “Vision Move” plans a robot picking path successfully, the rest unused vision results will be discarded. If this feature is enabled, the vision result that leads to a planning failure and the rest unused vision results will be retained and used for the next planning. This feature should be used with the “Is Vision Result Used Up” Step.

When there are multiple pickable objects in one vision result and **it can be guaranteed that picking one object will not affect the poses of other objects**, this option can be selected. The vision result will be reused and no new image will be captured until all vision results are used up.

Application Example


Vision Look_1
Check Look_1
Fixed-Point Move_1
Wait_1
Relative Move_1
Vision Move_1
Relative Move_2
Is Vision Result Used Up_1

▼ *Share Vision Result*

This feature enables multiple “Vision Move” Steps in which the same vision service is selected to share the vision result.

When one “Vision Move” Step plans successfully, the corresponding vision result will be used, and the unused vision result will be used by the next “Vision Move” Step with this feature enabled. After all “Vision Move” Steps that share the vision result complete planning, the rest results will be discarded.

In addition, this feature can be used in conjunction with the “Reuse Vision Result” feature. When “Reuse Vision Result” is selected, the rest vision results will not be discarded even after all “Vision Move” Steps have completed one round of planning.

Application Example

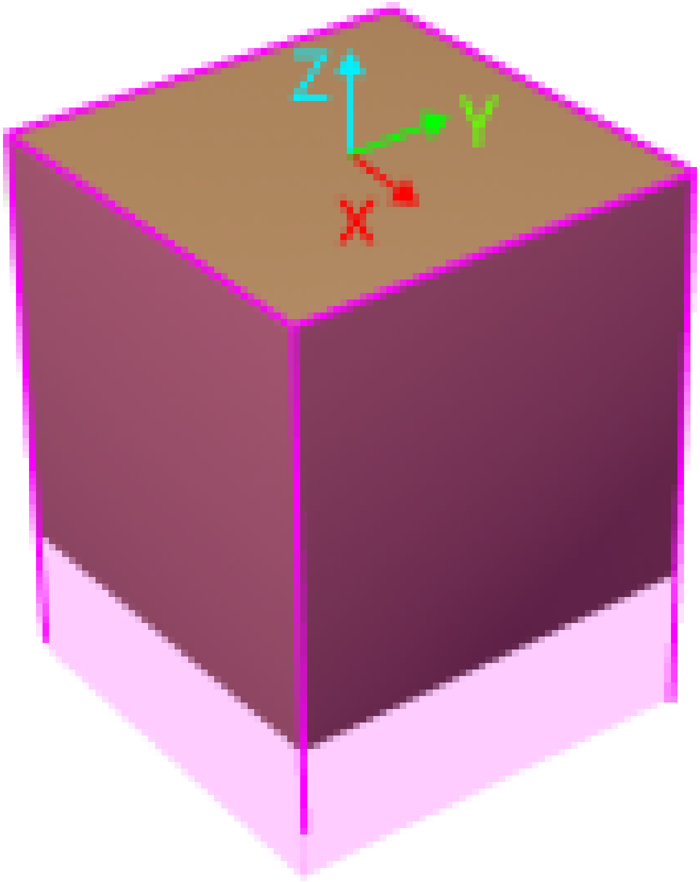
(To be supplemented)

Remove Point Cloud of Target Workobject

▼ *Remove Point Cloud of Target Workobject*

Select to remove the point cloud of the target workobject to be picked, and only collision between the robot end tool and non-target workobjects will be detected.

To avoid picking non-target boxes, the software will remove the point cloud of the target box and keep the point cloud of its adjacent boxes. During path planning, the software will discard a solution that may lead to a wrong picking due to the detected collision between the model of the vacuum gripper’s buffer and the point cloud of adjacent boxes.

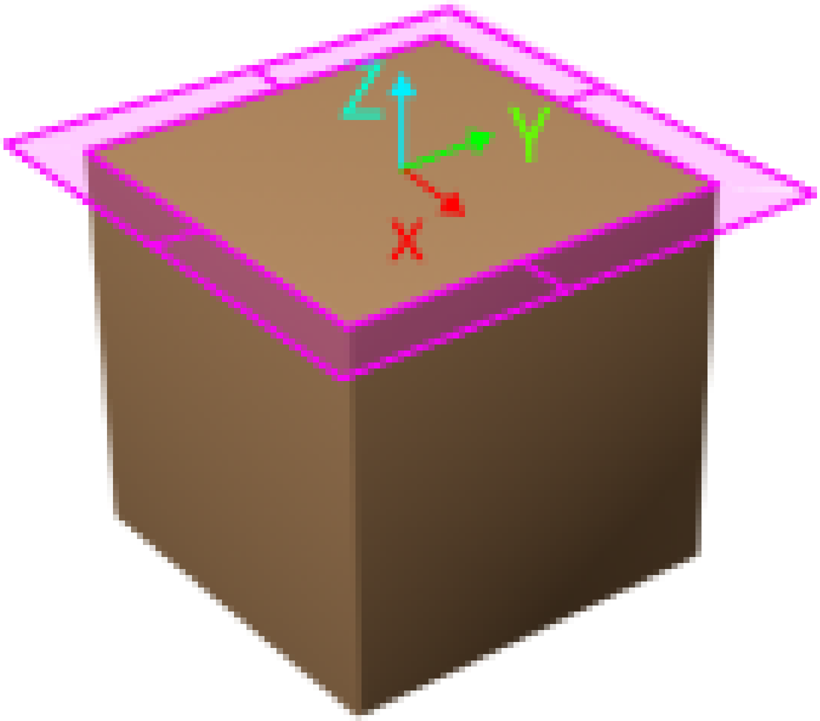


The default range for point cloud removal is a cuboid whose top surface is the top surface of the target box with an infinite height whose direction is the Z-direction of the box's vision pose.

You can adjust the "XY-Plane Point Cloud Removal Range Expansion" and "Z-Direction Point Cloud Removal Range Expansion" to expand the point cloud removal range for removing the point cloud near the target workobject.

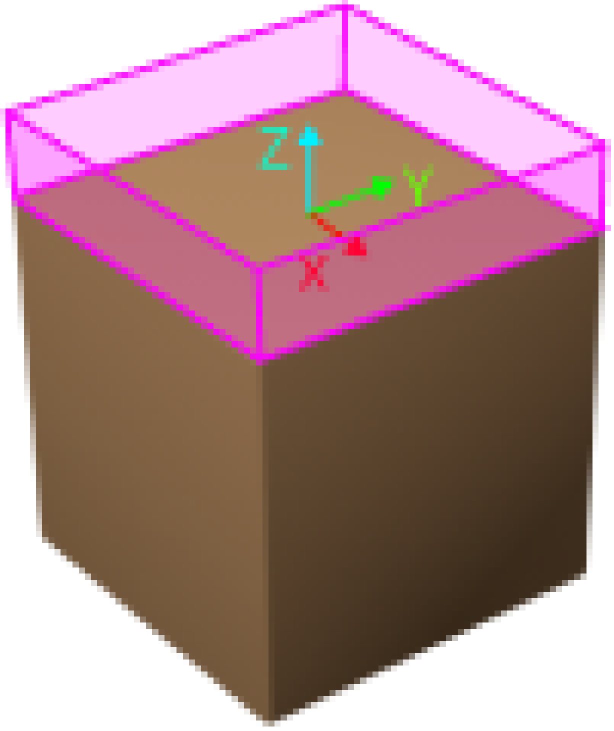
▼ *XY-Plane Point Cloud Removal Range Expansion*

Based on the original workobject dimensions, the XOY plane of the workobject pose is the top surface for the point cloud removal range. This parameter specifies the extended length from actual workobject edges on this surface.



▼ *Z-Direction Point Cloud Removal Range Expansion*

Based on the original workobject dimensions, the extended height in the positive Z-direction of the workobject pose for the point cloud removal range.



Supplementary Features

▼ *Update Bin Pose with Vision*

Once the “Update Bin Pose with Vision” parameter is selected, the bin will be recognized and located simultaneously while the camera captures the image for workobject recognition. Therefore, the bin pose in the simulation area can be updated dynamically, which facilitates the collision detection algorithm to effectively prevent the robot from colliding with the bin.

This feature will provide three fields, “scene_object_names,” “scene_object_sizes,” and “scene_object_poses,” in the vision result to define the name, dimensions, and position of the scene object to be updated.

▼ *Result Must Be in Specified Bin*

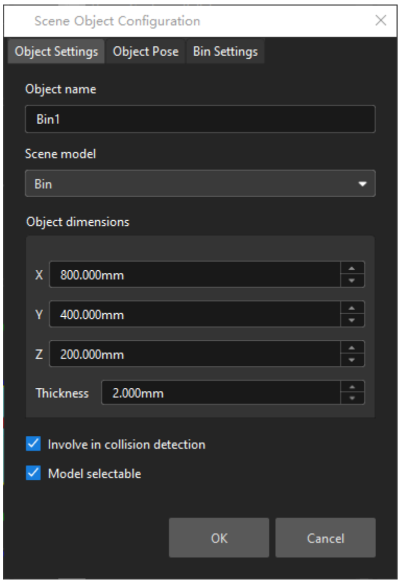
Result Must Be in Specified Bin

This feature limits that the vision result received by the current “Vision Move” Step must be in a specified bin. The vision result out of the effective bin range will not be used.

The effective range for each bin can be configured separately in the corresponding configuration window. Only bins with “Set effective range for pose” enabled can be selected in the drop-down menu.

Application Example

Add “Bin1”, and select the “Set effective range for pose” parameter for this bin.



Object name: Bin1

Scene model: Bin

Object dimensions

X: 800.000mm

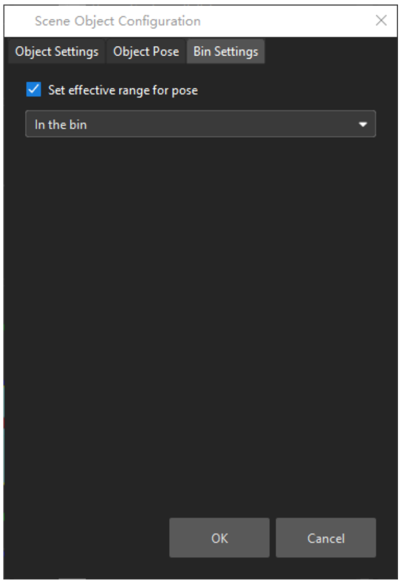
Y: 400.000mm

Z: 200.000mm

Thickness: 2.000mm

☑ Involve in collision detection

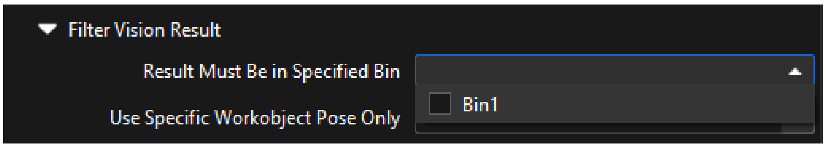
☑ Model selectable



☑ Set effective range for pose

In the bin

Select “Bin1” in the drop-down menu.



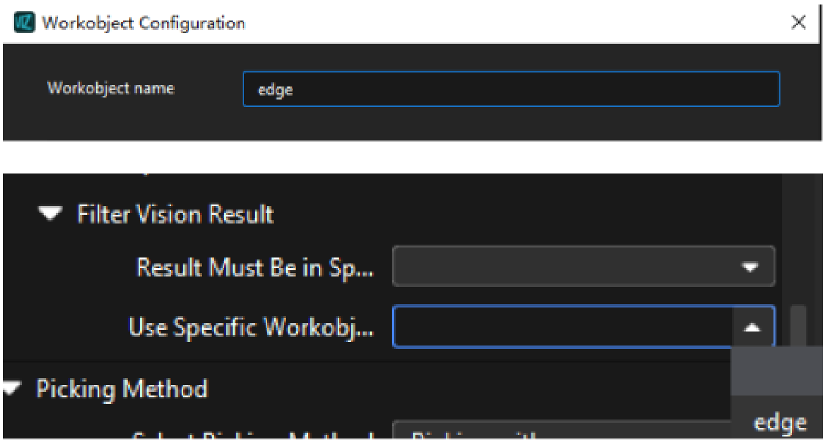
Result Must Be in Specified Bin: □ Bin1

Use Specific Workobject Pose Only

▼ *Use Specific Workobject Pose Only*

This parameter is used to filter the workobjects corresponding to the specified “labels” field in the vision results.

If you need to use this parameter, you need to add the corresponding workobjects in the resource tree whose name is the same as the “labels” field name, and then select the workobject in the dropdown menu on the right side of the parameter.



Workobject name: edge

▼ *Filter Out Unlikely Pick Point*

This parameter group is mainly used to avoid picking on the same pick point in scenarios where the object cannot be picked successfully.

Once the “Filter Out Unlikely Pick Point” parameter is selected, you can adjust the following parameters.

- Filter Subject
  - Pick point: only pick points with high failure rates will be deprioritized in the picking and discarded in some rounds.
  - Workobject: if any one pick point on the workobject is considered to have a high failure rate, all pick points on this workobject will be deprioritized in the picking and discarded in some rounds.
- Priority Downgrade Radius

  A sphere with this parameter value as its radius and the pick point successfully planned in the last round as its center will be introduced. If a pick point in the latest vision result falls within this sphere, the pick point will be put off for picking with a decreased picking priority.
- Discard Attempt Radius

  A sphere with this parameter value as its radius and the pick point successfully planned in the last round as its center will be introduced. If a pick point in the latest vision result falls within this sphere, the pick point will be discarded in this round of planning.

  For example, if the robot only moves the crankshaft but fails to pick it on the first attempt, there is a possibility of successful picking on the next attempt. Therefore, **Priority Downgrade Radius** can be set to downgrade the priority of the pick point, while the pick point will not be discarded. When the robot fails to move the crankshaft at all in the first attempt, it is highly unlikely that the crankshaft can be picked successfully in the next attempt, and therefore **Discard Attempt Radius** can be set to discard the pose directly.
- List Length

  A list of pick points with high failure rates. When the list length exceeds the upper limit, the pick point that is added to the list earliest will be removed from the list and can be used in

the next round of planning.

Picking Sequence

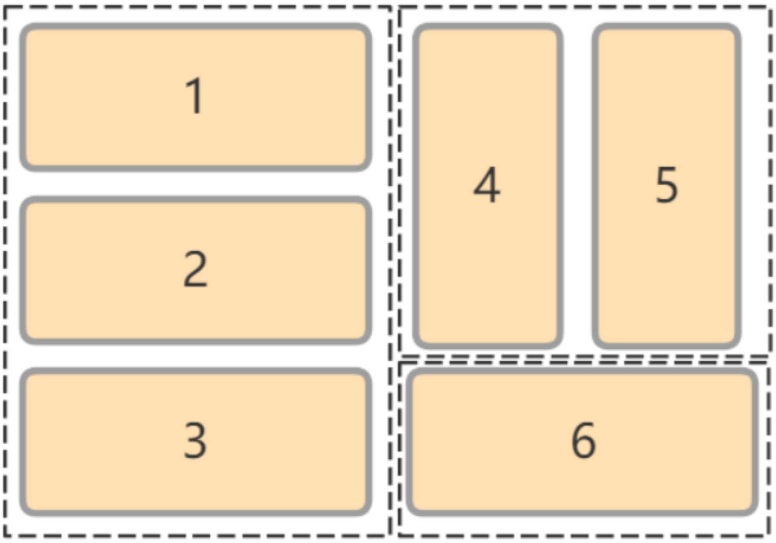
▼ *Sorting Strategy*

This parameter is used to specify the sorting strategy when picking workobjects.

- **Sort by the workobject count in the combination**

  The workobject combinations will be sorted by the number of workobjects they contain, and those with more workobjects will be picked first.

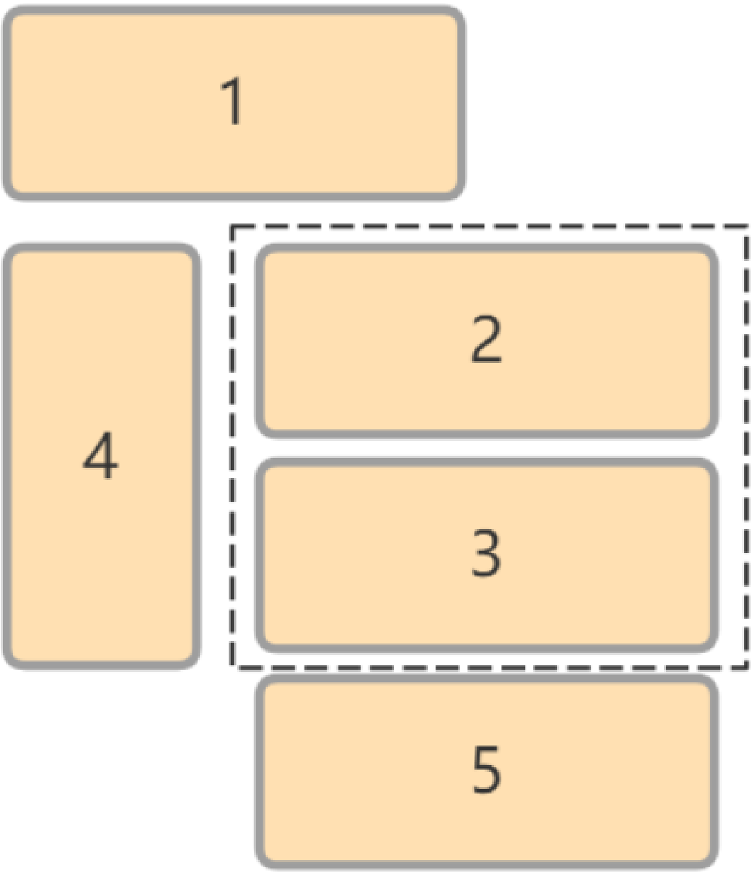
  If there are three combinations, 1-2-3, 4-5, and 6, the picking sequence will be 1-2-3, 4-5, and 6.



- **In the strict order of vision result**

  Pick in the order of the result sent by the vision service. If the path planning fails by using a vision result, the current result will not be skipped and the path planning will fail.

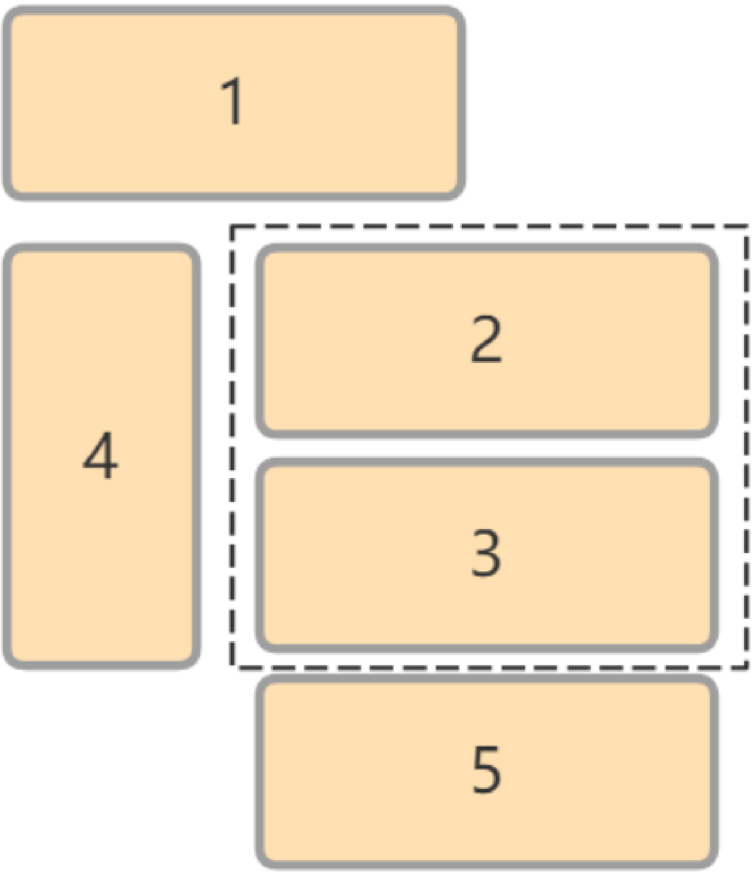
  For example, as shown in the figure below, the picking sequence is 1, 2-3, 4, and 5. If the planning of the box group 2-3 fails, only box 1 can be picked successfully.



- In the partial order of vision result

  Pick in the order of the result sent by the vision service. If the path planning fails by using a vision result, the current result will be skipped. Then the next vision result will be used for the subsequent path planning.

  For example, as shown in the figure below, the picking sequence is 1, 2-3, 4, and 5. If the planning of the box group 2-3 fails, the planning of box 4 and 5 can be continued.



Box grouping

There are three grouping strategies in total, and each strategy contains different parameters.

▼ *Default Strategy*

Group boxes along the X-axis or Y-axis of the box pose.

▼ *Group along Box Frame Axis*

One box will be used as the reference, and other adjacent boxes along the X-axis or Y-axis of this box's pose will be grouped.

- **Grouping Direction**

  Specify the axis of the reference pose for box grouping.

- **Grouping Restrictions**
- Group Row Count Upper Limit

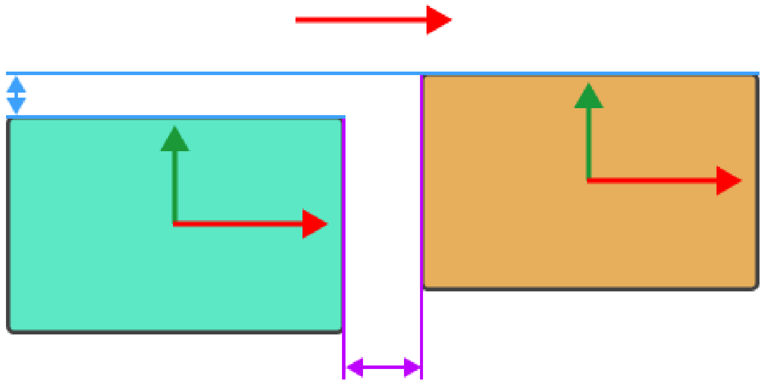
  The allowable maximum number of rows in one box group for multi-pick depalletizing.

- Spacing Upper Limit

  This parameter limits the maximum distance between two adjacent boxes in the grouping direction. When the distance is smaller than this value, the boxes can be considered to be in a group.

Please note that the value of this parameter cannot be greater than the box width in the vision result. If the set value is greater than the box width, the value will be automatically modified to the box width in the actual execution.

The range indicated by the purple arrow below is the Spacing Upper Limit. When the actual distance is smaller than this value, the boxes can be considered to be in the same group.

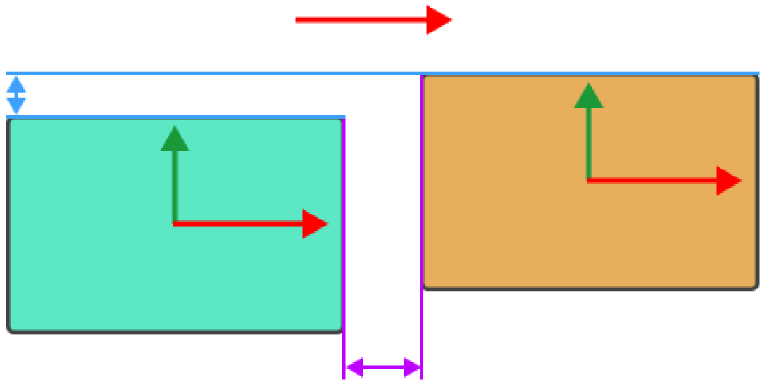


- Deviation Upper Limit

  This parameter limits the maximum deviation distance of a box in the direction perpendicular to the grouping direction. When the deviation distance is smaller than this value, the box can be considered to be in the group.

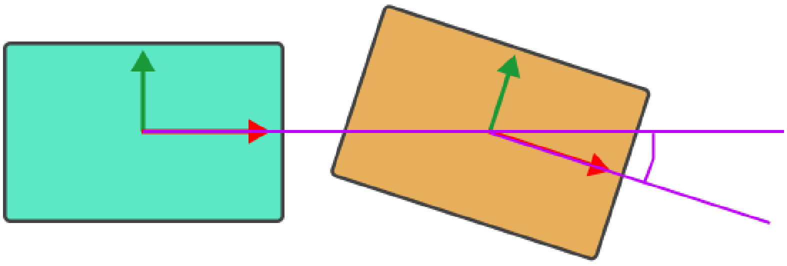
  Please note that the value of this parameter cannot be greater than the box width in the vision result. If the set value is greater than the box width, the value will be automatically modified to the box width in the actual execution.

  The range indicated by the blue arrow below is the Deviation Upper Limit. When the actual distance is smaller than this value, the box can be considered to be in the group.



- Angle Deviation Upper Limit

  This parameter limits the angle of rotational deviation of a box relative to the grouping direction. When the rotation angle is smaller than this value, the box can be considered to be in the group.

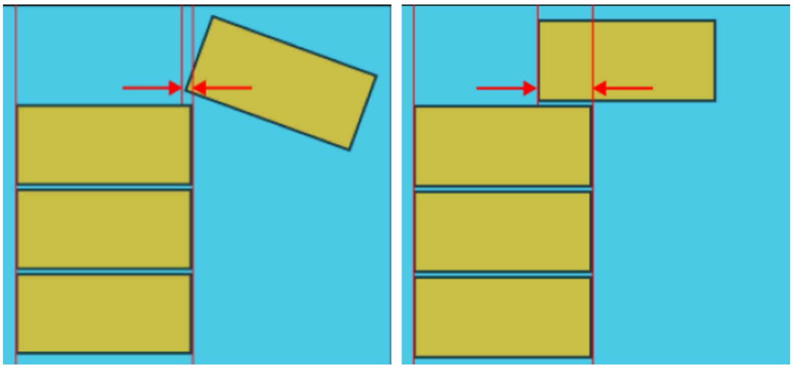


- Only Group Complete Row

  In actual applications, a complete row of boxes might be neglected because there are part of other boxes at ends in the grouping direction. Setting an **Obstruction Tolerance Distance** can preclude the intruding box from affecting the grouping result.

- Obstruction Tolerance Distance

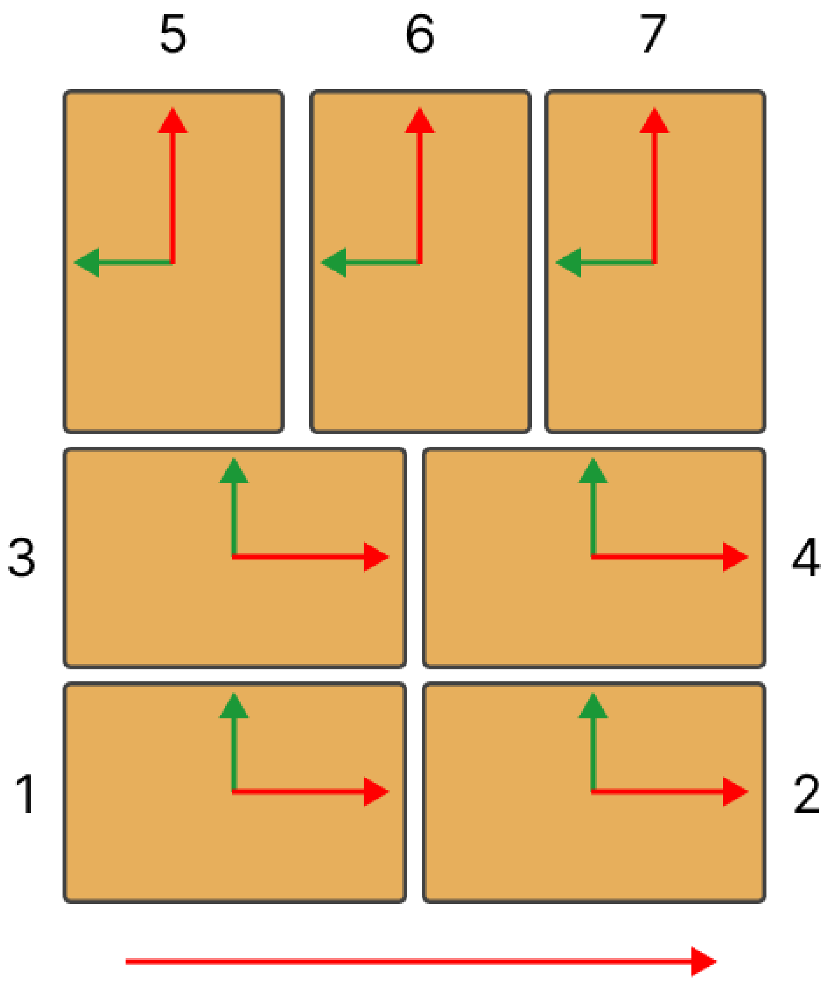
  For a box group, if any other box intrudes in the direction perpendicular to the grouping direction, and the intrusion distance is smaller than this threshold, the intruding box will be ignored, and the box group will be considered a complete row. Otherwise, the box group will be discarded.



The set value of Obstruction Tolerance Distance cannot exceed the edge length of the boxes.

▼ *Group along Custom Frame Axis*

As shown in the figure below, if the 7 boxes are grouped along the axis in box reference frame:



- The possible groups along X-axis of the box are: 1-2, 3-4, 5, 6, 7.
- The possible groups along Y-axis of the box are:1-3, 2-4, 5-6-7.

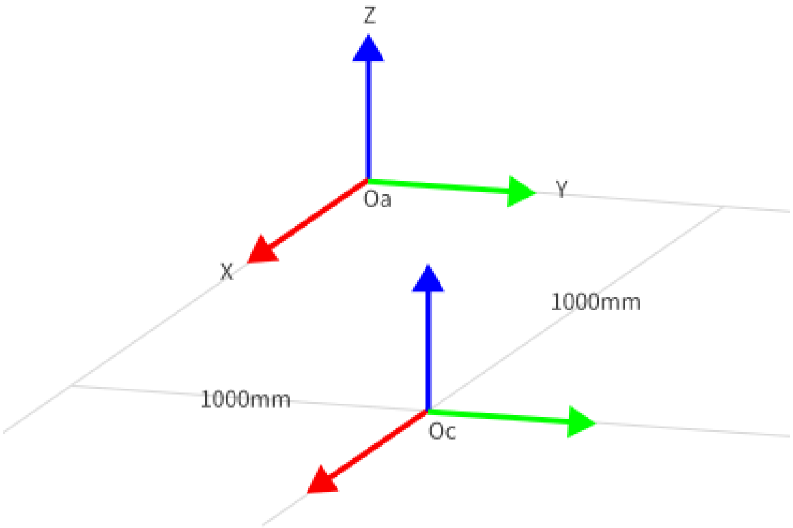
If you want the boxes to be grouped into 1-2, 3-4, 5-6-7, all boxes should be grouped along the direction indicated by the red arrow at the bottom of the figure. The red arrow is an axis of a custom reference frame.

- Grouping Direction

  Specify the axis of the reference pose for box grouping.

- X/Y/Z Coord. of Custom Frame Origin

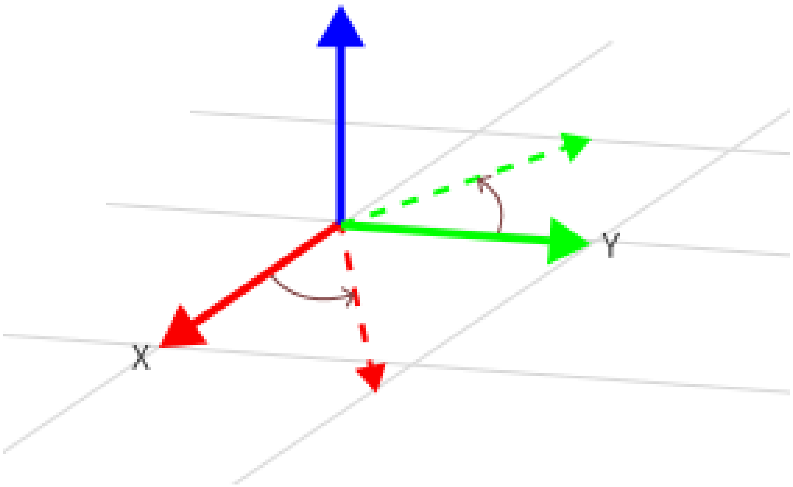
  Specify three coordinates to define the origin of the custom reference frame.



Oa indicates the robot reference frame, and Oc indicates the custom reference frame.

- Custom Frame Rotation Angle

  Specify the angle that the custom reference frame rotates around its Z-axis.



  - Group Row Count Upper Limit

    The allowable maximum number of rows in one box group for multi-pick depalletizing.

  - Spacing Upper Limit

    This parameter limits the maximum distance between two adjacent boxes in the grouping direction. When the distance is smaller than this value, the boxes can be considered to be in a group.

    Please note that the value of this parameter cannot be greater than the box width in the vision result. If the set value is greater than the box width, the value will be automatically modified to the box width in the actual execution.

The range indicated by the purple arrow below is the Spacing Upper Limit. When the actual distance is smaller than this value, the boxes can be considered to be in the same group.

[sapcing and deviation upper limit] | *sapcing_and_deviation_upper_limit.png*

- Deviation Upper Limit

  This parameter limits the maximum deviation distance of a box in the direction perpendicular to the grouping direction. When the deviation distance is smaller than this value, the box can be considered to be in the group.

  Please note that the value of this parameter cannot be greater than the box width in the vision result. If the set value is greater than the box width, the value will be automatically modified to the box width in the actual execution.

  The range indicated by the blue arrow below is the Deviation Upper Limit. When the actual distance is smaller than this value, the box can be considered to be in the group.

  [sapcing and deviation upper limit] | *sapcing_and_deviation_upper_limit.png*

- Angle Deviation Upper Limit

  This parameter limits the angle of rotational deviation of a box relative to the grouping direction. When the rotation angle is smaller than this value, the box can be considered to be in the group.

  [angle deviation upper limit] | *angle_deviation_upper_limit.png*

- Only Group Complete Row

  In actual applications, a complete row of boxes might be neglected because there are part of other boxes at ends in the grouping direction. Setting an **Obstruction Tolerance Distance** can preclude the intruding box from affecting the grouping result.

- Obstruction Tolerance Distance

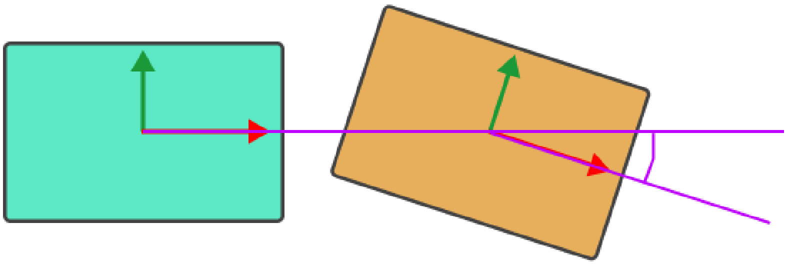
  For a box group, if any other box intrudes in the direction perpendicular to the grouping direction, and the intrusion distance is smaller than this threshold, the intruding box will be ignored, and the box group will be considered a complete row. Otherwise, the box group will be discarded.

  [obstruction tolerance distance] | *obstruction_tolerance_distance.png*

The set value of Obstruction Tolerance Distance cannot exceed the edge length of the boxes.

- **Box-Frame Angle Threshold**

  Before the boxes are grouped, the angle between the box reference frame and the custom reference frame will be calculated. If the angle exceeds this threshold, the corresponding box will not be grouped.



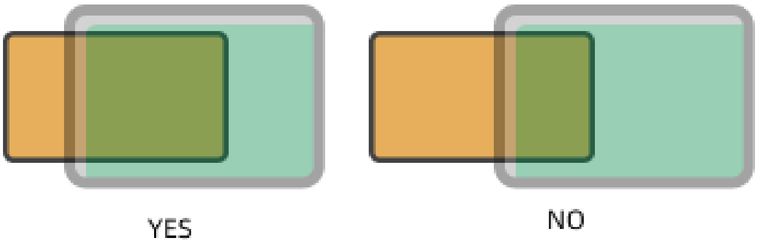
Suction Cup Offset

▼ *Box Surface Coverage Lower Limit*

This parameter specifies the minimum proportion of a box's upper surface covered by the vacuum gripper when the vacuum gripper attempts to pick a box. If the coverage is below the threshold, it is considered that the vacuum gripper cannot attach to the box firmly and the picking is not allowed.

In addition, when the upper surface of a single box is larger than the entire vacuum gripper, and all sections of the vacuum gripper are activated, the coverage can reach 100%.

Supposing that this parameter is set to 50%, the picking feasibilities are shown as below.

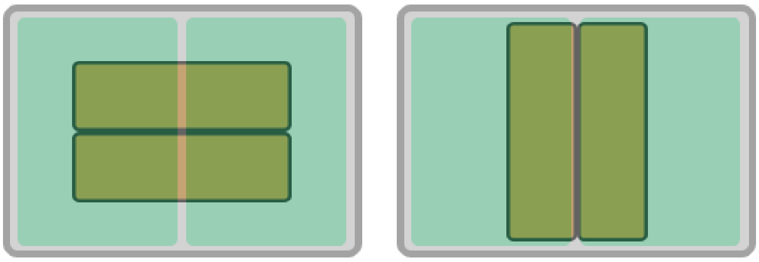


During the depalletizing process of the boxes, the fixed and non-replaceable suction cups are usually used. Due to various sizes of the boxes, the suction cups need to handle different boxes of various sizes. To avoid picking non-target boxes, the robot may not pick the target box with the center of the vacuum gripper. Instead, the corner or other parts of the vacuum gripper will be used. By setting the orientation of the suction cups and configuring the picking strategy, the problem of mis-suction can be effectively avoided, ensuring stable picking of boxes of various sizes.

▼ *Suction Cup Orientation*

This parameter specifies the orientation of the vacuum gripper relative to the box group during picking.

- Random orientation: The orientation of the suction cup is not specified.
- Custom: You can set the **Suction Cup X-axis Orientation** and **Orientation Reference**, and there are four relative positional relationships in total:
  - X-axes of objects are parallel to that of the vacuum gripper.
  - X-axes of objects are perpendicular to that of the vacuum gripper.
  - The X-axis of the vacuum gripper is parallel to the direction along which the boxes are grouped.
  - The X-axis of the vacuum gripper is perpendicular to the direction along which the boxes are grouped.



The figure on the left shows the effect when the X-axis of the vacuum gripper is parallel to the long edge of the box group, and the figure on the right shows the effect when the X-axis of the vacuum gripper is perpendicular to the long edge of the box group.

There are four offset strategies in total, and each strategy contains different parameters.

▼ *Default strategy*

The software sequentially applies three strategies—Center to center, Edge midpoint to edge midpoint, and Corner to corner—to attempt the vacuum gripper offset.

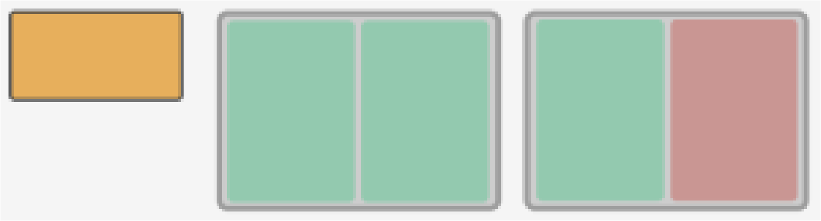
▼ *Custom strategy priority*

The software adheres to user-defined offset priorities to attempt the vacuum gripper offset.

There are three custom strategies in total, which are Center to center, Edge midpoint to edge midpoint, and Corner to corner. You can set the priority levels according to actual requirements.

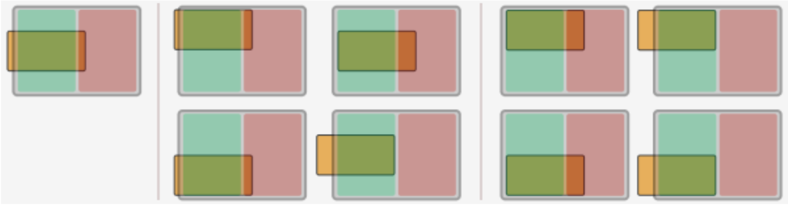
- **Center to center**: The center of the activated section of the vacuum gripper matches that of the box.
- **Edge midpoint to edge midpoint**: The edge midpoint of the activated section of the vacuum gripper matches that of the box.
- **Corner to corner**: The corner of the activated section of the vacuum gripper matches that of the box.

In the figure below, the orange rectangle represents the box, and the gray rectangle represents the vacuum gripper. The green part of the vacuum gripper indicates that the section is activated, while the red part indicates that the section is deactivated.



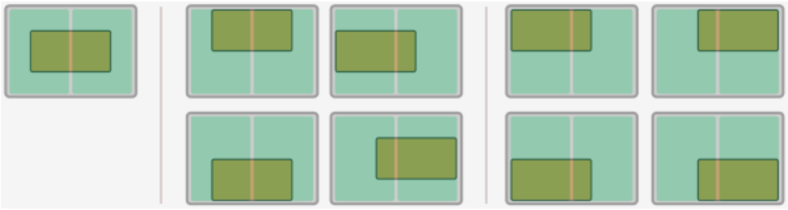
- When a single section is activated

  The figures below demonstrate the effects of using Center to center, Edge midpoint to edge midpoint, and Corner to corner from left to right.



- When both sections are activated

  The figures below demonstrate the effects of using Center to center, Edge midpoint to edge midpoint, and Corner to corner from left to right.

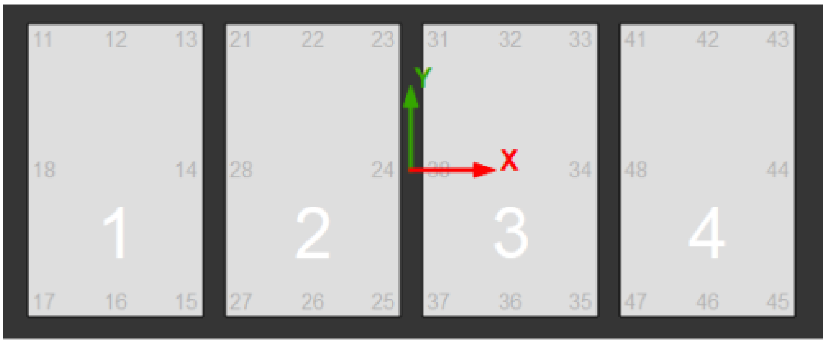


- The effect is similar when there are more than two activated sections of the vacuum gripper.

▼ *By edge/corner ID sequence*

The software attempts the offset according to the user-specified **Edge/Corner ID Sequence** (e.g., 11, 21, 31, 41).

The edge/corner IDs are generated automatically in the vacuum gripper configurator. The two-digit numbers at the edge of each section are edge/corner IDs, as shown below.



▼ *By distance to reference point*

You need to set a reference point to enable this feature. The offset strategies will be sorted according to the distance between the TCP and the reference point, and the offset strategy with a TCP closer to the reference point has a higher priority.

- Reference Point X/Y

  Specify the position of the reference point. The reference point will appear in the 3D simulation area.

- Keep Only High Priority Offset

  Only offset solutions with a shorter distance to the reference point will be kept.

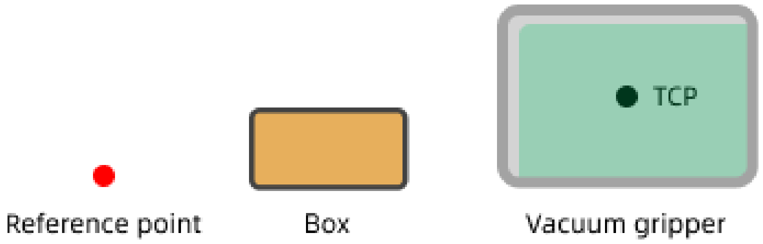
- Max Plan Result Count

  The maximum number of offset solutions that will be kept.

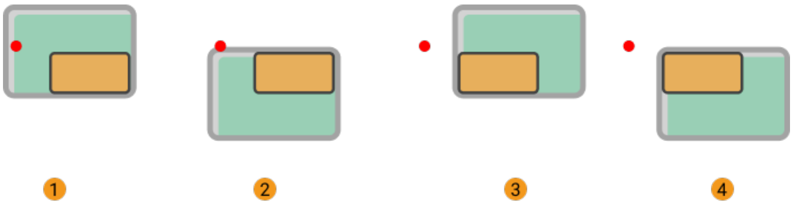
  If there are 4 possible solutions for the vacuum gripper offset in total, and the value of this parameter is set to 2, the 2 solutions in which the vacuum gripper TCPs are closer to the reference point will be kept.

If there are 2 possible solutions for the vacuum gripper offset in total, and the value of this parameter is set to 4, only 2 solutions will be kept.

- Application Example



The software will attempt the offsets in the order shown below.



Carton Drop Detection

In actual box depalletizing projects, DI check points will be added to the working surface of the suction cups to monitor whether the boxes are dropped during picking and placing by comparing the sensor signal changes.

After the DI check points are added to the vacuum gripper configurator, the software will determine which sensors on the working surface of the vacuum gripper are covered by the picked boxes, and therefore the DI signals that should be detected can be confirmed.

▼ *Remove Edge DI*

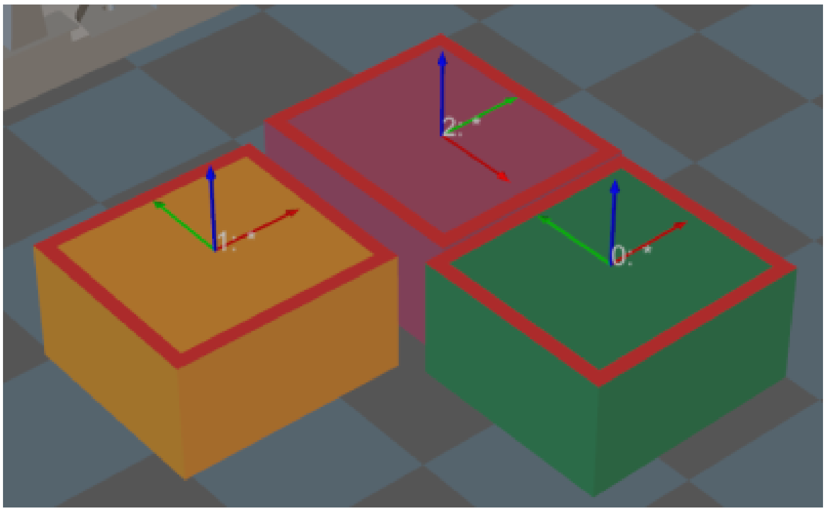
In actual depalletizing process, when the sensors attached to the vacuum gripper near the edge of the box, the software may mistakenly detect that the box is dropped due to large gap between boxes, loose suction at the edge, or other reasons.

To prevent these issues, you can select this parameter and set **Dist from Carton Edge To Remove DI**.

▼ *Dist from Carton Edge To Remove DI*

DI sensors that are within the removal range will not be taken into account in the box drop detection.

The width of the red frame shown in the figure below represents the Dist from Carton Edge To Remove DI. For actual application, please adjust this parameter according to the on-site requirement.



Count of Picked Workobjects

▼ *Limit Total Count*

Select Limit Total Count and you can set the maximum Planned Count and other parameters.

▼ *Limit Count per Picking*

- No limit: There is no limit to the number of workobjects that can be picked at a time.
- Maximum value: The maximum workobjects that can be picked at a time.
- Specified value: Set a fixed limit on the number of workobjects to be picked at a time.

5.5.9.7.4. Picking with array

When the **Picking Method** is set to **Picking with array**, you can configure the following parameters.

In addition, you should refer to Configure the Tool to configure the corresponding end tool.

Move-Type Step Common Parameters

▼ *Send Waypoint*

Selected by default to send waypoint poses to the receiver, such as the robot. When this option is unselected, the waypoint pose will not be sent. However, the waypoint will remain in the planned path.

▼ *Try Continuously Running through Succeeding Non-Moves*

Unselected by default. When **non-move Steps**, such as Vision Look, Set DO, Check DI, etc., are connected between **move-type Steps**, the robot's path planning will be interrupted, and the actual robot will take a short pause, reducing the smoothness of running.

When this option is selected, the project will continue to run without waiting for the current move-type Step to complete execution, and therefore the robot can move in a smooth way without pauses. However, enabling this option may cause the execution of the Step to end prematurely.

Why will this option cause the execution of the Step to end prematurely?

Mech-Viz will send multiple poses simultaneously to the robot when the project is running. When the currently returned JPs of the robot correspond to the last pose sent by Mech-Viz, Mech-Viz will assume that the robot has moved to the last position.

For example, there are 10 move-type Steps in a path, and the pose of the 5th move-type Step is the same as that of the last move-type Step. When the robot moves at low speed, it sends JPs to Mech-Viz when it moves to the 5th move-type Step, Mech-Viz may mistakenly determine that the robot has finished the move-type Steps and prematurely ends the Steps since the poses of the 5th move-type Step and the last move-type Step are the same in the path.

▼ *Do Not Check Collision with Placed Workobject*

Unselected by default, namely that the collision with the already placed objects will not be detected. When this option is selected, the collisions between the robot, end tool, and placed objects will be detected.

In palletizing scenarios, the two possible cases of error are as follows:

1. When the robot is placing a carton, the robot may come into light contact with the placed cartons while no deformation will be caused. After Mech-Viz detects this collision in simulation, it will plan other positions for placing the carton, and therefore a full stack cannot be formed.
2. Usually, the TCP of a suction cup is inside the suction cup model instead of on the surface of it. Under this circumstance, the suction cup may be embedded in the model of the picked carton in the simulation of picking, while the software does not detect the collision between the end tool and the picked object. After the robot places the object and the carton model turns into an object model in the scene, a collision between the suction cup and the carton will be detected and the palletizing cannot be completed.

When this option is selected, no collision between the robot, end tool, and the placed object will be detected, and the above two cases of errors can be avoided.

▼ *Point Cloud Collision Detection Mode*

Select the proper mode according to the requirement of the on-site situation. Usually, the default setting **Auto** can be used. **Do not check collision** mode can be used in move-type Steps before the robot picks the object, and **Check collision** mode can be used after the robot picks the object.

Auto	Default setting. Collision with point cloud is checked only for the “Vision Move” Step and the “Relative Move” Step that depends on the “Vision Move” Step, but not for all move-type Steps.
Do not check collision	Point cloud collisions for all move-type Steps will not be detected.
Check collision	Point cloud collisions for all move-type Steps will be detected.

When Collisions › Collision detection configuration › Detect collision between point cloud and others is switched on, Mech-Viz will detect collisions between the robot model, end tool model, and point cloud when planning the path. Normally, detects whether the robot collides with other objects during picking and placing. When there are point cloud outliers, non-exiting collisions will be detected, which leads to errors in path planning.

▼ *Ignore Workobject Symmetry*

This parameter will only take effect when **Waypoint type** of the Step is set to **Workobject pose**.

None	Default setting, i.e., do not disable symmetry on any axis.

Around workobject frame Z axis	Only disable symmetry on Z-axis of the workobject reference frame.
Around workobject frame X&Y axis	Disable symmetry on X-axis and Y-axis of the workobject reference frame.
Around all axes	Once the object symmetry is disabled, the robot will place the objects strictly according to the workobject poses.

> In some special cases, objects are not pickable due to their peculiar poses. Setting **Rotational symmetry** under Workobjects › Workobject Configuration in Resources may solve this problem. Based on the settings of Rotational symmetry, when the default object pose is not pickable, Mech-Viz will calculate candidate poses and determine whether the candidate poses are pickable. As the candidate poses calculated based on the settings of Rotational symmetry are different from the original poses output from Mech-Vision, the consistency of the objects' place poses cannot be guaranteed.

▼ *Plan Failure Out Port*

After selecting this parameter, this Step will add "Plan failure" out port.

During the planning process, planning is carried out along the branch after the "Success" out port; if the planning fails in the current Step, the branch process after the "Planning failure" out port will be executed.

Held Workobject Collision Detection Settings

▼ *Do Not Check Collision with Scene Object/Robot*

Unselected by default. Once this option is selected, the collisions between the **held workobject** with the **scene objects or robot** will not be detected, and therefore the calculation workload of collision detection will be reduced, the planning speed can be increased, and the cycle time can be shortened. It is usually enabled in the first one or two move-type Steps after the robot picks the object.

Please enable this option cautiously as there may be collision risks.

When **Detect collision between held workobject and others** under Collision detection configuration › Configuration on held workobject is enabled, the software will detect whether the model of the held object collides with the models of the scene objects and the robot.

In palletizing projects, the calculated carton dimensions have millimeter-level errors with the actual dimensions, and frictions between cartons may occur during picking but no collisions will occur. For some move-type Steps that will obviously not cause collisions, detecting such collisions only adds to the calculation workload and planning time, and consequently extending the cycle time. In palletizing projects, enabling Do Not Check Collision with Scene Objects does not affect the collision detection between the held carton and the placed cartons. This option can be enabled when there are scene objects under the stack to avoid failure of finding the palletizing solution.

▼ *Do Not Check Collision with Point Cloud*

Unselected by default. Once this option is selected, the collisions between the **held workobject** with the **point clouds in the scene** will not be detected, and therefore the calculation workload of collision detection will be reduced, the planning speed can be increased, and the cycle time can be shortened.

- When both **Detect collision between held workobject and others** under Collision detection configuration › Configuration on held workobject and **Configuration on point cloud ¬ Detect collision between point cloud and others** are enabled, the software will detect whether the model of the held workobject will collide with the point cloud in the scene.

- When Mech-Vision sends the point cloud and object model to Mech-Viz, the point cloud and the object model are fitted together. After the robot picks the object, the model moves along the planned path, and the collision between the model of the held workobject and the point cloud will occur.
- It is known that the model of the held workobject will have false collisions with the point cloud. Detecting such collisions unnecessarily adds to the calculation workload and extends the planning time.

Vision Result Global Usage

▼ *All Vision Poses in One Move*

Usually, Vision Move will only use one pose of the corresponding vision result. However, once this parameter is selected, the robot will move through all waypoints corresponding to the vision poses at one time.

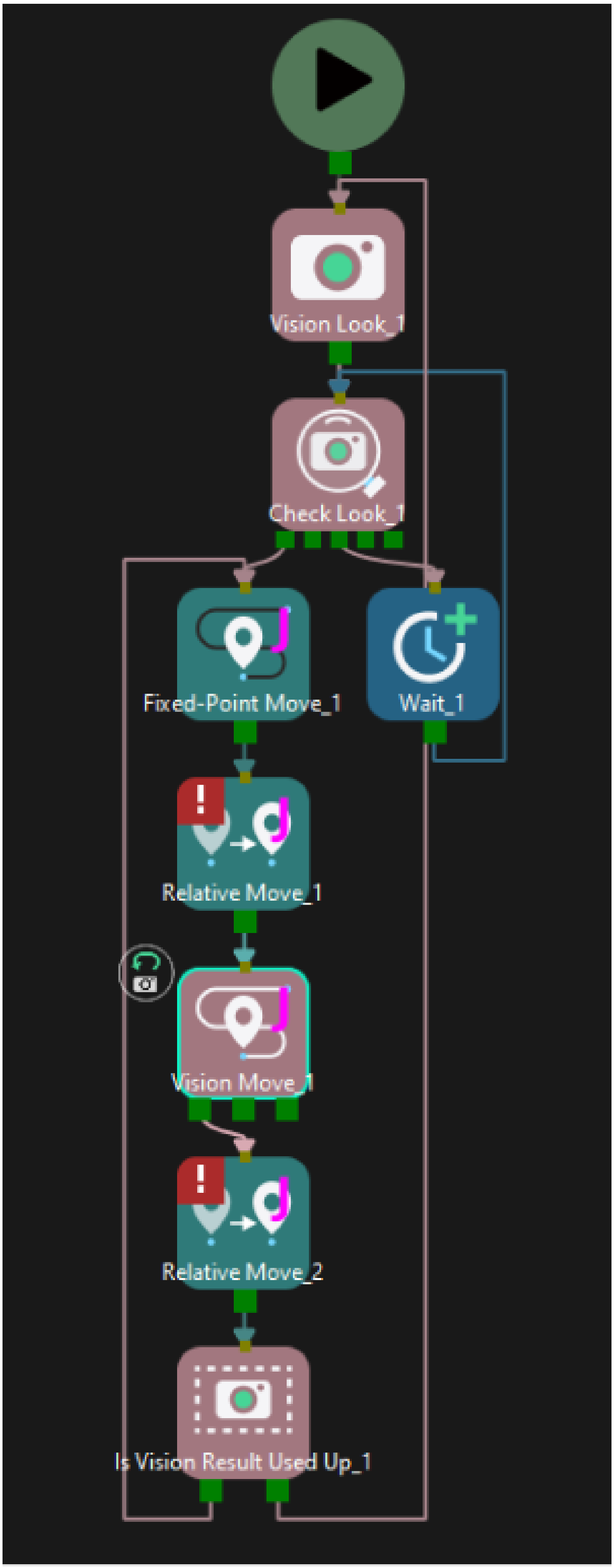
This parameter is usually used when the robot moves in a fixed path where no DO signals will be sent, such as the gluing application.

▼ *Reuse Vision Result*

When this feature is not enabled, each time “Vision Move” plans a robot picking path successfully, the rest unused vision results will be discarded. If this feature is enabled, the vision result that leads to a planning failure and the rest unused vision results will be retained and used for the next planning. This feature should be used with the “Is Vision Result Used Up” Step.

When there are multiple pickable objects in one vision result and **it can be guaranteed that picking one object will not affect the poses of other objects**, this option can be selected. The vision result will be reused and no new image will be captured until all vision results are used up.

Application Example


Vision Look_1
Check Look_1
Fixed-Point Move_1
Wait_1
Relative Move_1
Vision Move_1
Relative Move_2
Is Vision Result Used Up_1

▼ *Share Vision Result*

This feature enables multiple “Vision Move” Steps in which the same vision service is selected to share the vision result.

When one “Vision Move” Step plans successfully, the corresponding vision result will be used, and the unused vision result will be used by the next “Vision Move” Step with this feature enabled. After all “Vision Move” Steps that share the vision result complete planning, the rest results will be discarded.

In addition, this feature can be used in conjunction with the “Reuse Vision Result” feature. When “Reuse Vision Result” is selected, the rest vision results will not be discarded even after all “Vision Move” Steps have completed one round of planning.

Application Example

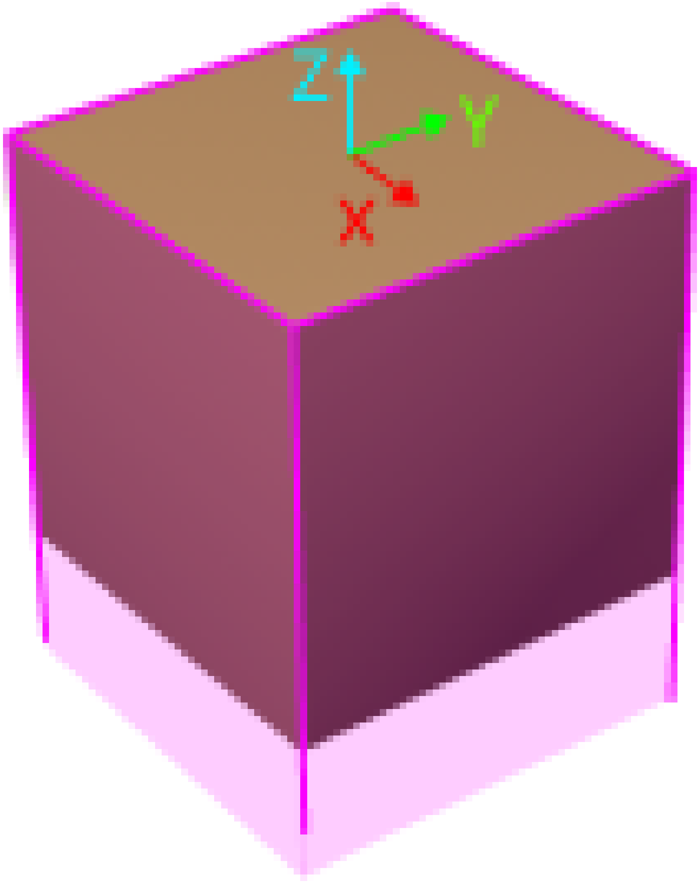
(To be supplemented)

Remove Point Cloud of Target Workobject

▼ *Remove Point Cloud of Target Workobject*

Select to remove the point cloud of the target workobject to be picked, and only collision between the robot end tool and non-target workobjects will be detected.

To avoid picking non-target boxes, the software will remove the point cloud of the target box and keep the point cloud of its adjacent boxes. During path planning, the software will discard a solution that may lead to a wrong picking due to the detected collision between the model of the vacuum gripper’s buffer and the point cloud of adjacent boxes.

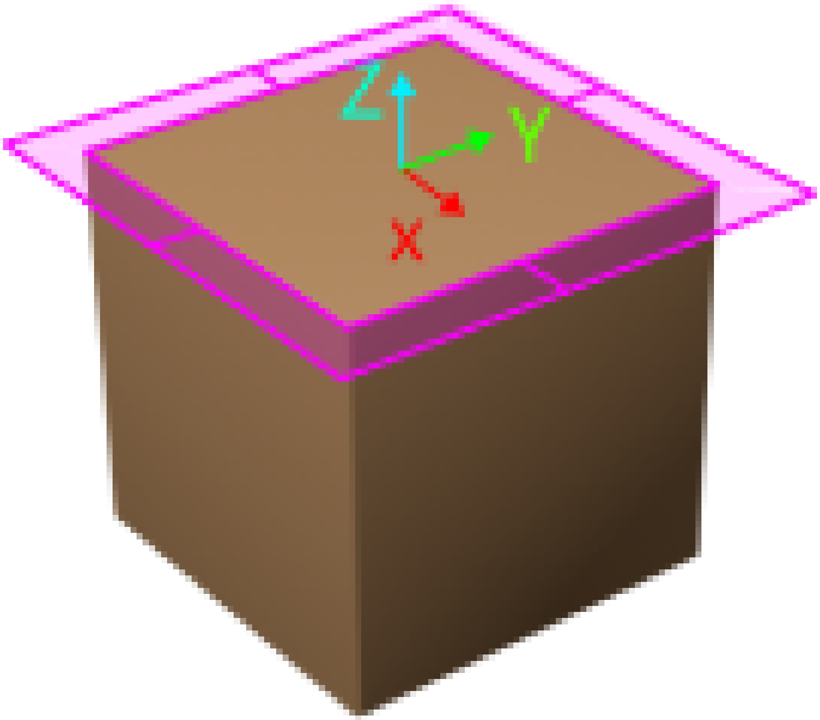


The default range for point cloud removal is a cuboid whose top surface is the top surface of the target box with an infinite height whose direction is the Z-direction of the box's vision pose.

You can adjust the "XY-Plane Point Cloud Removal Range Expansion" and "Z-Direction Point Cloud Removal Range Expansion" to expand the point cloud removal range for removing the point cloud near the target workobject.

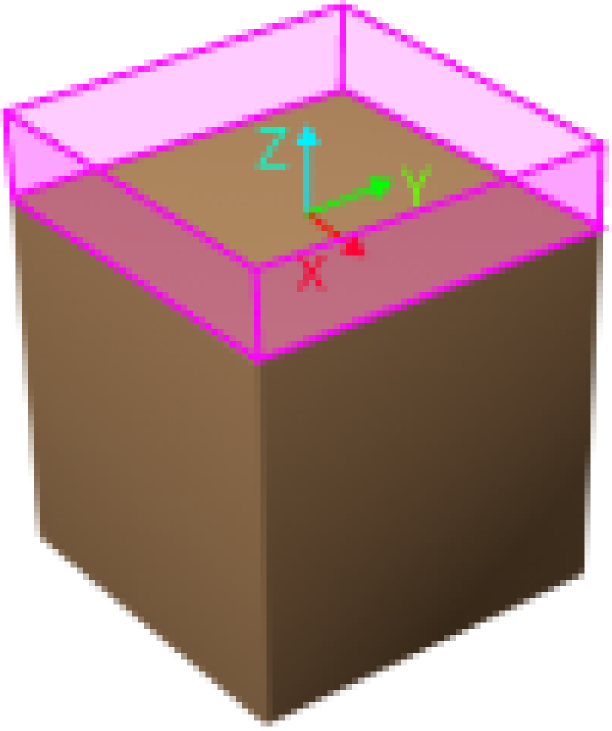
▼ *XY-Plane Point Cloud Removal Range Expansion*

Based on the original workobject dimensions, the XOY plane of the workobject pose is the top surface for the point cloud removal range. This parameter specifies the extended length from actual workobject edges on this surface.



▼ *Z-Direction Point Cloud Removal Range Expansion*

Based on the original workobject dimensions, the extended height in the positive Z-direction of the workobject pose for the point cloud removal range.



Supplementary Features

▼ *Update Bin Pose with Vision*

Once the “Update Bin Pose with Vision” parameter is selected, the bin will be recognized and located simultaneously while the camera captures the image for workobject recognition. Therefore, the bin pose in the simulation area can be updated dynamically, which facilitates the collision detection algorithm to effectively prevent the robot from colliding with the bin.

This feature will provide three fields, “scene_object_names,” “scene_object_sizes,” and “scene_object_poses,” in the vision result to define the name, dimensions, and position of the scene object to be updated.

▼ *Result Must Be in Specified Bin*

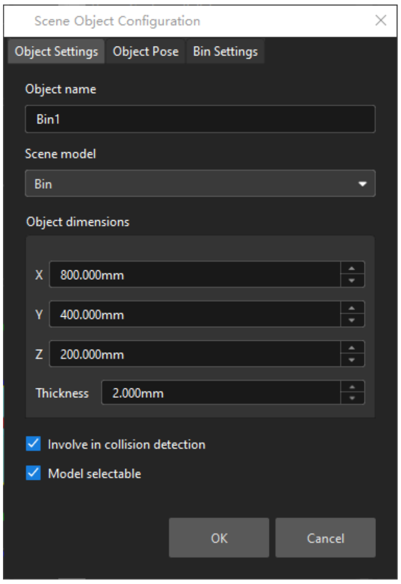
Result Must Be in Specified Bin

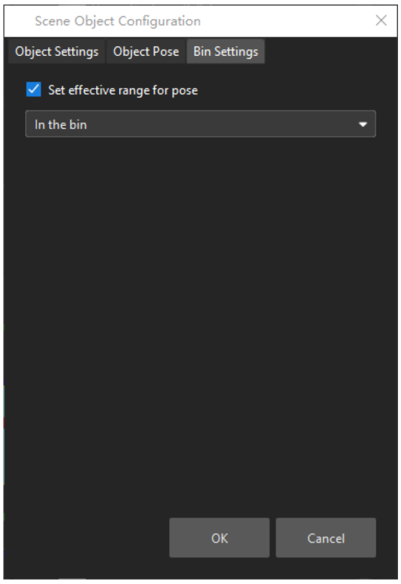
This feature limits that the vision result received by the current “Vision Move” Step must be in a specified bin. The vision result out of the effective bin range will not be used.

The effective range for each bin can be configured separately in the corresponding configuration window. Only bins with “Set effective range for pose” enabled can be selected in the drop-down menu.

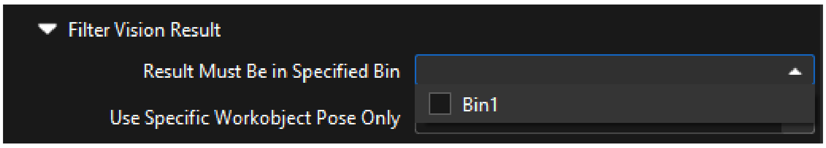
Application Example

Add “Bin1”, and select the “Set effective range for pose” parameter for this bin.


Scene Object Configuration
Object Settings | Object Pose | Bin Settings
Object name: Bin1
Scene model: Bin
Object dimensions
X 800.000mm
Y 400.000mm
Z 200.000mm
Thickness 2.000mm
☑ Involve in collision detection
☑ Model selectable


Scene Object Configuration
Object Settings | Object Pose | Bin Settings
☑ Set effective range for pose
In the bin

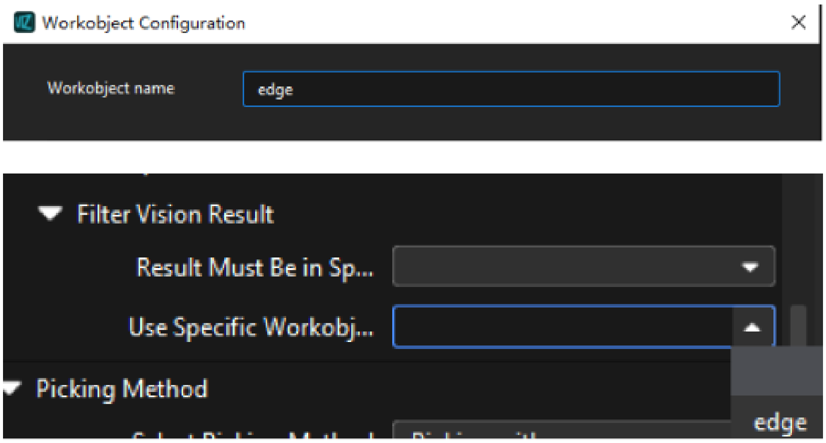
Select “Bin1” in the drop-down menu.


▼ Filter Vision Result
Result Must Be in Specified Bin
□ Bin1
Use Specific Workobject Pose Only

▼ *Use Specific Workobject Pose Only*

This parameter is used to filter the workobjects corresponding to the specified “labels” field in the vision results.

If you need to use this parameter, you need to add the corresponding workobjects in the resource tree whose name is the same as the “labels” field name, and then select the workobject in the dropdown menu on the right side of the parameter.



Workobject name: edge

Use Specific Workobj...: edge

▼ *Filter Out Unlikely Pick Point*

This parameter group is mainly used to avoid picking on the same pick point in scenarios where the object cannot be picked successfully.

Once the “Filter Out Unlikely Pick Point” parameter is selected, you can adjust the following parameters.

- Filter Subject
  - Pick point: only pick points with high failure rates will be deprioritized in the picking and discarded in some rounds.
  - Workobject: if any one pick point on the workobject is considered to have a high failure rate, all pick points on this workobject will be deprioritized in the picking and discarded in some rounds.
- Priority Downgrade Radius

  A sphere with this parameter value as its radius and the pick point successfully planned in the last round as its center will be introduced. If a pick point in the latest vision result falls within this sphere, the pick point will be put off for picking with a decreased picking priority.
- Discard Attempt Radius

  A sphere with this parameter value as its radius and the pick point successfully planned in the last round as its center will be introduced. If a pick point in the latest vision result falls within this sphere, the pick point will be discarded in this round of planning.

  For example, if the robot only moves the crankshaft but fails to pick it on the first attempt, there is a possibility of successful picking on the next attempt. Therefore, **Priority Downgrade Radius** can be set to downgrade the priority of the pick point, while the pick point will not be discarded. When the robot fails to move the crankshaft at all in the first attempt, it is highly unlikely that the crankshaft can be picked successfully in the next attempt, and therefore **Discard Attempt Radius** can be set to discard the pose directly.
- List Length

  A list of pick points with high failure rates. When the list length exceeds the upper limit, the pick point that is added to the list earliest will be removed from the list and can be used in

the next round of planning.

Picking Sequence

▼ *Sorting Strategy*

- Sort by the workobject count in the combination: The workobject combinations will be sorted by the number of workobjects they contain, and those with more workobjects will be picked first.
- In the strict order of vision result: Pick in the order of the result sent by the vision service. If the path planning fails by using a vision result, the current result will not be skipped and the path planning will fail.
- In the partial order of vision result: Pick in the order of the result sent by the vision service. If the path planning fails by using a vision result, the current result will be skipped. Then the next vision result will be used for the subsequent path planning.

Matching Condition

▼ *Array Gripper Matching Strategy*

You can choose between “Match position only” and “Match position and orientation.”

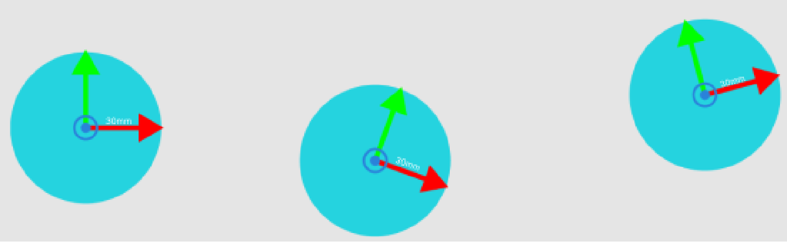
- Match position only: Only the positions of the workobjects will be taken into account for matching, while the angles between the TCP and workobject poses will not.
- Match position and orientation: Both the positions of the workobjects and the angles between the TCP and workobject poses will be taken into account for matching.

▼ *Distance Threshold*

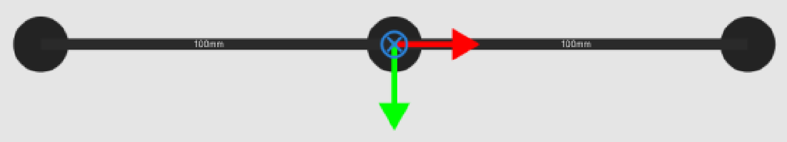
A circle with the “Distance Threshold” as its radius and the origin of workobject pose as its center will be introduced. If every end of the multi-end tool can fall within this circle, the matching is considered successful.

Application Example

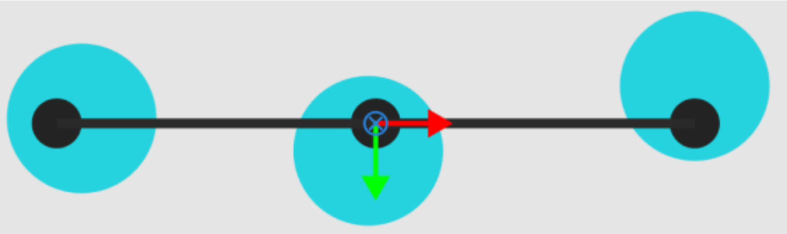
1. The three workobjects are shown below, and the Distance Threshold is set to 30 mm.



2. An array gripper with 3 end tools that are 100 mm away from each other is shown in the figure below.



3. In the path planning, the software will find a position where all three TCPs fit in the blue circles as shown below.

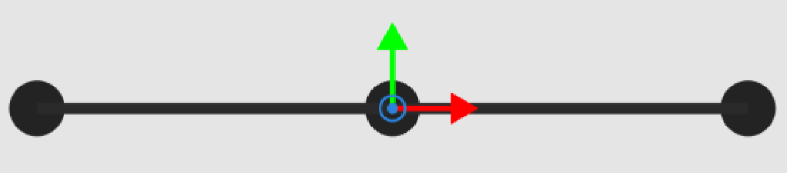


If three TCPs cannot be matched with the possible areas for picking at the same time, the software will try to match two TCPs instead.

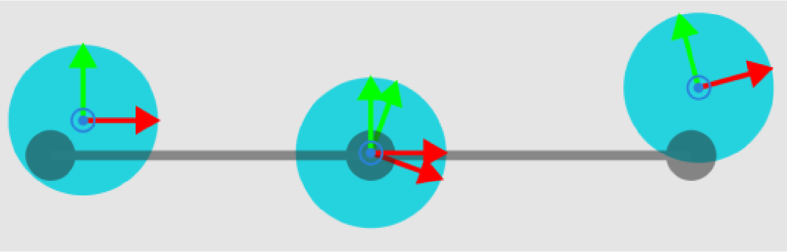
▼ *Angle Threshold*

If the angle between the X-axis of the TCP pose and the X-axis of each workobject pose is less than the “Angle Threshold,” the matching is considered successful.

1. The TCP will be rotated 180° around its own X-axis so that its Z-axis will be upward and parallel with that of the workobject pose.



2. Calculate the angle between the X-axis of the workobject pose and each axis of the TCP pose. If the angle is within the **Angle Threshold**, the matching is considered successful, otherwise it fails.



In the Array gripper mode, it is acceptable that the object in the middle of a combination is missed as long as the combination is not rotationally symmetric.

For example, the array gripper has 4 ends, and the ends are numbered 0, 1, 2, and 3. If there are 3 objects to be picked whose position can be represented by OOXO (X for empty space), the 0, 1, and 3 ends need to be started. After being rotated 180°, the combination changes from OOXO to OXOO, which is not symmetric.

Count of Picked Workobjects

▼ *Limit Total Count*

Select Limit Total Count and you can set the maximum Planned Count and other parameters.

▼ *Limit Count per Picking*

- No limit: There is no limit to the number of workobjects that can be picked at a time.
- Maximum value: The maximum workobjects that can be picked at a time.
- Specified value: Set a fixed limit on the number of workobjects to be picked at a time.

5.5.10. Others

5.5.10.1. Stop Workflow

Function

In complex usage scenarios where one Procedure may nest another Procedure, you can use this Step to stop the entire project when a certain Step in the Procedure does not output a result.

Parameter Description

None.

Exit Reason

You can record the reason why you want to end the project here, which facilitates to make the project logic clear.

5.5.11. General Parameters

5.5.11.1. Move-Type Step Common Parameters

▼ *Send Waypoint*

Selected by default to send waypoint poses to the receiver, such as the robot. When this option is unselected, the waypoint pose will not be sent. However, the waypoint will remain in the planned path.

▼ *Try Continuously Running through Succeeding Non-Moves*

Unselected by default. When **non-move Steps**, such as Vision Look, Set DO, Check DI, etc., are connected between **move-type Steps**, the robot's path planning will be interrupted, and the actual robot will take a short pause, reducing the smoothness of running.

When this option is selected, the project will continue to run without waiting for the current move-type Step to complete execution, and therefore the robot can move in a smooth way without pauses. However, enabling this option may cause the execution of the Step to end prematurely.

Why will this option cause the execution of the Step to end prematurely?

Mech-Viz will send multiple poses simultaneously to the robot when the project is running. When the currently returned JPs of the robot correspond to the last pose sent by Mech-Viz, Mech-Viz will assume that the robot has moved to the last position.

For example, there are 10 move-type Steps in a path, and the pose of the 5th move-type Step is the same as that of the last move-type Step. When the robot moves at low speed, it sends JPs to Mech-Viz when it moves to the 5th move-type Step, Mech-Viz may mistakenly determine that the robot has finished the move-type Steps and prematurely ends the Steps since the poses of the 5th move-type Step and the last move-type Step are the same in the path.

▼ *Do Not Check Collision with Placed Workobject*

Unselected by default, namely that the collision with the already placed objects will not be detected. When this option is selected, the collisions between the robot, end tool, and placed objects will be detected.

In palletizing scenarios, the two possible cases of error are as follows:

1. When the robot is placing a carton, the robot may come into light contact with the placed cartons while no deformation will be caused. After Mech-Viz detects this collision in simulation, it will plan other positions for placing the carton, and therefore a full stack cannot be formed.
2. Usually, the TCP of a suction cup is inside the suction cup model instead of on the surface of it. Under this circumstance, the suction cup may be embedded in the model of the picked carton in the simulation of picking, while the software does not detect the collision between the end tool and the picked object. After the robot places the object and the carton model turns into an object model in the scene, a collision between the suction cup and the carton will be detected and the palletizing cannot be completed.

When this option is selected, no collision between the robot, end tool, and the placed object will be detected, and the above two cases of errors can be avoided.

▼ *Point Cloud Collision Detection Mode*

Select the proper mode according to the requirement of the on-site situation. Usually, the default setting **Auto** can be used. **Do not check collision** mode can be used in move-type Steps before the robot picks the object, and **Check collision** mode can be used after the robot picks the object.

Auto	Default setting. Collision with point cloud is checked only for the “Vision Move” Step and the “Relative Move” Step that depends on the “Vision Move” Step, but not for all move-type Steps.
Do not check collision	Point cloud collisions for all move-type Steps will not be detected.
Check collision	Point cloud collisions for all move-type Steps will be detected.

> ❗ When Collisions › Collision detection configuration › Detect collision between point cloud and others is switched on, Mech-Viz will detect collisions between the robot model, end tool model, and point cloud when planning the path. Normally, detects whether the robot collides with other objects during picking and placing. When there are point cloud outliers, non-exiting collisions will be detected, which leads to errors in path planning.

▼ *Ignore Workobject Symmetry*

This parameter will only take effect when **Waypoint type** of the Step is set to **Workobject pose**.

None	Default setting, i.e., do not disable symmetry on any axis.
Around workobject frame Z axis	Only disable symmetry on Z-axis of the workobject reference frame.
Around workobject frame X&Y axis	Disable symmetry on X-axis and Y-axis of the workobject reference frame.
Around all axes	Once the object symmetry is disabled, the robot will place the objects strictly according to the workobject poses.

> ℹ In some special cases, objects are not pickable due to their peculiar poses. Setting **Rotational symmetry** under Workobjects › Workobject Configuration in Resources may solve this problem. Based on the settings of Rotational symmetry, when the default object pose is not pickable, Mech-Viz will calculate candidate poses and determine whether the candidate poses are pickable. As the candidate poses calculated based on the settings of Rotational symmetry are different from the original poses output from Mech-Vision, the

consistency of the objects' place poses cannot be guaranteed.

▼ *Plan Failure Out Port*

After selecting this parameter, this Step will add “Plan failure” out port.

During the planning process, planning is carried out along the branch after the “Success” out port; if the planning fails in the current Step, the branch process after the “Planning failure” out port will be executed.

5.5.11.2. Held Workobject Collision Detection Settings

▼ *Do Not Check Collision with Scene Object/Robot*

Unselected by default. Once this option is selected, the collisions between the **held workobject** with the **scene objects or robot** will not be detected, and therefore the calculation workload of collision detection will be reduced, the planning speed can be increased, and the cycle time can be shortened. It is usually enabled in the first one or two move-type Steps after the robot picks the object.

Please enable this option cautiously as there may be collision risks.

When **Detect collision between held workobject and others** under Collision detection configuration › Configuration on held workobject is enabled, the software will detect whether the model of the held object collides with the models of the scene objects and the robot.

In palletizing projects, the calculated carton dimensions have millimeter-level errors with the actual dimensions, and frictions between cartons may occur during picking but no collisions will occur. For some move-type Steps that will obviously not cause collisions, detecting such collisions only adds to the calculation workload and planning time, and consequently extending the cycle time. In palletizing projects, enabling Do Not Check Collision with Scene Objects does not affect the collision detection between the held carton and the placed cartons. This option can be enabled when there are scene objects under the stack to avoid failure of finding the palletizing solution.

▼ *Do Not Check Collision with Point Cloud*

Unselected by default. Once this option is selected, the collisions between the **held workobject** with the **point clouds in the scene** will not be detected, and therefore the calculation workload of collision detection will be reduced, the planning speed can be increased, and the cycle time can be shortened.

- When both **Detect collision between held workobject and others** under Collision detection configuration › Configuration on held workobject and **Configuration on point cloud ¬ Detect collision between point cloud and others** are enabled, the software will detect whether the model of the held workobject will collide with the point cloud in the scene.
- When Mech-Vision sends the point cloud and object model to Mech-Viz, the point cloud and the object model are fitted together. After the robot picks the object, the model moves along the planned path, and the collision between the model of the held workobject and the point cloud will occur.
- It is known that the model of the held workobject will have false collisions with the point cloud. Detecting such collisions unnecessarily adds to the calculation workload and extends the planning time.

5.5.11.3. General Parameters of Non-Move Steps

▼ *Skip Execution*

None Default setting. Do not skip the current Step.

Simulation only Skip the current Step during simulation. The exit port is specified by **Out Port When Skip**.

Always Skip the current Step when the project is simulating or running. The exit port is specified by **Out Port When Skip**.

Instruction When **Simulation only** or **Always** is selected, the current Step will be skipped and the subsequent Step will be executed when running the project. If this parameter is set to None in “Check DI” Step in the project, and there is no external input signal during simulation, the project will be stopped when executing to this Step. In this case, setting the parameter to Simulation only or Always enables the simulation to continue.

▼ *Out Port When Skip*

This parameter will take effect when **Skip Execution** is set to **Simulation only** or **Always**. It specifies the exit port when skipping a Step.

5.5.11.4. Basic Move Settings

Basic move settings of move-type Steps are used to set the **velocity** and **motion type** for the robot when it moves to the waypoint.

▼ *Pick or place*

Unspecified Default value

Pick Please select this option when the move-type Step is before “Vision Move”.

Place Please select this option when the move-type Step is after “Vision Move”.

Setting this parameter facilitates checking the logic of the Mech-Viz project. Please select a proper option according to the on-site situation and ensure that the Step used to pick is placed before the Step used to place.

▼ *Motion type*

Joint move Joint motion, which guides the robot to move in a curved path. It is less likely to reach singularities in the path for joint motion. This motion type is applicable to scenarios where the requirement of path accuracy is not strict and the robot moves in a large space.

Linear move Linear motion, which guides the robot to move linearly. This motion type is applicable to scenarios where there is a strict requirement for path accuracy, such as welding, gluing, and certain types of picking.

▼ *Velocity & Acceleration*

Velocity and acceleration determine how fast the robot can move. Usually, the set acceleration should be lower than the velocity. When the set acceleration is higher than the velocity, the robot will move in a choppy way.

The velocities of Vision Move and its prior and subsequent Steps should be relatively low to ensure that the objects can be picked steadily.

▼ *Blend radius*

Default value 50.00 mm

Notes Usually, the default setting can be used.

The blend radius refers to the distance between the target point and the point where the robot starts to turn. The larger the blend radius, the more smoother the robot motion transitions are. If the robot moves in a relatively small space, please set the blend radius to a smaller value.

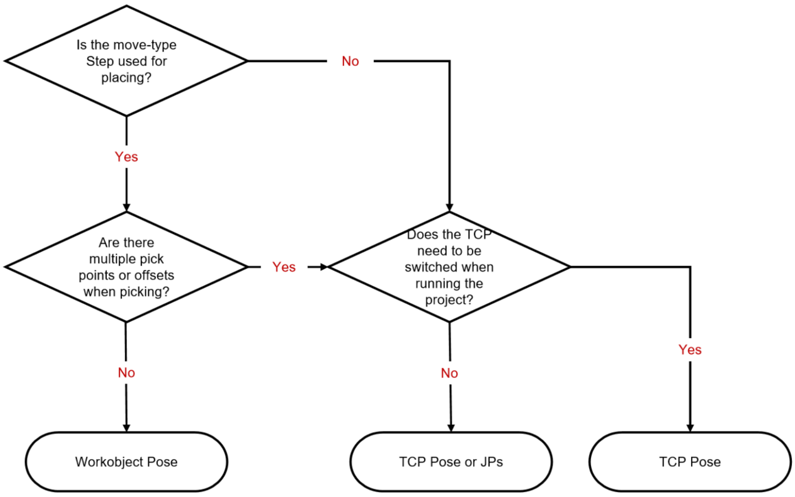
If the robot moves in a relatively large space without obstacles and the distance between two consecutive path segments is long, please set the blend radius to a larger value.

5.5.11.5. Waypoint Type

▼ *Waypoint Type Description*

Tool	The waypoint will be represented by the X, Y, Z values, Euler angles, quaternions or rotation vectors in the tool reference frame. The tool pose is the pose of the tool, which is described as a relative pose of the tool reference frame to the robot reference frame. This description is in line with the task space, focusing on describing the position and orientation of the tool (such as truss arms, welding torch) when the robot is performing tasks.
JPs	The waypoint will be represented by the joint positions of the robot. Joint positions describe the angle or position of each joint of the robot. This description can represent the state of each joint in line with the hardware structure of the robot.
Workobject pose	The waypoint will be represented by the X, Y, Z values, Euler angles, quaternions or rotation vectors in the object reference frame. It focuses on describing the position and orientation of the actual workobjects that need to be operated or handled when the robot is performing tasks. Workobject pose is the pose of a certain point on the workobject relative to the robot reference frame.

The method to determine the waypoint type is shown in the figure below.


Is the move-type Step used for placing?
No
Yes
Are there multiple pick points or offsets when picking?
Yes
Does the TCP need to be switched when running the project?
No
No
Yes
Workobject Pose
TCP Pose or JPs
TCP Pose

▼ *Parameter Description*

Edit pose	Enter quaternions or Euler angles directly to adjust the pose. You can also copy and paste the pose.
Transform pose	Transform the current pose to a new one. It is suitable for fine-tuning.
Rectify pose	Set the coordinate of P1, P2, and P3 according to the instruction, and then rectify the object pose with the three-point method. It is suitable for scenarios where the objects are tilted and their poses cannot be easily determined. For instance, when a cuboid tilts, its pose is hard to determine. Rectify pose can be used here to calculate the pose of the cuboid and therefore the robot can move to the rectified pose.
Edit JPs	Similar to Edit Pose above; edit the JPs by copying and pasting or editing the JPs in either radians or degrees, which can be switched based on the actual requirement.
→ Get	Click to move the robot back to the specified waypoint.
→ Set	Get the pose of the simulated robot and set it as the waypoint.
Show	Show all the JPs solutions to the current waypoint.

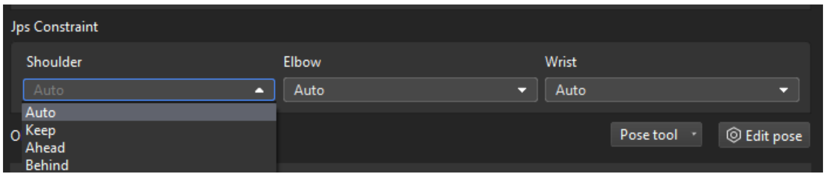
5.5.11.6. JPs Constraint

▼ *Definition of Terms*

Shoulder The relative positional relationship between the center of the wrist and Axis1. Axis1 refers to the rotation center axis of the robot axis 1.

Elbow Relative positional relationship between the wrist and lower arm, which refers to the line connecting the rotation centers of the robot axis 2 and axis 3.

Wrist Wrist refers to axis 5 of the robot. The negative wrist JPs suggests that the robot’s wrist can rotate in the negative direction.



▼ *Value list*

Auto
: There is no constraints on the motion of this joint. The optimal solution is the one in which each axis rotates the least.

Keep
: Constrain the next JPs solution according to the current JPs status. For example, if the current JP of the axis 3 is positive, only solutions in which the JP of axis 3 is positive will be considered for the next waypoint.

Ahead
: The wrist center is ahead of axis 1.

Behind
: The wrist center is behind axis 1.

After clicking Show, a window showing all the JPs solutions to the current pose will pop up. After clicking one of the solutions, the corresponding pose of the solution will be displayed in the 3D simulation area. Therefore, you will know the possible solutions under different constraints, as shown below.

1. JPs constraint settings only apply to 6-axis robots. It is considered that 4-axis robots do not have abnormal shoulder, elbow, or wrist positions.
2. Relative Move, Custom Pallet Pattern, and Predefined Pallet Pattern do not support JPs constraint settings. There are default JPs constraint settings for these Steps, that is, the statuses of the shoulder, elbow, and wrist will not change, and the robot’s motion does not cross solution systems (does not go through singularities).

5.6. Built-in Tools

5.6.1. Model Editor

As a built-in tool of Mech-Viz, Model Editor is mainly used to adjust the reference frame of the end tool model and create convex hulls for the end tool model.

The collision models of end tools used in Mech-Viz should be **OBJ models that are composed entirely of convex hulls (convex polyhedra)**. If the end tool model you use is in STL, STP, STEP, or OBJ format and does not meet this requirement, you should use Model Editor to **create convex hulls** on it and **export it as an OBJ model**.

> A convex hull refers to the smallest convex shape that completely encloses a set of points in a multi-dimensional space. In simpler terms, a convex hull is a convex polygon or polyhedron that wraps around a set of points or a collection of objects in such a way that no part of the polygon or polyhedron “bends inward.”

This section is intended to help you get familiar with and use the Model Editor. Please click the link below to get started with the Model Editor.

- Model Editor Application Example
- Model Editor User Interface

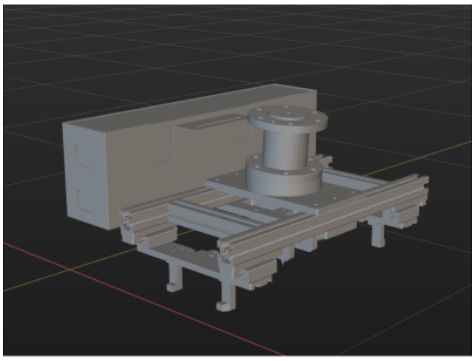
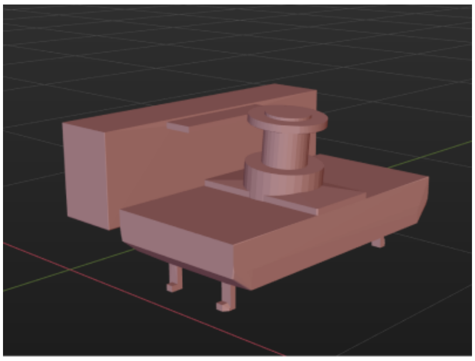
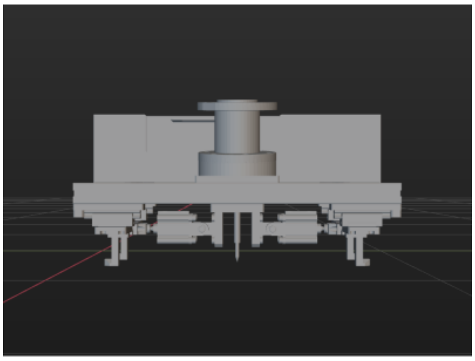
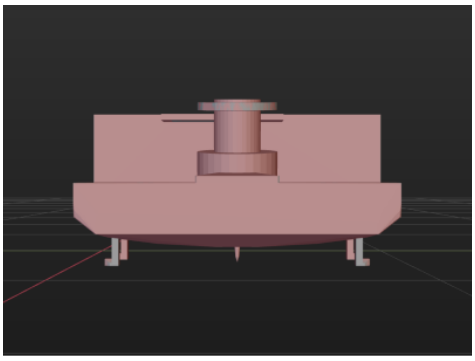
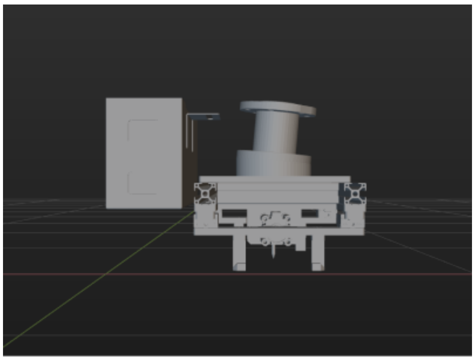
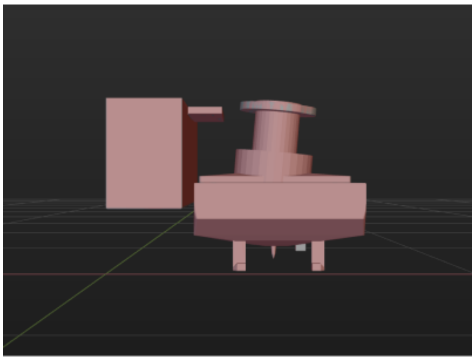
- Use Model Editor to Create Convex Hulls

5.6.1.1. Application Examples

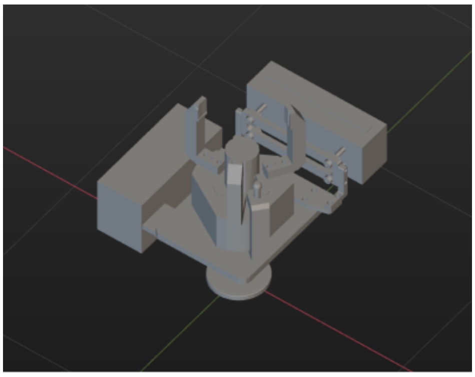
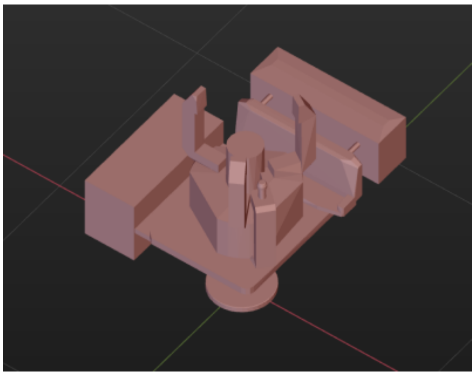
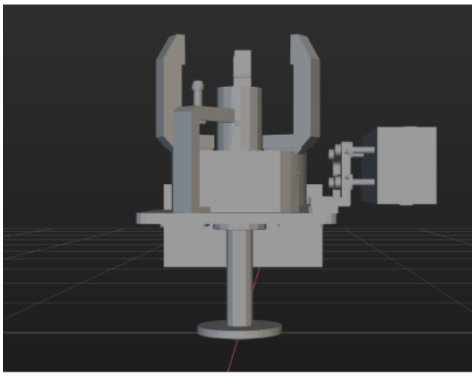
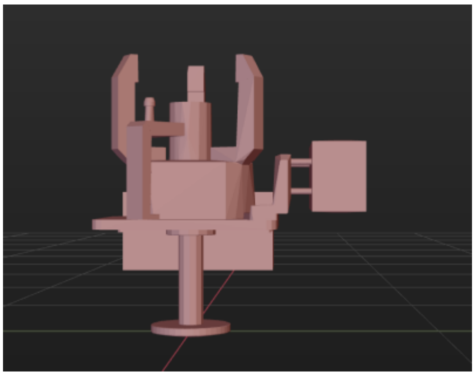
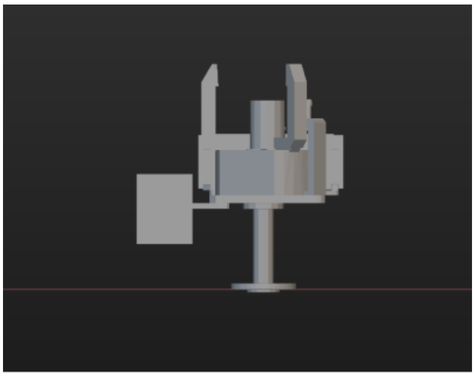
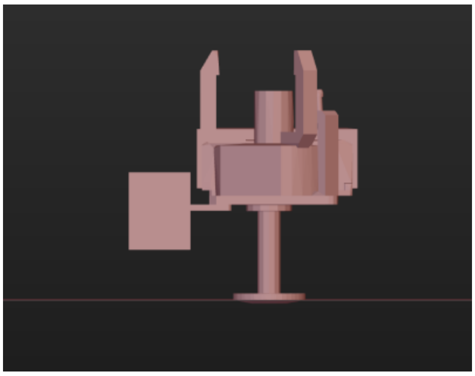
To save time when creating a collision model for the end tool, it's not always necessary for the convex hulls you create to replicate every detail of the original model. You can omit certain details based on the specific requirements of the model.

Please refer to the examples below, which illustrate the end tool models before and after they have been enclosed by convex hulls.

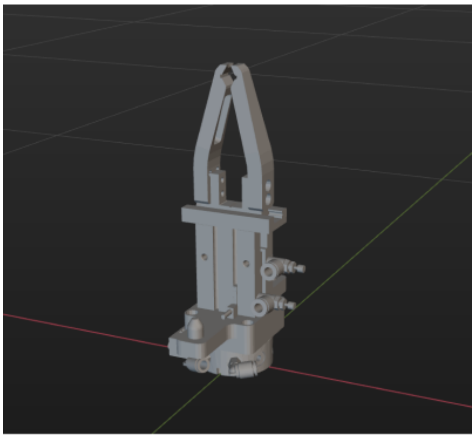
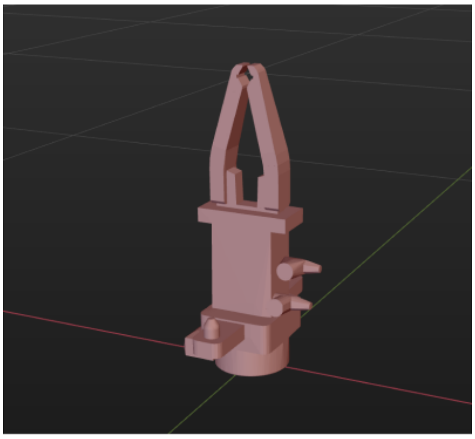
- Example 1:

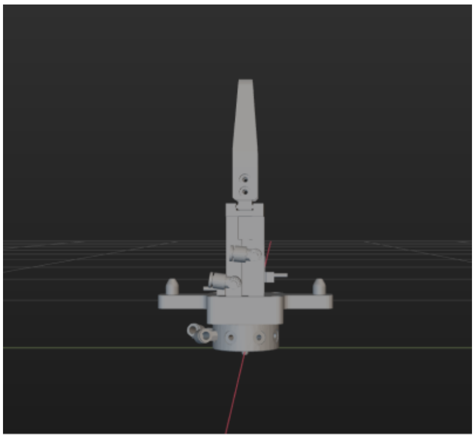
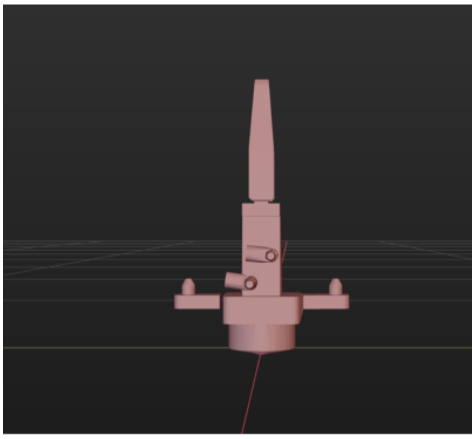
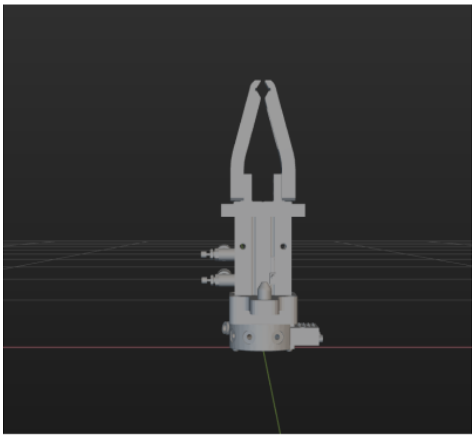
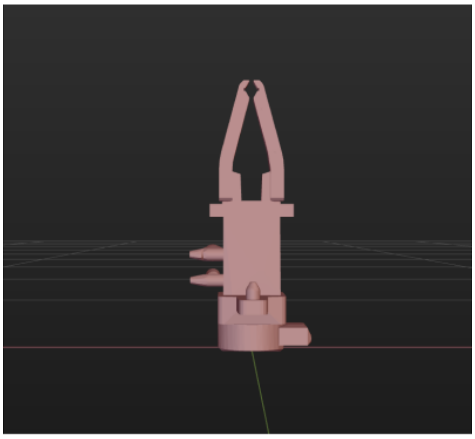
View	Before	After
Custom view		
Front view		
Side view		

- Example 2:

View	Before	After
Custom view		
Front view		
Side view		

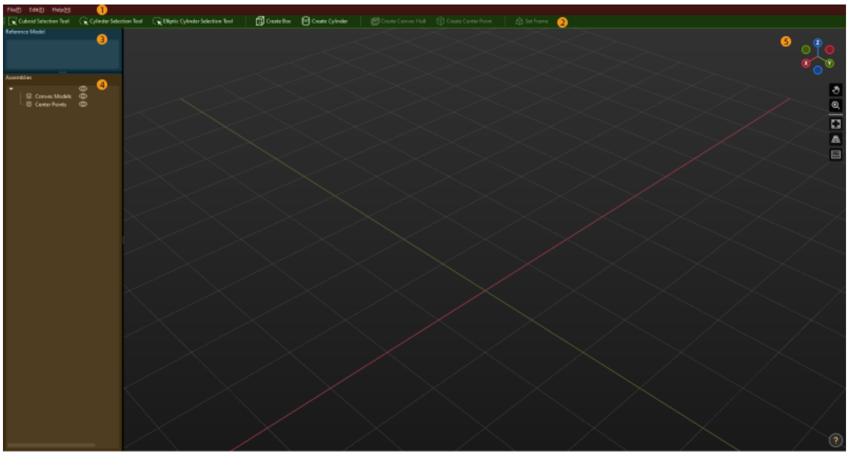
- Example 3:

View	Before	After
Custom view		

View	Before	After
Front view		
Side view		

5.6.1.2. Interface Introduction

The interface of Model Editor consists of the following 5 main parts.



① Menu Bar, ② Toolbar, ③ Reference Model, ④ Assemblies ⑤ 3D Editing Workspace

Menu Bar

Provides functions as importing and exporting the model file, editing, and opening the user guide.

File

Option	Description	Shortcut
New	Create a new editing project.	Ctrl + N
Open	Open an existing M3D file to edit.	Ctrl + O
Save	Save the editing project as an M3D file.	Ctrl + S
Save As	Save the editing project as an M3D file to the specified directory.	Ctrl + Shift + S
Import	Reference Model: import the reference model.	N/A
Export	Reference Model: export the reference model; Assemblies: export the assembly.	N/A

Edit

Option	Description	Shortcut
Undo	Undo the change.	Ctrl + Z
Redo	Redo the change.	Ctrl + Y

Help

Option	Description
User Manual	Open the online user guide of the Model Editor.

Toolbar

Provides basic tools for the Model Editor.

Option	Description	Shortcut
Cuboid Selection Tool	Creates a cuboid selection.	N/A
Elliptic Cylinder Selection Tool	Creates an elliptic cylinder selection.	N/A
Cylinder Selection Tool	Creates a cylinder selection.	N/A
Create Box	Creates a cuboid convex hull.	N/A
Create Cylinder	Creates a cylinder convex hull.	N/A
Create Convex Hull	Creates a convex hull based on the vertices in the cuboid selection.	Shift + C
Create Center Point	Creates a center point based on the vertices in the cuboid selection.	N/A

Option	Description	Shortcut
Create the Reference Frame	Creates a reference frame based on the center point or the vertices.	N/A

Reference Model

The names of the imported files will be displayed in the Reference Model panel.

- Click the model file name, and the vertices of the model will be displayed in the 3D editing workspace.
- Once you select the model file name, click 👁 will hide the model only and the vertices of the model will still be displayed.
- If the model file name is not selected, click 👁 will hide the entire model.

Assemblies

The geometric solids, convex hulls, and center point will be displayed in the Assemblies panel. The geometric solid with convex hulls form a convex model.

Click 👁 to hide the corresponding convex hulls or center points.

Right-click the name of the geometric solid and you can see the following options.

Option	Description	Shortcut
Copy	Copy the selected geometric solid.	Ctrl + C
Paste	Paste the geometric solid.	Ctrl + V
Delete	Delete the selected geometric solid.	Delete
Rename	Rename the selected geometric solid.	N/A

Right-click the name of the convex hull and you can see the following options.

Option	Description	Shortcut
Copy	Copy the selected convex hull.	Ctrl + C
Paste	Paste the convex hull.	Ctrl + V
Delete	Delete the selected convex hull.	Delete
Rename	Rename the selected convex hull.	N/A
Show 3D Selection Tool At Creation	Display the 3D selection tool which is used to create the convex hull.	N/A

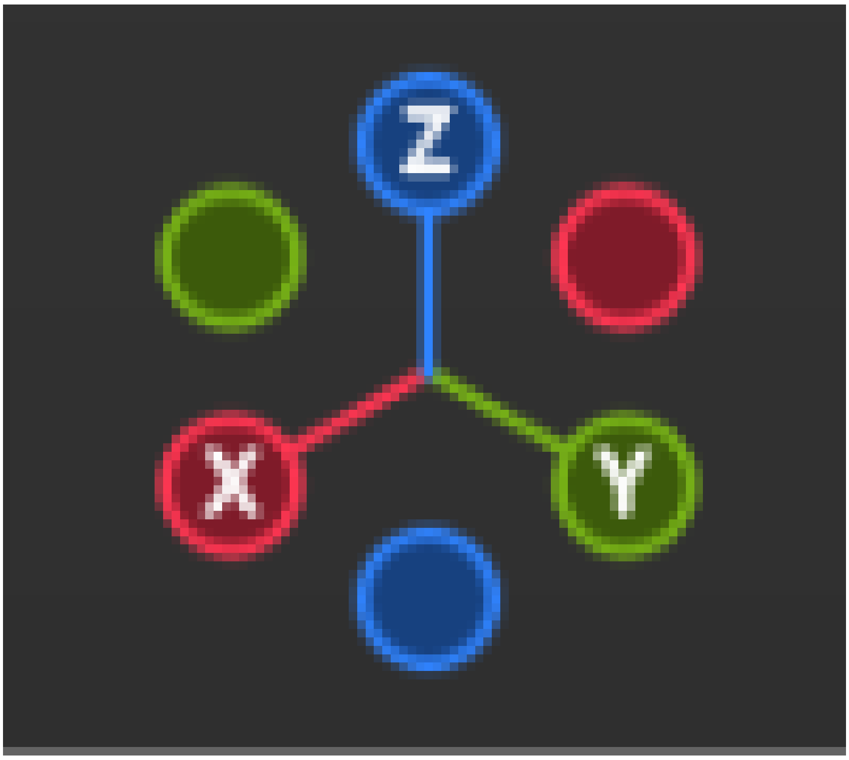
Right-click the name of the center point and you can see the following options in the context menu.

Option	Description	Shortcut
Copy	Copy the selected center point.	Ctrl + C
Paste	Paste the center point.	Ctrl + V
Delete	Delete the selected center point.	Delete
Rename	Rename the selected center point.	N/A
Show 3D Selection Tool At Creation	Display the 3D selection tool which is used to create the center point.	N/A
Show On Top	Select to view the center point inside the model.	N/A

3D Editing Workspace

The 3D Editing Workspace is used to display and edit the model.

The navigation gizmo shows the current view in the workspace. Click the icon of each axis (X, Y, Z, -X, -Y, -Z) to align the view with the axis. Press and hold the left mouse button and drag the navigation gizmo to rotate the view.

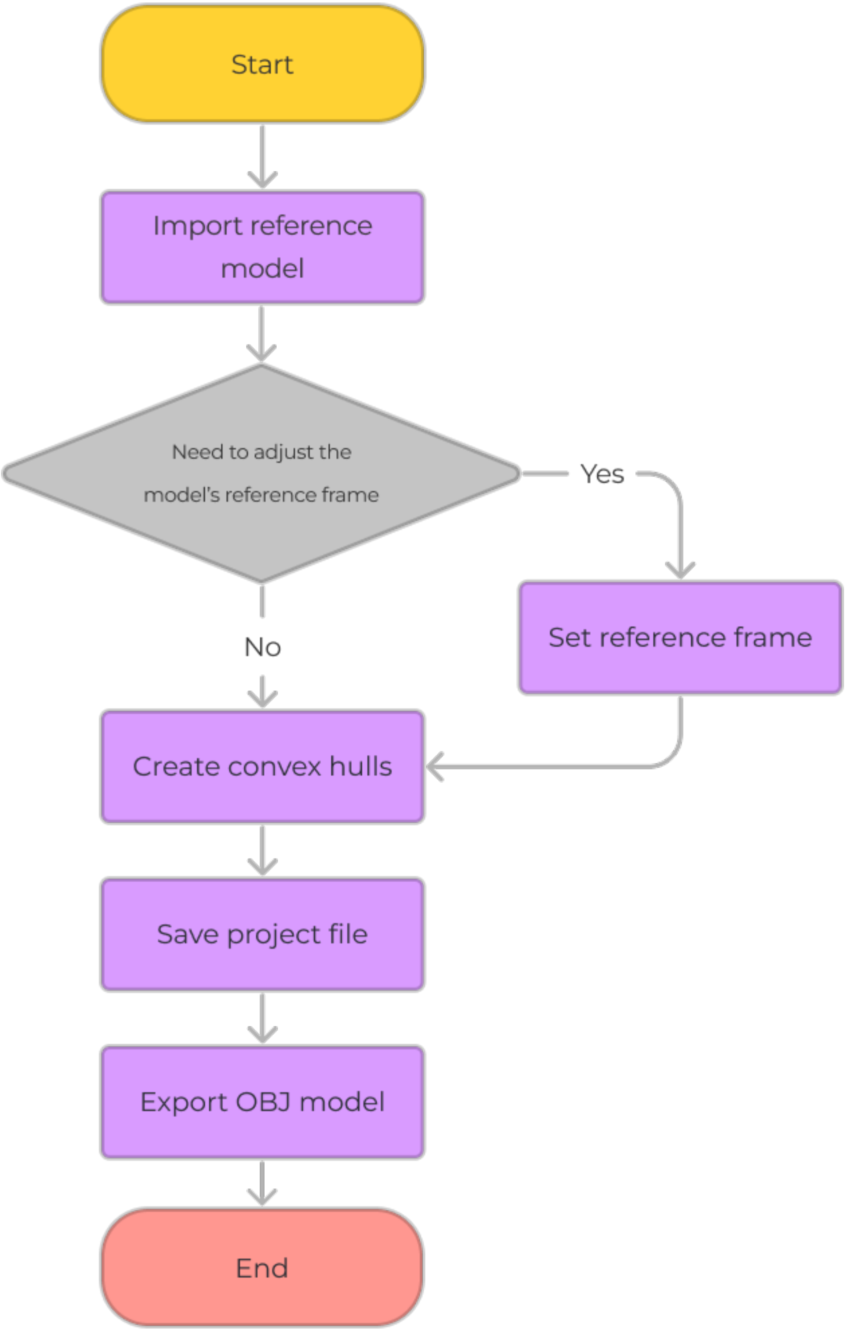


Rotate the view	Press and hold the **scroll wheel** and drag in any direction, or press and hold the **left mouse button** and drag the navigation gizmo in any direction.
Pan the view	Press and hold ✋ and drag in any direction, or press and hold **Shift + scroll wheel** and drag in any direction.

Zoom in/out	Press and hold 🔍 to zoom in by dragging upwards and zoom out by dragging downwards, or **scroll the mouse wheel**.
Fit the view	Click ⛶.
Switch to perspective/orthographic view	Click ▦.
Show pressed keyboard keys and mouse buttons	Once ☰ is selected (highlighted in blue), the keyboard keys and mouse buttons you pressed will be displayed in the lower right corner in the 3D Editing Workspace.

5.6.1.3. Use Model Editor to Create OBJ Models

The steps to convert STL models, STP models, STEP models, and ineligible OBJ models to OBJ models that are composed entirely of convex hulls are as follows.



1. Import Reference Model

Supported file formats of imported reference models include STL, STP, STEP, and OBJ.

Follow either of these steps to import the reference model.

- Click File › Import › Reference Model and select the reference model file in the file selection window.
- Drag and drop the reference model file to the interface of the Model Editor.

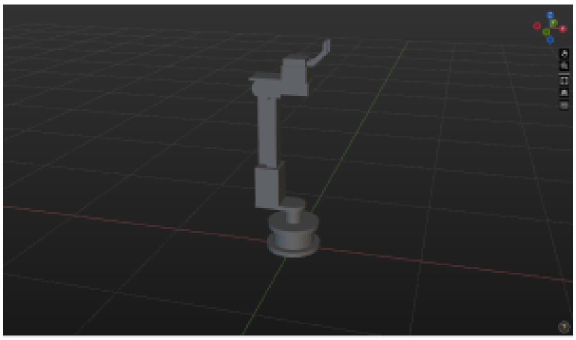
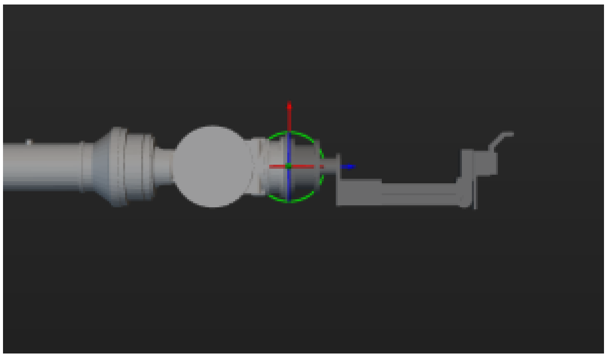
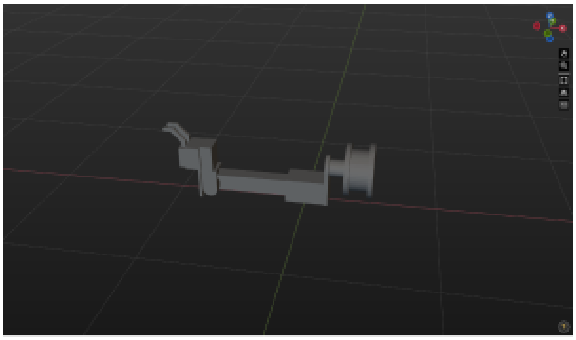
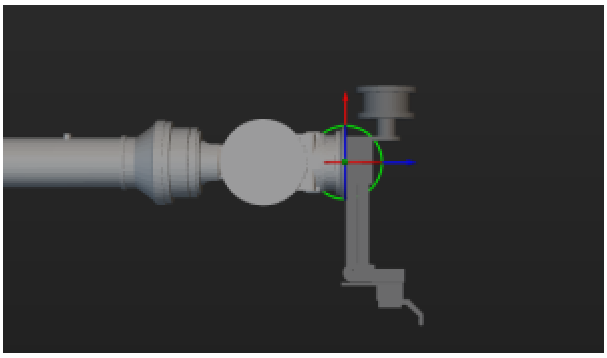
Select the unit according to the actual dimensions of the model in the pop-up window, and click [OK].

If an error message “Failed to load the model” pops up, please refer to Determine Whether the

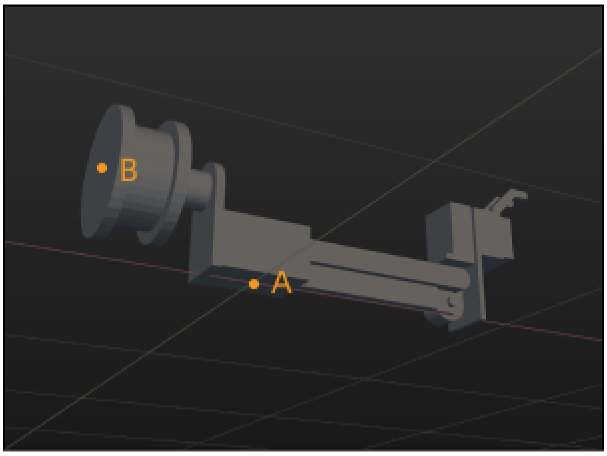
STEP/STP File Is Eligible to check the model file.

2. Set Frame (Optional)

The end tool models with different poses in the Model Editor are illustrated in the table below, including the default pose that is assigned when the tool is initially added via the Tool Configuration window.

Pose of end tool model in the Model Editor	Default tool pose when added via Tool Configuration
	
	

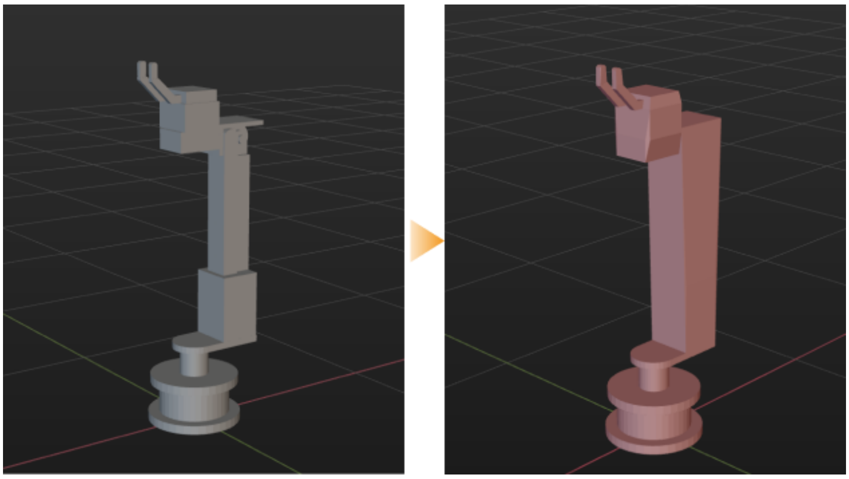
From the above table, we can see that if the “reference frame origin of the model’s installation position” does not coincide with the “reference frame origin of the Model Editor’s world reference frame,” you should adjust the position or size of the end tool model when adding the end tool. To avoid the need for adjusting the model’s position every time it is used, please refer to the content in Adjust the Reference Frame of the Model to adjust the reference frame of the model.



A: World frame origin in the Model Editor; B: Reference frame origin of the model’s installation position

3. Create Convex Hulls

The collision models of tools used in Mech-Viz should be OBJ models that are composed entirely of convex hulls, and a convex hull should be created based on the reference model. The final model should be composed entirely of convex hulls and reflect the shape of the original reference model.

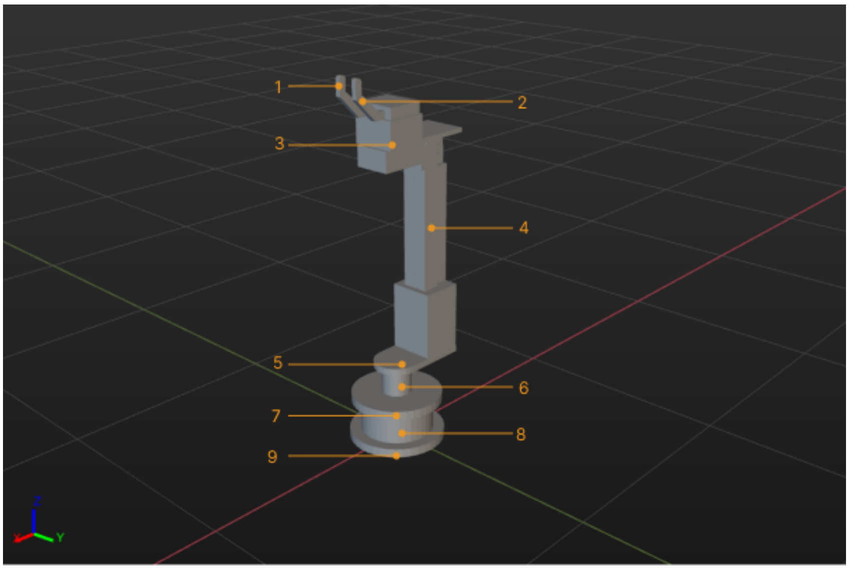


There are two ways to create convex hulls, please select the one suitable for the actual model.

- Use the selection tool to select vertices and create a convex hull based on the vertices.
- Create a box or cylinder directly.

Before proceeding to read the following content, please read Selection Tool Usage Guide and Create a Box or Cylinder to learn about how to use the relative tools.

Do not use the 3D selection tool to cover the whole model to create a convex hull. It is recommended to segment the model into multiple parts according to its structure, and use the 3D selection tool to select each part and create convex hulls separately. For example, the model in the figure below can be segmented into 9 parts.



The steps are as follows:

1. Select the model name in the “Reference Model” panel, and click Cuboid Selection Tool, Cylinder Selection Tool, or Elliptic Cylinder Selection Tool.
2. Adjust the selection tool to cover the vertices on the part to be processed, and click **[Create Convex Hull]**.
3. Repeat step 2 until all 9 parts of the model are covered by convex hulls.
4. Click 👁 on the left of the reference model name to hide the reference model and view the convex model.

- Since the convex hull or center point are generated based on the vertices of the model, please ensure that the vertices on the part to be processed are within the selection tool.
- For parts that cannot be accurately selected by a selection tool to create convex hulls, you can create a box or cylinder.

4. Save the Project File

To facilitate re-editing of the model, you can follow these steps to save the model to an M3D file.

1. Click File › Save.
2. Specify a save path and enter a file name.
3. Click **[Save]**.

5. Export the Model after Editing

Follow these steps to save the edited model to an OBJ file.

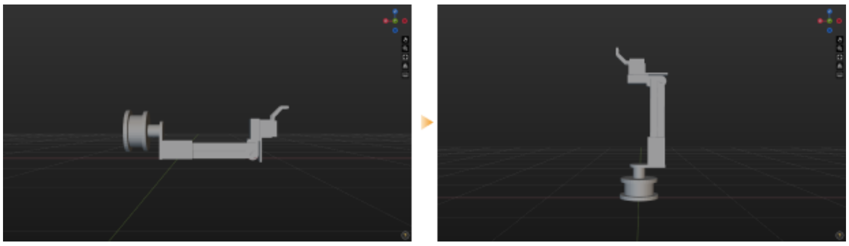
1. Click File › Export › Reference Model.
2. Specify a save path and enter a file name.
3. Save as type obj(*.obj).
4. Click **[Save]**.

Only the models set to visible in the Assemblies panel in the Model Editor will be saved, while the hidden ones will not be exported.

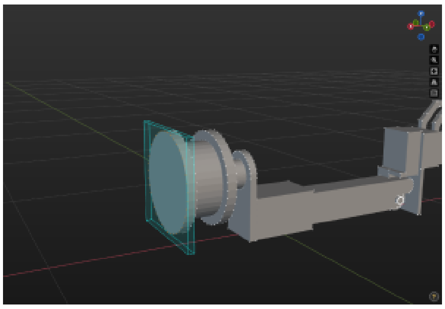
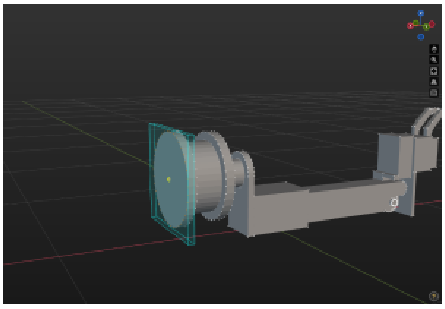
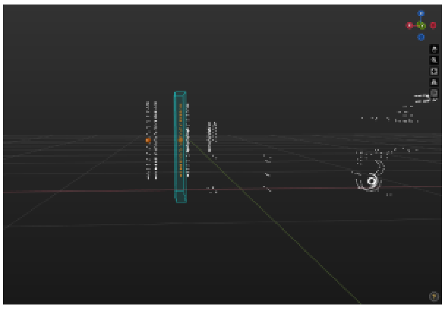
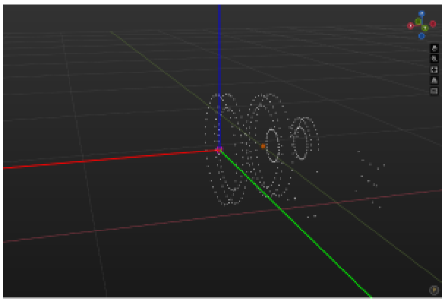
Once the model is exported successfully, you can add the model to Project resource tree › Model library. Then you can set the corresponding collision model of end tools in Project resource tree › Tools.

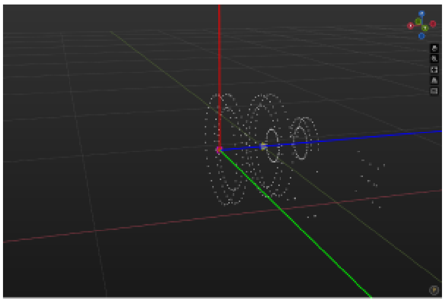
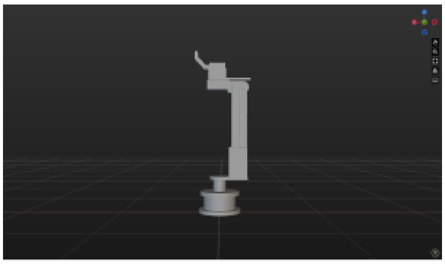
5.6.1.3.1. Adjust the Reference Frame of the Model

You should **set the reference frame origin at the model’s actual installation position**. Doing so will align the **origin of the model’s reference frame** with the **origin of the Model Editor’s world reference frame**.



Please read Selection Tool Usage Guide before proceeding to read the following content.

1. Select the model name in the “Reference Model” panel. 2. Click [**Cuboid Selection Tool**], and adjust the selection frame to enclose all the outermost vertices at the end of the flange. Since the center point is generated based on the vertices of the model, please ensure that the vertices on the part to be processed are within the selection tool.	
3. Click [**Create Center Point**], and the point will be used as the origin of the reference frame. To visualize the center point, you can right-click the center point name in the Assemblies panel and select “Show On Top” in the context menu. Or click the 👁 icon to the left of the model name to hide the model.	
4. Click [**Cuboid Selection Tool**] and adjust the selection frame to enclose the vertices that are parallel to the outermost vertices. 5. Click [**Create Center Point**], and this point will be used as the reference point of the Z-axis.	
6. Click [**Set Frame**]. 7. Check **Select center point** in the Coordinate Parameters panel on the right. 8. Click [**Unselected**] next to **Coordinate Origin**. 9. Click the center point in the 3D editing workspace, and a reference frame with the center point as the origin will be created.	

10. Click **[Unselected]** next to **Axis Z**. 11. Click the reference point of Z-axis you created before in the 3D editing workspace.	
12. Click **[OK]**. Then the model pose will change, and the origin of the world reference frame in the Model Editor will coincide with that of the model's actual installation reference frame.	

5.6.1.3.2. Selection Tool Usage Guide

The selection tools include **Cuboid Selection Tool**, **Elliptic Cylinder Selection Tool**, and **Cylinder Selection**.

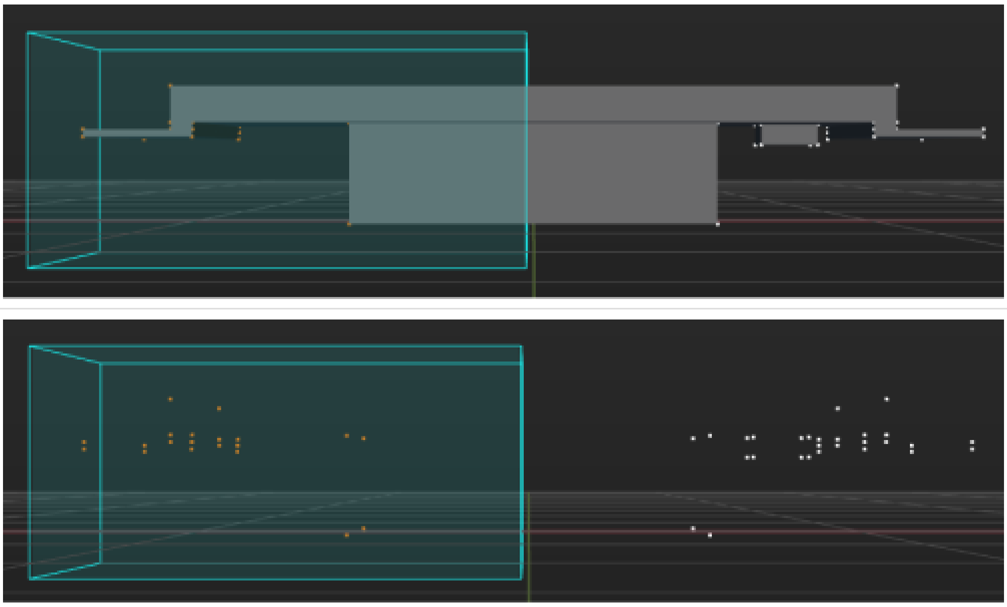
What can the selection tools be used for?

1. Selection tools can select the vertices of the reference model, and then a convex hull can be created based on the selected vertices.

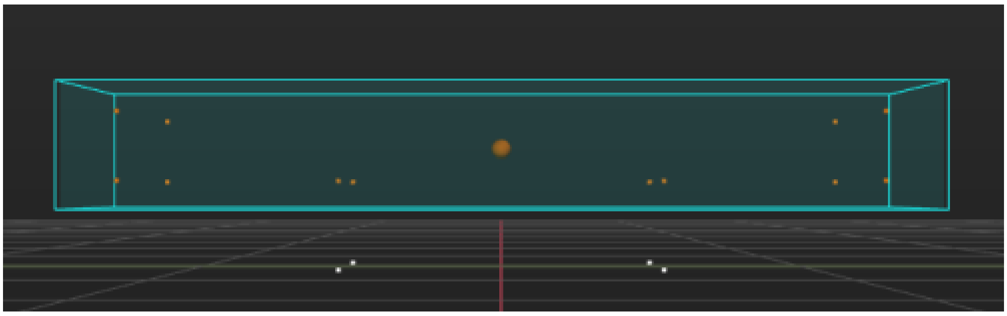
   As shown in the figure below, the vertices within the cuboid selection are selected, while vertices outside the cuboid selection are not selected.

   > Once you select the name of the reference model, click 👁 will hide the model only and the vertices of the model will still be displayed.

   The selected vertices will be displayed in yellow, and unselected vertices are white.



2. Selection tools can select the vertices of the reference model, and then a center point can be generated based on the selected vertices. The center point can be used to set the reference frame.



To display the center point on top without being obscured by the model, right-click the name of the center point in the Assemblies panel and select “Show On Top” in the context menu.

How to adjust the size, position, and orientation of the selection tool?

Click ▤ on the right of the interface to display the key and buttons you pressed. Click it again to disable the display.

The examples provided below are all based on the “Cylinder Selection Tool.” Unless otherwise specified, the usage of the “Elliptic Cylinder Selection Tool” and the “Cuboid Selection Tool” is the same as that of the “Cylinder Selection Tool.”

Translation

Press and hold the **Ctrl** key and click any surface of the selection tool until the yellow arrows appear. Then drag in any direction.

▶ https://docs.mech-mind.net/download/ModelEditor/translate-selection-tool-en.mp4 *(video)*

Custom Scaling

The “Cuboid Selection Tool” and “Cylinder Selection Tool” support custom scaling.

Press and hold the **Ctrl** key to activate the anchor points, and click on any green anchor point until it turns yellow. Then drag in any direction.

▶ https://docs.mech-mind.net/download/ModelEditor/scale-selection-tool-en.mp4 *(video)*

Proportional Scaling

Press and hold **Ctrl** + **Shift**, click on any vertex’s anchor point and drag it in any direction.

▶ https://docs.mech-mind.net/download/ModelEditor/proportional-scaling-selection-tool-en.mp4 *(video)*

Centroid Scaling

Press and hold **Ctrl** + **Alt**, click on any anchor point and drag it in any direction.

▶ https://docs.mech-mind.net/download/ModelEditor/centroid-scaling-selection-tool-en.mp4 *(video)*

Proportional Centroid Scaling

Press and hold **Ctrl** + **Alt** + **Shift**, click on any vertex's anchor point and drag it in any direction.

▶ https://docs.mech-mind.net/download/ModelEditor/proportional-centroid-scaling-selection-tool-en.mp4 *(video)*

Translate and Rotate Selection Tool Manipulator

Press and hold **Ctrl** + **Space** to activate the manipulator, click on any handle of the manipulator and drag in any direction.

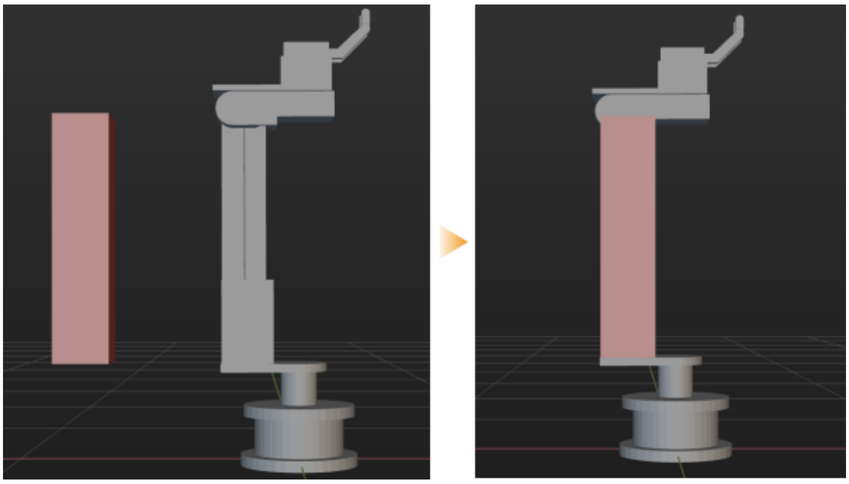
▶ https://docs.mech-mind.net/download/ModelEditor/translate-and-rotate-selection-tool-manipulator-en.mp4 *(video)*

In addition, you can set the parameters under **Dimensions**, **Center**, and **Rotation** in the 3D Selection Tool panel to adjust the size, position, and pose of the selection tool.

5.6.1.3.3. Create a Box or Cylinder

What can the box and cylinder be used for?

The created boxes and cylinders are regular convex bodies. The created convex bodies can enclose part of the model structure directly, saving you the need to manually select vertices for creating convex hulls.



How to use the box and cylinder?

Follow these steps to create a box. (The steps to create a cylinder are similar.)

1. Click **[Create Box]**.
2. Name the object in the pop-up window, and roughly modify the dimensions of the object, and then click **[OK]**. By default, the center of the newly created box coincides with the origin of the world reference frame in the 3D editing workspace.
3. Click the box you created, press and hold the `Ctrl` key and the left mouse button and drag the manipulator to move the box to the part to be enclosed.
4. Double-click the cuboid model, adjust the dimensions and pose in the Convex Object Configuration window to make the it enclose the part as accurately as possible.

5.6.1.3.4. Determine Whether the STEP/STP File Is Eligible

If any one of the following requirements of the STEP/STP file is not met, the model cannot be loaded successfully.

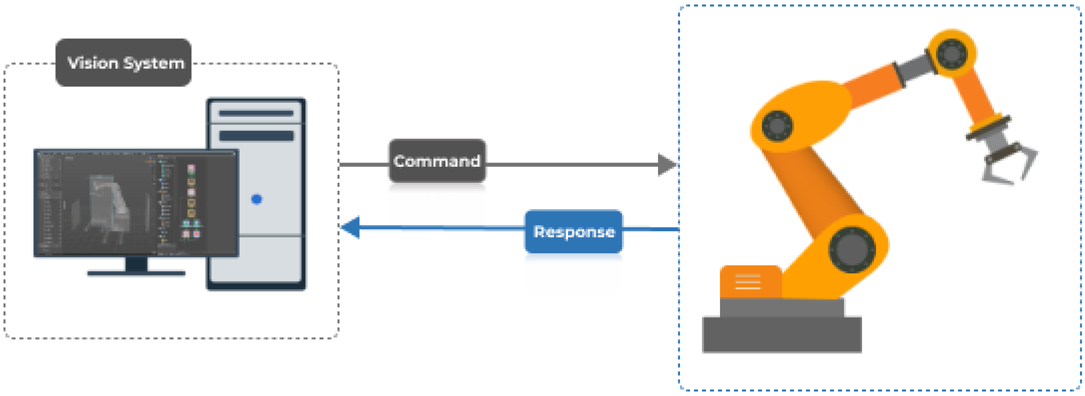
- The STEP/STP file is an ASCII file.
- The format of the STEP/STP file is defined by ISO 10303-21, and “ISO 10303-21” is on the first line of the file.
- Except for the marker on the first line, the HEADER and DATA sections are used to describe the information and data in the file respectively.

In addition, if the STEP/STP file is displayed in the garbled code after being opened, the file may be encrypted and you will need to contact the person who provided the file to decrypt it.

5.6.2. Master-Control Robot

In the main interface of the software, click Master-Control Robot in the toolbar to connect the robot.

Mech-Viz and the robot constitute a master/slave control or communication model after you load the monitoring program to the robot. Mech-Viz functions as the master device to send commands, while the robot functions as the slave device, performing tasks according to the commands received from the master device.



Please follow the steps below to master-control the robot:

1. Ensure that the **robot model** you select in the current project is the same as the **real robot model**. You can check the robot model in the project resource tree.
2. Refer to Master-Control Communication to load the Master-Control program to the robot and configure the communication settings. (Skip this step if you have finished the configuration.)
3. Execute the Master-Control program at the robot side.
4. Click [**Master-control Robot**] in the toolbar.
5. Enter the robot IP address.
6. Click [**Connect the robot**].
7. Check whether the Master-Control of the robot is successful.

If the current status shows **connected**, the robot is connected successfully. Then click **[Sync Robot]**, and the pose of the simulated robot will be synchronized with that of the real robot.

If you cannot connect to the robot successfully, please follow these guidelines to troubleshoot:

- Ensure that the **robot model** you select in the current project is the same as the **real robot model**.
- Ensure that the entered robot IP address is correct.
- Ensure that the Master-Control program is loaded successfully at the robot side.

5.6.3. Depalletizing Vacuum Gripper Configuration

This section introduces how to configure depalletizing vacuum grippers.

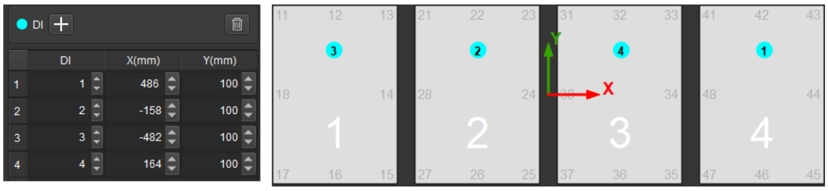
The vacuum gripper configuration is based on the selected tool type. Please refer to Configure End Tool to complete the tool configuration first. In the Tool Configuration window, set the **Tool type** to **Depalletizing vacuum gripper**, and click the **Configure depalletizing vacuum gripper** button at the bottom to open the vacuum gripper configurator.

Introduction

You can configure both single-block vacuum grippers and multi-block vacuum grippers. However, for multi-block vacuum grippers, the rectangular blocks should be of the same size and arranged in a single row.

Configuration Process

1. Configure the vacuum grippers structure.
   a. Set the **Number of Blocks** according to the number of blocks of the actual vacuum gripper. If multiple blocks share the same DO signal, the blocks can be viewed as a whole, and you can set the number of blocks to 1.
   b. Set the **Spacing** according to the actual interval between the blocks.
   c. Set the **Block length and Block width** according to the actual vacuum gripper size.
   d. Set **Buffer Thickness**, which will be added to the plane where the TCP is located for collision detection.
2. Set the **Angle** parameter according to the TCP settings in the tool configuration window.
3. Set the **DO/DI active value** parameter according to the actual situation.
4. Configure DI Check Point
   a. Click **[+]** to add a new DI check point. By default, the DI is 0, and the check point is at the center of the vacuum gripper.
   b. Modify the value of **X and Y** or drag the check point to adjust the position of the pick point.
   c. Modify the **DI** parameter to change the ID of the check point.



	DI	X(mm)	Y(mm)
1	1	486	100
2	2	-158	100
3	3	-482	100
4	4	164	100

If you want to delete a check point, select it and click 🗑.

1. Configure the corresponding **DO Value** according to the control port number of each vacuum gripper block.
2. Select File › Save to save the vacuum gripper configuration to the corresponding tool.

Export the Vacuum Gripper Configuration File

Select File › Export to store the configuration file to a specified path.

Import the Vacuum Gripper Configuration File

Select File › Import to import the vacuum gripper configuration file.

5.6.4. Array Gripper Configuration

This chapter introduces how to configure array grippers.

The array gripper configuration is based on the selected tool type. Please refer to Configure End Tool to complete the tool configuration first. In the Tool Configuration window, set the **Tool type** to **Array gripper**, and click the **Configure array gripper** button at the bottom to open the array gripper configurator.

Configuration Process

1. Set the **Number of tools** and **Spacing** according to the actual array gripper.
2. Set the **DO active value** parameter according to the actual situation.
3. Set the **Array angle** parameter according to the TCP settings in the Tool Configuration window.
4. Select File › Save to save the vacuum gripper configuration to the corresponding tool.

Export the Array Gripper Configuration File

Select File › Export to store the configuration file to a specified path.

Import the Array Gripper Configuration File

Select File › Import to import the vacuum configuration file.

5.6.5. Pallet Editor

The pallet editor is used to edit the customized pallet pattern in the Custom Pallet Pattern Step. You can click **[Pallet Editor]** in the lower left corner of the parameter panel of the Custom Pallet Pattern Step to open the tool.

Preparation

Before editing the pallet pattern, please ensure that you have the following information based on the on-site situation.

- Length and width of the pallet tray
- Number of layers

- Layer height
- Layout of each layer
- Length and width of the carton

Edit Pallet

You will need to edit the **Layer Layouts** and **Pallet Layers** to complete editing a pallet.

> Click File › Load From File to open the existing pallet configuration file.

Layer Layouts

You will need to set the pallet size, the number of layer layouts, and object positions in the Layer Layouts panel.

1. Set the **Pallet Length** and **Pallet Width**. They are the length and width of the pallet tray respectively.
2. Click **[Add Position]** to add cartons. You can drag the carton to adjust the positions.
3. In the **Position Setting** panel, set the size of the carton, rotational angle, and entry angle.
4. Repeat steps 2 and 3 to complete the editing of the current layer layout.
5. If you want to add multiple layer layouts, click **[Add Layer Layout]**.

Pallet Layers

You will need to set the number of layers, layer height, and layer layout for each layer in the Pallet Layers panel.

1. Click **[Add Layer]** until the number of layers meets the requirement.
2. Select a layer and enter the corresponding **Layer Height** to set the height of the current layer.
3. Select a layer layout in the drop-down list of **Change Layout** for the current layer.
4. Repeat steps 2 and 3 to complete the editing of all pallet layers.

Save and Use

Select File › Save (Ctrl + S) to save the pallet configuration, and the pallet pattern will be displayed in the 3D simulation area of the software.

Select File › Save to File and specify a directory to save the configuration file in **JSON** format. In the parameter panel of the Custom Pallet Pattern Step, click **[Load Pattern]** to select the pallet configuration file and therefore load the pallet pattern.

5.6.6. Key Query Tool

> This tool is available only when the Developer mode is enabled. Click Tools in the menu bar and select Key Query Tool to open it.

This tool is used to query the interfaces and interface parameters of each Step, typically used when writing Standard Interface programs and Adapter programs.

Enter the Step name you are looking for in the search box, select the Step in the Step list and the detailed parameter information will be displayed on the right.

By default, only frequently used Steps will be displayed. Click ▣ to show all Steps.

5.6.7. Robot Model Library

Building a Mech-Viz project usually starts with selecting a robot model. The robot model selected in the robot model library should match the robot model used in the actual applications.

Select Robot Model

To ensure that you can obtain and download the latest robot models, it is recommended to click **[Load online robot model list]** when the network access is available. If you are currently unable to connect to the internet, you can only use local robot models.

Obtaining the online robot model list does not mean that the local robot model list will be updated. After you obtain the online robot model list, click **[Update local robot model]** in the right-click shortcut menu of the robot model card with “Updatable” label, then you will update the local robot models. Please note that this will overwrite the local robot model file.

1. You can either enter the robot model into the search box in the top left corner or filter robot models by setting the filtering conditions.
2. Hover your mouse cursor over the robot model card. If the local robot model is not available, click **[Download]**, wait for the download to complete, then double-click the robot model card or click **[Select]**. If the local robot model is available, you can select it directly.

Additionally, clicking the robot model card will display the communication mode(s) supported by the current robot and its features in Master-Control communication.

Robot Communication Mode

- A checkmark (√) after Master-Control communication indicates that the current robot supports Master-Control Communication.

  Master-Control Communication: In this mode, the Vision System functions as the master device to send commands, while the robot functions as the slave device, performing tasks according to the commands received from the master device. This is generally used in scenarios where the robot does not need to communicate with devices other than the Vision System.

- A checkmark (√) after Standard Interface example programs indicates that there are example programs available in the software for the robot brand.

  Standard Interface Communication: In this mode, the robot side calls an interface command and instructs the Vision System to process it and return the result. Subsequently, the robot processes the received result to further derive its desired outcome. This is generally used in scenarios where robot programs are more complex, and the Vision System only serves as a vision data provider for the robot system. In theory, all robots support Standard Interface Communication.

Master-Control Communication Features

Mech-Viz supports different features for different robots, which affects Mech-Viz control over certain robot features.

After selecting a robot, the robot’s Master-Control Communication features will be displayed. A checkmark (√) indicates the feature is supported, while a cross (×) indicates not supported.

FAQs

Why do some robots have an “Updatable” label?

Robot model cards with an **Updatable** label indicate that the local robot model and the online robot model do not match.

You can use the local robot model without updating, or right-click the card and select **[Update local robot model]** to download the online robot model file. Please note that this will overwrite the local robot model file. After downloading, the robot model will be labeled as **Local**.

How to use self-prepared robot model files?

- To create robot models and import them into Mech-Viz, please refer to Robot Model Creating and Importing.
- To import robot installation package file (.mrob), please perform the following steps.

  > To export the robot installation package file (.mrob), you can go to Tools > Robot Library Tools > Export Current Robot.

  1. Click **[Install local robot model installation package]** on the lower-left corner of the Robot Model Library.
  2. Select the robot installation package file (.mrob) and click **[Open]**.
  3. If a window saying “Robot xxx resource file already exists. Do you want to overwrite? ” pops up, click **[Yes]**.
  4. Search for the robot installation package file name, move your mouse to the robot model card, and click **[Select]**.

Welcome to provide feedback on the robot models. If you think a certain robot model performs well, you can give it a thumbs-up. If you find any issue with the robot model, you can give it a thumbs-down. Your feedback will help us further enhance the accuracy of robot models!

5.7. Appendix

5.7.1. Save and Use Vision Records

With the use of vision records, you can run the Mech-Viz project individually without running the corresponding Mech-Vision project.

Save Vision Records

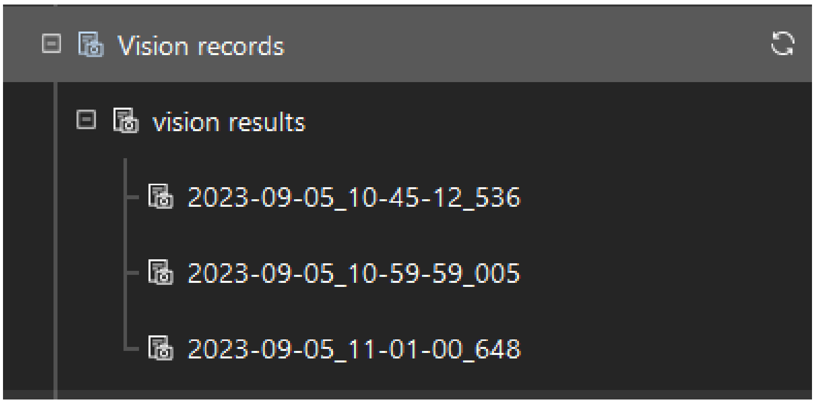
The vision records will be automatically saved to the project folder when simulating with the vision service or running the project with the real robot. 5 vision records will be saved by default. You can right-click “Vision records” in the project resource tree on the left and set the **Number of Saved Records** in the context menu.

Simulate by Using Vision Records

Follow the steps below to simulate by using vision records:

1. Open the Mech-Viz project, and ensure that the vision records of the current project are

available, as shown below.



Vision records

vision results

2023-09-05_10-45-12_536

2023-09-05_10-59-59_005

2023-09-05_11-01-00_648

2. Close Mech-Vision.
3. Click ▼ on the right of the [**Simulate**] button, and select **Use vision records**.
4. Click [**Simulate**] and a “Select Vision Record to Use” window will pop up.
5. Select the vision service and vision record you need.

   The selected record will be highlighted. The right arrow next to it indicates that the vision records will be retrieved from this one as a start. In addition, you can modify the retrieve mode or specify the pose in the vision record.

   > When there are multiple vision services, the vision records used for the simulation is determined by the “Vision Service Name” specified in “Vision Look” and “Vision Move”. Under this circumstance, you should select the vision record under the corresponding vision service.

6. Click [**Start simulation**] to simulate by using the vision records of the selected vision service.
7. Click the [**Stop**] button to make it stop.

Modify Retrieve Mode for Vision Records

You can modify the retrieve mode in the **Select Vision Record to Use** window according to your actual requirement.

Repeat all	The default retrieve mode, i.e., starting from the selected record, cycle through all vision records, and each record can be used multiple times.
Repeat one	Repeat the selected single vision record only.
Use in turn	Starting from the selected record, cycle through all vision records, and each record will be used only once.

Specify Pose in the Vision Record

Once a pose in the vision record is selected, only the pose with the specified index will be used for simulation.

The operation procedure is as follows.

1. In the **Select Vision Record to Use** window, select **Only use pose with specified index**.

2. Select the pose index in the drop-down list of the corresponding vision record.

5.7.2. Robot Model Creating and Importing

When the robot model library does not include the robot you need, the basic procedures for creating and importing a robot model are shown below.

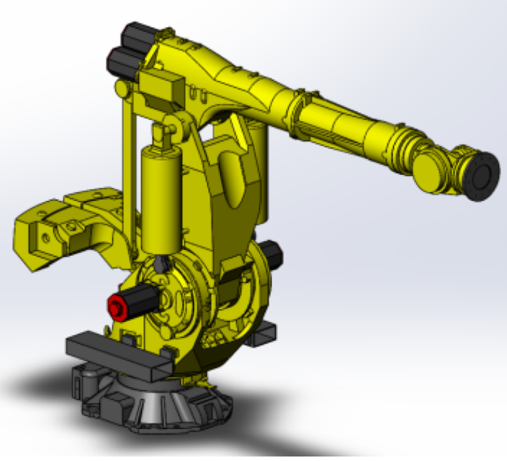
A FANUC M-900iB/400L robot will be used as an example.

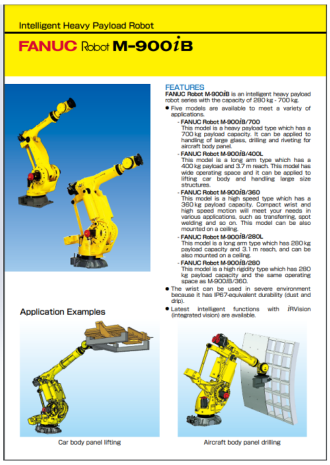
Obtain the CAD File and Specifications of the Robot

Before creating the robot model, please obtain the CAD file and the datasheet which contains DH parameters and motion ranges of each joint of the robot.

You can download the relevant documentation on the official website of the robot.

The figure below shows the CAD file and descriptions of FANUC M-900iB/400L.





Intelligent Heavy Payload Robot

FANUC Robot M-900*i*B

FEATURES

FANUC Robot M-900*i*B is an intelligent heavy payload robot series with the capacity of 280 kg - 700 kg.

- Five models are available to meet a variety of applications.
  - **FANUC Robot M-900*i*B/700**
    This model is a heavy payload type which has a 700 kg payload capacity. It can be applied to handling of large glass, drilling and riveting for aircraft body panel.
  - **FANUC Robot M-900*i*B/400L**
    This model is a long arm type which has a 400 kg payload and 3.7 m reach. This model has wide operating space and it can be applied to lifting car body and handling large size structures.
  - **FANUC Robot M-900*i*B/360**
    This model is a high speed type which has a 360 kg payload capacity. Compact wrist and high speed motion will meet your needs in various applications, such as transferring, spot welding and so on. This model can be also mounted on a ceiling.
  - **FANUC Robot M-900*i*B/280L**
    This model is a long arm type which has 280 kg payload capacity and 3.1 m reach, and can be also mounted on a ceiling.
  - **FANUC Robot M-900*i*B/280**
    This model is a high rigidity type which has 280 kg payload capacity and the same operating space as M-900*i*B/360.
- The wrist can be used in severe environment because it has IP67-equivalent durability (dust and drip).
- Latest intelligent functions with *i*RVision (integrated vision) are available.

Application Examples

Car body panel lifting

Aircraft body panel drilling

Some robot websites will provide models in X_T format. Comparing to STEP files, X_T files perform better and takes less time in reconstruction. Therefore, it is recommended to use models in X_T format.

Write the (robot)_algo.json File

(robot)_algo.json Template File

The template file of (robot)_algo.json is stored in `C:/Users/Administrator/AppData/Roaming/Mmind/robot/profile_example.json in the installation directory of Mech-Viz. You can create a new (robot)_algo.json file based on the template.

For the definition of the parameters in the (robot)_algo.json file, please refer to Description of [robot]_profile.json File.

Obtain the Configuration of the Robot (algo_type)

Different robots have different configurations which are described by different frames of reference and DH parameters.

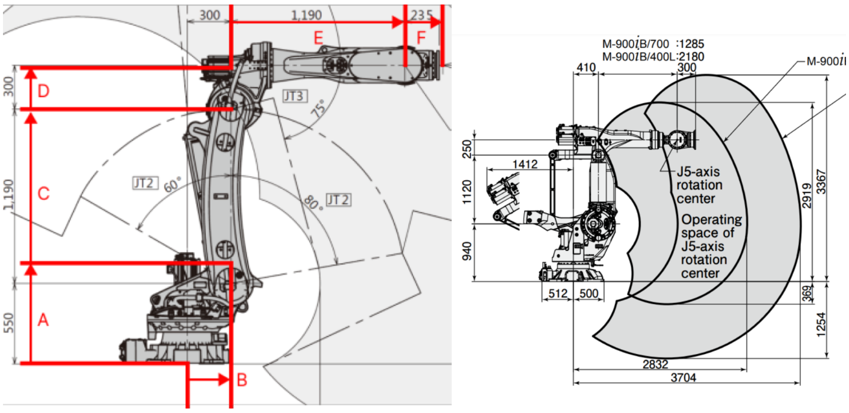
Please refer to Robot Configurations for examples of different robot configurations.

FANUC M-900iB/400L is a common 6-axis split-ball type wrist industrial robot, which belongs to the SphericalWrist_SixAxis configuration.

Obtain the DH Parameters (dh, dhPassive)

Please refer to Robot Configurations and find the corresponding configuration file of the robot. Then you can confirm the DH parameter value in (robot)_algo.json according to the DH parameters in the specifications of the robot.

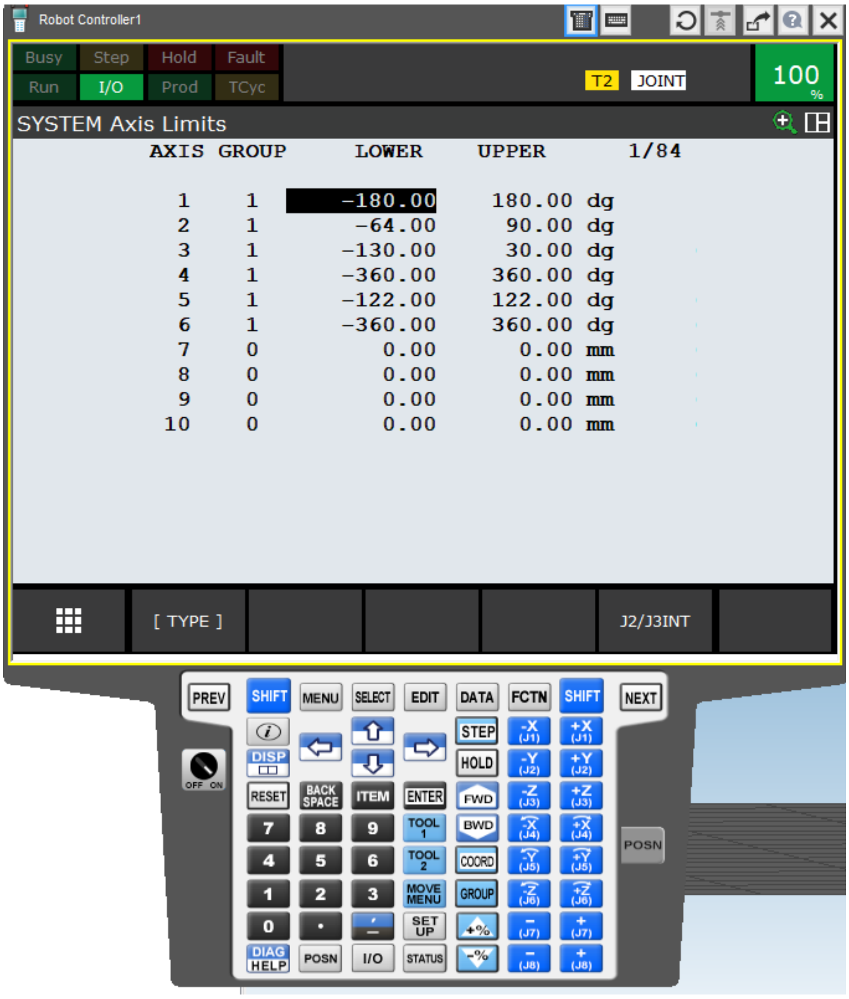
The DH parameters of FANUC M-900iB/400L are a=0.940, b=0.410, c=1.120, d=0.250, e=2.180, f=0.300.



In the above figure, the configuration diagram of SphericalWrist_SixAxis robot is on the left, and the specifications of FANUC M-900iB/400L is on the right.

Obtain the Motion Ranges of Each Joint (minlimits, maxlimits)

Usually, the motion ranges of joints are specified in the documentation of the robot. However, you need to open the robot simulation software RoboGuide of FANUC robots and then search for the motion ranges in this case.



SYSTEM Axis Limits

```
        AXIS  GROUP      LOWER      UPPER        1/84

          1     1       -180.00     180.00 dg
          2     1        -64.00      90.00 dg
          3     1       -130.00      30.00 dg
          4     1       -360.00     360.00 dg
          5     1       -122.00     122.00 dg
          6     1       -360.00     360.00 dg
          7     0          0.00       0.00 mm
          8     0          0.00       0.00 mm
          9     0          0.00       0.00 mm
         10     0          0.00       0.00 mm
```

Obtain Other Parameters

For common robot brands, you can refer to the parameters of other created robot models. If there are not available parameters to reference, please contact the site engineers for other parameters, i.e., mastering_joints, axis_flip, base_z_offset, axis_flip, and mastering_joints.

- Please use English punctuation marks in the JSON parameter file.
- The mastering_joints, axis_flip, base_z_offset attributes should correspond to those specified in the **robot simulation software** after creating a robot.
- The axis_flip attribute is coupled with minlimits and maxlimits. Sometimes it is necessary to switch the value of the upper and lower limits and reverse the positive and negative signs of the values at the same time.
- J6 in the mastering_joints attribute can be neglected easily. This attribute affects **Euler angles** and need to be checked and saved carefully.

Write the [robot]_profile.json Parameter File

[robot]_profile.json template file is stored in `resource/robot/profile_example.json` in the installation directory of Mech-Viz. You can create a new _profile.json file based on the template.

- For the definition of the parameters in the [robot]_profile.json file, please refer to

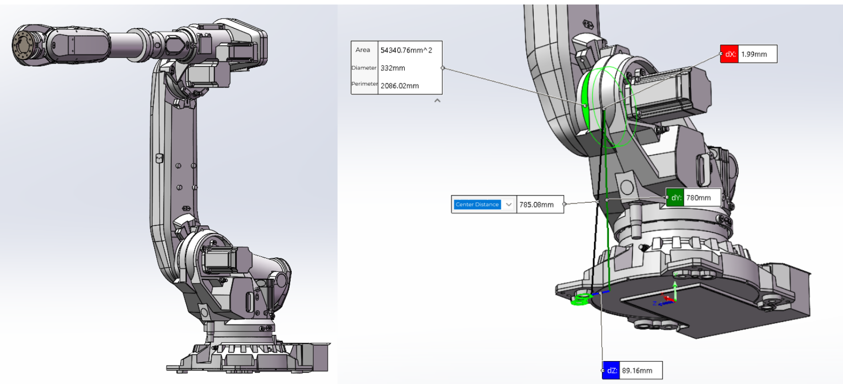
Description of [robot]_profile.json File.

- This file is not mandatory.

Use Solidworks to Construct a STL Model of the Entire Robot

Import the CAD Model in SolidWorks

Open the CAD model of the robot in SolidWorks, as shown below.



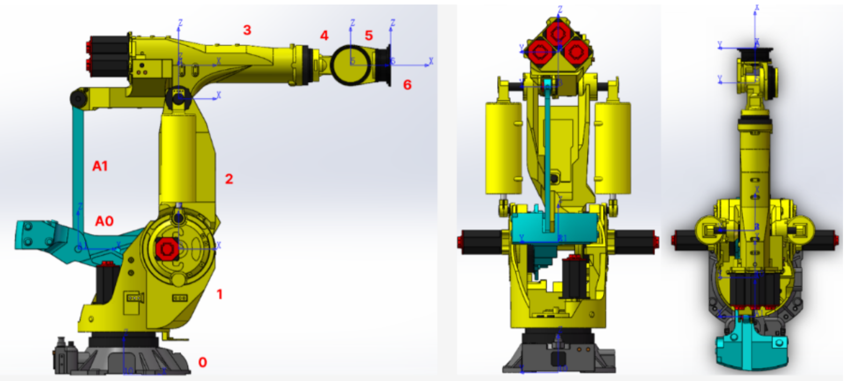
In the above figure, the CAD robot model is on the left and an assembly reference diagram is on the right.

- It is recommended to download an entire robot model that has been assembled completely.
- If you can download only the parts, you will need to assemble them yourself and check each joint according to the DH parameters.
- You may remove the details of the model and keep only the structure that may affect collision detection to reduce the time to import the model.

Create Reference Frames

Please refer to Robot Configurations to create the reference frame of each axis.

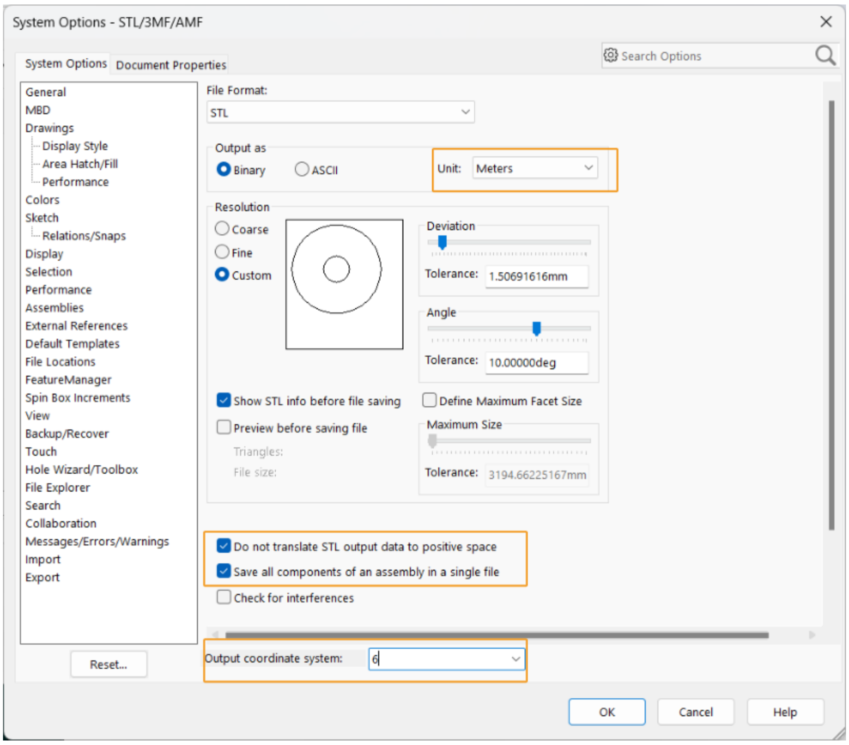
As for FANUC M-900iB/400L, the configuration is SphericalWrist_SixAxis, and the robot pose should be like the figure below: J1 axis is aligned with the base, J2 axis stays vertically upward, J3 axis and J4 axis stays horizontally forward, and J5 axis stays forward rather than downward. Once the pose is determined, all parts should be fixed.



Export STL Models of Robot Joints

It is recommended to hide other parts on the assembly when exporting the joint model in turn.

Save the exported file in STL format, set the output to binary and the unit to meter. Please refer to Robot Configurations for detailed information. Other settings are shown below.



System Options - STL/3MF/AMF

System Options | Document Properties

Search Options

General
MBD
Drawings
- Display Style
- Area Hatch/Fill
- Performance

Colors
Sketch
- Relations/Snaps

Display
Selection
Performance
Assemblies
External References
Default Templates
File Locations
FeatureManager
Spin Box Increments
View
Backup/Recover
Touch
Hole Wizard/Toolbox
File Explorer
Search
Collaboration
Messages/Errors/Warnings
Import
Export

File Format: STL

Output as: ● Binary ○ ASCII

Unit: Meters

Resolution: ○ Coarse ○ Fine ● Custom

Deviation — Tolerance: 1.50691616mm

Angle — Tolerance: 10.00000deg

☑ Show STL info before file saving

□ Preview before saving file

Triangles:

File size:

□ Define Maximum Facet Size

Maximum Size — Tolerance: 3194.66225167mm

☑ Do not translate STL output data to positive space

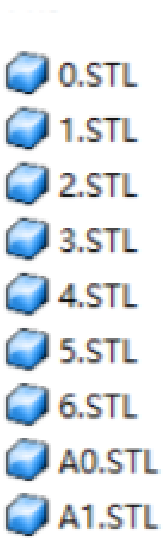
☑ Save all components of an assembly in a single file

□ Check for interferences

Output coordinate system: 6

Reset... | OK | Cancel | Help

The entire robot model of FANUC M-900iB/400L in STL format is shown below.



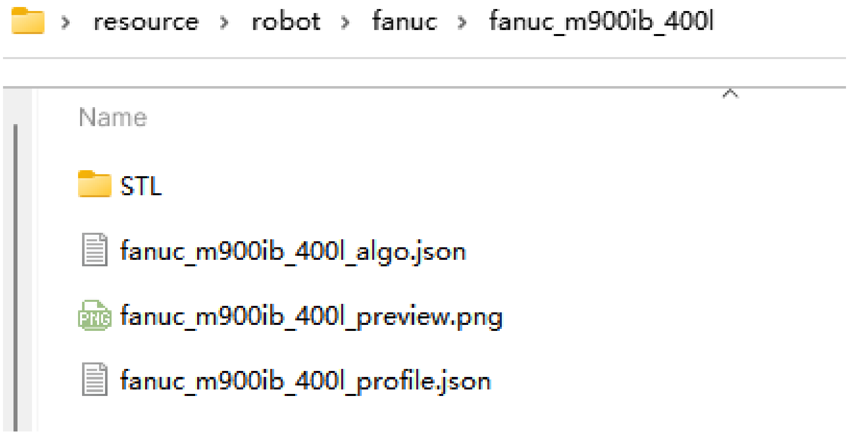
0.STL
1.STL
2.STL
3.STL
4.STL
5.STL
6.STL
A0.STL
A1.STL

Now you have completed creating a robot model.

Use the Created Robot Model

The robot library of Mech-Viz is stored in `resource/robot` in the installation directory. Robot models of the same brand are stored in the same folder which is named after the brand name, and all the files of the robot model are stored in the folder which is named after the model name.

As for FANUC M900iB/400L, the model files are stored in `resource/robot/fanuc_m900ib_400l`. The descriptions of each file are as follows.



resource › robot › fanuc › fanuc_m900ib_400l

Name

STL

fanuc_m900ib_400l_algo.json

fanuc_m900ib_400l_preview.png

fanuc_m900ib_400l_profile.json

- STL: the robot model file
- fanuc_m900ib_400l_algo.json: robot kinematic parameter file
- fanuc_m900ib_400l_preview.png: image of the robot
- fanuc_m900ib_400l_profile.json: parameter description file of the robot

> - Use lower case letters to name the folders.
> - The robot model file supports STL, DAE, and OBJ format. Different formats of model files are stored in a folder which is named after the format name respectively.
> - STL model can be used for both display and collision detection.
> - DAE model can only be used for display.
> - OBJ model can only be used for collision detection.
> - A complete robot model set should include at least one model used for display and one for collision detection.

Copy and paste all files into the corresponding directory and the robot model will be imported successfully.

Then you can select the newly imported robot from the Robot Model Library in Mech-Viz.

5.7.2.1. Robot Configurations

> Configuration support indicates that this robot can use Mech-Viz to conduct simulation and path planning.

Basic Configuration Lists

Six-Axis	UR_UR5_Like (six-axis collaborative robot similar to UR5)
	Painting_SixAxis (six-axis spray painting robot)
	SphericalWrist_SixAxis (common industrial six-axis spherical wrist robot)
	SphericalWrist_SixAxis_KAWASAKI_BX250L_Like (six-axis spherical wrist robot with positive X-direction and Z-direction offsets (G, D) on the J3)
	SphericalWrist_SixAxis_STAUBLI_TX2_60_Like (six-axis spherical wrist robot with a positive Y-direction offset (H) on the J3)
Five-Axis	SphericalWrist_FiveAxis (similar to the six-axis spherical wrist robot with J4 not able to roll)
Four-Axis	Palletizer (four-axis palletizing robot)
	Palletizer_KAWASAKI_CP180L_Like (four-axis palletizing robot with a positive Z-direction offset (D) on the J3)
	Scara_FourAxis_Translation_1st (four-axis SCARA robots with axis 1 as translation axis)
	Scara_FourAxis (four-axis SCARA)

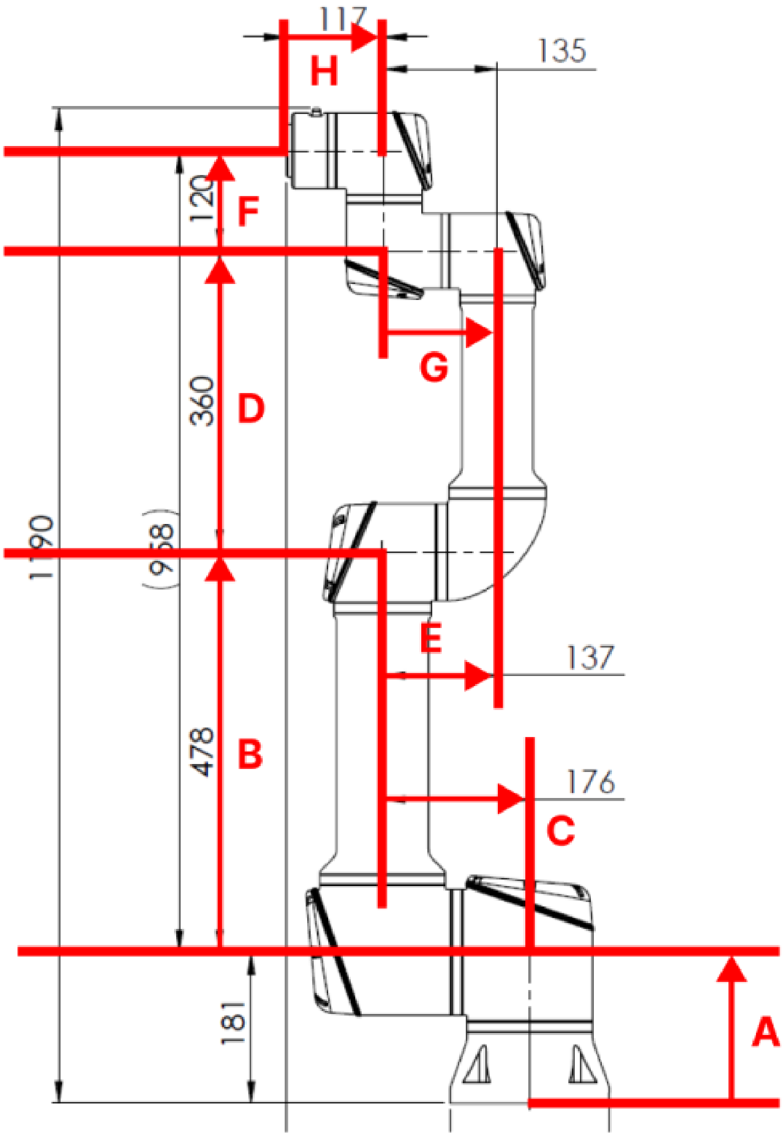
Special Configuration Lists

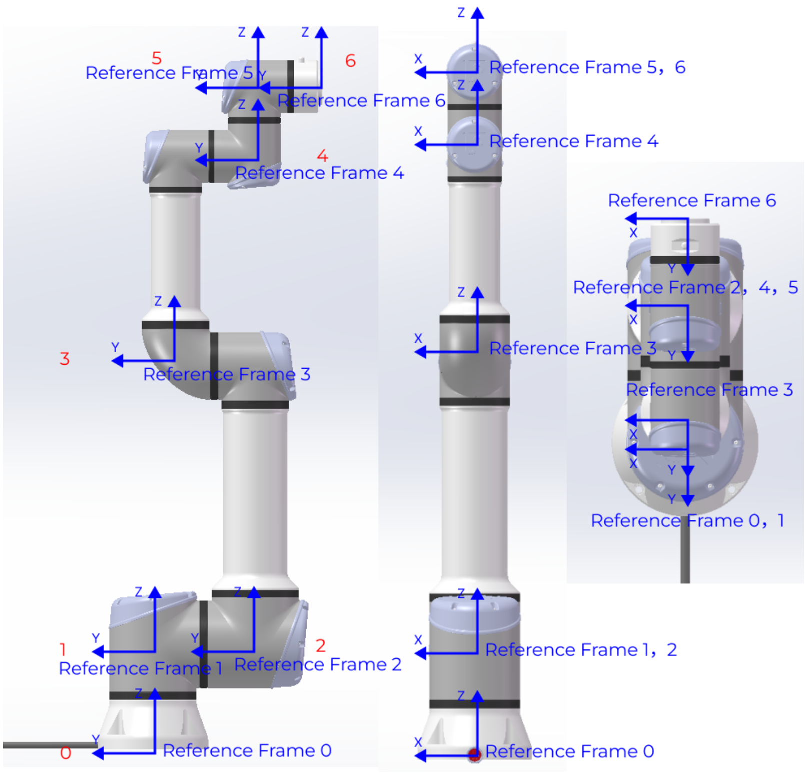
Six-Axis	OffsetWrist_Y_SixAxis_FANUC_CRX_10IA_Like (there is a negative Y-direction offset (L) on the J5)
	OffsetWrist_Y_SixAxis_Rokae_SR4_Like (there is a positive X-direction offset (G) on the J2, and a negative Y-direction offset (L) on the J5)
	OffsetWrist_Y_SixAxis_DENSO_COBOTTA_PRO_1300_Like (there is a negative Y-direction offset (H) on the J3, and a positive Y-direction offset (L) on the J5)
	OffsetWrist_Y_SixAxis_DENSO_COBOTTA_Like (there are positive Y-direction and Z-direction offsets (H, D) on the J3, and a negative Y-direction offset (L) on the J5)
	OffsetWrist_Z_SixAxis_ABB_CRB15000_Like (there is a positive Z-direction offset (M) on the J5)

5.7.2.1.1. Basic Configurations

UR_UR5_Like six-axis collaborative robot (similar to UR5)

Axes	Six
Configuration name	UR_UR5_Like
Description	Six-axis collaborative robot similar to UR5

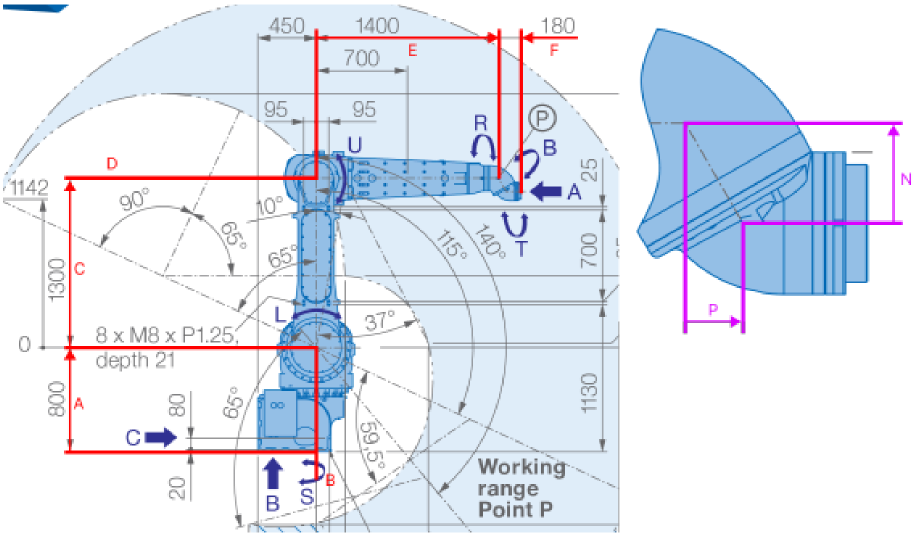
DH parameters	 117 135 H 120 F G 360 D 1190 (958) E 137 176 478 B C 181 A

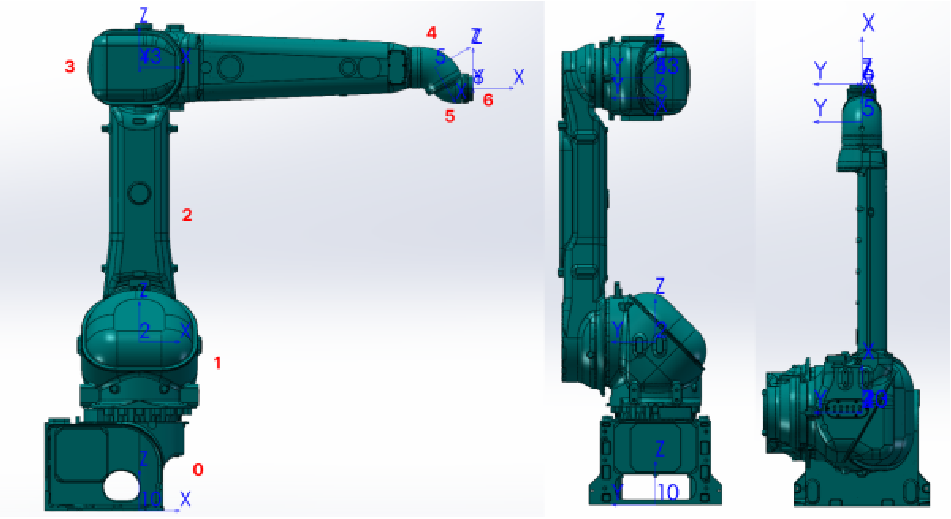
Reference frame and axis No.	 5 Reference Frame 5 6 Reference Frame 6 4 Reference Frame 4 3 Reference Frame 3 1 Reference Frame 1 2 Reference Frame 2 0 Reference Frame 0 Reference Frame 5，6 Reference Frame 4 Reference Frame 3 Reference Frame 1，2 Reference Frame 0 Reference Frame 6 Reference Frame 2，4，5 Reference Frame 3 Reference Frame 0，1 Z Y X
Correlation between link and reference frame	0: Reference frame 0
	1: Reference frame 1
	2: Reference frame 2
	3: Reference frame 3
	4: Reference frame 4
	5: Reference frame 5
	6: Reference frame 6
Classification of axes	Wrist: 5, 6, 4
	Forearm: 3
	Upper arm: 2
	Base: 0, 1
Robot model list	UR, TM, ELITE, AUBO, JAKA, SINSUN collaborative robots, DOBOT collaborative robots

[robot]_algo example

```
{
    "algo_type": "UR_UR5_Like",
    #DH parameters [meter]
    "dh": [ A, B, D, G, F, H ],
    "shoulder_offset": C,
    "elbow_offset": E,
    #joint limits [degree]
    "min_limits": [ J1 min, J2 min, J3 min, J4 min, J5 min, J6 min ],
    "max_limits": [ J1 max, J2 max, J3 max, J4 max, J5 max, J6 max ],
    #joint positions at standard pose [degree]
    "mastering_joints": [ J1, J2, J3, J4, J5, J6 ],
    #joint orientations [0/1]
    "axis_flip": "J1J2J3J4J5J6",
    #robot base location [meter,meter,meter,degree,degree,degree]. If robot has
not been moved, robot base frame equals to World frame.
    "robot_frame_in_robot_base_frame": [ x, y, z, rx, ry, rz ],
    #robot flange frame orientation [x/y/z]
    "flange_frame_orientation": "z"
}
```

Painting_SixAxis (six-axis spray painting robot)

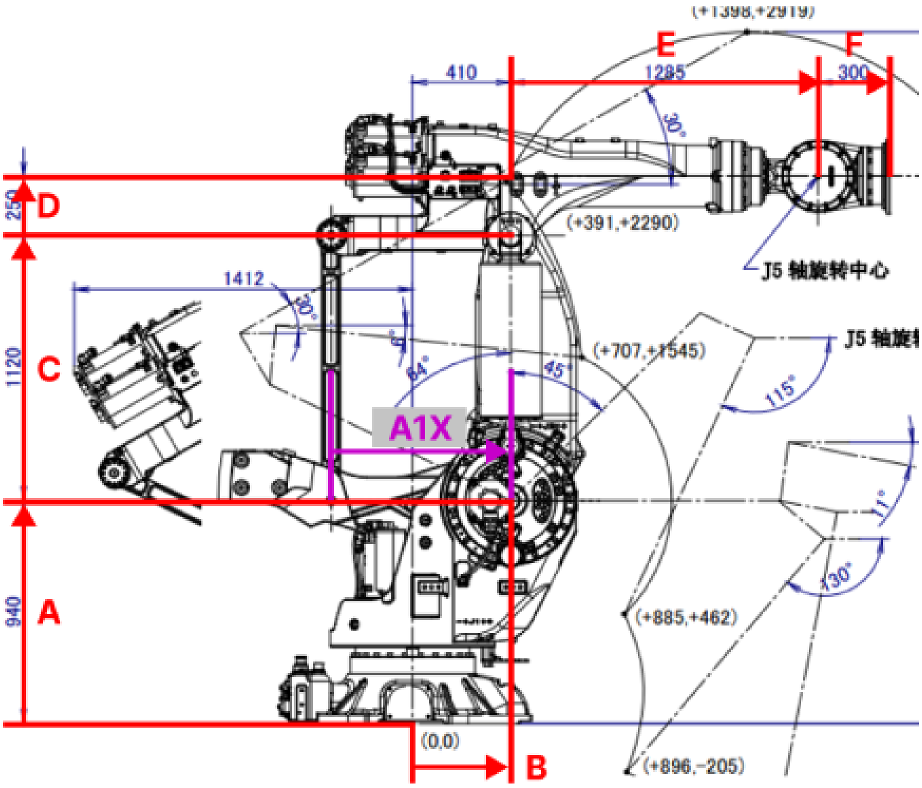
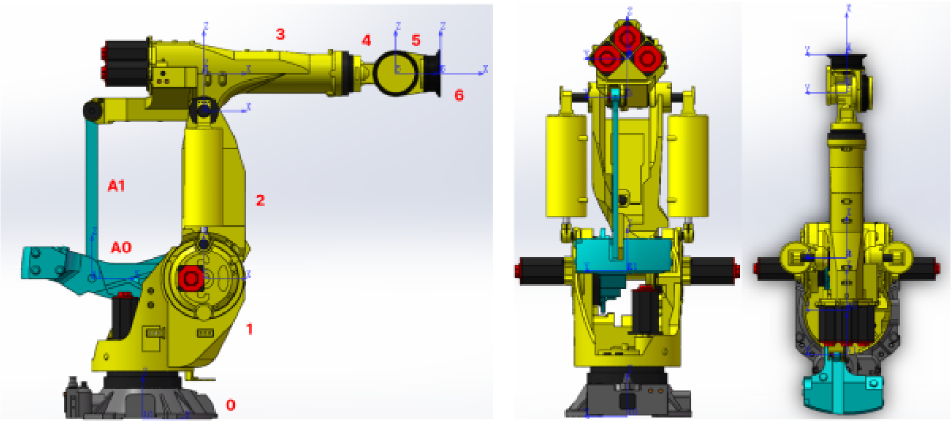
Axes	Six
Configuration name	Painting_SixAxis
Description	Six-axis spray painting robot
DH parameters	 450 1400 180 700 E F 95 95 R P B U D 1142 25 A 90° 10° 65° 115° 140° T 700 1300 C 65° L 37° 0 8 x M8 x P1.25, depth 21 800 A 80 65° 1130 C 59.5° B S B 20 Working range Point P N P

Reference frame and axis No.	 Z X3 X 3 4 Z 5 X 6 5 2 Z 2 X 1 Z 0 10 X Z Y Y Z Y Z 10 X Y Y Z 5 X Y 20
Correlation between link and reference frame	0: Reference frame 0, 1
	1: Reference frame 0, 1
	2: Reference frame 2
	3: Reference frame 3
	4: Reference frame 4
	5: Reference frame 5
	6: Reference frame 6
Classification of axes	Wrist: 5, 6, 4
	Forearm: 3
	Upper arm: 2
	Base: 0, 1
Robot model list	Yaskawa MPX3500, EFORT GR6150 1500

[robot]_algo example

```
{
    "algo_type": "Painting_SixAxis",
    #DH parameters [meter]
    "dh": [ A, B, C, D, E, F, N, P ],
    #joint limits [degree]
    "min_limits": [ J1 min, J2 min, J3 min, J4 min, J5 min, J6 min ],
    "max_limits": [ J1 max, J2 max, J3 max, J4 max, J5 max, J6 max ],
    "link3_dynamic_limits": [Min, Max],
    #joint positions at standard pose [degree]
    "mastering_joints": [ J1, J2, J3, J4, J5, J6 ],
    #joint orientations [0/1]
    "axis_flip": "J1J2J3J4J5J6",
    #robot base location [meter,meter,meter,degree,degree,degree]. If robot has
not been moved, robot base frame equals to World frame.
    "robot_frame_in_robot_base_frame": [ x, y, z, rx, ry, rz ],
    #robot flange frame orientation [x/y/z]
    "flange_frame_orientation": "z"
}
```

SphericalWrist_SixAxis (common industrial six-axis spherical wrist robot)

Axes	Six
Configuration name	SphericalWrist_SixAxis
Description	Common industrial six-axis spherical wrist robot
DH parameters	 (+1398,+2919) E F 410 1285 300 30° 250 D (+391,+2290) J5 轴旋转中心 1412 30° 1120 C (+707,+1545) J5 轴旋转 45° 64° 115° A1X 11° 130° 940 A (+885,+462) (0,0) B (+896,-205)
Reference frame and axis No.	

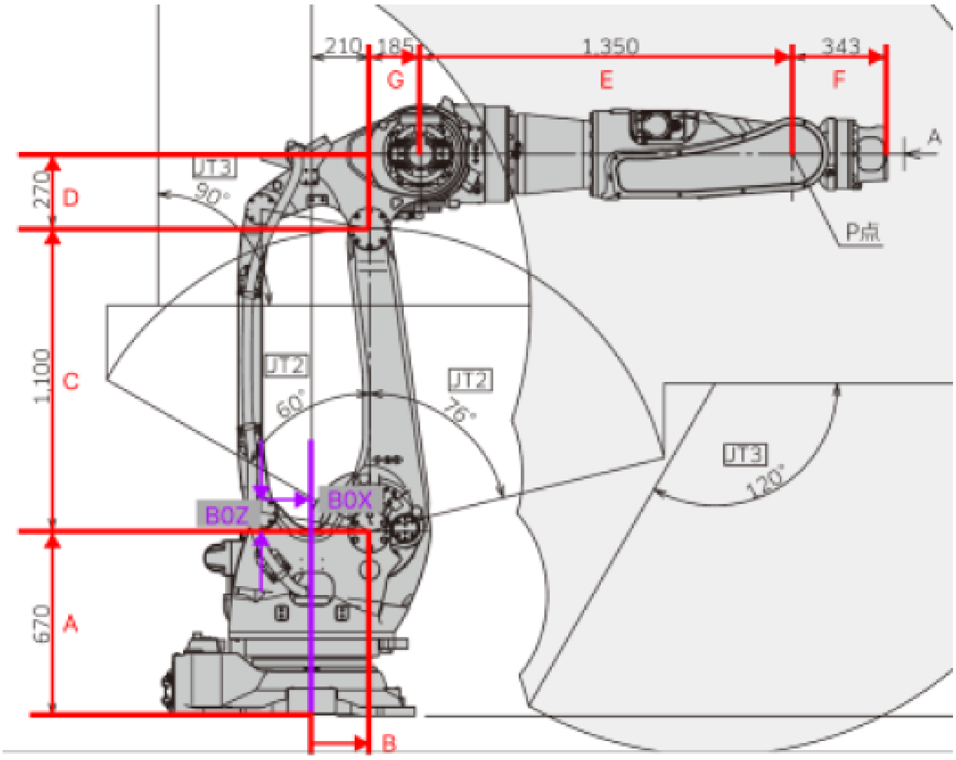
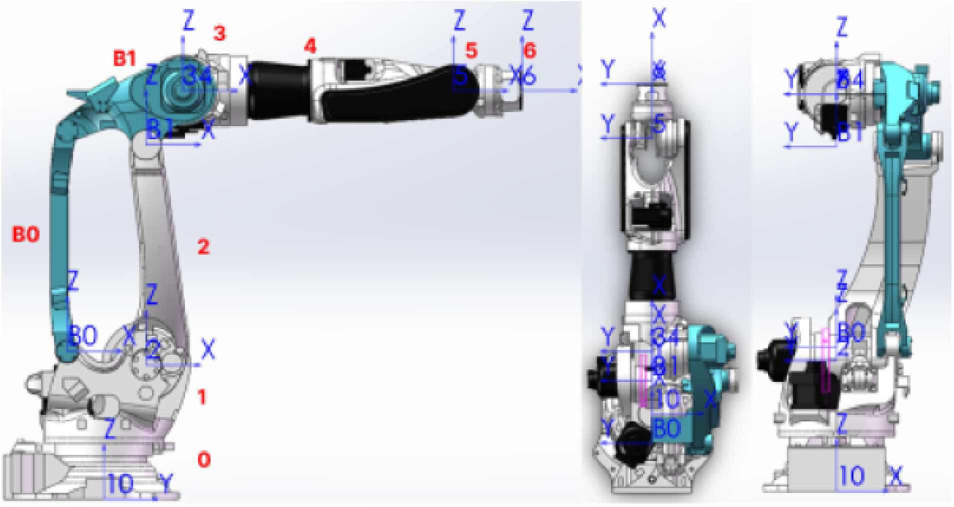
Correlation between link and reference frame	0: Reference frame 0, 1
	1: Reference frame 0, 1
	2: Reference frame 2
	3: Reference frame 3
	4: Reference frame 4
	5: Reference frame 5
	6: Reference frame 6
	A0: Reference frame 2
	A1: Reference frame A1
Classification of axes	Wrist: 5, 6
	Forearm: 3, 4
	Upper arm: 2, A1
	Base: 0, 1, A0
Robot model list	Yaskawa MPX3500, EFORT GR6150 1500 Kawasaki CX110L, ABB IRB6700 150-320, FANUC R-2000iC 210F, FANUC M-900iB 700, YASKAWA GP400, Kawasaki MX series

[robot]_algo example

```
{
    "algo_type": "SphericalWrist_SixAxis",
    #DH parameters [meter]
    "dh": [ A, B, C, D, E, F ],
    "dhPassive": [ A1X, A1Z, B0X, B0Z, B2X, B2Z ],
    #joint limits [degree]
    "min_limits": [ J1 min, J2 min, J3 min, J4 min, J5 min, J6 min ],
    "max_limits": [ J1 max, J2 max, J3 max, J4 max, J5 max, J6 max ],
    "link3_dynamic_limits": [Min, Max],
    #joint positions at standard pose [degree]
    "mastering_joints": [ J1, J2, J3, J4, J5, J6 ],
    #joint orientations [0/1]
    "axis_flip": "J1J2J3J4J5J6",
    #robot base location [meter,meter,meter,degree,degree,degree]. If robot has not been moved, robot base frame equals to World frame.
    "robot_frame_in_robot_base_frame": [ x, y, z, rx, ry, rz ],
    #robot flange frame orientation [x/y/z]
    "flange_frame_orientation": "z"
}
```

SphericalWrist_SixAxis (six-axis spherical wrist robot with positive X-direction and Z-direction offsets (G, D) on the J3)

Axes	Six

Configuration name	SphericalWrist_SixAxis_KAWASAKI_BX250L_Like
Description	Six-axis spherical wrist robot with positive X-direction and Z-direction offsets (G, D) on the J3
DH parameters	 210 185 1,350 343 G E F A JT3 90° 270 D P点 1,100 C JT2 60° JT2 76° JT3 120° B0X B0Z 670 A B
Reference frame and axis No.	 Z 3 B1 4 Z 5 Z 6 34 X B1 X X6 B0 2 Z Z B0 X X 1 Z 0 10 Y X Y Y 5 X Y 34 B1 10 X Y B0 Z Y 34 Y B1 Z B0 Z 10 X

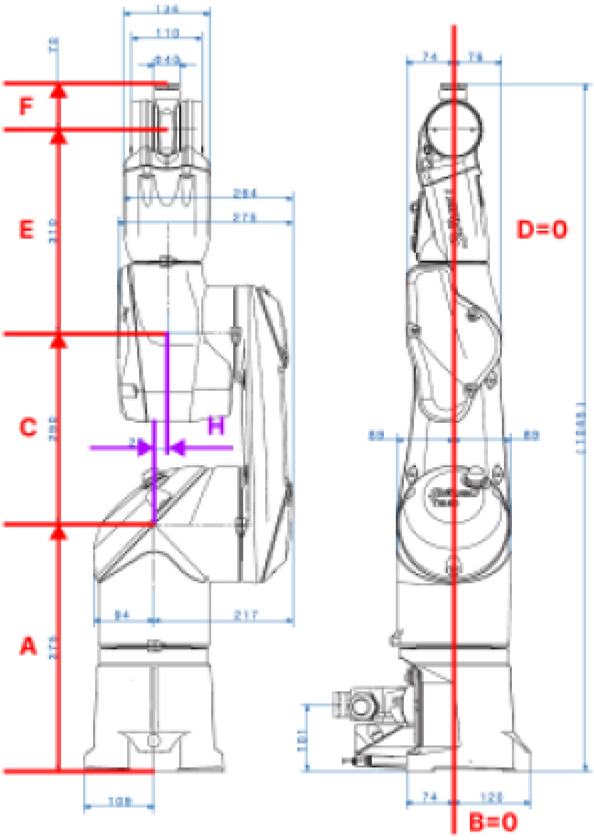
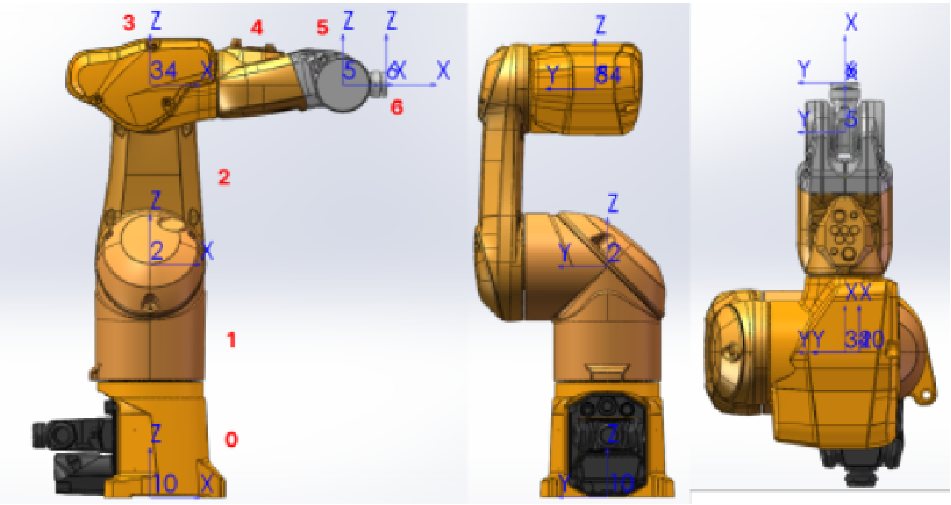
Correlation between link and reference frame	0: Reference frame 0, 1
	1: Reference frame 0, 1
	2: Reference frame 2
	3: Reference frame 3
	4: Reference frame 4
	5: Reference frame 5
	6: Reference frame 6
	B0: Reference frame B0
	B1: Reference frame B1
Classification of axes	Wrist: 5, 6, 4
	Forearm: 3
	Upper arm: 2, B0, B1
	Base: 0, 1
Robot model list	KAWASAKI BX250L, 300L, 200X

[robot]_algo example

```
{
    "algo_type": "SphericalWrist_SixAxis_KAWASAKI_BX250L_Like",
    #DH parameters [meter]
    "dh": [ A, B, C, D, G, E, F ],
    "dhPassive": [ A1X, A1Z, B0X, B0Z, B2X, B2Z ],
    #joint limits [degree]
    "min_limits": [ J1 min, J2 min, J3 min, J4 min, J5 min, J6 min ],
    "max_limits": [ J1 max, J2 max, J3 max, J4 max, J5 max, J6 max ],
    "link3_dynamic_limits": [Min, Max],
    #joint positions at standard pose [degree]
    "mastering_joints": [ J1, J2, J3, J4, J5, J6 ],
    #joint orientations [0/1]
    "axis_flip": "J1J2J3J4J5J6",
    #robot base location [meter,meter,meter,degree,degree,degree]. If robot has not been moved, robot base frame equals to World frame.
    "robot_frame_in_robot_base_frame": [ x, y, z, rx, ry, rz ],
    #robot flange frame orientation [x/y/z]
    "flange_frame_orientation": "z"
}
```

SphericalWrist_SixAxis_STAUBLI_TX2_60_Like (six-axis spherical wrist robot with a positive Y-direction offset (H) on the J3)

Axes	Six

Configuration name	SphericalWrist_SixAxis_STAUBLI_TX2_60_Like
Description	Six-axis spherical wrist robot with a positive Y-direction offset on the J3
DH parameters	 F E C H A D=0 B=0 110 84 217 108 74 74
Reference frame and axis No.	 3 4 5 6 2 1 0 Z X Y 34 5 6 2 10 84 320

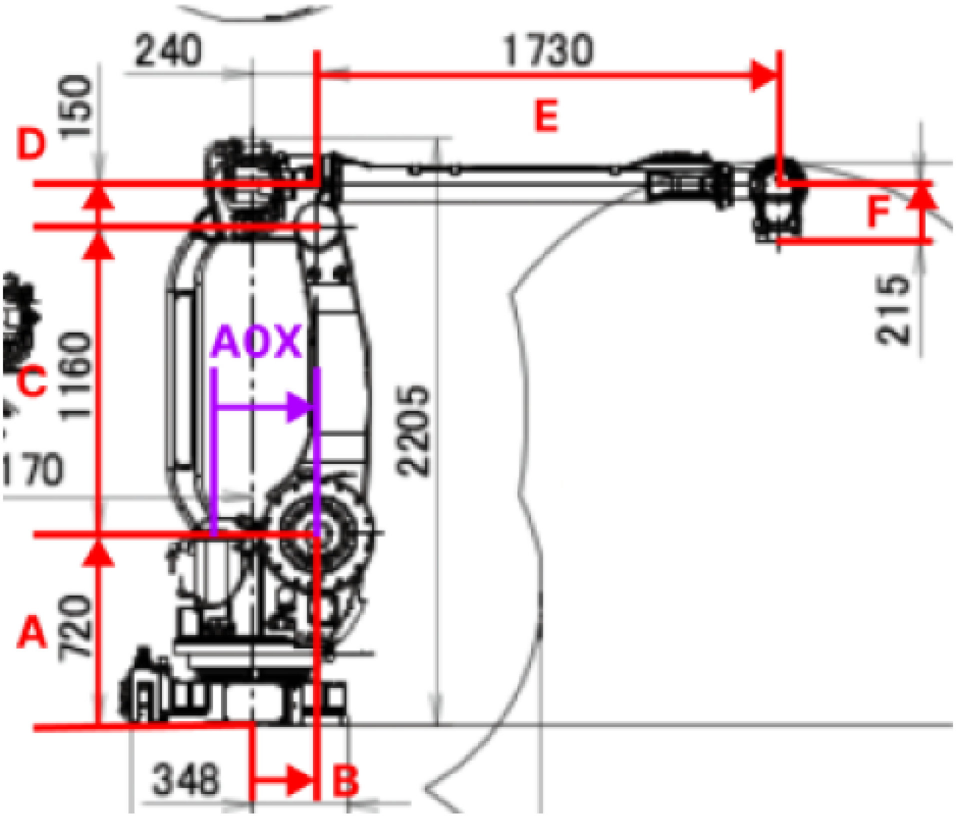
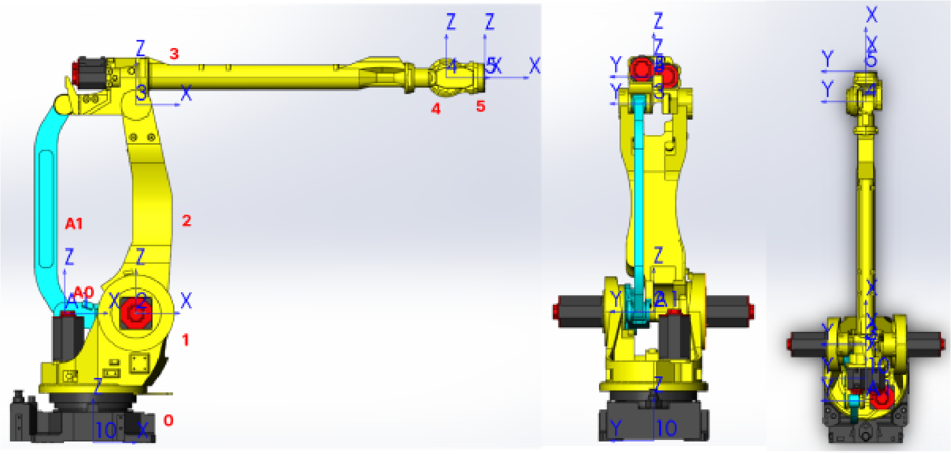
Correlation between link and reference frame	0: Reference frame 0, 1
	1: Reference frame 0, 1
	2: Reference frame 2
	3: Reference frame 3
	4: Reference frame 4
	5: Reference frame 5
	6: Reference frame 6
Classification of axes	Wrist: 5, 6
	Forearm: 3, 4
	Upper arm: 2
	Base: 0, 1
Robot model list	STAUBLI TX2-60

[robot]_algo example

```
{
    "algo_type": "SphericalWrist_SixAxis_STAUBLI_TX2_60_Like",
    #DH parameters [meter]
    "dh": [ A, B, C, D, E, F, H ],
    "dhPassive": [ A1X, A1Z, B0X, B0Z, B2X, B2Z ],
    #joint limits [degree]
    "min_limits": [ J1 min, J2 min, J3 min, J4 min, J5 min, J6 min ],
    "max_limits": [ J1 max, J2 max, J3 max, J4 max, J5 max, J6 max ],
    "link3_dynamic_limits": [Min, Max],
    #joint positions at standard pose [degree]
    "mastering_joints": [ J1, J2, J3, J4, J5, J6 ],
    #joint orientations [0/1]
    "axis_flip": "J1J2J3J4J5J6",
    #robot base location [meter,meter,meter,degree,degree,degree]. If robot has not been moved, robot base frame equals to World frame.
    "robot_frame_in_robot_base_frame": [ x, y, z, rx, ry, rz ],
    #robot flange frame orientation [x/y/z]
    "flange_frame_orientation": "z"
}
```

SphericalWrist_FiveAxis (similar to the six-axis spherical wrist robot with J4 not able to roll)

Axes	Five
Configuration name	SphericalWrist_FiveAxis
Description	Similar to the six-axis spherical wrist robot with J4 not able to roll

DH parameters	 240 1730 150 D E F 215 1160 C AOX 2205 170 A 720 348 B
Reference frame and axis No.	 Z 3 X Z Z X 4 5 A1 2 A0 1 0 10 Y A1 10
Correlation between link and reference frame	0: Reference frame 0, 1
	1: Reference frame 0, 1
	2: Reference frame 2
	3: Reference frame 3
	4: Reference frame 4
	5: Reference frame 5
	A0: Reference frame 2
	A1: Reference frame A1

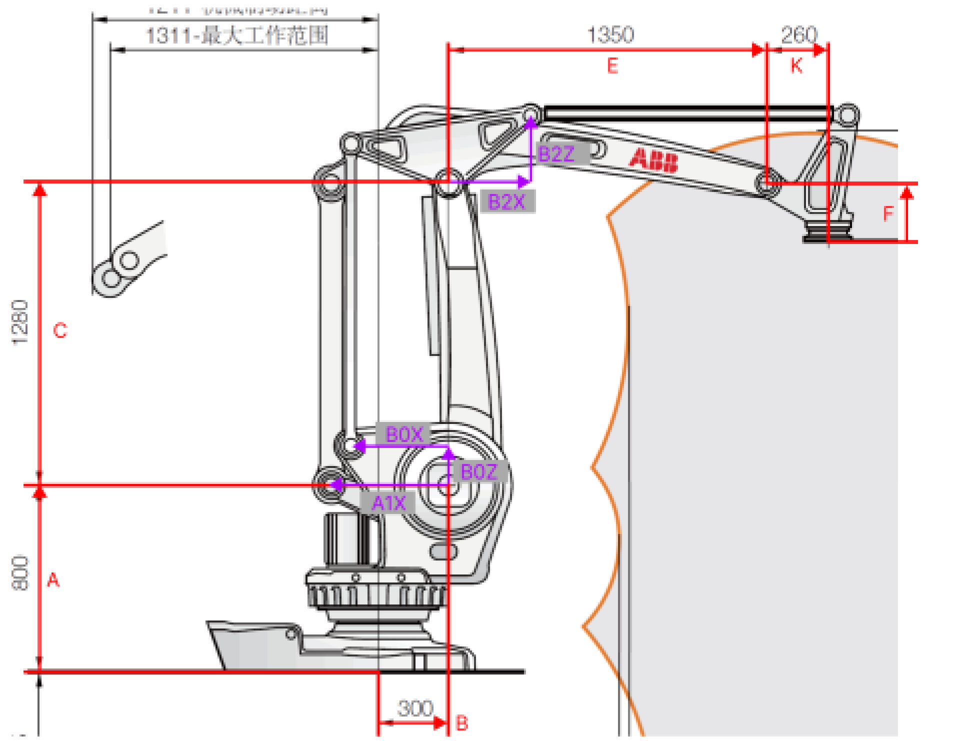
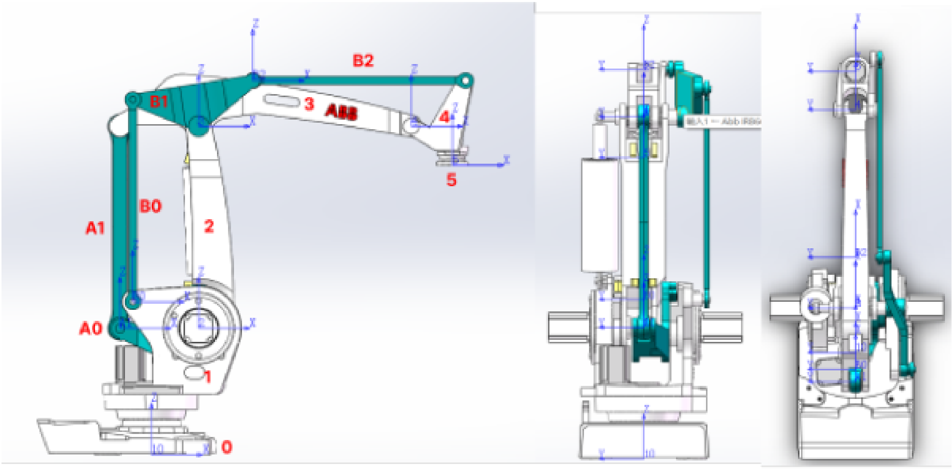
Classification of axes	Wrist: 4, 5
	Forearm: 3
	Upper arm: 2, A1
	Base: 0, 1, A0
Robot model list	YASKAWA MPL80II, FANUC M-410iB 140H

[robot]_algo example

```
{
    "algo_type": "SphericalWrist_FiveAxis",
    #DH parameters [meter]
    "dh": [ A, B, C, D, E, F ],
    "dhPassive": [ A1X, A1Z, B0X, B0Z, B2X, B2Z ],
    #joint limits [degree]
    "min_limits": [ J1 min, J2 min, J3 min, J4 min, J5 min ],
    "max_limits": [ J1 max, J2 max, J3 max, J4 max, J5 max ],
    "link3_dynamic_limits": [Min, Max],
    "link4_dynamic_limits": [Min, Max],
    #joint positions at standard pose [degree]
    "mastering_joints": [ J1, J2, J3, J4, J5 ],
    #joint orientations [0/1]
    "axis_flip": "J1J2J3J4J5",
    #robot base location [meter,meter,meter,degree,degree,degree]. If robot has not been moved, robot base frame equals to World frame.
    "robot_frame_in_robot_base_frame": [ x, y, z, rx, ry, rz ],
    #robot flange frame orientation [x/y/z]
    "flange_frame_orientation": "z"
}
```

Palletizer (four-axis palletizing robot)

Axes	Four
Configuration name	Palletizer
Description	Four-axis palletizing robot

DH parameters	 1311-最大工作范围 1350 E 260 K B2Z B2X F 1280 C B0X B0Z A1X 800 A 300 B
Reference frame and axis No.	
Correlation between link and reference frame	0: Reference frame 0, 1
	1: Reference frame 0, 1
	2: Reference frame 2
	3: Reference frame 3
	4: Reference frame 4
	5: Reference frame 5
	A0: Reference frame 2
	A1: Reference frame A1
	B0: Reference frame B0
	B1: Reference frame 3
	B2: Reference frame B2

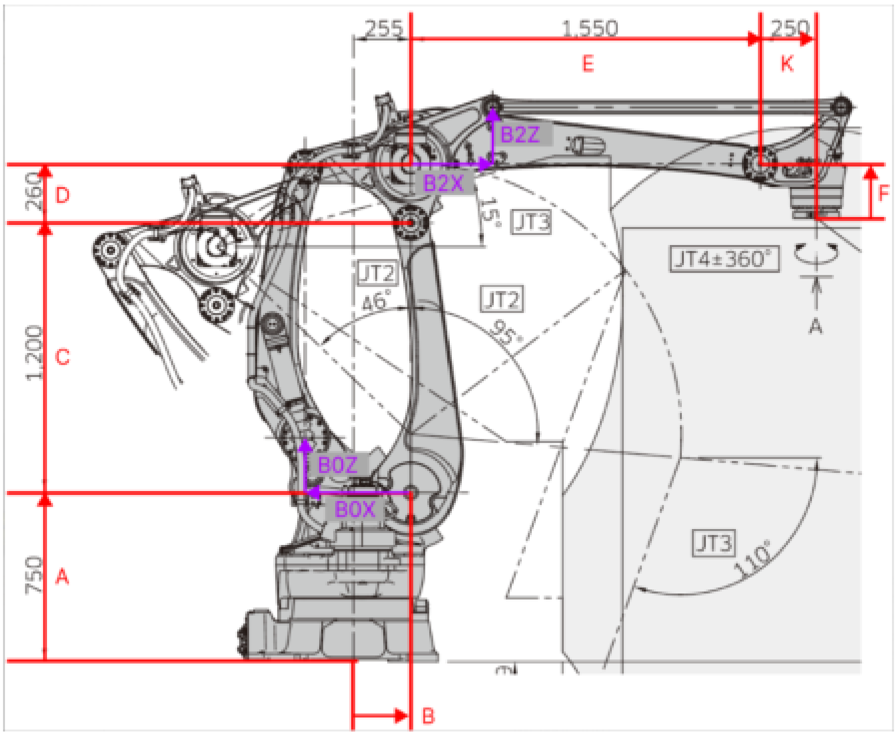
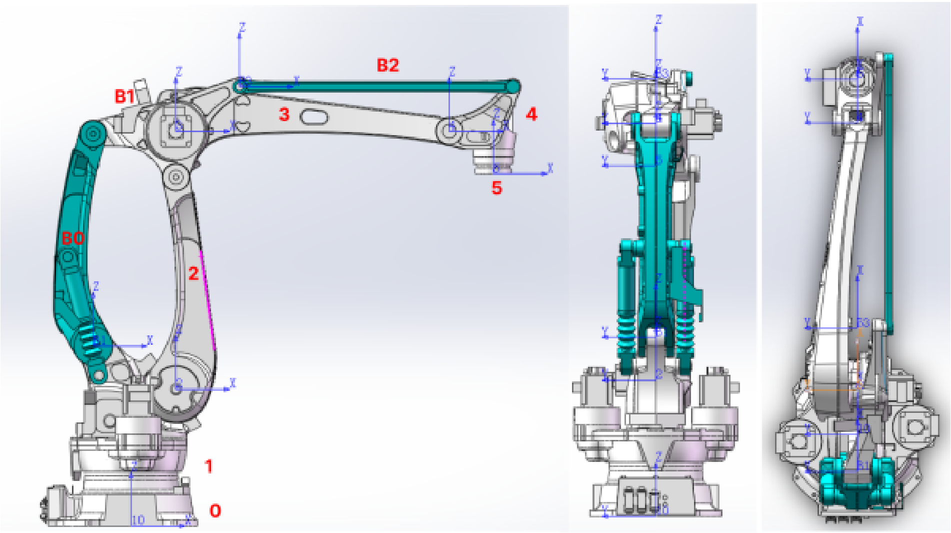
Classification of axes	Wrist: 4, 5
	Forearm: 3, B2, B1
	Upper arm: 2, A1, B0
	Base: 0, 1, A0
Robot model list	YASKAWA MPL80II, FANUC M-410iB 140H ABB IRB660 180 315, FANUC M-410iC 110, YASKAWA MPL800II

[robot]_algo example

```
{
    "algo_type": "Palletizer",
    #DH parameters [meter]
    "dh": [ A, B, C, E, K, F ],
    "dhPassive": [ A1X, A1Z, B0X, B0Z, B2X, B2Z ],
    #joint limits [degree]
    "min_limits": [ J1 min, J2 min, J3 min, J4 min ],
    "max_limits": [ J1 max, J2 max, J3 max, J4 max ],
    "link3_dynamic_limits": [Min, Max],
    #joint positions at standard pose [degree]
    "mastering_joints": [ J1, J2, J3, J4 ],
    #joint orientations [0/1]
    "axis_flip": "J1J2J3J4",
    #robot base location [meter,meter,meter,degree,degree,degree]. If robot has
not been moved, robot base frame equals to World frame.
    "robot_frame_in_robot_base_frame": [ x, y, z, rx, ry, rz ],
    #robot flange frame orientation [x/y/z]
    "flange_frame_orientation": "z"
}
```

Palletizer_KAWASAKI_CP180L_Like (four-axis palletizing robot with a positive Z-direction offset (D) on the J3)

Axes	Four
Configuration name	Palletizer_KAWASAKI_CP180L_Like
Description	Four-axis palletizing robot with a positive Z-direction offset (D) on the J3

DH parameters	 255 1,550 250 E K B2Z B2X 260 D 15° JT3 JT4±360° A JT2 46° JT2 95° 1,200 C F B0Z B0X JT3 110° 750 A B
Reference frame and axis No.	
Correlation between link and reference frame	0: Reference frame 0, 1
	1: Reference frame 0, 1
	2: Reference frame 2
	3: Reference frame 3
	4: Reference frame 4
	5: Reference frame 5
	B0: Reference frame B0
	B1: Reference frame 3
	B2: Reference frame B2

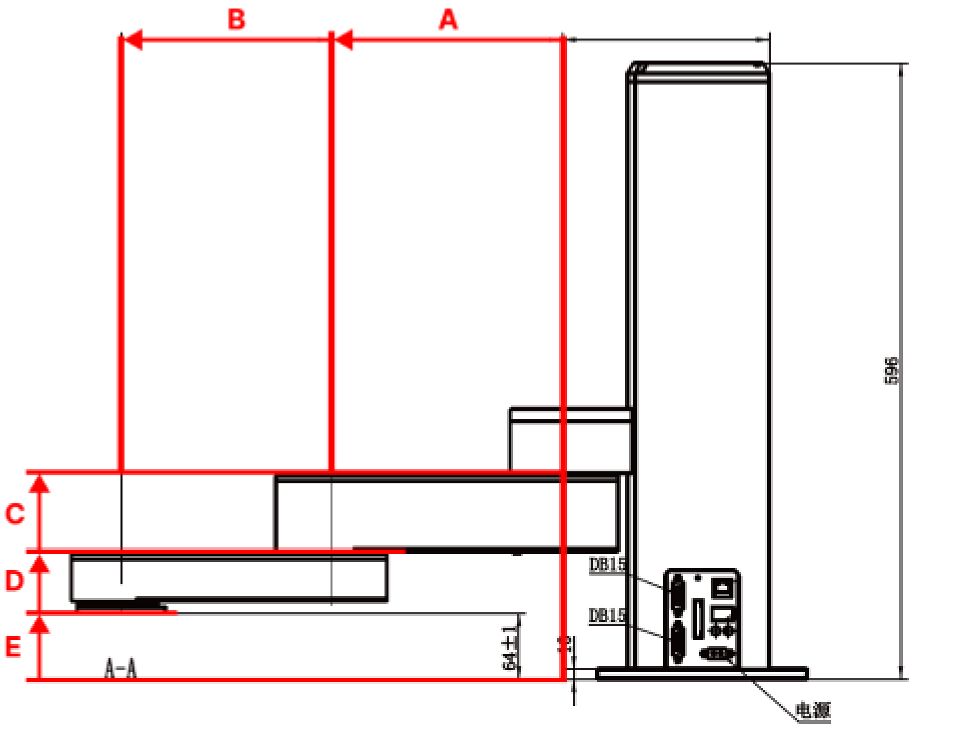
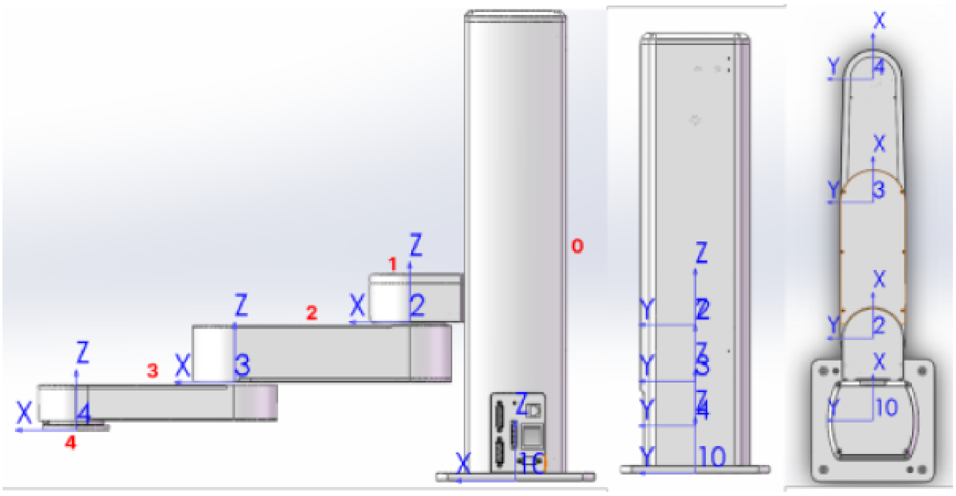
Classification of axes	Wrist: 4, 5
	Forearm: 3, B2, B1
	Upper arm: 2, B0
	Base: 0, 1
Robot model list	KAWASAKI CP130L, 180L, 300L, 500L, 700L, FANUC M-410iB 450

[robot]_algo example

```
{
"algo_type": "Palletizer_KAWASAKI_CP180L_Like",
    #DH parameters [meter]
    "dh": [ A, B, C, D, E, K, F ],
    "dhPassive": [ A1X, A1Z, B0X, B0Z, B2X, B2Z ],
    #joint limits [degree]
    "min_limits": [ J1 min, J2 min, J3 min, J4 min ],
    "max_limits": [ J1 max, J2 max, J3 max, J4 max ],
    "link3_dynamic_limits": [Min, Max],
    #joint positions at standard pose [degree]
    "mastering_joints": [ J1, J2, J3, J4 ],
    #joint orientations [0/1]
    "axis_flip": "J1J2J3J4",
    #robot base location [meter,meter,meter,degree,degree,degree]. If robot has
not been moved, robot base frame equals to World frame.
    "robot_frame_in_robot_base_frame": [ x, y, z, rx, ry, rz ],
    #robot flange frame orientation [x/y/z]
    "flange_frame_orientation": "z"
}
```

Scara_FourAxis_Translation_1st (four-axis palletizing robot with a positive Z-direction offset (D) on the J3)

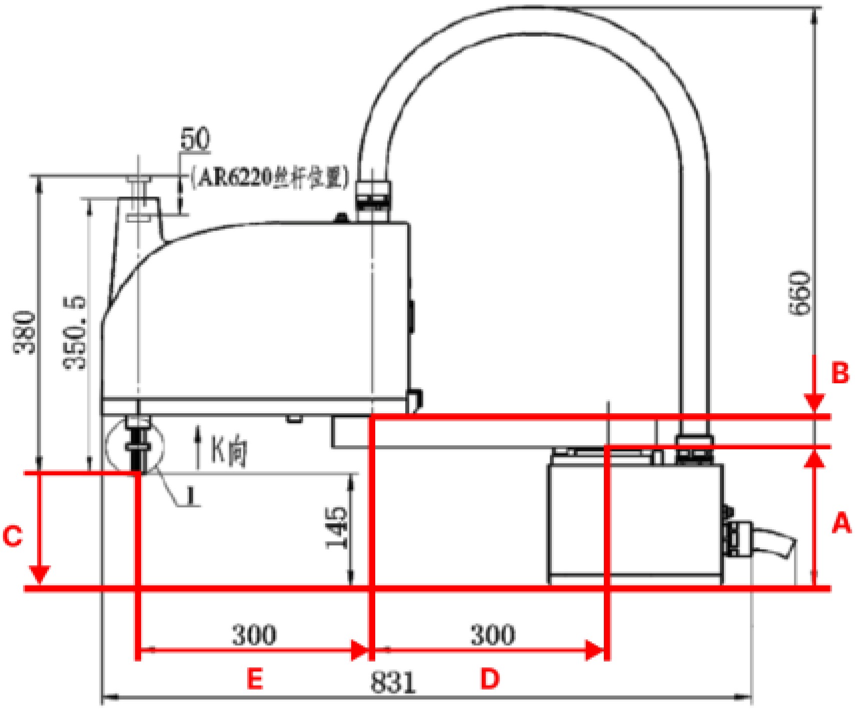
Axes	Four
Configuration name	Scara_FourAxis_Translation_1st
Description	Four-axis SCARA robots with axis 1 as translation axis

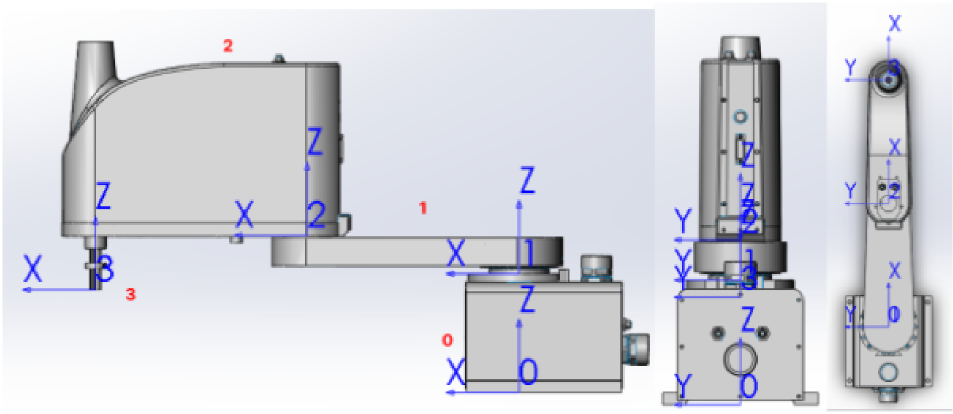
DH parameters	 B A C D E 596 DB15 DB15 64±1 A-A 电源
Reference frame and axis No.	
Correlation between link and reference frame	0: Reference frame 0, 1
	1: Reference frame 0, 1
	2: Reference frame 2
	3: Reference frame 3
	4: Reference frame 4
Classification of axes	Wrist: 4
	Forearm: 3
	Upper arm: 2
	Base: 0, 1
Robot model list	HITBOT Z-Arm 2442

[robot]_algo example

```
{
    "algo_type": "Scara_FourAxis_Translation_1st",
    #DH parameters [meter]
    "dh": [ A, B, C, D, E ],
    #joint limits [millimeter,degree,degree,degree]
    "min_limits": [ J1 min, J2 min, J3 min, J4 min ],
    "max_limits": [ J1 max, J2 max, J3 max, J4 max ],
    #joint positions at standard pose [meter,degree,degree,degree]
    "mastering_joints": [ J1, J2, J3, J4 ],
    #joint orientations [0/1]
    "axis_flip": "J1J2J3J4",
    #robot base location [meter,meter,meter,degree,degree,degree]. If robot has
not been moved, robot base frame equals to World frame.
    "robot_frame_in_robot_base_frame": [ x, y, z, rx, ry, rz ],
    #robot flange frame orientation [x/y/z]
    "flange_frame_orientation": "z"
}
```

Scara_FourAxis (four-axis SCARA robots)

Axes	Four
Configuration name	Scara_FourAxis
Description	Four-axis SCARA
DH parameters	 50 (AR6220丝杆位置) 380 350.5 660 K向 I 145 B A C 300 300 E 831 D

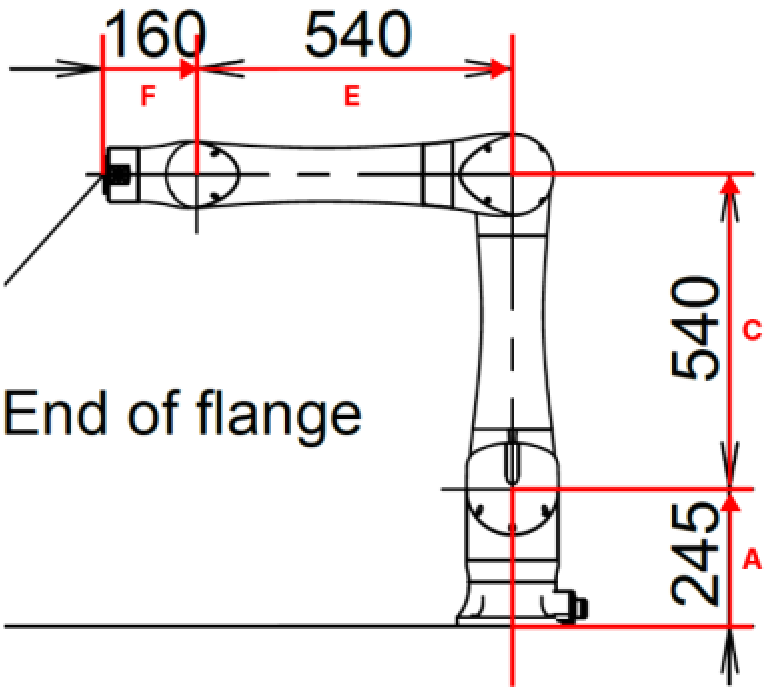
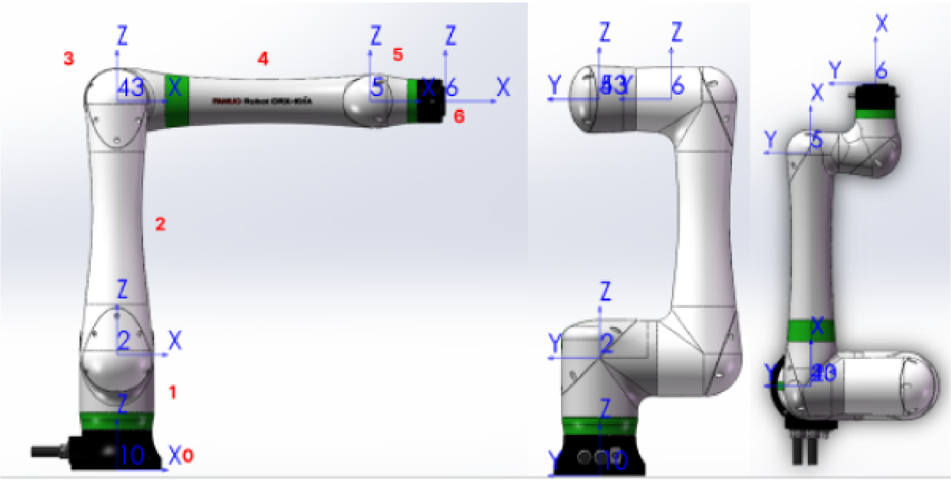
Reference frame and axis No.	 2 Z Z X 2 1 Z X 3 3 X 1 Z 0 X 0 Y Z Y Z Y 1 Z Y 0 X Y X Y X Y 0
Correlation between link and reference frame	0: Reference frame 0, 1
	1: Reference frame 0, 1
	2: Reference frame 2
	3: Reference frame 3
Classification of axes	Wrist: 3
	Forearm: 2
	Upper arm: 1
	Base: 0
Robot model list	ADTECH AR6215

[robot]_algo example

```
{
    "algo_type": "Scara_FourAxis",
    #DH parameters [meter]
    "dh": [ A, B, C, D, E ],
    #joint limits [degree,degree,millimeter,degree]
    "min_limits": [ J1 min, J2 min, J3 min, J4 min ],
    "max_limits": [ J1 max, J2 max, J3 max, J4 max ],
    #joint positions at standard pose [degree,degree,meter,degree]
    "mastering_joints": [ J1, J2, J3, J4 ],
    #joint orientations [0/1]
    "axis_flip": "J1J2J3J4",
    #robot base location [meter,meter,meter,degree,degree,degree]. If robot has not been moved, robot base frame equals to World frame.
    "robot_frame_in_robot_base_frame": [ x, y, z, rx, ry, rz ],
    #robot flange frame orientation [x/y/z]
    "flange_frame_orientation": "z"
}
```

5.7.2.1.2. Special Configurations

OffsetWrist_Y_SixAxis_FANUC_CRX_10IA_Like (there is a negative Y-direction offset (L) on the J5)

Axes	Six
Configuration name	OffsetWrist_Y_SixAxis_FANUC_CRX_10IA_Like
Description	There is a negative Y-direction offset (L) on the J5
DH parameters	 160 540 F E 540 C End of flange 245 A
Reference frame and axis No.	 Z 3 4 Z 5 Z 43 X 5 X 6 X 6 2 Z 2 X 1 Z 10 X0 Z Z Y 53 Y 6 Z Y 2 Z 10 X Y 6 X Y 5 X Y 40

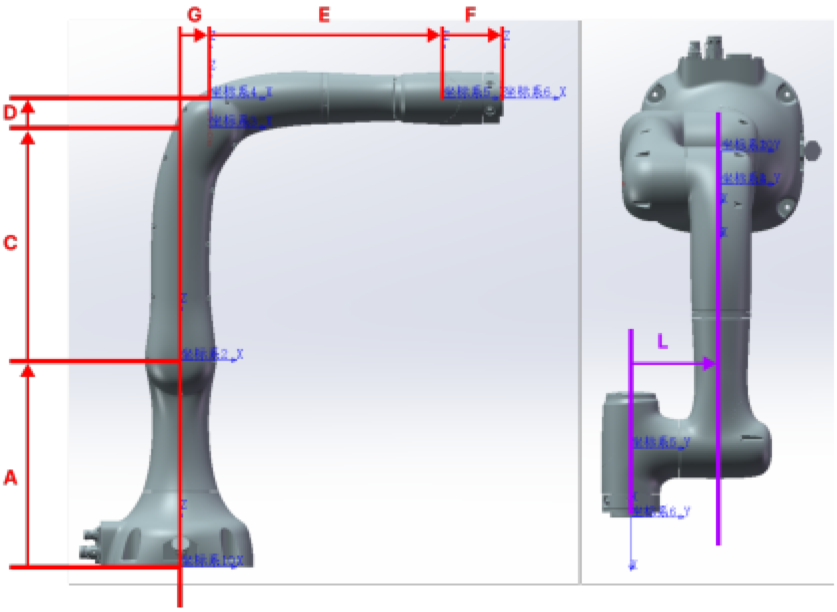
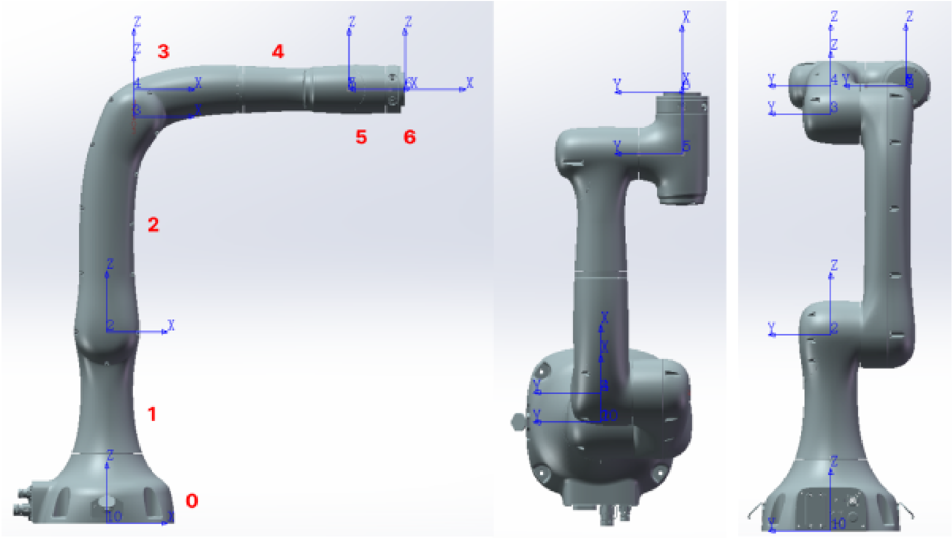
Correlation between link and reference frame	0: Reference frame 0, 1
	1: Reference frame 0, 1
	2: Reference frame 2
	3: Reference frame 3, 4
	4: Reference frame 3, 4
	5: Reference frame 5
	6: Reference frame 6
Classification of axes	Wrist: 5, 6
	Forearm: 3, 4
	Upper arm: 2
	Base: 0, 1
Robot model list	FANUC CRX-10iA, FANUC CRX-10iAL, YASKAWA HC10

[robot]_algo example

```
{
    "algo_type": "OffsetWrist_Y_SixAxis_FANUC_CRX_10IA_Like",
    #DH parameters [meter]
    "dh": [ A, C, E, F, -L ],
    #joint limits [degree]
    "min_limits": [ J1 min, J2 min, J3 min, J4 min, J5 min, J6 min ],
    "max_limits": [ J1 max, J2 max, J3 max, J4 max, J5 max, J6 max ],
    #joint positions at standard pose [degree]
    "mastering_joints": [ J1, J2, J3, J4, J5, J6 ],
    #joint orientations [0/1]
    "axis_flip": "J1J2J3J4J5J6",
    #robot base location [meter,meter,meter,degree,degree,degree]. If robot has
not been moved, robot base frame equals to World frame.
    "robot_frame_in_robot_base_frame": [ x, y, z, rx, ry, rz ],
    #robot flange frame orientation [x/y/z]
    "flange_frame_orientation": "z"
}
```

OffsetWrist_Y_SixAxis_Rokae_SR4_Like (there is a positive X-direction offset (G) on the J2, and a negative Y-direction offset (L) on the J5)

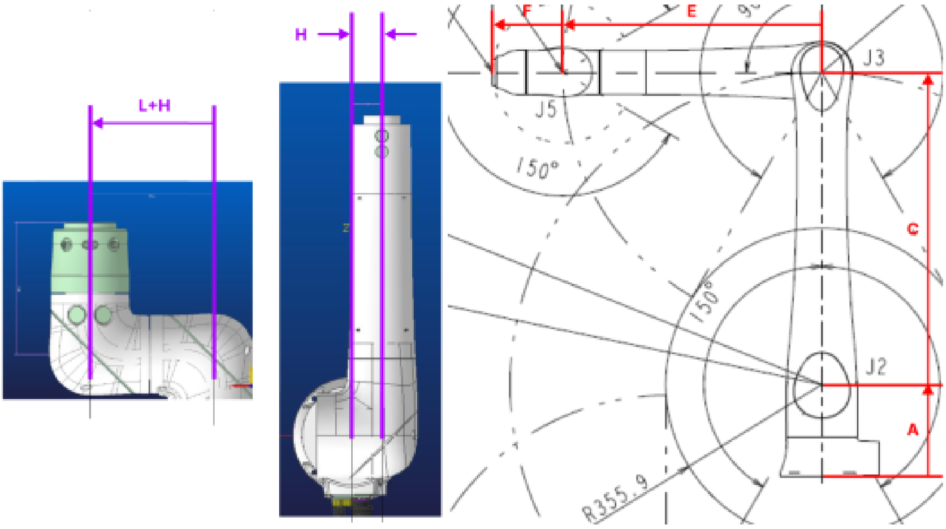
Axes	Six
Configuration name	OffsetWrist_Y_SixAxis_Rokae_SR4_Like
Description	There is a positive X-direction offset (G) on the J2, and a negative Y-direction offset (L) on the J5

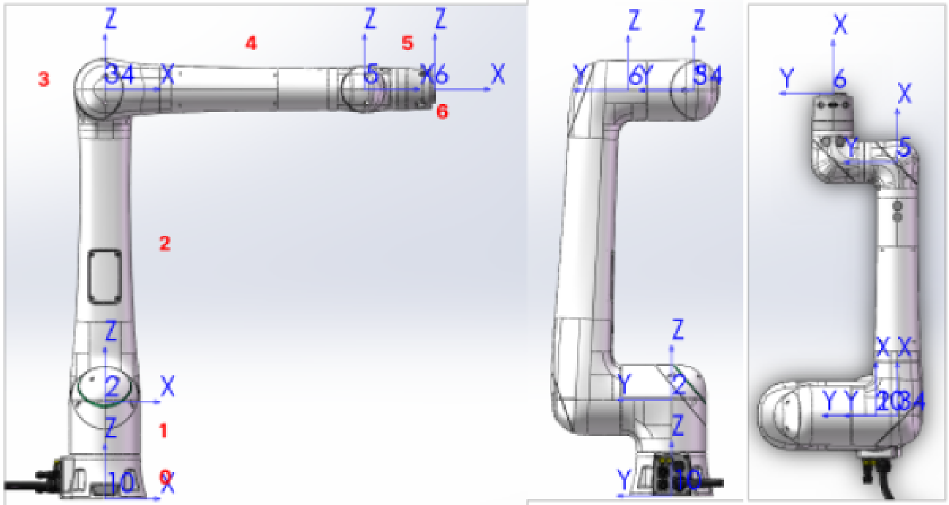
DH parameters	
Reference frame and axis No.	
Correlation between link and reference frame	0: Reference frame 0, 1
	1: Reference frame 0, 1
	2: Reference frame 2
	3: Reference frame 3
	4: Reference frame 4
	5: Reference frame 5
	6: Reference frame 6
Classification of axes	Wrist: 5, 6
	Forearm: 3, 4
	Upper arm: 2
	Base: 0, 1
Robot model list	Rokae SR3, SR4

[robot]_algo example

```
{
    "algo_type": "OffsetWrist_Y_SixAxis_ROKAE_SR4_Like",
    #DH parameters [meter]
    "dh": [ A, C, D, G, E, F, -L ],
    #joint limits [degree]
    "min_limits": [ J1 min, J2 min, J3 min, J4 min, J5 min, J6 min ],
    "max_limits": [ J1 max, J2 max, J3 max, J4 max, J5 max, J6 max ],
    #joint positions at standard pose [degree]
    "mastering_joints": [ J1, J2, J3, J4, J5, J6 ],
    #joint orientations [0/1]
    "axis_flip": "J1J2J3J4J5J6",
    #robot base location [meter,meter,meter,degree,degree,degree]. If robot has
not been moved, robot base frame equals to World frame.
    "robot_frame_in_robot_base_frame": [ x, y, z, rx, ry, rz ],
    #robot flange frame orientation [x/y/z]
    "flange_frame_orientation": "z"
}
```

OffsetWrist_Y_SixAxis_DENSO_COBOTTA_PRO_1300_Like (there is a negative Y-direction offset (H) on the J3, and a positive Y-direction offset (L) on the J5)

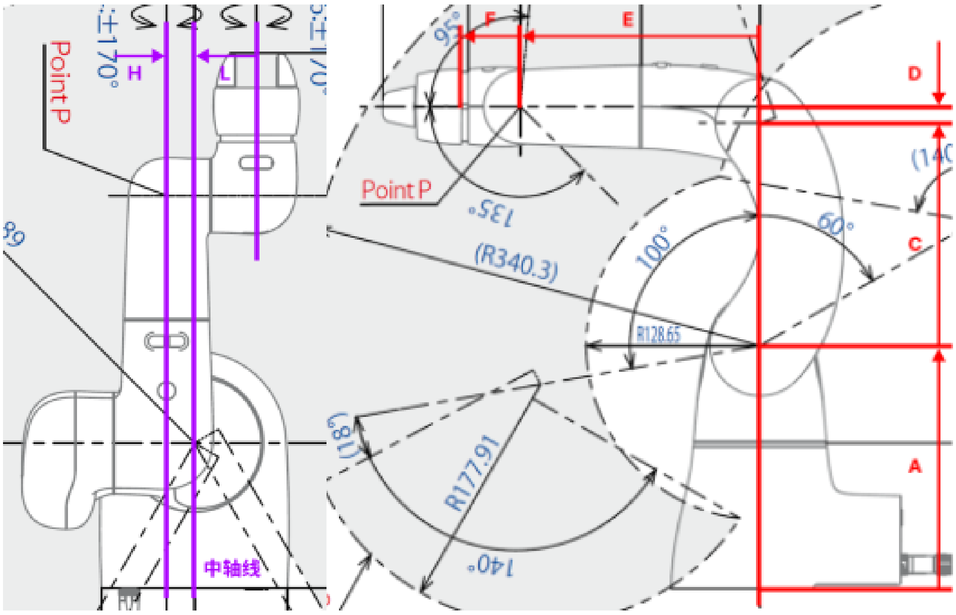
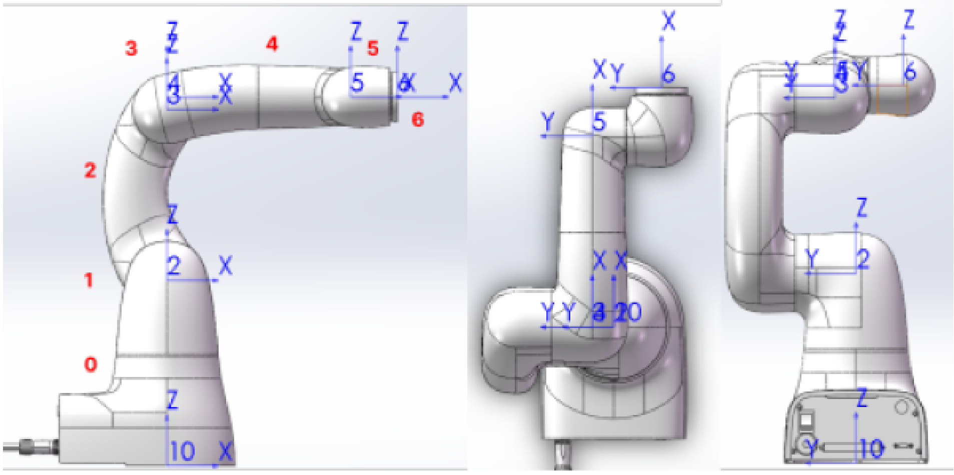
Axes	Six
Configuration name	OffsetWrist_Y_SixAxis_DENSO_COBOTTA_PRO_1300_Like
Description	There is a negative Y-direction offset (H) on the J3, and a positive Y-direction offset (L) on the J5
DH parameters	 L+H H F E J3 J5 150° 150° C J2 A R355.9

Reference frame and axis No.	 Z 4 5 3 34 X 6 2 1 10 0 Y 6Y 54
Correlation between link and reference frame	0: Reference frame 0, 1
	1: Reference frame 0, 1
	2: Reference frame 2
	3: Reference frame 3, 4
	4: Reference frame 3, 4
	5: Reference frame 5
	6: Reference frame 6
Classification of axes	Wrist: 5, 6
	Forearm: 3, 4
	Upper arm: 2
	Base: 0, 1
Robot model list	Denso COBOTTA PRO 900, COBOTTA PRO 1300

[robot]_algo example

```
{
    "algo_type": "OffsetWrist_Y_SixAxis_DENSO_COBOTTA_PRO_1300_Like",
    #DH parameters [meter]
    "dh": [ A, C, E, F, -H, L ],
    #joint limits [degree]
    "min_limits": [ J1 min, J2 min, J3 min, J4 min, J5 min, J6 min ],
    "max_limits": [ J1 max, J2 max, J3 max, J4 max, J5 max, J6 max ],
    #joint positions at standard pose [degree]
    "mastering_joints": [ J1, J2, J3, J4, J5, J6 ],
    #joint orientations [0/1]
    "axis_flip": "J1J2J3J4J5J6",
    #robot base location [meter,meter,meter,degree,degree,degree]. If robot has
not been moved, robot base frame equals to World frame.
    "robot_frame_in_robot_base_frame": [ x, y, z, rx, ry, rz ],
    #robot flange frame orientation [x/y/z]
    "flange_frame_orientation": "z"
}
```

OffsetWrist_Y_SixAxis_DENSO_COBOTTA_Like (there are positive Y-direction and Z-direction offsets (H, D) on the J3, and a negative Y-direction offset (L) on the J5)

Axes	Six
Configuration name	OffsetWrist_Y_SixAxis_DENSO_COBOTTA_Like
Description	There are positive Y-direction and Z-direction offsets (H, D) on the J3, and a negative Y-direction offset (L) on the J5
DH parameters	 Point P ±170° H L 中轴线 95° F E D Point P 135° (R340.3) (140 60° C 100° R128.65 (81) R177.91 A 140°
Reference frame and axis No.	

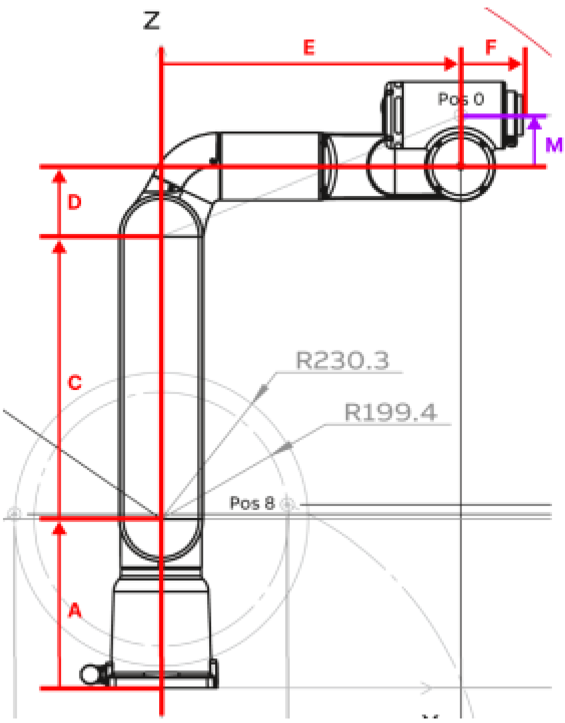
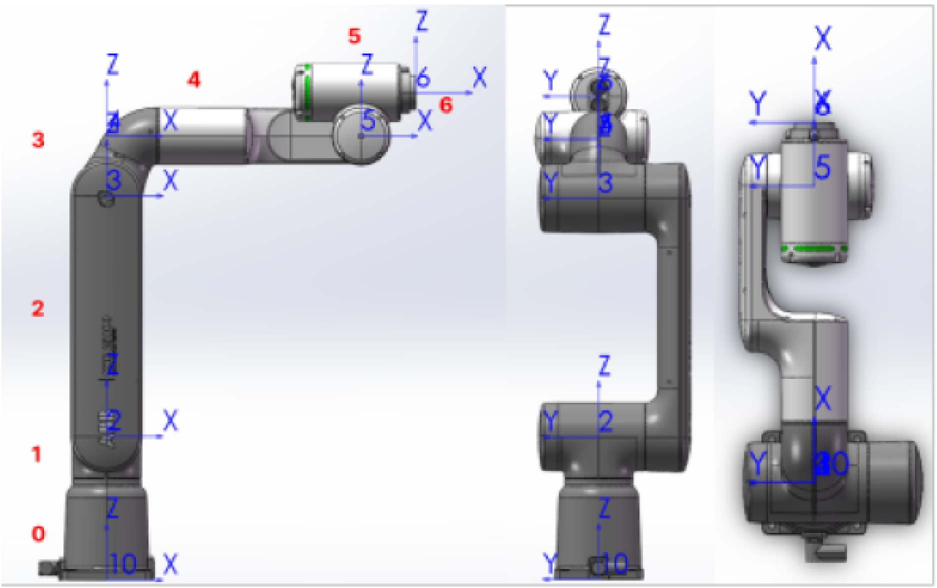
Correlation between link and reference frame	0: Reference frame 0, 1
	1: Reference frame 0, 1
	2: Reference frame 2
	3: Reference frame 3, 4
	4: Reference frame 3, 4
	5: Reference frame 5
	6: Reference frame 6
Classification of axes	Wrist: 5, 6
	Forearm: 3, 4
	Upper arm: 2
	Base: 0, 1
Robot model list	Denso COBOTTA

[robot]_algo example

```
{
    "algo_type": "OffsetWrist_Y_SixAxis_DENSO_COBOTTA_Like",
    #DH parameters [meter]
    "dh": [ A, C, D, E, F, H, -L ],
    #joint limits [degree]
    "min_limits": [ J1 min, J2 min, J3 min, J4 min, J5 min, J6 min ],
    "max_limits": [ J1 max, J2 max, J3 max, J4 max, J5 max, J6 max ],
    #joint positions at standard pose [degree]
    "mastering_joints": [ J1, J2, J3, J4, J5, J6 ],
    #joint orientations [0/1]
    "axis_flip": "J1J2J3J4J5J6",
    #robot base location [meter,meter,meter,degree,degree,degree]. If robot has
not been moved, robot base frame equals to World frame.
    "robot_frame_in_robot_base_frame": [ x, y, z, rx, ry, rz ],
    #robot flange frame orientation [x/y/z]
    "flange_frame_orientation": "z"
}
```

OffsetWrist_Z_SixAxis_ABB_CRB15000_Like (there is a positive Z-direction offset (M) on the J5)

Axes	Six
Configuration name	OffsetWrist_Z_SixAxis_ABB_CRB15000_Like
Description	There is a positive Z-direction offset (M) on the J5

DH parameters	 Z E F Pos 0 M D C R230.3 R199.4 Pos 8 A
Reference frame and axis No.	
Correlation between link and reference frame	0: Reference frame 0, 1
	1: Reference frame 0, 1
	2: Reference frame 2
	3: Reference frame 3, 4
	4: Reference frame 3, 4
	5: Reference frame 5
	6: Reference frame 6

Classification of axes	Wrist: 5, 6
	Forearm: 3, 4
	Upper arm: 2
	Base: 0, 1
Robot model list	ABB CRB15000-5/0.95

[robot]_algo example

```
{
    "algo_type": "OffsetWrist_Z_SixAxis_ABB_CRB15000_5_0_95_Like",
    #DH parameters [meter]
    "dh": [ A, C, D, E, F, M ],
    #joint limits [degree]
    "min_limits": [ J1 min, J2 min, J3 min, J4 min, J5 min, J6 min ],
    "max_limits": [ J1 max, J2 max, J3 max, J4 max, J5 max, J6 max ],
    #joint positions at standard pose [degree]
    "mastering_joints": [ J1, J2, J3, J4, J5, J6 ],
    #joint orientations [0/1]
    "axis_flip": "J1J2J3J4J5J6",
    #robot base location [meter,meter,meter,degree,degree,degree]. If robot has
not been moved, robot base frame equals to World frame.
    "robot_frame_in_robot_base_frame": [ x, y, z, rx, ry, rz ],
    #robot flange frame orientation [x/y/z]
    "flange_frame_orientation": "z"
}
```

5.7.2.2. Description of [robot]_algo.json File

[robot]_algo.json is a robot configuration file which records the parameter information of the robot. It defines the robot type, DH parameters, range of motion, home position of joints, rotation of joints, etc.

The code fragment below is extracted from [robot]_algo.json file as an example.

```
{
    "algo_type": "SphericalWrist_SixAxis",
    #DH parameters [meter]
    "dh": [ A, B, C, D, E, F ],
    "dhPassive": [ A1X, A1Z, B0X, B0Z, B2X, B2Z ],
    #joint limits [degree]
    "min_limits": [ J1 min, J2 min, J3 min, J4 min, J5 min, J6 min ],
    "max_limits": [ J1 max, J2 max, J3 max, J4 max, J5 max, J6 max ],
    "link3_dynamic_limits": [Min, Max],
    #joint positions at standard pose [degree]
    "mastering_joints": [ J1, J2, J3, J4, J5, J6 ],
    #joint orientations [0/1]
    "axis_flip": "J1J2J3J4J5J6",
    #robot base location [meter,meter,meter,degree,degree,degree]. If robot has
not been moved, robot base frame equals to World frame.
    "robot_frame_in_robot_base_frame": [ x, y, z, rx, ry, rz ],
    #robot flange frame orientation [x/y/z]
    "flange_frame_orientation": "z"
}
```

[robot]_algo.json File

Robot Configurations

Parameter	algo_type
Example	"algo_type": "SphericalWrist_SixAxis"
Description	The configuration type of the robot. The robot models are divided into 15 categories in Mech-Viz 1.8.0. For more information about the robot configuration, please refer to Robot Configurations.

Robot DH Parameter

Parameter	dh
Example	"dh": [0.465, 0.2, 0.88, 0.21, 1.100, 0.225]
Description	The DH parameters of the active joints of the robot. For more information about the definition and calculation methods, please refer to the illustrations in the Robot Configurations.

Parameter	dhPassive
Description	The DH parameters of the passive joints of the robot. For more information about the definition and calculation methods, please refer to the illustrations in the Robot Configurations. dhPassive parameters are not divided individually in line with the robot configurations, but are all in the form of [A1X, A1Z, B0X, B0Z, B2X, B2Z] with zero fill in the blank space.

Robot Joint Motion Range

Parameter	minlimits
Example	"min_limits": [-160, -65, -77, -360, -125, -360]
Description	This parameter defines the minimum motion range of axes of the robot.

Parameter	maxlimits
Example	"max_limits": [160, 120, 90 , 360, 125, 360]
Description	This parameter defines the maximum motion range of axes of the robot.

- For most of the robots, you can obtain the accurate motion range of each axis of the robot from the robot’s product manual. The ranges of each axis on the product manuals of Nachi, Hyundai, and other brands are inaccurate and cannot be used directly.
- For every joint, the two parameters, minlimits and maxlimits, are related to axis_flip. Under certain circumstances, the two parameters and their negative/positive values should be reversed if the value of axis_flip changes. For example, given that "axis_flip": "101010", "link3_dynamic_limits": [-204,80] , if "axis_flip": "100010", then "link3_dynamic_limits": [-80， 240].

Parameter	link3_dynamic_limits
Example	"link3_dynamic_limits": [-204,80]
Description	The parameter of dynamic limits.

Parameter	link4_dynamic_limits
Description	The parameter of dynamic limits.

Pose of the Robot with the Angle of Each Axis Being 0°

Parameter	mastering_joints
Example	"mastering_joints": [-90, 0, 0, 0, 0, 90]
Description	This parameter defines the pose of the robot with the angle of each axis being 0°. When the value of parameters are all set to 0, the robot pose is the same as the one during modeling.

For some robots such as KUKA robots of all series, when the value of J2 and J3 are set to 0, its pose is not same as the default pose in Mech-Viz, and therefore the value of J2 and J3 in mastering_joints should be adjusted.

Rotation Directions of the Robot Axes

Parameter	axis_flip
Example	"axis_flip": "101010"
Description	This parameter defines the rotation directions of the robot axes.

- After modeling the robot, you need to check whether the rotation orientation of each axis displayed in Mech-Viz is the same with that in the robot simulation software or real robot. If there is difference, you need to adjust the parameter.
- This attribute will affect other attributes such as minlimits, maxlimits, link3_dynamic_limits, link4_dynamic_limits, etc.

Robot Reference Frame Parameter

Parameter	robot_frame_in_robot_base_frame
Description	This parameter defines the position of the robot reference frame with respect to the robot foundation.

- For the robots of FANUC and YASKAWA brands, the robot reference frame is one dh1 above the foundation, so this parameter should be [0, 0, dh1, 0, 0, 0]. That is, the robot reference frame is located at position dh1 in the +Z direction of the robot base frame.

- For the robots of KAWASAKI brand facing in the +Y direction, this parameter should be [0, 0, 0, 0, 0, -90]. That is, with respect to the robot base frame, the robot reference frame goes in a counterclockwise 90° rotation around the Z-axis of the base frame.
- For YASKAWA MPL3500 Wall robot which has a special robot reference frame, adjust this parameter according to actual situation, and finally it should be [0, 0, 0.8, 0, -90 , 90]. That is, the robot reference frame is one dh1 higher above the robot base frame, and rotates around its Y-axis and Z-axis.

Robot Flange Frame Orientation

Parameter	"robot_flange_orientation": "x/y/z"
Example	"robot_flange_orientation": "x"
Description	For the robots of some brands such as TURIN, the Z-axis of their flange frames is not pointing outward relative to the flange surface as usual. Instead, the X-axis of the flange reference frame points outward. By configuring the parameter as "robot_flange_orientation": "x", you can change the axis pointing outward to the X-axis in the flange reference frame. When this parameter is not specified, the axis pointing outward defaults to the Z-axis.

Notes When Using a Newly Created Robot Model

When using a newly created robot model, please make sure that the settings of the following parameters correspond with the real robot:

- axis_flip

  Compare the axis rotation direction of the real robot with that of the simulated robot in Mech-Viz to see if they agree.

- dh

  You can check if the pose of the simulated robot in Mech-Viz corresponds to that of the real robot by using the following two methods.

  - Synchronize the JPs to compare the TCP.
  - Synchronize the TCP to compare JPs.

    The closer the DH parameters of the real robot are to the theoretical values in the parameter file, the more precise the robot is. An error within 1mm is generally considered acceptable.

- mastering_joints

  Check if the pose of the simulated robot in Mech-Viz corresponds to that of the real robot.

  Please pay attention to axis-1, axis-4, and axis-6 while checking.

- min_limits/max_limit

  Check if the pose of the simulated robot in Mech-Viz corresponds to that of the real robot.

  The soft limits of the robot must not exceed its hard limits.

5.7.2.3. Description of [robot]_profile.json File

[robot]_profile.json records some basic information of the robot, including the robot name, payload, reach, number of axes, home position, and maximum velocity.

The code fragment below is extracted from [robot]_profile.json file as an example.

```
{
     #information in the library
     "robot_display_names": ["name1","name2","name3"],
     "payload": [40,40,60],
     "reach": 2.55,
     "axes": 6,
     #robot home gesture
     "home_jps":[0,90,0,0,90,0],
     #singularity detection
     "max_tcp_vel": 7,
     "max_tcp_acc": 2,
     "max_joint_vel": 500,
     "max_joint_acc": 100
}
```

Information Displayed on the Robot Model Card

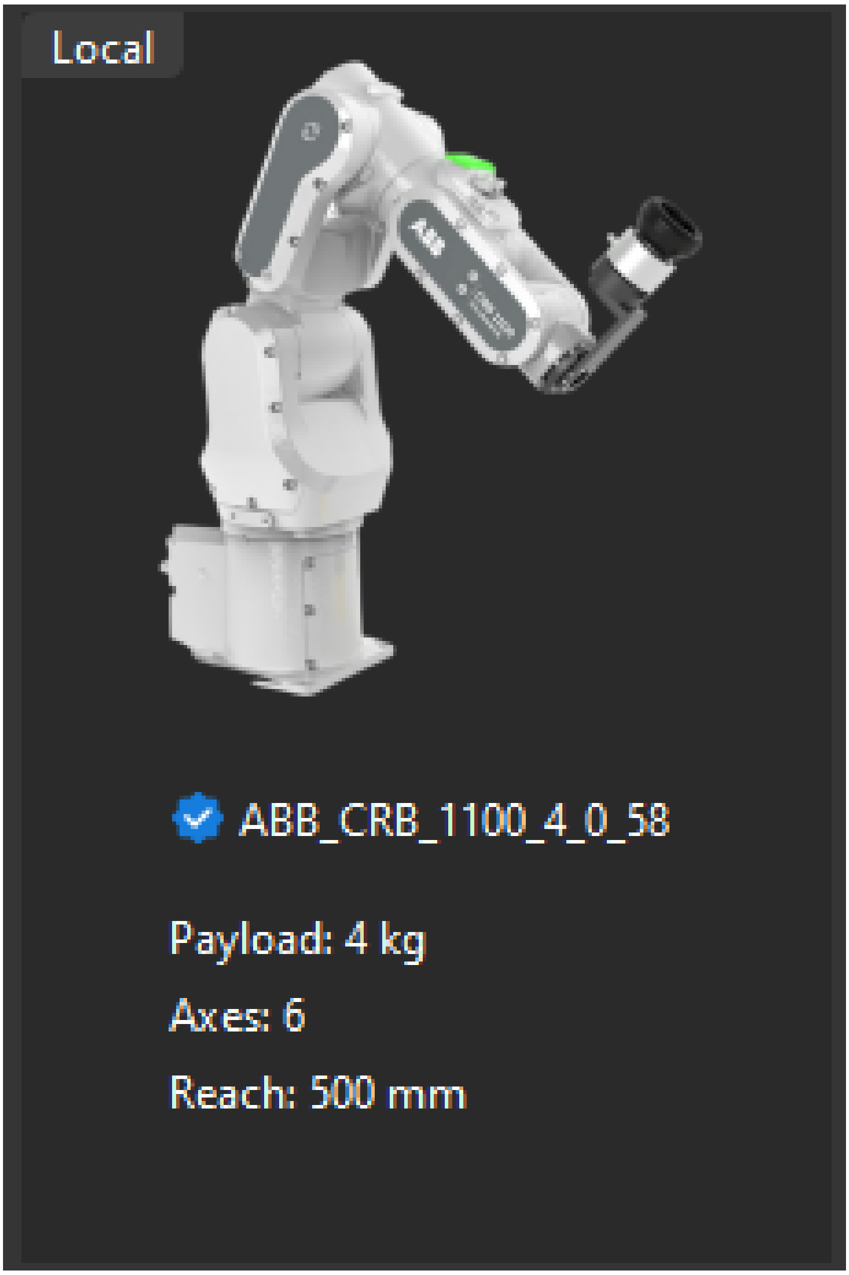
Parameter	robot_display_names
Example	"robot_display_names": ABB_CRB_1100_4_0_475
Description	Name of the robot model.

Parameter	"payload": [payload]
Example	"payload": [4]
Description	Rated payload (kg).

Parameter	reach
Example	"reach":0.475
Description	Reachable radius (m).

Parameter	axes
Example	"axes":6

Description	Number of movable axes.



ABB_CRB_1100_4_0_58

Payload: 4 kg

Axes: 6

Reach: 500 mm

Initial Pose of Robot

Parameter	home_jps
Example	"home_jps":[0,90,0,0,90,0]
Description	Default pose (unit: °) of the robot model in the software.

Singularity Detection

The following parameters are used for singularity detection.

Parameter	max_tcp_vel
Example	"max_tcp_vel": 7
Description	Maximum TCP velocity (m/s).

Parameter	max_tcp_acc
Example	"max_tcp_acc": 2
Description	Maximum TCP acceleration (m/s²).

Parameter	max_joint_vel
Example	"max_joint_vel": 500
Description	Maximum joint velocity (°/s).

Parameter	max_joint_acc
Example	"max_joint_acc": 100
Description	Maximum joint acceleration (°/s²).

5.7.3. Others

5.7.3.1. Options

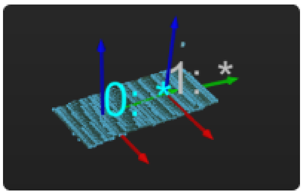
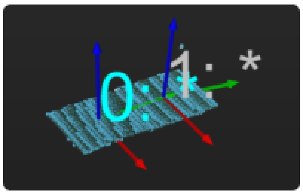
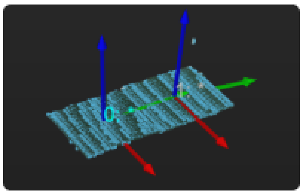
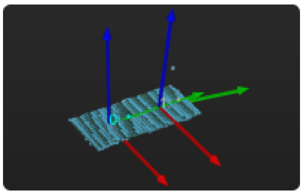
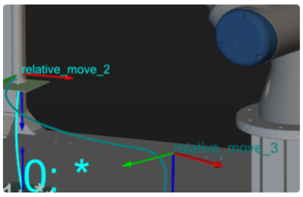
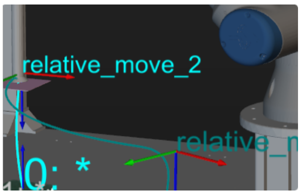
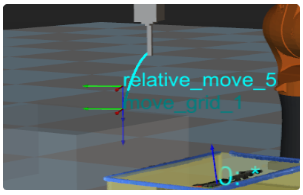
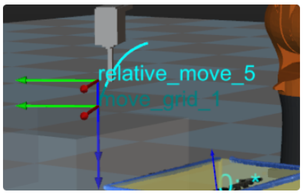
You can configure the settings below in the Options window.

- Common
  - Use custom resource path
  - Parameters related to Log
  - Parameters related to the 3D simulation area
  - Close window settings
  - Language, unit of length, and unit of angle
- Basic Settings
  - Enable Real Robot Pause
  - Ground Style (in the 3D simulation area)

Options with * will only take effect after the software is restarted.

5.7.3.2. Display Settings

In Display Settings, more details can be seen to debug the project.

Parameter	Value	Illustration
Vision Pose Tag Size	0.05	
	0.07	
Vision Pose Dragger Size	0.05	
	0.07	
Path Text Size	0.02	
	0.04	
Path Dragger Size	0.05	
	0.1	

6. Appendix

6.1. Non-standard IPC Setup

Mech-Mind allows you to use your own computer or laptop (“non-standard IPC” for short) as the industrial personal computer (IPC) for installing and running Mech-Mind’s software. Please note that Mech-Mind does not guarantee that the functions and performance of a non-standard IPC are the same as those of a standard IPC.

To ensure normal operation of Mech-Mind’s software on a non-standard IPC, you need to complete the settings in this chapter.

> The standard IPC provided by Mech-Mind is already fully set up. It provides the standard operation environment for Mech-Mind’s software and therefore can maximize the function and performance of the software. Therefore, it is recommended to use the standard IPC provided by Mech-Mind. For instructions about the Mech-Mind’s standard IPC, please refer to the Mech-Mind IPC User Manual.

Check System Requirements

Please confirm that the IPC running Mech-Vision and Mech-Viz or Mech-DLK meets the system configuration requirements (including authorization requirements).

- System Requirements for Mech-Vision and Mech-Viz
- System Requirements for Mech-DLK
- System Requirements for Mech-DLK SDK

Check Graphics Card Driver

To carry out deep learning training or inference, Mech-DLK and Mech-Vision rely on NVIDIA’s high hash rate and require that the GPU driver meets the version requirement.

- For Mech-Vision 1.7.0 and above, as well as for Mech-DLK 2.3 and above, the graphics card driver version should be no lower than 472.50.
- For Mech-Vision 1.6.1 and 1.6.2, the graphics card driver version should be no lower than 471.68.

Check Driver Version

Follow the steps below to check the driver’s version.

1. Type **NVIDIA control panel** in the search box of the taskbar to open the panel.
2. Check the graphics card type and driver’s version on the homepage.
    - If the driver version is lower than 472.50, please update graphics card driver.
    - If the driver version is higher than 472.50, no update is required.

Update Graphics Card Driver

Follow the steps below to update the graphics card driver.

1. Go to the NVIDIA Download Drivers page.
2. Choose the IPC’s graphics series in the **Product Series** drop-down menu, and choose the graphics type in the **Product** drop-down menu. Make sure the other options are correct and then click **[Search]**.
3. Click **[Download]** on the page.
4. Click **[Download]** again on the **Download** page to start downloading the driver.
5. When the download is completed, double-click the EXE file to install the driver.

> After the driver is successfully installed, please restart Mech-Vision to ensure normal operation.

Check CPU

Mech-Vision and Mech-Viz require that the CPU supports AVX2 instruction set.

Click the link to confirm whether your CPU supports the AVX2 instruction set.

Set IPC Region and Language

The **region** and **language** of the IPC should match. Otherwise, garbled texts will appear in the log of Mech-Vision or Mech-Viz.

Set IPC Region

Please follow the steps below to set the IPC region:

On the IPC, go to Settings › Time & language › Regions. In **Country or region**, choose the country or region where you are located.

Set IPC Language

The IPC language should be set according to the IPC region. Follow the steps below to complete the setting:

1. On the IPC, go to Settings › Time & language › Language. Choose the language in **Windows display language**. If your language is not listed in **Windows display language**, proceed to the next step.
2. In **Preferred languages**, click **Add a preferred language** to open the window **Choose a language to install**.
3. Type a language or country/region name in the search box, select the language that you wish to install, and click **[Next]** to open the window **Install language features**.
4. Check the option **Set as my Windows display language**, and then click **[Install]** to install the language and set it as the display language for your IPC.

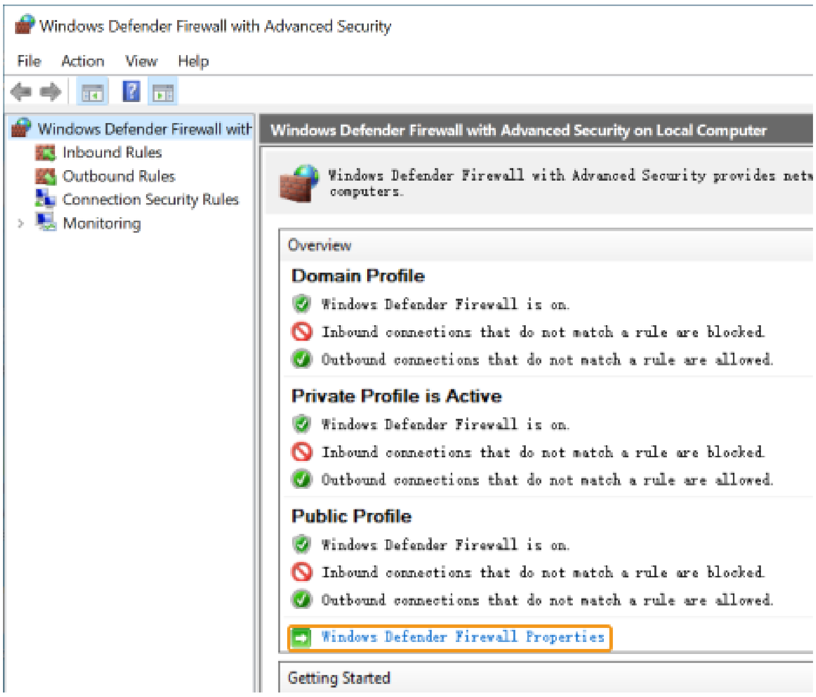
> If the option **Set as my Windows display language** was not checked while you installed the language, you need to manually switch the language in **Windows display language**.

Turn off Windows Defender Firewall

The software communicates with external devices connected to the IPC (such as Mech-Eye Industrial 3D Camera, robot controller, and PLC) through Ethernet ports, whereas Windows Defender Firewall might block such communication. Therefore, it is necessary to turn off Windows

Defender Firewall's protection for these Ethernet ports.

1. On the IPC, go to Settings › Control panel › System and Security › Windows Defender Firewall › Advanced settings.
2. In the pop-up window, click **Windows Defender Firewall Properties**.



Windows Defender Firewall with Advanced Security

Windows Defender Firewall with Advanced Security on Local Computer

Windows Defender Firewall with Advanced Security provides netw computers.

Overview

Domain Profile

- Windows Defender Firewall is on.
- Inbound connections that do not match a rule are blocked.
- Outbound connections that do not match a rule are allowed.

Private Profile is Active

- Windows Defender Firewall is on.
- Inbound connections that do not match a rule are blocked.
- Outbound connections that do not match a rule are allowed.

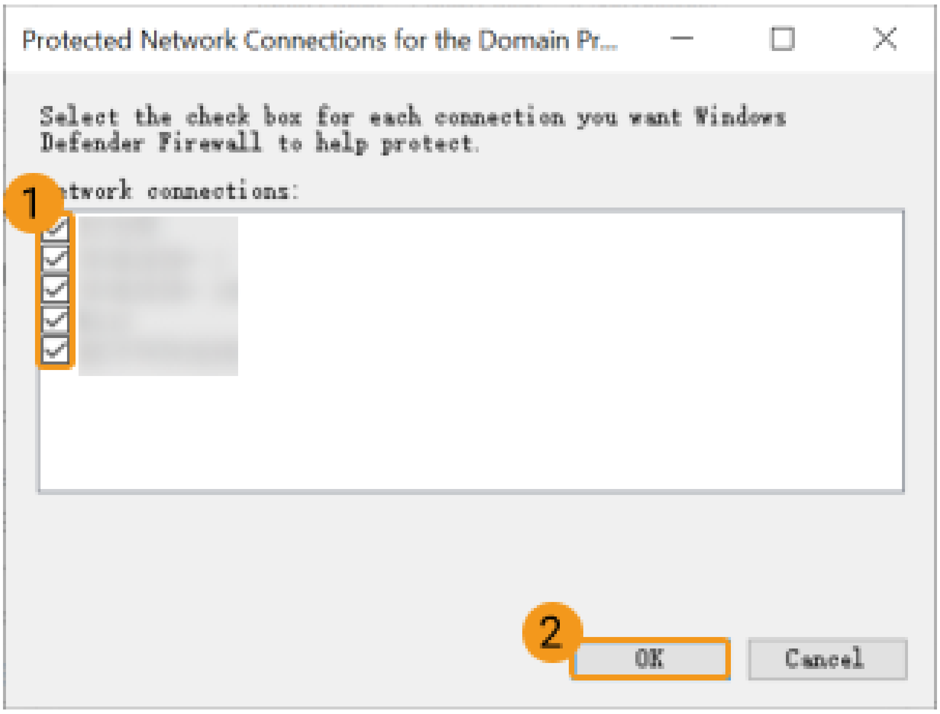
Public Profile

- Windows Defender Firewall is on.
- Inbound connections that do not match a rule are blocked.
- Outbound connections that do not match a rule are allowed.

Windows Defender Firewall Properties

Getting Started

3. Under the **Domain Profile** tab, click **Customize…** on the right side of **Protected network connections**.
4. In the pop-up window, uncheck **all** the connected Ethernet ports. Then, click **[OK]**.



Protected Network Connections for the Domain Pr...

Select the check box for each connection you want Windows Defender Firewall to help protect.

twork connections:

5. Repeat steps 3 and 4 for the **Private Profile** and **Public Profile** tabs.

Prevent Windows from Updating

In order to complete updates, Windows might force the IPC to shut down/restart, which obstructs normal production. Therefore, it is highly recommended to prevent Windows from updating to avoid unexpected downtime.

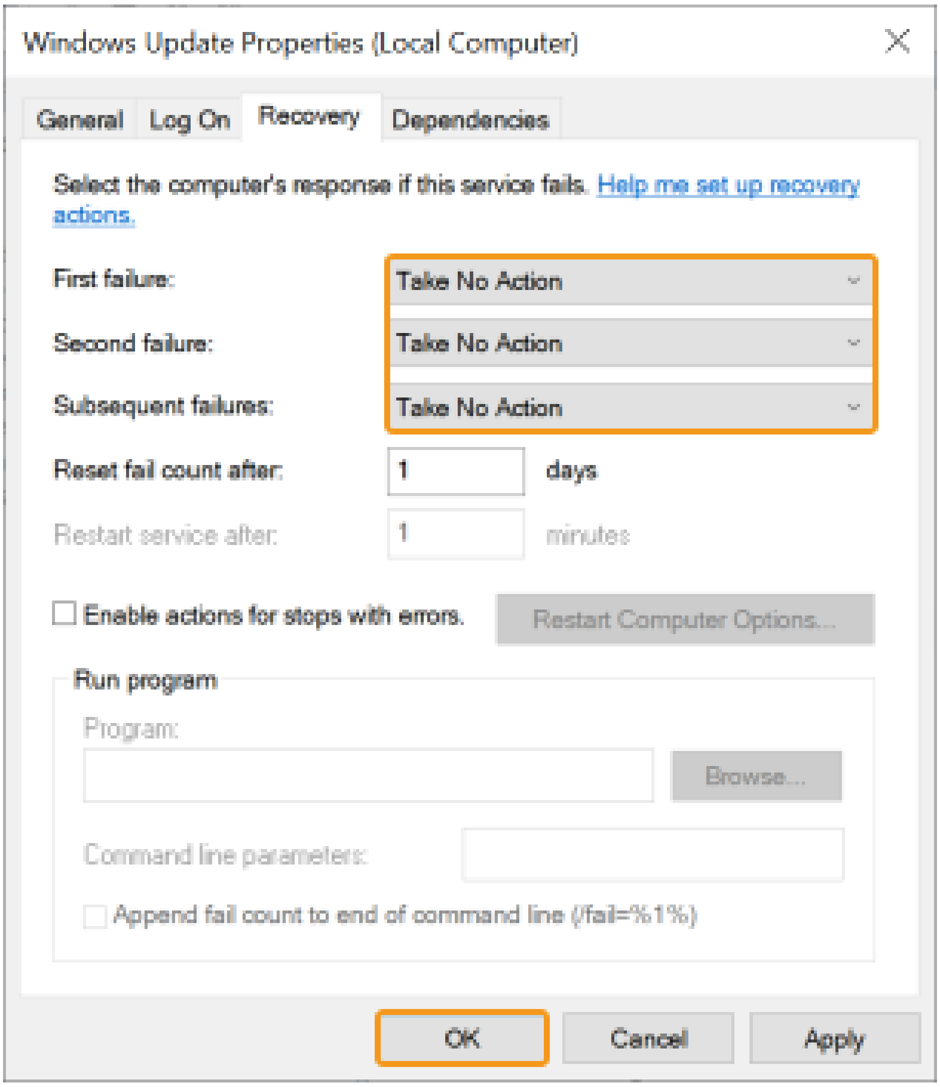
> If you choose to keep Windows Update enabled, please make sure that the shutdown/restart occurs during planned downtime. For example, you can set active hours for Windows Update.

Windows Update Medic Service fixes problems in Windows Update and ensures that your computer continues to receive updates. That is, even if you have disabled Windows Update, Windows Update Medic Service will eventually re-enable it. Therefore, it is necessary to disable Windows Update Medic Service as well.

Disable Windows Update

Follow the steps below to disable Windows Update:

1. Search for **Services** in the search box of the IPC taskbar, and open **Services** from the search results.
2. Scroll down to find **Windows Update**. Double-click it to open **Windows Update Properties**.
3. In the **General** tab, click **[Stop]** to stop the service first. In the drop-down menu of **Startup type**, select **Disabled**, and then click **[Apply]**.
4. Click the **Recovery** tab, and change the response for all failures to **Take No Action**. Then, click **[OK]**.



Windows Update Properties (Local Computer)

Select the computer's response if this service fails. Help me set up recovery actions.

First failure: Take No Action

Second failure: Take No Action

Subsequent failures: Take No Action

Reset fail count after: 1 days

Restart service after: 1 minutes

□ Enable actions for stops with errors.

Run program

Program:

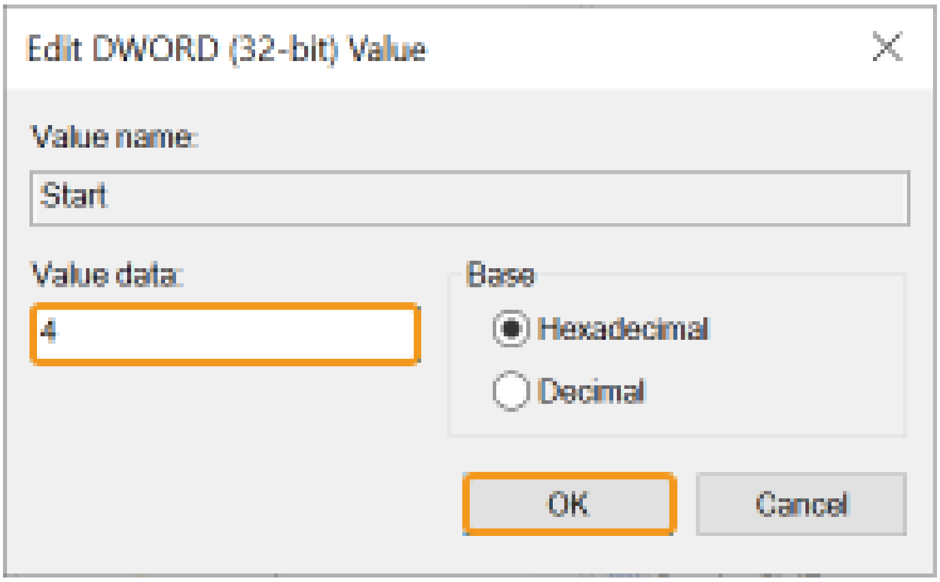
Command line parameters:

□ Append fail count to end of command line (/fail=%1%)

Disable Windows Update Medic Service

Follow the steps below to disable Windows Update Medic Service:

1. Open the **Registry Editor** by one of the following methods:
   - On the IPC, click the search box in the lower left corner of the desktop, and enter **Registry Editor**.
   - On the IPC, click the search box in the lower left corner of the desktop, and enter **regedit**.
   - Hold `win` and `R` to open the Run window, enter **regedit** in the Open input box. Then click [OK].
2. In the left panel, navigate to HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\WaaSMedicSvc. Then in the right panel, right-click **Start** and select **Modify**.
3. In the pop-up window, change **Value data** to 4, and then click [OK] to close the window.



Value name:

Start

Value data:

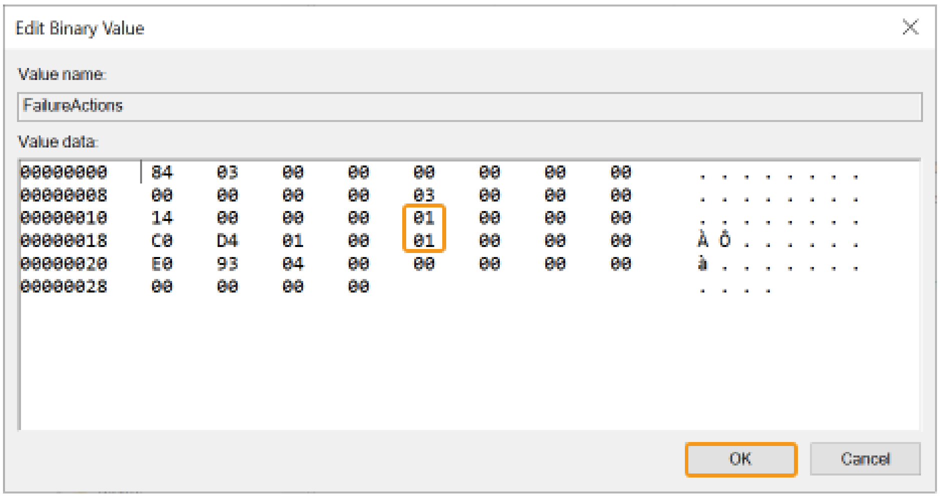
4

Base

● Hexadecimal

○ Decimal

4. In the right panel, right-click **FailureActions** and select **Modify**.
5. In the pop-up window, change the values selected in the box below to **01** each. Then, click [OK].



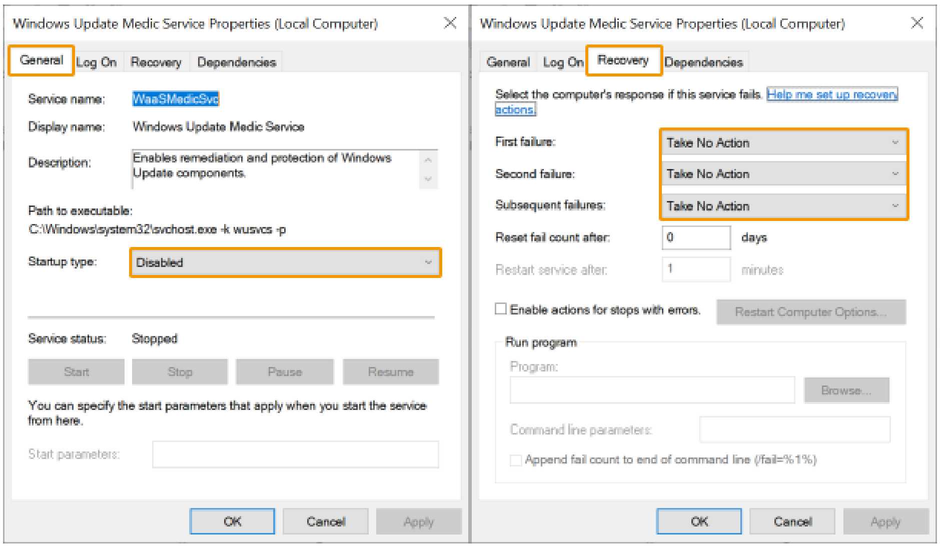
Value name:

FailureActions

Value data:

```
00000000  84  03  00  00  00  00  00  00     . . . . . . . .
00000008  00  00  00  00  03  00  00  00     . . . . . . . .
00000010  14  00  00  00  01  00  00  00     . . . . . . . .
00000018  C0  D4  01  00  01  00  00  00     À Ô . . . . . .
00000020  E0  93  04  00  00  00  00  00     à . . . . . . .
00000028  00  00  00  00                     . . . .
```

6. Check if the changes are effective:
   a. Search for **Services** in the search box of the IPC taskbar, and open **Services** from the search results.
   b. Find **Windows Update Medic Service** and double-click to open it. Now the Startup type in the **General** tab should be **Disabled**, and all responses for failures in the **Recovery** tab should be **Take No Action**.



Windows Update Medic Service Properties (Local Computer)

Service name: WaaSMedicSvc

Display name: Windows Update Medic Service

Description: Enables remediation and protection of Windows Update components.

Path to executable:
C:\Windows\system32\svchost.exe -k wusvcs -p

Startup type: Disabled

Service status: Stopped

You can specify the start parameters that apply when you start the service from here.

Start parameters:

Windows Update Medic Service Properties (Local Computer)

Select the computer's response if this service fails. Help me set up recovery actions.

First failure: Take No Action

Second failure: Take No Action

Subsequent failures: Take No Action

Reset fail count after: 0 days

Restart service after: 1 minutes

□ Enable actions for stops with errors. Restart Computer Options...

Run program

Program: Browse...

Command line parameters:

□ Append fail count to end of command line (/fail=%1%)

6.2. Software License

Mech-Mind uses CodeMeter from Wibu-Systems as the license system for its software. Starting from version 1.6.0, CodeMeter installer is included in the installation package of Mech-Mind software.

Mech-Mind provides two types of license: trial license and full license.

Trial License

Trial software licenses are temporary and are given only when the license dongle cannot be received in time.

If you need to obtain a trial license, please contact Mech-Mind sales to get a **ticket** code. Once you get the ticket code, follow the steps below to activate the trial license:

1. Go to Mech-Mind License WebDepot.
2. On the Welcome page, copy and paste the ticket code into the **Ticket** field, and click `Next`.
3. On the page **My Licenses**, click `Activate Licenses`.
4. On the page **Available Licenses**, click `Activate Selected Licenses Now`.
5. When the message License transfer completed successfully! appears, click `OK`.

If the license is successfully activated, you will see the license displayed in CodeMeter Control Center.

Full License

Before proceeding, please make sure that:

- Plug the license dongle into a USB port of the IPC.
- Run the CodeMeter installer to install CodeMeter and then run it.

CodeMeter will read the license automatically the first time it operates.

For full licenses that do not have perpetual authorization, follow the steps below to update the license:

1. After receiving a message from CodeMeter that the license is about to expire, please export the license request file and send it to Mech-Mind.
2. Mech-Mind will send back a license update file after updating the license.
3. Update the License and check the license expiration date.

Export the License Request File

When CodeMeter informs you that the License is about to expire, please export the license request file as instructed below:

1. Click ● in the IPC system tray to open CodeMeter Control Center.
2. Select the license dongle whose license needs to be updated, and click **[License Update]**.
3. Click **[Next]** in the pop-up window.
4. In the **CmFAS Assistant** window, select **Create license request**, and then click **[Next]**.
5. Select **Extend existing license**, and then click **[Next]**.
6. Check the **Mech-Mind** box, and then click **[Next]**.
7. On the next page, click **[..]** to select a location for saving the license request file, and then click **[Commit]**.

   If licenses on multiple license dongles need to be updated, please repeat steps 2 to 7 to export a license request file for each dongle.

8. Send the license request file to Mech-Mind.

Update the License

After you send the license request file, Mech-Mind will send back a license update file in **WIBUCMRAU** format. You can update the license automatically or manually.

Automatic update: Double click the license update file to update the corresponding license.

Manual update:

1. Click ● in the IPC system tray to open CodeMeter Control Center.
2. Select the license update file in **WIBUCMRAU** format returned by Mech-Mind and click **[License Update]**.
3. Click `Next` in the pop-up window.
4. In the **CmFAS Assistant** window, select **Import license request**, and then click **[Next]**.
5. Click **[..]** to select the license update file, and then click **[Commit]**.

It is recommended to try **automatic update** first. If **automatic update** fails, perform **manual update** instead.

Check the License

Please check whether the license is updated successfully. Follow the steps below to check the license in Mech-Vision and Mech-Viz respectively.

- Check the license in Mech-Vision

  In the main interface of Mech-Vision, click Help › About in the menu bar, and you can check the expiration time in the pop-up "About" window to see whether the license is updated successfully.

- Check the license in Mech-Viz

  After opening Mech-Viz, click **[Help]** in the lower left corner of the File page, and you can check the expiration time in the "Version information" panel to see whether the license is updated successfully.

If the license has been successfully updated, you can end the license configuration. If not, please update the license again. If the problem persists, please contact Mech-Mind.

6.3. Calibration Board

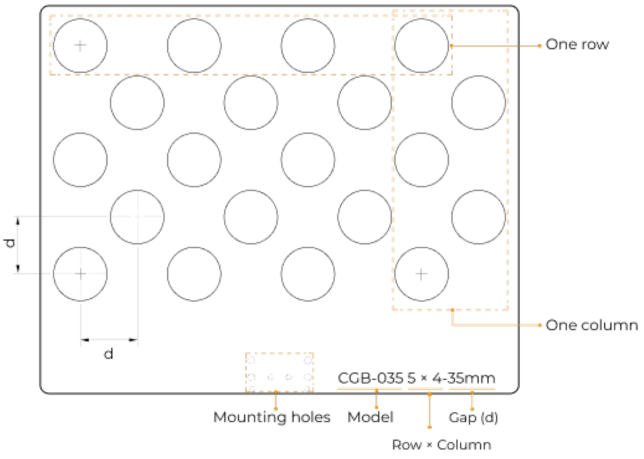
Calibration board is usually used for intrinsic parameter check and hand-eye calibration. This chapter introduces the types, models, and mounting methods of calibration boards.

Calibration Board Types and Models

Mech-Mind provides three types of calibration boards: BDB, CGB and OCB.

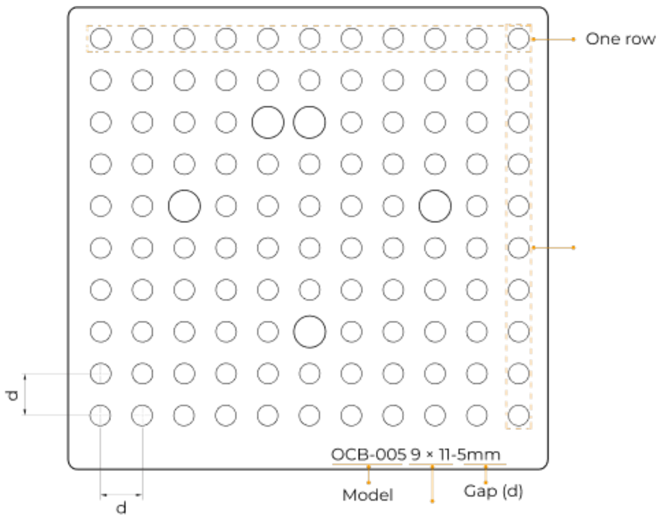
CGB

For V4 cameras.

Model	Functional Diagrams	d (mm)	Row	Column
CGB-020	 One row One column d d CGB-035 5 × 4-35mm Mounting holes Model Gap (d) Row × Column	20	5	4
CGB-035		35	5	4
CGB-050		50	5	4

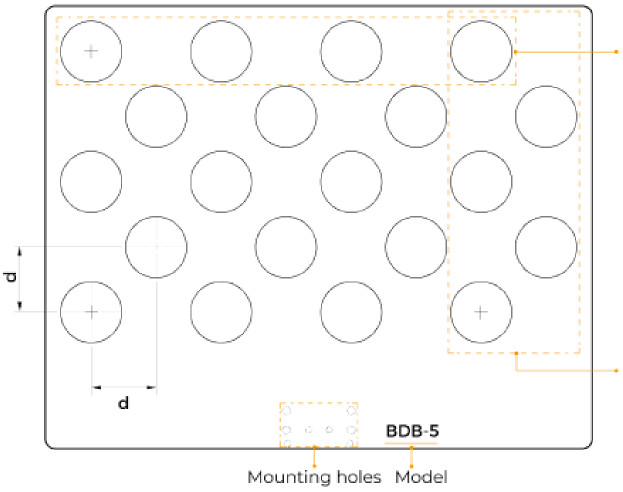
OCB

OCB calibration boards have no mounting holes and therefore cannot be mounted on the last joint of the robot. It can be used only when the camera is mounted in the eye in hand (EIH) mode.

Model	Functional Diagrams	d (mm)	Row	Column
OCB-005		5	9	11
OCB-010		10	9	11
OCB-020		20	9	11

BDB

For V2 and V3 cameras.

Model	Functional Diagrams	d (mm)	Row	Column
BDB-5		20	5	4
BDB-6		35	5	4
BDB-7		50	5	4

Mounting Method

In the following calibration scenarios, you can mount the calibration board on the last joint of the robot.

- Automatic Calibration in the Eye to Hand (ETH) Scenario
- Manual Calibration in the ETH Scenario (Multiple Random Calibration Board Poses)

For other calibration scenarios and intrinsic parameter checks, there is no need to mount the calibration board. Please place the calibration board in the center of the object plane.

Common ways to mount the calibration board are shown below:

Direct mounting	Mounting through the flange adapter

6.4. Camera Mounting

6.4.1. Camera Mounting Frame Designs

Camera Mounting Frame Forms

The mounting frame required is different through different camera mounting methods, as detailed in the table below.

Mounting Frames in the Eye to hand (ETH) Setup	Aluminum Profile Frame Square Tube Frames Sliding Camera Mounting Frames (Horizontal) Vertical Camera Mounting Frames
Mounting Frames in the Eye in hand (EIH) Setup	The camera is above the robot’s J6 axis flange The camera is below the robot’s J6 axis flange

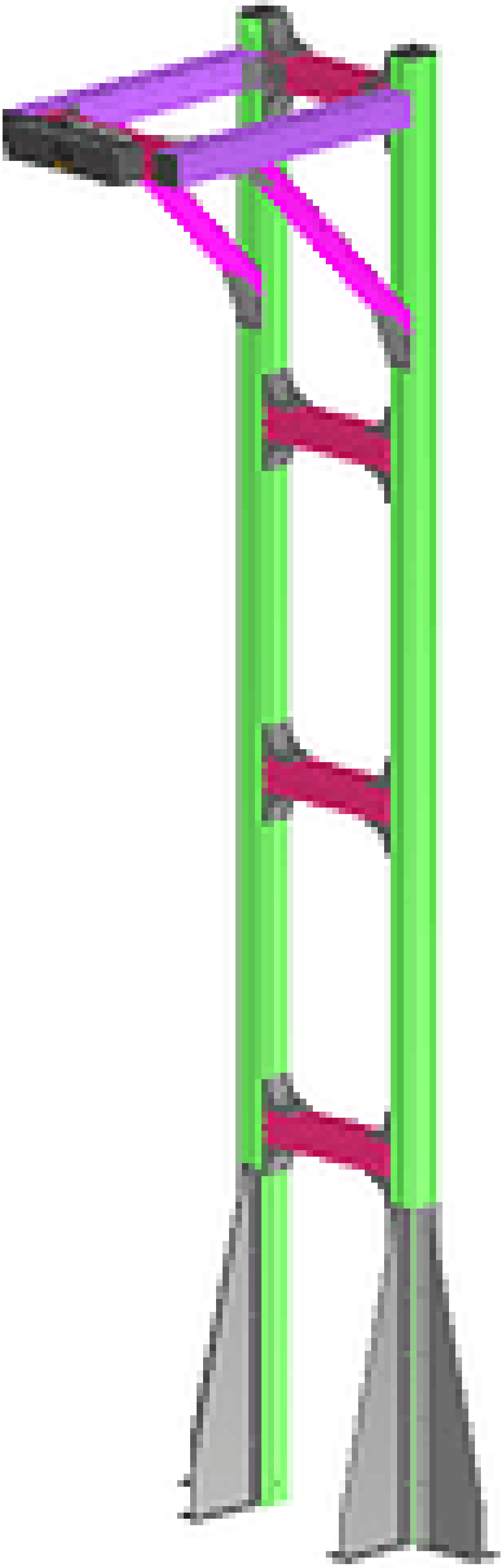
For details on the design requirements and applicable scenarios of various mounting frames, please refer to Mounting Frame Designs and Mounting Methods in the ETH Setup and Mounting Frame Designs and Mounting Methods in the EIH Setup.

Mounting Frame Designs and Mounting Methods in the ETH Setup

When designing a mounting frame in the ETH Setup, you need to arrange trays on the surface of the column to facilitate the routing of camera cables.

Aluminum Profile Frame Design

The aluminum profile frame is constructed with double-column aluminum profiles, and you need to choose European standard 8080 aluminum profiles. The specific design is shown in the figure below.

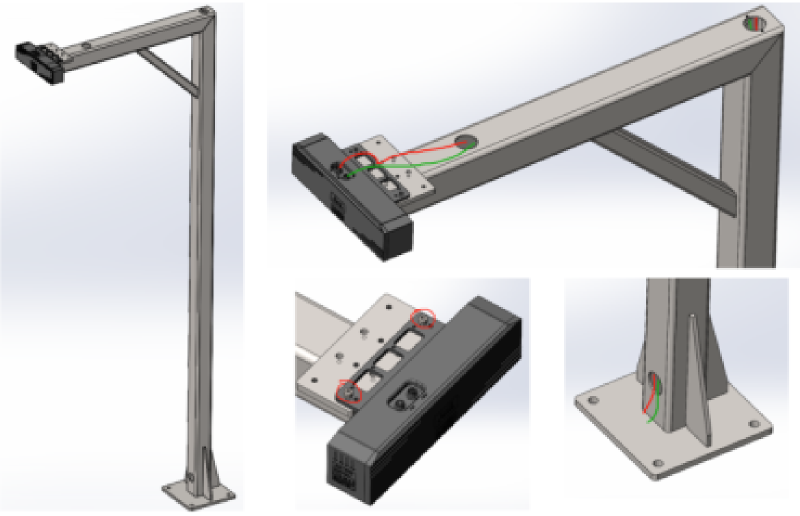


Square Tube Frame Designs (Recommended)

When using a square tube frame, you need to choose national standard hollow square tubes for welding. Hollow square tubes possess high strength and stability and can provide a robust support frame. You can choose the single-column or portal frame according to your needs. Specific forms and requirements are as follows:

- Single-column Frame Design

  This form is suitable for environments without noticeable vibrations. It features a simple structure and easy installation.

  Design Requirements:

  - The specifications of the square steel should be no less than 100*100*5.
  - It is recommended to use integral welding.
  - The crossbar and column of the camera mounting frame should be equipped with necessary diagonal braces and ribs.

- Portal Frame Design

  This form is suitable for environments with noticeable vibrations. The portal frame provides a more robust support frame. Therefore, it can better resist external vibrations and impacts, and ensure the camera’s stability.

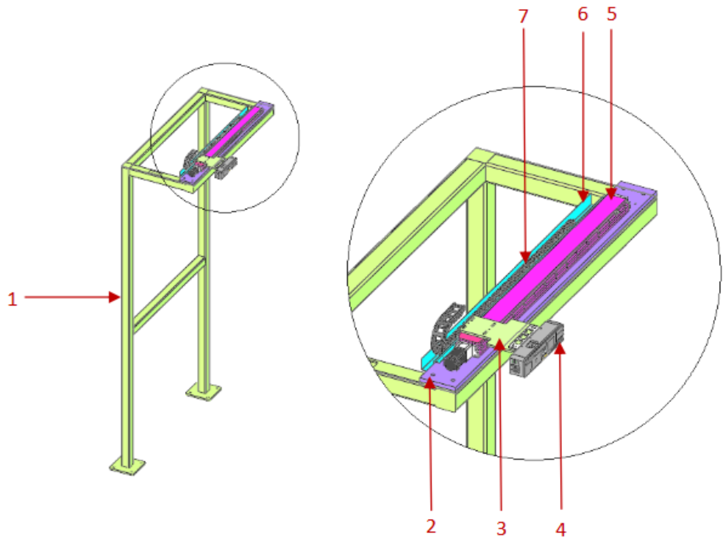
Sliding Camera Mounting Frame Designs

Feasible: Adjacent stacks share the same camera.

Based on different power components, sliding camera mounting frames can be divided into: servo module driven frames and pneumatic cylinder driven frames.

- Servo Module Driven Frame

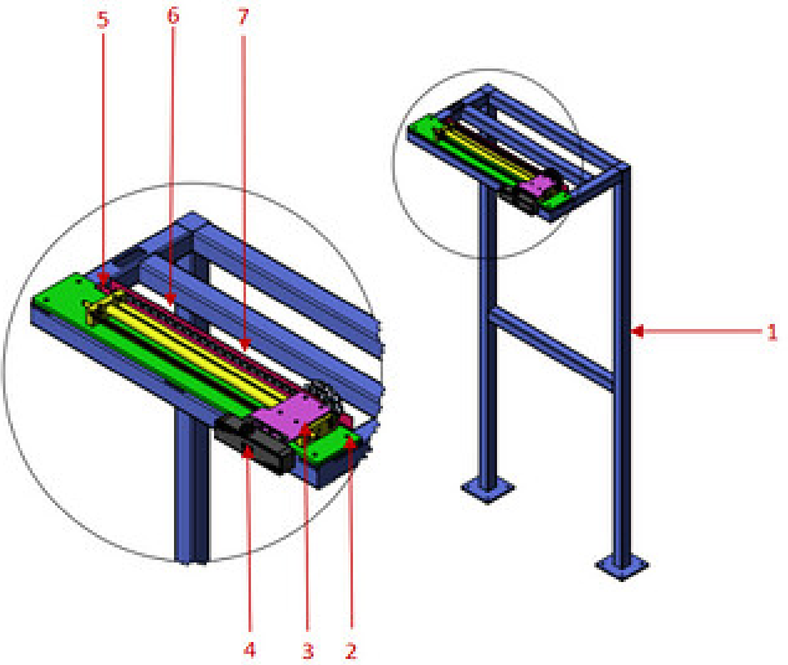
  The servo module driven frame requires a smooth operation, with no impact on stopping and starting. Recommended normal operating speed: ≤250 mm/s. The components of a servo module driven frame are shown in the figure below: 1-frame body, 2-module mounting plate, 3-camera mounting plate, 4-camera, 5-servo module, 6-drag chain mounting plate, and 7-drag chain.



Frame body: Welded from 100*100 square steel. If the whole frame is high, you may add diagonal braces or use other securing methods. **Module mounting plate and camera mounting frame**: High strength and rigidity are required to ensure the stability of the camera when the servo module is in operation. **Servo module**: The repeat positioning precision of servo modules generally includes ±0.01 mm, ±0.02 mm, ±0.05 mm, ±0.1 mm, ±0.2 mm, etc. You can select the linear servo module according to the actual precision requirements of your project. It is recommended to use a servo motor for the driving part. **Drag chain**: Select standard market-available parts based on the diameter of your bus cable. Generally, the cross-sectional area occupied by the bus cable should be less than 60% to 80% of the internal volume of the drag chain. **Drag chain mounting plate**: Fabricated using sheet metal bending as required.

- Pneumatic Cylinder Driven Frame (Not Recommended)

  The positioning precision of pneumatic cylinders is slightly lower than that of servo modules., with a typical precision of ±0.1 mm. The pneumatic cylinder driven frame requires a smooth operation, with no impact on stopping and starting. Recommended normal operating speed: ≤200 mm/s. The components of a pneumatic cylinder driven frame are shown in the figure below: 1-frame body, 2-pneumatic cylinder mounting plate, 3-camera mounting plate, 4-camera, 5-pneumatic cylinder, 6-drag chain mounting plate, and 7-drag chain.



Pneumatic cylinder: Choose the cylinder stroke according to the actual distance of the on-site stacks. For the cylinder type, it is recommended to choose a guide rod cylinder. Otherwise it must be used with a linear guide. The pneumatic cylinder must be used with an oil buffer and a limit block. In addition, it should be equipped with an exhaust throttle valve. For how to design and select the frame body, pneumatic cylinder mounting plate, drag chain mounting plate and drag chain, please refer to the servo module driven frame.

Considering factors such as impacts on the pneumatic cylinder, overall operation precision, camera vibrations at the docking position, and long-term stability, this method is not recommended.

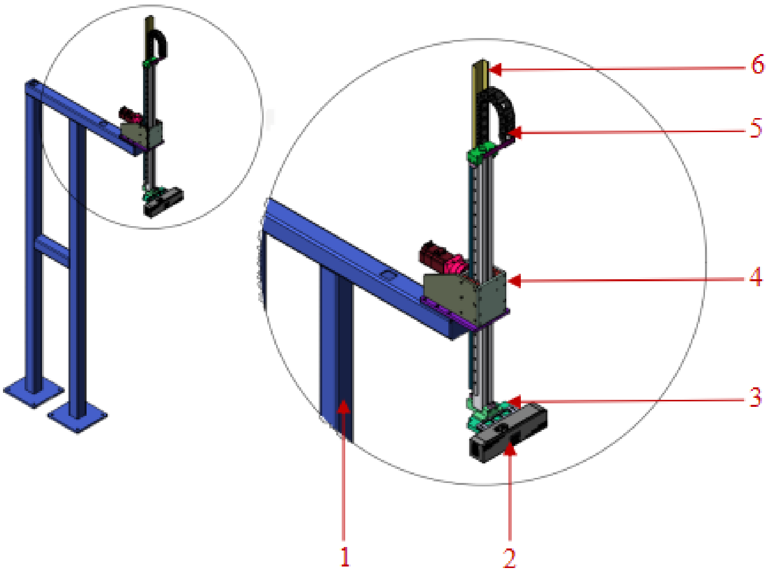
Vertical Camera Mounting Frame Designs

Feasible: Suitable for scenarios where the pallet pattern is tall, its length or width dimension is too large, and the point clouds of the sacks at the bottom have poor quality.

Based on different driving forces, vertical camera mounting frames can be divided into: pneumatic vertical camera mounting frames and electric vertical camera mounting frames.

- Electric Vertical Camera Mounting Frame

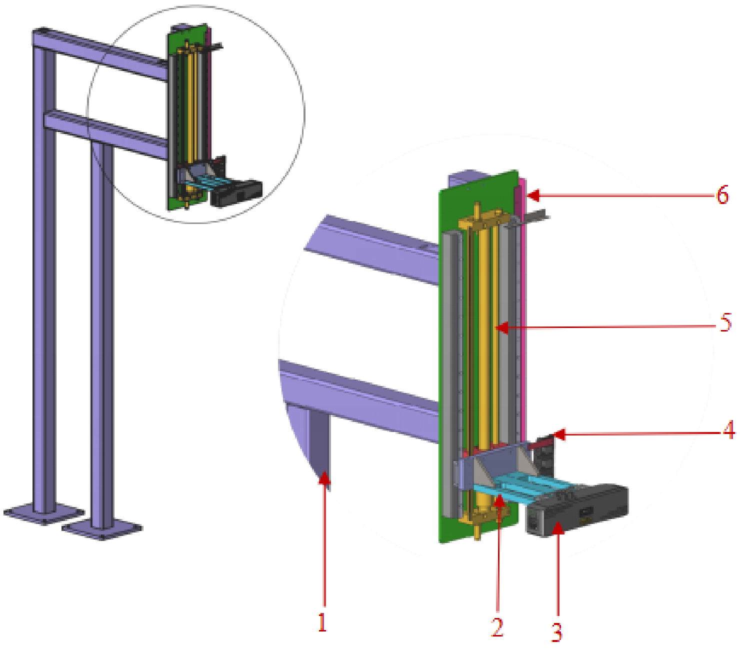
  Design Requirements: The camera mounting frame should perform smoothly, with no severe impacts during operation; the movement speed should be less than 250 mm/s; ensure smooth and stable operation of the moving mechanism. The camera mounting frame should be rigid to ensure no vibrations while working. The components of an electric vertical camera mounting frame are shown in the figure below: 1-frame body, 2-3D camera, 3-camera mounting plate, 4-gear and rack assembly, 5-drag chain, and 6-drag chain mounting plate.



Gear and rack assembly: It is recommended to choose the brake motor to prevent the camera from falling in free fall after an unexpected power outage.

- Pneumatic Vertical Camera Mounting Frame (Not Recommended)

Design Requirements: The camera mounting frame should perform smoothly, with no severe impacts during operation; the movement speed should be less than 200 mm/s; ensure smooth and stable operation of the moving mechanism.The camera mounting frame should be rigid to ensure no vibrations while working. The components of a pneumatic vertical camera mounting frame are shown in the figure below: 1-frame body, 2-camera mounting plate, 3-3D camera, 4-drag chain, 5-rodless cylinder, and 6-drag chain mounting plate.



Pneumatic cylinder: Choose the cylinder stroke according to the actual distance of the on-site stacks. The pneumatic cylinder must be used with an oil buffer and a limit block. In addition, it should be equipped with an exhaust throttle valve. Choose the center-closed directional control valve for your pneumatic cylinder. This prevents the camera from falling in free fall after the air is shut off.

Considering factors such as impacts on the pneumatic cylinder, overall operation precision, camera vibrations at the docking position, and long-term stability, this method is not recommended.

Mounting Requirements for Mounting Frames in the ETH Setup

When mounting the frames in the ETH Setup, you should use chemical bolts or expansion bolts to secure the camera mounting frame to the ground. Those securing methods provide stable and reliable support and ensure that the camera does not shake.

For details about camera mounting requirements and guidelines for mounting camera cables, please refer to:

- Camera Mounting Requirements
- Guidelines for Mounting Camera Cables

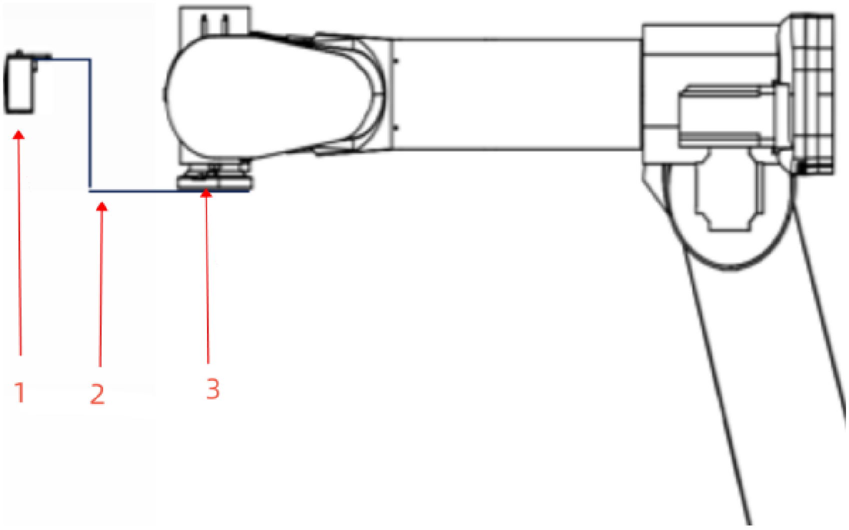
Mounting Frame Designs and Mounting Methods in the EIH Setup

The following section introduces two common design forms for mounting frames in the EIH Setup.

The camera is above the robot’s J6 axis flange

- Advantage: When the robot’s arm span is small, this form helps raise the camera to expand the field of view.
- Disadvantage: The rotation range of the robot’s J6 axis is limited and the mounting frame or the camera and the robot body may interfere with each other.

Structure diagram: 1-camera, 2-camera mounting frame, 3-robot’s J6 axis flange

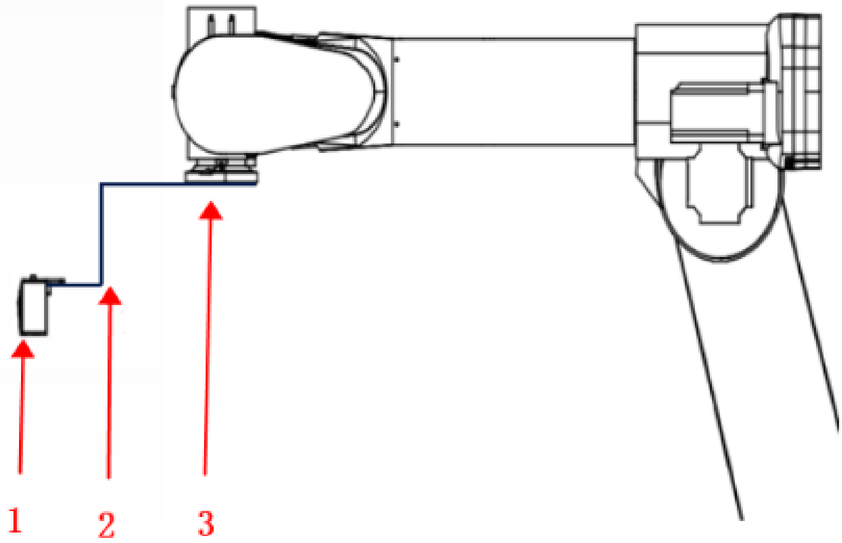


The camera is below the robot’s J6 axis flange

- Advantage: The robot’s J6 axis rotation is not affected by the camera or camera mounting frame.
- Disadvantage: When the robot lifts to the same height, the image-capturing position is not high,

resulting in a smaller field of view.

Structure diagram: 1-camera, 2-camera mounting frame, 3-robot’s J6 axis flange



Mounting Requirements for Mounting Frames in the EIH Setup

When mounting frames in the EIH Setup, you should take measures to prevent them from loosening, such as applying thread glue to the mounting bolts, using anti-loosening washers, etc.

For details about camera mounting requirements and guidelines for mounting camera cables, please refer to:

- Camera Mounting Requirements
- Guidelines for Mounting Camera Cables

6.4.2. Camera Mounting Requirements

There are two common methods to mount a camera:

- Eye to hand (ETH): The camera is mounted on a camera mounting frame independent of the robot.
- Eye in hand (EIH): The camera is mounted on the last joint of the robot and moves with the robot.

The requirements for mounting the camera mounting frame and cabling vary with different mounting methods.

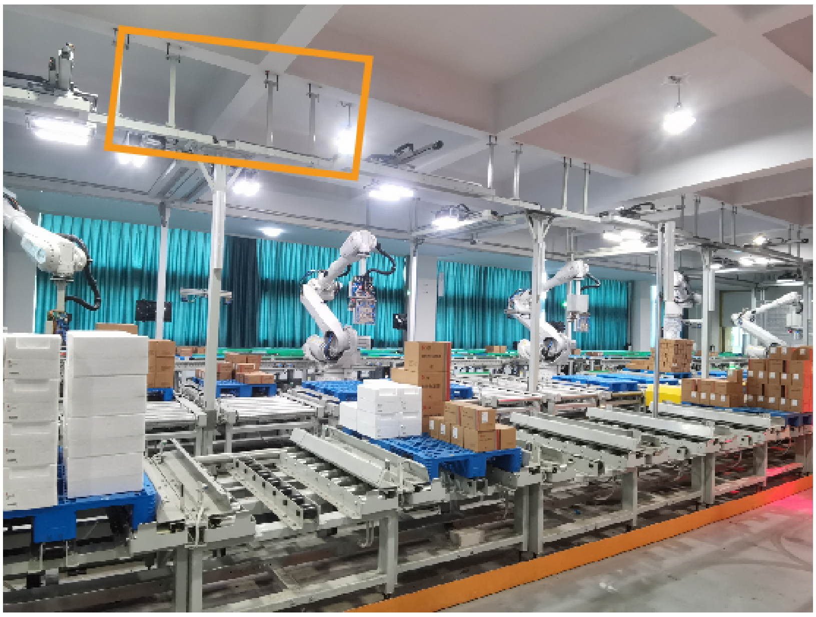
Camera Mounting Requirements (ETH)

When the camera is mounted on a camera mounting frame (stand) independent of the robot, you need to consider its stability, reliability and cabling to ensure the accuracy of image capturing. Specific notes are as follows:

- Secure the camera mounting frame

  You can use either chemical bolts or expansion bolts to secure the camera mounting frame to the ground, wall, etc., which can provide stable and reliable support to ensure that the camera will not shake.

The number and type of foundation bolts depend on the height and structure of the camera mounting frame. When the camera is not mounted on the ground, you may need to reinforce the camera mounting frame by side walls, the floor, or the ceiling, as shown in the figure below.

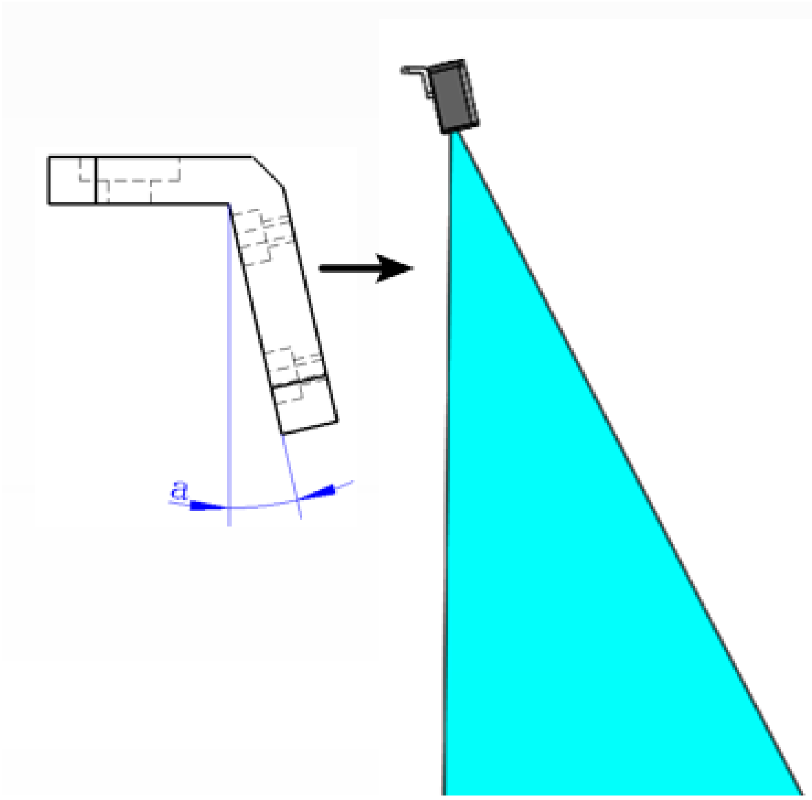


- Cabling

  Install the camera cables in the cable trays arranged on the surface of the columns for neat and organized routing. This not only ensures the safety and aesthetic appeal of the cables but also facilitates easy maintenance.

- Adjust camera mounting angle

  Normally, the camera is mounted vertically downward. However, for project sites with limited ceiling height, you can tilt the camera to adjust the mounting angle. In the early stages of designing your project, it is recommended to perform emulation and simulation tests on your project to ensure that the angle meets the actual requirements.



Camera Mounting Requirements (EIH)

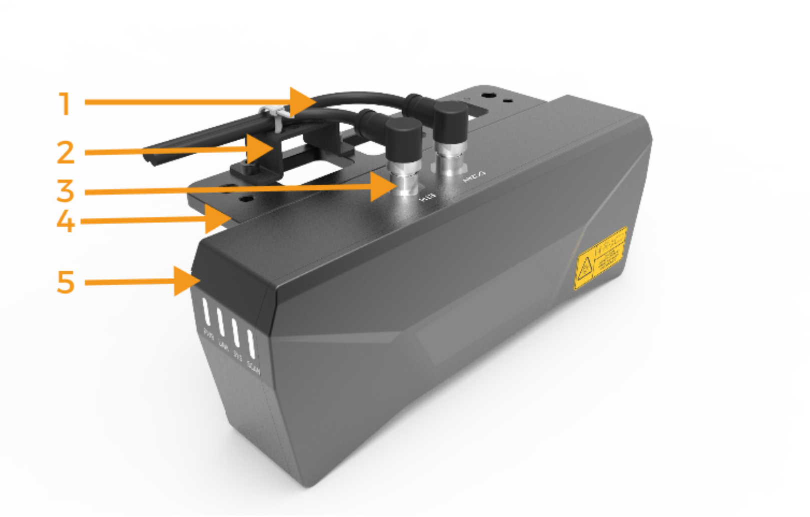
Pay attention to the following notes when you mount the camera on the last joint of the robot:

- Install the camera mounting frame

  When you install the camera mounting frame (bracket), you should take measures to prevent it from loosening, such as applying thread glue to the mounting bolts, using anti-loosening washers, etc.

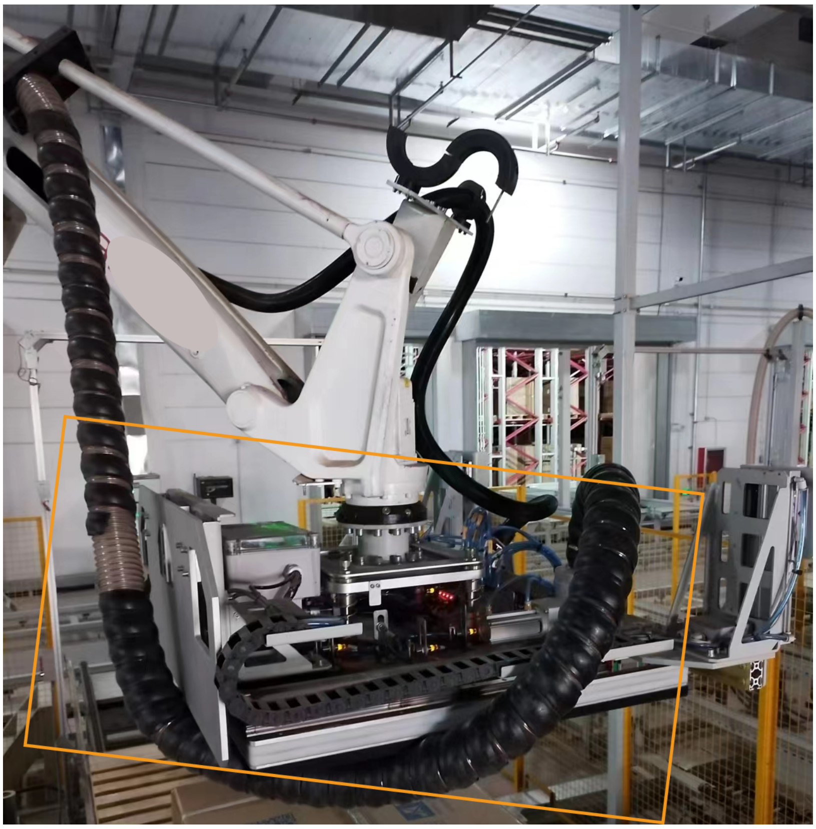
- Secure cables

  It is necessary to secure the cables near the camera connector to prevent force on it. When bundling cables, it is important to consider the rotational allowance of the robot's end flange to avoid insufficient cable release, which could lead to pulling on the camera cables, and even cause irreversible damage to the camera cable connector. The way to secure cables is shown in the figure below: 1-camera cable, 2-tube clamp, 3-cable connector, 4-camera bracket, 5-camera.



- Cabling

  The routing of cables should be arranged properly to avoid the cables in the dresspack being too long or too short. The following figure shows an incorrect example of excessively long cables.



- Cable protection

  Use the dresspack to protect cables and pay attention to the following notes:

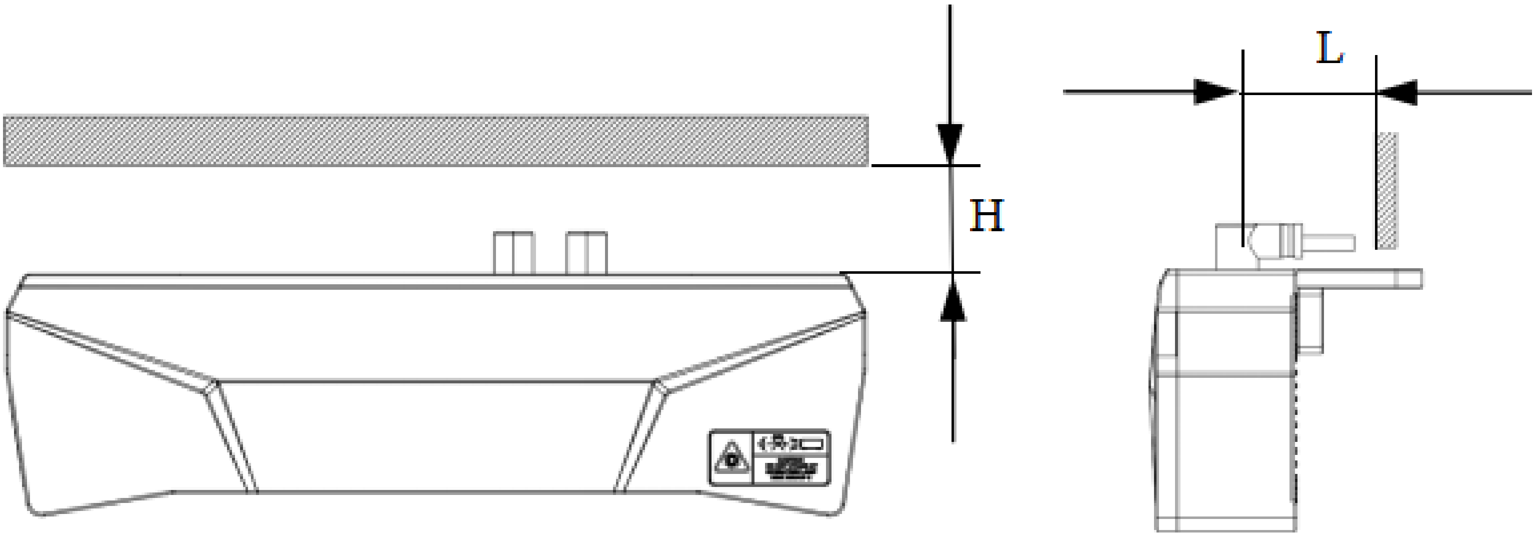
  - The dresspack shall not use self-winding cable conduits or fabric cable wrap.
  - Choose standard complete sets of corrugated pipe fittings for the dresspack (including but not limited to: tube clamps, half-shells, protection sleeves, etc.).

- Both the dresspack and cables should be secured and assembled properly to prevent additional torque on the cables during robot motion.

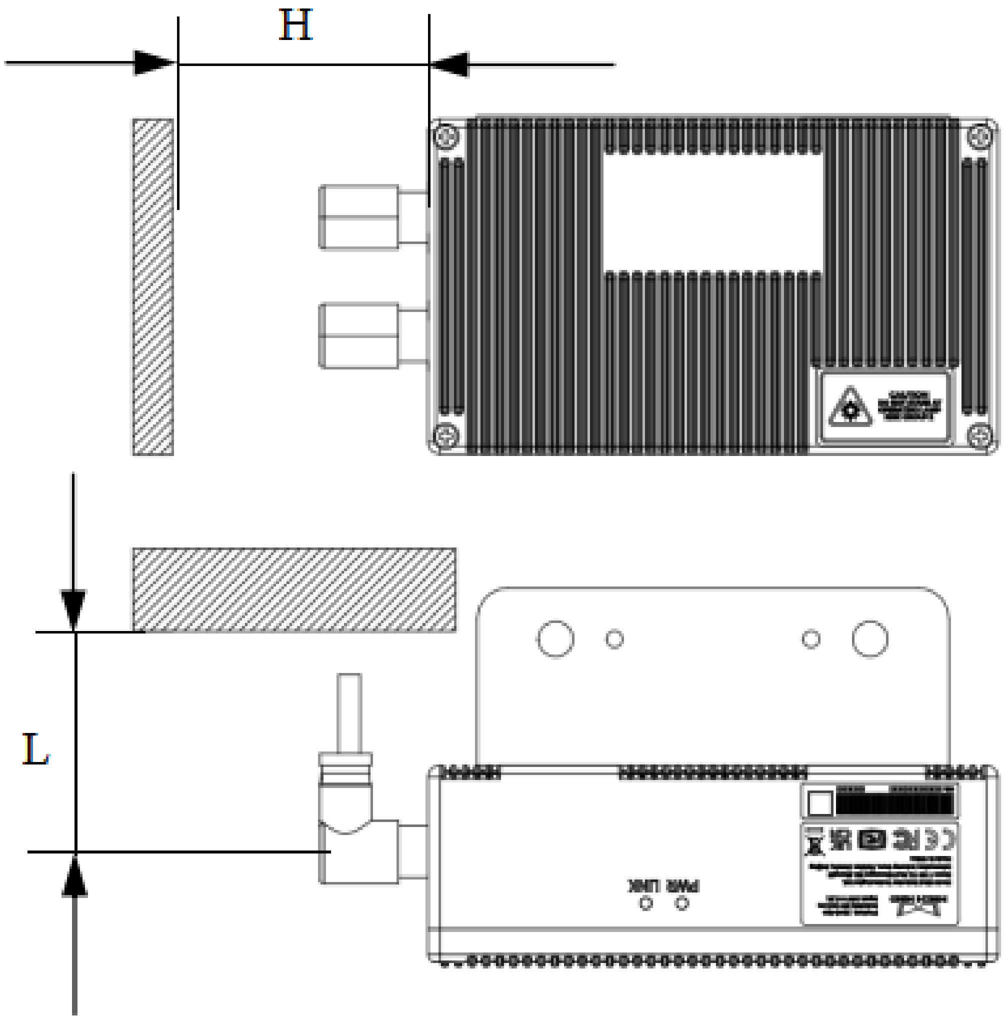
6.4.3. Guidelines for Mounting Camera Cables

Installation Specifications for Camera Cable Terminals

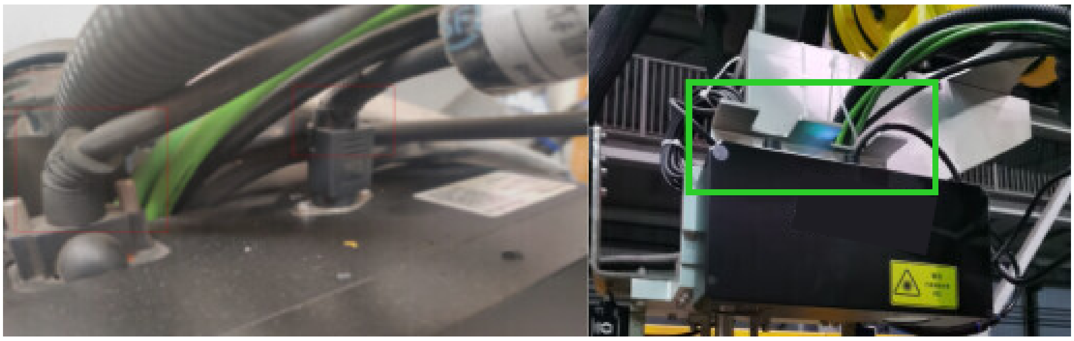
To facilitate the installation and removal of the cable terminal, when installing the camera, the distance in the following two directions should meet these two requirements: H ≥ 70 mm; L ≥ 65 mm.



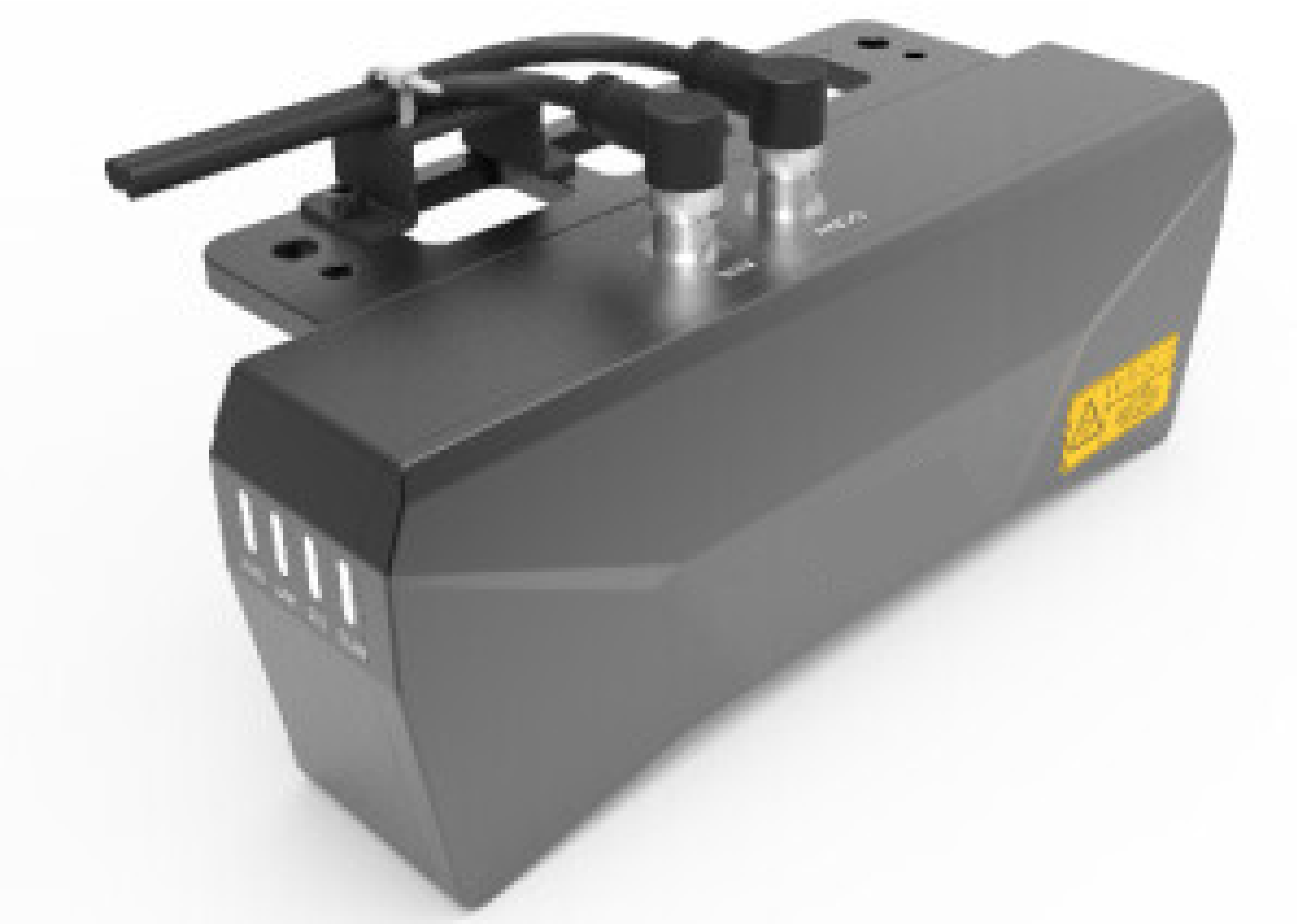
NANO series cameras have side cable outlets. L and H are shown below in the real design:



When mounting the camera using the eye in hand (EIH) mode, there are many cables at the end of the robot. In this case, the terminals should be protected from additional external force.



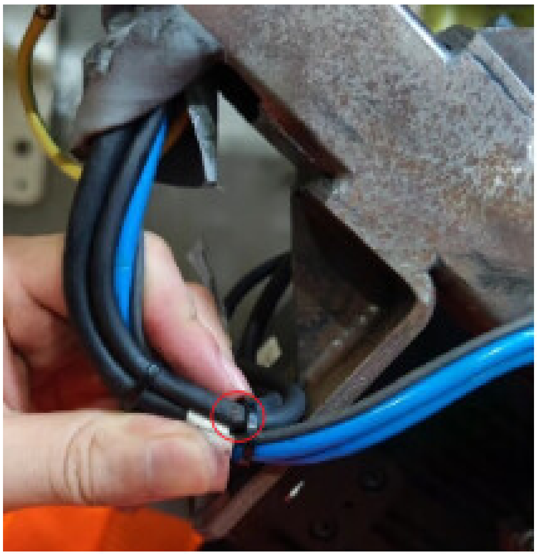
The following bracket can be used to secure camera cables onsite.



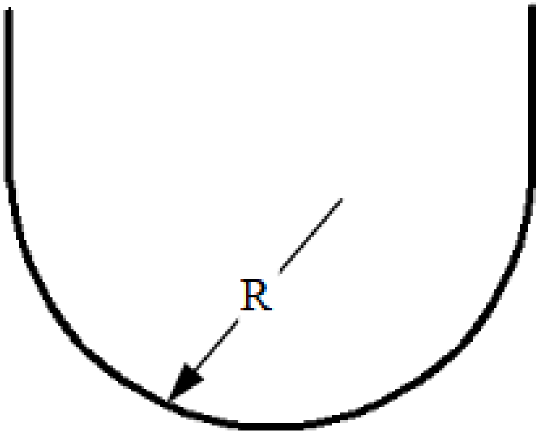
Technical Specifications for Cabling

When **cabling**, the camera cables should not be stretched tight but should have an appropriate amount of slack; the cabling should be laid out in a straight and neat manner, avoiding any crossing or entanglement; at the points where the cables make turns, they should be securely tied with appropriate tension.

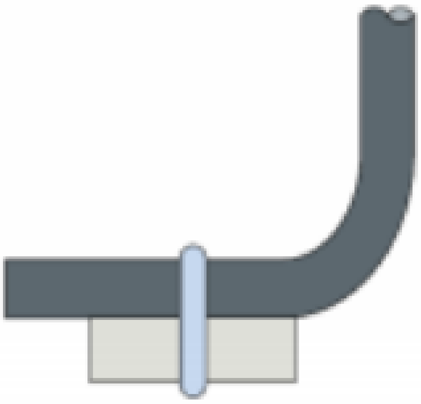
When **wiring** and **routing**, observe whether there are sharp edges or corners on various workobjects and components. Avoid these areas when routing to prevent damaging the cables.

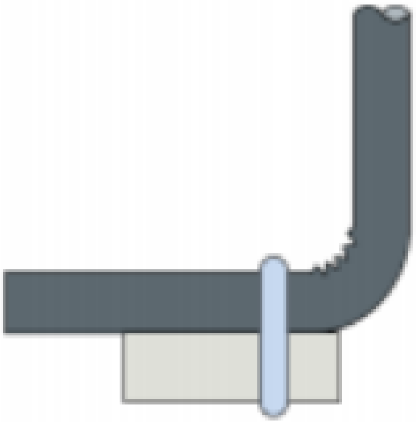


When **changing the direction** of the cables, try to make the bending radius (R) as large as possible and leave adequate slack, as shown in the figure below. Avoid excessive bending of the cables to prevent any impact on the quality and functionality of the cables. The recommended value of R is more than 7.5 times the outer diameter of the cable.



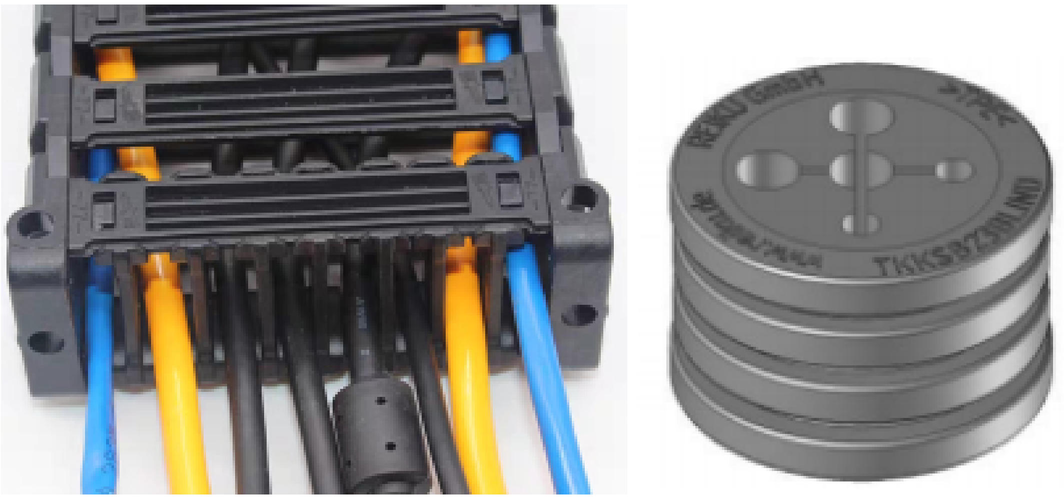
When **securing** the cables, avoid sharp bends as shown in the right figure. Please refer to the left figure for installation.



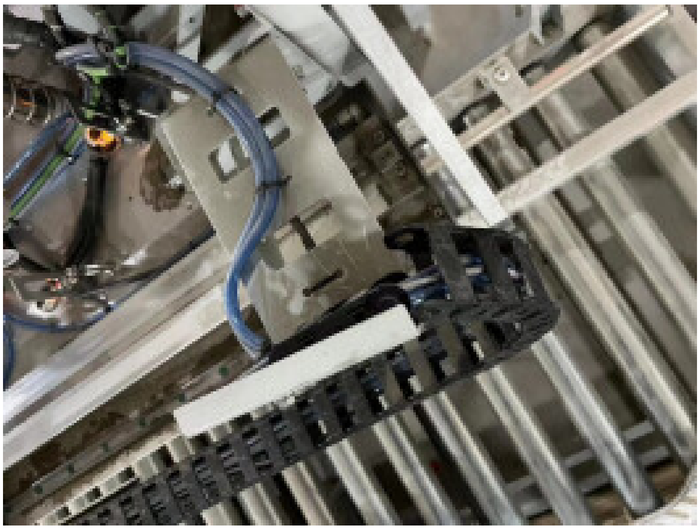


Routing Requirements for Cables inside Dresspack or Drag Chain

Do not secure the cables inside the drag chain or dresspack. Otherwise, the cables may not be able to disperse, move, and bend. Secure the cables at the entrance of the dresspack or drag chain to prevent friction between cables. Please refer to the figure below for specific securing methods.



The general requirement is that at least 3 to 4 sections of the drag chain do not move with the camera.



When routing cables inside the drag chain,do not subject the cables to additional tension. That is, the cables should not be pulled too tightly (as shown in the left figure) or left too loose (as shown in the right figure). Otherwise, friction inside the drag chain can lead to wear of the cable’s insulation layer.

When routing cables through corrugated pipe, it is important to ensure that the cables inside the pipe do not become tangled or entwined with each other.

6.5. Compatibility Mode Introduction

In compatibility mode, the following situation exists for some robots: when the pose of the real robot is synchronized with that of the simulated robot in the software simulation area, the robot tool pose displayed on the teach pendant does not match the tool pose in the software interface. The problem occurs in the robots including all FANUC and YASKAWA four/five/six-axis robots, six-axis robots of Kawasaki R series, all STAUBLI six-axis robots, and all AUBO and HANS robots.

In compatibility mode, when software communicates with the robot through Standard Interface or Master-Control, the difference in the tool pose will be compensated automatically. However, when you perform manual calibration, teach the pick point, or send the planning or recognition result in tool poses to the robot through Adapter, the difference in the tool pose should be compensated manually.

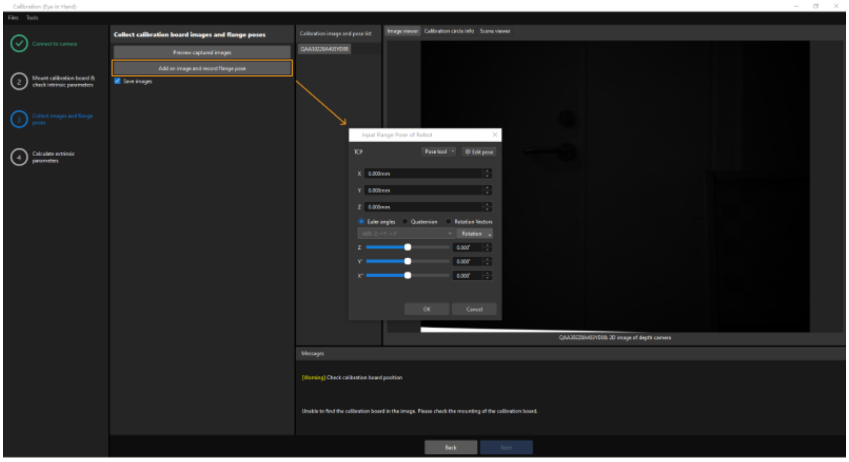
Scenarios Require Manual Compensation

If the robot you use belongs to FANUC four/five/six-axis robots, YASKAWA four/five/six-axis robots, Kawasaki R series six-axis robots, STAUBLI six-axis robots, AUBO, or HANS robots, you need to manually compensate the difference when you perform manual calibration, teach the pick point, and establish Adapter communication.

Manual Calibration

In the process of manual calibration, click **[Collect images and flange poses]**, and then enter the flange pose data in the **Input Flange Pose of Robot** window.

The compensation value should be added to the Z-value of the tool pose displayed on the teach pendant, and then the result can be used as the Z parameter in the software.


Calibration (Eye in Hand)
Files Tools
Connect to camera
Mount calibration board & check intrinsic parameters
Collect images and flange poses
Calculate extrinsic parameters
Collect calibration board images and flange poses
Preview captured images
Add an image and record flange pose
Save images
Calibration image and pose list
QAA30228A403YD08
Image viewer
Calibration circle info
Scene viewer
Input Flange Pose of Robot
TCP
Pose tool
Edit pose
X
Y
Z
Euler angles
Quaternion
Rotation Vectors
ABB: Z->Y'->X''
Rotation
Z
Y'
X''
OK
Cancel
QAA30228A403YD08: 2D image of depth camera
Messages
Back
Next

X 0.000mm

Y 0.000mm

Z 0.000mm

● Euler angles ○ Quaternion ○ Rotation Vectors

Z 0.000°

Y' 0.000°

X'' 0.000°

[Warning] Check calibration board position

Unable to find the calibration board in the image. Please check the mounting of the calibration board.

Teach the Pick Point

When adding pick point by teaching in the **Matching Model and Pick Point Editor**, the image capturing pose and picking pose should be entered in the form of robot flange poses.

The compensation value should be added to the Z-value of the tool pose displayed on the teach pendant, and then the result can be used as the Z parameter in the software.

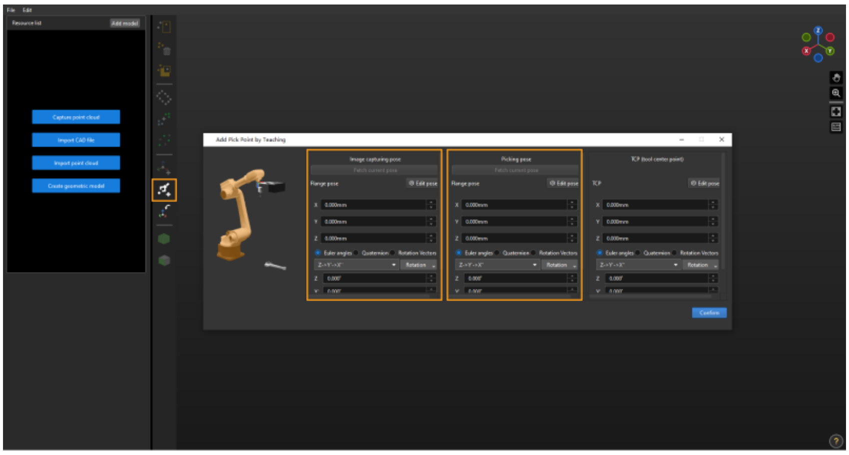


Image capturing pose: Flange pose X 0.000mm, Y 0.000mm, Z 0.000mm; ● Euler angles; Z->Y'->X''; Z 0.000°

Picking pose: Flange pose X 0.000mm, Y 0.000mm, Z 0.000mm; ● Euler angles; Z->Y'->X''; Z 0.000°

TCP (tool center point): TCP X 0.000mm, Y 0.000mm, Z 0.000mm; ● Euler angles; Z->Y'->X''; Z 0.000°

Adapter Communication

Adapter communication is the customized communication method for engineers. You need to compensate the difference in the script.

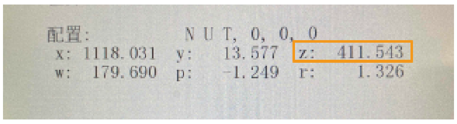
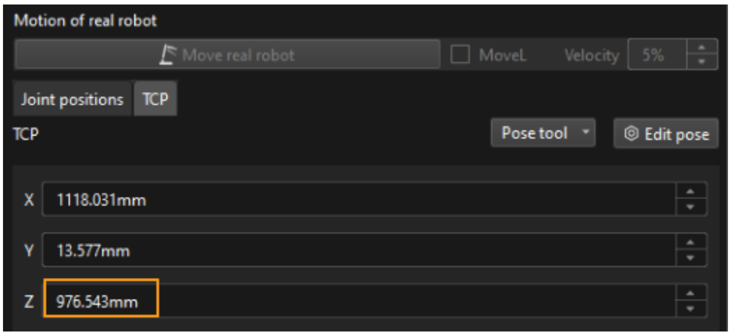
The compensation value should be added to the Z-value of the tool pose displayed on the teach pendant, and then the result can be used in the code.

How to Determine Compensation Value

You can determine compensation value through the following methods:

- Synchronize the real robot with the simulated robot in the software simulation area. The compensation value is the difference of Z values between the robot tool pose displayed on the teach pendant and in the software.

  Take the robot in the following table as an example.

	
The tool pose displayed on the teach pendant	The tool pose displayed in Mech-Viz

```
配置:            N U T, 0, 0, 0
 x: 1118.031  y:    13.577  z:   411.543
 w:  179.690  p:    -1.249  r:     1.326
```

X 1118.031mm

Y 13.577mm

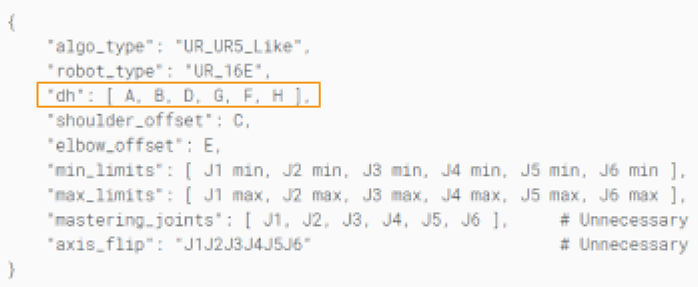
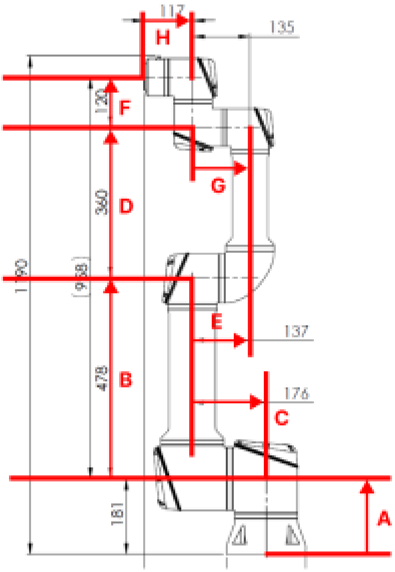
Z 976.543mm

The Z values of tool pose displayed on the teach pendant and Mech-Viz are respectively 411.543, and 976.543, so the compensation value is 976.543 - 411.543 = 565 mm.

- Check the parameter document of the robot. The compensation value is the value of the first DH parameter (DH1), or the value of Zoffset in the parameter document.

  Take the robot in the following table as an example.

  From the UR_16E_algo.json file, we know that DH1 corresponds to A. By checking the DH specifications, we know that A equals 181 mm, so the compensation value is 181 mm.

	
UR_16E_algo.json	DH Specification

```
{
    "algo_type": "UR_UR5_Like",
    "robot_type": "UR_16E",
    "dh": [ A, B, D, G, F, H ],
    "shoulder_offset": C,
    "elbow_offset": E,
    "min_limits": [ J1 min, J2 min, J3 min, J4 min, J5 min, J6 min ],
    "max_limits": [ J1 max, J2 max, J3 max, J4 max, J5 max, J6 max ],
    "mastering_joints": [ J1, J2, J3, J4, J5, J6 ],      # Unnecessary
    "axis_flip": "J1J2J3J4J5J6"                         # Unnecessary
}
```

6.6. Guide for Robot Model Parameter Verification

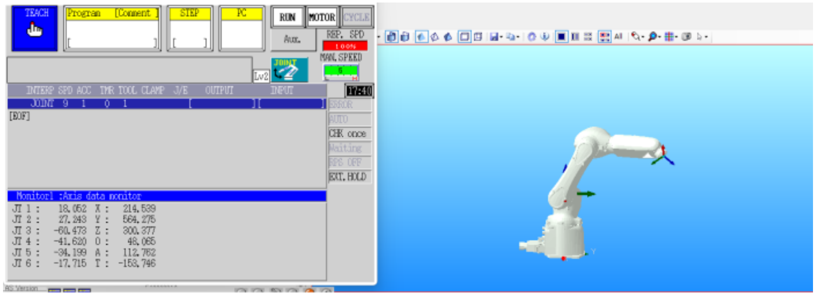
When the robot pose or tool pose of the simulated robot in the software is not consistent with those of the real robot, please refer to this section to verify the parameters of the robot model.

Example of Inconsistent Pose

A Kawasaki RS007N robot is used in the following example.

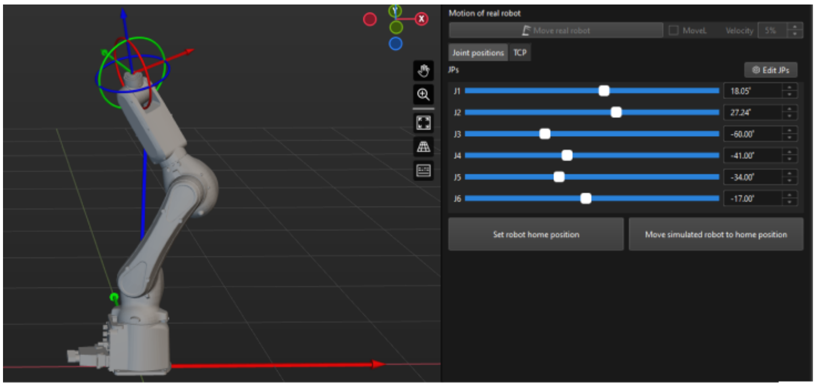
In the following example, the virtual robot displayed in the simulation software provided by the robot manufacturer represents the real robot.

1. Use the teach pendant to move the real robot to a random pose, and record the values of the JPs and tool pose.



```
INTERP SPD ACC  TMR TOOL CLAMP  J/E    OUTPUT        INPUT
  JOINT  9   1    0   1               [          ][          ]
[EOF]

Monitor1 :Axis data monitor
JT 1 :    18.052  X :   214.539
JT 2 :    27.243  Y :   564.275
JT 3 :   -60.473  Z :   300.377
JT 4 :   -41.620  O :    48.065
JT 5 :   -34.199  A :   112.762
JT 6 :   -17.715  T :  -153.746
```

2. Search for Kawasaki RS007N in the Robot Model Library and select the robot model.
3. Enter the JPs values of the real robot into the software, and the pose of the simulated robot is shown below.



JPs	
J1	18.05°
J2	27.24°
J3	-60.00°
J4	-41.00°
J5	-34.00°
J6	-17.00°

As you can tell from the two figures above, the robot poses are very different.

Verification Procedure

Please check the following items in order:

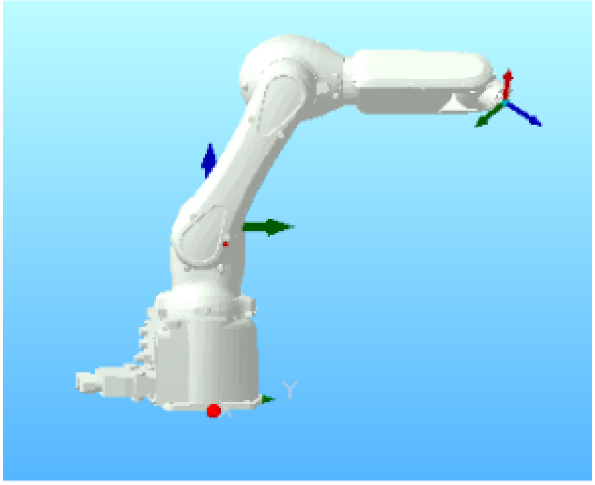
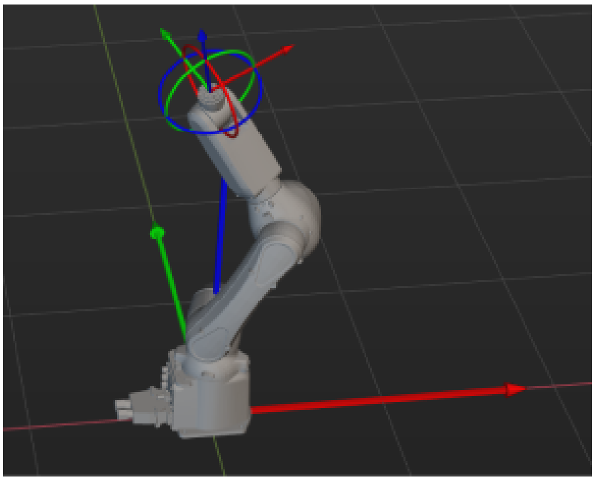
1. The orientations of the axes in the robot reference frame.
2. The rotation directions of the robot joints.
3. The poses of the robot joints.
4. The origin of the robot reference frame.

The items above are defined in the [robot]_algo.json file.

- Right-click the robot name in the project resource tree, and select **[Open Robot File Directory]** in the context menu to open the kawasaki_RS007N_algo.json file.
- Please refer to Description of [robot]_algo.json File for the description of the parameters in [robot]_algo.json.

Orientations of the Axes in the Robot Reference Frame

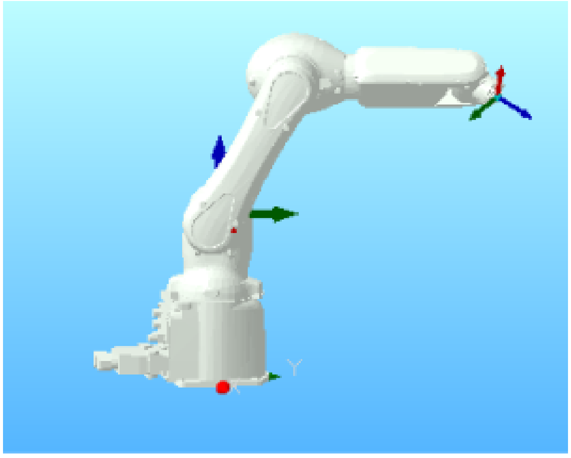
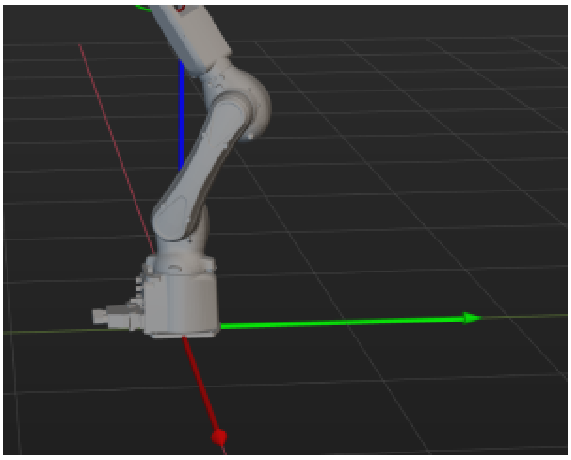
As you can see in the following figure, the real robot faces towards the positive Z-direction of its own robot reference frame, while the simulated robot faces towards the positive X-direction of its own robot reference frame.

	
Real robot	Simulated robot

Please skip this part if the orientations of the axes in the reference frame of the real robot are consistent with their simulated counterparts.

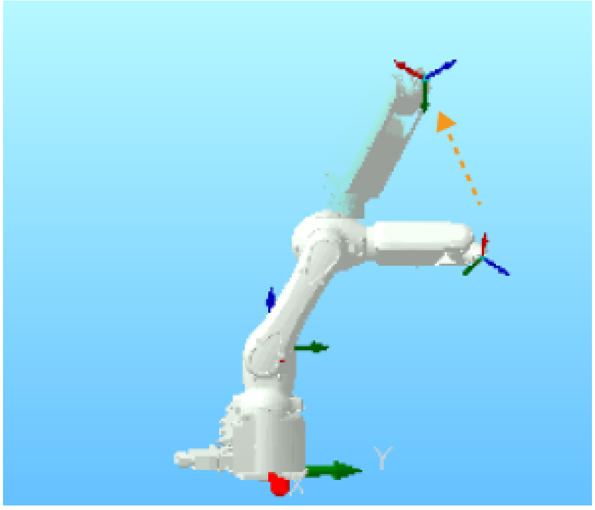
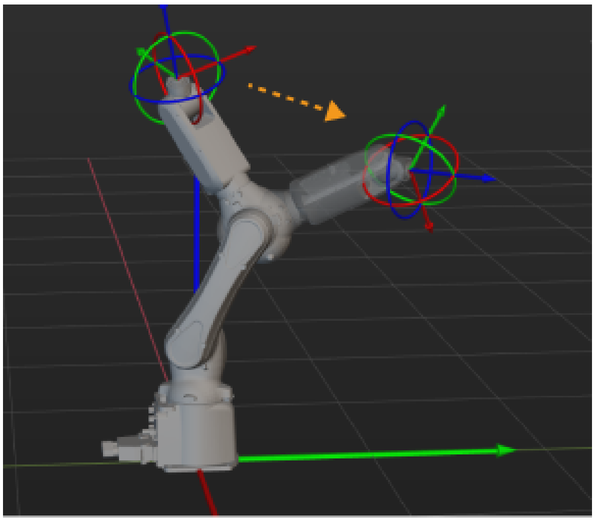
In this case, the X-axis and Y-axis in the robot reference frame of the simulated robot should be rotated 90° counterclockwise around the Z-axis. Please follow the steps below to resolve the difference:

1. Set the value of the robot_frame_in_robot_base_frame parameter in the kawasaki_RS007N_algo file to [0,0,0,0,0,-90] and then save the change.
2. Right-click the robot name in the project resource tree and select **[Reload Robot]** in the context menu.
3. Re-enter the JPs values of the real robot into the software, and compare the robot poses.

	
Real robot	Simulated robot

Rotation Directions of the Robot Joints

Adjust the joint positions on the teach pendant and in the software accordingly, and compare the directions the joints rotate for the real and simulated robot.

	
Real robot	Simulated robot

Please skip this part if the robot joint rotation directions of the real robot are consistent with their simulated counterparts.

In this case, the J3s of the real and simulated robot rotate in opposite directions. Please follow the steps below to resolve the difference:

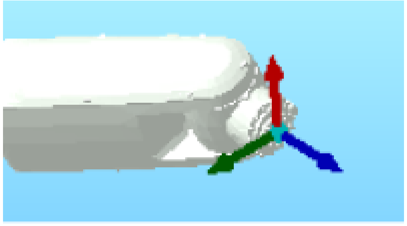
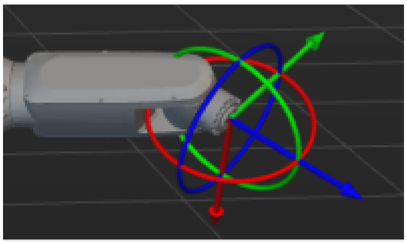
1. Modify the value of the axis_flip parameter in the kawasaki_RS007N_algo file from 100010 to 101010 and then save the change.

   Changing 0 to 1 or 1 to 0 can flip the rotation direction of the corresponding joint.

2. Right-click the robot name in the project resource tree and select **[Reload Robot]** in the context menu.
3. Rotate the joints of the robot separately and compare the axis rotation direction of the real robot with that of the simulated robot.

Poses of the Robot Joints

After correcting the joint rotation directions, re-enter the JPs or the real robot into the software and compare the poses of the real and simulated robot.

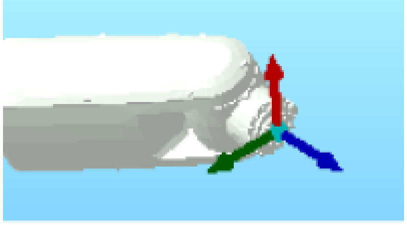
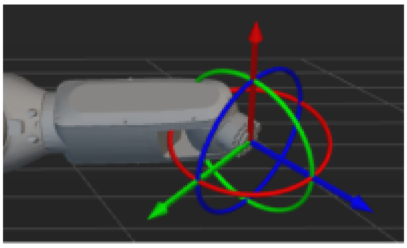
	
Real robot	Simulated robot

Please skip this part if the robot joint poses of the real robot are consistent with their simulated counterparts.

According to the above robot poses, the orientations of J6 differ by 180°. Please follow the steps below to resolve the difference:

1. Modify the value of the mastering_joints parameter in the kawasaki_RS007N_algo file from [0, 0, 90, 0, 0, -90] to [0, 0, 90, 0, 0, 90] and then save the change.
2. Right-click the robot name in the project resource tree and select **[Reload Robot]** in the context menu.

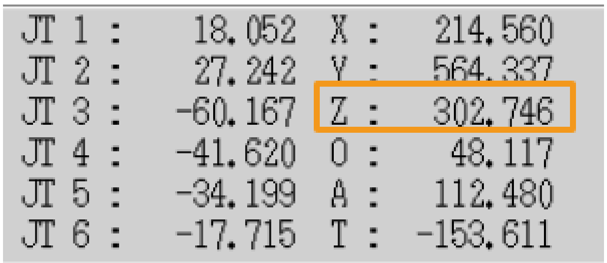
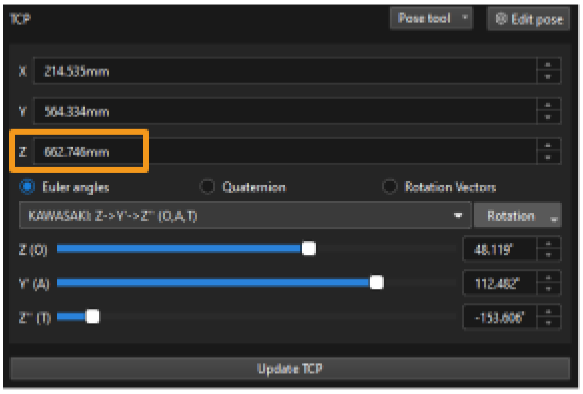
Re-enter the JPs values of the real robot into the software, and compare the robot poses.

	
Real robot	Simulated robot

Origin of the Robot Reference Frame

Before verifying the origin of the robot reference frame, please ensure that no end tool is installed on the real robot and there is no tool configuration in the software.

Compare the tool poses when the JPs of the real and simulated robot are the same.

	
Tool pose of the real robot	Tool pose of the simulated robot

```
JT 1 :    18.052  X :   214.560
JT 2 :    27.242  Y :   564.337
JT 3 :   -60.167  Z :   302.746
JT 4 :   -41.620  O :    48.117
JT 5 :   -34.199  A :   112.480
JT 6 :   -17.715  T :  -153.611
```

TCP

X 214.535mm

Y 564.334mm

Z 662.746mm

● Euler angles ○ Quaternion ○ Rotation Vectors

KAWASAKI: Z->Y'->Z'' (O,A,T)

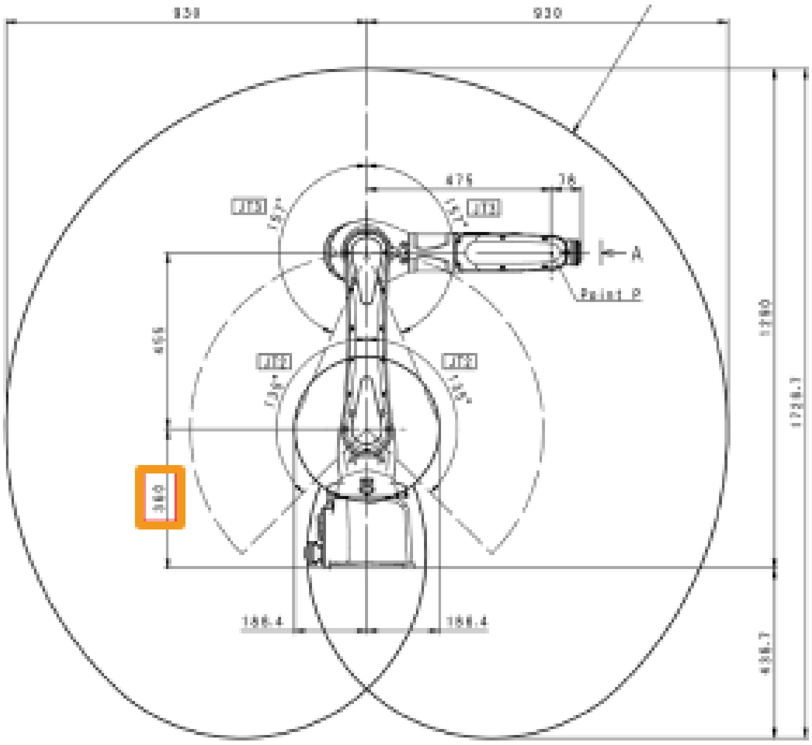
Z (O) 48.119°

Y' (A) 112.482°

Z'' (T) -153.606°

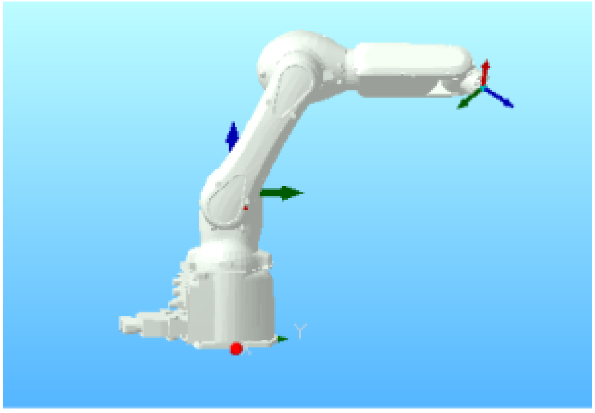
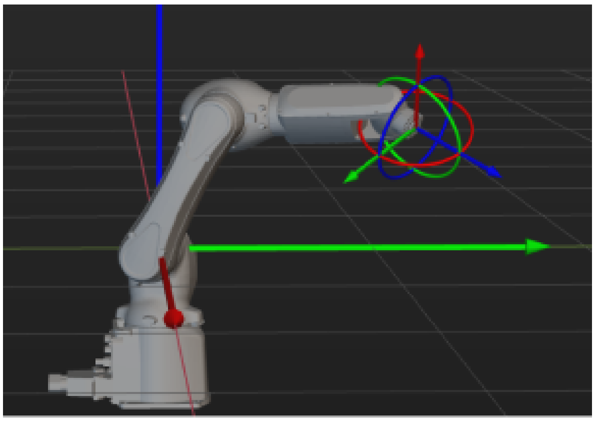
Please skip this part if the origin of the reference frame of the real robot is consistent with its simulated counterpart.

As you can tell from the above figures, there is a 360 mm difference between the Z-values of the tool poses. By referring to the DH specification, the value of DH1 is 360. All this indicates that the reference frame of the real robot is not located at the bottom of the robot but at the J2.



Please follow the steps below to resolve the difference:

1. Set the value of the robot_frame_in_robot_base_frame parameter in the kawasaki_RS007N_algo file to [0,0,0.36,0,0,-90] and then save the change.
2. Right-click the robot name in the project resource tree and select **[Reload Robot]** in the context menu.
3. Re-enter the JPs values of the real robot into the software, and compare the robot poses.

	
Real robot	Simulated robot

Final Check

After checking the above items, move the real robot to another pose, and enter the JPs at that pose into the software to check whether the poses of the real and simulated robot are consistent.

If the poses are consistent, you have completed the verification of robot model parameters.

6.7. Terminology

Mech-Mind Vision System

Mech-Mind Vision System is the full 3D vision-based solution provided by Mech-Mind Robotics, including Mech-Eye Industrial 3D Cameras Mech-Eye, the industrial personal computer (IPC) and Mech-Mind Software Suite. By integrating Mech-Mind vision system into the robot system, a complete vision workstation can be built to achieve the purpose of 3D vision guiding the robot to perform intelligent tasks.

Vision Processing: Mech-Vision

Software Function Structure

Solution

Solution is a collection of configurations and data of robot communication, vision processing, path planning and other functions, which are all integral components of a vision application.

Mech-Vision Project

Projects refer to Mech-Vision projects. One or more projects make up a solution. Projects cannot be used independently and must belong to a solution.

Step

Steps are the basics of a project. A step is a minimum algorithm unit for data processing. By connecting different steps in a project, you can achieve different data process tasks.

Procedure

A procedure is a combination of multiple steps. There are often consistent or similar algorithm processing processes in different projects. By encapsulating and combining these fixed algorithm processing steps, they can be easily reused. .

Parameter Recipe

Parameter recipes are sets of parameter settings that need to be adjusted according to different situations for the same project. With parameter recipes, you do not need to build multiple projects with the same logic and configure their parameters differently to meet different on-site requirements. Instead, you will only need to switch between parameter recipes in one project to make it applicable to various scenarios and therefore the productivity can be increased.

Before Processing

Scene

Everything captured by the camera, including the background, bins, objects, etc.

Target Object

The objects are things in the scene on which the poses need to be calculated and which need to be processed/picked by the robot. Objects can be workobjects, partitions, bins, etc. depending on requirements.

Background

The scene without target objects.

Region of Interest (ROI)

The region excluding surrounding parts unnecessary for vision data processing in the scene. Such parts may be background objects, pallets, bin edges, etc. An ROI can be either set by selecting a 3D box on the point cloud, or a 2D box on the depth map/image.

Calibration Pose

The robot pose for calibration. It is in the form of TCP, flange pose or joint positions as required.

Calibration Circle

The circles on the calibration board.

During Processing

3D Matching

The process of fitting an object point cloud model onto the point cloud of an object in the scene, to find the poses of the objects in the scene. An object point cloud model reflects object shape and features and carries the defined object pose.

Model

The target object point cloud model used in 3D matching. The model can be made by the Matching Model and Pick Point Editor, either from the target object point cloud or an STL model.

Surface Model

The model used to match target objects by their surface features. A surface model includes the object surface feature parts and excludes other unnecessary parts.

Surface Matching

When the surface of the target object has obvious recognizable features (such as crankshafts, rotors, and steel bars), it is recommended to use surface matching.

Edge Model

The model used to match target objects by their edge features. An edge model includes the object edge feature parts and excludes other unnecessary parts.

Edge Matching

When the object is relatively flat but present clear and regular edge features under the camera, edge matching is recommended. Such objects include panels, track shoes, connecting rods, brake discs, etc. The Matching Model and Pick Point Editor can help generate an edge model.

2D Matching

The process of fitting a 2D template that reflects object shape and features and carries the defined object pose onto to the image of the object in the scene, in order to determine the object's poses within the scene.

2D Template

The 2D shape that reflects object shape and features used in 2D matching.

Deep Learning

In Mech-Vision, deep learning is a technique usually used to recognize and classify objects and find object poses. A trained deep learning model is exported from Mech-DLK and used by deep learning steps in Mech-Vision.

Inference

Use a trained deep learning model to make predictions on the actual vision data to obtain information including poses, classification labels, etc.

After Processing

Vision Result

A vision result is the output by the execution of a Mech-Vision project at a time. A vision result may contain multiple vision points and other data.

Vision Point

A vision point refers to a calculated pose and its associated data, as follows.

Object Pose (vision pose)

The pose of the object calculated by the Mech-Vision project. A pose contains the position information (X, Y, Z coordinates) and the orientation information (either in Euler angles or in quaternions).

Label

The string label attached to each pose. Usually for indicating the object type.

Object Dimensions

The dimensions of the corresponding object of the object pose. In the form of (length, width, height), or (radius, height), or others.

Pick Point

The pose on the object on which the robot can pick the workobject. In some cases, the pick point is equivalent to workobject pose. The pick point and the robot’s picking pose (in the form of TCP) usually coincide but have opposite Z axes.

The information contained in a vision point may include other custom types of data associated to the object pose in the vision point, such as recommended robot velocity, pose offset, etc. You can customize the data types at step “Procedure Out” in Mech-Vision.

Highest Layer

The part in the scene defined by a specified range of height, usually on the top of the scene containing point clouds of the target objects most convenient for processing/picking. Highest layer point cloud extraction is frequently used in carton palletizing/depalletizing.

Eye in Hand (EIH)

EIH is the setup where the camera is installed on the flange located on the end of the robot.

Eye to Hand (ETH)

ETH is the setup where the camera is installed on a bracket independent from the robot.

Intrinsic Parameter

Intrinsic parameters are measures of the properties of a camera itself, including the focal length, the lens distortion, etc. These parameters are usually calibrated and stored in the camera before the camera leaves the factory.

Extrinsic Parameter

Extrinsic parameters define how poses are transformed between the camera reference frame and the world reference frame.

Point Cloud

A point cloud is a set of data points on the appearance of a product obtained by measuring instruments.

Mask

Specified images, shapes, or objects are usually selected to mask the image (partial or all) to control the area to process or the processing. The particular image or object used for masking is called a mask.

Static Background

The background and information of the collected depth maps and color images.

Visualized Output

Visual output refers to displaying the output results of steps that can be visualized in Mech-Vision.

Procedure

A procedure is a customizable special step that combines multiple steps.

Instance Segmentation

Instance segmentation is an image processing method that classifies images pixel by pixel, and objects are assigned different values to distinguish them regardless of whether they are of the same category.

Image Classification

Image classification is an image processing method that classifies object images based on the object categories in them.

Morphological Transformation

Morphological transformations refer to some simple operations on an image, such as erosion, dilation, etc.

Erosion

Erosion is one of the fundamental operations in morphological image processing. It “erodes” spots with high brightness in the input image and outputs an image with reduced bright regions.

Dilation

Dilation is one of the fundamental operations in morphological image processing. Contrary to erosion, it “expands” spots with high brightness in the input image and outputs an image with enhanced bright regions. Please note that erosion and dilation processes are not reversible.

Confidence Interval

Confidence intervals are used to estimate the range of parameter values.

Confidence

Confidence refers to the reliability of the confidence interval.

Normal

The vector represented by a straight line perpendicular to a plane is the normal of the plane.

Threshold

Threshold refers to the lowest or highest value that an effect can reach.

Boolean

Boolean is a data type. A Boolean value is either “True” or “False”.

Hash

The hash value refers to the smaller data formed by mapping a long piece of data through a hash algorithm. It can be simply understood as the ID of a piece of data.

Robot Path Planning: Mech-Viz

Software Function Structure

Mech-Viz Project

Projects refer to the robot path planning projects created in Mech-Viz. Once you have completed the necessary setup of the project in Mech-Viz, you can use the project to plan a path and guide the robot to move. All the configurations of the project are stored in the folder with the same name as the project.

Project Resources

Project resources refer to various basic resources used in the project, including the robot, tools, workobjects, and scene objects.

Workflow

The workflow refers to the robot motion control program created in Mech-Viz in the form of a flowchart.

Step

A Step refers to a function module of robot programming.

Procedure

A procedure contains multiple connected steps.

Robot&Object Settings

Simulation Space

The space containing all contents involved in the workflow, including the robot, workobjects, bins, and other objects.

Scene Object

Scene objects refer to any solid bodies aside from the robot and the workobjects, including the bin, the pallet, elements of the working platform, etc.

Visualization Model

The 3D solid body simulation for visualizing the corresponding thing in the space. It will not be used for collision detection.

Collision Model

The 3D solid body simulation for detecting collisions of the corresponding thing in the space during path planning.

Workobject

The object that the robot needs to process/pick.

Workobject Symmetry

The property that, after a rotation around the rotational symmetry axis for a certain angle, the appearance of a workobject is considered to be coincident with that before the rotation.

N-fold Symmetry

The property that after a rotation by an angle of 360°/N, the workobject shape is considered unchanged.

Number of Symmetry Folds

The value of N in the definition of “N-fold symmetry”.

Symmetry Angle

The value of 360°/N in the definition of “N-fold symmetry”.

> ℹ number of symmetry folds * symmetry angle = 360°

TCP

TCP refers to the tool center point. The TCP is on the tip of the tool. In order to complete tasks such as picking workobjects, we usually say that the robot should move to a specific point in space, which actually means its TCP should move to that point.

Workobject Pick Point

The pose on the object on which the robot can pick the workobject. In some cases, the pick point is equivalent to the workobject reference frame. In other cases, the pick point is obtained by offsetting the workobject reference frame, especially when one workobject has multiple pick points. The pick point and the robot’s picking pose (in the form of TCP) usually coincide but have opposite Z axes.

Picking Relaxation

The permission to rotate the tool around the workobject frame to make attempts at different angles to facilitate picking.

Robot

A robot refers to a system composed of rigid bodies connected by joints.

Tool

The device mounted on the robot end that performs processing/picking jobs.

Depalletizing Vacuum Gripper

Rectangular suction cups used for depalletizing. Multi-block suction cups are supported.

Array Gripper

An array of tools at the end of the robot that can be used to perform pick-or-place tasks concertedly.

Edge-corner ID

The numbers used to identify the specific edges or corners of suction cups on a vacuum gripper.

Path Planning

Path

A path is a sequence of waypoints that the robot needs to reach one by one.

Waypoint

A point (presented as a robot pose in JPs or TCP) in the path that the robot needs to reach. A waypoint can contain additional information including label, motion type (linear/joint move), velocity, acceleration, etc.

Home Position

A default robot pose that the robot should return to before the start of a job or after the completion of a job.

Initial Pose

The robot pose before the beginning of a job. The path planning needs to accept the initial pose and take it into consideration.

Trajectory

A trajectory is the record of a sequence of waypoints that robot has physically reached, carrying the timestamps.

Robot Pose

The status of the robot in the 3D space, presented in the form of TCP, JPs, or flange pose.

Workobject Waypoint

The waypoint at which the robot processes/picks the workobject.

Picking Waypoint

The waypoint at which it is planned that the robot should pick a workobject.

Picking Pose

The pose of the robot when it picks the workobject.

Placing Waypoint

The waypoint at which it is planned that the robot should place a workobject.

Placing Pose

The pose of the robot when it places the workobject.

Robot Singularity

The robot singularity refers to a situation in which the robot joint speeds theoretically need to be infinite (although not achievable in practice) when the robot’s end effector reaches a specific position and angle. This results in a reduction in the robot’s degrees of freedom.

Singularity Threshold

A singularity threshold is the maximum joint angular velocity that the robot is allowed to reach. It is used to check singularities in Mech-Viz.

Singularity Vel Decelerate Ratio

Singularity velocity decelerate ratio refers to the lowest accepted velocity decelerate ratio that a robot should have when the robot encounters a singularity.

Joint Motion

Joint motion refers to any movement of the tool between two target points without path control or posture control.

Linear Motion

Linear motion refers to the motion type of the robot in which the tool moves in a straight line between two target points.

Euler Angles

Euler angles are used to describe the orientation of an object in the 3D space. The object’s rotation in the 3D space can be denoted by 3 angles, i.e., pitch, yaw, and roll.

Quaternions

Quaternions is a parameter group consisting of four quaternion values that define the 3D orientation of an object. Unlike Euler angles, it solves the problem of Euler angles in which the rotation system is limited to rotating only on the vertical axis.

Point Cloud Cube

The cube simulated around each point in the point cloud to define point cloud volume for collision detection.

Point Cloud Cube Size

The edge length of a point cloud cube.

Collision Volume

During simulation, if one party involved in the collision detection is the point cloud, the collision volume is the number of point cloud points that overlap with the other party’s collision model times the volume of a point cloud cube.

Communication

Endian

Big-endian

Big-endian is an order that high-order bytes are arranged at the low-address end of memory, and low-order bytes are arranged at the high-address end of memory.

```
Low address        High address

0x12 | 0x34 | 0x56 | 0x78
```

Little-endian

Little-endian is an order that low-order bytes are arranged at the low-address end of memory, and high-order bytes are arranged at the high-address end of memory.

```
Low address        High address

0x78 | 0x56 | 0x34 | 0x12
```

Others

Flange

The flange is the part that connects two shafts and is mainly used for attachment.

License Dongle

A license dongle is an encryption product used to authorize software.

Industrial Personal Computer (IPC)

An industrial personal computer is a ruggedized computer intended for industrial purposes. It can be used as an industrial controller.

Programmable Logic Controller (PLC)

A PLC is a logic controller that is used for automated controls.

Takt Time

This is the overall processing time taken from capturing the image to the robot completing a certain task. Specifically, it includes the time required to capture the image on the camera, process in Mech-Vision, plan in Mech-Viz, and complete the motion on the robot.