Mech-Vision Manual

Mech-Mind

Jul 01, 2022

CONTENTS

1 Mech-Vision Quick Facts 3

2 Getting Started with Mech-Vision 5

3 Typical Applications 16

4 Guide to Steps 69

5 Common Procedures 438

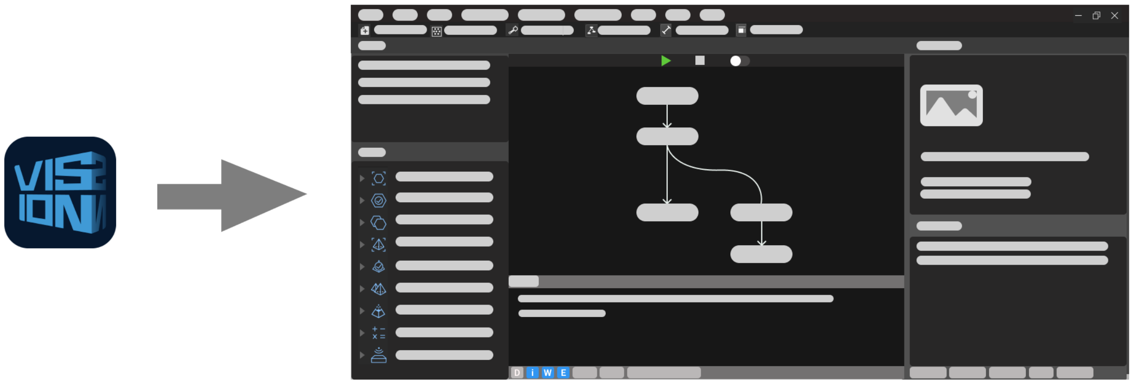
6 Hand-Eye Calibration Guide 542

7 Supplementary Tools 655

8 Measurement Mode Guide 780

9 Advanced Guide 790

10 FAQ 841



Mech-Vision is a **graphical machine vision software** independently developed by Mech-Mind Robotics. Working in conjunction with Mech-Viz and Mech-Center, Mech-Vision processes images collected from industrial sites, and then sends processed visual data to the robot control system in order to accomplish various vision-based intelligent robot tasks.

Read the sections below for **an overview of Mech-Vision**.

Mech-Vision Quick Facts

Getting Started with Mech-Vision

Read the section below to learn more about **typical application projects** of Mech-Vision.

Typical Applications

Steps are fundamental units of a project. Read the sections below to learn about **Steps**.

Guide to Steps

Common Procedures

To solve practical problems more efficiently, Mech-Vision provides various supplementary tools. Read the sections below to learn about **Supplementary Tools**.

Hand-Eye Calibration Guide

Supplementary Tools

Measurement Mode Guide

For those who are familiar with the basics of Mech-Vision, the **advanced guide** below provides more information on features and typical application projects.

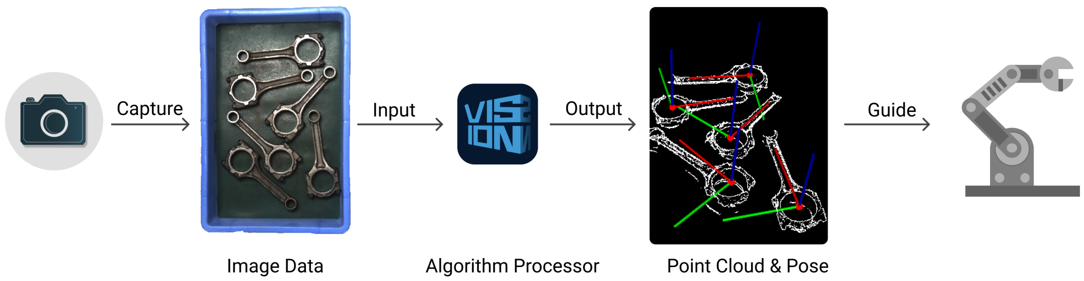
Advanced Guide

Please refer to the section below for **solutions to common problems**.

FAQ

MECH-VISION QUICK FACTS

Mech-Vision is a **graphical machine vision software** independently developed by Mech-Mind Robotics. Working in conjunction with Mech-Viz and Mech-Center, Mech-Vision processes images collected from industrial sites, and then sends processed visual data to the robot control system in order to accomplish various vision-based intelligent robot tasks.

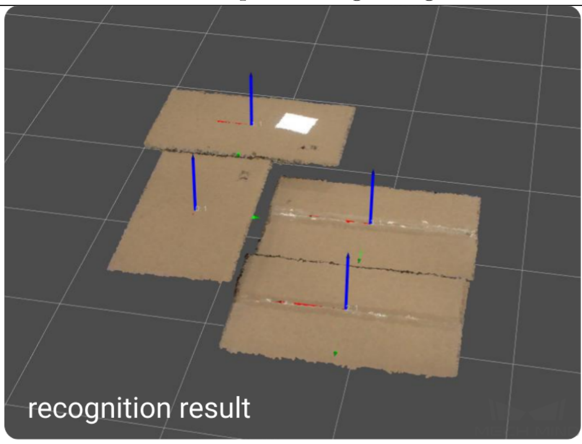


Point Cloud is a set of data points that represent the surface of an object.
Pose is the combination of positional and orientational information used to guide the robot to pick.

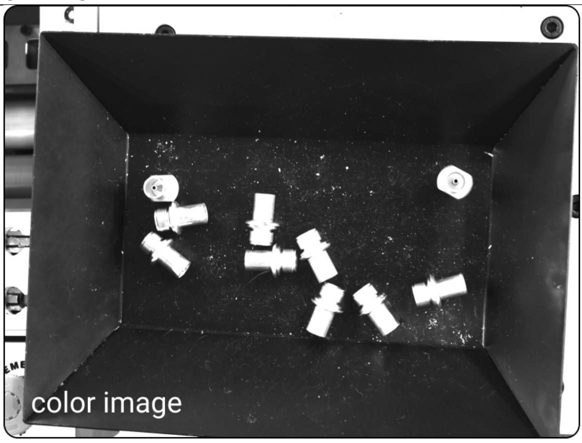
Typical Applications

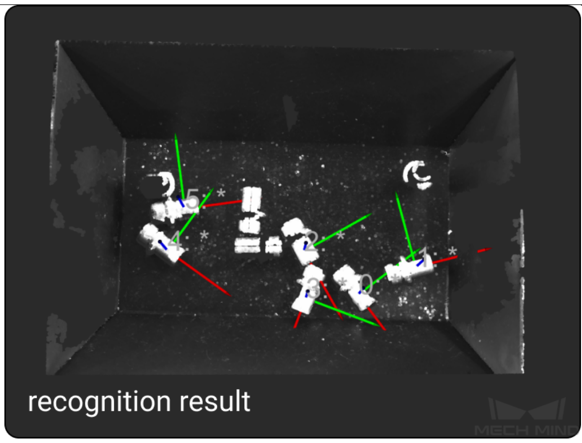
Mech-Vision can solve different problems in different applications.

Depalletizing
Segment individual boxes from closely stacked boxes and calculate the poses for guiding the robot

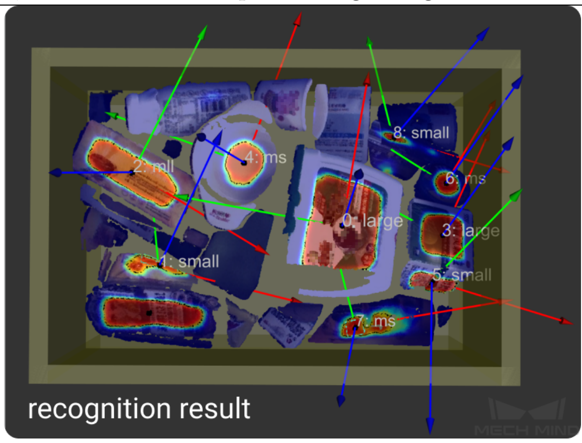


Machine Tending
Recognize pickable workpieces from a pile of randomly placed workpieces and calculate the poses for guiding the robot

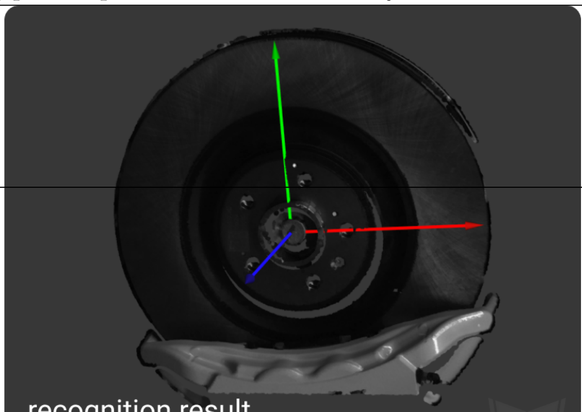




Piece Picking
Recognize objects of varied shapes, sizes, and colors, and calculate the poses for guiding the robot



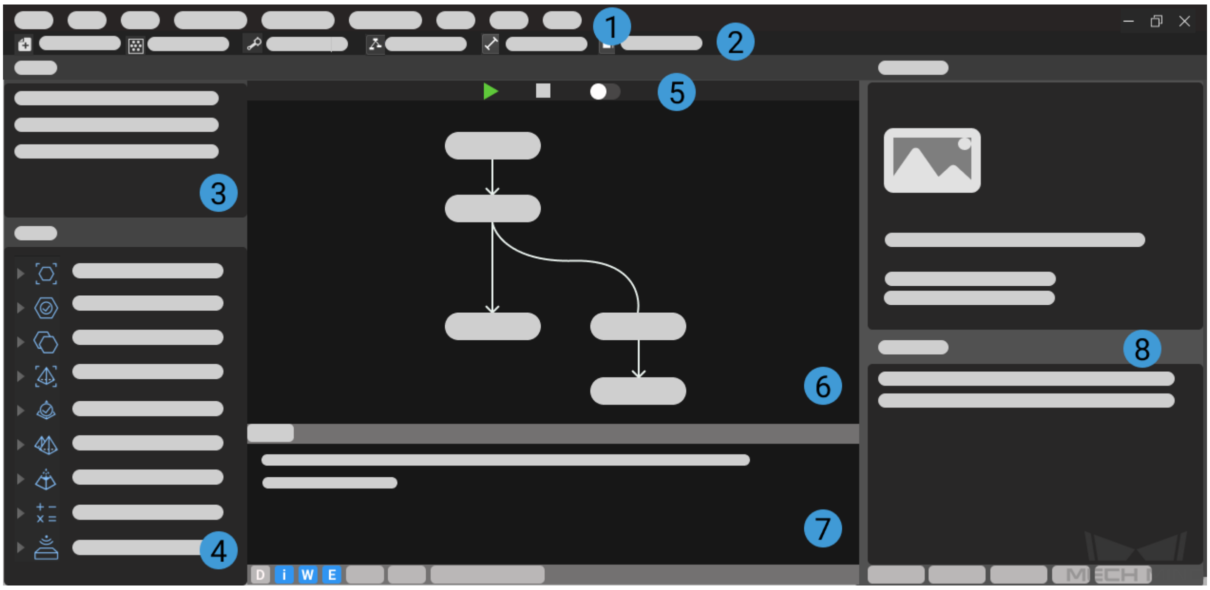
High Precision Locating
Obtain high-precision pose of the object, providing precise positional data of the object to the robot



CHAPTER

TWO

GETTING STARTED WITH MECH-VISION



The user interface of Mech-Vision consists of the following parts:

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 projects.

Options	Description	Shortcut
New Project	Create a new empty project	Ctrl+N
Open Project	Open an existing project	Ctrl+O
Open Recent	Show recently opened projects. Click on a project name to open it	N/A
Open Executable File In Explorer	Open the installation folder of Mech-Vision	N/A
Save Project	Save the changes to the current project	Ctrl+S
Save Project To JSON	Save the project, and save the .vis file as a .json file	N/A
Save Project As	Save the project under a specified directory	Ctrl+Shift+S
Exit	Close the project and exit Mech-Vision	Ctrl+Q

Edit

Options	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 Mech-Vision.

Options	Description
Projects List	Display/Hide the Projects panel; checked by default
Step Library	Display/Hide the Step Library panel; checked by default
Debug Output	Display/Hide the Debug Output panel; checked by default
History	Display/Hide the History panel
Project Assistant	Display/Hide the Project Assistant tab
Step Quick Info	Display/Hide the Step Quick Info tab
Step Comment List	Display/Hide the Step Comment List tab
Step Parameters	Display/Hide the Step Parameters tab
Step Input Source Selection	Display/Hide the Step Input Source Selection panel
Measurement Output	Display/Hide the Measurement Output panel
Log	Display/Hide the Log panel; checked by default
Toolbar	Display/Hide the Toolbar; checked by default

Typical Applications

Create projects for typical applications including palletizing, deplattizing, machine tending, etc.

Options	Description	Shortcut
New Typical Application Project	Create a typical application project by following the guidance in the pop-up window	Ctrl+Shift+N

Camera

Options	Description	Shortcut
Camera Viewer	Capture and save images	Ctrl+Shift+V
Camera Calibration	Standard: standard calibration process Quick: quick calibration using existing parameters	N/A

Deep Learning

Options	Description	Shortcut
Deep Learning Server	Start/Stop deep learning server(s).	Ctrl+Alt+D

Toolkit

Providing supplementary tools that are commonly used for debugging.

Options	Description
Matching Model and Pick Point Editor	Generate point cloud model, edit point cloud model, and add pick point
Model Editor Tools (old)	Include the old tools for point cloud model and pick point, now replaced by Matching Model and Pick Point Editor
Data Playback	Replay the image data in the project for identifying any error in images
Plugin Tools	Test Project Precision: used to calculate error rate rate Mech Glue Wizard: acquire images from the camera and generate trajectory images used for labeling; save image, trajectory, and point cloud files

Settings

Providing commonly used settings.

Options	Description	Shortcut
Set Mech-Center Address	Set the IP address of Mech-Center	N/A
Lock Project	Check the box to lock opened projects	N/A
Options	Change common settings like language, unit, etc	Ctrl+Shift+O
Log Level	Select log level, and information of the selected level and above will be logged	N/A

Help

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

Options	Description
About	Check current version information
Change Log	Open the Change Log in browser
Manual	Open Mech-Vision User Manual in browser

2. Toolbar

Providing quick access to commonly used tools and buttons for shifting project mode.

Option	Description
New Typical Application Project	Quickly create a typical application project
Camera Calibration (Standard)	Start hand-eye calibration (standard mode)
Matching Model and Pick Point Editor	Generate, edit, and add pick point(s) to model point clouds of the target projects
Standard Editing Mode	The default mode of Mech-Vision
Measurement Mode	Used for measurement and defect detection in the 3C industry
Operator Interface (Custom)	Mech-Vision enters this mode when a typical application project is created

3. Projects List

This panel displays the name(s) of opened project(s).

4. Step Library

This panel provides all the Steps that can be used to build a project.

5. Project Toolbar

Providing buttons for running, stopping and debugging the project.

Option	Description
Run	Run the project
Stop	Stop the project
Debug Output	Turn the debug output on/off

6. Graphical Programming Workspace

This is where you can connect and debug the Steps of a project. Only available in standard editing mode.

7. Log

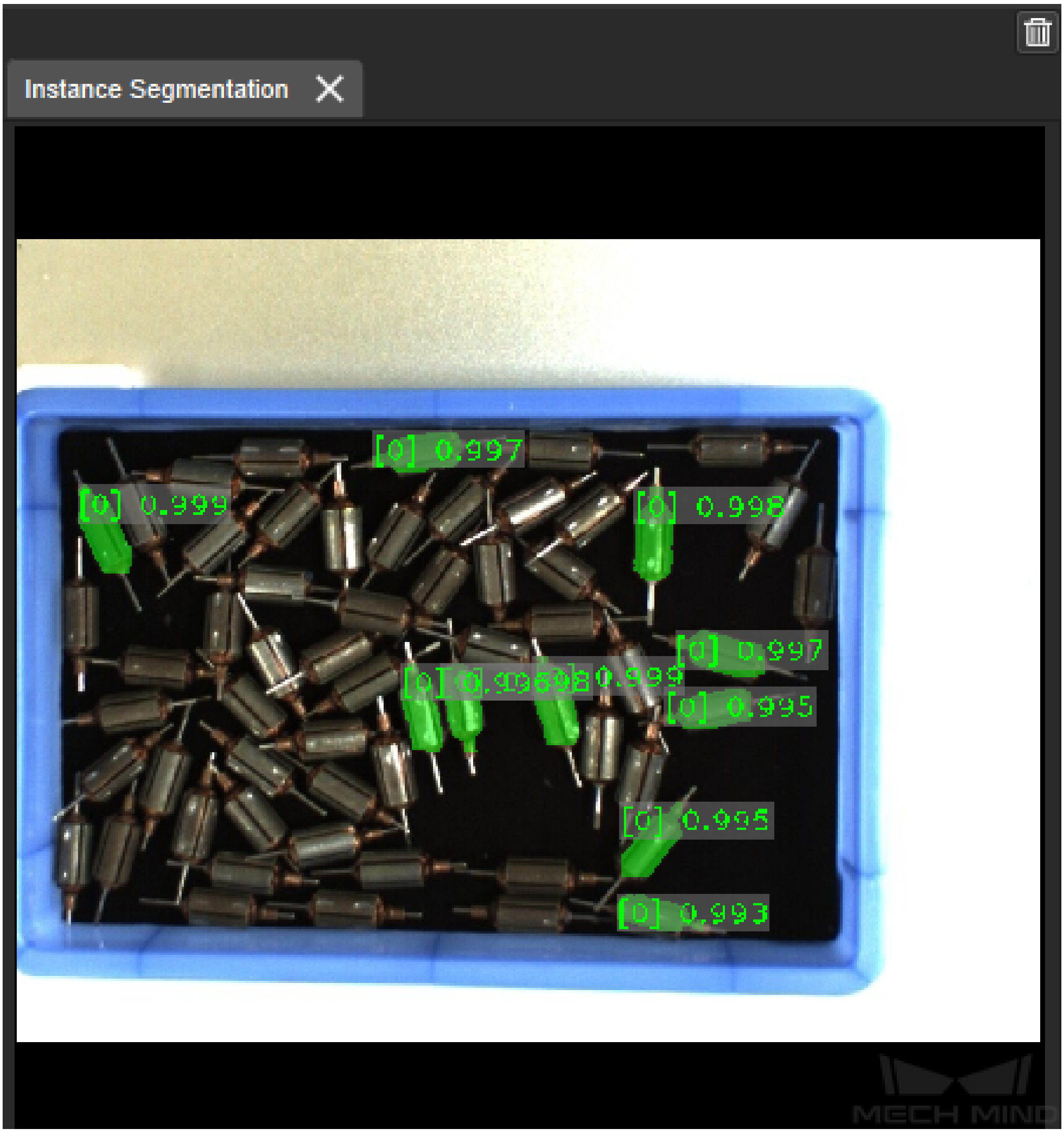
This panel displays the log in real-time while running a project, where you can check the execution record of a certain moment.

8. Project Configuration Panel

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

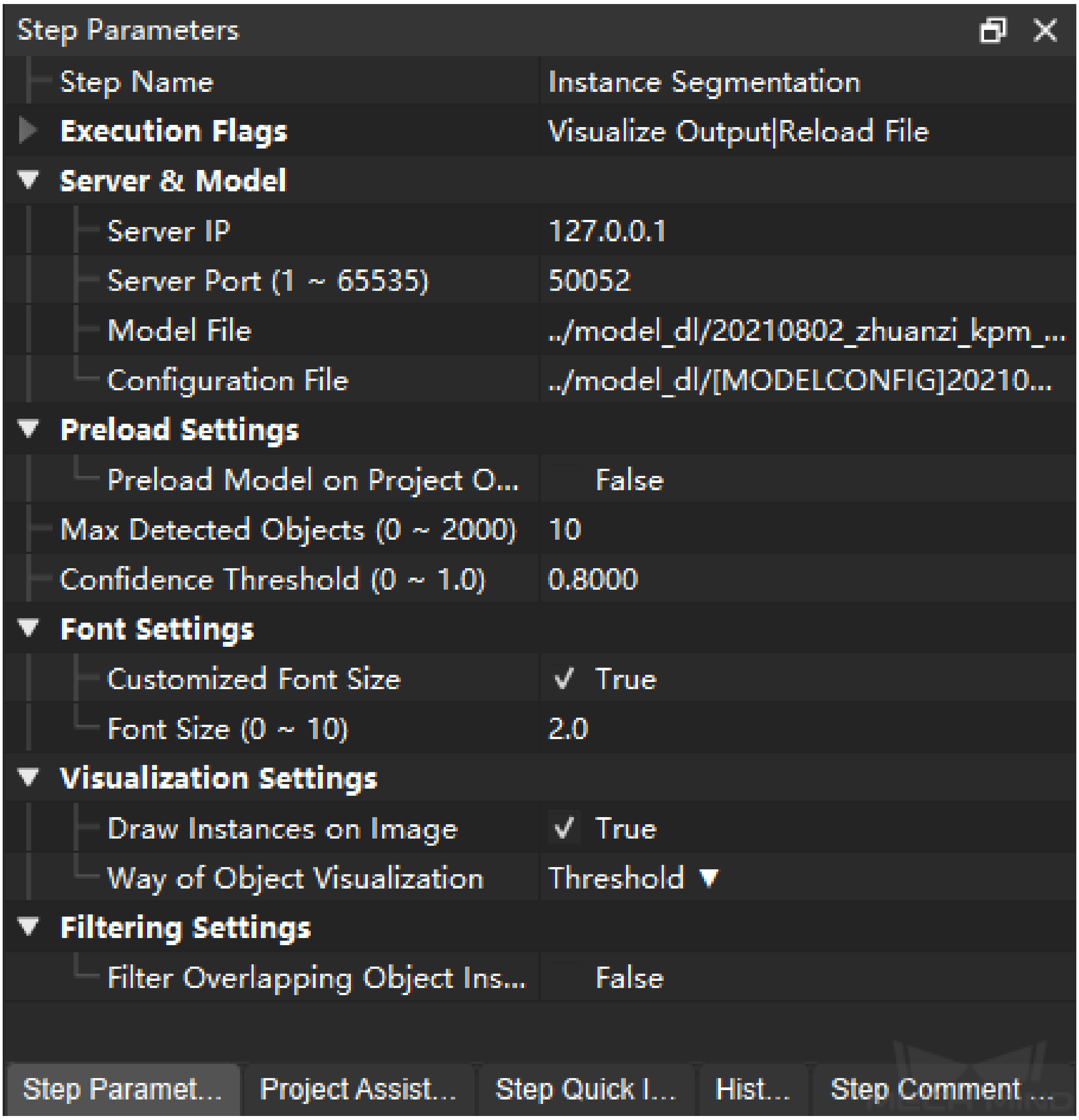
Debut Output

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*.



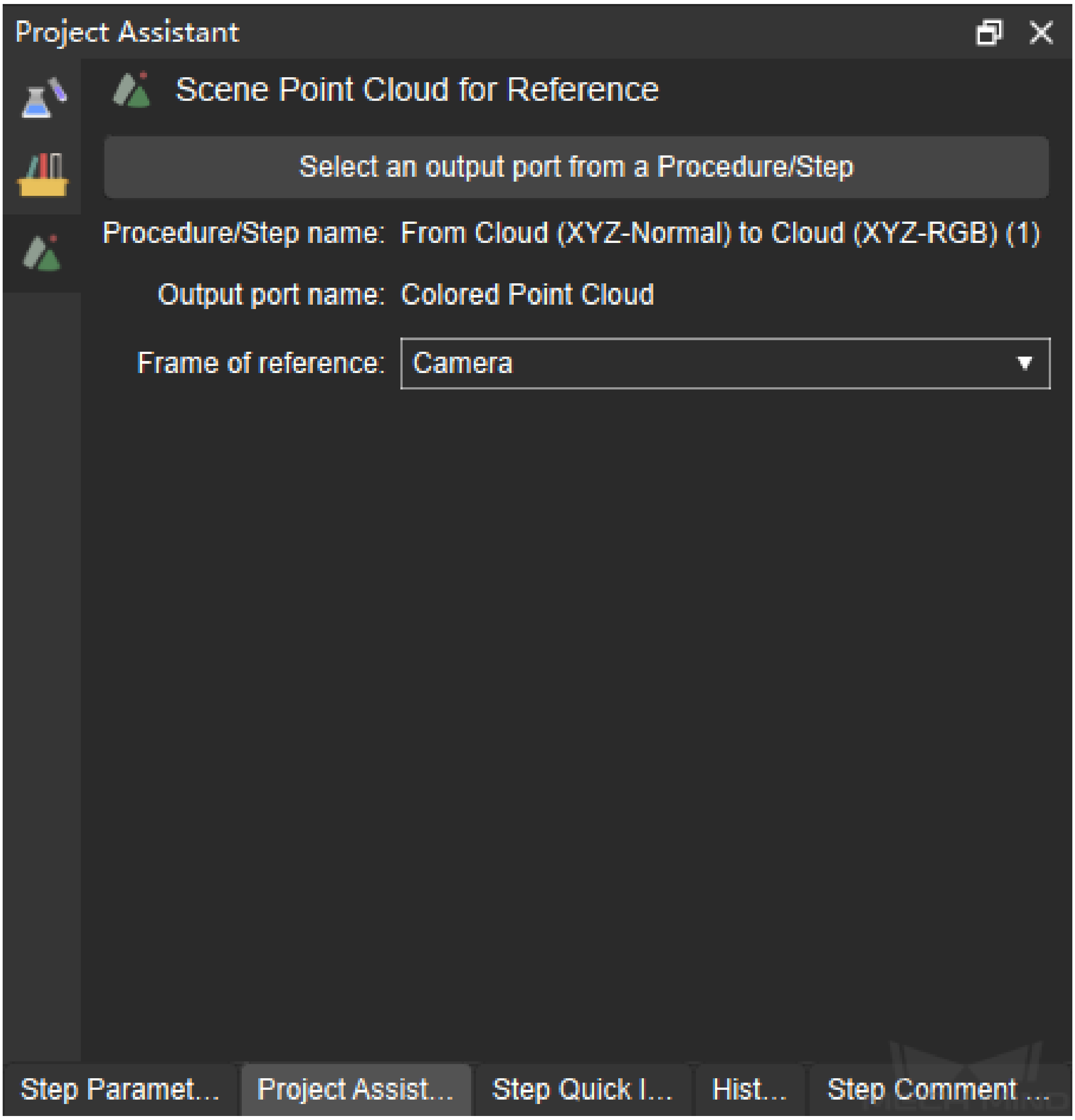
Step Parameters

This tab displays the common parameters of the selected Step.



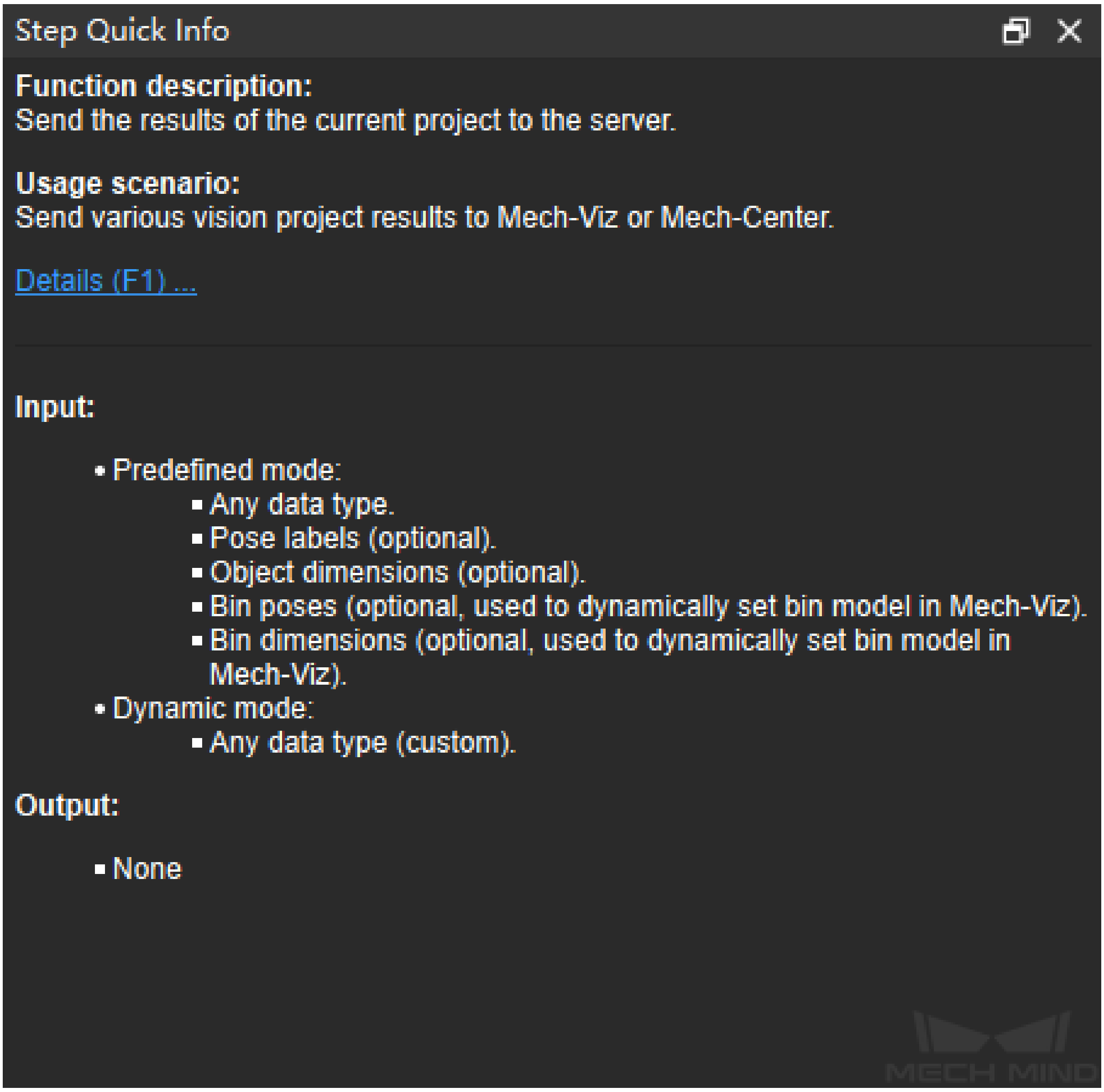
Project Assistant

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



Step Quick Info

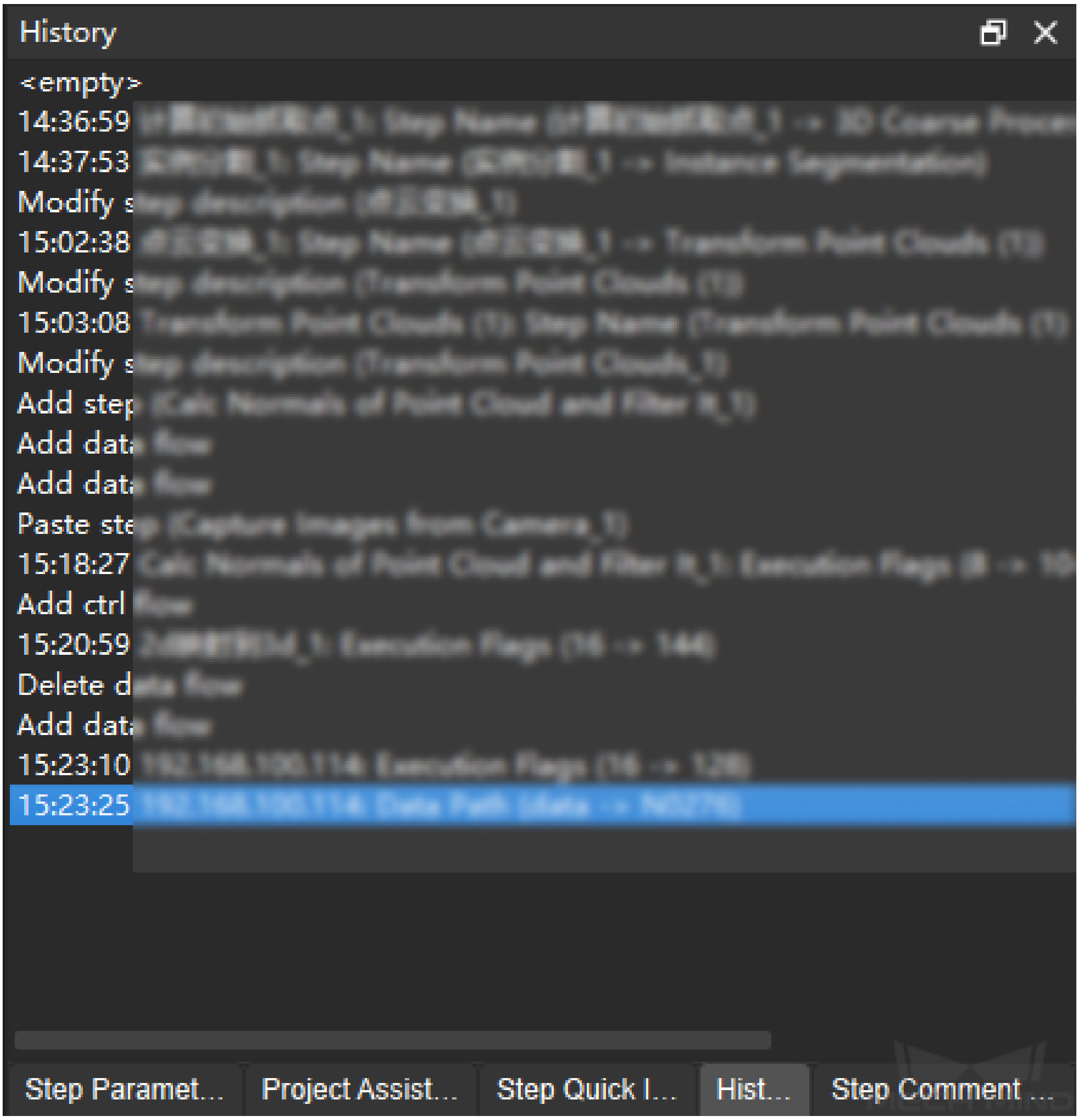
This tab displays a description of the function, usage scenario, input and output of the selected Step, as shown below.



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

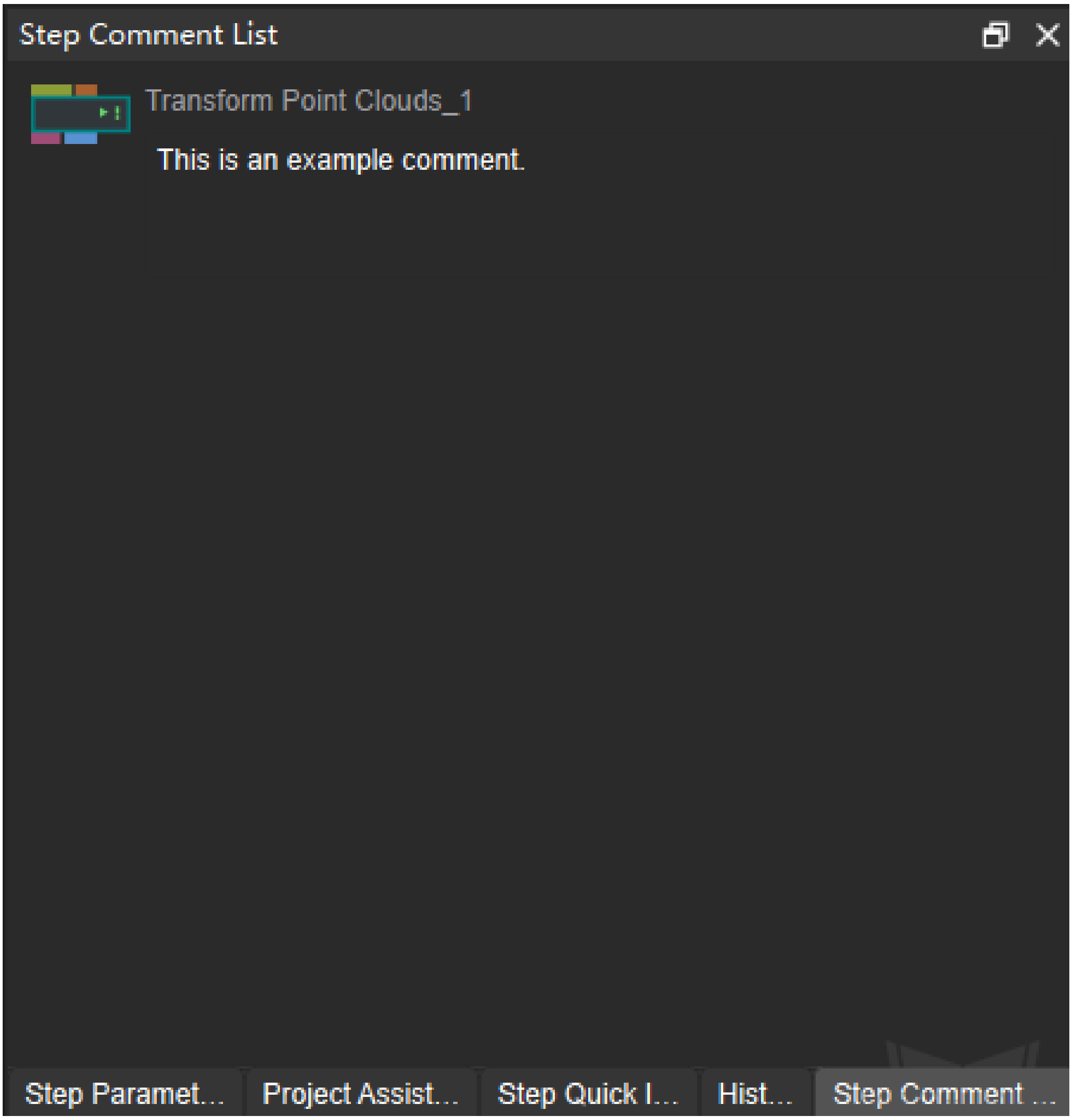
History

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



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*.

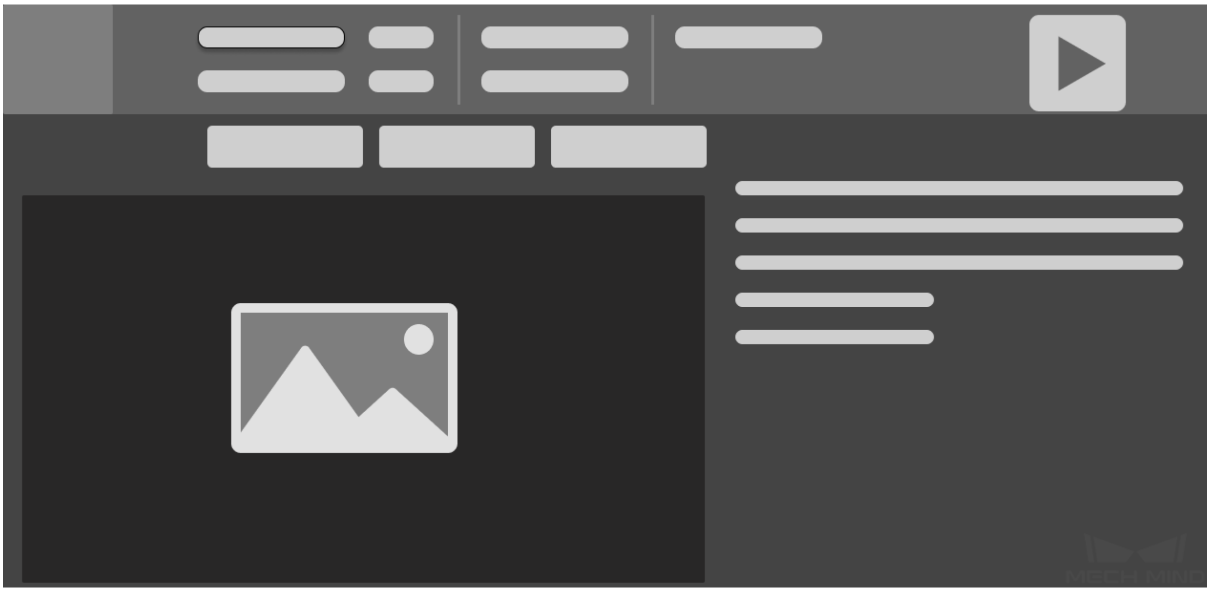


Tip: If you need to use the Step Input Source Selection or Measurement Output tabs, check the corresponding options in the View menu in the Menu Bar.

How to quickly create a project in Mech-Vision?

Click on *New Typical Application Project* in the Toolbar to create a project in the following window:

After the project is deployed, Mech-Vision will display the following operator interface:



The Operator Interface can be used to:

1. Check the execution status of the project, such as the top-level point cloud, deep learning results, poses, etc.
2. Adjust parameters conveniently to optimize output results.

Please refer to *Typical Applications* for Mech-Vision's typical application projects in different scenarios.

TYPICAL APPLICATIONS

To create a new typical application project, please go to *Typical Applications* → *New Typical Application Project* on the Menu bar.

To successfully run a project in the field, please make the following preparations:

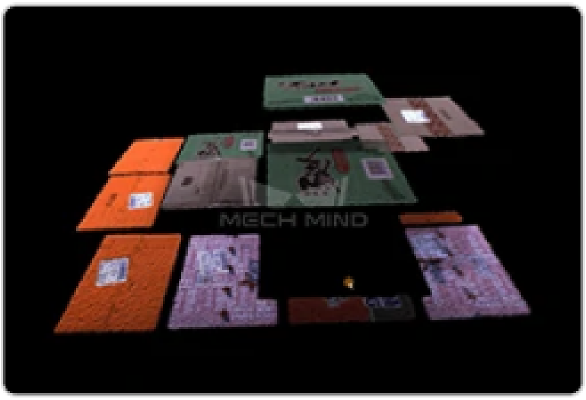
- Mount, secure, and wire the camera to make sure the software can connect to the camera. Please see Mounting the Camera and Connect a Camera for details.
- Prepare a calibration board, which should have clear circle grid patterns and no noticeable scratches or bending deformation.
- Preparation works are slightly different between different projects. The other types of preparation work will be elaborated in corresponding sections below.

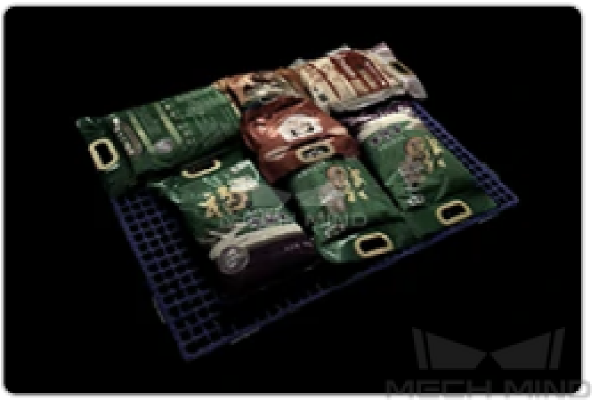
Practical guides corresponding to different application scenarios will be provided later, which will involve:

1. Preparation
2. Project deployment
3. Running and debugging

Please refer to different sections based on actual needs of the project.

Depalletizing





Applicable to the logistics industry. Usually used to recognize objects as cartons, sacks, etc. A variety of deep learning algorithms will be used in the project, and objects can be recognized either when they are closely stacked or placed randomly.

3.1 Sacks

In sack depalletizing scenarios, instance segmentation is usually utilized to identify each sack in the image and recognize poses of the sacks. Mech-Mind Robotics provides a super model specifically used for sack depalletizing. It can be directly applied in Mech-Vision to correctly segment most of the sacks without deep learning training.

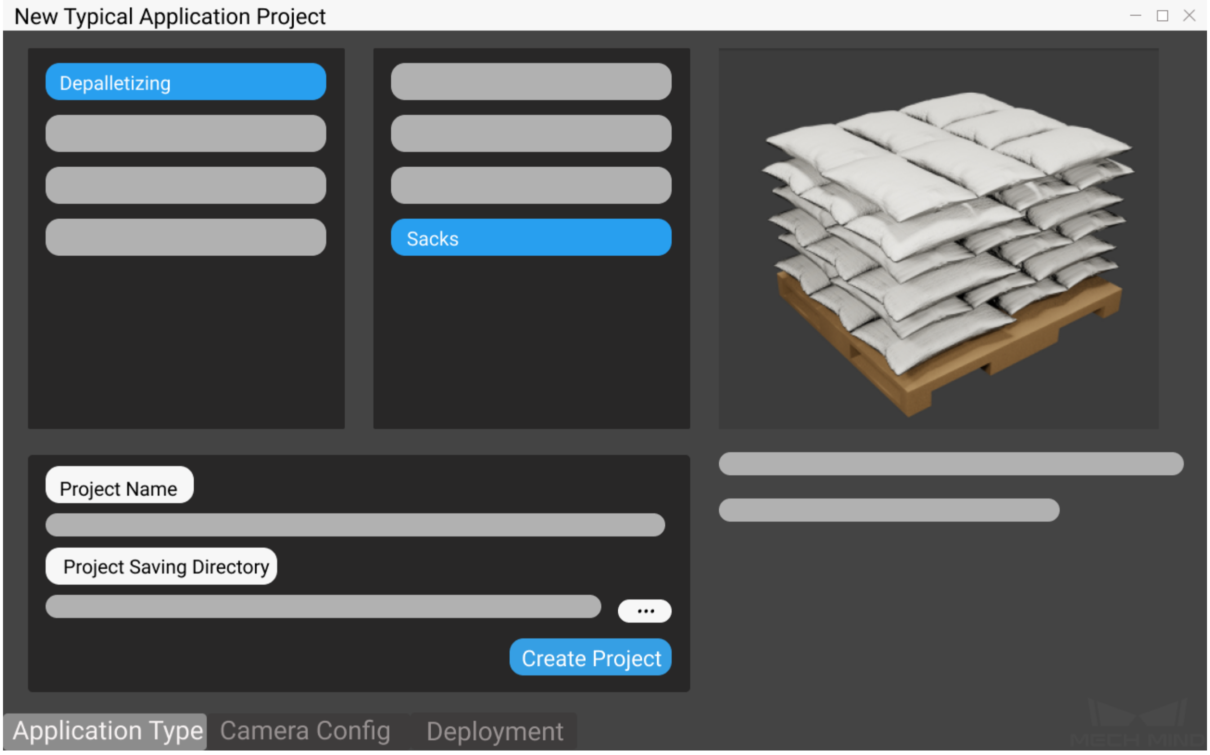
3.1.1 Preparation

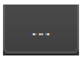
Before deploying the project, please confirm sack dimensions.

3.1.2 Instructions for Creating a New Typical Application Project

1. Create Project

Click on *Typical Applications* → *New Typical Application Project* in the Menu Bar or *New Typical Application Project* in the Toolbar to open the following window.



1. Select **Sacks**.
2. Name the project.
3. Click on  to select a folder to save the project (it is recommended to create an empty folder), then click *Create Project* .

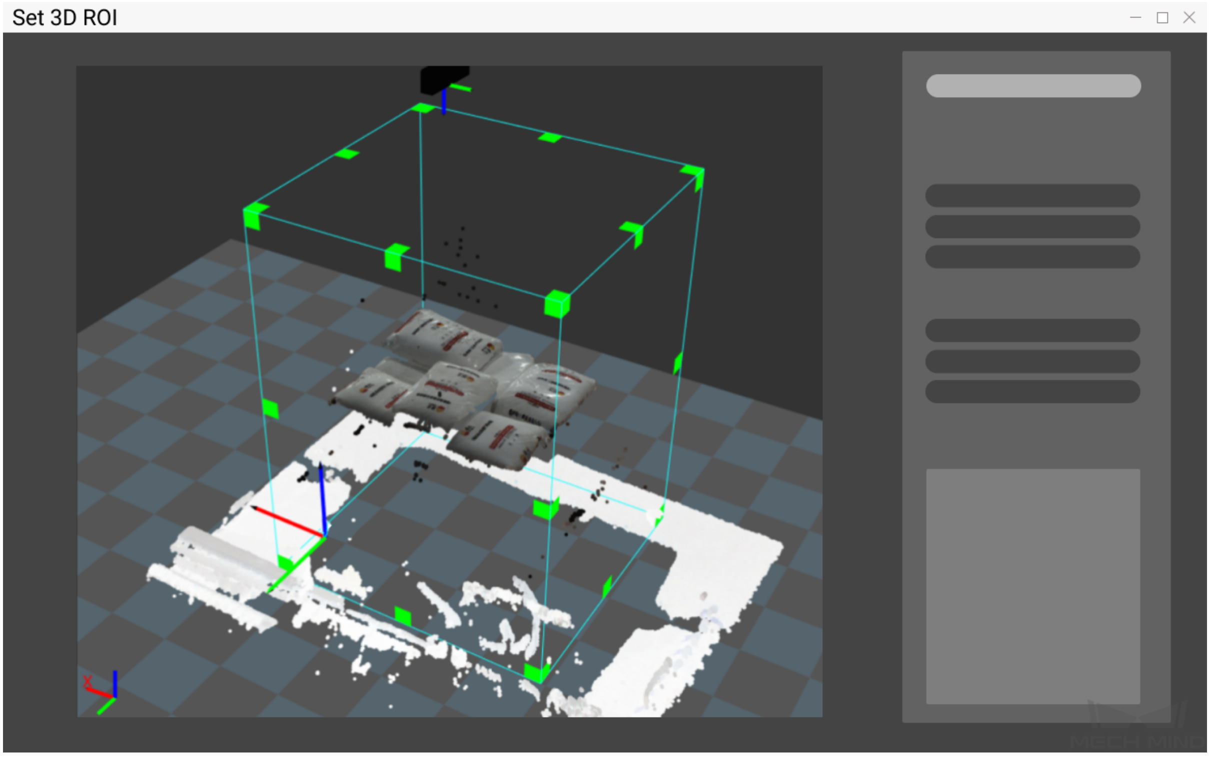
2. Camera Configuration and Calibration

It is necessary to ensure that the point cloud fully reflects the scene information. Please refer to camera_configuration to complete the related operations of camera configuration.

3. Related Parameter Settings and Object Recognition

Parameter Settings

1. Click on *3D ROI* to set the region of interest in 3D space and capture the height information. Hold down the Ctrl key and adjust with the left mouse button.

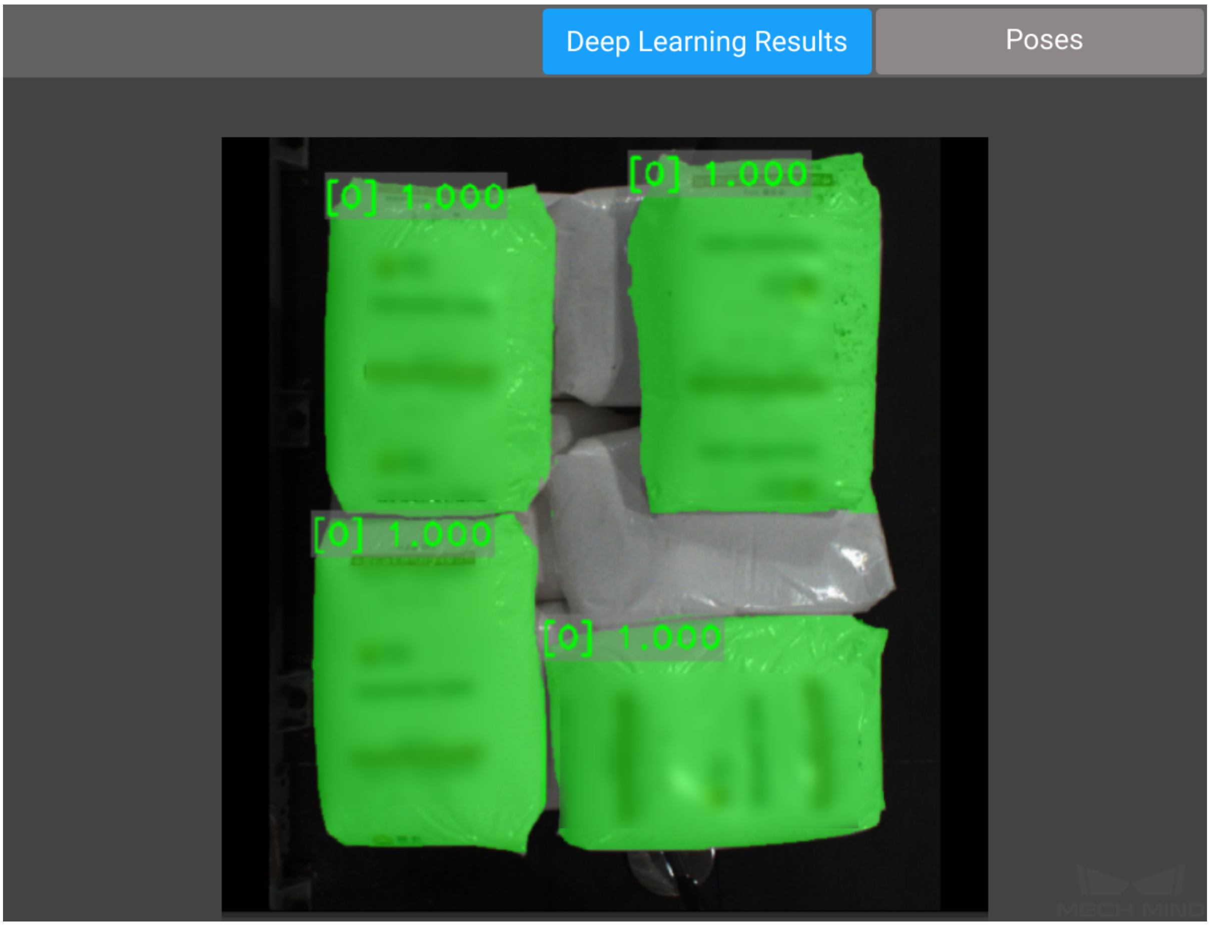


2. Enter the dimensions of sacks to be recognized in the current scene.

3. Please click on the right of the model path to select the corresponding deep learning model file in .pth format. After the selection, click on the right of the configuration file path to select the corresponding configuration file in .py format.

Object Recognition

1. Click on *Generate Pose* to recognize the object and get the object pose. Click *Deep Learning Results* and *Pose* to view the recognition results.



2. After viewing the pose results, click on *Edit Pose* to enter *Pose Editor* to sort and align the poses.

4. Finish Deployment

Please preview the result first. If there is no problem, you can finish deploying the project in typical application project. If you have other communication requirements, please make corresponding settings on the Mech-Center interface.

Click *Finish* in the deployment interface to open the operator interface.

3.1.3 Running and Debugging

Click on ▶ in the upper right corner of the application interface to run the project, and you can view the **Top-Layer Point Cloud**, **Deep Learning Results**, and **Pose** while running the project.

When the project running result is poor, you can refer to the following ideas to try to solve the problem.

- Confirm whether the sack dimensions input are correct.
- The top-layer point cloud is of poor quality.
  - The point cloud is truncated: reselect the ROI.
  - The point cloud has irregular missing parts: adjust the camera parameters.
- The deep learning result is poor.
  - Adjust the confidence threshold of deep learning.
    - Increasing the threshold will increase the number of results output.
    - Decreasing the threshold will decrease the number of results output.
  - Contact the support team to replace the deep learning model, or re-train the deep learning model.
- The pose results are poor.
  - Re-edit the poses.

Note:

- To change the deep learning model, see Box Palletizing/Depalletizing.
- You can switch to **standard editing mode** to view the detailed structure of the project. After reading and understanding *Guide to Steps*, you can try adding, deleting related steps and modifying the parameters of the steps in the project to meet the needs of more complex scenarios.

After successfully running and debugging the project, if you need to save the on-site data for future reference or you find that certain part of the project is not performing well and would like to optimize the Step or tune the parameters in an off-site situation, the tools *Data Storage* and *Data Playback* can be very useful.

Prerequisites for using the tool *Data Playback*:

1. A project file in which the project can run correctly without errors.
2. On-site source data which is gathered during the whole period while running the project, including 2D color images, depth maps, camera parameter file. Please refer to *Data Storage* for instructions on how to save the data.

3.2 Cartons (Mixed Case)

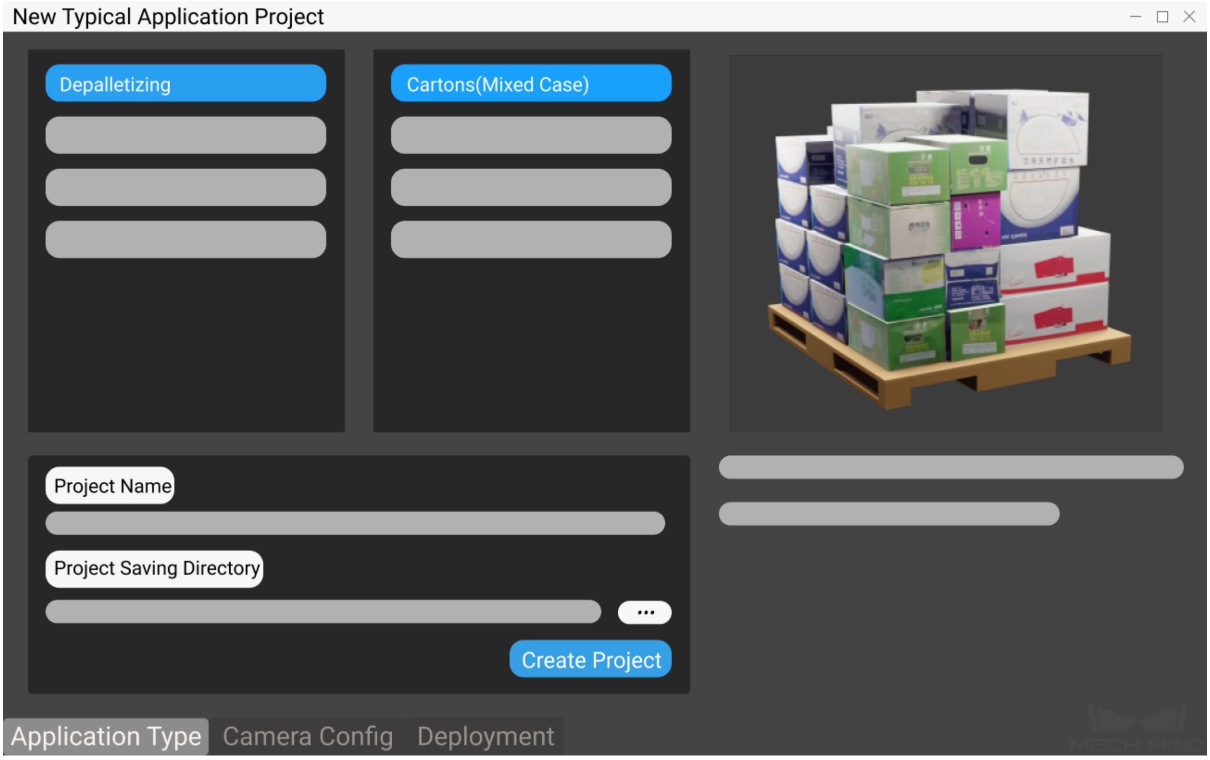
Mixed-case depalletizing means depalletizing where a single stack has cartons of a variety of dimensions.

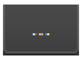
In carton depalletizing scenarios, instance segmentation is usually utilized to identify each carton in the image and recognize poses of the cartons. Mech-Mind Robotics provides a super model specifically used for carton depalletizing. It can be directly applied in Mech-Vision to correctly segment most of the cartons without deep learning training.

3.2.1 Instructions for Creating a New Typical Application Project

1. Create Project

Click on *Typical Applications* → *New Typical Application Project* in the Menu Bar or *New Typical Application Project* in the Toolbar to open the following window.



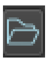
1. Select **Cartons (Mixed Case)**.
2. Name the project.
3. Click on  to select a folder to save the project (it is recommended to create an empty folder), then click *Create Project* .

2. Camera Configuration and Calibration

Please refer to camera_configuration to complete the related operations of camera configuration.

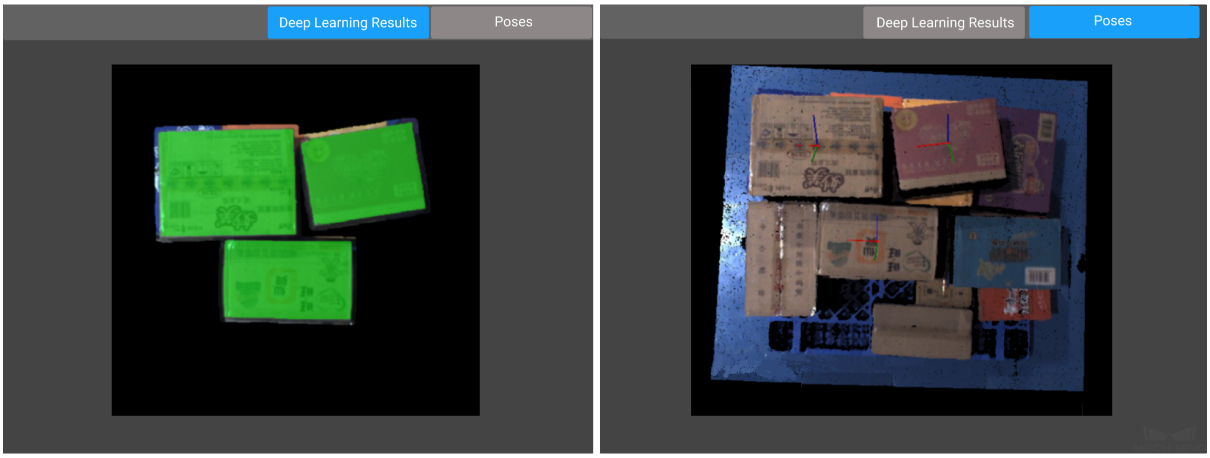
3. Related Parameter Settings and Object Recognition

Parameter Settings

1. Click on *2D ROI* to select the region of interest (the selection range should completely cover the target object). Click on *Refresh Image* to recapture the image, and click on *OK* to complete the setting.
2. Click on *3D ROI* to set the region of interest in 3D space and get height information. Hold down Ctrl key and adjust with the left mouse button.
3. Mech-Vision will automatically fill in the model path and configuration file path of the built-in model. To make a change, click on  on the right side of the model path to select the corresponding deep learning model file. After the selection is complete, click on  on the right side of the configuration file path to select the corresponding configuration file.

Object Recognition

1. Click on *Generate Pose* to recognize the objects and obtain the object poses. Click on *Deep Learning Results* or *Pose* to view the recognition results.



2. After viewing the pose results, click on *Edit Pose* and enter *Pose Editor* to sort and align the poses.

4. Finish Deployment

Please preview the result first. If there is no problem, you can finish deploying the typical application project. If you have other communication requirements, please make corresponding settings on the Mech-Center interface.

Click *Finish* in the deployment interface to open the operator interface.

3.2.2 Running and Debugging

Click on 

in the upper right corner of the application interface to run the project, and you can view the **Top-Layer Point Cloud**, **Deep Learning Results**, and **Pose** while running the project.

When the project running result is poor, you can refer to the following ideas to try to solve the problem.

- The top-layer point cloud is of poor quality.
  - The point cloud is truncated: reselect the ROI.
  - The point cloud has irregular missing parts: adjust the camera parameters.
- The deep learning result is poor.
  - Adjust the confidence threshold of deep learning.
    - Increasing the threshold will increase the number of results output.

    * Decreasing the threshold will decrease the number of results output.

  – Contact the support team to replace the deep learning model, or re-train the deep learning model.

- The pose results are poor.

  – Re-edit the poses.

Note:

- To change the deep learning model, please see Box Palletizing/Depalletizing.
- You can switch to **standard editing mode** to view the detailed structure of the project. After reading and understanding *Guide to Steps*, you can try to add, delete related steps and modify the parameters of the steps in the project to meet the needs of more complex scenarios.

After successfully running and debugging the project, if you need to save the on-site data for future reference or you find that certain part of the project is not performing well and would like to optimize the Step or tune the parameters in an off-site situation, the tools *Data Storage* and *Data Playback* can be very useful.

Prerequisites for using the tool *Data Playback*:

1. A project file in which the project can run correctly without errors.
2. On-site source data which is gathered during the whole period while running the project, including 2D color images, depth maps, camera parameter file. Please refer to *Data Storage* for instructions on how to save the data.

3.3 Cartons (Single Case, Input Dimensions)

Single-case depalletizing means depalletizing where a single stack has cartons of the same dimensions.

In carton depalletizing scenarios, instance segmentation is usually utilized to identify each carton in the image and recognize poses of the cartons. Mech-Mind Robotics provides a super model specifically used for carton depalletizing. It can be directly applied in Mech-Vision to correctly segment most of the cartons without deep learning training.

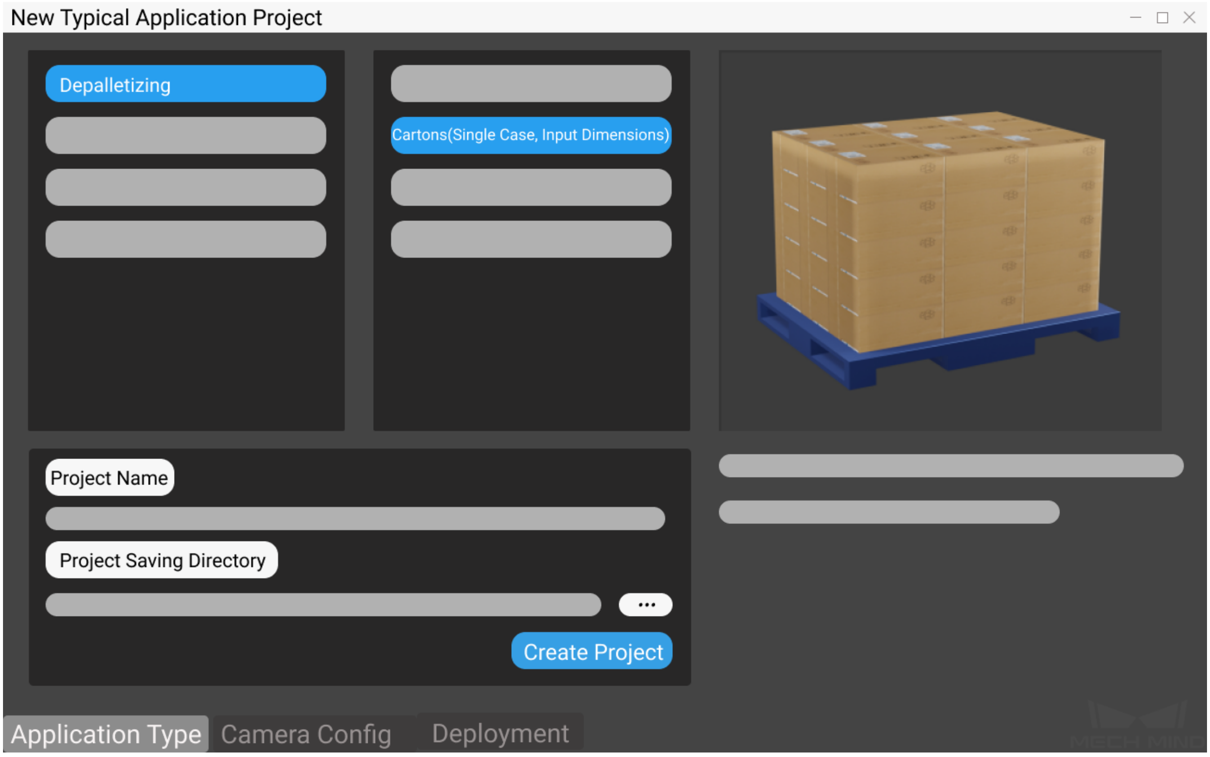
3.3.1 Preparation

Before deploying the project, please confirm the carton dimensions.

3.3.2 Instructions for Creating a New Typical Application Project

1. Create Project

Click on *Typical Applications* → *New Typical Application Project* in the Menu Bar or *New Typical Application Project* in the Toolbar to open the following window.



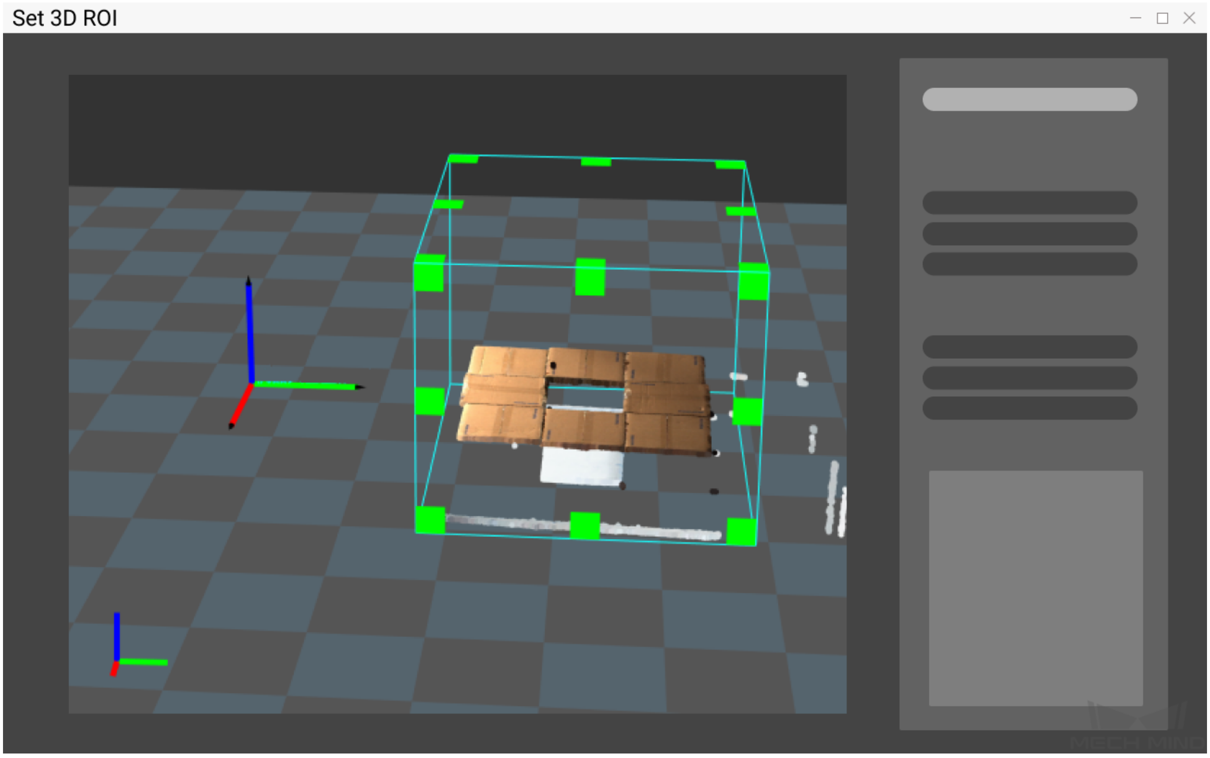
1. Select **Cartons (Single Case, Input Dimensions)**.
2. Name the project.
3. Click on ... to select a folder to save the project (it is recommended to create an empty folder), then click *Create Project* .

2. Camera Configuration and Calibration

Please refer to camera_configuration to complete the related operations of camera configuration.

3. Related Parameter Settings and Object Recognition

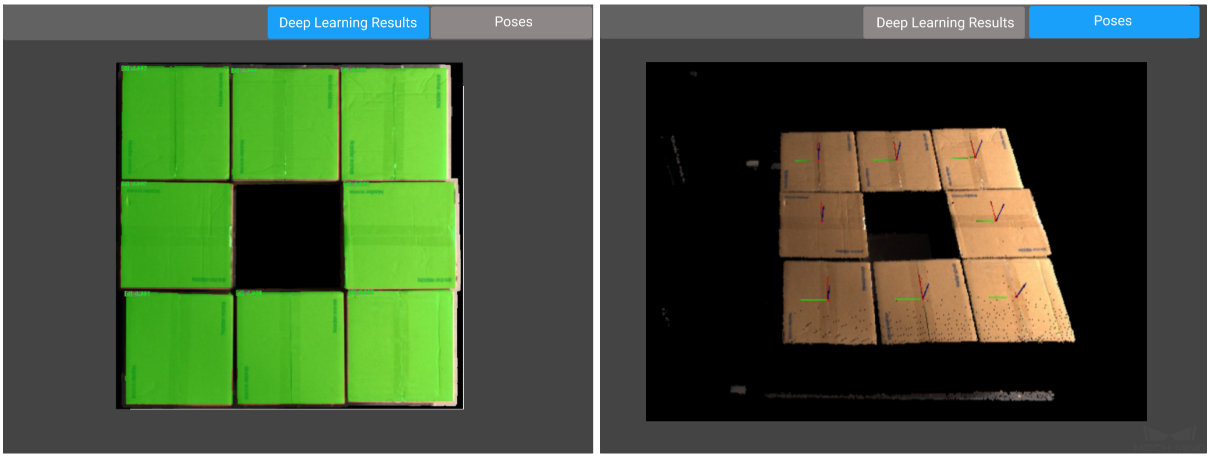
1. Click on *2D ROI* to select the region of interest (the selection range should completely cover the target object). Click on *Refresh Image* to recapture the image, and click on *OK* to complete the setting.
2. Click on *3D ROI* to set the region of interest in 3D space. Hold down Ctrl key and adjust with the left mouse button.



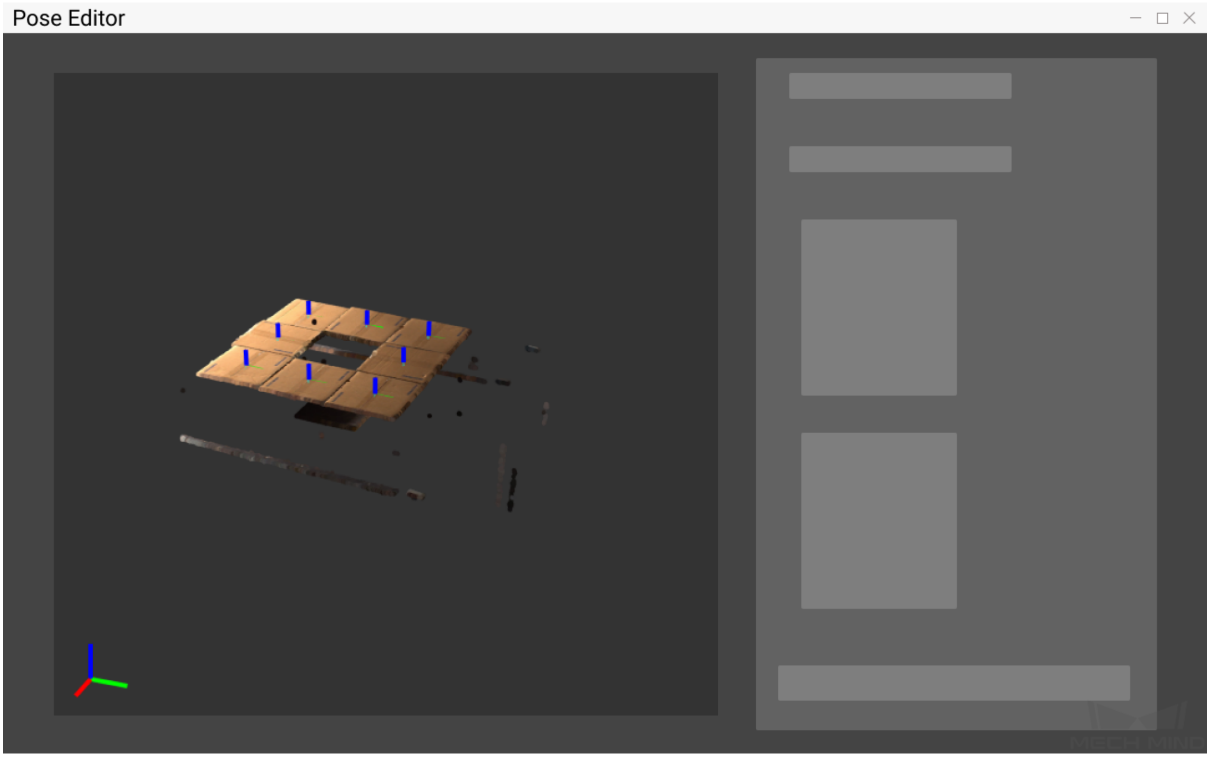
3. Enter the dimensions of the cartons to be recognized in the current scene.

Object Recognition

1. Click on *Generate Pose* to recognize the objects and get the object poses. Click on *Deep Learning Results* or *Pose* to view the recognition results.



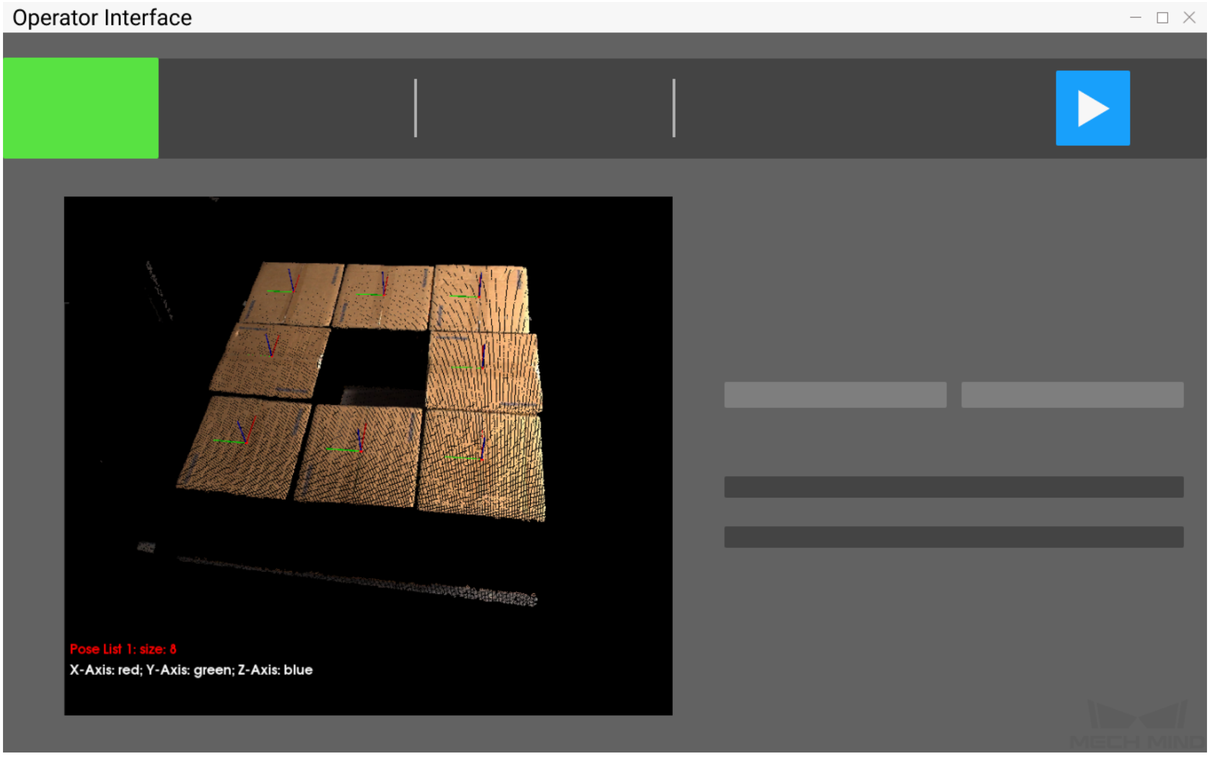
2. After viewing the pose results, click on *Edit Pose* and enter *Pose Editor* to sort and align the poses.



4. Finish Deployment

Please preview the result first. If there is no problem, you can finish deploying the typical application project. If you have other communication requirements, please make corresponding settings in the Mech-Center interface.

Click *Finish* in the deployment interface to open the operator interface.



3.3.3 Running and Debugging

Click on  in the upper right corner of the application interface to run the project, and you can view the **Top-Layer Point Cloud**, **Deep Learning Results**, and **Pose** while running the project.

When the project running result is poor, you can refer to the following ideas to try to solve the problem.

- Confirm whether the box dimensions input are correct.
- The top-layer point cloud is of poor quality.
  - The point cloud is truncated: reselect the ROI.
  - The point cloud has irregular missing parts: adjust the camera parameters.
- The deep learning result is poor.
  - Adjust the confidence threshold of deep learning.

- ∗ Increasing the threshold will increase the number of results output.
- ∗ Decreasing the threshold will decrease the number of results output.

– Contact the support team to replace the deep learning model, or re-train the deep learning model.

- The pose results are poor.
  – Re-edit the poses.

Note:

- To change the deep learning model, please see Box Palletizing/Depalletizing.
- You can switch to **standard editing mode** to view the detailed structure of the project. After reading and understanding *Guide to Steps*, you can try to add, delete related steps and modify the parameters of the steps in the project to meet the needs of more complex scenarios.

After successfully running and debugging the project, if you need to save the on-site data for future reference or you find that certain part of the project is not performing well and would like to optimize the Step or tune the parameters in an off-site situation, the tools *Data Storage* and *Data Playback* can be very useful.

Prerequisites for using the tool *Data Playback*:

1. A project file in which the project can run correctly without errors.
2. On-site source data which is gathered during the whole period while running the project, including 2D color images, depth maps, camera parameter file. Please refer to *Data Storage* for instructions on how to save the data.

3.4 Cartons (Single Case, Not Input Dimensions)

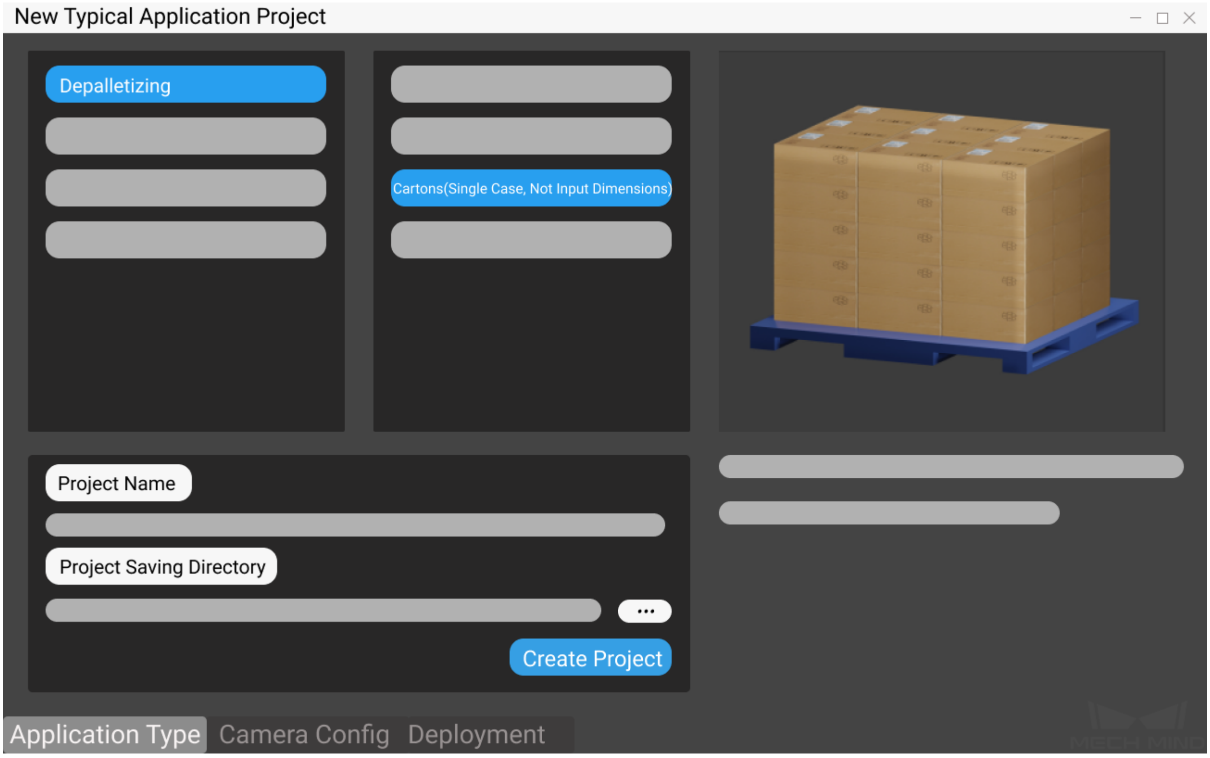
Single-case depalletizing means depalletizing where a single stack has cartons of the same dimensions.

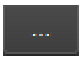
In carton depalletizing scenarios, instance segmentation is usually utilized to identify each carton in the image and recognize poses of the cartons. Mech-Mind Robotics provides a super model specifically used for carton depalletizing. It can be directly applied in Mech-Vision to correctly segment most of the cartons without deep learning training. In addition, since the deep learning model will measure the carton dimensions automatically, you do not need to confirm the carton dimensions in advance.

3.4.1 Instructions for Creating a New Typical Application Project

1. Create Project

Click on *Typical Applications* → *New Typical Application Project* in the Menu Bar or *New Typical Application Project* in the Toolbar to open the following window.



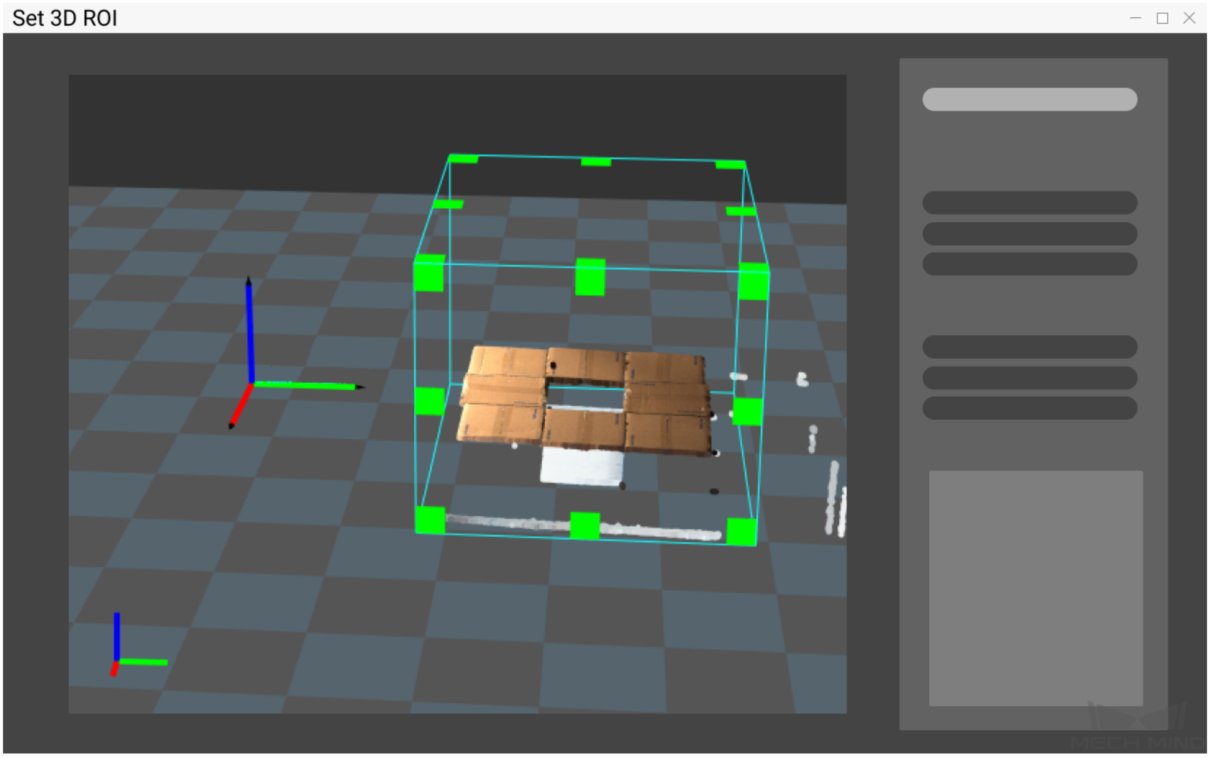
1. Select **Cartons (Single Case, Not Input Dimensions)**.
2. Name the project.
3. Click on  to select a folder to save the project (it is recommended to create an empty folder), then click *Create Project* .

2. Camera Configuration and Calibration

Please refer to camera_configuration to complete the related operations of camera configuration.

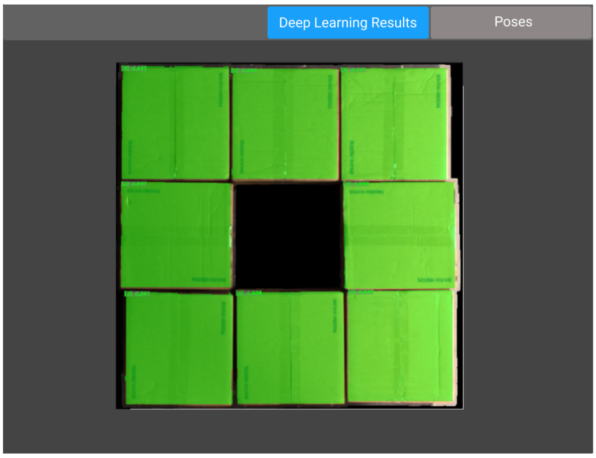
3. Related Parameter Settings and Object Recognition

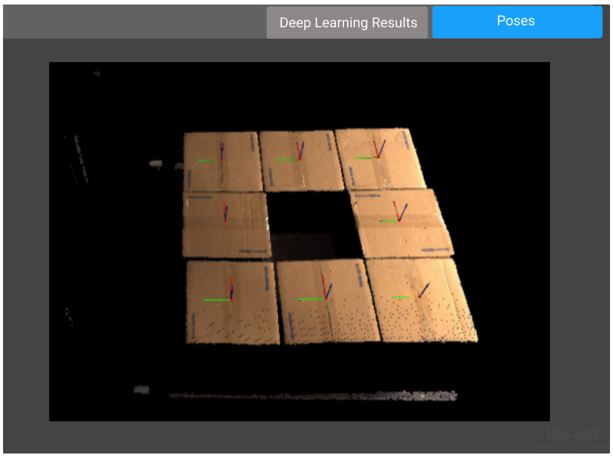
1. Click on *2D ROI* to select the region of interest (the selection range should completely cover the target object). Click on *Refresh Image* to recapture the image, and click on *OK* to complete the setting.
2. Click on *3D ROI* to set the region of interest in 3D space. Hold down Ctrl key and adjust with the left mouse button.



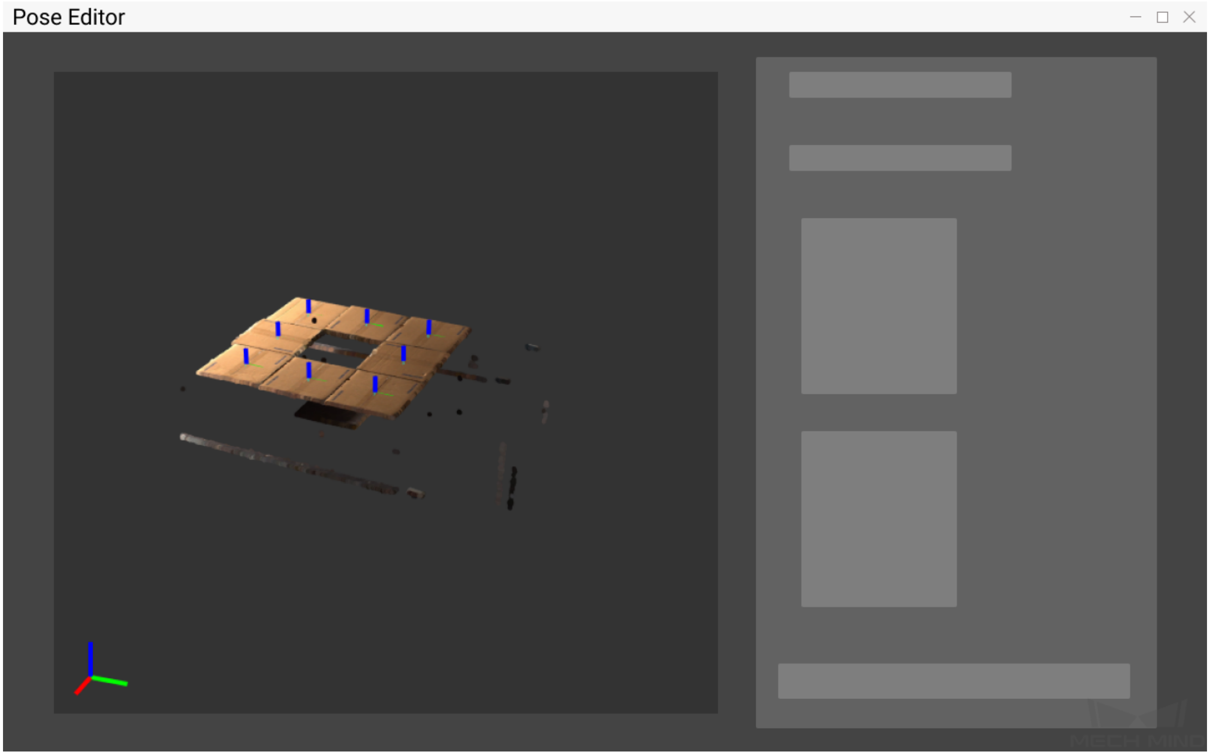
Object Recognition

1. Click on *Generate Pose* to recognize the objects and get the object poses. Click on *Deep Learning Results* or *Pose* to view the recognition results.





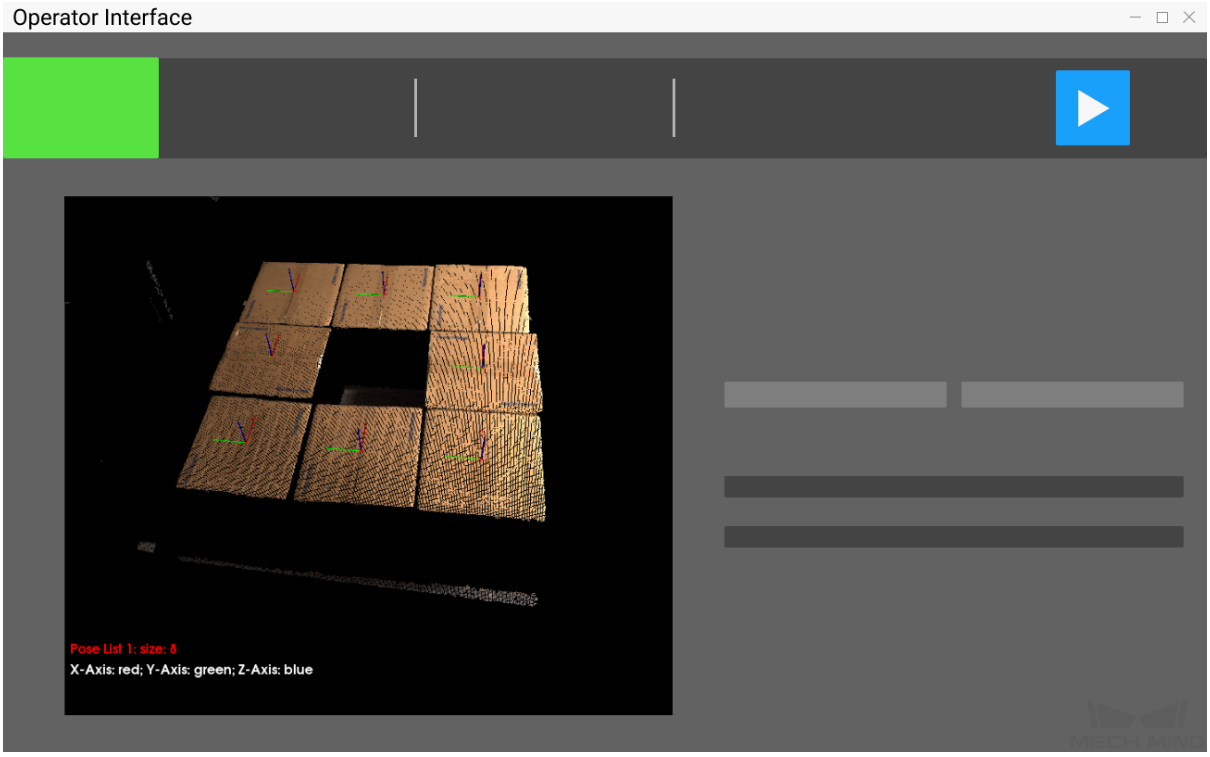
2. After viewing the pose results, click on *Edit Pose* and enter *Pose Editor* to sort and align the poses.



4. Finish Deployment

Please preview the result first. If there is no problem, you can finish deploying the typical application project. If you have other communication requirements, please make corresponding settings in the Mech-Center interface.

Click *Finish* in the deployment interface to open the operator interface.



3.4.2 Running and Debugging

Click on in the upper right corner of the application interface to run the project, and you can view the **Top-Layer Point Cloud**, **Deep Learning Results**, and **Pose** while running the project.

When the project running result is poor, you can refer to the following ideas to try to solve the problem.

- Confirm whether the carton dimensions input are correct.
- The top-layer point cloud is of poor quality.
  - The point cloud is truncated: reselect the ROI.
  - The point cloud has irregular missing parts: adjust the camera parameters.
- The deep learning result is poor.
  - Adjust the confidence threshold of deep learning.

    * Increasing the threshold will increase the number of results output.
    * Decreasing the threshold will decrease the number of results output.
  - Contact the support team to replace the deep learning model, or re-train the deep learning model.

- The pose results are poor.
  - Re-edit the poses.

Note:

- To change the deep learning model, please see Box Palletizing/Depalletizing.
- You can switch to **standard editing mode** to view the detailed structure of the project. After reading and understanding *Guide to Steps*, you can try to add, delete related steps and modify the parameters of the steps in the project to meet the needs of more complex scenarios.

After successfully running and debugging the project, if you need to save the on-site data for future reference or you find that certain part of the project is not performing well and would like to optimize the Step or tune the parameters in an off-site situation, the tools *Data Storage* and *Data Playback* can be very useful.

Prerequisites for using the tool *Data Playback*:

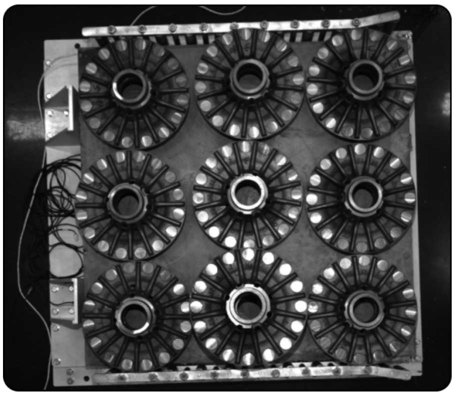
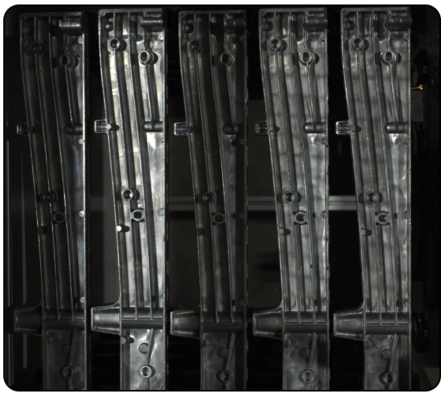
1. A project file in which the project can run correctly without errors.
2. On-site source data which is gathered during the whole period while running the project, including 2D color images, depth maps, camera parameter file. Please refer to *Data Storage* for instructions on how to save the data.

Machine Tending

Applicable to pick and move complex workpieces, structural parts, irregular parts, and other objects in the automotive, steel, machinery, and other industries. Able to handle a variety of complex situations, and able to deal with metals with varying reflectivity to a certain extent.

According to the features of the objects to pick, machine tending in Mech-Vision can be categorized into the following three scenarios:

3.5 Large Non-Planar Workpieces Loading

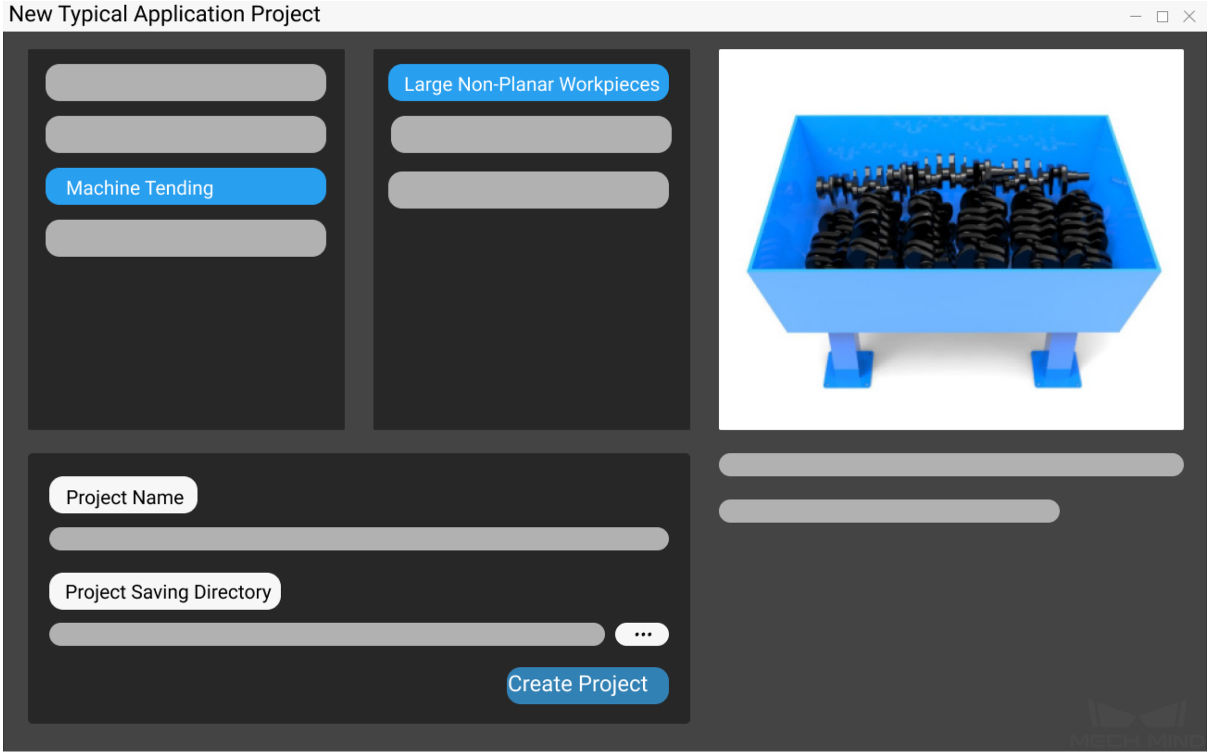
 

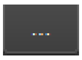
The image above shows the result of large non-planar workpiece loading. For this scenario, the number of workpieces to be processed is relatively small, while the sizes of workpieces are relatively large and the features of their curved surfaces are relatively obvious. In this project, 3D matching algorithms are used to recognize the large non-planar workpieces.

3.5.1 Create a New Typical Application Project

Create Project

Click on *Typical Applications* → *New Typical Application Project* in the Menu Bar or *New Typical Application Project* in the Toolbar to open the following window.



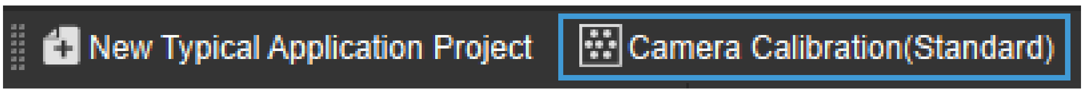
1. Select **Large Non-Planar Workpieces**.
2. Name the project.
3. Click on  to select a folder to save the project (it is recommended to create an empty folder), then click *Create Project* .

3.5.2 Preparation

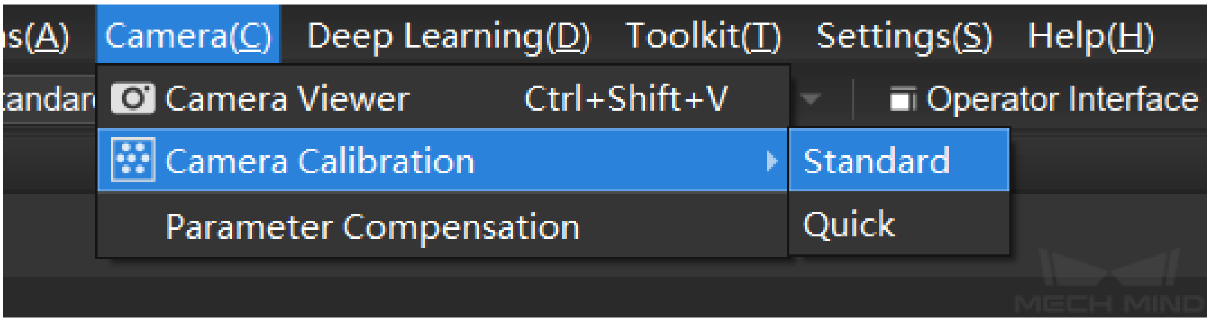
1. Calibrate the Camera

Calibrating the camera is to obtain the parameter file of the camera, which is used to determine the spatial relationship between the robot and camera.

Click on *Camera Calibration(Standard)* in the Toolbar to calibrate the camera.



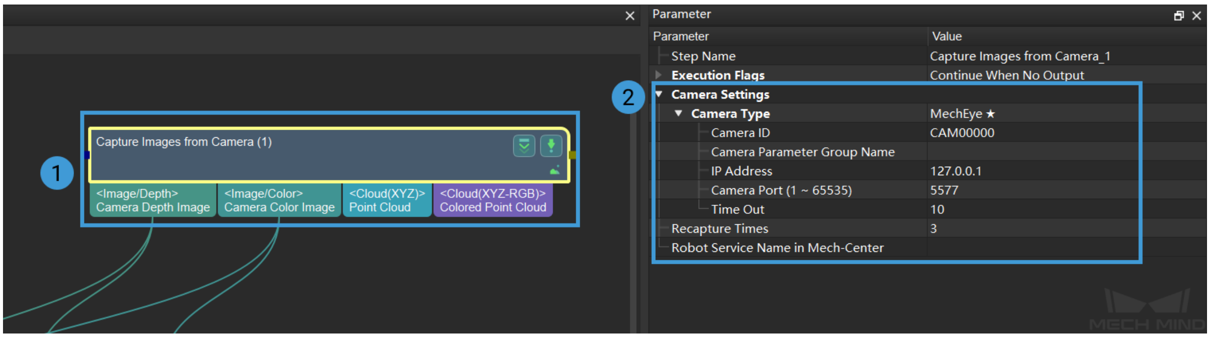
Alternatively, you can click on *Camera* → *Camera Calibration* → *Standard* in the Menu Bar to open the same window.



For more information about camera calibration, please refer to *Start Calibration - Standard Mode*.

2. Configure the Camera

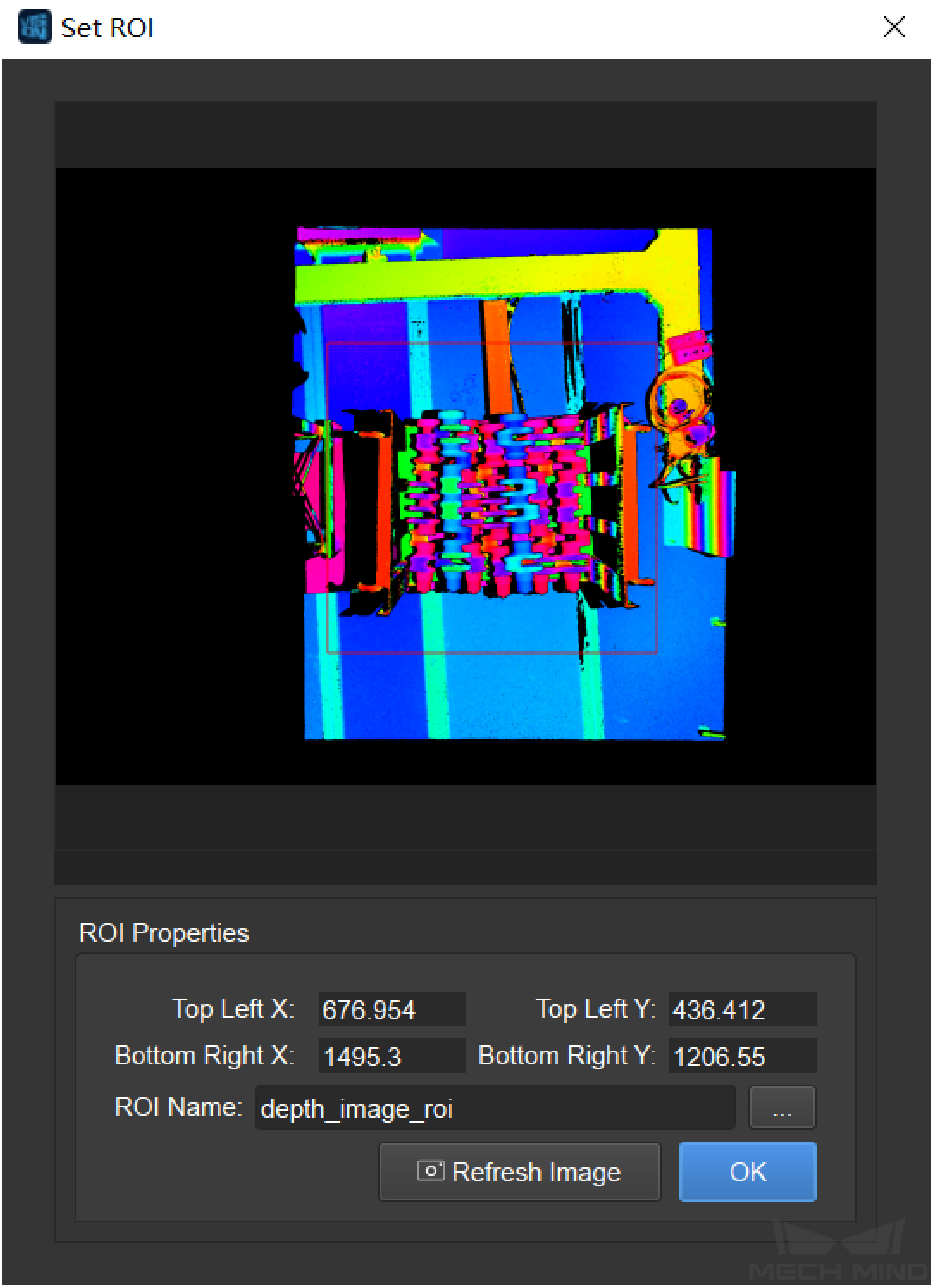
Before using the camera to capture images, you need to set the type, parameter group name, IP address and other parameters of the camera. For detailed instructions, please see *Capture Images from Camera*.



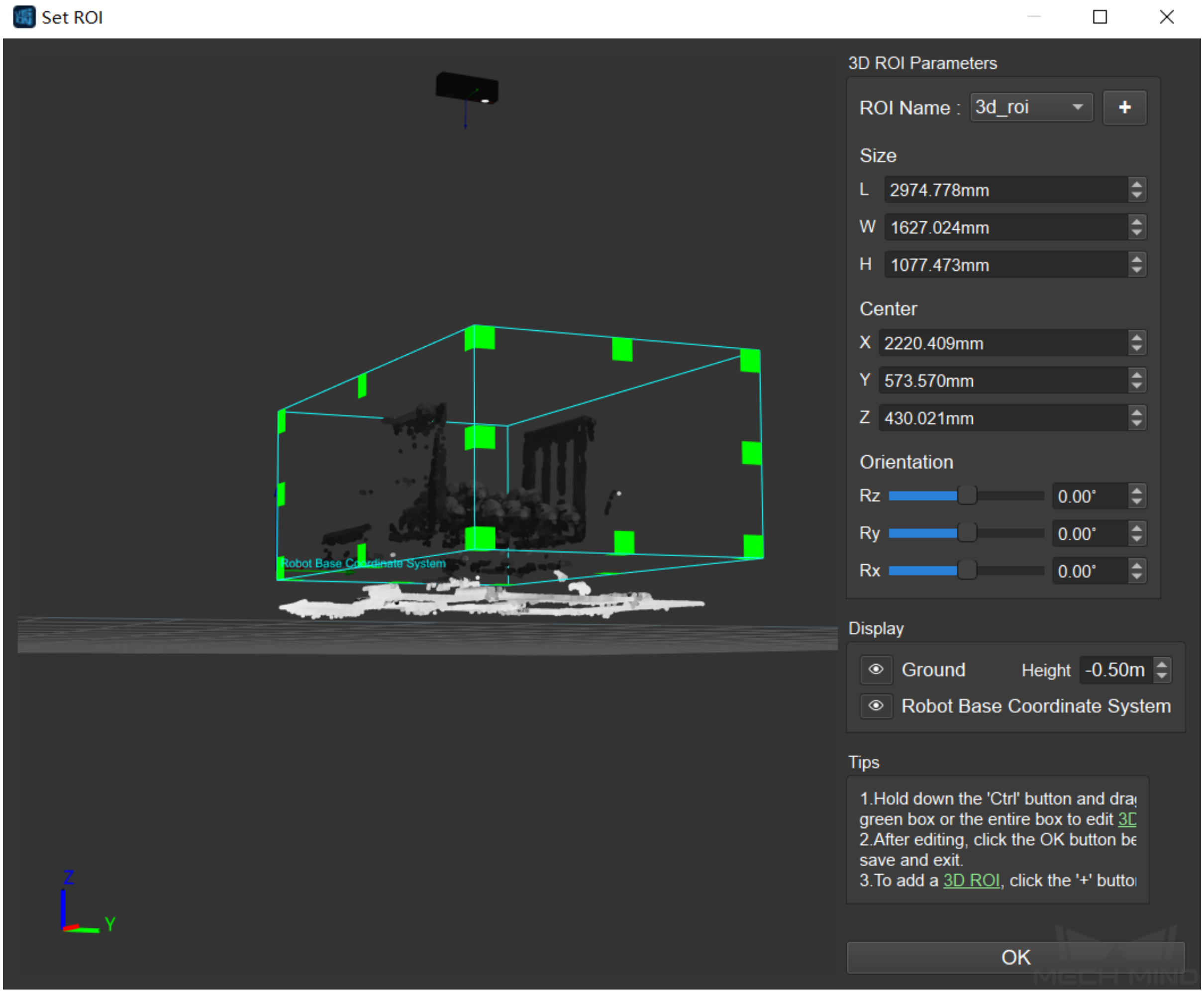
3. Set 2D and 3D ROI

Double click on the Procedure **Point Cloud Pre-Processing** to display the detailed structure, and set 2D ROI and 3D ROI in the Step *From Depth Map to Point Cloud* and *Extract 3D Points in 3D ROI* respectively.

Setting a 2D ROI can avoid generating unnecessary point cloud and therefore increase the pre-processing speed. Please refer to *Instructions for Setting 2D ROI* for detailed instructions on setting 2D ROI.


Set ROI
ROI Properties
Top Left X: 676.954
Top Left Y: 436.412
Bottom Right X: 1495.3
Bottom Right Y: 1206.55
ROI Name: depth_image_roi
...
Refresh Image
OK

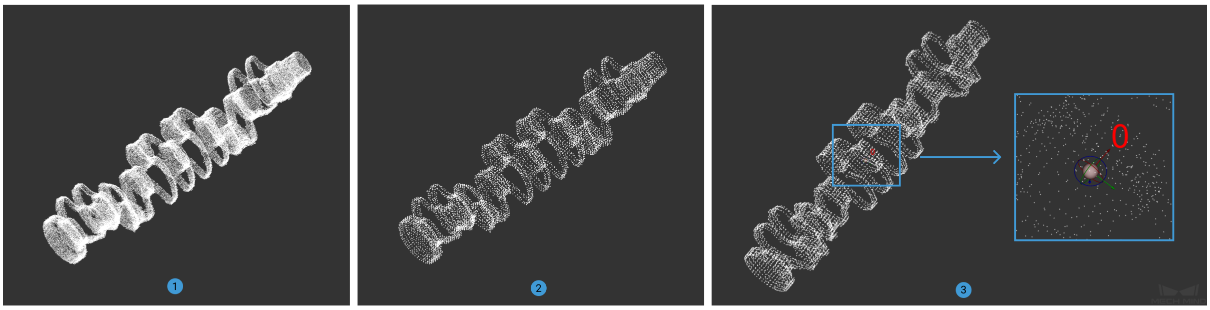
Setting a 3D ROI can extract the point cloud of the target objects and filtered the unwanted points in the backgroung. Please refer to *Instructions for Setting 3D ROI* for detailed instructions on setting 3D ROI.



4. Generate a Model Point Cloud

Please use the tool *Matching Model and Pick Point Editor* to generate a model point cloud of the target object, so that Mech-Vision can compare the point cloud of the target object to the model point cloud and then generate an actual picking pose.

The model point cloud and pick point generated using the Matching Model and Pick Point Editor are shown below.



is the generated model point cloud of the large non-planar workpiece; is the downsampling result; is the pick point.

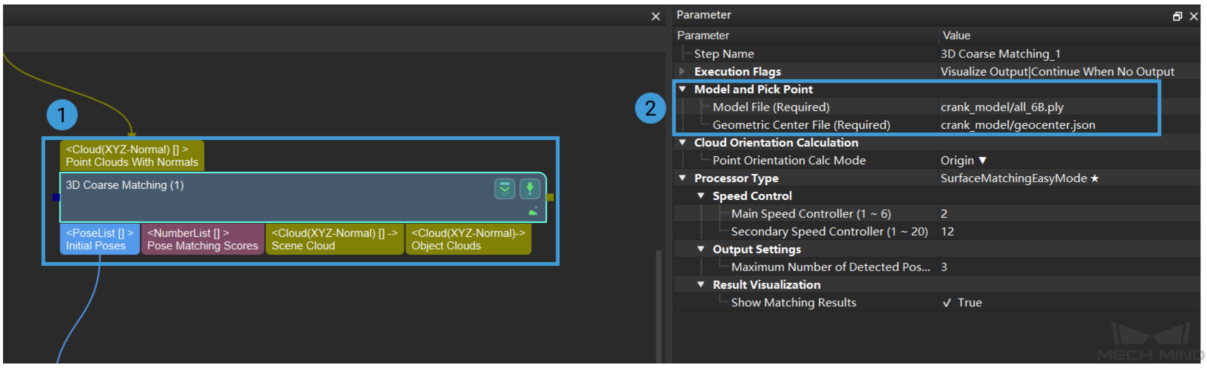
The generated model file will be saved in the project folder.

3.5.3 Project Deployment

During project deployment phase, you will need to set relevant parameters of Steps, and add the configuration files obtained in the preparation phase to corresponding Steps before actually running the project.

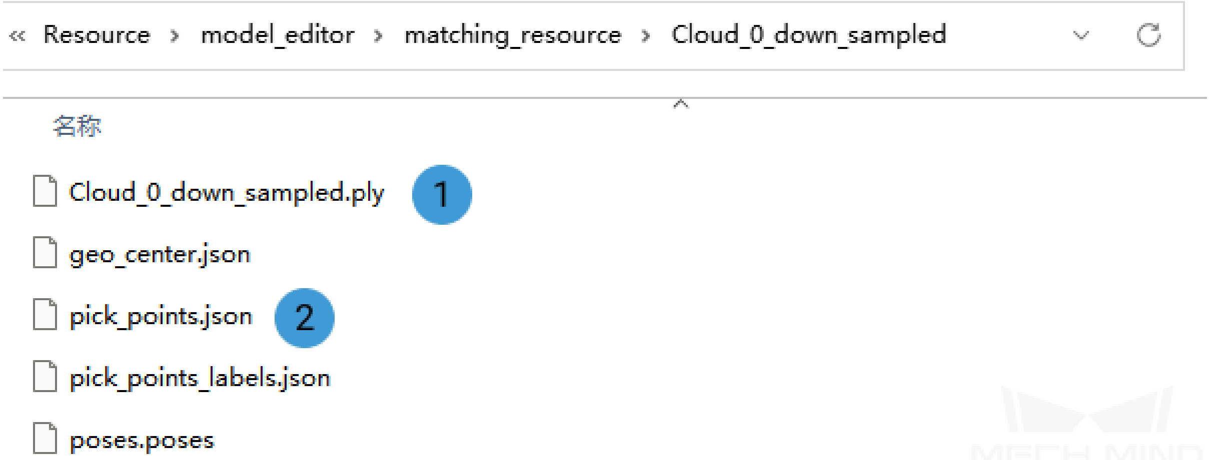
1. Set the Model File and Geometric Center File

- Double-click on the Procedure **3D Matching** to display the detailed structure.
- Select the Step **3D Coarse Matching** and click on  to set the model file and geometric center file in the Parameter, as shown below.



- The way to set model file and geometric center file in **3D Fine Matching** is the same as it is in **3D Coarse Matching**.

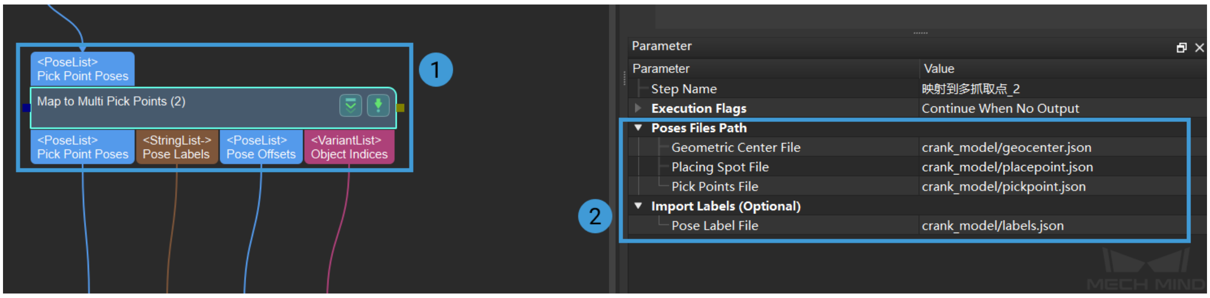
Click to see the example of each file



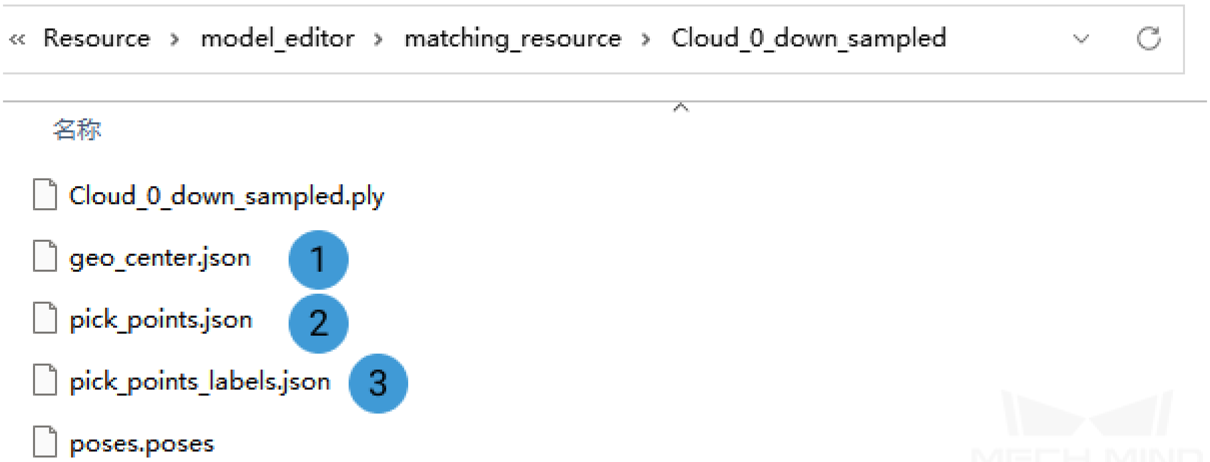
is a point cloud model file, and is a geometric center file.

2. Set Poses Files

- Select the Step **Map to Multi Pick Points** and click on to set the poses files and important lables in Parameter, as shown below.



Click to see the example of each file



is a geometric center file, is a pose label file, and is a placing spot file.

3.5.4 Running and Debugging

After completing the project deployment, click on 

to run the project.

- Please see *Run the Project and Debug* for how to run and debug the project.
- Please see *Small Non-Planar Workpiece Loading* to learn about the algorithms and parameter adjustment of the project.

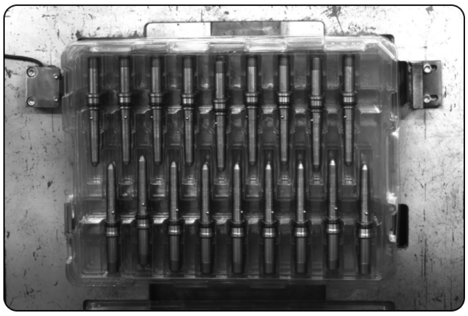
Tip: If you need to save the images or parameters when debugging the project or before training a deep learning model, you can use the tool *Data Storage*.

After successfully running and debugging the project, if you need to save the on-site data for future reference or you find that certain part of the project is not performing well and would like to optimize the Step or tune the parameters in an off-site situation, the tools *Data Storage* and *Data Playback* can be very useful.

Prerequisites for using the tool *Data Playback*:

1. A project file in which the project can run correctly without errors.
2. On-site source data which is gathered during the whole period while running the project, including 2D color images, depth maps, camera parameter file. Please refer to *Data Storage* for instructions on how to save the data.

3.6 Small Non-Planar Workpiece Loading

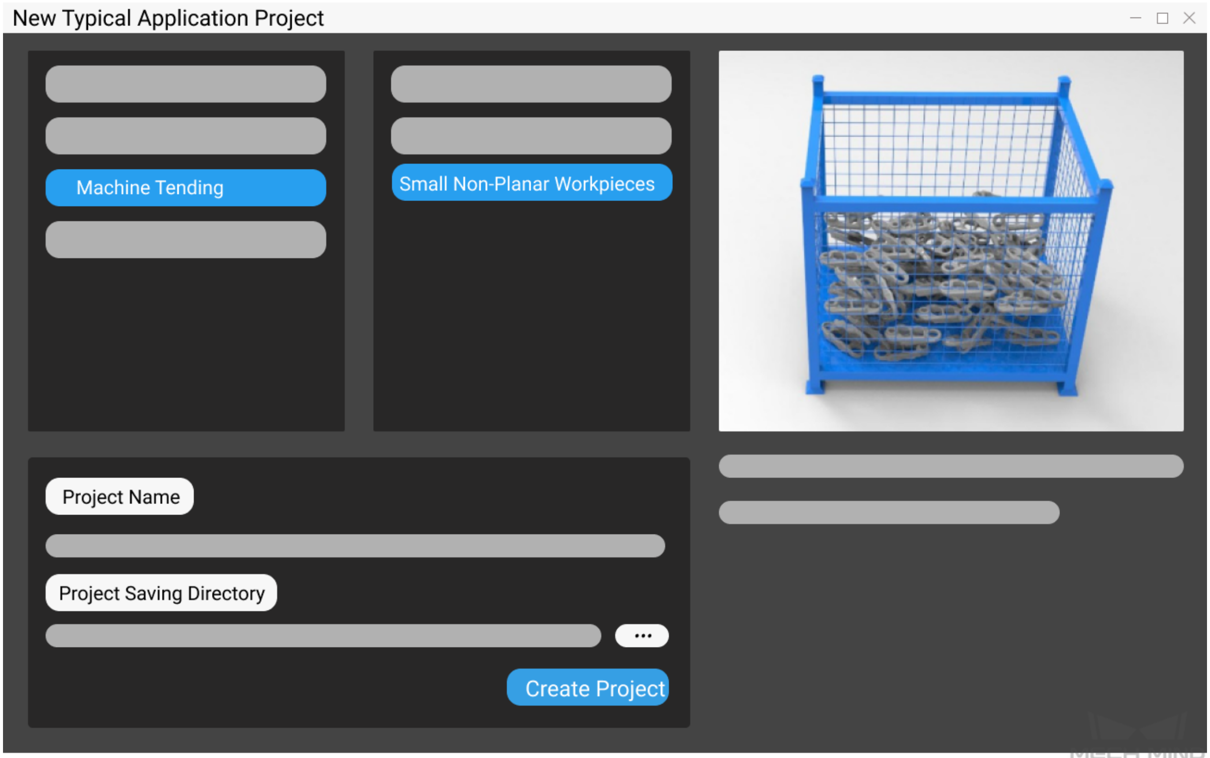



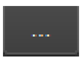
The image above shows the result of small non-planar workpiece loading. For this scenario, the number of the workpieces to be processed is large and their size is small. In this project, deep learning algorithms and 3D matching algorithms are used to recognize the small non-planar workpieces.

3.6.1 Create a New Typical Application Project

Create a Project

Click on *Typical Applications* → *New Typical Application Project* in the Menu Bar or *New Typical Application Project* in the Toolbar to open the following window.



1. Select **Small Non-Planar Workpieces**.
2. Name the project.
3. Click on  to select a folder to save the project (it is recommended to create an empty folder), then click *Create Project* .

3.6.2 Preparation

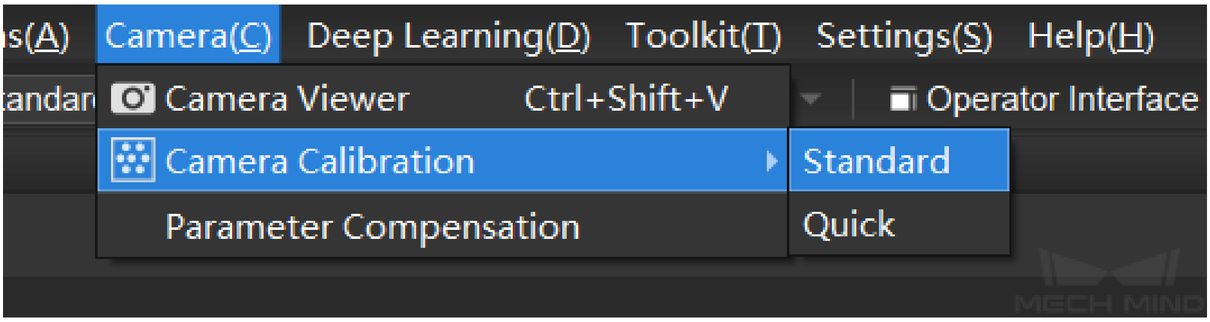
Before deploying the project, please complete the following preparation:

1. Calibrate the Camera

Calibrating the camera is to obtain the parameter file of the camera, which is used to determine the spatial relationship between the robot and camera.

Click on *Camera Calibration(Standard)* in the Toolbar to calibrate the camera.

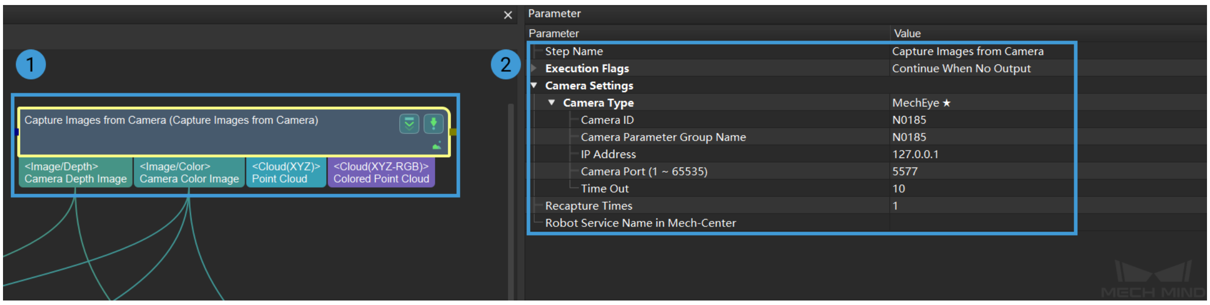
Alternatively, you can click on *Camera* → *Camera Calibration* → *Standard* in the Menu Bar to open the same window.



For more information about camera calibration, please refer to *Start Calibration - Standard Mode*.

2. Configure the Camera

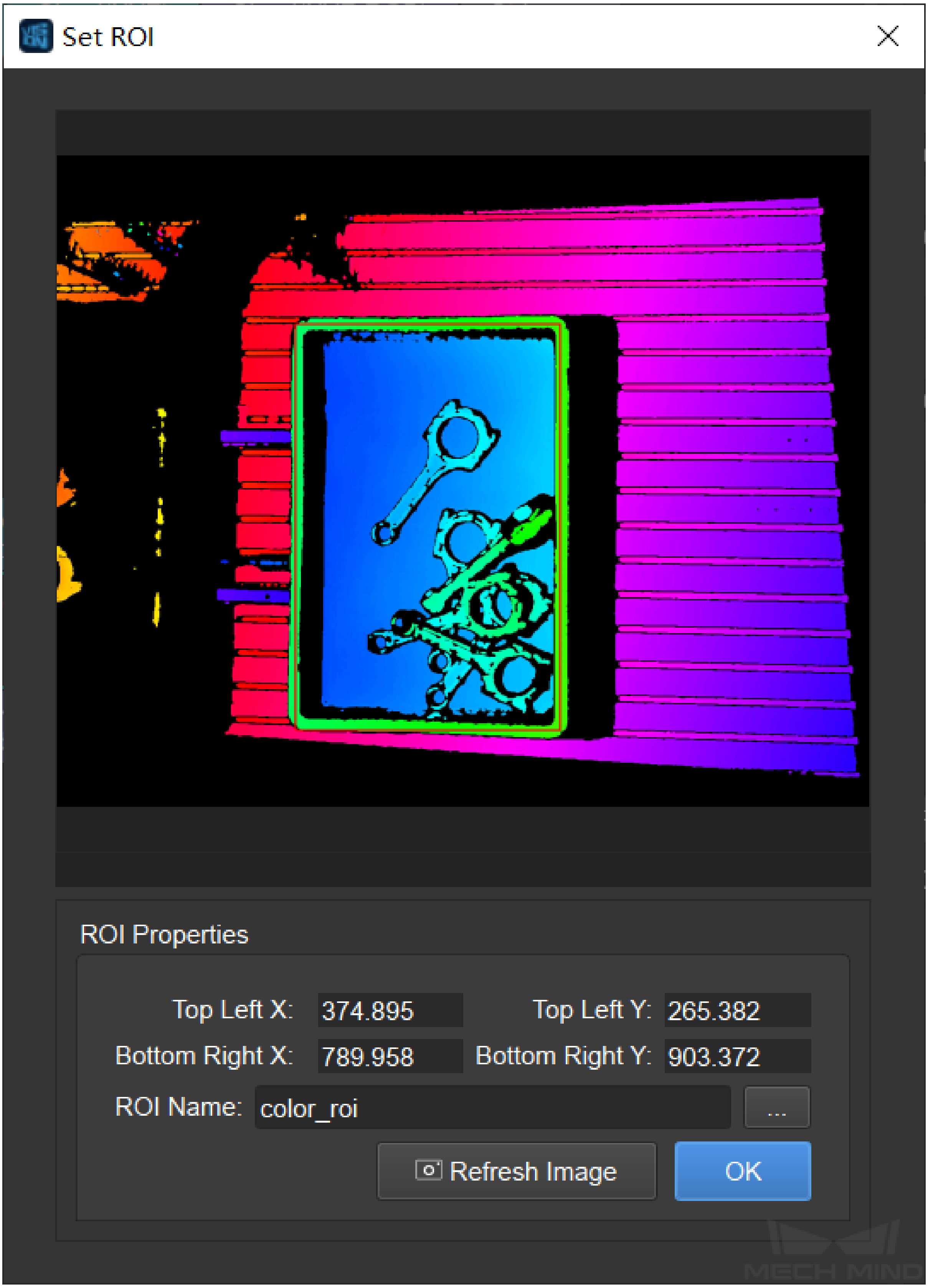
Before using the camera to capture images, you need to set the type, parameter group name, IP address and other parameters of the camera. For detailed instructions, please see *Capture Images from Camera*.



3. Set 2D and 3D ROI

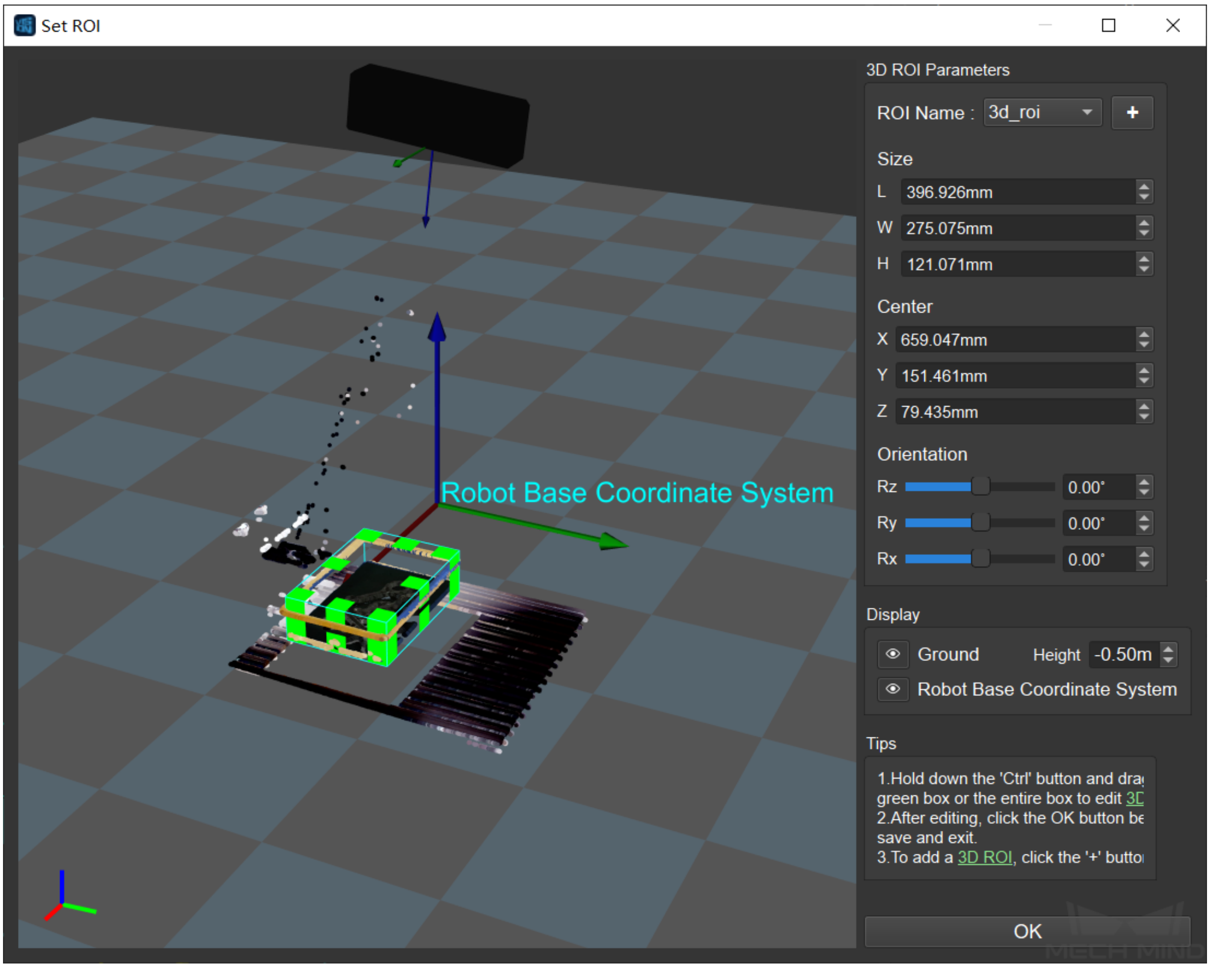
Double click on the Procedure **Point Cloud Pre-Processing** to display the detailed structure, and set 2D ROI and 3D ROI in the Step *From Depth Map to Point Cloud* and *Extract 3D Points in 3D ROI* respectively.

Setting a 2D ROI can increase the deep learning pre-processing speed. Please refer to *Instructions for Setting 2D ROI* for detailed instructions on setting 2D ROI.


Set ROI
ROI Properties
Top Left X: 374.895
Top Left Y: 265.382
Bottom Right X: 789.958
Bottom Right Y: 903.372
ROI Name: color_roi
...
Refresh Image
OK

Attention: The way to set a 2D ROI in Step *From Depth Map to Point Cloud* of the Procedure **Point Cloud Pre-Processing** is the same as above.

Setting a 3D ROI can extract the point cloud of the target objects and filtered the unwanted points in the backgroung. Please refer to *Instructions for Setting 3D ROI* for detailed instructions on setting 3D ROI.



Tip: Usually, the small non-planar workpieces are piled randomly in a bin. Setting a 3D ROI is not able to facilitate the algorithms to distinguish the bin from workpieces, and the points in the background cannot be filtered effectively. In these circumstances, *Set Static Background for Project* may improve the performance of filtering unwanted points.

1. **Instance Segmentation**

> **Note:** If you already have a super model, you can skip this step and start configuring the deep learning model file.

Instance segmentation is used for detecting and locating each distinct target object in an image, as shown below.

Please see Instance Segmentation for detailed information about training a deep learning model.

5. Generate a Model Point Cloud

Please use the tool *Matching Model and Pick Point Editor* to generate a model point cloud of the target object, and therefore Mech-Vision can compare the point cloud of the target object to the model point cloud and then generate an actual picking pose.

The model point cloud and pick point generated using the Matching Model and Pick Point Editor are shown below.

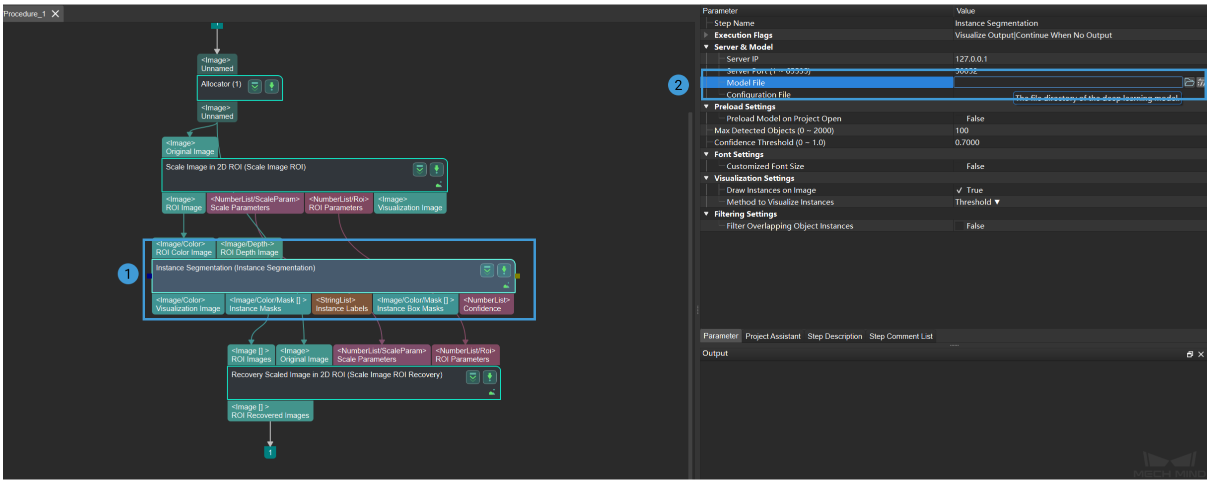
The generated model file will be saved in the project folder.

3.6.3 Project Deployment

During project deployment phase, you will need to set relevant parameters of Steps, and add the configuration files obtained in the preparation phase to corresponding Steps before actually running the project.

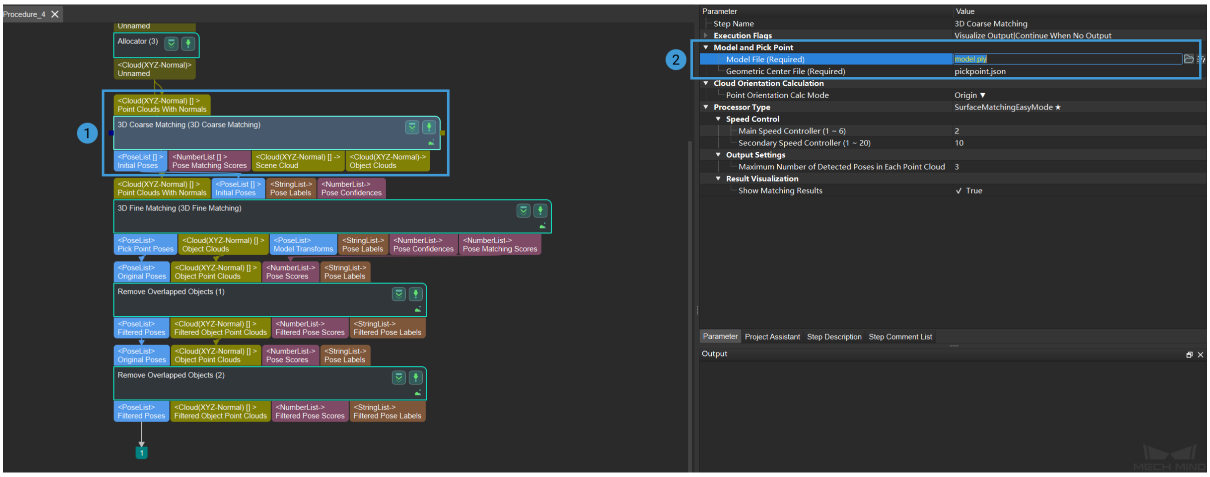
1. Set the DL Model File and Configuration File

- Double-click on the Procedure **Instance Segmentation** to display the detailed structure.
- Select the Step **Instance Segmentation** and click on the folder icon to set the model file and configuration file in the Parameter, as shown below.



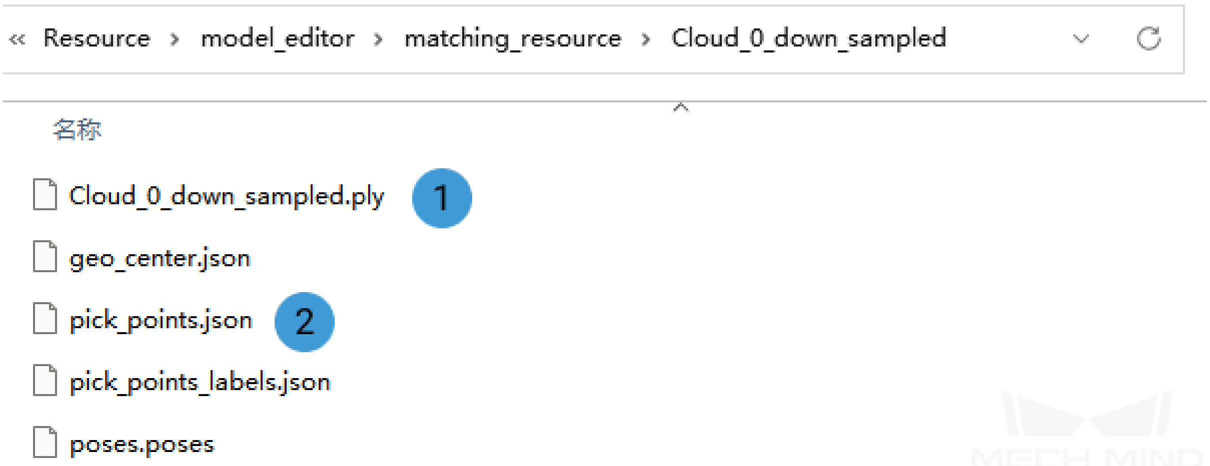
2. Set the Model File and Geometric Center File

- Double-click on the Procedure **3D Matching** to display the detailed structure.
- Select the Step **3D Coarse Matching** and click on the folder icon to set the model file and geometric center file in the Parameter, as shown below.



- The way to set model file and geometric center file in **3D Fine Matching** is the same as it is in **3D Coarse Matching**.

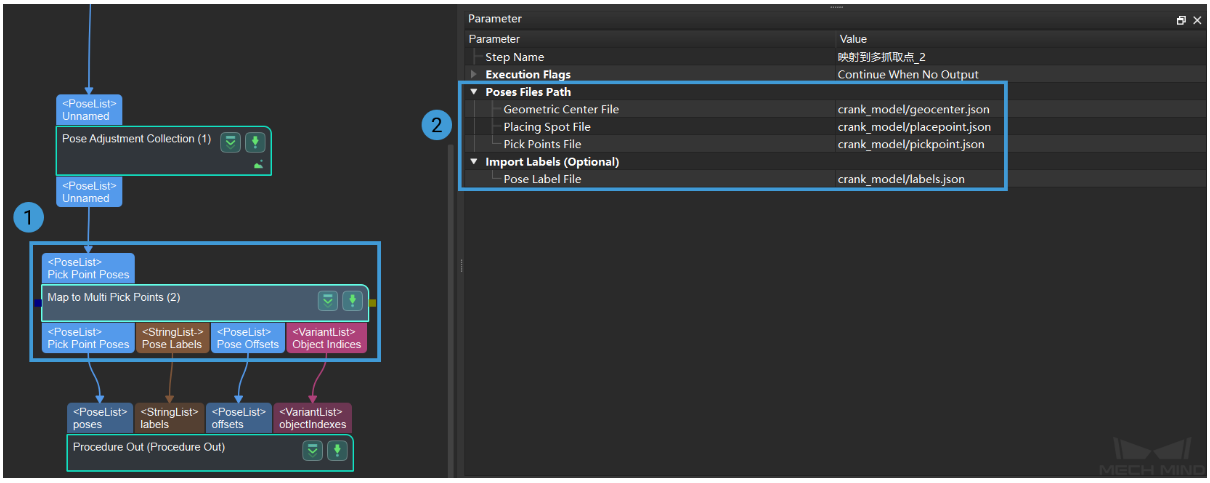
Click to see the example of each file



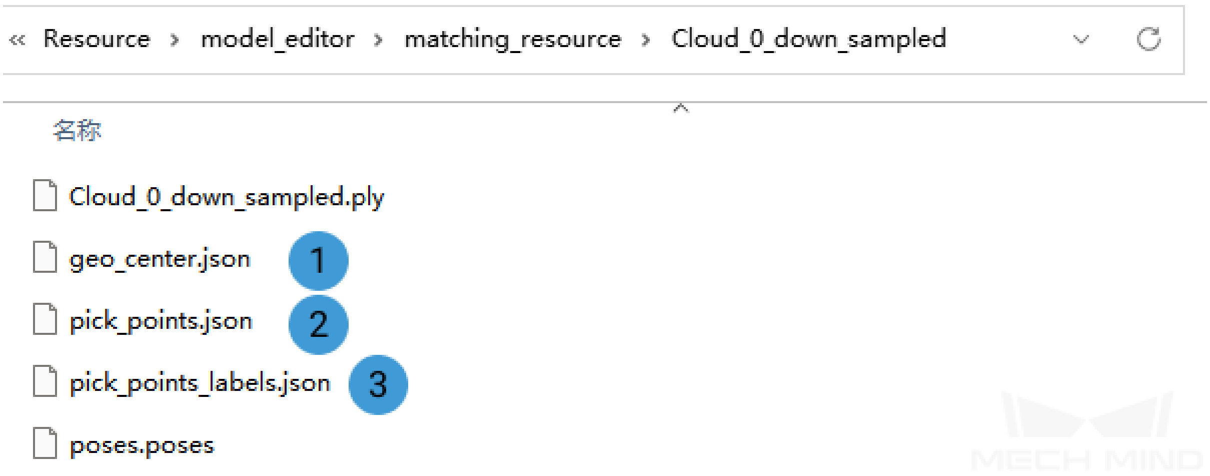
is a point cloud model file, and is a geometric center file.

3. Set Poses Files

- Select the Step **Map to Multi Pick Points** and click on  to set the poses files and important lables in Parameter, as shown below.



Click to see the example of each file



is a geometric center file, is a placing spot file, and is a pose label file.

3.6.4 Running and Debugging

After completing the project deployment, click on ▶ Run to run the project.

- Please see *Run the Project and Debug* for how to run and debug the project.
- Please see *Small Non-Planar Workpiece Loading* to learn about the algorithms and parameter adjustment of the project.

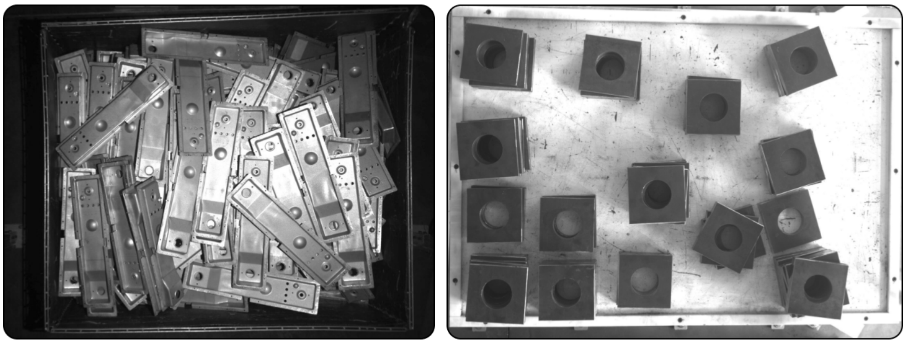
Tip: If you need to save the images or parameters when debugging the project or before training a deep learning model, you can use the tool *Data Storage*.

After successfully running and debugging the project, if you need to save the on-site data for future reference or you find that certain part of the project is not performing well and would like to optimize the Step or tune the parameters in an off-site situation, the tools *Data Storage* and *Data Playback* can be very useful.

Prerequisites for using the tool *Data Playback*:

1. A project file in which the project can run correctly without errors.
2. On-site source data which is gathered during the whole period while running the project, including 2D color images, depth maps, camera parameter file. Please refer to *Data Storage* for instructions on how to save the data.

3.7 Overlapping Planar Workpiece Loading



In overlapping planar workpiece loading scenario, the number of the workpieces to be processed is large and their size is small. In this project, deep learning algorithms and 3D matching algorithms are used to recognize the overlapping planar workpieces.

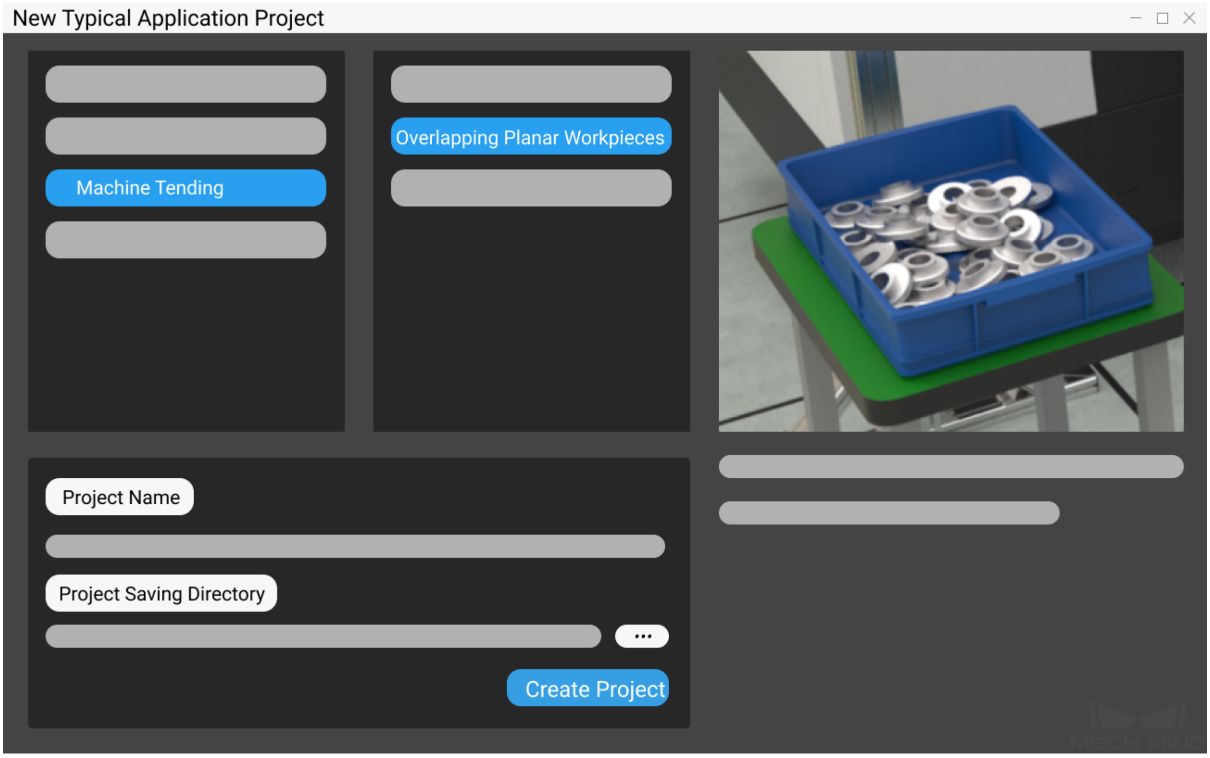
In actual projects, the top and bottom surfaces of some types of workpieces are dissimilar, and therefore the algorithms need to distinguish the top surface from bottom surface during picking. For other workpieces, their top and bottom surfaces are similar, and there is no need to distinguish the two surfaces during picking. This section focuses on the latter case where the algorithms do not need to distinguish the top surface from bottom surface of workpieces.

If a super model of the typical workpiece is available, it can be used in the project directly without deep learning training. You can contact the technical team to inquire if there is an available super model.

3.7.1 Create a New Typical Application Project

Create a Project

Click on *Typical Applications* → *New Typical Application Project* in the Menu Bar or *New Typical Application Project* in the Toolbar to open the following window.



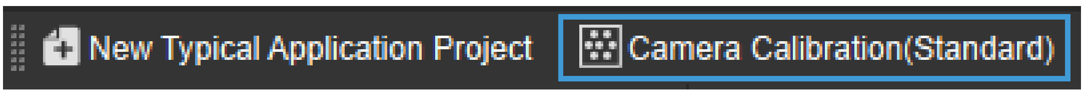
1. Select **Overlapping Planar Workpieces**.
2. Name the project.
3. Click on ... to select a folder to save the project (it is recommended to create an empty folder), then click *Create Project* .

3.7.2 Preparation

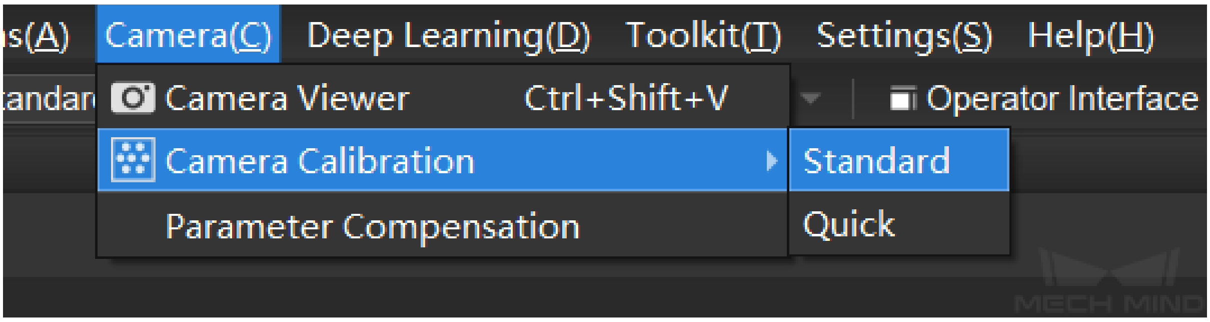
1. Calibrate the Camera

Calibrating the camera is to obtain the parameter file of the camera, which is used to determine the spatial relationship between the robot and camera.

Click on *Camera Calibration(Standard)* in the Toolbar to calibrate the camera.



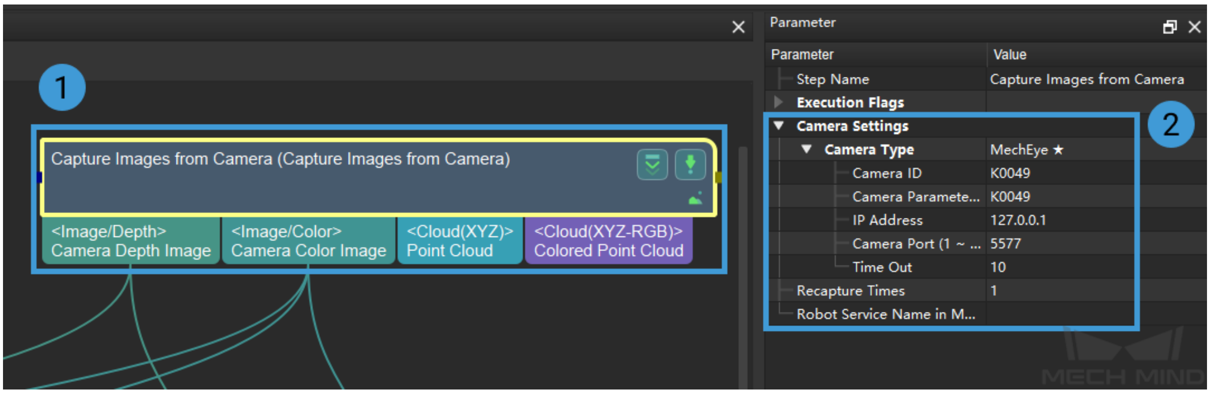
Alternatively, you can click on *Camera* → *Camera Calibration* → *Standard* in the Menu Bar to open the same window.



For more information about camera calibration, please refer to *Start Calibration - Standard Mode*.

2. Configure the Camera

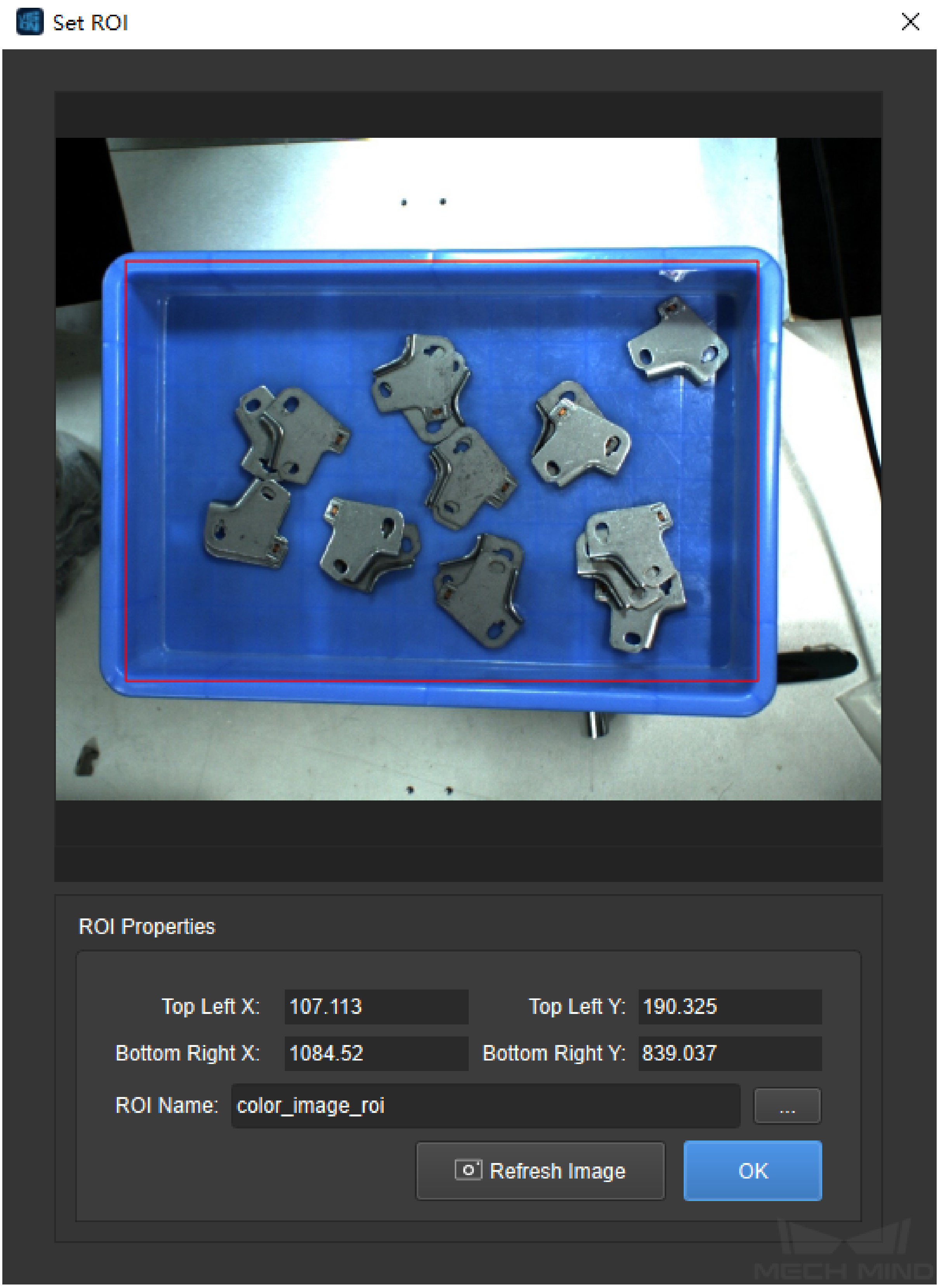
Before using the camera to capture images, you need to set the type, parameter group name, IP address and other parameters of the camera. For detailed instructions, please see *Capture Images from Camera*.



3. Set 2D and 3D ROI

Double click on the Procedure **Instance Segmentation** to display the detailed structure, and set 2D ROI in the Step **Scale Image in 2D ROI**.

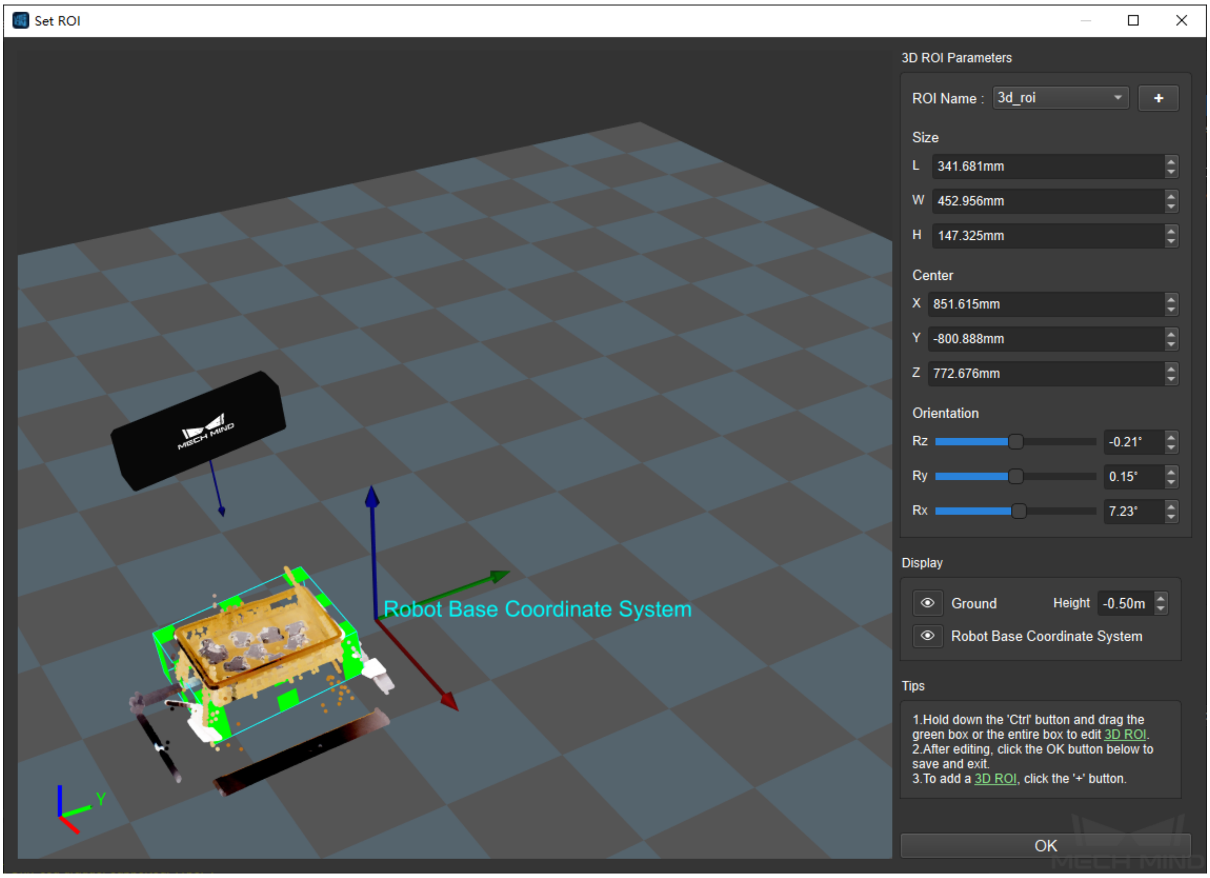
Setting a 2D ROI can avoid generating unnecessary point cloud and therefore increase the pre-processing speed. Please refer to *Instructions for Setting 2D ROI* for detailed instructions on setting 2D ROI.


Set ROI
ROI Properties
Top Left X: 107.113
Top Left Y: 190.325
Bottom Right X: 1084.52
Bottom Right Y: 839.037
ROI Name: color_image_roi
...
Refresh Image
OK

Attention: The way to set a 2D ROI in Step *From Depth Map to Point Cloud* of the Procedure **Point Cloud Pre-Processing** is the same as above.

Double click on the Procedure **Point Cloud Pre-Processing** to display the detailed structure, and set 3D ROI in the Step **Extract 3D Points 3D ROI**.

Setting a 3D ROI can extract the point cloud of the target objects and filtered the unwanted points in the backgroung. Please refer to *Instructions for Setting 3D ROI* for detailed instructions on setting 3D ROI.



1. Instance Segmentation

Note: If you already have a super model, you can skip this step and start configuring the deep learning model file.

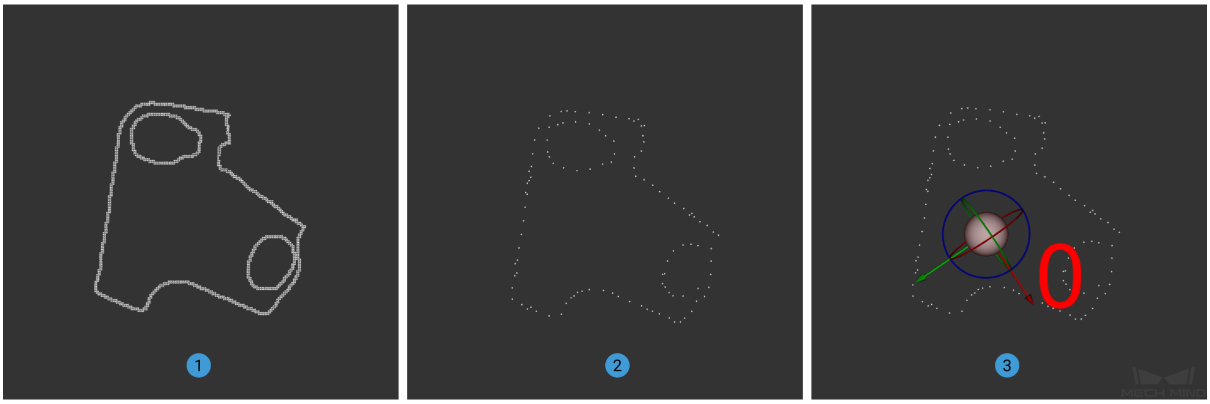
Instance segmentation is used for detecting and locating each distinct target object in an image, as shown below.

Please see Instance Segmentation for detailed information about training a deep learning model.

5. Generate a Model Point Cloud

Please use the tool *Matching Model and Pick Point Editor* to generate a model point cloud of the target object, so that Mech-Vision can compare the point cloud of the target object to the model point cloud and then generate an actual picking pose.

The model point cloud and pick point generated using the Matching Model and Pick Point Editor are shown below.



is the generated point cloud model of the overlapping planar workpiece; is the downsampling result; is the pick point.

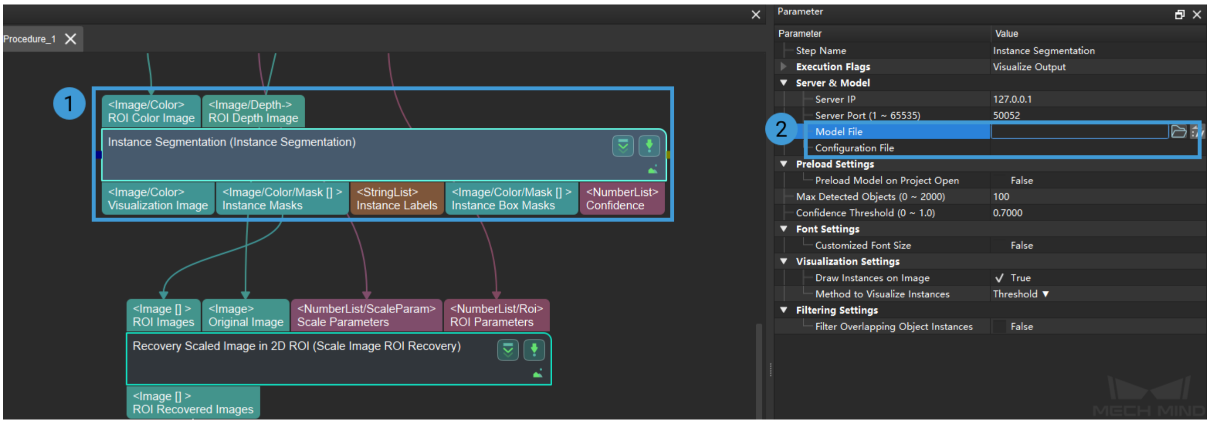
The generated model file will be saved in the project folder.

3.7.3 Project Deployment

During project deployment phase, you will need to set relevant parameters of Steps, and add the configuration files obtained in the preparation phase to corresponding Steps before actually running the project.

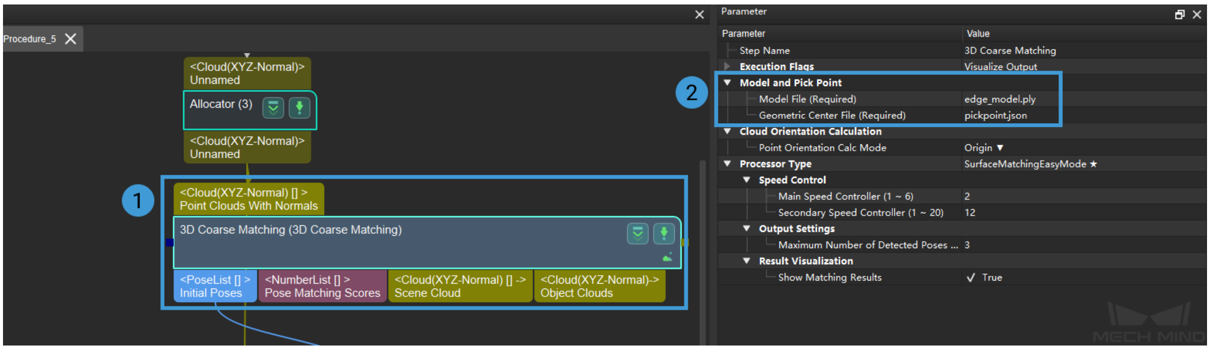
1. Set the DL Model File and Configuration File

- Double-click on the Procedure **Instance Segmentation** to display the detailed structure.
- Select the Step **Instance Segmentation** and click on to set the model file and configuration file in the Parameter, as shown below.



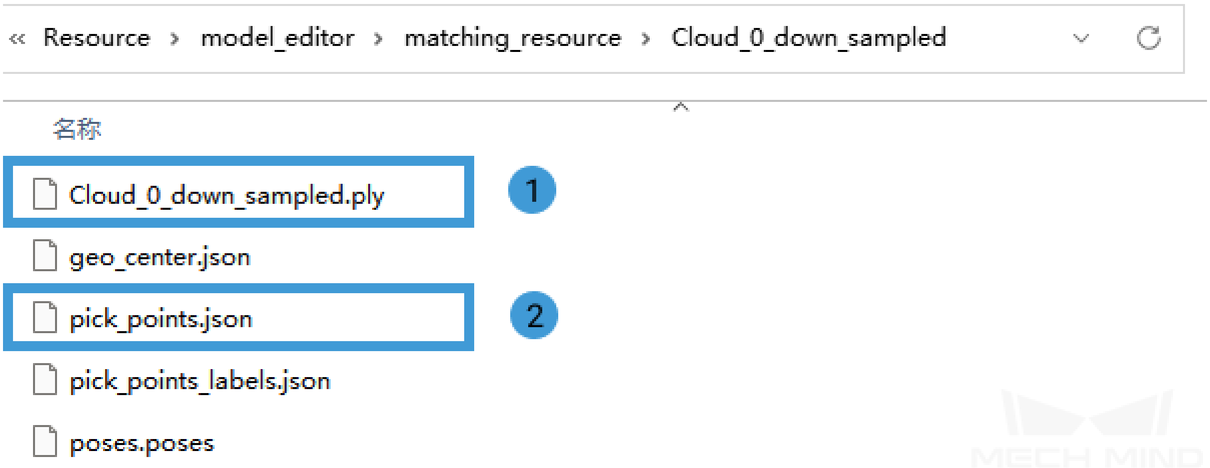
2. Set the Model File and Geometric Center File

- Double-click on the Procedure **3D Matching** to display the detailed structure.
- Select the Step **3D Coarse Matching** and click on to set the model file and geometric center file in the Parameter, as shown below.



- The way to set model file and geometric center file in **3D Fine Matching** is the same as it is in **3D Coarse Matching**.

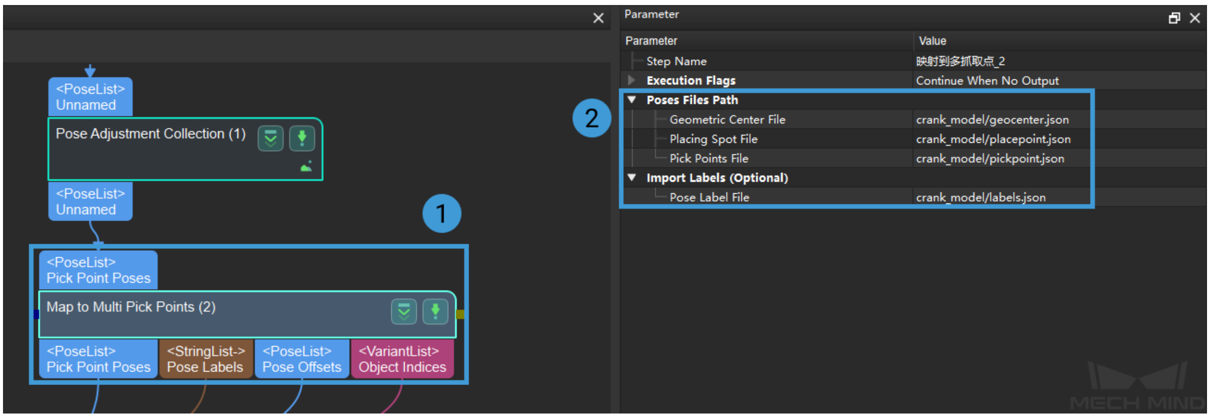
Click to see the example of each file



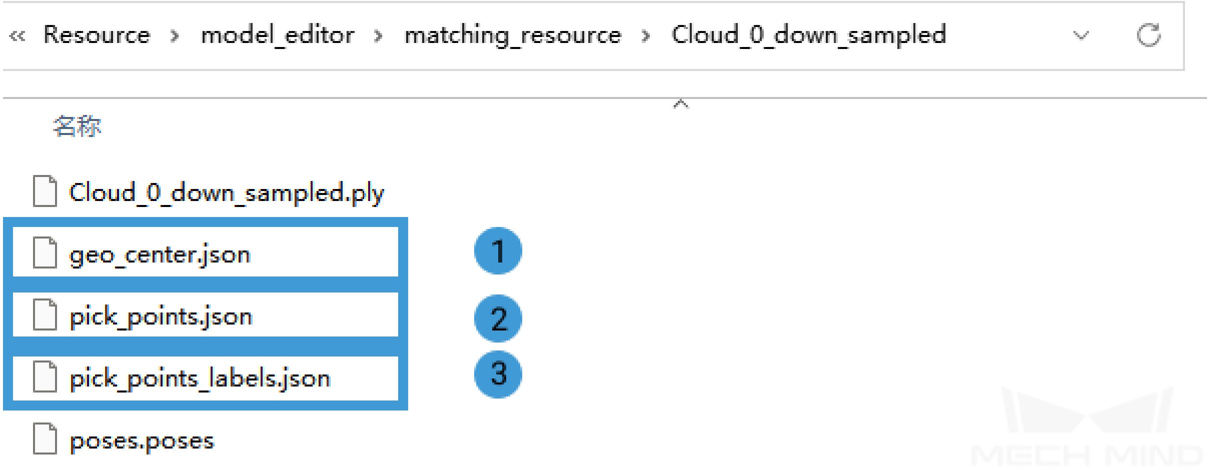
is a point cloud model file, and is a geometric center file.

3. Set Poses Files

- Select the Step **Map to Multi Pick Points** and click on to set the poses files and important lables in Parameter, as shown below.



Click to see the example of each file



is a geometric center file, is a placing spot file, and is a pose label file.

3.7.4 Running and Debugging

After completing the project deployment, click on  to run the project.

- Please see *Run the Project and Debug* for how to run and debug the project.
- Please see *Overlapping Planar Workpiece Loading* to learn about the algorithms and parameter adjustment of the project.

After successfully running and debugging the project, if you need to save the on-site data for future reference or you find that certain part of the project is not performing well and would like to optimize the Step or tune the parameters in an off-site situation, the tools *Data Storage* and *Data Playback* can be very useful.

Prerequisites for using the tool *Data Playback*:

1. A project file in which the project can run correctly without errors.
2. On-site source data which is gathered during the whole period while running the project, including 2D color images, depth maps, camera parameter file. Please refer to *Data Storage* for instructions on how to save the data.

Piece Picking (by Vacuum Gripper)


Applicable to the express industry. Able to deal with packaging cartons, parcels, envelopes, boxes, bags, bottles, etc. Able to work against reflective tapes, complex surface patterns, and parcel labels.

According to whether the object is placed in a bin, Piece Picking can be categorized into the following two scenarios:

3.8 Piece Picking

In piece picking, since there is no single model that can be applied to all objects of varying types and shapes, deep learning algorithms are needed to estimate pick points.

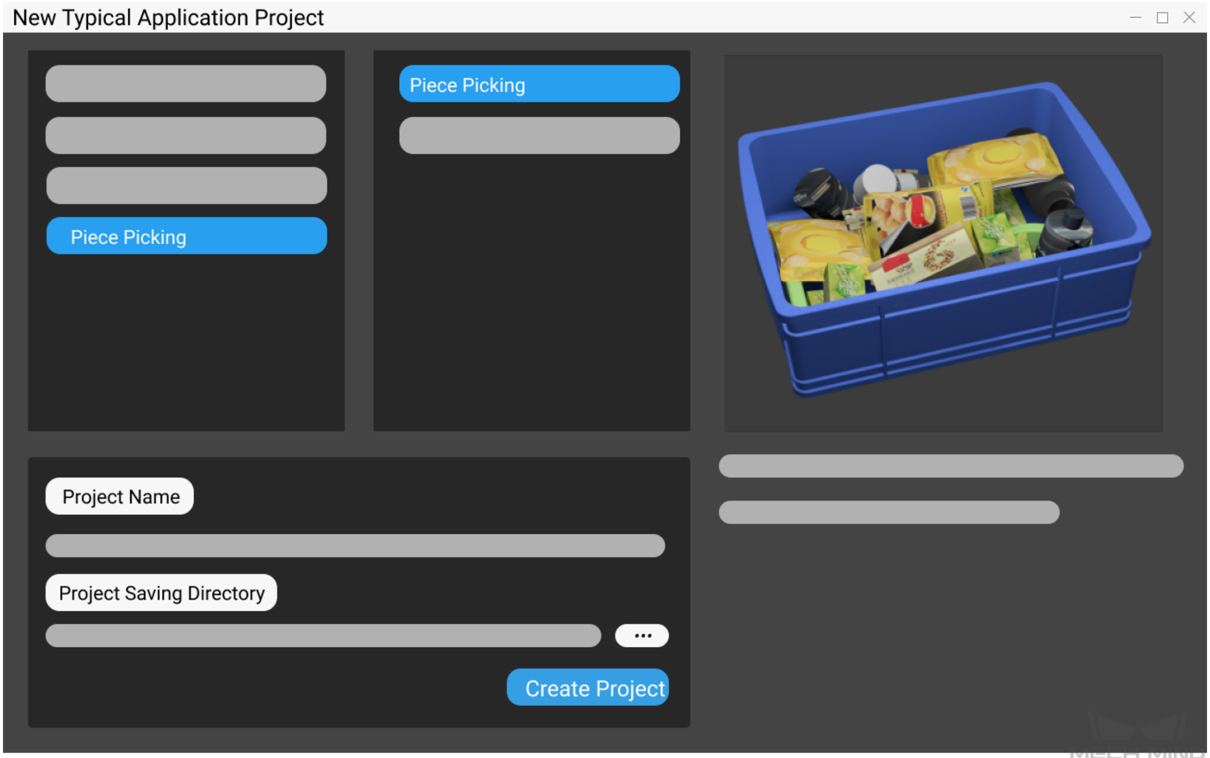
3.8.1 Preparation

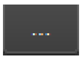
Before deploying the project, please obtain the super model (please contact the support team of Mech-Mind Robotics).

3.8.2 Instructions for Creating a New Typical Application Project

Create Project

Click on *Typical Applications* → *New Typical Application Project* in the Menu Bar or *New Typical Application Project* in the Toolbar to open the following window.



1. Select **Piece Picking**.
2. Name the project.
3. Click on  to select a folder to save the project (it is recommended to create an empty folder), then click *Create Project* .

3.8.3 Running and Debugging

- Before running the project, please refer to *Hand-Eye Calibration Guide* to complete the camera calibration.
- After completing the calibration, please make preparation to capture images by adjusting *Capture Images from Camera* after reading *Guide to Steps*.
- Please see *Run the Project and Debug* for how to run and debug the project.
- Please see *Piece Picking (with Bin)* to learn about the algorithms and parameter adjustment of the project.

After successfully running and debugging the project, if you need to save the on-site data for future reference or you find that certain part of the project is not performing well and would like to optimize the Step or tune the parameters in an off-site situation, the tools *Data Storage* and *Data Playback* can be very useful.

Prerequisites for using the tool *Data Playback*:

1. A project file in which the project can run correctly without errors.

2. On-site source data which is gathered during the whole period while running the project, including 2D color images, depth maps, camera parameter file. Please refer to *Data Storage* for instructions on how to save the data.

3.9 Piece Picking (Without Bin)

In piece picking, since there is no single model that can be applied to all objects of varying types and shapes, deep learning algorithms are needed to estimate pick points.

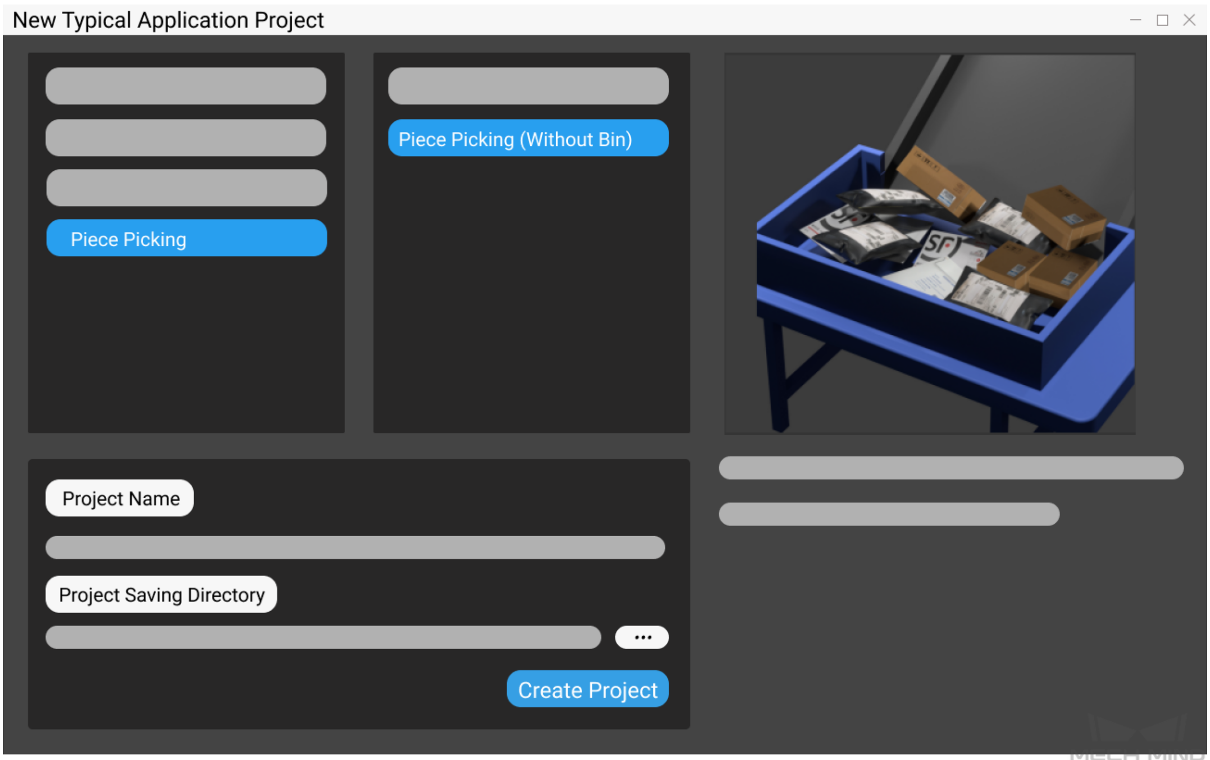
3.9.1 Preparation

Before deploying the project, please obtain the super model (please contact the support team of Mech-Mind Robotics).

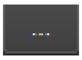
3.9.2 Instructions for Creating a New Typical Application Project

Create Project

Click on *Typical Applications* → *New Typical Application Project* in the Menu Bar or *New Typical Application Project* in the Toolbar to open the following window.



1. Select **Piece Picking (Without Bin)**.
2. Name the project.

3. Click on  to select a folder to save the project (it is recommended to create an empty folder), then click *Create Project* .

3.9.3 Running and Debugging

- Before running the project, please refer to *Hand-Eye Calibration Guide* to complete the camera calibration.
- After completing the calibration, please make preparation to capture images by adjusting *Capture Images from Camera* after reading *Guide to Steps*.
- Please see *Run the Project and Debug* for how to run and debug the project.
- Please see *Piece Picking (without Bin)* to learn about the algorithms and parameter adjustment of the project.

After successfully running and debugging the project, if you need to save the on-site data for future reference or you find that certain part of the project is not performing well and would like to optimize the Step or tune the parameters in an off-site situation, the tools *Data Storage* and *Data Playback* can be very useful.

Prerequisites for using the tool *Data Playback*:

1. A project file in which the project can run correctly without errors.
2. On-site source data which is gathered during the whole period while running the project, including 2D color images, depth maps, camera parameter file. Please refer to *Data Storage* for instructions on how to save the data.

High Precision Positioning

Suitable for applications that require a high precision in positioning, trajectory planning (such as gluing), mounting/assembly, etc. Commonly used in the furniture, automotive, home appliance, and other industries.

3.10 High Precision Positioning

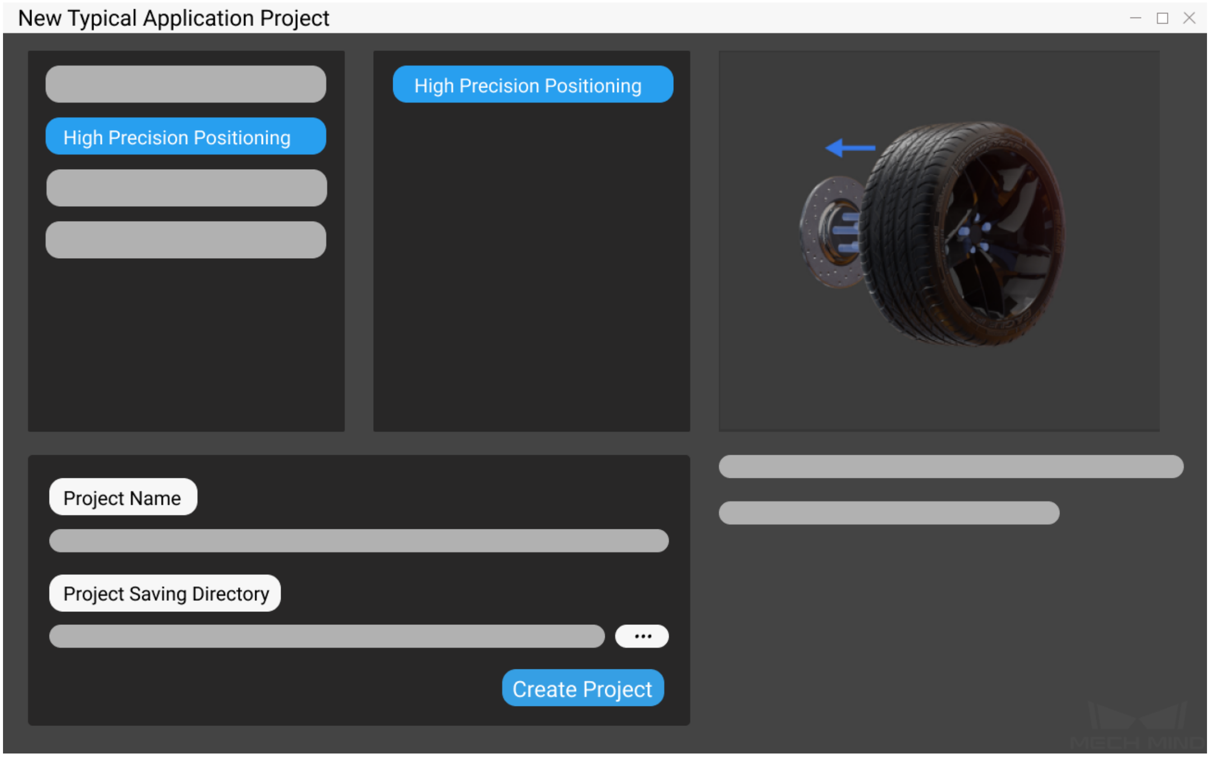
A high precision positioning is required in many scenarios, such as wheel hub positioning, tire tightening, and gearbox assembly in automotive industry. However, robots cannot directly receive high-precision position information of objects in the process for traditional automated manufacturing. Therefore, human intervention cannot be avoided, which results in reduced efficiency.

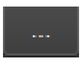
To solve this problem, Mech-Mind Robotics provides vision recognition results and 3D vision positioning technology enabling multi-model matching, which can obtain the precise three-dimensional pose information of objects and therefore improve the efficiency of high-precision assembly.

3.10.1 Instructions for Creating a New Typical Application Project

Create Project

Click on *Typical Applications* → *New Typical Application Project* in the Menu Bar or *New Typical Application Project* in the Toolbar to open the following window.



1. Select **High Precision Positioning**.
2. Name the project.
3. Click on  to select a folder to save the project (it is recommended to create an empty folder), then click *Create Project* .

3.10.2 Model Point Cloud Creation

After deploying the project, please complete the following preparation:

- Use *Matching Model and Pick Point Editor* to create a model point cloud of the target object for 3D matching Steps.

Hint: For instructions on model point cloud creation, please see *Matching Model and Pick Point Editor*.

3.10.3 Running and Debugging

- Before running the project, please refer to *Hand-Eye Calibration Guide* to complete the camera calibration.
- After completing the calibration, please make preparation to capture images by adjusting *Capture Images from Camera* after reading *Guide to Steps*.
- Please see *Run the Project and Debug* for how to run and debug the project.
- Please see *High Precision Positioning* to learn about the algorithms and parameter adjustment of the project.

After successfully running and debugging the project, if you need to save the on-site data for future reference or you find that certain part of the project is not performing well and would like to optimize the Step or tune the parameters in an off-site situation, the tools *Data Storage* and *Data Playback* can be very useful.

Prerequisites for using the tool *Data Playback*:

1. A project file in which the project can run correctly without errors.
2. On-site source data which is gathered during the whole period while running the project, including 2D color images, depth maps, camera parameter file. Please refer to *Data Storage* for instructions on how to save the data.

GUIDE TO STEPS

As Mech-Vision' s minimum algorithm unit for data processing, Steps are the basis of project construction.

A project is built by Steps that are connected together to process the input data by user-defined rules.

You do not need to write code to use Mech-Vision. Instead, you only need to drag and connect Steps to create your Mech-Vision Project.

Steps are divided into three types based on their roles.

Type	Description
Input	Input-type Steps input raw data to Mech-Vision and output data to processing-type Steps. They only have output ports at the bottom.
Processing	Processing-type Steps process data received from Input-type Steps or preceding processing-type Steps and output the data to Output-type Steps or succeeding Steps. They have input ports on the top and output ports at the bottom.
Output	Output-type Steps output processed data received from processing-type Steps. They only have input ports on the top and usually used to end the project.

Example of input-type Steps:

Example of processing-type Steps:

Example of output-type Steps:

The section below introduces the basic features of Steps and how to edit Steps in the graphical programming workspace.

4.1 Basic Features of Steps

Click the section below to learn about how to search for, add, and delete Steps:

Search/Add/Delete Steps

Click the section below to learn about the inputs and outputs of Steps:

Input/Output of Steps

Click the section below to learn about how to create and delete connections between Steps:

Create/Delete Connections

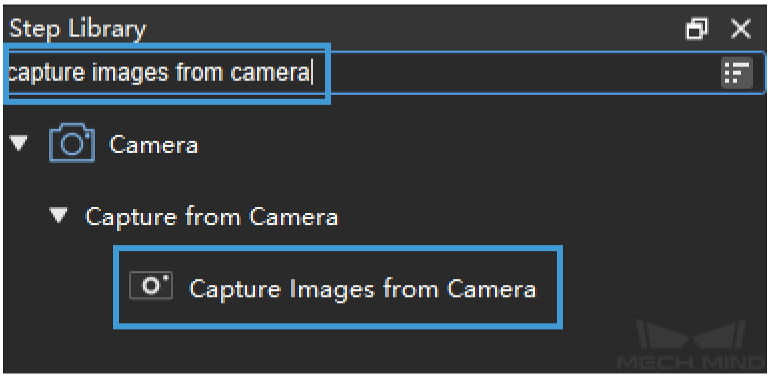
Click the section below to learn about different statuses of Steps:

Statuses of Steps

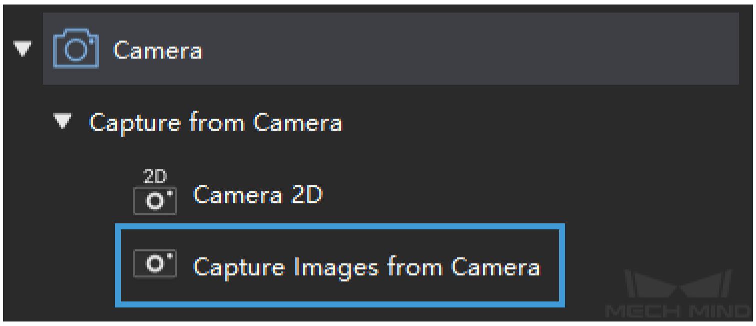
4.1.1 Search/Add/Delete Steps

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).



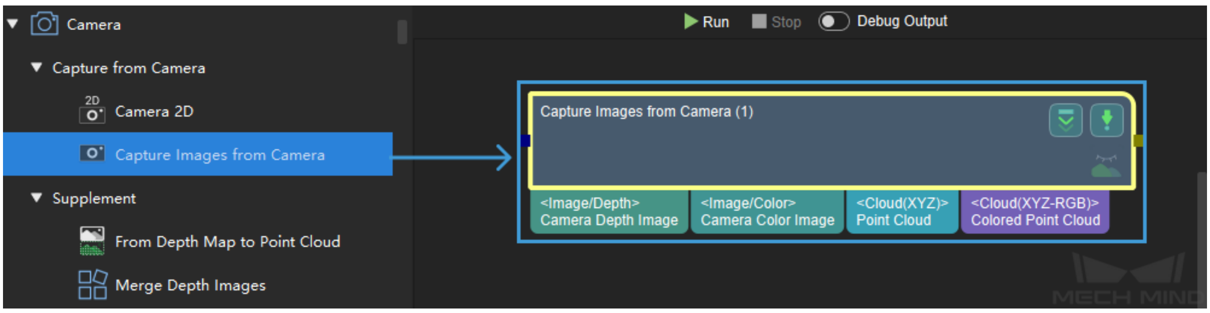
Click the corresponding Step category to expand the hierarchical list.



Add a Step

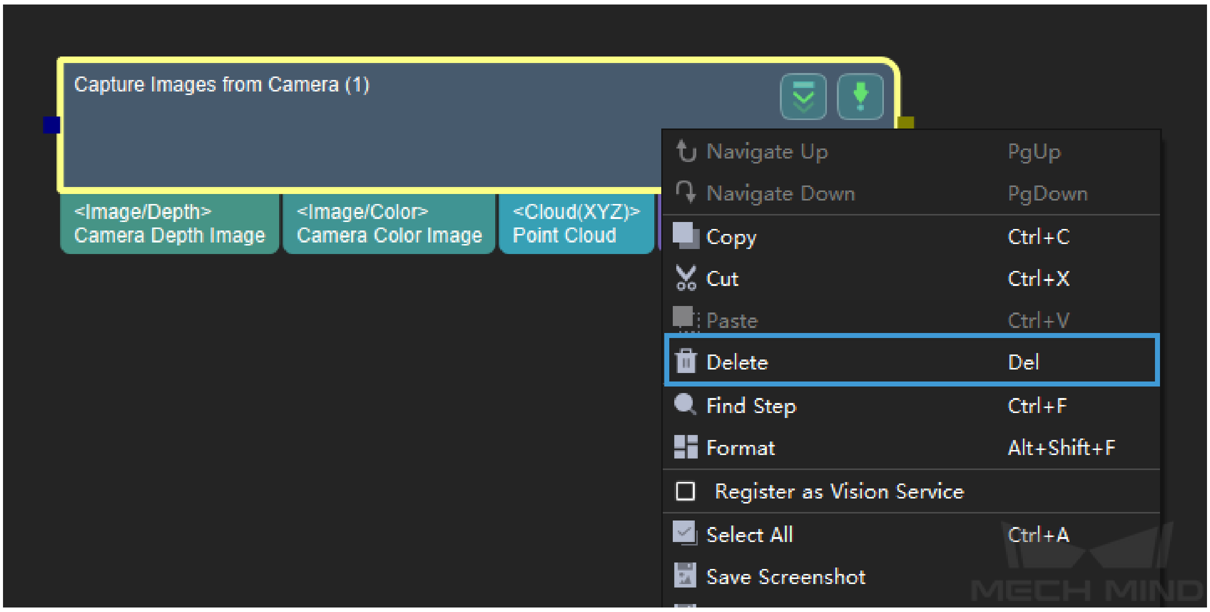
> **Attention:** You can add Steps only after you have *created a new project* or *opened an existing project*.

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



Delete a Step

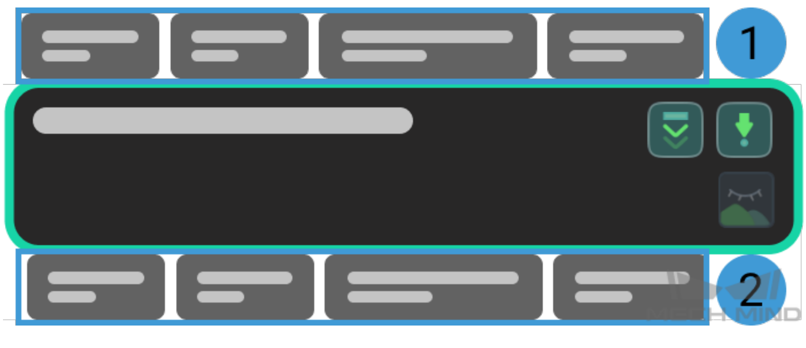
Left-click on a Step and press Delete on the keyboard. Right-click on a Step and click on *Delete* in the context menu.



4.1.2 Input/Output of Steps

You can view the input/output data type of a Step by hovering the mouse over the corresponding port.

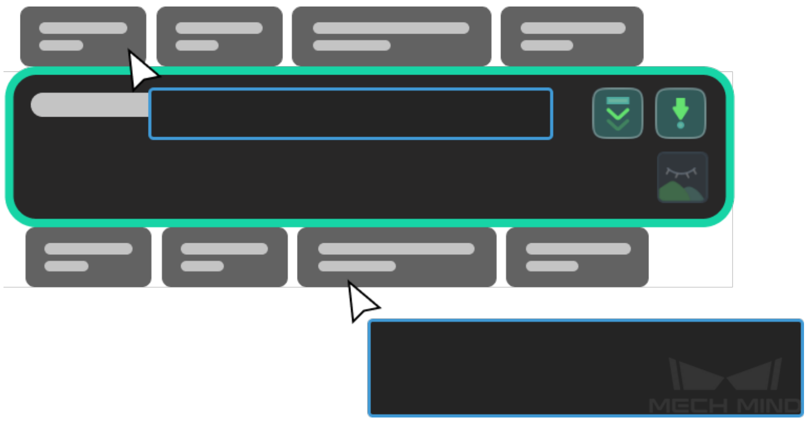
The input/output ports of each step are on the top or at the bottom of the Step, as shown below.



1. Input ports

2. Output ports

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



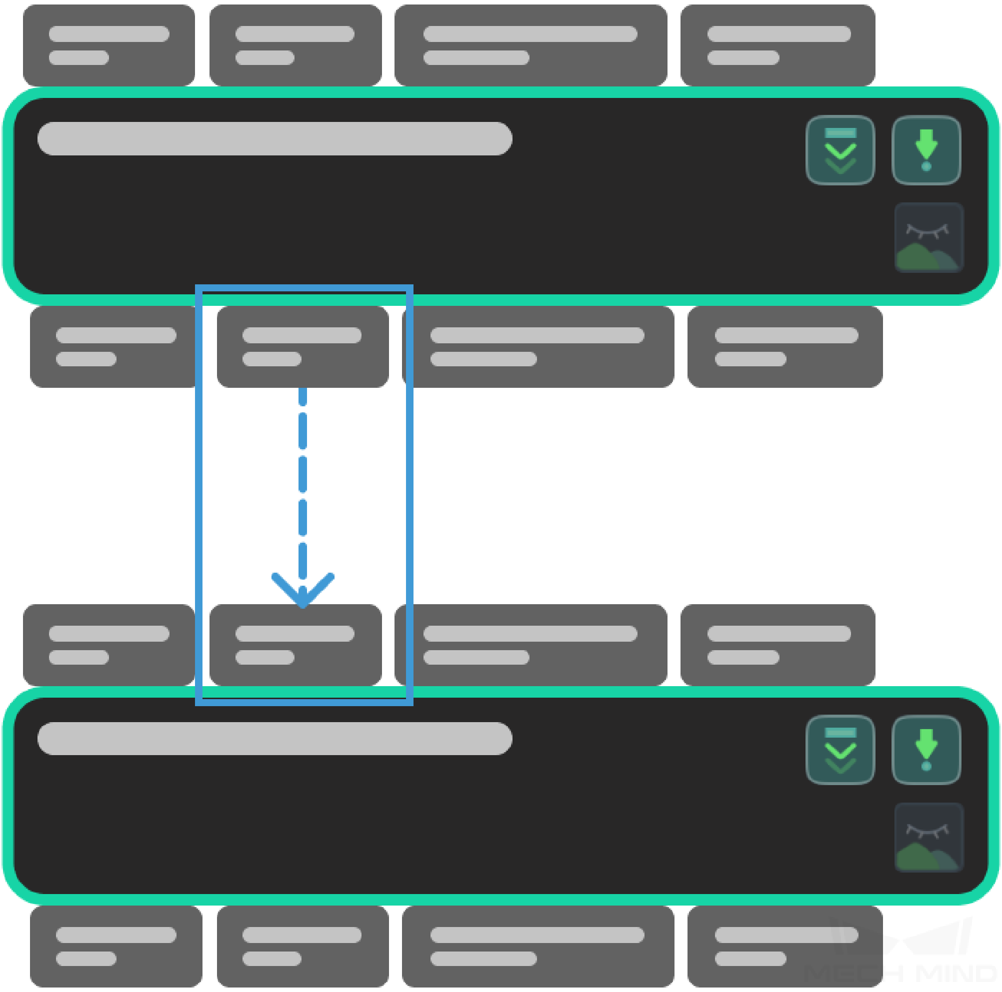
4.1.3 Create/Delete Connections

Create a Connection

> **Attention:** Connections can only be created between ports with the same data type. For example, an **Image/Color** output port can only be connected to an **Image/Color** input port. Ports of different data types cannot be connected.

Hover the mouse cursor 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.



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* .

4.1.4 Statuses of Steps

Regular	
Selected	
Warning	
Error	

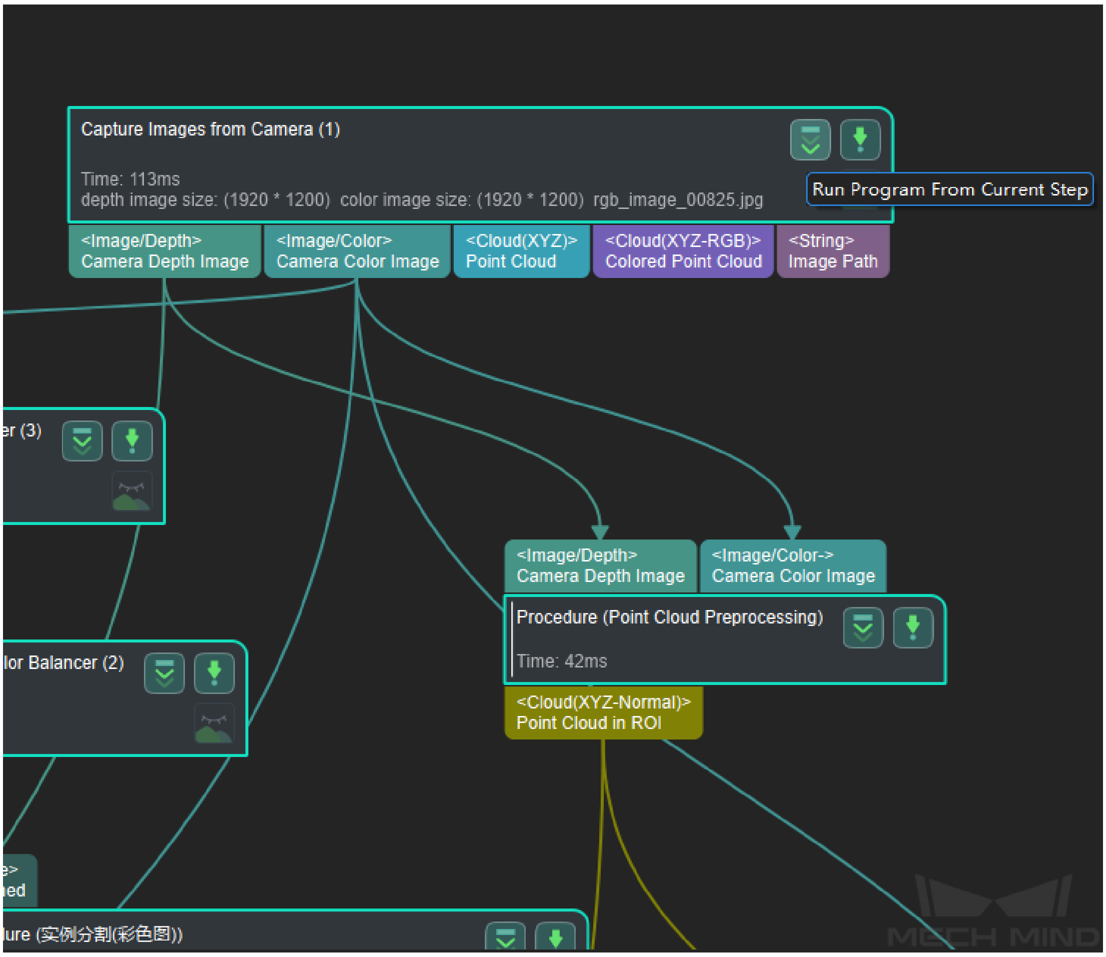
The section below introduces how to run Steps and view the outputs of individual results.

4.2 Run Steps and View Outputs

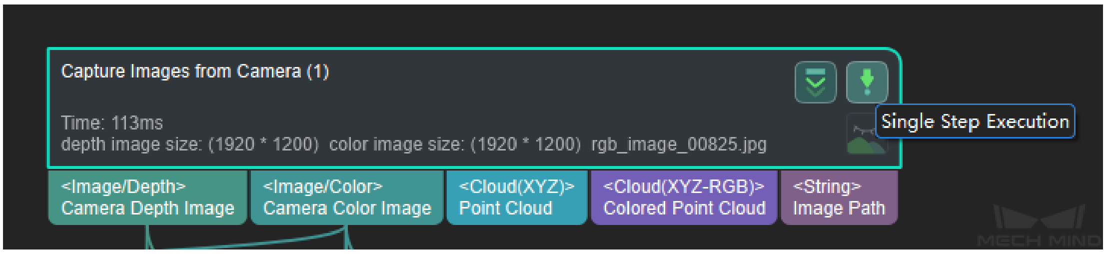
4.2.1 How to Run

- Clicking on 

in the Project Toolbar will run the entire project.
- Clicking on  on a Step will run the project down from the current Step.

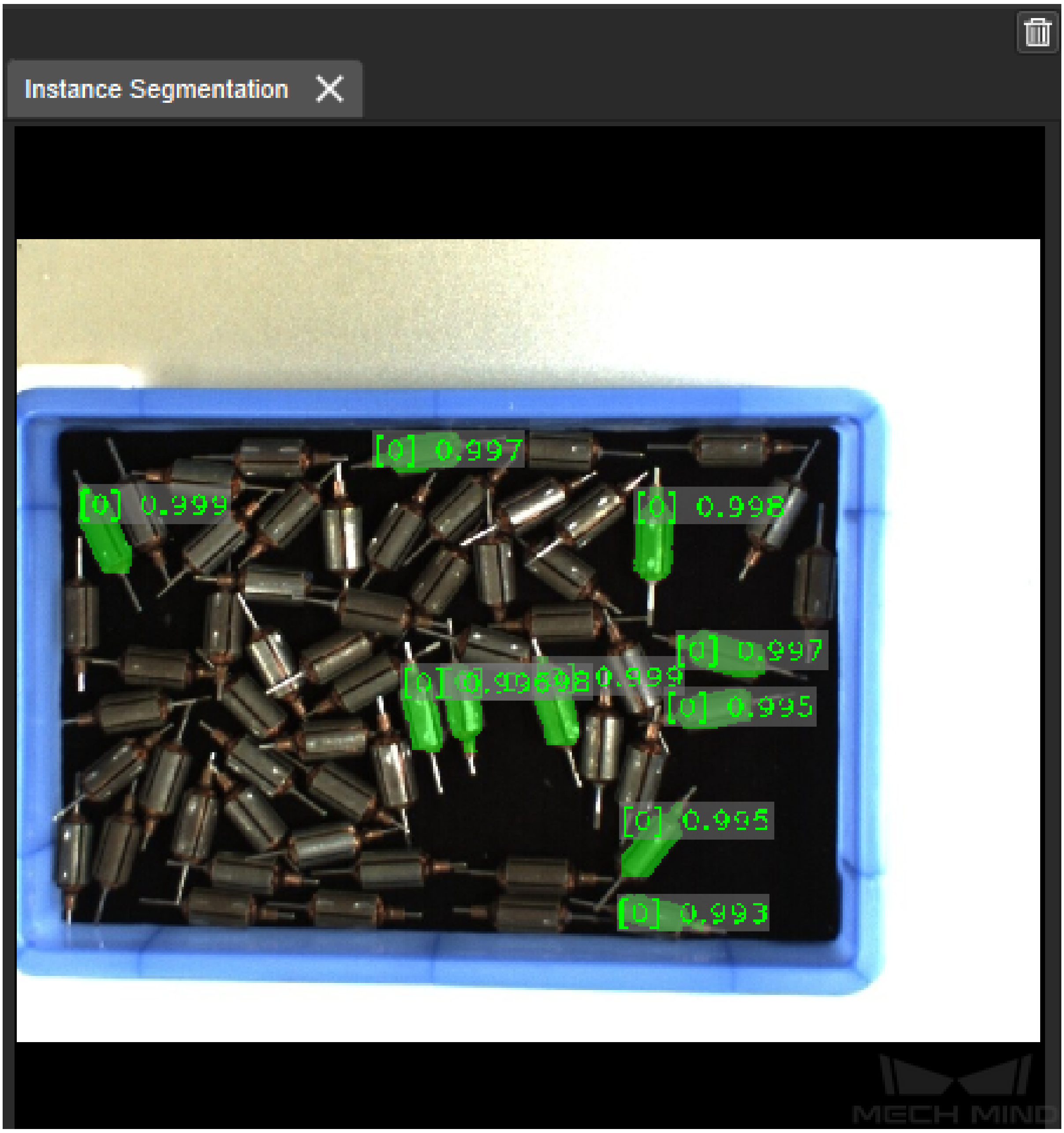


- Click on the button to run the current Step independently.



4.2.2 Debug Output

1. Click on Debug Output to switch on Debug Output.
2. Run the project; the visualized outputs of all Steps with the / icon will be displayed in the **Debug Output** panel in the upper right of Mech-Vision.



Hint:

- To view the visualized output of any Step after debugging, just run the Step independently using .
- If Debug Output is switched off, no visualized output will be displayed when running the project.
- Click on in the upper right of the **Debug Output** panel to close all the tabs displayed.

To view the visualized output of a single Step

- Method 1: Click on and hold, drag the mouse to the right and select *Visualize Output* in the pop-up menu. The visualized output of this Step will be displayed in the **Debug Output** panel.
- Method 2:
  1. Click on Debug Output to switch on Debug Output.
  2. Click on on a Step to turn on output visualization (the icon becomes if output visualization is turned on).
  3. Click on to run the Step independently, and the visualized output will be displayed in the **Debug Output** panel.

4.2.3 How to View the Output

After running a Step, double-click the connection under the port to display the output data of the port.

Tip: You can connect the Step *Accept All* to any output ports from any Steps to view Step output.

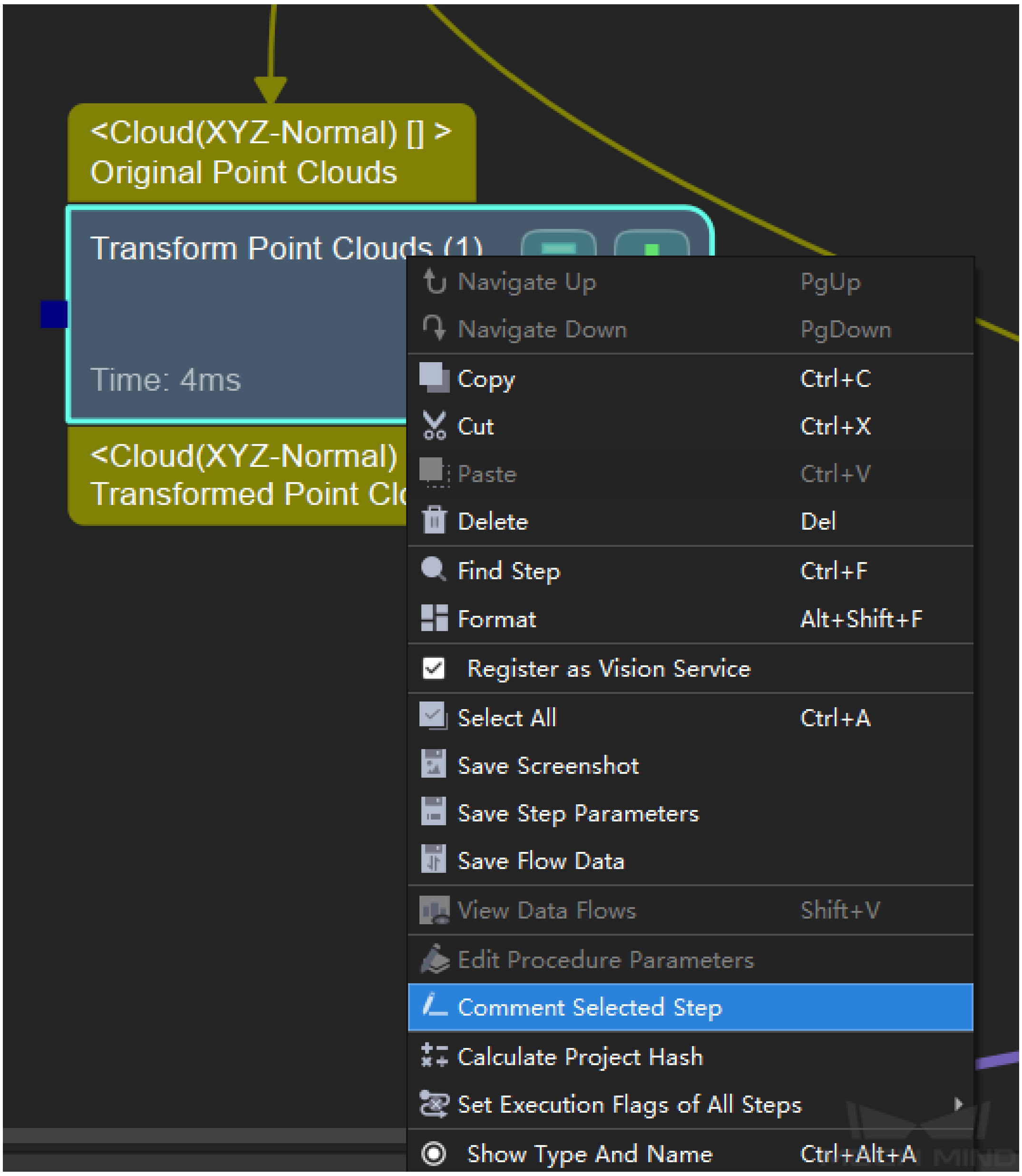
The section below describes how to add comments to Steps for project editing and debugging purposes.

4.3 Introduction to Step Comments

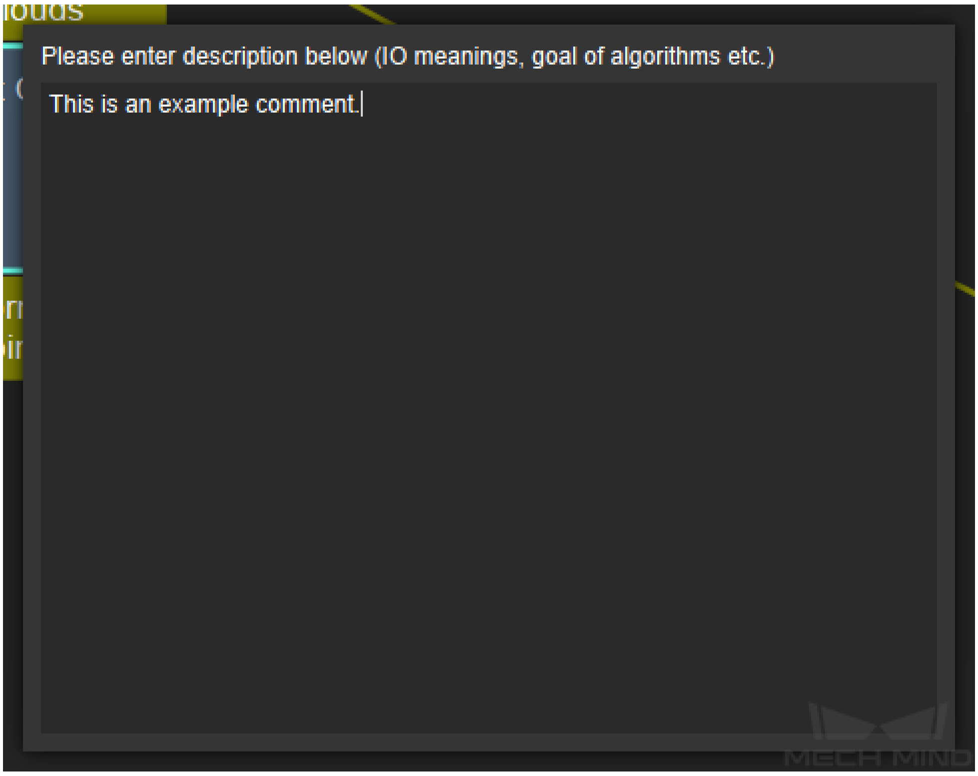
You can add a comment or description to a certain Step or Procedure in the project.

4.3.1 How to Add Comments

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

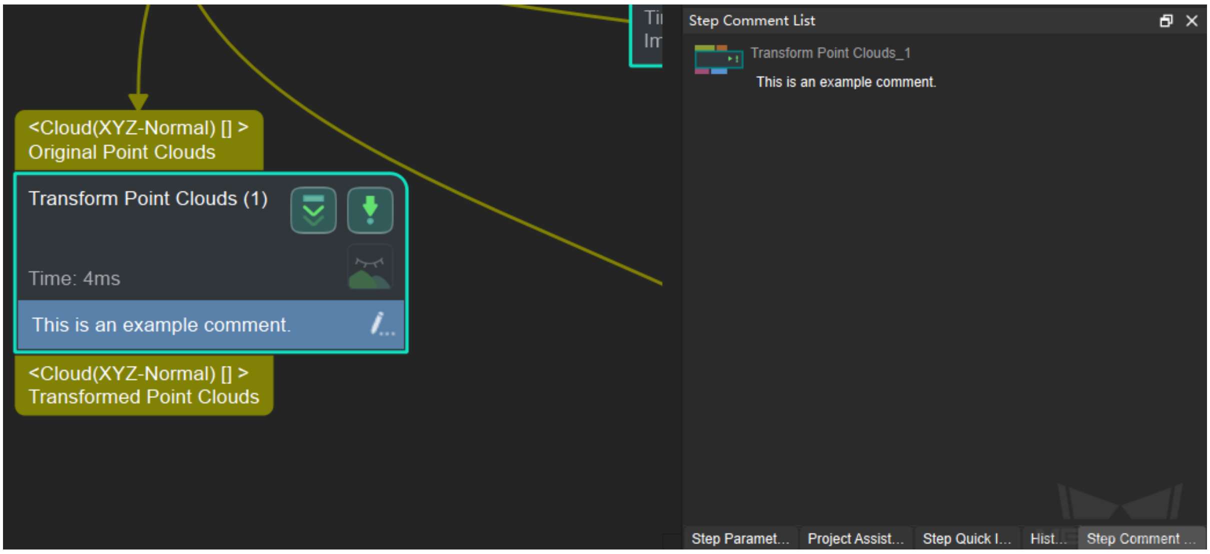


2. Enter your comment in the pop-up window, and then click on anywhere in Mech-Vision to close the window.



4.3.2 How to View the Comment

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



Each Step has many adjustable parameters. Adjusting parameters of a Step will affect the outputs of the Step and thus affect the output of the entire project.

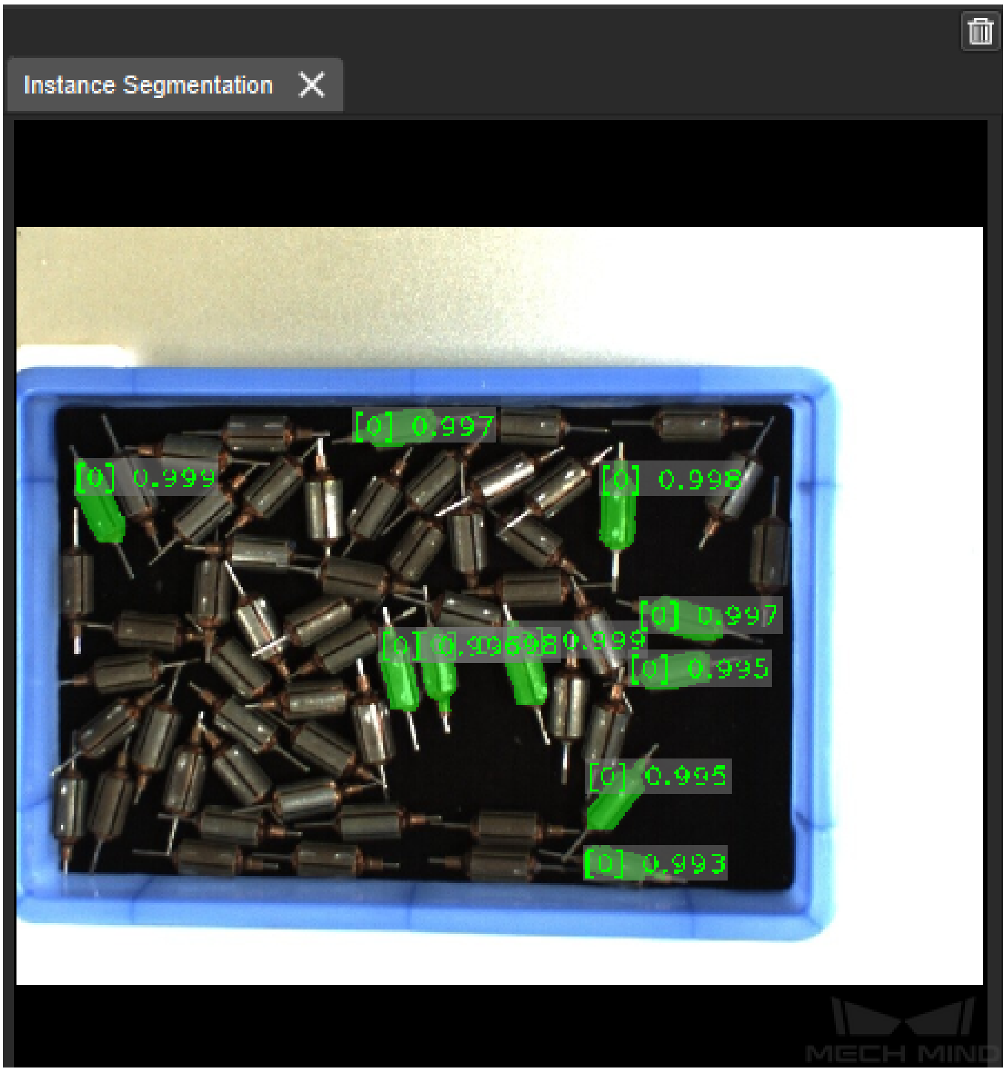
The section below describes the general parameters of the Steps.

4.4 Introduction to General Parameters of Steps

4.4.1 Execution Flags

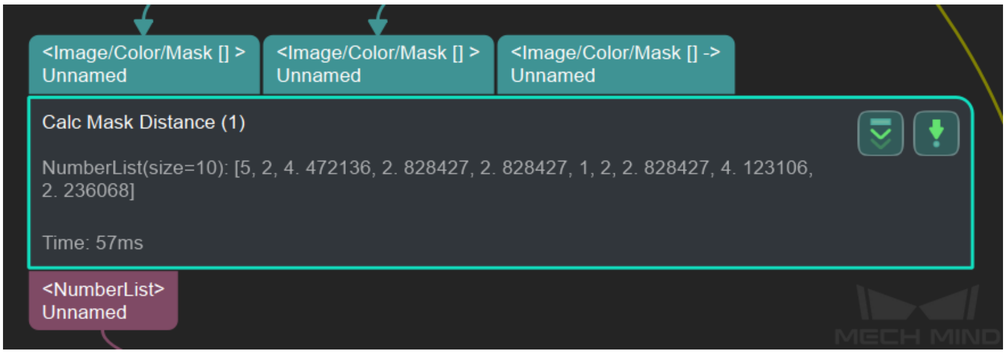
Visualize Output

Visualize Output displays the Step’s output as an image, as shown below. This feature helps determine whether the output is correct, and thus the parameters in the Step can be optimized to achieve an desired result.



Textualize Output

Textualize Output displays the Step's output as texts, as shown below. This feature helps determine whether the output is correct, and thus the parameters in the Step can be optimized to achieve an desired result.



Reuse Input

Adjust **Reuse Input** does not affect the actual output. It is only used as a mark to show that the Step supports 1xn input.

Continue When No Output

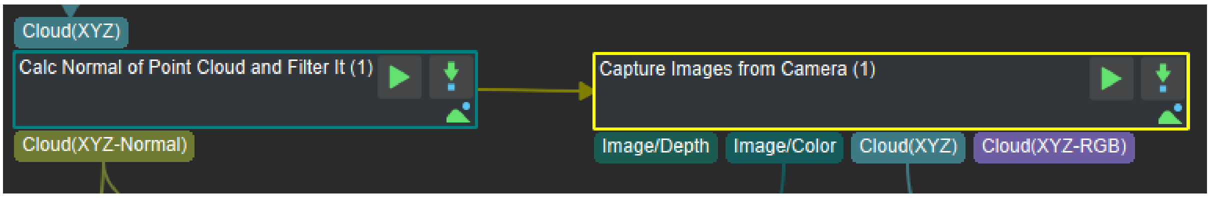
Continue execution when there is no output from the Step while running the project.

Reload File

Reload files each time the project is run. If set to `True` and the path to save the required file is changed, a file path error will be prompted when running the project.

Trigger Control Flow When No Output

Trigger a control flow when the current Step outputs no data. Take the example shown below, when there is no output in the *Calc Normals of Point Cloud and Filter It* Step, the *Capture Images from Camera* Step will be run.



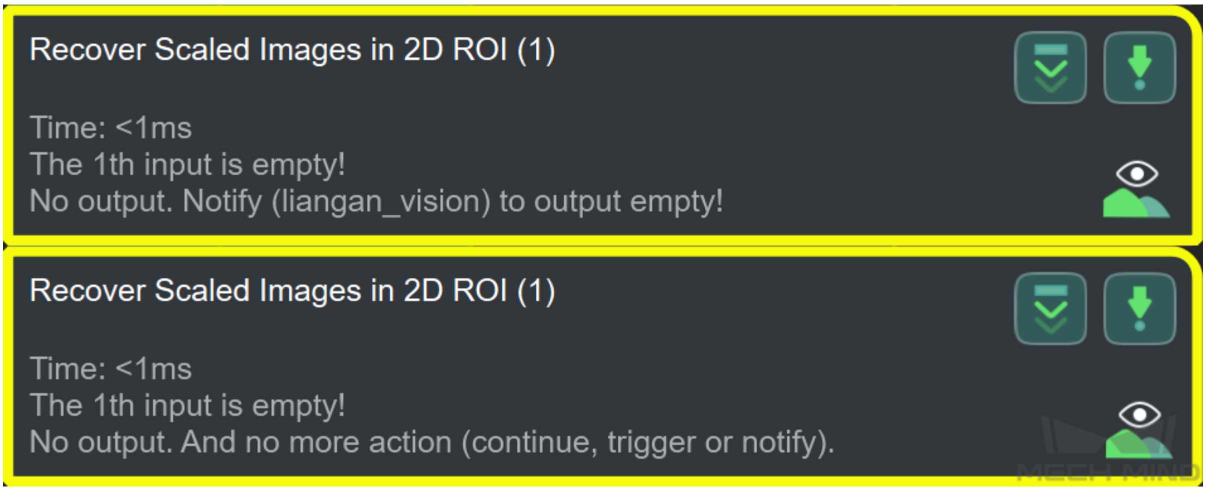
Trigger Control Flow When Output

Trigger a control flow when the current Step outputs data. Take the example shown above, when there is output in the *Calc Normals of Point Cloud and Filter It* Step, the *Capture Images from Camera* Step will be run.

Figure1 Trigger Control Flow

Notify Procedure Out When No Output

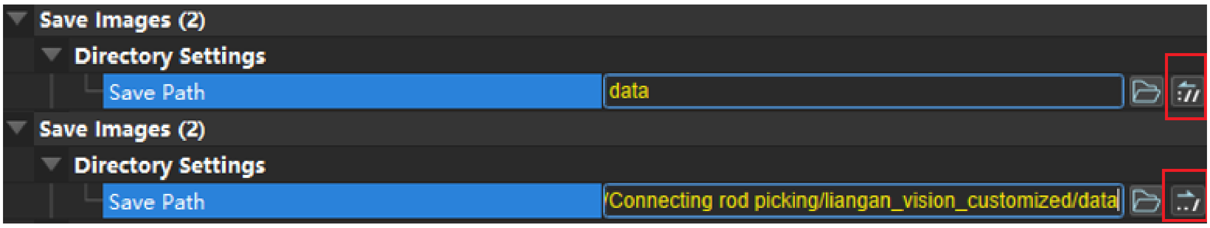
When the Step outputs no data, a message stating that the project has no output and exit will be prompted as shown below. The top figure is the running results when this flag is set to `True` and the bottom figure `False`.



4.4.2 Switch between Relative Path and Absolute Path

In Steps such as *Save Images and Step Parameters*, *Save Images*, *Save Results to File*, *Capture Images from Camera*, etc., which involve selecting the path to read or save the data, the default data path is the relative path.

- Click to convert the relative path to an absolute path.
- Click to convert the absolute path to a relative path.



Hint: You can search the keyword or check Contents on the left side of this webpage to open the pages of Steps you need to learn about.

The section below introduces commonly-used Steps in typical application projects.

4.5 Introduction to commonly used Steps in Typical Projects

4.5.1 Use and Adjustment of commonly used Steps in the "Depalletizing" scenario

Invalidate Depth Pixels Outside 3D ROI
Get Highest Areas in Depth Image
Extract 3D Points in 3D ROI
Scale Image in 2D ROI
Instance Segmentation
Read Object Dimensions
2D Matching
Remove Polygons outside Mask
Remove Overlapped Polygons
Show Images

4.5.2 Use and Adjustment of commonly used in the "Machine Tending" scenario

Point Filter
Point Cloud Clustering
Merge Point Clouds
Estimate Point Cloud Edges by 2D Method
Estimate Point Cloud Edges by 3D Method
3D Coarse Matching
3D Coarse Matching (Multiple models)
3D Fine Matching
3D Fine Matching (Multiple Models)
Flip Poses' Axes
Transform Poses
Show Point Clouds and Poses

4.6 2D Matching

In this section, Steps that are associated with 2D Matching will be introduced.

4.6.1 2D Matching

2D Matching

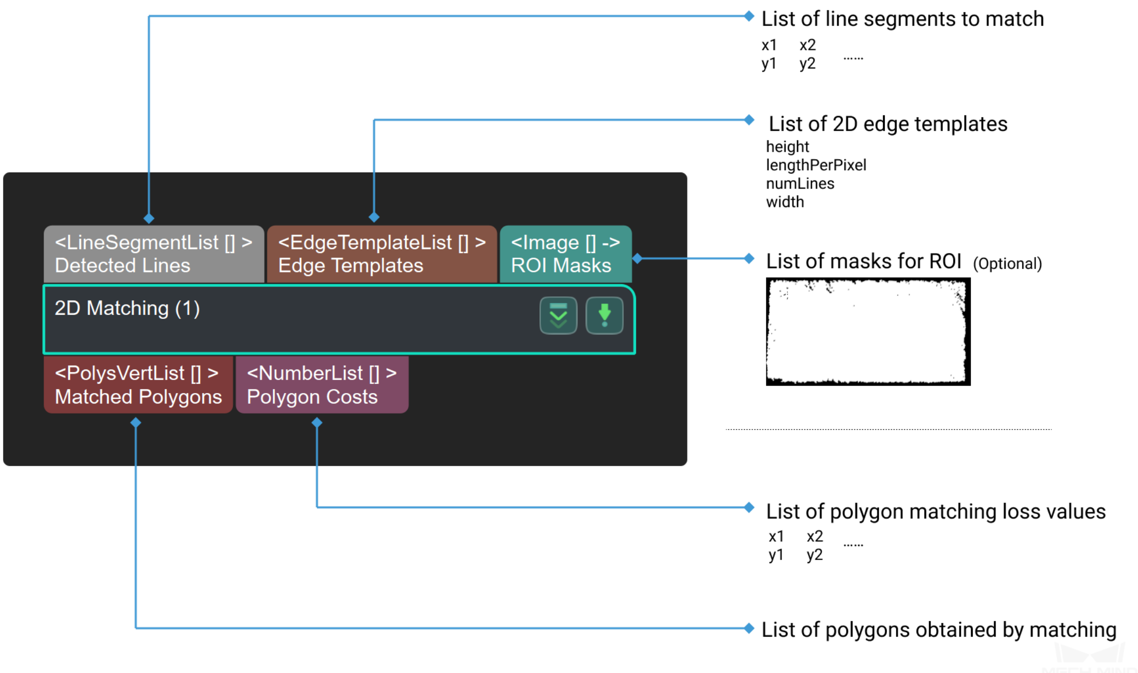
Function

Perform template matching on the specified shapes using a 2D matching algorithm.

Sample Scenario

Locate polygon objects on the 2D image, to provide detected polygon features for the subsequent calculation of the 3D pose of the polygons. Usually used with the Step *Remove Polygons outside Mask*.

Input and Output



Parameters

Template Variant

Line Angle Quantization(0~360)

Default Value: 60l

Instruction: Quantitative decomposition of the contour direction. For rectangles: quantize 180 degrees. (The rectangle has 180 degrees rotational symmetry.) The parameter to be filled in is the number of quantified copies:

if the parameter is set to `90` , then 180°/90=2°. The more quantization, the finer the edges, and more time is needed to complete the process as well.

Example:

When the parameter value is set to `30` , the step running time and the quantization result are shown in *Figure 1*.

Figure 1. The Result of setting the parameter value to 30

When the parameter value is set to `90` , the step running time and the quantization result are shown in *Figure 2*.

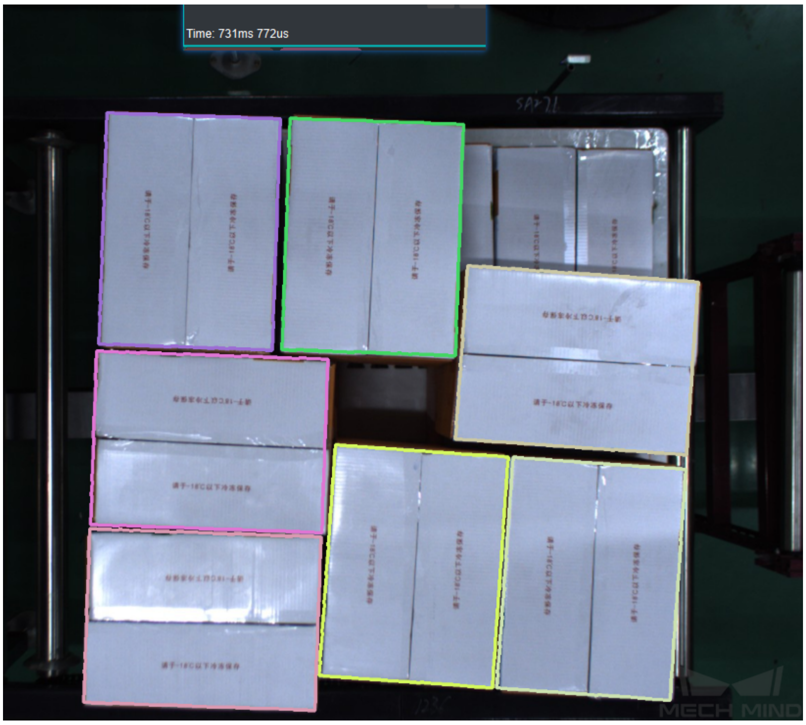


Figure 2. The Result of setting the parameter value to 90

Direction Cost Weight

Default Value: 0.1

Instruction: The higher the value, the more accurate the result, but more time-consuming as well.

Base Scale

Default Value: 1.1

Instruction: The template will be scaled by "Base Scale Ratio" ^" Scale Index" . The "Scale Index" is an integer within ["Min Scale Exponent" ," Max Scale Exponent"]. The higher the value, the more accurate the result, but more time-consuming as well.

Min Scale Exponent

Default Value: 0

Instruction: Reference value: -3, -2, -1, 0, the lower the value, the greater the possibility of matching the target correctly, but more time-consuming as well.

Max Scale Exponent

Default Value: 0

Instruction: Reference value: 0, 1, 2, 3. The higher the value, the greater the possibility of matching the target correctly, but more time-consuming as well.

Base Aspect Ratio

Default Value: 1.1

Instruction: The template will be scaled by "Base Aspect Ratio" ^" Aspect Ratio Index" . The "Base Aspect Ratio" is an integer within ["Min Aspect Exponent" , "Max Aspect Exponent"]. The higher the value, the greater the possibility of a correct match, but it is more time-consuming.

Min Aspect Exponent

Default Value: 0

Instruction: Reference value: -3, -2, -1, 0, the lower the value, the greater the possibility of matching the target correctly, but more time-consuming as well.

Max Aspect Exponent

Default Value: 0

Instruction: Reference value: 0, 1, 2, 3. The higher the value, the greater the possibility of matching the target correctly, but more time-consuming as well.

Candidate Search Settings

Coarse Orientation of Objects

Default Value: 0

Instruction: The rough direction of objects in the scene. For a rectangle, the direction of the horizontal side is defined as the 0 degree direction, and the vertical direction is defined as the 90 degree direction.

Search Range of Orientation

Default Value: 90

Instruction: The range of objects' poses in the scene.

Translation Step of Template

Default Value: 0.005

Instruction: The lower the value, the more accurate the result, but more time-consuming.

Rotation Step of Template

Default Value: 1.5

Instruction: The lower the value, the more accurate the result, but more time-consuming.

Threshold Setting

Max Mean Distance Error

Default Value: 0.008

Instruction: The maximum average distance error of each matching template. The higher the value, the more accurate the result.

Example: As shown in *Figure3*, the left side is the output result in the default state, and the right side is the output result with the parameter set to 0.002.

Figure 3. Max mean distance error

Overlap Threshold

Default Value: 0.5

Instruction: If the overlap of the results is higher than this value, they will be deleted. The higher the value, the more overlapping results tend to be output.

Example: As shown in *Figure4*, the left side is the output result when the parameter is set to 0.2, and the right side is the output result when the parameter is set to 0.5.

Figure 4. Overlap threshold

4.6.2 Calc Pose

Calc Poses and Dimensions of Rectangles

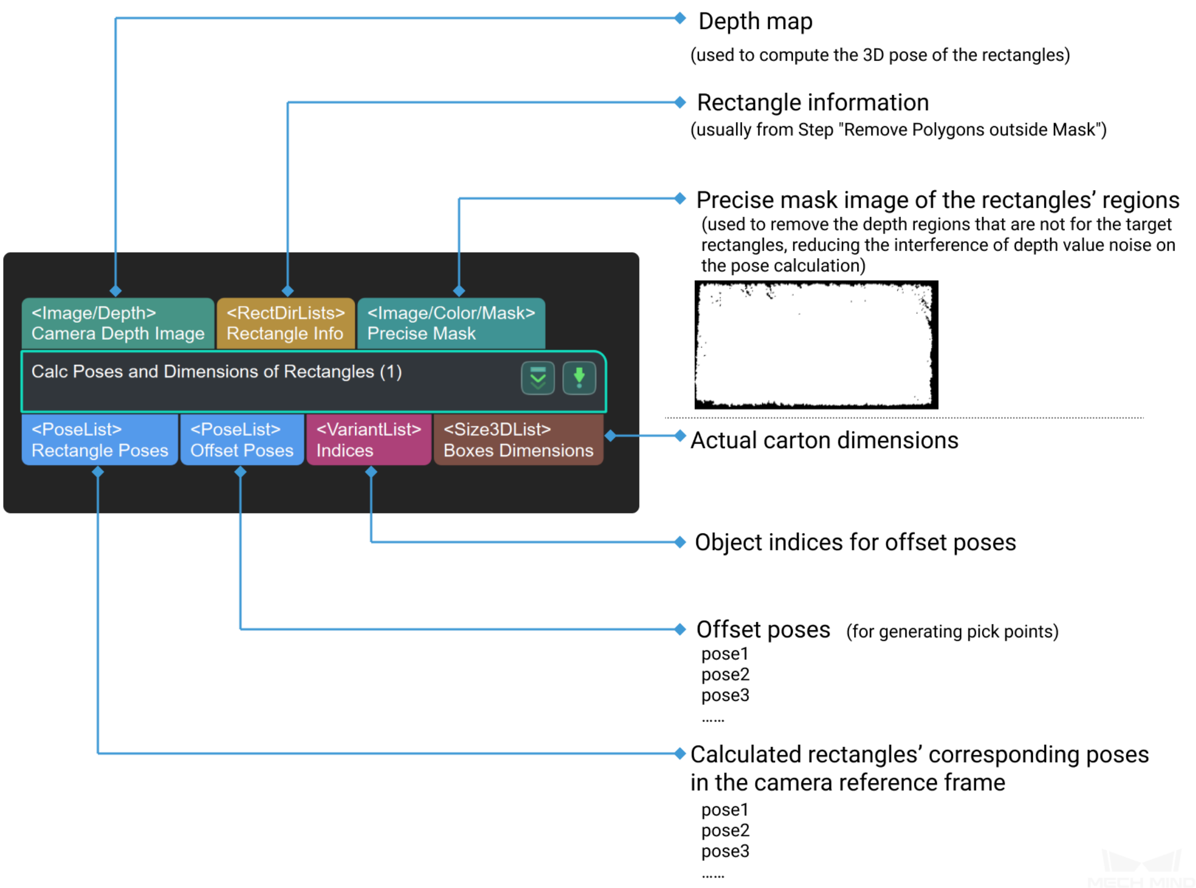
Function

Calculate the corresponding poses of the rectangles in 3D space by the depth map and rectangle information.

Sample Scenario

Fixedly used with the Step *Remove Overlapped Polygons*, to calculate the poses of the rectangular objects obtained by *2D Matching* in the camera reference frame.

Input and Output



Parameters

inputType

Instruction: This paramater is used to determine the type of input. | Default Value: Depth | Suggested Value: To set according to the real scenarios. | List of Values: Depth,VertexAndSize.

InputDepth

searchRadius

Instruction: This parameter is used to adjust the search radius of the depth of the given position in the image(in pixel). The parameter is only valid when *inputType* is 'Depth'.

Default Value: 20

Suggested Value: 20

Valid Range:[0,+∞)

suckerLengthX

Instruction: This parameter is used to adjust the size of the area covered by the sucker in X direction(in m). When the size of the sucker is smaller than the target object, the paramater should be decreased in order to avoid sucking the adjacent objects.

Default Value：0

Suggested Value：To set according to the real scenarios. When the size of the sucker is smaller than the target object, the paramater are suggested to be set to zero.

| Valid Range:[0,+∞)

suckerLengthY

Instruction：This parameter is used to adjust the size of the area covered by the sucker in Y direction(in m). When the size of the sucker is smaller than the target object, the paramater should be decreased in order to avoid sucking the adjacent objects.

Default Value：0

Suggested Value：To set according to the real scenarios. When the size of the sucker is smaller than the target object, the paramater are suggested to be set to zero.

| Valid Range:[0,+∞)

suckerMaskOtherRectRatio

Instruction: This parameter is the upper limit of the overlapping ratio of the sucker and adjacent boxes and decides whether to perform the sucking operation in order to avoid the condition that the adjacent boxes are also be sucked. When the overlapping ratio of the sucker and adjacent boxes is smaller than this value, the sucking operation would be performed.

Default Value: 0.3

Suggested Value: To set according to the real scenarios.

output strategy

outputOnePoseForOneBox

Instruction: This parameter decides whether to output single pose for single box.

Default Value: True

Suggested Value: True

List of Values: True, False

Calc Rect 2D Pose

4.6.3 Generate Template

Generate Rectangular Edge Templates of Specified Sizes

Function

Generate a list of rectangular edge templates given the dimensions of the objects (in pixels). Usually used before the Step *2D Matching*.

Sample Scenario

Fixedly used with the Step *From Actual Dimensions to Dimensions in Pixels* to generate edge templates for the Step *2D Matching*.

Input and Output

Parameters

templateDiagonalQuantization

Instruction：This parameter is used to set the diagonal quantization of generated template and equal to the length of rectangle template’s diagonal divided by diagonal quantization, thus getting the number of quantization pixels of edges.

Default Value：10

Suggested Value：[10,15]

Valid Range: [0,+∞)

Load 2D Templates

4.6.4 Remove Polygons

Remove Overlapped Polygons

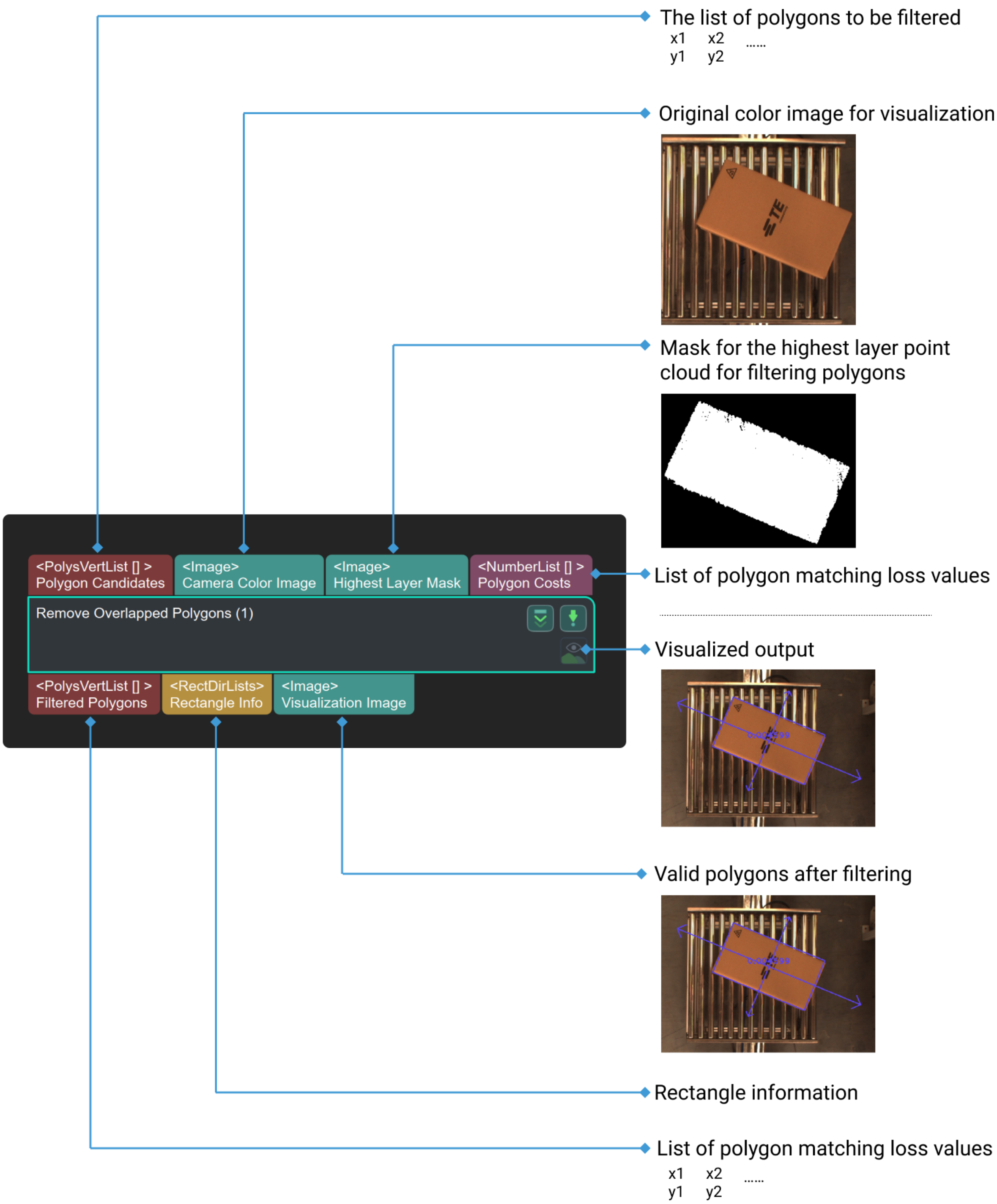
Function

Filter out invalid polygons that overlap or are outside the masks.

Sample Scenario

Fixedly used with the Steps *Remove Polygons outside Mask* and *Calc Poses and Dimensions of Rectangles* to filter out polygons that do not reflect the contours of real objects.

Input and Output



Parameters

Mask Settings

Usage of Input Masks

Default Value: SingleMask

List of Values: SingleMask (one inputted image corresponds to one mask image), CorrespondingMasks (all the inputted images corresponds to one mask image)

Instruction: To change the way the mask is used. The user can choose to apply one mask to all polygons, or apply a different mask to each polygon.

Inspection Criteria

Reached Out Ratio Min Threshold

Default Value: 0.45

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. Calculate the ratio of the area of the shifted mask beyond the highest layer of the mask to the area of the mask itself, 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.

Reached Out Ratio Max Threshold

Default Value: 0.6

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. Calculate the ratio of the area of the shifted mask beyond the highest layer of the mask to the area of the mask itself, 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.

Output Strategy

Position Restraint

Default Value: AtCorners

List of Values: AtCorners, None

Instruction: As shown in *Figure1*,the left side is the output image under the default setting, and the right side is the output image when set to None.

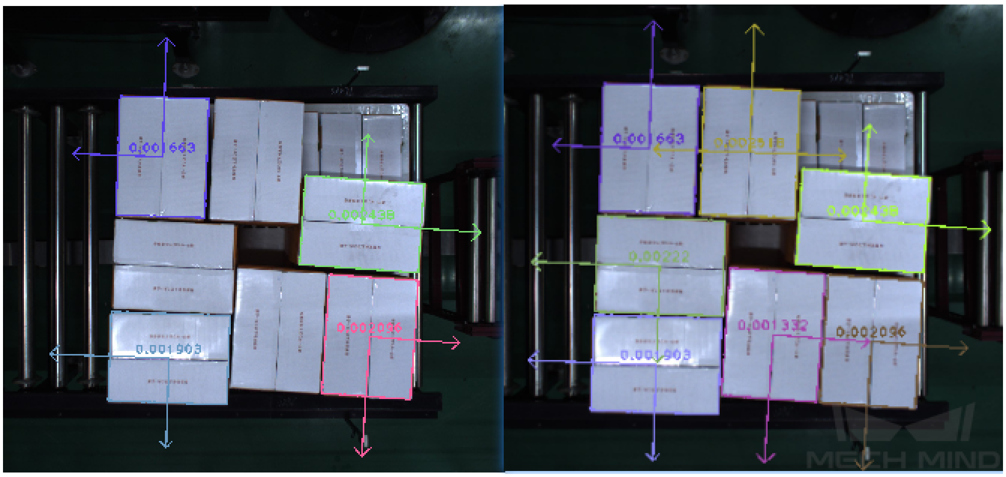


Figure1 Position Restraint

Remove Polygons outside Mask

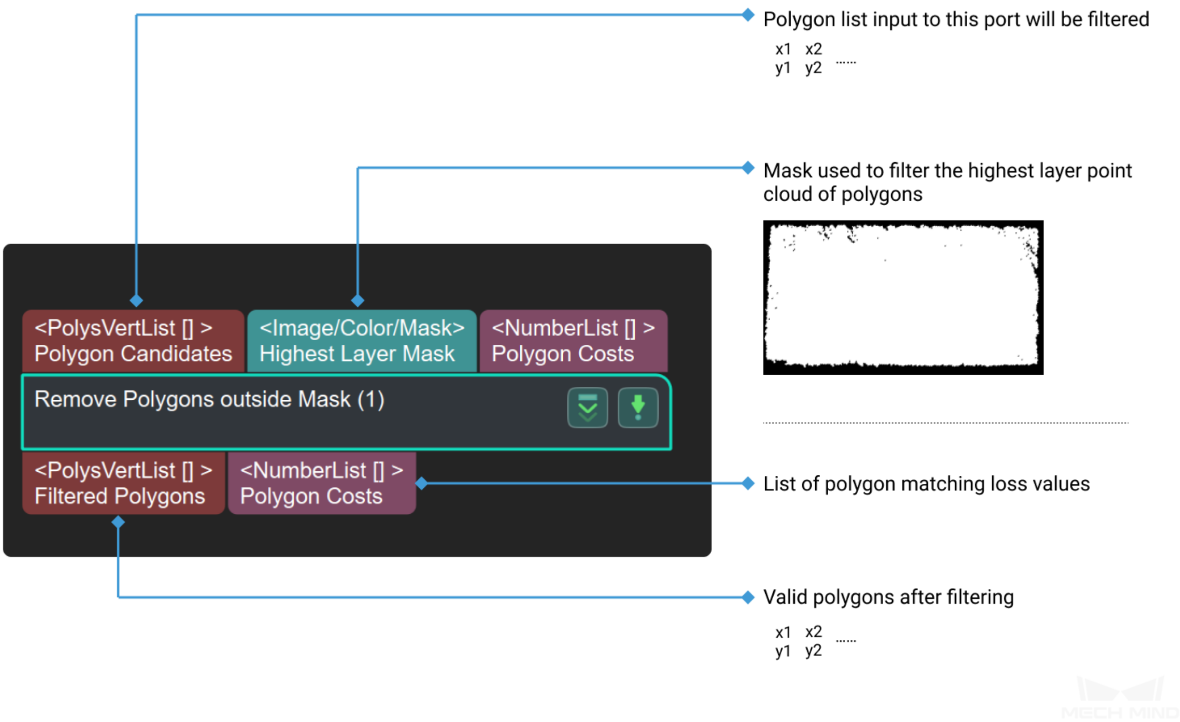
Function

Filter out interfering polygons whose overlapping area with the mask is smaller than the set threshold.

Sample Scenario

Fixed used with the Step *2D Matching* and the Step *Remove Overlapped Polygons* to filter out the interfering polygons from matching by the mask.

Input and Output



Parameters

Usage of Input Masks

Default Value: SingleMask

List of Values: SingleMask (one inputted image corresponds to one mask image), CorrespondingMasks (all the inputted images corresponds to one mask image)

Instruction: To change the way the mask is used. The user can choose to apply one mask to all polygons, or apply a different mask to each polygon.

Min Intersection Ratio(0~1.0)

Default Value: 0.8

Instruction: 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.

4.6.5 Others

Count Boxes

From Actual Dimensions to Dimensions in Pixels

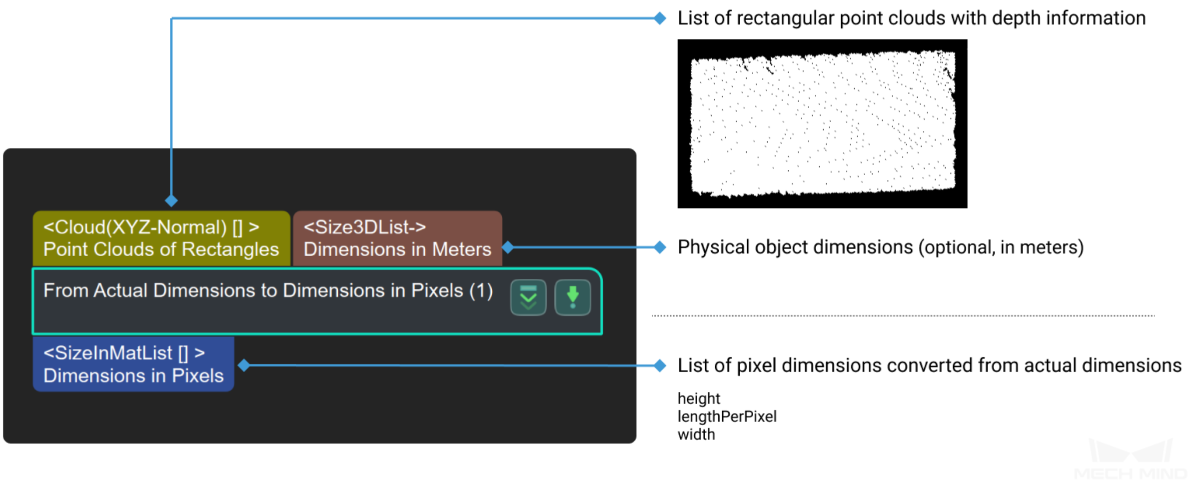
Function

Given the depth information of a rectangular point cloud and its physical dimensions, calculate the dimensions of the corresponding object in pixels.

Sample Scenario

Usually used with the Step *Generate Rectangular Edge Templates of Specified Sizes* to provide edge templates for subsequent 2D matching.

Input and Output



Parameters

fullConnection

Instruction: When this paramter is 'False', one of the real sizes corresponds to only one point cloud. When this paramter is 'True', one of the real sizes corresponds to several point clouds and every point cloud will match with all the inputted real sizes.

Default Value: False

Suggested Value: To set according to the real scenarios.

List of Values: True, False

inputType

Instruction: This parameter determines which type of input is used in this step.

Default Value: Depth

Suggested Value: To set according to the real scenarios.

List of Values: Depth,CloudNormal

searchRadius

Instruction: This parameter determines the radius of searching(in pixel) and is valid only when *inputType* is 'Depth'.

Default Value: 15

Suggested Value: To set according to the real scenarios.

Valid Range:$[0,+\infty)$

Generate Rectangle Candidates

4.6.6 Calc Pixel Size at Specified Height

4.7 2D Feature Detector

In this chapter, steps that are associated with 2D Feature Detector will be introduced.

4.7.1 Circle

Detect Circle Centers

Detect Inscribed Circles

4.7.2 Line

Detect Edges

Detect Line Segments

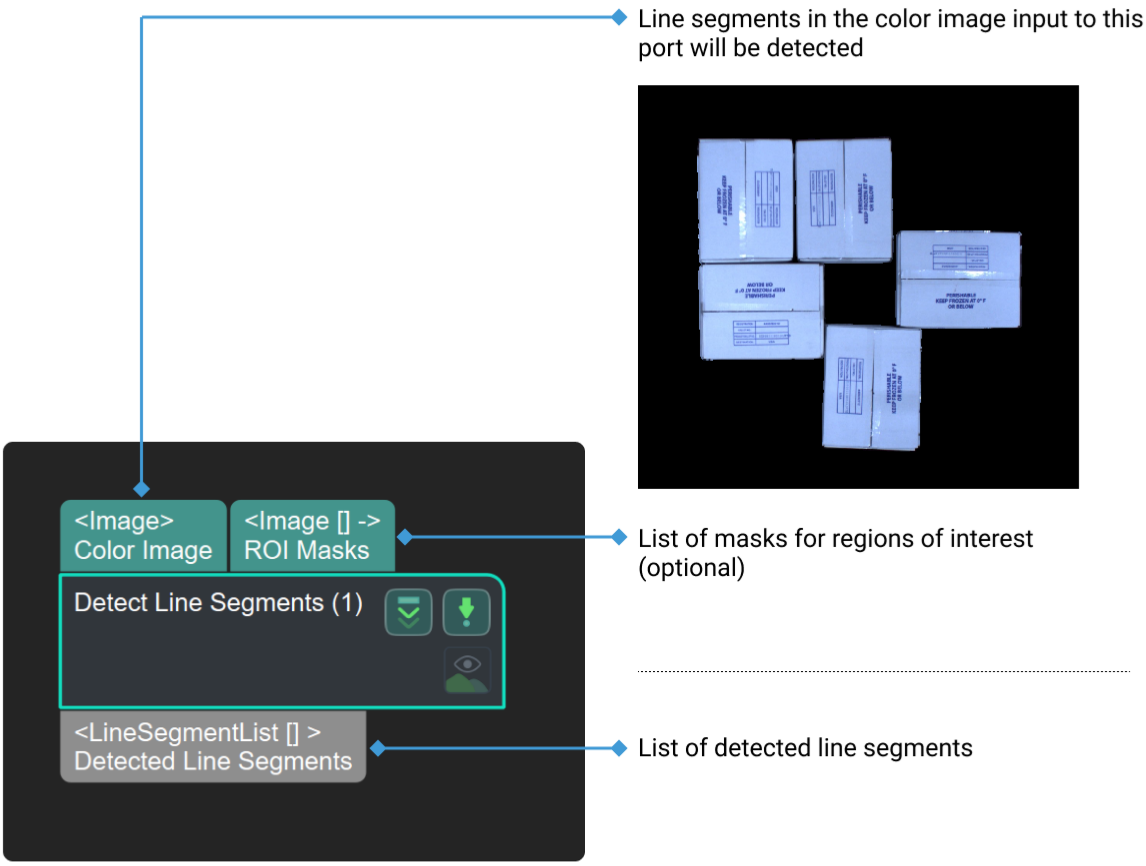
Function

Detect line segments in an image.

Sample Scenario

Extract the line segment features of objects in the 2D image for subsequent matching. This Step can be used with the Step *2D Matching*.

Input and Output



Parameters

LineProperty

gradientQuant

Instruction: This parameter is used to set the lower limit of the gradient of the pixels in the images. When the gradient of a pixel is smaller than the limit, the pixel will be discarded and no further process of line detection will be performed on this pixel.

Default Value: 2

Suggested Value: 1, 2

angleTolerance

Instruction: This parameter is the upper limit of the angle difference(in °) between the direction of the line(called level-line) that perpendicular to the direction of the pixel' s gradient and the main direction of minimal circumscribed rectangle of the area that a line may exist. When the angle difference is less than this value, the current pixel is considered to be a point belonging to the line. Otherwise, the number of detected lines will decrease while the accuracy of lines will increase.

Default Value：30

Suggested Value：30

minLineLength

Instruction: This parameter is the lower limit of the length of the detected lines(in pixel). If the detected lines were shorter than this limit, they will be discarded and the number of lines will decrease. Otherwise, shorter lines could also be detected so that there would be more detected lines.

Default Value：10

Suggested Value：10, 20, 30

filterLineByMask

Instruction: This parameter decides whether to filter out the detected lines outsides the given masks or not.

Default Value: False

Suggested Value: To set according to the actual scenarios.

List of Values: True, False

4.7.3 Others

Detect Corners

Detect Fiducial Markers

Detect N Largest Rectangles in Mask

Detect Rectangles Given Corners and Sizes in Pixels

Detect Shape Feature of Region

Find 2D Contour at Specified Hierarchical Level

Find Corners

Find Hole 2D

Read QR Code

4.8 2D General Processing

This section describes steps related to 2D general processing.

4.8.1 Color Image Processing

Convert Image Type

Image Brightness and Color Balancer

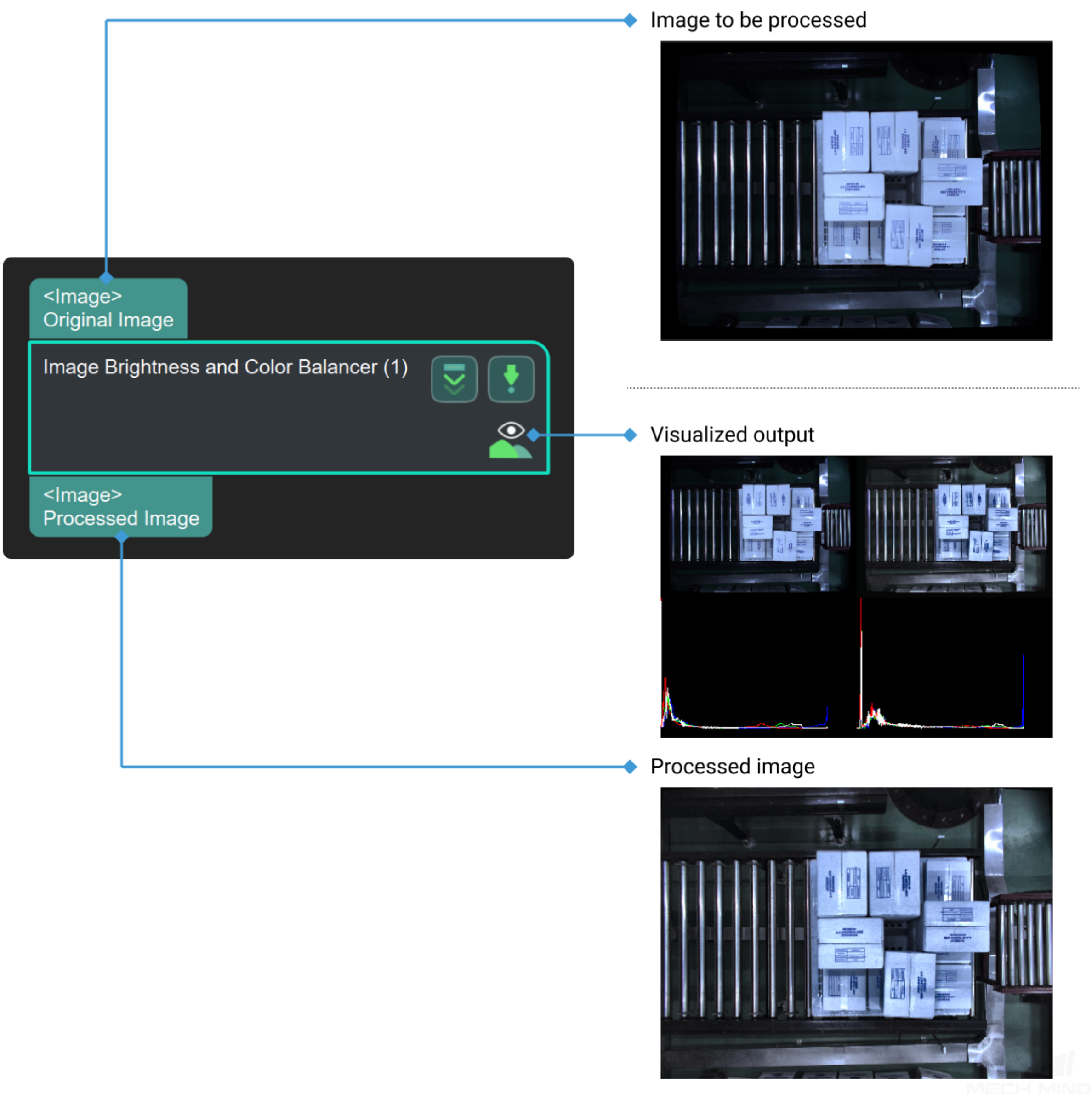
Function

This Step utilizes 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 as edge detection, identification, etc.

Sample Scenario

Please consider using this Step in scenes with large variations in light intensity. This Step 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

AdaptiveColorBalancer

Instruction: There is no parameters for this kind of balancer. The image balancing process is based on the RGB channels' values of the image.

Default Value: None

Suggested Value: None

CLAHE

Instruction: This balancer is used to adjust the contrast of the image and only one parameter named ClipLimit in this balancer. When the parameter increases, the degree of contrast decreases. Otherwise, the contrast increases, which means bright zones would be birghter and dark zones would be darker.

Default Value: 4

Suggested Value: To set according to the real scenarios.

ColorBalancer

Instruction: There are parameters in this balancer, 'Lightness', 'greenToRed' and 'blueToYellow' included.' Lightness' is used to adjust brightness. 'greenToRed' and 'blueToYellow' are used to adjust the dominant hue of the image. When the parameters increase, images would obviously tend to be red or yellow.

Default Value: Lightness=0;greenToRed=0;blueToYellow=0

Suggested Value: To set according to the real scenarios.

GammaCorrection

Instruction: This balancer is used to adjust the contrast of the image and only one parameter in this balancer. When the parameter increases, the degree of contrast decreases. Only few pixels would be displayed if this value is set to 0.

Default Value: 1

Suggested Value: To set according to the real scenarios.

Illumination Normalization

There are three kinds of methods and one common parameter included in this kind of balancer.

The method and the meanIllumination could be set in the branch named 'common setting'. When meanIllumination increases, image would be brighter, which is usually used to process images taken in low light situations.

common setting

meanillumination

Instruction: This parameter is used to adjust the mean value of illumination and valid for all the three following methods.

Default Value: 100

Suggested Value: To set according to the real scenarios.

method

Instruction: This parameter is used to determine which kind of method is performed in this balancer. Three methods are BgAdjust, Retinex_SSR and Retinex_MSR. | Default Value: BgAdjust | Suggested Value: To set according to the real scenarios.

BgAdjust Setting

Instruction: The method focuses on the processing of ROI(region of interrest) and there are four parameters. 'X' and 'Y' determine the starting coordinate of ROI. 'Width' and 'Height' determine the size(width and height) of the ROI. | Default Value:X=30,Y=700,Width=200, Height=200 | Suggested Value: To set according to the real scenarios.

Retinex SSR setting

Instruction: ‘KernelSize’ represent the size of Gauss kernel and only can be odd number. The edges of objects in the image would be more obvious when this value increases.

Default Value:kernelSize=21

Suggested Value: To set according to the real scenarios.

Retinex MSR setting

Instruction:This method combines three kinds of Gauss kernel with different kernelsize to process the given image. Adjusting sizes of three kernels could bring images with different edge effects. Usually, Retinex MSR method could obtain more egde information than Retinex SSR.

Default Value:kernelSizeSmall=15;kernelSizeMedium=81;kernelSizeLarge=201

Suggested Value:To set according to the real scenarios.

Image Filtering

Description

This step could perform filtering process on the input images with different kinds of filters.

Parameters of Input and Output

Image/Color [Input] original images

Image/Color [Output]filtered images

Parameters

BilateralFilter

Instruction：There are three parameters in this filter. The diameter determines the range of filtering area. When diameter increases, more original image information would be retained, so that the image is still clear. Otherwise, smaller diameter would blur the image. The paramter ‘sigmaColor’ represents the differnce of pixel value between pixels and ‘sigmaSpace’ represents the difference of spatial distance between pixels. These two values affect the filter result jointly.

Default Value：diameter=3; sigmaColor=3;sigmaSpace=3.

Suggested Value：To set according to the real scenarios.

Boxfilter

Instruction：The only parameter is the kernel size during the filtering process and its unit is pixel. When the parameter is increased, the image is blurred.

Default Value：kernelSize=100.

Suggested Value：To set according to the real scenarios.

DfFilter

Instruction: The full name of the filter is Discrete Fourier Filter and it mainly used to filter out the signals with high frequencies, noise and edges for example. This filter is capable of noise reduction and edge enhancement.

Default Value：None

Suggested Value：None

GaussianFilter

Instruction: This kind of filter processes the images with the Gauss functions. There are four parameters here: sizeX and sizeY, referring to the size of the Gauss kernel and when they increase, the image would be blurred; sigmaX and sigmaY, referring to the variances in the Gauss function and the increase of them would also blur the image.

Default Value: sizeX=3,SizeY=3;sigmaX=3,sigmaY=3.

Suggested Value: To set according to the real scenarios.

HaarFilter

Instruction：Smooth noise reduction processing, which can filter out part of noise.

Default Value：None

Suggested Value：None

4.8.2 Image Transform

Image Transform

Perspective Transform

Rotate Images By Provided Poses

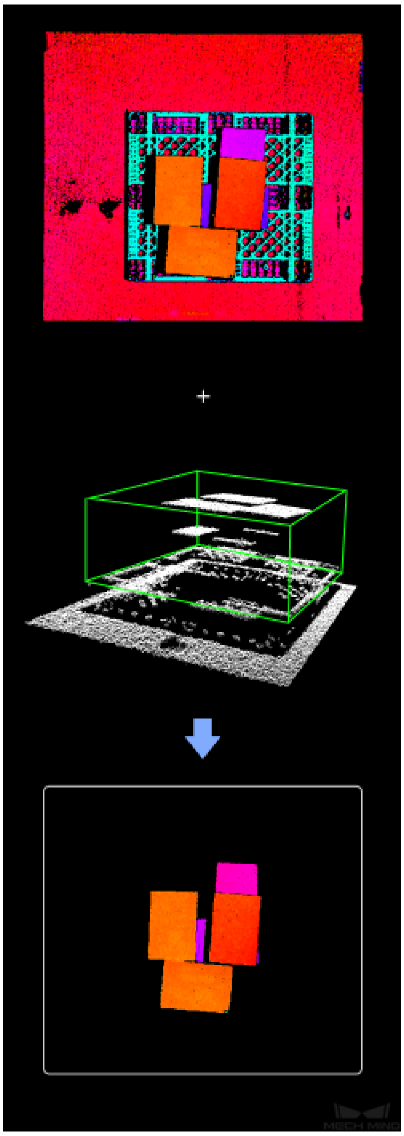
4.8.3 ROI

Extract Empty Areas in Depth Image Within 3D ROI

Invalidate Depth Pixels Outside 3D ROI

Function

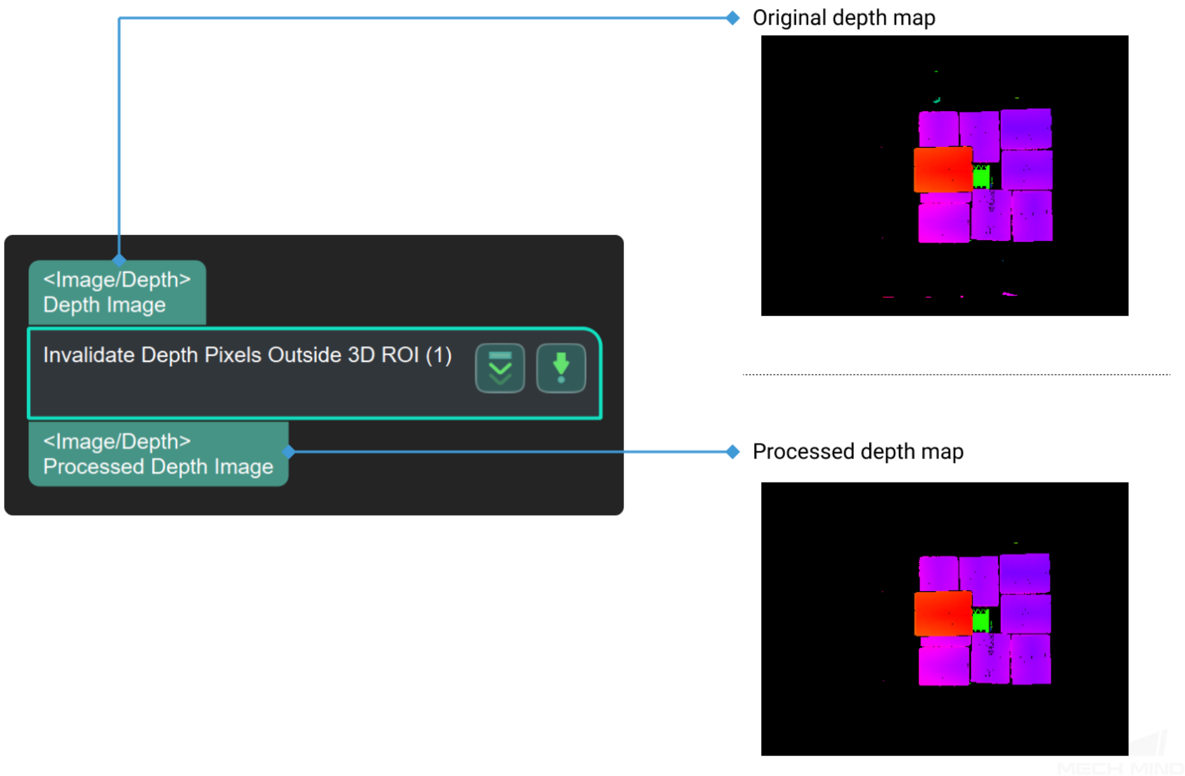
Set a Region of Interest (ROI) in the depth image and set the pixel values outside the region to zero.



Sample Scenario

This Step can be used to extract a target area in the original depth image and therefore avoid interferences from background and other regions. It is usually connected with Steps as *Segment Depth Image*, *Get Highest Areas in Depth Image*, *Merge Mask Images*, etc.

Input and Output



Preparation

Before setting a 3D ROI, please go to **Project Assistant** and select the data source of scene point cloud. For detailed settings, please refer to *Scene Point Cloud*.

Parameters

3D ROI Settings

3D ROI Name Instruction: Click  on the right side of the 3D ROI Name to enter the **Set ROI** window, as shown in *Figure1*. For detailed settings, please refer to *Instructions for Setting 3D ROI*.

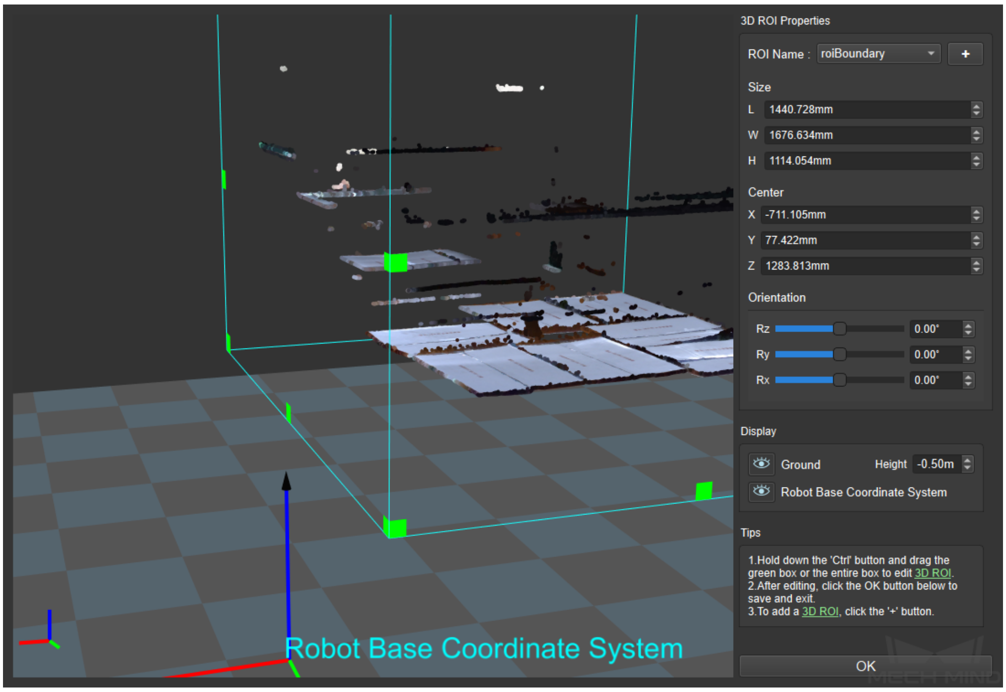


Figure1 Set Target Area

Afetr intercepting the depth ROI, the pixel values outside the region of the depth image' s interest will be set to zero. *Figure2* is the input image, and *Figure3* is the output image.

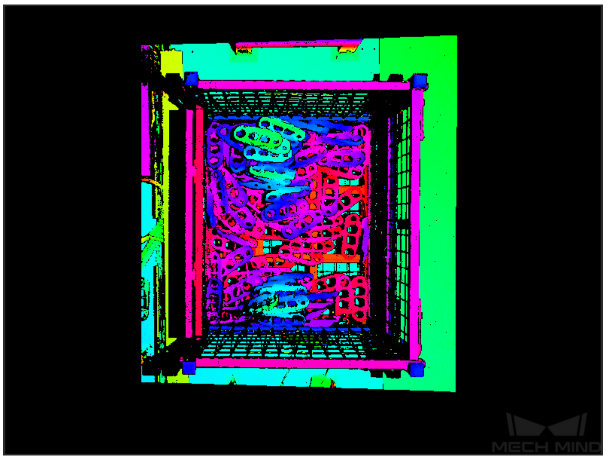


Figure2 Input Image

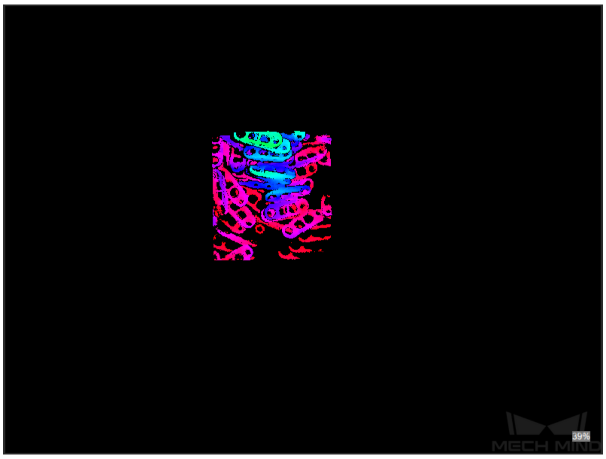


Figure3 Output Image

Threshold Setting

Min Points in ROI

Instruction: 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 in *Figure 4*. The actual information engineering can set the threshold for the point cloud threshold.

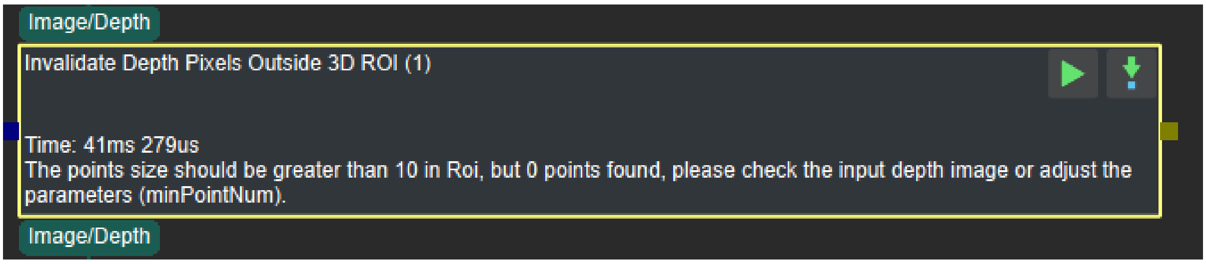


Figure4 Warning Message

4.8.4 Segment Area

Binarize Image

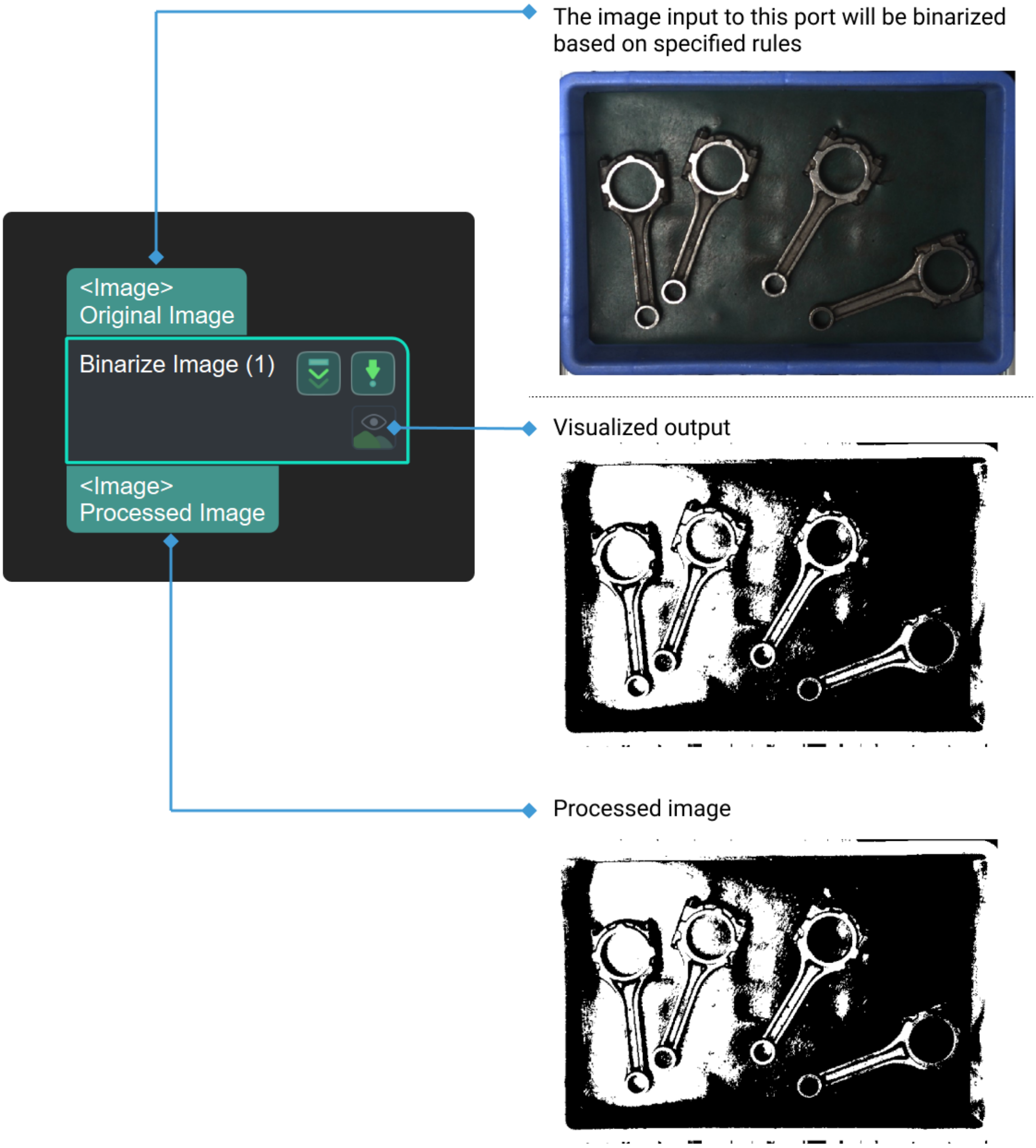
Function

Filter image pixels according to the set threshold, and binarize the pixels above and below the threshold according to the set rules.

Sample Scenario

A general image processing Step. Generally used to segment feature objects and backgrounds on 2D images.

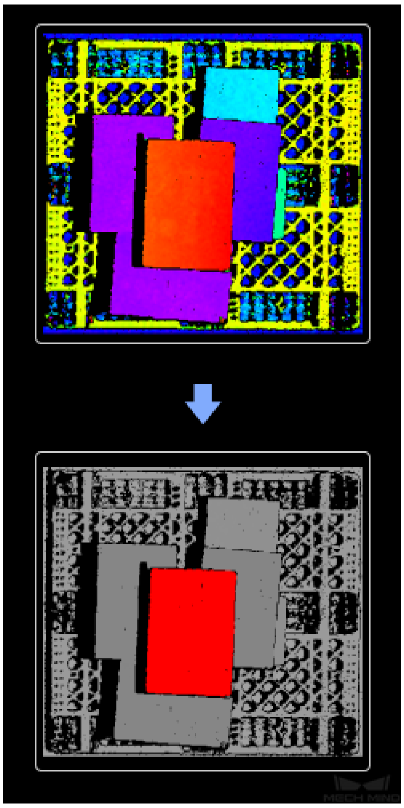
Input and Output



Get Highest Areas in Depth Image

Function

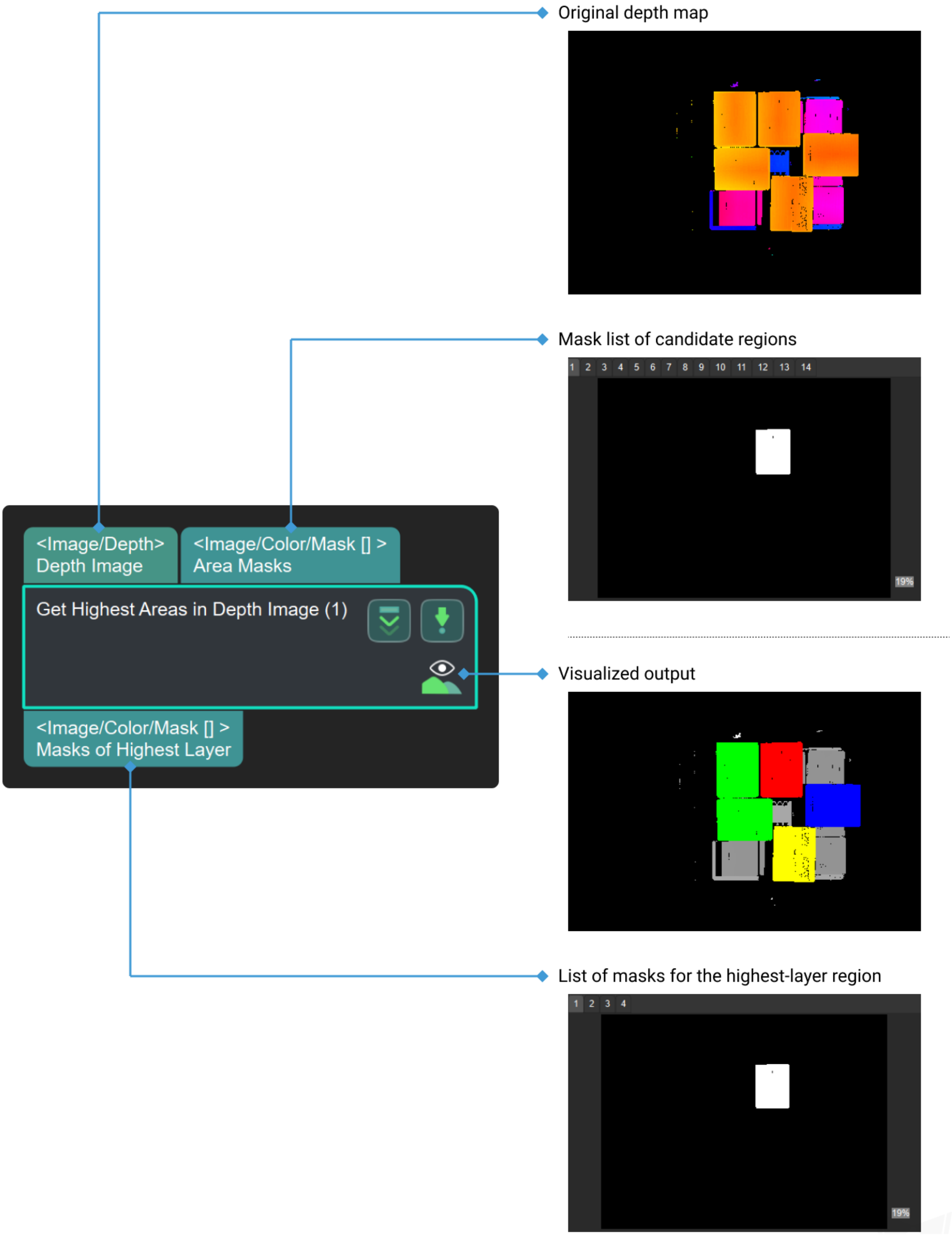
This Step utilizes the given depth map and mask list to segment the image into multiple areas, and output the area that is closest to the camera, while other areas will be discarded.



Sample Scenario

This Step is usually used in carton depalletizing applications to identify the objects on the highest layer. It is usually connected with Steps as *Invalidate Depth Pixels Outside 3D ROI*, *Merge Mask Images*, *Segment Depth Image*, etc.

Input and Output



Parameters

LayerHeight

Default Value: 0.1

Instruction: The height of the top.

LayerDiffTolerance

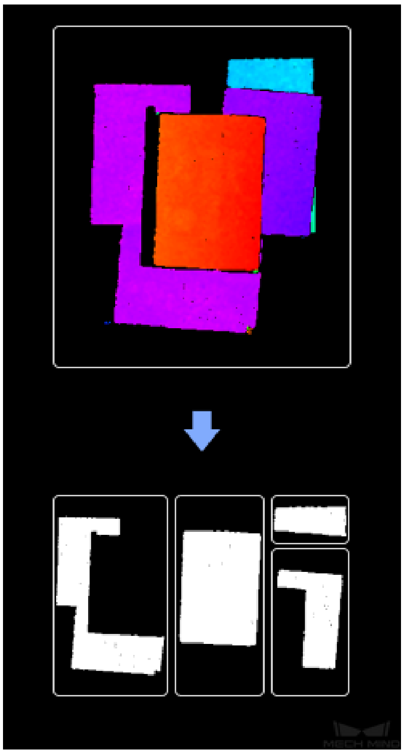
Default Value: 0.5

Instruction: This parameter is the upper limit of the ratio of height difference between the different areas with different height to **layerHeight** . The areas whose difference of height are less than this paramter are believed to have the same height. Otherwise, areas are considered to have different height when this value is small.

Segment Depth Image

Function

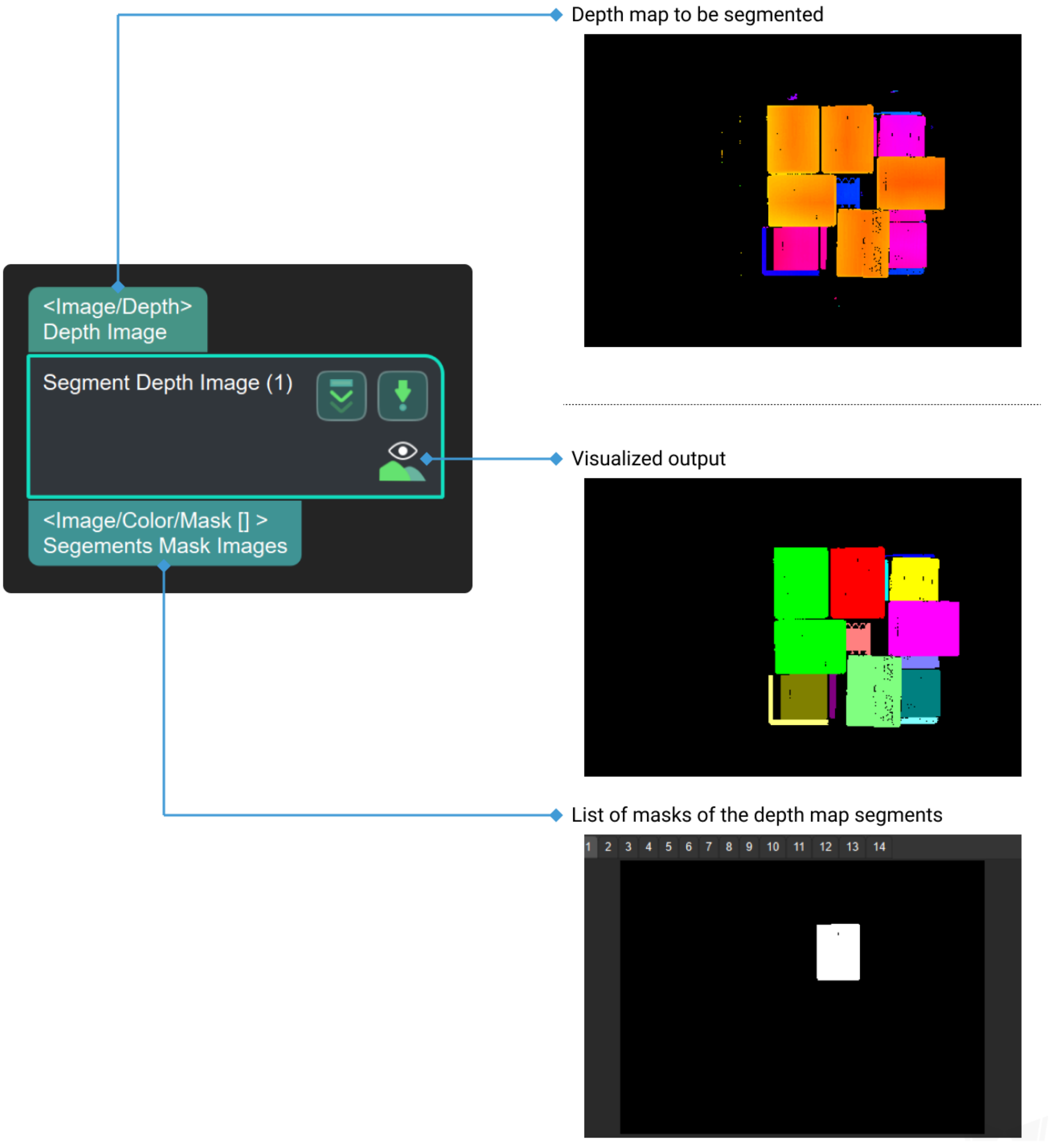
This Step is used to segment depth image according to specified rules and output areas that meet the rules.



Sample Scenario

This Step is widely used for depth image processing, and is usually connected with Steps as *Invalidate Depth Pixels Outside 3D ROI*, *Get Highest Areas in Depth Image*, *Merge Mask Images*, etc.

Input and Output



Parameters

floodThreshold

Instruction: This parameter is used to determine the tolerance of depth difference(in mm) in depth segmentation process. When the value increases, the adjacent areas with obvious depth difference will be considered to have the same depth. Otherwise, these areas will be separated into several groups with different depths.

Default Value: 7

Suggested Value: To set according to the real scenarios.

floodMinArea

Instruction: This parameter is the lower limit of the size of the segmentation results (in pixel). When the number of pixel in the result is less than the limit, the result will be discarded.

Default Value: 1500

Suggested Value: To set according to the real scenarios.

depthRoiPath

Instruction: When the path is empty, this step will process the entire image. When the path of ROI(region of interest) exists, the step will only perform segmentation process on ROI. Once the path changes, this step would reload the file.

Default Value: depth_roi.json

Suggested Value: The absolute path or the relative path of the ROI file.

4.8.5 Others

Place Polygons

4.8.6 Determine Pixel Size

4.8.7 OCR

4.8.8 Read Bar Code

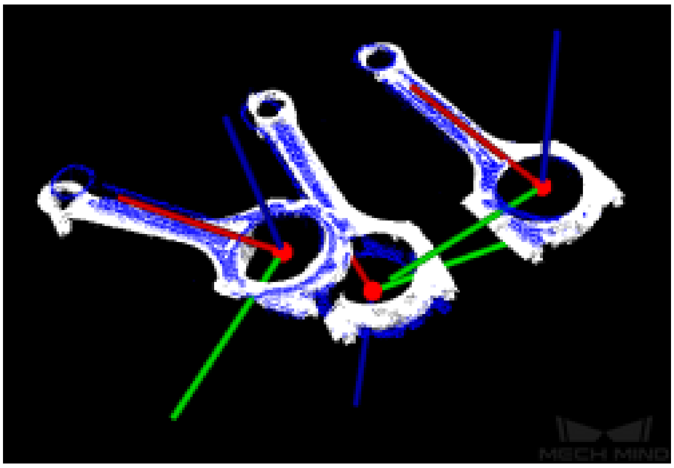
4.9 3D Matching

In this section, steps that are associated with 3D Matching will be introduced.

4.9.1 3D Coarse Matching

Function

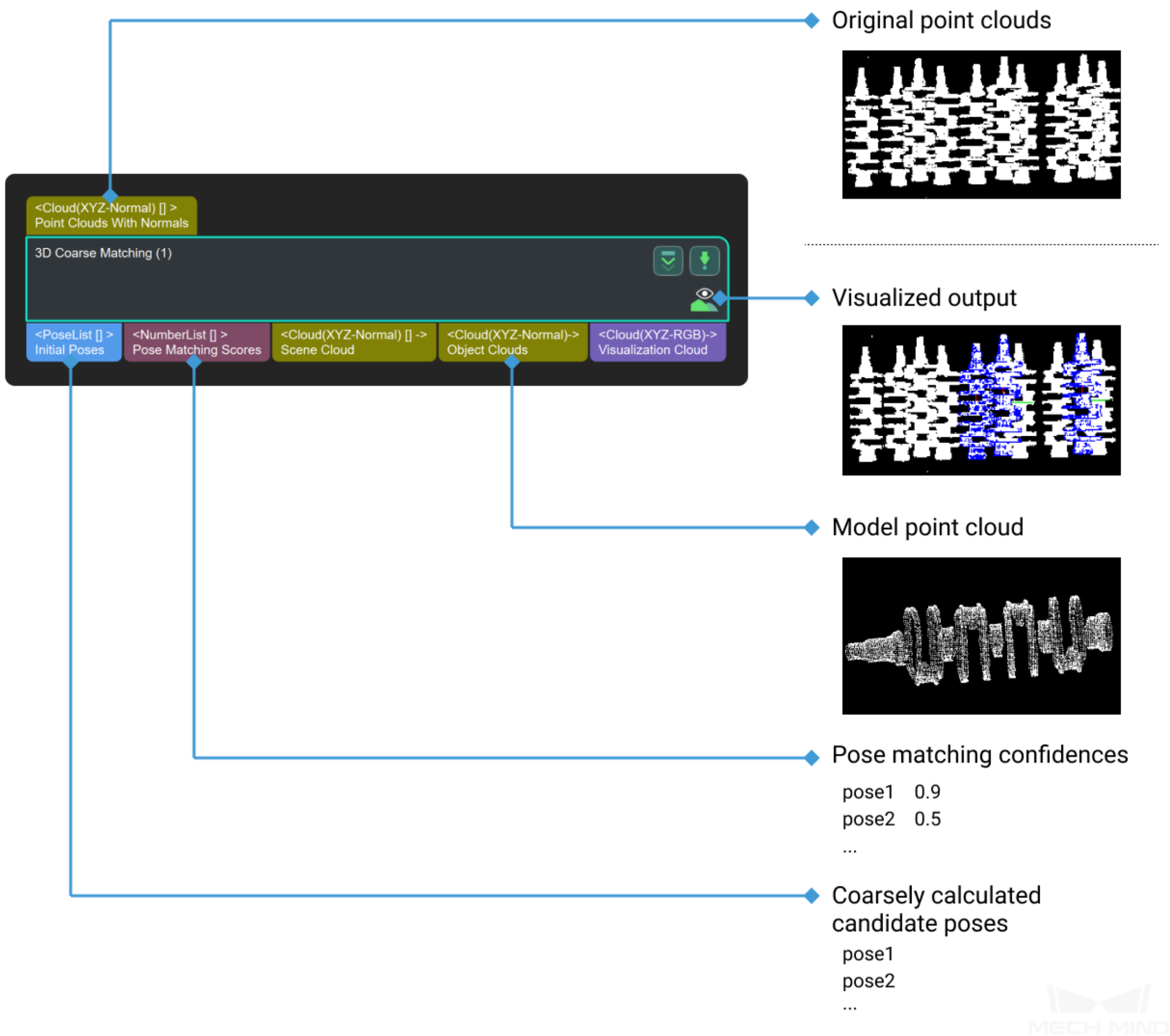
Roughly match the model point cloud with the original point cloud, and output approximate candidate poses of the target candidates.



Sample 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 connected with the Step *3D Fine Matching*.

Input and Output



Parameters

Model and Pick Point

The path of model file and pick point file.

ModelFile

Default Value: model.ply

Instruction: The path of 3D model file, the construction process can refer to the complete model point cloud stitching document. **If the file path or name is changed, the file will be read again when the next time this step is used.** Both full path and relative path of the model file location can be identified.

Geometric Center File

Instruction: The pick point file is based on the 3D model file. The pick point file generation method can refer to *Add Pick Point*. **If the file path or name is changed, the file will be read again when the next time**

this step is used. Both full path and relative path of the pick point file location can be identified.

Cloud Orientation Calculation

Point Orientation Calc Mode

Default Value: Origin

List of Values: Origin, StandardMode, EdgeTangent, EdgeNormal

Instruction: There are four methods for calculating point direction in point cloud, which are selected according to the actual situation of the project:

- **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 point 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.

Note: When using the **EdgeTangent** or **EdgeNormal** method, 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: There are two types of this algorithm. The **Resultant Visualisation** is set to parameters that can be adjusted for both algorithms. The algorithm type parameters are adjusted using *Figure 1*as an example of an input point cloud, 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 Sttings** and **Pose Verification Settings**

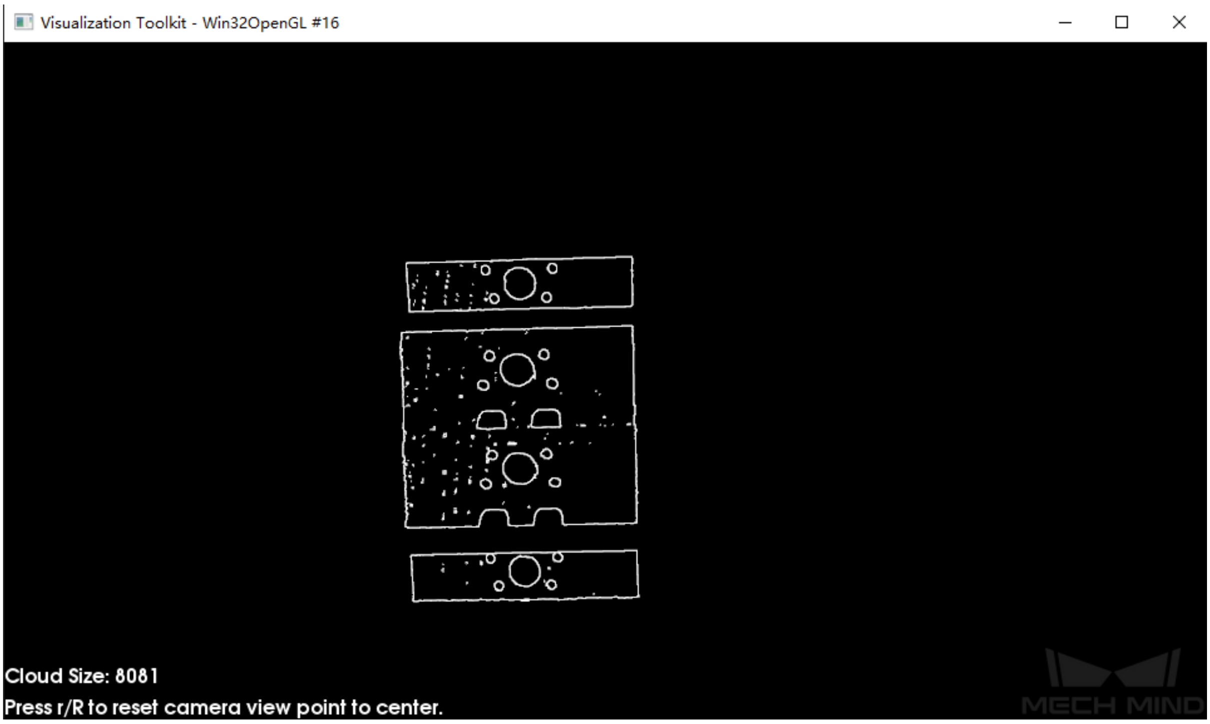


Figure 1 Example of Input Point Cloud

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** .

Example: As shown in *Figure 2* . The left figure shows the result when this value is `2` , and the right figure shows the result when this value is `5` . It is obvious that the matching accuracy decreases after adjustment.

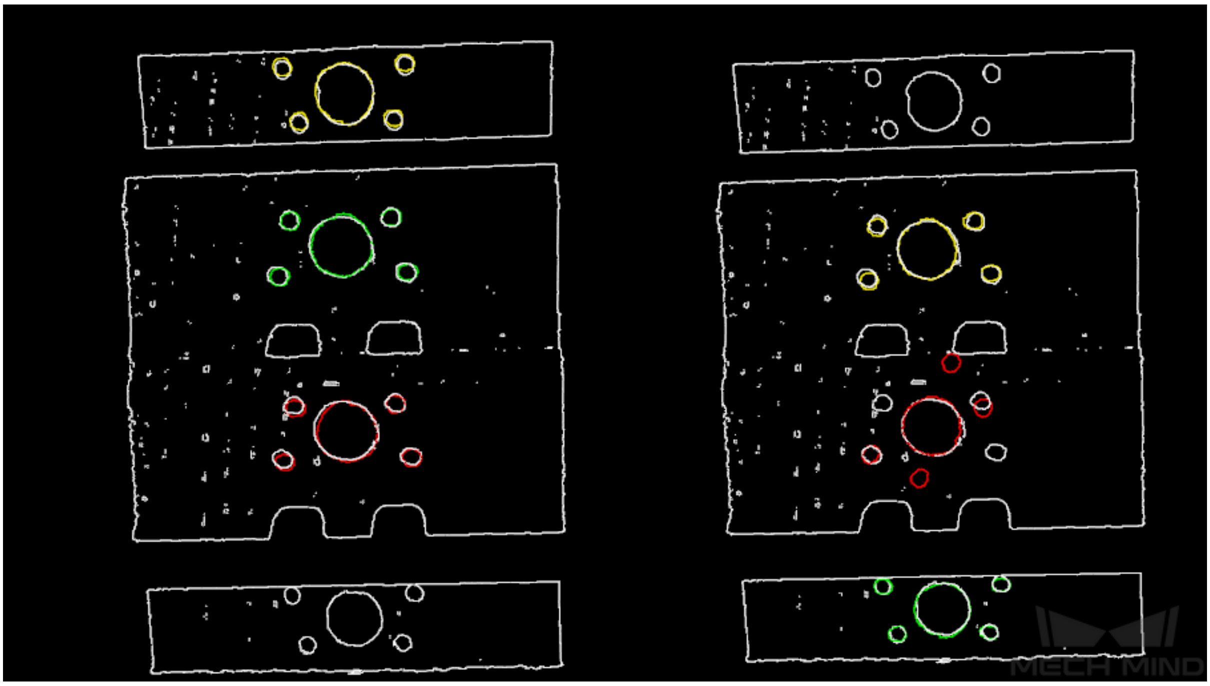


Figure 2 Comparison of Adjustment Results of Main Speed Controller

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**.

Example: As shown in *Figure 3*. The figure on the left shows the result when this value is `10` , and the figure on the right shows the result when this values 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.

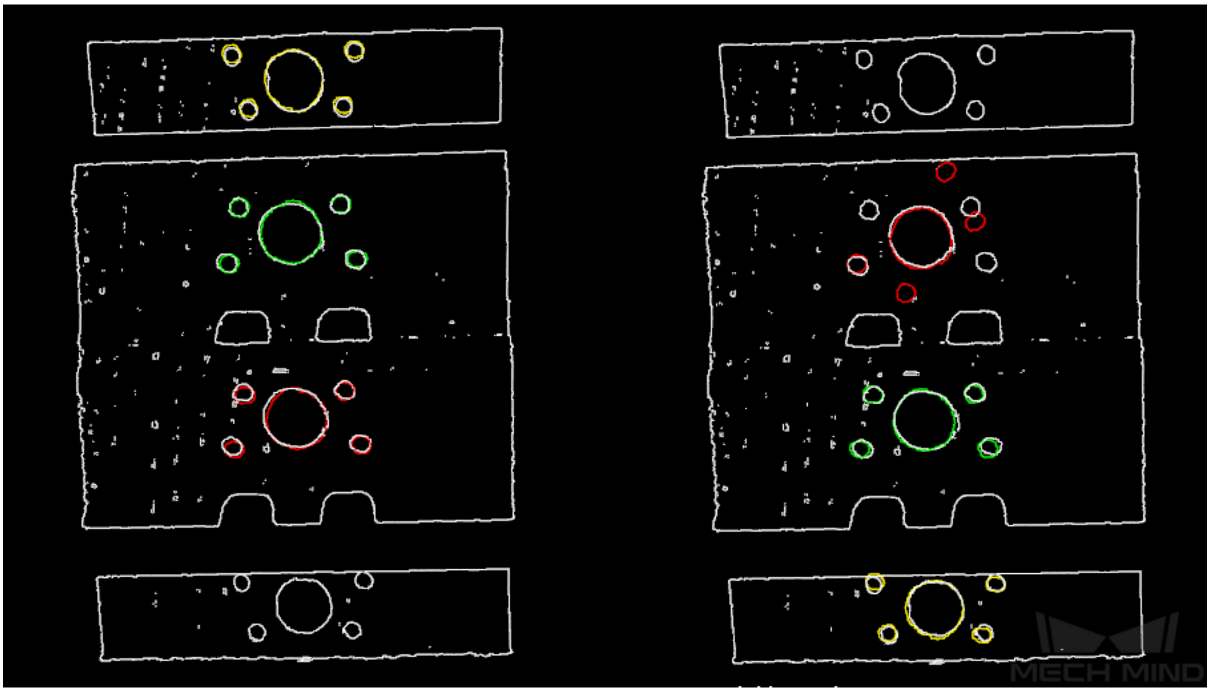


Figure 3 Comparison of Adjustment Results of Secondary Speed Controller

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 *Figure 4*. The left picture shows the result when the parameter is `3`, and the right picture shows the result when the parameter is `10`.

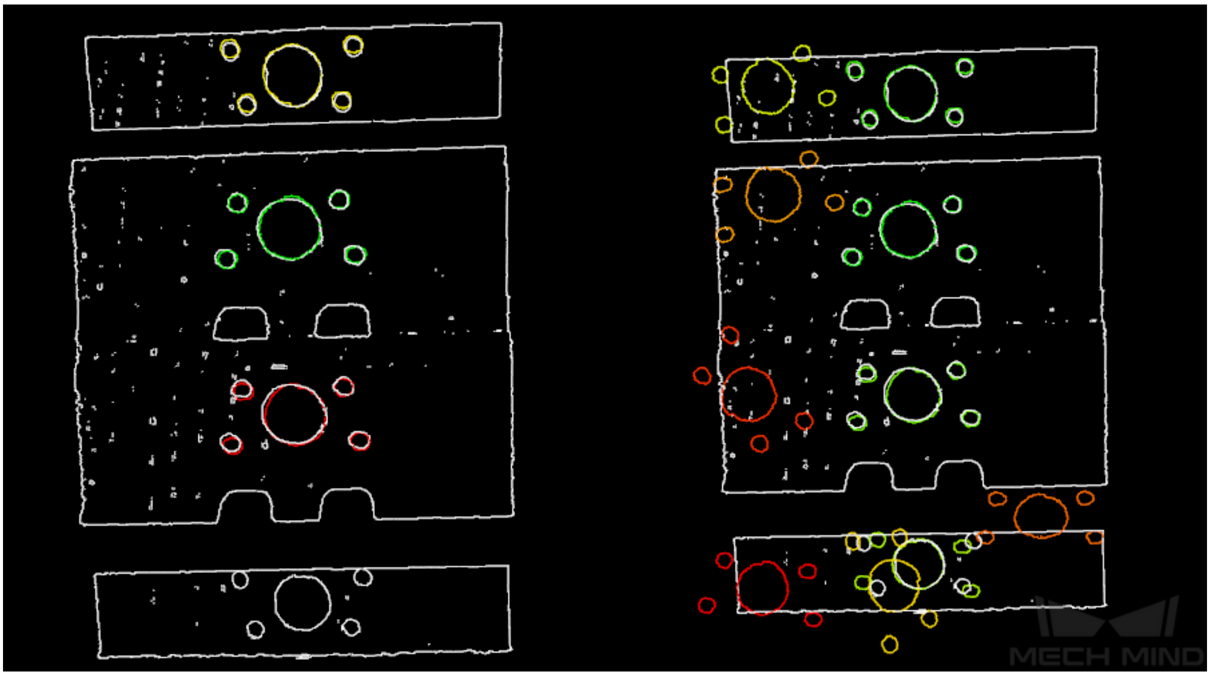


Figure 4 Comparison of Parameter Adjustment Results of Maximum Number of Detected Poses in Each Point Cloud

---

The parameters of **SurfaceMatching** are described below.

Sample Settings

Enable Automatic Downsampling

Default Value: True

List of Values: True, False

Instruction: This parameter is used to determine whether to use automatic downsampling. If set to `True`, the sampling interval parameter of model point cloud 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 True, 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 model point cloud after downsampling. It sets an upper limit for the number of points in the model point cloud after downsampling. If the matching effect is not ideal, it is recommended to increase the value of this parameter. If the project has a high requirement on the matching speed, it is recommended to decrease the value of this parameter.

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 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 can not 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: 0.01

Instruction: This parameter is used to adjust the maximum distance between points in the sampling point cloud. When the sampling interval of model point cloud 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 *Figure 5*. 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`.

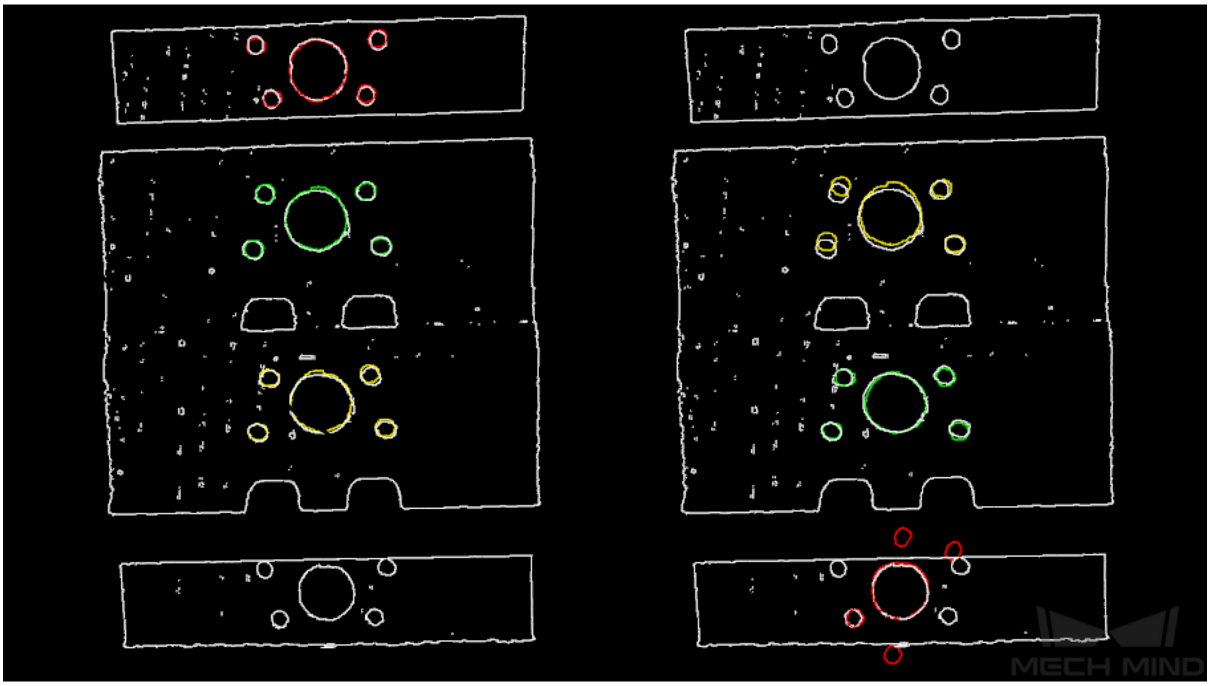


Figure 5 Comparison of Sample Interval Parameter Adjustment Results of Model Point Cloud

Min Sampling Interval

Default Value: 0.003

Instruction: This parameter is used to calculate the sampling interval. It is effective when the value of **Enable Automatic Downsampling** is True. 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 (Distance Interval = **Distance Quantification** * **Sampling Interval**). 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 point's normals (Angle Interval = 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.

Reference Point Step

Default Value: 5

Instruction: This parameter is used to sample points in the point cloud to make a point pairs, from the point cloud with the step size as an interval sampling point. 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 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.

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 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.

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: True

List of Values: True, Flase

Instruction: This parameter determines whether pose validation is used. If `True` is checked, 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

Default Value: 3

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

Example of adjustment: The left side of *Figure 6* is the result when the parameter value is `3`, and the right side is the result when the parameter value is `5`.

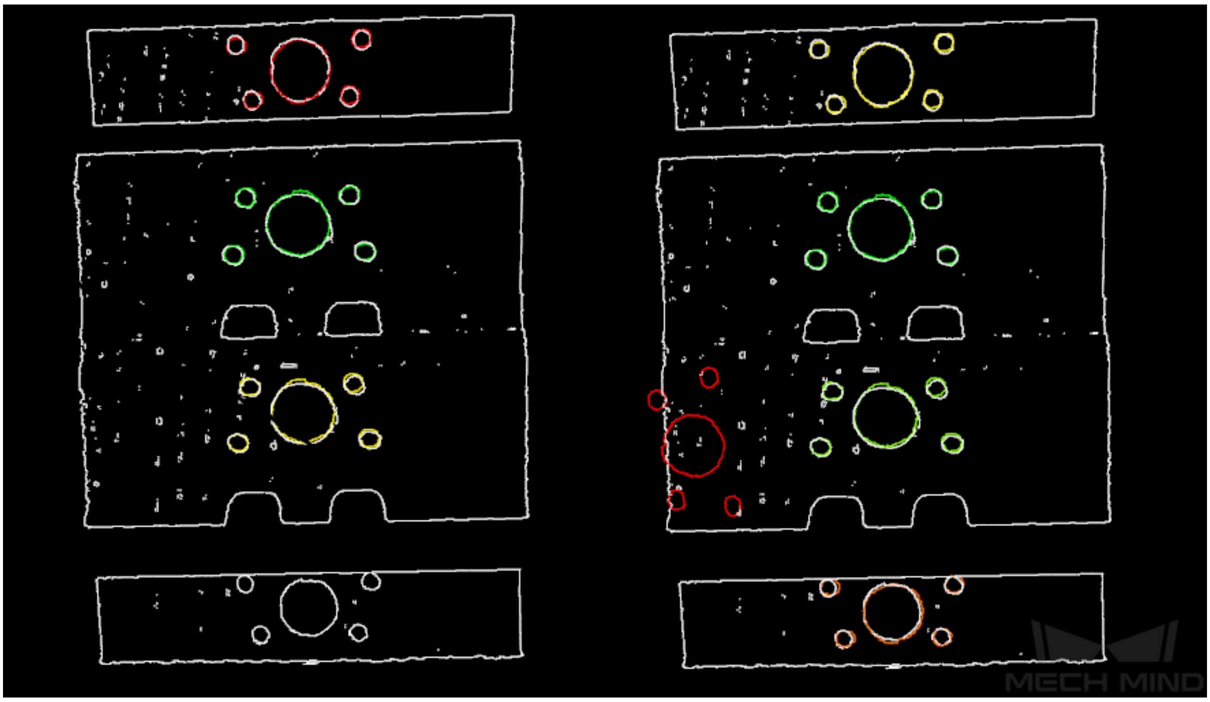


Figure 6 Comparison of Maximum Number Parameter Adjustment Results

Results Visualization

Show Sampled Model Cioud

Default Value: False

List of Values: True, Flase

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

Show Sampled Scene Cioud

Default Value: False

List of Values: True, Flase

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

Show Matching Results

Default Value: True

List of Values: True, Flase

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

4.9.2 3D Coarse Matching (Multiple models)

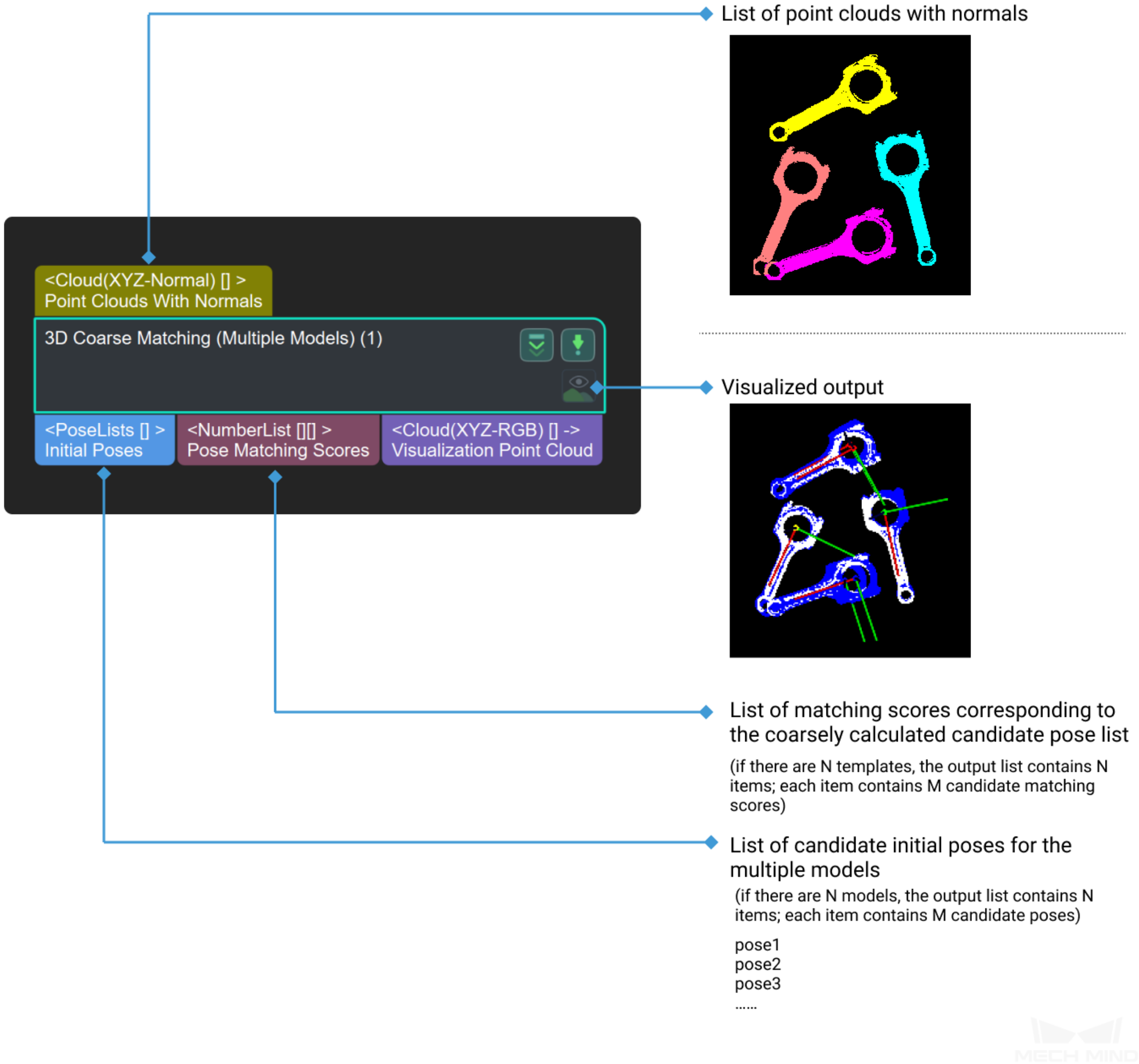
Function

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

Sample Scenario

This Step is an extended version of *3D Coarse Matching* and is used in multi-model scenarios. This Step is usually followed by *3D Fine Matching (Multiple Models)* to obtain accurate poses.

Input and Output



Parameters

Model and Pick Point

The path of model file and pick point file.

ModelFile

Default Value: model.ply

Instruction: The path of 3D model file, the construction process can refer to the complete model point cloud stitching document. **If the file path or name is changed, the file will be read again when the next time this step is used.** The full path or relative path of the model file location can be identified.

Geometric Center File

Instruction: The pick point file is based on the 3D model file. The pick point file generation method can refer to *Add Pick Point*. **If the file path or name is changed, the file will be read again when the next time this step is used.** The full path or relative path of the pick point file location can be identified.

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 *Figure 1*. Different files are separated by ;.

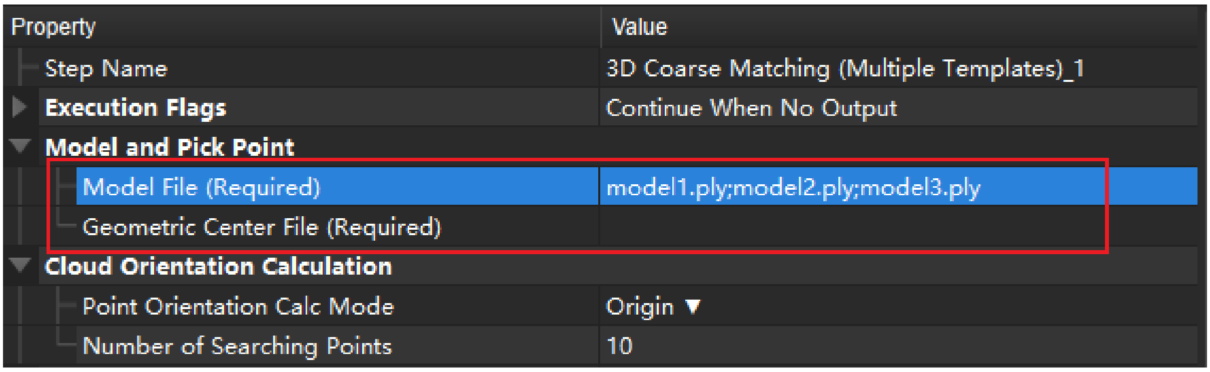


Figure 1 Example of Model and Pick Point

Cloud Orientation Calculation

Point Orientation Calc Mode

Default Value: Origin

List of Values: Origin, StandardMode, EdgeTangent, EdgeNormal

Instruction: There are four methods for calculating point direction in point cloud, which are selected according to the actual situation of the project:

- **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 point 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.

Note: 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: There are two types of this algorithm. The **Resultant Visualisation** is set to parameters that can be adjusted for both algorithms. The algorithm type parameters are adjusted using *Figure 1*as an example of an input point cloud, 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 Sttings** and **Pose Verification Settings**

Figure 2 Example of Input Point Cloud

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** .

Example of adjustment：As shown in *Figure 3*. The left figure shows the result when this value is 2 , and the right figure shows the result when this value is 5 . It is obvious that the matching accuracy decreases after adjustment.

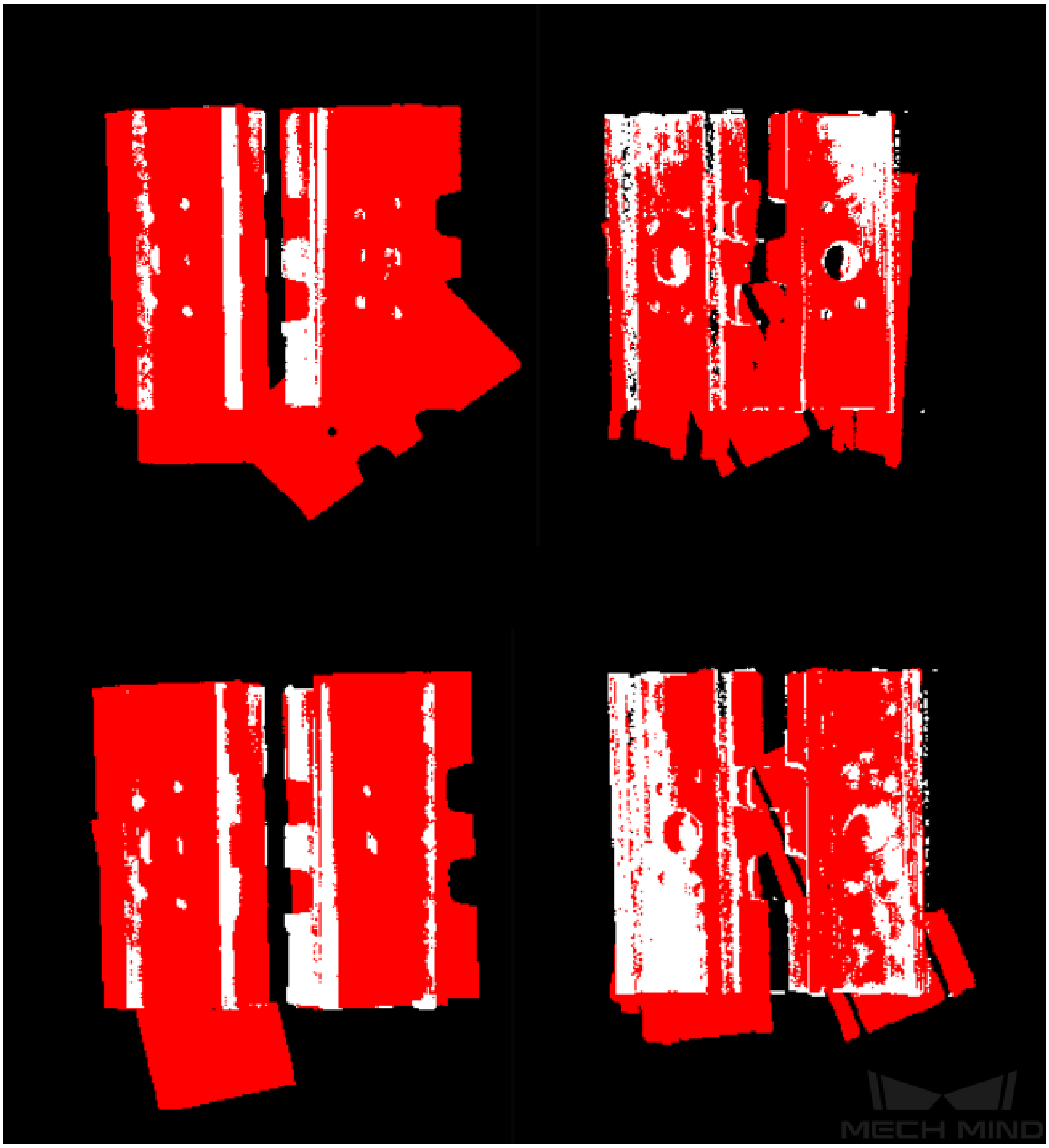


Figure 3 Comparison of Adjustment Results of Main Speed Controller

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**.

Example of adjustment：As shown in the *Figure 4*. 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.

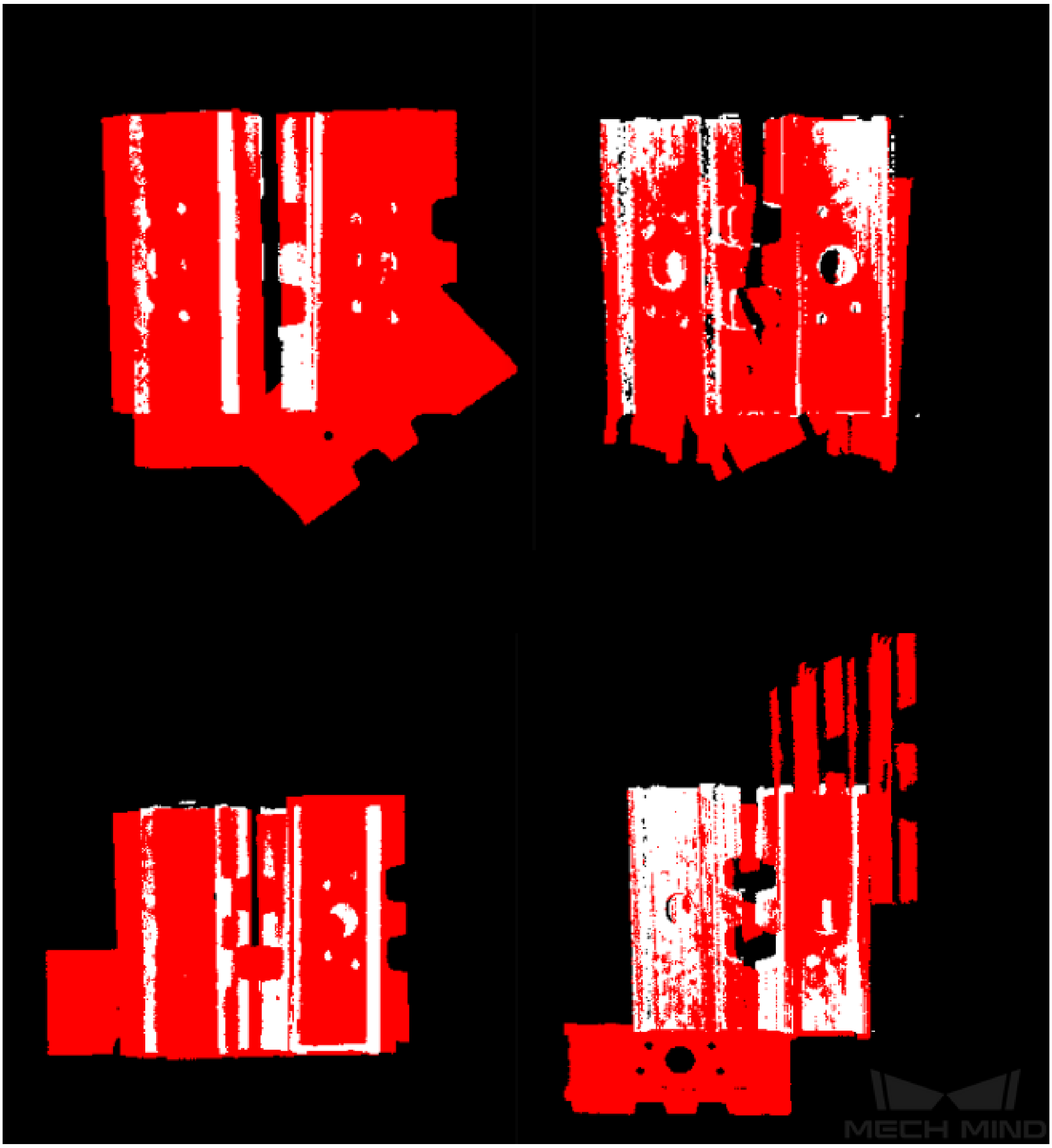


Figure 4 Comparison of Adjustment Results of Secondary Speed Controller

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 *Figure 4*. The left picture shows the result when the parameter is `3`, and the right picture shows the result when the parameter is `10`.

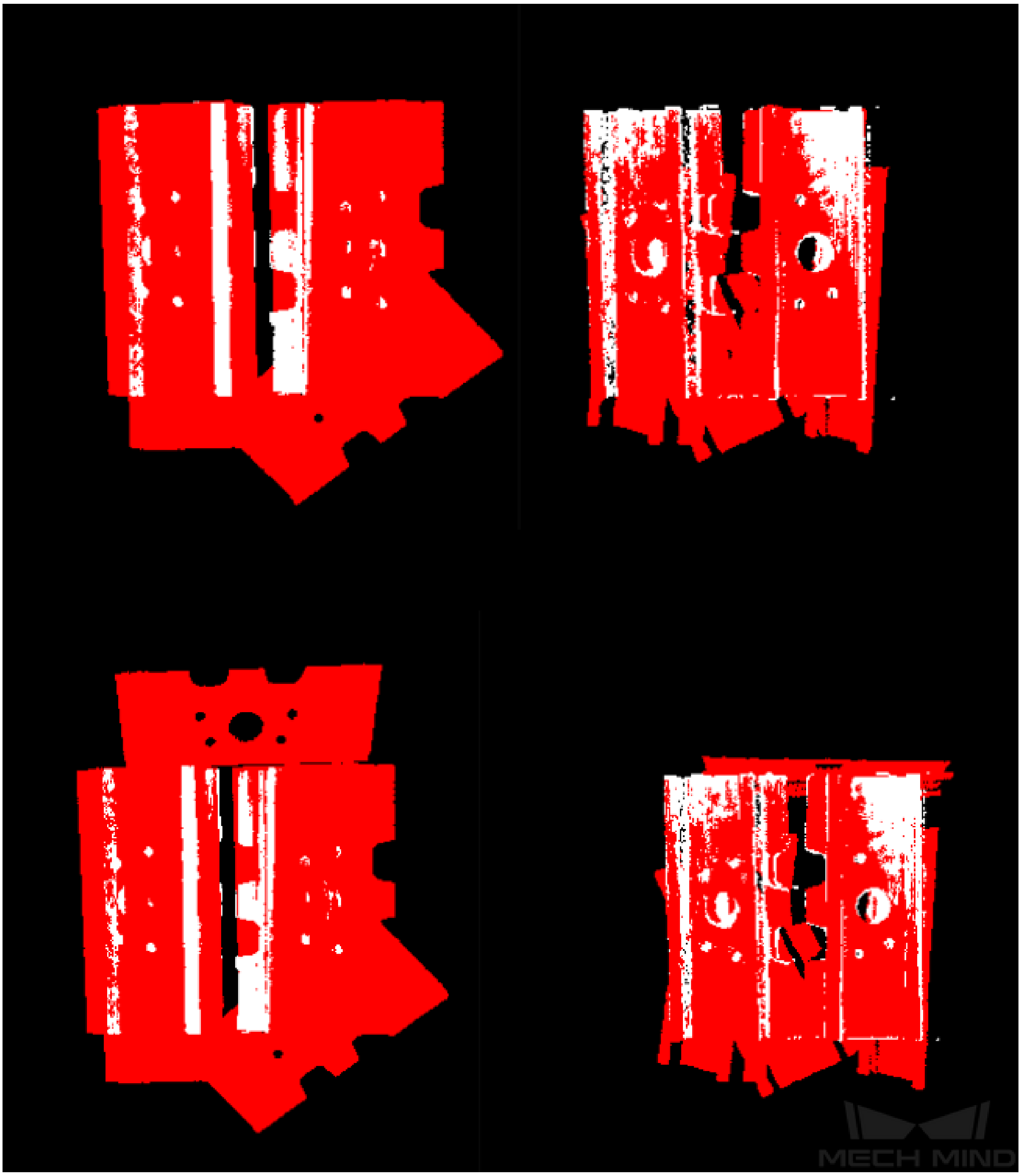


Figure 5 Comparison of Parameter Adjustment Results of Maximum Number of Detected Poses in Each Point Cloud

The parameters of **SurfaceMatching** are described below.

Sample Settings

Enable Automatic Downsampling

Default Value: True

List of Values: True, False

Instruction: This parameter is used to determine whether to use automatic downsampling. If set to `True`, the sampling interval parameter of model point cloud 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 Downsamplin** is True, 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 themodel point cloud after downsampling. It sets an upper limit for the number of points in ourmodel point cloud 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: 0.01

Instruction: This parameter is used to adjust the maximum distance between points in the sampling point cloud. When the sampling interval of model point cloud 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 *Figure 6*. 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`.

Figure 6 Comparison of Sample Interval Parameter Adjustment Results of Model Point Cloud

Min Sampling Interval

Default Value: 0.003

Instruction: This parameter is used to calculate the sampling interval. It is effective when the value of **Enable Automatic Downsampling** is True. 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 (Distance Interval = **Distance Quantification** * **Sampling Interval**). 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 point' s normals (Angle Interval = 2 * 3.14 / **Angle Quantification**). The bigger the value is, the less the result tends to do.

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.

Reference Point Step

Default Value: 5

Instruction: This parameter is used to sample points in the point cloud to make a point pairs, from the point cloud with the step size as an interval sampling point. 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 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.

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 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.

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: True

List of Values: True, Flase

Instruction: This parameter determines whether pose validation is used. When the parameter value is `True`, 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

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 6* is the result when the parameter value is 3, and the right side is the result when the parameter value is 5.

Figure 7 Comparison of Maximum Number Parameter Adjustment Results

Results Visualization

Show Sampled Model Cioud

Default Value: False

List of Values: True, Flase

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

Show Sampled Scene Cioud

Default Value: False

List of Values: True, Flase

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

Show Matching Results

Default Value: True

List of Values: True, Flase

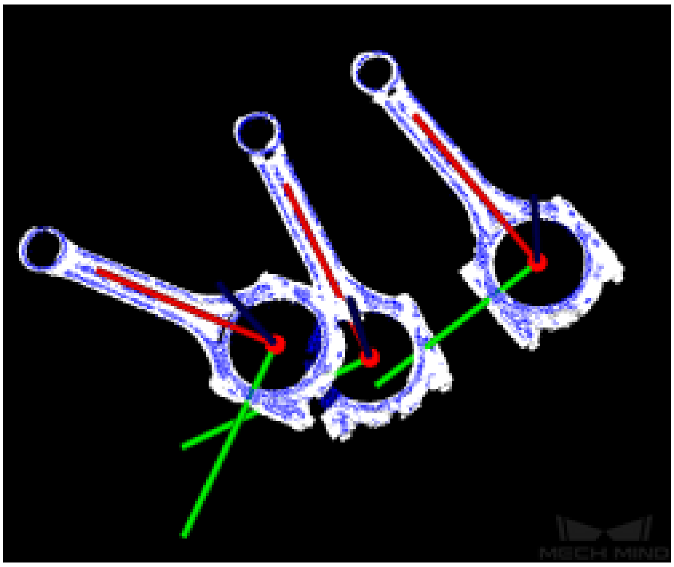
Instruction: This parameter is used to display the matched model and field point cloud.

4.9.3 3D Fine Matching

Function

Accurately match the model point cloud with the original point cloud, and outputs more accurate poses of target objects.

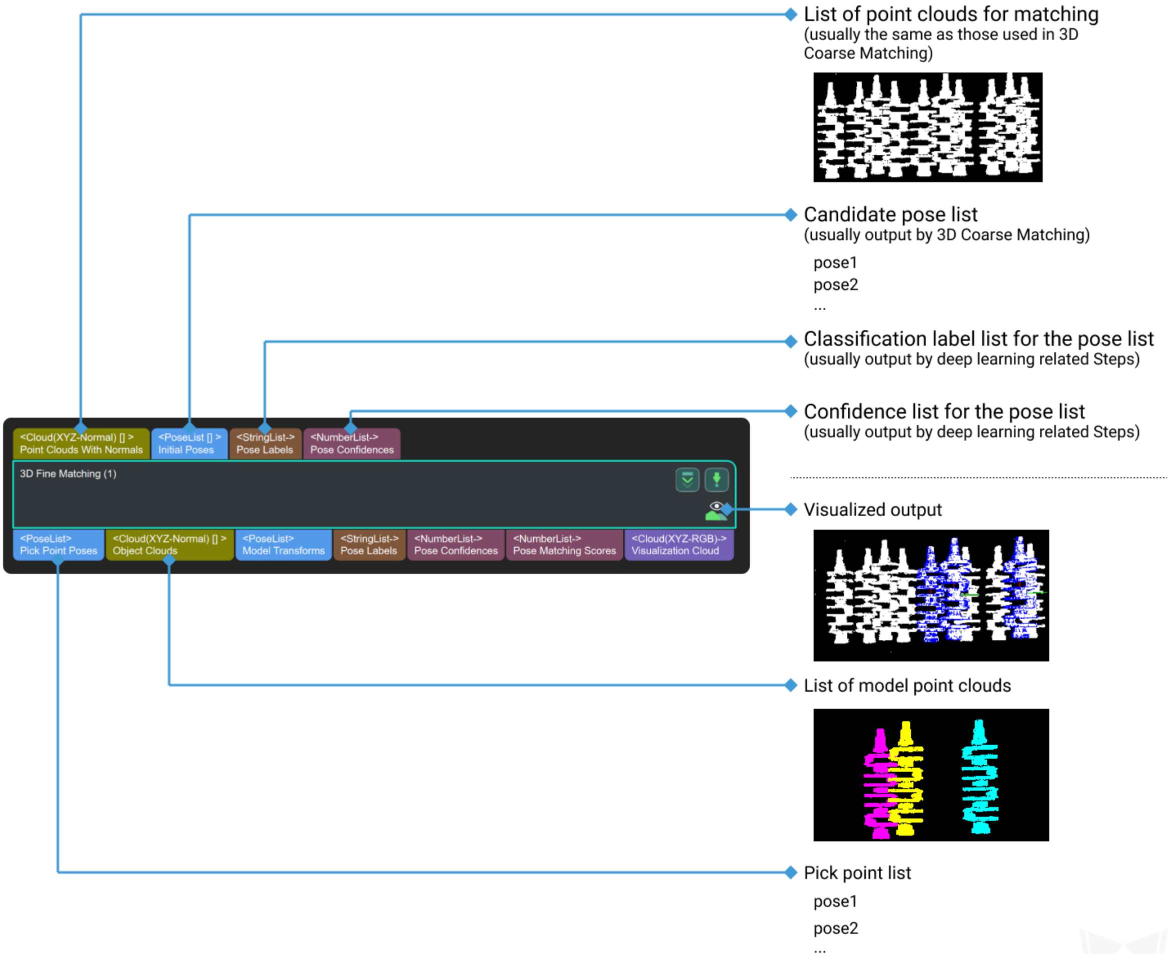
Hint: Please refer to *Generate Point Cloud Model* to create a point cloud model.



Sample Scenario

This Step performs fine matching based on the matching results of *3D Coarse Matching*, and outputs more accurate object poses, which can be used as pick points.

Input and Output



Parameters

The **Parameters Tuning Level**, found under **Execution Flags** under, sets the extent of completeness that the parameters are displayed for adjustment.

> **Options**: `Basic`, `Advanced`
>
> **Default**: `Basic`
>
> - `Basic`: Only commonly used parameters are displayed for adjustment.
> - `Advanced`: All parameters are displayed for adjustment.

Advanced parameters are marked with below.

Key parameters	Introduction
Model and Pick Point	Set the model file path and the geometric center file path
Cloud Orientation Calculation	Choose the calculation method of point cloud orientation
Correspondence Settings	Select the method to match the model point cloud with the object point cloud
Sample Settings	Set the details of sampling the point cloud
Symmetry Settings	Tell the matching algorithm the symmetricity of the objects
Model Weight in Validation	The weight added to object parts with obvious features to facilitate matching
Validation Settings	Settings on matching result validation
Output Settings	Settings on matching result output
Acceleration Settings	Whether to use GPU to accelerate the running
Results Visualization	What parts of the results to visualize

Model and Pick Point

Model and Pick Point set paths of the .ply file that contains the point cloud(s) of a single model (multiple point clouds taken from different perspectives of a single model object when necessary) and the .json file that contains the information of pick point(s) of the model object.

When the paths are changed, the next time this step is called, the files will be re-read. Both absolute paths and relative paths can be recognized.

Model File

Default: `model.ply`

Description: The file path of the .ply file that contains the model point cloud. Please see *Generate Point Cloud Model* for details about generating model point clouds.

Geometric Center File

Default: `model/pickPoint.json`

Description: This file contains the pick point information of the model point cloud. This file can contain geometric center information for testing purposes, but in practice, when the robot performs actual picking, the file must contain pick point information. Please see *Add Pick Point* for details about generating pick point files.

Cloud Orientation Calculation

Point Cloud Calc Mode

Default: `Origin`

Options: `Origin`, `StandardMode`, `EdgeTangent`, `EdgeNormal`

Description: The above options represent four methods for calculating the direction (normal) of the model point cloud required for the matching algorithm. The normal of the point cloud is perpendicular to the image plane of the depth map on which the point cloud is generated.

- `Origin`: Directly obtain the normal from the model point cloud when the file contains the normal of the model point cloud.

- `StandardMode`: Select a target point, search for the k points closest to the target point in the vicinity of the target point, and find the minimum feature vector as the normal by principal component analysis.
- `EdgeTangent`: Calculate the tangent at the edge of the model point cloud, and take the tangent as the normal. This method can distinguish different objects whose outer contours are mirror images of each other.
- `EdgeNormal`: Calculate the normal at the edge of the model point cloud. This method is recommended for relatively flat objects.

Note: When using `EdgeTangent` or `EdgeNormal`, please ensure that each edge point cloud does not contain multiple objects (each object’s point cloud should be separate and no object should overlap in the point cloud).

Number of Searching Points

Default: `10`

Description: This parameter is for `StandardMode` mentioned above. It sets the k value, i.e., the number of points in the vicinity of the target point to search for. The minimum value is `1`. The larger the value, the more accurate the result, the slower the computation.

Correspondence Settings

Correspondence Settings

Default: `GMM`

Options: `GMM`, `nearest-neighbor`

Description: Select the method to match the model point cloud with the object point cloud. The model point cloud will be moved and rotated within the set range to match the object point cloud.

Applicability: In most scenarios, GMM has better anti-interference performance and higher matching speed, so it is generally recommended; in rare cases, when GMM cannot meet the requirement, nearest-neighbor can be selected.

Please see *Settings under GMM* and *Settings under Nearest-Neighbor* below for parameter adjustment instrutions of the two methods.

The figure below is an example of a comparison between the input point cloud and the visualized output matching results via nearest-neighbor.

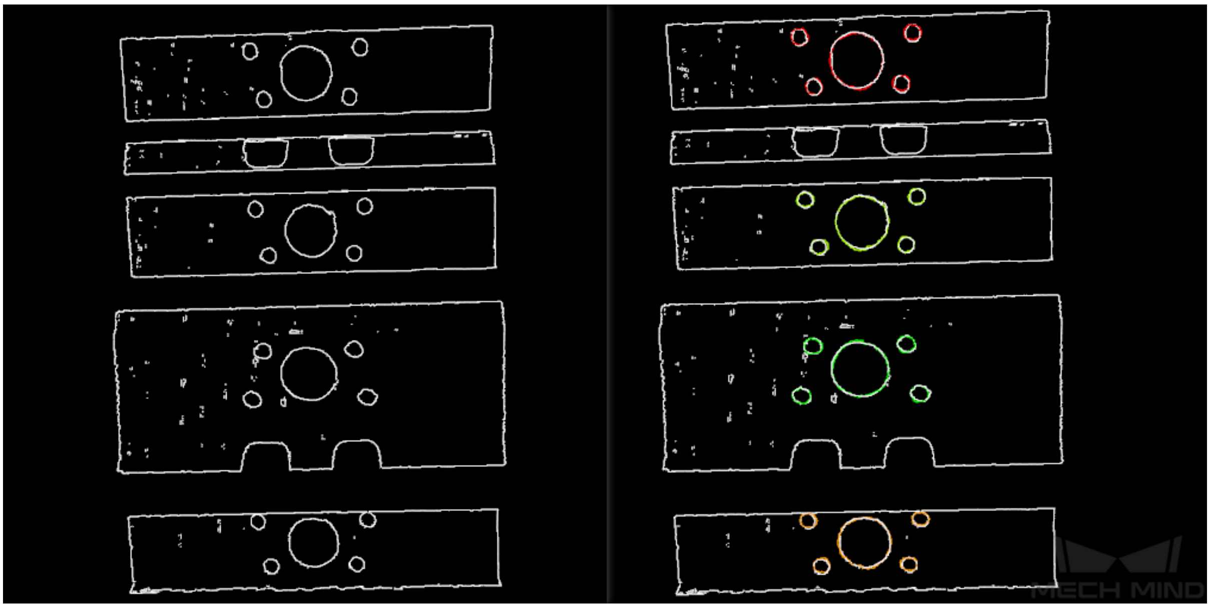


Figure 1. Input/output of matching via nearest-neighbor

Settings under GMM

Matching Mode

Default: `Standard`

Options: `HighSpeed`, `Standard`, `HighPrecision`

Description: please select based on the accuracy and speed requirements.

- **HighSpeed**: The fastest but less accurate.
- **Standard**: Relatively stable.
- **HighPrecision**: The most accurate but slower.

Number of Iterations

Default: `30`

Description: 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, the greater the number of times of matching calculations, the longer the running time, the higher the accuracy.

Standard Deviation

Default: `0.005 m`

Description: This value is a measure of the expected deviation of the matched pose of the model from the actual pose of the object after *3D Coarse Matching*. The closer the expectation to the actual value, the faster the calculation can be, but a higher **Standard Deviation** will lead to less accurate matching results. Normally, this value will be no greater than `0.01 m`. |

Example: The figure below is a comparison between matching results when **Standard Deviation** is `0.04 m` and `0.01 m` respectively.

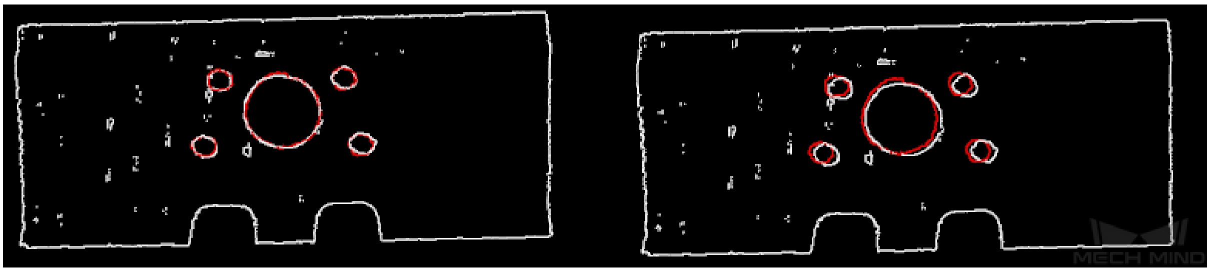


Figure 2. Comparison of results under different **Standard Deviations**

Standard Deviation Update Step Number

Default: `3`

Description: During the matching, the standard deviation gradually decreases, until it reaches the **Minimum Standard Deviation**. The value of this parameter is the number of times the standard deviation is updated in the matching algorithm. Normally this parameter does not need adjustment.

Minimum Standard Deviation

Default: `0.001`

Description: The target value the standard deviation needs to reach.

Apply Weight in Iteration

Default: `False`

Options: `True`, `False`

Description: If set to `True`, the weight of the points in the model point cloud will be used in the iteration to facilitate the matching.

Please see *Weight in Point Cloud Matching* for why and how weight works.

Settings under Nearest-Neighbor

Iteration Settings

Number of Iterations

Default: `30`

Description: Iteration refers to repeatedly 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, the greater the number of times of matching calculations, the longer the running time, the higher the accuracy.

Nearest Point Search Radius

Default: `0.01`

Minimum: `0.001`

Description: It adjusts the search radius of the closest point, its unit is m, and its value should correspond to the input initial pose deviation, i.e., the pose deviation after 3D coarse matching. If the deviation is large, the search radius should be large. The value of this parameter should be no lower than `0.001`; otherwise, some sparsely distributed points of the object point cloud may not be matched to their nearest points in the model point cloud, thus affecting the output result.

Example: When the initial deviation is large while this parameter is set to the minimum value `0.001`, the matching will be incomplete. When this parameter is increased to a larger value like `0.01`, the search radius increases, and the matching will be comparatively complete.

Below are two parameters that set the standard by which the matching iteration ends. The iteration ends when either of the standards set by **MSE Threshold** and **Window Size** is met.

MSE Threshold

Default: `0.001`

Description: It adjusts the threshold of the mean square error (MSE), which is a measure of the closeness of the matching. The MSE is recorded and compared with the **MSE Threshold** after each iteration. When the MSE after an iteration is less than the threshold, the iteration ends.

Window Size

Default: `10`

Minimum: `3`

Description: When the number of iterations during which the deviation’s fluctuation remains trivial reaches the **Window Size**, the iteration ends. The larger the **Window Size**, the longer the computation may take, but a **Window Size** may lead to suboptimal results. For instance, if the **Window Size** is set to the minimum value `3`, the iteration ends when the deviations after three consecutive iterations show trivial fluctuation, and the matching result may not be optimal because had the iteration continued, yet smaller deviation would have been produced.

Show Corresponse

Default: `False`

Options: `True`, `False`

Description: If it is set to `True`, the paired point cloud points will be displayed for each iteration.

Complex Object

Default: `False`

Options: `True`, `False`

Description: It decides whether the algorithm will allocate resources to treat complex object point cloud matching. If the target object is a complex shaped object (not a simple geometric shape) and **Automatic Weight** is set to `True`, please set it to `True`.

Automatic Weight

Default: `False`

Options: `True`, `False`

Description: If set to `True`, the function of automatically applying weight to the point cloud points will be enabled.

Please see *Weight in Point Cloud Matching* for why and how weight works.

Point Pair Rejection

Reject Pair

Default: False

Options: True, False

Description: Nearest-neighbor search will find multiple matches for one point, so the function of rejecting pairs should be enabled to filter those matches. If set to True, the function will be enabled. **N Sigma Threshold** and **Point Pair Angle Diff Thre** are parameters to the function of rejecting invalid pairs. If it is set to False, the two parameters below will not take effect.

N Sigma Threshold

Default: 1.000,0

Description: It sets the threshold for the distance between the points in a pair. Suppose the mean standard deviation of the distances in all point pairs in the point cloud is , the mean of the distances in all point pairs in the point cloud is D. If a point pair's distance is higher than D + threshold* , the pair is deemed invalid and is rejected.

Point Pair Angle Diff Thre

Default: 45°

Description: If the angle difference between the normals of the points in a pair is greater than this value, the point pair will be eliminated.

Apply Weights on Pairs

Default: False

Options: True, False

Description: If set to True, weight will be applied in the iteration. Please see *Weight in Point Cloud Matching* for why and how weight works.

Remove Repeat Correspond

Default: False

Options: True, False

Description: If set to True, the function of removing redundant point pairs will be enabled, and the settings of **Remove Repeat Correspond Type** will take effect. It is recommended to set to True when there are more points in the model point cloud than in the point cloud of a single object in the scene.

Remove Repeat Correspond Type

Default: MinDis

Options: MinDis, GlobalMin

Description: It is to select the method of finding and deleting the redundant correspondences between a point in the object point cloud and points in the model point cloud.

- MinDis: Only keep the pair with the shortest distance.
- GlobalMin: Obtain the mean distance in all pairs, and only keep the pair with the distance closest to the mean distance.

Sample Settings

Sampling Interval

Default: `0.005 m`

Description: It is the sampling interval used for downsampling the model point cloud and the object point cloud. The larger the value, the fewer the points will remain in the point clouds, and the lower the accuracy of matching.

Example: *Figure 3* is a comparison of the output results when **Sampling Interval** is set to `0.025m` and `0.005m`, respectively. The white parts are the input object point clouds, and the colored lines are the matching results. The comparison of the two figures shows that the smaller the sampling interval (right), the more accurate the output matching result can be.

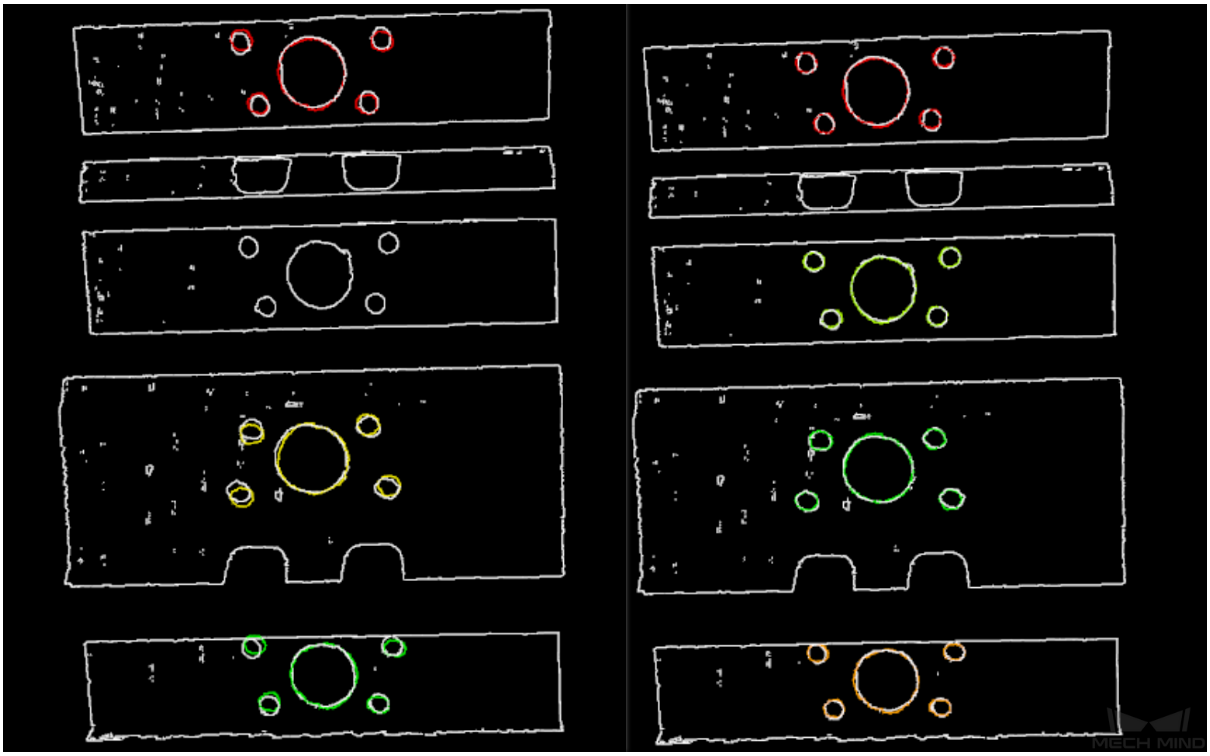


Figure 3. Comparison of outputs with small and large sampling intervals

Symmetry Settings

Symmetry Axis

Default: `ROTATE_BY_Z`

Options: `ROTATE_BY_X`, `ROTATE_BY_Y`, `ROTATE_BY_Z`

Description: For objects with symmetry, this parameter is to select the symmetry axis of the model point cloud.

Angle Step (0–360)

Default: `360°`

Description: This parameter is to adjust the symmetry angle step. For instance, if an object looks the same before and after rotating 60° around the symmetry axis, its symmetry angle step is 60°.

Min Rotation Angle (-180–180)

Default: `-180°`

Max Rotation Angle (-180–180)

Default: `180°`

Description: **Min Rotation Angle** and **Max Rotation Angle** together tell the matching algorithm the range of the possible angle the object is positioned within the angle step around the symmetry axis. For instance, if the angle step is 60°, then the range can be -180° to -120°. Usually, the range does not need to be adjusted.

For more information about symmetry, please see symmetry-label.

Model Weight in Validation

File of Model Fragments with Heigh Weight

Description: This .ply file contains the part of the model point cloud that needs to have a higher weight. The weight value is set by **Weight of Each Point** below.

For the reason of setting the weight, please see *Weight in Point Cloud Matching* below.

Weight of Each Point

Default: `2.0`

Description: This parameter is to set the weight value for the parts with salient features in the model point cloud. The initial weight of all points is 1. After applying the weight, the points in the target part of the model point cloud will have a weight of this parameter.

Search Radius When Set Weight

Default: `0.003,0`

Description: This parameter is used to set the search radius when applying the weight. The weight setting is an operation performed on the original model point cloud. The model point cloud will be down-sampled before being used in the matching process, resulting in the shifting or loss of some points that need to be weighted. When setting the weight of the points near a missing point, the search radius needs to be set by this parameter.

Validation Settings

Confidence Threshold

Default: `0.500`

Minimum: `0`

Description: The confidence threshold decides the standard by which each match between an object point cloud and its matched model point cloud is considered valid. Matches with confidences higher than the threshold are valid and those lower are not. The confidence of a match is a measure of the spatial closeness between the object point cloud points and its matched model’s point cloud points. Therefore, a lower threshold means more matches are valid for output and the output is less accurate, and a higher one means less valid matches but higher accuracy of the output.

Limitation: Please note mismatches may still occur given a higher threshold, especially when the object’s different parts are of the same shape.

Instructions:

The **Confidence Threshold** needs to be tuned during testing for the optimal effect, its minimum is 0, but its maximum `max` should be calculated as follows:

Suppose the value of *Weight of Each Point* is `w`. For the reason of setting the weight, please see *Weight in Point Cloud Matching* below.

The ratio of weighted points to all the points in a model point cloud is `r`, that is:

`r` = point count of *File of Model Fragments with Heigh Weight* / point count of *Model File*.

`max` = `w` * `r` + (`1` - `r`)

Please set **Confidence Threshold** to a value between `0` and `max`.

The point count of a point cloud is displayed on the lower left corner of the window that pops up by double clicking on the input/output ports in the graphical programming workspace. as shown below:

Figure 4. Point count of a point cloud

Example: *Figure 4* below is an example showing two valid matches obtained under small and large confidence thresholds respectively. The confidence threshold of (a) is `0.50` and that of (b) is `0.92`. The match of (a) is more accurate than that of (b). Even though the comparison is not prominent, as 3D fine matching itself is a process of fine-tuning, the difference can still be observed, especially in the yellow boxes.

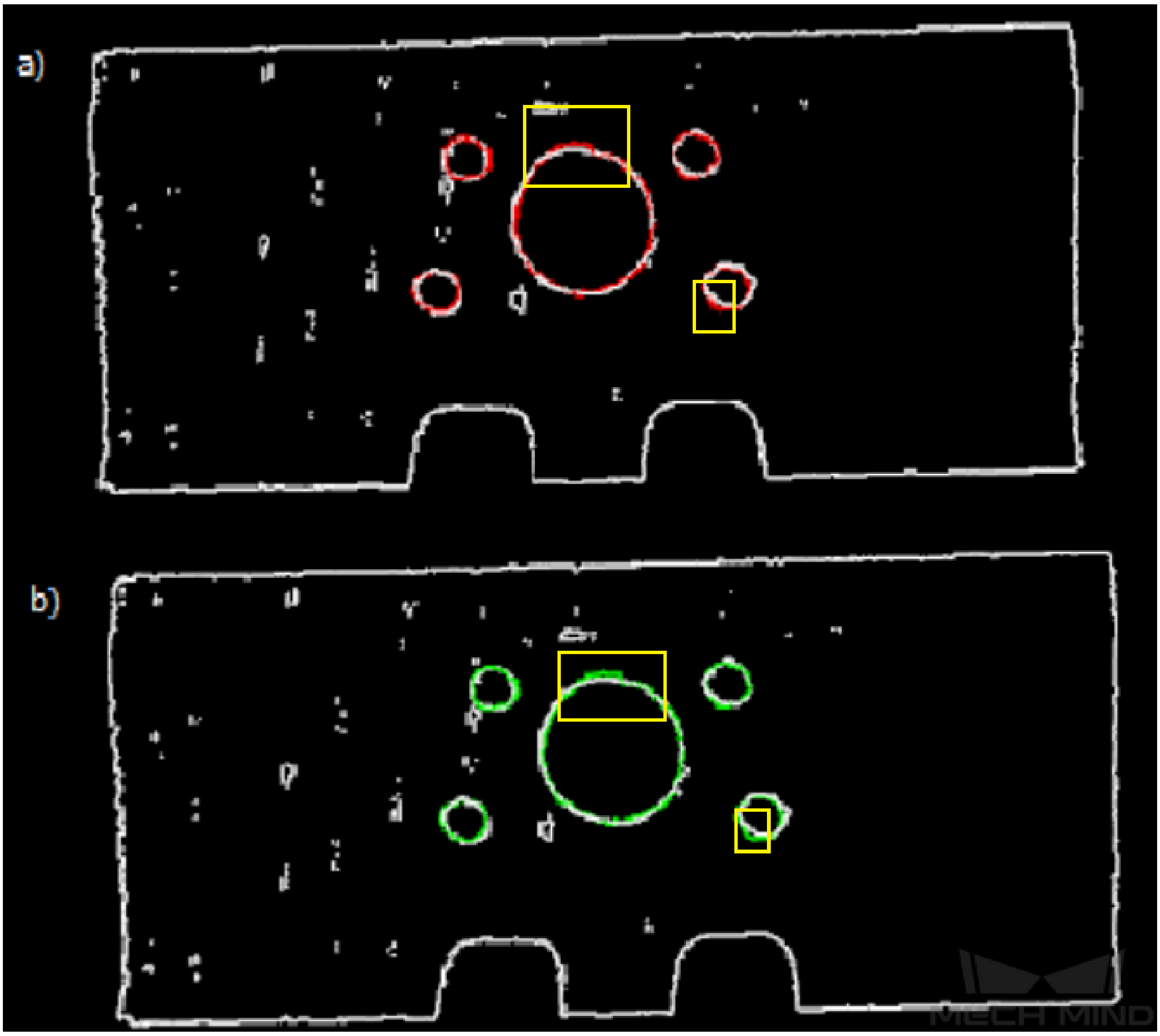


Figure 5. Comparison of matching results under different confidence thresholds with marks

Search Radius

Default: `0.01 m`
Minimum: `0`
Description: It is the search radius when calculating the confidence. It is set by the sparseness of the object point cloud. If the point cloud is sparse, it needs to be set slightly larger.

Apply Angle Deviation in Validation

Default: `False`

Options: `True`, `False`

Description: If set to `True`, during the validation, the angle difference of the normals of the corresponding points in a pair will be considered, and the number of output results will be reduced, but the output will be more accurate.

Multiply Corresponding Points Ratio in Scene

Default: `False`

Options: `True`, `False`

Description: If set to `True`, the validation score of matching will be multiplied by the ratio of the number of the points with correspondence to the number of points in the entire point cloud. The validation score is printed in the log for debugging, and is printed only when the log level is set to `Debug`. It should be set to `False` for obtaining the poses of multiple objects in a point cloud. *Figure 6* is an example of validation scores printed in the log when debugging.

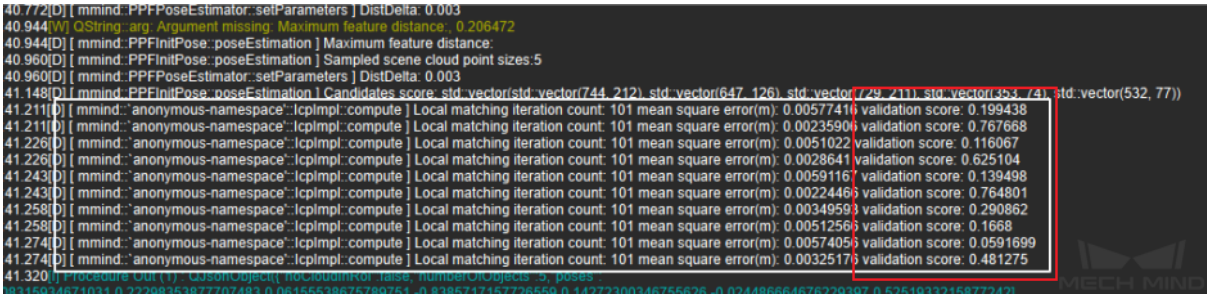


Figure 6. Validation scores in the log

Output Settings

Maximum Number of Detected Poses in Each Point Cloud

Default: `1`

Description: The larger the value, the more matching pairs will be output.

Example: *Figure 7* is a comparison of the output results when it is set to `1` and `4` respectively.

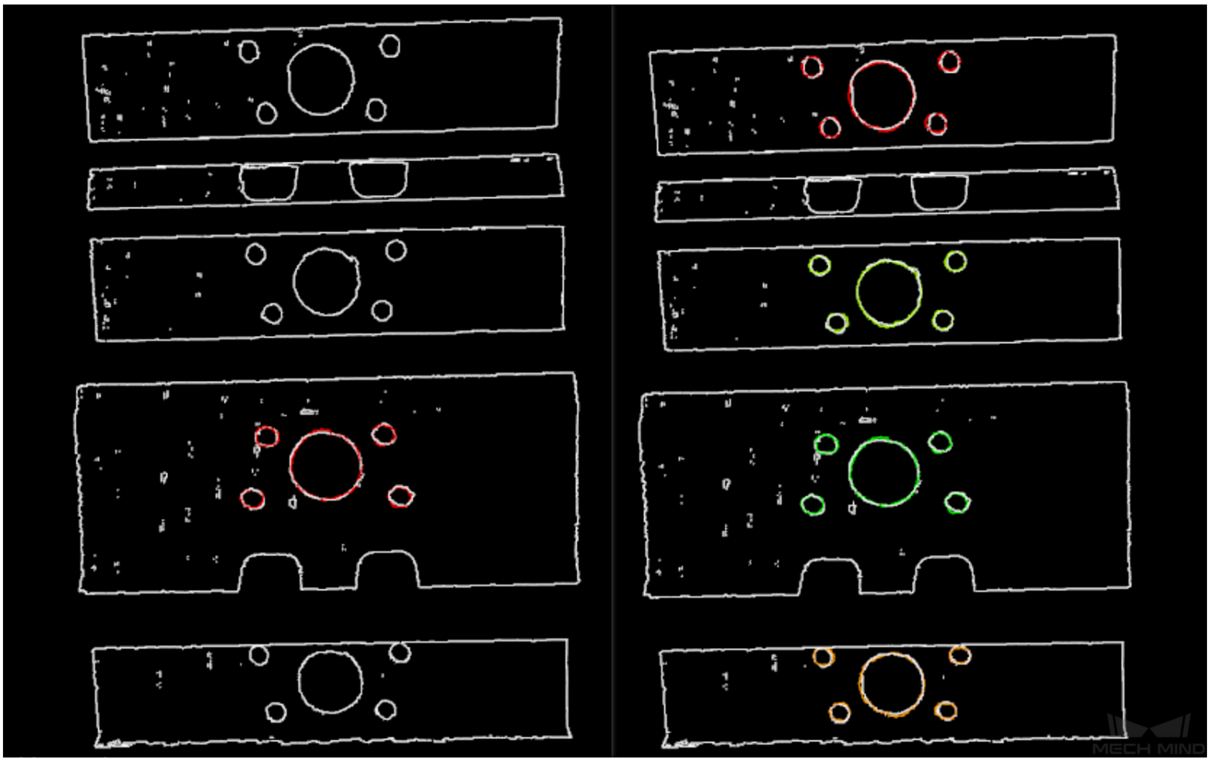


Figure 7. Comparison of single and multiple output results

Acceleration Settings

Acceleration with GPU

Default: `False`
Options: `True`, `False`
Description: If set to `True`, GPU acceleration will be enabled. Please set by actual needs.

Results Visualization

Show Sampled Model Cloud

Default: `False`
Options: `True`, `False`
Description: If set to `True`, the sampled model point cloud will be displayed in the output result. Please set by actual needs.

Show Sampled Scene Cloud

Default: `False`
Options: `True`, `False`
Description: If set to `True`, the sampled object point cloud will be displayed in the output result. Please set by actual needs.

Show Validation Point Correspondences

Default: `False`

Options: `True`, `False`

Description: If set to `True`, the correspondence between the model and the object point clouds will be displayed in the output result. Please set by actual needs.

Show Matching Results

Default: `False`

Options: `True`, `False`

Description: Display the output model and object point clouds when set to `True`.

Weight in Point Cloud Matching

Of an object, some parts have salient features, such as the corners, the holes, etc., and those parts should be paid special attention to by the matching algorithm to make the matching easier. Therefore, weights should be applied to the points of those parts in the point cloud so that the algorithm gives importance to those parts. In particular, weight should be applied when matching without weight does not work well.

Normally, among the points in a point cloud in Mech-Vision, a point is either not weighted or weighted by a fixed multiplier.

Yet please note that, after applying weight, mismatches may still occur especially when the object's different parts are of the same shape or have similar features.

Parameters involved in weighting:

- *Weight of Each Point*
- *File of Model Fragments with Heigh Weight*
- *Model File*

4.9.4 3D Fine Matching (Multiple Models)

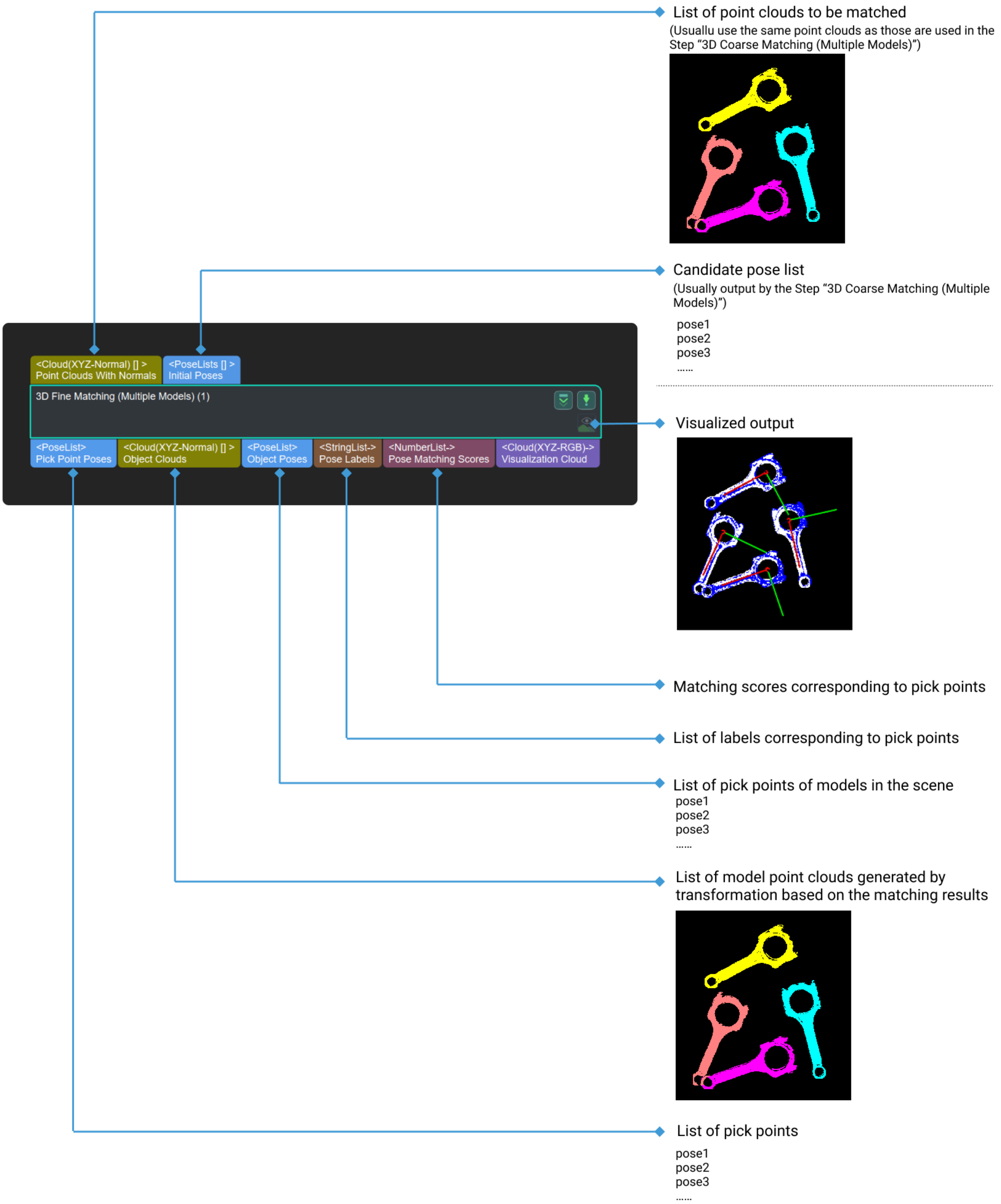
Function

Use multiple models to accurately match objects in the scene, and output accurate pick points of the target objects.

Sample 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



Parameters

Parameters 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 and Pick Point

The path of model file and pick point file.

Model File

Default Value：model.ply

Instruction：This parameter is used to set the path of the 3D model file. If the file path or name is changed, the file will be re-read when the step is called next time. Both full path and relative path of the model file can be recognized.

Geometric Center File

Default Value：model/pickPoint.json

Instruction：The pick point file is based on the 3D model file. For the method of generating pick point files, please refer to *Add Pick Point*. If the file path or name is changed, the file will be re-read when the step is called next time. Both full path and relative path of the model file can be recognized.

Model Label File

Instruction: Fill in the path of the model label file in json format. Multiple file paths are allowed to be set and separated by semicolons. If the file path or name is changed, the file will be re-read when this Step is called next time.

Example: Ensure that the order of the file paths is consistent: if the front file path of an object is entered first in the **Model Label File**, the corresponding positive file path should also be entered before the negative one in both **Model File** and **Geometric Center File**, as shown in *Figure 1*.

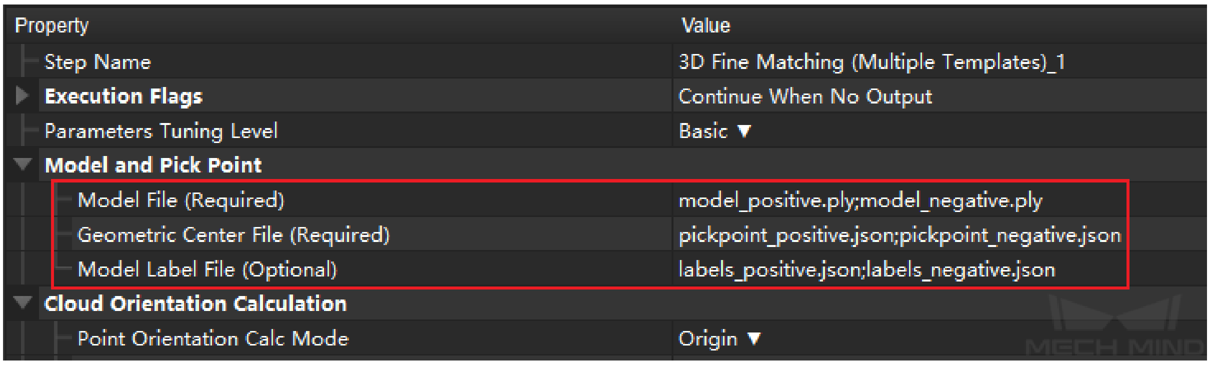


Figure 1 Model and Pick Point File Path Example

After inputting file paths correctly, the output point cloud image is shown in *Figure 2*. Both the front and back templates of the object can match the scene point cloud well.

Figure 2 Multi-Template 3D Fine Matching Output

Cloud Orientation Calculation

Point Orientation Calc Mode

Default Value: Origin

List of Values: Origin, StandardMode, EdgeTangent, EdgeNormal

Instruction: There are four methods for calculating the direction of the point, which can be selected according to the actual situation of the project:

- **Origin**: Directly use the original normal direction of the input point cloud;
- **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 point, and use PCA (Principal Component Analysis) to obtain the minimum eigenvector as the point' s normal direction.
- **EdgeTangent** : Calculate the tangent of the input edge point cloud, and use the tangent direction of the point 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: Distinguish the front (positive) and back (negative) of the object. Use the *3D Fine Matching (Multiple Models)* step, and the model point cloud is the positive and negative template of the object;
  - Scenario 2: No need to distinguish the front and back of the object. Use the *3D Fine Matching* (single template) step.
- **EdgeNormal** : Calculate the normal direction of the input edge point cloud. This mode can also be used when matching the edge point cloud of a flat object.

Note: When using **EdgeTangent** or **EdgeNormal** methods, please ensure that each edge point cloud does not contain multiple objects (each object' s point cloud has been divided already).

Number of Searching Points

Default Value: 10

Instruction: This parameter is used to set the number of used points when calculating the orientation. 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 model point cloud with the scene point cloud. The model point cloud will move and rotate according to the set range to match the point cloud in the scene obtained in the previous step match. 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 to try.

GMM Mode

Matching Mode

Default Value: Standard

List of Values: HighSpeed, Standard, HighPrecision

Instruction: There are three matching modes, and 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: 0.005

Instruction: The value here needs to macth 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. 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 `0.01m`.

Standard Deviation Update Step Number

Default Value: 3

Instruction: During the matching process, the **Standard Deviation** gets smaller and smaller until it reaches the value of **Minimum Standard Deviation**. The number of times that the standard deviation decays in this process is the value of this parameter. For some cases with large standard deviation, the more attenuation steps, the more stable the entire attenuation process. Under normal circumstances, this parameter does not need to be adjusted.

Minimum Standard Deviation

Default Value: 0.001

Instruction: The lower limit of the **Standard Deviation** .

Apply Weight in Iteration

Default Value: False

List of Values: True, False

Instruction: If checked, the weight of the model will be used during the iteration of the algorithm.

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. The parameter set here is the maximum number of iterations. In the nearest point mode, the iteration stops when the calculation meets the accuracy requirements of the algorithm. 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: 0.01 m

Instruction: This parameter is used to adjust the search radius of the nearest point (in m), 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 model point cloud 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 *Figure 3*; 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 *Figure 3*.

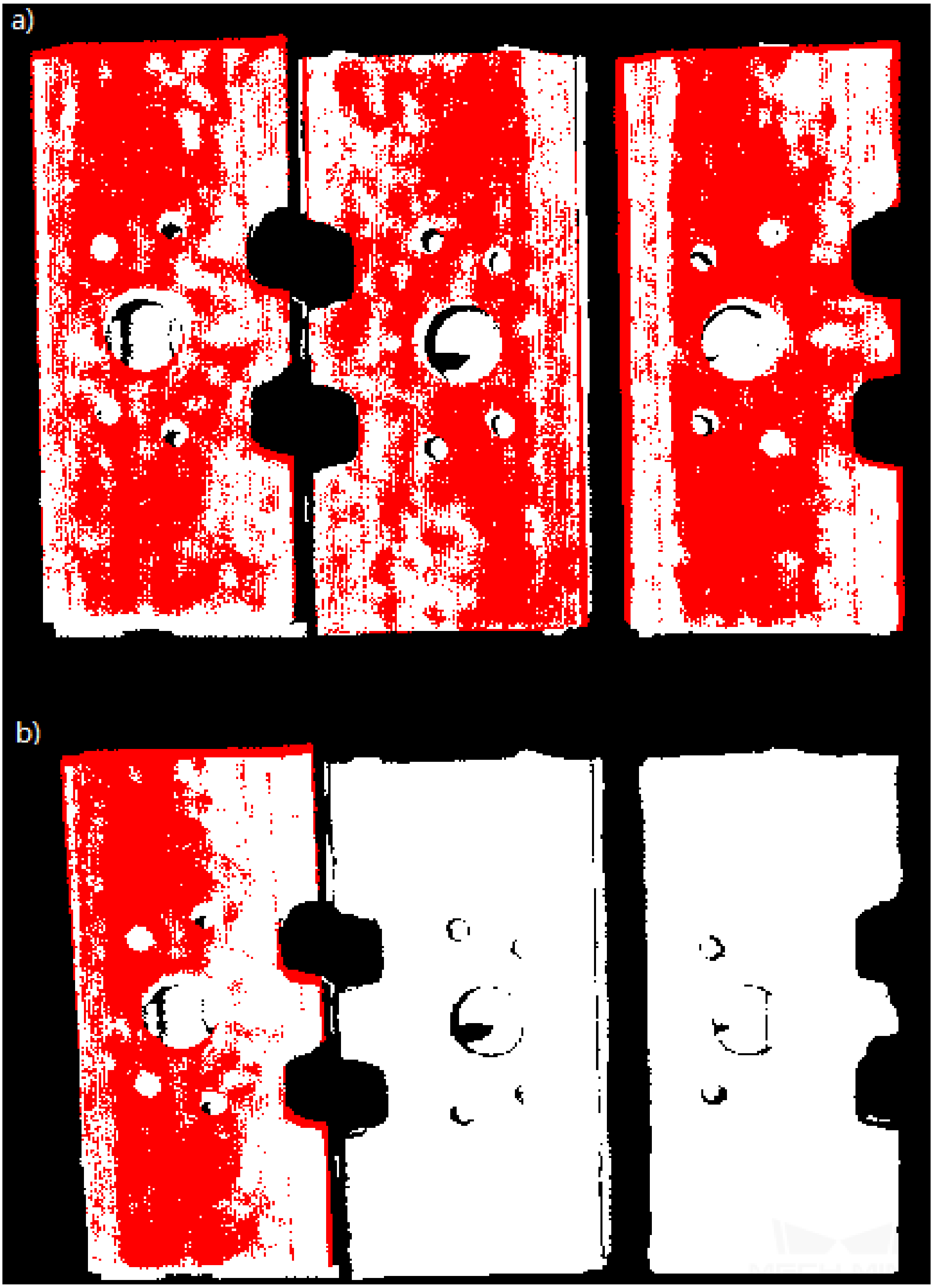


Figure 3 Comparison of Matching Results with Different Search Radius

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. 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 Correspondence

Default Value: False

List of Values: True, False

Instruction: If True is checked, the corresponding points will be displayed in each iteration.

Complex Object

Default Value: False

List of Values: True, False

Instruction: Check True if the target object is a complex shape object (not a simple geometric shape) or if the **Automatic Weight** function is required.

Automatic Weight

Default Value: False

List of Values: True, False

Instruction: If True is checked, 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: False

List of Values: True, False

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 True is checked, 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: False

List of Values: True, False

Instruction: If True is checked, 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: False

List of Values: True, False

Instruction: If True is checked, 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 model point cloud has more points than that of a single object in the scene.

Remove Repeat Correspond Type

Default Value: MinDis

List of Values: MinDis, 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.

Sampling Settings

Sampling Interval

Default Value: 0.005

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: *Figure 4* and *Figure 5* are the output results when this parameter is set as `0.005` and `0.025`, 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 (*Figure 4*), the more accurate the output result will be.

Figure 4 Output with a Small Sampling Interval

Figure 5 Output with a Large Sampling Interval

Search Points Number

Default Value: 20

Instruction: When **Parameters 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.

Validation Settings

Confidence Threshold

Default Value: 0.5

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.

Result evaluation search radius

Default Value: 0.01 m

Instruction: This parameter is the search radius when calculating the confidence level, which cannot be negative. It is set according to the point cloud situation of the object and its value cannot be negative. If the point cloud is sparse, this value needs to be set slightly larger.

Output Settings

Maximum Number of Detected Poses in Each Point

Default Value: 1

Instruction: The larger the value, the more matching output results.

Example: By adjusting this parameter, different matching output results can be obtained. *Figure 6* is the output result of the parameter when the default value is `1`, *Figure 7* is the output when the parameter is adjusted to `4`.

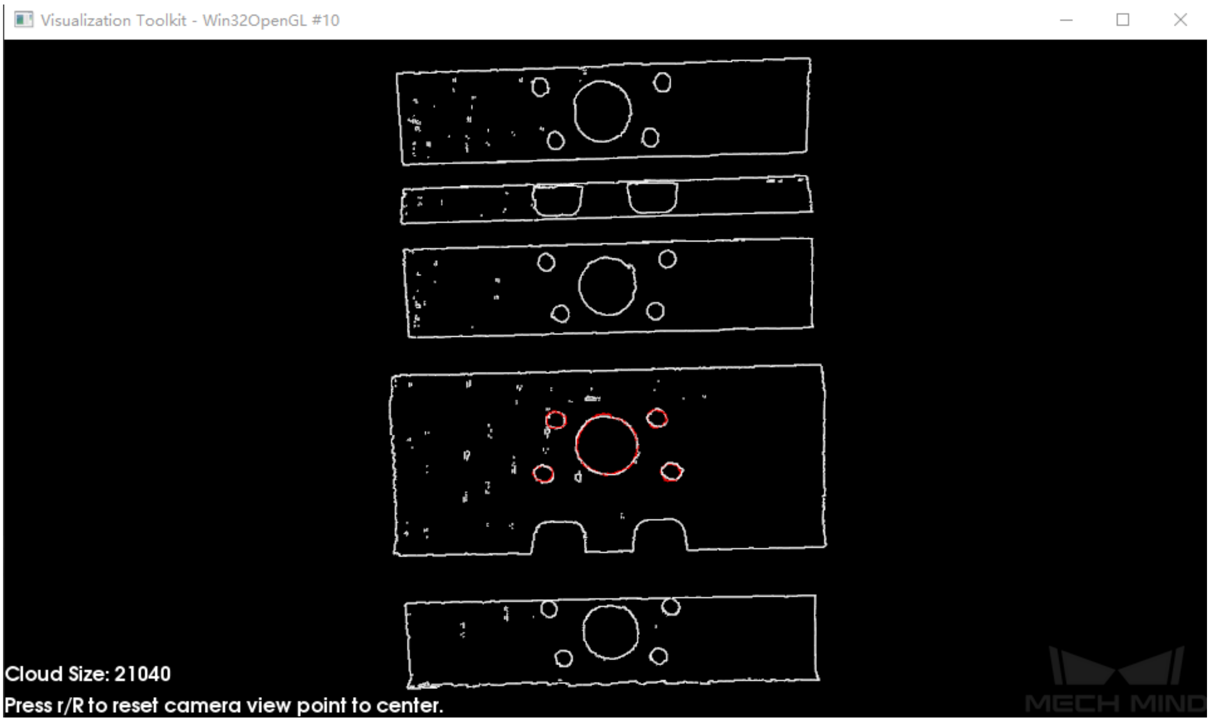


Figure 6 Single Result

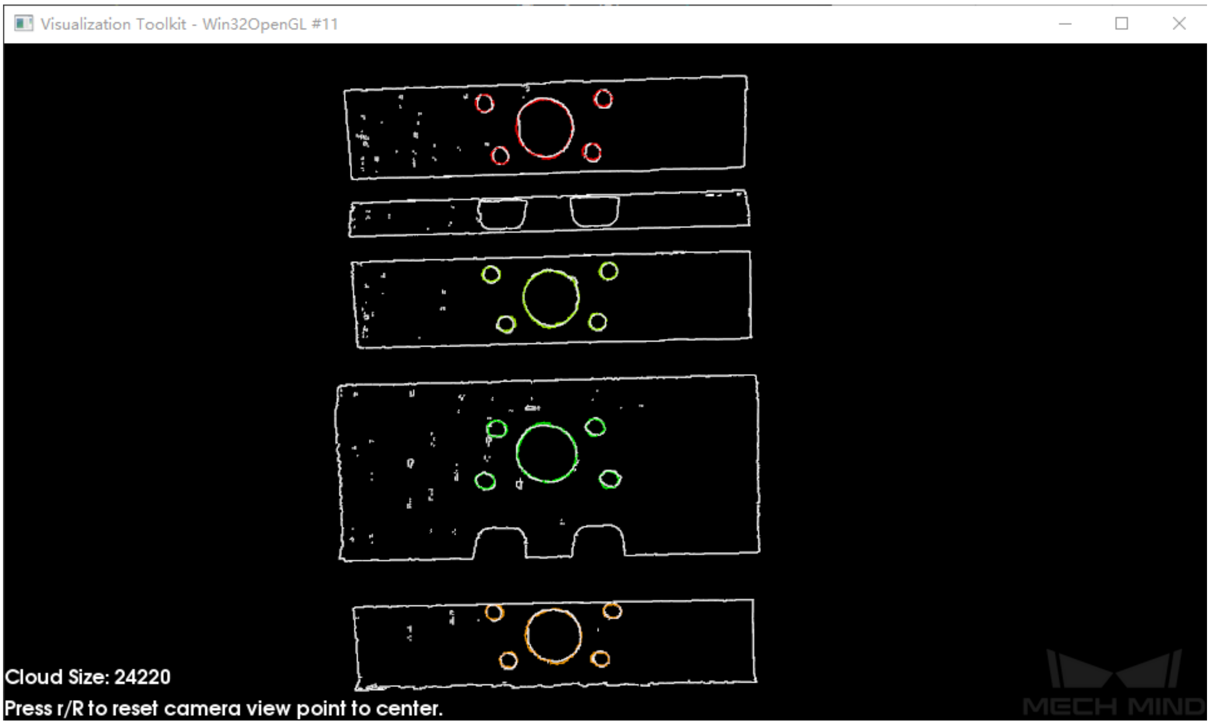


Figure 7 Multiple Results

Results Visualization

Show Matching Results

Default Value: True

Instruction: If True is checked, the estimated pose of objects (by rendering the transformed model and the scene point cloud) will be visualized.

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

Symmetry Axis

Default Value: ROTATE_BY_Z

List of Values: ROTATE_BY_X, ROTATE_BY_Y, 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** .

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.

Model Weight in Validation

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 higer the value is, the better the matching result of the model fragments tend to be.

Example: Assuming that the weight coefficient of all initial point clouds is `1`, after setting this parameter for the target point cloud, the weight coefficient of the target point cloud will become [1*this parameter], so as to achieve the purpose of emphasis in the subsequent matching steps.

Search Radius When Set Weight

Default Value: 0.0030

Instruction: This parameter is used to set the radius for setting weights. The bigger the value is, the more points in the model will have the set weight. The weight setting is an operation performed on the original model point cloud. The model point cloud 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: False

List of Values: True, False

Instruction: If True is checked, 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: False

List of Values: True, False

Instruction: If True is checked, 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.

Accleration Settings

Accleration with GPU

Default Value: False

List of Values: True, False

Instruction: If True is checked, GPU acceleration will be enabled to speed up the process.

Result Visualization

Show Sampled Model Cloud

Default Value: False

List of Values: True, False

Instruction: If you check True, the sampled model point cloud will be visualized in the output.

Show Sampled Scene Cloud

Default Value: False

List of Values: True, False

Instruction: If True is checked, the sampled scene point cloud will be visualized in the output.

Show Validation Point Correspondences

Default Value: False

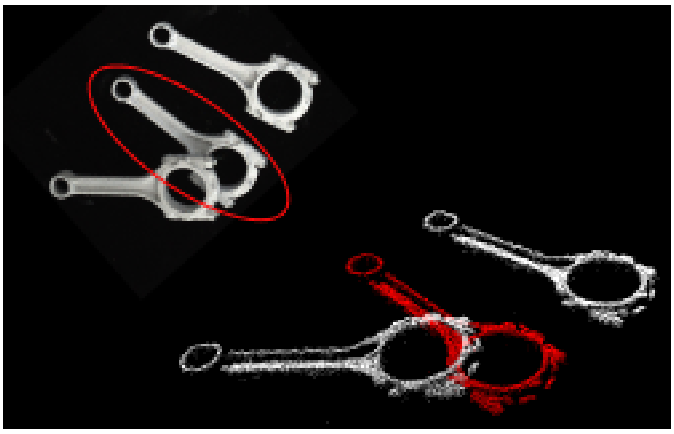
List of Values: True, False

Instruction: If True is checked, the corresponding relationship between the model and the scene point cloud will be visualized in the output.

4.9.5 Remove Overlapped Objects

Function

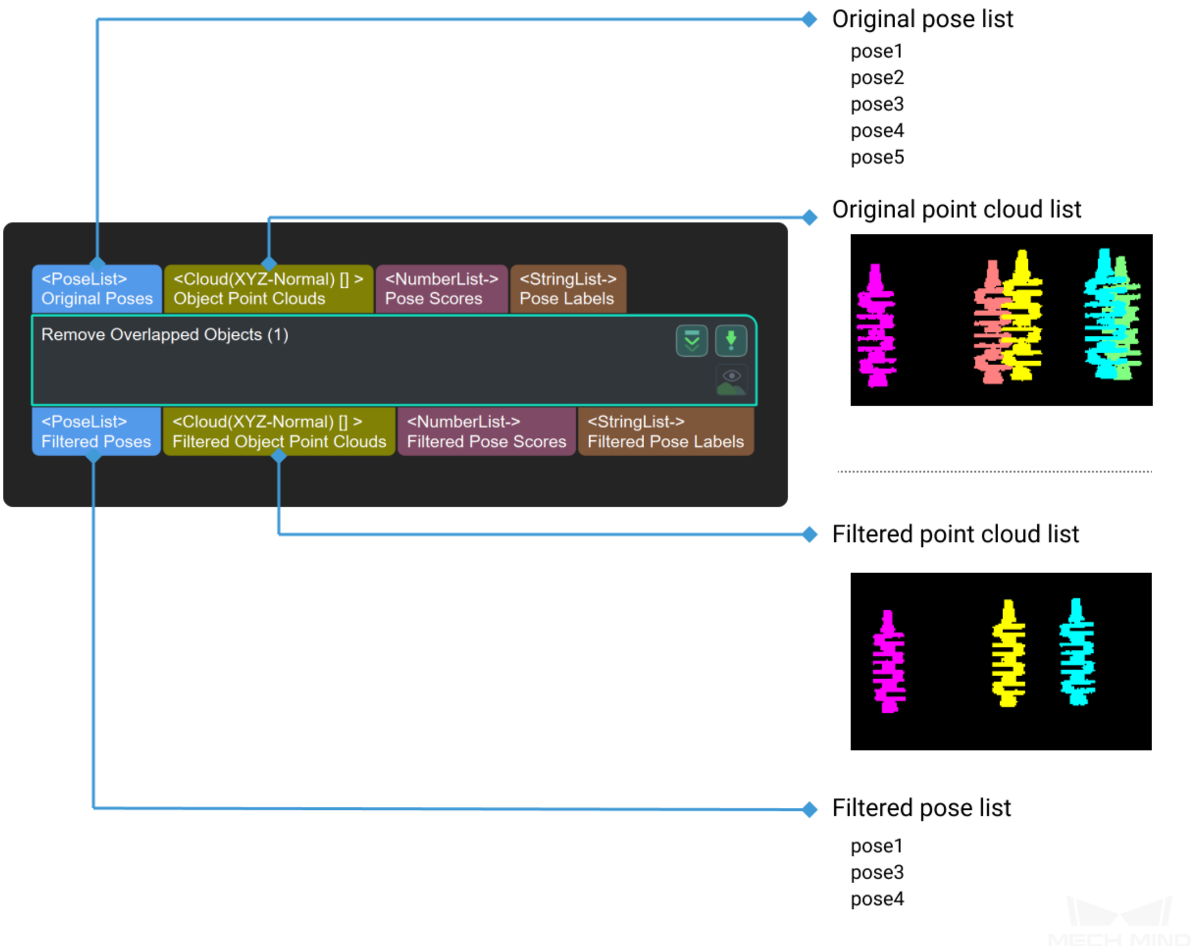
Remove the point clouds of the overlapped objects according to user-defined rules.



Sample Scenario

This Step is usually used after the Step *3D Fine Matching* to filter the matched poses according to different needs and therefore the poses of overlapped objects can be removed. It is recommended to try the projection algorithm (**ProjectionOfObjectIn2DBased**) first.

Input and Output



Parameters of Input and Output

PoseList[] [Input]the calculated picking poses

Cloud(Normal)[] [Input]the point clouds of model transformed by the calculated poses

DoubleList- [Input]the confidence of the point clouds

StringList- [Input]the labels of the corresponding point clouds

PoseList [Output]the filtered poses

Cloud(Normal)[] [Output]the point clouds of model transformed by the filtered poses

DoubleList- [Output]the confidence of corresponding point clouds

StringList- [Output]the labels of corresponding point clouds

Parameters

algoType

Instruction：This parameter is used to select the algorithms used in the process of removing overlapping objects,’UsingBoundingBox’and ‘UsingOverlap’included.

Default Value：UsingBoundingBox

Suggested Value：To set according to the actual scenarios.

overlap

overlapRatioThre

Instruction：This parameter is used to set the lower limit of the overlapping ratio threshold, If the overlapping ratio is greater than the limit, there exists overlapping objects.

Default Value：0.6000

Suggested Value：To set according to the actual scenarios.

resolution

leafsize

Instruction：This parameter is used to adjust the resolution of the point clouds bounding box(in m).

Default Value：0.0030

Suggested Value：To set according to the actual scenarios.

expand bounding box

Instruction：This parameter is used to set the expanding size of the bounding box in every direction.

Default Value：expandSizeRatioInX=1.0, expandSizeRatioInY=1.0, expandSizeRatioInZ=1.0.

Suggested Value：To set according to the actual scenarios.

maskClosingKernelSize

Instruction：This parameter is used to adjust the size of mask-closing kernel(in pixel).

Default Value：3

Suggested Value：To set according to the actual scenarios.

depthImgDilateKernelSize

Instruction：This parameter is used to adjust the size of dilating kernel of depth images(in pixel).

Default Value：4

Suggested Value：To set according to the actual scenarios.

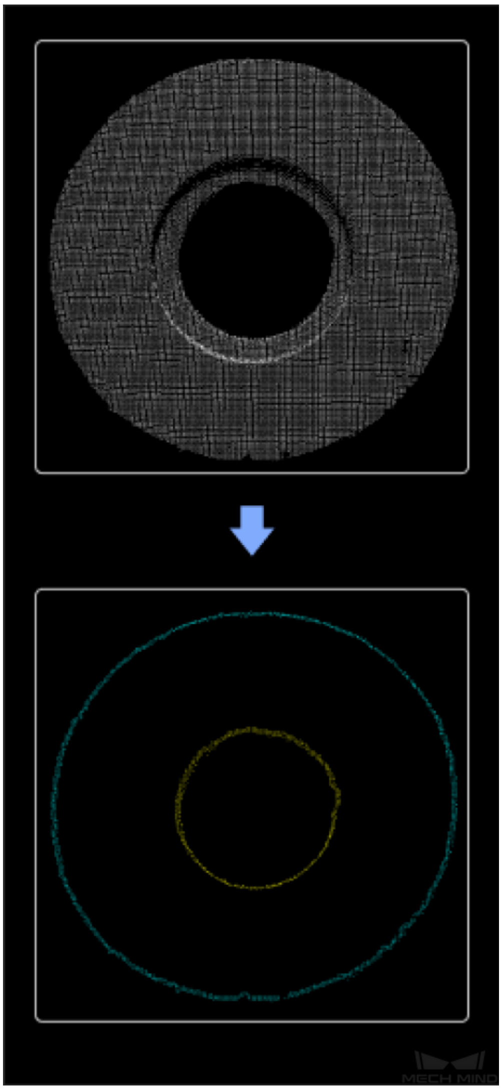
4.10 3D Feature Detector

In this chapter, steps that are associated with 3D Feature Detector will be introduced.

4.10.1 Estimate Point Cloud Edges by 2D Method

Function

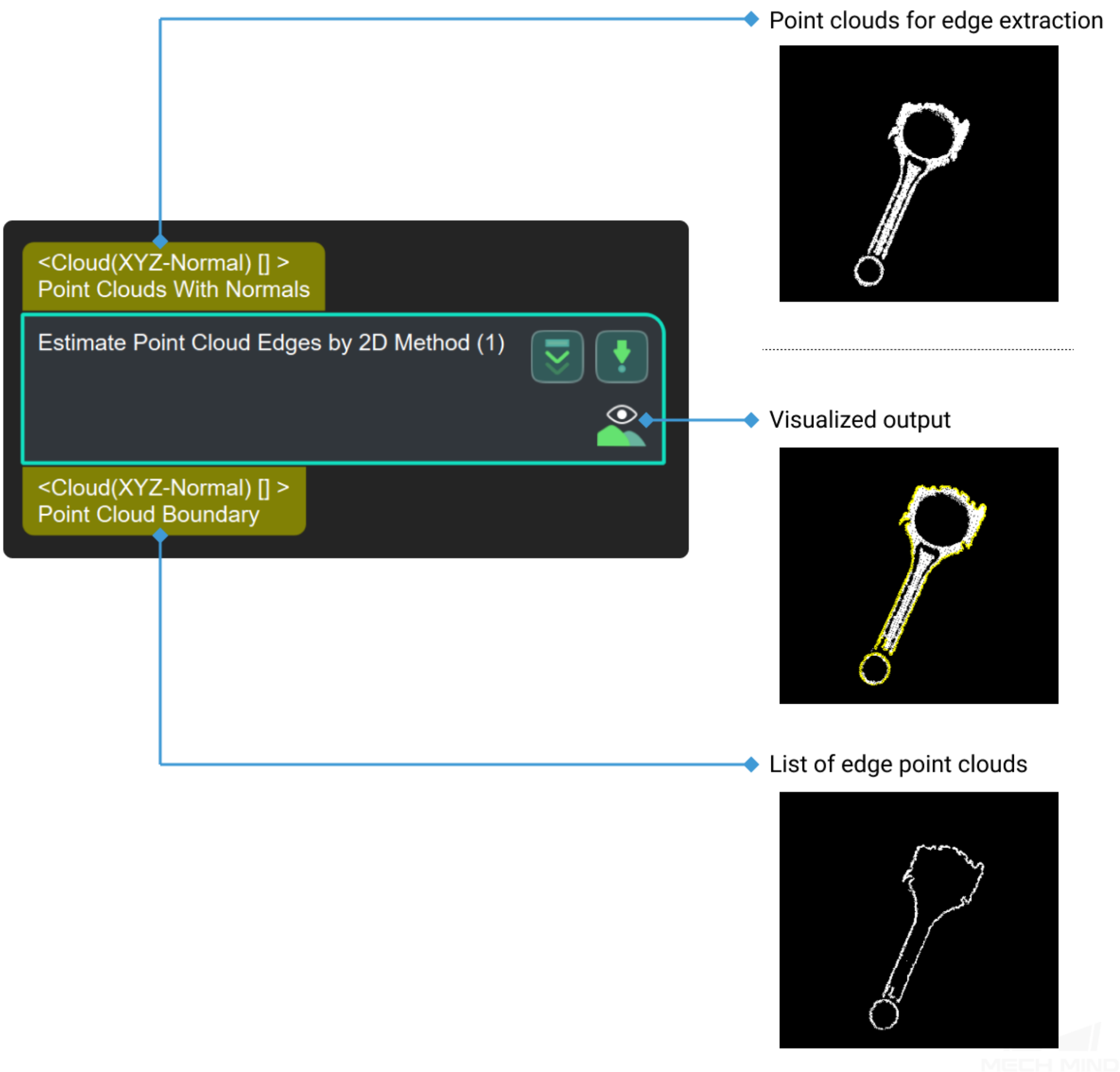
Extract the edges from input point clouds.



Sample Scenario

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



Parameters

Contour Settings

2D Line Width

Default Value: 3

Instruction: This parameter is used to adjust the line size of the profile. When the value is increased, the extracted contour becomes thicker. When the value is reduced, the extracted contour becomes thinner.

Contour Retrieval Mode

Default Value: CV_RETR_EXTERNAL

List of Values: CV_RETR_EXTERNAL, CV_RETR_LIST, CV_RETR_CCOMP, CV_RETR_TREE, CV_RETR_FLOODFILL

Instruction:

- **CV_RETR_EXTERNAL**: only detects the contours and neglects inner holes.
- **CV_RETR_LIST**: detects all contours but does not establish mutual inclusion relation.
- **CV_RETR_TREE**: detects all contours and establish mutual inclusion relation.
- **CV_RETR_CCOMP**: detects all contours and establish mutual inclusion relation of two layers.

Min Length

Default Value: 20

Instruction: This parameter provides the contour filtering function. When the number of pixels contained in the extracted contour is smaller than the value, the contour will be filtered out. When the number is larger than the value, the contour will be displayed.

Merge Contours in Same Point Cloud

Default Value: False

List of Values: True, False

Instruction: This parameter determines whether all profiles belonging to the same input point cloud will be merged together.

Morphological Operation Settings

Dilation Kernel Size

Default Value: 3

Instruction: This parameter is used to control the expansion size used by the expansion algorithm in the process of filling holes. When this parameter is increased, the filling effect of hole caused by expansion effect is obvious.

Erosion Kernel Size

Default Value: 3

Instruction: This parameter is used to control the size of the corrosion dimension used when the corrosion algorithm is used in the process of filling holes. When this parameter is increased, the hole enlargement caused by corrosion effect is obvious.

4.10.2 Estimate Point Cloud Edges by 3D Method

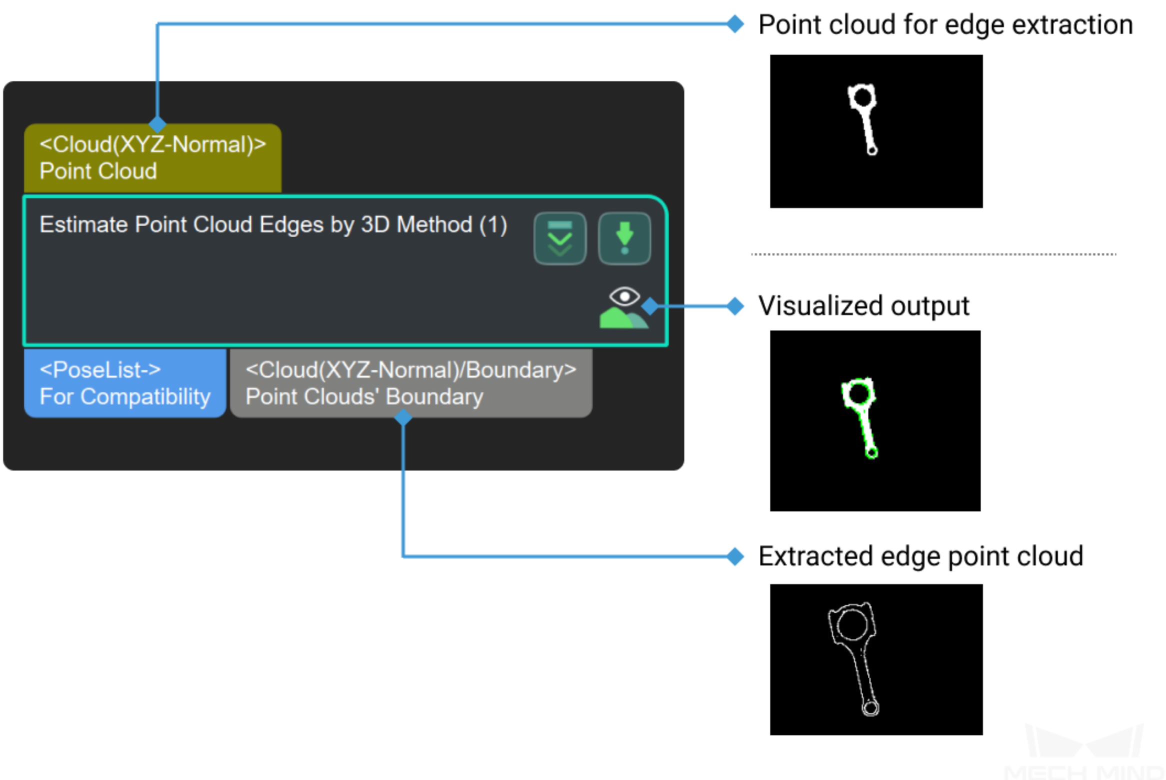
Function

Extract the edges of a 3D point cloud.

Sample Scenario

Usually used for matching based on point cloud edges, which is faster than matching based on full point clouds.

Input and Output



Parameters

Boundary Estimations Settings

Angle Threshold

Default Value: 90

Instruction: This parameter determines whether the point is identified as a boundary point. When the boundary is relatively flat, it is recommended that the value be set to a smaller value. When the boundary is sharp, it is recommended that the value be set to a larger value. The value ranges from 0 to 360.

Neighbour Search Settings

Nearest Neighbour Search Type

Default Value: KNN

List of Values: KNN, Radius

Instruction: Search for adjacent points by selecting different ways.

- **KNN**: The search of adjacent points is related to the number of adjacent points.
- **Radius**: The proximity search is related to the search radius.

Search Number

Default Value: 20

Instruction: This parameter is used to control the size of searche numbers in the process of obtaining the boundary. The search mode takes effect when **KNN** is selected in **Nearest Neighbour Search Type**. When the value is increased, part of contour details will be ignored and the overall recognition effect is good. When the value is set to small, subtle boundary details are displayed, some of them are invalid boundary.

Search Radius

Default Value: 0.0050

Instruction: This parameter determines the size of the search radius in the process of obtaining the boundary. The search mode takes effect when Radius is selected in **Nearest Neighbour Search Type**. When the value is increased, the search Radius becomes larger, resulting in fewer detected contours but better overall recognition effect. When the value is set to small, the search radius becomes smaller and subtle contour information can be detected.

Speed Up

Apply GPU

Default Value: False

List of Values: True, False

Instruction: This parameter controls whether to enable GPU acceleration.

4.10.3 Point Cloud Shape Detector

Function

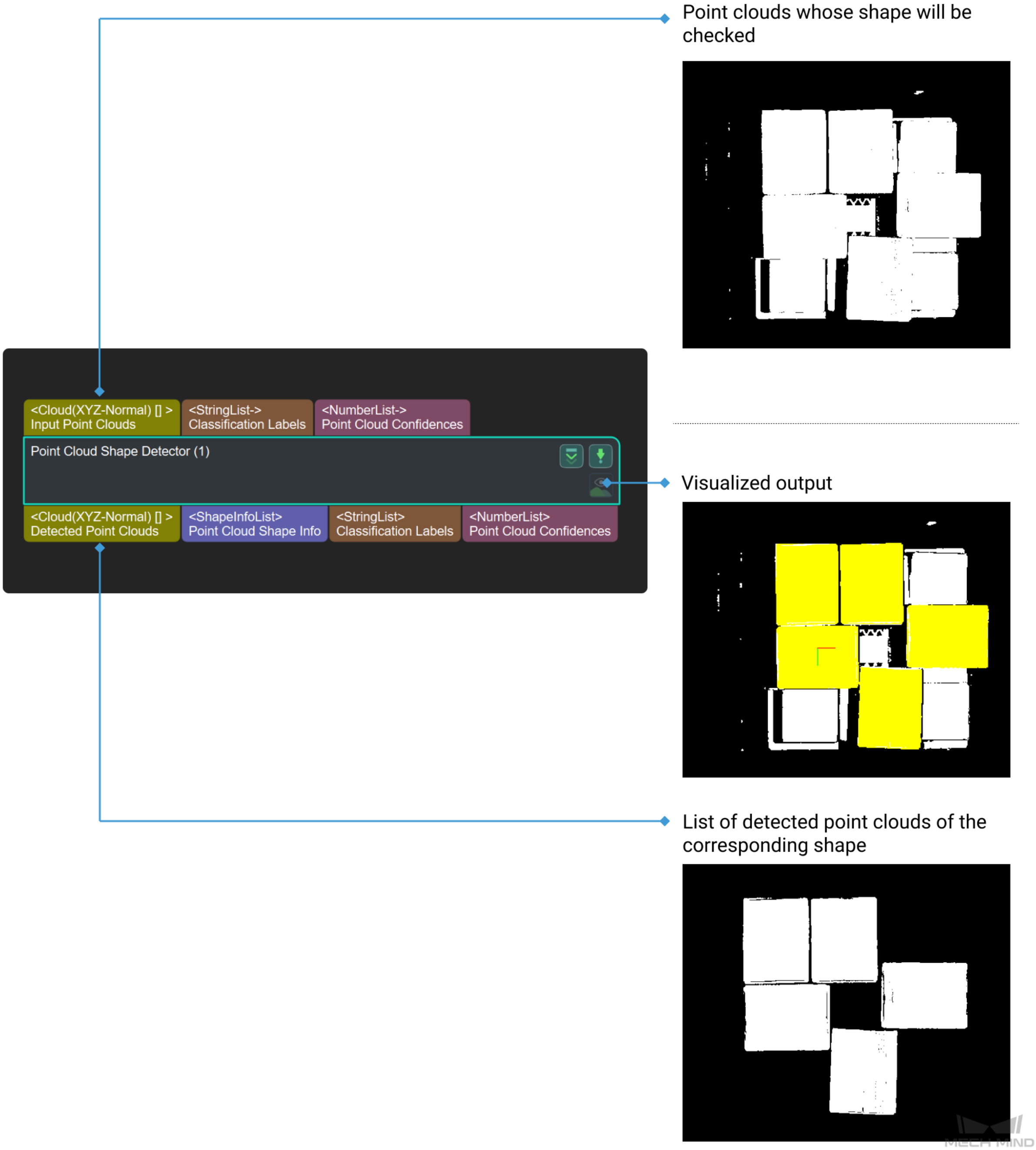
Detect and output point clouds of a plane or cylindrical shape; the shape can be selected in the **Step Parameters**.

Sample Scenario

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



Parameters of Input and Output

Cloud(Normal)[] [Input]inputted clouds

StringList- [Input]the labels of inputted clouds

DoubleList- [Input]the confidence of clouds

Cloud(Normal)[] [Output]detected clouds of primitive shapes

ShapeInfoList [Output]information of detected shapes

StringList [Output]the labels of corresponding outputted clouds

DoubleList [Output]confidence of corresponding clouds

Parameters

Shape Type

Instruction: This parameter is used to choose the target shape type, plane, cylinder and sphere included.

Default Value：plane

Suggested Value：To set according to the actual scenarios.

Detection Property

detector

Instruction：This parameter is used to choose which kind of detector used in the detection process, ALGO1 and ALGO2 included.

Default Value：ALGO1

Suggested Value：ALGO1

normalDeviation

Instruction：This parameter is used to set the upper limit of the angle difference(in °) between the normal of points in the image and the normal of projected theoretically corresponding standard shape.

Default Value：30

Suggested Value：To set according to the actual scenarios.

epsilon

Instruction: This parameter is the upper limit(in m) of the acceptable range of the point cloud fluctuation error.

Default Value：0.0050.

Suggested Value：[0.0030,0.0050]

clusterEpsilon

Instruction: This parameter is the lower limit(in m) of the interval of point cloud connectivity judgement.

Default Value: 0.0050

Suggested Value: To set according to the actual scenarios.

cloudResolution

Instruction: This parameter is used to control the number and intensity of points. Greater number and intensity correaponds to higher accuracy and longer computing time.

Default Value: 0.0050

Suggested Value: Usually twice the value of *clusterEpsilon*.

candidateTopNum

Instruction: This parameter determines the number of final outputted result for every point cloud.

Default Value: 1

Suggested Value: To set according to the actual scenarios.

sortBySize

Instruction: This parameter decides whether to sort the final result according to the size of point cloud or not.

Default Value: True

Suggested Value: To set according to the actual scenarios.

successProbability

Instruction: This parameter is used to adjust the success rate of detection.

Default Value: 0.99000

Suggested Value: To set according to the actual scenarios.

Candidates Filter

minCoverageRatio

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.

Default Value：0.1000

Suggested Value：To set according to the actual scenarios.

minMeanDistError

Instruction：This parameter is used to set the lower limit of the mean distance deviation(in m) between real points and the points of corresponding detected theoretical shape.

Default Value：0.0100

Suggested Value：To set according to the actual scenarios.

minMeanAmgleError

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.

Default Value：45.0000

Suggested Value：To set according to the actual scenarios.

Cylinder Property

minRadius

Instruction：This parameter is used to set the lower limit of the cylinder(in m).

Default Value：0.0100

Suggested Value：To set according to the actual scenarios.

MaxRadius

Instruction：This parameter is used to set the upper limit of the cylinder(in m).

Default Value：0.1000

Suggested Value：To set according to the actual scenarios.

ifOptimizeCylinderCoefficients

Instruction: This parameter decides whether to optimize the shape paramaters of cylinder, center and axis included.

Default Value: False

Suggested Value:

List of Values: True,False

4.11 3D General Processing

In this chapter, steps that are associated with 3D General Processing will be introduced.

4.11.1 Calc Pose

Calc Cylinder Poses

Description

This step could calculate the pose of cylinder center.

Parameters of Input and Output

Cloud(Normal)[] [Input]list of cylinder point clouds

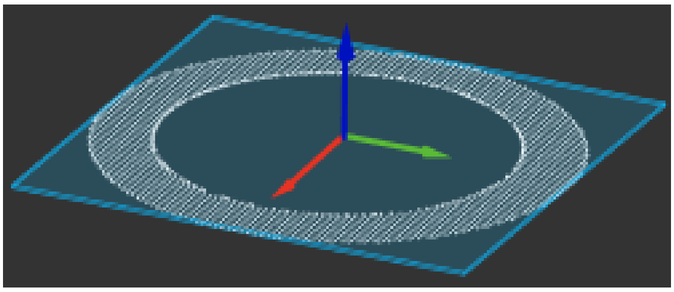
ShapeInfoList [Input]shape information

PoseList [Output]the poses of cylinders' centers

Calc Poses and Dimensions from Planar Point Clouds

Function

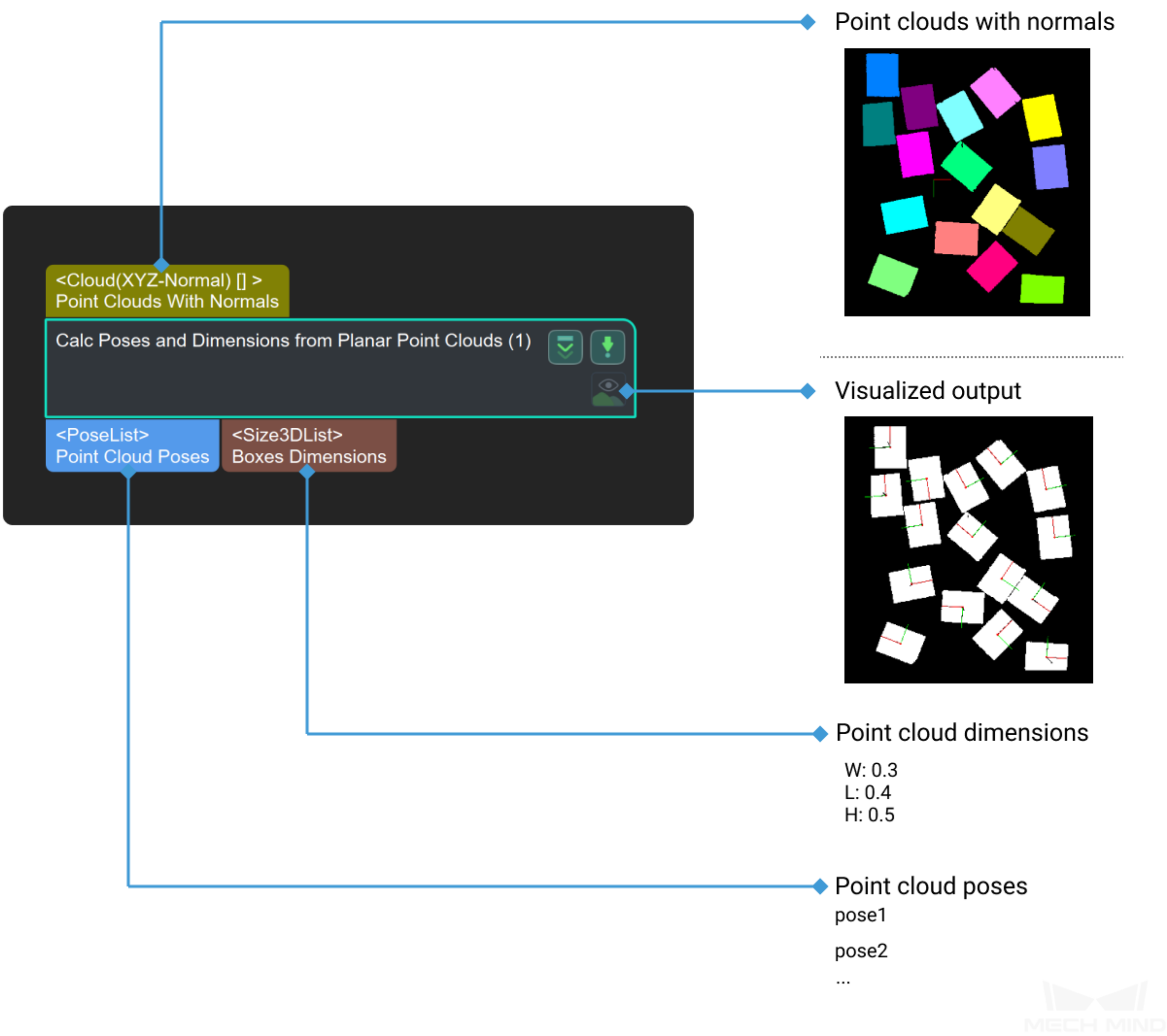
Calculate the poses of planar point clouds and the dimensions of the point cloud bounding boxes.



Sample Scenario

This Step is usually used to calculate pick points and dimensions of planar point clouds of objects as cartons, sacks, etc.

Input and Output



Parameters

Pose Combination

The method for calculating the position and pose of gripping point

translationType

Instruction: This parameter is used to choose the translation type.' BoundingRectCenter' represents the center of bounding box and 'CloudCenter' represents the geometric center of point cloud.

Default Value: BoundingRectCenter

Suggested Value: BoundingRectCenter

List of Values: BoundingRectCenter, CloudCenter

rotationType

Instruction: This parameter is used to choose the translation type used to choose the rotation type. When the 'BoundingRectOrientation' is selected, the calculation of gripping poses will be based on the poses of bounding boxes while ‘CloudOrientation’ based on the direction of the main element of the point cloud.

Default Value：BoundingRectOrientation

Suggested Value：BoundingRectOrientation

List of Values: BoundingRectOrientation, CloudOrientation

boxZLen

The property determines that whether to consider the height difference between the bottom of objects and the current coordinate origin or not.

calcBoxZlen

Instruction：This parameter decides whether to calculate the height of the box or not.

Default Value：False

Suggested Value: To set according to the actual scenarios.

List of Values: True, False

useMaxDistAlongSpecifiedDir

Instruction：This parameter decides whether to use the maximum distance in specified direction or not.

Default Value：False

Suggested Value: To set according to the actual scenarios.

List of Values: True, False

backgroundHeight

Instruction：This parameter is used to set the background height(in m) in the coordinate of industrial robot in order to calculate the length of the box in Z direction. This parameter is valid when *calcBoxZlen* is ‘True’.

Default Value：0.0

Suggested Value: To set according to the real background height.

specifiedDir

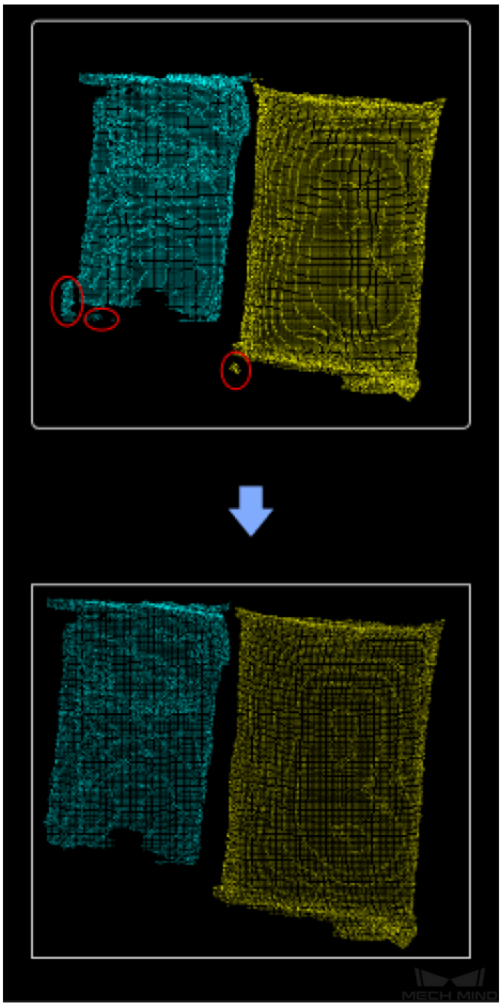
Instruction：This parameter is used to specify the direction and only valid when *useMaxDistAlongSpecifiedDir* is ‘True’. | Default Value：x=0;y=0;z=0。| Suggested Value：To set according to the actual scenarios.

4.11.2 Cluster

Cluster Point Clouds and Output Eligible Point Clouds

Function

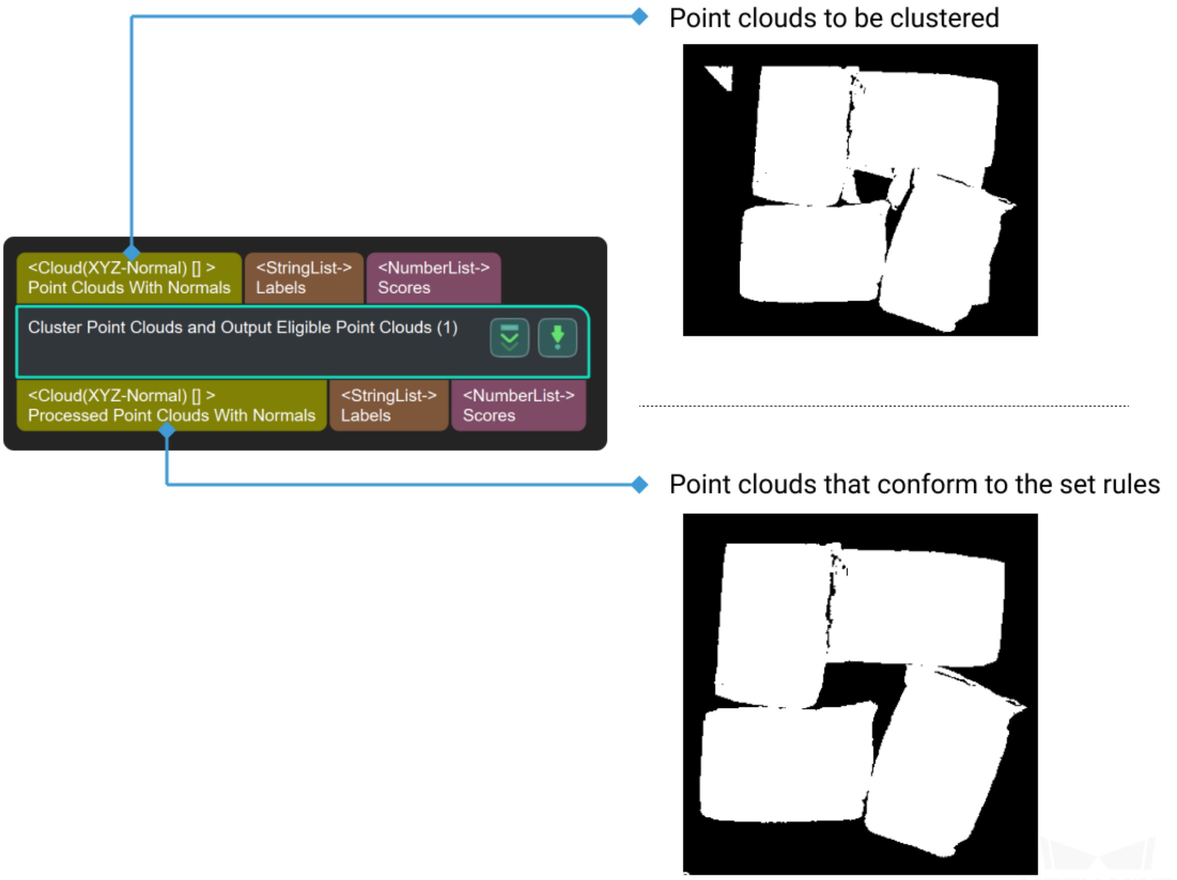
Perform clustering operation on point clouds, and output the point clouds that conform to the set rules in the clustering result.



Sample Scenario

This Step is usually used to denoise point clouds of individual objects.

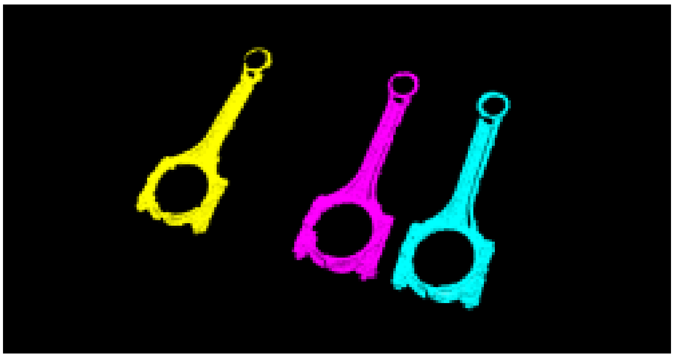
Input and Output



Point Cloud Clustering

Function

Perform point cloud clustering according to specified rules; often used for segmentation of target objects.



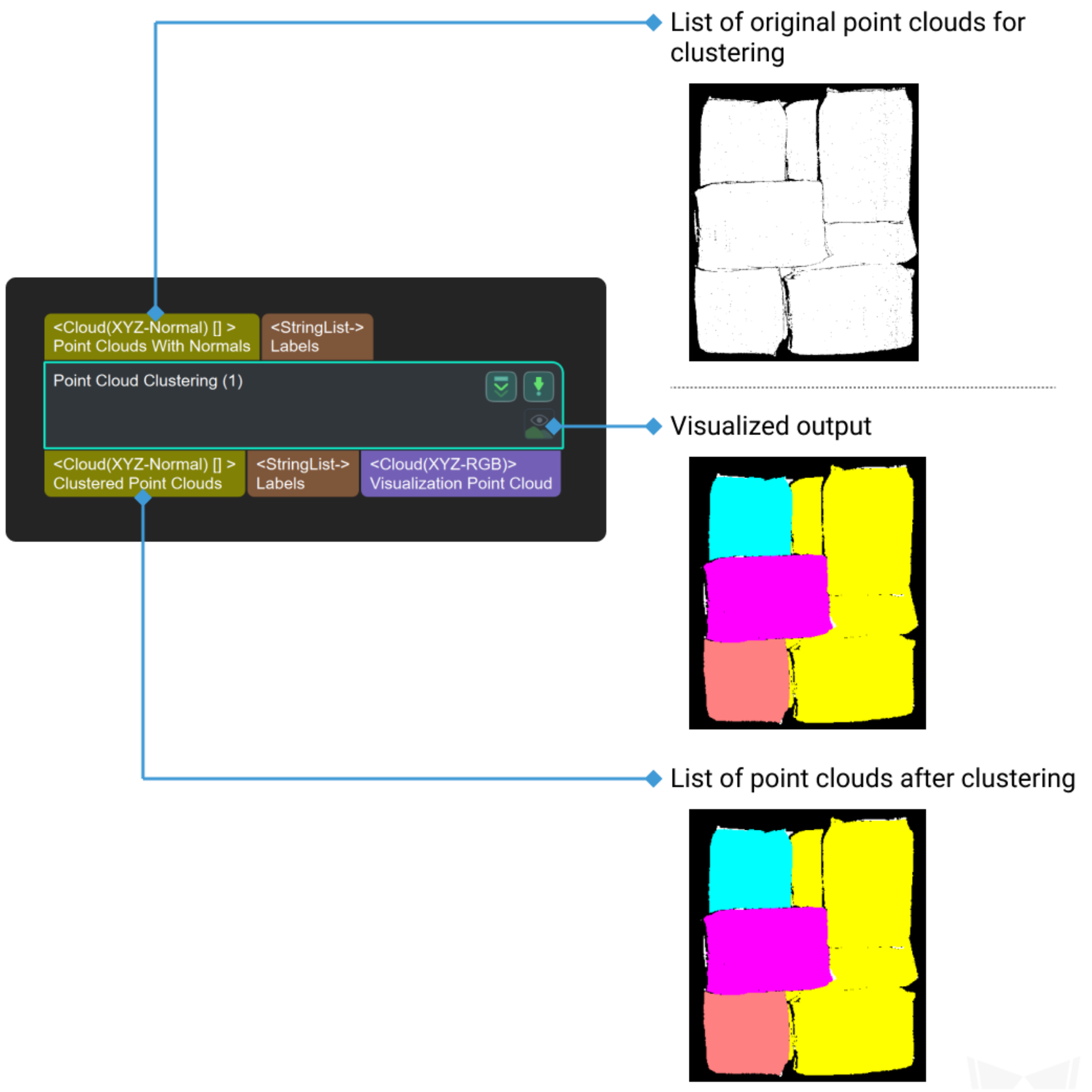
Sample Scenario

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



Parameters

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

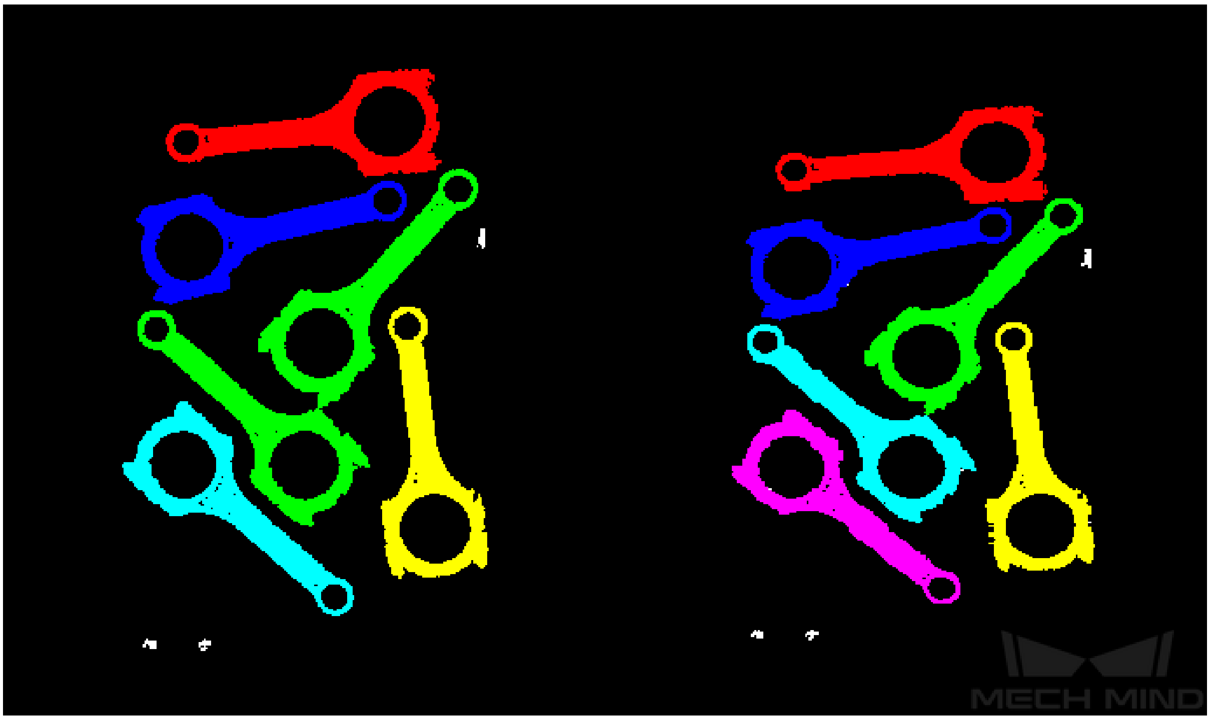
EuclideanCluster

Max Distance of Neighboring Points in Output Clusters

Default value: 0.0030

Instruction: This parameter is the clustering tolerance. 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 below, the left panel shows the result of this parameter at the default value of `0.003` and the right panel shows the result of this parameter adjusted to `0.005`. The user can see that after the adjustment, the green point cloud in the middle is divided into different categories.



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, resulting 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 neighbouring 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

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: 3,000,000 | 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 5 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.

GPU

Default Value: False
List of Values: True, False
Instruction: This parameter controls whether to enable **GPU** acceleration.

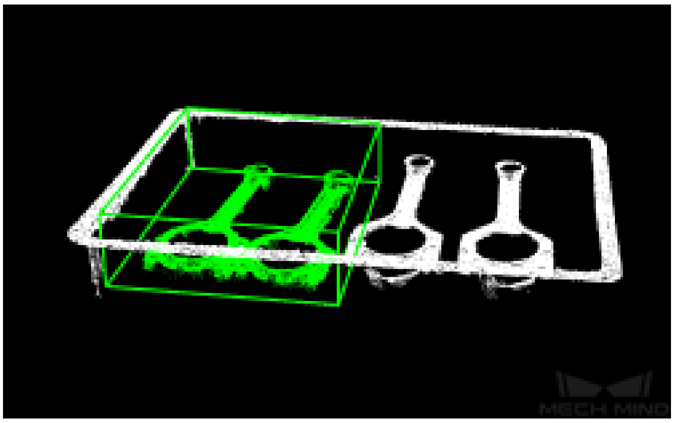
4.11.3 Extract Points by Geometry

Clouds in 3D Box

Extract 3D Points in 3D ROI

Function

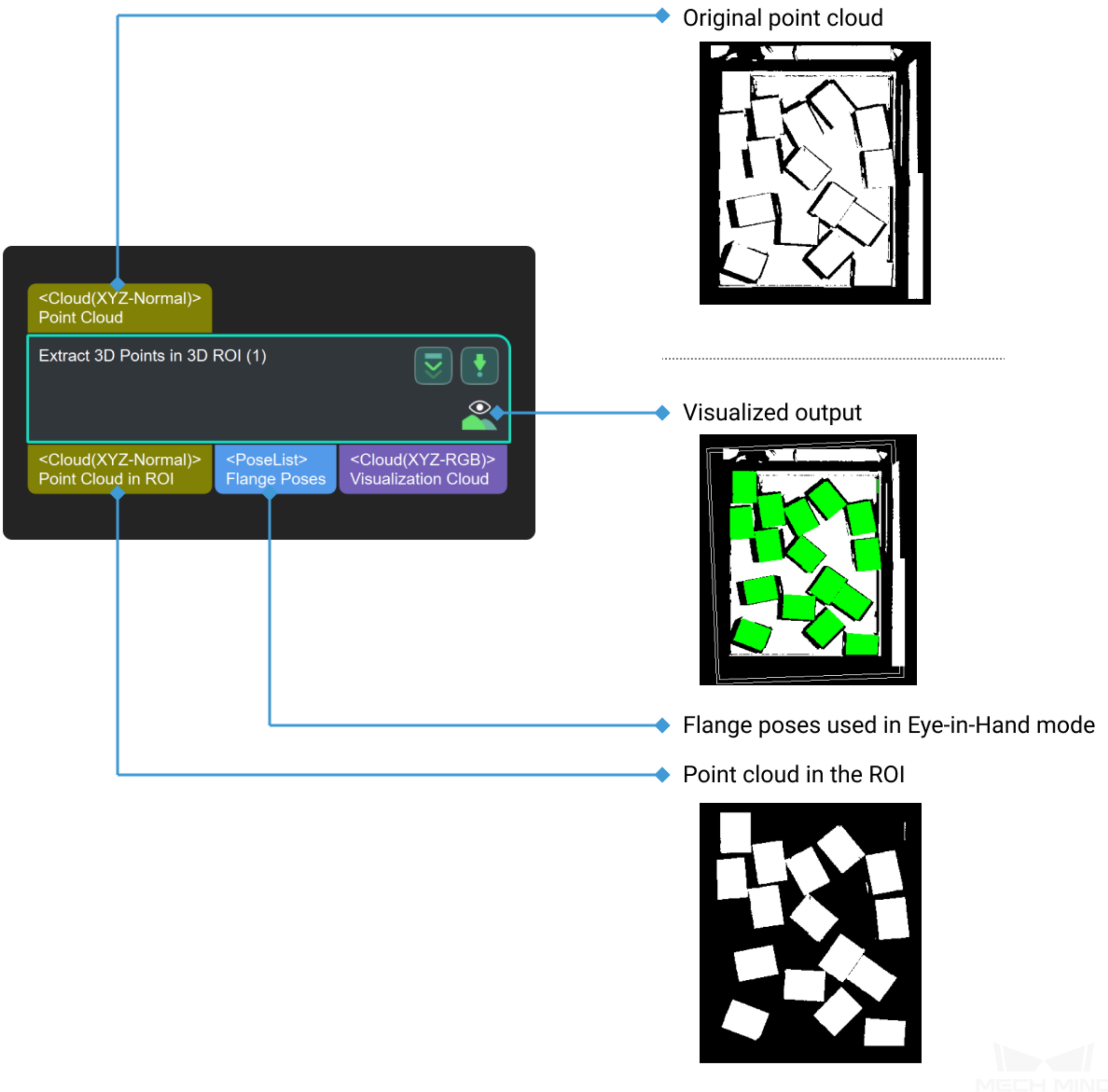
Set a Region of Interest (ROI) in 3D space, and the points within the region will be extracted and points outside the region will be eliminated.



Sample Scenario

This Step enables to extract point cloud of the target object and avoid interference from background and outliers.

Input and Output



Preparation

Before setting a 3D ROI, please go to **Project Assistant** and select the data source of scene point cloud. For detailed settings, please refer to *Scene Point Cloud*.

Parameters

3D ROI Settings

3D ROI Name Instruction: Click ✎ on the right side of the 3D ROI Name to enter the **Set ROI** window, as shown in *Figure 1*. For detailed settings, please refer to *Instructions for Setting 3D ROI*.

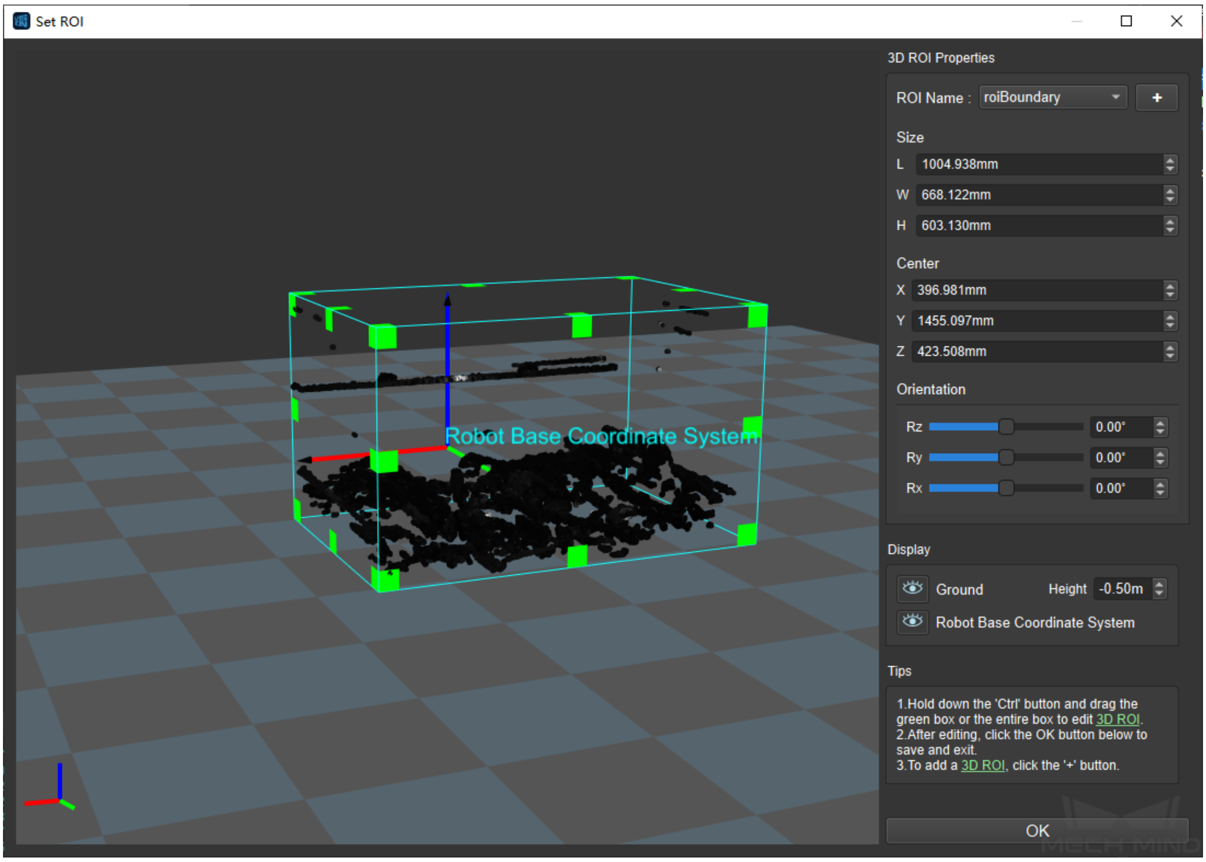


Figure1 Set Target Area

Input Cloud Coordinate Type

Default Value: Camera Coordinate

List of Values: Camera Coordinate, Robot Coordinate

Instruction: Change the coordinate system of the input point cloud. As shown in *Figure 2*, the left side is the input point cloud in the default state, and the right side is the input point cloud in the robot coordinate system.

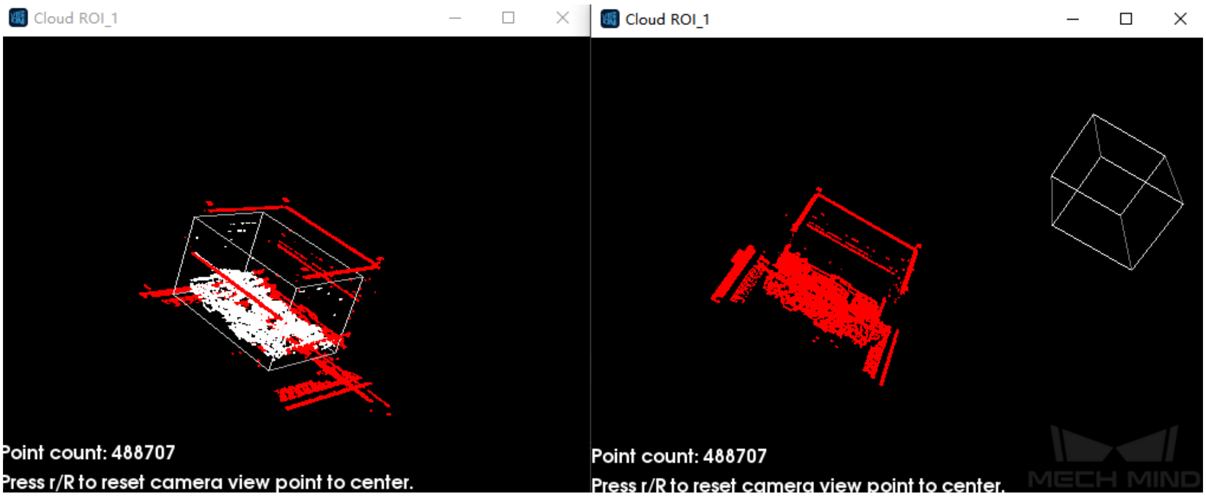


Figure2 Input Cloud Coordinate

Empty Status Settings

Min Points Number in 3D ROI

Default Value: 0

Instruction: This parameter is the lower limit of the number of the points in the 3D ROI. If the points in the 3D ROI are less than this value, it means that there is no point cloud that meets the requirements in the 3D ROI.

Send Empty Status of Points in 3D ROI

Default Value: True

List of Values: True, False

Instruction: This parameter determines whether to send the presence or absence information of the point cloud in the area of interest. In the default state, if the actual points in the 3D ROI are less than the value set in **Minimum Points in 3D ROI** , a prompt like *Figure 3* will be generated.

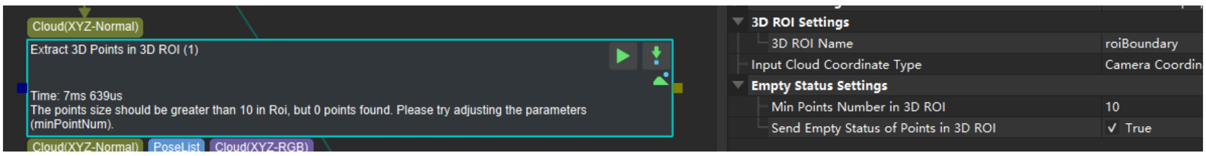


Figure3 Prompt When Set to True

If set to False, there will be no prompt, as shown in *Figure 4*.

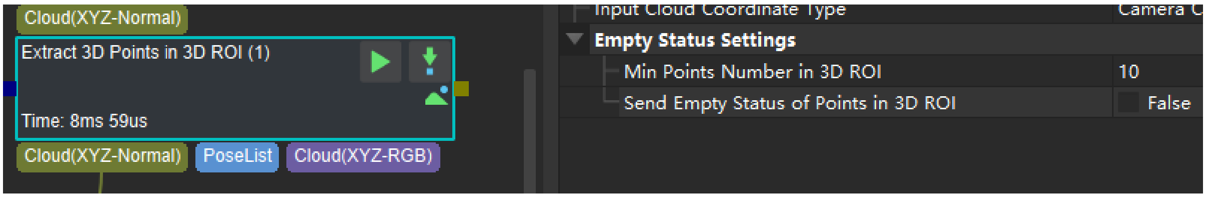


Figure4 Prompt When Set to False

Possible Preceding Step

Image Brightness and Color Balancer

Apply Masks to Point Cloud

Calc Normals of Point Cloud and Filter It

Possible Succeeding Step

Show Point Clouds and Poses

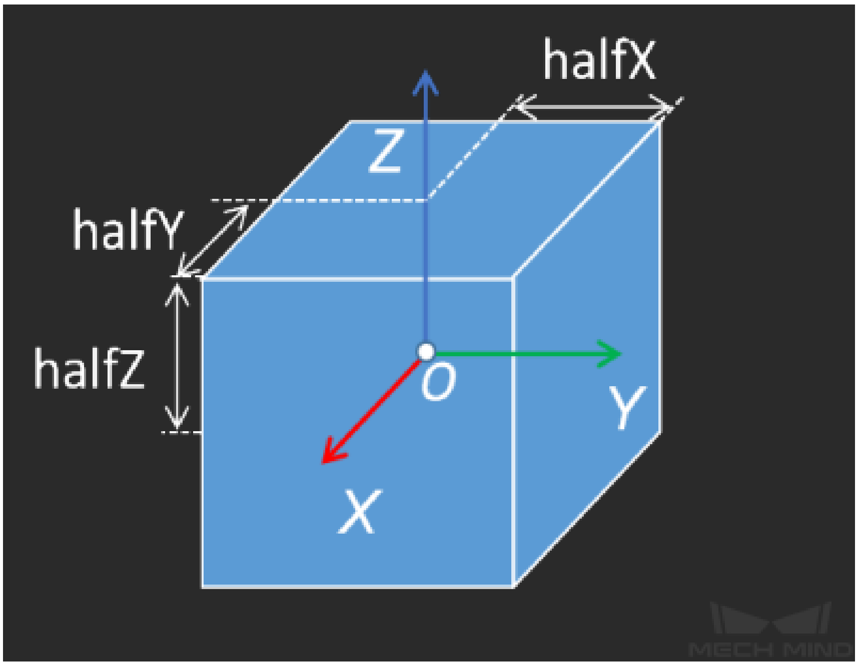
Merge Point Clouds

Apply Masks to Point Cloud

Extract 3D Points in Cuboid

Function

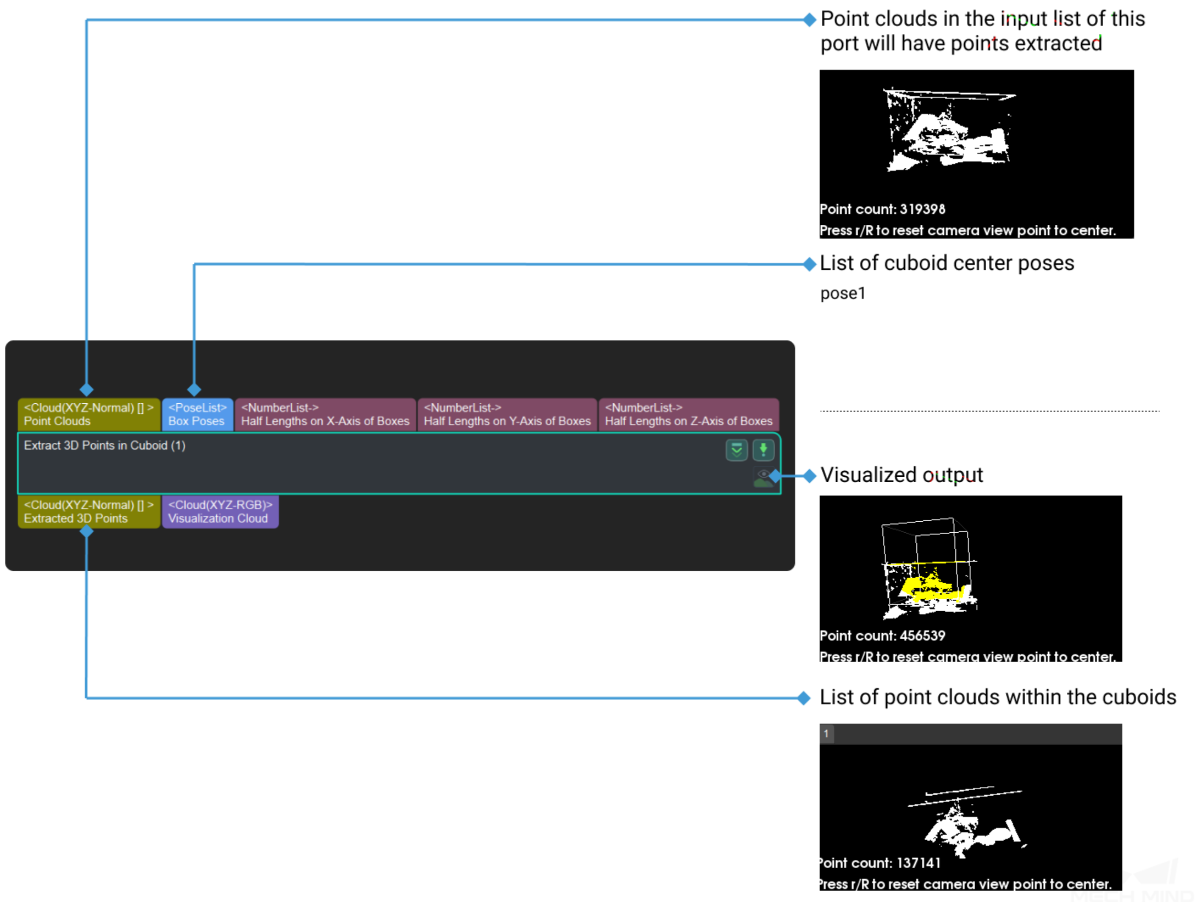
Extract point clouds inside specified cuboids, and discard the point clouds outside the cuboids.



Sample 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.

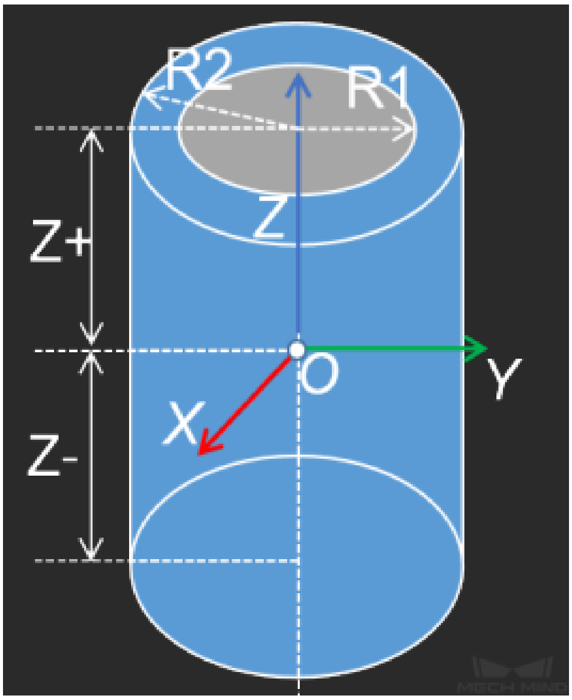
Input and Output



Extract 3D Points in Cylinder

Function

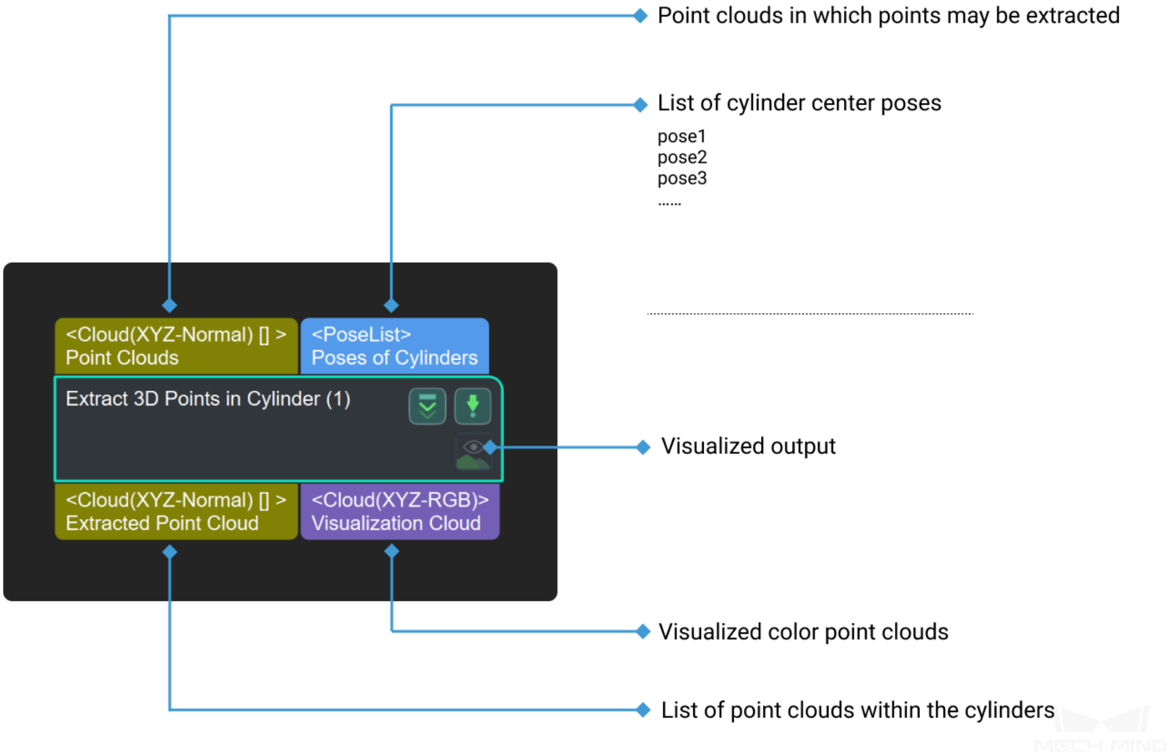
Extract point clouds inside specified cylinders, the point clouds outside the cylinders will be discarded.



Sample Scenario

A general point cloud extraction Step. No fixed usage scenarios.

Input and Output



Parameters

Shape Type

Default Value: Ring

List of Values: Cylinder, Ring, Sector

Instruction: 3D points inside this specified shape will be extracted.

Z Direction Properties

Min Z Value

Default Value: -0.1 m

Instruction: This parameter is used to set the lower limit of distance threshold (in m) along Z axis of the reference coordinate, which corresponds to Z- in *Figure 1*.

Max Z Value

Default Value: 0.1 m

Instruction: This parameter is used to set the upper limit of distance threshold (in m) along Z axis of the reference coordinate, which corresponds to Z+ in *Figure 1*.

Ring and Sector Radius Properties

Inner Radius

Default Value: 0.05 m

Instruction: The minimum distance on XY-plane, which corresponds to R1 in *Figure 1*. To obtain a solid cylinder, this parameter needs to be set as `0` .

Outer Radius

Default Value: 0.1 m

Instruction: The maximum distance on XY-plane, which corresponds to R2 in *Figure 1*.

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: False

List of Values: True, False

Instruction: If checked, the center of the object point cloud is used to check if it is inside the cylinder.

Input Point Cloud Usage

Default Value: All

List of Values: All, FirstOnly

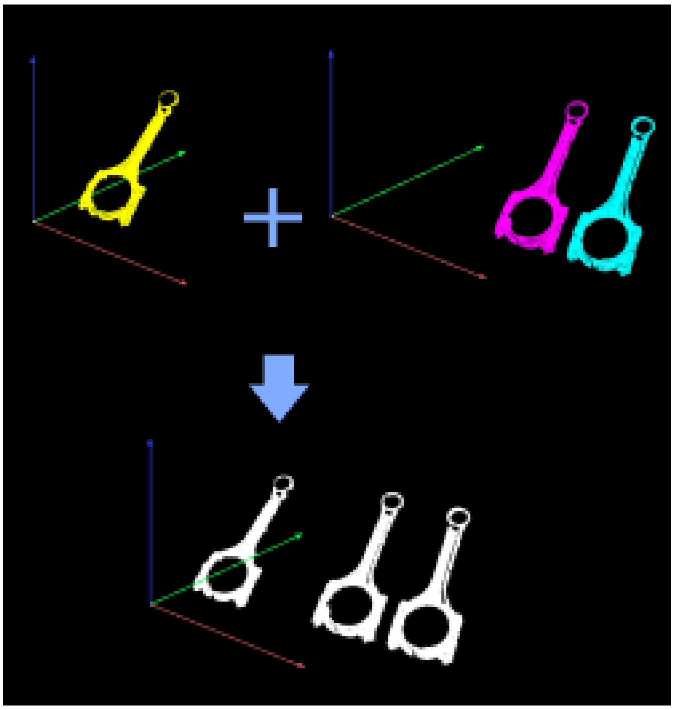
Instruction: This parameter decides whether to only use the first input point cloud.

4.11.4 Merge

Merge Point Clouds

Function

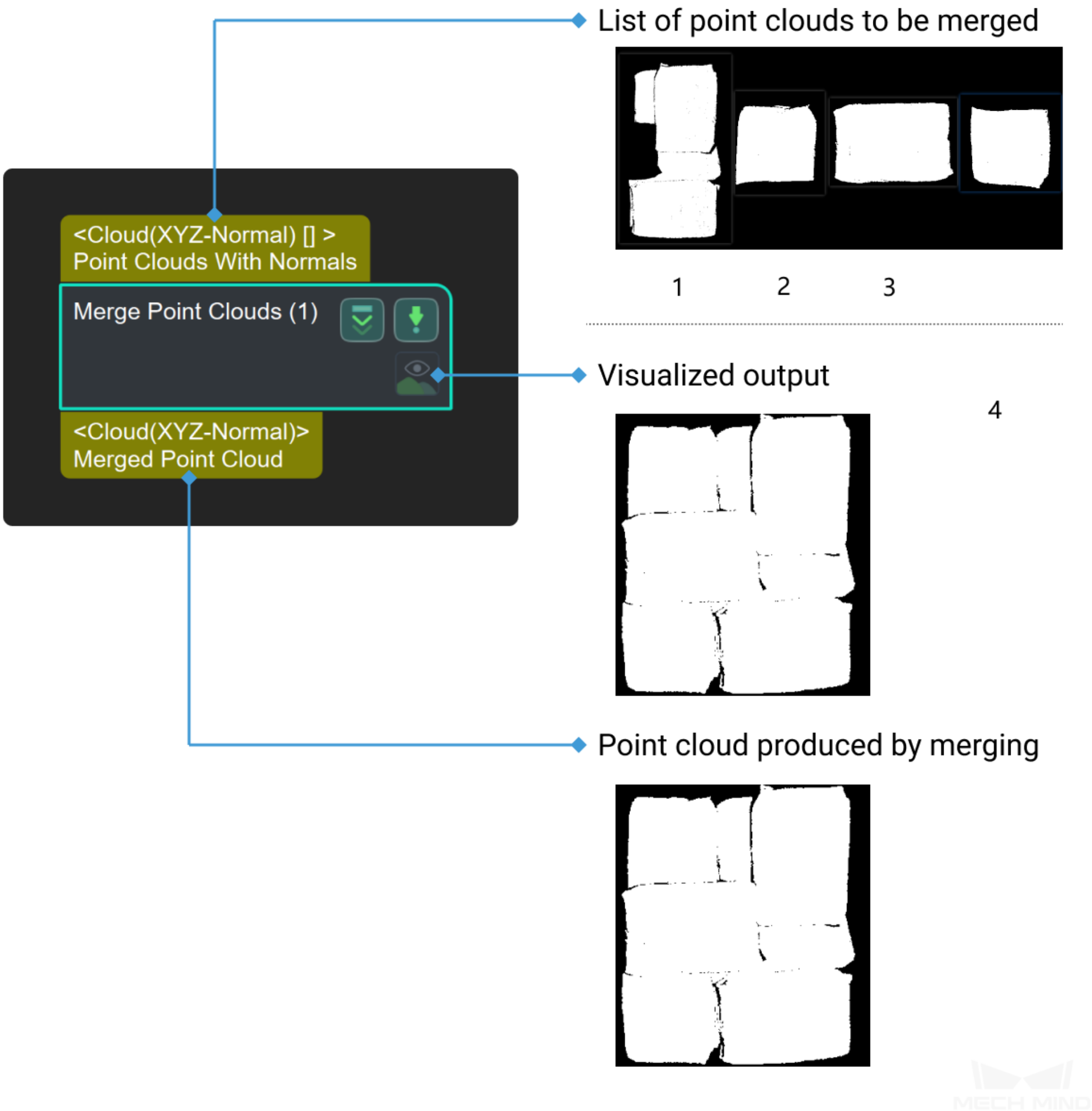
This Step merges multiple point clouds into one and outputs it.



Sample Scenario

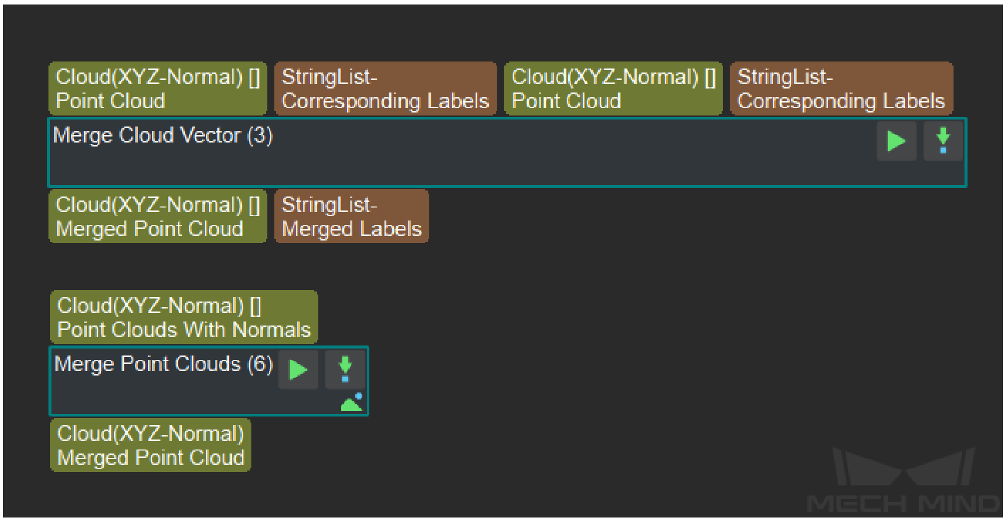
This Step usually follows Steps as *Point Cloud Clustering* and *Apply Masks to Point Cloud*. It can merge multiple point clouds in the point cloud list into a single point cloud to facilitate further processing.

Input and Output



Comparison with Similar Steps

When merging two lists of point clouds, **Merge Point Clouds** needs a preceding **Merge** Step, but **Merge Cloud Vector** can directly merge two point cloud lists into one single point cloud and merge their label lists into one at the same time.



Merge Point Clouds with similar height

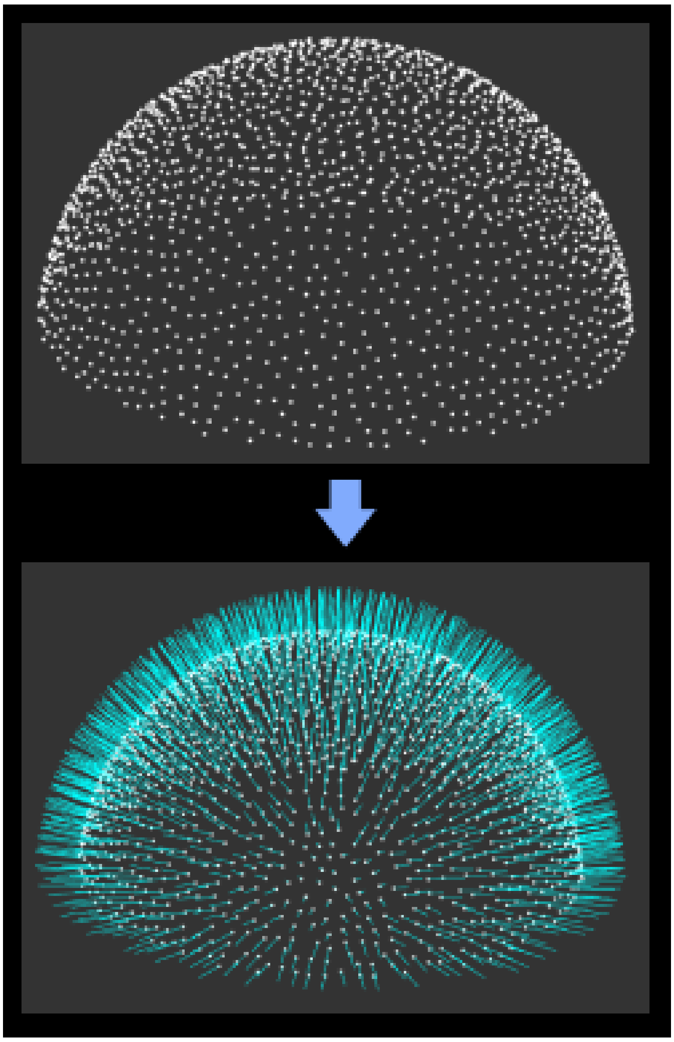
4.11.5 Normal/Curvature

Calc Edge Points Normal

Calc Normals of Point Cloud and Filter It

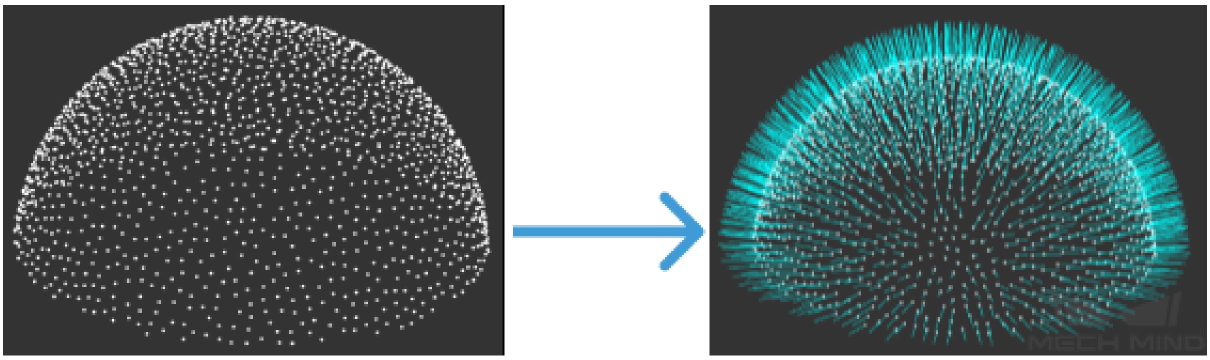
Function

Calculate point cloud normals and remove outliers.

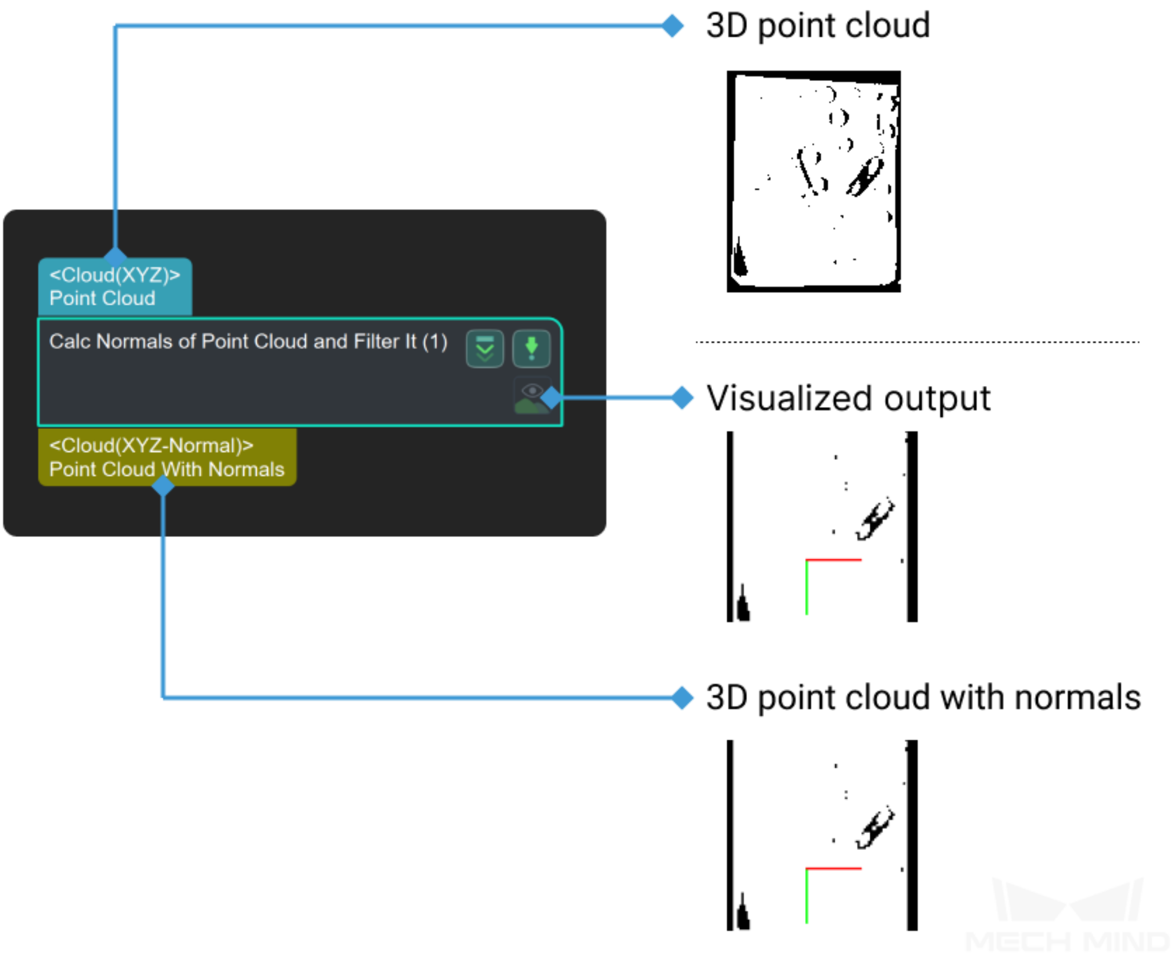


Sample Scenario

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



Calc Point Cloud Curvature

Cloud Smooth and Normal Estimation

4.11.6 Remove Points

Down-Sample Point Cloud

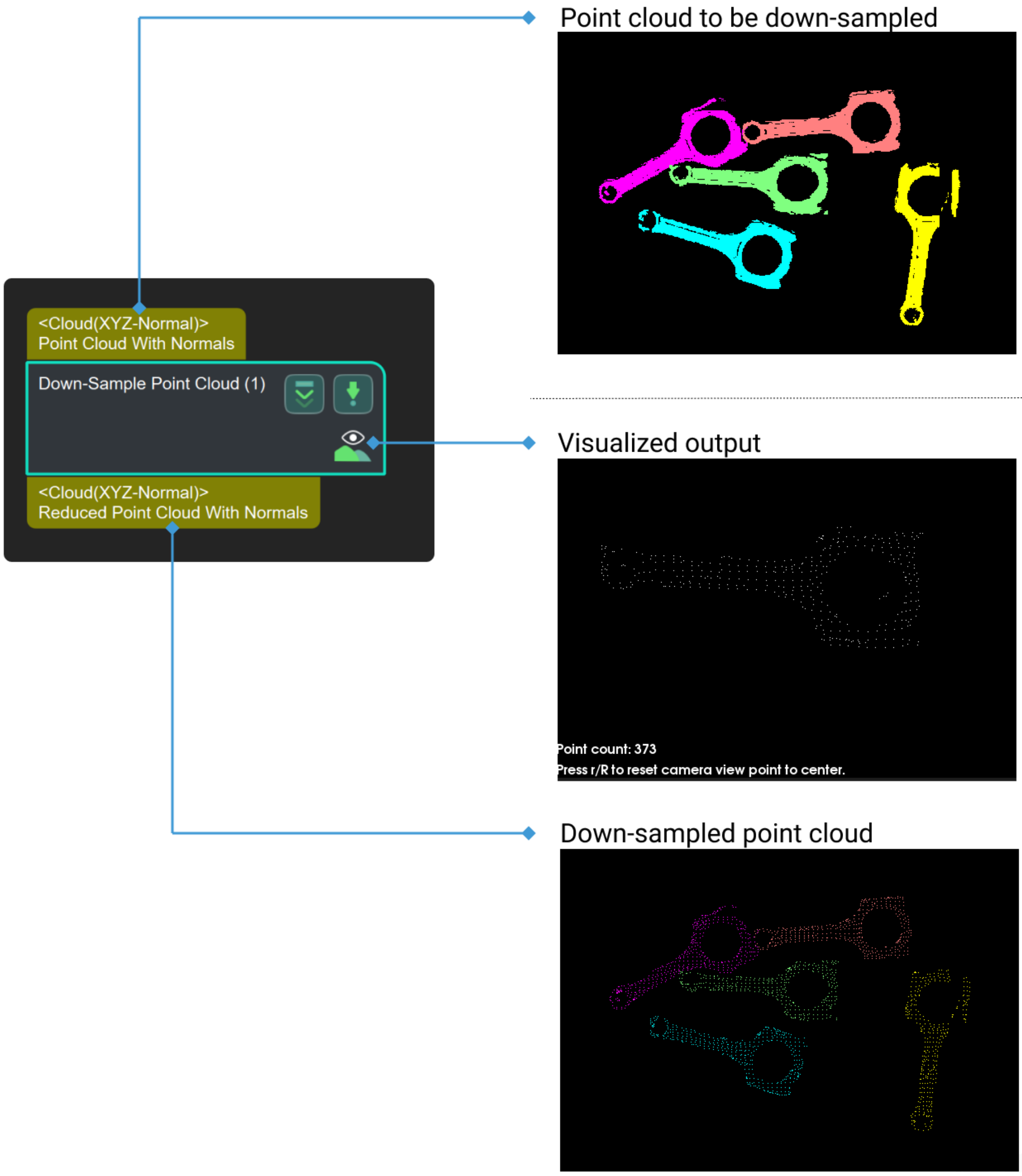
Function

Down-sample a point cloud to reduce the number of points.

Sample Scenario

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

Point Filter

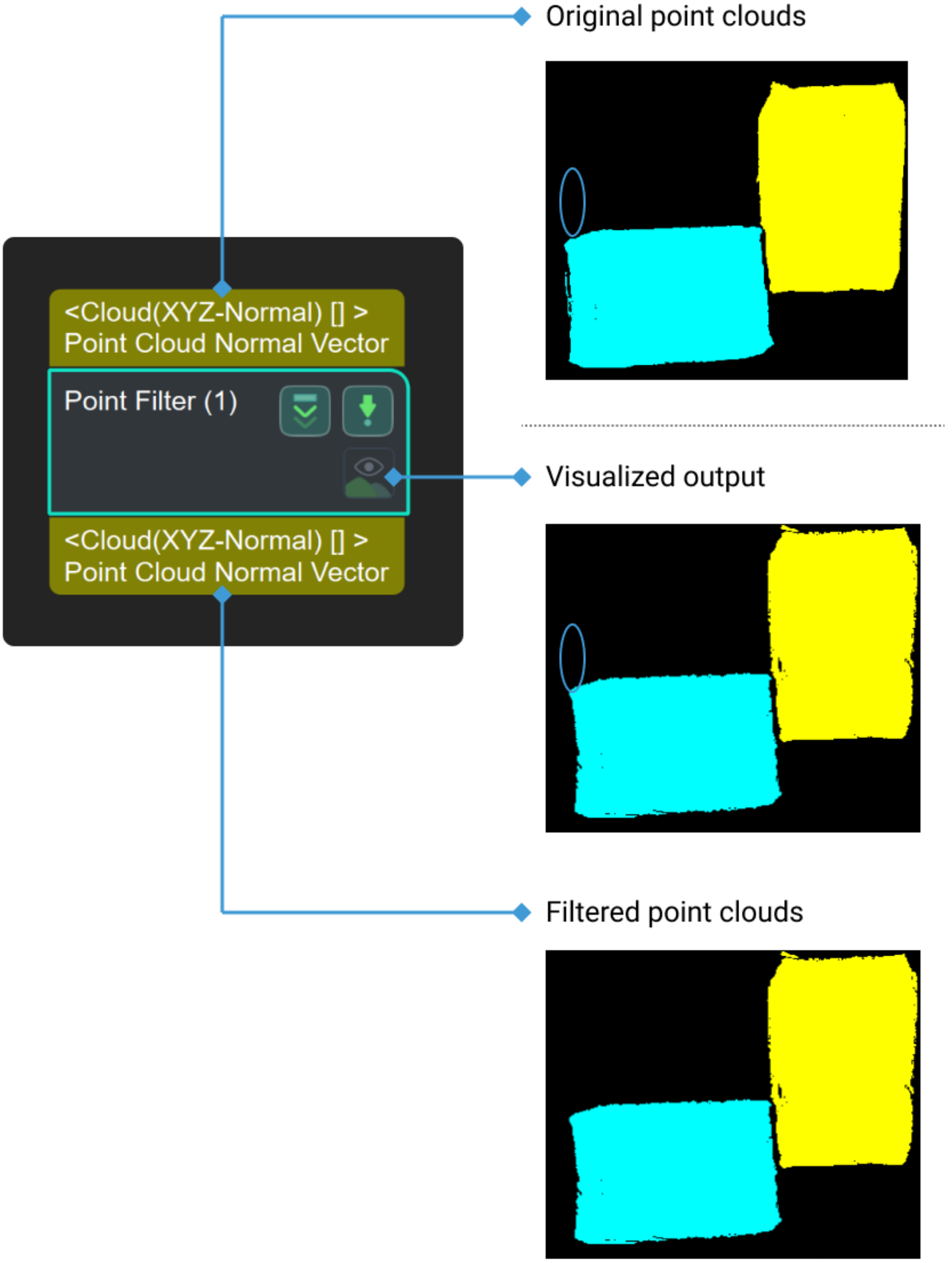
Function

Filter points based on the rules specified by the user.

Sample 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


Original point clouds
<Cloud(XYZ-Normal) [] >
Point Cloud Normal Vector
Point Filter (1)
Visualized output
<Cloud(XYZ-Normal) [] >
Point Cloud Normal Vector
Filtered point clouds

Parameters

Points Filter Type

Default Value: StatisticalOutlierFilter

List of Values: StatisticalOutlierFilter, NormalsFilter

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 in *figure 1*. This method parameters are introduced first.
- *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.Shown the red box in *figure 2*.

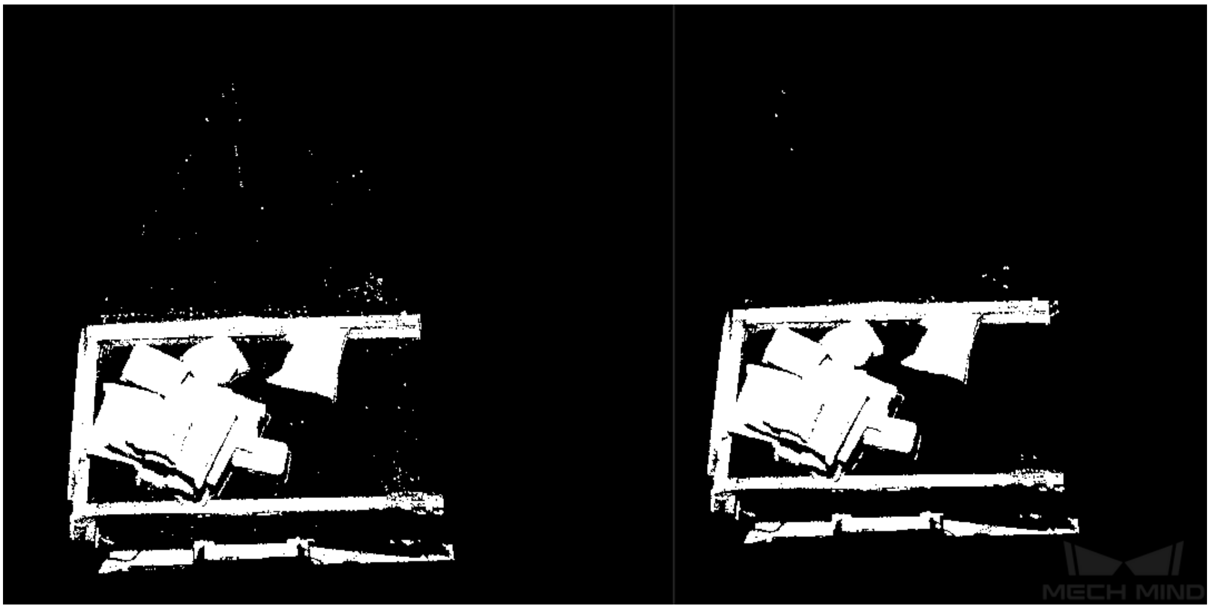


Figure1 StatisticalOutlierFilter Results Comparison

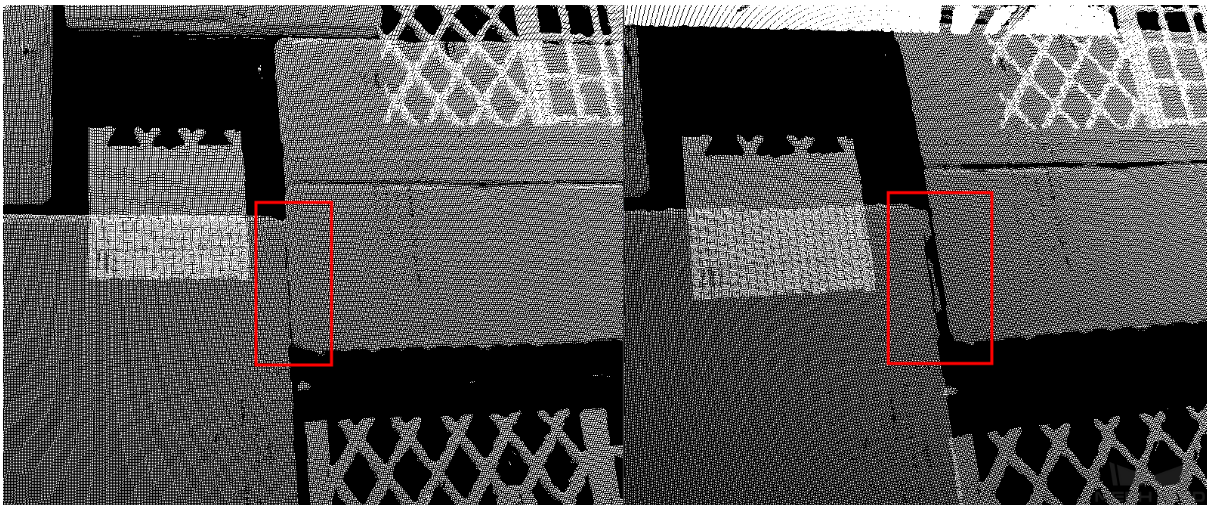


Figure2 NormalsFilter Results Comparison

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.

Stddev Mul Thresh

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.

Reference Direction

Use Robot Z in Camera

Default Value: False

List of Values: True, Flase

Instruction: This parameter decides whether to use the Z axis of robot in camera or not. If this parameter is set to `True`, **Align X Align Y** and **Align Z** is 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 set to `True`, the three parameters are invalid and the reference axis is robot Z-axis.

Align Y

Default Value: 0

Instruction: Same as **Align X** adjustment method.

Align Z

Default Value: -1

Instruction: Same as **Align X** adjustment method.

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 unit of °. 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 unit of °. When the Angle difference between a point' s normal vector and its reference vector exceeds this threshold, the point is filtered out.

Remove Cloud Points from Point Cloud

4.11.7 Reshape/Transform/Move

Cloud Distortion Correction

Cloud Scale

Move Cloud along Set Dir

Project a Point onto a Plane

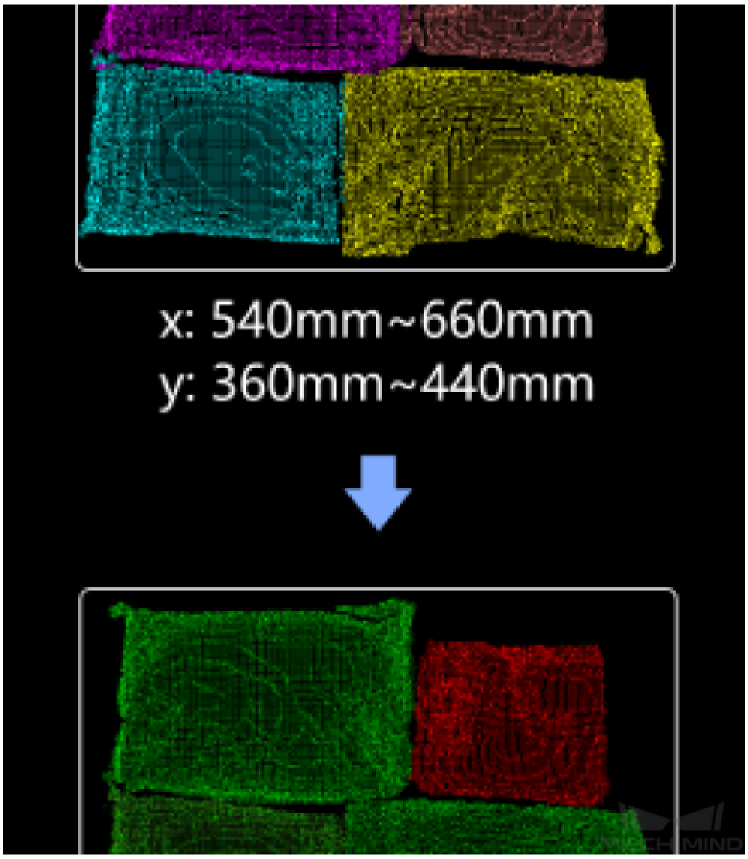
Transform Plane Cloud to Align Direction

4.11.8 Validate and Get Clouds

Classification by Point Clouds' Sizes

Function

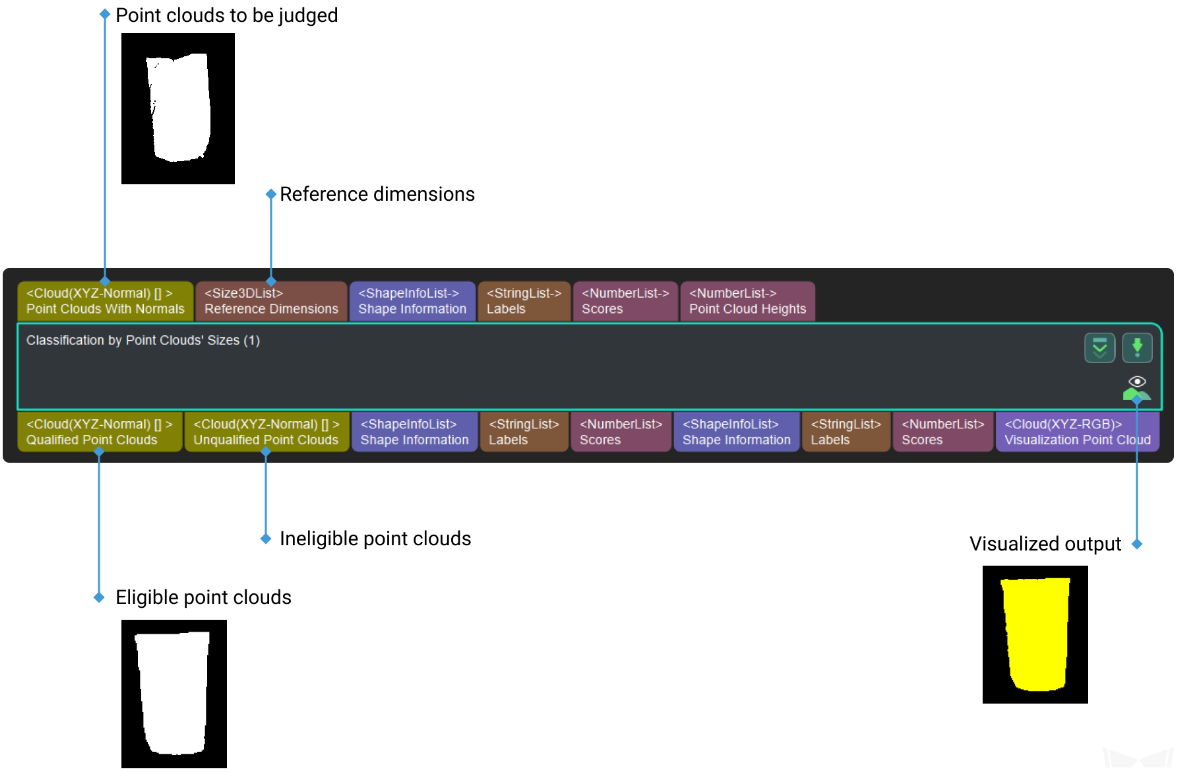
Based on the reference dimensions, judge whether each of the input point clouds is qualified. If the deviation between the dimensions of an input point cloud and the reference dimensions is within the threshold range, the point cloud is eligible.



Sample 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 points from previous Steps are valid or not. This Step is usually used with Steps as *Apply Masks to Point Cloud*, *Get Highest Layer Clouds*, etc.

Input and Output



Parameters of Input and Output

Could(Normal)[] [Input]the list of the inputted point clouds

Size3dList [Input]the sizes used to classify the point clouds

ShapeInfoList- [Input]the shape information of detected point clouds

StringList- [Input]the labels of inputted point clouds

DoubleList- [Input]the confidence of point clouds

DoubleList- [Input]the height of point clouds

Could(Normal)[] [Output]the point clouds with correct sizes

Cloud(Normal)[] [Output]the point clouds with incorrect sizes

ShapeInfoList [Output]the shape information of detected point clouds

StringList [Output]the labels of the point clouds with correct sizes

DoubleList [Output]the confidence of the point clouds with correct sizes

ShapeInfoList [Output]the shape information of detected point clouds

StringList [Output]the labels of the point clouds with incorrect sizes

DoubleList [Output]the confidence of the point clouds with incorrect sizes

Parameters

length diff ratio thre

lengthDiffLowerRatioThre

Instruction：When the longer side of the real point cloud size is smaller than the longer side of the input specification size, this parameter will be used for the following calculation to determine whether the longer side meets the specifications: real size < (1 + lengthDiffLowerRatioThre)* inputted length.

Default Value：0.2500

Suggested Value：To set according to the actual scenarios.

lengthDiffUpperRatioThre

Instruction：When the longer side of the real point cloud size is greater than the longer side of the input specification size, this parameter will be used for the following calculation to determine whether the longer side meets the specifications: real size < (1 + lengthDiffLowerRatioThre)* inputted length.

Default Value：0.2500

Suggested Value：To set according to the actual scenarios.

width diff ratio thre

widthDiffLowerRatioThre

Instruction: When the shorter side of the real point cloud size is smaller than the shorter side of the input specification size, this parameter will be used for the following calculation to determine whether the shorter side meets the specifications: real size < (1 + widthDiffLowerRatioThre)* inputted width.

Default Value：0.2500

Suggested Value：To set according to the actual scenarios.

widthDiffUpperRatioThre

Instruction: When the shorter side of the real point cloud size is greater than the shorter side of the input specification size, this parameter will be used for the following calculation to determine whether the shorter side meets the specifications: real size < (1 + widthDiffUpperRatioThre)* inputted width.

Default Value：0.2500

Suggested Value：To set according to the actual scenarios.

height diff ratio thre

heightDiffLowerRatioThre

Instruction: When the height of the real point cloud size is smaller than the height of the input specification size, this parameter will be used for the following calculation to determine whether the height meets the specifications: real size < (1 + heightDiffLowerRatioThre)* inputted height.

Default Value：0.2500

Suggested Value：To set according to the actual scenarios.

heightDiffUpperRatioThre

Instruction: When the height of the real point cloud size is greater than the height of the input specification size, this parameter will be used for the following calculation to determine whether the height meets the specifications: real size < (1 + heightDiffLowerRatioThre)* inputted height.

Default Value：0.2500

Suggested Value：To set according to the actual scenarios.

baseHeight

Instruction: This parameter is used to set the base height(in m).

Default Value：0.0000

Suggested Value：To set according to the actual scenarios.

aspect ratio thre

aspectRatioThre

Instruction: This parameter is used to set the difference between the aspect ratio of the point cloud size and the aspect ratio of the input specifications.

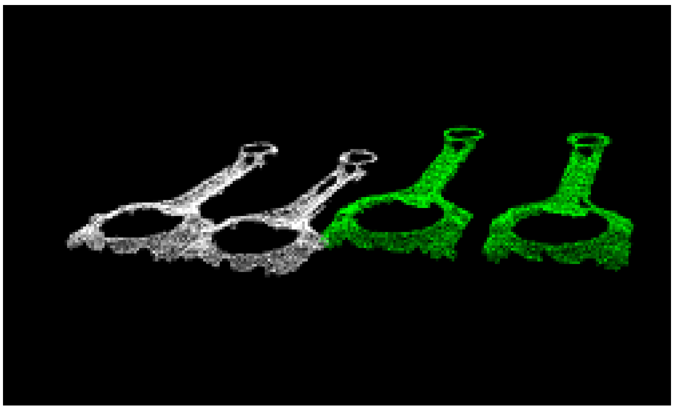
Default Value：1.0000

Suggested Value：To set according to the actual scenarios.

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.



Sample Scenario

Keep point clouds that meet the requirements along the user-specified reference direction. Unlike *Get Highest Layer Points*, this Step process the point cloud directly and usually follows Steps including *Point Cloud Clustering* 、*Apply Masks to Point Cloud*, etc.

Input and Output

Parameters

Reference Direction

Default Value: x=0; y=0; z=1.0

Instruction: This parameter is used to select the reference direction, including X, Y and Z.

Layer Settings

Only Get Highest One

Default Value: False

List of Values: True, False

Instruction: This parameter decides whether to only get the highest result along the specified direction or not.

Layer Height

Default Value: 0.1 m

Instruction: This parameter is used to set the height of every layer (in m).

Layer Tolerance

Default Value: 1.0

Instruction: This parameter is used to set the upper limit of the height tolerance (in m) between the highest layer and other layers. If the distance from one item to the highest item is smaller than the multiplication of **Layer Height** and **Layer Tolerance** , this item belongs to the highest layer.

Cloud Settings

Use Max Size Cloud Normal

Default Value: False

List of Values: True, False

Instruction: This parameter decides whether to use the normal of the point cloud with the greatest size as the reference direction or not. If checked True, **Reference Direction** will be invalid.

Max Size Cloud Normal Z Dir

Default Value: Negative

List of Values: Positive, Negative

Instruction: Specify the normal direction of the max size point cloud as positive or negative in the current coordinate. This parameter is only valid when **Use Max Size Cloud Normal** is checked as True.

Get Highest Layer Points

Description

This step is used to get the points in highest layer along the specified direction.

Parameters of Input and Output

Could(Normal)[] [Input]one inputted point cloud.

PoseList- [Input]the pose of the cloud

Cloud(Normal) [Output]the highest point along the reference direction

Parameters

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：To set according to the actual scenarios.

Layer property

onlyGetHighestOne

Instruction：This parameter decides whether to only get the highest point along the specified direction or not.

Default Value：False

Suggested Value：To set according to the actual scenarios.

List of Values: True, False

LayerHeight

Instruction：This parameter is used to set the height of every layer(in m).

Default Value：0.1

Suggested Value：To set according to the actual scenarios.

layerDiffTolerance

Instruction:This parameter is used to set the upper limit of the height tolerance(in m) between the highest layer and other layers. Tolerance equals the multiplication of ‘LayerHeight’ and ‘layerDiffTolerance’. If the height differene is less than the parameter, they belong to the same layer(the highest layer).

Default Value：1.0

Suggested Value：To set according to the actual scenarios.

useInputPoseZdir

Instruction: This parameter decides whether to use Z direction of the inputted pose as the reference direction or not. If the parameter is ‘True’, ‘reference direction’ is not valid anymore.

Default Value: False

Suggested Value: To set according to the actual scenarios.

List of Values: True, False

Get Valid Ring Clouds

Validate Point Clouds

Function

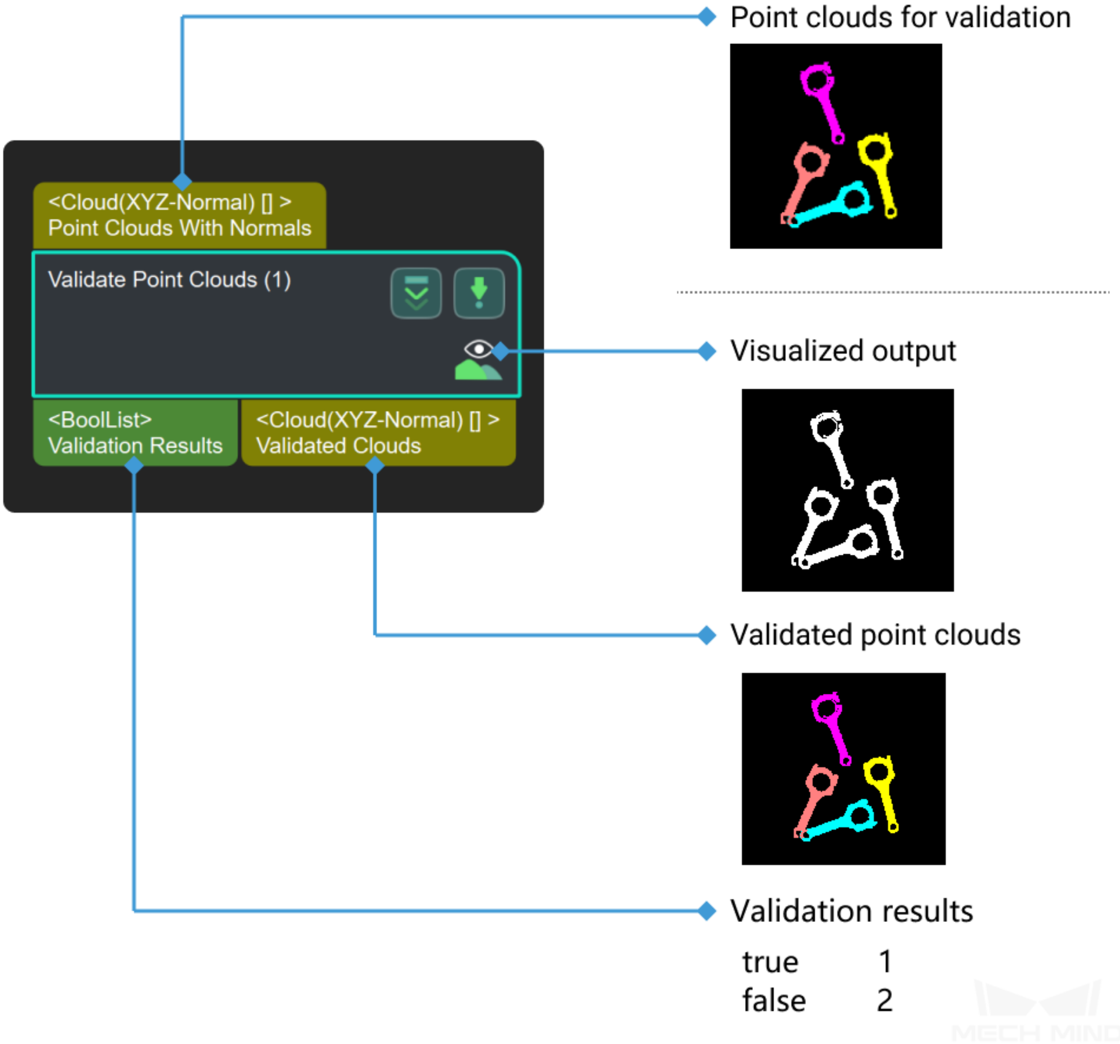
Use the set rules to judge whether the input point clouds meet the requirements.

Sample Scenario

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

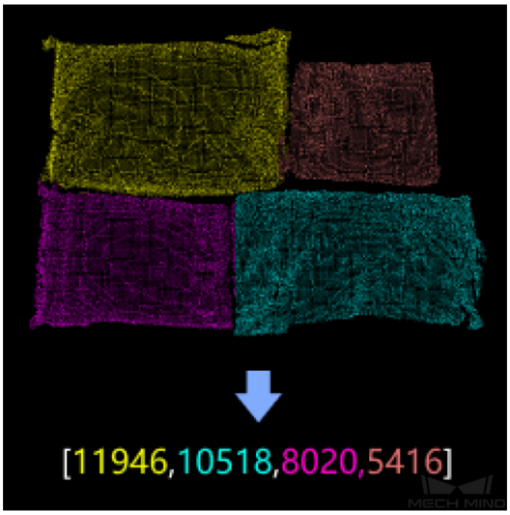


4.11.9 Others

Count 3D Points

Function

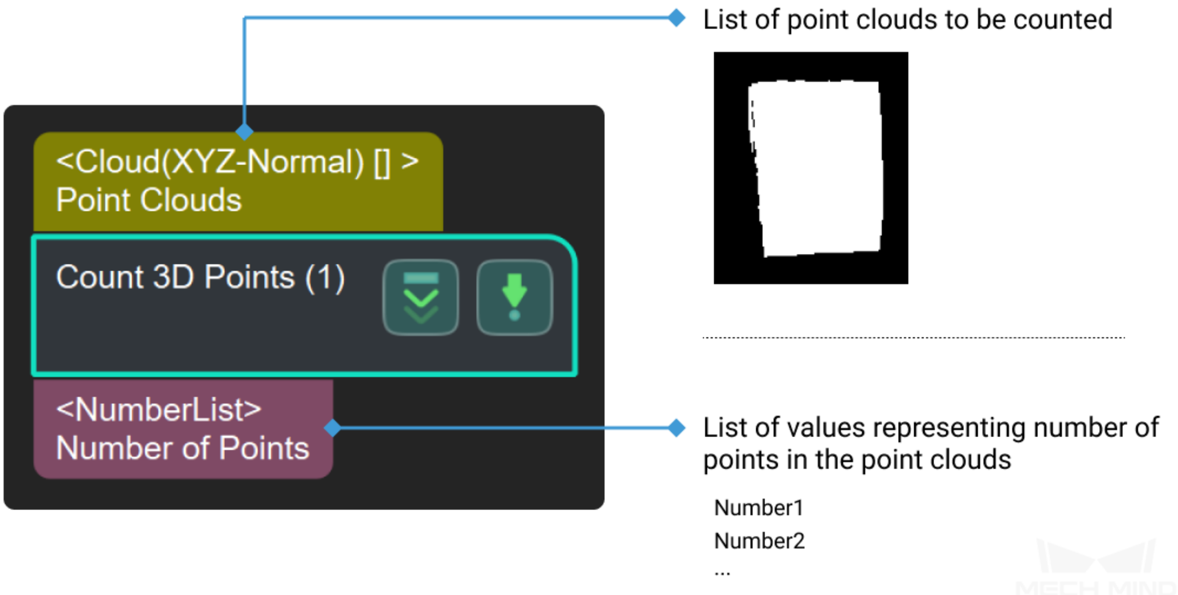
Count the points in the input point clouds.



Sample Scenario

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*, etc.

Input and Output



Detect Obscured Objects

From Cloud (XYZ-Normal) to Cloud (XYZ-RGB)

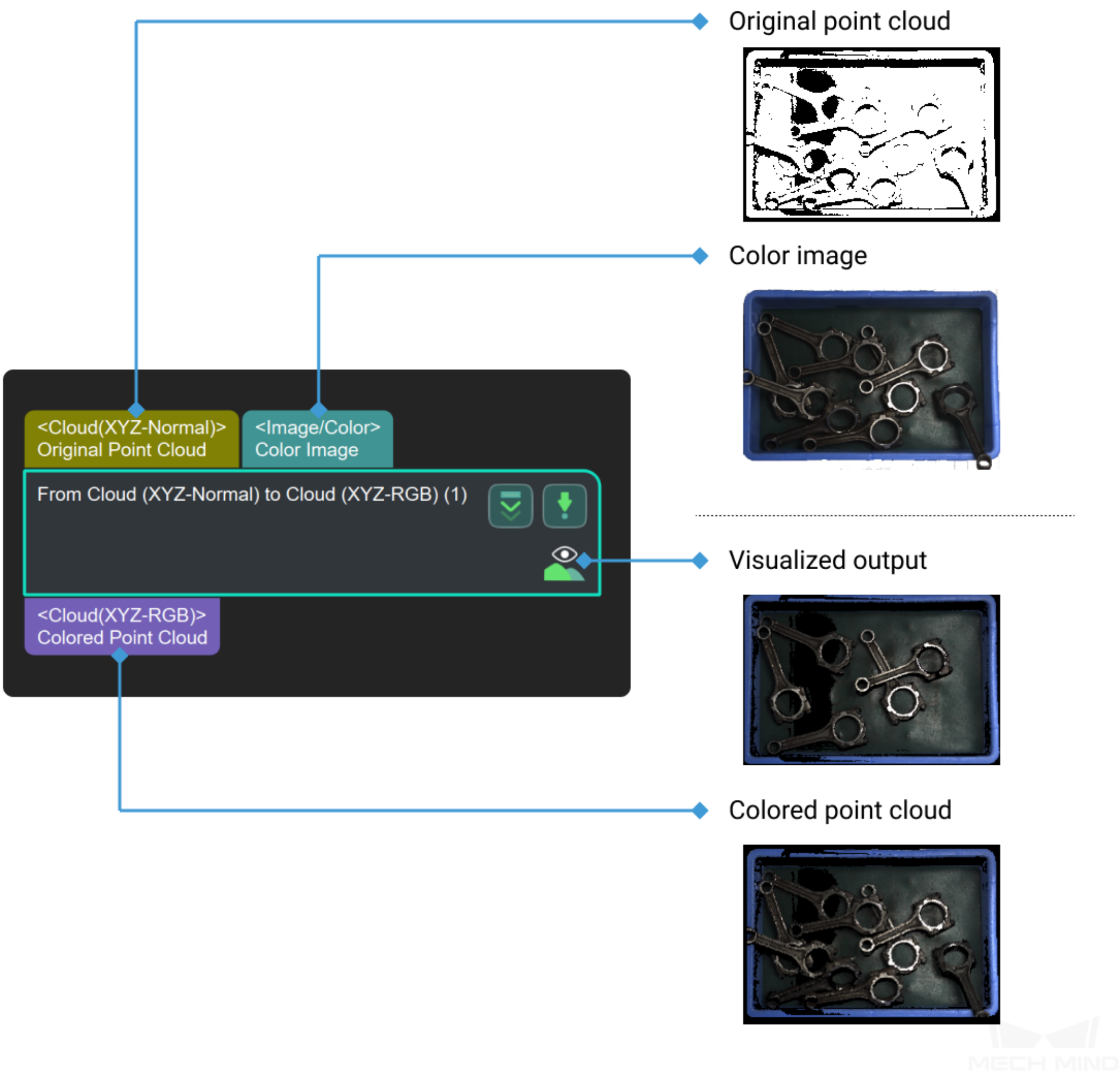
Function

Convert a colorless point cloud with normals to a colored point cloud using a 2D color image.

Sample Scenario

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



Generate Point Cloud Model

4.12 Arithmetic

In this chapter, steps that are associated with Arithmetic will be introduced.

4.12.1 Compare Values

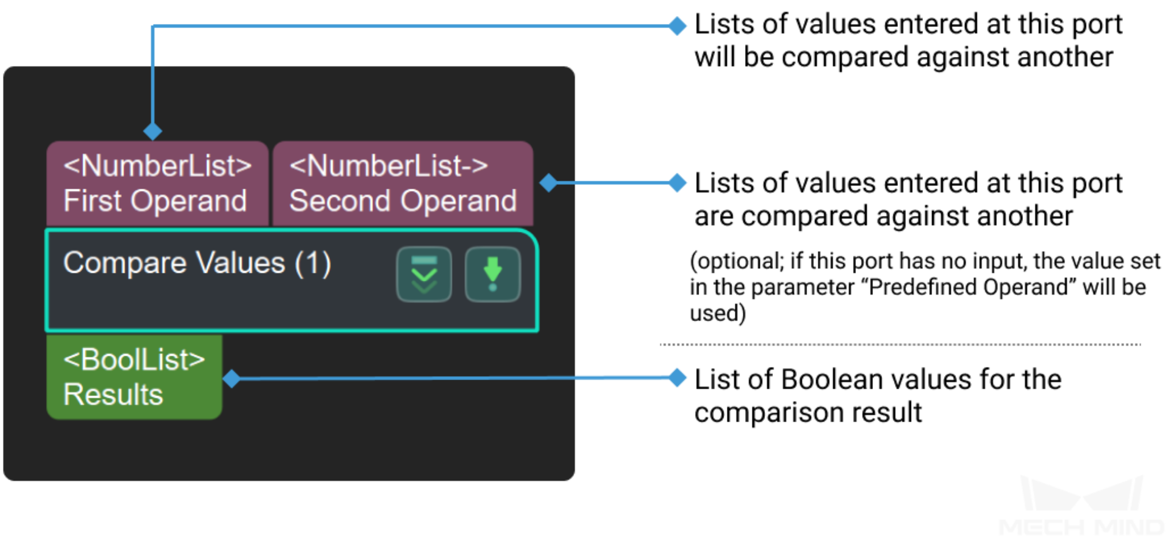
Function

According to the set operation type, compare the two input value lists, and output the Boolean value (True/False) list of the comparison result.

Sample Scenario

A general value comparison Step. No fixed usage scenarios.

Input and Output



4.12.2 Numeric Operation

Function

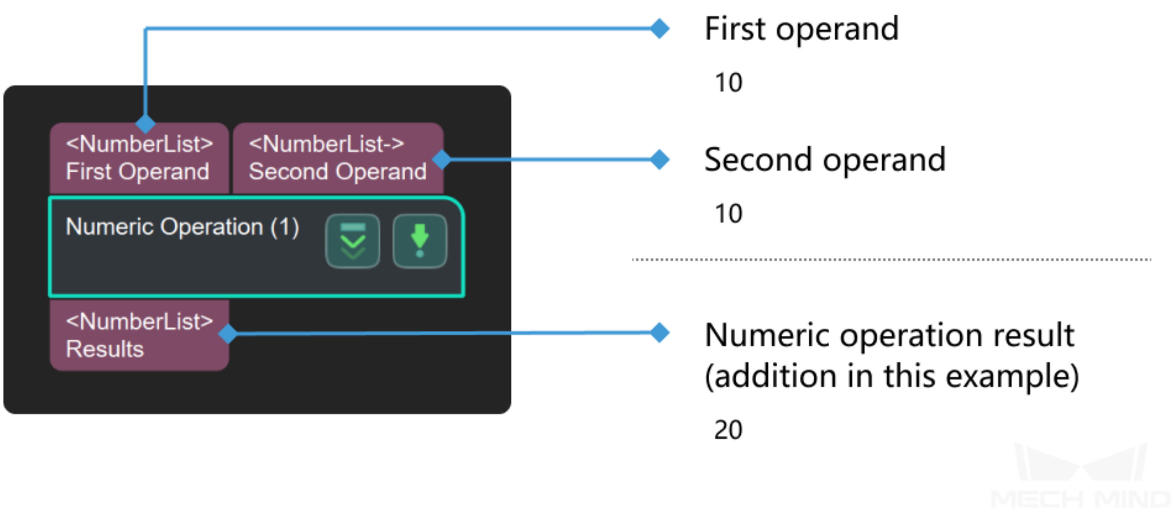
Perform numeric operations on the input values. Supported operations include addition, subtraction, multiplication, division, etc.

Sample 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



4.13 Bin Detection

This section describes the use of Steps under **Bin Detection**.

4.13.1 Detect Bin Four Sides

4.13.2 DetectBinLargestInscribedRect

4.13.3 Detect Bin Two Sides

4.13.4 Filter Poses outside Bin

4.14 Camera

In this chapter, steps that are associated with Camera will be introduced.

4.14.1 Capture from Camera

Camera 2D

Capture Images from Camera

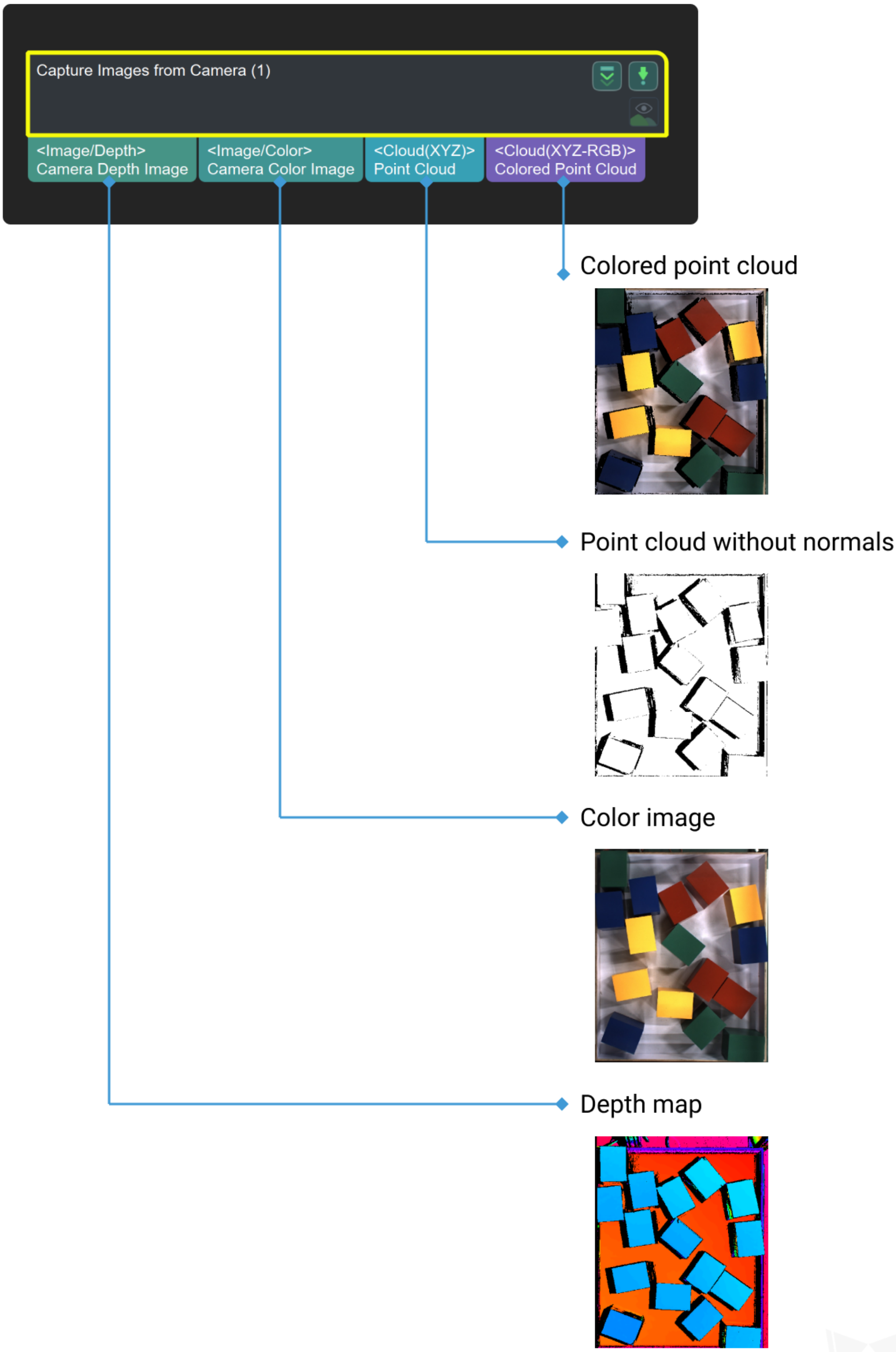
Function

Obtain color images, depth maps, and point clouds from a real or virtual camera.

Sample Scenario

Usually used as the initial input entry of a vision project. This Step collects data through Mech-Eye Industrial 3D Cameras or third-party cameras or acts as a simulated data source from a virtual camera.

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
Colored point cloud
Point cloud without normals
Color image
Depth map

Parameters

Camera Settings

Camera Type

Default Value: MechEye

List of Values: MechEye, VirtualCamera,External2D

Instruction: Select MechEye when connecting a real camera, and VirtualCamera when using camera parameter for simulation.

Camera ID

Default Value: CAM00000

Instruction: Click *open an editor* to enter the "Camera Connection Assistant" window. Select the camera that needs to be connected and establish a connection. After the connection is successful, click *Get Default Camera Parameter* to get the intrinisic and extrinsic parameters of the camera (If the camera parameters have been obtained before, there is no need to obtain them again). At the same time, the Mech-Eye interface is activated after the connection is successful, and the user can enter the Mech-Eye Viewer window. If the user need to set up in the Mech-Eye Viewer, please refer to Mech-Eye Viewer.

Camera Parameter Group Name

Automatically fill in after the camera number is set.

IP Address

Default Value: 5577

Instruction: the real IP address of the camera

Port Number

Default Value: 127.0.0.1

Instruction: Camera interface

TimeOut

Default Value: 10s

Recommended Value: 10s

Instruction: If the camera fails to capture the picture within the set time, it is a timeout. Camera will try to capture the picture again, and 3 consecutive timeouts will cause an exception report.

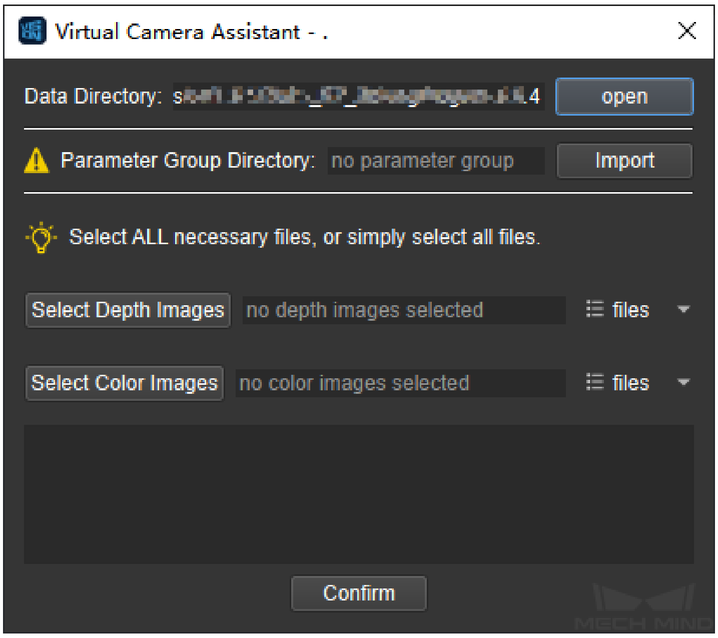
VirtualCamera

Data Path

1. Use the Eye To Hand project for simulation:

Instruction: When selecting the image data path, if the image data in the selected path is not stored in the format shown in the following figure, the “Virtual Camera Assistant” will be triggered to assist in the selection of image data:

Instructions for adjusting the virtual camera assistant:



Data Directory: Click *open* to re-select the file path here. After the correct selection, it will prompt that the setting is completed;

Camera ID/Parameter Group: Select the intrinsic and extrinsic parameter files of the camera to be loaded, and click √ to update the parameter group;

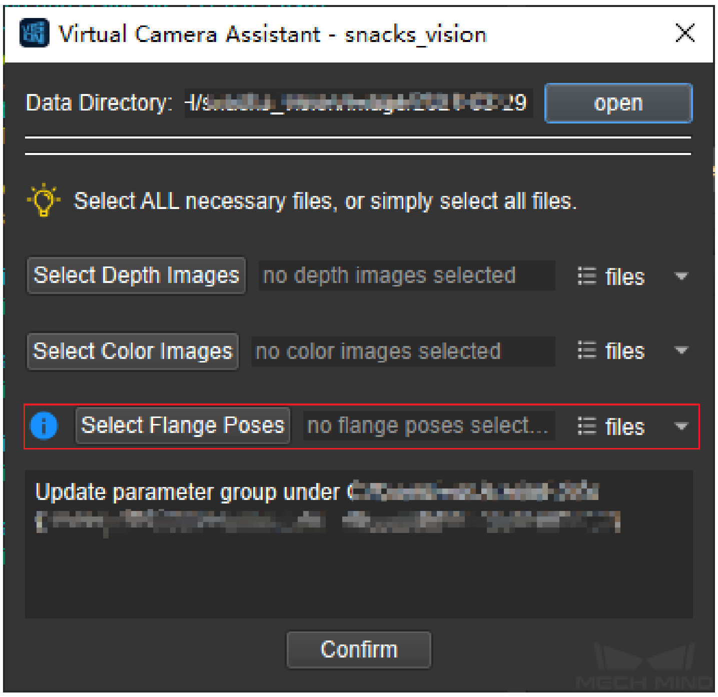
Select Depth/Select Color Image: Select several color and depth images respectively, and click *File* to view the selected image.

After all settings are completed, click "OK" and the "Virtual Camera Setting Complete" window will pop up. Then click "OK" to exit.

2.Use the Eye In Hand project for simulation

Instruction: When selecting the image data path, if the image data in the selected path is not stored in the format shown in the following figure (Compared to the ETH project, the EIH project has one more folder to store the flange poses), the "Virtual Camera Assistant" will be triggered to assist in the selection of image data:

Instructions for adjusting the virtual camera assistant:



Select Flange Poses: Select the flange poses file to be added, click *file* to view the selected file.

Play Mode

Default Value: normal

List of values: normal, repeat one, repeat all, Shuffle

Instruction: change the mode of reading the image in the file

Image Name Type

Default Value: Complete route

List of values: Complete path, File Name, Base name

Instruction: Change the picture name type of the output image

ReCapture Times

Default Value: 3

Recommended Value: 3

Instruction: The number of retakes when the photo fail

Robot Service Name in Mech-Center

Instruction: When using EyeInHand for extrinsic parameter calibration, select the robots by name if there are multiple robots.

4.14.2 Supplement

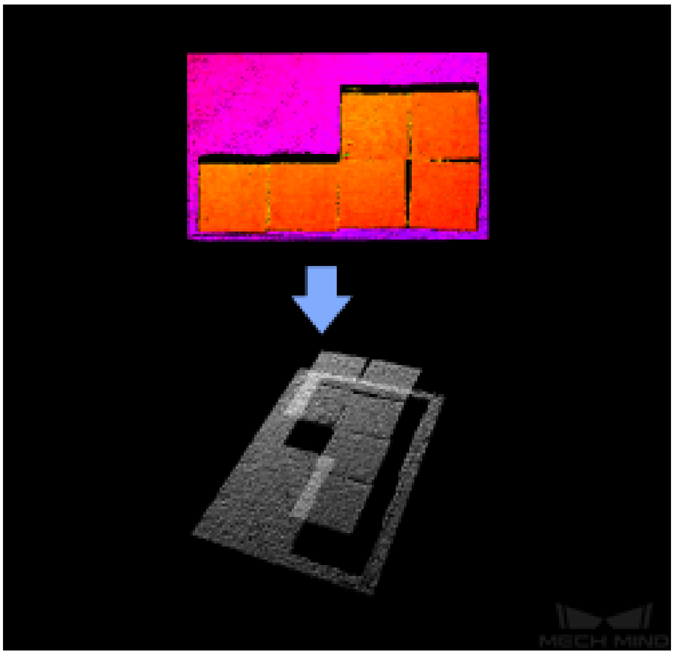
Calc Disparity Image

Compare Two Depth Image

From Depth Map to Point Cloud

Function

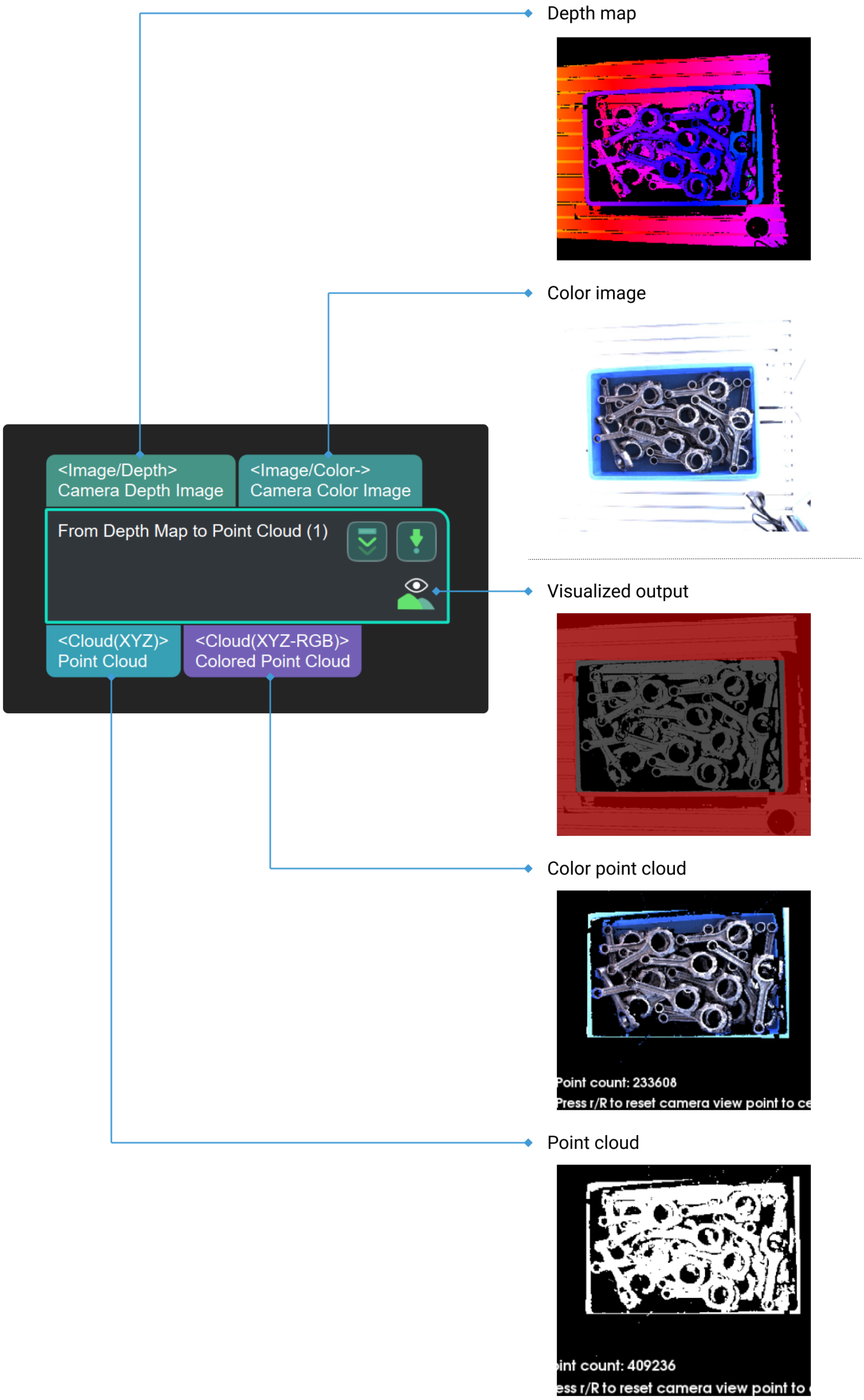
Generate a point cloud from a depth map and a corresponding color image.



Sample 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


Depth map
Color image
Visualized output
Color point cloud
Point count: 233608
Point cloud
<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

Parameter

- **Depth ROI File Name**

  Description: this parameter is the name ("depth_image_roi.json" by default) of the depth map ROI file.

  Instructions on setting the ROI:

  1. Click on the parameter value and click on the right to pop up the Set ROI window.
  2. Set the ROI in the window.
  3. Click on *OK* to save the settings and a file named "depth_image_roi.json" will be generated.

  Example: comparison of different ROI settings.

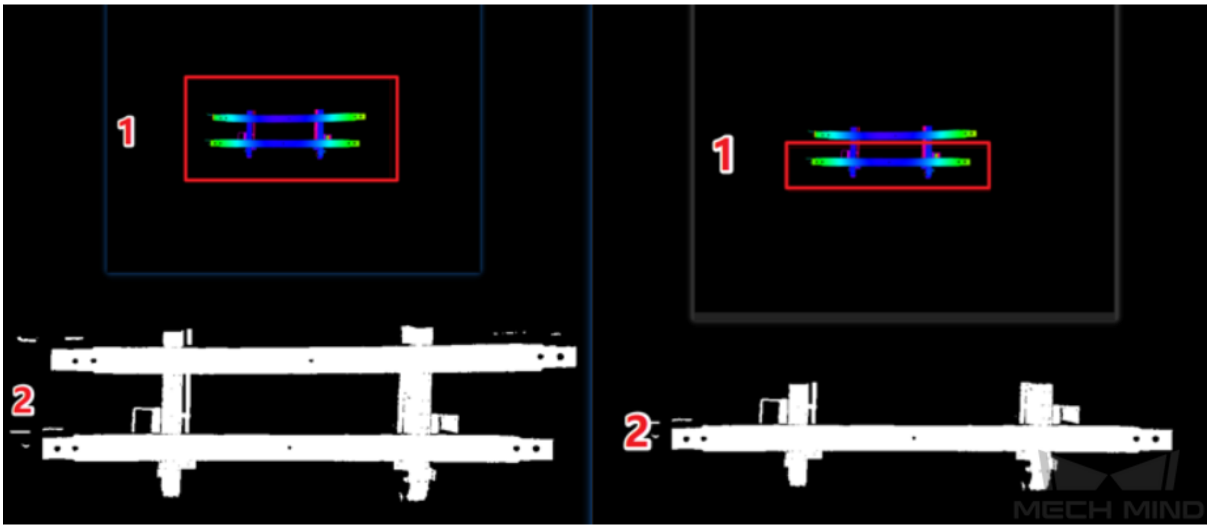


Figure 2. Comparison of different ROI settings

  1. ROIs in the depth map.
  2. Output point clouds.

- **Remove Background by Depth Image**

  When set to `True`, the background will be removed based on the settings of the two parameters **Depth Background Filename** and **Variation of Background Depth**. The two parameters are displayed only when **Remove Background by Depth Image** is set to `True`.

- **Depth Background Filename**

  Instructions: Click on the parameter value and click on the right to pop up the Set Background window.

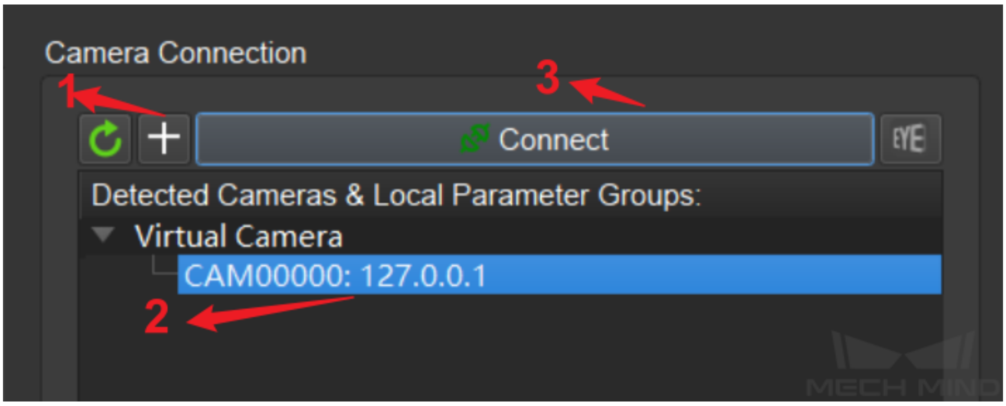


Figure 3. Connect to a virtual camera

1. Add a virtual camera.
2. Select a virtual camera.
3. Connect to the selected camera.

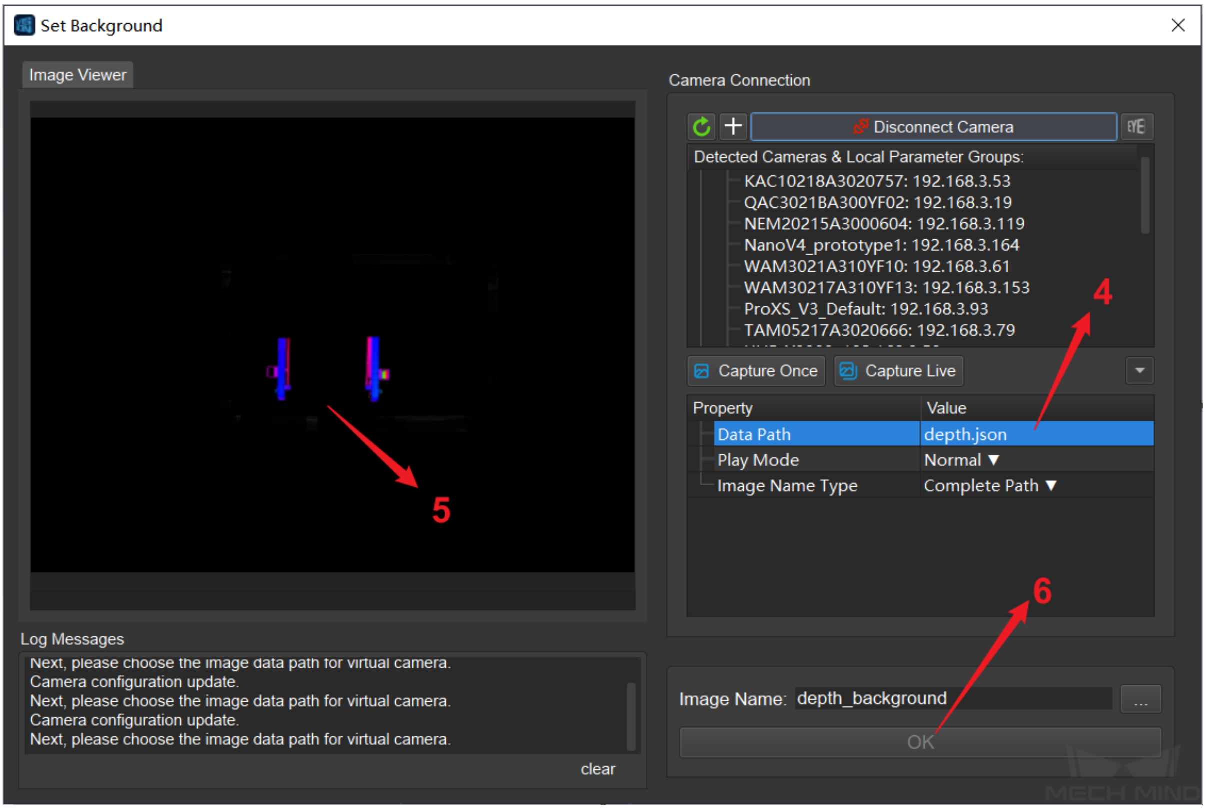


Figure 4. Capture the background depth map

4. Set the file path.
5. Display the background depth map.
6. Save the background depth map to the path.

- **Variation of Background Depth**

  Description: In input depth maps, points within the variation of the depth map in the file specified by **Depth Background Filename** will be regarded as background points and will be deleted.

  Example: *Figure 5* compares background removal effects under different values of **Variation of Background Depth**.

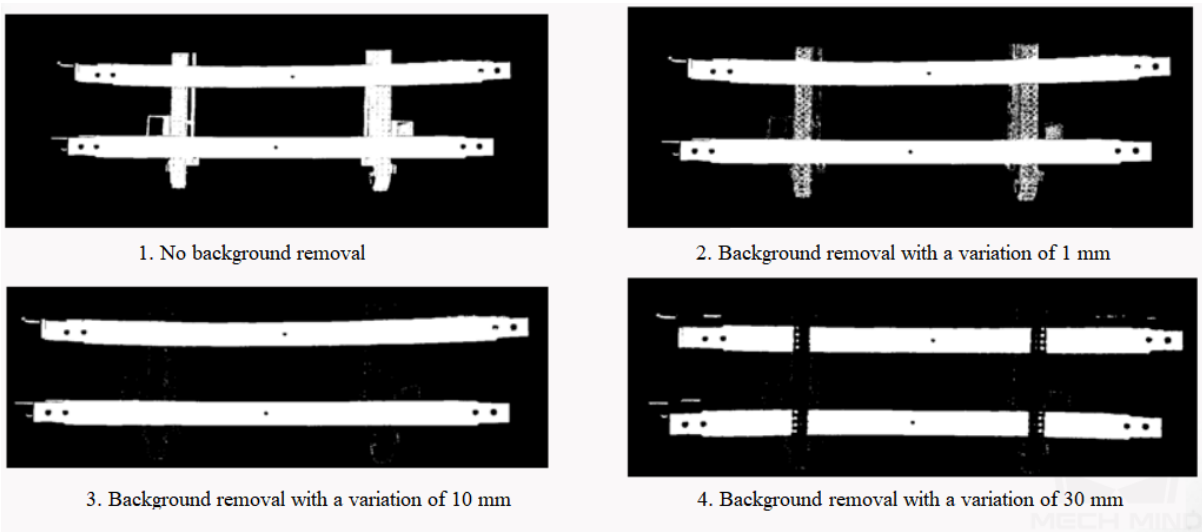


Figure 5. Background removal effects under different values of **Variation of Background Depth**

If the variation is too small, the background cannot be completely removed. If the variation is too large, the object part will also be removed.

Please set the variation to an appropriate value. Usually, the recommended value is 10 mm.

Parameter Tuning Example

An example of parameter tuning for removing the background part based on the background depth file and only keeping the object(s) is shown below.

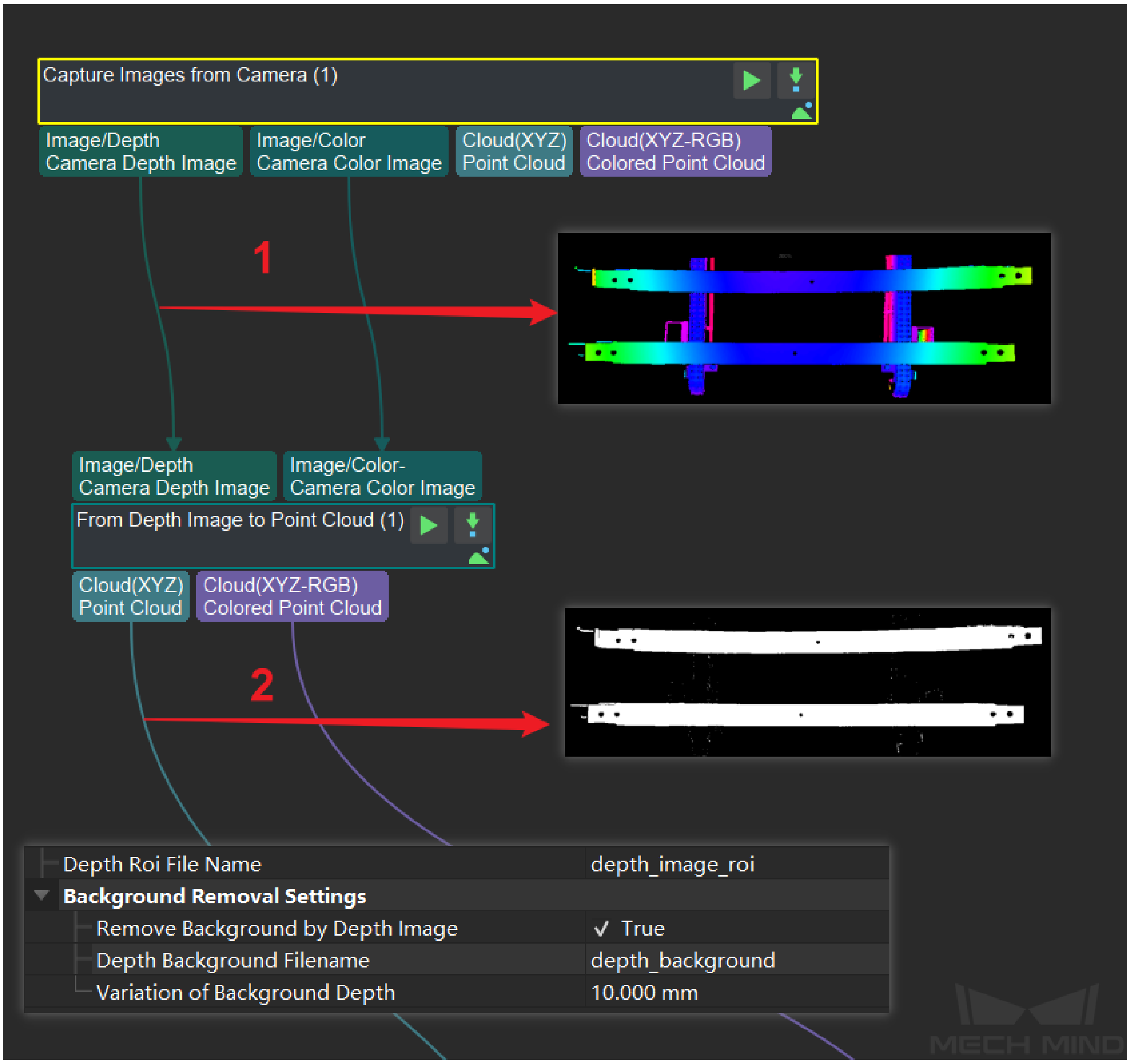


Figure 6. Parameter tuning and comparison of input and output

1. Input depth map.
2. Input point cloud.
3. Parameter tuning (please see the links above for instructions).

Comparison of Similar Steps

Capture Images from Camera can also output point cloud and color point cloud.

However, to accelerate project running, please use **From Depth Map to Point Cloud** to obtain point cloud and color point cloud.

Tip: To accelerate project running, please set an ROI in **From Depth Map to Point Cloud** to avoid generating unnecessary point cloud parts.

From Disparity Image to Depth Image

Merge Depth Images

Validate and Calc Parameter Compensation

4.14.3 Laser Profiler

4.15 Communication

In this chapter, steps that are associated with Communication will be introduced.

4.15.1 Notify

4.15.2 Send Point Cloud to External Service

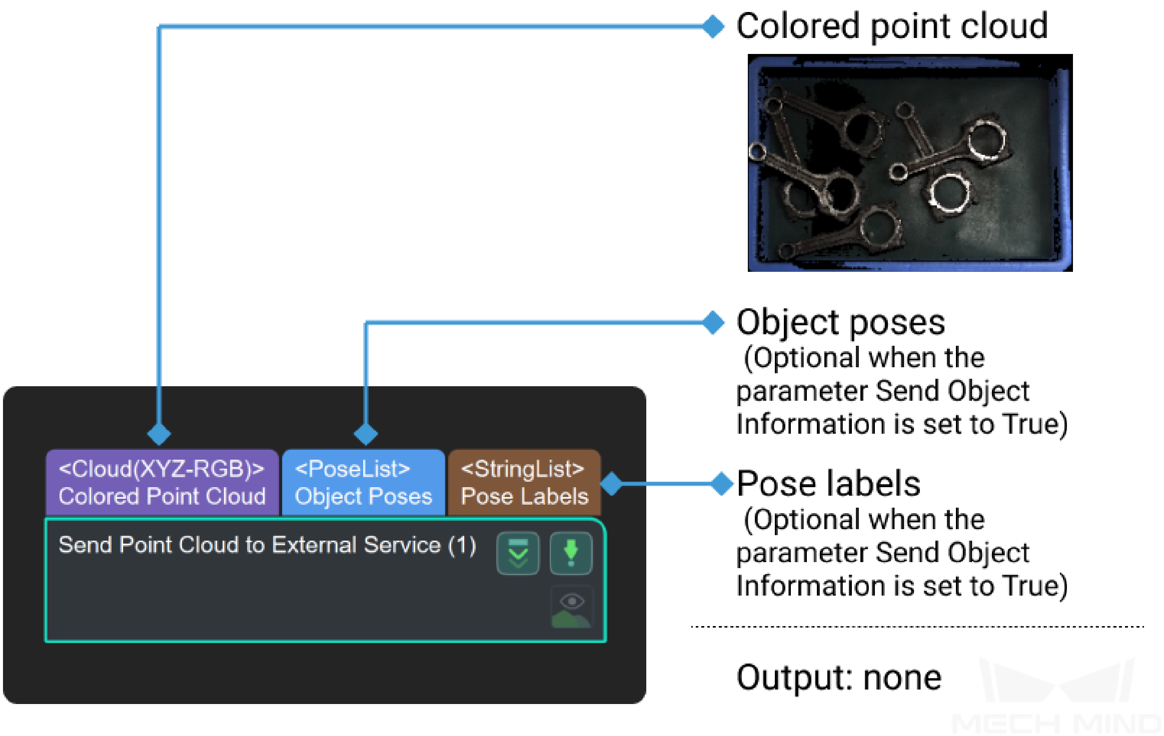
Function

Send point cloud information to Mech-Viz.

Sample Scenario

Send point cloud to Mech-Viz for simulation and debugging or for checking the actual point cloud processing results.

Input and Output



Parameters

Input point cloud type

Default Value: Color Point Cloud (CloudXYZRGB)

List of Values: color point cloud (CloudXYZRGB), point cloud (CloudXYZ), normal point cloud (CloudNORMAL)

Instruction: The type of point cloud to be sent.

Send Object Information

Default Value: False

List of Values: True, False

Instruction: Choose whether to send pose and label information according to actual needs.

Whether to convert to robot coordinate system

Default Value: True

List of Values: True, False

Instructions: If selecting True, the input point cloud information will be converted to the robot coordinate system and then sent to Mech-Viz.

4.16 Deep Learning

In this chapter, steps that are associated with Deep Learning will be introduced.

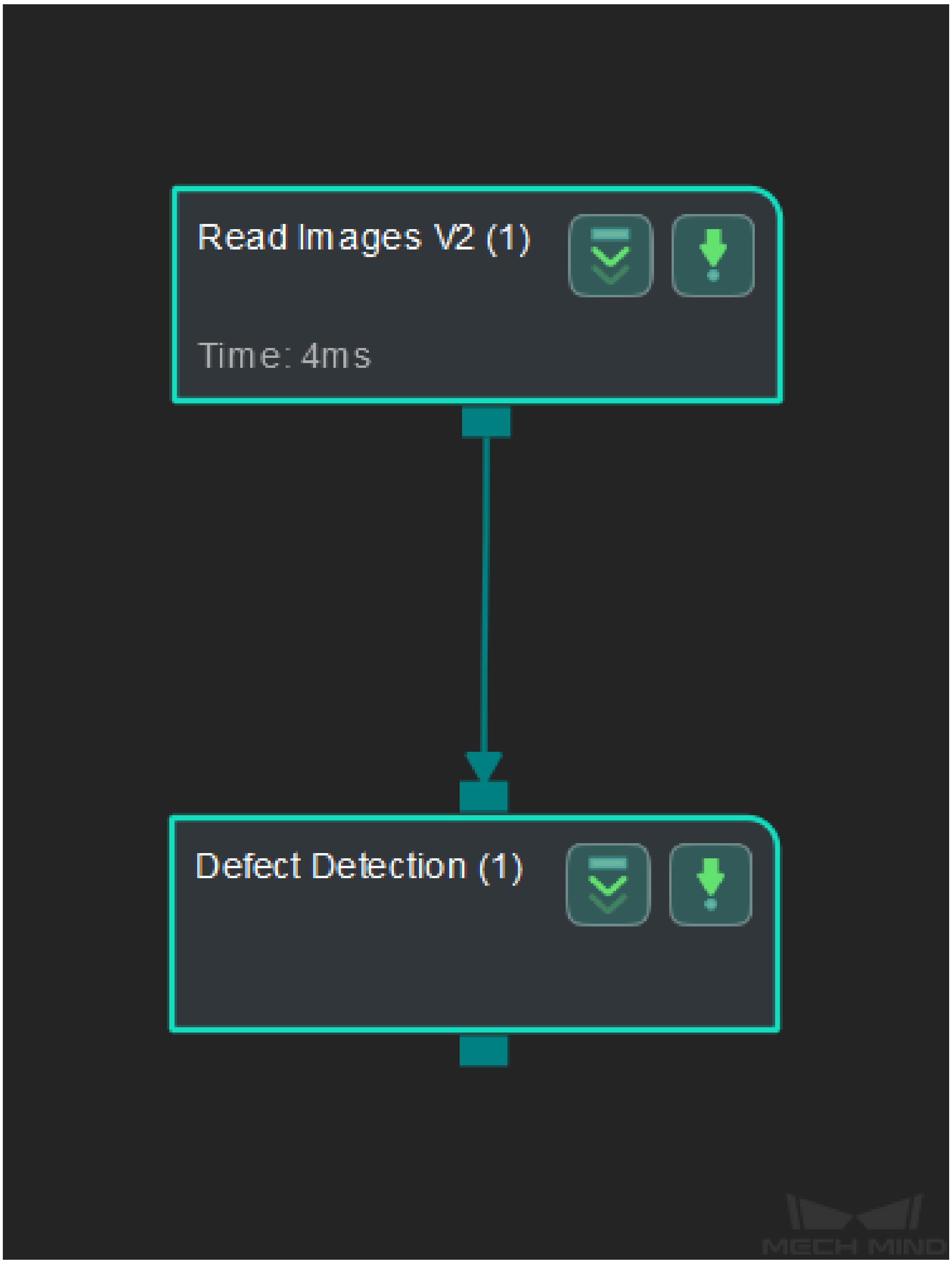
4.16.1 Defect Detection

Defect Detection

Function

This Step utilizes deep learning algorithms to detect whether there are defects in the input image and further classify the types of the defects.

Sample Application (in Measurement Mode)



Connect *Read Point Cloud V2* with *Defect Detection* to detect whether there are defects in the input image.

Connect Steps

Attention: Please prepare a deep learning model of the object to be detected in advance.

Click on *Defect Detection*, go to the **Step Input Source Selection** panel, and then select **Read Images V2_1_Color Image** as **Input 1 (Color Image)**.

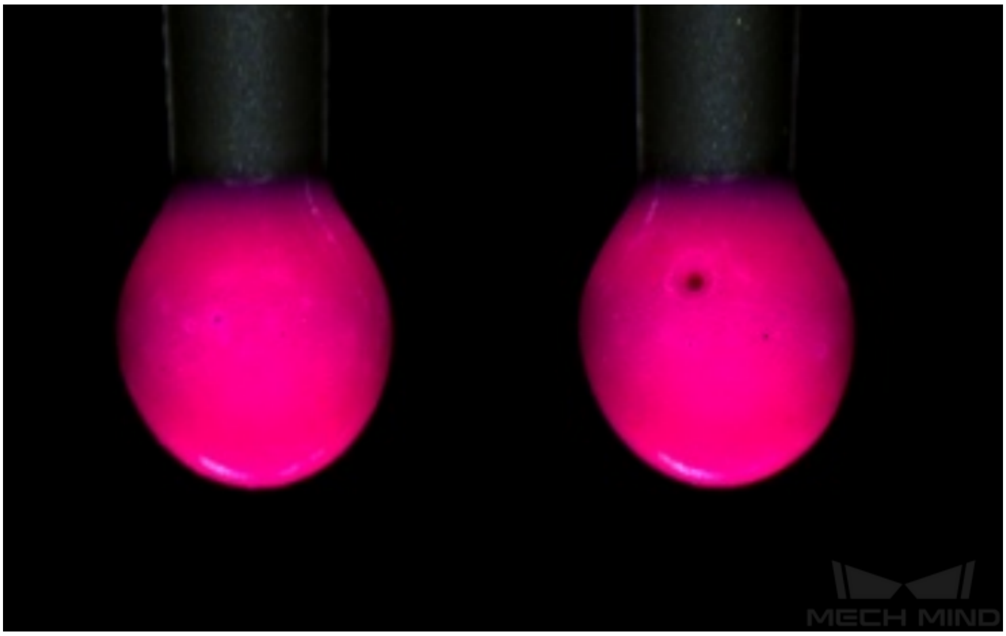
Configure parameters

Note: Please prepare the color image(s) that contain the object(s) to be measured in advance.

- Configure in *Read Images V2*

  Click on the Step, and configure the **Imgae Source**, **Image File/Folder Path**, and other parameters in the **Step Parameters** panel.

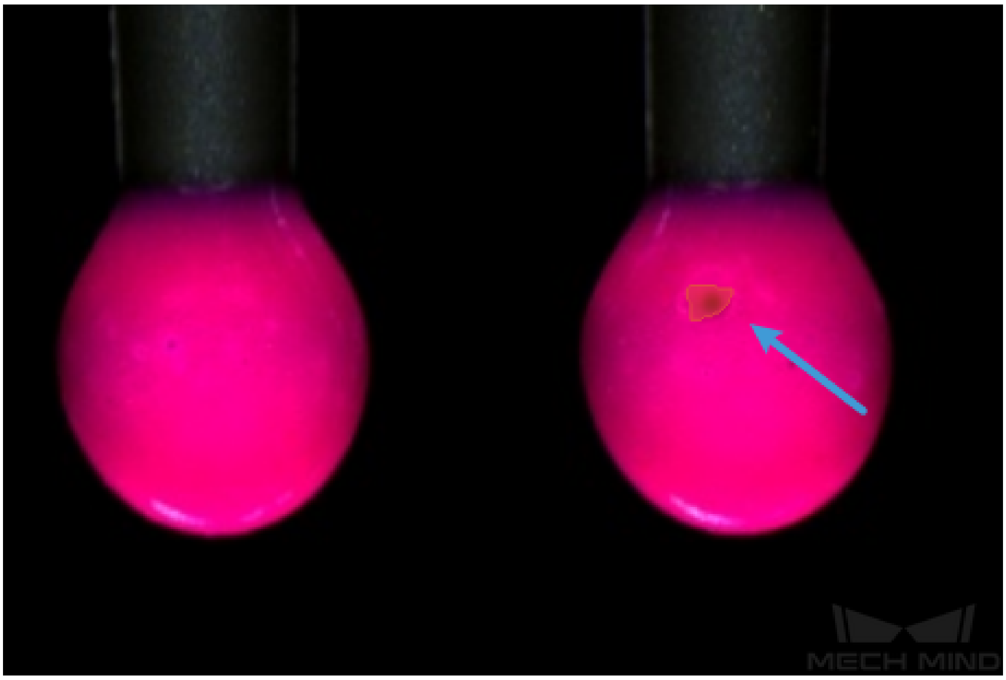
  The objects to be detected are shown as below.



- Configure in *Defect Detection*

  1. Click on the Step, go to **Step Parameters** panel and configure the **Model File** and **Configuration File**.
  2. Go to *Deep Learning→ Deep Learning Server* in the Menu Bar to check whether the server has been started.
  3. After configuration, click on *Run* to run the project.

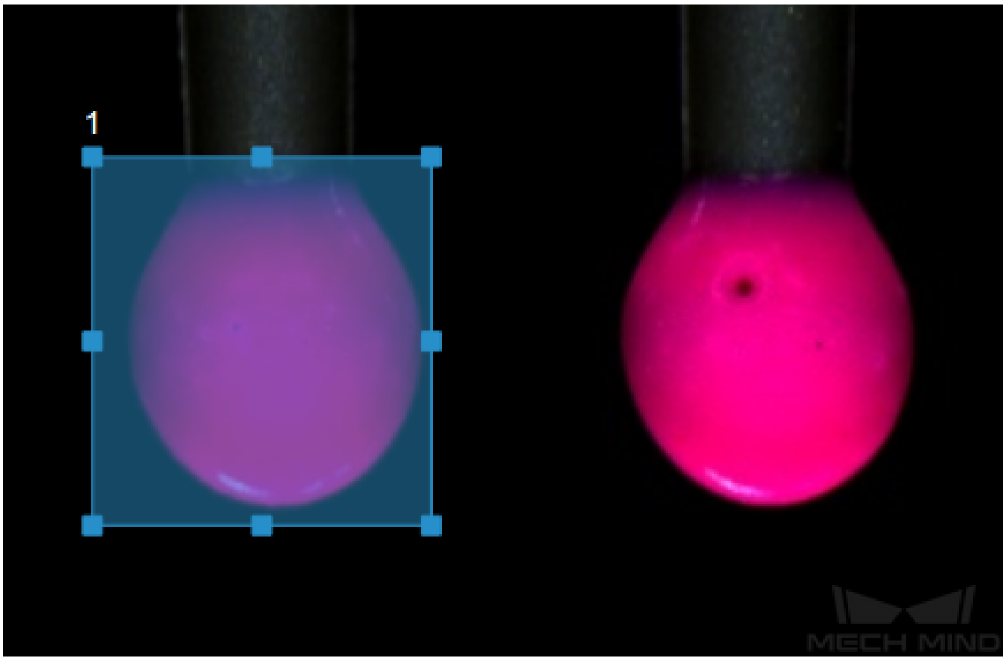
  If the project runs successfully, the defect will be annotated.



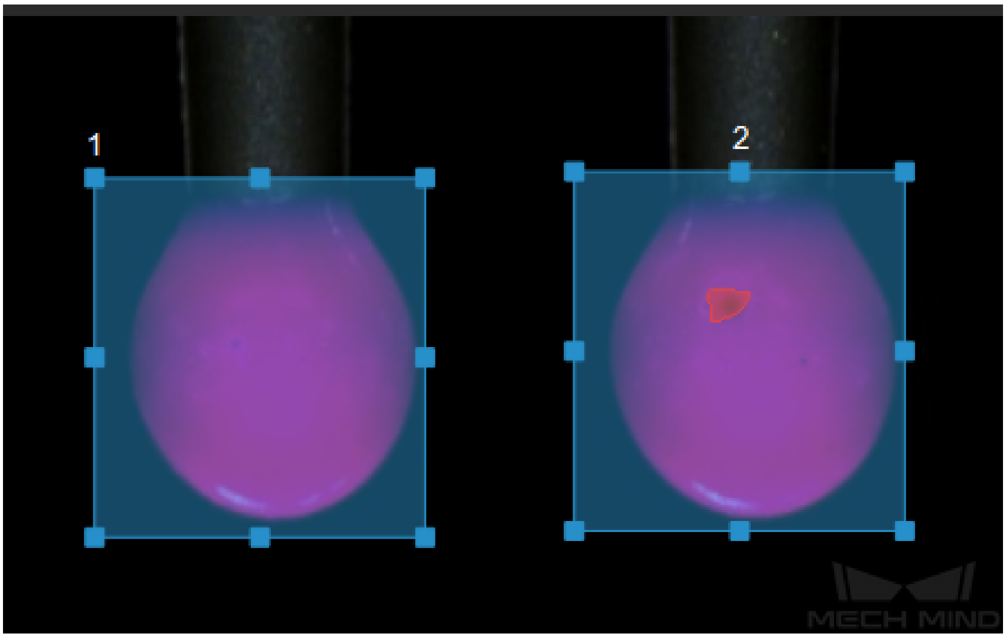
- In addition, you can select an ROI for defect detection.

  Click on *Defect Detection*, go to *Step Parameters* → *Filter Settings*. Click on *Empty* and then + to add one or more ROIs.

  Add one ROI:

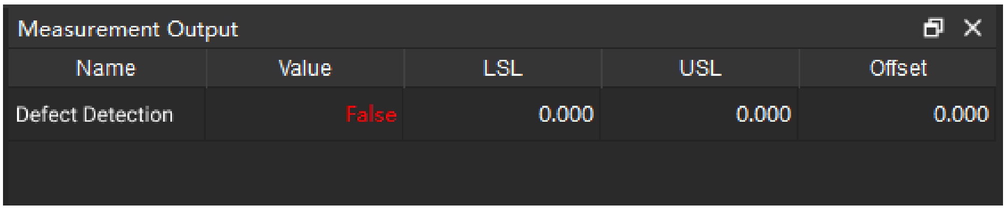


  Add two ROIs:



Read the result

- The measurement result will be displayed in **Measurement Output** and **Result View** panels. In this sample application, a defect was detected, and the measurement value is **False**. If there is not any defect, the value will be **True**.



Measurement Output

Name	Value	LSL	USL	Offset
Defect Detection	False	0.000	0.000	0.000

4.16.2 Detect Edges

Detect Edges

4.16.3 Image Classification

Image Classification

4.16.4 Instance Segmentation

Instance Segmentation

Function

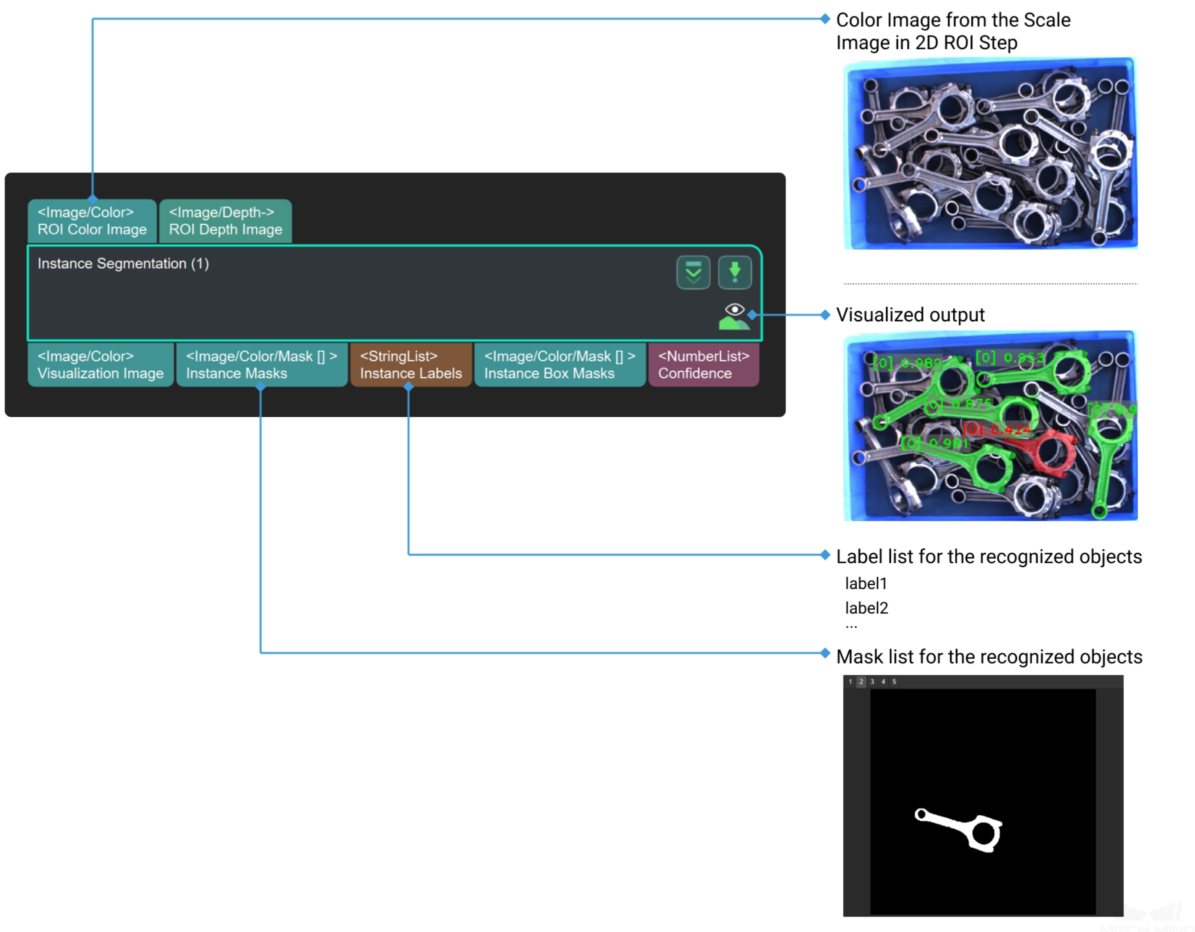
Recognize target objects in an image and locate them at the pixel scale through deep learning.

Sample Scenario

You can use this Step to infer with model files from Mech-DL Kit and Mech-DLK. The results of inference are closely related to the data on which the model was trained. If you find that the inference results are not as expected, try the following methods in sequence:

1. Please check after setting the region of interest and scaling whether the sizes of the target objects in the image are similar to those in the training data (if you are using supermodel and this method does not solve the problem, please contact the support team).
2. Please check whether the objects, ambient light, background, etc. in the input color image are similar to those of the training data.
3. Adjust the confidence threshold and use the first output port to check the visualized recognition results.

Input and Output



Parameters

Server & Model

Server Address

Default Value: 127.0.0.1:50052

Instruction: This attribute is to set the IP and port of the deep learning server, generally use the default value. If multiple instance segmentation steps are enabled, multiple corresponding deep learning servers need to be opened, and the port setting needs to correspond to the port of the deep learning server one-to-one.

Model File

Instruction: The absolute path or relative path of the deep learning model file is stored. The deep learning model file for instance segmentation is mainly a .pth format file, which can be obtained through the Mech-DLK training model.

Configuration File

Instruction: The absolute path or relative path of the deep learning configuration file is stored. The deep learning configuration file of the instance segmentation step is mainly a .py format file, which can be obtained through the Mech-DLK training model.

Preload Setting

Preload Model on Project Open

Default Value: False

List of Values: True,False

Instruction: When checked as True, the existing deep learning model will be automatically loaded after the project is started, which can greatly improve the speed of the first deep learning run after starting the project. It is recommended to enable the editing of this parameter after the project is debugged.

Max Detected Objects

Default Value: 100

Instruction: This attribute is used to set the maximum number of instances that can be displayed in the image, and the range is 0~500, which can be adjusted according to actual needs.

Acceptable Mean for Gray-scaled Input

This attribute judges whether the average gray value of the input image is in the set gray average value interval, and filters overexposed or over dark images to avoid incorrect instance segmentation, as shown in *Figure 1* .

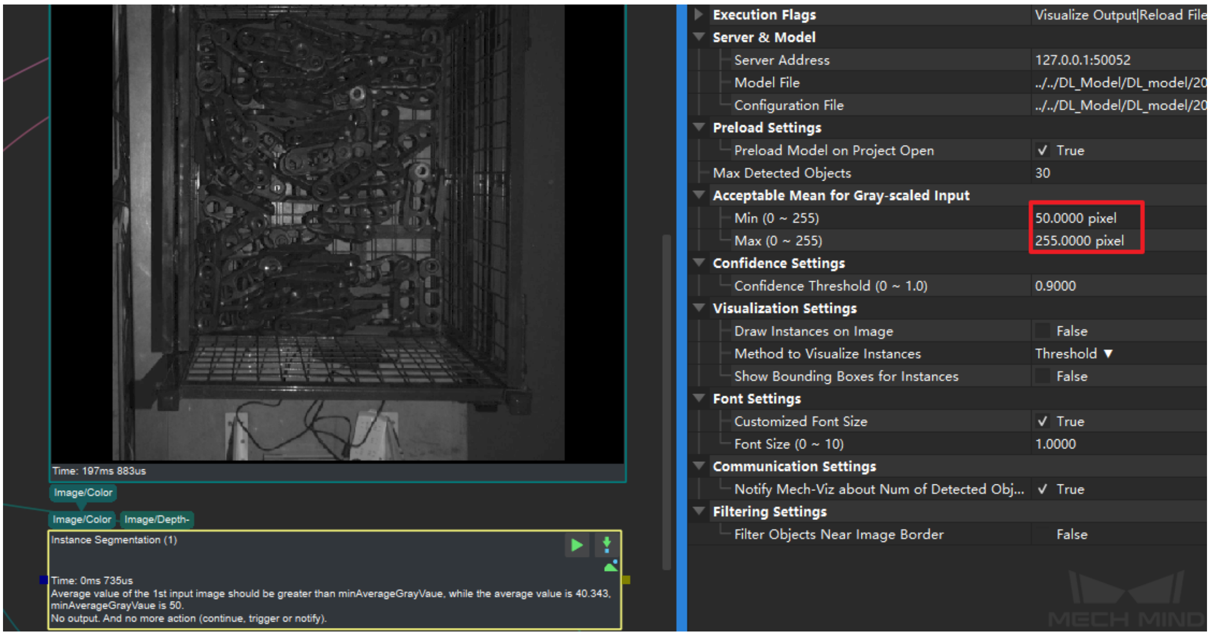


Figure1 Filter Out Dark Images

Min

Default Value: 0

Instruction: If the average gray value of the picture is less than the value of this parameter, the picture will not be processed. The parameter can set the value range: 0~255.

Max

Default Value:255

Instruction: If the average gray value of the picture is greater than the value of this parameter, the picture will not be processed. The parameter can be set in the range of 0~255.

Confidence Settings

Confidence Threshold

Default Value: 0.7

Instruction: After calculation and processing through deep learning, instances with confidence greater than the threshold will be retained as the output result.

Example: The confidence threshold can be adjusted by first enabling the attribute **Draw Instance on image** and using the visualization output to obtain the confidence of each instance. Then adjust the confidence threshold accordingly, as shown in *Figure 2* .

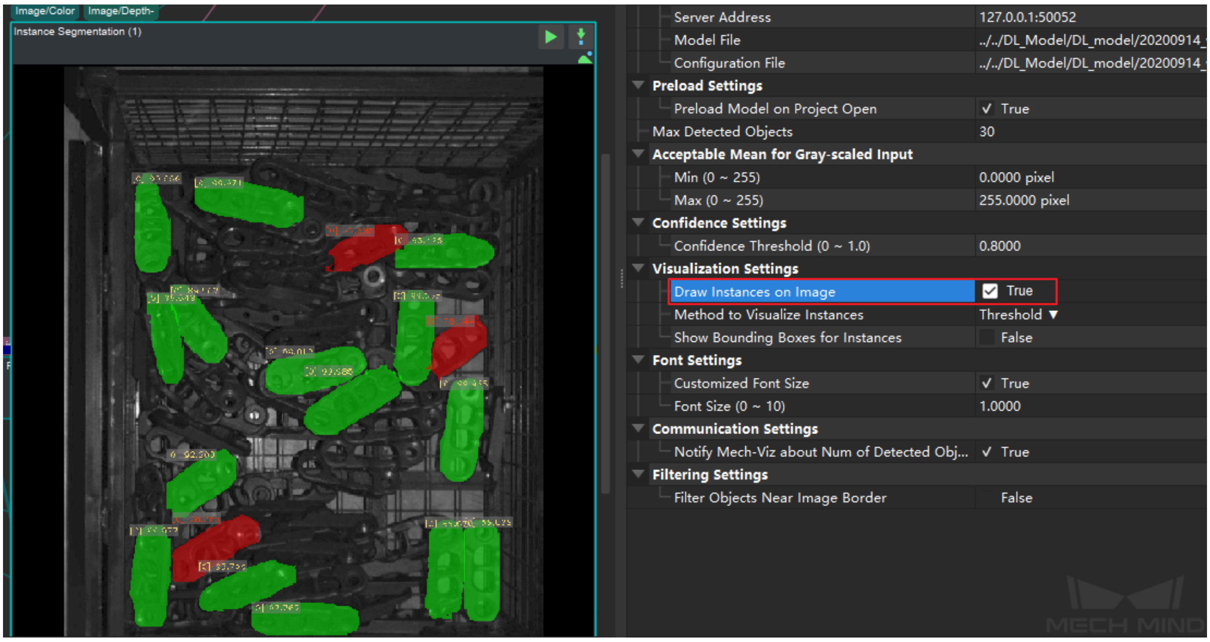


Figure2 Visualize Confidence Threshold

If the number of instances obtained by running at the current threshold is small or the result is not good enough, as shown in *Figure 3* , the red data in the figure are instances with confidence lower than the threshold , which will not be output result. At this point, you can try to lower the threshold, as shown in *Figure 4*. After lowering the threshold, the number of instances as output results has increased significantly.

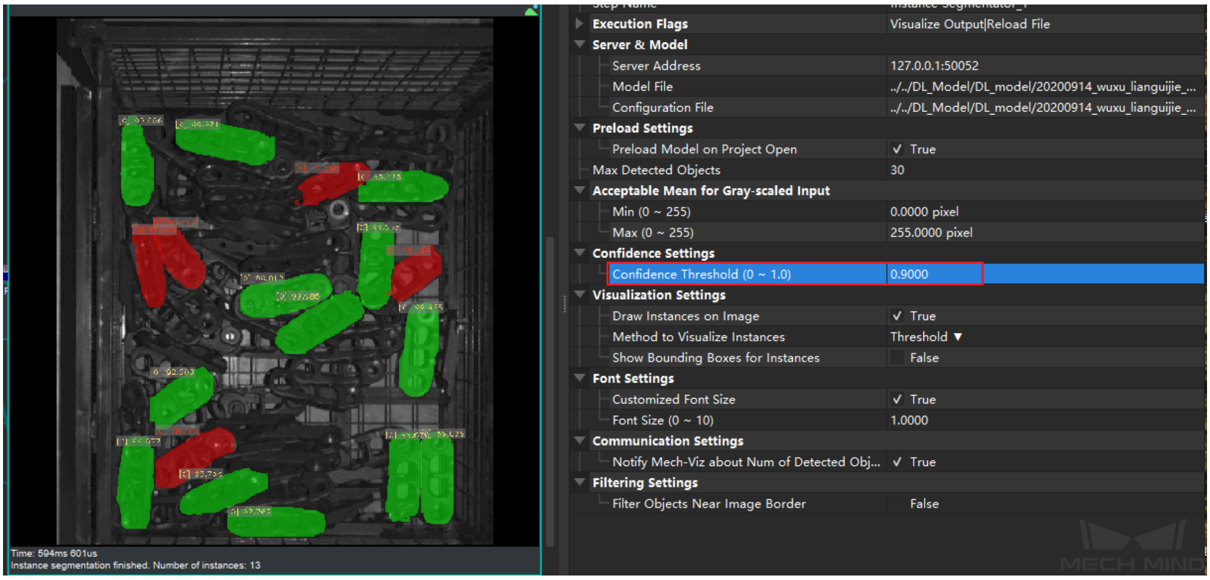


Figure3 Instance Segmentation Results under High Threshold

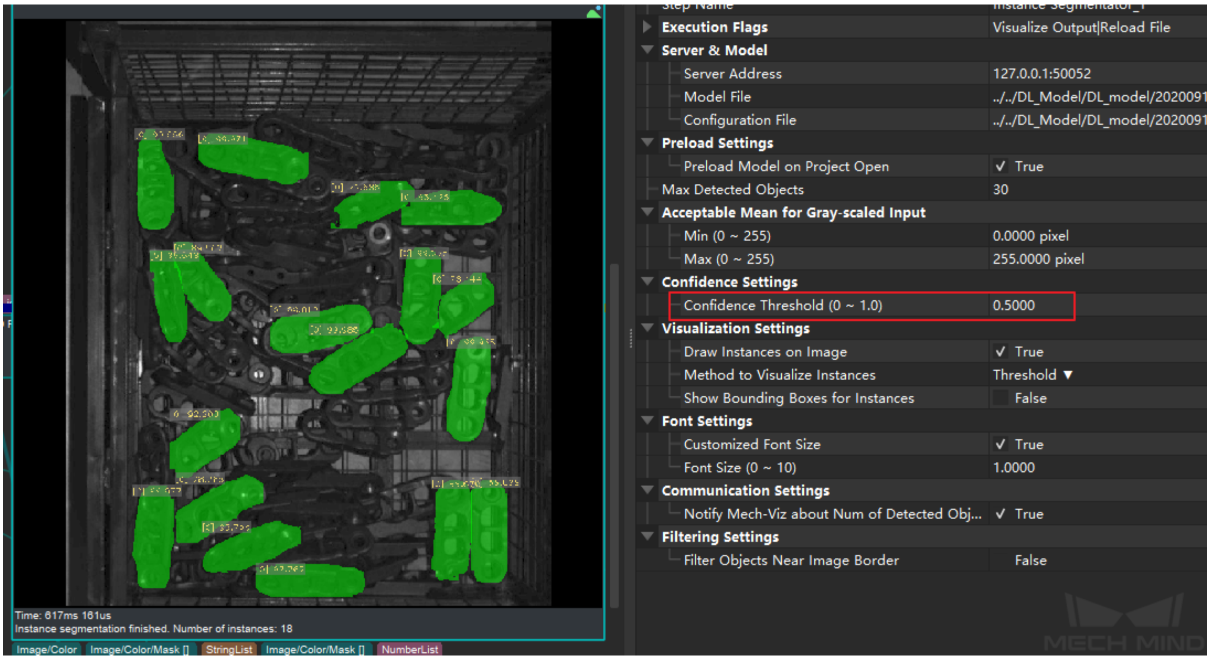


Figure4 Instance Segmentation Results under Low Threshold

If the result does not change much after adjusting the confidence threshold, the user can try to change the deep learning model, or check whether the object for instance segmentation is within the selected ROI.

Visualization Settings

Draw Instance on Image

Default Value: False

List of Values: False, True

Instruction: When set as **True** , the user can see the instance segmentation results in the visual output, but it takes longer to run. This parameter is recommended to be turned on during debugging, and turned off during actual operation.

Method to Visualize Instances

Default Value: Instance

List of Values: Threshold, Instances, Classes

Instruction: By selecting different parameters, the instance can be filled in different ways. Threshold: The filled color depends on the confidence of the instance. Instances with a confidence greater than the confidence threshold will be filled with green, otherwise they will be filled with red. Instances: Fill each instance with different colors. Classes: Fill the instances of the same label with the same color.

Show Bounding Boxes for Instances

Default Value: True

List of Values: True, False

Instruction: When set as **True** , the border of the divided instance will be displayed. This parameter needs to be enabled in the parameter **Draw Instances on Image** before it can take effect.

Font Settings

Customized Font Size

Default Value: False

List of Values: True,False

Instruction: When set as **True** , the font size displayed in the visual output can be customized to adjust. This parameter needs to be enabled in the parameter **Draw Instances on Image** before it can take effect.

Font Size

Default Value: 3.000

Instruction: This parameter needs to be set after the parameter **Customized Font Size** is enabled. Set according to actual needs. The range is 0-10. When it is set to 0, the font is not visible; when it is set to 10, the font is the largest.

Communication Settings

Notify Mech-Viz about Num of Detected Objects

Default Value: True

List of Values: True, False

Instruction: When set as **True**, the number of split instances will be sent to Mech-viz.

Filtering Settings

Filter Objects Near Image Border

Default Value: False

List of Values: False, True

Instruction: When set as **True** , the edge workpieces will be filtered out.

Width of Border

Default Value: 10 pixel

Instruction: This parameter is the allowable minimum distance between the left and right width of the workpiece from the edge, and the workpiece which is smaller than the set parameter value will be filtered out.

Height of Border

Default Value: 10 pixel

Instruction: This parameter is the minimum distance between the allowable workpiece in the upper and lower width from the edge, the workpiece which is smaller than the set parameter value will be filtered out.

Display Predefined ROI

Default Value: False

List of Values: True, False

Instruction: When set as True, the image edge size set by the attribute **Width of Border** and the attribute **Height of Border** will be displayed during the visual output.

Common Procedures

Procedure of Instance Segmentation

Instance Segmentation (CPU)

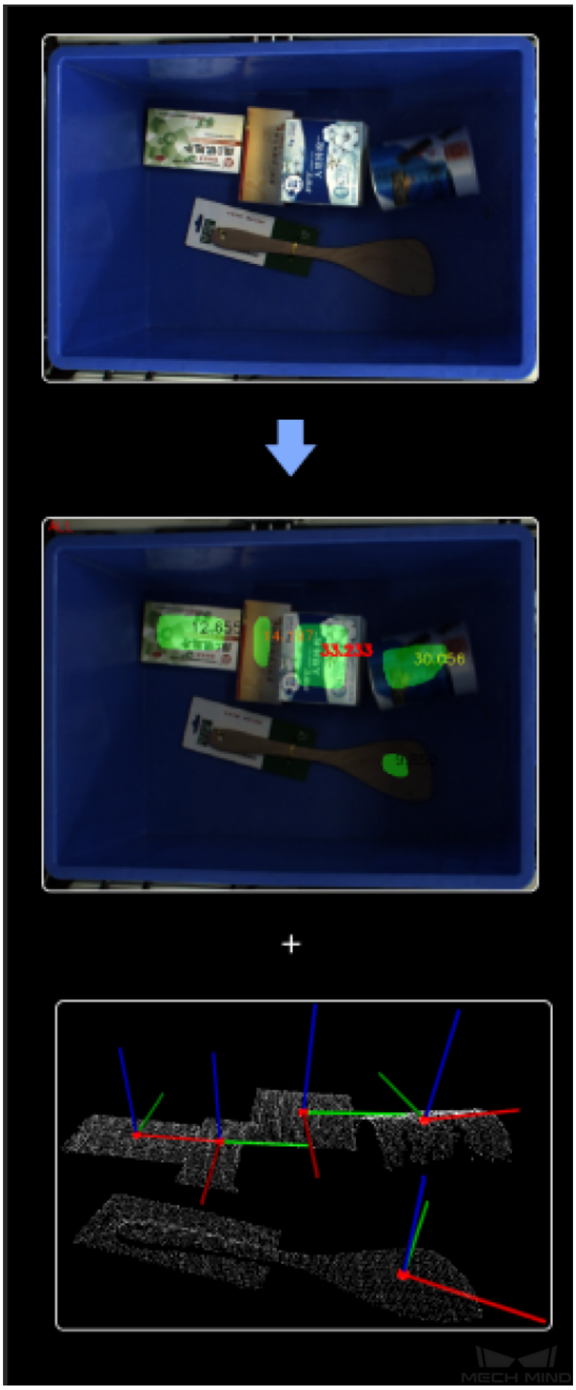
4.16.5 Novel Objects Picking

Calc Poses From Heat Map Of Graspability

Grasp Pose Estimation

Function

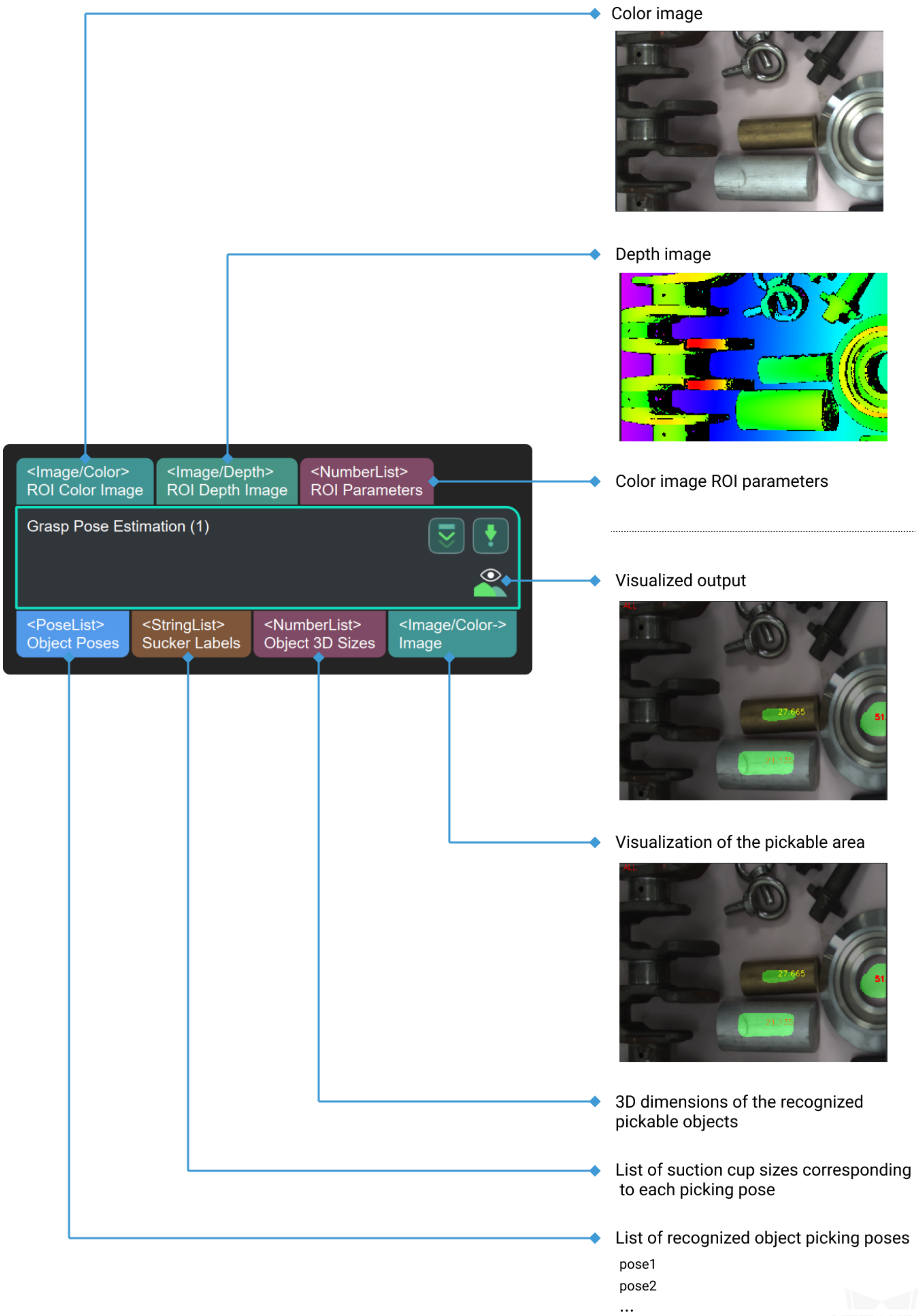
Based on 2D images and 3D depth images, this Step recognizes the pickable objects in the image and outputs corresponding picking poses.



Sample Scenario

This Step is usually used for sorting disparate items heaped or scattered.

Input and Output


Color image
Depth image
<Image/Color>
ROI Color Image
<Image/Depth>
ROI Depth Image
<NumberList>
ROI Parameters
Color image ROI parameters
Grasp Pose Estimation (1)
Visualized output
<PoseList>
Object Poses
<StringList>
Sucker Labels
<NumberList>
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 picking pose
List of recognized object picking poses
pose1
pose2
...

Pixel-wise Graspability Evaluation

4.16.6 Object Detection

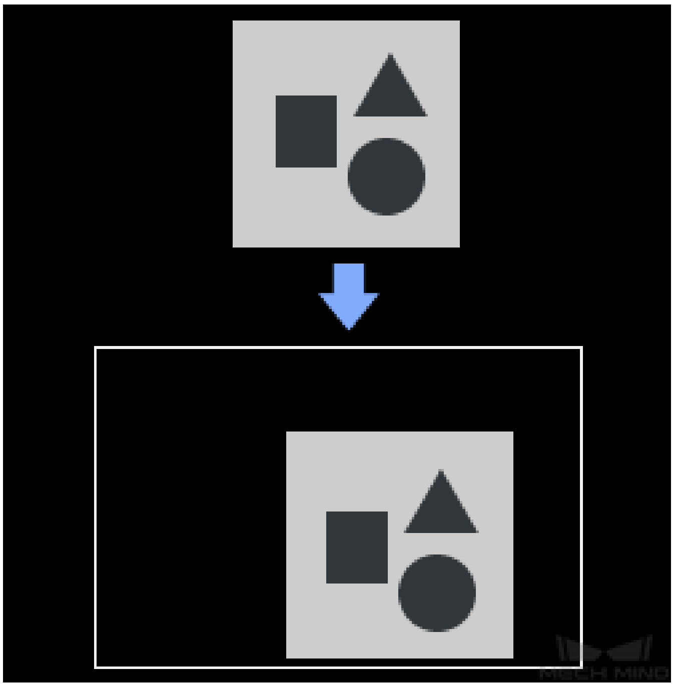
Object Detection

4.16.7 ROI

Recover Scaled Images in 2D ROI

Function

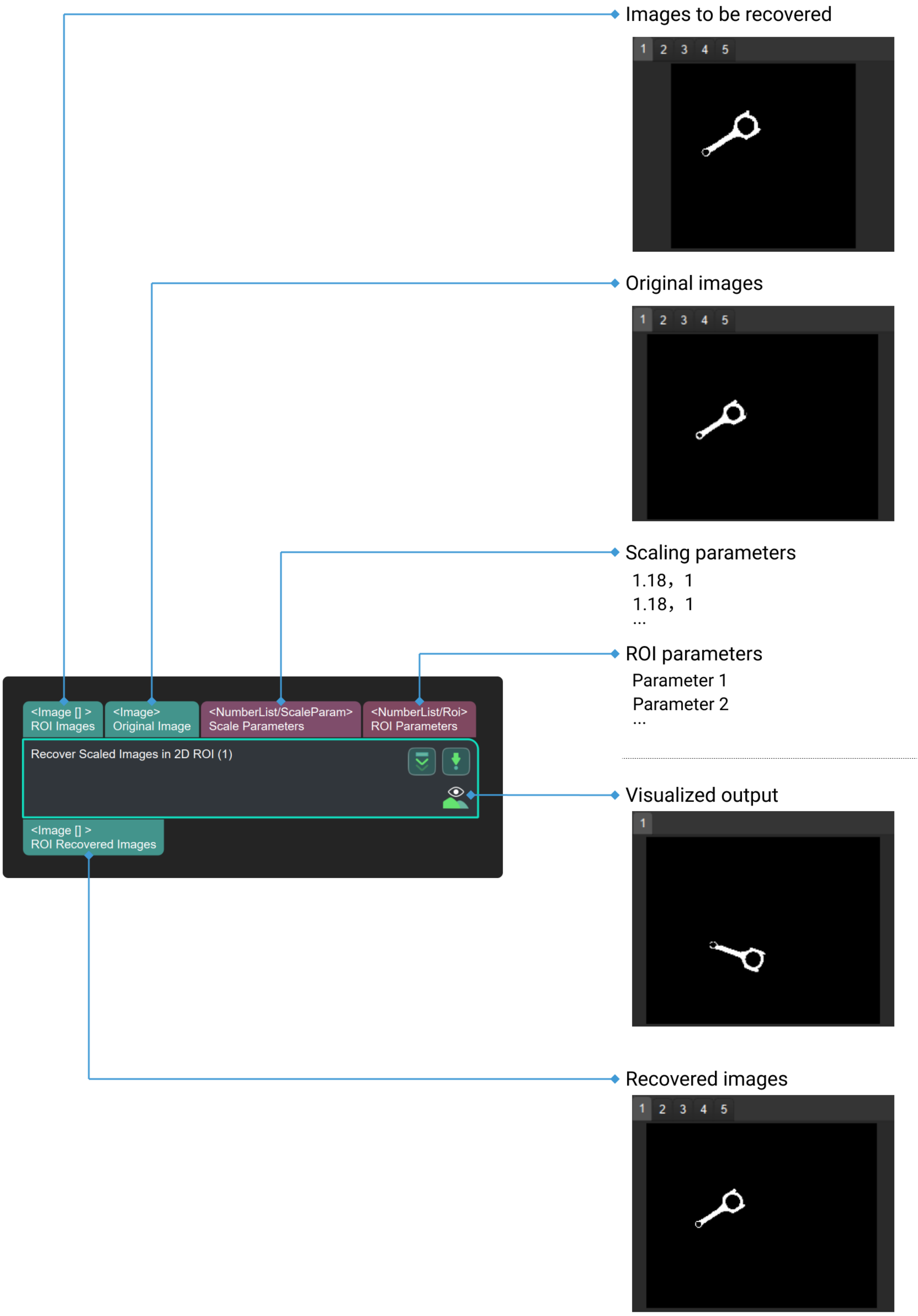
This Step can recover the input scaled 2D image in an ROI to its original size based on scaling parameters, and the recovered image will be displayed on a balck background with a specified size.



Sample Scenario

This Step is usually connected with *Scale Image in 2D ROI* and other Steps related to deep learning.

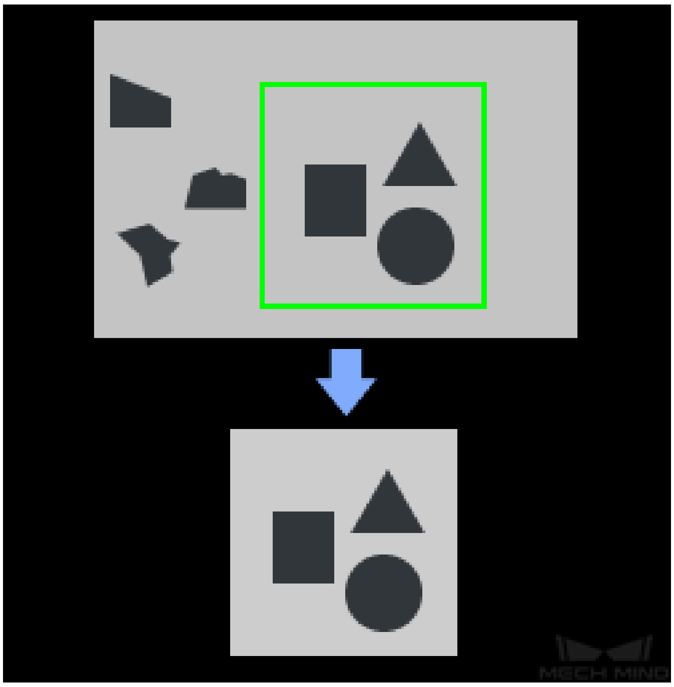
Input and Output


Images to be recovered
Original images
Scaling parameters
1.18，1
1.18，1
...
ROI parameters
Parameter 1
Parameter 2
...
Visualized output
Recovered images
<Image [] >
ROI Images
<Image>
Original Image
<NumberList/ScaleParam>
Scale Parameters
<NumberList/Roi>
ROI Parameters
Recover Scaled Images in 2D ROI (1)
<Image [] >
ROI Recovered Images

Scale Image in 2D ROI

Function

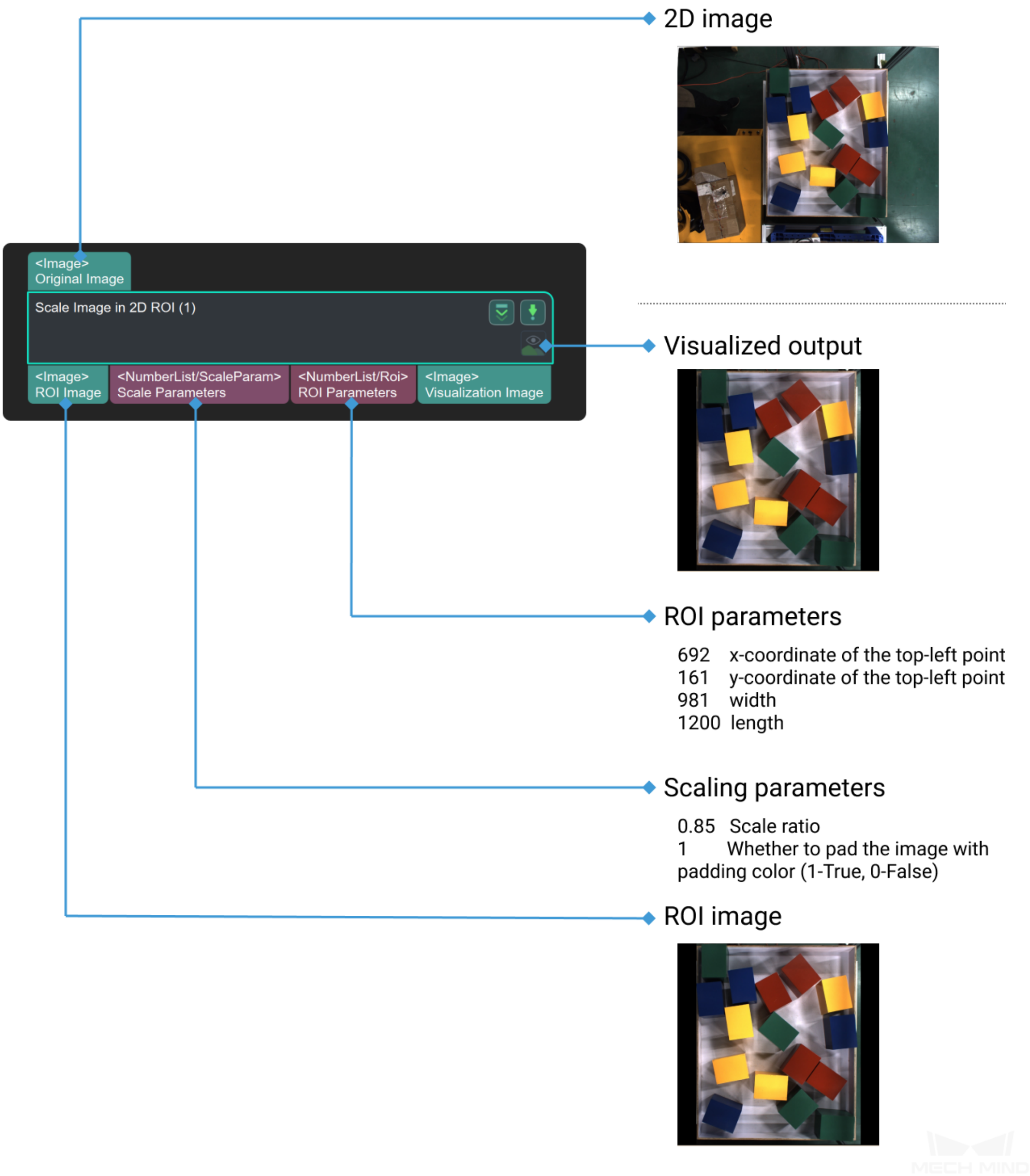
Scale the region of interest (ROI) in an image to a specified size.



Sample 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



Parameters

Color ROI Setting

Default Value: ROIByFile

List of Values: ROIByFile (Read color ROI from file), ROIByParam (Read color

ROI from parameters)

ROIByFile

Color roi File

Instruction: Click *Open an editor* to enter the **Set ROI** window. Create and connect a virtual camera when using parametric simulation. After selecting the image data to be loaded, the selected image appears in the "Image Display" column. If it is an actual application scenario, connect the real camera to collect the image and select the area of interest.

ROIByParam

Start X

Default Value: 0

Instruction: Manually set ROI. It can not be greater than the maximum X value of the image.

Start Y

Default Value: 0

Instruction: Manually set ROI. It can not be greater than the maximum Y value of the image.

Width

Default Value: 0

Instruction: Manually set ROI. It can not be greater than the maximum width of the image.

Height

Default Value: 0

Instruction: Manually set ROI. It can not be greater than the maximum height of the image.

Parameter Adjustment Instructions

As shown in *Figure 2* , 1 is **Start X** and 2 is **Start Y** to determine the upper left corner of the ROI area; 3 is **Width** and 4 is **Heigth** to determine the ROI area.

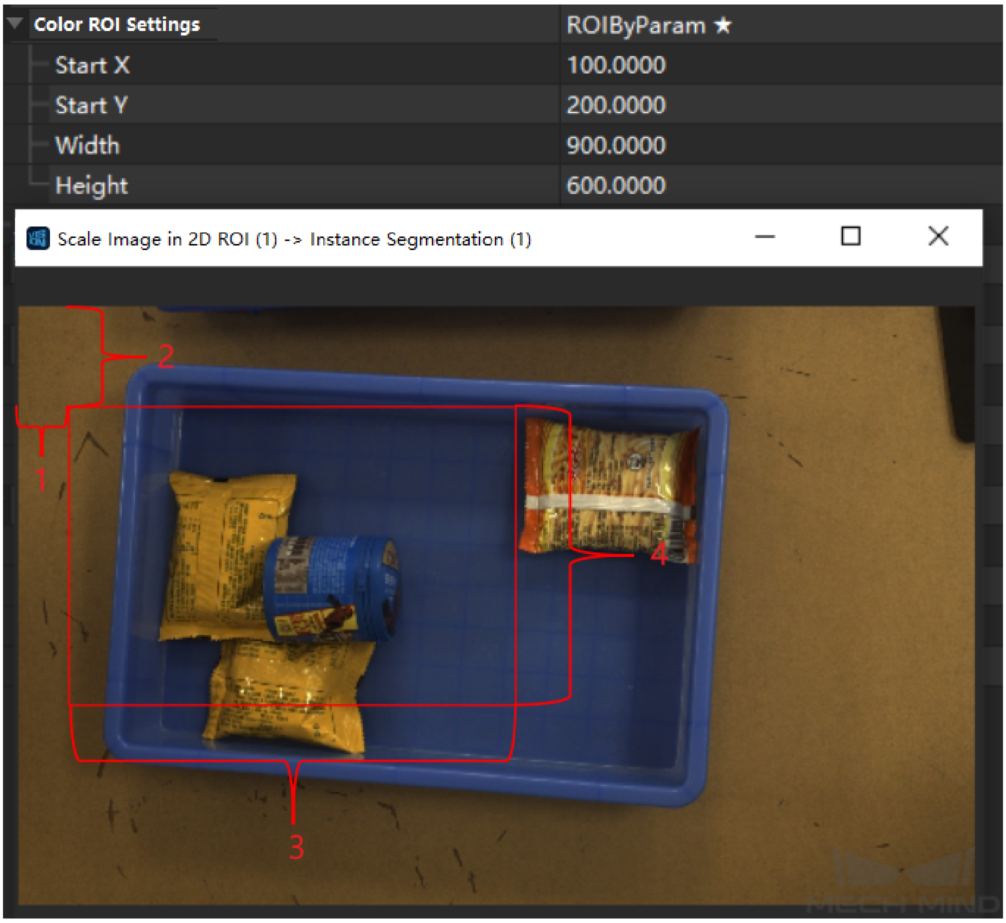


Figure 2 Example of Setting ROI Area With Parameters

Method to Update Color ROI

Default Value: Origin
List of Values: Origin, AdaptivemodifiedRoiOrigin, BoundRect

- Origin: Extract color ROI according to setting.
- AdaptivemodifiedRoiOrigin: Use the border of the image to adjust the color ROI.
- BoundRect: Extract the bounding rectangle area of the input image.

Instruction: Adjust the area corresponding to the input image. **Origin** uses the original image. **AdaptivemodifiedRoiOrigin** adaptively adjusts the center of the non-zero point area of the image to the center of the ROI. **BoundRect** replaces the original image with the circumscribed rectangle of the non-zero point of the image to calculate the zoom ratio. When the position of the object to be detected in the input picture is uncertain and the proportion of the area of the whole picture is small, choose from the latter two according to actual needs.

Color ROI Scaling

Auto Scale

Default Value: True

List of Values: True, False

Instruction: When set to False, the zoom factor can be changed (default is `1`). As shown in the figure below, the left side is the output diagram in the default state, and the right side is the output diagram with the custom zoom factor changed to `0.5`.

Adjustment Example: The steps to calculate the scaling are described in detail in *Custom scaling factor calculation example*.

Figure 3 Changing The Scaling Factor

Ideal Destination Resolution

Same as Input Color ROI & Skip Scaling

Default Value: True

List of Values: True, False

Instruction: Adjust the input image. **Origin** means using the original image. **AdaptivemodifiedRoiOrigin** can adpatively adjust the center of the non-zero area of the image to the center of the ROI. **BoundRect** can replace the original image with the external rectangle of the non-zero area of the image. When the object to be detected accounts for a small proportion of the entire image and its position is uncertain, it is recommended that choose one from the latter two types.

When using the default value, the horizontal length (default is 1024 pixel) and vertical width (default is 1024 pixel) can be modified, and it can be set according to actual engineering requirements on time. In the figure below, the left image is the output image in the default state, and the right is the output image with the length changed to 512 pixels and the vertical width to 1024 pixels.

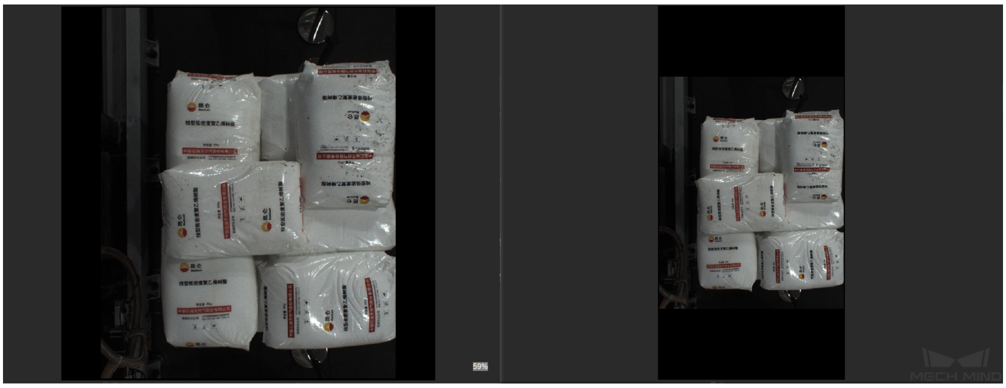


Figure 4 Modifying The Image Size

Padding Color

Padding

Default Value: True

List of Values: True, False

Instruction: The value of `True` fills the missing area with the specified colour. In general it will be filled to black (0,0,0) or grey (128,128,128).

Adjustment Example: As shown in *Figure 5* , the left-hand side shows the result for a fill value of `False` and the right-hand side shows the result for a fill value of (0,255,0).

Figure 5 Image Fill Comparison

r

Default Value: 0

Instruction: The number of red channel. The value range is 0~255. Set according to actual needs.

g

Default Value: 0

Instruction: The number of green channel. The value range is 0~255. Set according to actual needs.

b

Default Value: 0

Instruction: The number of blue channel. The value range is 0~255. Set according to actual needs.

Common Procedures

Procedure of Instance Segmentation

4.16.8 Deep Learning Inference

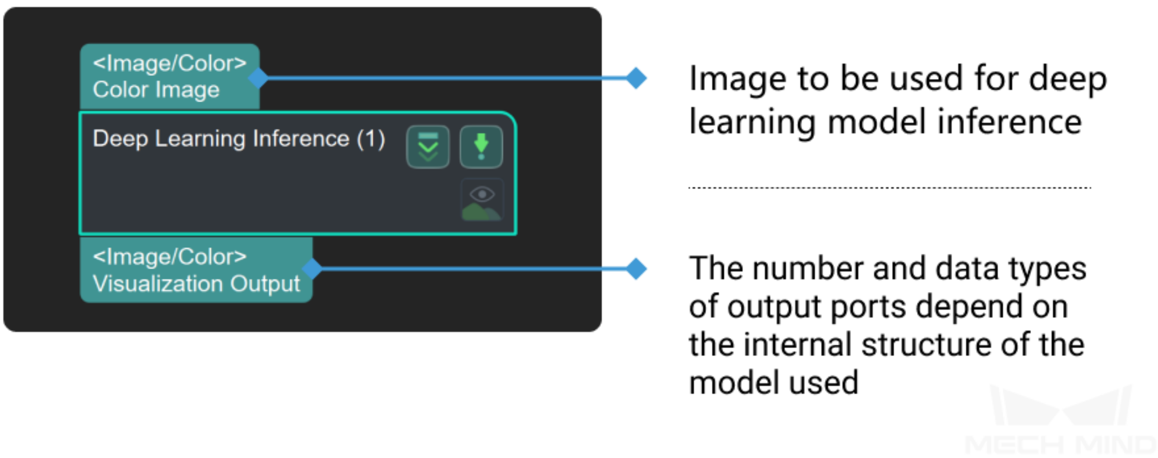
Function

Infer using the model package exported by Mech-DLK and output the inference results.

Sample Scenario

Usually used in classification, target detection, defect detection, and other scenarios, to infer using model packages exported by deep learning software on images from the project site.

Input and Output



4.17 Depalletizing/Palletizing

4.17.1 Pallet Information Recognition

4.17.2 Validate Box Dimensions

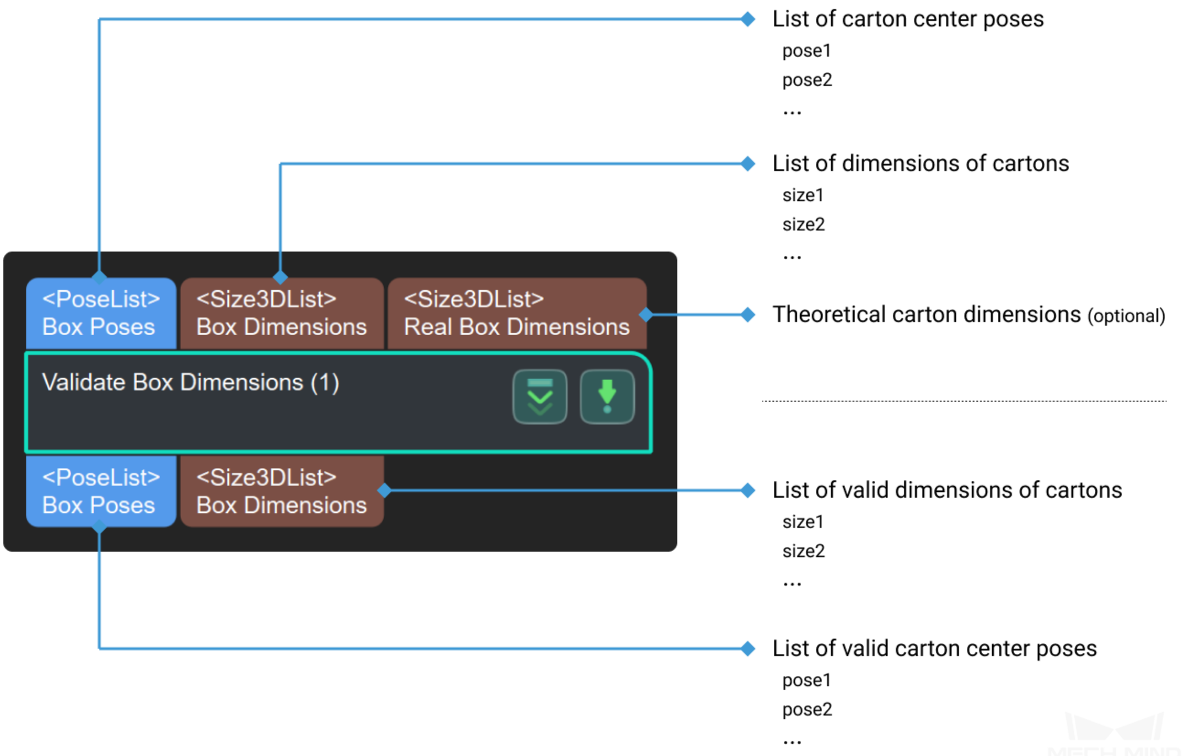
Function

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.

Sample Scenario

This Step is usually used in carton depalletizing applications to determine whether recognized cartons are eligible. It usually follows the Step *Calc Poses and Dimensions from Planar Point Clouds*.

Input and Output



4.17.3 Validate Box Masks

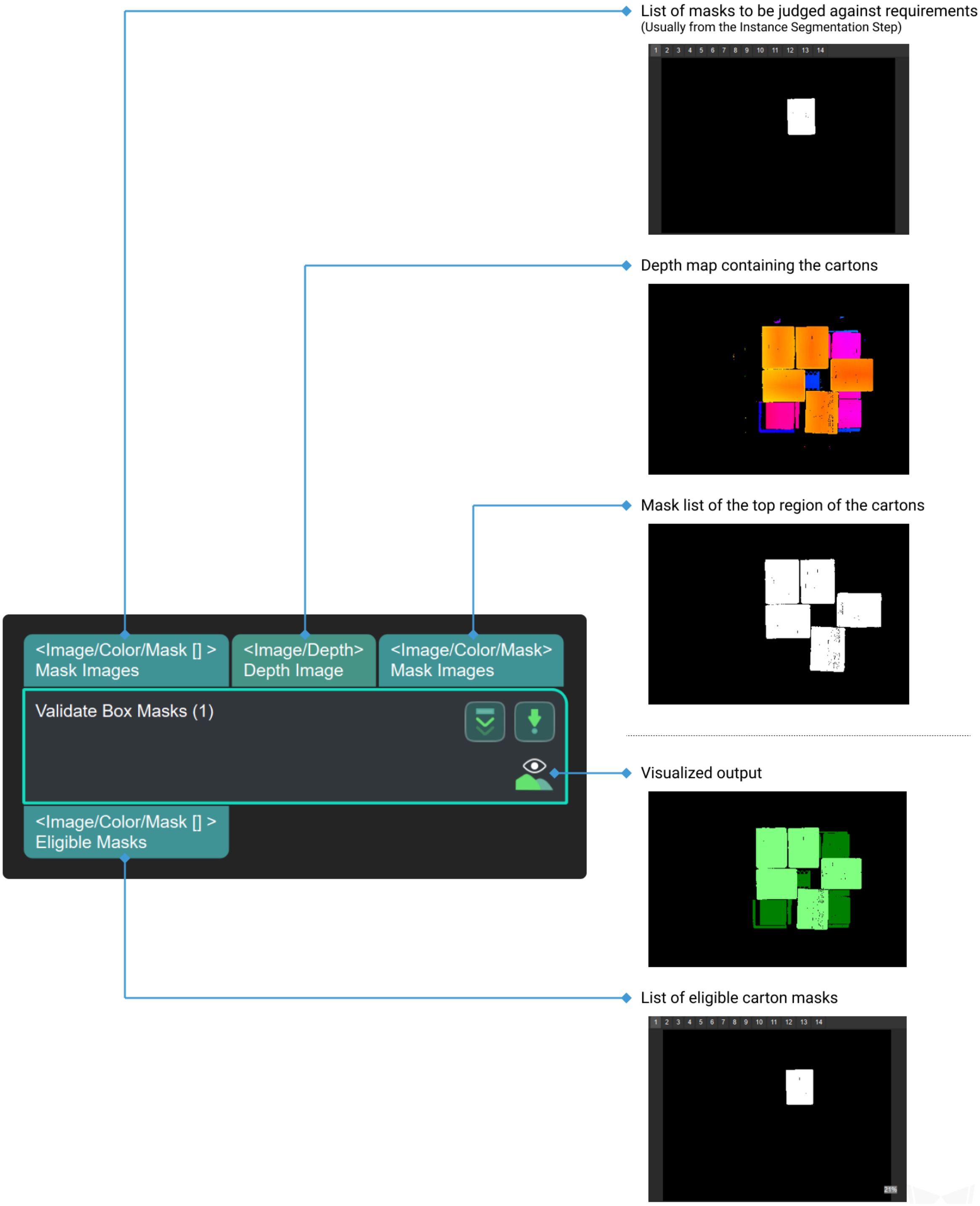
Function

Determine whether the carton masks meet the requirement, and output a list of eligible masks.

Sample 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
<Image/Color/Mask [] >
Mask Images
<Image/Depth>
Depth Image
<Image/Color/Mask>
Mask Images
Validate Box Masks (1)
Visualized output
<Image/Color/Mask [] >
Eligible Masks
List of eligible carton masks

4.18 Drawing

In this chapter, steps that are associated with drawing will be introduced.

4.18.1 Draw Min Circumscribed Rectangles of Masks

4.18.2 Draw Polygon Vertices

4.18.3 Visualize Information on Image

4.19 Label

In this chapter, steps that are associated with Label will be introduced.

4.19.1 Add Labels to Poses

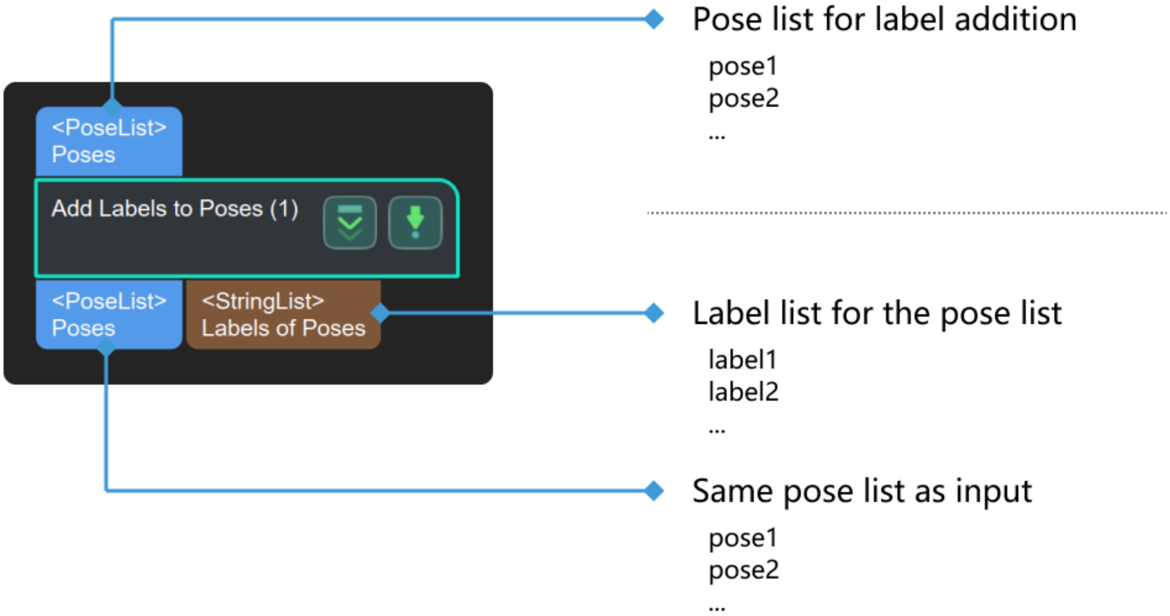
Function

Add the same label to each input pose.

Sample Scenario

General pose processing Step. No fixed usage scenarios.

Input and Output



Parameters

label file

labelFile

Default Value: addlabels.json
Instruction: The absolute path or relative path of the label file which need to include the required label names.

4.19.2 From Variants To Labels

4.19.3 Join Labels

4.19.4 Label Mapping

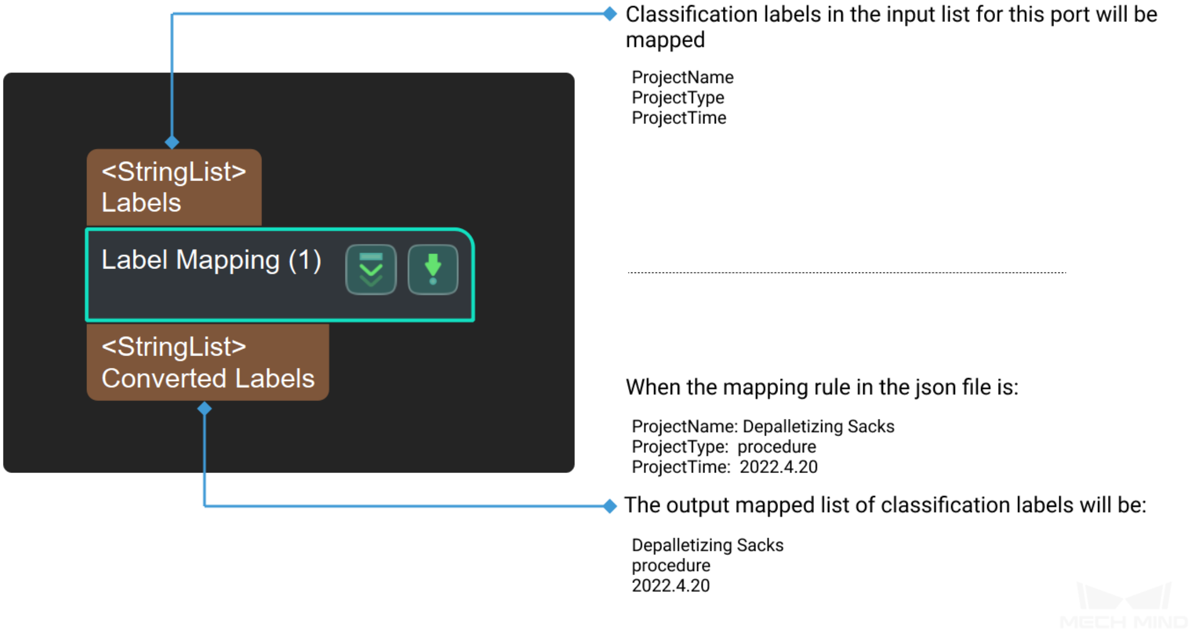
Function

Perform one-to-one mapping of the input classification labels by the mapping specified in the label file set in the **Step Parameters**.

Sample Scenario

This Step is used to rename labels.

Input and Output



Parameters

label file

labelFile

Default Value: labelmap.json

Instruction: Absolute path or relative path of the label file which includes the mapped label.

4.19.5 Stratify Values by Thresholds

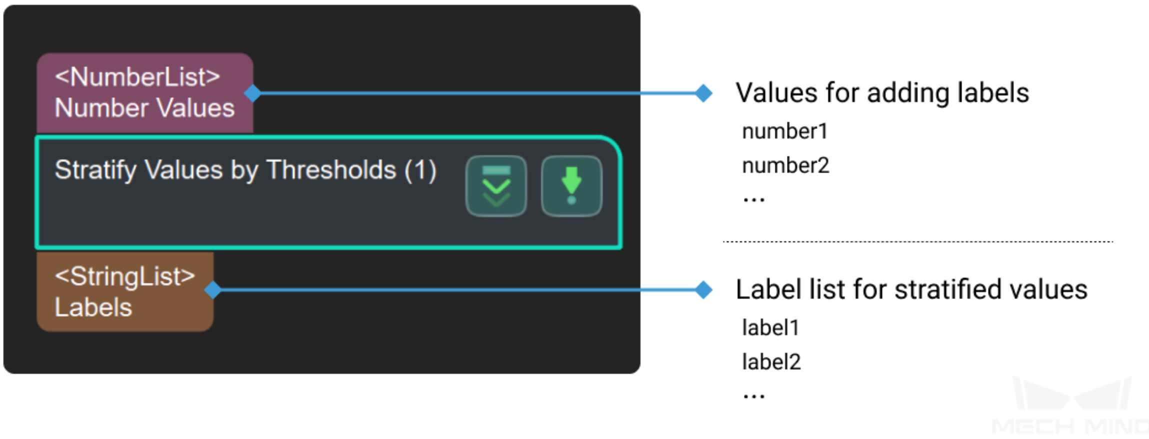
Function

Stratify input values by the set thresholds and output the labels.

Sample Scenario

Sort and label data items by values. Usually used with *Validate Labels and Output Flags* to filter the list of values by value intervals. If you need to divide the list of values 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

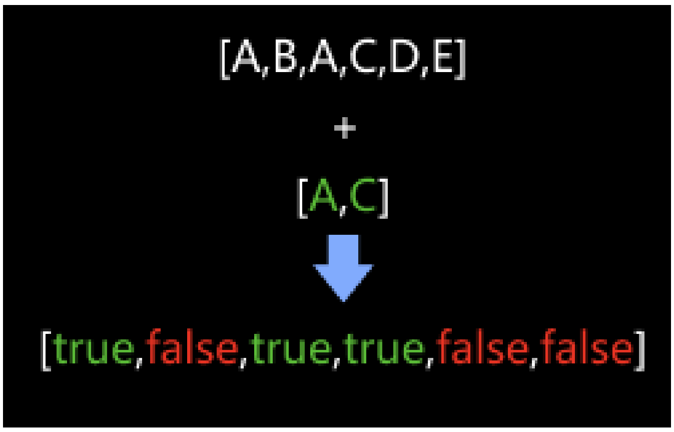
File Settings

- Threshold File Name
- Label File Name

4.19.6 Validate Labels and Output Flags

Function

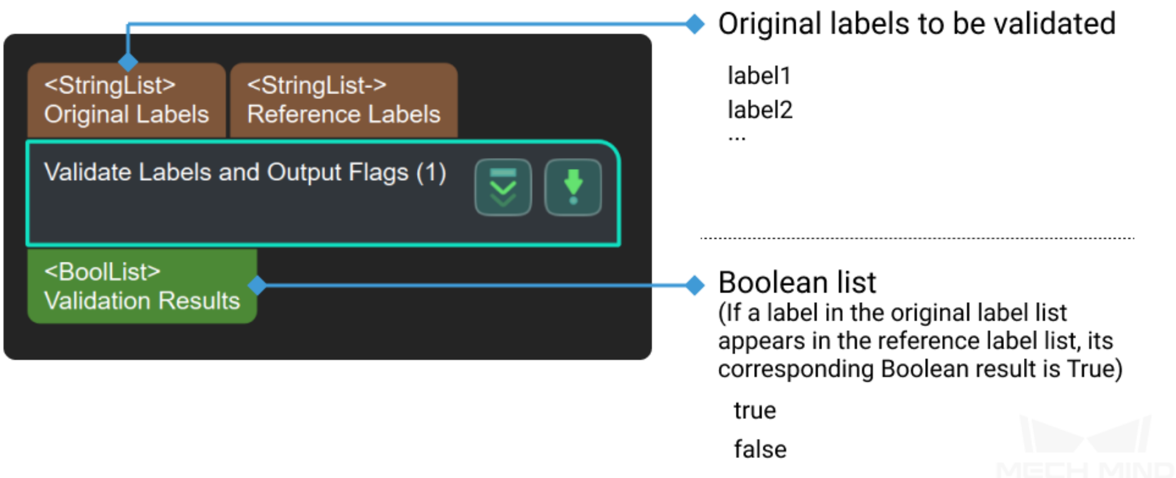
Determine whether each label of the input label list appears in the standard list, and output the corresponding Boolean value list. The standard list comes from a label file set in the parameters or the input standard label list.



Sample 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



Parameters

Label file name

Default Value: labelFilter.json

Instruction: This file contains the label that is used to validate the input. If the label file or input standard label contains the 1st input label list, output True.

4.20 Map

In this chapter, steps that are associated with Map will be introduced.

4.20.1 Conversion

2D Poses to 3D Poses Base Orthographic Projection

From 3D Poses to 2D Poses

From NumberList to Size3DList

From Numbers to Variants

From Variant to Variants

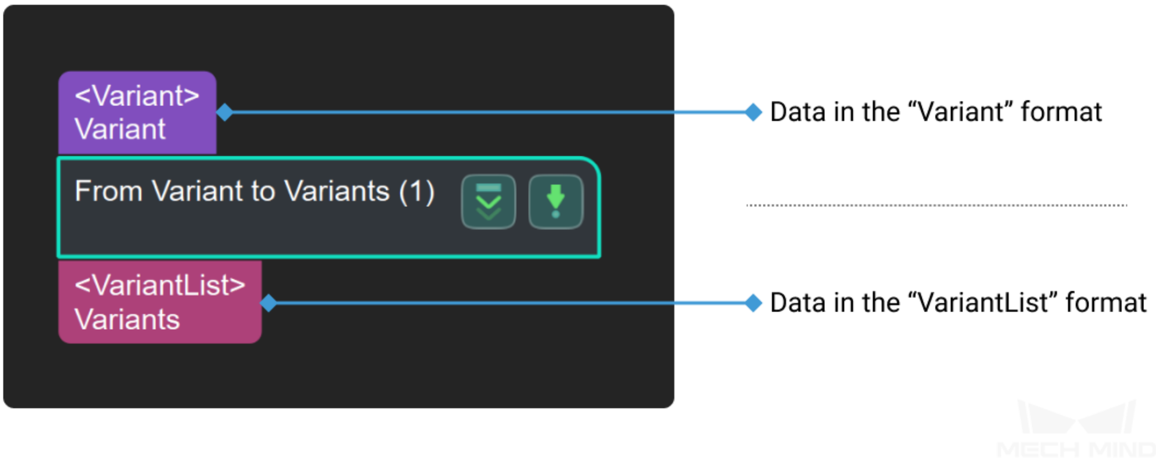
Function

Convert input from the “Variant” format to the “VariantList” format.

Sample Scenario

A general data type conversion Step. For example, used between the Step *Count Elements in Specified Dimension in Data List* and the Step *From Variants to Numbers*, converting the number of a list’ s elements into another list for output.

Input and Output



From Variants to Numbers

From Variants to Variant

4.20.2 Image - Image

Apply Masks to Image

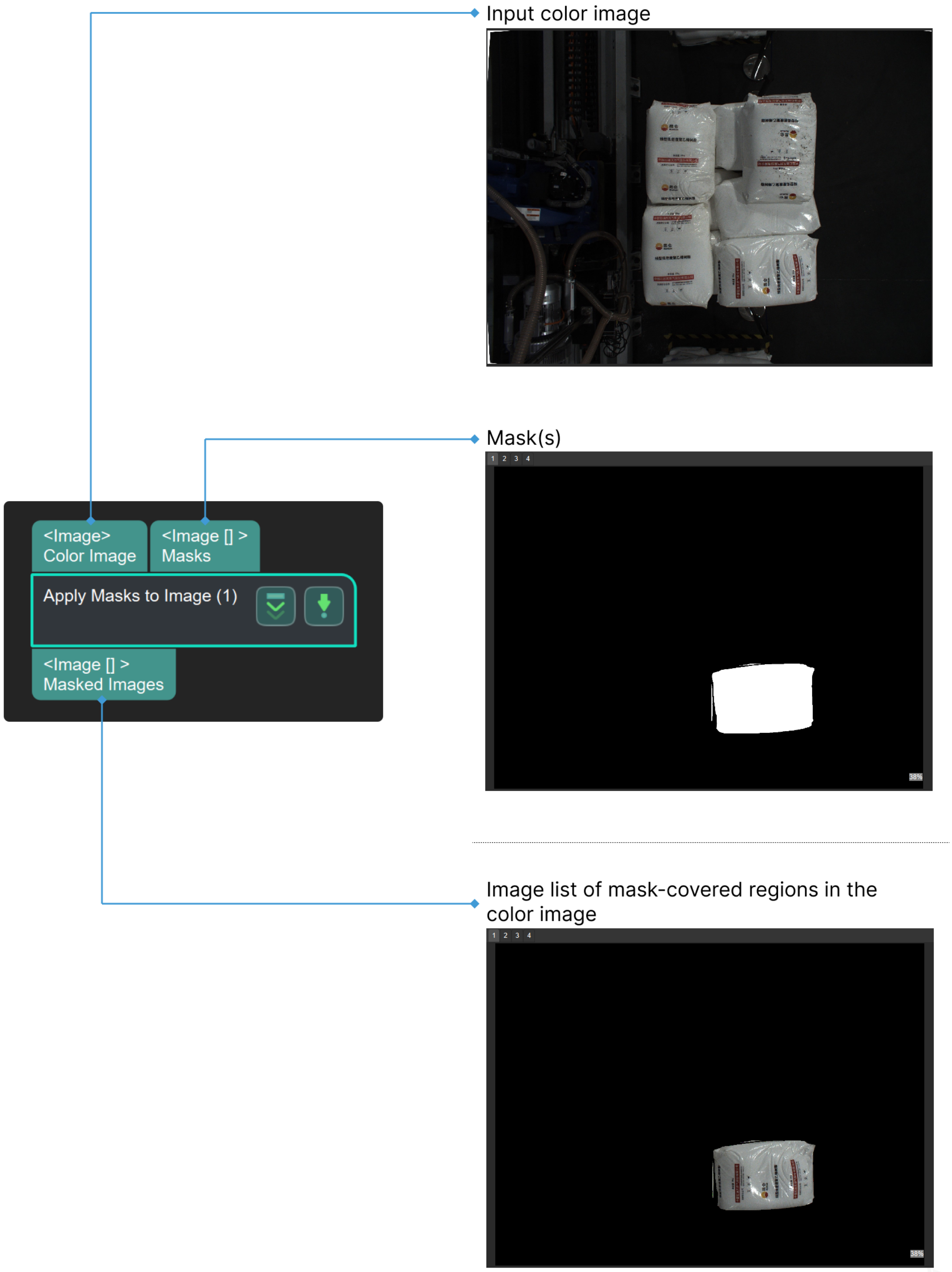
Function

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.

Sample 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


Input color image
Mask(s)
<Image>
Color Image
<Image [] >
Masks
Apply Masks to Image (1)
<Image [] >
Masked Images
Image list of mask-covered regions in the color image

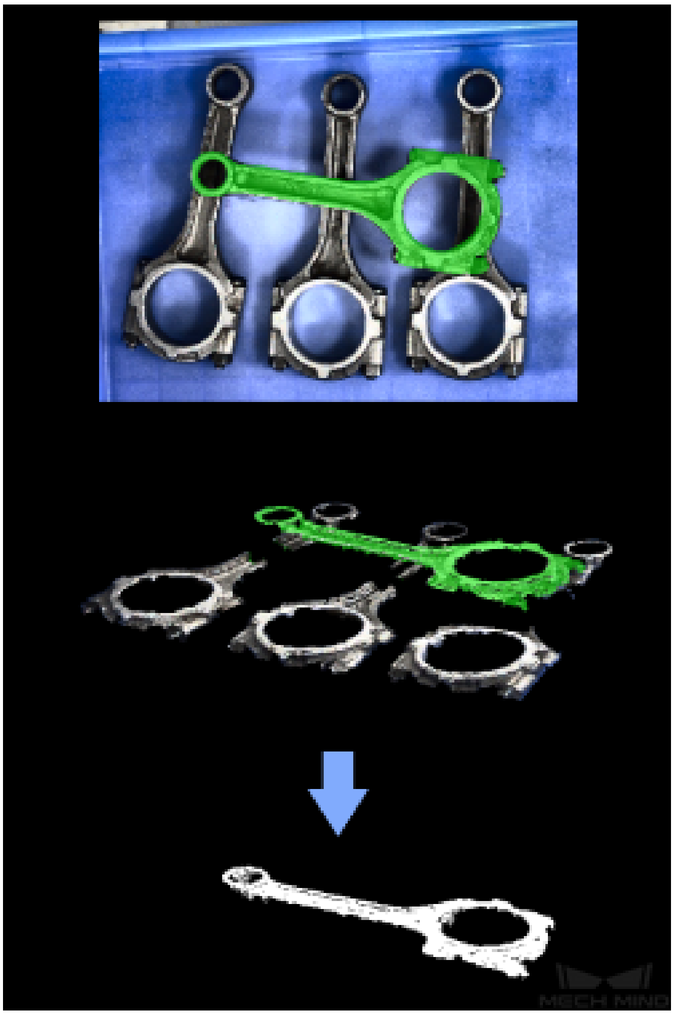
Map Depth to Rgb

4.20.3 Image - Point Cloud

Apply Masks to Point Cloud

Function

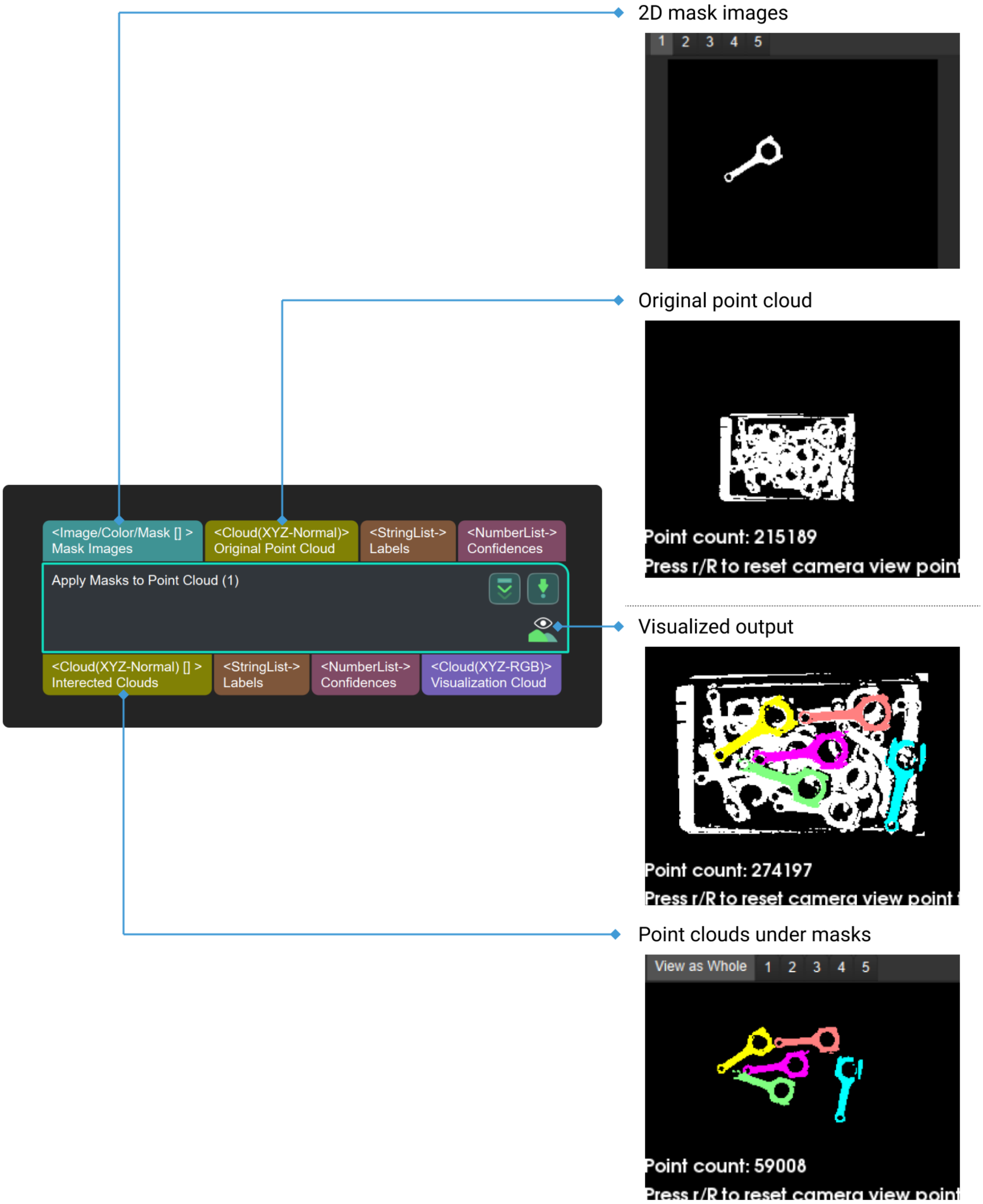
Apply 2D mask to point cloud, and extract points that are in the area of each mask. The points outside the mask will be eliminated.



Sample 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) [] >
Interceted 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

Convert 2D Points to 3D Points

From Cloud (XYZ-RGB) to Color Image

Orthographic Projection

Orthographic Projection Inverse

Project 3D Point Cloud to 2D Image

Function

Convert a 3D point cloud to a 2D image according to the specified projection type.

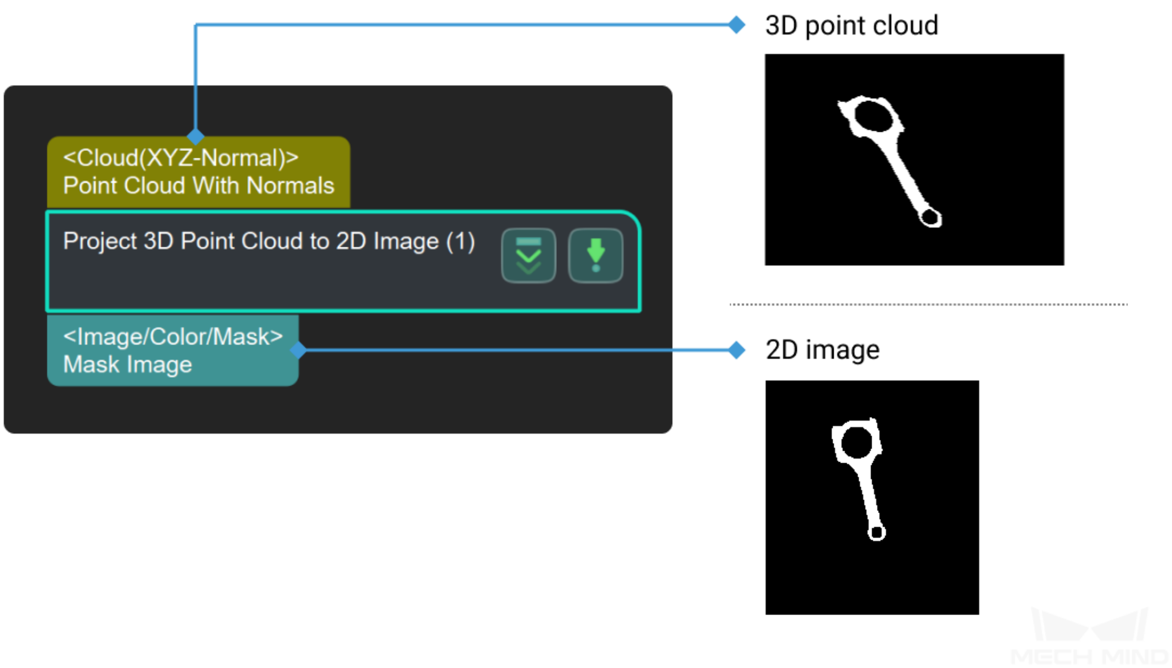
Sample Scenario

For point cloud preprocessing. First, project the 3D point cloud onto the 2D image. Then, after using 2D image processing methods (such as 2D morphological operations), convert the 2D image back to 3D point clouds through *Apply Masks to Point Cloud*.

For picking scenarios, it is recommended to set the **Projection Type** parameter to **PerspectiveProjection**.

For measurement scenarios, it is recommended to use *Orthographic Projection* instead.

Input and Output



Parameters

orthogonalProj

useOrthogonalProj

Default Value: False

List of Values: True, False

Instruction: Use orthogonal projection to project a 3D point cloud onto a 2D plane

projectAlongObjZAxis

Default Value: True

List of Values: True, False

Instruction: Select the projection axis according to actual needs.

closingOperateSize

Default Value: 7

Instruction: Used to bridge narrow discontinuities, small ravines and holes. The larger the core size, the more obvious the effect.

perspectiveProj

imageType

Default Value: Color

List of Values: Color, Depth

Instructions: When the color camera map and depth camera used in one project, you can choose to project to the corresponding camera.

dilateSize

Default Value: 21

Instruction: Expand the highlight or white part of the image, the larger the core size, the more obvious the effect.

erodeSize

Default Value: 21

Instruction: Reduce or thin the highlight or white part of the image, the larger the core size, the more obvious the effect.

Varying Normal Area

4.21 Mask Processing

In this chapter, steps that are associated with Mask Processing will be introduced.

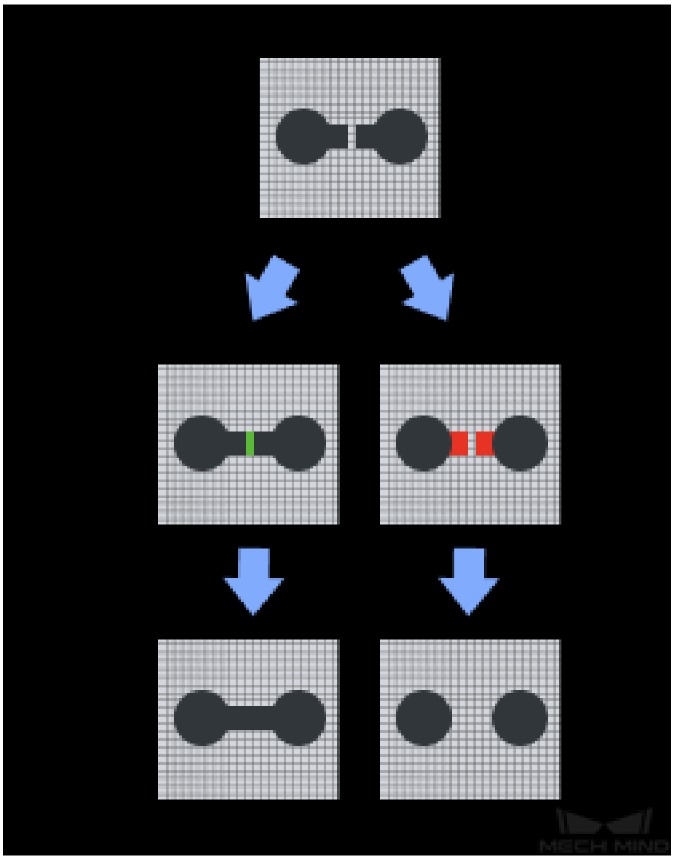
4.21.1 Add/Remove Areas in Masks

Mask Filter

Morphological Transformations

Description

Perocess an image using morphological operations.

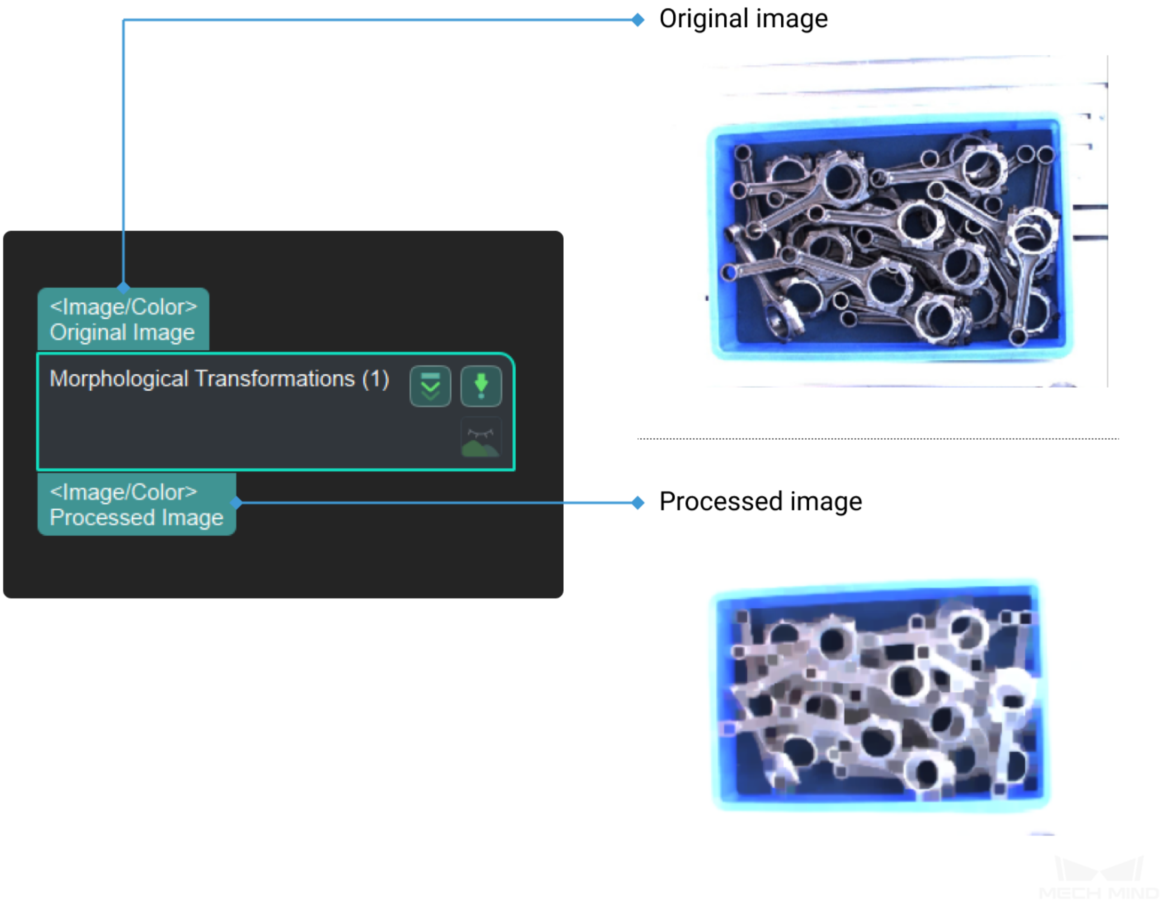


Sample Scenario

This Step is usually used for image pre-processing when the quality of images is not ideal and further adjustment is needed.

Apply operations such as dilation, erosion, and opening and closing to denoise, fill, etc. based on actual needs.

Input and Output



Parameters

morphOperatorType

List of Values: ClosingOperator, DilateOperator, ErodeOperator, OpeningOperator, Skelton

Dilate Operator

Expand the highlighted or white parts of the image, make objects more visible and fills in small holes in objects.

kernelSize

Default Value: 3

Instruction: The larger the core, the stronger the dilate effect, set according to the detection requirements.

morphShapes

Default Value: Rectangle (MORPH_RECT)

List of Values: rectangle (MORPH_RECT), cross (MORPH_CROSS), ellipse (MORPH_ELLIPSE)

Instruction: Image details: MORPH_RECT <MORPH_CROSS <MORPH_ELLIPSE, the processing time: MORPH_RECT

<MORPH_CROSS <MORPH_ELLIPSE. It is recommended to use the default value unless there are special requirements.

Erode Operator

The highlight or white part of the image is reduced and thinned. It removes islands and small objects so that only substantive objects remain..

kernelSize

Default Value: 3

Instruction: The larger the core, the stronger the erode effect, set according to the detection requirements.

morphShapes

Default Value: Rectangle (MORPH_RECT)

List of Values: rectangle (MORPH_RECT), cross (MORPH_CROSS), ellipse (MORPH_ELLIPSE)

Instruction: Image details: MORPH_RECT <MORPH_CROSS <MORPH_ELLIPSE, the processing time: MORPH_RECT <MORPH_CROSS <MORPH_ELLIPSE. It is recommended to use the default value unless there are special requirements.

Opening Operator

The opening operation erodes an image and then dilates the eroded image. It can be used to remove small objects from an image while preserving the shape and size of larger objects in the image.

kernelSize

Default Value: 20

morphShapes

Default Value: Rectangle (MORPH_RECT)

List of Values: rectangle (MORPH_RECT), cross (MORPH_CROSS), ellipse (MORPH_ELLIPSE)

Instruction: Image details: MORPH_RECT <MORPH_CROSS <MORPH_ELLIPSE, the processing time: MORPH_RECT <MORPH_CROSS <MORPH_ELLIPSE. It is recommended to use the default value unless there are special requirements.

Closing Operator

The closing operation dilates an image and then erodes the dilated image. It can be used to fill small holes from an image while preserving the shape and size of the objects in the image.

kernelSize

Default Value: 20

morphShapes

Default Value: Rectangle (MORPH_RECT)

List of Values: rectangle (MORPH_RECT), cross (MORPH_CROSS), ellipse (MORPH_ELLIPSE)

Instruction: Image details: MORPH_RECT <MORPH_CROSS <MORPH_ELLIPSE, the processing time: MORPH_RECT <MORPH_CROSS <MORPH_ELLIPSE. It is recommended to use the default value unless there are special requirements.

Skeleton

Extracting the skeleton of the binary map to refine the image. It is mainly used to refine the trajectory when automatically extracting the trajectory.

Remove Noise in Masks

4.21.2 Calc Mask Property

Calc Areas of Masks

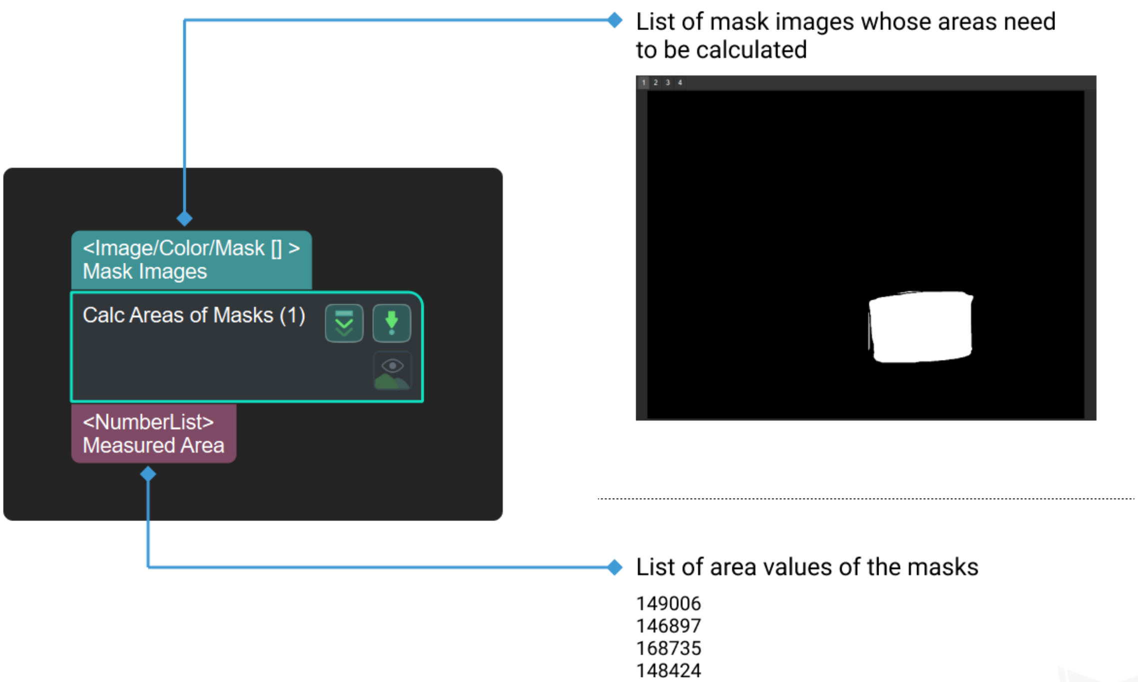
Function

Calculate the areas of multiple mask images.

Sample Scenario

This Step calculates mask areas which facilitate follow-up sorting, filtering, and other operations.

Input and Output



Calc Center Point Of Non-zero Areas

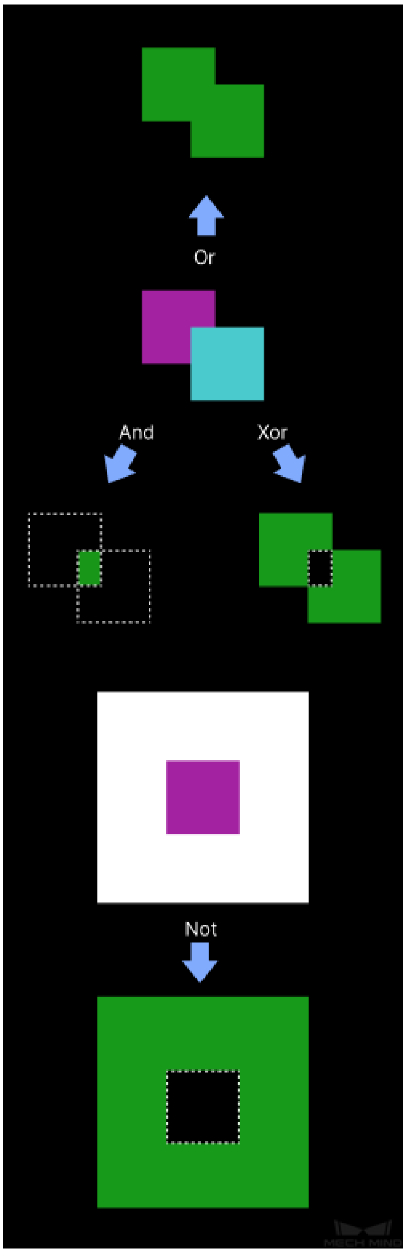
Calc Mask Distance

4.21.3 Mask Logical Operation

Mask Logical Operation

Function

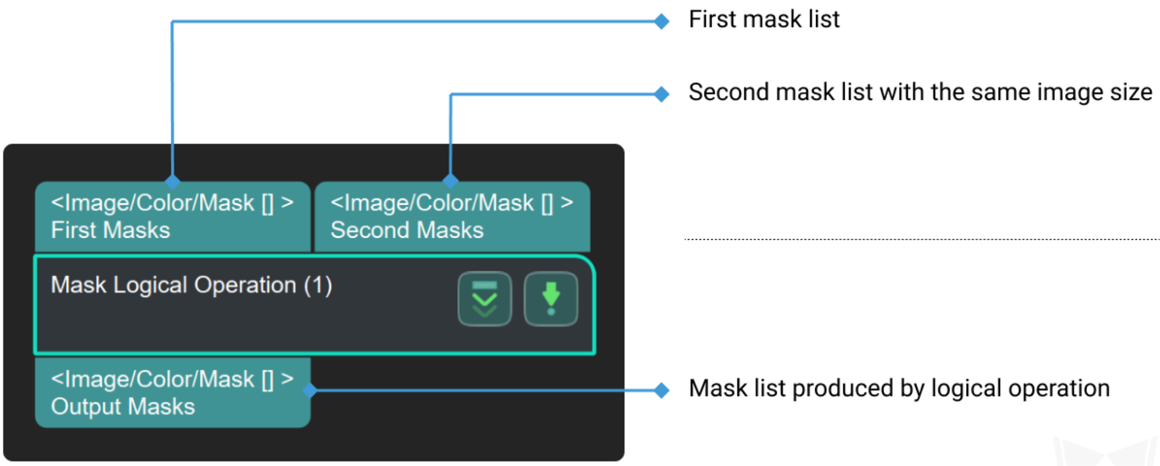
This Step is used to perform logical operations on two mask lists of the same size. It can combine two mask lists (“Or”Operation), extract the shared part of the two mask lists (“And”Operation) or remove the shared part (“Xor” Operation).



Sample Scenario

This Step is widely used in all scenarios concerning mask logical operations.

Input and Output



Parameters

operation

logicalOperator

Instruction: This parameter could be used to choose the type of logical operations, And, Or, Xor and Not included. When 'And', 'Or' or 'Xor' is used, there will be two input masks, while only one input mask for 'Not'.

Default Value: And

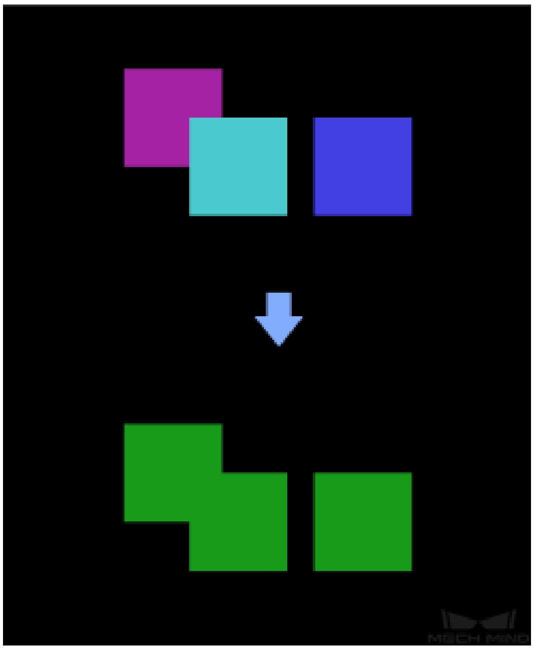
Suggested Value: To set according to the real scenarios.

List of Values: And, Or, Xor, Not.

Merge Mask Images

Function

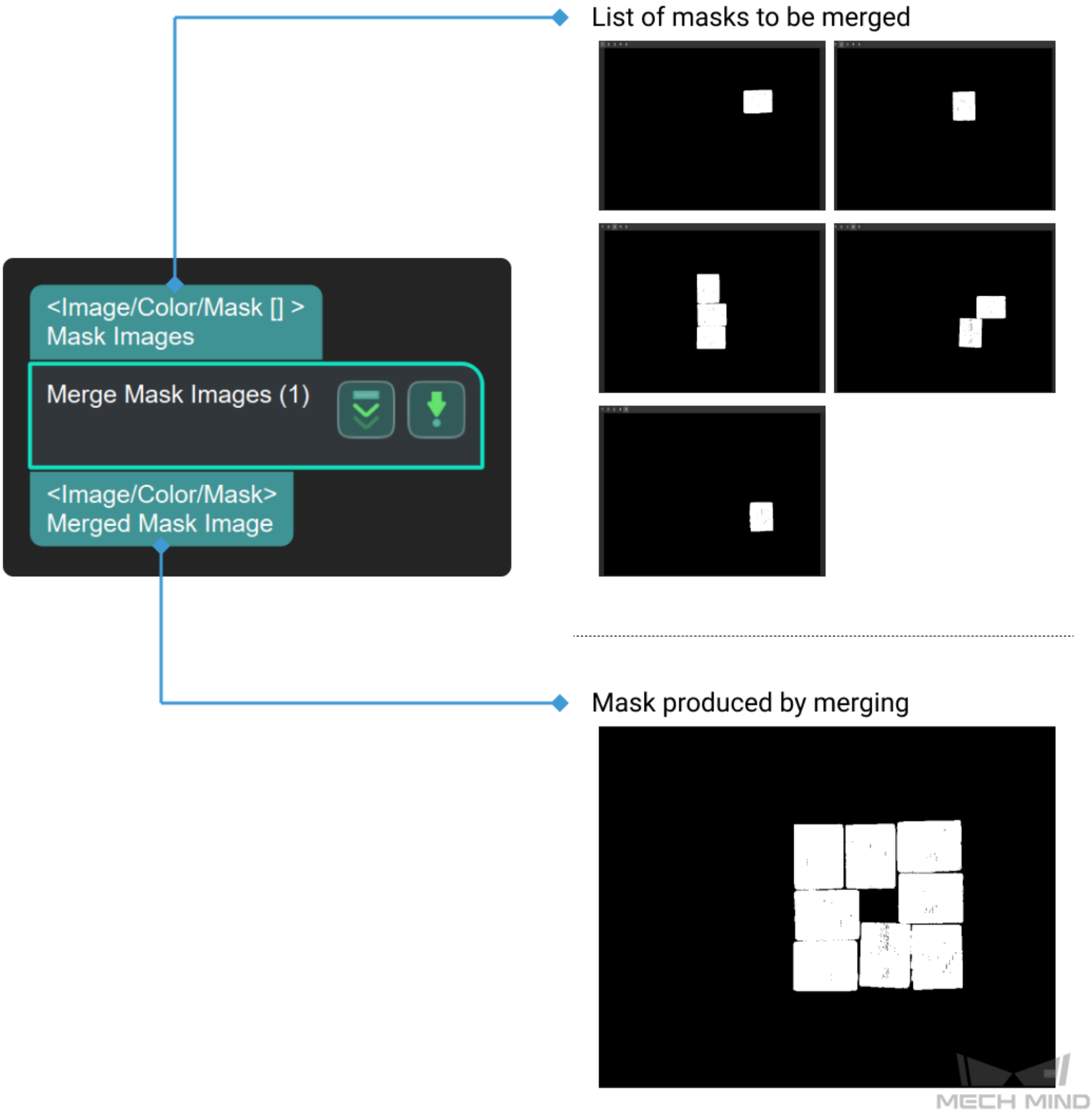
Merge multiple masks into on mask image.



Sample 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



4.21.4 Validate Masks by Poses

Filter Masks by Poses

Validate Masks Containing 2D Poses

4.21.5 Others

Mask Cluster

Mask Gridding

4.22 Measuring

This chapter will introduce the instructions for the measurement related steps

4.22.1 2.5D Measuring

Measure Height Difference Point to Point

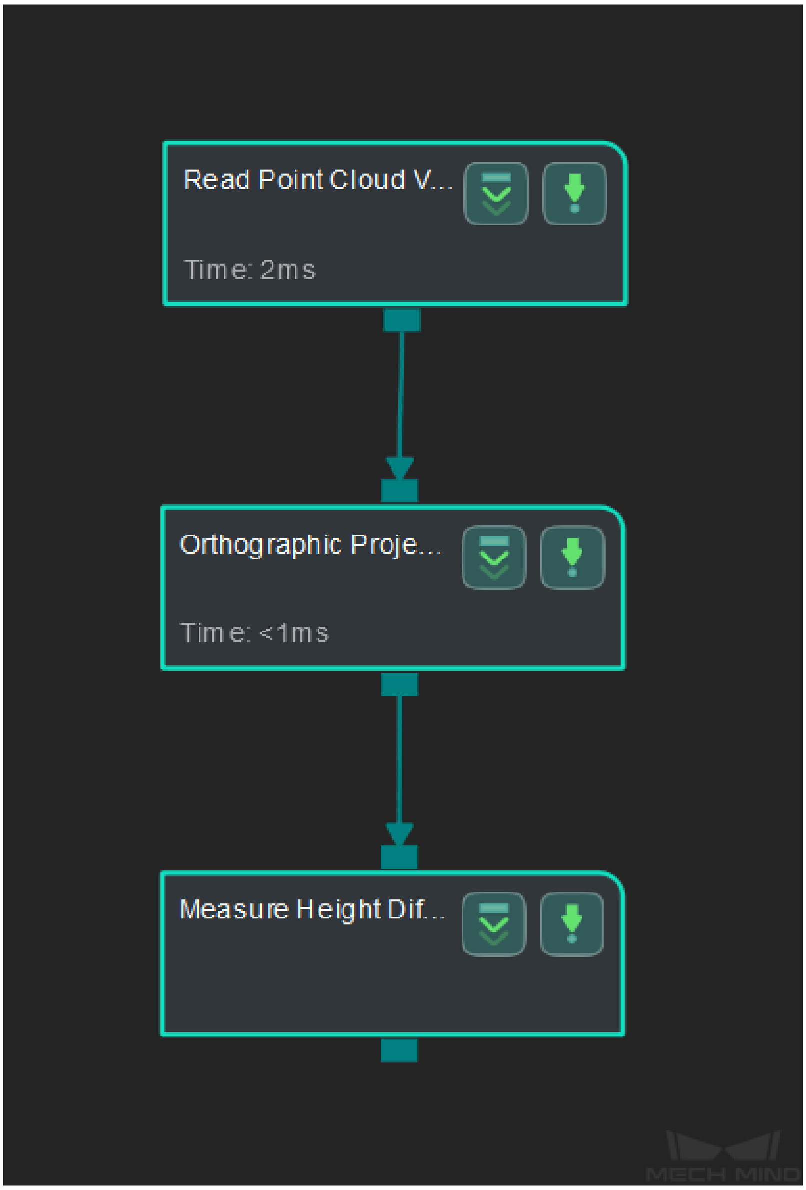
Function

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.

Hint: Before using this Step, please refer to *Getting Started with Measurememt Mode* to learn about basics of **Measurement Mode**.

Sample Application



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 on *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 on *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

Note: Please prepare the point cloud of the object to be measured in advance.

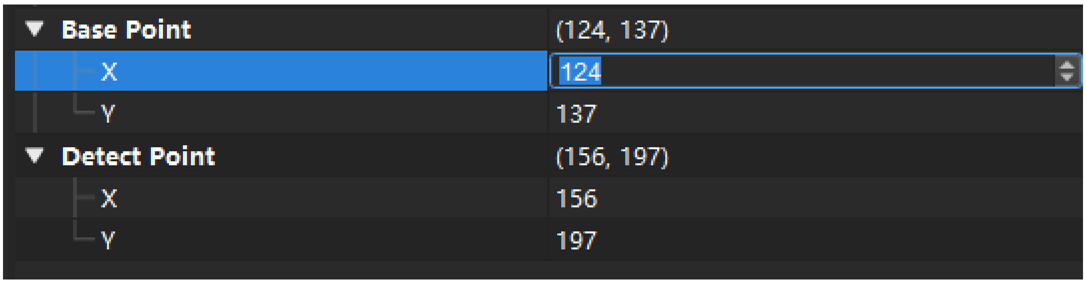
- Configure in *Read Point Cloud V2*

  Click on the Step, and configure the **Point Cloud Source** and other parameters in the **Step Parameters** panel.

- Configure in *Measure Height Difference Point to Point*

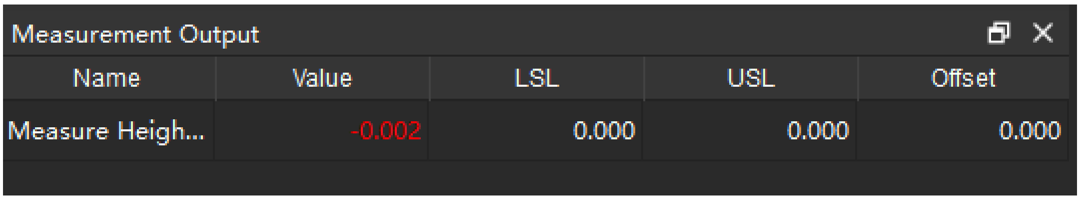
  The baseline point and detect point are located in the upper left corner in the sketchpad area by default. Please drag them onto the point cloud.

You can also adjust the position of the points in the **Step Parameters** panel, as shown below.



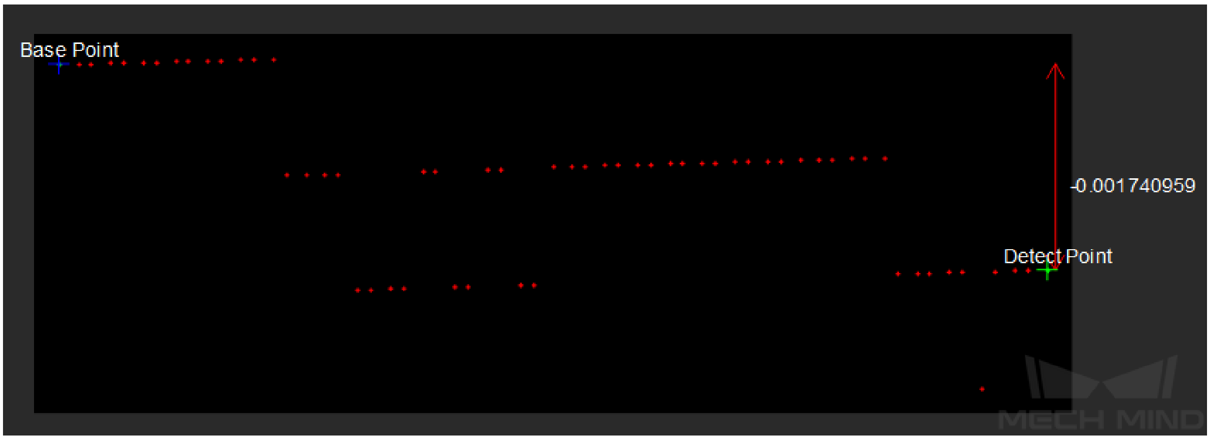
Read the result

After completeteing the configuration, click on *Run* to run the project. Click on *Measure Height Difference Point to Point* and check the measurement result in the **Measurement Output** panel.



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.



Hint: 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.

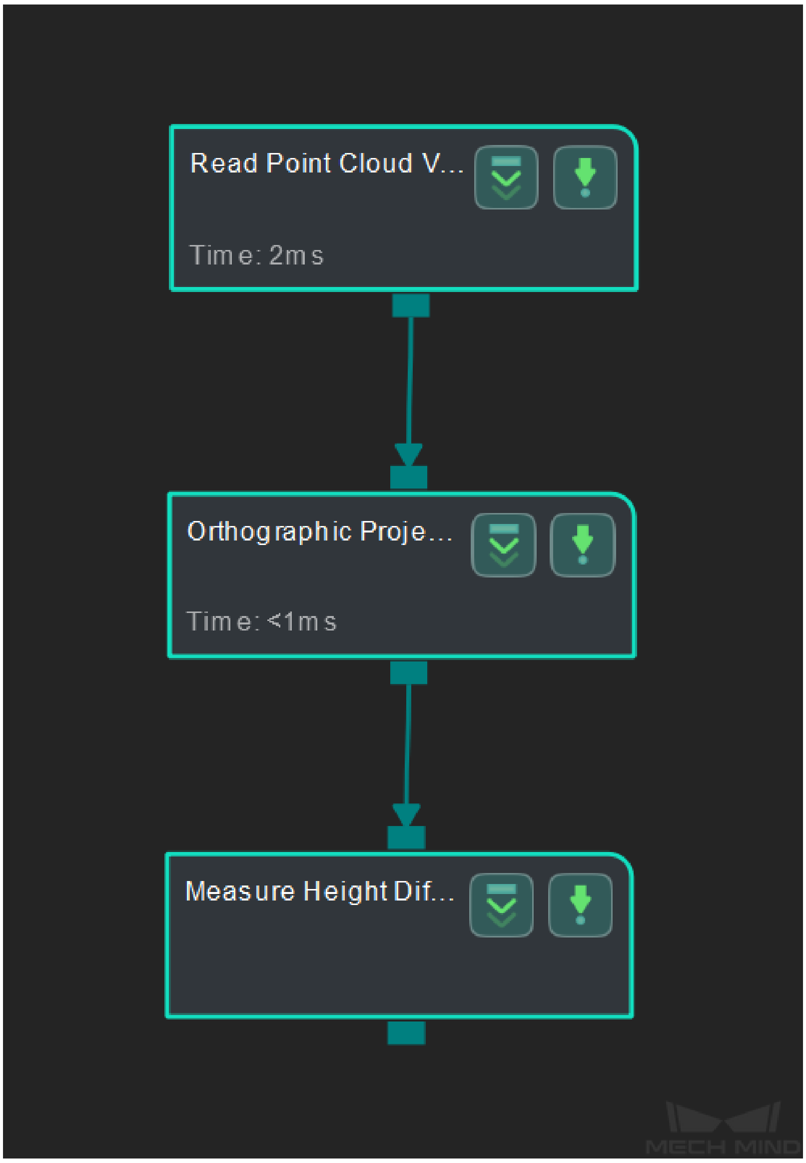
Measure Height Differences Points to Baseline

Function

This Step is used to calculate the height difference from points to the baseline.

Hint: Before using this Step, please refer to *Getting Started with Measurememt 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 on *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 on *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

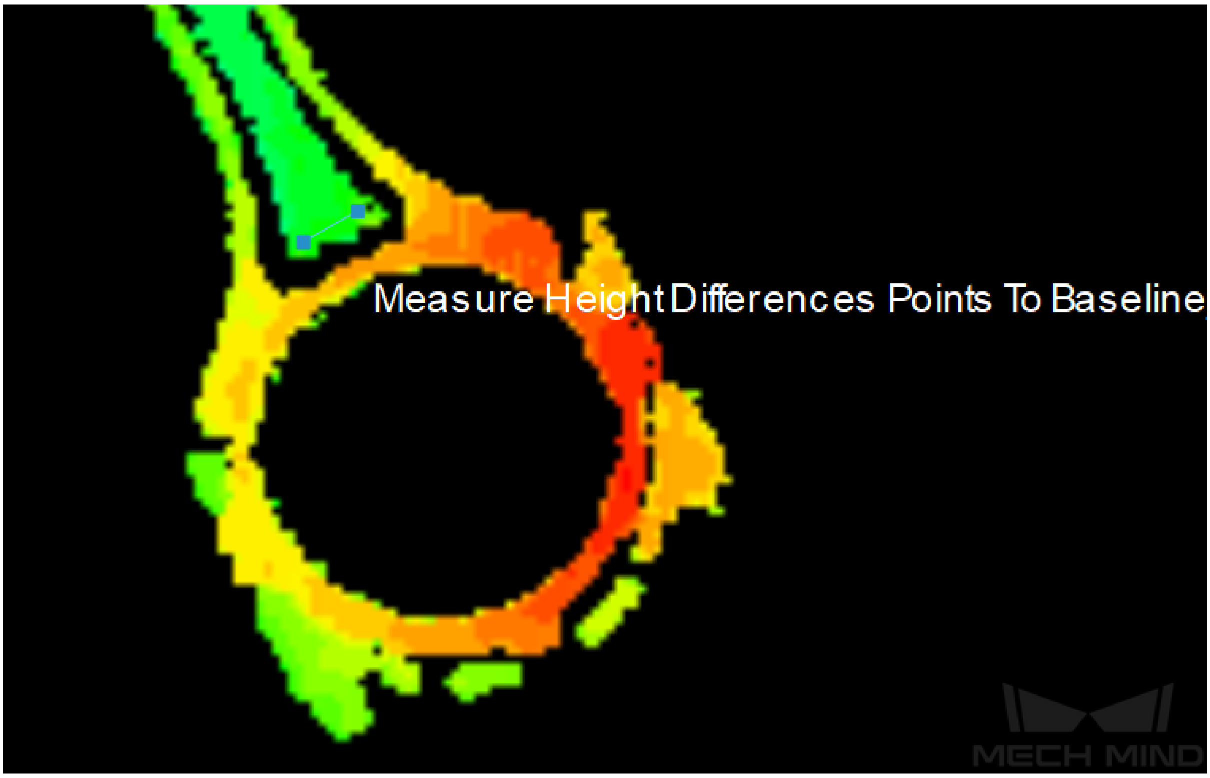
Note: Please prepare the point cloud of the object to be measured in advance.

- Configure in *Read Point Cloud V2*

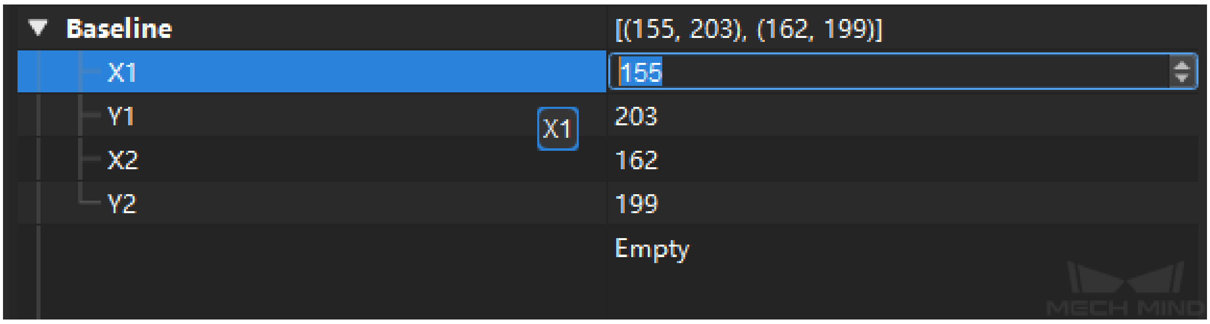
  Click on the Step, and configure the **Point Cloud Source** and other parameters in the **Step Parameters** panel.

- Configure in *Measure Height Differences Points to Baseline*

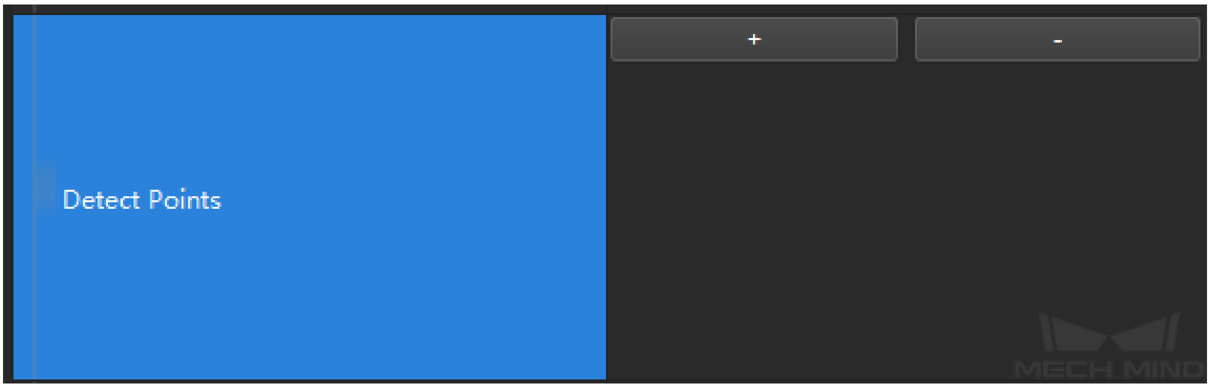
  The baseline is located in the upper left corner in the sketchpad area by default, drag it onto the point cloud and adjust the length.



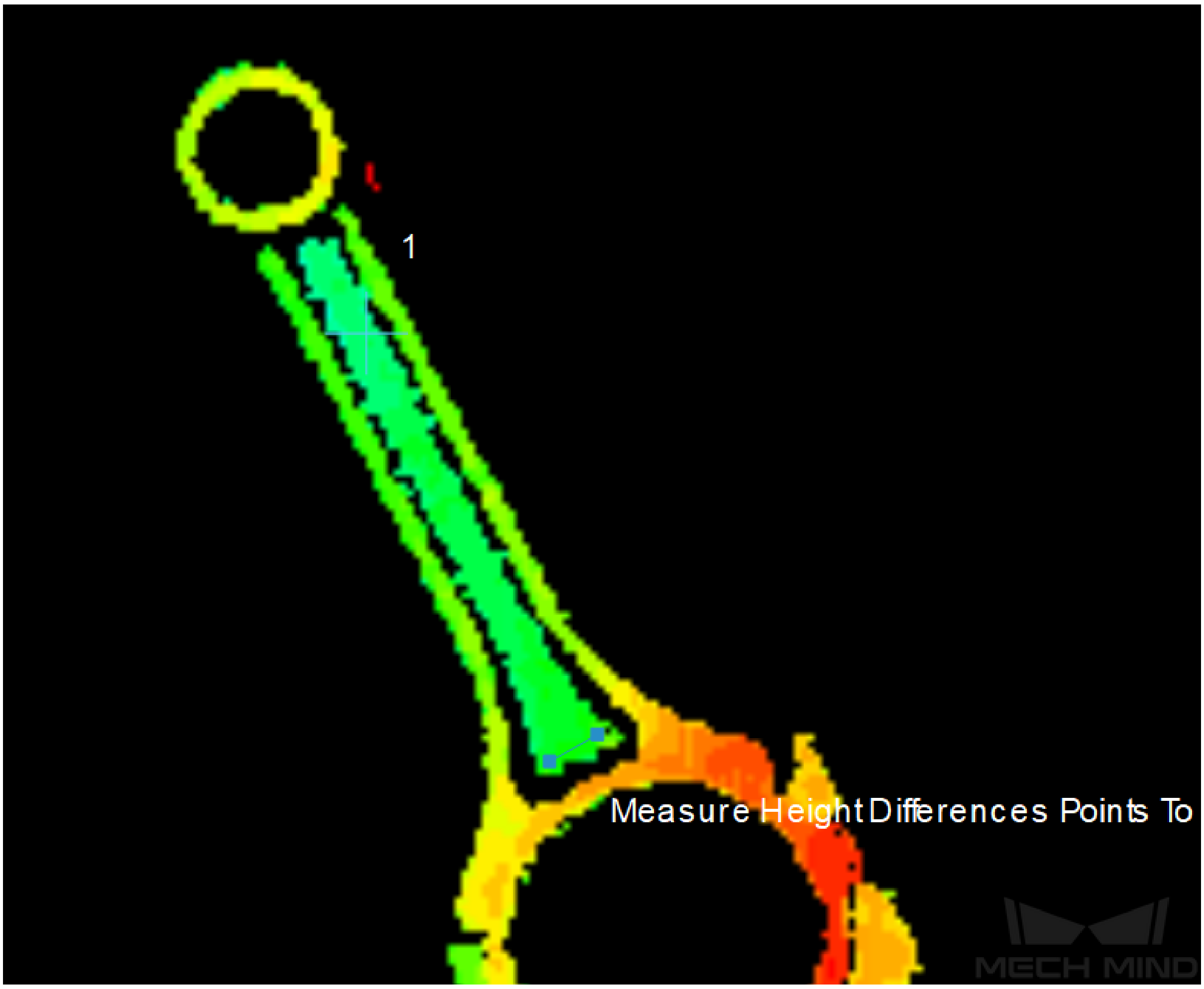
You can also adjust the position of the baseline in the **Step Parameters** panel, as shown below.



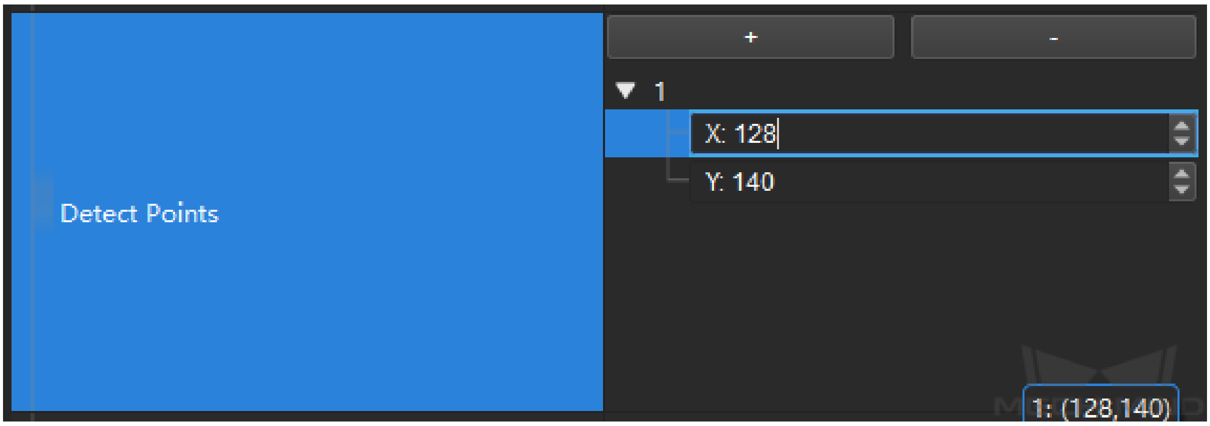
Click on *Empty* and then + to add one or more detect points.



Drag the newly added points from the upper left corner in the sketchpad area onto the point cloud.

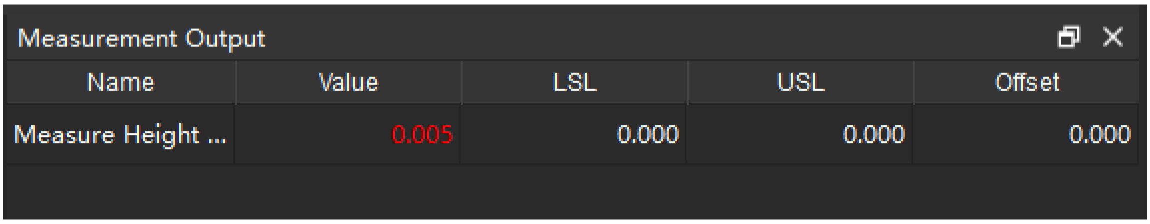


You can also adjust the position of the detect points in the **Step Parameters** panel, as shown below.



Read the result

After completeteing the configuration, click on *Run* to run the project. Click on *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

Note: If multiple detect points have been added, there will be multiple measurement results.

Hint: 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.

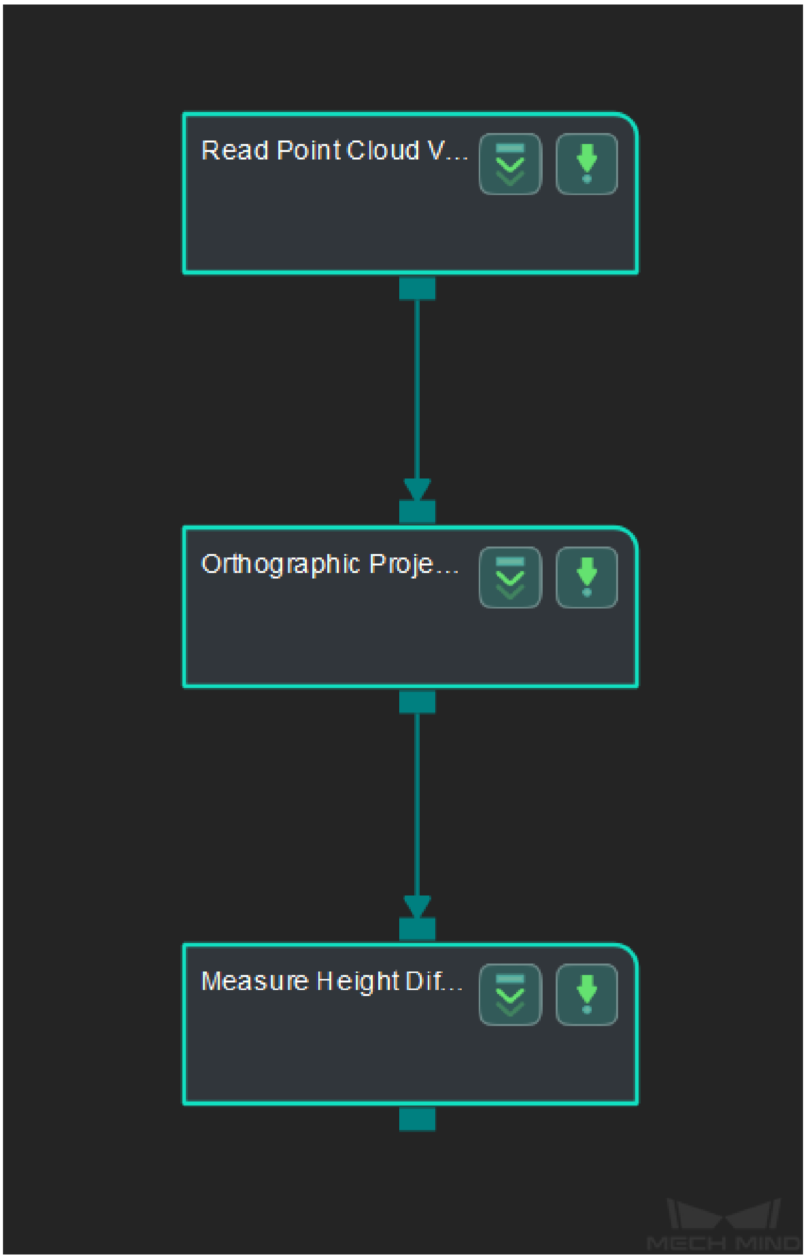
Measure Height Differences Points to Plane

Function

This Step is used to calculate the height difference from points to the plane in a point cloud.

Hint: Before using this Step, please refer to *Getting Started with Measurememt Mode* to learn about basics of **Measurement Mode**.

Sample Application



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 on *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 on *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

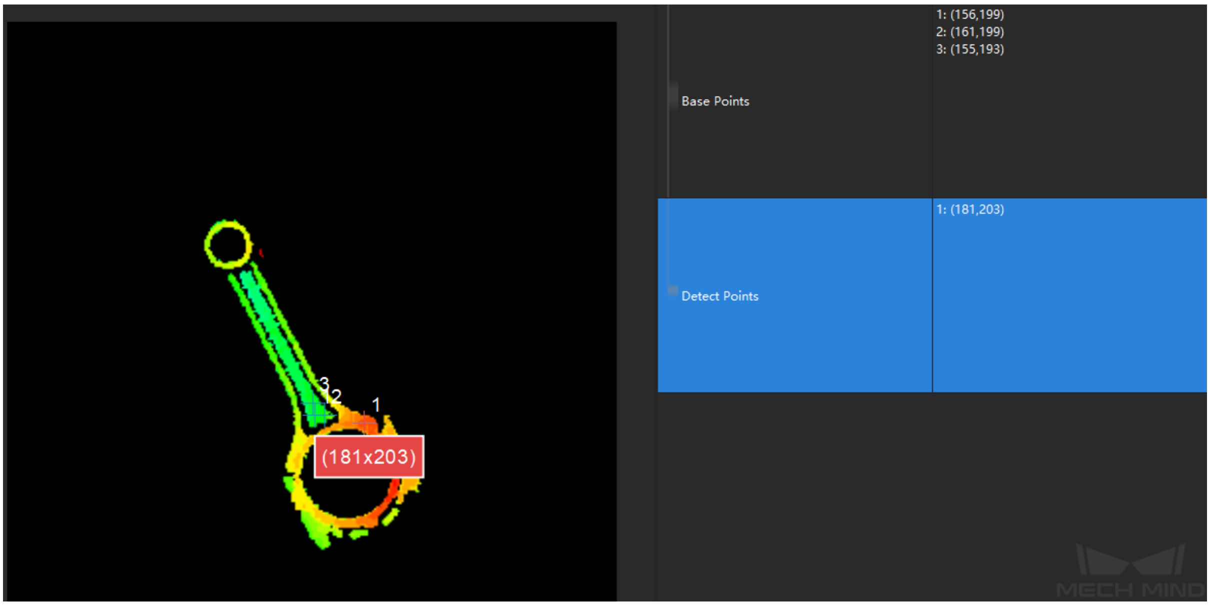
Note: Please prepare the point cloud of the object to be measured in advance.

- Configure in *Read Point Cloud V2*

  Click on the Step, and configure the **Point Cloud Source** and other parameters in the **Step Parameters** panel.

- Configure in *Measure Height Differences Points to Plane*
  - Click on the Step, go to **Step Parameters** panel.
  - Keep the default settings of **Max Cluster Height Difference**, **Max Search Radius**, and **Max Search Points Number**.
  - Add three **Base Points** to define a plane in the point cloud.
  - Add one **Detect Point**.

    Click on *Empty* and then + to add points, and drag the points from the upper left corner in the sketchpad area onto the point cloud, as shown below.



Read the result

After completeing the configuration, click on *Run* to run the project. Click on *Measure Height Differences Points to Plane* and check the measurement result in the **Measurement Output** panel.

Hint: 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.

4.22.2 2D Measuring

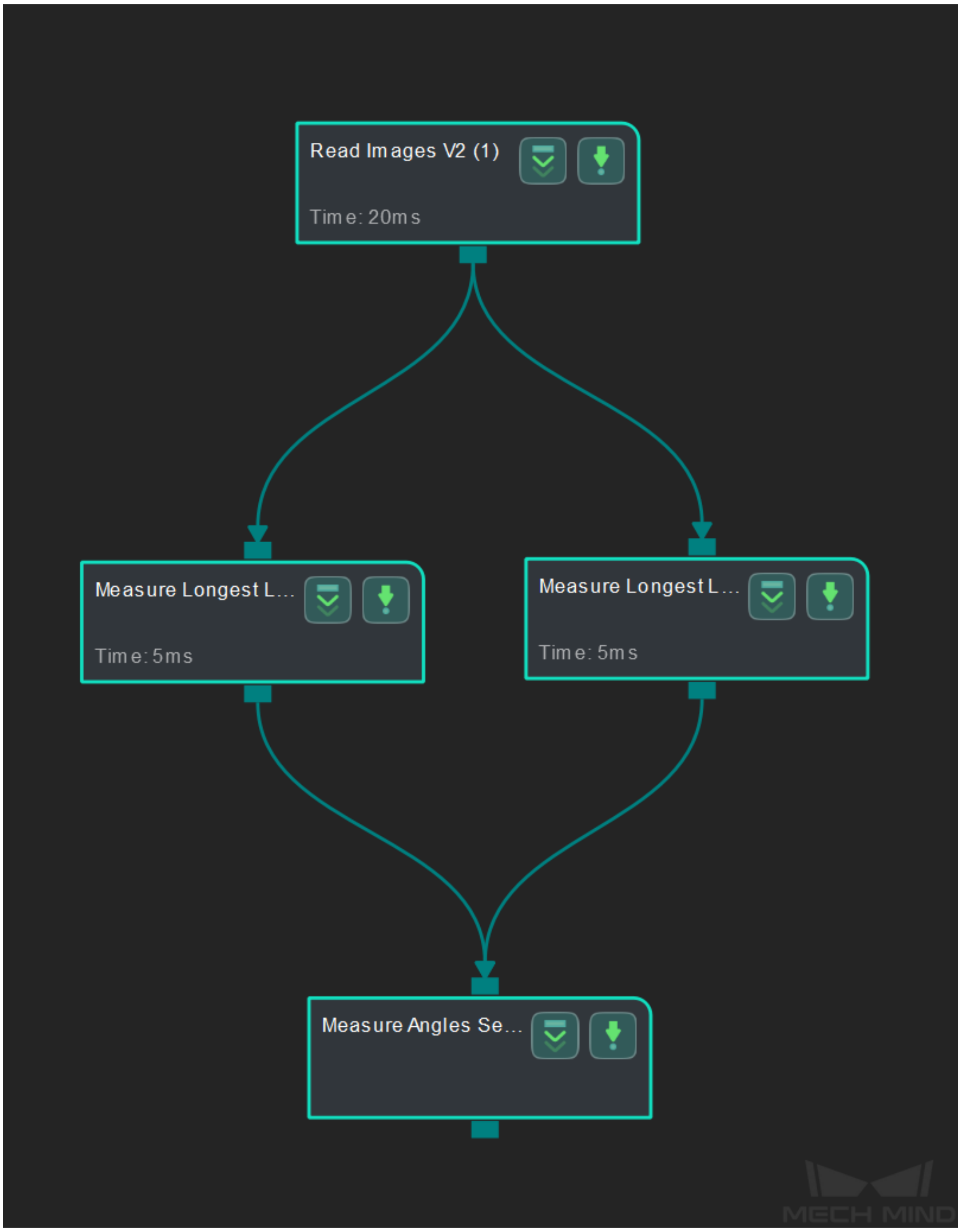
Measure Angles Segments to Segments

Function

This Step is used to measure the angles between line segments.

Hint: Before using this Step, please refer to *Getting Started with Measurememt 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 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 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

Note: Please prepare color image(s) that contain object(s) to be measured in advance.

- Configure in *Read Images V2*

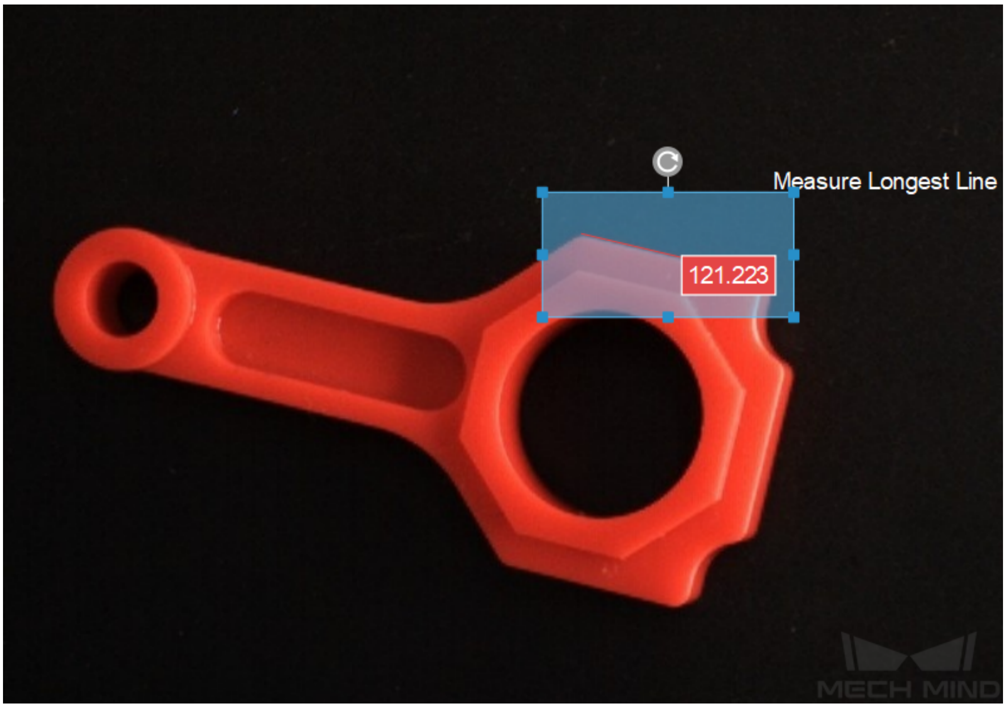
  Click on the Step, and configure the **Imgae Source**, **Image File/Folder Path**, and other parameters in the **Step Parameters** panel. After completeing the configuration, click on *Run* to run the project.

- Configure in *Measure Longest Line Segment* (1)

  Click on the Step, and select an ROI in the sketchpad area.

- Configure in *Measure Longest Line Segment* (2)

Click on the Step, and select an ROI in the sketchpad area.



- Configure in *Measure Angles Segments to Segments*

  Click on the Step, the angle (in degrees) 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.

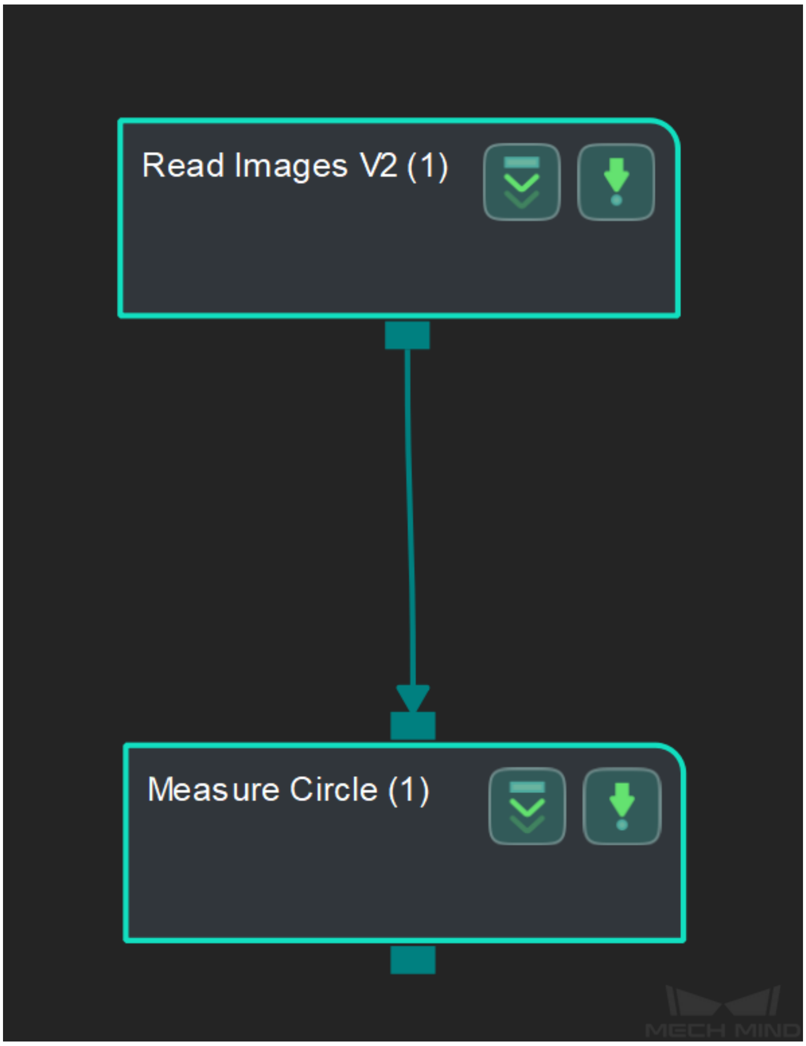
Measure Circle

Function

This Step is mainly used to measure the radius of a circle.

Hint: Before using this Step, please refer to *Getting Started with Measurememt Mode* to learn about basics of **Measurement Mode**.

Sample Application



Connect *Read Images V2* with *Measure Circle* to locate the circle in the ROI and measure its radius.

Connect Steps

Click on *Measure Circle*, 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

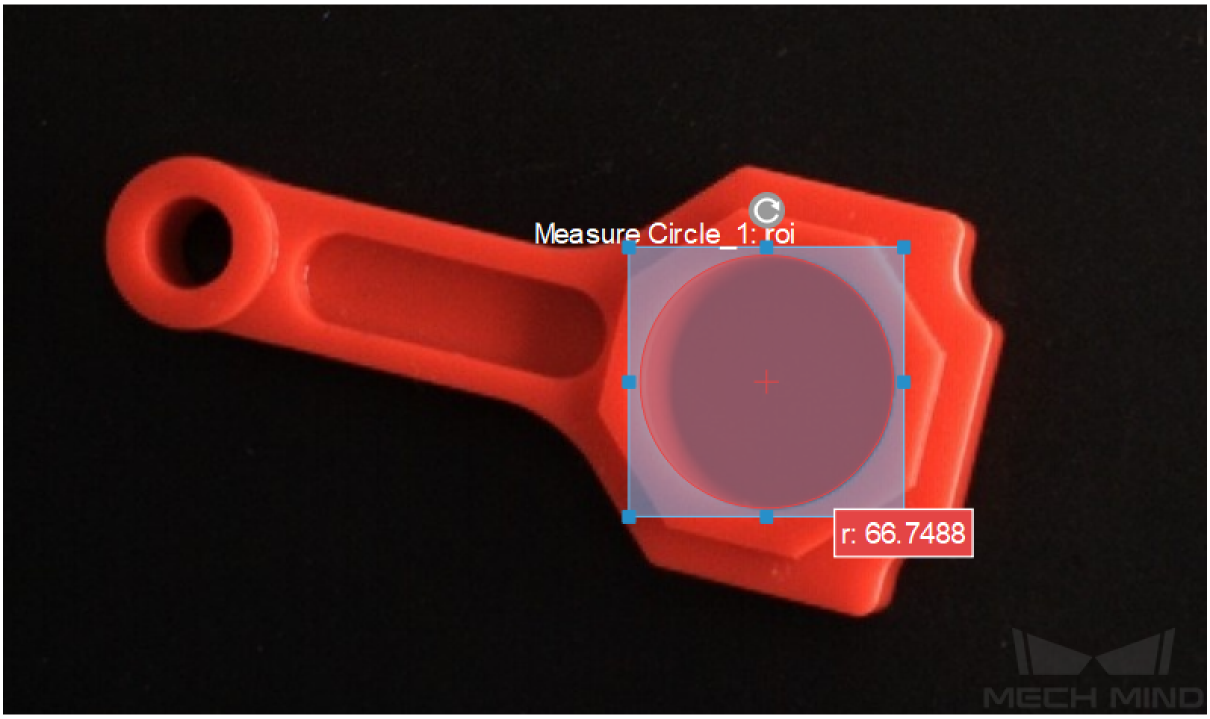
Note: Please prepare color image(s) that contain object(s) to be measured in advance.

- Configure in *Read Images V2*

  Click on the Step, and configure the **Imgae Source**, **Image File/Folder Path**, and other parameters in the **Step Parameters** panel. After completeing the configuration, click on *Run* to run the project.

- Configure in *Measure Circle*

  Click on 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.

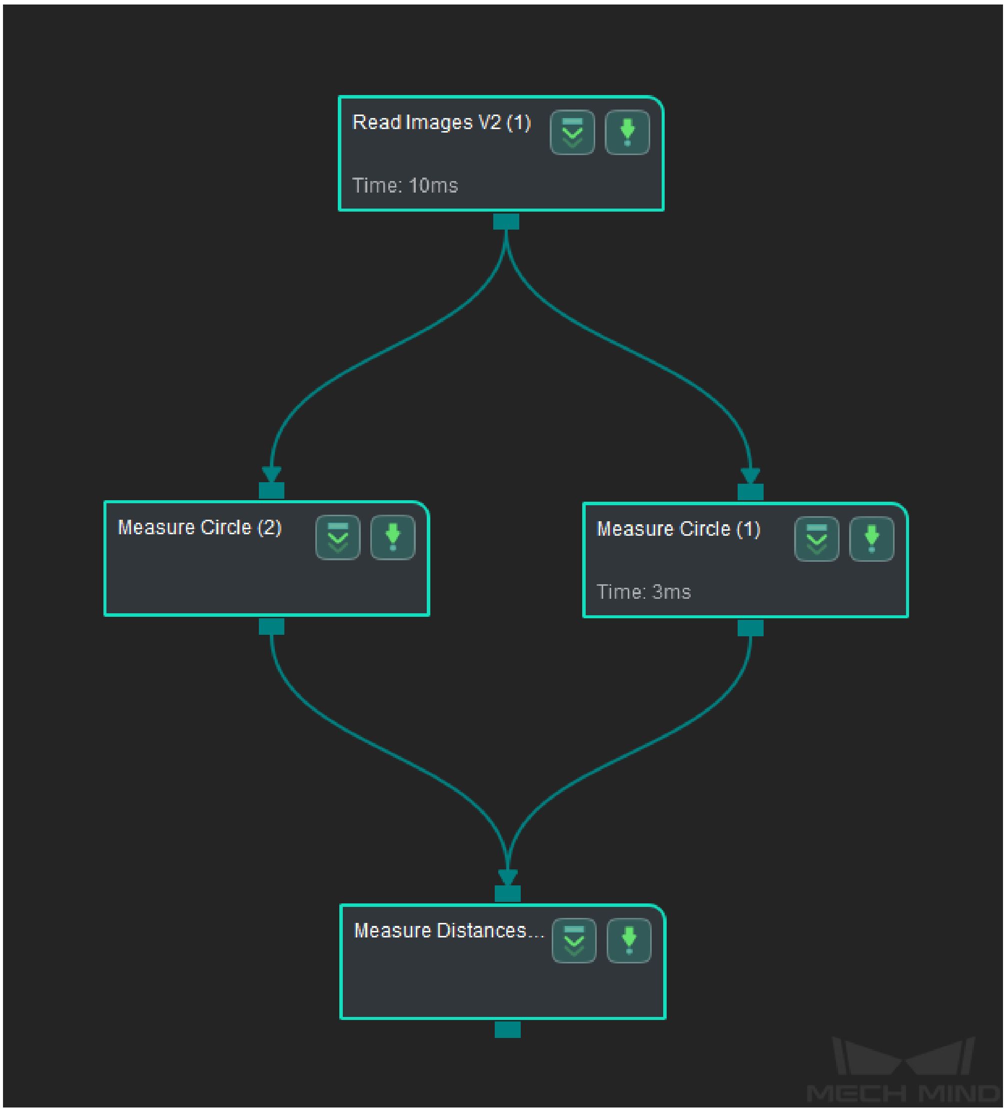
Measure Distances Circles to Circles

Function

This Step is used to measure the distances from circles to circles.

Hint: Before using this Step, please refer to *Getting Started with Measurememt Mode* to learn about basics of **Measurement Mode**.

Sample Application



Connect *Read Images V2*, *Measure Circle*, 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 on *Measure Circle* (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 Circle* (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 Circles to Circles*, go to the **Step Input Source Selection** panel, and then select **Measure Circle_1_Circles** as **Input 2 (Circles)** and **Measure Circle_2_Circles** as **Input 3 (Circles)**.

Configure parameters and read the result

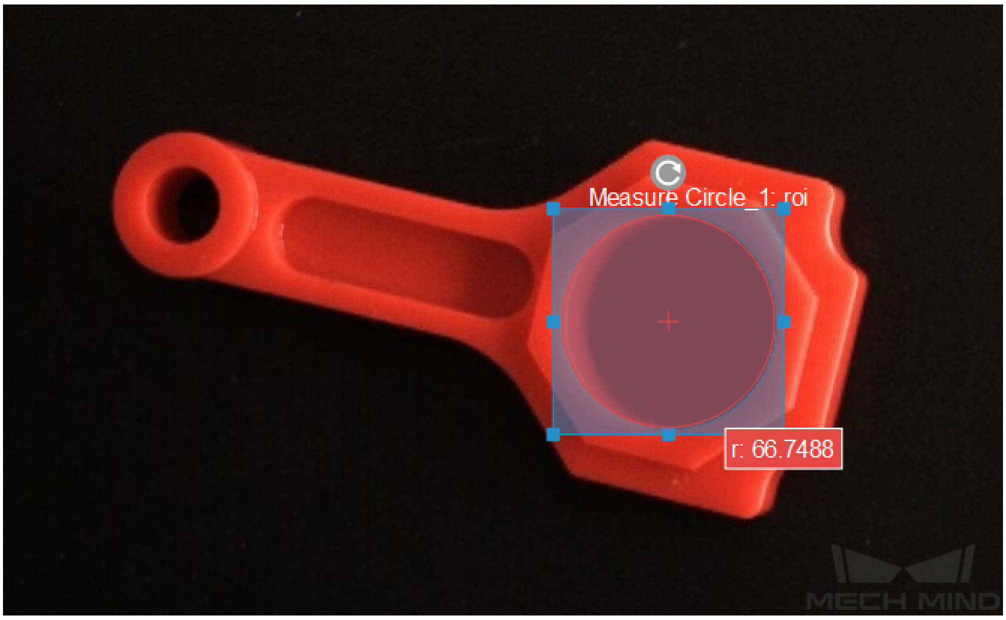
Note: Please prepare color image(s) that contain object(s) to be measured in advance.

- Configure in *Read Images V2*

  Click on the Step, and configure the **Imgae Source**, **Image File/Folder Path**, and other parameters in the **Step Parameters** panel. After completeing the configuration, click on *Run* to run the project.

- Configure in *Measure Circle* (1)

  Click on the Step, and select an ROI in the sketchpad area.

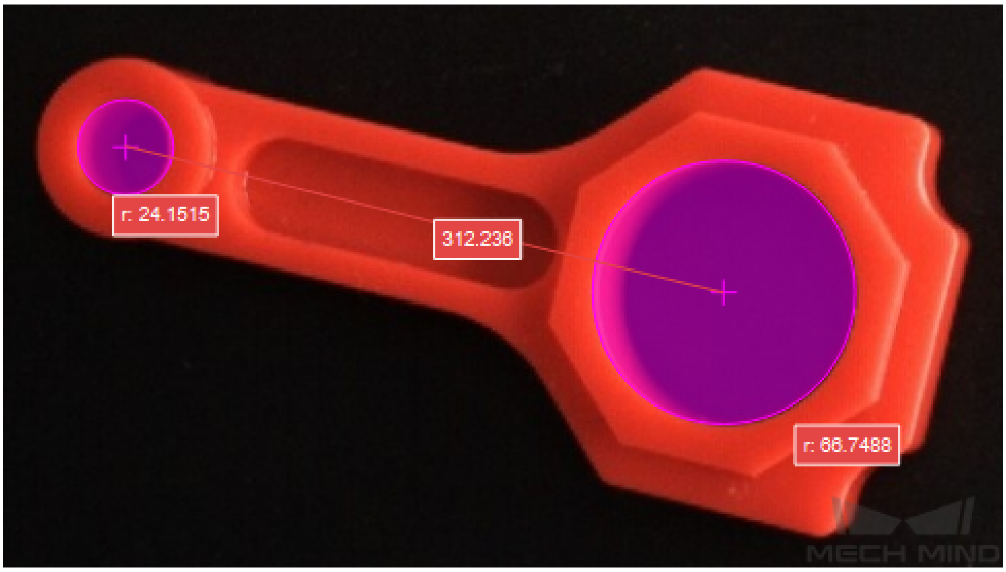


- Configure in *Measure Circle* (2)

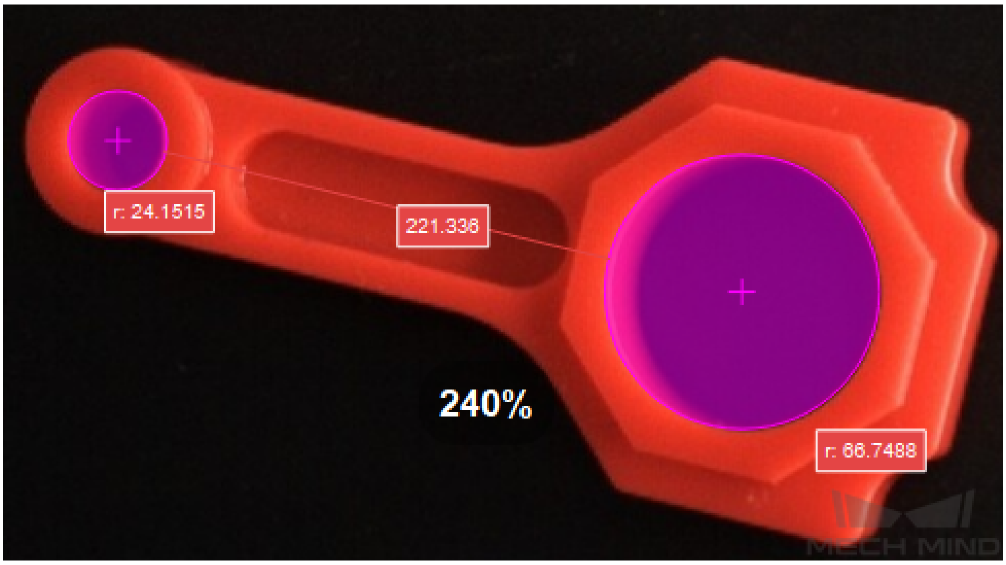
  Click on the Step, and select an ROI in the sketchpad area.

- Configure in *Measure Distances Circles to Circles*

  Click on the Step, 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 between two circles.



- 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.

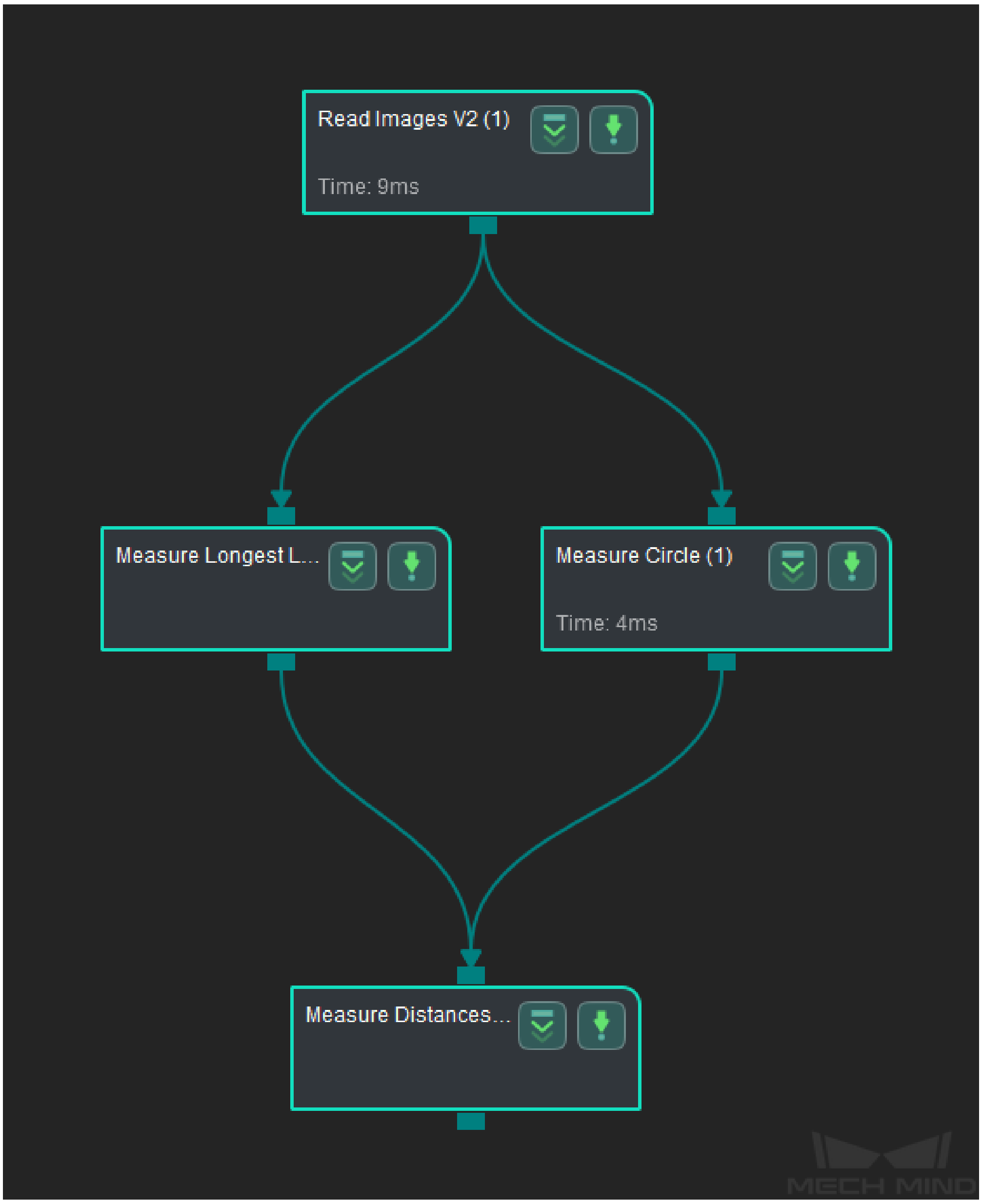
Measure Distances Circles to Segments

Function

This Step is used to measure the distances from circles to line segments.

Hint: Before using this Step, please refer to *Getting Started with Measurememt Mode* to learn about basics of **Measurement Mode**.

Sample Application



Connect *Read Images V2*, *Measure Circle*, *Measure Longest Line Segment*, and *Measure Distances Circles to Segments* to measure the distance from circles to line segments.

Connect Steps

1. Click on *Measure Circle* , 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* , 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 Circles to Segments*, go to the **Step Input Source Selection** panel, and then select **Measure Circle_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

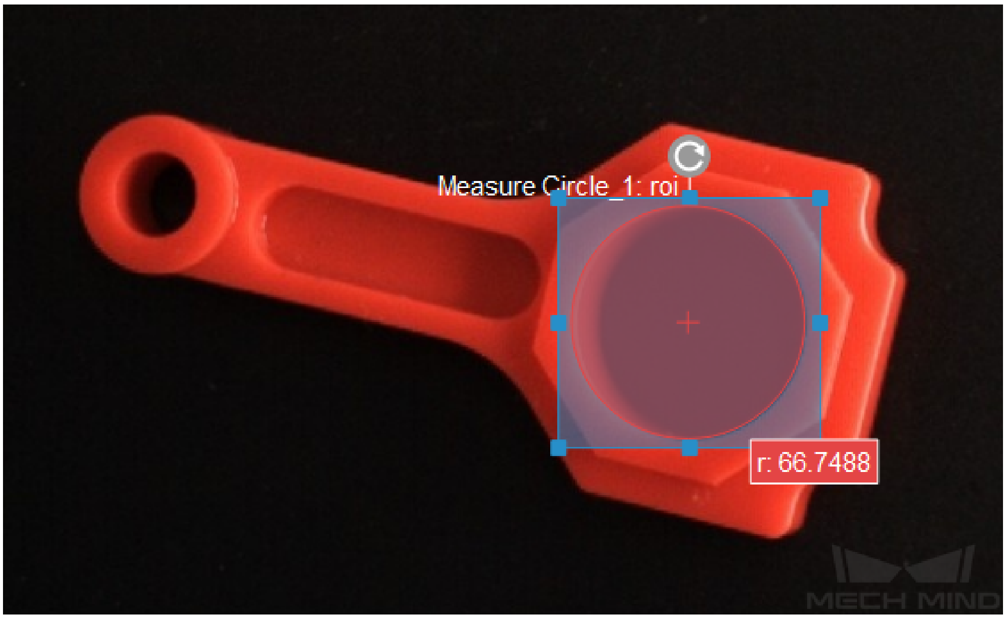
Note: Please prepare color image(s) that contain object(s) to be measured in advance.

- Configure in *Read Images V2*

  Click on the Step, and configure the **Imgae Source**, **Image File/Folder Path**, and other parameters in the **Step Parameters** panel. After completeing the configuration, click on *Run* to run the project.

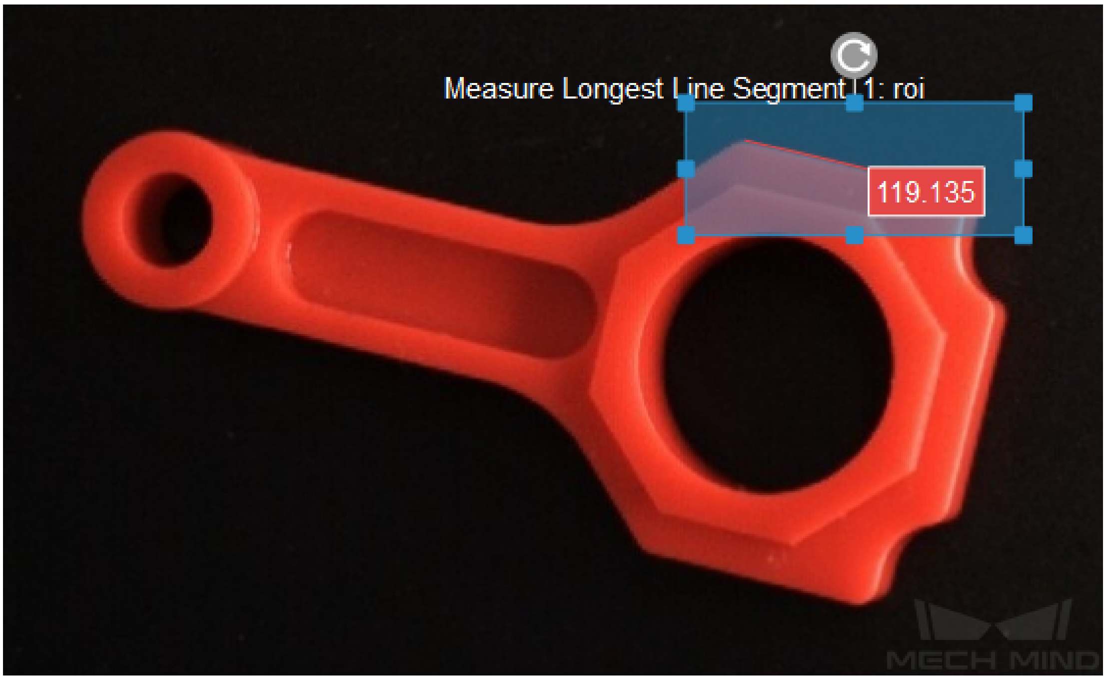
- Configure in *Measure Circle*

  Click on the Step, and select an ROI in the sketchpad area.



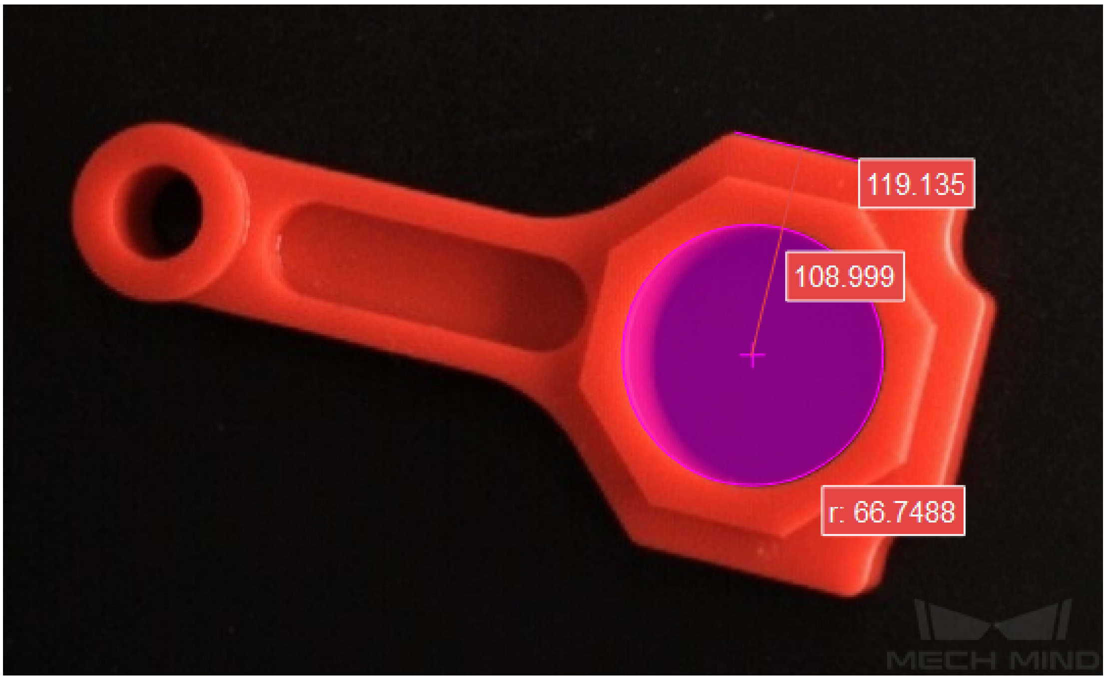
- Configure in *Measure Longest Line Segment*

  Click on the Step, and select an ROI in the sketchpad area.



- Configure in *Measure Distances Circles to Segments*

  Click on the Step, 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.

Measure Distances Points to Circles

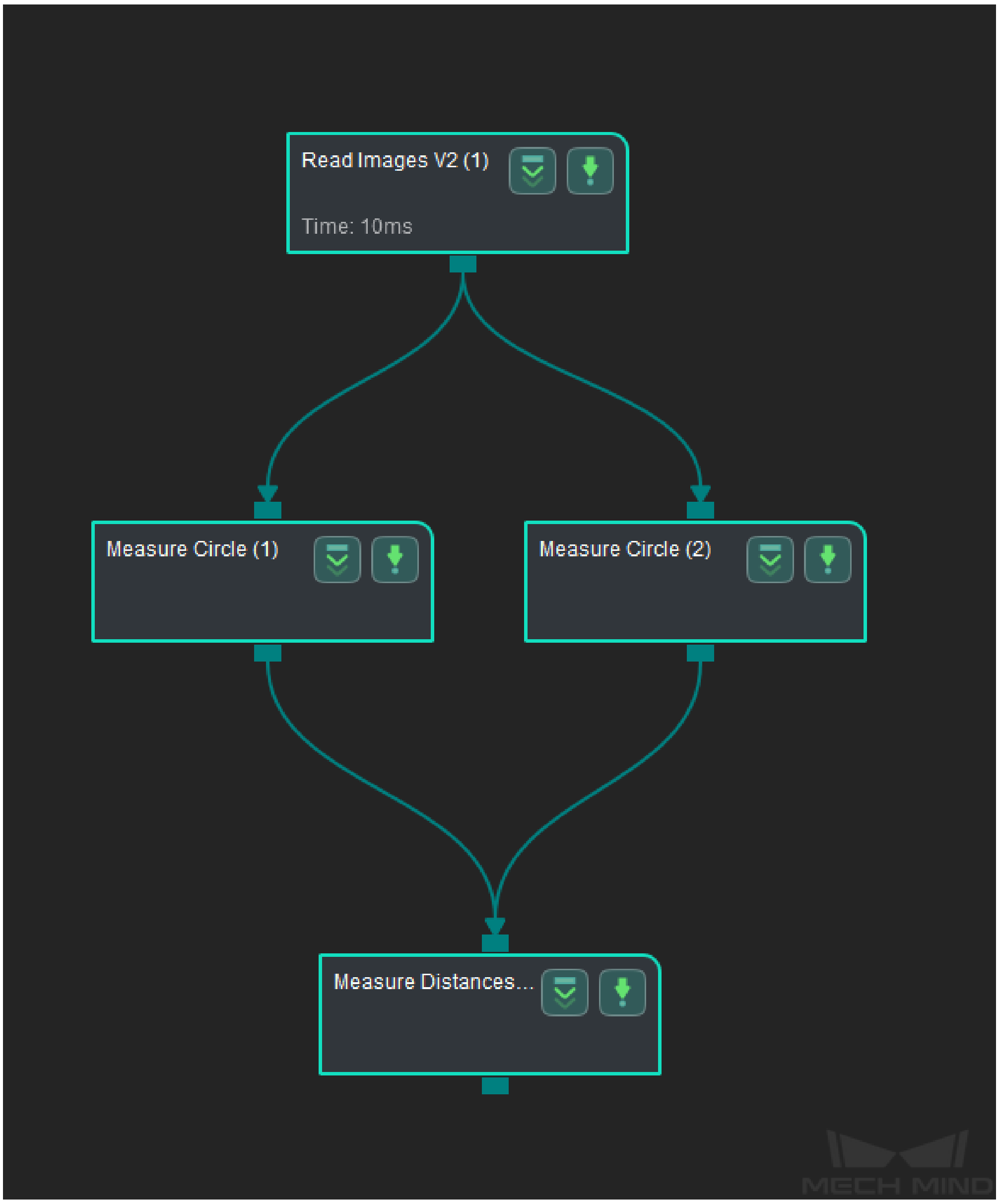
Function

This Step is used to measure the distances from points to circles.

Hint: Before using this Step, please refer to *Getting Started with Measurememt Mode* to learn about basics of **Measurement Mode**.

Attention: This Step does not support custom specified points, the points can only be obtained from *Measure Circle* or *Detect Circle Centers*.

Sample Application



Connect *Read Images V2*, *Measure Circle*, and *Measure Distances Points to Circles* to measure the distance from points to circles.

Connect Steps

1. Click on *Measure Circle* (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 Circle* (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 Circle_1_Circles** as **Input 2 (Points)** and **Measure Circle_2_Circles** as **Input 3 (Circles)**.

Configure parameters and read the result

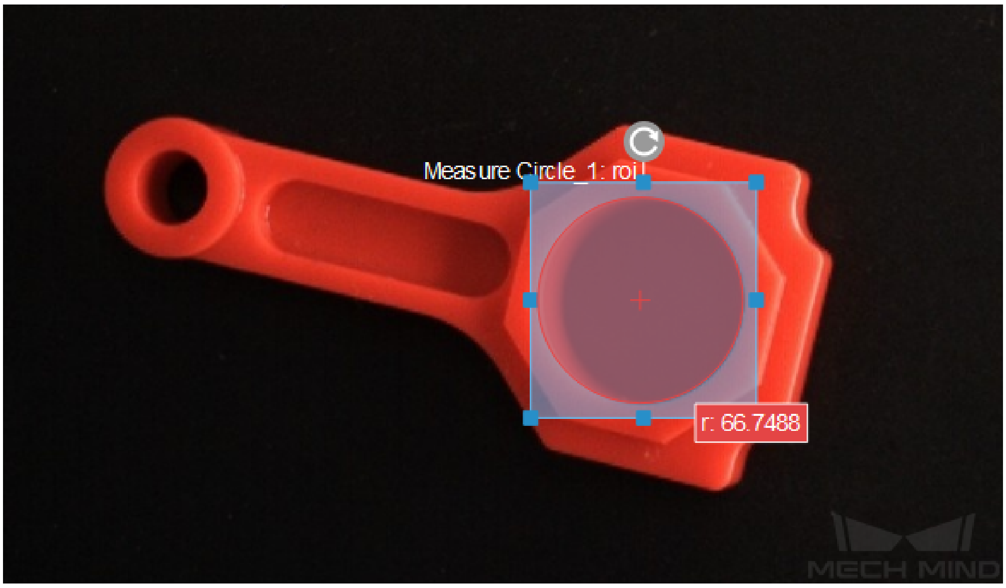
Note: Please prepare color image(s) that contain object(s) to be measured in advance.

- Configure in *Read Images V2*

  Click on the Step, and configure the **Imgae Source**, **Image File/Folder Path**, and other parameters in the **Step Parameters** panel. After completeing the configuration, click on *Run* to run the project.

- Configure in *Measure Circle* (1)

  Click on the Step, and select an ROI in the sketchpad area.

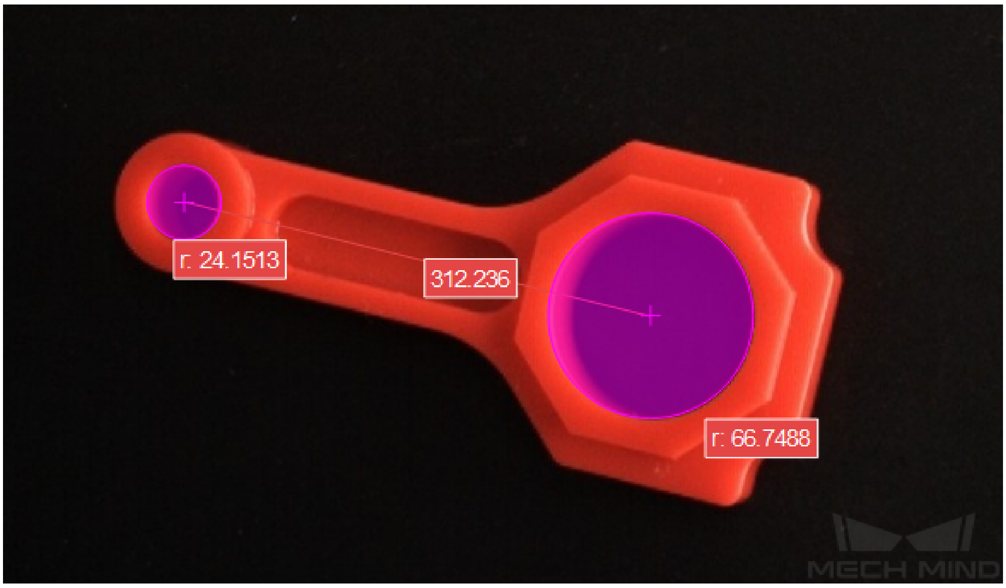


- Configure in *Measure Circle* (2)

  Click on the Step, and select an ROI in the sketchpad area.

- Configure in *Measure Distances Points to Circles*

  Click on the Step, 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.

Measure Distances Points to Points

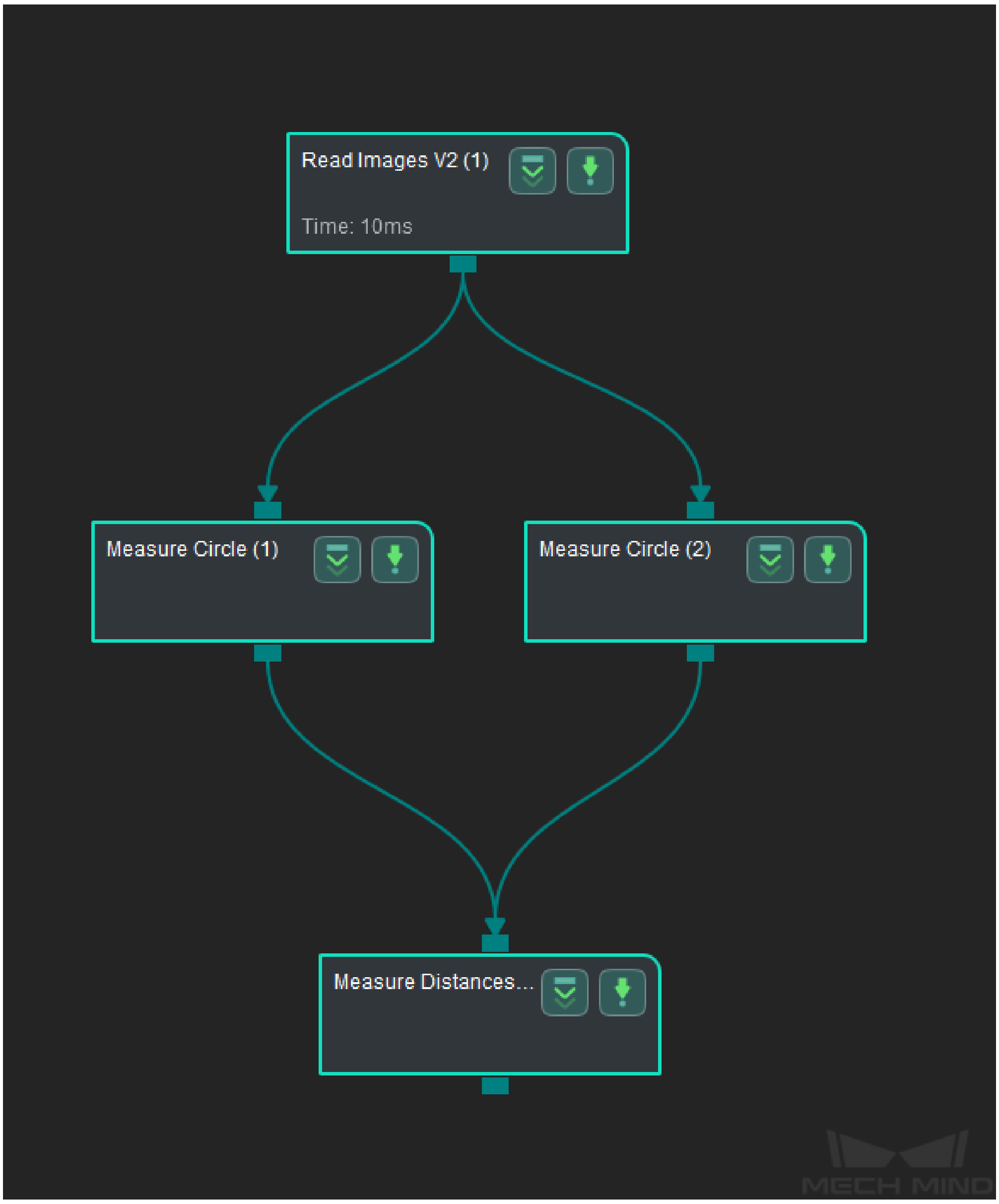
Function

This Step is used to measure the distances between two points.

Hint: Before using this Step, please refer to *Getting Started with Measurememt Mode* to learn about basics of **Measurement Mode**.

Attention: This Step does not support custom specified points, the points can only be obtained from *Measure Circle* or *Detect Circle Centers*.

Sample Application



Connect *Read Images V2*, *Measure Circle*, and *Measure Distances Points to Points* to measure the distance between two points.

Connect Steps

1. Click on *Measure Circle* (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 Circle* (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 Points*, go to the **Step Input Source Selection** panel, and then select **Measure Circle_1_Circles** as **Input 2 (Points)** and **Measure Circle_2_Circles** as **Input 3 (Points)**.

Configure parameters and read the result

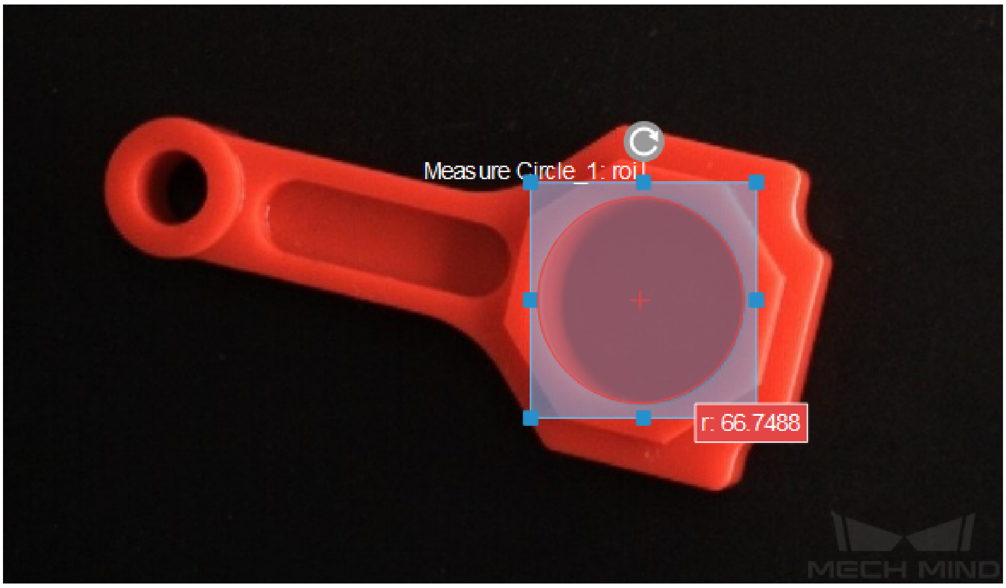
Note: Please prepare color image(s) that contain object(s) to be measured in advance.

- Configure in *Read Images V2*

  Click on the Step, and configure the **Imgae Source**, **Image File/Folder Path**, and other parameters in the **Step Parameters** panel. After completeing the configuration, click on *Run* to run the project.

- Configure in *Measure Circle* (1)

  Click on the Step, and select an ROI in the sketchpad area.

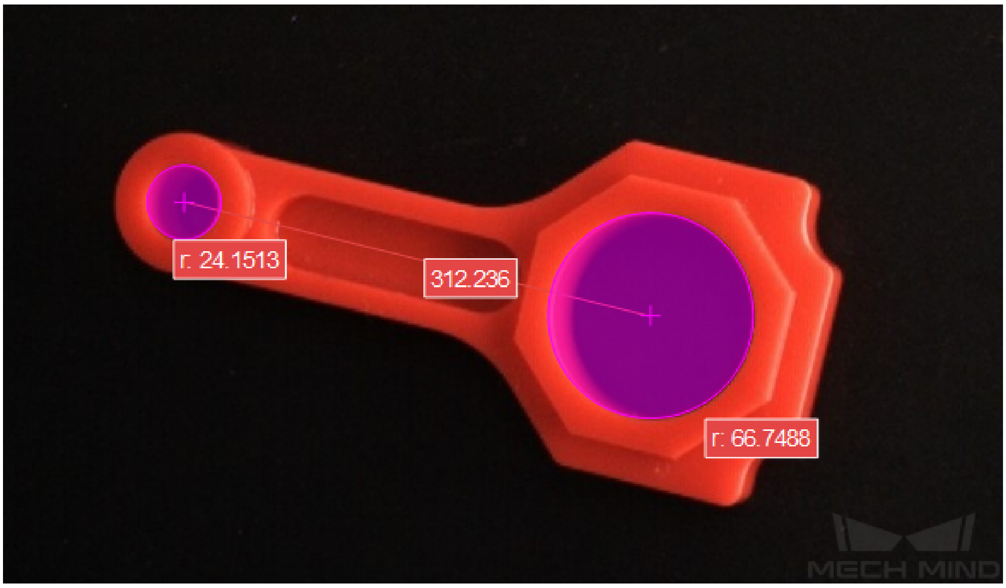


- Configure in *Measure Circle* (2)

  Click on the Step, and select an ROI in the sketchpad area.

- Configure in *Measure Distances Points to Points*

  Click on the Step, 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.

Measure Distances Points to Segments

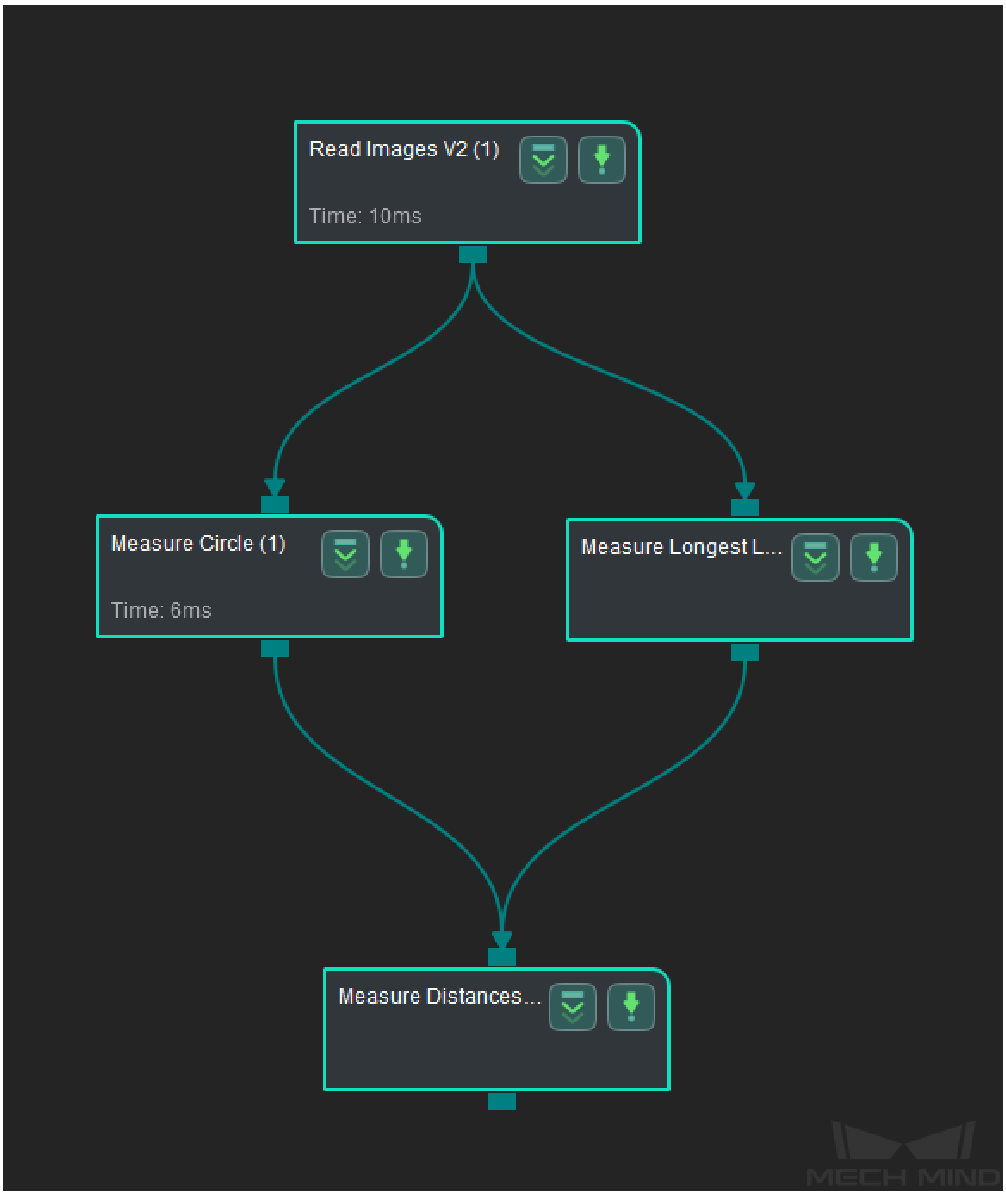
Function

This Step is used to measure the distances from points to line segments.

Hint: Before using this Step, please refer to *Getting Started with Measurememt Mode* to learn about basics of **Measurement Mode**.

Attention: This Step does not support custom specified points, the points can only be obtained from *Measure Circle* or *Detect Circle Centers*.

Sample Application



Connect *Read Images V2*, *Measure Circle*, *Measure Longest Line Segment*, and *Measure Distances Points to Segments* to measure the distance from points to line segments.

Connect Steps

1. Click on *Measure Circle* , 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* , 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 Segments*, go to the **Step Input Source Selection** panel, and then select **Measure Circle_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

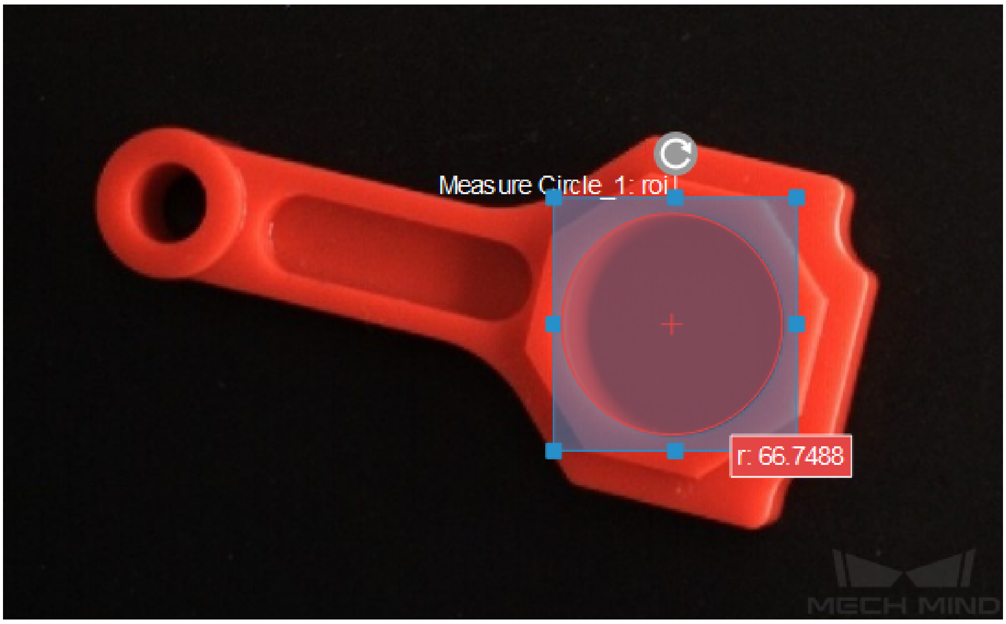
Note: Please prepare color image(s) that contain object(s) to be measured in advance.

- Configure in *Read Images V2*

  Click on the Step, and configure the **Imgae Source**, **Image File/Folder Path**, and other parameters in the **Step Parameters** panel. After completeing the configuration, click on *Run* to run the project.

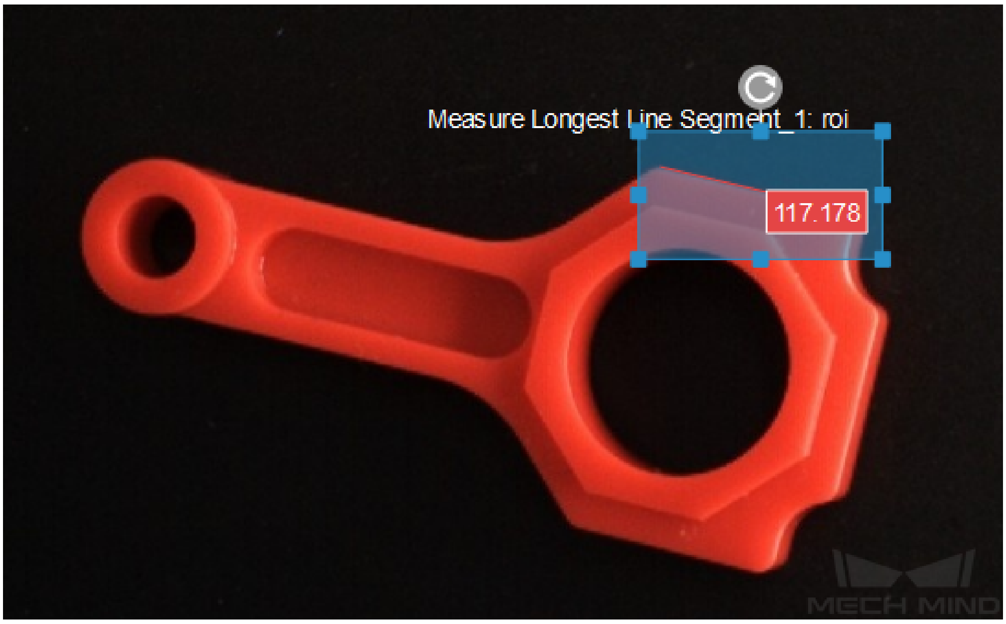
- Configure in *Measure Circle*

  Click on the Step, and select an ROI in the sketchpad area.



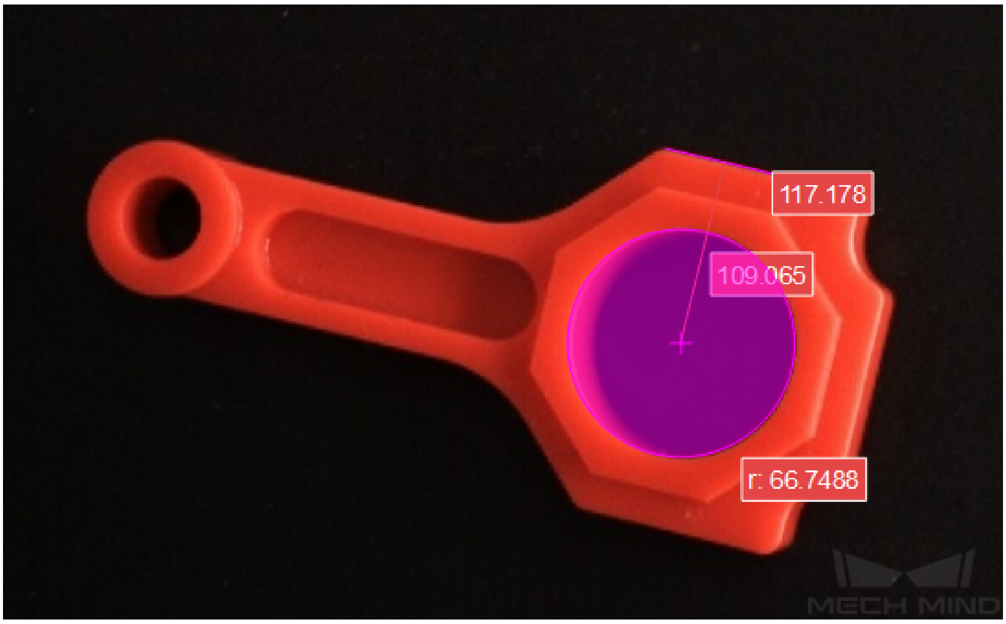
- Configure in *Measure Longest Line Segment*

  Click on the Step, and select an ROI in the sketchpad area.



- Configure in *Measure Distances Points to Segments*

  Click on the Step, the distance (in pixels) from the center of the circle to the line segment 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.

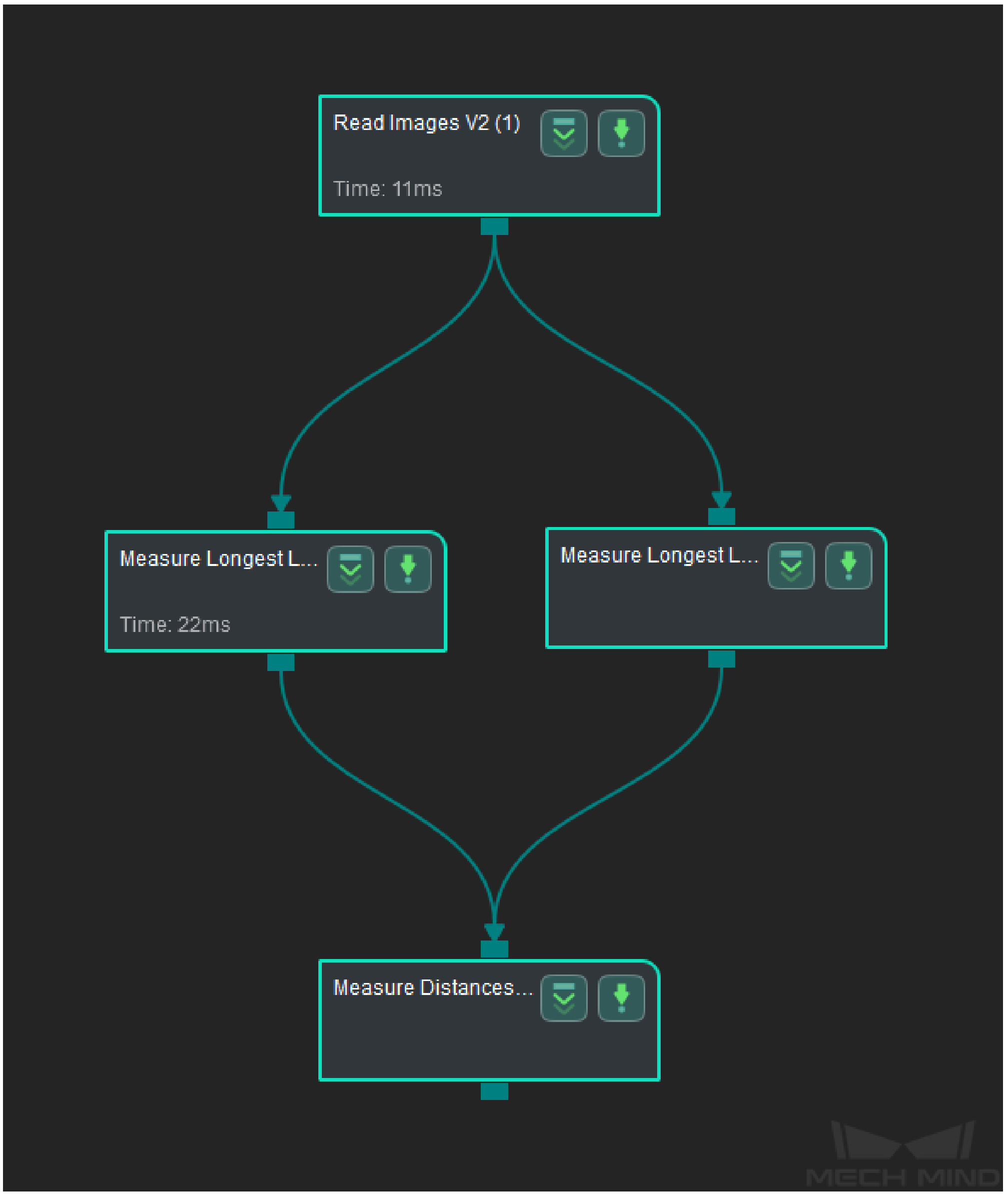
Measure Distances Segments to Segments

Function

This Step is used to measure the distances between line segments.

Hint: Before using this Step, please refer to *Getting Started with Measurememt 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

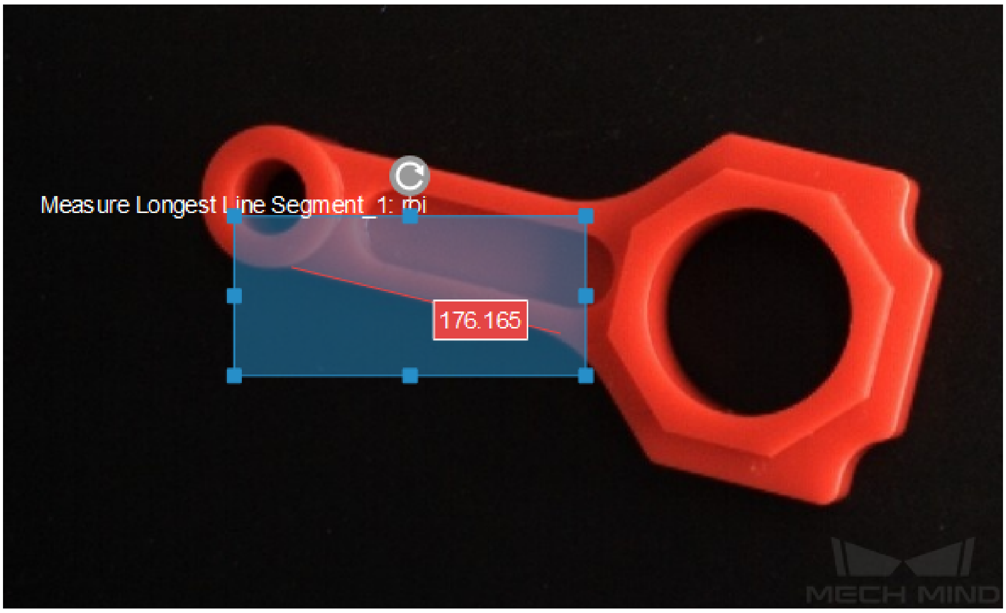
Note: Please prepare color image(s) that contain object(s) to be measured in advance.

- Configure in *Read Images V2*

  Click on the Step, and configure the **Imgae Source**, **Image File/Folder Path**, and other parameters in the **Step Parameters** panel. After completeing the configuration, click on *Run* to run the project.

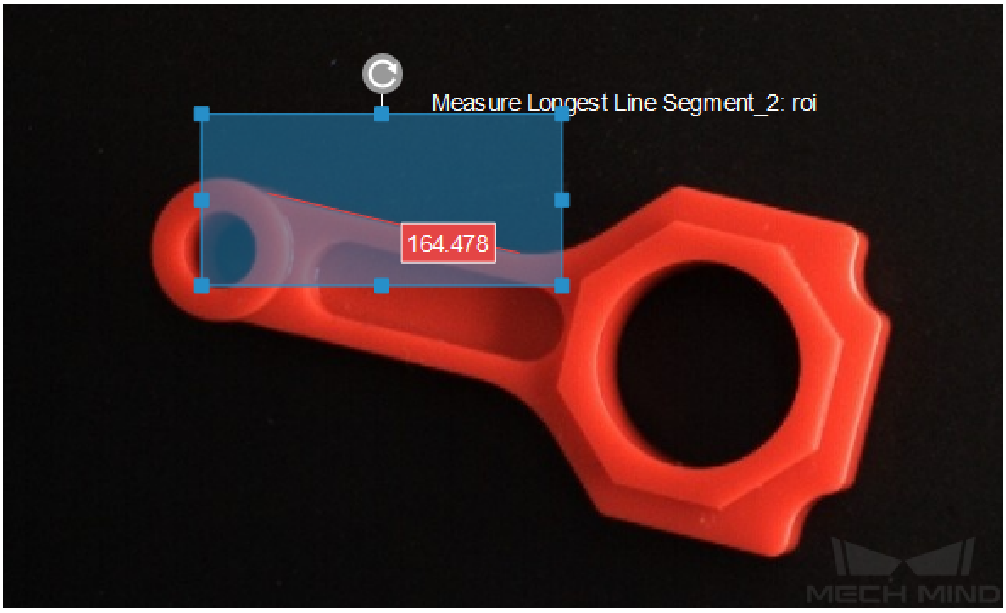
- Configure in *Measure Longest Line Segment* (1)

  Click on the Step, and select an ROI in the sketchpad area.



- Configure in *Measure Longest Line Segment* (2)

Click on the Step, and select an ROI in the sketchpad area.



- Configure in *Measure Distances Segments to Segments*

  Click on the Step, 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.

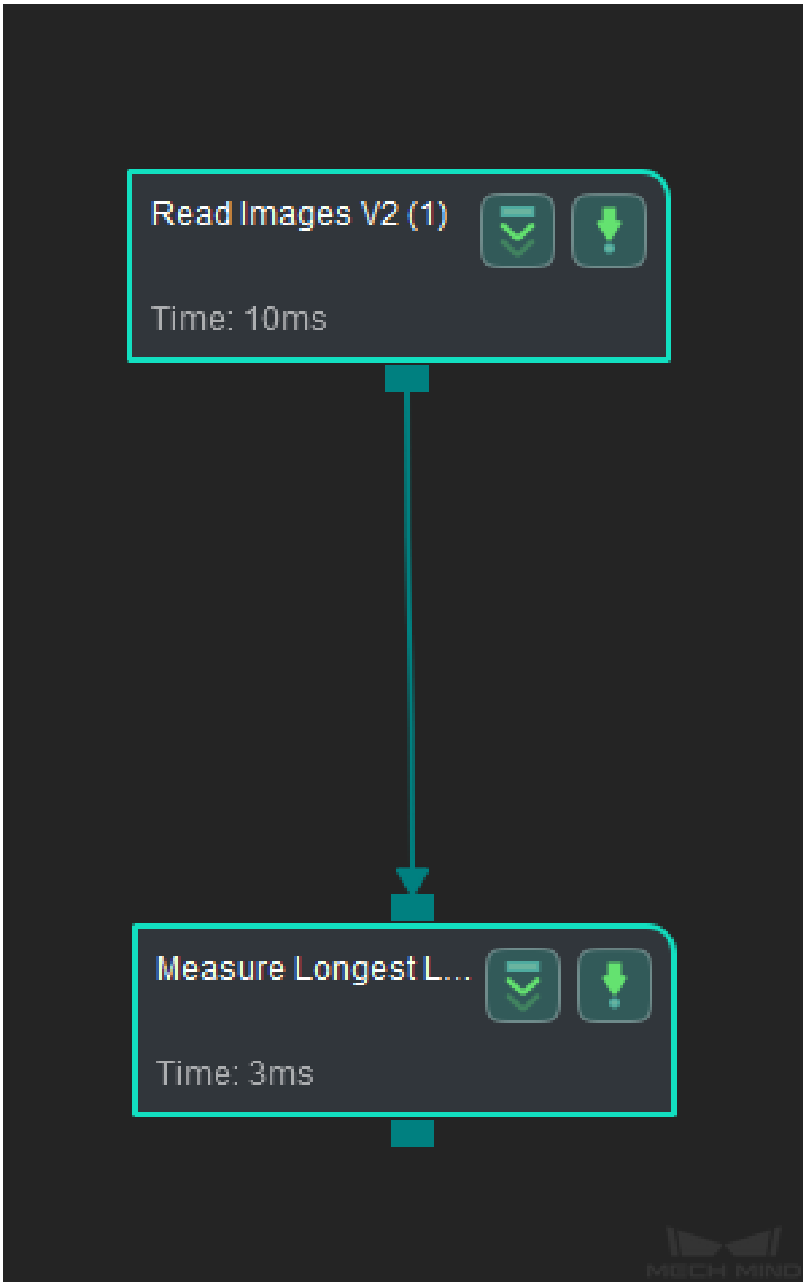
Measure Longest Line Segment

Function

This Step is used to locate and measure the longest line segment in the ROI.

Hint: Before using this Step, please refer to *Getting Started with Measurememt 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 on *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

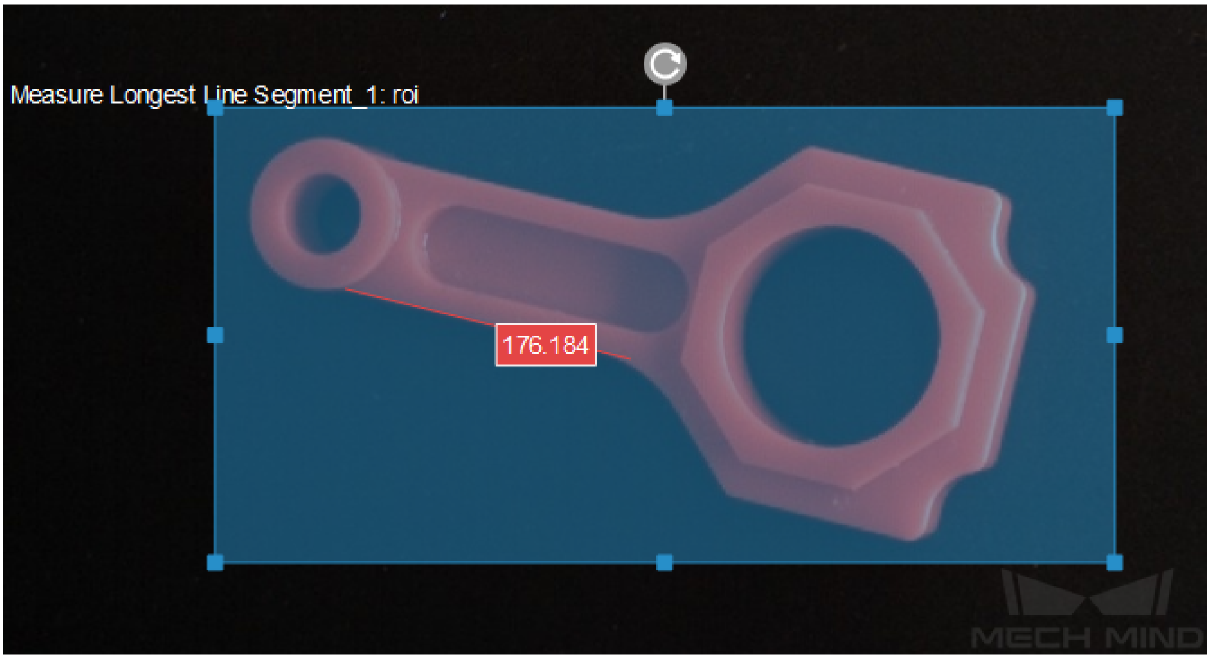
Note: Please prepare color image(s) that contain object(s) to be measured in advance.

- Configure in *Read Images V2*

  Click on the Step, and configure the **Imgae Source**, **Image File/Folder Path**, and other parameters in the **Step Parameters** panel. After completeing the configuration, click on *Run* to run the project.

- Configure in *Measure Longest Line Segment*

  Click on 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.

4.22.3 3D Length/Distance

This section introduces Steps under **3D Length / Distance**.

Calc Diagonal Length

Calc Distance between Two Poses along Specified Direction

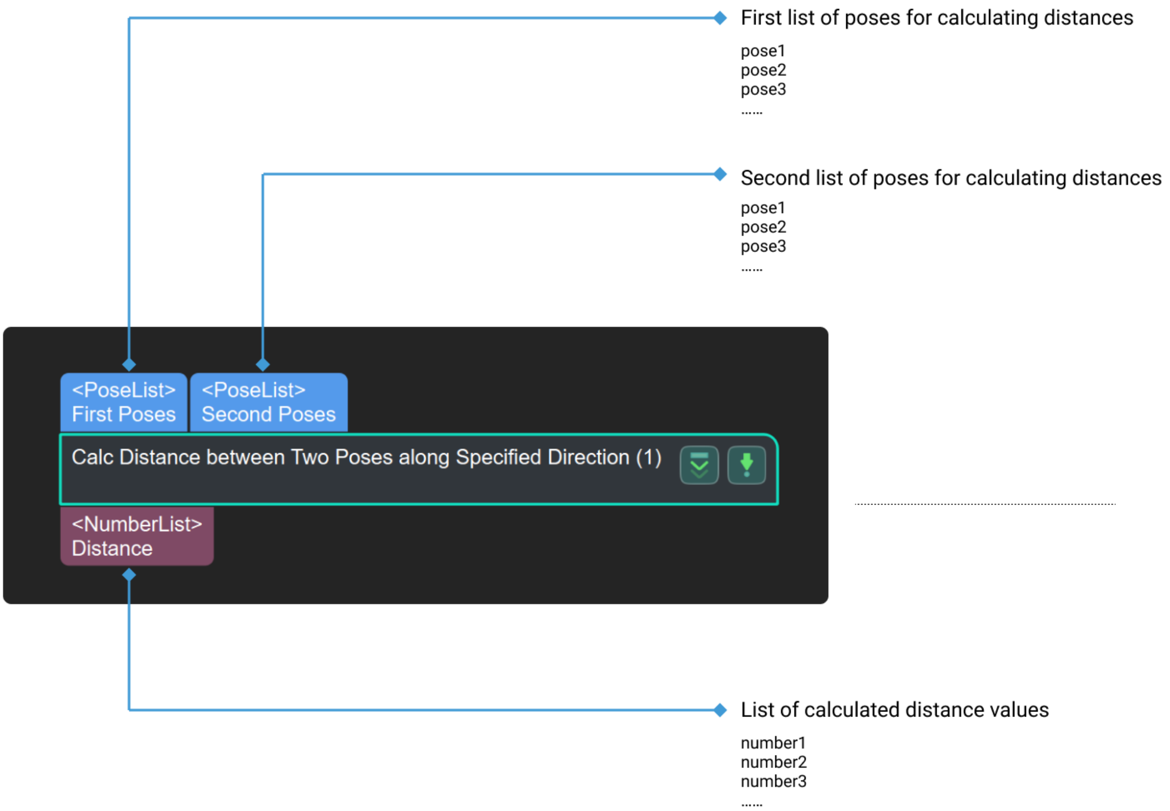
Function

Calculates the projection length of the vector formed by the two pose centers on a specified axis or combination of axes.

Sample Scenario

A general pose distance calculation Step. No fixed usage scenarios.

Input and Output



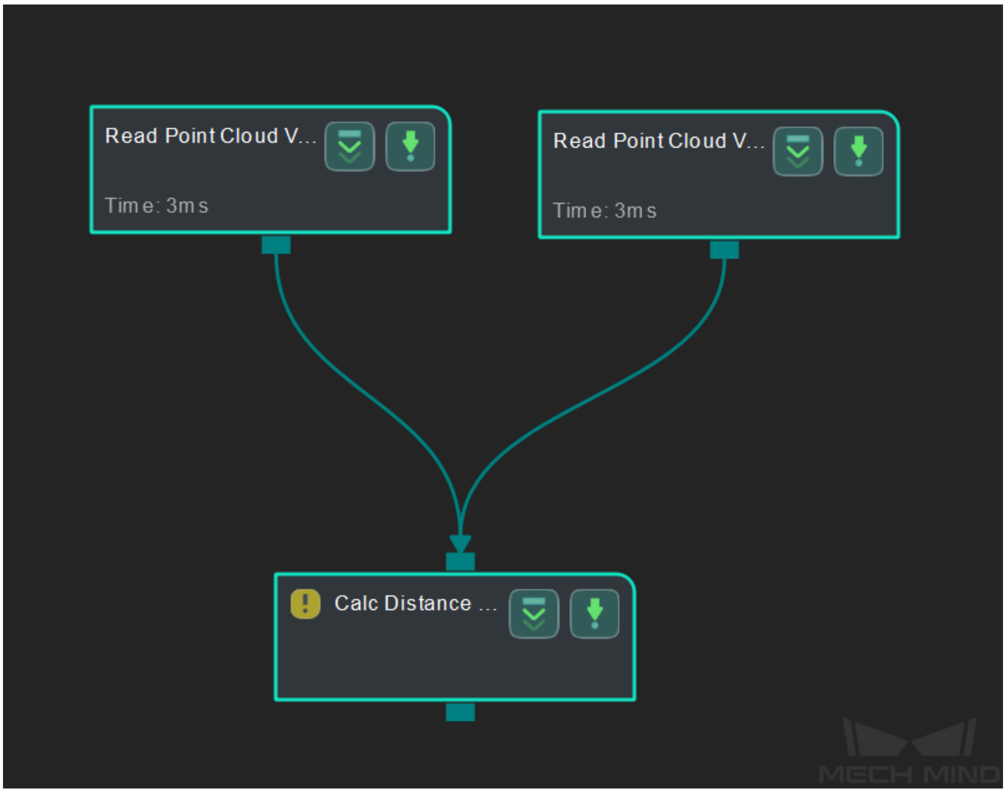
Calc Distance from 3D Points to Plane

Function

This Step is used to calculate the distance from 3D points to the plane.

Hint: Before using this Step, please refer to *Getting Started with Measurememt Mode* to learn about basics of **Measurement Mode**.

Sample Application



Connect *Read Point Cloud V2* with *Calc Distance from 3D Points to Plane* to calculate the distance from 3D points to the plane.

Connect Steps

Click on *Calc Distance from 3D Points to Plane*, go to the **Step Input Source Selection** panel, and then select **Read Point Cloud V2_1_Point Cloud With Normals** as **Input 1 (Point Clouds With Normals)** and **Read Point Cloud V2_2_Point Cloud With Normals** as **Input 4 (Clouds for Fitting)**.

Configure parameters

Note: Please prepare the point cloud of the object to be measured in advance.

- Configure in *Read Point Cloud V2* (1)

  Click on the Step, and configure the **Point Cloud Source** and other parameters in the **Step Parameters** panel.

- Configure in *Read Point Cloud V2* (2)

  Click on the Step, and configure the **Point Cloud Source** and other parameters in the **Step Parameters** panel.

- After configuration, click on *Run* to run the project.

Read the result

Click on *Calc Distance from 3D Points to Plane* and check the measurement in the **Measurement Output** panel.

Hint: 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.

Calc Distance to Reference Pose

Calc Length along Axis

Calc Mask' s Span on Given Line

4.22.4 Global Dimensioning and Tolerancing

Calc Flatness Error

Function

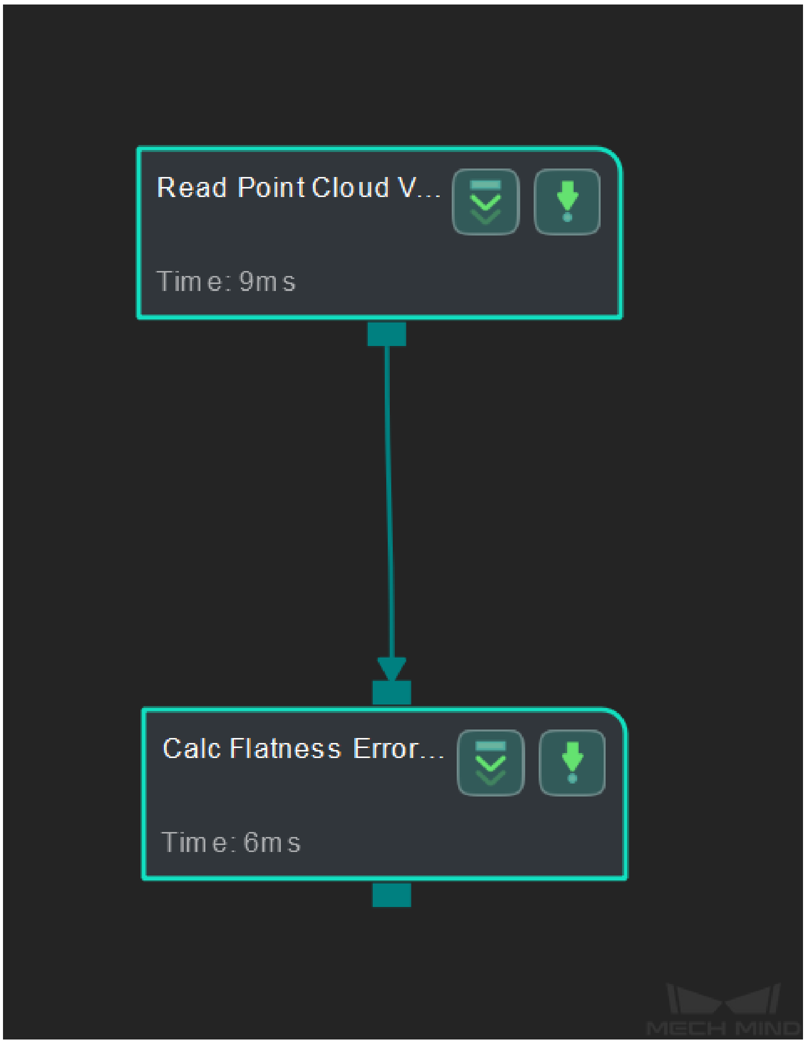
This Step is used to calculate the flatness error of the planar point cloud.

Note:

- 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.

Hint: Before using this Step, please refer to *Getting Started with Measurememt Mode* to learn about basics of **Measurement Mode**.

Sample Application



Connect *Read Point Cloud V2* with *Calc Flatness Error* to calculate the flatness error of the planar point cloud.

Connect Steps

Click on *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)**.

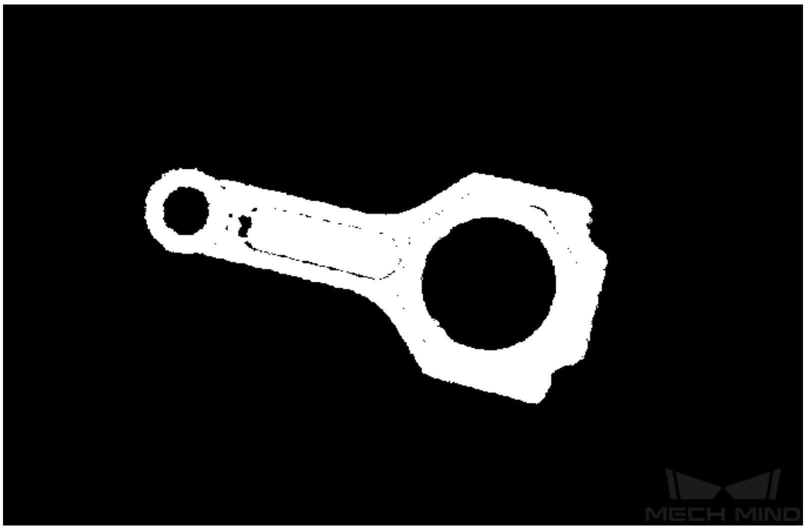
Configure parameters

Note: Please prepare the point cloud of the object to be measured in advance.

Configure in *Read Point Cloud V2*

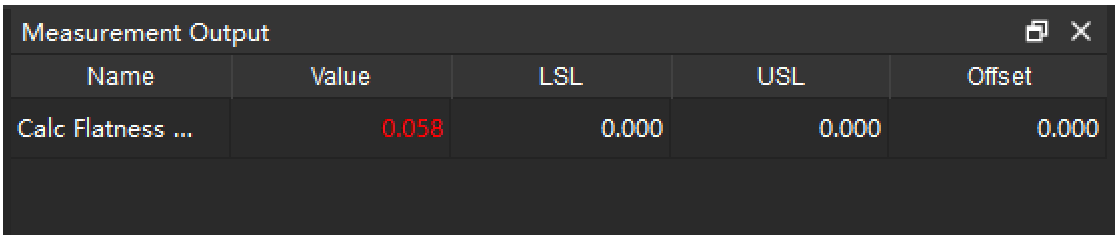
Click on the Step, and configure the **Point Cloud Source** and other parameters in the **Step Parameters** panel. After completeing the configuration, click on *Run* to run the project.

The point cloud used in this sample application is shown as below.



Read the result

After running the project, click on *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

Hint: 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.

Calc Parallelism Error

Function

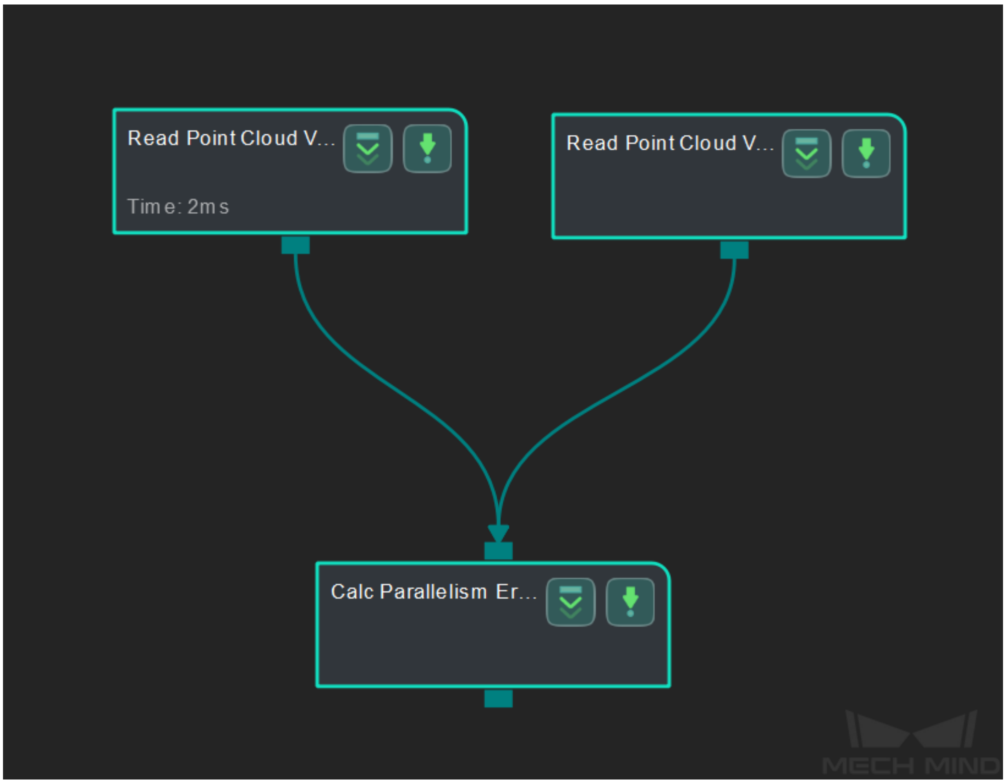
This Step is used to calculate the parallelism error from the target planar point cloud to the reference planar point cloud.

Note:

- Parallelism is defined as the quality or condition of two planes/edges being parallel. It is measured by the maximum allowable error between one plane/edge and the other.
- Parallelism error is the difference between the measured plane/edge and the reference plane/edge.

Hint: Before using this Step, please refer to *Getting Started with Measurememt Mode* to learn about basics of **Measurement Mode**.

Sample Application



Connect *Read Point Cloud V2* with *Calc Parallelism Error* to calculate the parallelism error from the target planar point cloud to the reference planar point cloud.

Connect Steps

Click on *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 **Read Point Cloud V2_2_Point Cloud With Normals** as **Input 2 (Reference Planar Point Cloud)**.

Configure parameters

Note: Please prepare the point cloud of the object to be measured and the reference point cloud in advance.

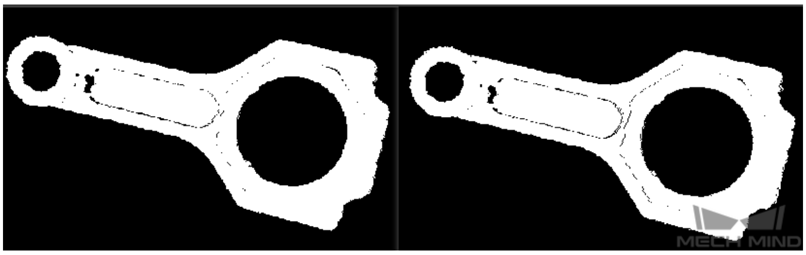
- Configure in *Read Point Cloud V2* (1)

  Click on the Step, and configure the **Point Cloud Source** (target point cloud) and other parameters in the **Step Parameters** panel.

- Configure in *Read Point Cloud V2* (2)

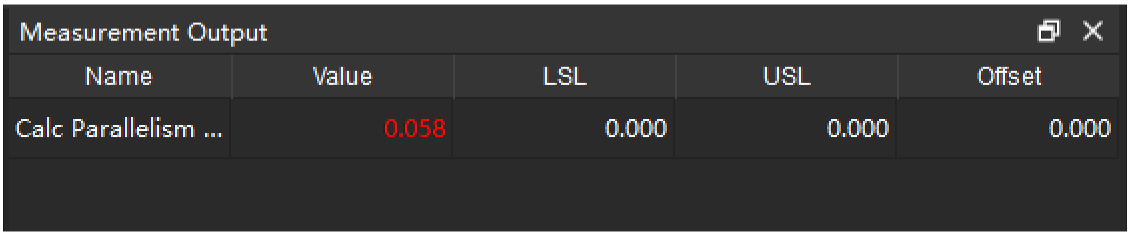
Click on the Step, and configure the **Point Cloud Source** (reference point cloud) and other parameters in the **Step Parameters** panel.

The point clouds used in this sample application are shown as below (target point cloud on the left and reference point cloud on the right).



Read the result

After running the project, click on *Calc Flatness Error* and check the parallelism error in the **Measurement Output** panel.



Hint: 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.

4.22.5 Others

Calc Box Dimensions

Calc Mean Gray Value

4.23 Meta

In this chapter, steps that are associated with Meta will be introduced.

4.23.1 Arrange by Index

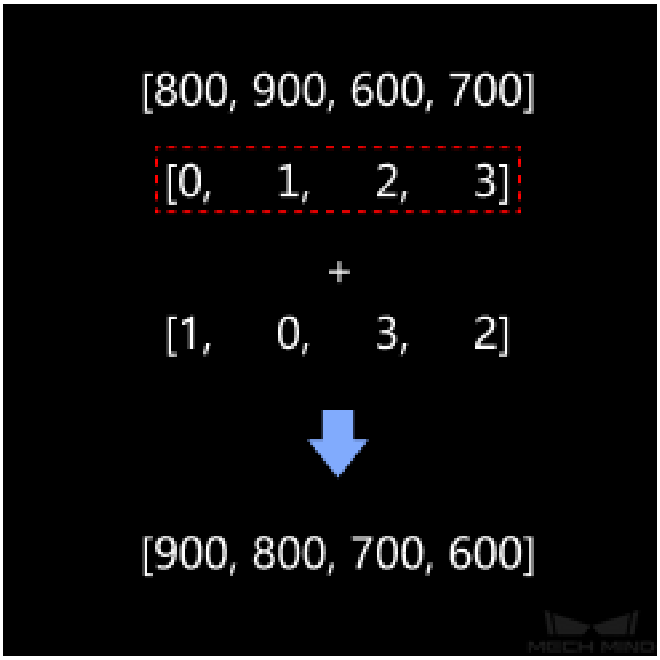
This chapter will introduce the steps in **Arrang by Index** .

Group Data

Reorder by Index List

Function

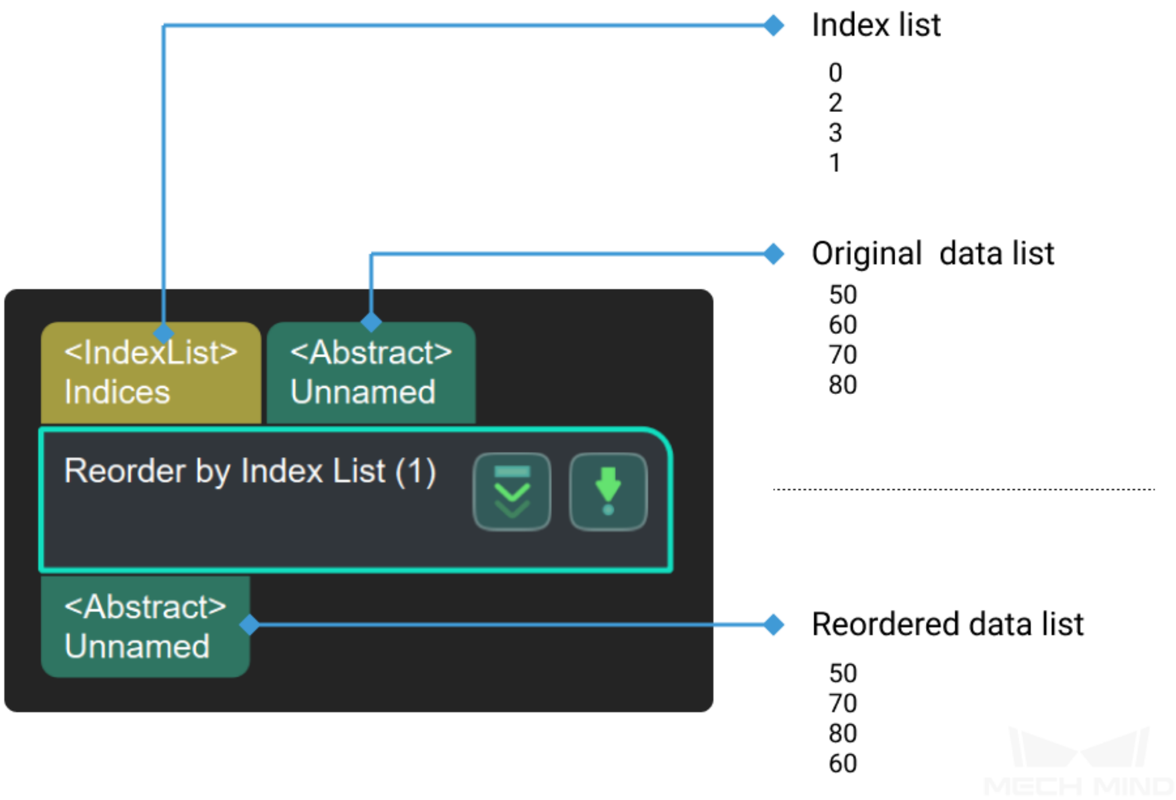
Reorder the original data according to the index.



Sample Scenario

Reorder the original data according to the input index. This Step usually follows Steps in the **Sort** category.

Input and Output



4.23.2 Change Data Dimension

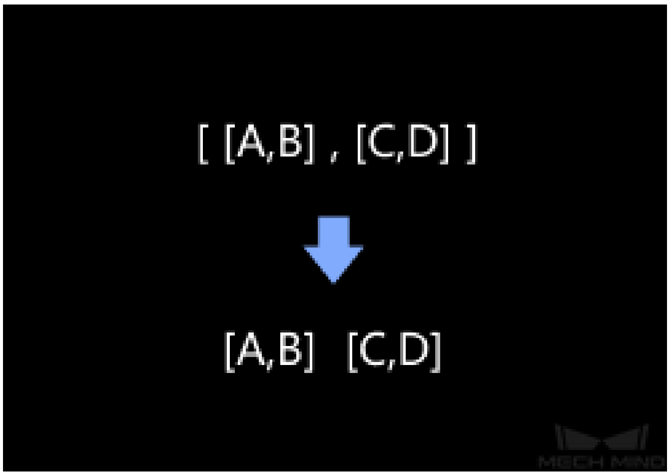
This chapter will introduce the steps in **Change Data Dimension** .

Pack

Unpack Data

Function

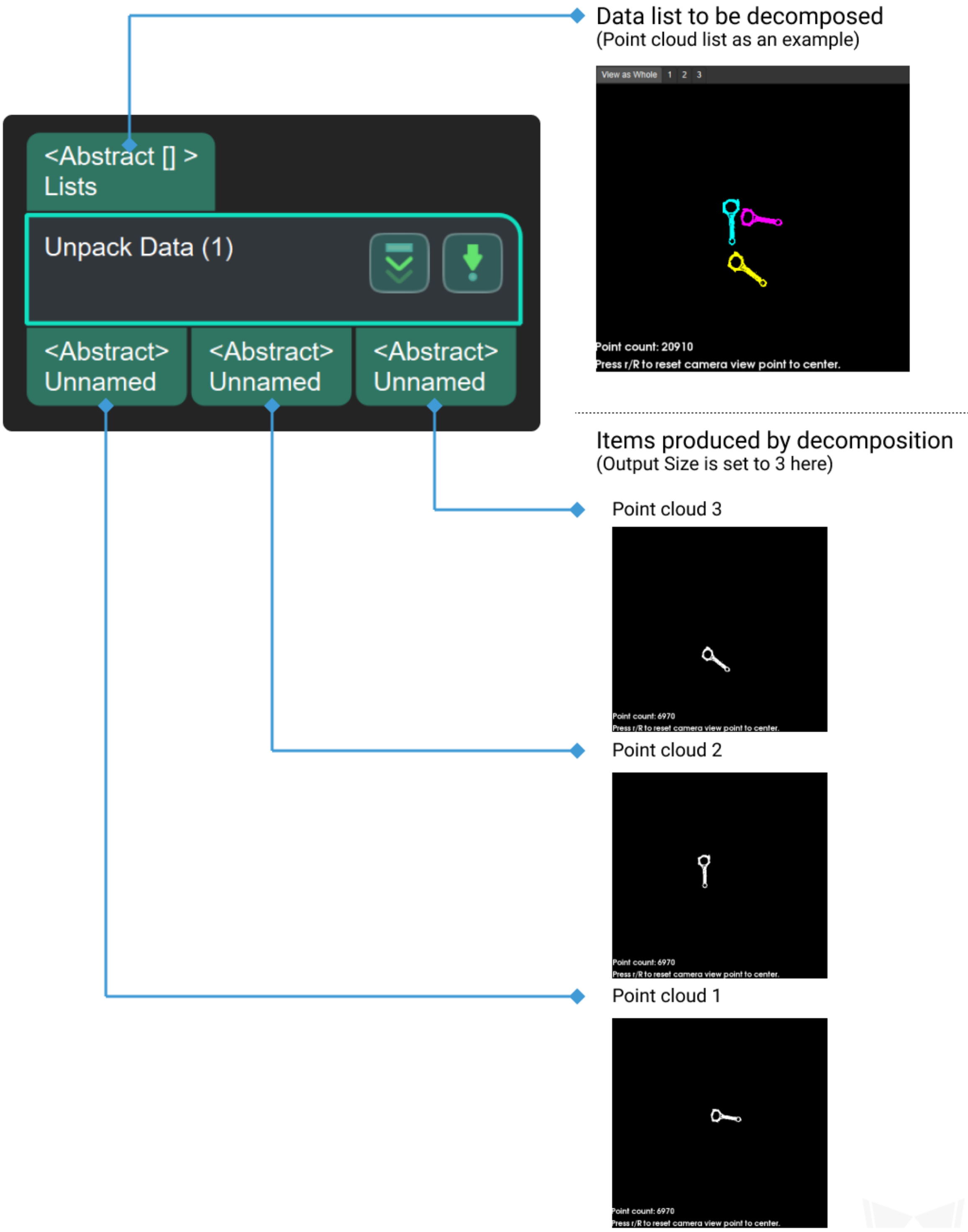
Decompose the list and obtain the first N elements in the list. The N value can be set in **Step Parameters**.



Sample 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].

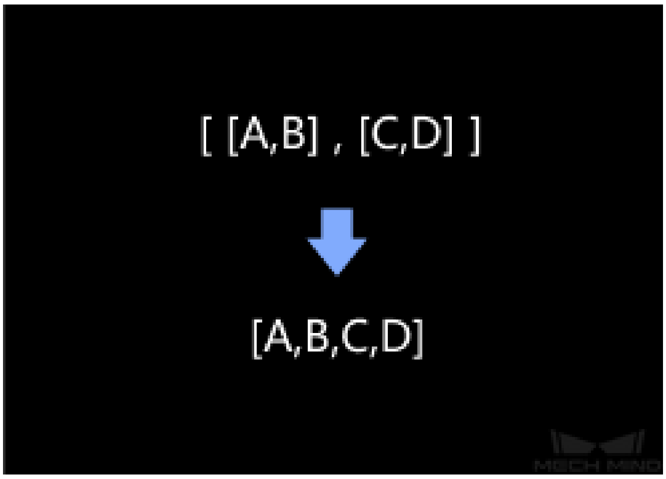
Input and Output


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.
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.

Unpack and Merge Data

Function

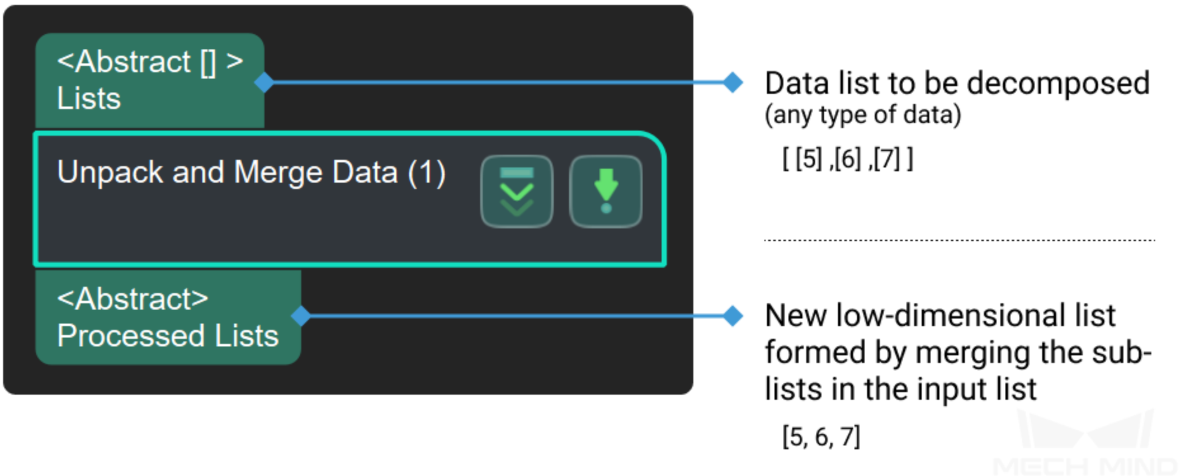
Decompose a high-dimensional data list and merge the sub-lists in it.



Sample 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



4.23.3 Procedure Out

Procedure Out

Description

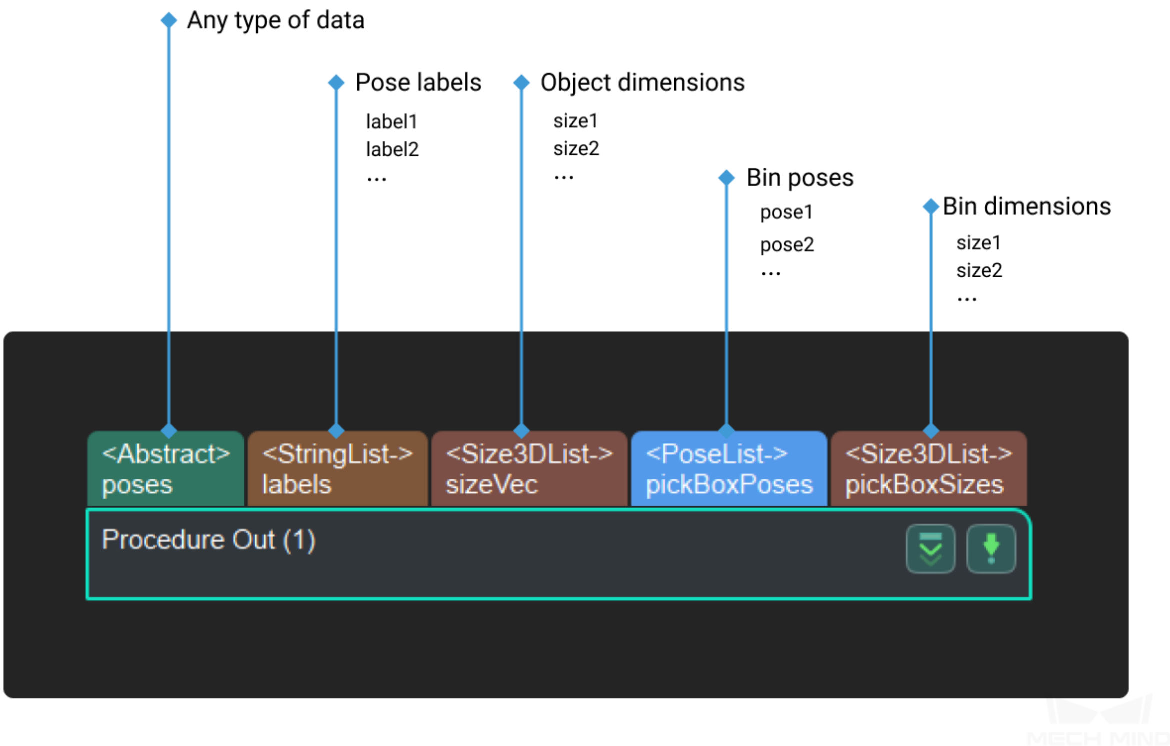
Send the results of the current project to the server.

Sample Scenario

Send various vision project results to Mech-Viz or Mech-Center via server.

Input and Output

- The input data types are shown in the figure below in predefined mode.



- The input data type can be any data type in dynamic mode.

Parameters

Output type

Default Value: Predefined

List of Values: Predefined, Dynamic

Instruction: When Predefined is selected, the output parameter corresponds to the input of the predefined mode; when Dynamic is selected, the parameter specified in the key information is output.

Output key information

Instruction: It is enabled when the step is in dynamic mode. Output the parameters with specific types or names. It can send the string key corresponding to the data and different key can be separated by “;” . One case is shown below

Dynamic mode example

4.23.4 Others

This chapter will introduce other steps in **Meta** .

Count Elements in Specified Dimension in Data List

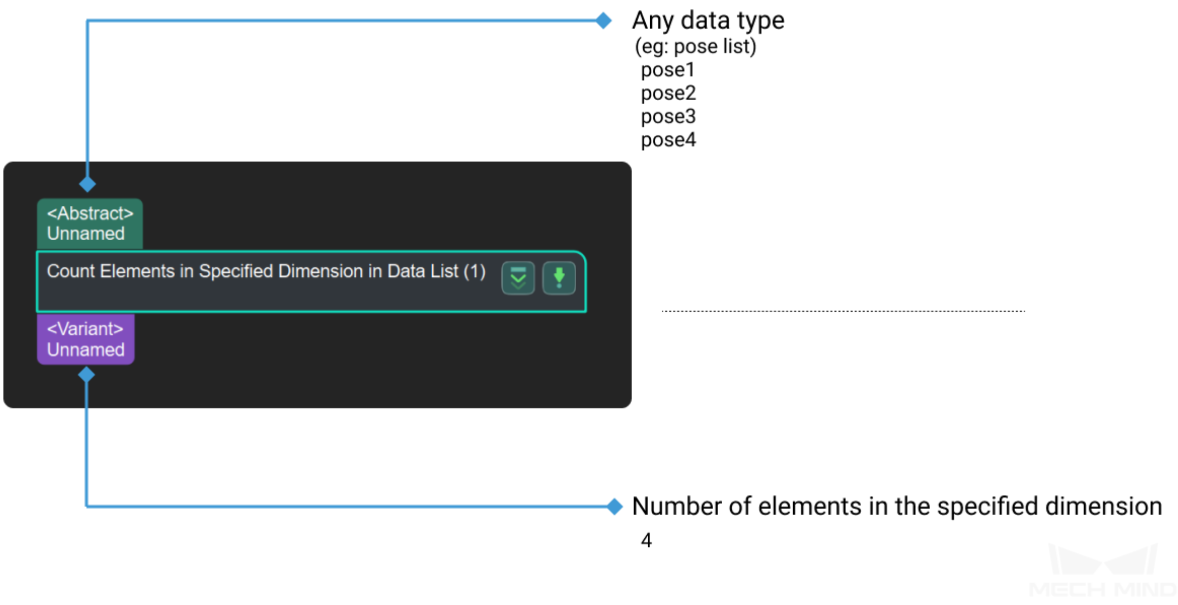
Function

Count the number of elements in a specified dimension of the input data.

Sample Scenario

A general element count Step. 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.

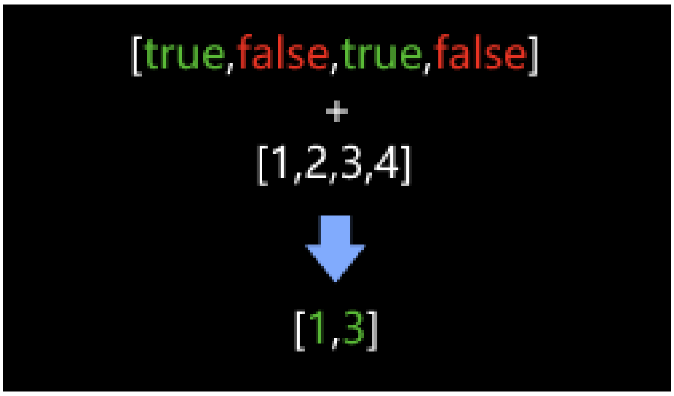
Input and Output



Filter

Function

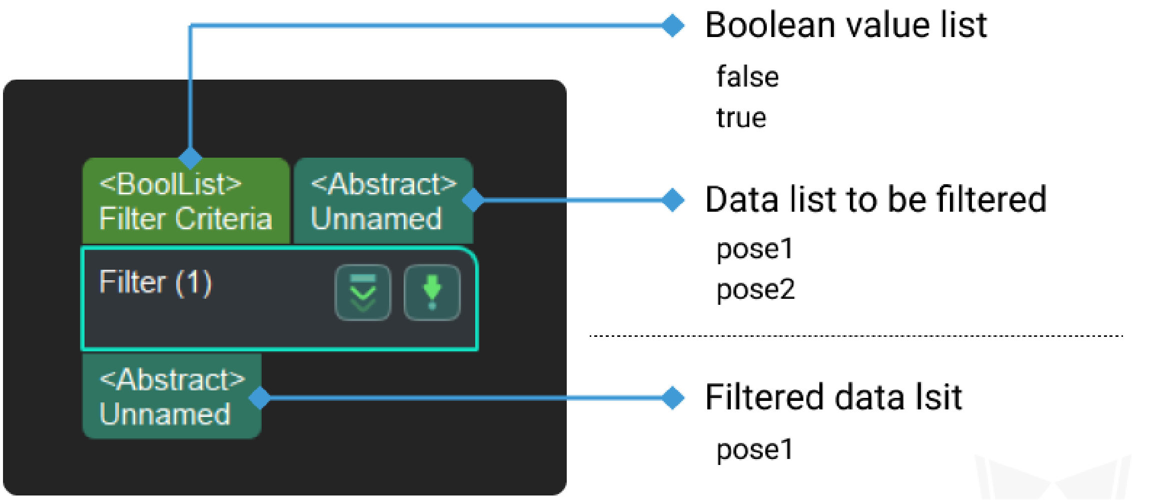
Generic filtering for any array-like variant. It takes a Boolean array of True/False as criteria and outputs the values corresponding to True values in the Boolean array as a new array.



Sample Scenario

This Step filters a data array based on a Boolean list. This Step usually follows Steps that output Boolean lists, including *Dichotomize Values by Threshold*, *Validate Labels and Output Flags*, *Validate Poses by Included Angle to Reference Direction*, *Validate Point Clouds*, etc.

Input and Output



Parameters

Ports Settings

Port Number

Default Value: 1

Value Range: 1 ~ 15

Instruction: This parameter indicates the number of the input and output ports.

Operation Layer

Default Value: 0

Value Range: 0 ~ 14

Instruction: If the operation layer is `0` , the operation will perform on every elements in the lists.

Reverse Bool List

Default Value: False

List of Values: True, False

Instruction: When checked as `False` , the items with False in input BoolList will be filtered, and return items with True; When checked as `True` , the items with True in input BoolList will be filtered, and returen items with False.

Merge Data

Function

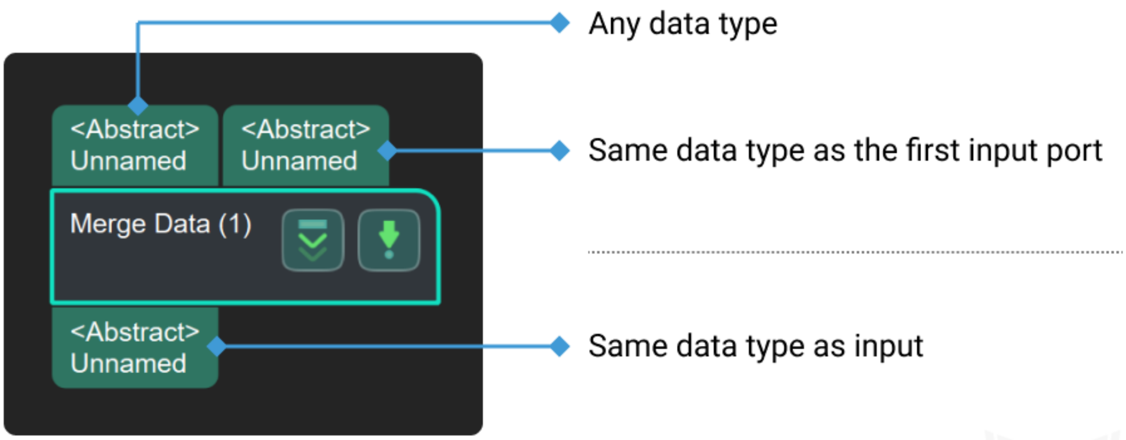
General data merging Step.

Sample Scenario

To merge data of types such as pose, number, etc.

This Step is new, and it is recommended to replace earlier version Steps, including *Merge Pose Lists* and Merge Cloud Vector, with this Step.

Input and Output



Repeat Data and Concatenate Copies

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].

Sample 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

Reverse

Trim Input List

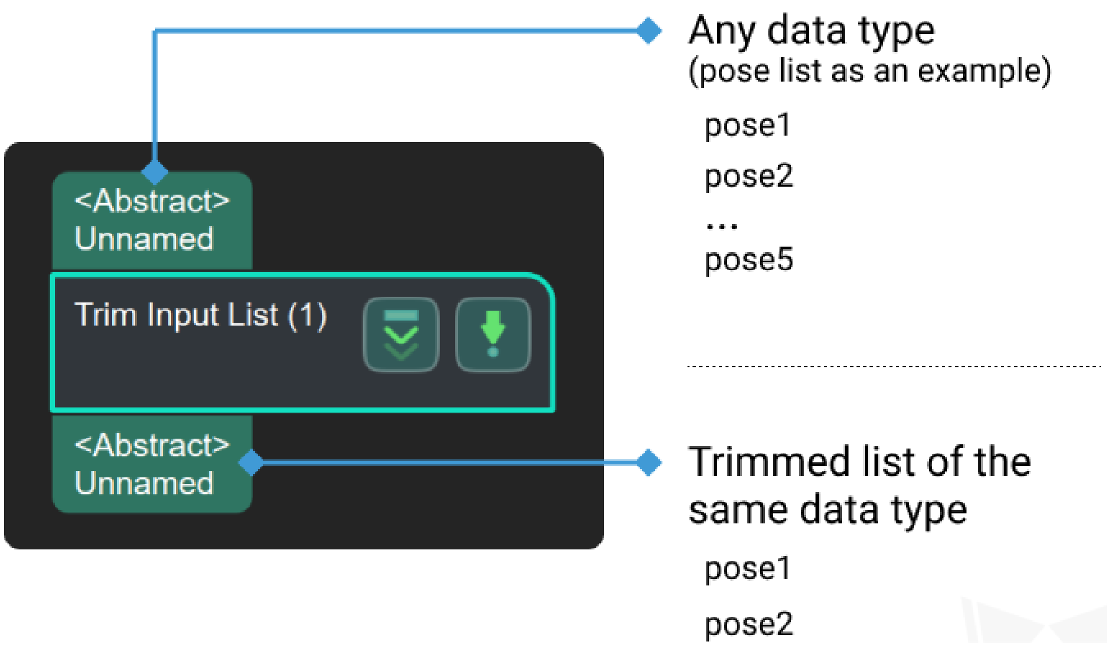
Function

Keep the first N items in a data list and output. Items after the Nth item will be discarded.

Sample Scenario

Extract the first N items in a list, and the N value can be set in the parameters by project requirements.

Input and Output



4.24 Pose

This section includes descriptions of Steps related to poses.

4.24.1 2D Pose

Evaluate 2D Poses

Group 2D Poses

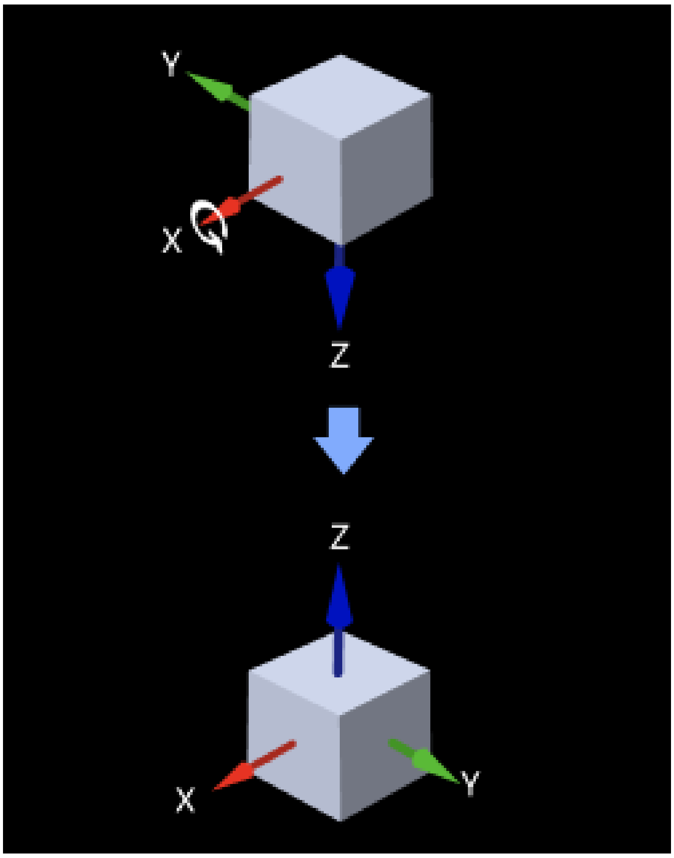
4.24.2 Adjust Orientation

Adjust 3D Poses by 2D Poses

Flip Poses' Axes

Function

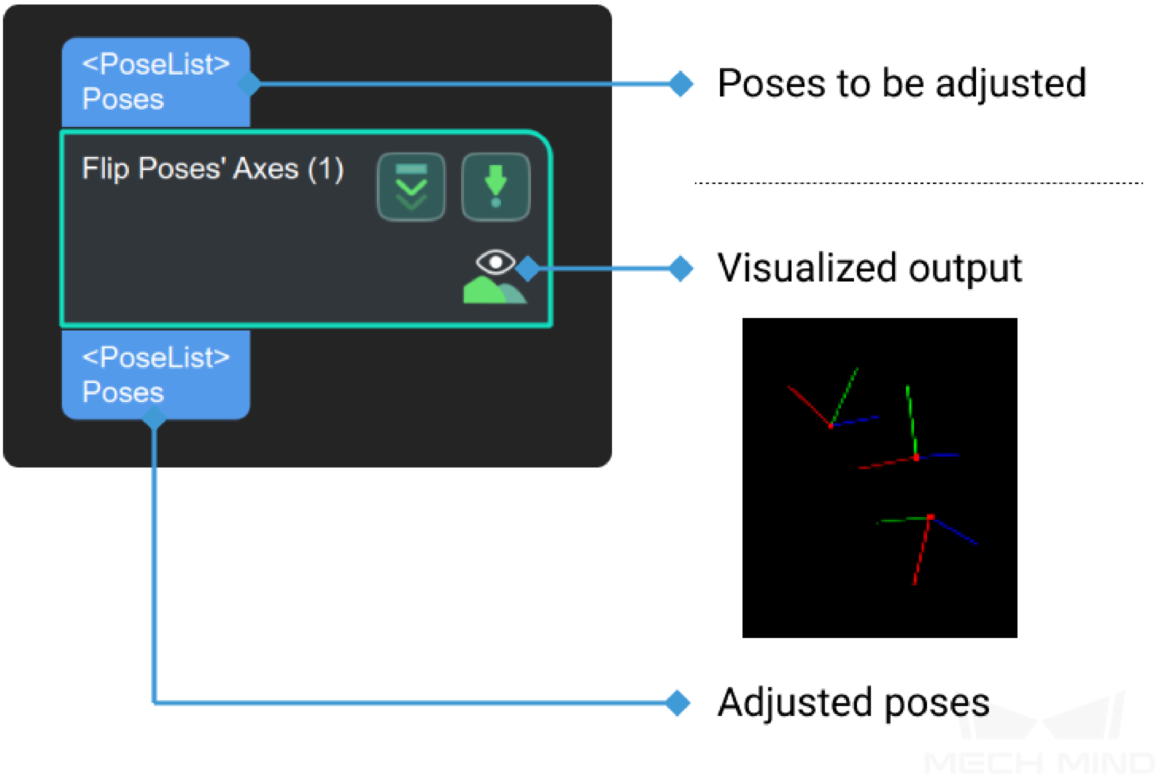
Flip the selected axis to the positive or negative directions around a specified axis of rotation, as shown below.



Sample 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



Parameters

axis property

axisType

Default Value: Z

List of Values: X, Y, Z

Instruction: Axis to be flipped.

directionType

Default Value: Negative

List of Values: Positive, Negative

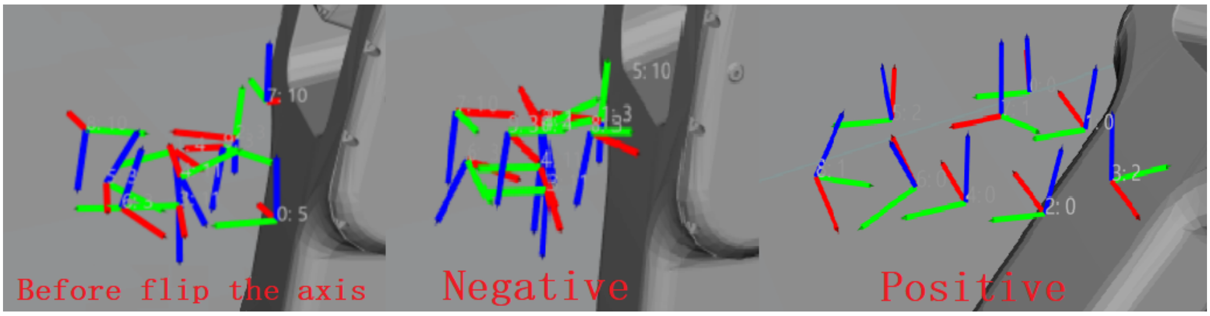
Instruction: the direction of the axis to be flipped. **Positive** means the angle between the setting axis and the positive direction of the same axis of world coordinate must less than 90°. If the angle is more than 90°, it's necessary to flip the setting axis, otherwise does not need to flip the axis. **Negative** means the angle between the setting axis and the negative direction of the same axis of world coordinate must less than 90°. If the angle is more than 90°, it's necessary to flip the setting axis, otherwise does not need to flip the axis.

rotateByAxis

Default Value: X
List of Values: X, Y, Z
Instruction: Decide which axis the rotation will around.

Example

Take the axis as the Z axis and the rotation axis as the X axis as an example :



Point Axes of Poses to Given Direction

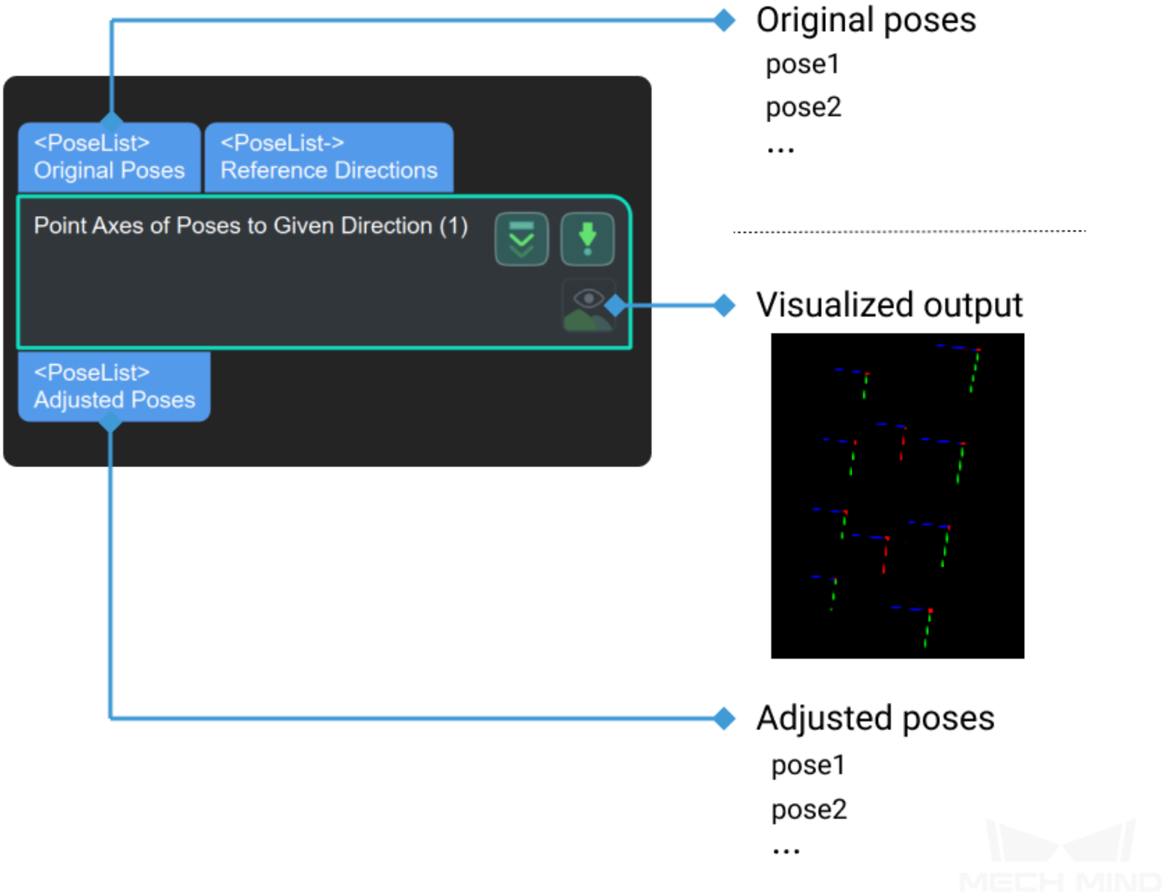
Function

Adjust the orientations of axes of poses by a set reference direction.

Sample Scenario

For various scenarios where pose orientations need to be adjusted. This Step belongs to an earlier version. If you need to rotate poses by object symmetricity (that is, to set the parameter **Rectification Method** of this Step to **ROTATION**), please instead use the newer Step *Rotate Poses to Directions with Symmetry Constraint* and use it with the Step *Easy Create Vector3Ds*.

Input and Output



Parameters

method

rectifyMethod

Default Value: HARD_RECTIFY
List of Values: HARD_RECTIFY, ROTATION

HARD_RECTIFY

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, using all reference poses' corresponding axis type. If there isn't a second input (reference pose), "reference Dir" determines the reference direction.

ROTATION

According to the symmetry of the object, set the symmetrical angle so that the direction of the rotation axis is roughly consistent with the specified reference axis after rectification.

axis

axisType

Default Value: Z

List of Values: X, Y, Z

Instruction: The rotation axis to be rectified.

used by rotation method

symmetryAngle

Default Value: Z

List of Values: X, Y, Z

Instruction: reference axis.

useFirstRefPose

Default Value: Z

List of Values: X, Y, Z

Instruction: It can be modified only when there exist a second input (reference pose). Select the reference direction using the first reference pose.

reference direction

x

Default Value: 0

Instruction: The direction vector of the reference axis x.

y

Default Value: 0

Instruction: The direction vector of reference axis y.

z

Default Value: 1

Instruction: The direction vector of the reference axis z.

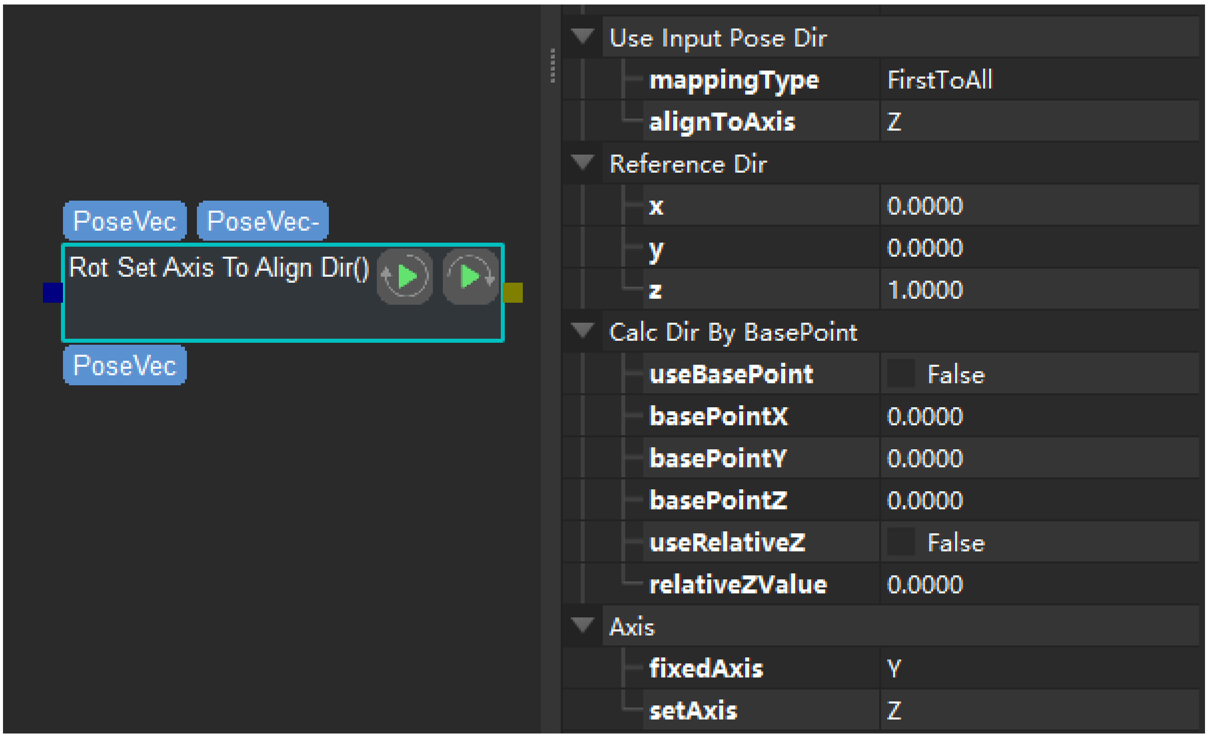
Typical case

It can be used to fix a certain axis of the object' s pose. For example, the x-axis direction of the cube object is randomly set, and some robots can' t rotate their grippers to certain angle, causing the unreachable pick point.

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.

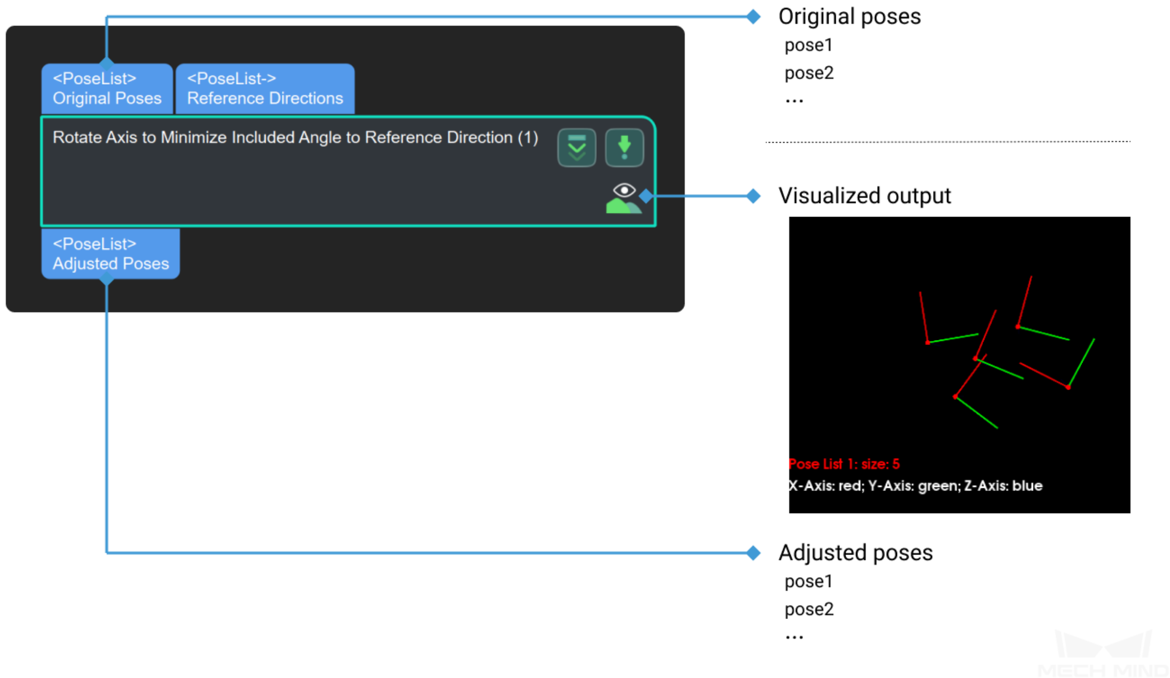


Step interface and parameters

Sample Scenario

For various scenarios where pose orientations need to be adjusted. This Step belongs to an earlier version. Please instead use *Rotate Poses to Directions Freely* which is newer and has a wider range of functions, and use it with *Easy Create Vector3Ds*.

Input and Output



Parameters

use Input Pose Dir

It is enabled when there exists a second input (reference pose).

mappingType

Default Value: FirstToAll

List of Values: FirstToAll, OneToOne

Instruction: Input mapping method of posture and reference posture. FirstToAll maps the first reference pose to all input poses; OneToOne maps the reference poses to the input poses one by one, which requires the number of reference poses and input poses are equal.

alignToAxis

Default Value: Z

List of Values: X, Y, Z

Instruction: Select the reference axis from the reference pose.

Reference Direction

x

Default Value: 0

Instruction: The direction vector of the reference axis x.

y

Default Value: 0

Instruction: The direction vector of reference axis y.

z

Default Value: 1

Instruction: The direction vector of the reference axis z.

Calc Dir By BasePoint

useBasePoint

Default Value: False

List of Values: True, False

Instruction: Whether to use the base point to calculate the reference axis.

basePointX

Default Value: 0

Instruction: X coordinate value of base point.

basePointY

Default Value: 0

Instruction: Y coordinate value of base point.

basePointZ

Default Value: 0

Instruction: Z coordinate value of base point.

useRelativeZ

Default Value: False

List of Values: True, False

Instruction: If the height of the object will change during the production process, it is recommended that set it to true.

relativeZValue

Default Value: 0

Instruction: The Z-direction value of the object = base point Z coordinate value + the relative Z value.

Axis

fixedAxis

Default Value: Y

List of Values: X, Y, Z

Instruction: Rotation axis

setAxis

Default Value: Z

List of Values: X, Y, Z

Instruction: Axes that needs to be rotated.

Typical case

It can be used to set the coordinate axes of multiple objects to the same direction. For example, when sending visual point, in order to calculate the offset, the directions of all objects' coordinate axes should be the same.

Rotate Poses around Given Axis

Function

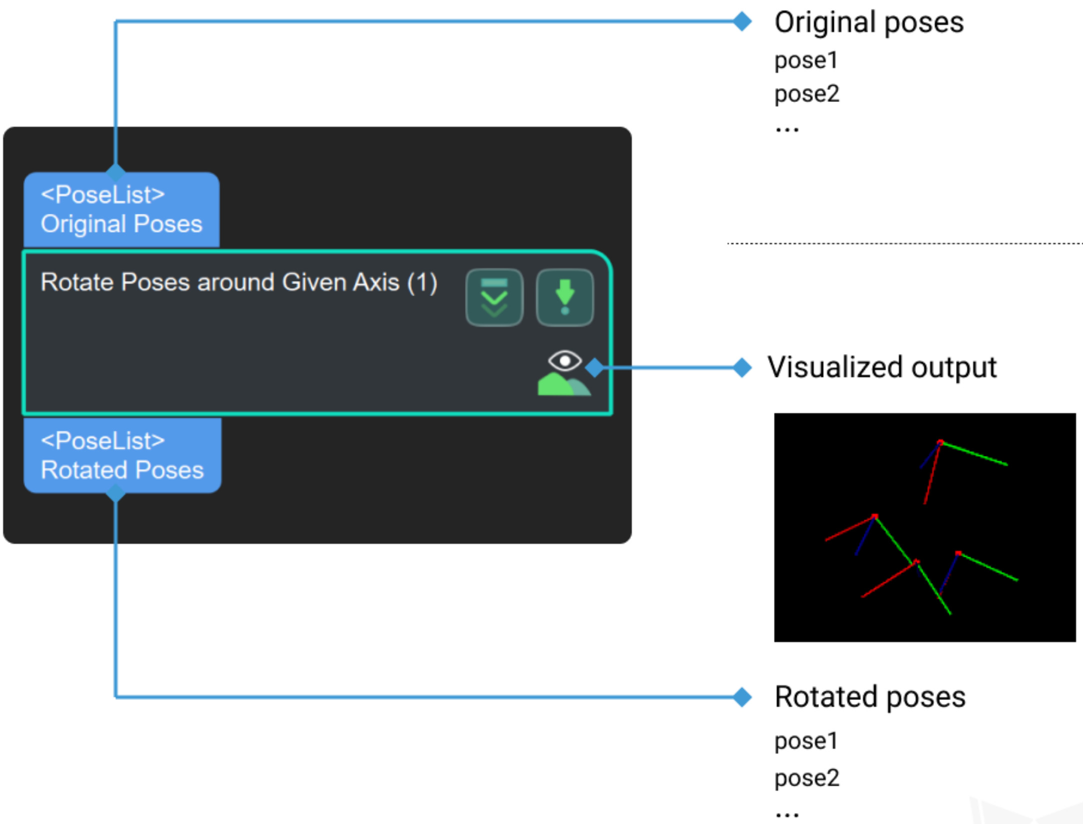
Rotate poses around a specified axis.

Sample Scenario

General pose rotation Step. No fixed usage scenarios.

This Step belongs to an earlier version. Please instead use the newer Step, *Rotate Poses around Axis by Angle*, in conjunction with *Easy Create Vector3Ds*.

Input and Output



Set Pose Quaternion

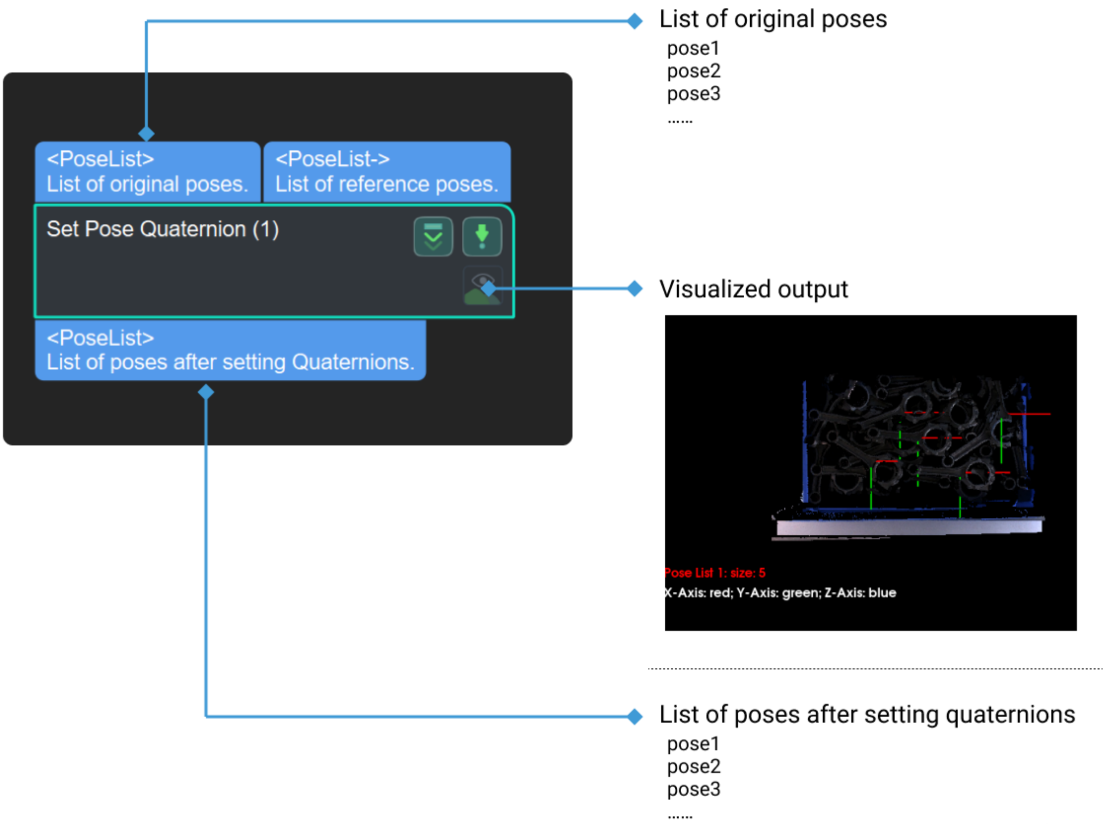
Function

Set the orientations of poses (in quaternion) by the reference pose input from the second port, or the quaternion set in the parameter.

Sample 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



4.24.3 Adjust Translation

Adjust Poses by Offsets

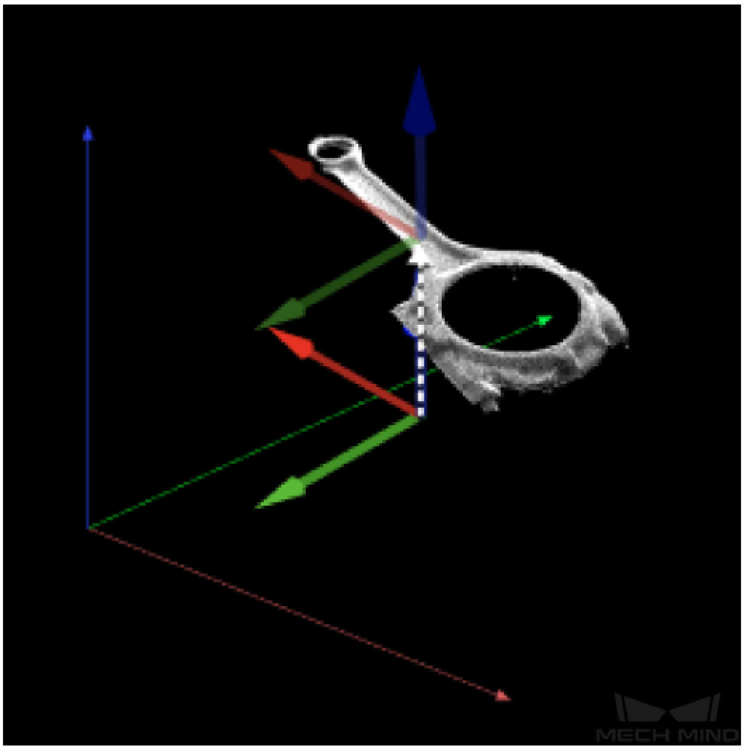
Adjust Poses by Tilt

Adjust Poses to Point Cloud Surface

Move Poses to Point Cloud Surfaces along Z-Axis

Function

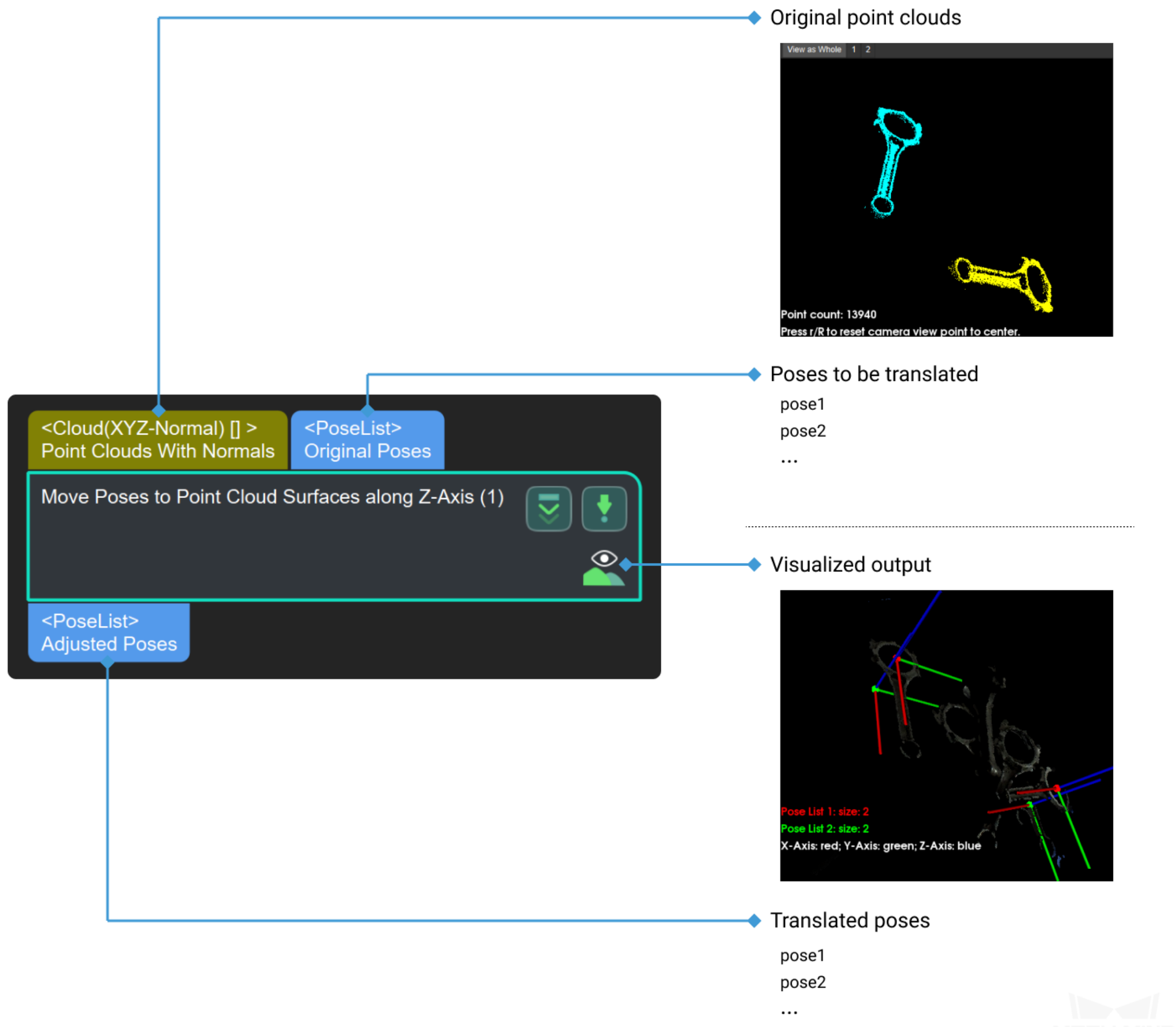
Move poses to point cloud surface along the Z-axis of a specified reference frame according to user-set rules.



Sample 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, concave/convex surface. It is usually used after the Step *Calc Poses and Dimensions from Planar Point Clouds*.

Input and Output



Offset Poses in Cylinder

Set Pose Translation

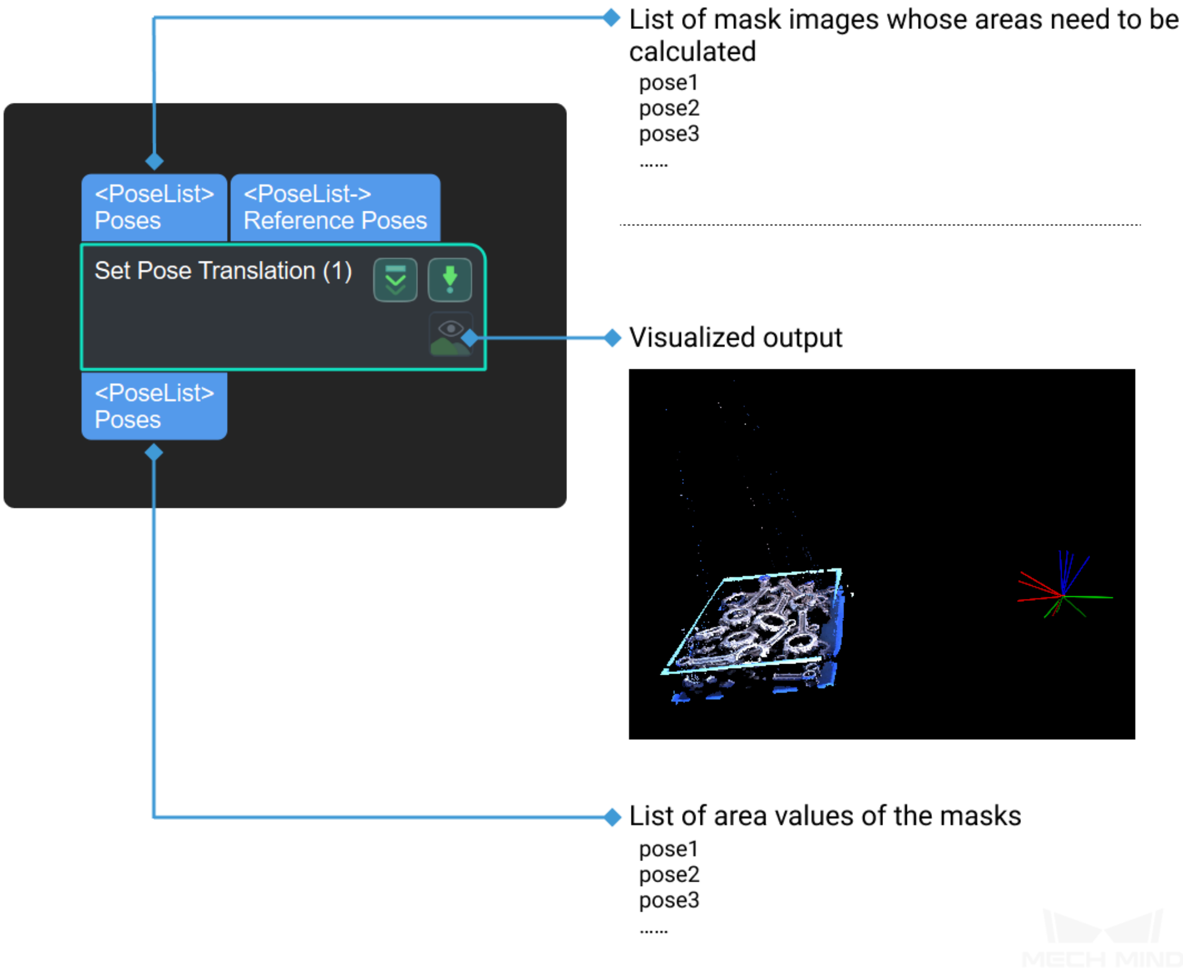
Function

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**.

Sample 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



Translate Poses along Given Direction

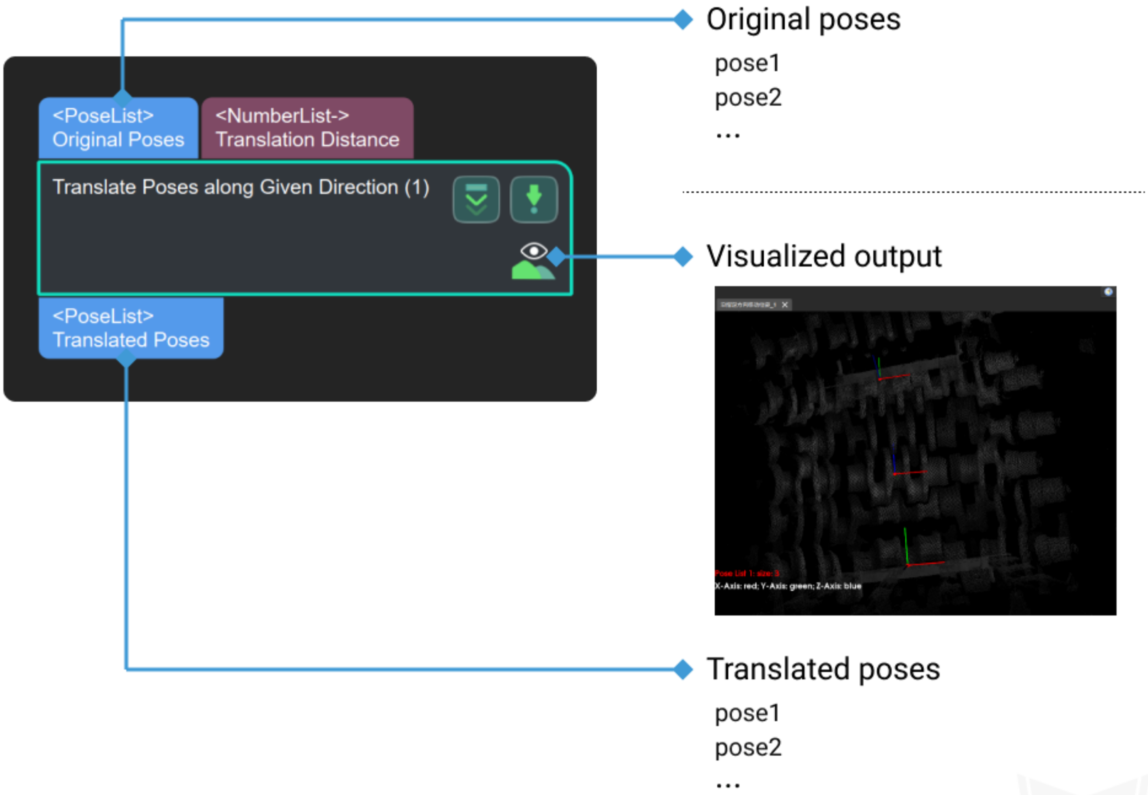
Function

Translate poses along a given direction.

Sample Scenario

Translate poses according to the direction and distance set by the user.

Input and Output



Parameters

Translation Direction Settings

- Translation Distance
- **Translation Direction Source**
  - CustomizedDirection:
  - ObjectAxis:

Customized Direction

- X
- Y
- Z

4.24.4 Adjust Translation & Orientation

Adjust Inaccurate Poses Caused by Camera Distortion

Adjust Poses by Obstacles V2

Rectify Ring Pose

4.24.5 Collect/Remove

Get Highest Layer Poses

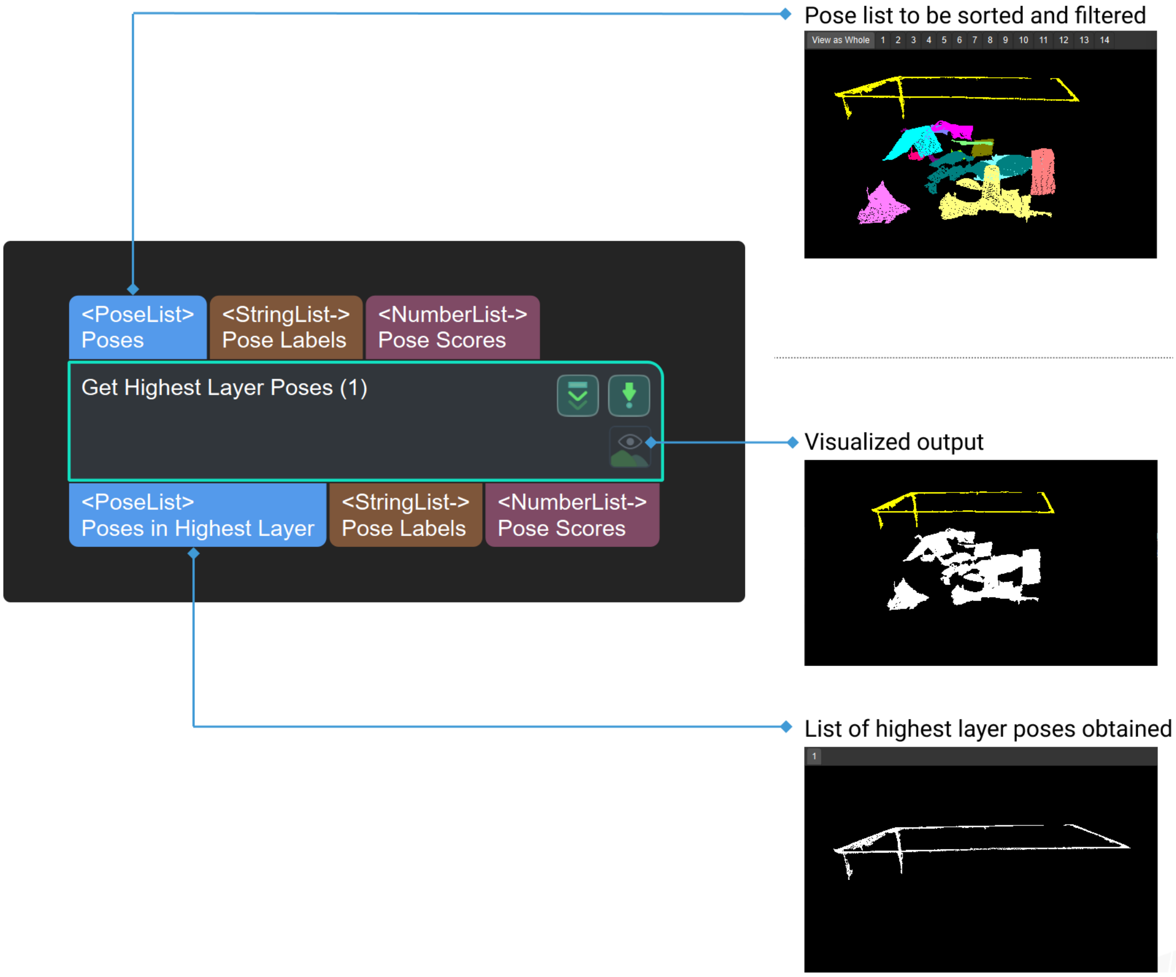
Function

Sort the poses in descending order along the specified direction, and then output the highest layer poses.

Sample Scenario

A general pose filtering Step. Output poses that meet the requirements according to the set parameters.

Input and Output



Keep Poses Distributed in Regular Polygon

Remove All Overlapping Poses

4.24.6 Conversion

From PoseList to Matrix4D

From PoseList to PoseLists

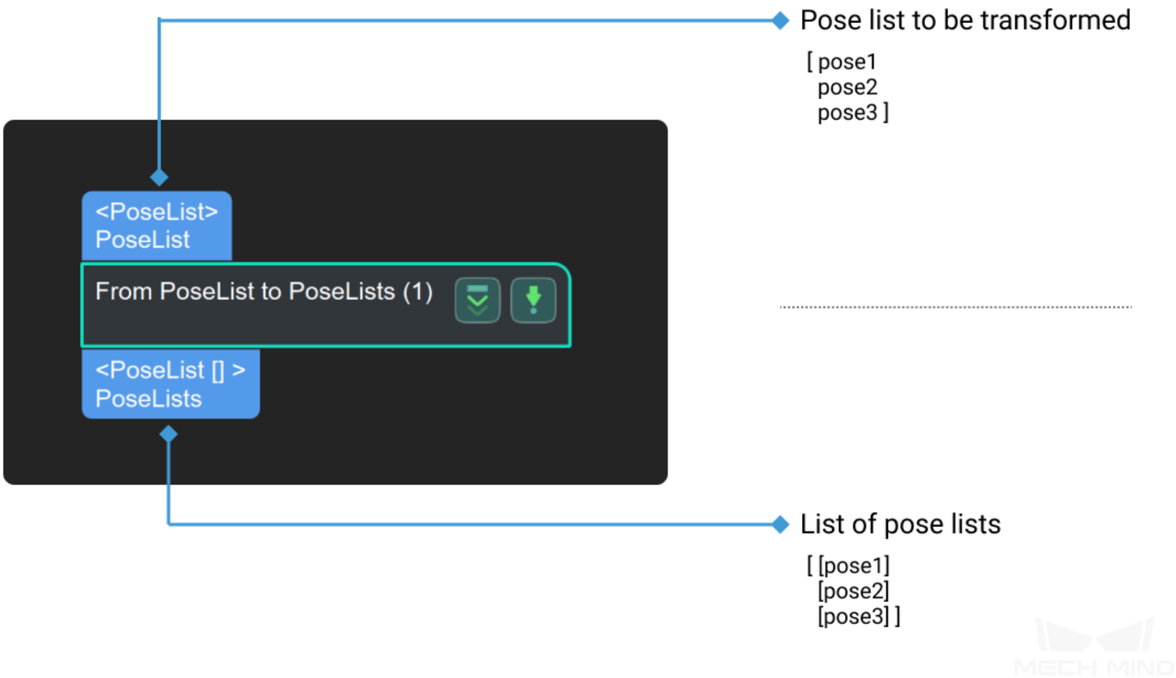
Function

Change each pose in the pose list to a one-element pose list.

Sample Scenario

A general data structure transformation Step. For instance, 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



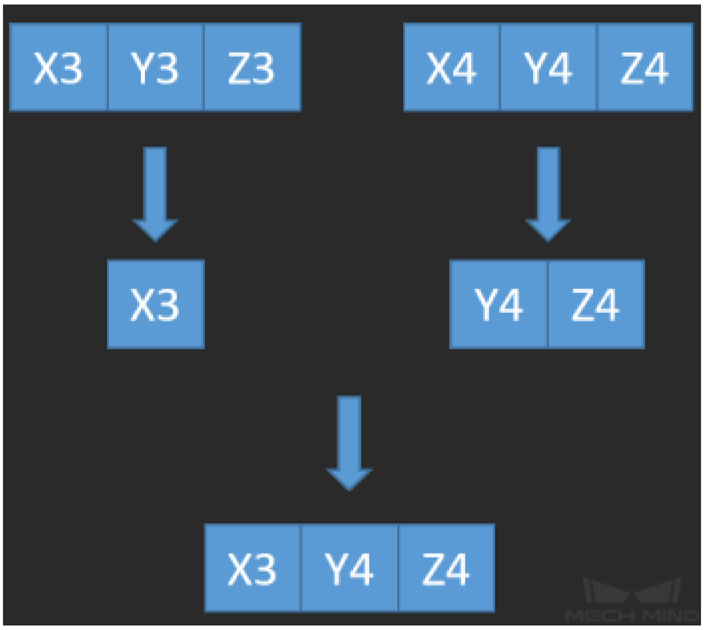
From Poses to Euler Angles

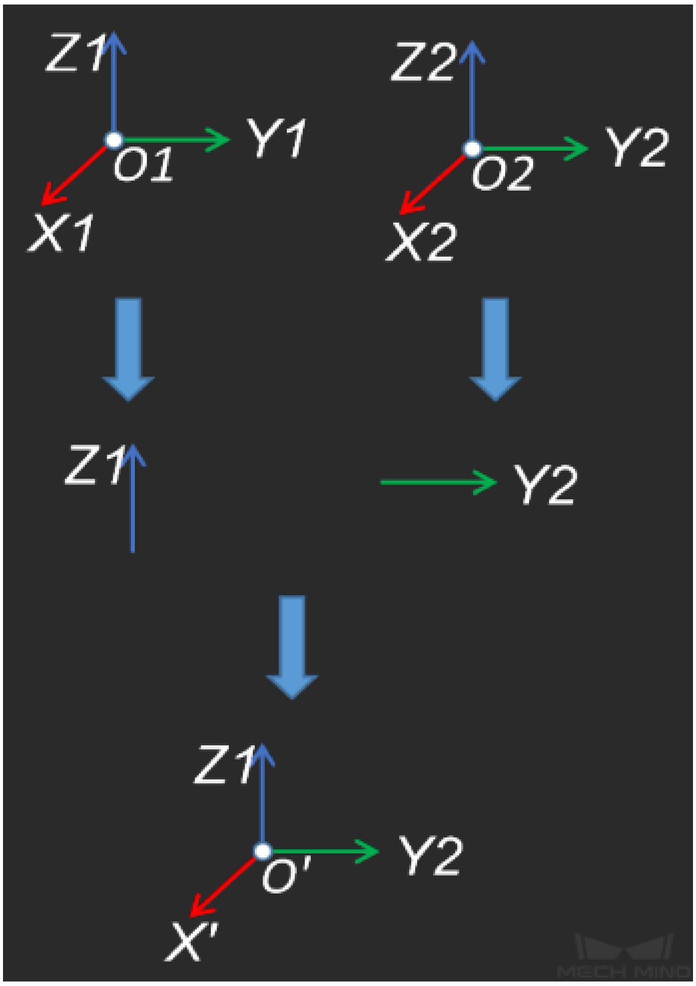
4.24.7 Generate Pose Offset

Compose New Poses by Combining Parts of Input Poses

Function

Combines the specified parts of the input poses to form a new pose.

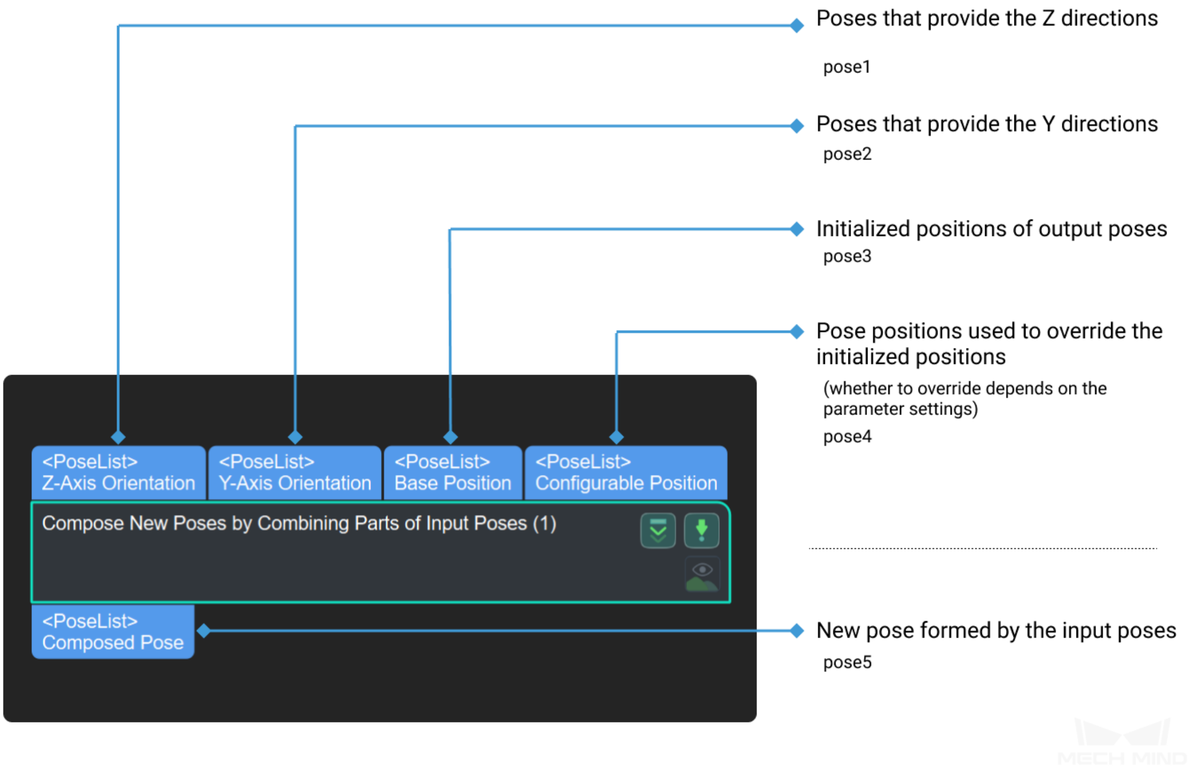

X3
Y3
Z3
X4
Y4
Z4
X3
Y4
Z4
X3
Y4
Z4


Z1
Y1
O1
X1
Z2
Y2
O2
X2
Z1
Y2
Z1
Y2
O'
X'

Sample Scenario

A general pose calculation Step. No fixed usage scenarios.

Input and Output



Parameters

translationStrategy

Default Value: X

List of Values: X, Y, Z, XY, XZ, YZ, XYZ, NONE

Instruction: Select the translation parameter, for example, select X, the corresponding operation is to set the X axis with the third input while Y and Z axes with the fourth input.

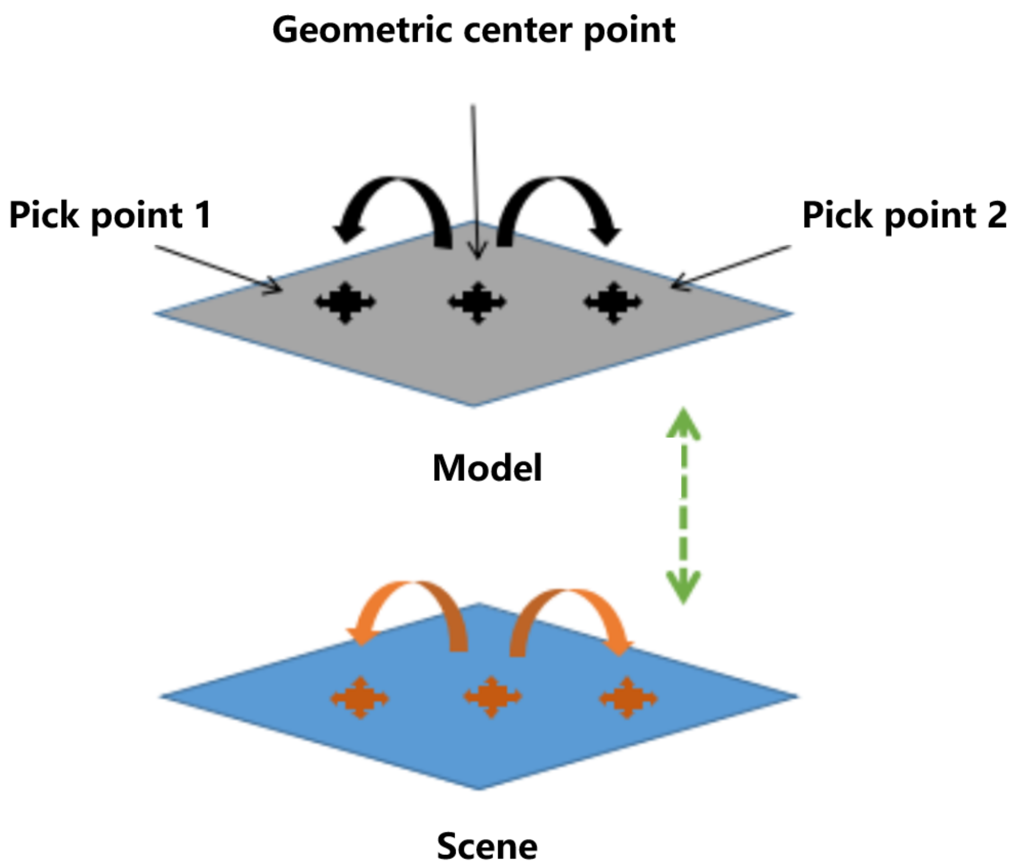
Generate Discrete Poses Revolving around Reference Pose

Generate Pick Points

Map to Multi Pick Points

Function

Calculate other pick points of objects through object geometric center points and their mapping relationships with the multiple pick points.



Schematic diagram of mapping to multiple pick points

Sample Scenario

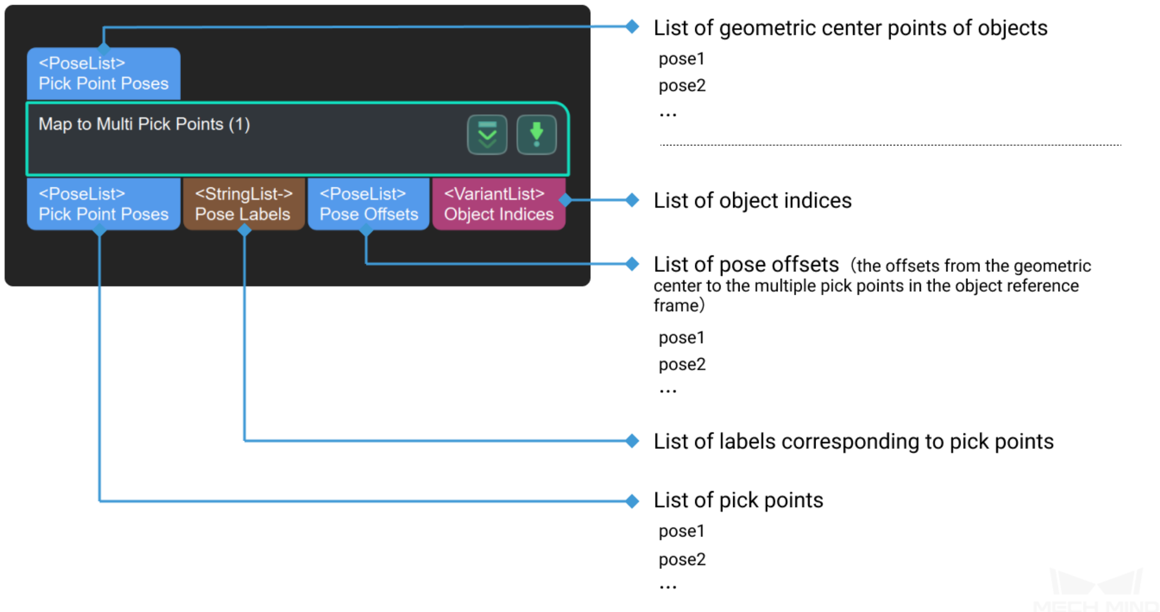
This Step is required if the objects' pick points do not coincide with their geometric center points, or if there is more than one pick point on each object.

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



Parameters

geoCenterPointPath

The pose of the object' s geometric center point on the template.

Instruction: Absolute path or relative path of the geometric center point file.

ObjPlacePointPath

Pose of object placement point.

Instruction: The absolute path or relative path of object placement point file.

pickpointsPath

The pose of the object' s pick point on the template.

Instruction: The absolute or relative path of the pick point file.

labelFilePath

It has labels corresponding to the pick point.

Instruction: Absolute path or relative path of label file.

4.24.8 Validate

Is Z Value of Input Greater than Threshold

Validate 2D Poses within Mask

Validate Existence of Poses in 3D ROI

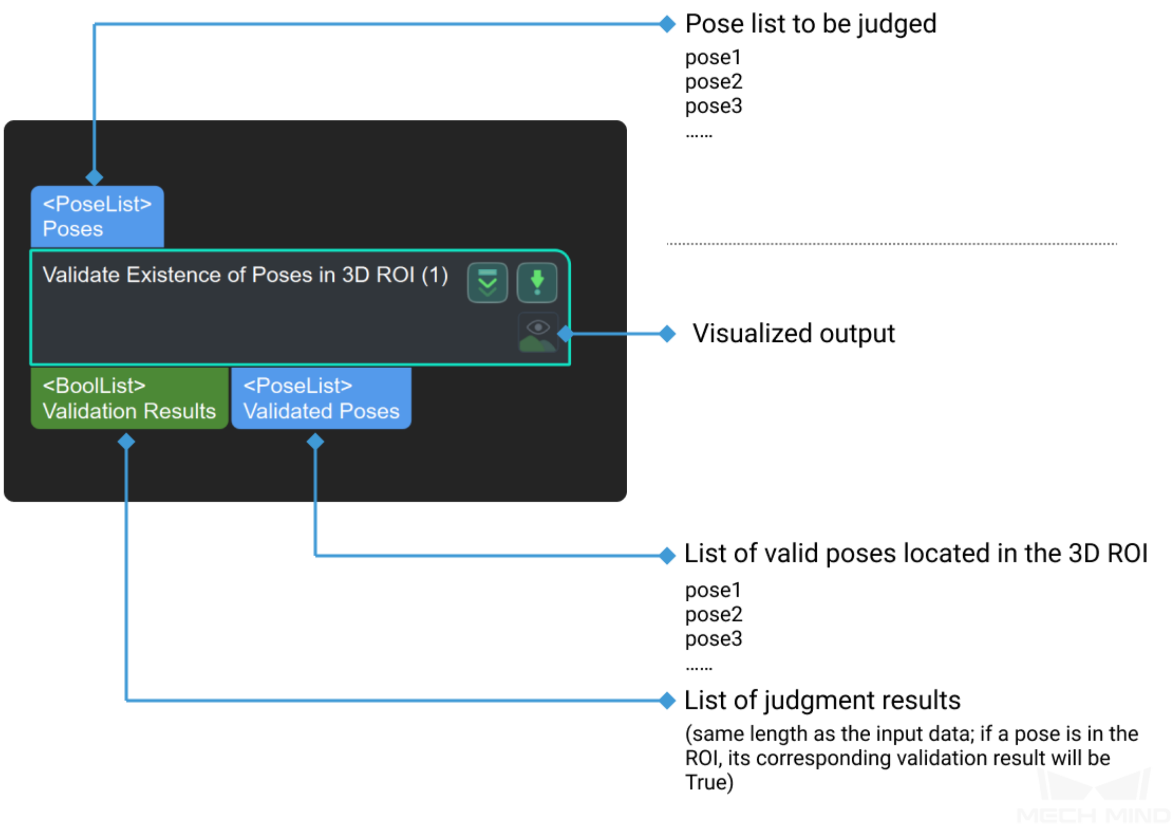
Function

Determine whether the input poses are in the set 3D ROI, and output the validation result and the poses located in the 3D ROI.

Sample Scenario

A general pose filtering Step, filtering by 3D ROI and outputting the validation results.

Input and Output



Validate Poses by Included Angle to Reference Direction

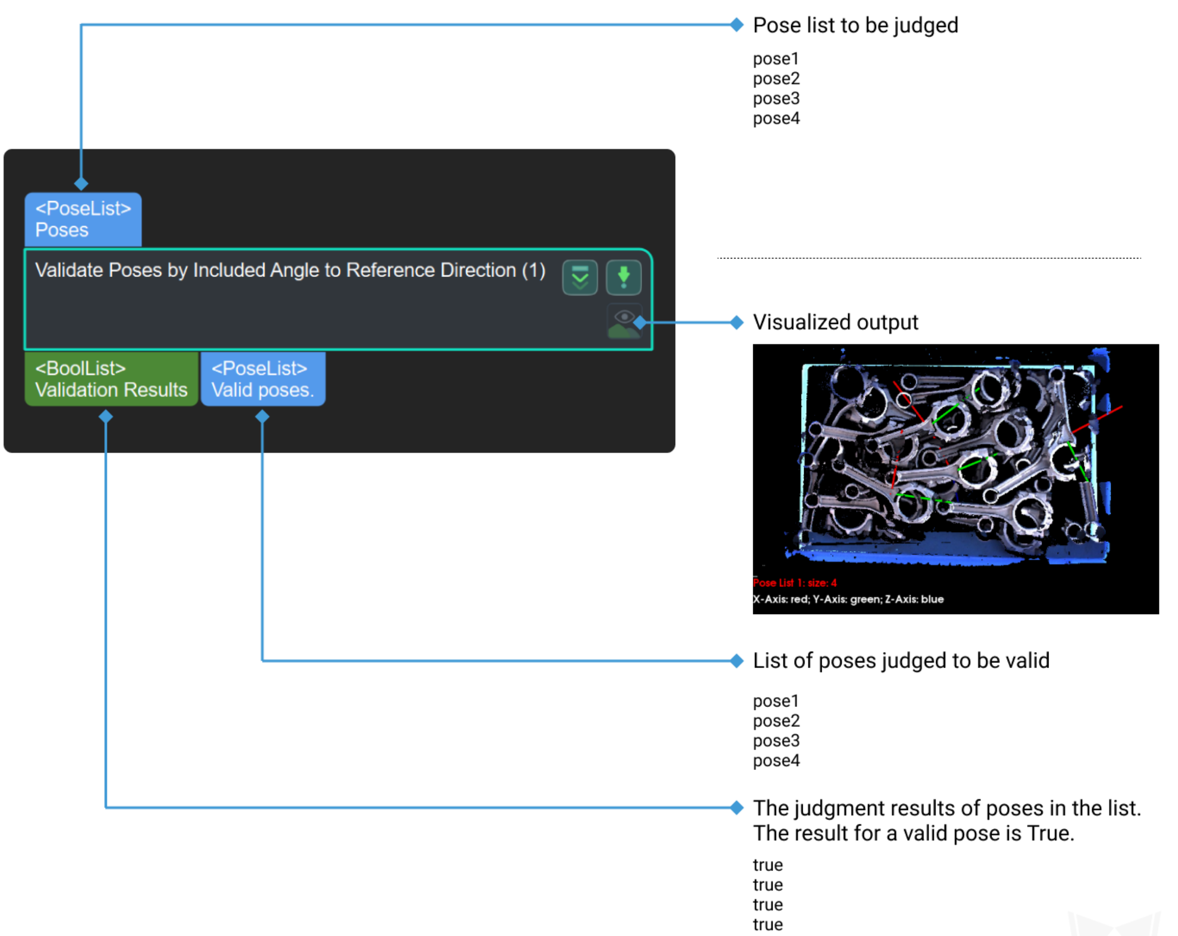
Function

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.

Sample Scenario

A general pose filtering Step. No fixed usage scenarios.

Input and Output



Validate underlying Poses

4.24.9 Others

Calc Included Angle between Specified Axis of Poses

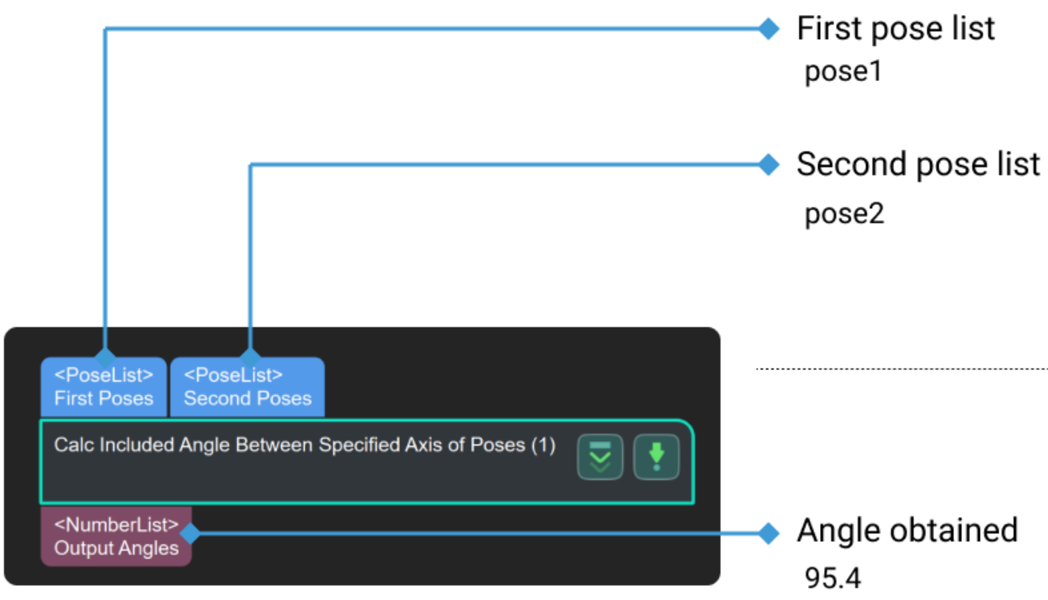
Function

Calculate angles between specified axes of input pose pairs.

Sample Scenario

General angle calculation Step. No fixed usage scenarios.

Input and Output



Calc Projection Length along Reference Direction

Find Correspondence between Poses and Offsets

4.25 Pose2

4.25.1 ConvenientHelpers

Easy Create Index List

Easy Create Number List

Easy Create Poses

Easy Create Quaternions

Easy Create String List

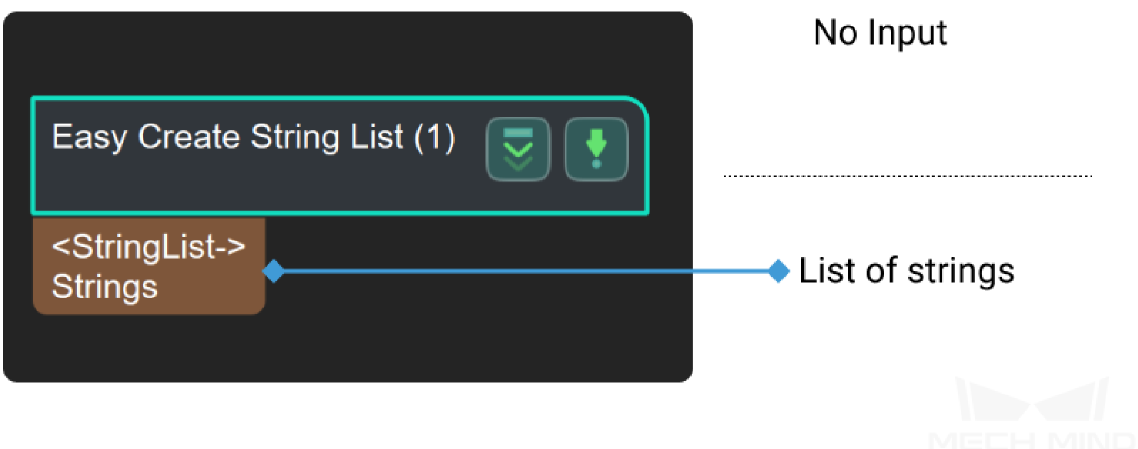
Function

Create a list of labels from a list of strings entered in the parameter.

Sample Scenario

A general label list creation Step.

Input and Output



Easy Create Vector3Ds

4.25.2 Coordination Transform

Easy Coordinate Transform

4.25.3 Inversion

Inverse Poses

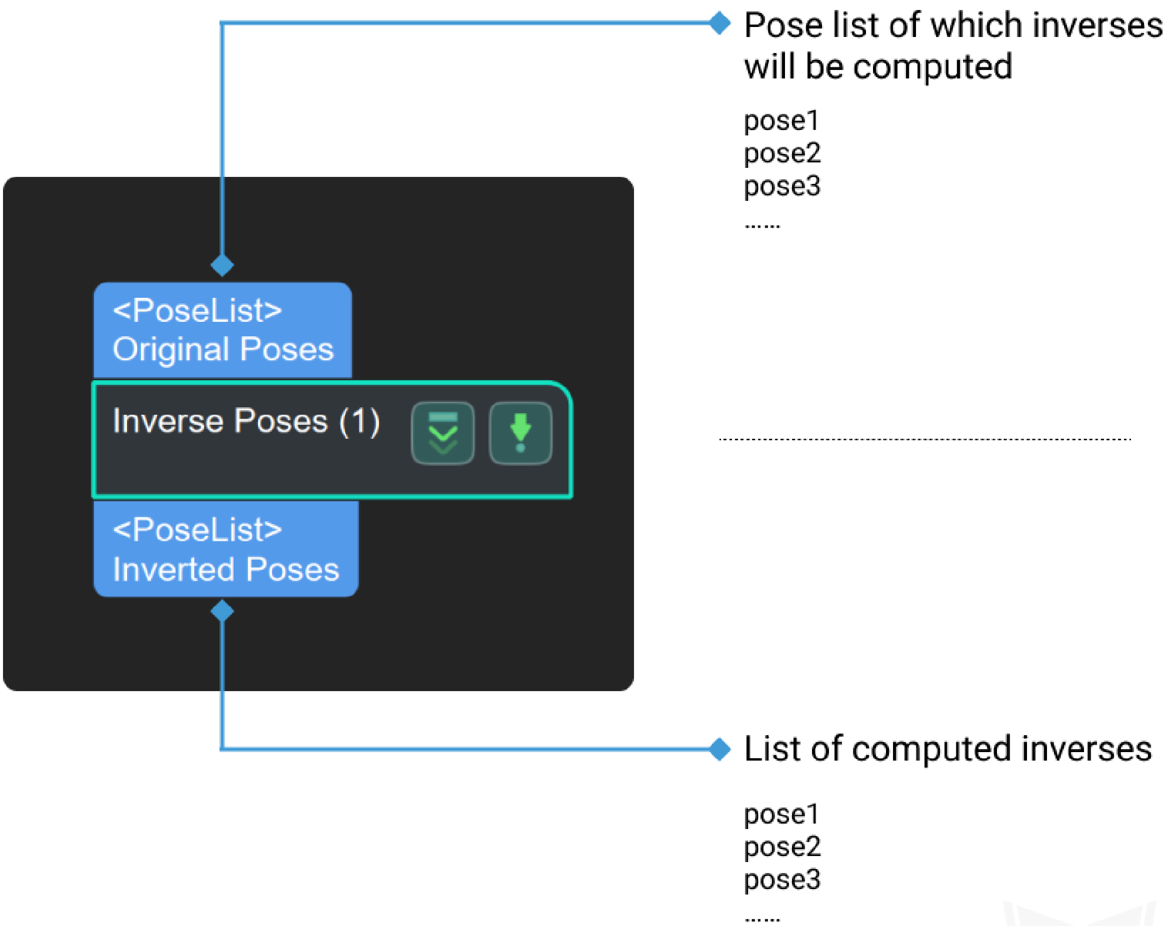
Function

Obtain the inverses of the poses in the input list.

Sample Scenario

A general pose calculation Step. Can be used for transformation between reference frames.

Input and Output



Inverse Quaternions

4.25.4 Pose Composition & Decomposition

Compose Pose from Quaternion and Translation

Compose Quaternion from Axis and Angle

Compose Quaternion from Two Axis (Right-Hand)

Decompose Poses to Quaternions and Translations

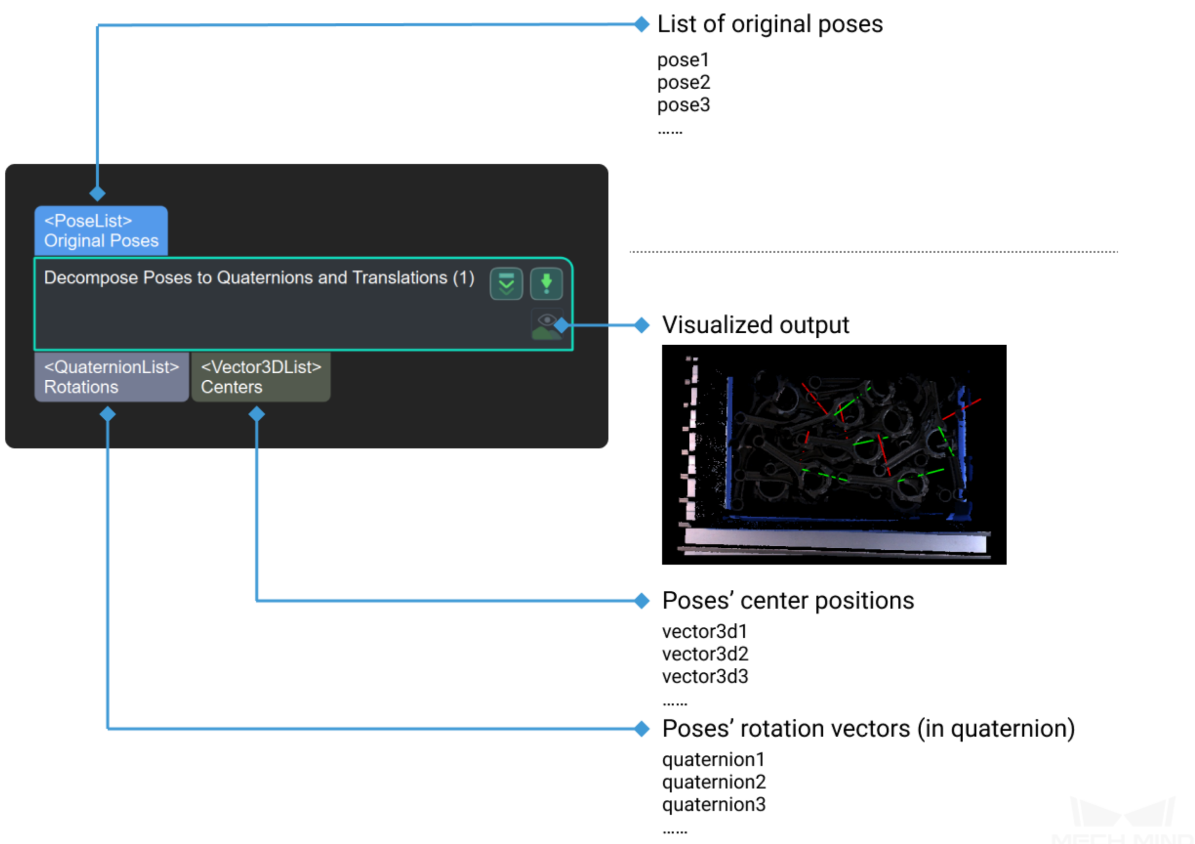
Function

Decompose each input poses into a Quaternion and a translation vector.

Sample 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



Decompose Quaternions to X- Y- Z- Axes

4.25.5 Pose Rotation

Easy Point to Reference Place

Make Poses Point to Reference Place

Rotate Poses around Axis by Angle

Rotate Poses Locally

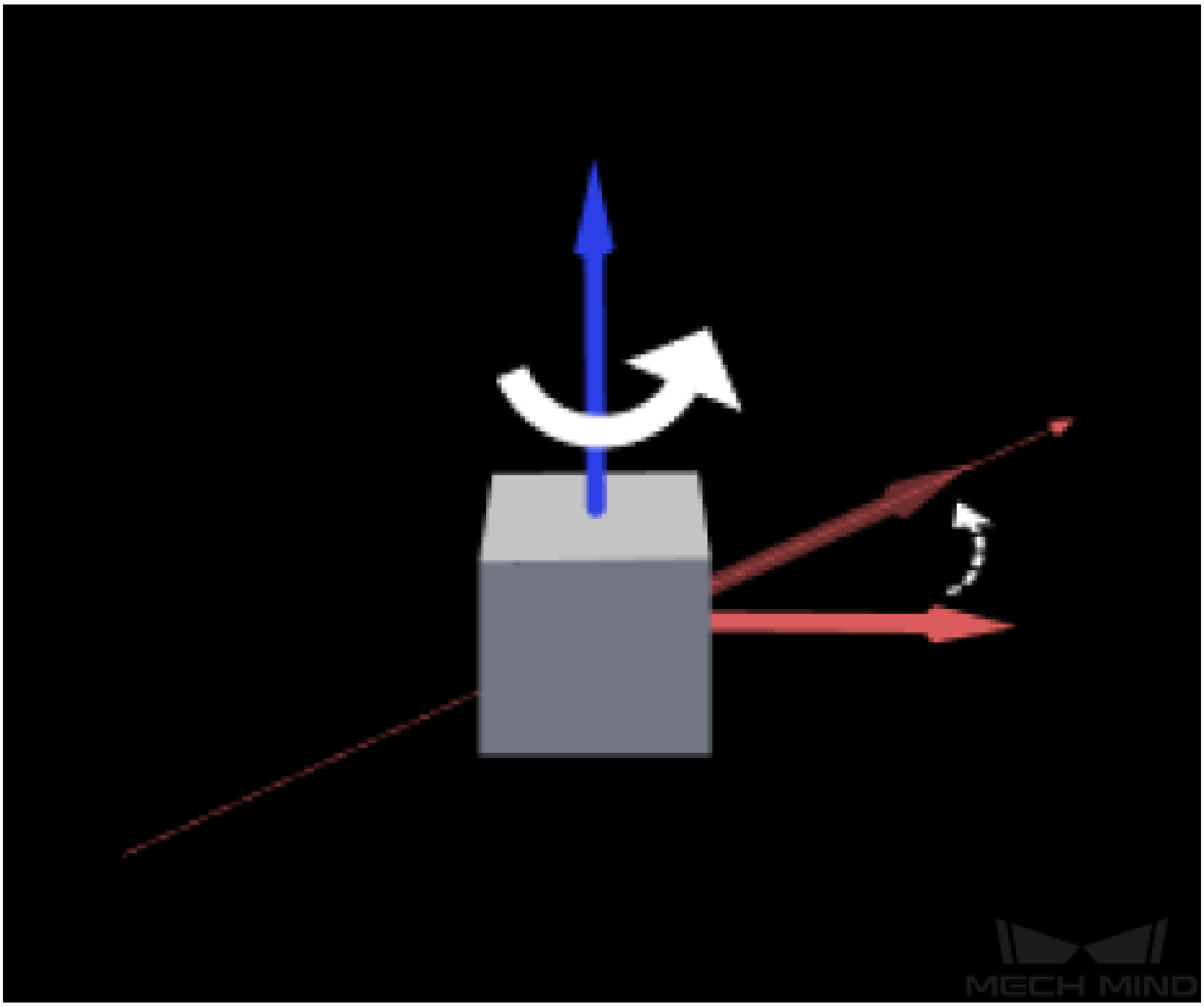
Rotate Poses to Directions Freely

Rotate Poses to Directions with Symmetry Constraint

Rotate Poses to Goal Direction

Function

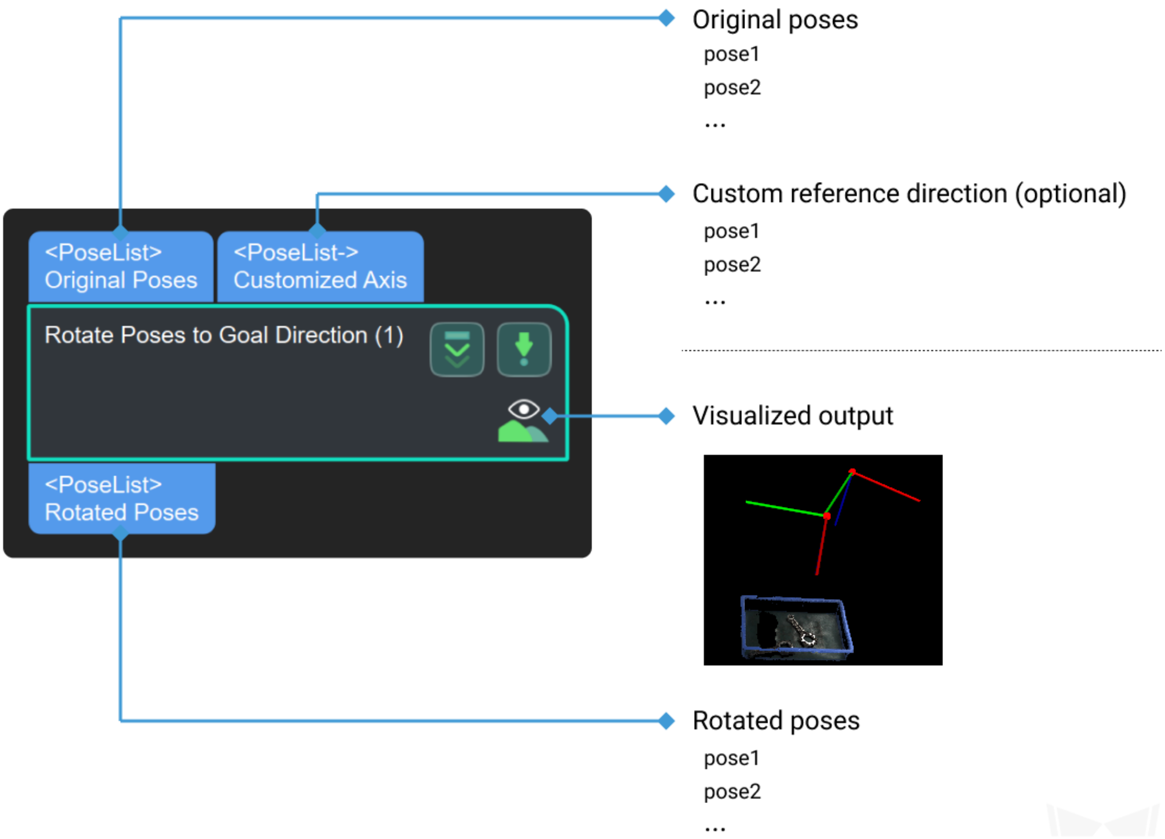
Rotate poses around fixed axes, aligning the specified axes with the reference direction.



Sample Scenario

A general pose adjustment Step. Often used to align the pointing of multiple poses.

Input and Output



4.25.6 Pose Sorting

Those Steps sort poses based on specific rules.

Sort Poses by Input Scores

Function

This Step sorts poses by their scores and outputs the sorted list and its index list.

- Input: the pose list to be sorted, the score list for the pose list. Please note that the two lists' lengths should be the same. 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.
- Output: the sorted pose list, the index list of the sorted list.

Sample Scenario

Scenario 1

For machine tending tasks of metal workpieces, poses can be sorted based on the matching scores, and therefore workpieces with the most accurately calculated poses can be prioritized for picking.

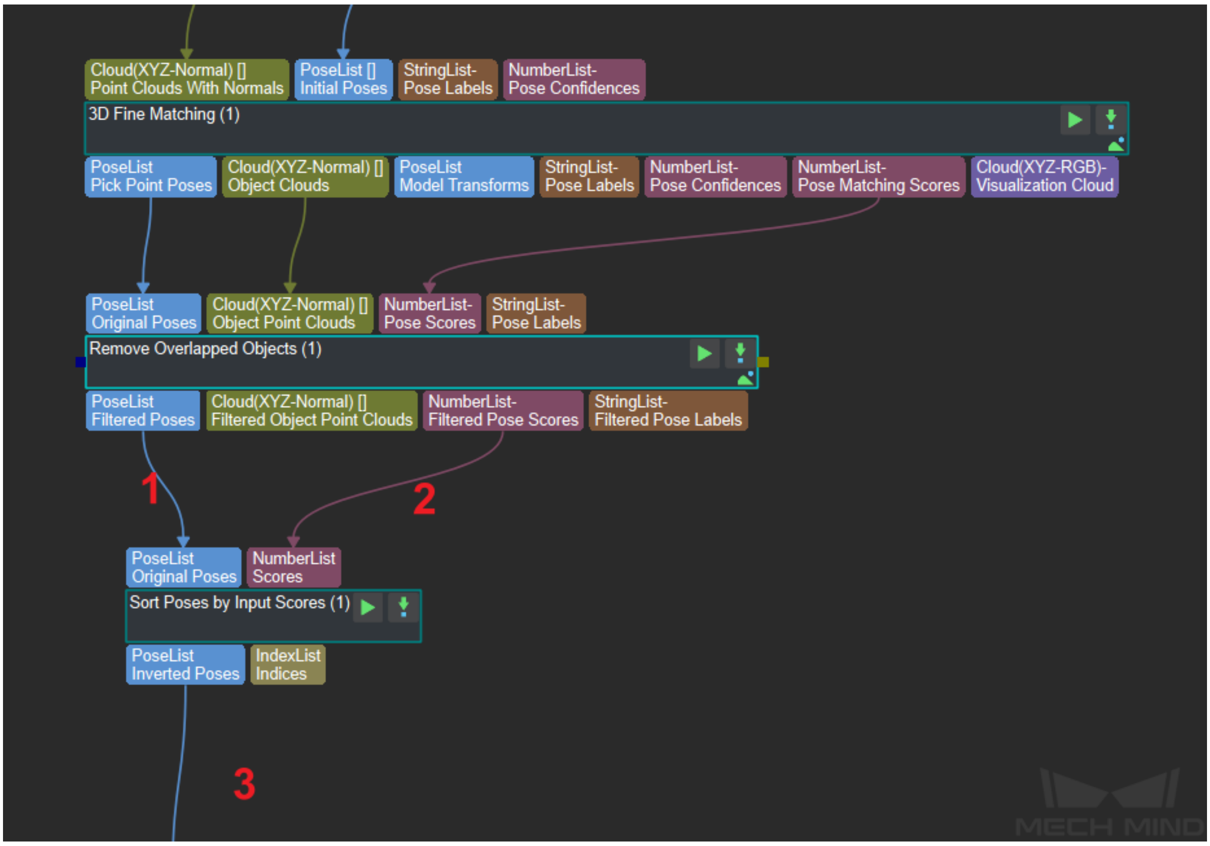


Figure 1. Recommended programming for sorting poses based on matching scores

1. Pose list to be sorted.
2. Matching score list corresponding to the pose list.
3. Sorted pose list.

Scenario 2

Sorting poses and associated masks based on the mask areas output from a deep learning Step.

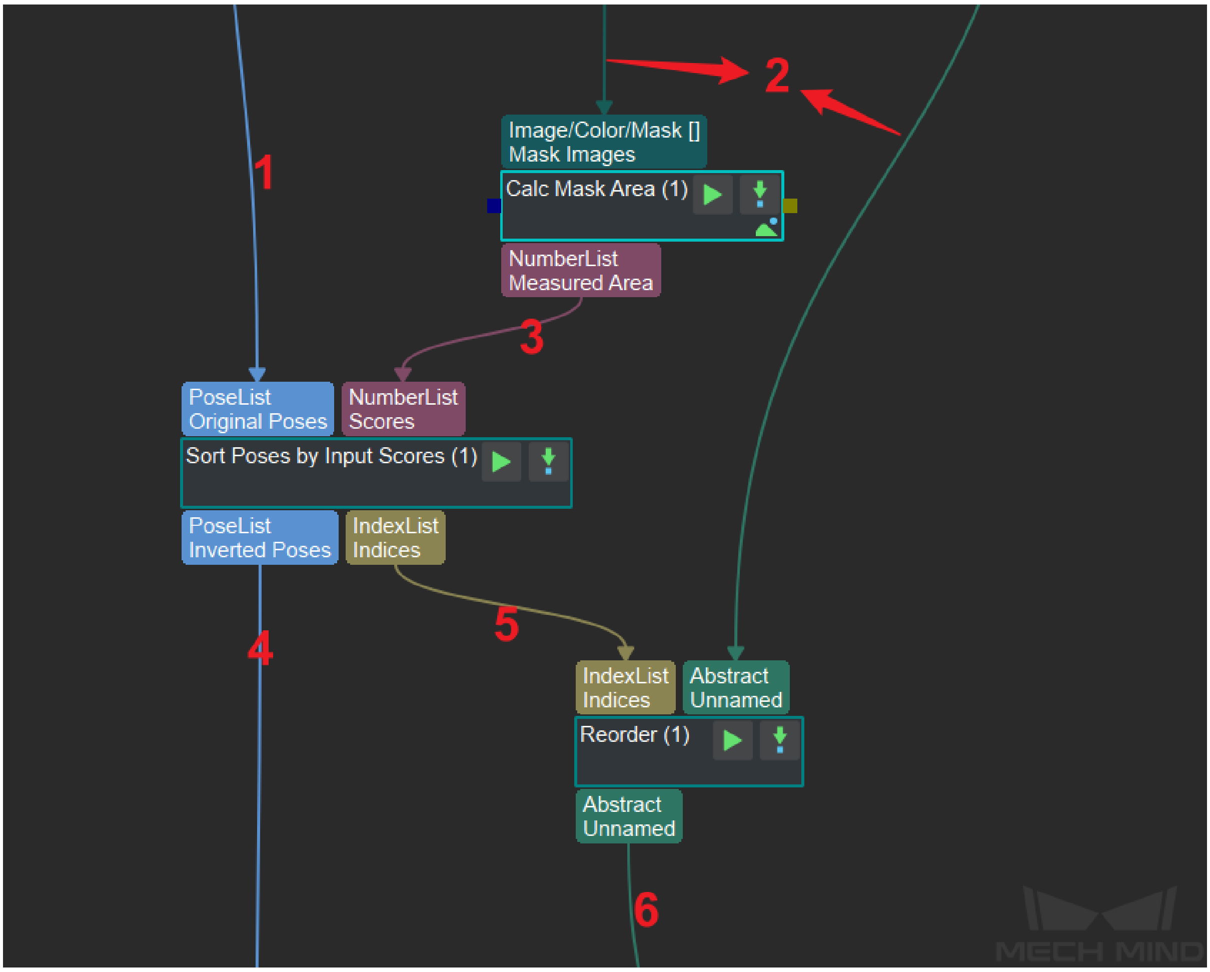


Figure 2. Recommended programming for sorting poses based on mask areas

1. Pose list to be sorted.
2. Mask list to be sorted.
3. Mask area list.
4. Sorted pose list.
5. Index list of the sorted pose list.
6. Reordered mask list based on the sorting result.

Parameters

- **Sort Order**

  Description: Decide whether to sort in ascending or descending order.

  Values:

  - `Ascending`
  - `Descending`

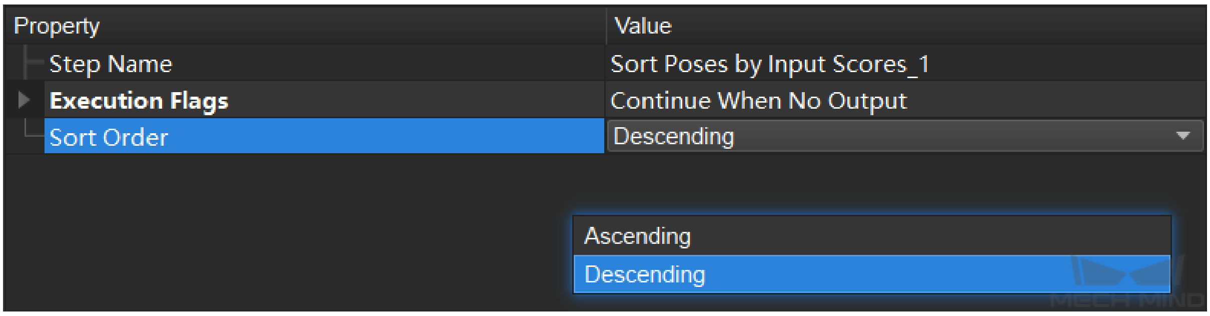


Figure 2. Parameters of **Sort Poses by Input Scores**

Sample Data Flow

Figure 3 is an example of data flow in pose sorting based on matching scores in descending order.

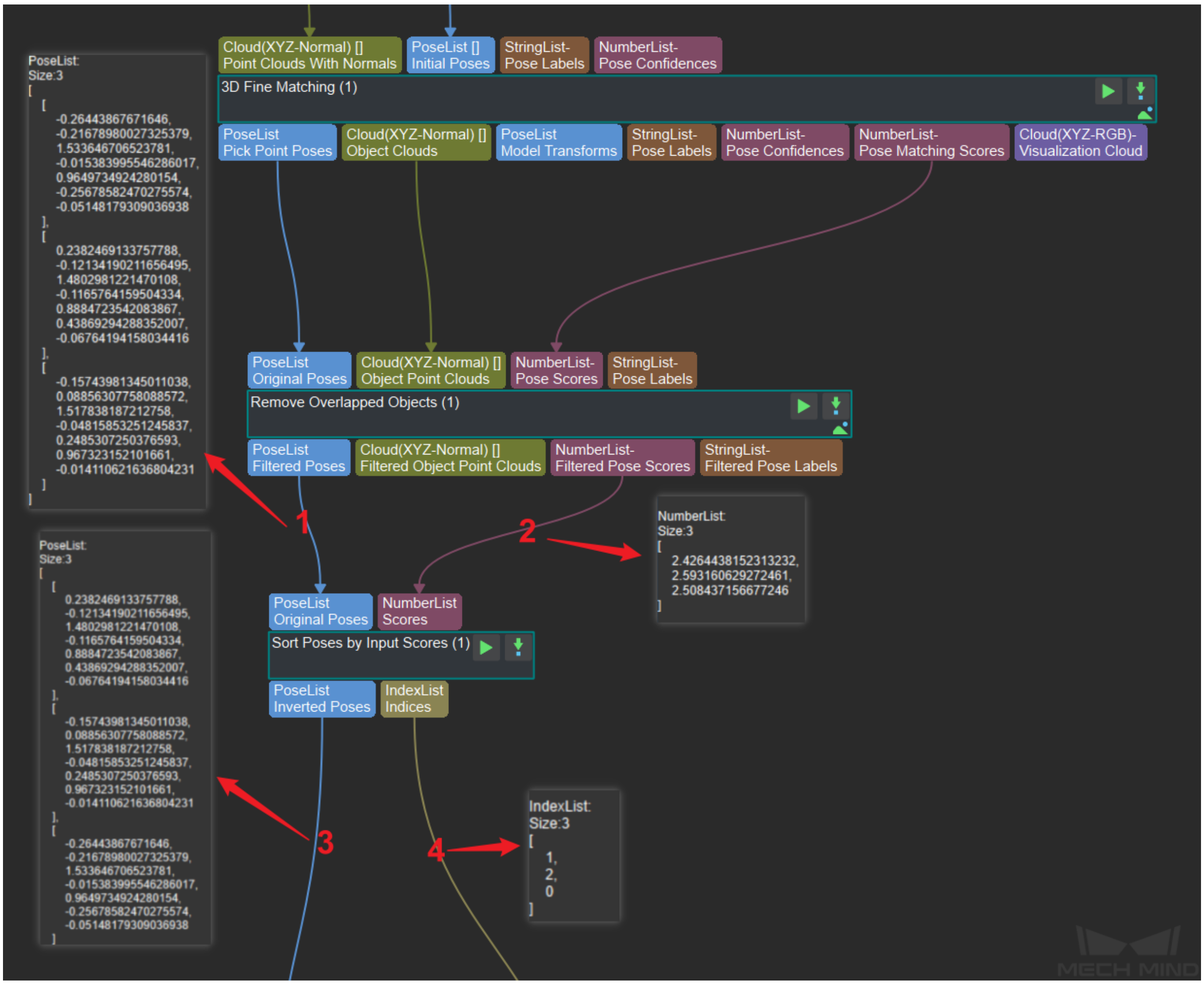


Figure 3. Sample data flow

1. Pose list to be sorted.
2. Matching score list corresponding to the pose list.
3. Sorted pose list.
4. Index list of the sorted pose list.

Comparison of Similar Steps

Sort Poses by Input Scores sorts in the same way as *Sort 3D Poses* when the parameter Sort Method of *Sort 3D Poses* is set to SORT_BY_CONFIDENCE, as shown in *Figure 4*.

Sort Poses by Input Scores outputs the index list of the sorted poses.

Sort Poses by Input Scores can also be used with a **Reorder** Step to reorder lists of data related to poses, such as masks.

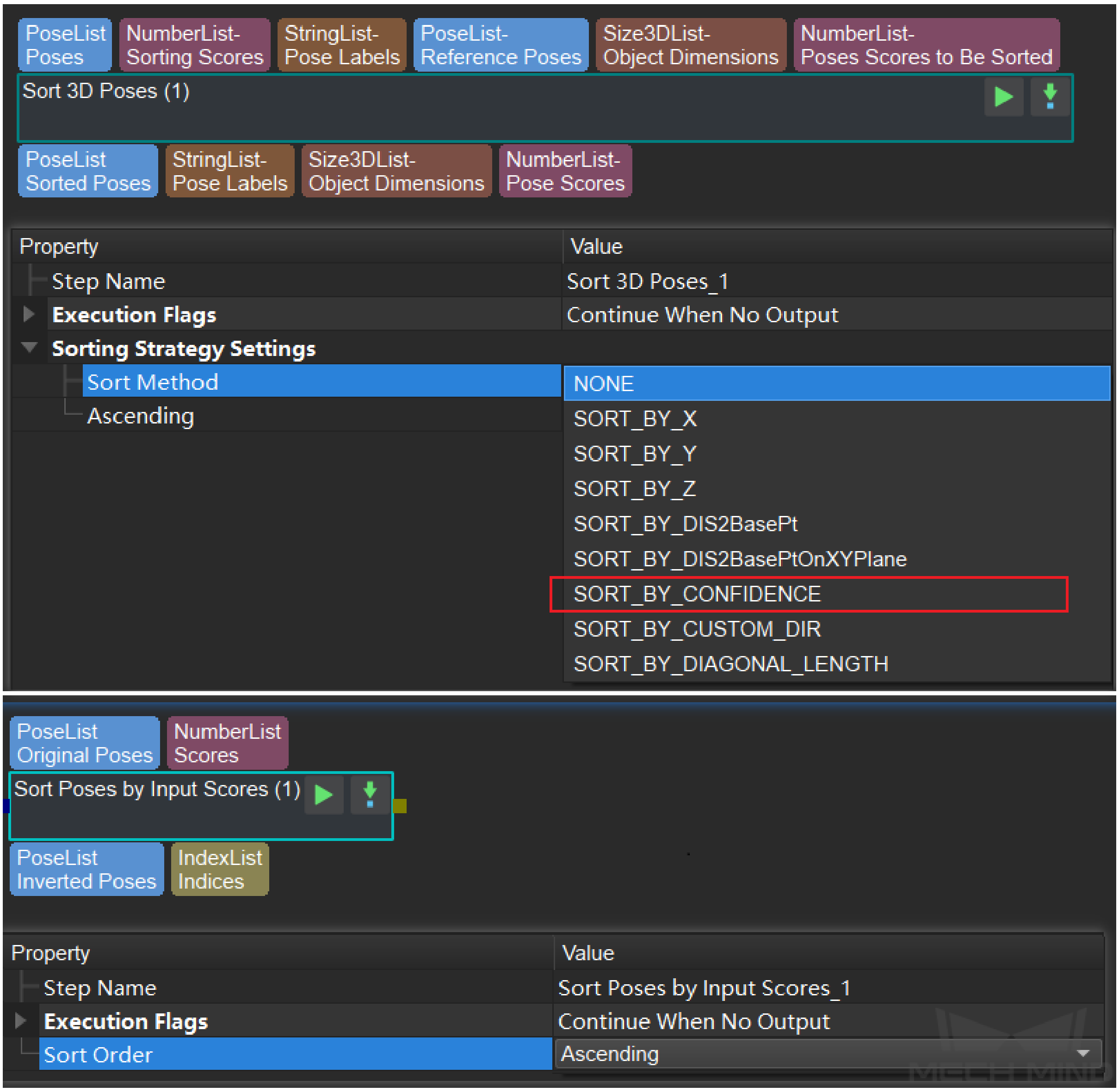


Figure 4. Comparison of similar Steps

Sort Poses by XYZ Values

Functions

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.

- Input: a list of poses to be sorted.
- Output: the sorted list of poses and the index list of the sorted list.

Sample Scenario

This Step can be used when a pose list needs to be sorted based on the poses' X, Y, or Z values, and when other lists of data related to poses, such as masks and labels, need to be reordered based on the sorting result.

When the mask list output from an **Instance Segmentation** step needs to be reordered based on the sorting result, the programming as shown in *Figure 1* can be used.

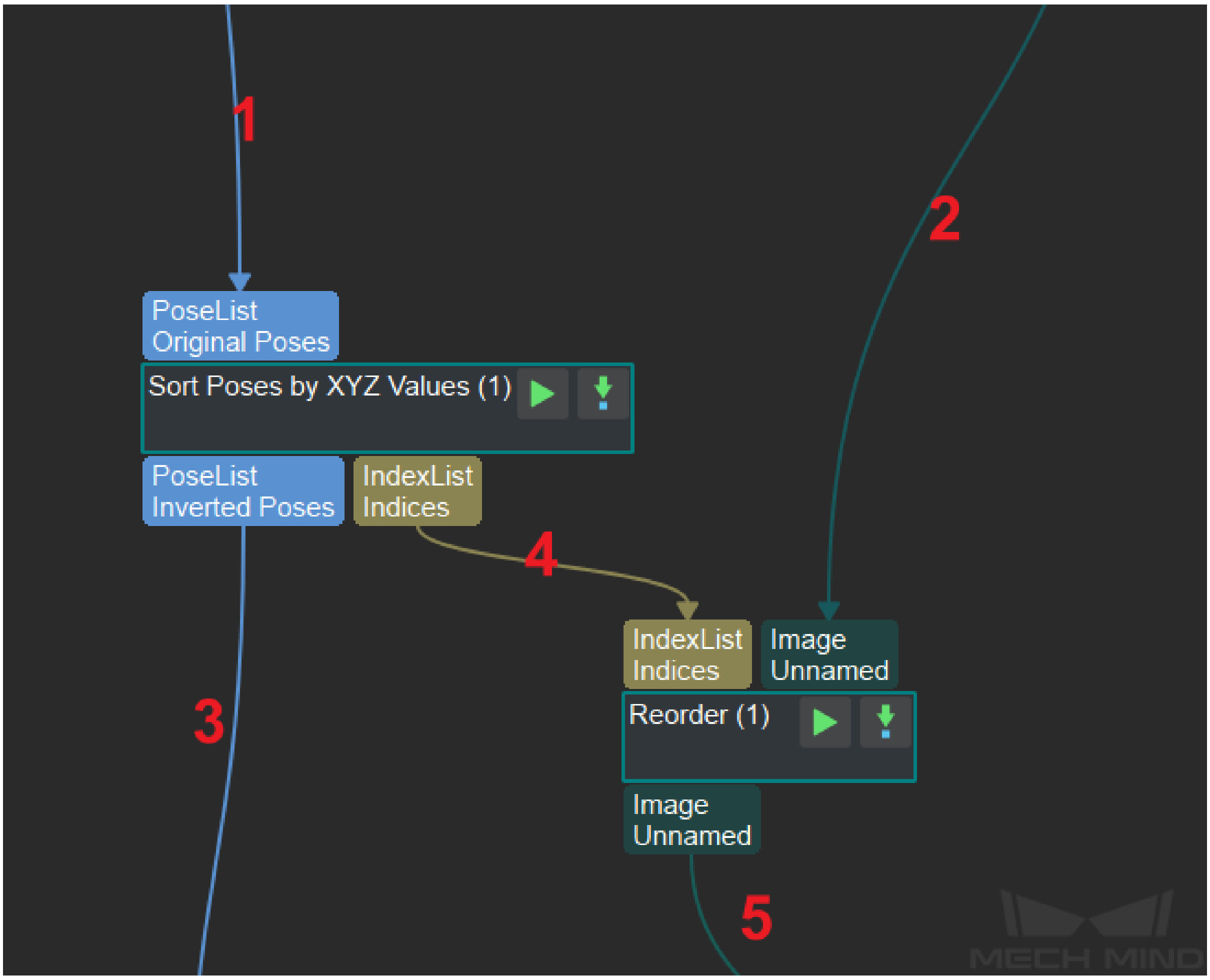


Figure 1. Recommended programming for reordering based on sorting

1. Pose list to be sorted.
2. Mask list to be sorted.
3. Sorted pose list.
4. Index list of sorted pose list.
5. Reordered mask list based on the sorting result.

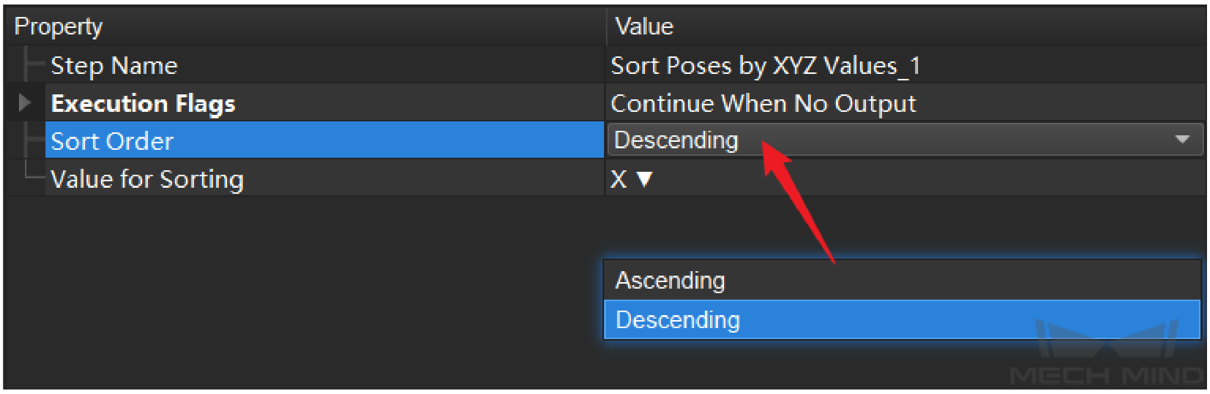
Parameters

Sort Order

Description: Decide whether to sort in ascending or descending order.

Values:

- `Ascending`
- `Descending`

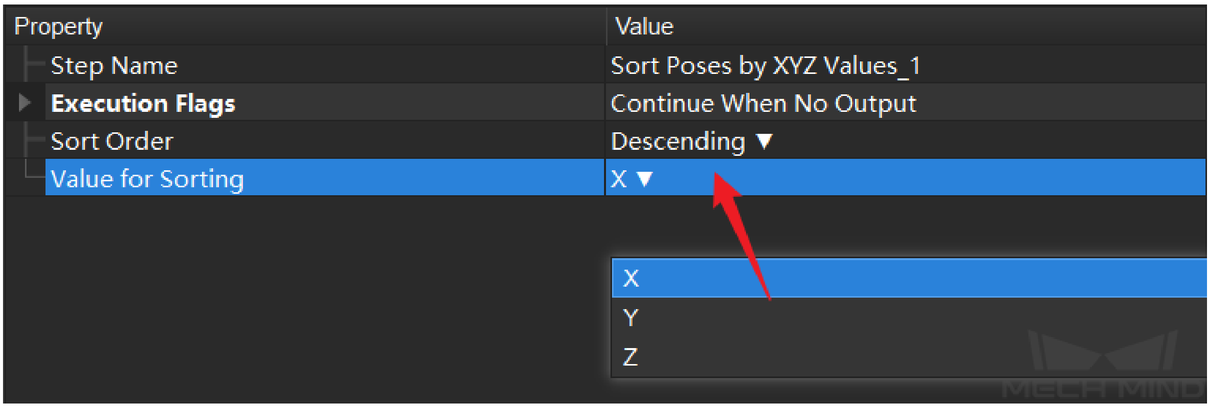


Value for Sorting

Description: Decide whether X, Y, or Z values of the poses is the basis of the sorting.

Values:

- `X`: sort based on the X values of the poses.
- `Y`: sort based on the X values of the poses.
- `Z`: sort based on the X values of the poses.



Parameter Tuning Example

If the **Sort Order** is set to `Ascending` and the **Value for Sorting** is set to `X`, then the sorting is based on X in ascending order. The data flow is as shown in *Figure 2*.

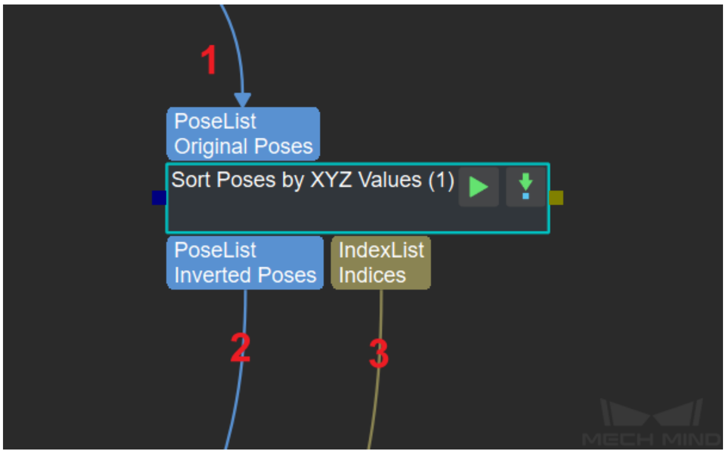


Figure 2. Sorting based on X in ascending order

1. Pose list to be sorted:

```
PoseList:
Size:3
[
    [7, 0, 0, 1, 0, 0, 0],
    [2, 1, 0, 1, 0, 0, 0],
    [5, 2, 1, 1, 0, 0, 0]
]
```

2. Sorted pose list:

```
PoseList
Size:3
[
    [2, 1, 0, 1, 0, 0, 0],
    [5, 2, 1, 1, 0, 0, 0]
    [7, 0, 0, 1, 0, 0, 0],
]
```

3. Index list of the sorted pose list:

```
IndexList:
Size:3
[
    1,
    2,
    0
]
```

Comparison of Similar Steps

Sort Poses by XYZ Values sorts in the same way as *Sort 3D Poses* when the sortMethod parameter of *Sort 3D Poses* is set to SORT_BY_X, SORT_BY_Y, or SORT_BY_Z.

Sort Poses by XYZ Values has an additional output for the index list of the sorted pose list, as shown in *Figure 3*.

The output index list can be used to reorder data lists related to poses, such as masks and labels. The ordering is usually done by **Sort Poses by XYZ Values** together with **Reorder** Step.

In on-site applications, if it suffices to use *Sort 3D Poses*, please use *Sort 3D Poses* for higher project running speed.

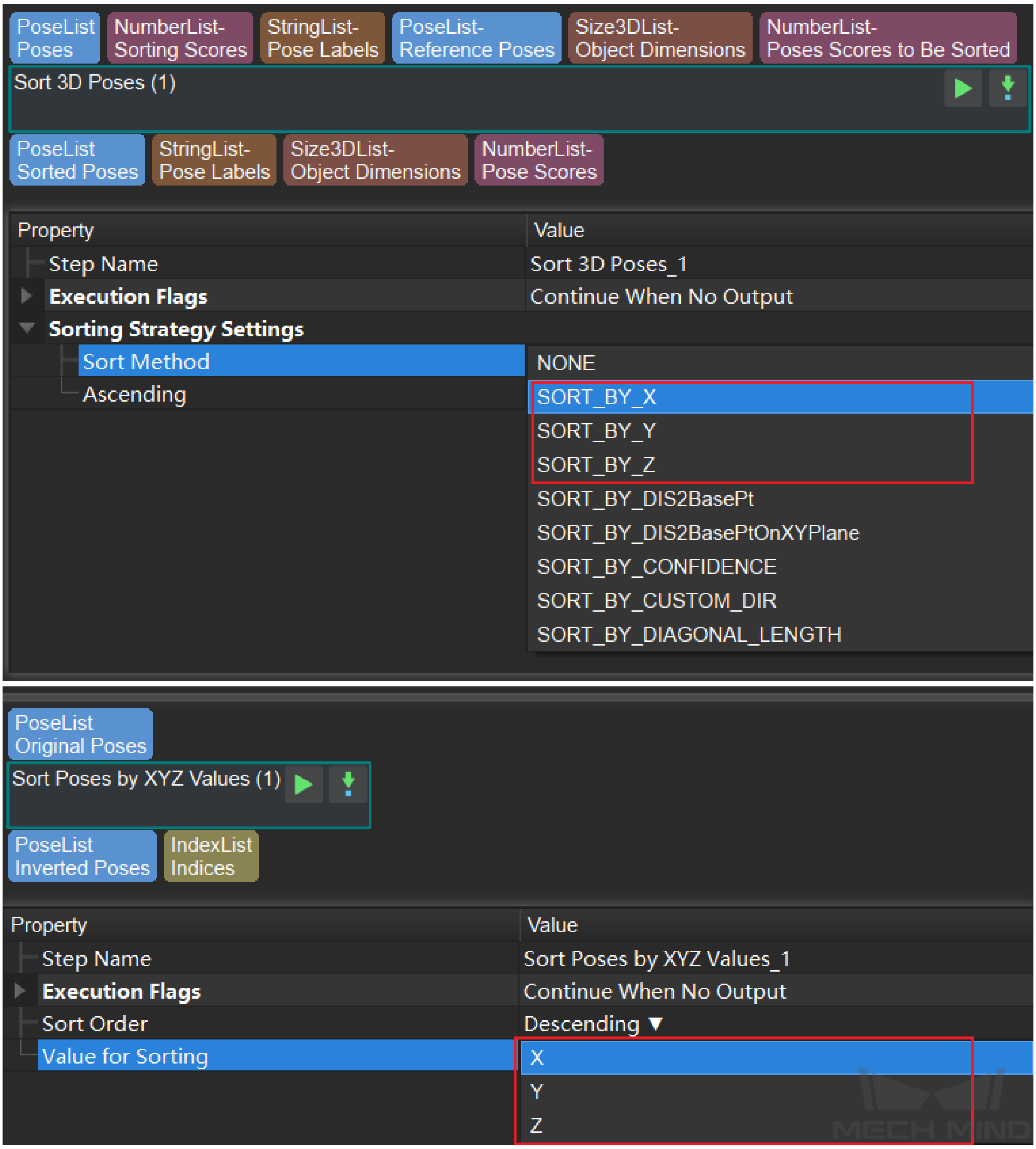


Figure 3. Comparison of similar steps

4.25.7 Pose Transformation

Pose Transformed by Pose2 in Object Coordinate

Pose Transformed by Quaternion in Object Coordinate

4.25.8 Pose Translation

Translate Poses along Customized Direction by Distances

Translate Poses Locally by Distances

Translate Poses by Direction and Distance

Translate Poses by Vector3d

4.25.9 Vector3D Composition & Decomposition

Compose Vector3D from Numbers

Decompose Vector3D to Numbers

4.25.10 Vector3dRelated

Calc Angle between Vector3D

Calc Cross Product of Vector3D

Calc Dot Product of Vector3D

Calc Length of Vector3D

Calc Normalized Vector3D

Vector3D Arithmetic Operations

4.25.11 Pose Adjustment Collection

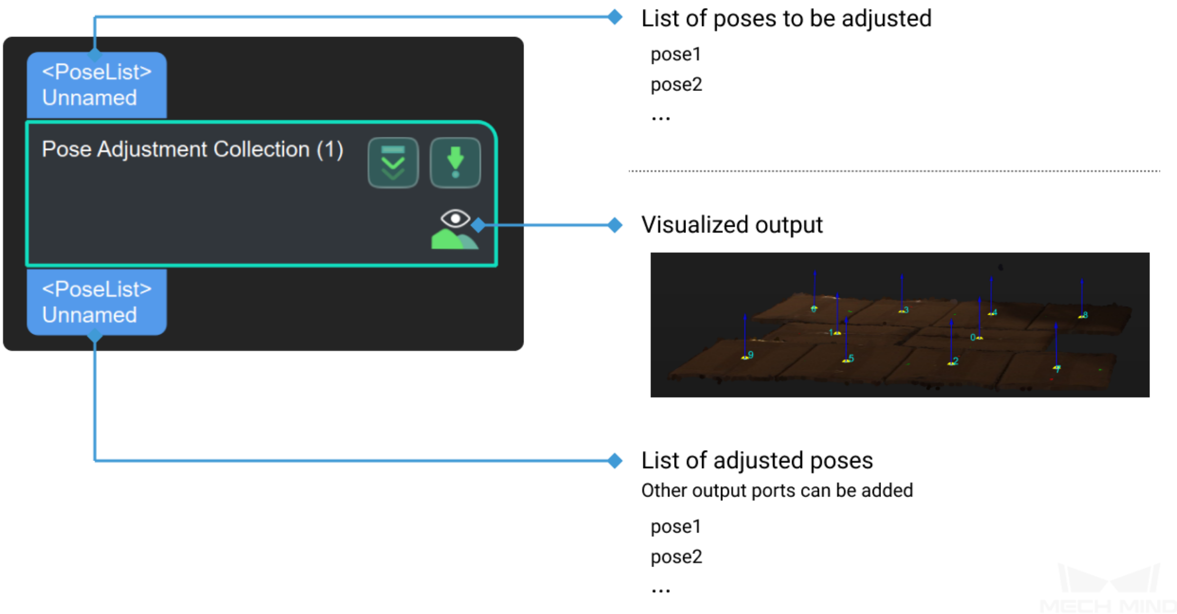
Function

Transform and adjust poses. A separate pose editing interface can be opened from this Step to select preset or custom rules to adjust poses.

Sample Scenario

This Step can be used to adjust or transform poses in a variety of scenarios.

Input and Output



4.26 Reader and Saver

This section describes the steps under **Reader and Saver**.

4.26.1 Read

Read 3D ROI Center

Read Images V2

Read Object Dimensions

Function

Read a list of object dimensions.

Sample Scenario

There are two ways to input the object dimensions in this Step. One is inputting from a JSON file containing dimensions of object(s), and the other is inputting directly in the **Step Parameters** panel.

For dynamically setting box dimensions in depalletizing projects, or for the subsequent point cloud classification operations that require object dimensions.

Input and Output

Parameters

Read Settings

Read Size from Properties

Default Value: True

List of Values: True, False

Instruction: When set as **True**, 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 set to **False**, the size information of the object will be read through the parameters in the specified file "boxSizesList.json", as shown in the *Figure1*. 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 meters , 0.3 meters, 0.2 meters respectively.

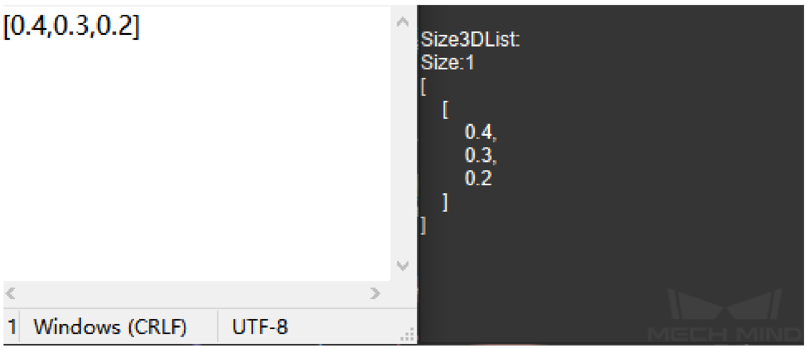


Figure1 Read Size from Properties

Box Size Settings

Auto Generate Box Sizes

Default Value: False

List of Values: True, False

Instruction: This attribute is used for multiple template matching and is not commonly used. By default, one set of size values are output, as shown on the left of *Figure2*. When set as **True** , three sets of size values are output. The original x/y/z value in the attribute is xyz/yzx /zxy output, as shown on the right side of *Figure2*.

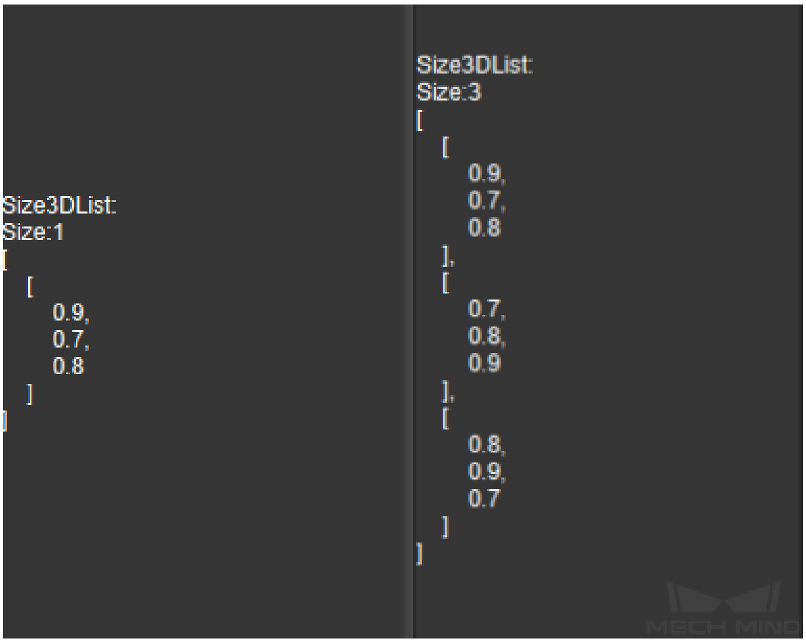


Figure2 Auto Generate Box Sizes

Length on X-axis

Default Value: 0.1

Instruction: Fill in the actual length of the object on the X axis (the unit is m). The valid range is (0, $+\infty$).

Length on Y-axis

Default Value: 0.1

Instruction: Fill in the actual length of the object on the Y axis (the unit is m). The valid range is (0, $+\infty$).

Length on Z-axis

Default Value: 0.1

Instruction: Fill in the actual length of the object on the Z axis (the unit is m). The valid range is (0, $+\infty$).

Read Point Cloud V2

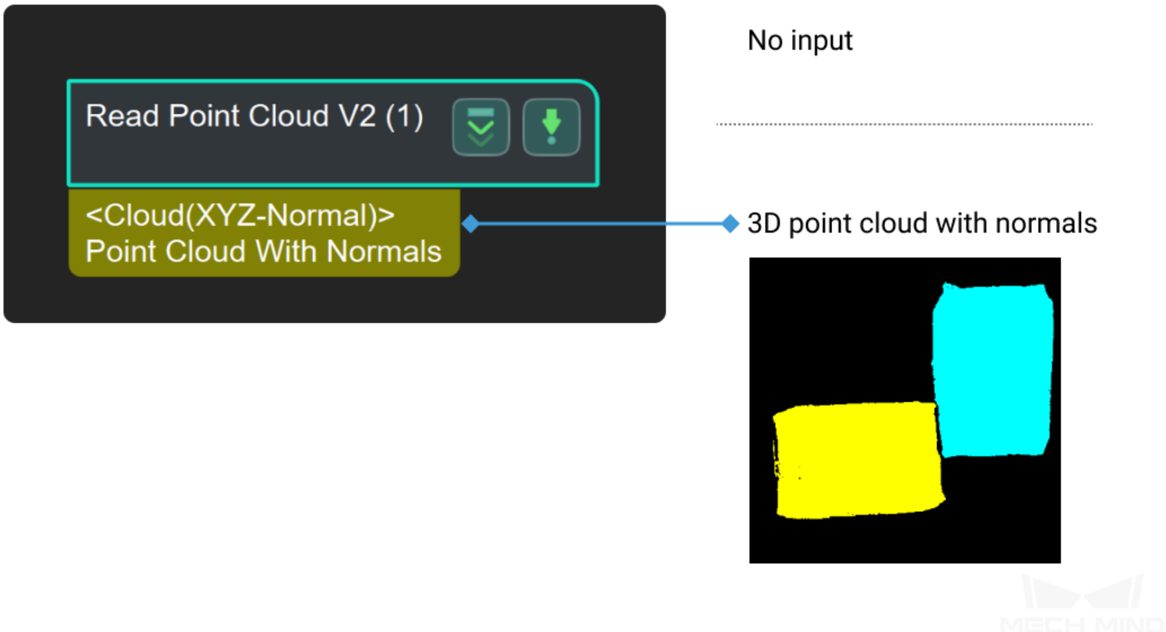
Function

Read point cloud from a file or files in a folder.

Sample Scenario

A general point cloud reading Step. Read the point cloud in PLY/PCD format from a specified path.

Input and Output



Read Poses from File

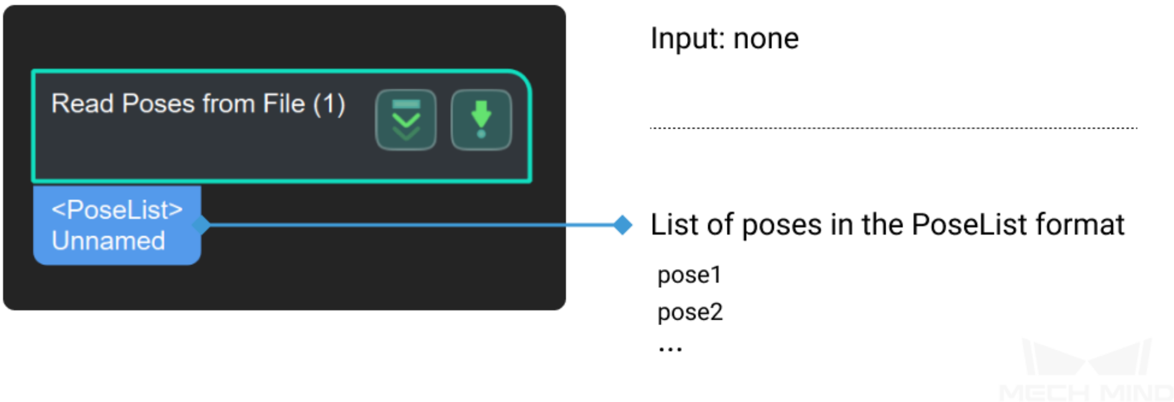
Function

Read a pose list from a local file.

Sample 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

Pose Vec File default poses.json file

Read STL

4.26.2 Save

Save Images

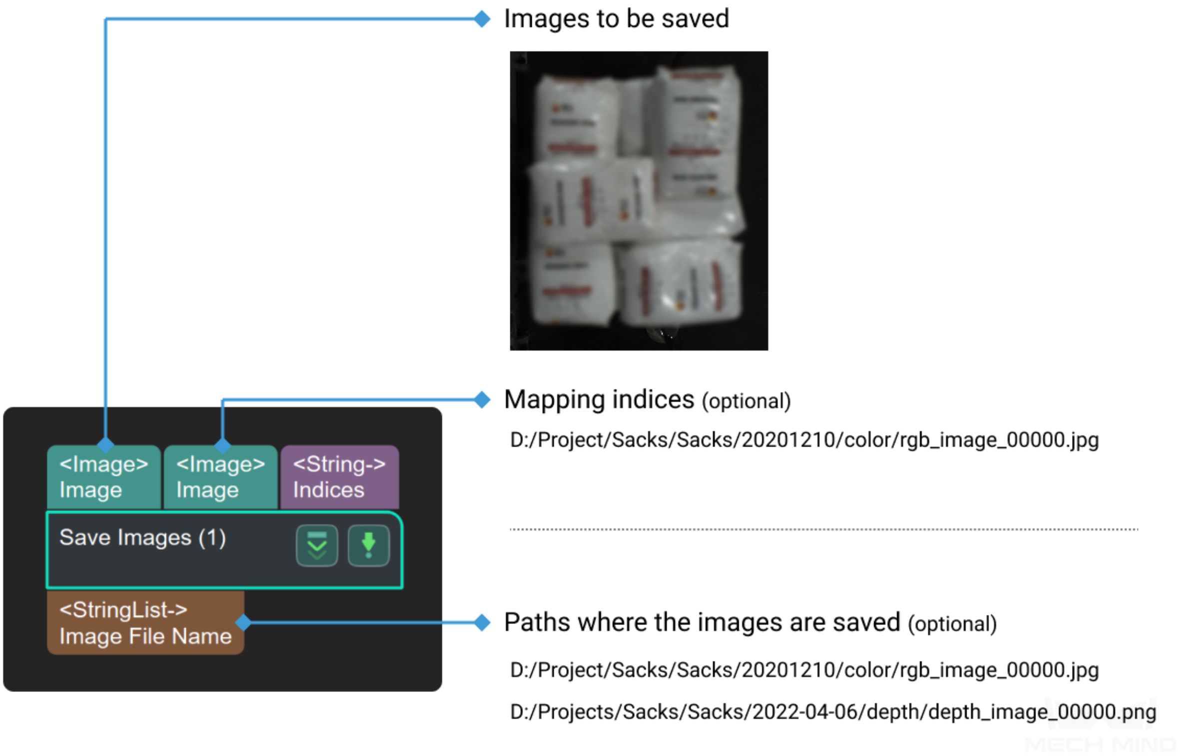
Function

Save the color images and depth maps to a specified path.

Sample 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



Parameters

dir setting

saveDir

Instruction: Set the path to save images.

createSubDirByDate

Default Value: True
List of Values: True, False
Instruction: Whether to create a subfolder named with the date.

saveCameraParas

Default Value: True
List of Values: True, False
Instruction: Whether to save the camera parameters, it is recommended to use the default value.

saveExpireDays

Default Value: 7
Instruction: The maximum number of days a file can be kept, expired files will be deleted.

targetName

Instruction: Whether to name the target object.

subFoldersJoined

Instruction: Subfolders can save images in different formats. To creat it, use “;” between subfolder names, such as: color; depth.

image index

maximumImgNum

Default Value: 1000

Instruction: The effective range is (1, +∞).

customSavingIndex

Default Value: -1

Instruction: Customize the serial number for saved image.

file name

saveColorImgAsPNG

Default Value: False

List of Values: True, False

Instruction: whether to save the color image in .png format.

fileNamePrefix

Instruction: Whether to add a prefix to the file name.

Save Local Areas around Poses as 3D ROI

Save Result to XML File

Save Results to File

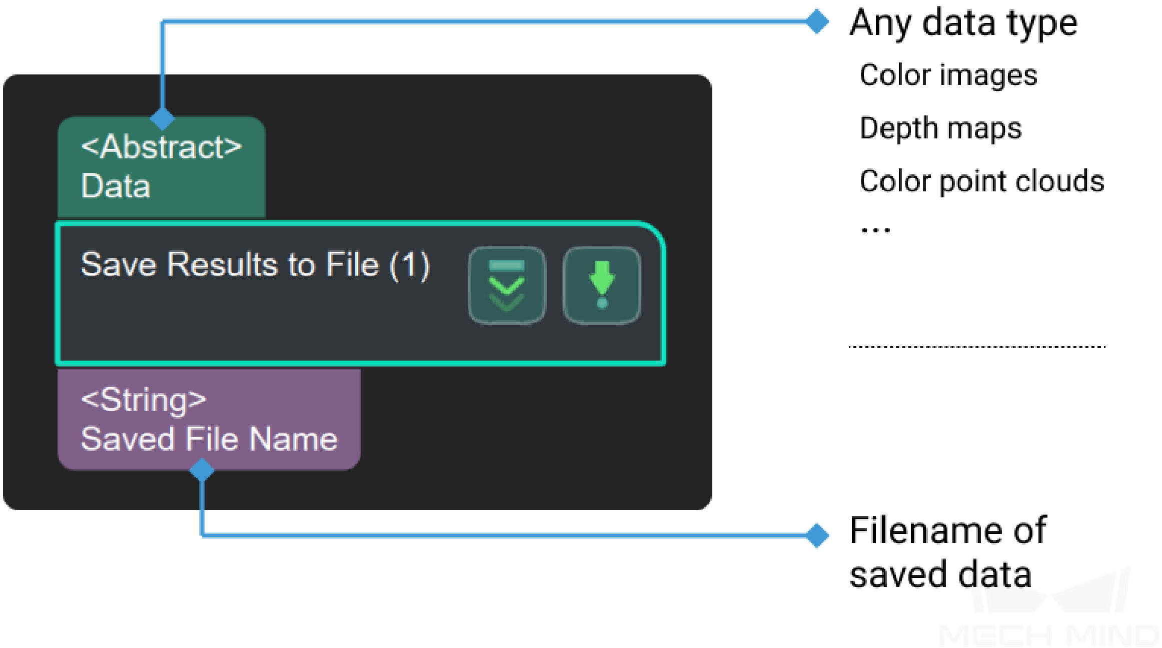
Function

Save project data to local files.

Sample Scenario

This Step saves data of all types, but if you need to save a 2D image, it is recommended to use *Save Images* instead.

Input and Output



Save Step Parameters to File

Function

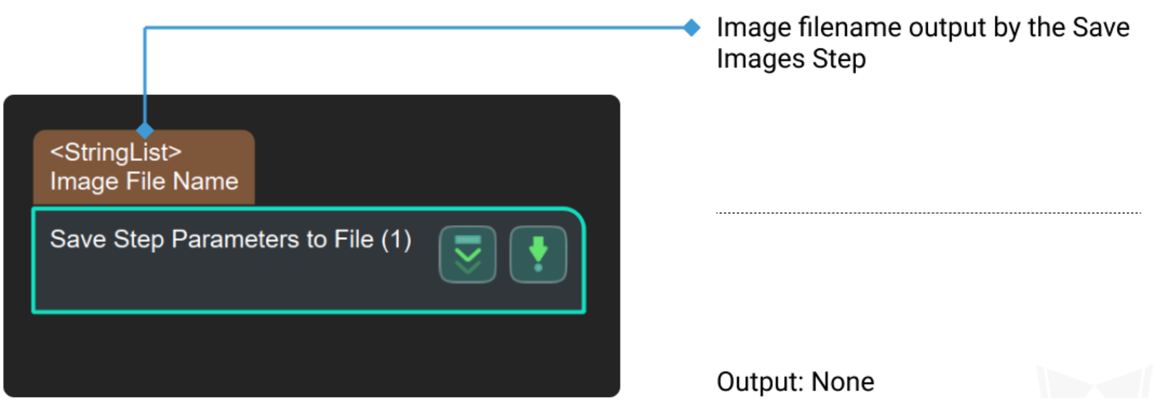
Save the parameters of a specified Step to a local file.

Sample Scenario

Export the parameters of other Steps. This Step must follow the *Save Images* Step.

Instructions: right click in the Graphical Programming Interface, click on **Save Step Parameters** in the menu, and select the Step parameters to be saved in the pop-up window.

Input and Output



Save Trajectory Points

4.27 Sort

In this chapter, steps that are associated with Sort will be introduced.

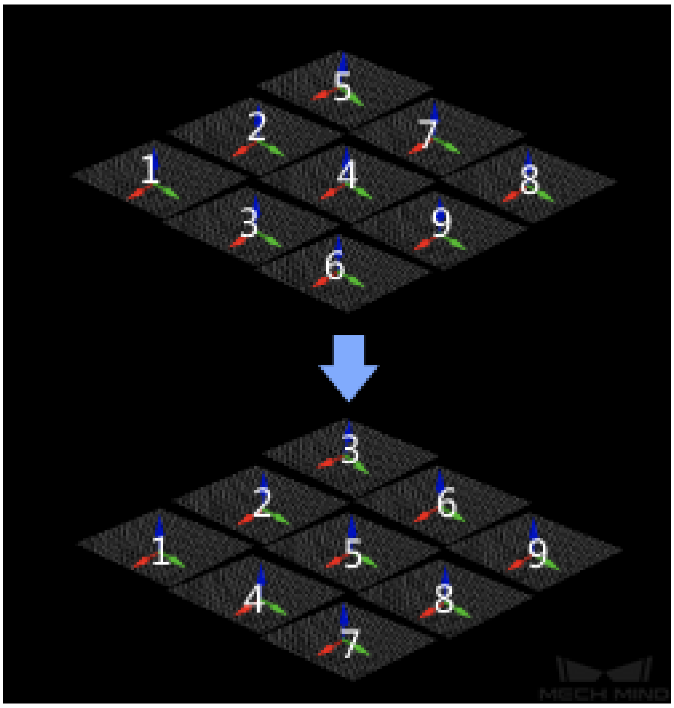
4.27.1 Calc Specified Property of Point Clouds

4.27.2 Sort 2D Poses

4.27.3 Sort 3D Poses

Function

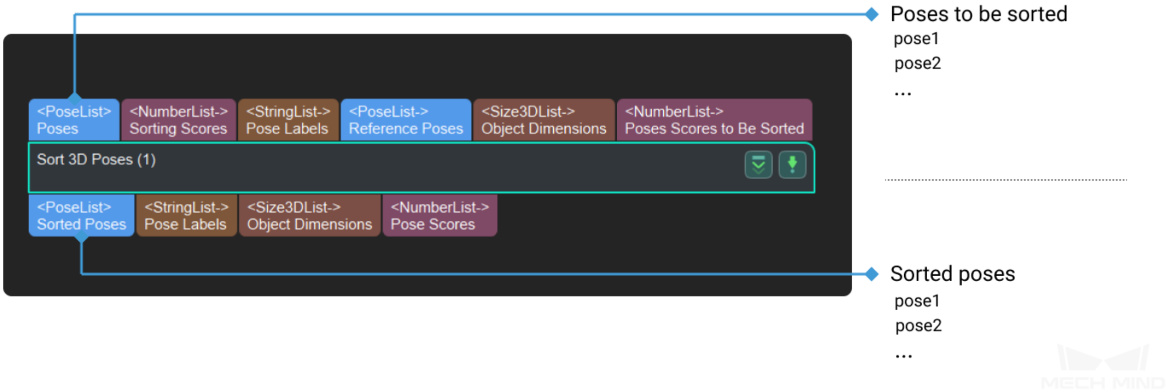
Sort the 3D poses according to specified rules.



Sample Scenario

Sort the poses output from other Steps according to the required rules for subsequent pose processing or robot trajectory planning.

Input and Output



Parameters

sort strategy

sortMethod

Default Value: NONE

List of Values: NONE, SORT_BY_X, SORT_BY_Y, SORT_BY_Z, SORT_BY_DIS2BasePt, SORT_BY_DIS2BasePtOnXYPlane, SORT_BY_CONFIDENCE, SORT_BY_CUSTOM_DIR

Instruction: SORT_BY_X, SORT_BY_Y, and SORT_BY_Z refer to sorting poses according to the length of corresponding componet along X, Y, and Z axes. SORT_BY_DIS2BasePt means sorting poses according to the distance between the pose and the base point. To use this method, setting the base point in the below part is necessary. SORT_BY_DIS2BasePtOnXYPlane is sorting poses according to the projection length of the line connecting the pose to the base point in the XY plane. SORT_BY_CONFIDENCE is sorting pose according to the confidence, and this method requires the second input. SORT_BY_CUSTOM_DIR is customized direction sorting, which need to modify the custom direction in the below.

isAscending

Default Value: True

List of Values: True, False

Instruction: Whether the pose is ranked in ascending order of parameters.

base_point

baseX

Default Value: 0

Instruction: X coordinate value of the base point.

baseY

Default Value: 0

Instruction: Y coordinate value of the base point.

baseZ

Default Value: 0
Instruction: X coordinate value of the base point.

custom_direction

dirX

Default Value: 0
Instruction: X-axis direction vector.

dirY

Default Value: 0
Instruction: Y-axis direction vector.

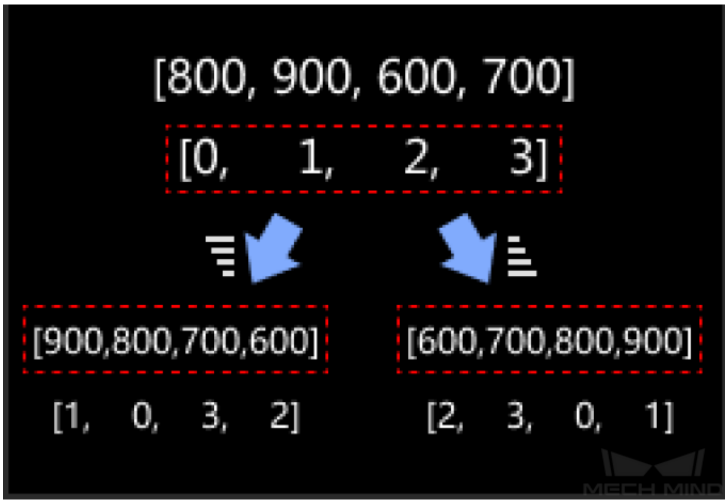
dirZ

Default Value: 1
Instruction: Z-direction direction vector.

4.27.4 Sort and Output Index List

Function

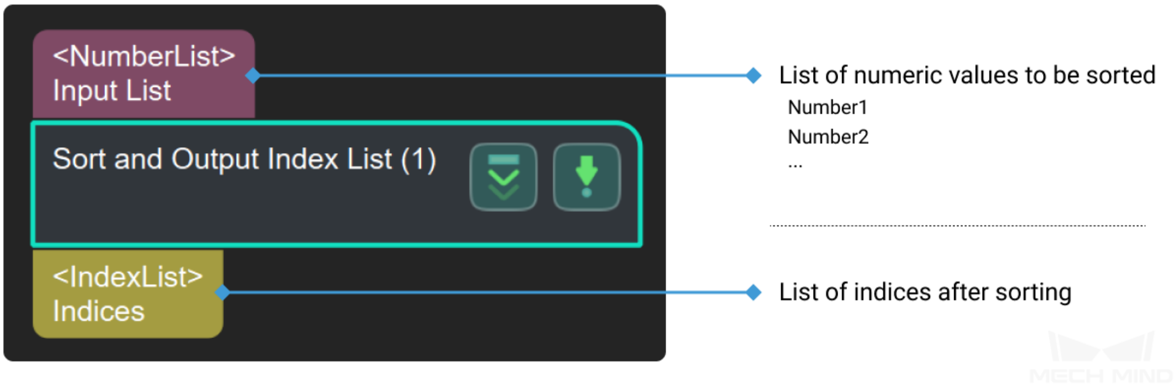
Sort the input list in ascending or descending order, and output the index list according to the sorting.



Sample 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



4.27.5 Sort and Stratify

4.28 Tools

This section describes Steps under Tools.

4.28.1 Image

Copy Images

Evaluate Variation of Depth Image

4.28.2 Pose

Calculate Calib-board Pose

Get Flange Pose List

Measure Result

Poses Repeatability Statistics

Verify Pick Points

4.28.3 Script Engine

Evaluate Results by HDevEngine

Evaluate Results by JavaScript Engine

4.28.4 Step IO

Accept All

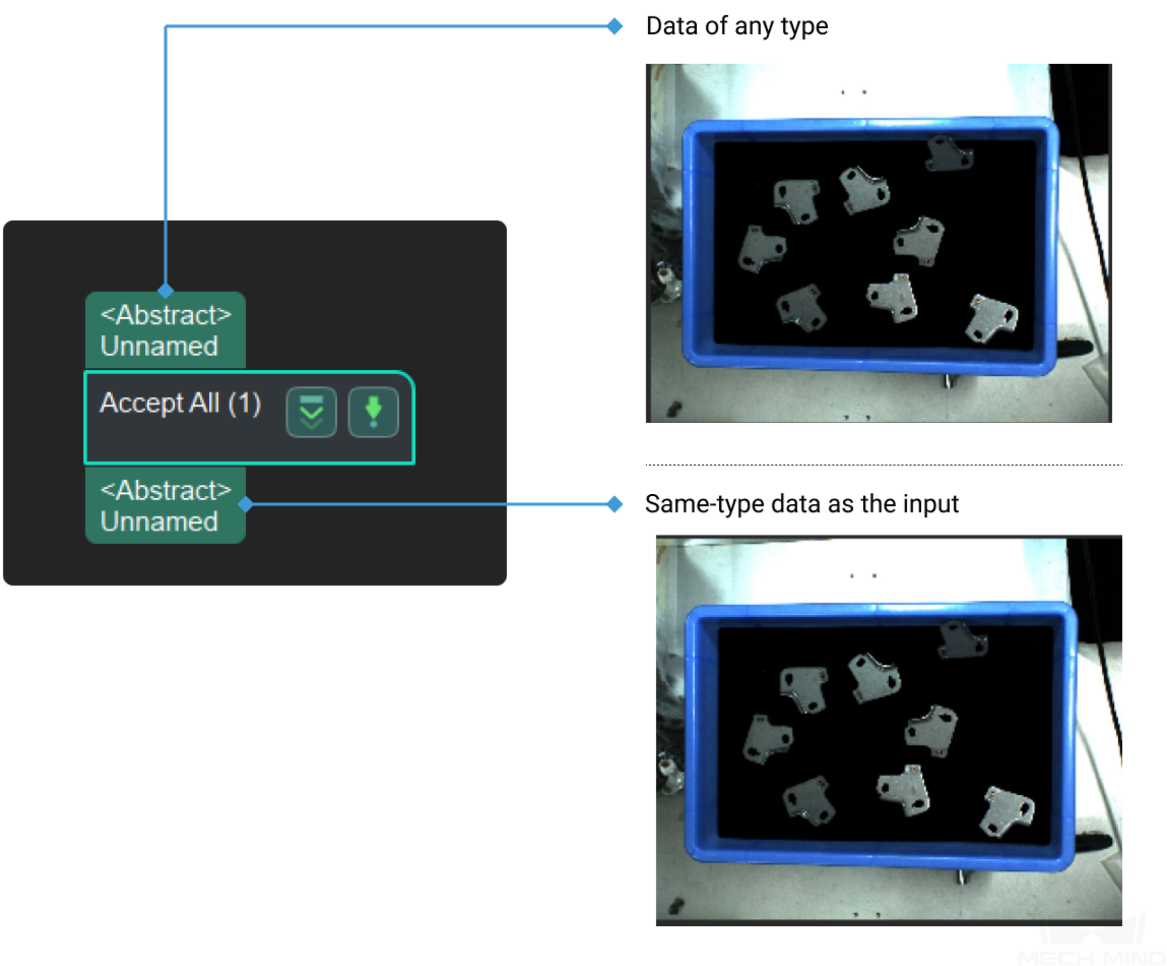
Functions

This Step can accept the output data from any other Steps.

Sample Scenario

This Step is usually used to check the data outputs from certain port during debugging. It can also be used to temporarily save data from certain Step if its execution time is long and therefore the total debugging time can be shorten.

Input and Output



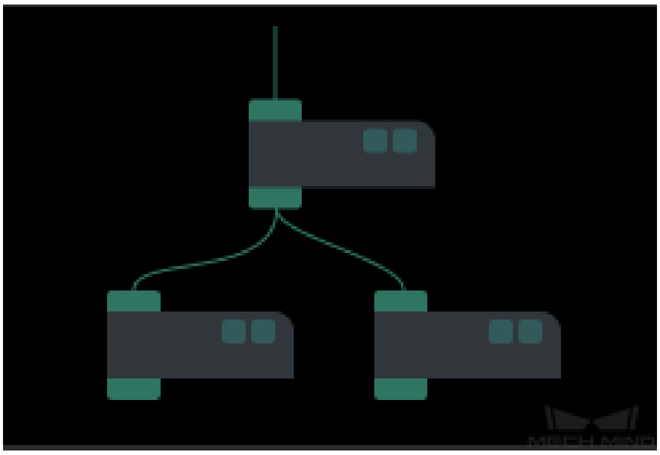
Allocator

Function

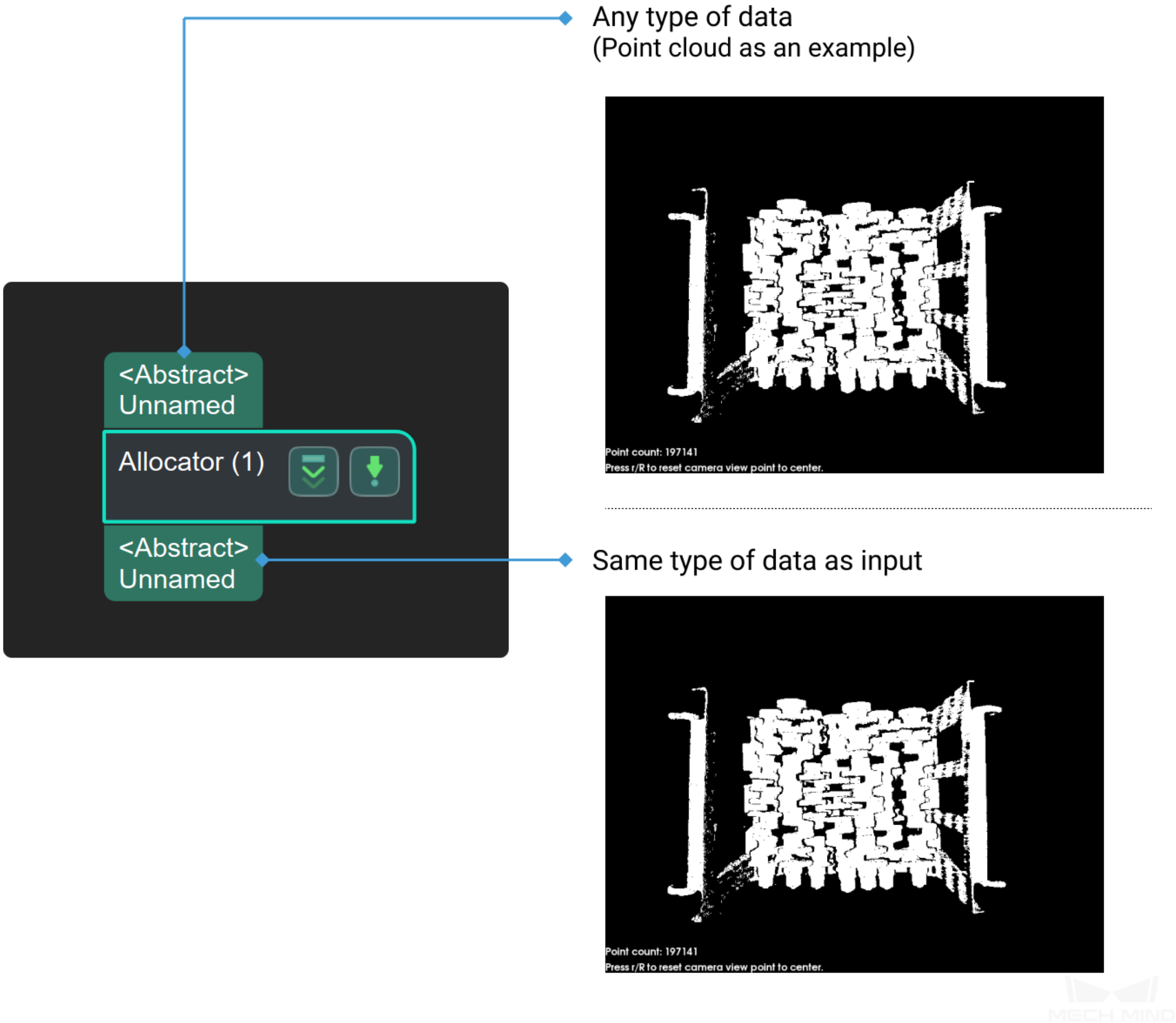
Used to assign input ports for Procedures.

Sample 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



4.28.5 Trigger

Periodic Trigger

Trigger

Trigger Control by Flag

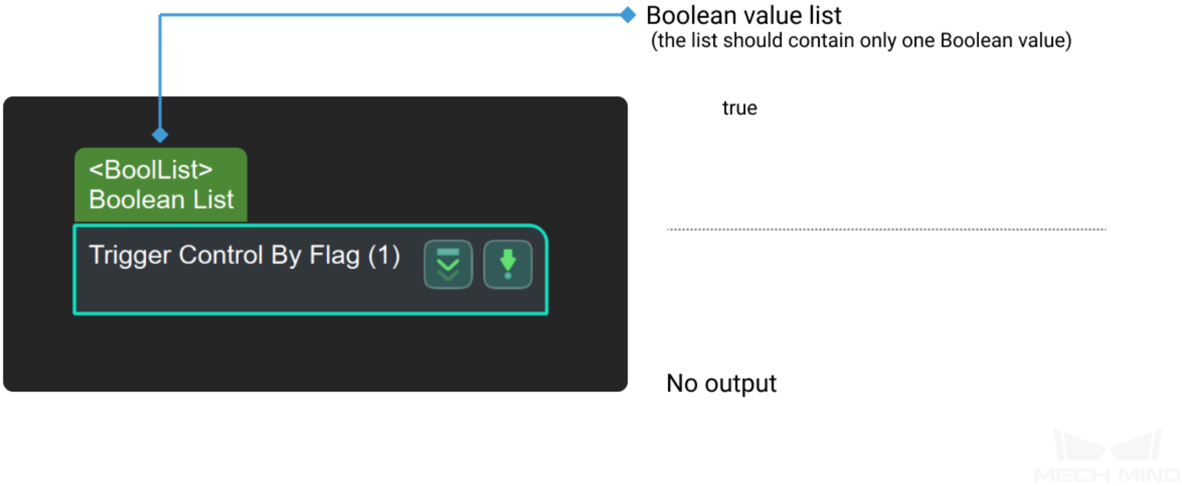
Function

Trigger control flow based on an input Boolean value (True/False) and a specified rule.

Sample 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



4.28.6 Others

This section describes steps under **Others**.

Boolean List Logical Operation

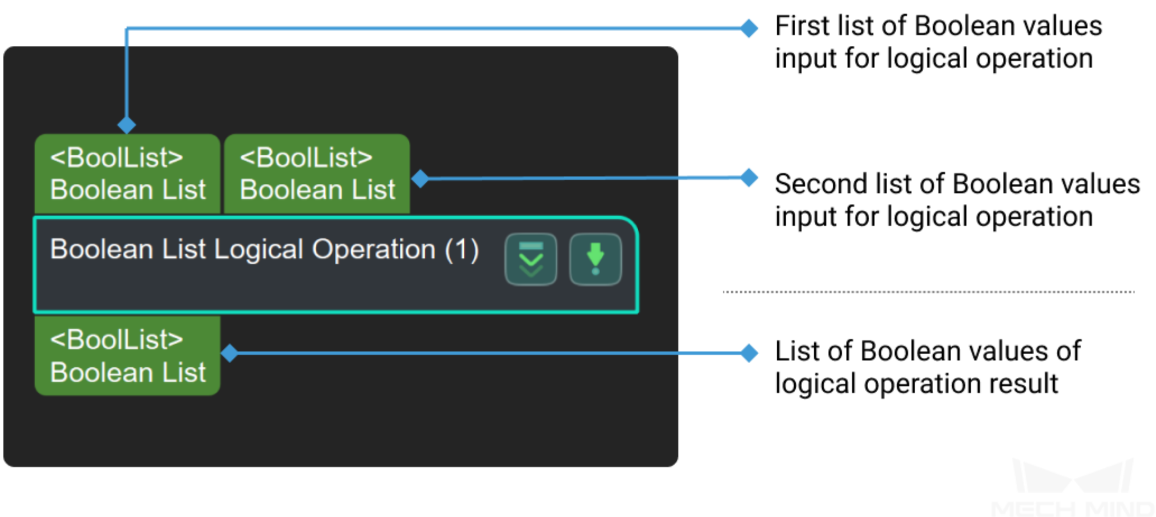
Function

According to the set logical operator, perform a logical operation on the input Boolean value (True/False) lists.

Sample Scenario

A general logic operation Step. No fixed usage scenarios.

Input and Output



Decompose Object Dimensions

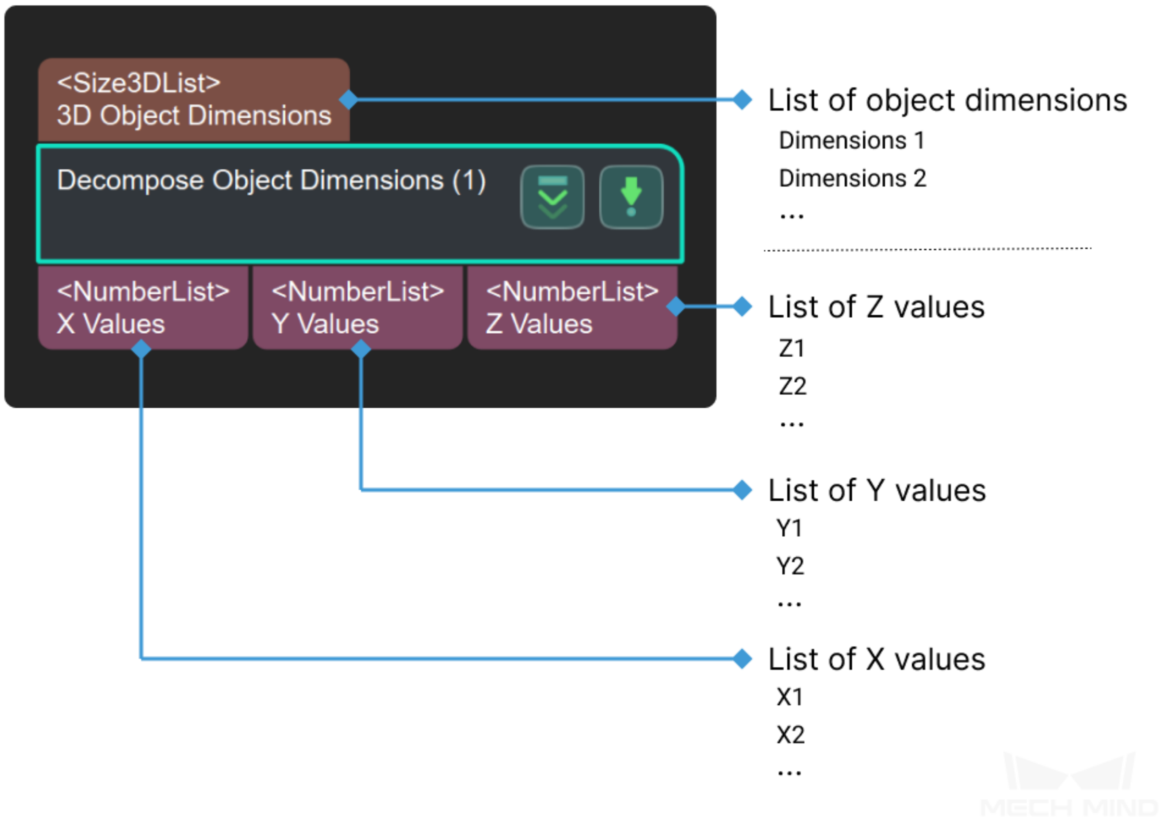
Function

Split the lengths, widths, and heights (X, Y, Z) of the objects in the input list into three separate lists

Sample Scenario

For facilitating subsequent separate processing of object lengths, widths, and heights after obtaining the values.

Input and Output



Dichotomize Values by Threshold

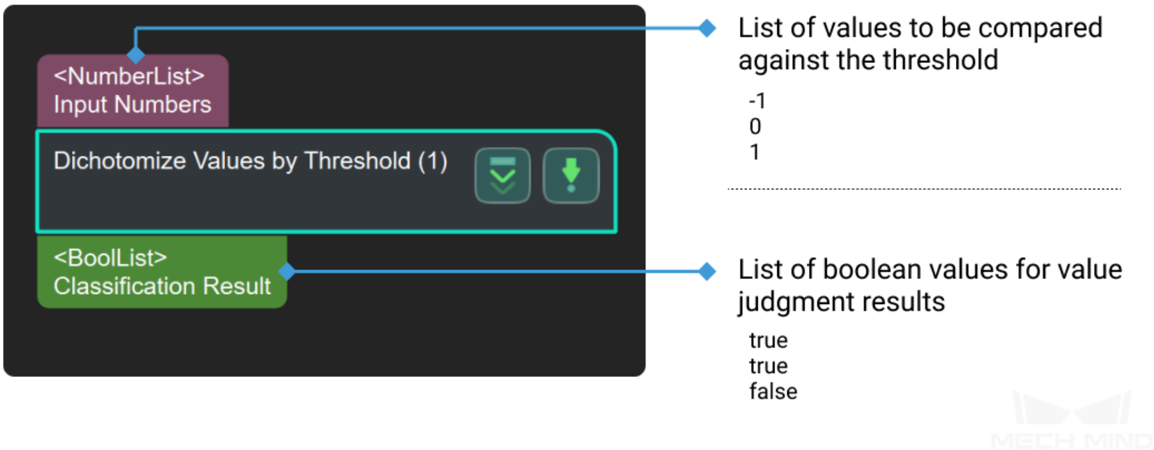
Function

Perform a value judgment on each input list item based on a set threshold.

Sample 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



4.28.7 External Interface

4.29 Trajectory

4.29.1 Adjust Trajectory

Adjust Poses to Obtain Accurate Trajectory

Adjust Trajectory Circular Motion

Calc Midpoint of Specified Side of Rectangle

Insert End Points and Send Motion Params

Smooth Trajectory

4.29.2 Generate Trajectory

Auto Trajectory

Generate Rect Traj

Generate Spiral Traj

Generate Traj by Contour

Generate Trajectory Given Depth Image

Generate Zigzag Traj

4.29.3 Load Trajectory

Load 2D Trajectory

Load Poses in Trajectory and Apply Affine Transform

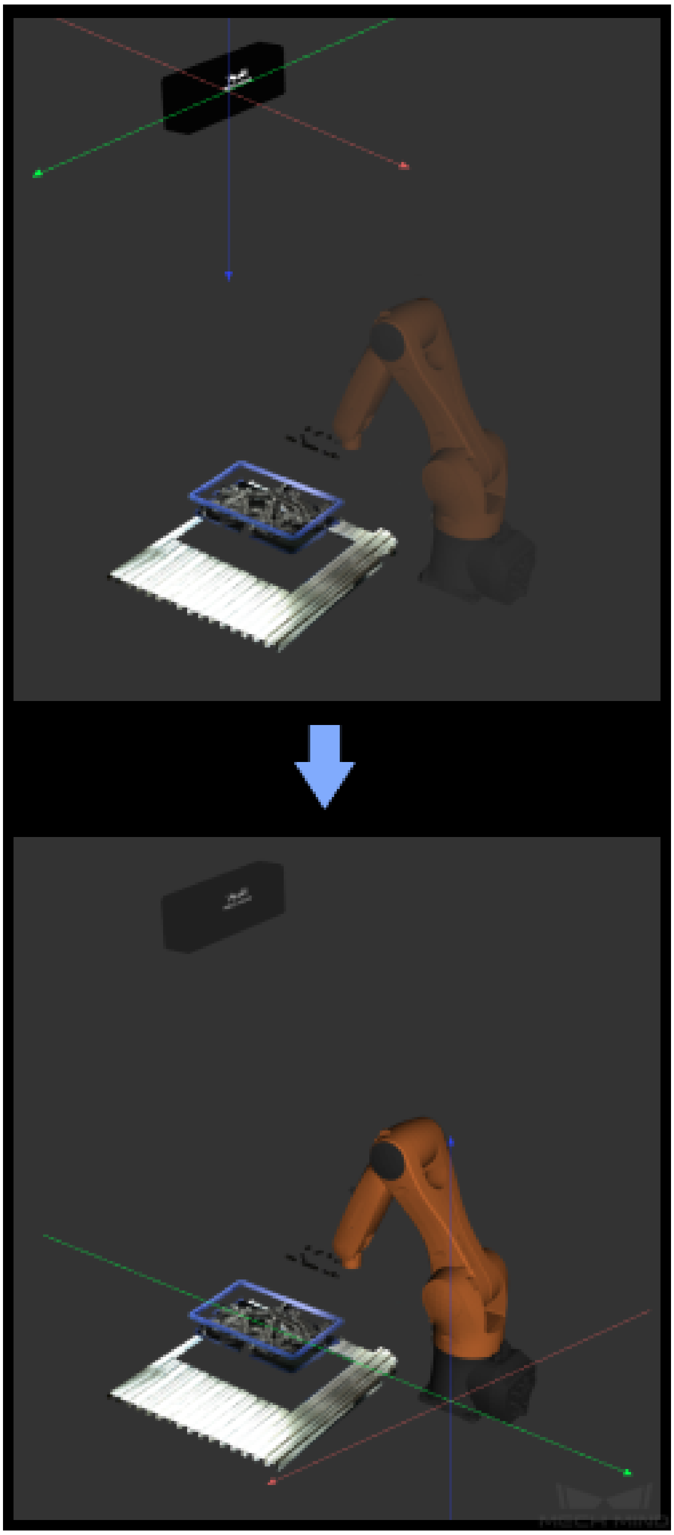
4.30 Transform

In this chapter, steps that are associated with Transform will be introduced.

4.30.1 Transform Point Clouds

Function

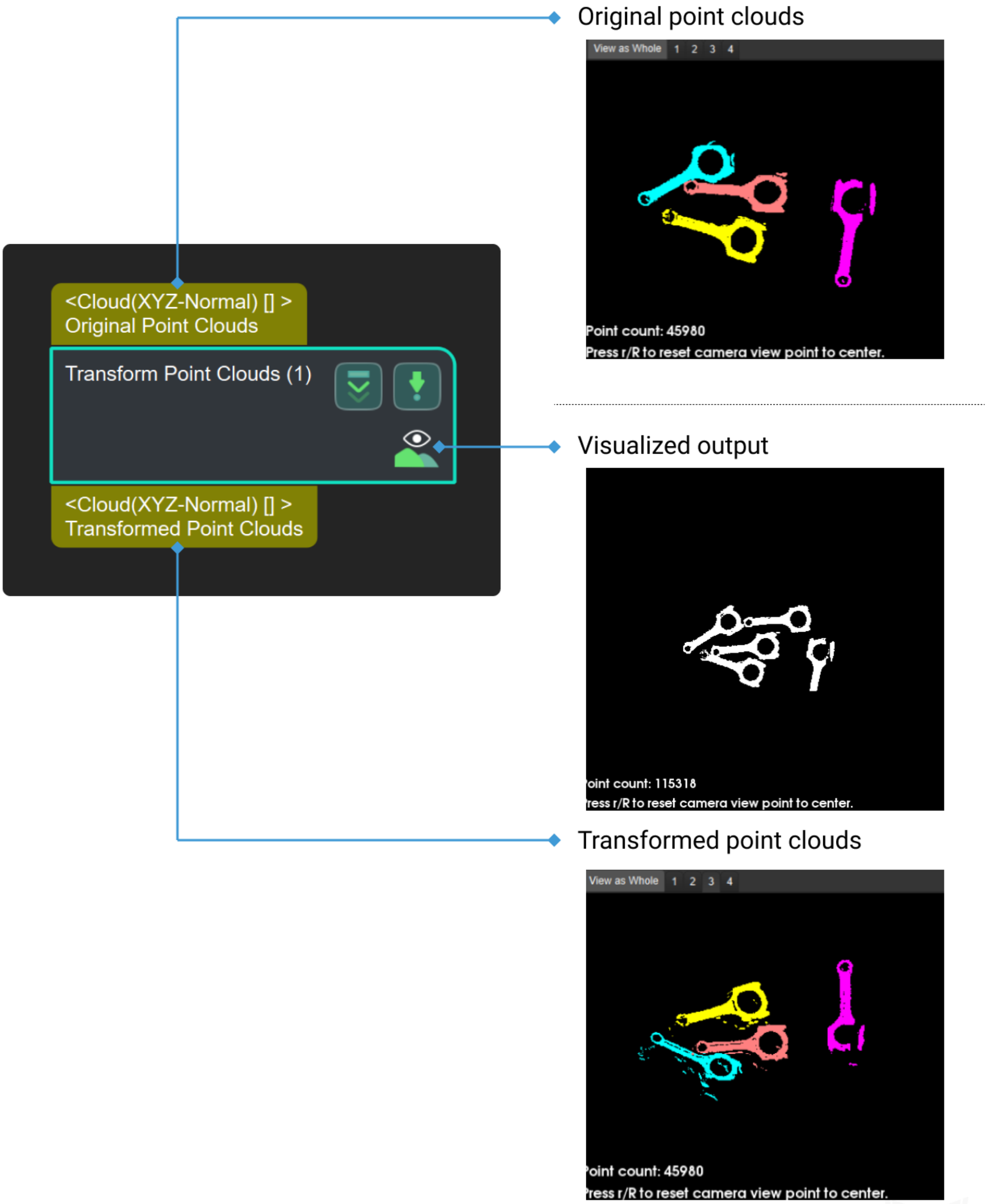
This Step can transform point clouds from one reference frame to another, or transform within its own reference frame according to the reference pose.



Sample Scenario

This Step is usually used to transform point clouds between camera reference frame and robot reference frame.

Input and Output



Parameters

CameraToRobot: transform point clouds from camera reference frame to robot reference frame.

RobotToCamera: transform point clouds from robot reference frame to camera reference frame.

AllWithFrst: transform all point clouds to the reference frame of the first reference pose.

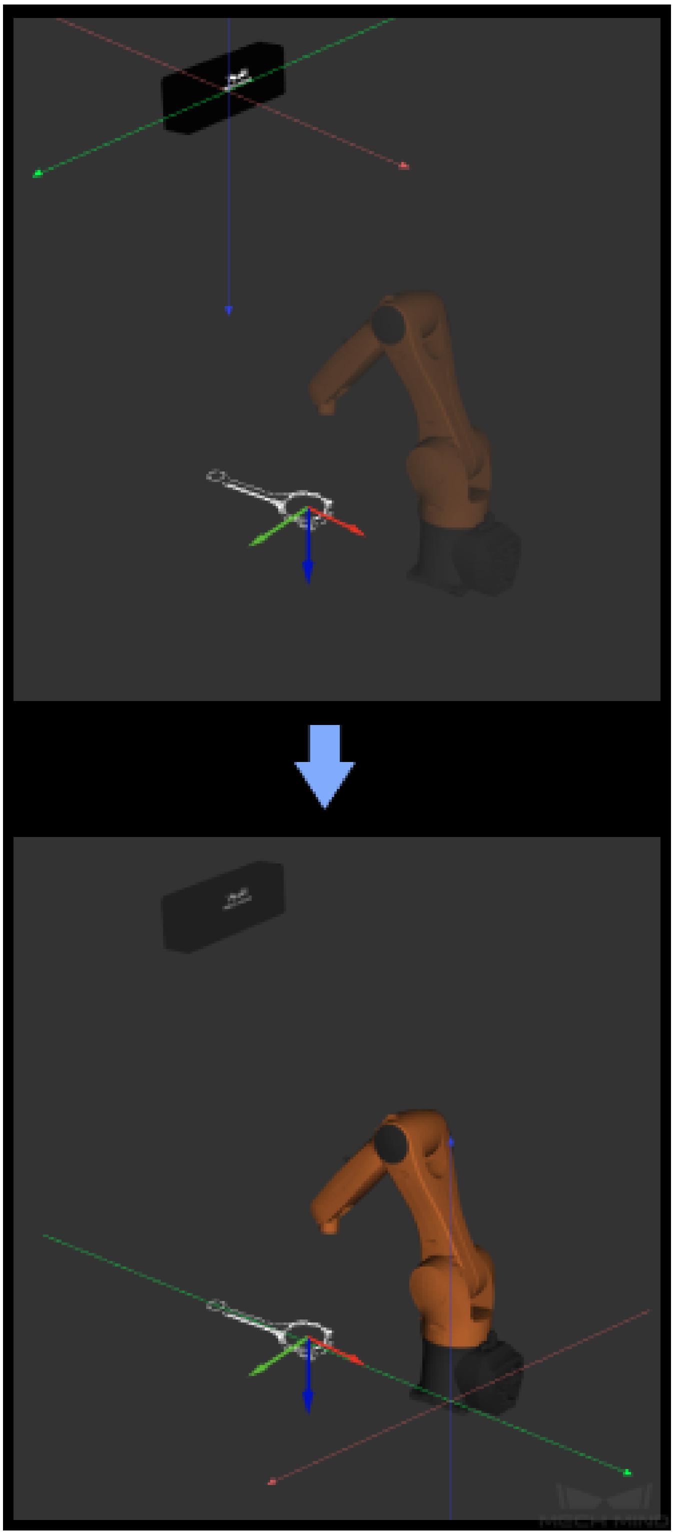
FirstWithAll: transform the first point cloud to the reference frame of reference pose.

UseCorrespondenceInput: transform all point clouds to the reference frame of the corresponding reference pose respectively.

4.30.2 Transform Poses

Function

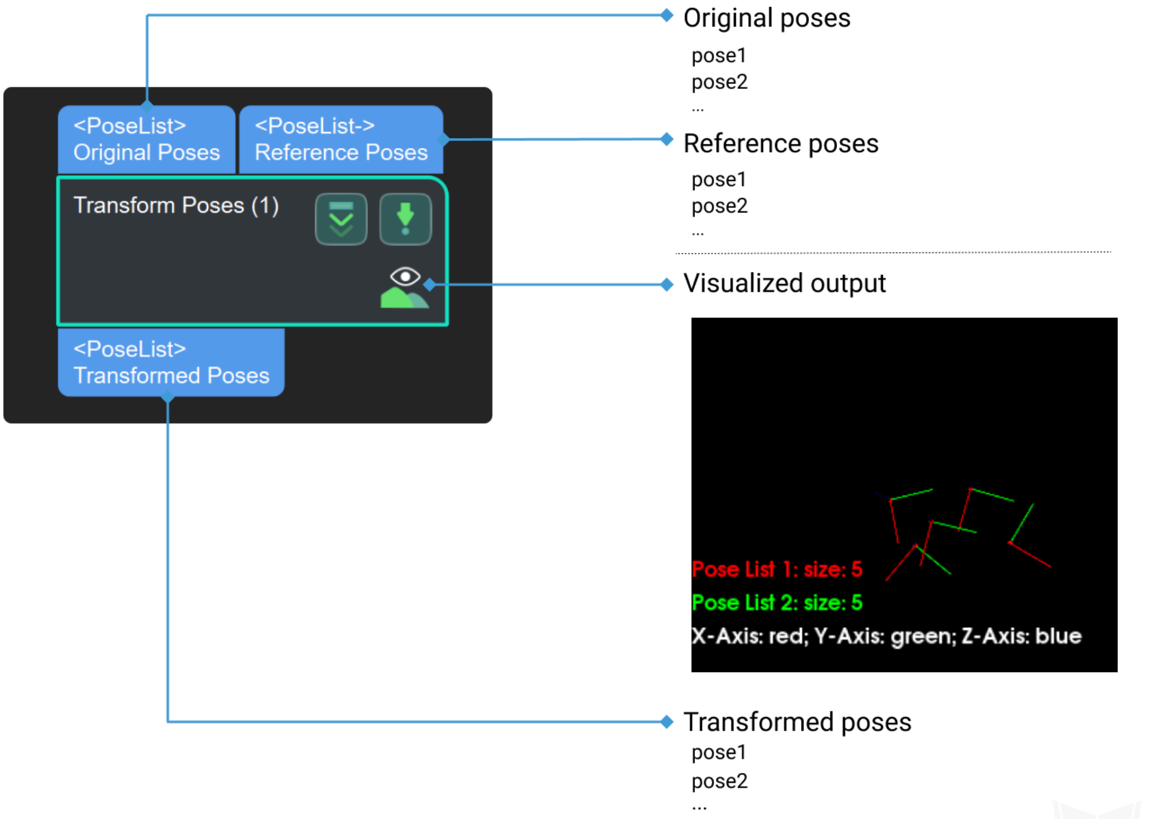
This Step can transform poses from one reference frame to another, or transform within its own reference frame according to the reference pose.



Sample Scenario

This Step is usually used to transform poses between camera reference frame and robot reference frame.

Input and Output



Parameters

type

transformType

Default Value: CameraToRobot (Camera coordinate System to robot Coordinate System)

List of Values: CameraToRobot, RobotToCamera, AllWithFirst, FirstWithAll, UseCorrespondenceInput.

Instruction: "CameraToRobot" means that transforming the pose in camera coordinate system to robot coordinate system while "RobotToCamera" works for inverse transformation. "AllWithFirst" is using the first input reference pose and "FirstWithAll" is using all reference poses to transform the first input pose. "UseCorrespondenceInput" is the one-to-one transformation between the input pose and the reference pose. When selecting the last three, the second input should exist.re should exist moreover for "UseCorrespondenceInput" , the number of input pose and the reference pose should be equal.

4.31 Visualization

In this chapter, steps that are associated with Visualization will be introduced.

4.31.1 Show Images

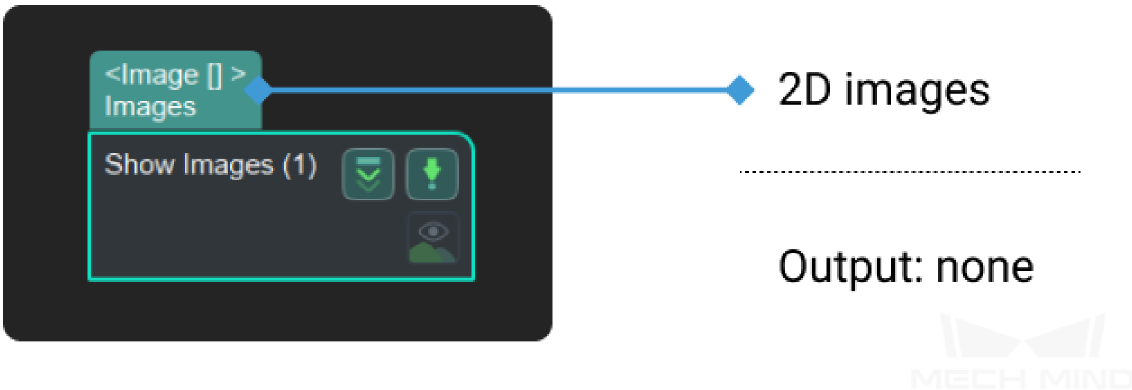
Function

This Step is used to display 2D images output by other Steps, such as some Steps in the **2D Matching**, **2D Feature Detector**, and **2D General Processing** categories.

Sample Scenario

You can tune parameters of previous Steps according to the results of output images until the output results meet requirements.

Input and Output



Parameters

Display Type

Default Value: MultiImages

List of Values: MultiImages, ImgAndPolyVerts

Instruction: **MultiImages**:Show images (as shown on the left side of *Figure1*).
ImgAndPolyVerts:Show images and candidate polygon (as shown on the right side of *Figure1*).

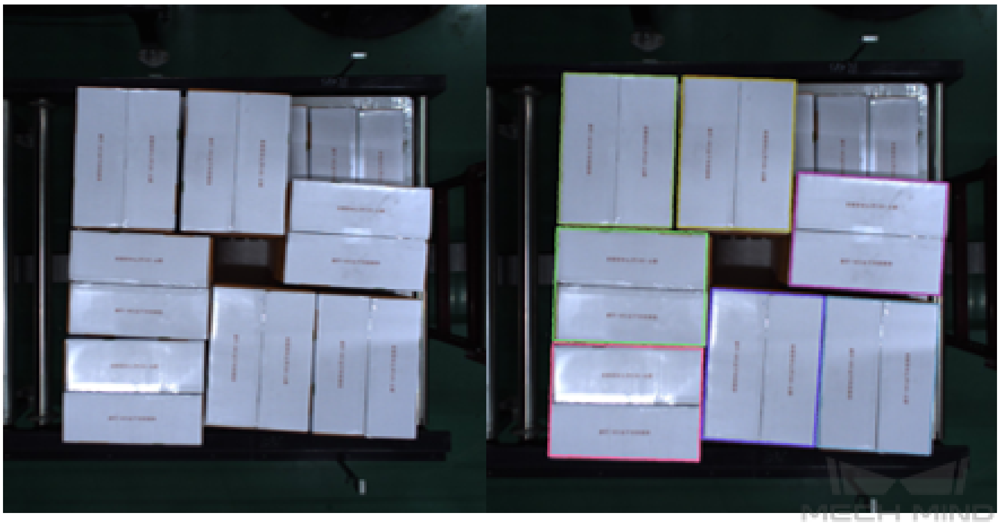


Figure1 Display Type

Show Image in Origin Size

Default Value: False

List of Values: False, True

Instruction: When the display type is **MultiImages**, this parameter can be set to choose whether to display the image according to the original size and original pixel value of the image. When the display type is **ImgAndPolyVerts**, this attribute does not appear.

Example: When using the default value, the image will be displayed in the step. As shown in *Figure2*.

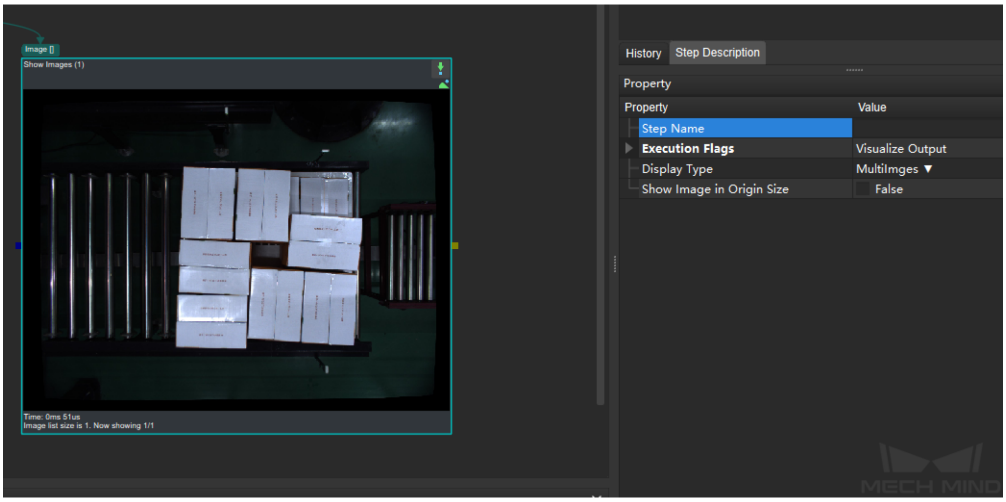


Figure2 Show Image in Origin Size

When it is changed to **True**, the **Image In Origin Size** window will pop up after running this step, and the original image will be displayed in this window.

As shown in *Figure3*.

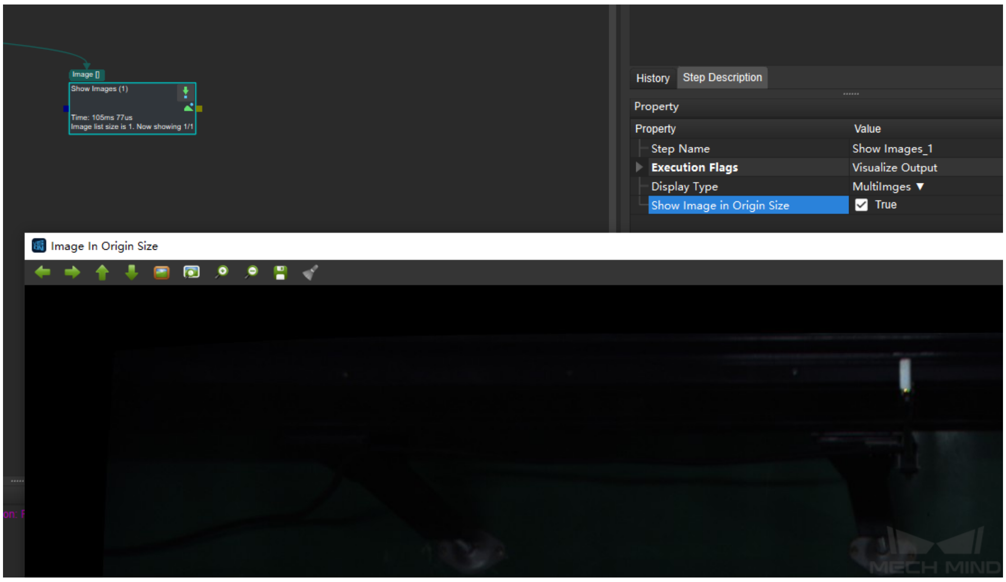


Figure3 Image In Origin Size

4.31.2 Show Point Clouds and Poses

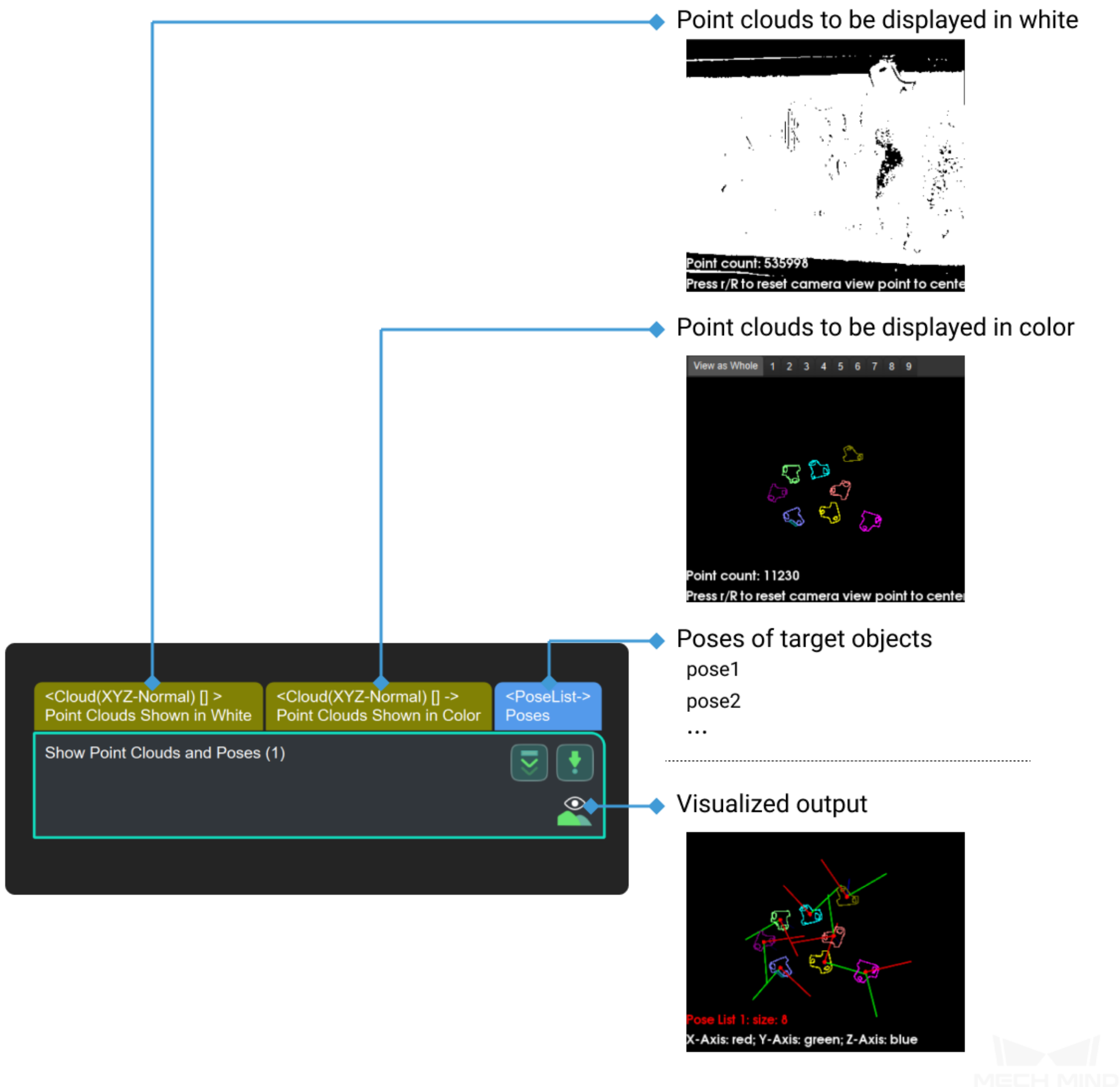
Function

Display Point Clouds and Poses.

Sample Scenario

This Step can display the relative position between the point cloud(s) and pose(s) and is usually used in debugging phase. For example, in a picking application, it can be used to check if the poses are correct.

Input and Output



Parameter Tuning

- **Normal Vector Visualization Settings**

  Show Normals

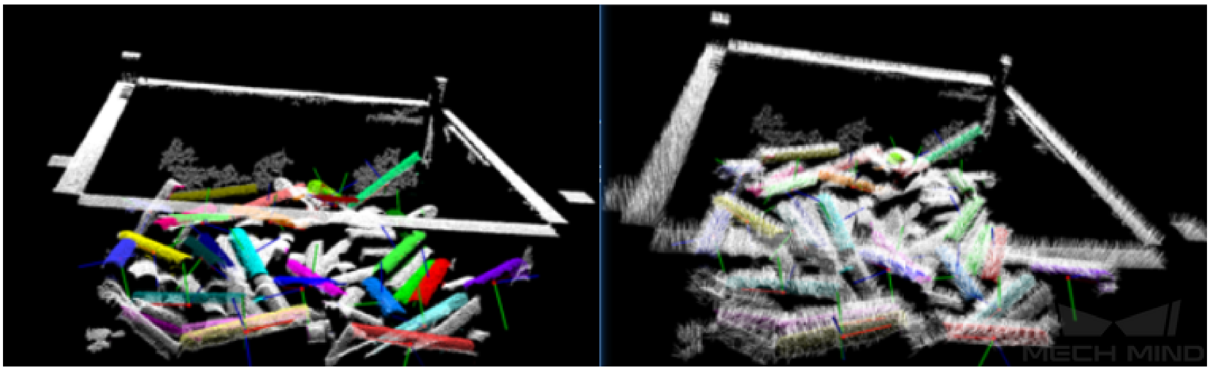
  Default: `False`
  Options: `True`, `False`
  Description: Whether to display the point cloud normal vectors or not.

  Display Interval of Normals

Default: 20

Instruction: This parameter is shown only when **Show Normals** is set to `True`. The unit is mm. The greater the value, the sparser the normal vectors are displayed.

Example: As shown below, the left part is the point cloud displayed without normals, and the right part is the point cloud displayed with normals at an interval of 20 mm.



- **Z Value Visualization Settings**

Visualize Z Value

Default: `False`

Options: `True`, `False`

Instruction: When set to `True`, points of the point cloud will be displayed in an ascending order of brightness along the positive Z-axis.

Upper Bound

Default: 0

Instruction: This parameter is only valid when **Visualize Z Value** is set to `True`. This value sets the upper bound of the range which the points need to be colored along the positive Z-axis. The larger the value, the dimmer the overall display.

Lower Bound

Default: 0

Instruction: This parameter is only valid when **Visualize Z Value** is set to `True`. This value sets the lower bound of the range which the points need to be colored along the positive Z-axis. The smaller the value, the brighter the overall display.

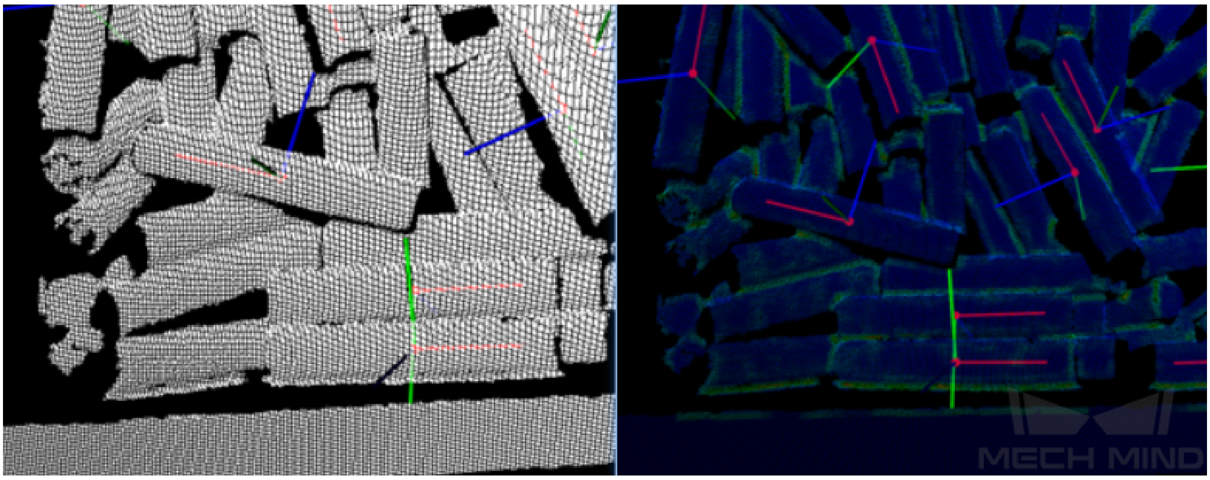
Show Curvature

Default: `False`

Options: `True`, `False`

Instruction: This parameter shows points of object parts with different curvatures in different colors. Red is used for large curvatures, blue for medium curvatures, and green for small curvatures.

Example: As shown below, the left part is the point cloud displayed without curvatures and the right part is the point cloud displayed with curvatures.



Comparison of Similar Steps

Show Point Clouds and Poses is used for checking the point clouds and poses. *Show Images* is used for checking the color images and masks.

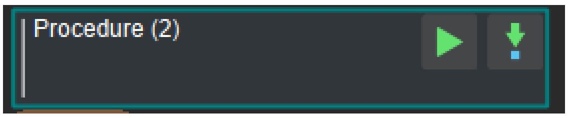
4.32 Others

In this chapter, steps that are associated with Others will be introduced.

4.32.1 Detect Graspable Rectangles

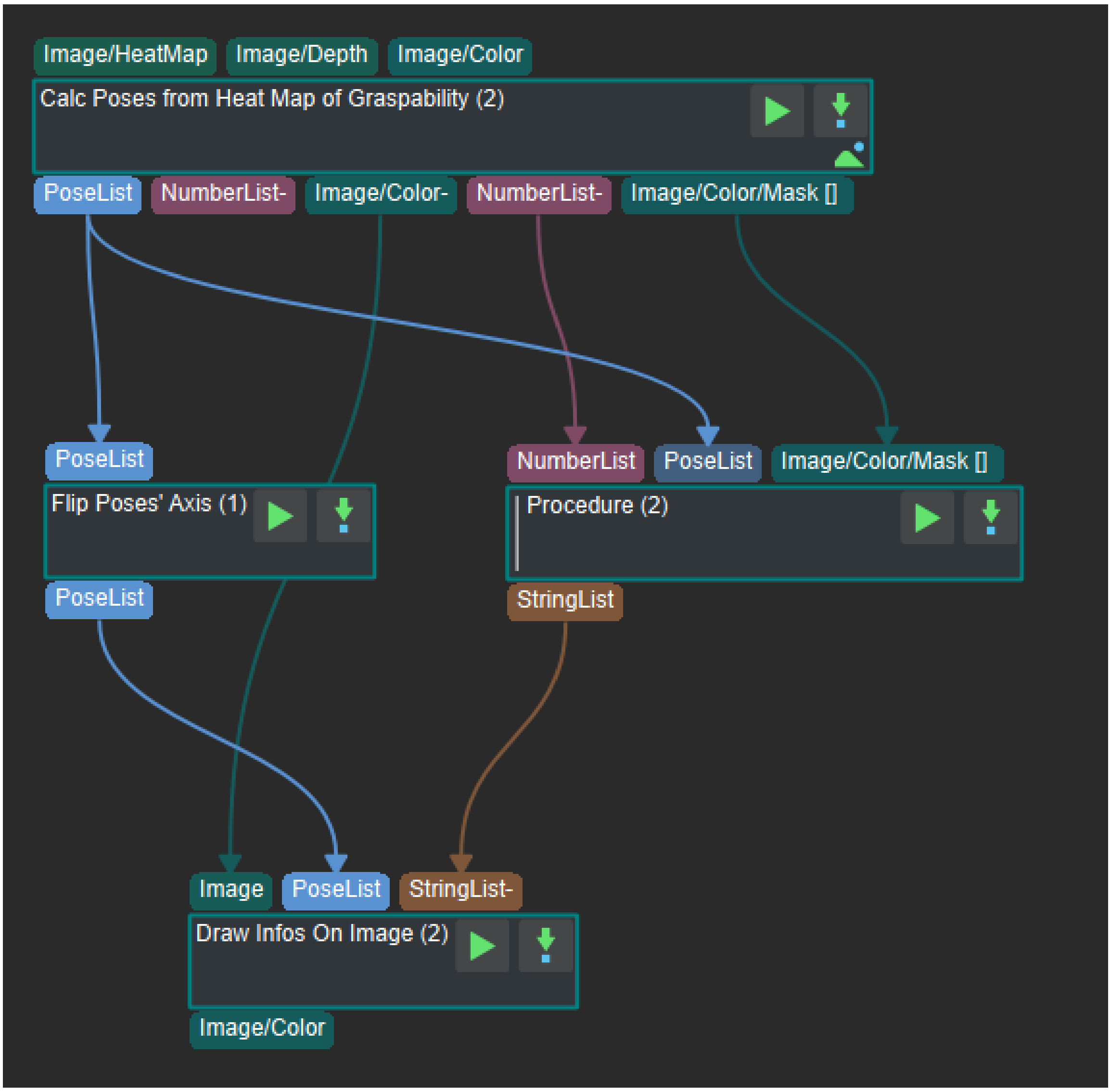
4.32.2 Procedure

Description



The procedure

Multiples steps can be put into a "procedure". Double-click procedure or right-click on the blank select "Navigate Down" to entre this "procedure", add the required steps into the "procedure" and connect them.

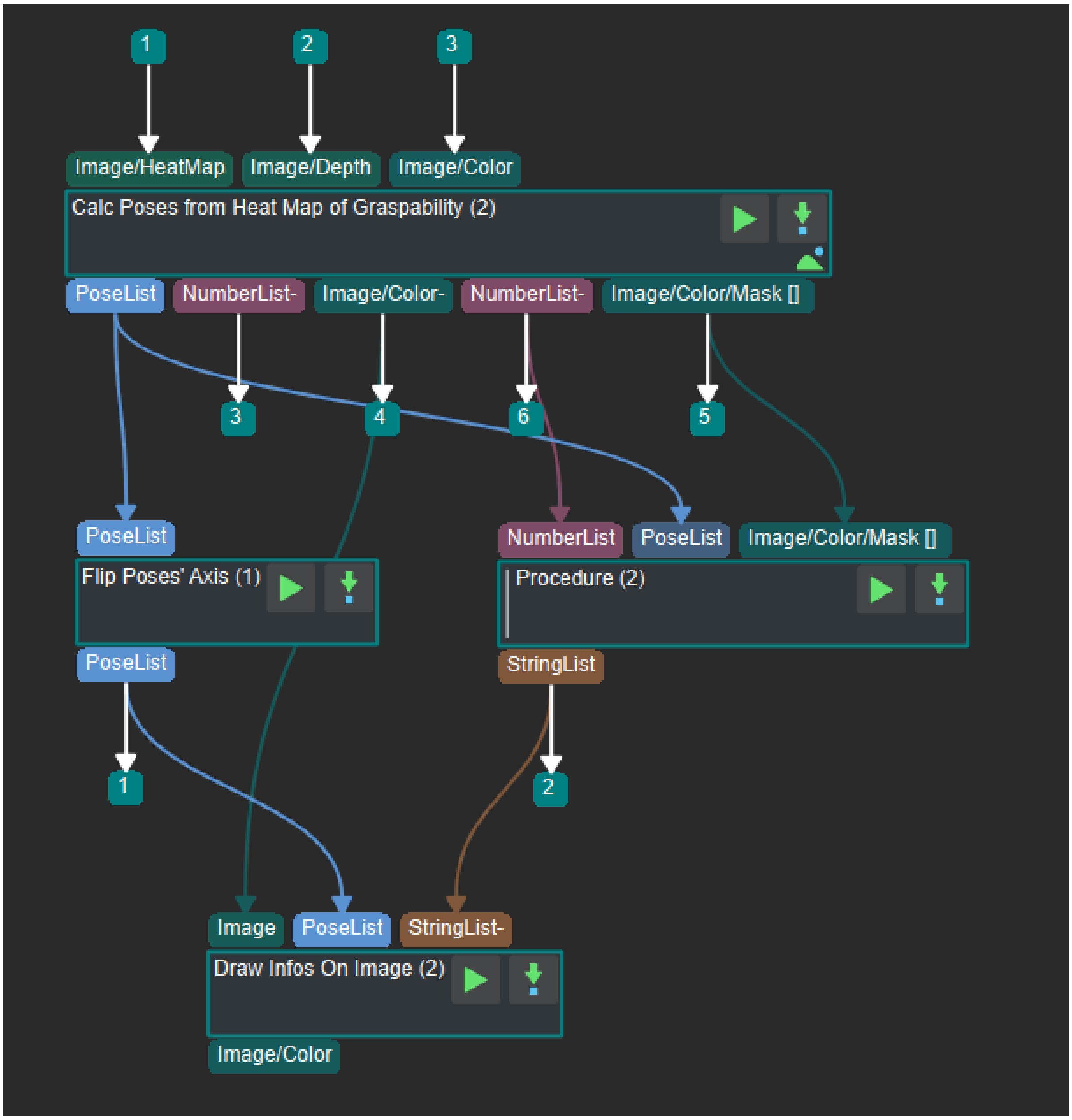


Add step into "procedure"

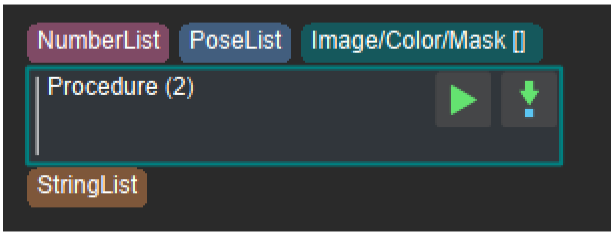
Right-click on the blank select "Navigate Up" to exit this procedure.

Set the Input and Output of Procedure

After the "procedure" is created, then need to set the input and output of this procedure. The setting method is relatively simple, 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.



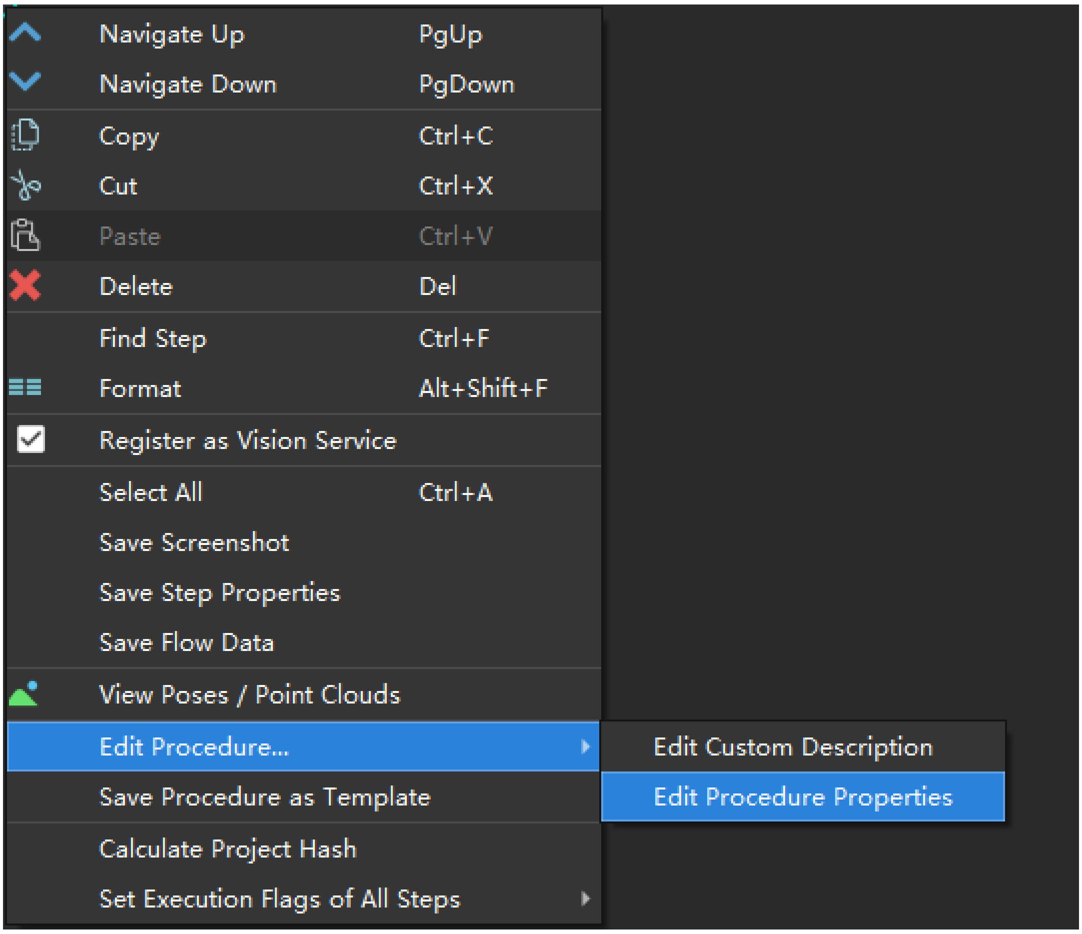
Set the input and iutput of procedure



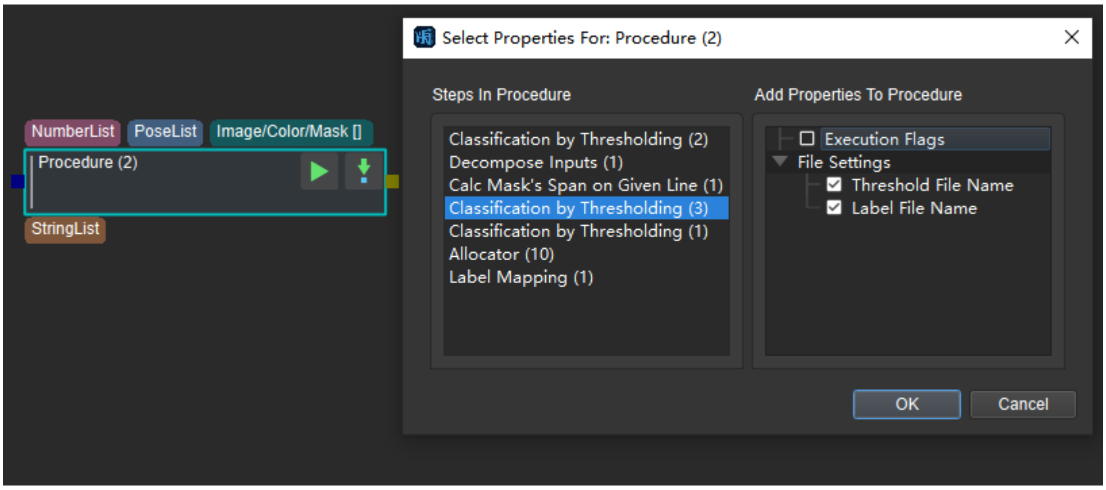
The input and output of procedure

Set the Parameters of Procedure

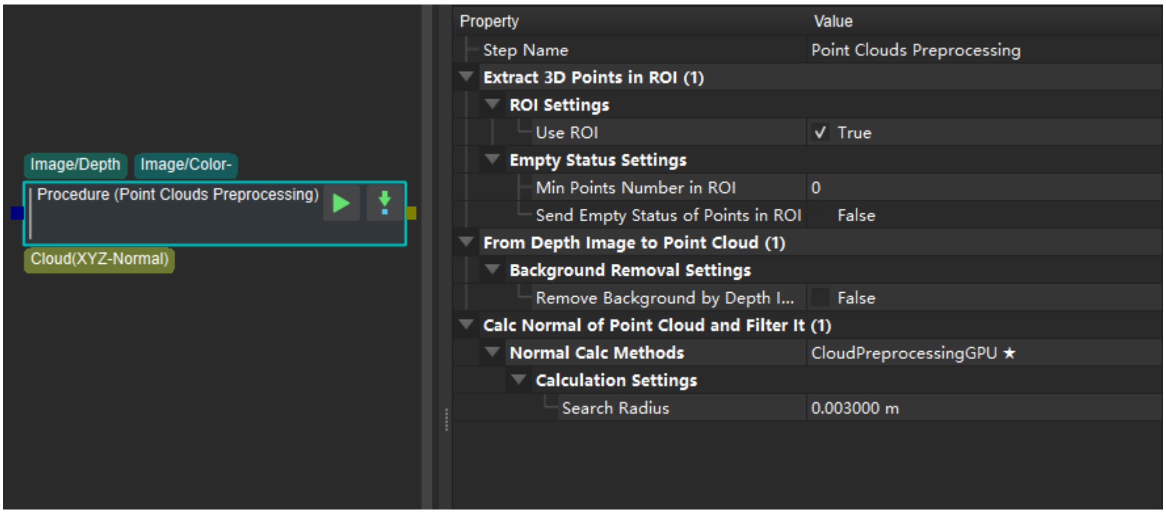
In order to facilitate the adjustment of the "procedure", choose the key parameter of each step in the "procedure" as the attribute of the "procedure". Select "procedure" and right click, select *Edit Procedure Parameters*. Select the attributes to be displayed in the opened window and click OK after selecting. The setting steps and effects are shown in the figure below.



Open procedure parameters setting



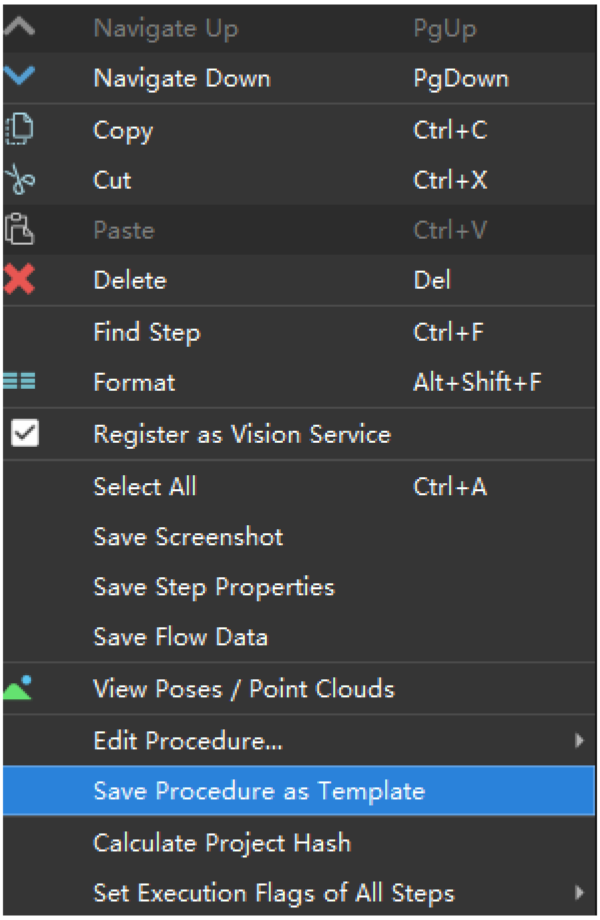
Set parameters for procedure



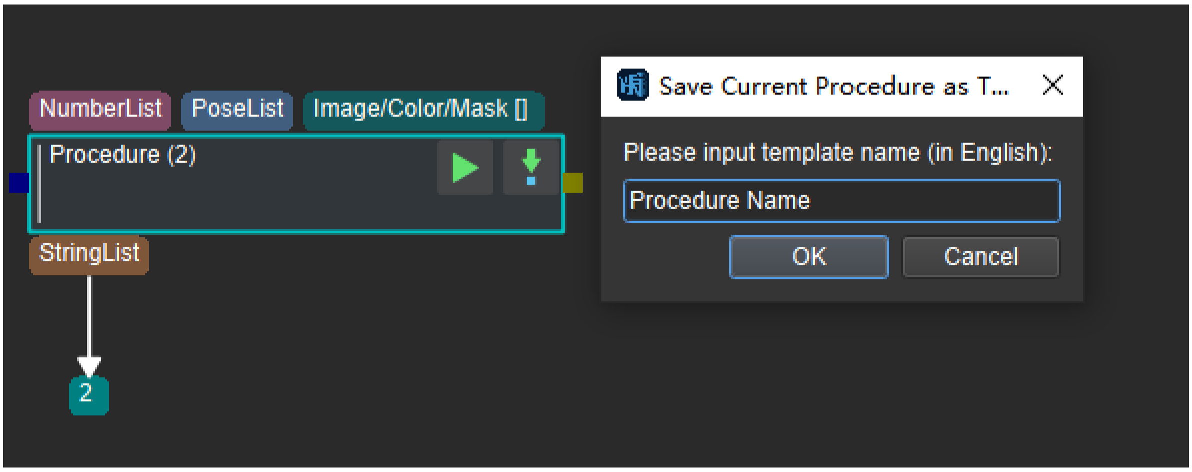
Procedure parameters

Save and Call a Procedure

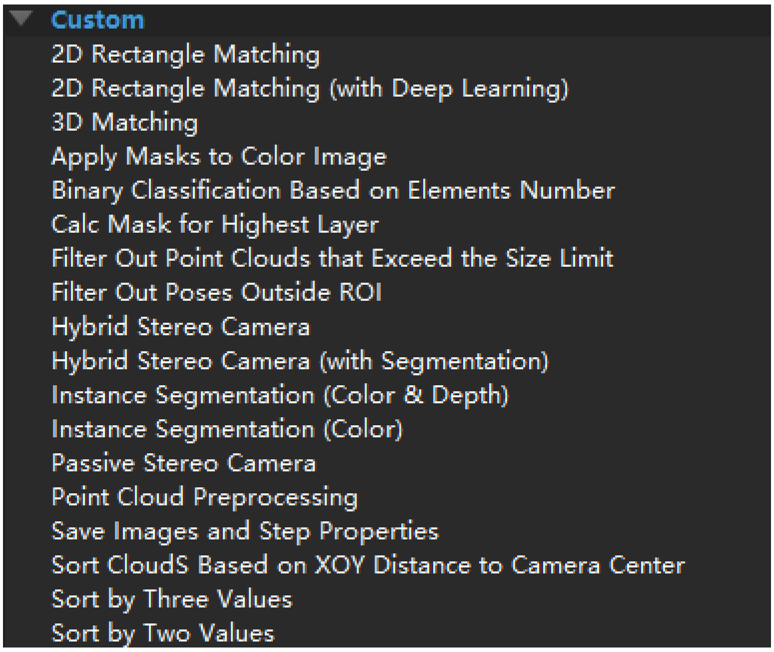
To reuse the Procedure in the future, you can save it as a template. Right-click it and select *Save Procedure as Template* . Enter the template name in the pop-up interface. The saved Procedure can be found under the “Custom group” in the “Step Library” .



Save the procedure as a template



Name and save the procedure template



The template of the saved procedure

4.33 Legacy

In this chapter, steps that are associated with Legacy will be introduced.

4.33.1 Collect/Remove

Collect Poses in 3D ROI

Function

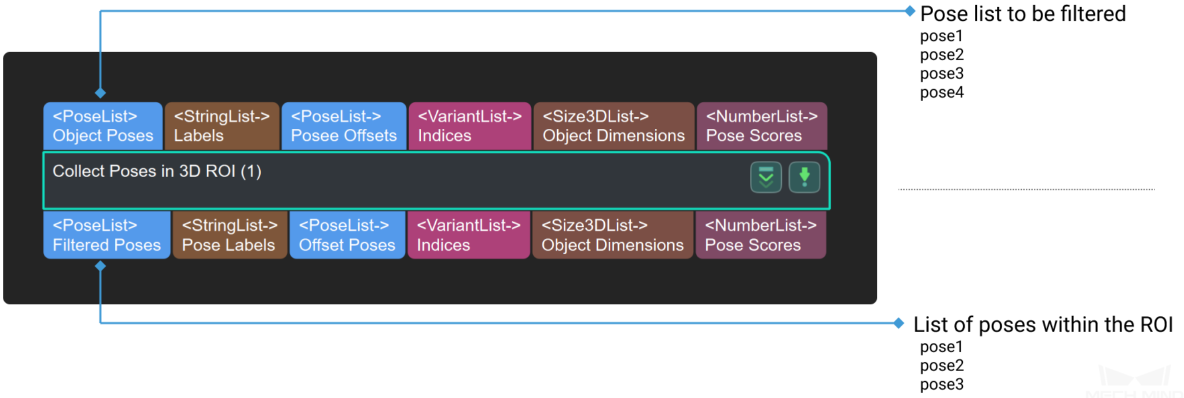
Filter the input pose list by the set 3D ROI. The poses inside the ROI will be retained, and the poses outside the ROI will be discarded.

Sample Scenario

A general pose filtering Step.

This Step is outdated. Please use the newer Step *Validate Existence of Poses in 3D ROI* instead.

Input and Output



Preparation

Before setting a 3D ROI, please go to **Project Assistant** and select the data source of scene point cloud. For detailed settings, please refer to *Scene Point Cloud*.

Parameters

ROI setting

3D ROI Name

Instruction: Click on the right side of the 3D ROI Name to enter the **Set ROI** window. For detailed settings, please refer to *Instructions for Setting 3D ROI*.

Input Poses Coordinate Type

Suppress Neighboring Poses with Low Scores (NMS)

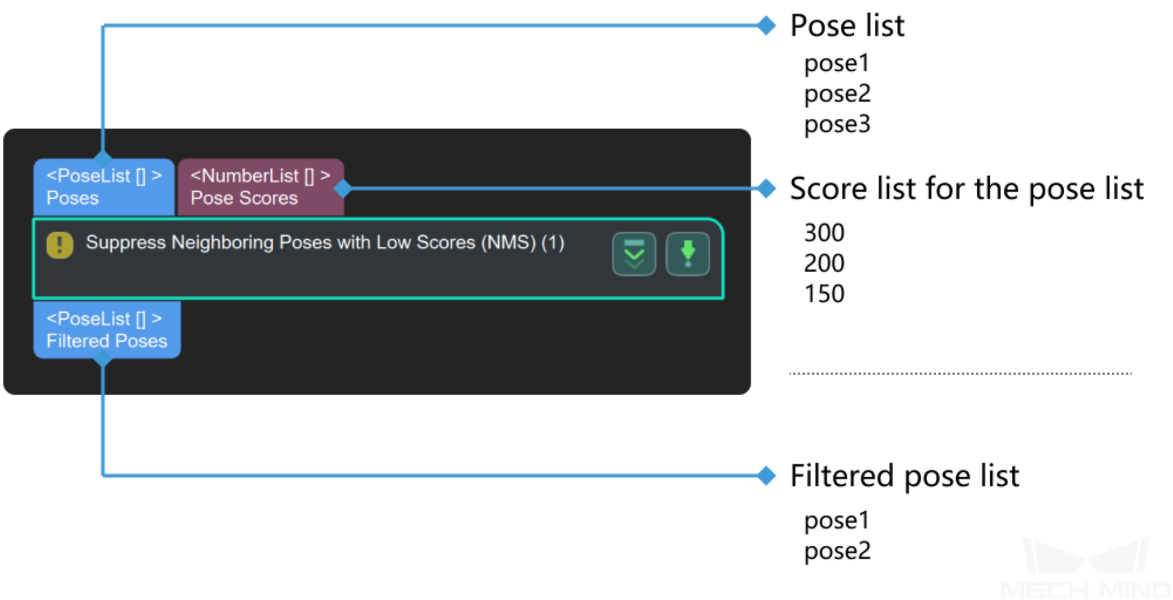
Function

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.

Sample Scenario

One of the methods for filtering poses. This Step belongs to an earlier version; the function has been integrated in *3D Coarse Matching*, and it is recommended not to use this step.

Input and Output



Parameters

Parameters

searchRadiusNMS

Default Value: 0.05m

Instruction: Define the range of the local search. If it is larger, more poses are involved in the calculation, which may result in a longer processing time for later step.

scoreRatioThreshold

Default Value: 0.02

Instruction: When the pose score is less than the threshold, it is not considered.

4.33.2 Image - Image

Map Mask Non-zero Area

4.33.3 Read

Read Images

Read Point Cloud

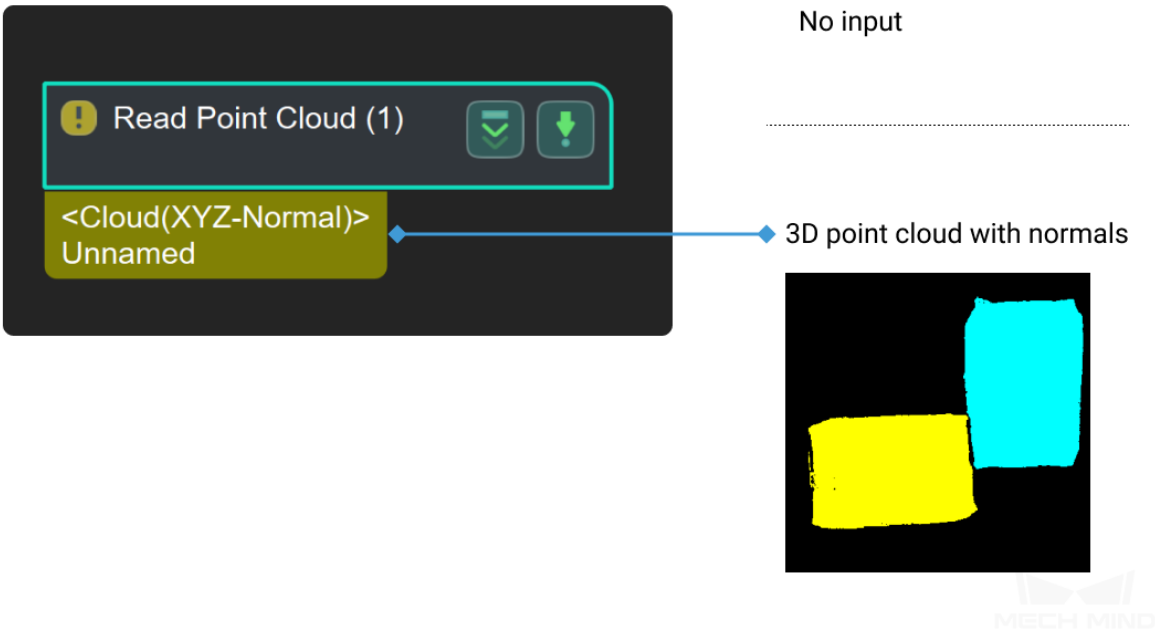
Function

Read point cloud in the PLY format.

Sample Scenario

Read the point cloud in PLY format from a specified path. This Step is outdated. Please use the newer Step *Read Point Cloud V2* under **Reader & Saver**.

Input and Output



4.33.4 Remove Points

Cloud Filter by Model and Pose

Remove Points from Point Cloud

4.33.5 Adjust Poses by Obstacles

4.33.6 Calc Absolute Values

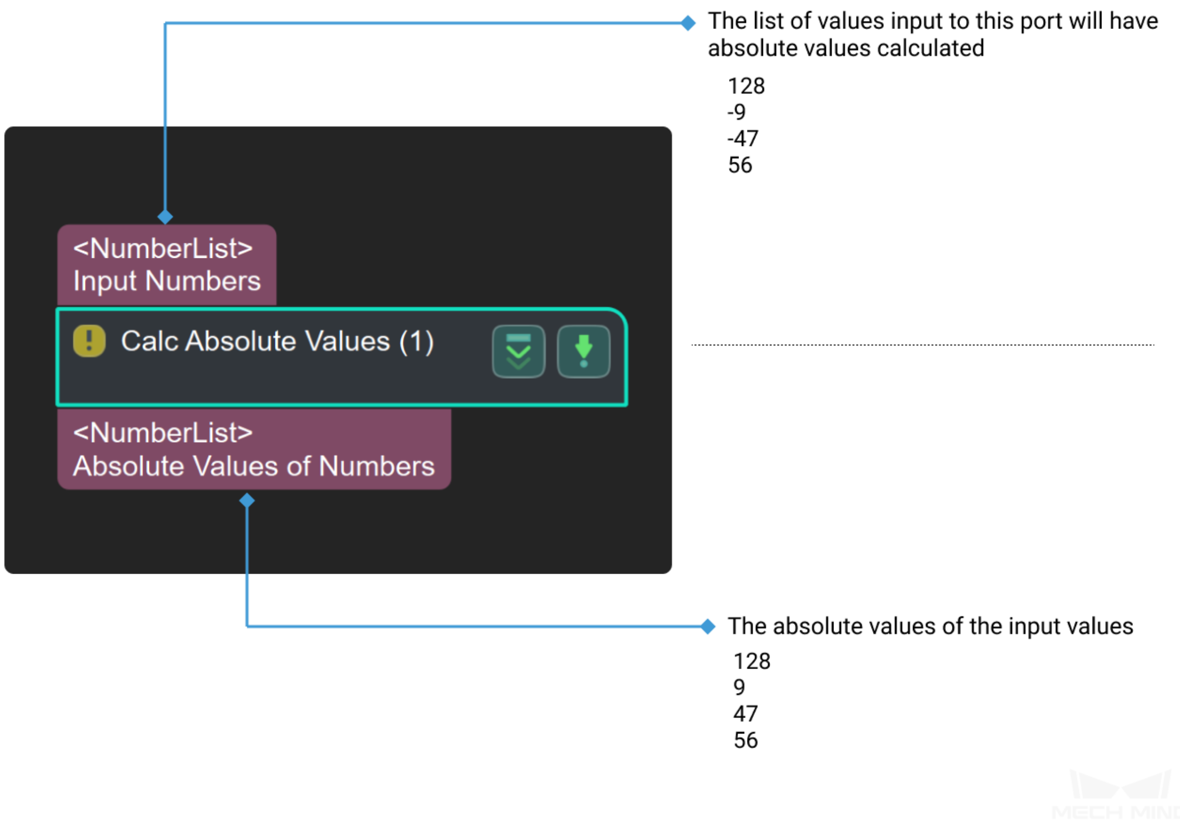
Function

Calculate the absolute values of a list of input values.

Sample Scenario

This Step is outdated. Please use the newer Step *Numeric Operation* that has more functions instead.

Input and Output



4.33.7 Cloud Processing (GPU)

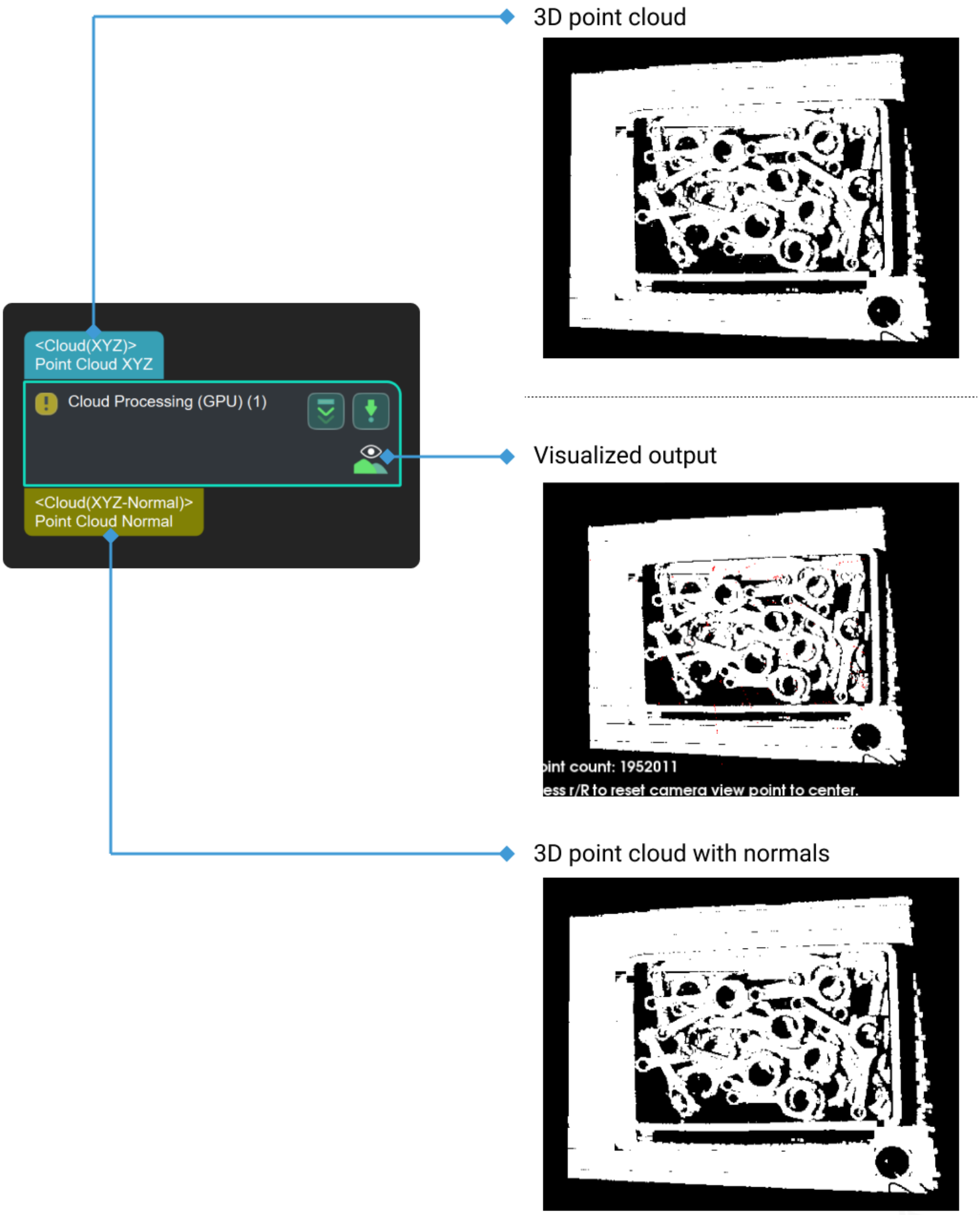
Function

Concatenate the two pose lists to merge them into one.

Sample Scenario

Add normals to the input point cloud and remove outliers. Usually used after *Capture Images from Camera*, *From Depth Map to Point Cloud*. This Step belongs to an earlier version. Please instead use *Calc Normals of Point Cloud and Filter It*.

Input and Output


3D point cloud
<Cloud(XYZ)>
Point Cloud XYZ
Cloud Processing (GPU) (1)
Visualized output
oint count: 1952011
ess r/R to reset camera view point to center.
<Cloud(XYZ-Normal)>
Point Cloud Normal
3D point cloud with normals

4.33.8 Filter by Labels

Function

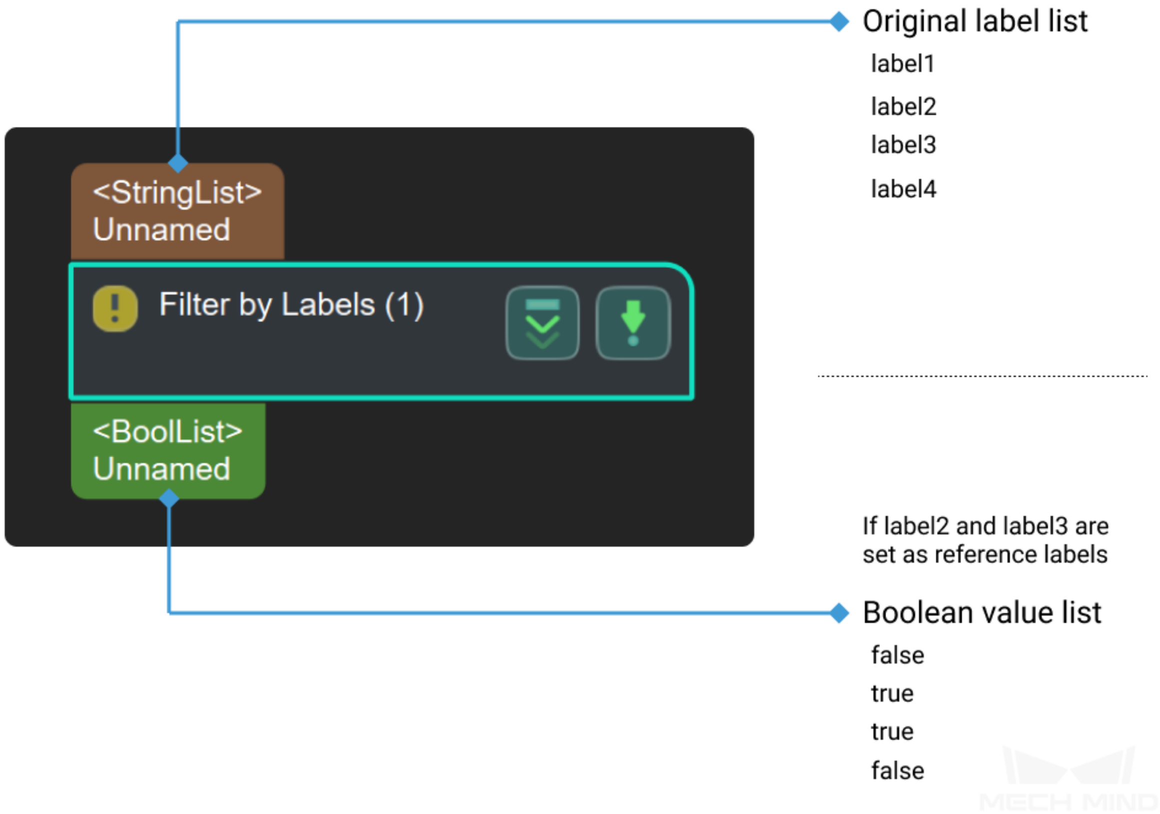
Check if each item in the original label list exists in the reference label list. The result of the check will be output as a list of Booleans. If an original label is in the reference label, then the corresponding item in the output list is True.

Sample Scenario

It is usually used with *Filter*. The list of Boolean values output by this Step will be used as the basis for filtering.

This Step belongs to an earlier version. In this Step, the reference label list can only be set by reading from the file. Please instead use *Validate Labels and Output Flags*, for which the reference list can be input through the port.

Input and Output



4.33.9 Get First Image from Image List

Function

Get the first image in an image list.

Sample Scenario

This Step belongs to an earlier version, it is recommended to use *Trim Input List* and *Unpack Data* in sequence.

Input and Output

4.33.10 Merge Pose Lists

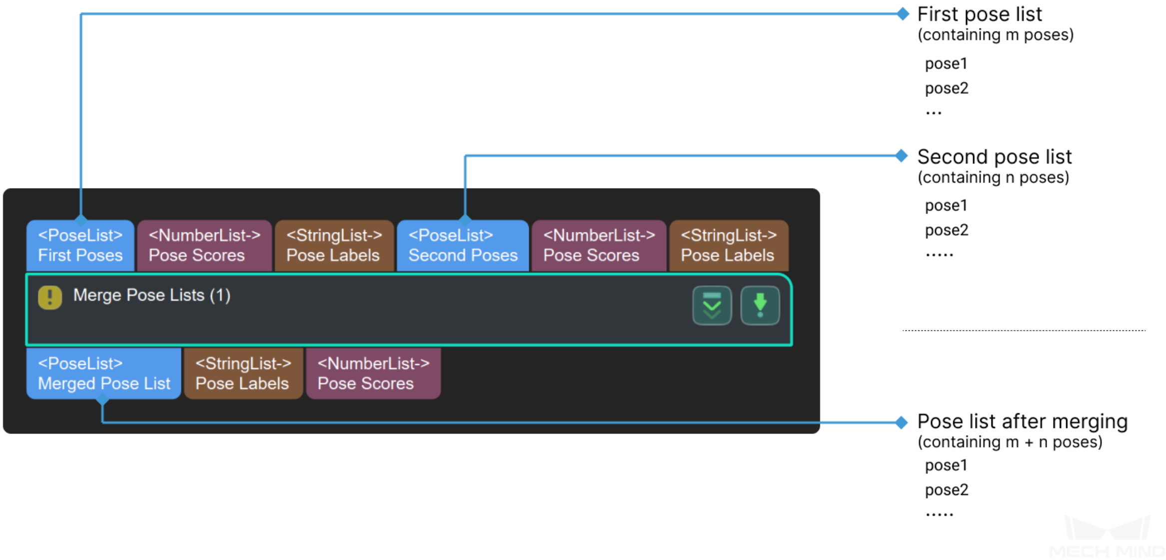
Function

Concatenate the two pose lists to merge them into one.

Sample Scenario

For various scenarios where pose data structures need to be adjusted. 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.

Input and Output



4.33.11 Number Scaling

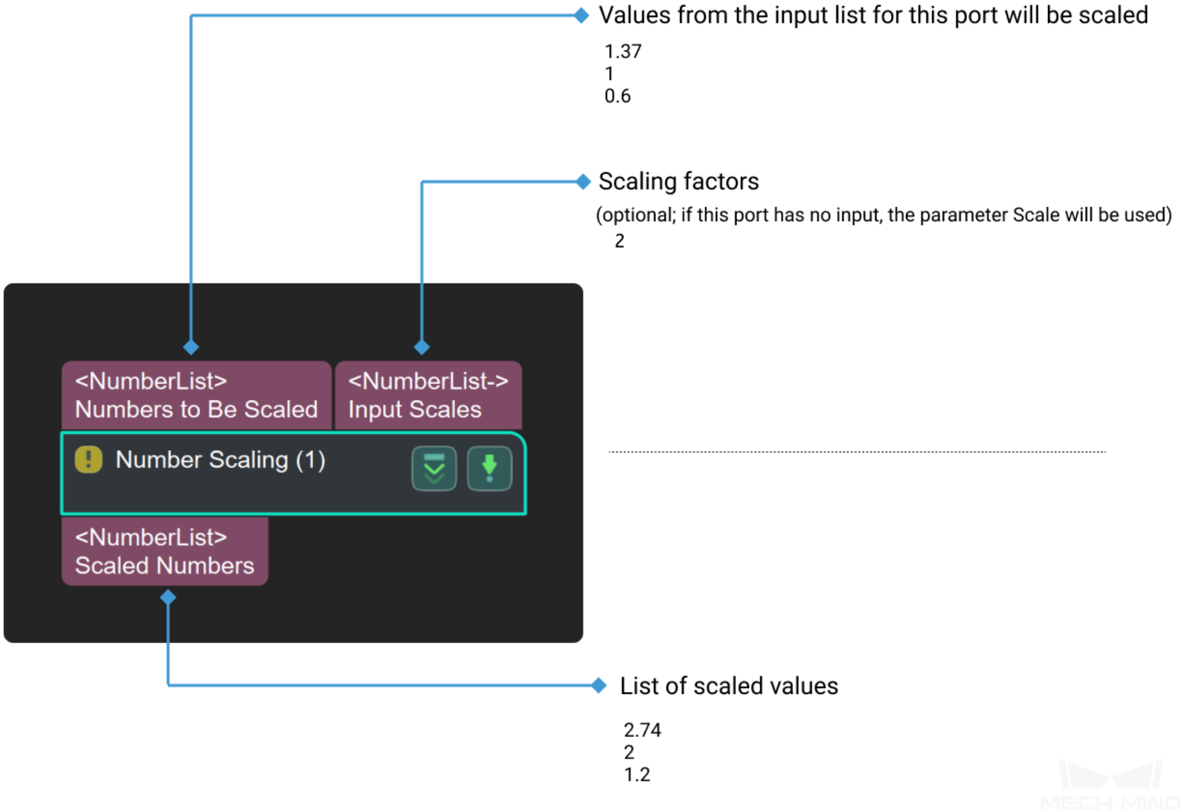
Function

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.

Sample Scenario

This Step is outdated, please use the newer Step *Numeric Operation* which has more functions instead.

Input and Output



4.33.12 Point Cloud Filter

Function

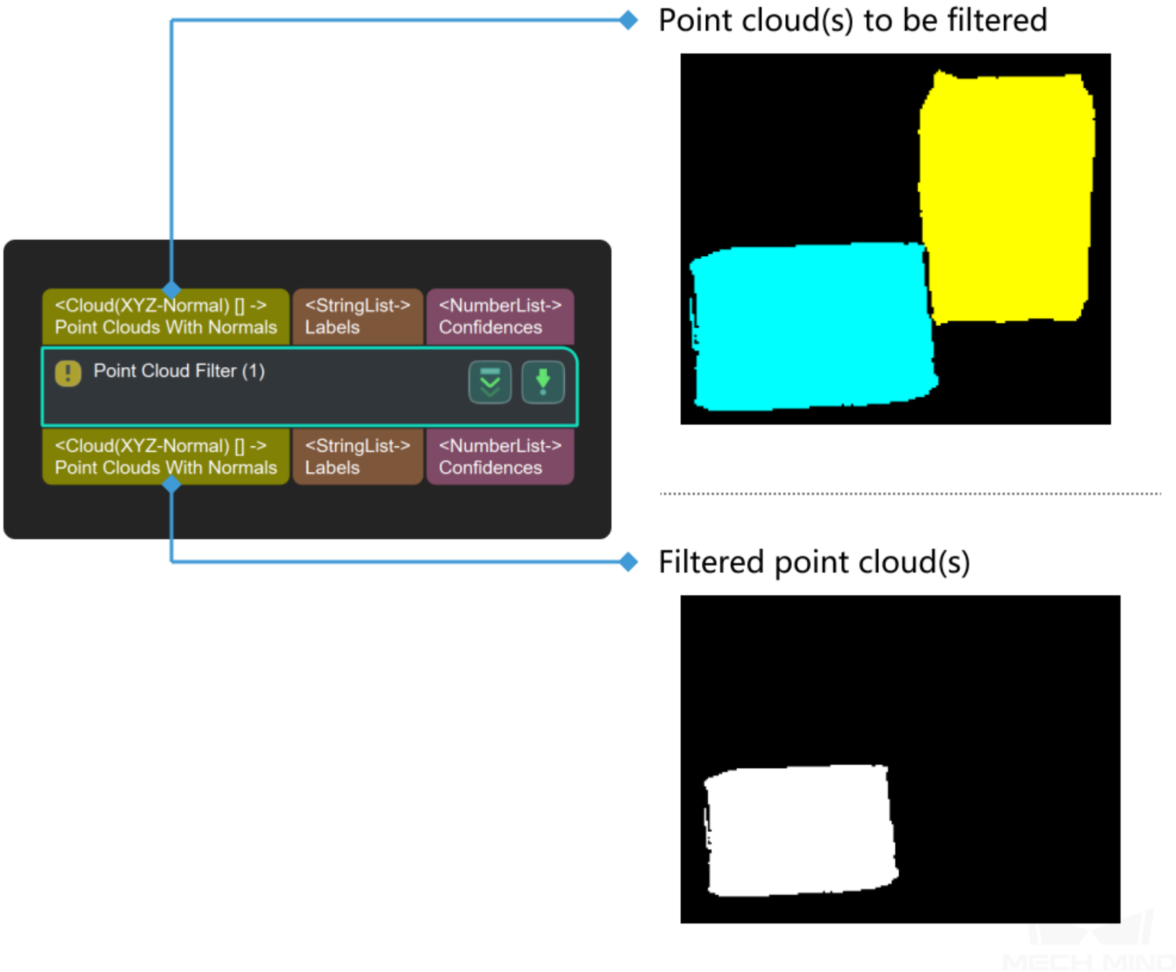
Filter the input point clouds according to the set rules.

Sample Scenario

Usually used to filter point clouds based on the number of points (setting the **Filter** parameter to **CloudCapacityFilter**).

This Step belongs to an earlier version and cannot output Boolean results; please use *Validate Point Clouds* instead.

Input and Output



Parameters

__filterType

Instruction: This parameter is used to select the filter type, CircleCloudsFilter, CloudCapacityFilter,RecCloudsFilter and SolidCloudsFilter included. | Default Value: CircleCloudsFilter | Suggested Value: To set according to the actual scenarios. | List of Values: CircleCloudsFilter, CloudCapacityFilter,RecCloudsFilter,SolidCloudsFilter

CircleCloudsFilter

'CircleCloudsFilter'could tranfer the 3D point clouds into 2D mask image to get the contour of the mask and calculate the circulariry and perimeter in order to filter out the point clouds that are not covered in the range.

contourRetrievalMode

Instruction: 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.

minAreaThreashold

Instruction：This parameter is the lower limit of the area of the circle clouds(in pixel).

Default Value：100

Suggested Value：To set according to the actual scenarios.

maxAreaThreashold

Instruction：This parameter is the upper limit of the area of the circle clouds(in pixel).

Default Value：1000

Suggested Value：To set according to the actual scenarios.

minCircularity

Instruction：This parameter is the lower limit of the circularity of the circle.

Default Value：0.5

Suggested Value：To set according to the actual scenarios.

maxCircularity

Instruction：This parameter is the upper limit of the circularity of the circle.

Default Value：1

Suggested Value：To set according to the actual scenarios.

minConvexity

Instruction：This parameter is the lower limit of the convexity of the circle.

Default Value：0.8

Suggested Value：To set according to the actual scenarios.

maxConvexity

Instruction：This parameter is the upper limit of the convexity of the circle.

Default Value：1

Suggested Value：To set according to the actual scenarios.

minInertia

Instruction：This parameter is the lower limit of the ratio between the length of short axis and the length of long axis .

Default Value：1000

Suggested Value：To set according to the actual scenarios.

maxInertia

Instruction：This parameter is the upper limit of the ratio between the length of short axis and the length of long axis .

Default Value：1000

Suggested Value：To 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.

minPointsNum

Instruction：This parameter is the lower limit of the number of the point in the point clouds.

Default Value：6000

Suggested Value：To set according to the actual scenarios.

Valid Range:(0, maxPointsNum)

maxPointsNum

Instruction：This parameter is the upper limit of the number of the point in the point clouds.

Default Value：360000

Suggested Value：To set according to the actual scenarios.

Valid Range:(minPointsNum, ∞)

RecCloudsFilter

minRectangularity

Instruction：This parameter is the lower limit of the rectangularity of the rectangle.

Default Value：0.9000

Suggested Value：To set according to the actual scenarios.

SolidCloudsFilter

ratio

Instruction：This parameter is the upper lower of the size ratio between the point cloud and the solid. | Default Value：0.9 | Suggested Value：To set according to the actual scenarios. | Valid Range:(0, 1]

removeSolidCloud

Instruction: This parameter decides whether to filter out the solid point clouds or not. | Default Value: True | Suggested Value: To set according to the actual scenarios. | List of Values: True, False

useOrthogonalProj

Instruction: This parameter decides whether to calculate the mask along the main normal or not. | Default Value: True | Suggested Value: To set according to the actual scenarios. | List of Values: True, False

dilateSize

Instruction：This parameter is used to adjust the size of the structural elements(in pixel) during the dilating process and only valid when ‘filterType’ is ‘CircleCloudsFilter’ or ‘CloudCapacityFilter’. | Default Value：3 | Suggested Value：3,5,7 | Valid Range:[1, ∞)

erodeSize

Instruction：This parameter is used to adjust the size of the structural elements(in pixel) during the eroding process and only valid when ‘filterType’ is ‘CircleCloudsFilter’ or ‘CloudCapacityFilter’. | Default Value：3 | Suggested Value：3,5,7 | Valid Range:[1, ∞)

4.33.13 ccording to Given Indices

4.33.14 Subtract Real Numbers

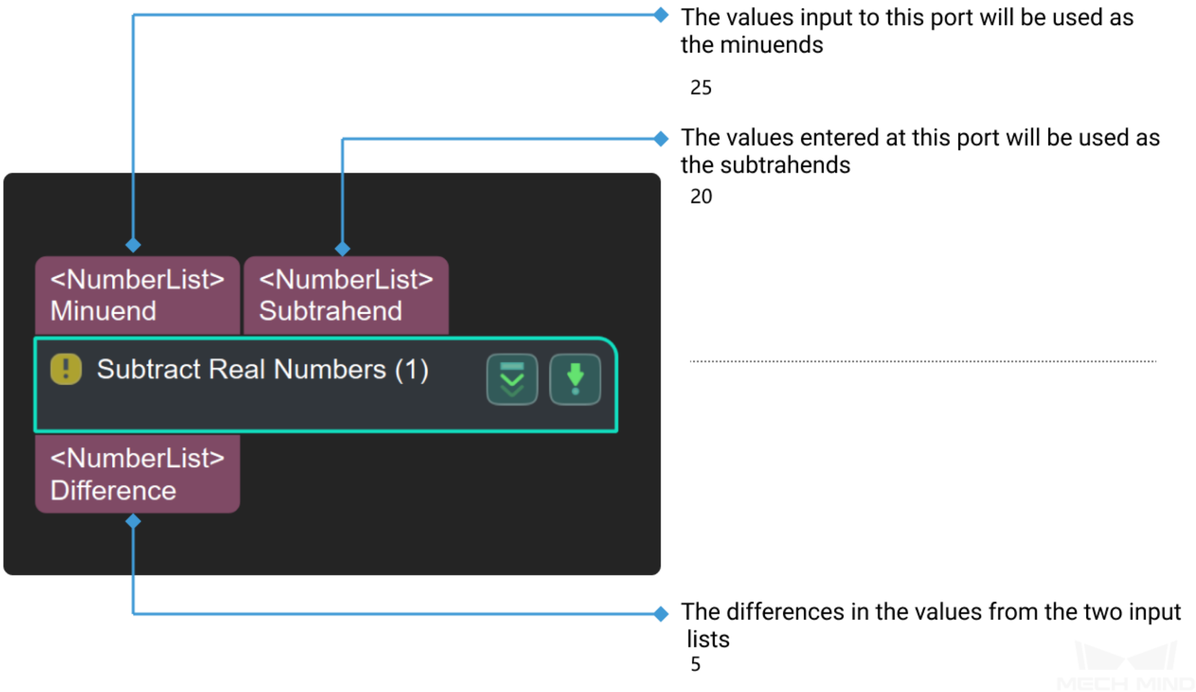
Function

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.

Sample Scenario

This Step is outdated. Please use the newer Step *Numeric Operation* that has more functions instead.

Input and Output



4.33.15 Trajectory Points Matching

4.33.16 Trim Pose List

Function

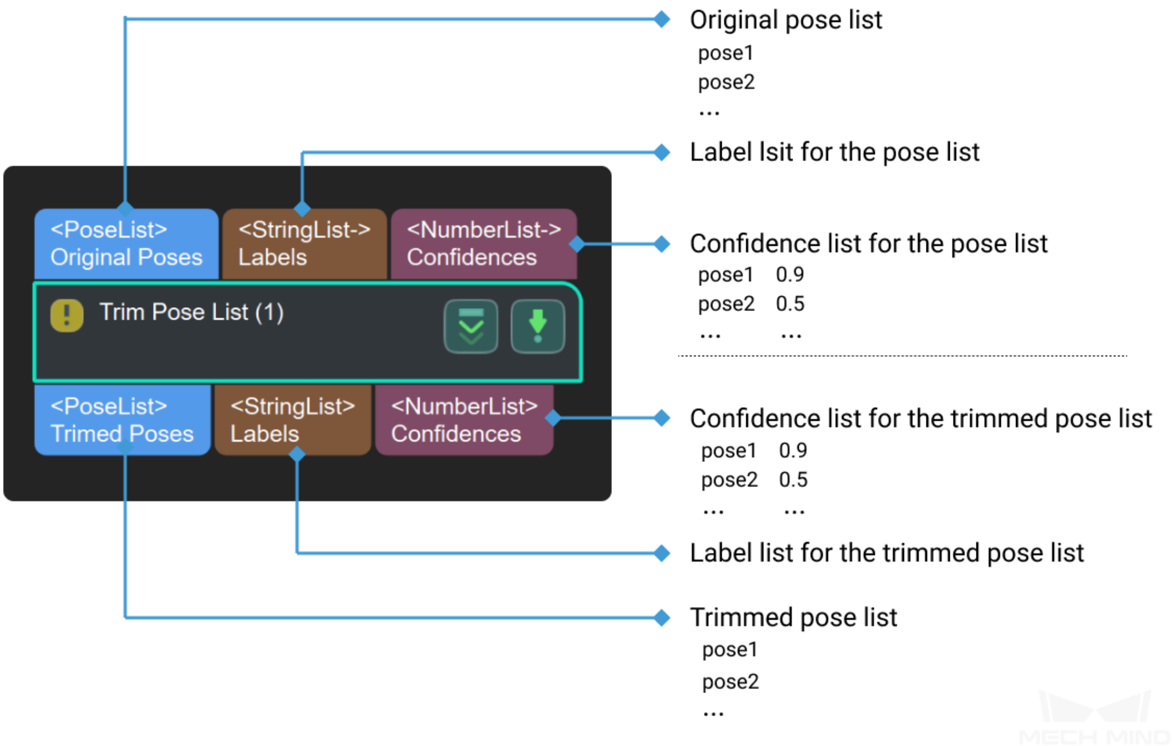
Keep the top N items in the pose list and output. Items after the Nth item will be discarded.

Sample Scenario

Extract the top N items in the pose list, and the N value can be set in the Parameter panel by requirements.

This Step belongs to an earlier version and can only be used for lists of pose-related data types. Please instead use the more general Step: *Trim Input List*.

Input and Output



Parameters

number limitation

poseNumLimit

Default Value: 1

Instruction: Specify the number of output poses.

4.33.17 Pose Filter

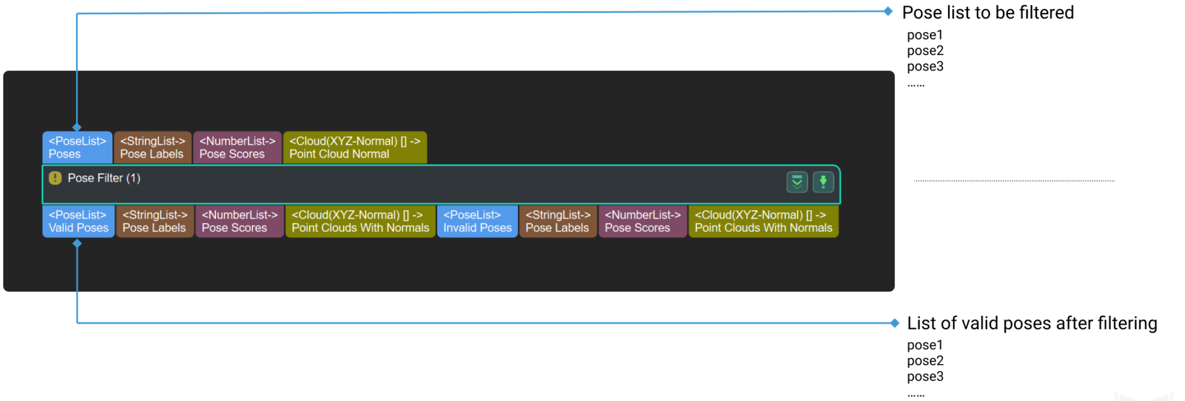
Function

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.

Sample Scenario

This Step is outdated. Please use the newer Step *Validate Poses by Included Angle to Reference Direction*. This Step can only calculate the angle between the Z-axis and a reference direction. The newer Step supports filtering poses by calculating the angle between any axes and a reference direction.

Input and Output



4.33.18 Normal Estimation

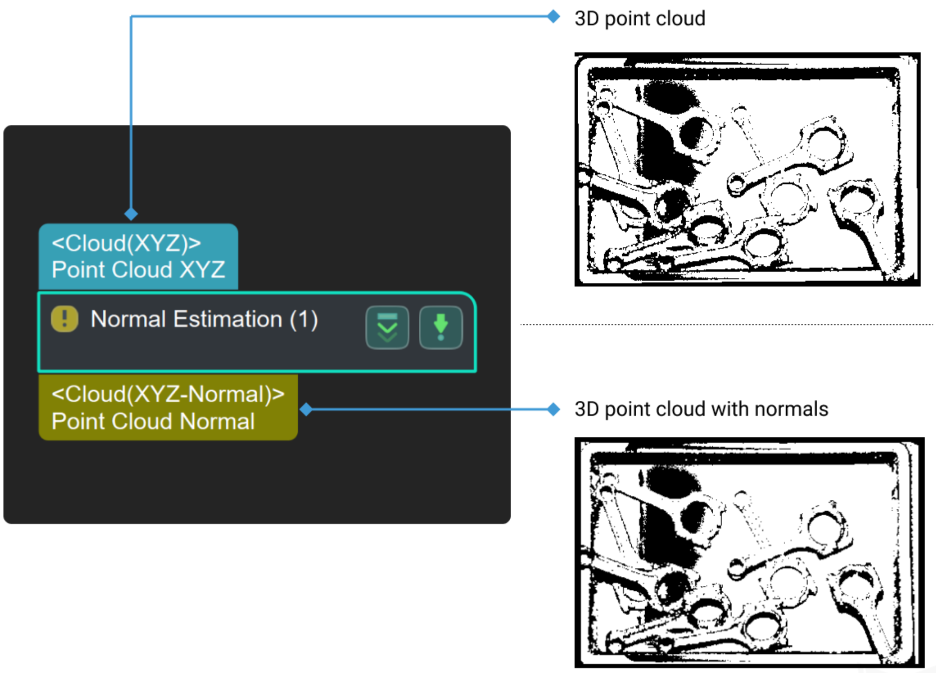
Function

Compute point cloud normal vectors.

Sample Scenario

This Step is outdated. Please use the **NormalEstimation** method in the Step *Calc Normals of Point Cloud and Filter It*.

Input and Output



CHAPTER

FIVE

COMMON PROCEDURES

This chapter introduces commonly used procedures that are used in projects to achieve different functions by connecting different steps. The Mech-Vision software stores the commonly used procedures in a **custom** directory in the step library, which can be used according to actual needs of the project.

To package multiple steps, please see *Procedure* .

5.1 2D Rectangle Matching(Deep Learning)

2D Rectangle Matching(Deep Learning) is capable of calculating locations and pick points of jointed and homogeneous cubes with complex textures. Inputs：original colored image, original depth image, masks from instance segmentation, the mask of the higest layer, original point cloud and actual size of the object Prior information of deep learning can provide the data of approximate location of objects. The main process includes 4 parts: generating rectangle edge templates, detecting lines in the image corresponding to the mask, 2D matching and calculating sizes and poses by correct rectangle information. as *Figure 1.* shown.

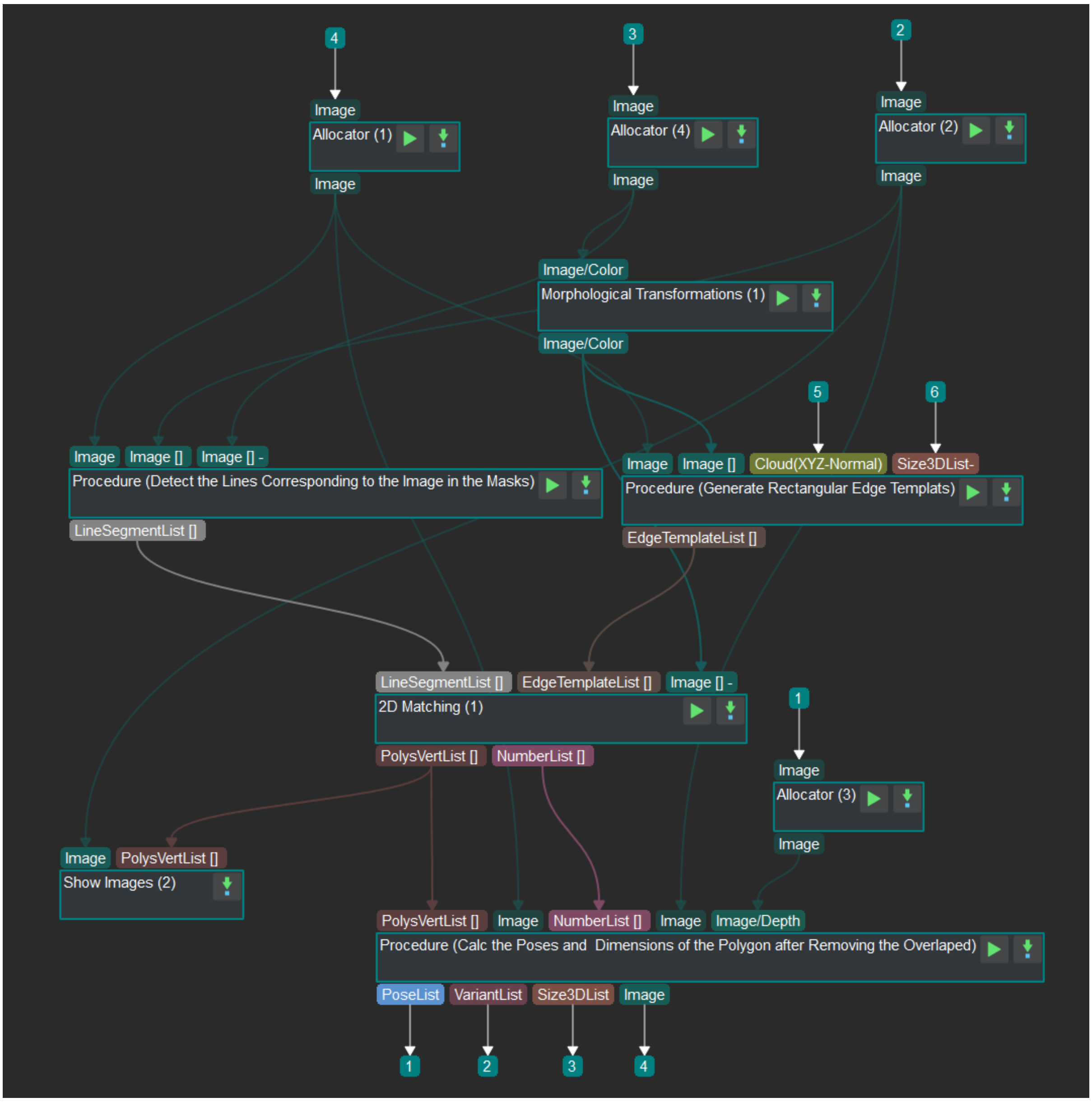


Figure 1. Procedure of 2D rectangle matching(Deep Learning)

5.1.1 Generating Rectangle Edge Templates

The procedure of generating rectangle edge templates is as shown in *Figure 2.*.

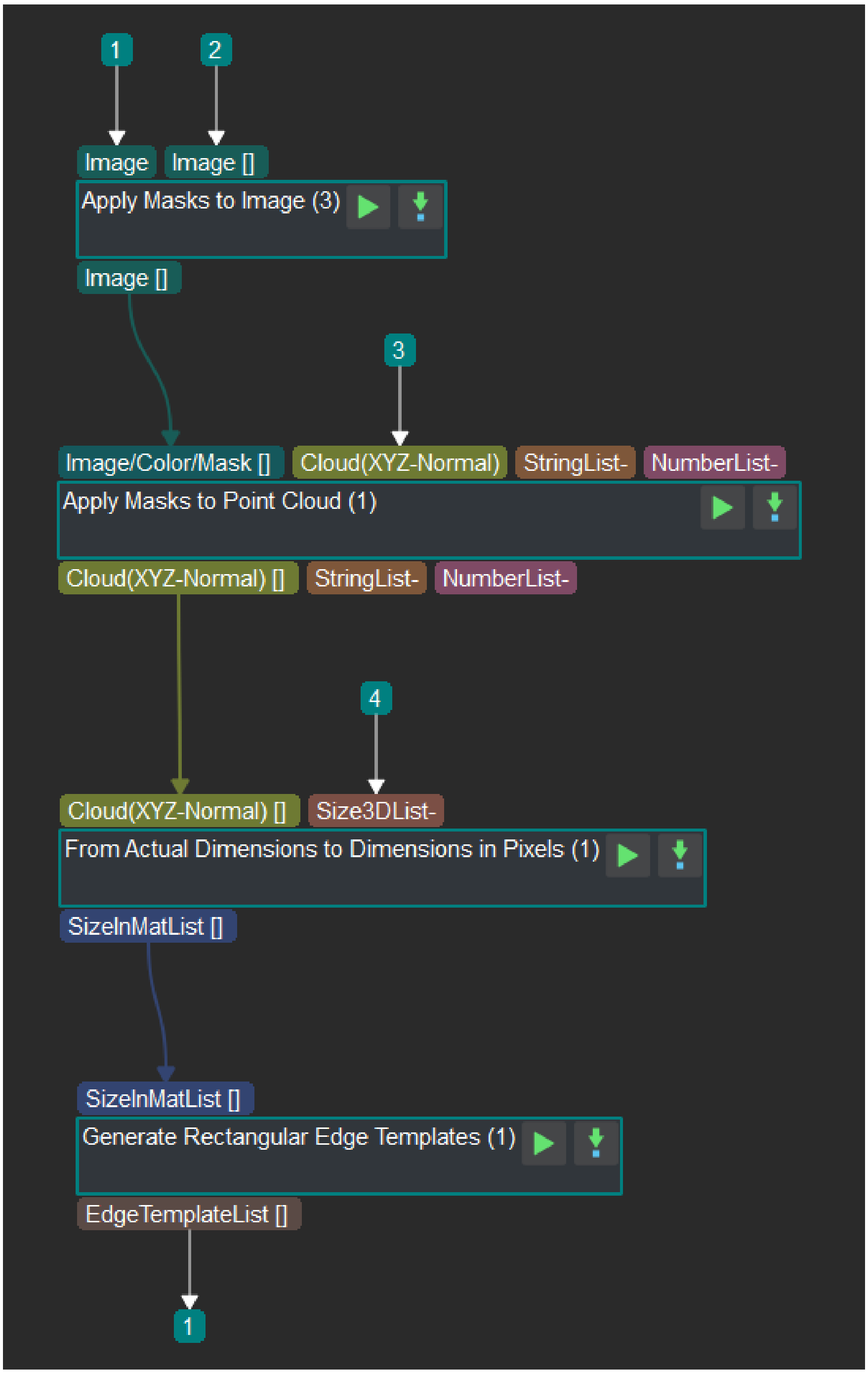


Figure 2. Procedure of generating rectangle edge templates

The introduction of each steps is as follows:

1. *Apply Masks to Image* Get the mask of the object after instance segmentation by calculating overlaps between the mask of the highest layer and the mask after instance segmentation.
2. *Apply Masks to Point Cloud* Get the point cloud of the object after instance segmentation by processing the object masks after instance segmentation and the original point cloud.
3. *From Actual Dimensions to Dimensions in Pixels* Calculate the size of the object(in pixel) with the point cloud after instance segmentation and the actual size of the object.
4. *Generate Rectangular Edge Templates of Specified Sizes* Generate the object surface(rectangle) edge template with the size of object(in pixel).

5.1.2 Detecting Lines In the Image

Detecting lines in the image corresponding to the mask is as *Figure 3.* shown.

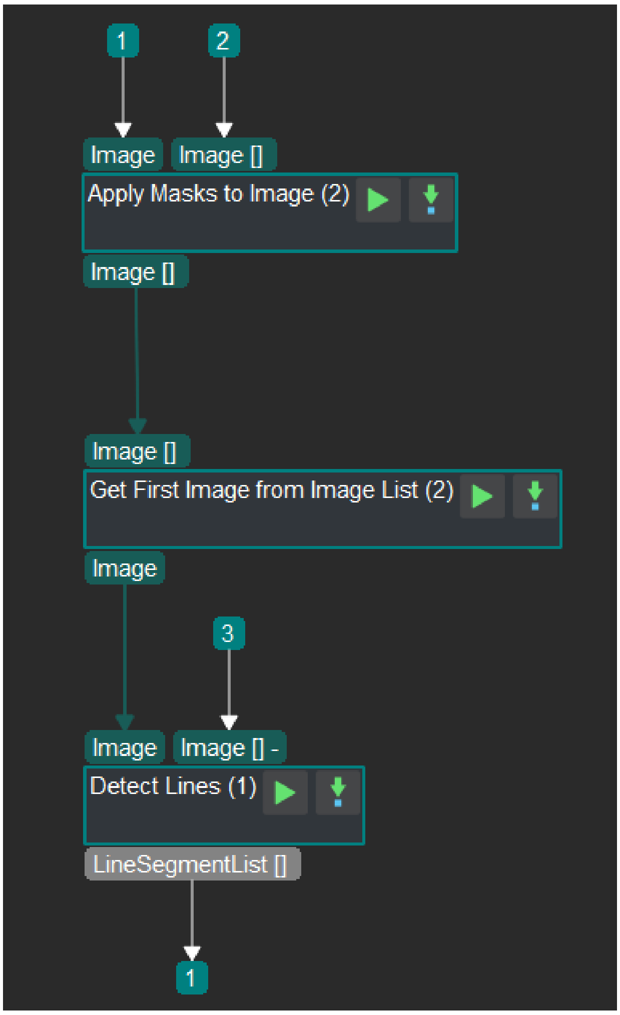


Figure 3. Procedure of detecting lines in the image corresponding to the mask

The introduction of each steps is as follows:

1. *Apply Masks to Image* Obtain the object mask of the higest layer by calculating overlaps between the mask of the higest layer and the original colored image.
2. *Get First Image from Image List* Get the image of the result corresponding to the first mask image.
3. *Detect Line Segments* Detect lines in segmented objects by processing the colored image corresponding to the mask and the mask generated by instance segmentation.

5.1.3 2D Matching

By matching detected lines with generated rectangle edge templates, Mech-Vision would output polygons consisted of intact lines and also polygon matching loss.

5.1.4 Calculating sizes and Poses

Calculating sizes and poses by correct rectangle information is as *Figure 4.* shows.

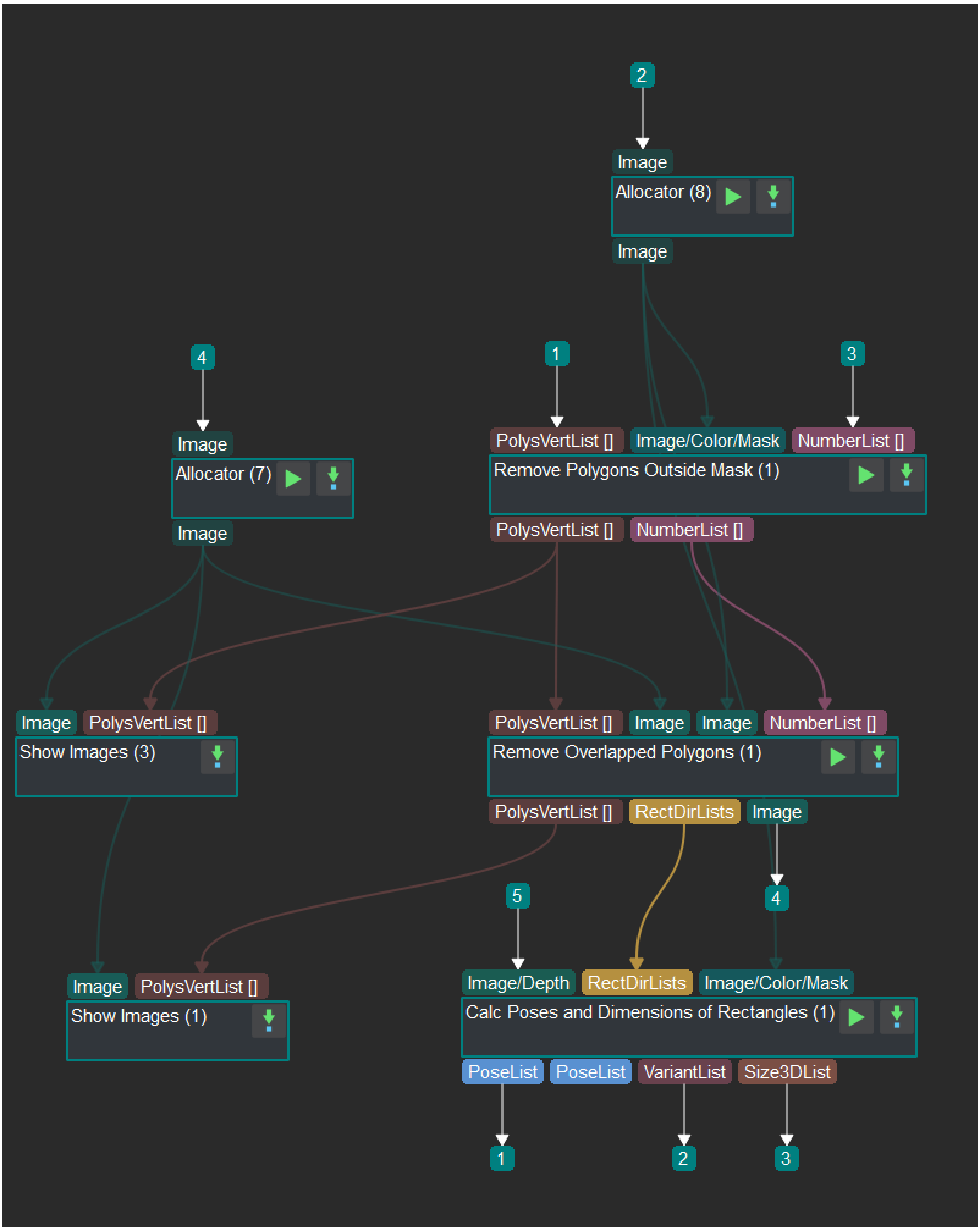


Figure 4. Calculating sizes and poses by correct rectangle information

The introduction of each steps is as follows:

1. *Remove Polygons outside Mask* Based on the 2D matching result and the mask of the higest layer, the polygons whose overlapping areas with the mask are smaller than the threshold would be removed. This step is to make sure that the object corresponding to the polygon is on the top layer.
2. *Remove Overlapped Polygons* The inputs are the result of remove_polygons_outside_mask, the original colored image and the mask of the highest object. This step is to remove overlapped polygons or polygons outside the mask. Make sure the object corresponding to the polygon is not overlapped.
3. *Calc Poses and Dimensions of Rectangles* This step needs the original depth image, the rectangle information after filtering and the mask of the higest layer. Pick points will be calculated with the depth information and rectangle vertexes.

Typical Application: *2D Matching*

5.2 2D Rectangle Matching(Non-deep Learning)

2D Rectangle Matching(Non-deep Learning) is capable of calculating locations and pick points of jointed and homogeneous cubes with complex textures and clear boundaries. Inputs：original colored image, original depth image, the mask of the higest layer, original point cloud and actual size of the object. The main process includes 4 parts: generating rectangle edge templates, detecting lines in the image corresponding to the mask, 2D matching and calculating sizes and poses by correct rectangle information. As *Figure 1.* shows.

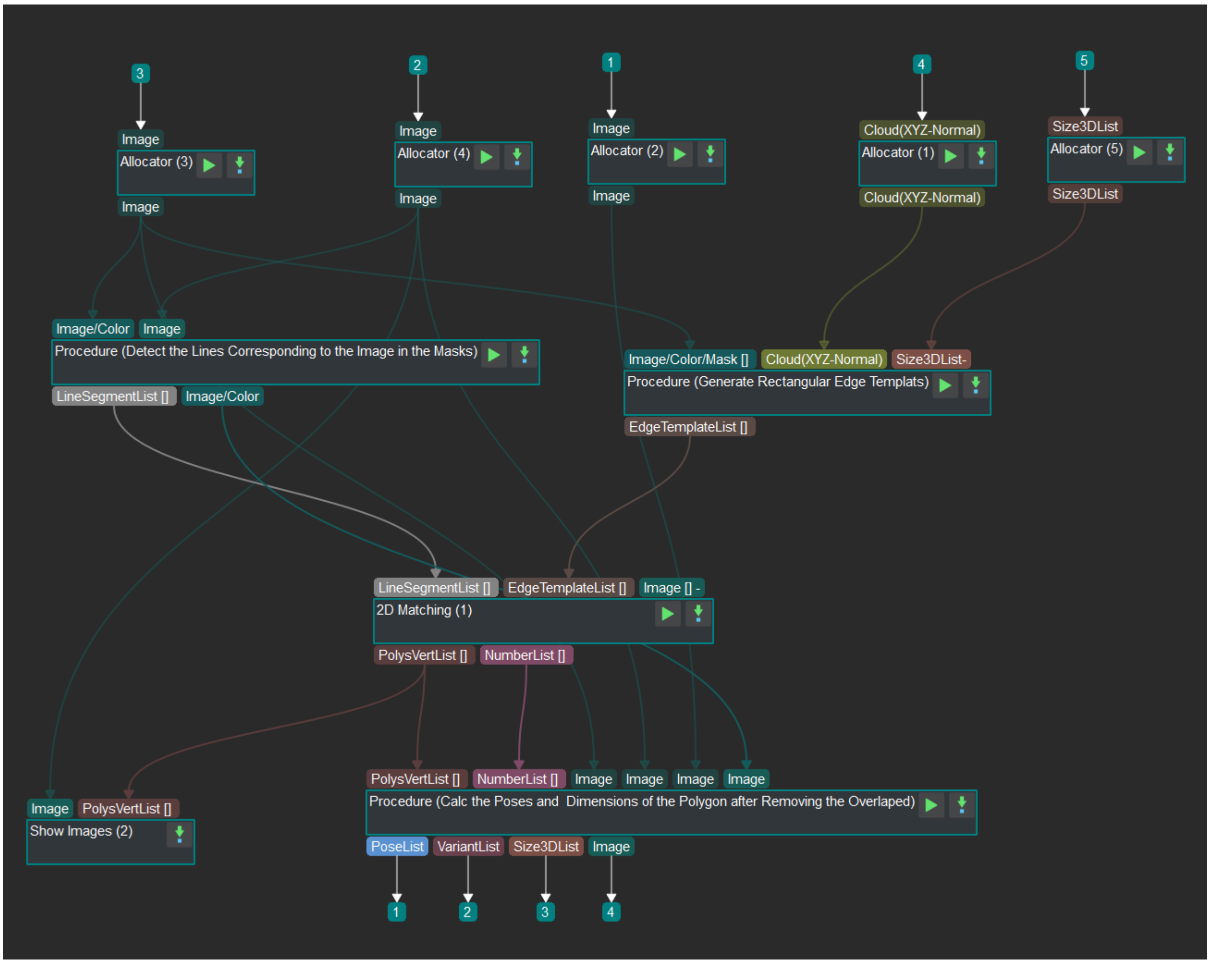


Figure 1. Procedure of 2D Rectangle Matching(Non-deep Learning)

5.2.1 Generating Rectangle Edge Templates

The procedure of generating rectangle edge templates is as shown in *Figure 2.*.

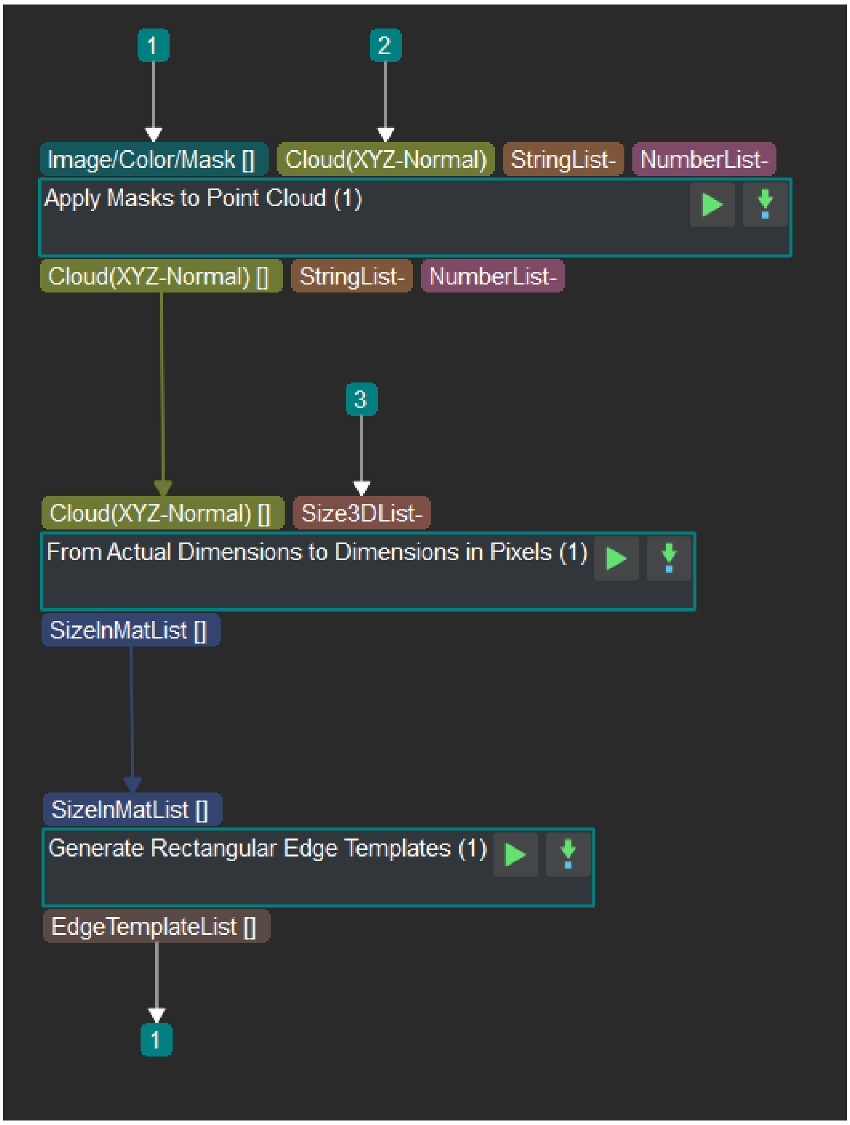


Figure 2. Procedure of generating rectangle edge templates

The introduction of each steps is as follows:

1. *Apply Masks to Point Cloud* Get the point cloud of the object after instance segmentation by processing the object masks after instance segmentation and the original point cloud.
2. *From Actual Dimensions to Dimensions in Pixels* Calculate the size of the object(in pixel) with the point cloud after instance segmentation and the actual size of the object.
3. *Generate Rectangular Edge Templates of Specified Sizes* Generate the object surface(rectangle) edge template with the size of object(in pixel).

5.2.2 Detecting Lines In the Image

Detecting lines in the image corresponding to the mask is as *Figure 3.* shown.

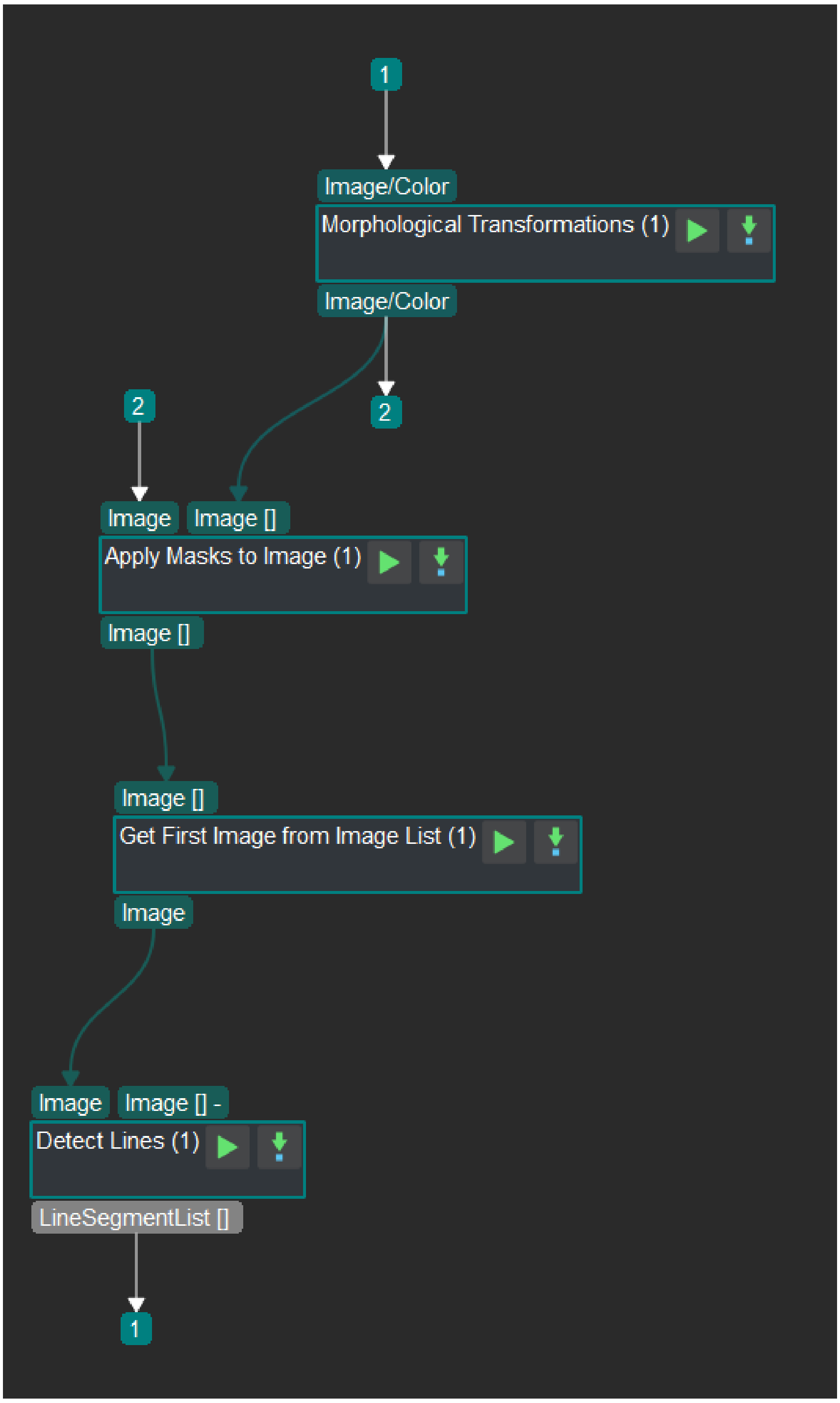


Figure 3. Procedure of detecting lines in the image corresponding to the mask

The introduction of each steps is as follows:

1. *Apply Masks to Image* Obtain the object mask of the higest layer by calculating overlaps between the mask of the higest layer and the original colored image.
2. *Get First Image from Image List* Get the image of the result corresponding to the first mask image.
3. *Detect Line Segments* Detect lines in the image by processing the colored image corresponding to the mask.

5.2.3 2D Matching

By matching detected lines with generated rectangle edge templates, Mech-Vision would output polygons consisted of intact lines and also polygon matching loss.

5.2.4 Calculating sizes and Poses

Calculating sizes and poses by correct rectangle information is as *Figure 4.* shows.

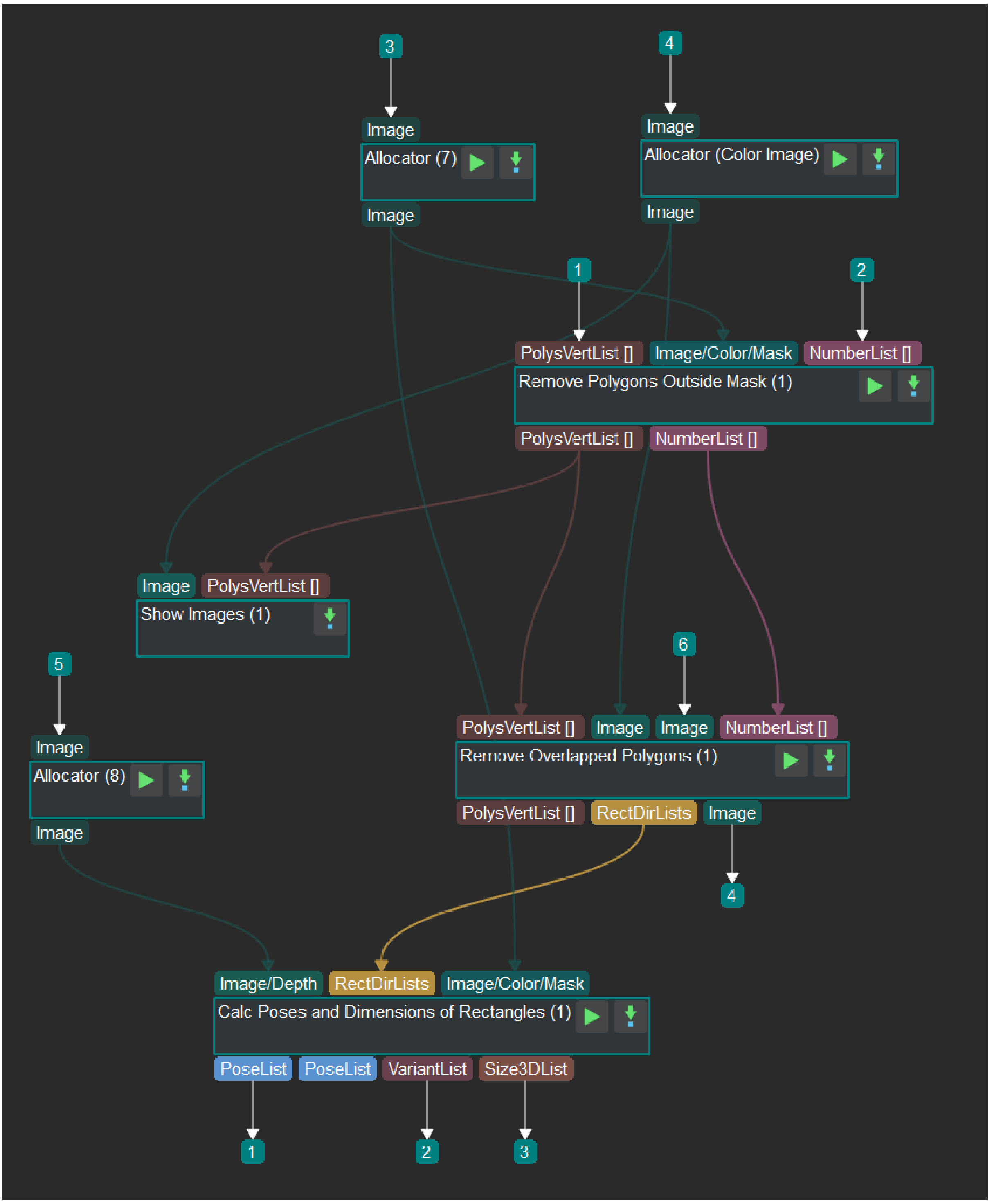


Figure 4. Calculating sizes and poses by correct rectangle information

The introduction of each steps is as follows:

1. *Remove Polygons outside Mask* Based on the 2D matching result and the mask of the higest layer,

the polygons whose overlapping areas with the mask are smaller than the threshold would be removed. This step is to make sure that the object corresponding to the polygon is on the top layer.

2. *Remove Overlapped Polygons* The inputs are the result of remove_polygons_outside_mask, the original colored image and the mask of the highest object. This step is to remove overlapped polygons or polygons outside the mask. Make sure the object corresponding to the polygon is not overlapped.

3. *Calc Poses and Dimensions of Rectangles* This step needs the original depth image, the rectangle information after filtering and the mask of the higest layer. Pick points will be calculated with the depth information and rectangle vertexes.

Typical Application: *2D Matching*

5.3 2D Sorting

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:

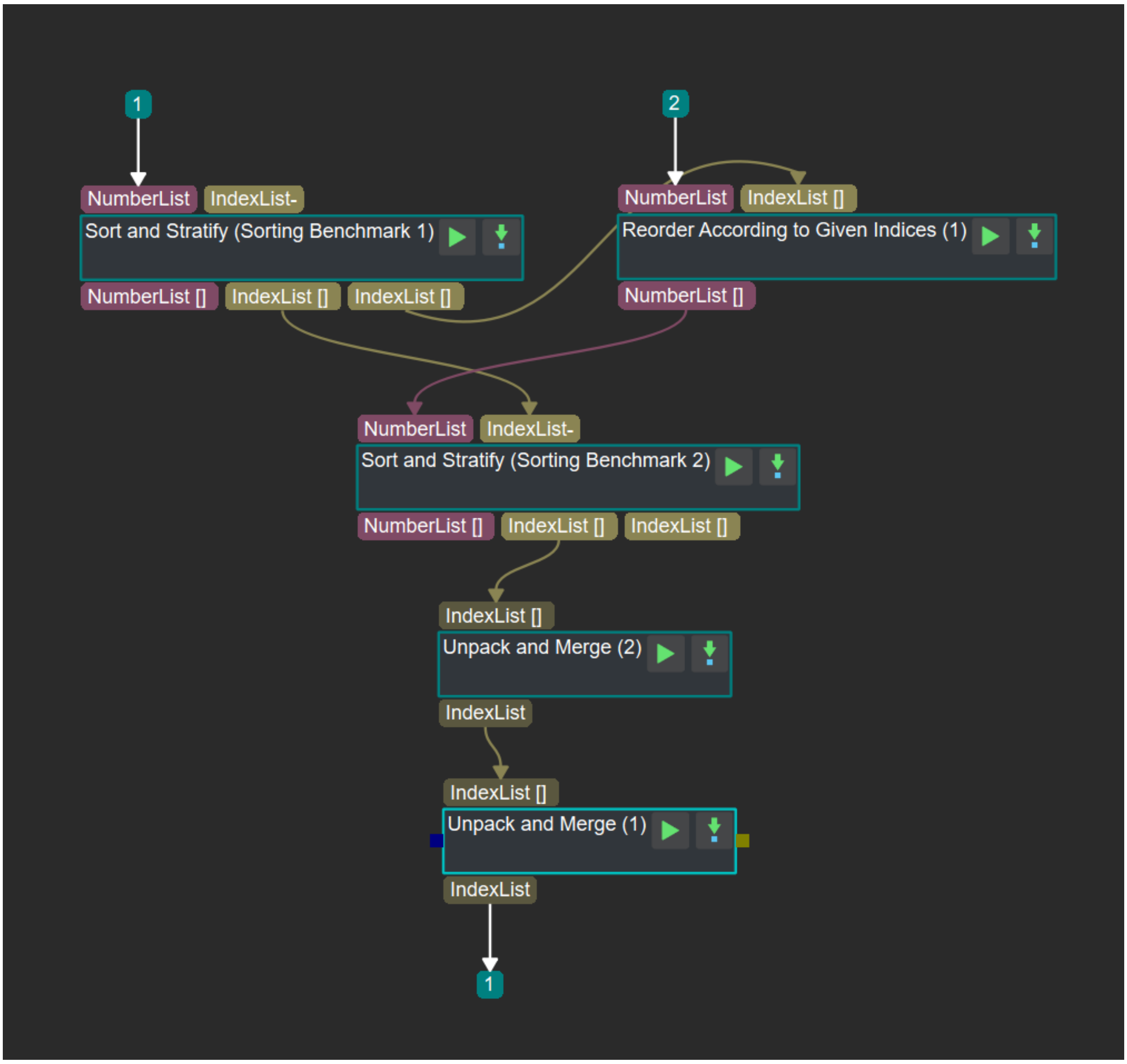


Figure 1. Procedure of 2D sorting

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. *ccording 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 are generated.
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.

5.4 3D Matching (High Precision)

3D matching (high precision) is suitable for projects that need the accurate position of objects. This procedure uses *3D Coarse Matching* and *3D Fine Matching* which are two calculation methods of the 3D vision processing of matching the workpiece template and the point cloud of the scene to calculate the rough position of the workpiece. And finally obtain more precise pick points by optimizing the matching result . The step procedure is shown in *Figure 1*.

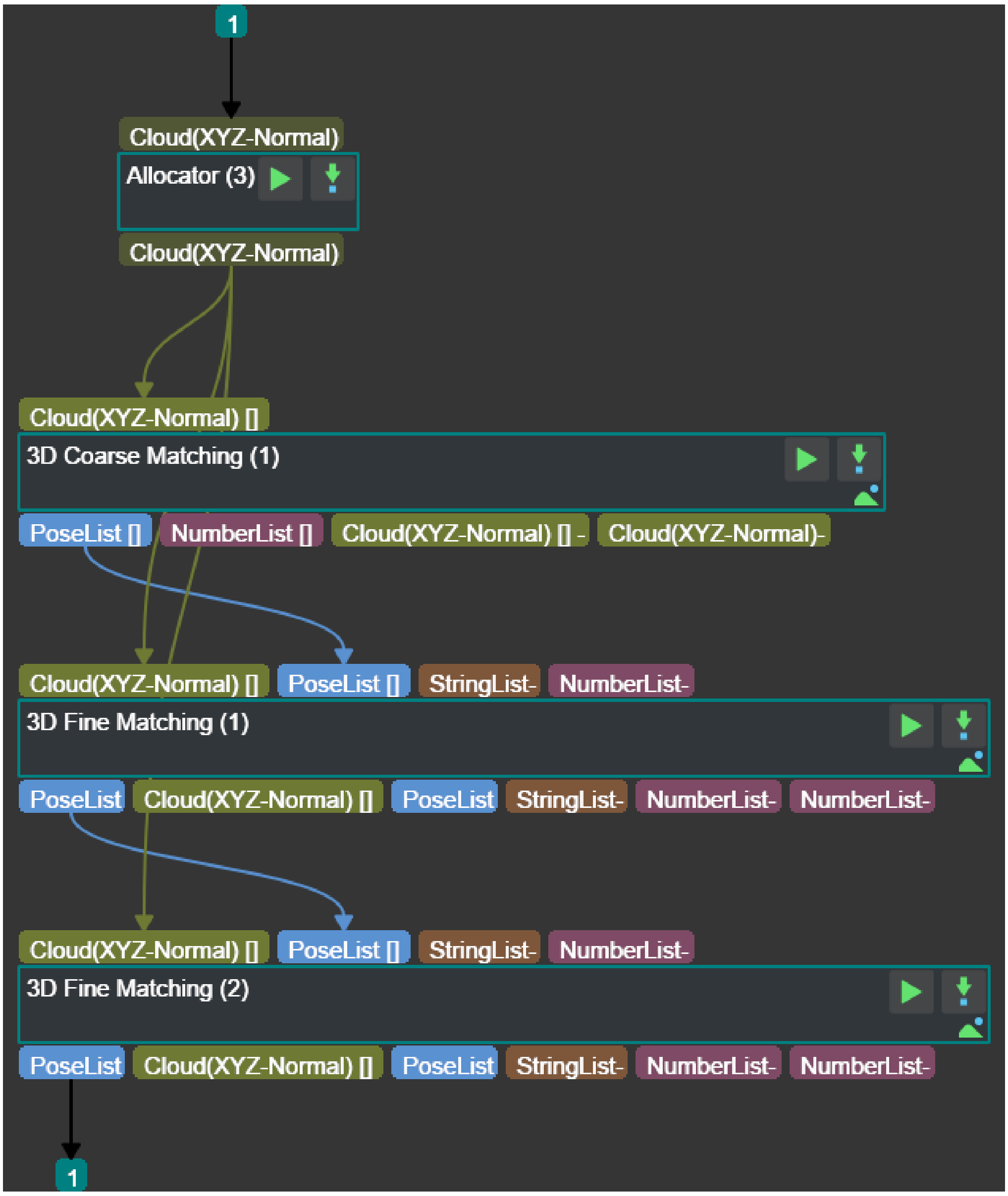


Figure 1 The procedure of 3D matching (high precision)

The role of each step is as follows:

1. *Allocator* Allocate the input point cloud with normal vector to the subsequent steps.
2. *3D Coarse Matching* Calculate the initial pick points of the object based on the input point cloud with normal vector and the template file in the setting. Then output the result.
3. *3D Fine Matching* Based on the input point cloud with normal vector, the initial pick point list from last Step, and the template file in the settings, calculate the more precise pick points of the object and output the result.
4. *3D Fine Matching* Based on the input point cloud with normal vector, the result of the previous step, and the template file in the settings, further optimize the pick points and output a more precise pick point list.

5.5 3D Matching

The main function of 3D matching is to detect objects in the scene and obtain their poses by point cloud template matching. This step usually follows the step *Filter Out Point Clouds That Exceed The Limit*. The combination is as *Figure 1.* shown.

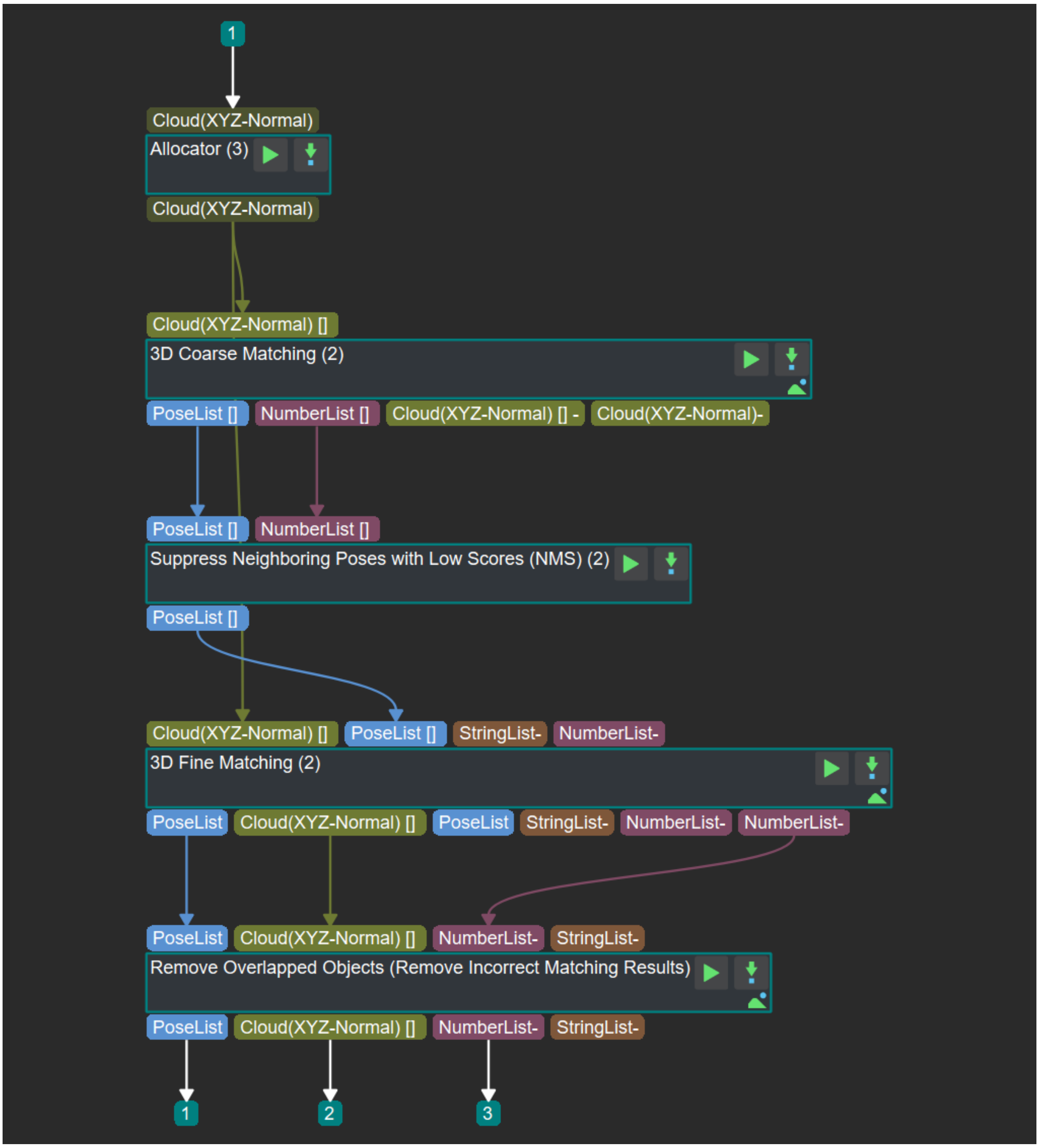


Figure 1. Procedure of 3D matching

Firstly, input the filtered point cloud into this combination, then users would get one or multiple initial poses by *3D Coarse Matching*. Then these poses are processed and the most reasonable one is selected using *Suppress Neighboring Poses with Low Scores (NMS)*. Finally, users could obtain the accurate pose by processing the filtered pose and point cloud with *3D Fine Matching*.

After selecting the object pose in horizontal direction, then *Remove Overlapped Objects* follows, which remove overlapping objects in the Z direction. Finally, the pose suitable for grabbing on the upper layer is selected.

5.6 3D Sorting

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:

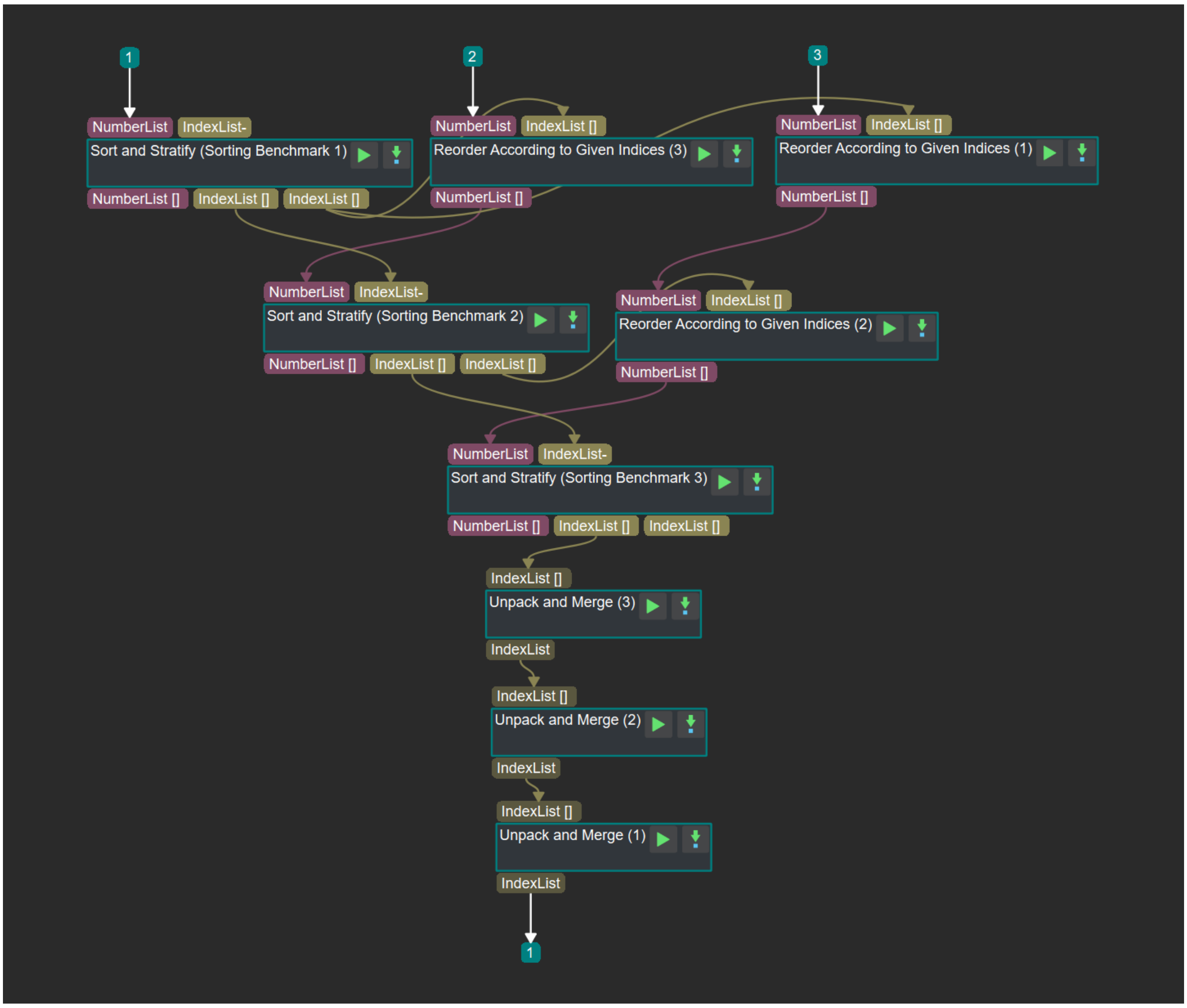


Figure 1. Procedure of 3D sorting

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. *ccording 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. *ccording 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. *ccording 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 "Unpack And Merge Data" twice, this step could turn the two-dimensional array-like variables into a one-dimensional array as the final output.

5.7 3D Boundary Extraction

The main function of 3D boundary extraction is to remove the features inside the object, and only keep the boundary of the object. The combination is as *Figure 1* shown.

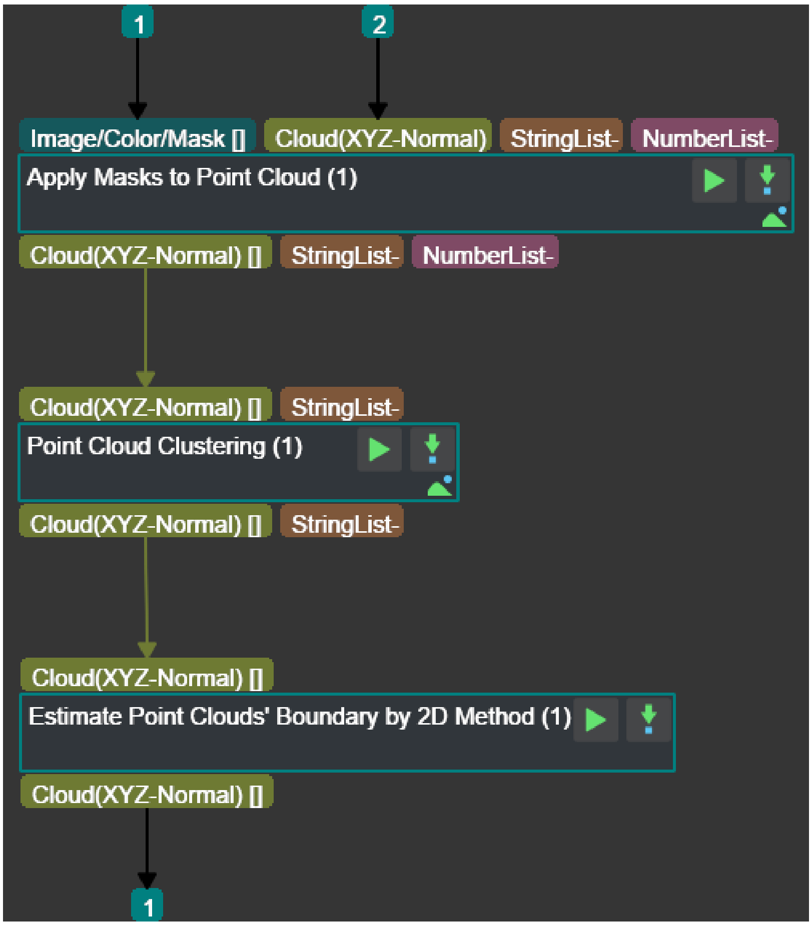


Figure 1 Procedure of 3D boundary extraction

The role of each Step is as follows:

1. *Apply Masks to Point Cloud* Take the point clouds of workpieces and masks as input to obtain the point cloud in the area of masks.
2. *Point Cloud Clustering* Take point cloud with normal vector as input, filter out some useless small point cloud or large background through point cloud clustering. Prepare for the subsequent boundary estimation of the point cloud.
3. *Estimate Point Cloud Edges by 2D Method* Take the point clouds after clustering as input, and estimate point cloud edges by 2D method.

5.8 Apply Masks to Color Image

5.8.1 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 as shown in *Figure 1*.

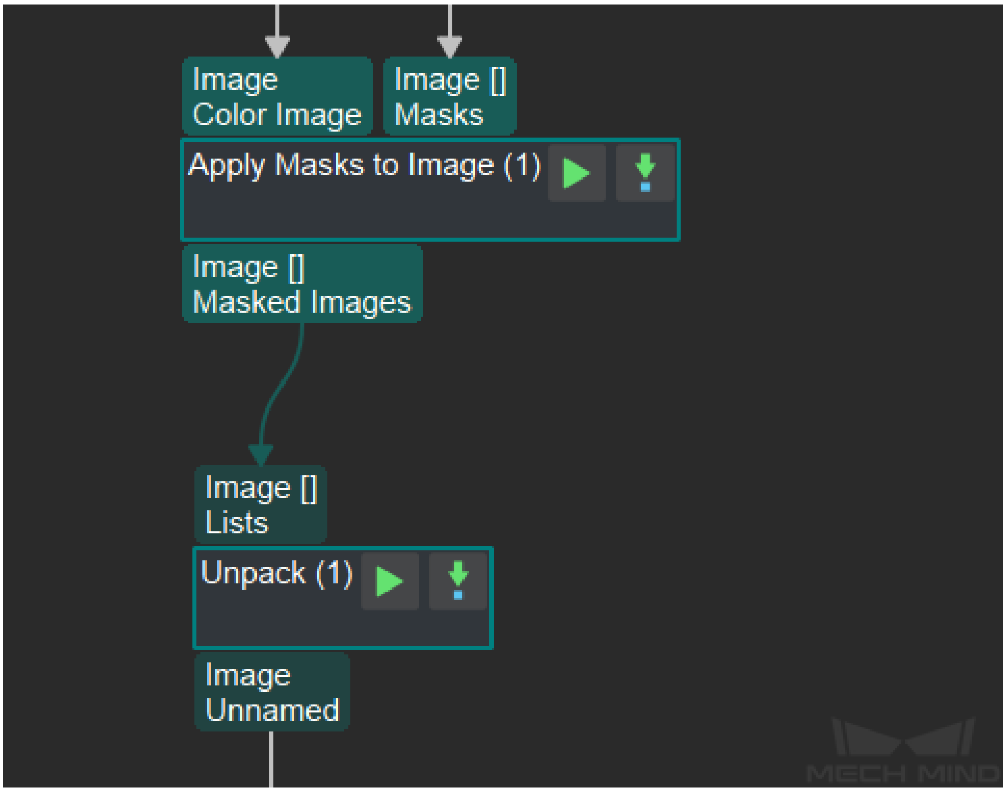


Figure 1. Graphical programming of **Apply Masks to Color Image**

5.8.2 Functions of Involved Steps

The functions of the Steps are as follows (please click on the link for detailed descriptions):

1. *Apply Masks to Image* obtains the image parts corresponding to the masks in the mask list.
2. Obtains the first image from the output of *Apply Masks to Image*.

5.8.3 Sample 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.

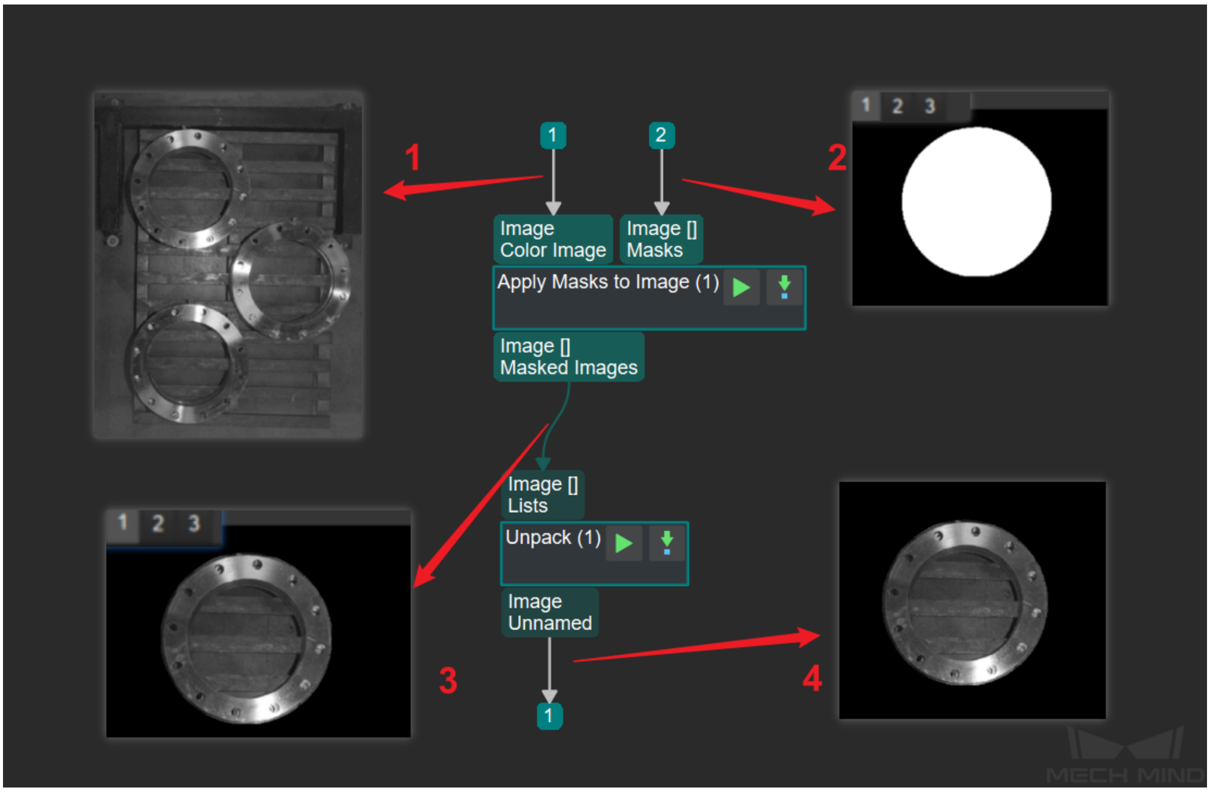


Figure 3. Data flow of the sample scenario

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.

5.8.4 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.

5.9 Calc Color Image for Highest Layer

5.9.1 Function

This Procedure defines the image area of the highest layer in the scene based on the depth map and then outputs the color image and the mask of the area.

- Input: depth map, color image
- Output: the color image of the highest layer, the merged mask of the highest layer

Note: the highest layer may include many segments and the masks of those segments will be merged into one before outputting.

The graphical programming of this Procedure is as shown in *Figure 1*.

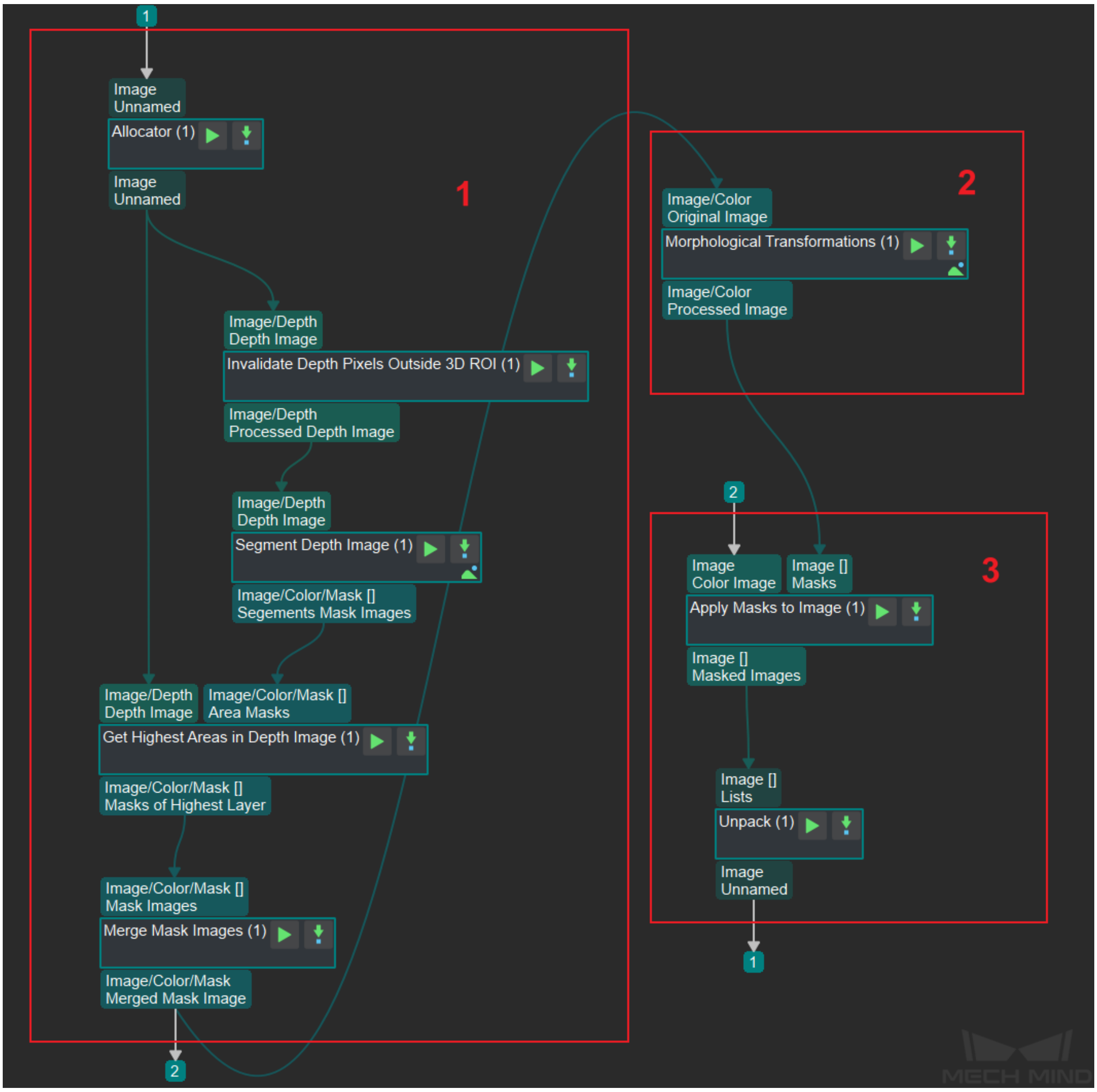


Figure 1. Graphical programming of **Calc Color Image for Highest Layer**

This Procedure consists of three sub-workflows:

1. Get the mask of the highest layer.
2. Morphological transformations.
3. Get the color image segment under the mask.

5.9.2 Functions of Involved Steps

The functions of the Steps involved in this Procedure are as follows (please click on the links for detailed descriptions of the Steps):

1. *Invalidate Depth Pixels Outside 3D ROI* invalidates the pixels out of the ROI in a depth map and outputs the depth map within the ROI.
2. *Segment Depth Image* segments the depth map within the ROI based on each pixel's depth and generates the corresponding mask(s).
3. *Get Highest Areas in Depth Image* obtains the highest area(s) in a depth map based on the masks and masks' depth information output from *Segment Depth Image*.
4. *Merge Mask Images* merges the masks input from *Get Highest Areas in Depth Image* if the input contains more than one mask, to facilitate the morphological transformations later.
5. *Morphological Transformations* dilates the mask(s) to facilitate recognition or picking later.
6. *Apply Masks to Image* obtains the color image segment(s) under the transformed mask(s).
7. *Unpack Data* obtains the color image of the first index in the list of image segments under masks.

5.9.3 Sample Scenario

This Procedure can extract the color image segment of the highest layer for deep learning recognition.

- The parameter tuning is as shown in *Figure 2*.

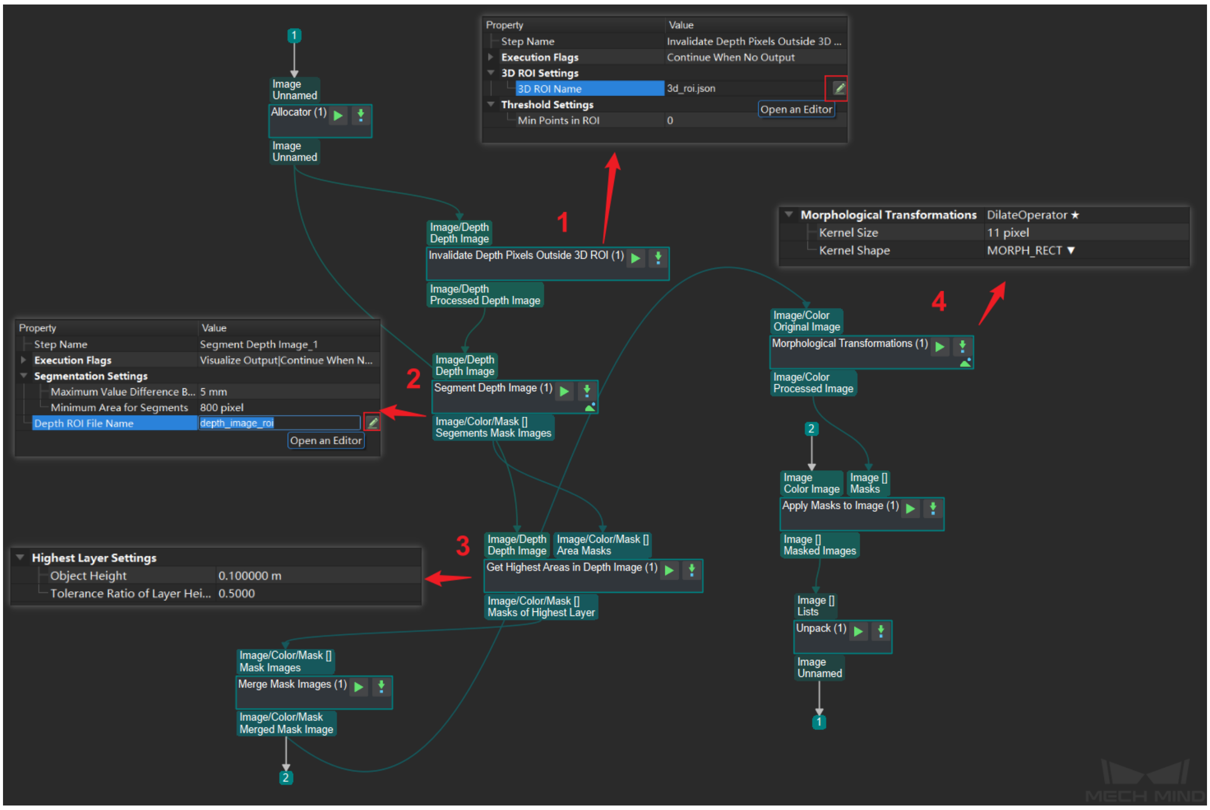


Figure 2. Parameter tuning

1. Click to pop up the window of setting the point cloud ROI.
2. Click to pop up the window of setting the depth map ROI.
3. Set parameters to appropriate values to obtain the highest layer in the depth map.
4. Set parameters to appropriate values to obtain the mask of the highest layer.

Note: Please see the corresponding descriptions in :ref: *steps_guide* for details about each Step.

- *Figure 3* is a comparison of the input and the output:

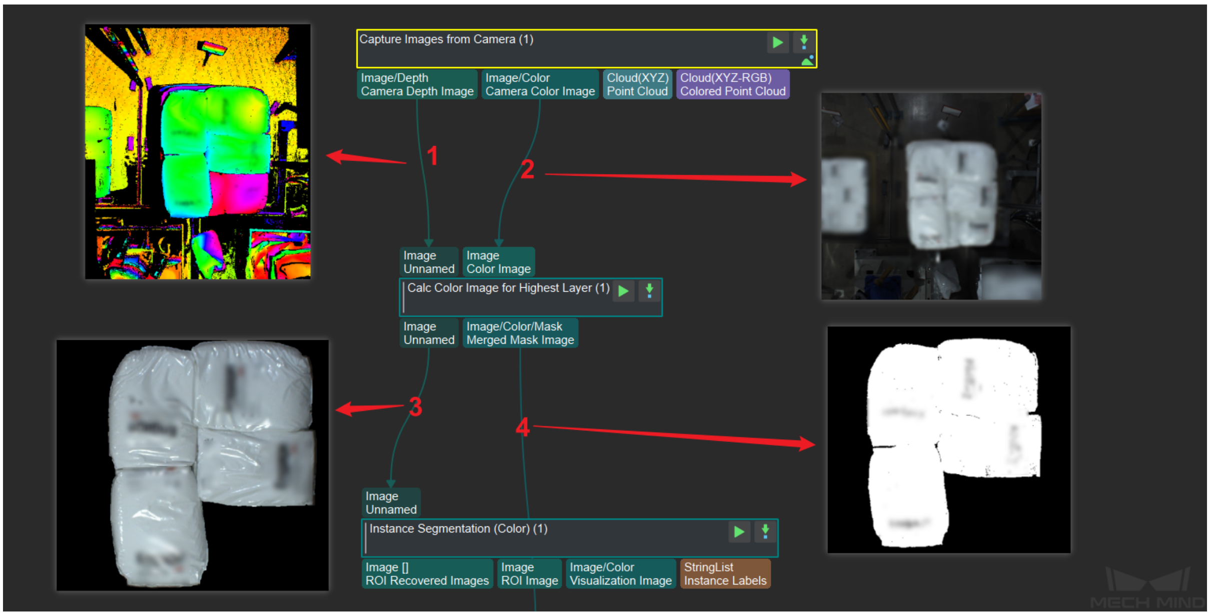


Figure 3. Comparison of the input and the output

1. The input depth map.
2. The input color image.
3. The obtained color image of the highest layer.
4. The merged mask of the highest layer.

Comparison of Similar Procedures

- *Get Mask of the Highest Layer* obtains the mask of the highest layer.
- *Calc Color Image for Highest Layer* obtains the mask and the color image of the highest layer.

5.10 Get Mask of the Highest Layer

The main function of this combination is to remove invalid regions and then seperate the depth image and getting the mask of the highest layer. Usually this combination is applied in the case that the depth situation is quite complex and should be seperated into layers. The combination is as *Figure 1.* shown.

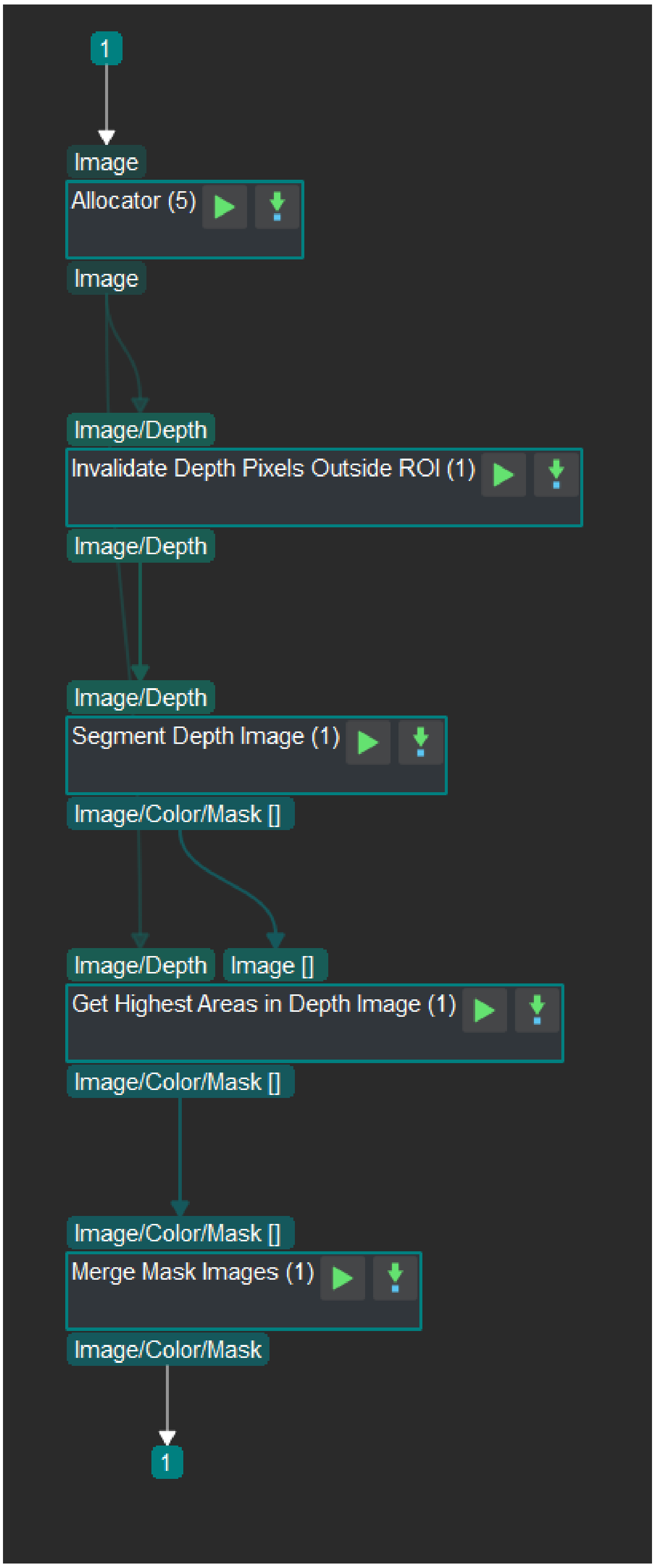


Figure 1. Procedure of getting the mask of the highest layer

The introduction of each steps is as follows:

1. *Invalidate Depth Pixels Outside 3D ROI* Filter the depth image and only remain regions of interest to remove useless data and accelerate the process.
2. *Segment Depth Image* Seperate the depth image and filter out points of noise.
3. *Get Highest Areas in Depth Image* Generate the mask of the closest valid region to the camera.
4. *Merge Mask Images* Combine multiple mask images closest to the camera at the same layer into one.

5.11 Calculate the Grasping Pose and Size of the Object

The main function of this procedure is using the target point cloud to calculate the grasping pose of it and size of the object. This procedure is usually used in depalletizing. The procedure is shown in *Figure 1*.

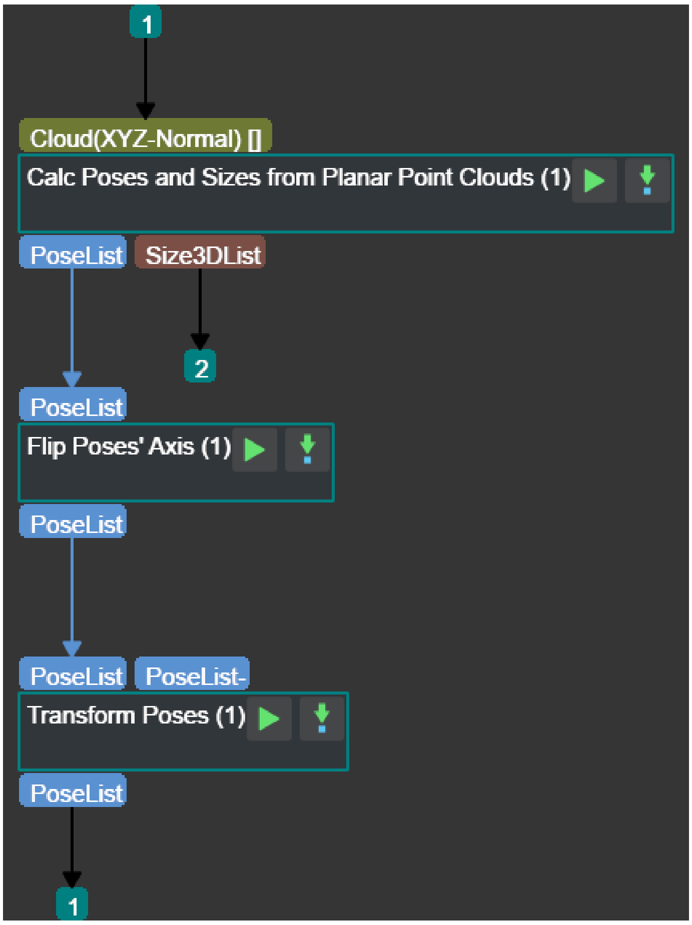


Figure 1 Procedure of calculating the grasping pose and size of the object

The role of each step is as follows:

1. *Calc Poses and Dimensions from Planar Point Clouds* Take the point clouds as input to calculate the poses and sizes of planar point clouds.
2. *Flip Poses' Axes* Adjust the coordinate axis for inappropriate poses.
3. *Transform Point Clouds* Transform the grasping pose from the camera coordinate system to the robot coordinate system (you can also customize the coordinate system transformation).

5.12 Roughly Calculate Pick Points according to the Heat Map of Grasp Probability

Take the heat map of grasp probability, the original depth image, the color image and the point cloud in the region of interest as input, calculate and filter the plane pick points. The step procedure is shown in *Figure 1*.

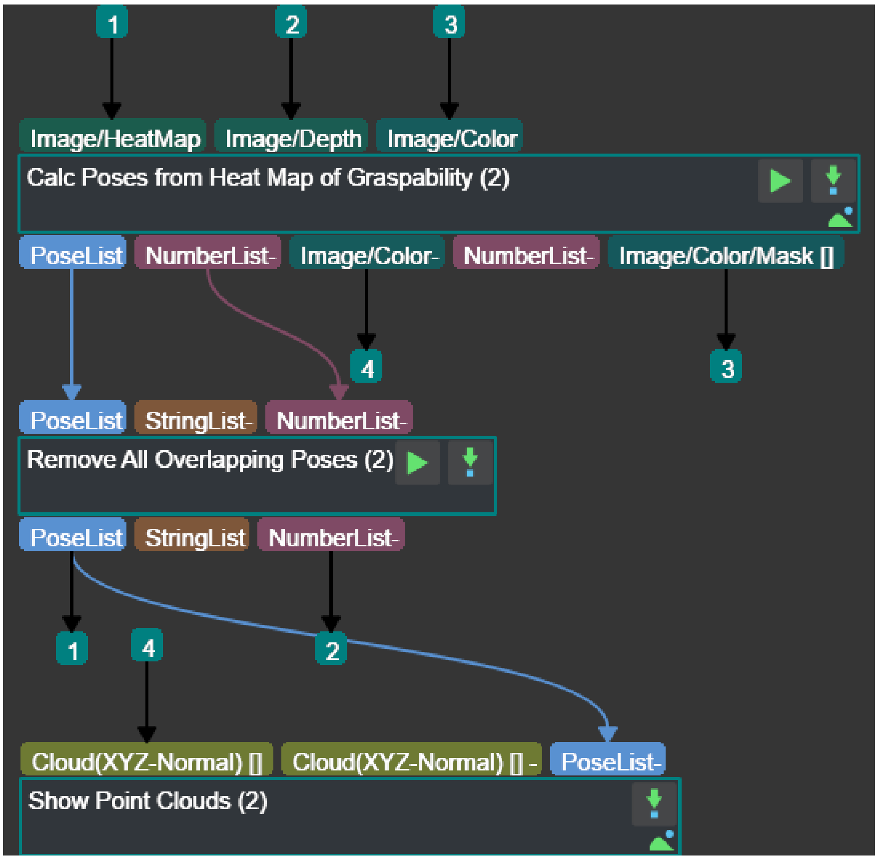


Figure 1 step procedure of calculating coarse pick points according to the heat map of grasp probability

The role of each step is as follows:

1. *Calc Poses From Heat Map Of Graspability* Based on the heat map of grasp probability, the original depth image and the color image, output the plane pick points that meet the requirements and the corresponding confidence. Mark the color image corresponding to the previous result and the graspable area of each pose.

2. *Remove All Overlapping Poses* Based on the plane pick points and their scores obtained from the previous step, remove the points that are too close and output the planar pick points that meet the requirements.

3. *Show Point Clouds and Poses* Displays the filtered plane pick points and the corresponding confidence on the color image.

5.13 Accurately Calculate Pick Points According to Poses

Take the point clouds with normal vector and the plane pose list as input to calculate the 3D pick points. The procedure is shown in *Figure 1*.

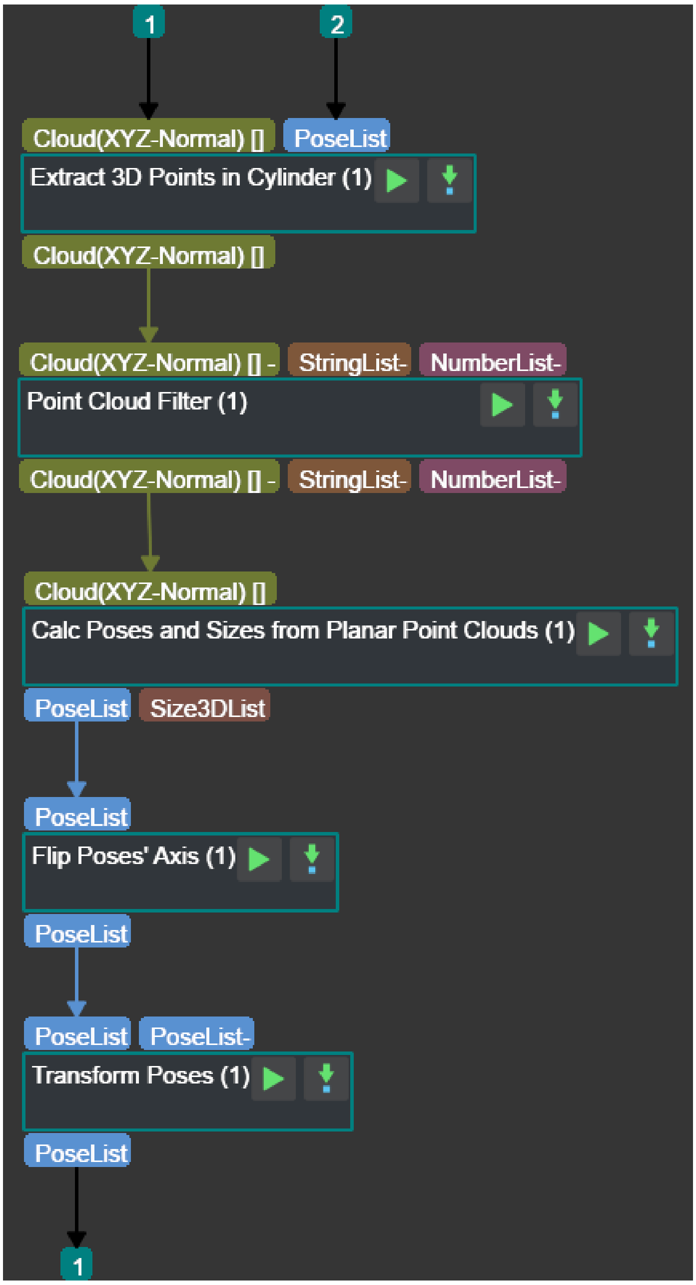


Figure 1 Procedure of calculating fine pick points according to coarse ones

The role of each step is as follows:

1. *Extract 3D Points in Cylinder* Take the point clouds with normal vector and the plane pick point list as input to extract the point cloud in the set cylinder.
2. *Point Cloud Filter* Take the point clouds which extracted in the previous step as input to remove the useless points according to the set parameters, and output the filtered point cloud.
3. *Calc Poses and Dimensions from Planar Point Clouds* Take the filtered point cloud as input to calculate the corresponding pick points of each point cloud and output the 3D pick point list.

4. *Flip Poses' Axes* Rotate the point clouds obtained from the previous Step to make their directions roughly the same to facilitate subsequent picking. Then output the flipped pick point list.
5. *Transform Poses* Input the result of the previous step. Transform the coordinate system of the pick points.

5.14 Calculate Pick Points According To The Heat Map Of Grasp Probability

Input the heat map of grasp probability of the pick points, the original depth image and the color image, calculate the planar pick points and assign the gripper (this project use a suction cup) label for it. The program of this procedure is shown in *Figure 1*.

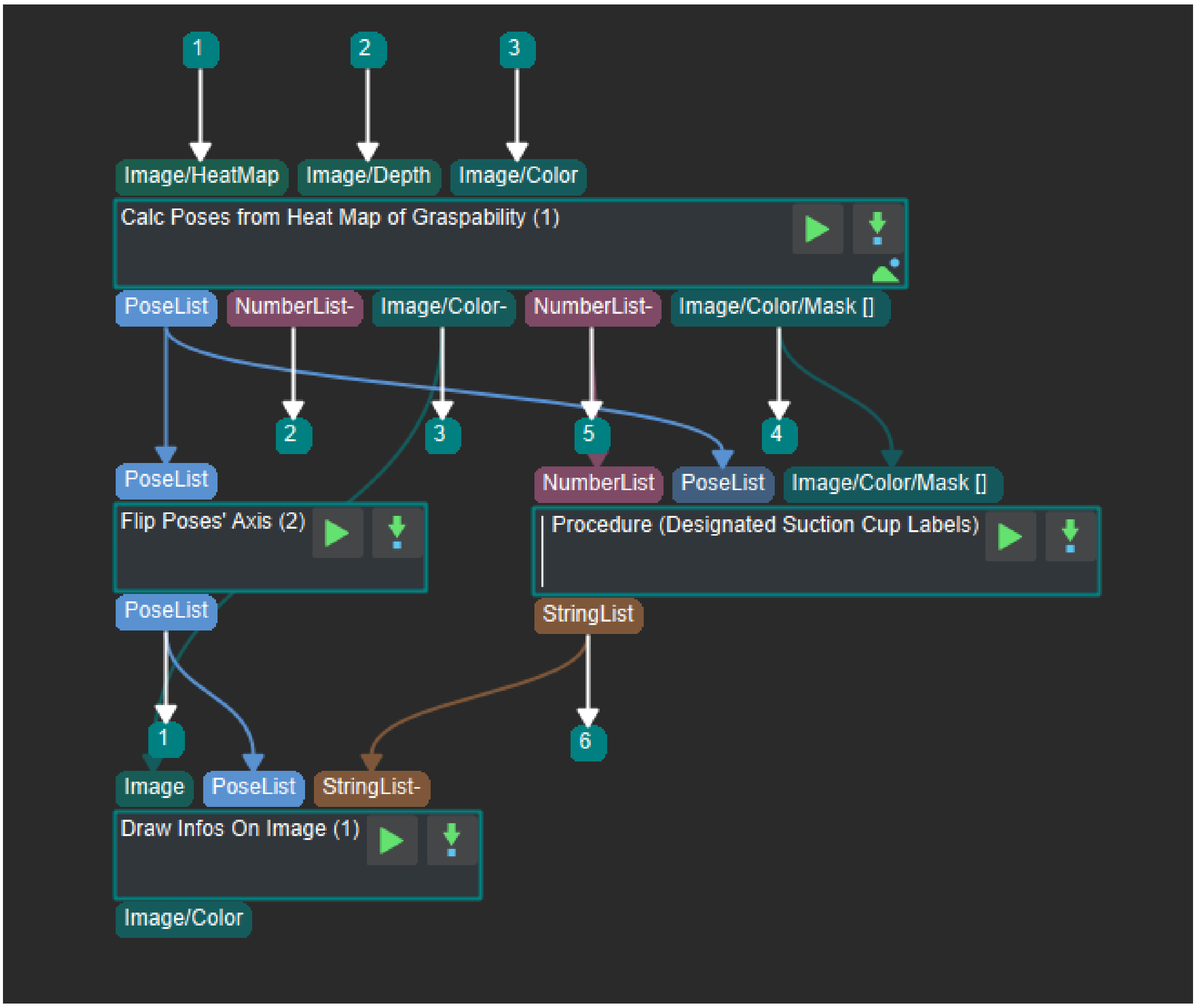


Figure 1 The program of calculating pick points (small value range) according to the heat map of grasp probability in Mech-Vision

The introduction of each step in the program ia as follows :

1. *Calc Poses From Heat Map Of Graspability* : Input the heat map of grasp probability, the original depth image and the color image, output the planar pick points which meet the requirement and the corresponding confidence, the color image with the above results, the diameter of the grippers corresponding to each Pick Point and the mask of the gripping area.
2. *Flip Poses' Axes* : Input the planar pick points calculated in the previous step, rotate the specified axis of the pick points to a specific direction and then output it.
3. *Assign Suction Cup Label* : Specify suction cup label for the planar pick points obtained in the first step.
4. *Visualize Information on Image* : Mark the the flipped planar pick points and their corresponding suction cup labels on the input color image.

5.14.1 Assign Suction Cup Label

Input the planar pick points, their confidence levels and the mask of the grabbing area obtained in *Calc Poses From Heat Map Of Graspability* , specify the classification standard of grabbing objects and generate the label file. The program of this procedure is shown in *Figure 2*.

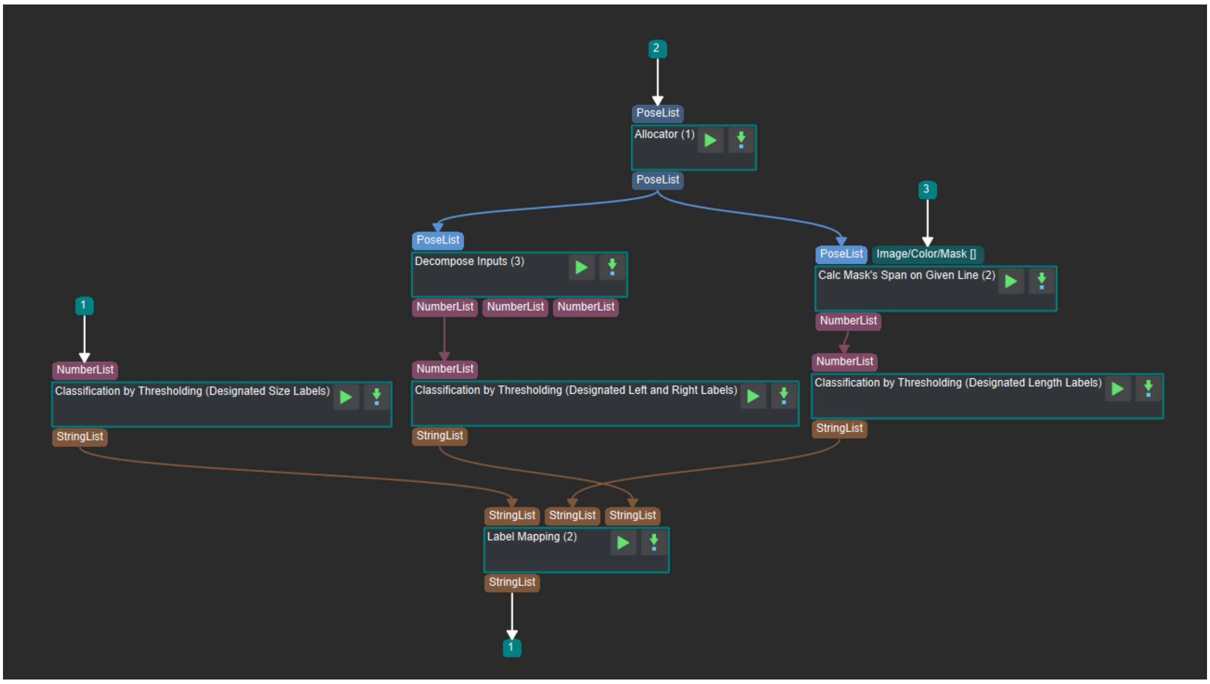


Figure 2 The program of assigning suction cup label in Mech-Vision

The introduction of each step in the program is as follows :

1. *Stratify Values by Thresholds* : Input the array which need to be classified, classify it according to the threshold and the label file in the setting then output the label.
2. *Calc Mask' s Span on Given Line* : Input the planar pick points and its corresponding mask in the grabbing area, calculate the length of the planar pick points in the mask and then output it.
3. *Allocator* : Allocate the inputs to the next few steps.

4. *Decompose Object Dimensions* : Decompose a 3D input into three 1D lists, output the corresponding lists according to the requirements.
5. *Label Mapping* : Input the labels obtained through different classification standards and obtained a single label list according to the rule of label setting.

5.15 Object Classification (Mask Size)

Divide objects into two categories according to the mask size, and add labels to the corresponding pick points for each category. The procedure is shown in *Figure 1*.

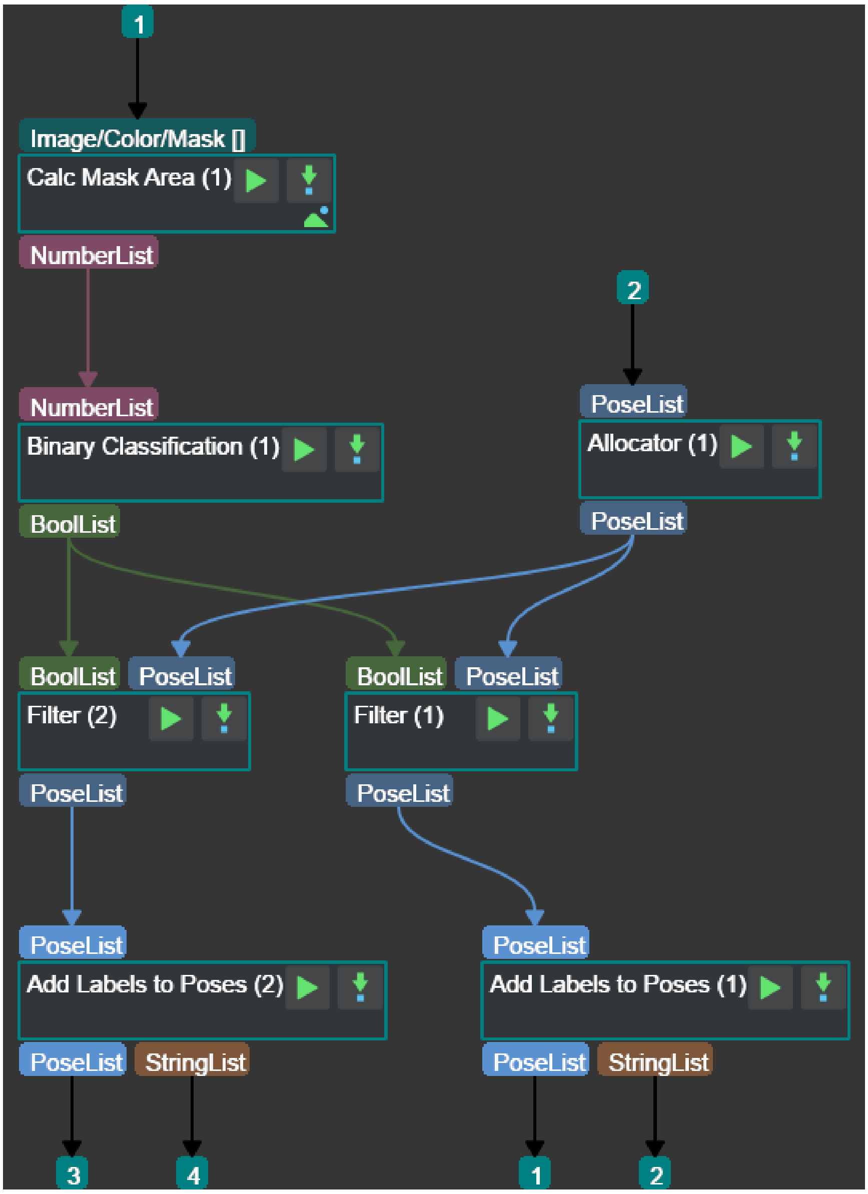


Figure 1 Procedure of object classification (mask size)

The role of each tep is as follows:

1. *Calc Areas of Masks* Take the list of mask pictures as input to calculate the size of each mask picture and output.
2. *Dichotomize Values by Threshold* Compare each element of the result from the previous step with the threshold, and output a list of Boolean values composed of True and False. This list is used to distinguish the types of objects corresponding to each mask.
3. *Allocator* Allocate the input pose list to the subsequent steps.
4. *Filter* Based on the Boolean value list and the pose list, output the pose corresponding to True or False to obtain the poses of different object types.
5. *Add Labels to Poses* Use the label file to add a label to each pose in the input pose list. Finally obtain the poses with label.

The functions of *Filter* and *Add Labels to Poses* on the other way are the same as those in 4 and 5 above, adding labels to the poses of another type of object.

5.16 Filtering Operations

There are many different combinations of filtering operations. In this section, **Filter by threshold** and **Filter by label** are used as examples and can be applied flexibly according to actual needs. The possible filtering operations include, but are not limited to point clouds, poses and labels. This section takes point clouds as an example.

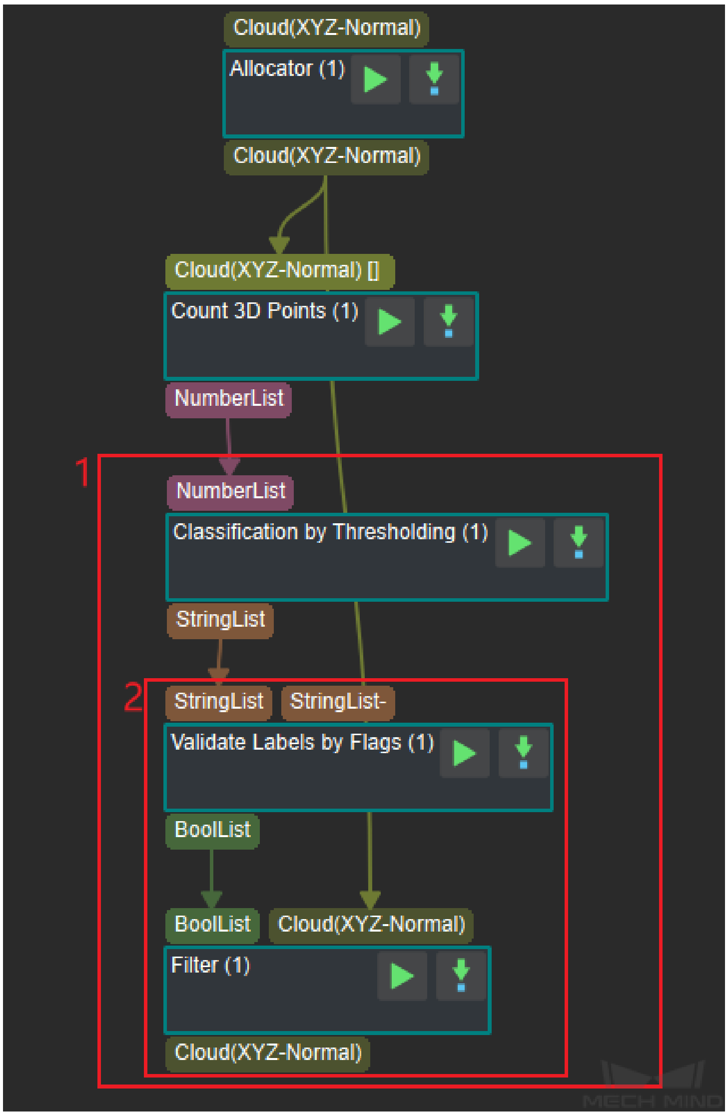


Figure 1 Example of procedure for filtering operations

1. The procedure **Filter by Threshold**
2. The procedure **Filter by Label**

5.16.1 Process Description

Filtering by label

1. *Validate Labels and Output Flags* step and **Label File** generate an array of flags with the same number of tags as the input, with the values `True` and `False`.
2. The *Filter* step keeps the input with the flag `True` and filters out the input with the flag `False` ; if **Reverse Bool List** is checked, the result is reversed.

Filtering by threshold

1. The *Stratify Values by Thresholds* step sorts the input array of values according to a threshold value. Less than the threshold is the previous category label and greater than or equal to the threshold is the next category label.
2. Use the **Filtering by label step combination** to filter the labels and achieve the threshold filtering effect.

5.16.2 Example of How to Achieve The Effect

Input point cloud

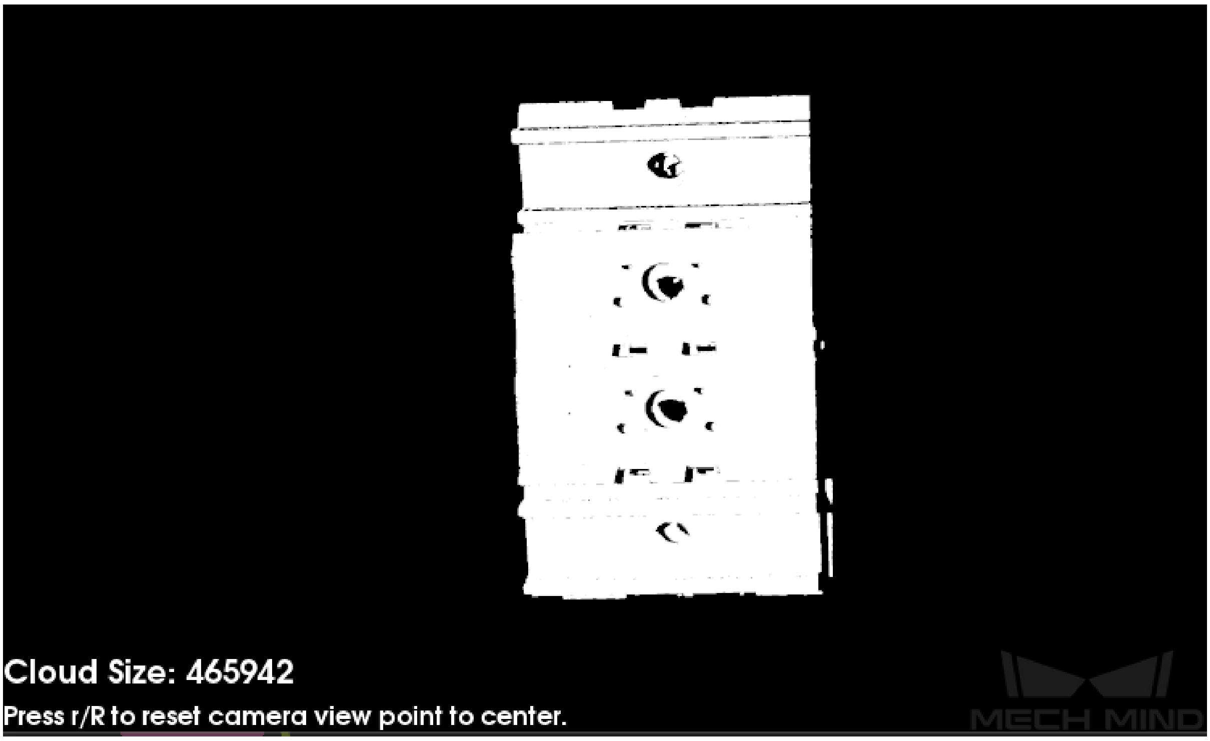


Figure 2 Input Point Cloud Image

Process flags

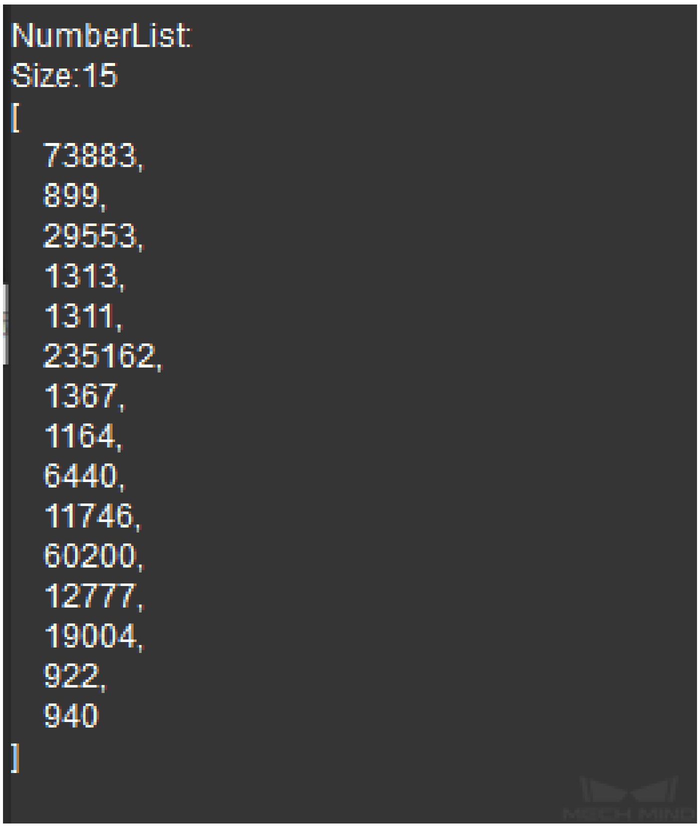


Figure 3 Input of **Stratify Values By Thresholds**

Here the threshold is 2000 and the labels are **ng, ok**. The required label is **ok** .

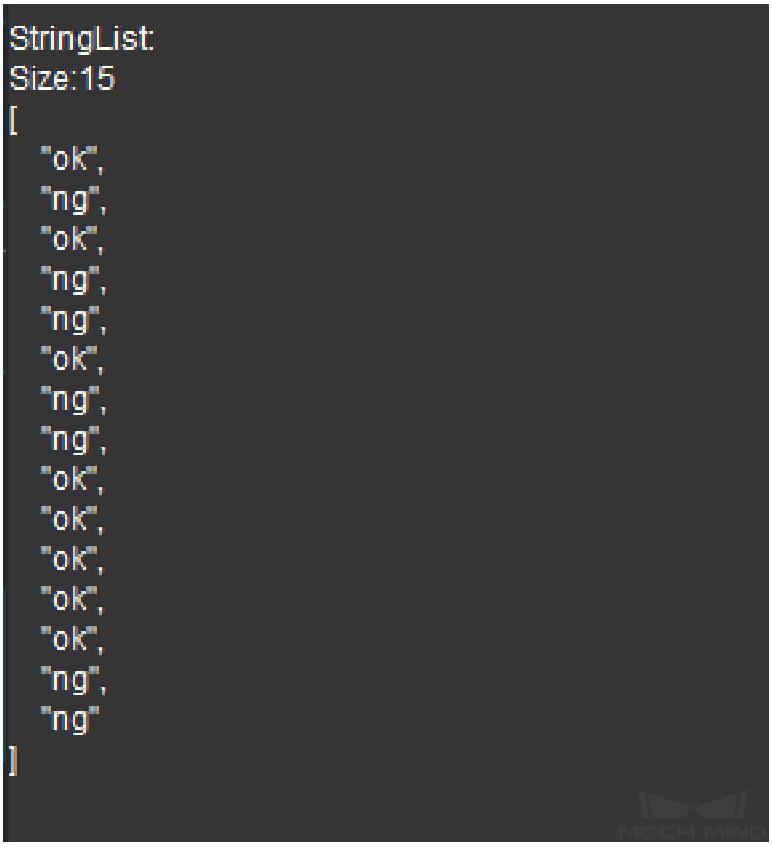


Figure 4 Input of **Validate Labels and Output Flags**

Retain the point cloud class with the flag `True` .

```
BoolList:
Size:15
[
  true,
  false,
  true,
  false,
  false,
  true,
  false,
  false,
  true,
  true,
  true,
  true,
  true,
  false,
  false
]
```

Figure 5 Input of **Filter**

Resulting point cloud

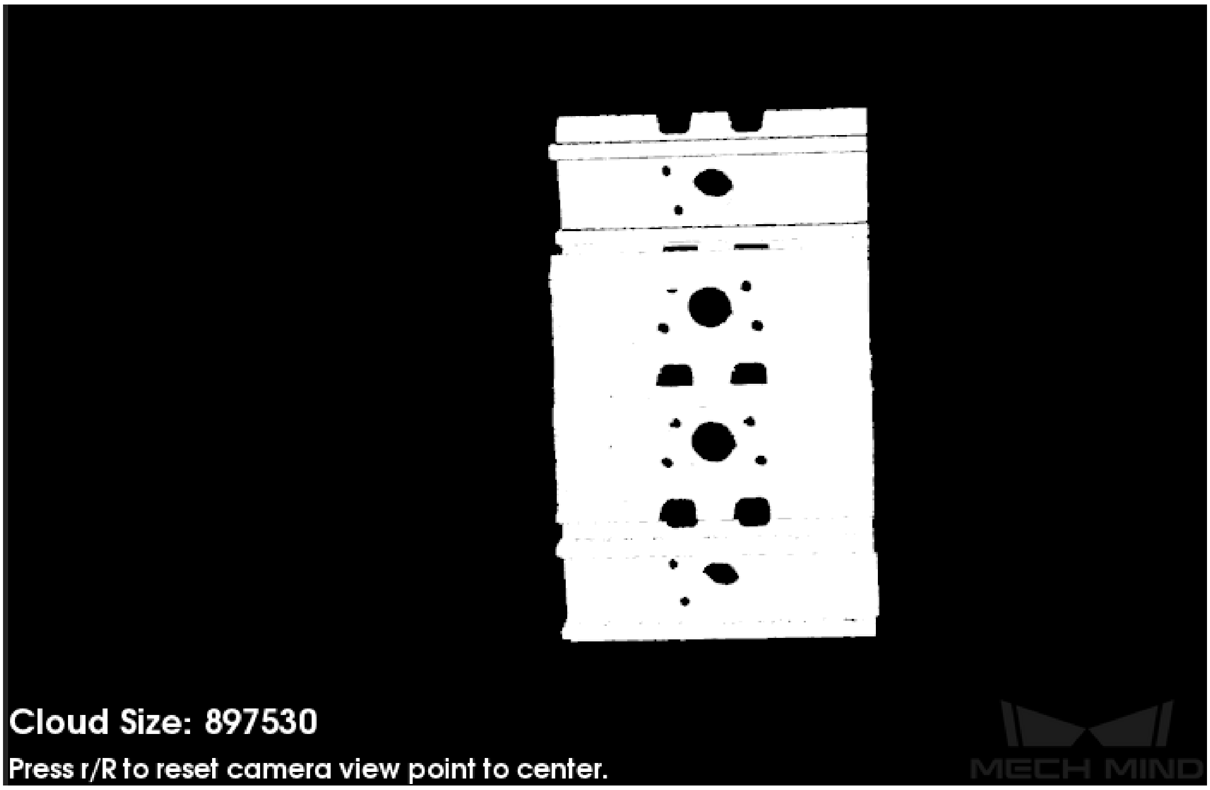


Figure 6 Resulting Point Cloud Image

5.17 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 *Figure 1.* shown.

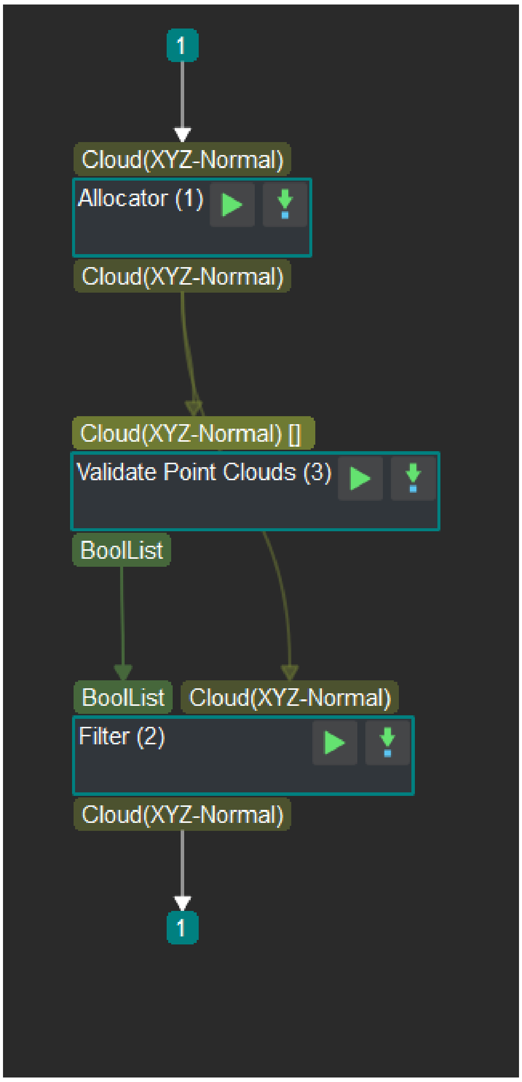


Figure 1. Procedure of filtering out point clouds exceed the limit

The introduction of each steps is as follows:

1. *Validate Point Clouds* Detect whether the number of the point cloud exceeds the limit.
2. *Filter* Remove the point cloud if the number of the point cloud exceeds the limit and output the filtered point cloud.

5.18 Filter Out Point Clouds That Exceed The Limit After Sorting

The main function of this procedure is similar to *Filter Out Point Clouds That Exceed The Limit*. The difference is this procedure sort the point clouds before filter out. The program of this procedure is shown in *Figure 1*.

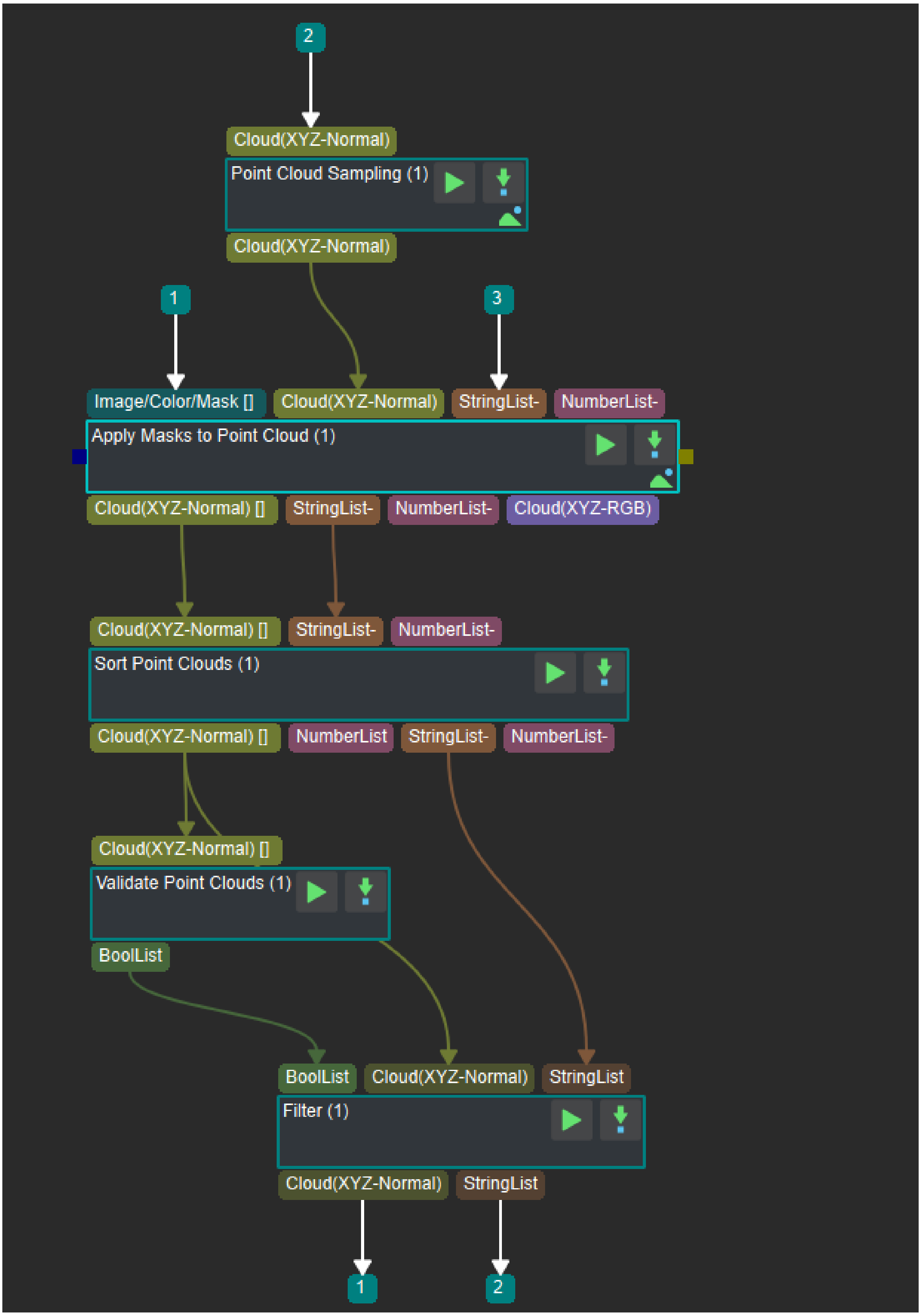


Figure 1 The program of filtering out point clouds that Exceed the limit after sorting in Mech-Vision

The introduction of each step is as follows :

1. *Down-Sample Point Cloud* : This step appropriately compress the point clouds to reduce the number of point clouds and speed up the operation.
2. *Apply Masks to Point Cloud* : Input point clouds and the mask and find out the point clouds under the mask.
3. sort_point_clouds : Sort the point clouds according to the rules
4. *Validate Point Clouds* : Input point clouds, validate the point cloud whether it is within the setting range.
5. *Filter* : Input the result of the previous step, keep the point cloud if it meet the requirements otherwise remove it and then get the point clouds after filtering.

5.19 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 as *Figure 1.* shown.

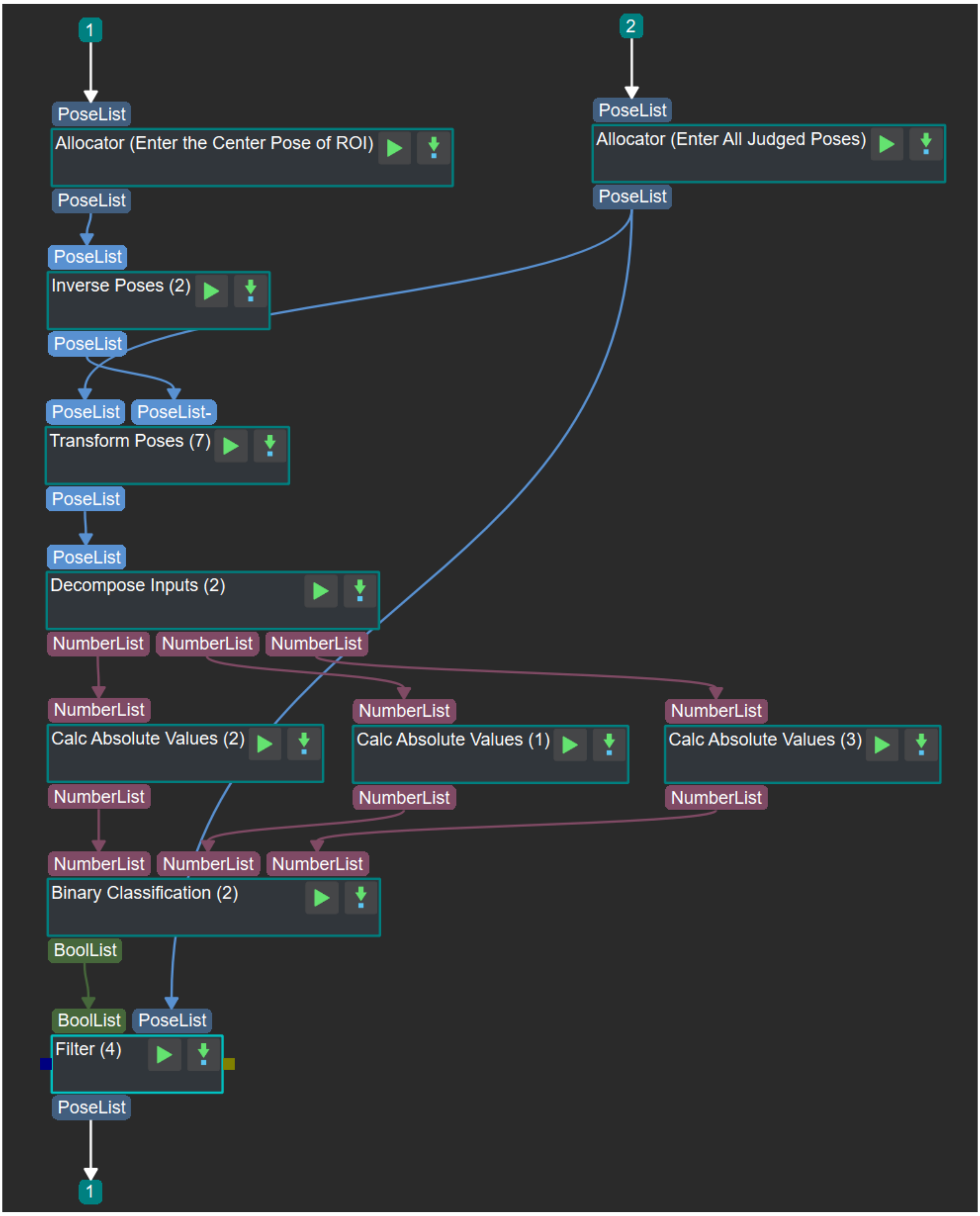


Figure 1. Procedure of filtering out poses out of ROI

The introduction of each steps is as follows:

1. *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.
2. *Inverse Poses* Calculate the inverse of the center pose of the ROI.
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.

5.20 Pick Points Filtering and Sorting

The main function of this procedure is sorting and filtering the pick points according to the specifications. According to the requirements of practical project, one or more different specifications can be set to obtain pick points with high grasp probability, which are used for subsequent planning of picking task. The procedure is shown in *Figure 1*.

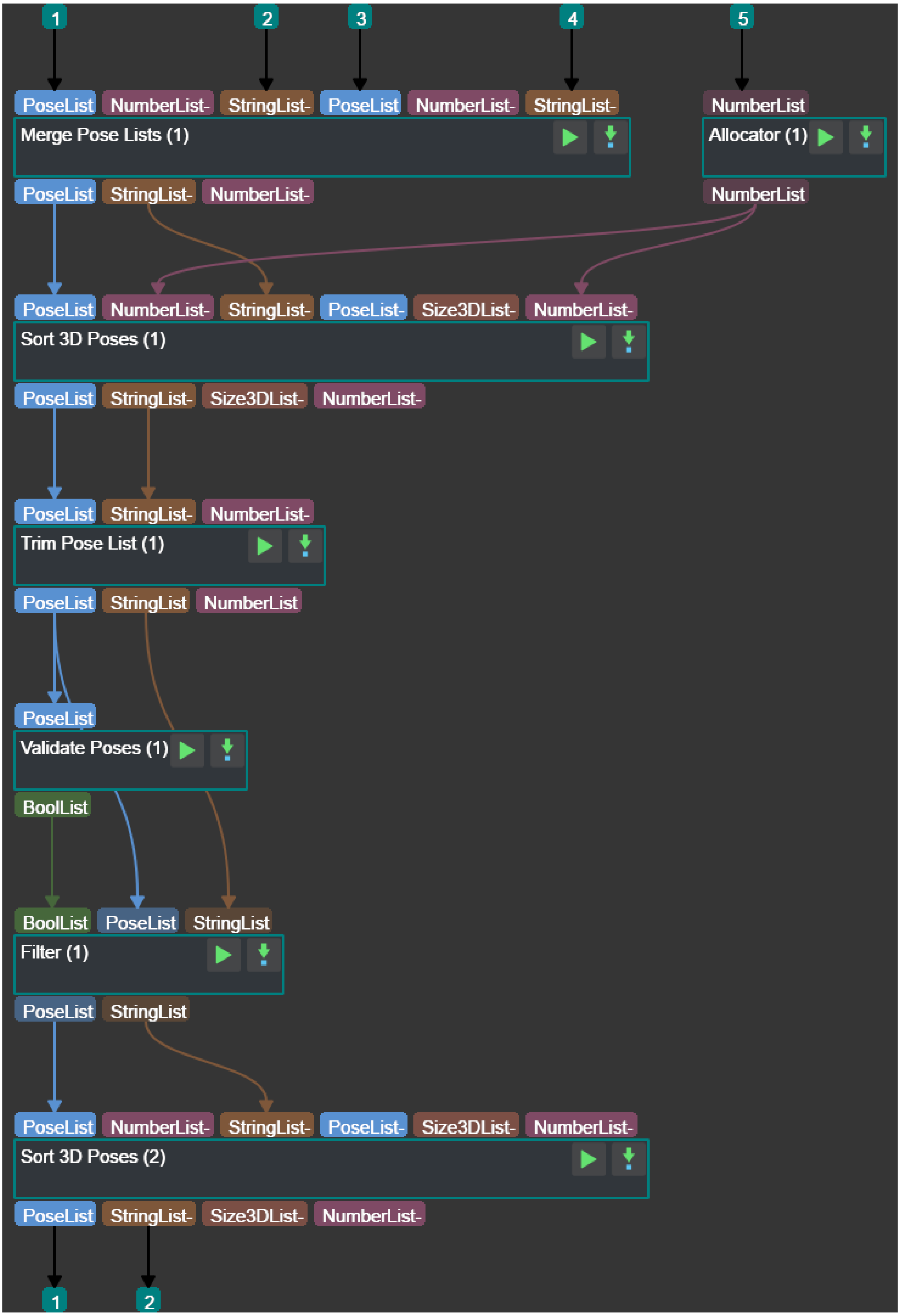


Figure 1 Procedure of pick points filtering and sorting

The role of each Step is as follows:

1. *Merge Pose Lists* Merge the two input pick point lists and their corresponding label lists into one and output them respectively.
2. *Allocator* Allocate the input list of pick points confidence to the subsequent Steps.
3. *Sort 3D Poses* Based on the input list of pick points and the corresponding confidence and label lists, sort the pick points according to the specified criteria (confidence), and output the sorted list of pick points and the corresponding label list.
4. *Trim Pose List* Input the result of the previous step and output several pick points with high confidence to speed up subsequent calculations.
5. *Validate Poses by Included Angle to Reference Direction* Take the list of selected pick points as input to convert each pick point to True or False according to the set value and output the Boolean value list.
6. *Filter* Based on the Boolean value list, the pick point list and the corresponding label list, choose the pick points corresponding to True or False, and output the list of selected pick point and its corresponding label list.
7. *Sort 3D Poses* Based on the filtered pick points and the corresponding labels, sort the pick points again according to the specified criteria (the value corresponding to the Z axis), and output the sorted pick points and the corresponding labels .

5.21 Acquire RGB Images Based on Depth Images

The principle behind this combination of steps is to get the RGB image based on the highest layer within the depth images ROI to extract the valid regions and to remove other regions from interfering with the deep learning recognition.

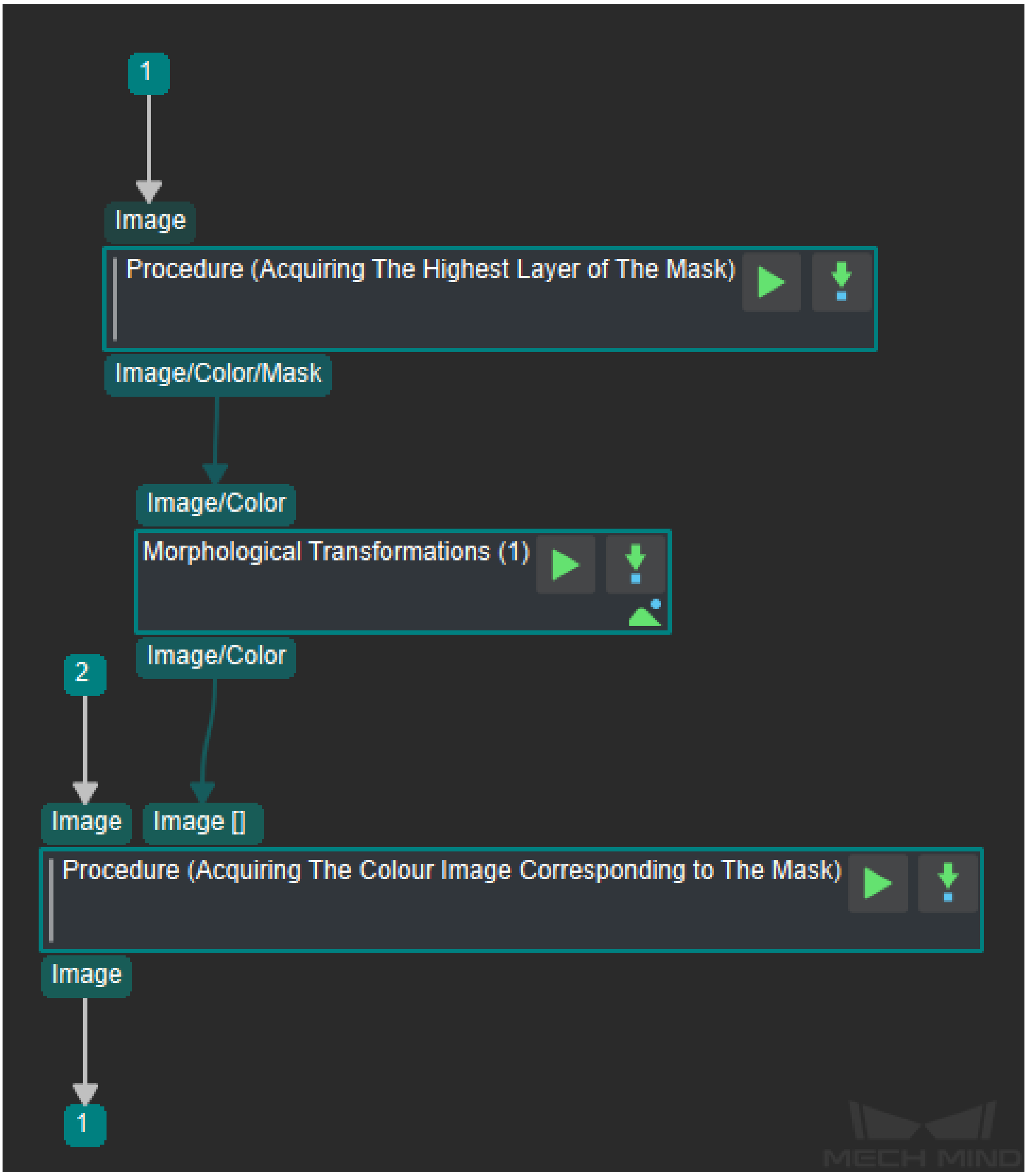


Figure 1 Combination of Steps to Acquire RGB Images Based on Depth Images

As shown in *Figure 1*, Acquire RGB Images Based on Depth Images procedure consists of three parts: **Acquiring The Highest Layer of The Mask** , **Morphological Transformations** and **Acquiring The Colour Image Corresponding to The Mask** .

5.21.1 Acquiring The Highest Layer of The Mask

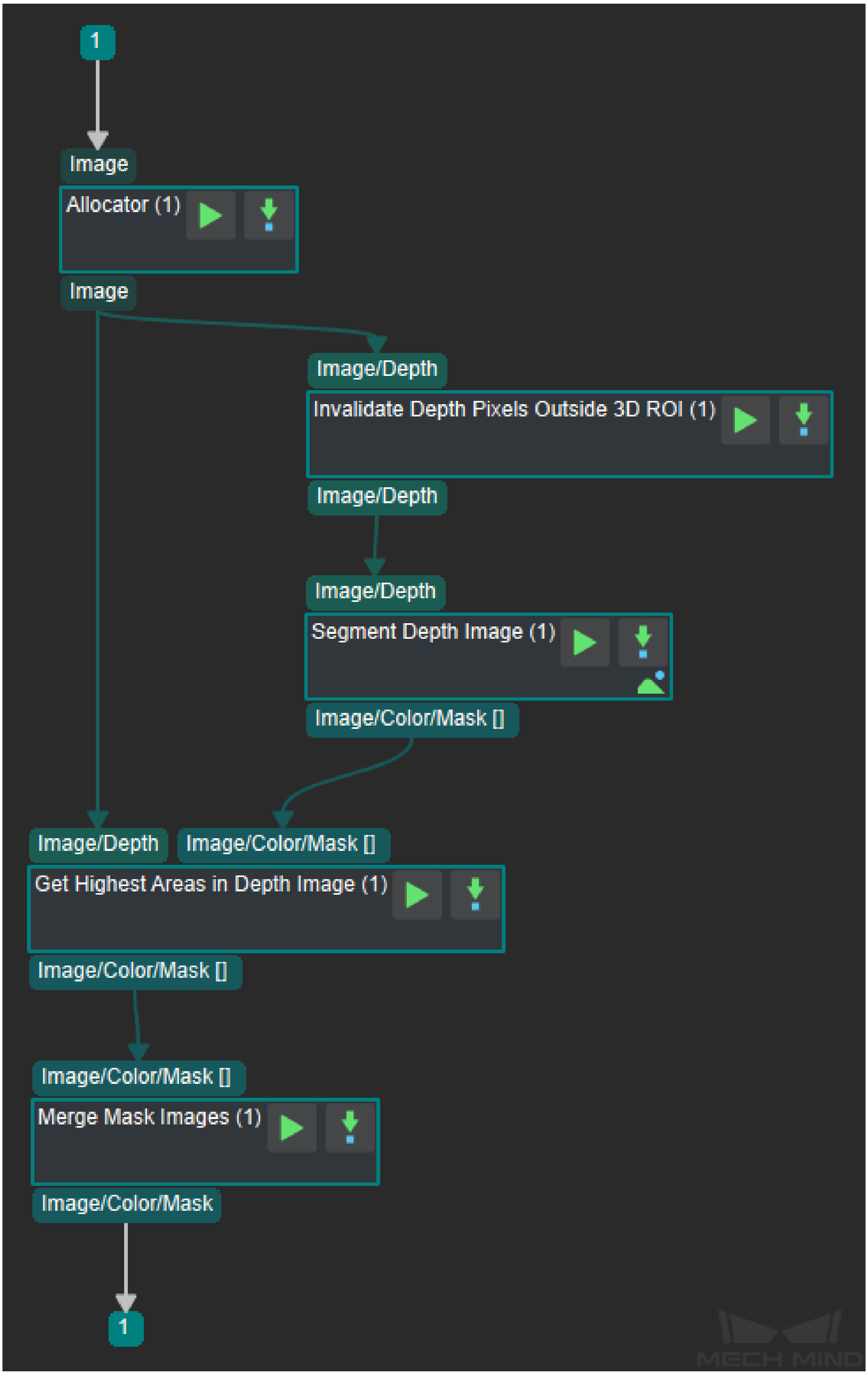


Figure 2 Combination of Steps to Acquiring The Highest Layer of The Mask

As shown in *Figure 2*, the process for **Acquiring The Highest Layer of The Mask** combination of steps is as follows.

1. **Invalidate Depth Pixels Outside 3D ROI** step extracts the depth data of the ROI area.
2. **Segment Depth Image** step to segment the depth image and filter small pieces of clutter.
3. **Get Highest Areas in Depth Image** Step to generate a mask of valid regions.
4. **Merge Mask Images** step to merge multiple mask images into one image.

Tip: There are two ways to set the ROI region, directly editing the json file and using Mech-Viz software. The essential difference between the two ways lies in the way the json file is manipulated, the second method is recommended.

5.21.2 Morphological Transformations

This step expands the masked image, which serves to expand the edge areas and preserve the edge information of the ROI in the image.

5.21.3 Acquiring The Colour Image Corresponding to The Mask

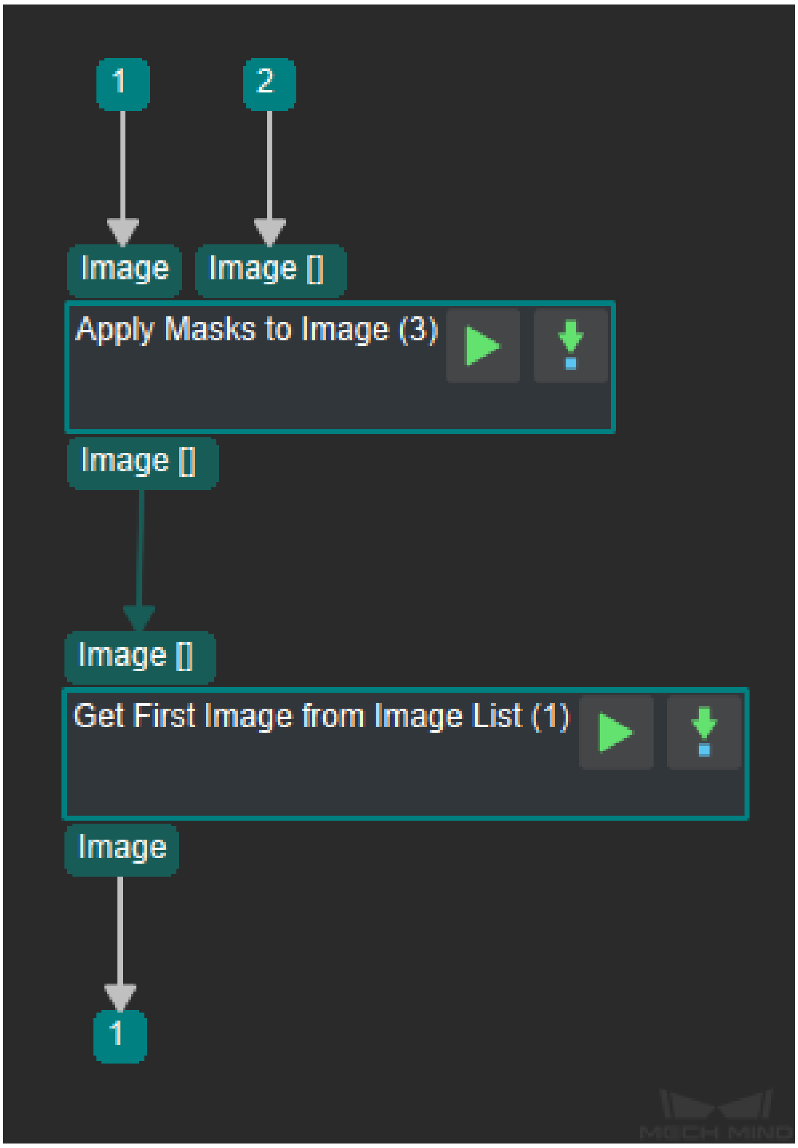


Figure 3 Combination of Steps to Acquiring The Colour Image Corresponding to The Mask

1. *Apply Masks to Image* step to extract the 2D image of the corresponding region of the mask image.
2. *Get First Image from Image List* step to get the result image corresponding to the first mask image.

Tip: If there is no *Get First Image from Image List* step in the list of steps on the left, the user can uncheck *Show only frequently used steps* in the settings.

5.21.4 Example Implementation

Input

Figure 4 RGB Original Image

Figure 5 Depth Original Image

Output

Figure 6 Result Image

Attention: When it is necessary to use Mech-Viz software to perform the ROI setting for the step **Invalidate Depth Pixels Outside 3D ROI**. The project must contain a **extract point cloud within ROI** step. Otherwise Mech-Viz will not complete the ROI setting and a message will pop up: ROI setting failed RPC failed.

5.22 Procedure of Instance Segmentation

Detecting the location of target objects in an image by instance segmentation provides segmentation masks and target object classes for point cloud processing and other operations.

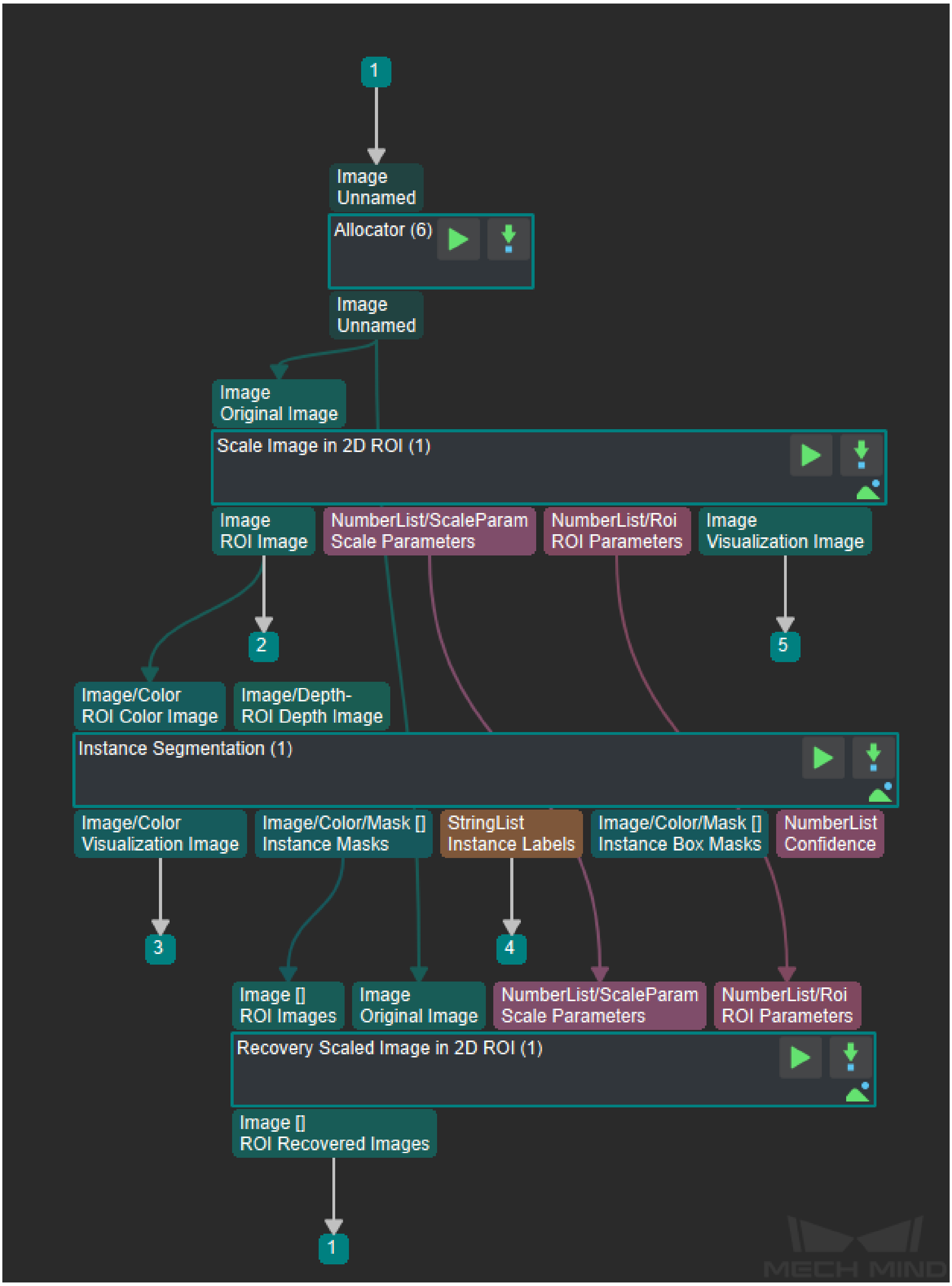


Figure 1 Combination Step of Instance Segmentation

5.22.1 Process Description

As shown in *Figure 1*, the flow of the **Instance Segmentation** procedure is as follows.

1. *Scale Image in 2D ROI* step extracts and scales the ROI region of the image.
2. *Instance Segmentation* step performs the recognition of 2D images and obtains the recognition result (category + mask).
3. The *Recover Scaled Images in 2D ROI* step scales the result of the *Instance Segmentation* step back to the original input image size to get the result of the instance segmentation on the original image.

5.22.2 Example implementation

Input

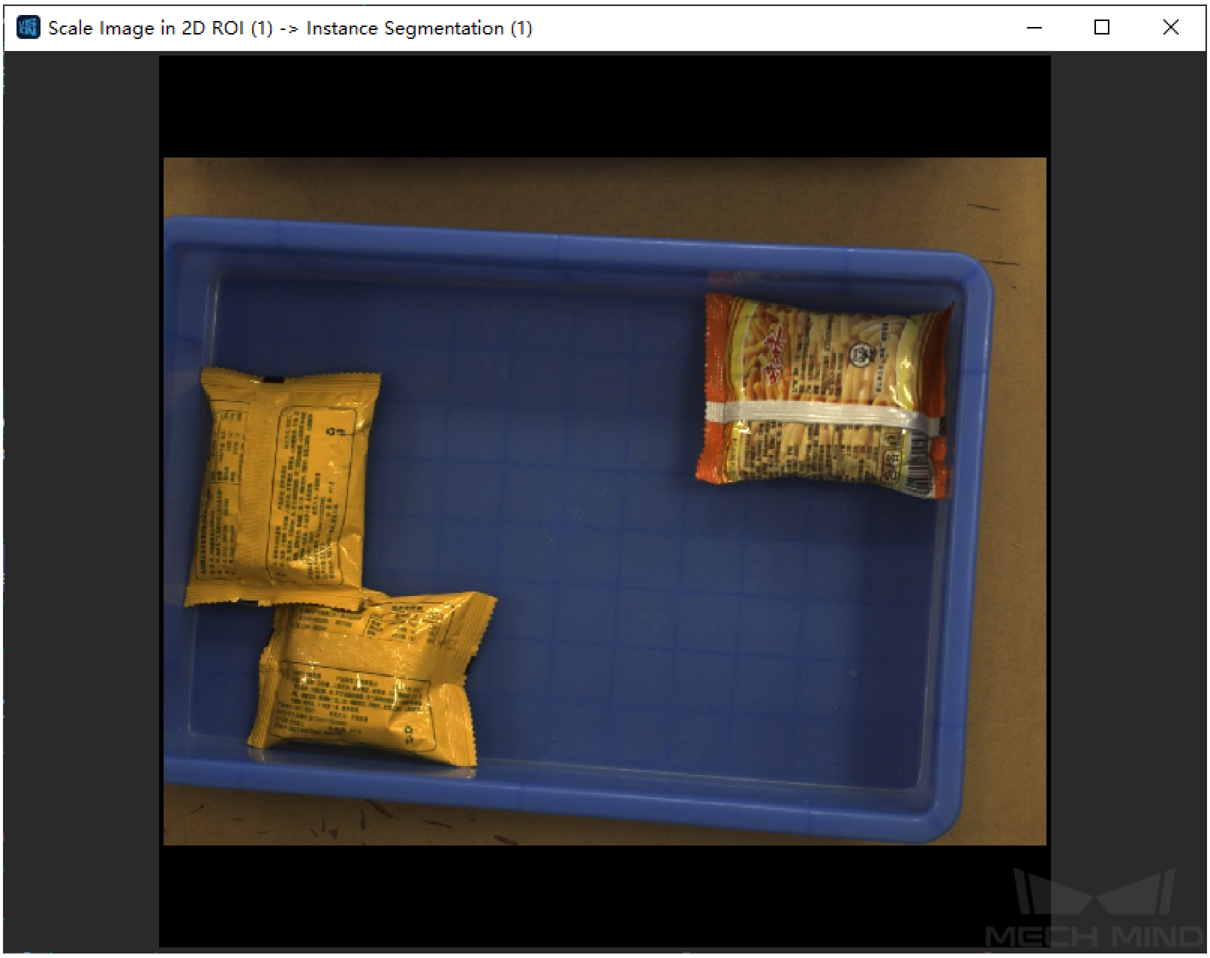


Figure 2 RGB Original Image

Identification Results

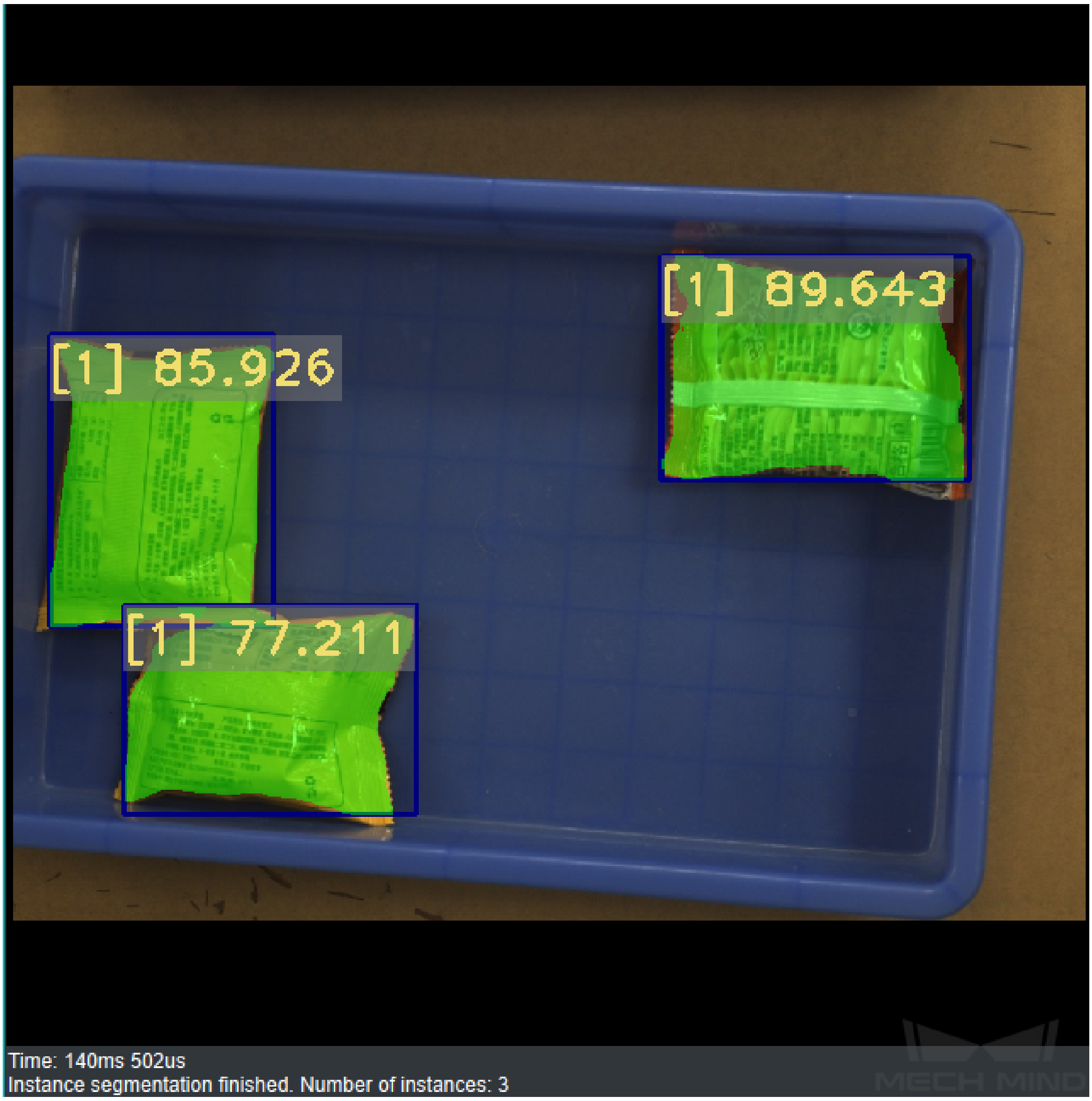


Figure 3 Image of Recognition Results

5.22.3 Precautions

1. The result of instance segmentation is largely affected by the ROI area and zoom ratio. If the user wants to obtain a more ideal instance segmentation result, the user must adjust the ROI and zoom ratio. For specific adjustment methods, refer to *Deep Learning Deployment ROI Settings*.
2. The maximum number of recognition of **Instance segmentation**: its modified value is too large (that is, the actual detection amount is large) . There may be insufficient GPU resources, and GPUs with higher computing power can be replaced according to specific project requirements.

```
INFO - 2020-10-26 11:08:14 - <string> - 77 - Request received: model type InstanceSeg model path c:\users\mech-mind-113\desktop\test_model\epoch_2500.pth config path c:\users\mech-mind-113\desktop\test_model\202010:
INFO - 2020-10-26 11:08:14 - <string> - 77 - Latency: 0.11864709854125977
INFO - 2020-10-26 11:08:15 - <string> - 77 - Request received: model type InstanceSeg model path c:\users\mech-mind-113\desktop\test_model\epoch_2500.pth config path c:\users\mech-mind-113\desktop\test_model\202010:
INFO - 2020-10-26 11:08:15 - <string> - 77 - Latency: 0.14062213897705078
INFO - 2020-10-26 11:08:16 - <string> - 77 - Request received: model type InstanceSeg model path c:\users\mech-mind-113\desktop\test_model\epoch_2500.pth config path c:\users\mech-mind-113\desktop\test_model\202010:
INFO - 2020-10-26 11:08:16 - <string> - 77 - Latency: 0.13164758682250977
INFO - 2020-10-26 11:08:17 - <string> - 77 - Request received: model type InstanceSeg model path c:\users\mech-mind-113\desktop\test_model\epoch_2500.pth config path c:\users\mech-mind-113\desktop\test_model\202010:
ERROR - 2020-10-26 11:08:18 - <string> - 86 - Failed to predict: CUDA out of memory. Tried to allocate 712.00 MiB (GPU 0; 4.00 GiB total capacity; 1.88 GiB already allocated; 16.14 MiB free; 2.60 GiB reserved in total by PyTorch)
INFO - 2020-10-26 11:11:33 - <string> - 77 - Request received: model type InstanceSeg model path c:\users\mech-mind-113\desktop\test_model\epoch_2500.pth config path c:\users\mech-mind-113\desktop\test_model\202010:
INFO - 2020-10-26 11:11:34 - <string> - 77 - Latency: 0.491652250289917
INFO - 2020-10-26 11:11:37 - <string> - 77 - Request received: model type InstanceSeg model path c:\users\mech-mind-113\desktop\test_model\epoch_2500.pth config path c:\users\mech-mind-113\desktop\test_model\202010:
INFO - 2020-10-26 11:11:37 - <string> - 77 - Latency: 0.1271655559539795
INFO - 2020-10-26 11:11:38 - <string> - 77 - Request received: model type InstanceSeg model path c:\users\mech-mind-113\desktop\test_model\epoch_2500.pth config path c:\users\mech-mind-113\desktop\test_model\202010:
INFO - 2020-10-26 11:11:38 - <string> - 77 - Latency: 0.4009270668029785
INFO - 2020-10-26 11:11:40 - <string> - 77 - Request received: model type InstanceSeg model path c:\users\mech-mind-113\desktop\test_model\epoch_2500.pth config path c:\users\mech-mind-113\desktop\test_model\202010:
ERROR - 2020-10-26 11:11:41 - <string> - 86 - Failed to predict: CUDA out of memory. Tried to allocate 528.00 MiB (GPU 0; 4.00 GiB total capacity; 1.66 GiB already allocated; 16.14 MiB free; 2.60 GiB reserved in total by PyTorch)
INFO - 2020-10-26 11:13:29 - <string> - 77 - Request received: model type InstanceSeg model path c:\users\mech-mind-113\desktop\test_model\epoch_2500.pth config path c:\users\mech-mind-113\desktop\test_model\202010:
INFO - 2020-10-26 11:13:29 - <string> - 77 - max_instances_per_image changed from 500 to 100
INFO - 2020-10-26 11:13:29 - <string> - 77 - Start loading model...
INFO - 2020-10-26 11:13:30 - <string> - 77 - Model loading finished.
INFO - 2020-10-26 11:13:30 - <string> - 77 - Latency: 0.5329568386077881
INFO - 2020-10-26 11:13:42 - <string> - 77 - Request received: model type InstanceSeg model path c:\users\mech-mind-113\desktop\test_model\epoch_2500.pth config path c:\users\mech-mind-113\desktop\test_model\202010:
INFO - 2020-10-26 11:13:42 - <string> - 77 - max_instances_per_image changed from 100 to 200
INFO - 2020-10-26 11:13:42 - <string> - 77 - Start loading model...
INFO - 2020-10-26 11:13:43 - <string> - 77 - Model loading finished.
ERROR - 2020-10-26 11:13:44 - <string> - 86 - Failed to predict: CUDA out of memory. Tried to allocate 712.00 MiB (GPU 0; 4.00 GiB total capacity; 1.88 GiB already allocated; 436.14 MiB free; 2.19 GiB reserved in total by PyTorch)
```

Figure4 Mech-Vision GPU Resource is Insufficient and Error Image is Reported

5.22.4 Deep Learning Deployment ROI Settings

ROI area and zoom ratio settings are divided into two situations. When the collected image is the same as the collected scene environment, the user can use the color image ROI file or parameter settings. When it is different from the scene of the collection environment, the user needs to calculate the zoom ratio by himself.

Consistent with the collection site environment

The following two methods are recommended to use the first one. When using the color image ROI file to set and zoom the ROI area, please refer to the method of obtaining the *color image ROI file*.

Use color image ROI file

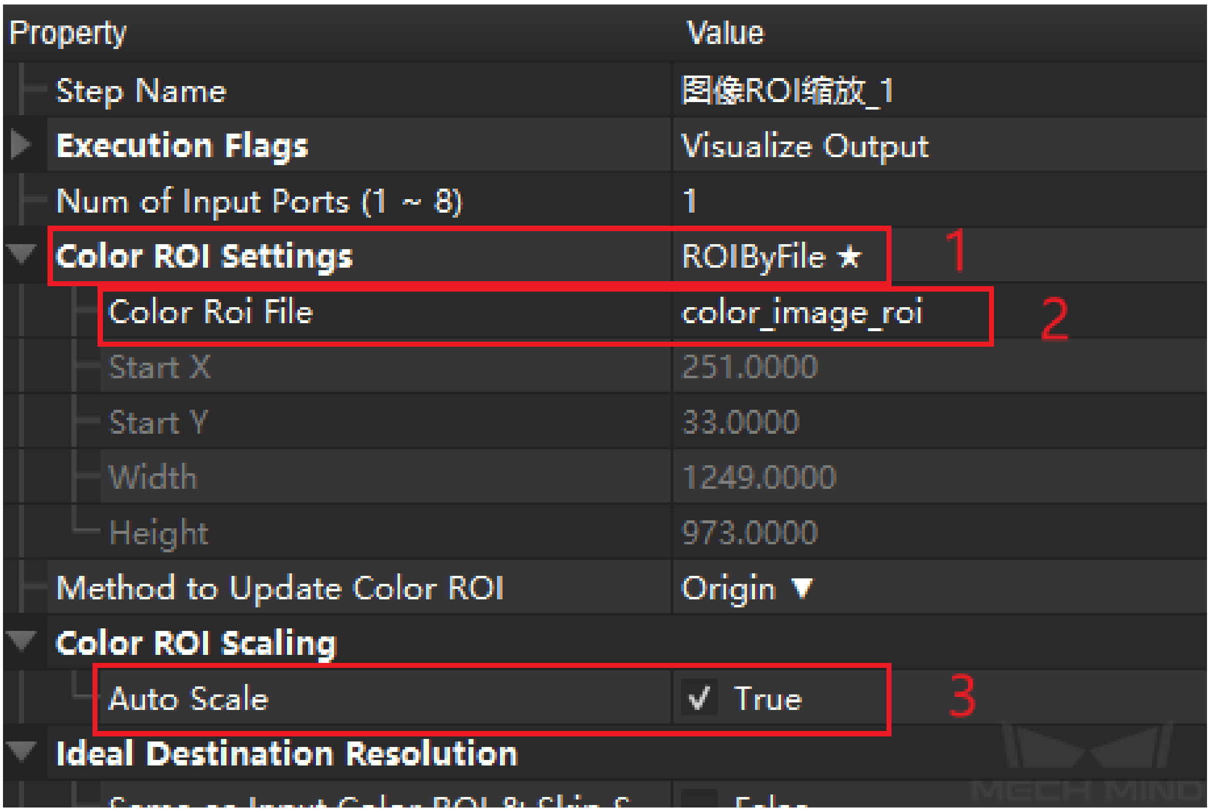


Figure5 Image ROI Zoom File Settings

1. Select **ROIByFile** in **Color Map ROI Settings** .
2. The obtained color ROI file.
3. Check **Auto zoom** .

Use parameters

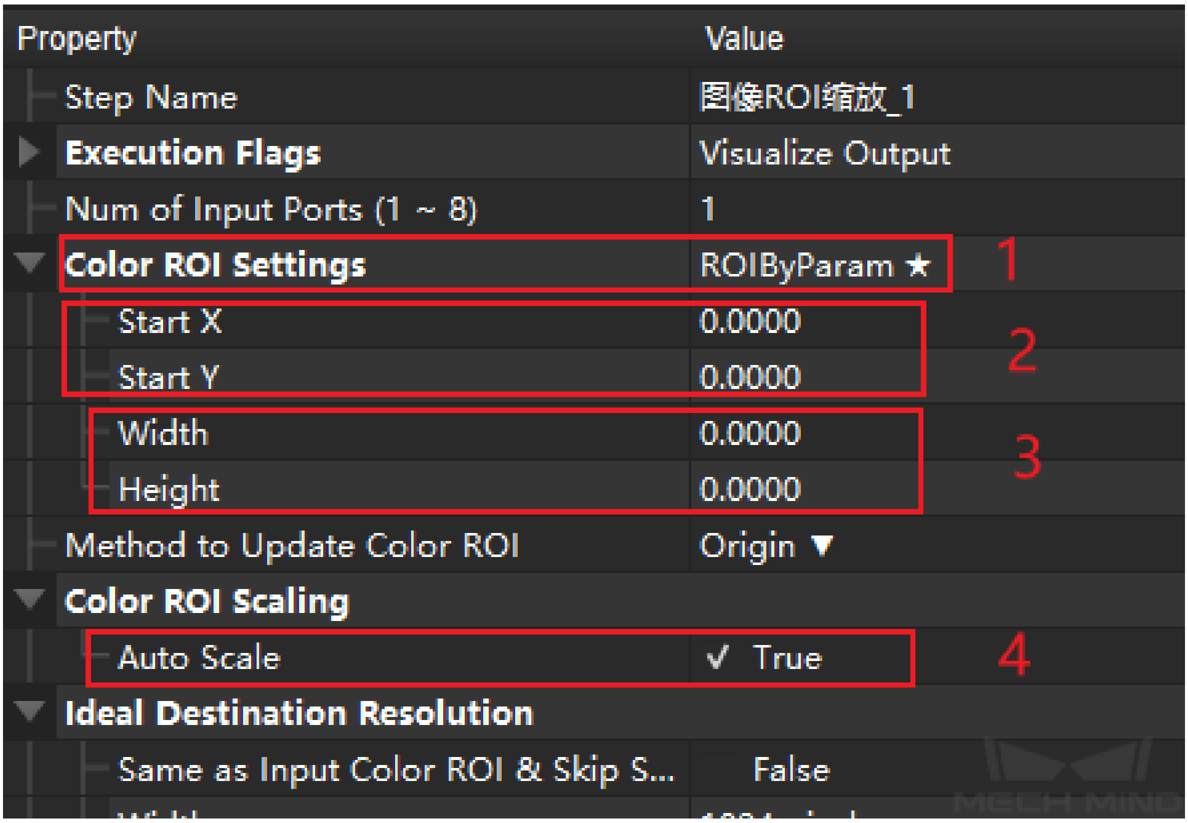


Figure6 Image ROI Scaling Parameter Settings

1. Select **ROIByParam** in **Color Map ROI Settings** .
2. Adjust the value of startX and startY to determine the starting point in the upper left corner.
3. When changing the size of the ROI area, try to ensure that the aspect ratio of the ROI is consistent with that during training. For example, during training, **width** is `1000`, **height** is `1200`, Then the new **width** is `1000 * some ratio` , and the new **height** is `1200 * some ratio` .
4. Check **Auto zoom** .

Inconsistent with the collection site environment

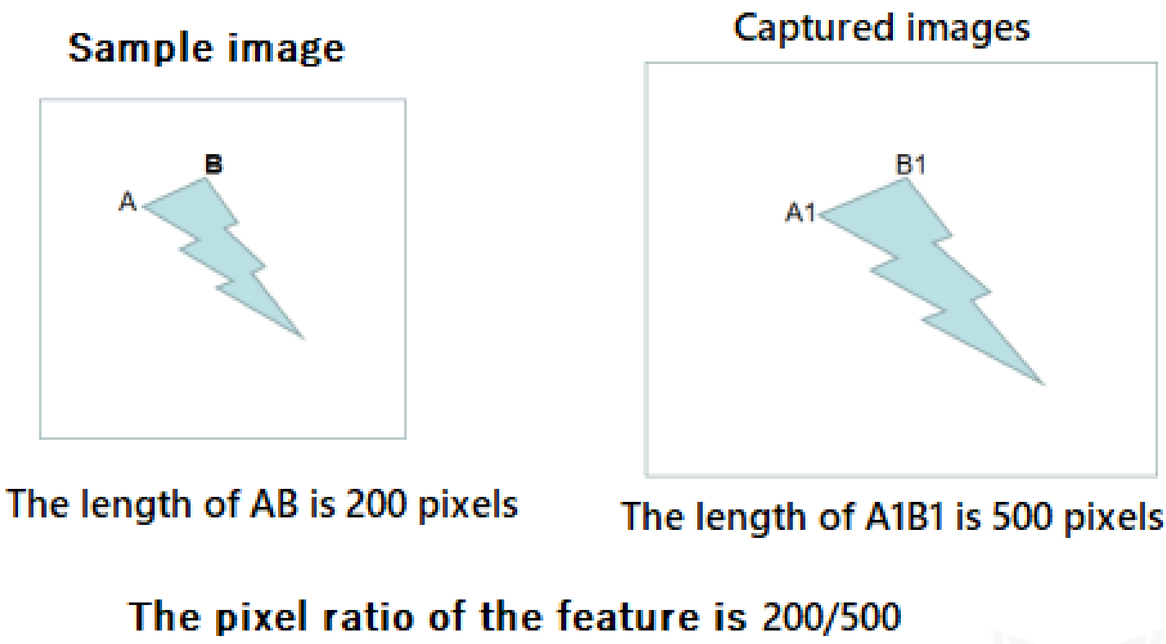


Figure7 Ratio Calculation

1. Use the Sample image in the model folder and the images collected on site to select the same feature of the object to be detected, such as *Figure 8*.In the calculation example, the selected feature is the AB edge.
2. Measure the pixel of this feature, and get the pixel length in the Sample image as 200, and the pixel length in the collected image as 500 .

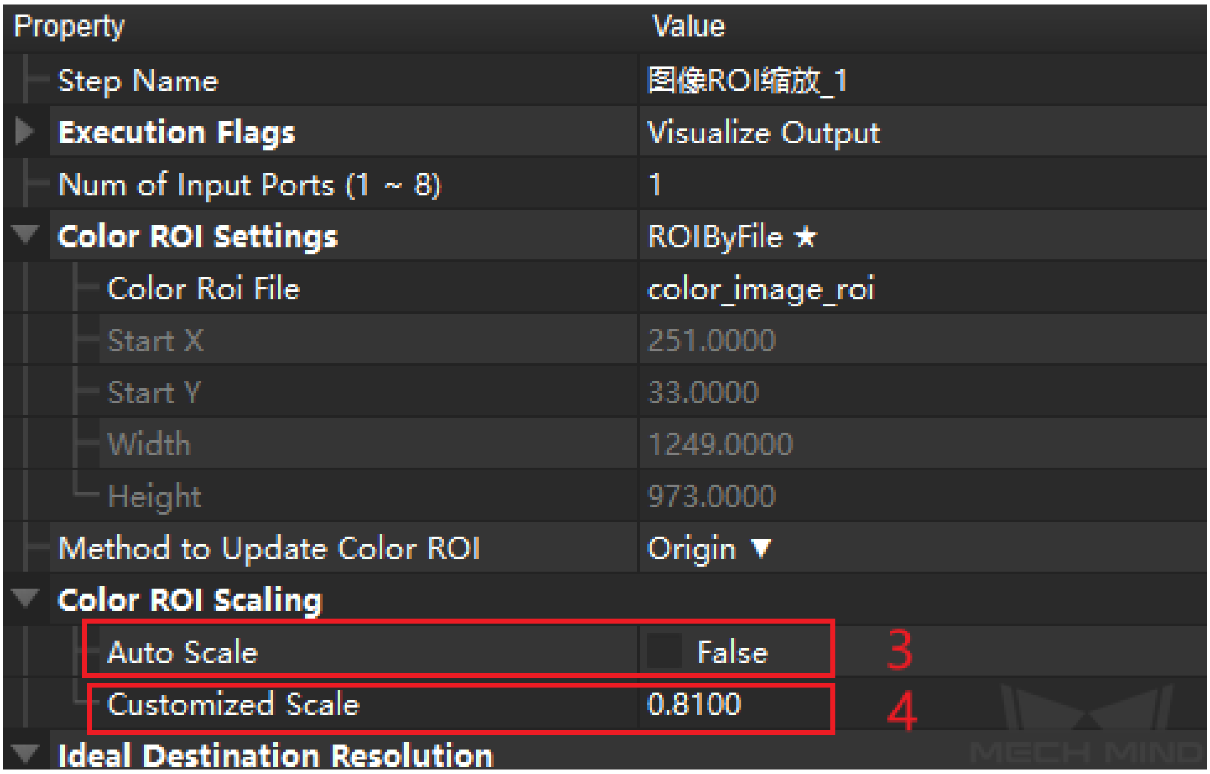


Figure8 Image ROI Zoom Setting

3. Uncheck **Auto zoom** .
4. Calculate the ratio of the two, that is, `200/500`, and fill in the value into the custom zoom factor.

5.23 Instance Segmentation (Colored Image)

The main function of this step is to obtain the list of the masks of the object by object detection and recognition of deep learning. It usually applies when we need to recognize and segment the object. The combination is as *Figure 1*.shown.

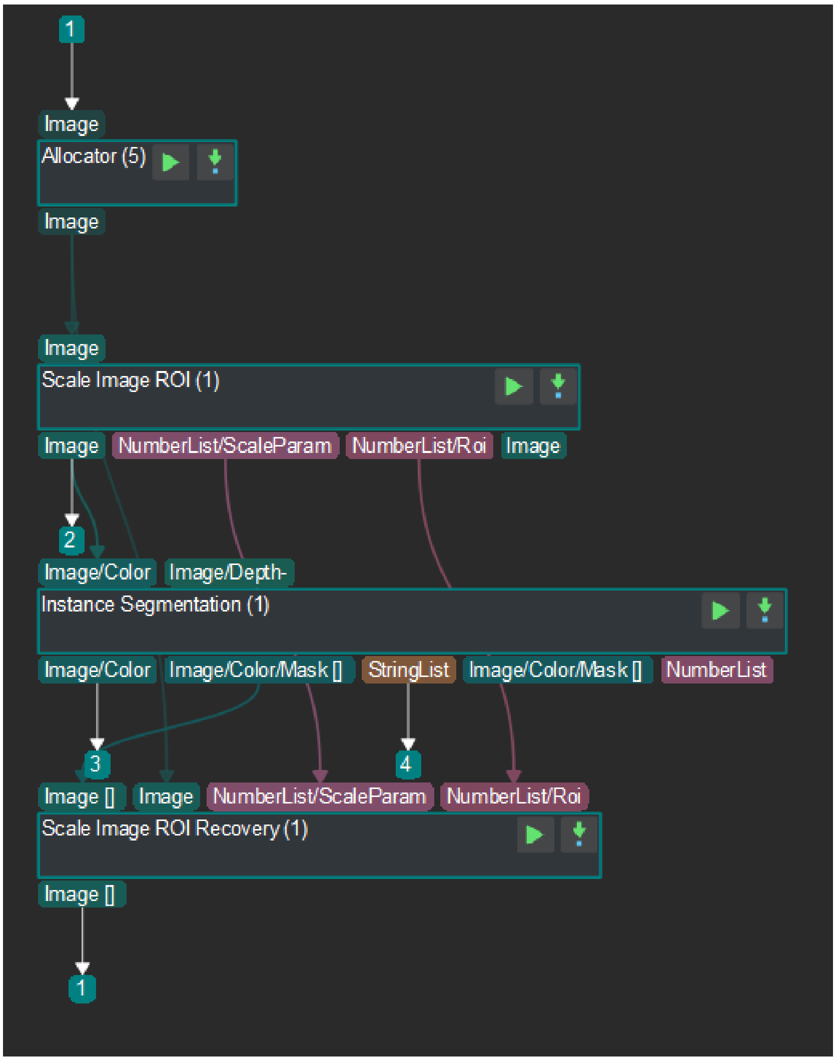


Figure 1. Procedure of instance segmentation(colored image)

The step *Scale Image in 2D ROI* would process the colored image and output the scaled ROI to the step *Instance Segmentation*. The list of the masks would be calculated by object detection and recognition of deep learning and then be fed into step *Recover Scaled Images in 2D ROI*. This step could restore the scaled ROI and finally output the ist of the masks.

To improve the detecting ability, we can also add some information of depth, please refer to *Instance Segmentation (Colored and Depth Image)*.

5.24 Instance Segmentation (Colored and Depth Image)

The main function of this step is to obtain the list of the masks of the object by object detection and recognition of deep learning. It usually applies when we need to recognize and segment the object. The combination is as *Figure 1* shown.

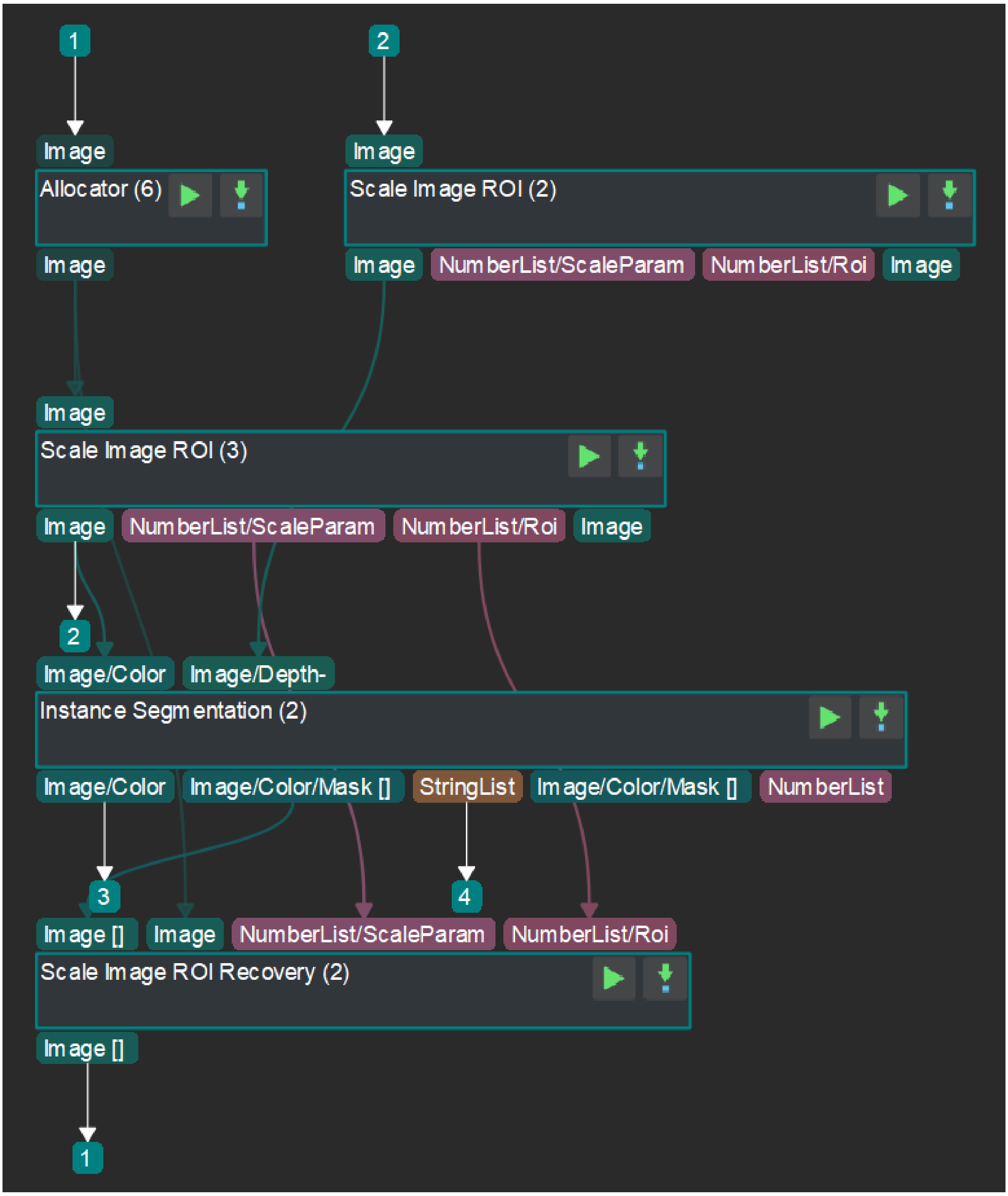


Figure 1. Procedure of instance segmentation(colored and depth image)

The step *scale_image_in_2d_roi* would process the colored and depth image and output the scaled ROI to the step *Instance Segmentation*. The list of the masks would be calculated by object detection and recognition of deep learning and then be fed into step *Recover Scaled Images in 2D ROI*. This step could restore the scaled ROI and finally output the ist of the masks.

Since we add some depth information in this step, the detecting ability should be higher than *Instance Segmentation (Colored Image)*.

5.25 Matching

According to the point cloud situation of the template, Match can be divided into edge template matching and full template matching. According to the sorts of the template, Match can be divided into single template matching and multiple templates matching. This chapter mainly introduces single template matching and multiple templates matching in full template.

Template selection

Template selection mainly decided by point cloud effect,project accuracy, beat and other important factors.

- **Edge template matching apply to:**
    1. Pose output scence without high precision. Such as using flexible fixture.
    2. Scenes with low quality point cloud.
    3. Scenes with certain restrictions on the beat.
- **Full template matching apply to:**
    1. Pose output scence with high precision. Such as using rigidity fixture.
    2. Scenes with high quality point cloud.
    3. For scenes where the beat requirement is not so high.

Selection of matching methods

Select according to the type of template that needs to be matched. Multiple object matching options, multiple template matching. For single object matching, select ordinary 3D matching.

Template making

1. Collect the original point cloud image: Obtain the point cloud image through the camera and save it.

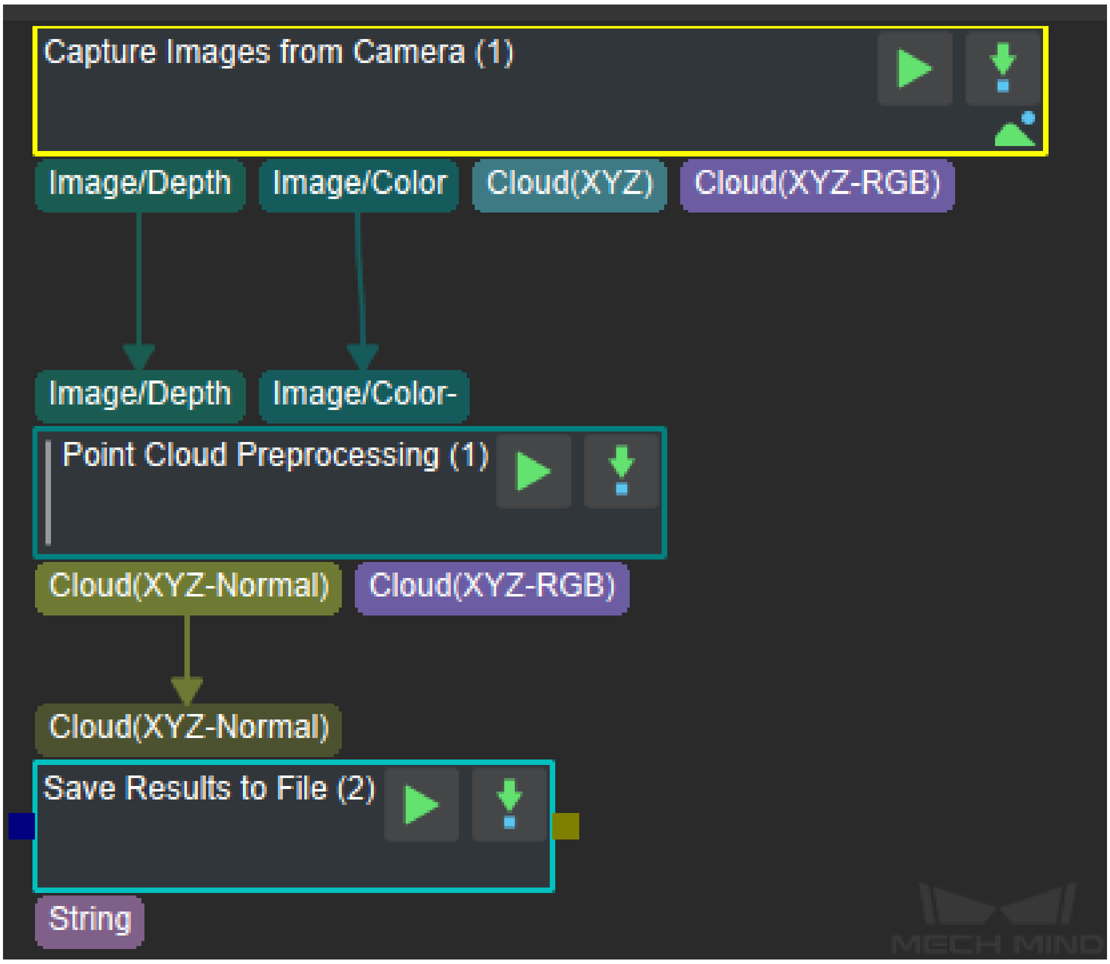


Figure1 Collect the Original Point Cloud Image

The user also can performs the collection operation in Mech-Vision's *Tools* → *Collection Point Cloud Template*.

2. Make point cloud full template file: Open the collected original point cloud image in the *Tools* → *Point Cloud Editor* of Mech-Vision, and edit the point cloud to remove noise.

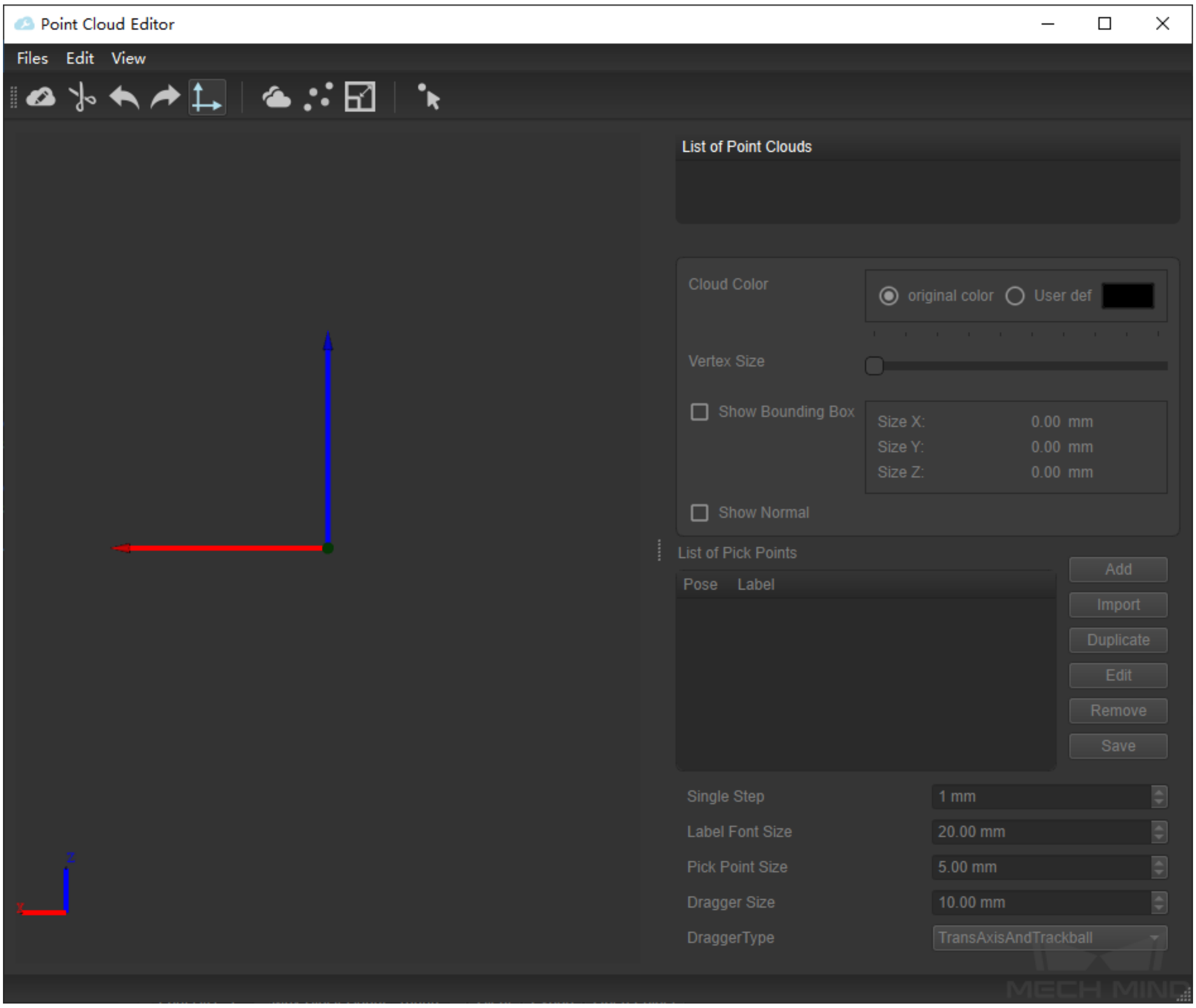


Figure2 Pose Cloud Editor

3. Edge extraction: Roughly edge extraction through the full template file.

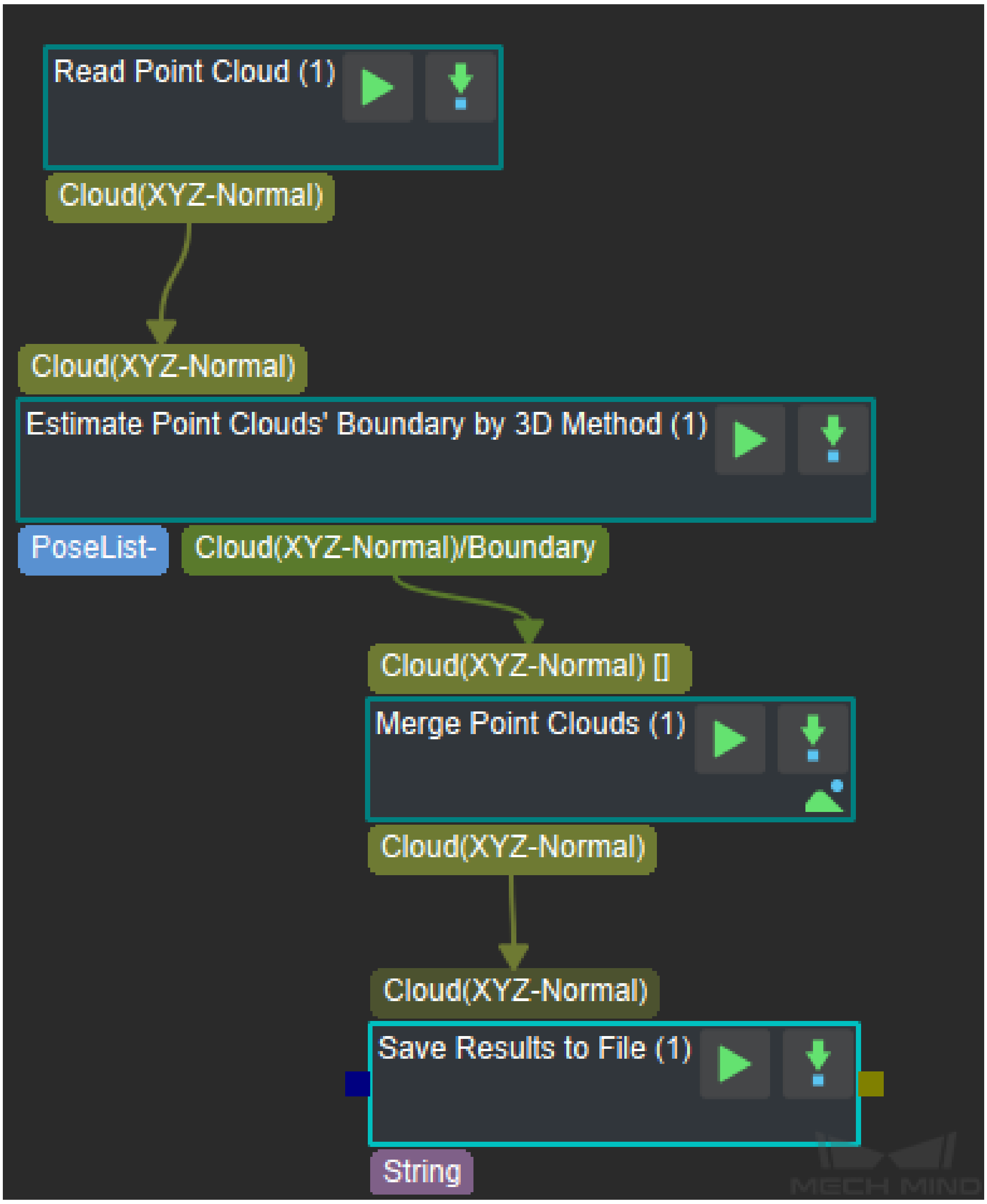


Figure3 Edge extraction

4. Create edge template file: Remove the noise of the point cloud edge file in **Point Cloud Editor** and save it as an edge template file.

Crawl point settings

Refer to *Add Pick Point* .

Matching basic step combination

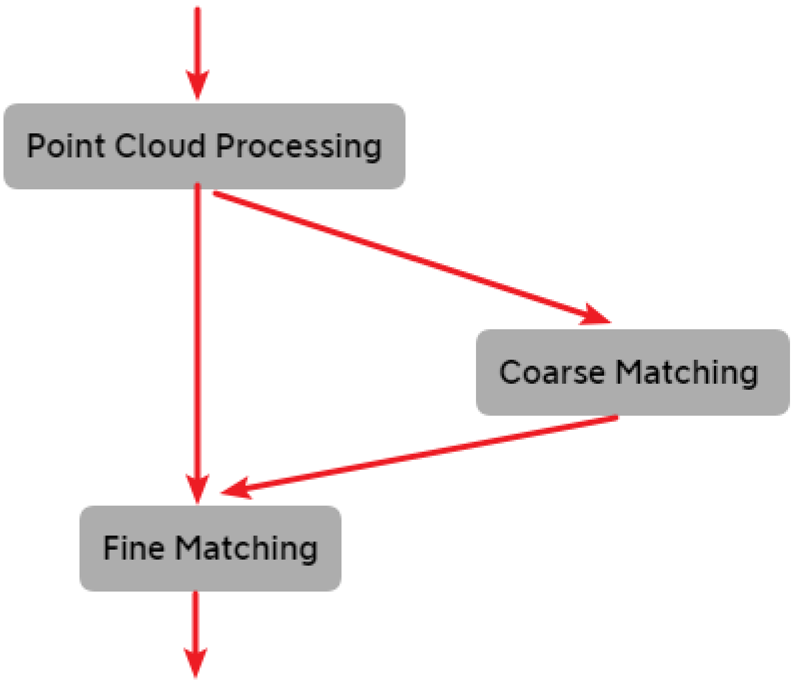


Figure4 Matching

As shown in *Figure 4*, the basic procedure of matching includes three parts: **Point Cloud Processing** , **Coarse Matching** and **Fine Matching** . The specific process is as follows:

1. The final optimization of the point cloud input into the matching is completed by the **Point Cloud Processing** part.
2. **Coarse Matching** through the matching between the template point cloud and the scene point cloud, the candidate pose conversion relationship between the scene point cloud and the template point cloud is calculated.
3. **Fine Matching** Partial matching of the template according to the iterative algorithm can improve the matching accuracy.

5.25.1 3D Matching

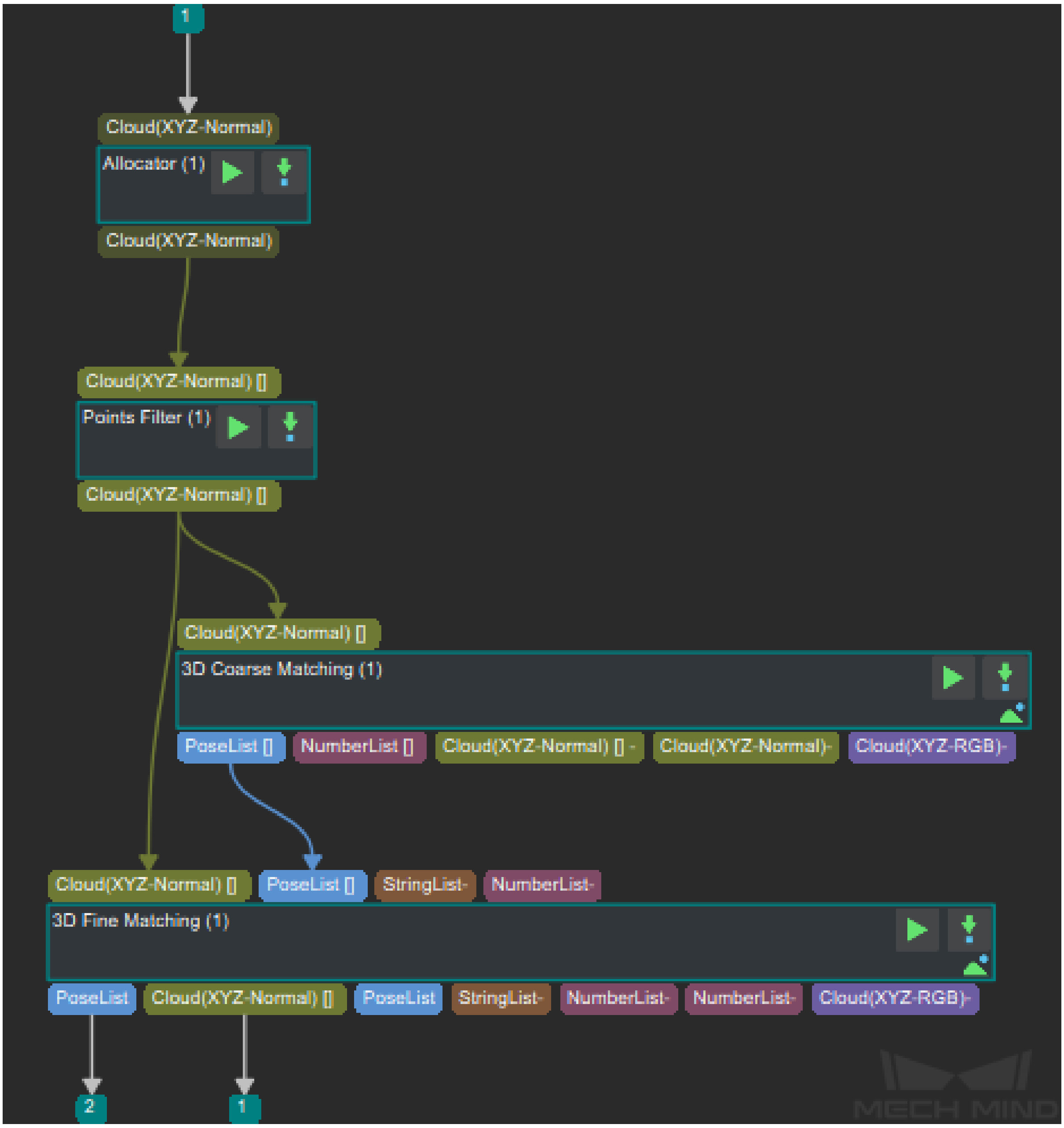


Figure 5 3D Matching

Example result

- Input point cloud image to be matched.

Figure6 Input Point Cloud Image to be Matched.

- Point cloud full template.

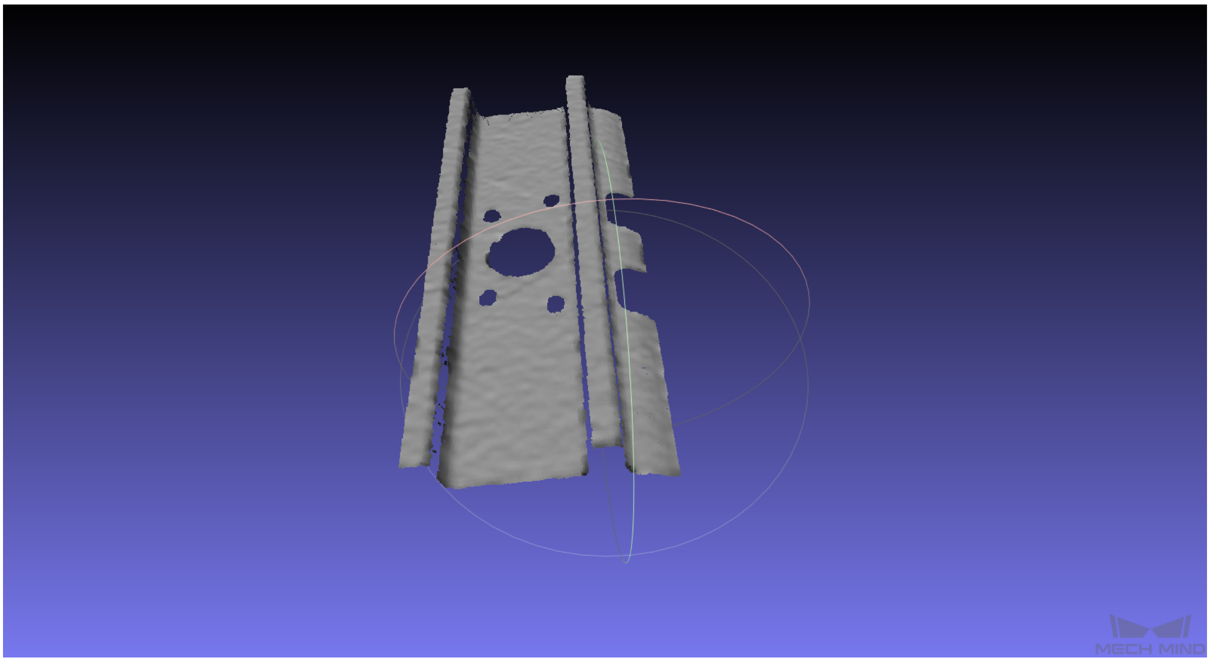


Figure7 Point Cloud Full Template

- 3D matching result

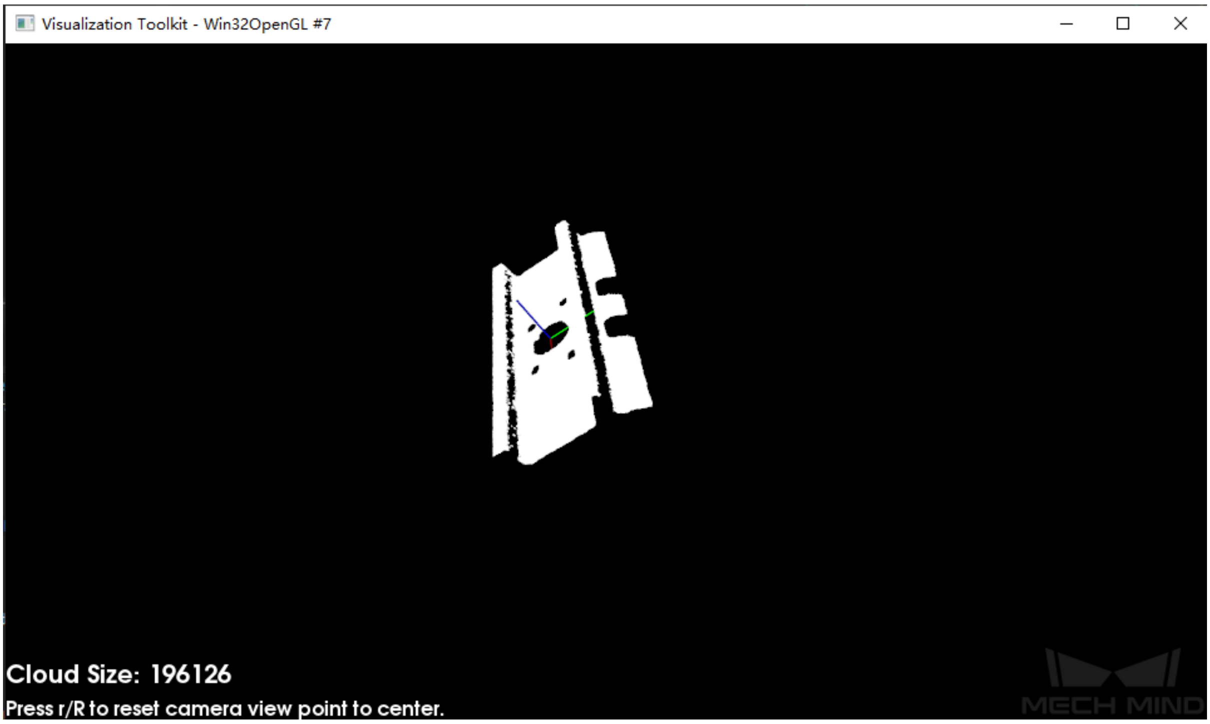


Figure8 3D Matching Result

5.25.2 3D Multiple Template Matching

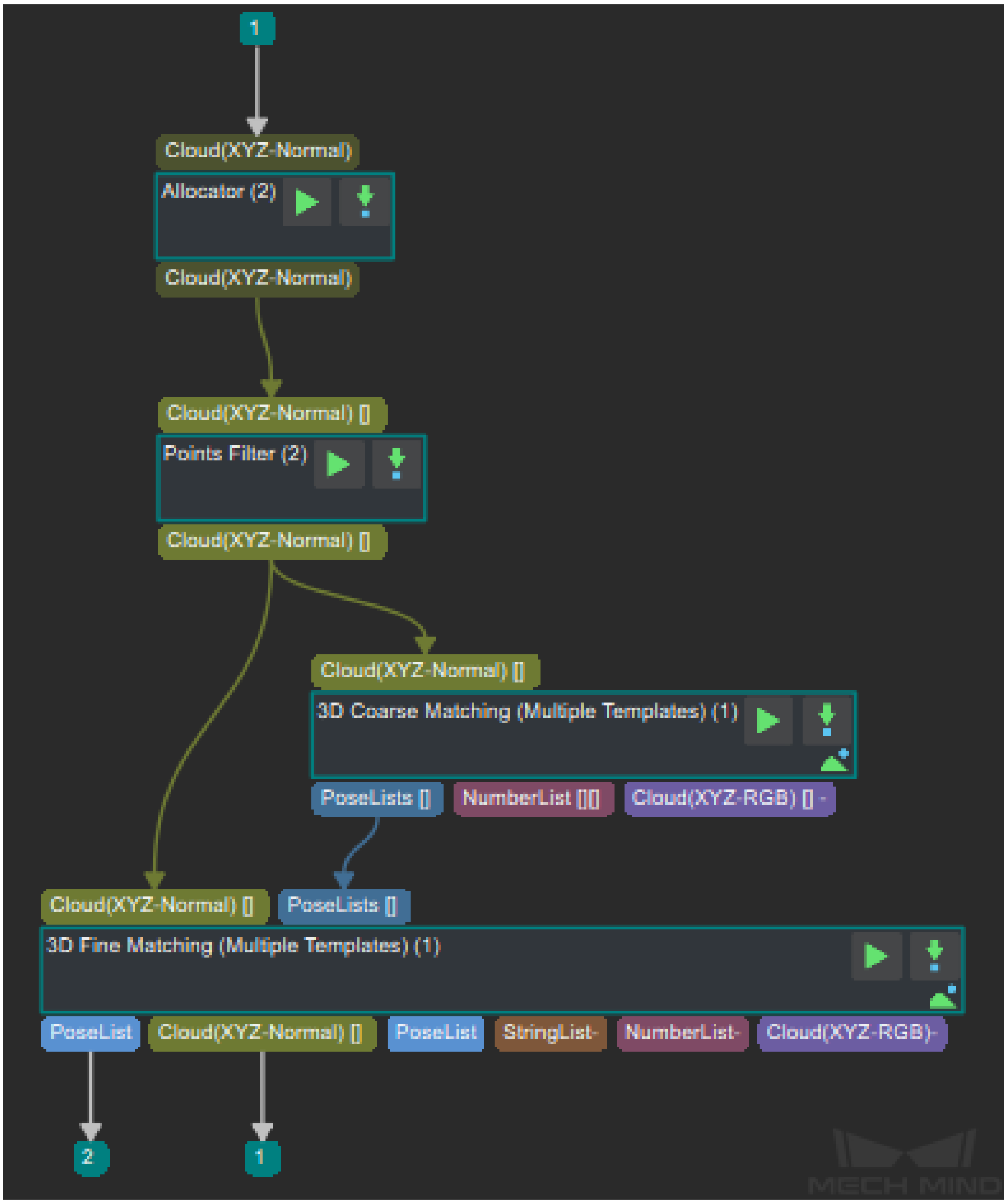


Figure9 3D Multiple Template Matching

Result example

- Input point cloud image to be matched.

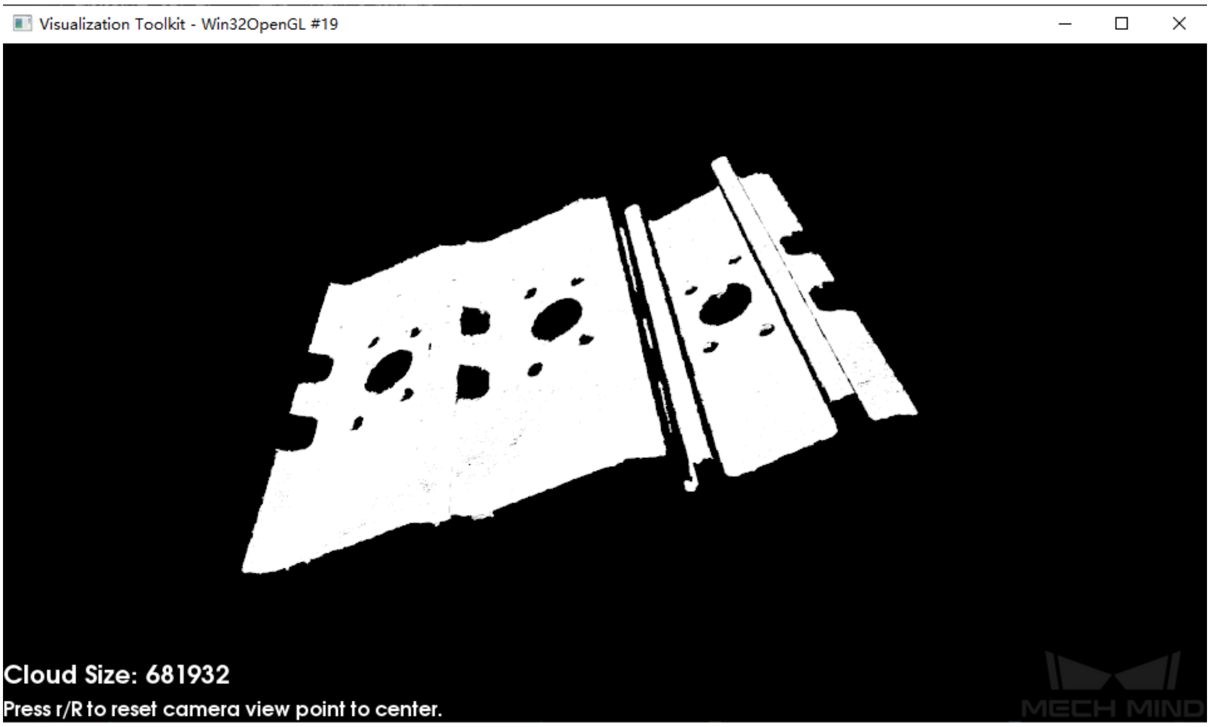


Figure10 Point Cloud Image to be Matched

- Point cloud full template.

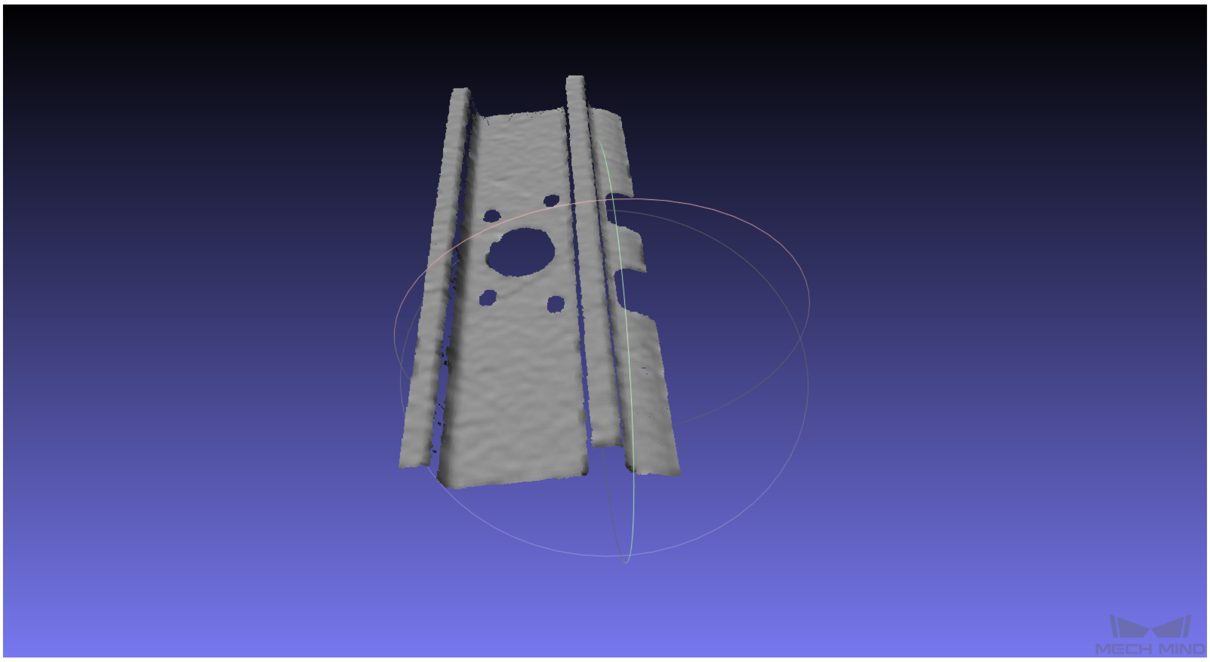


Figure11 Point Cloud Full Template1

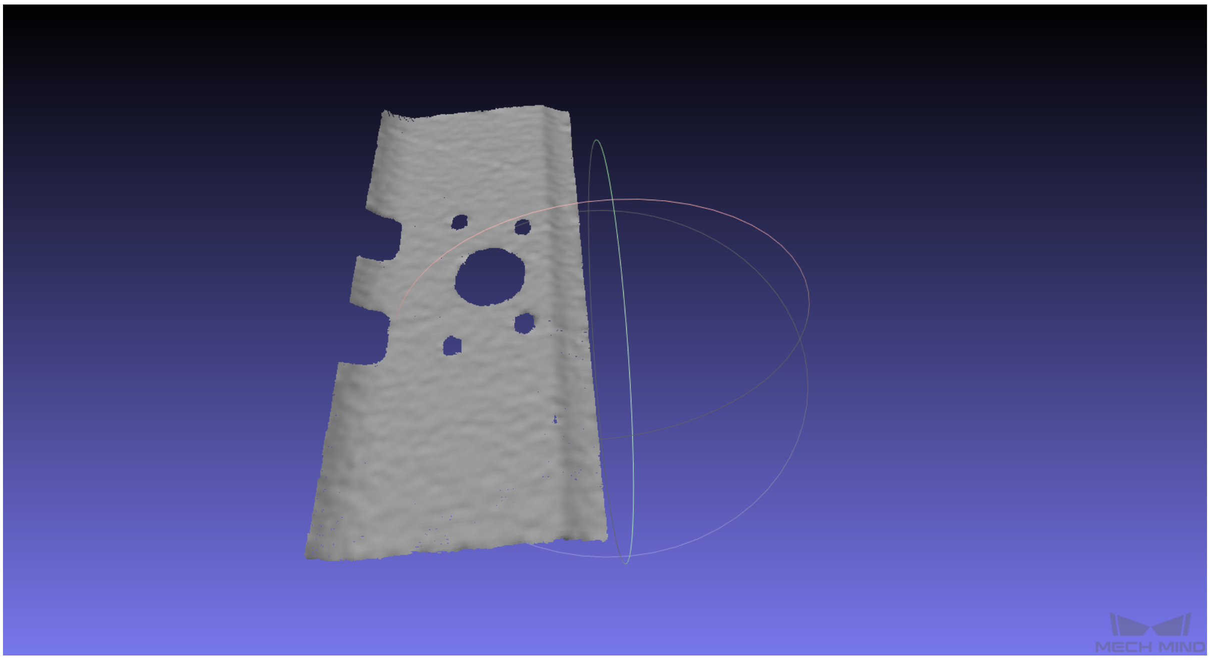


Figure12 Point Cloud Full Template2

- 3D meatching result

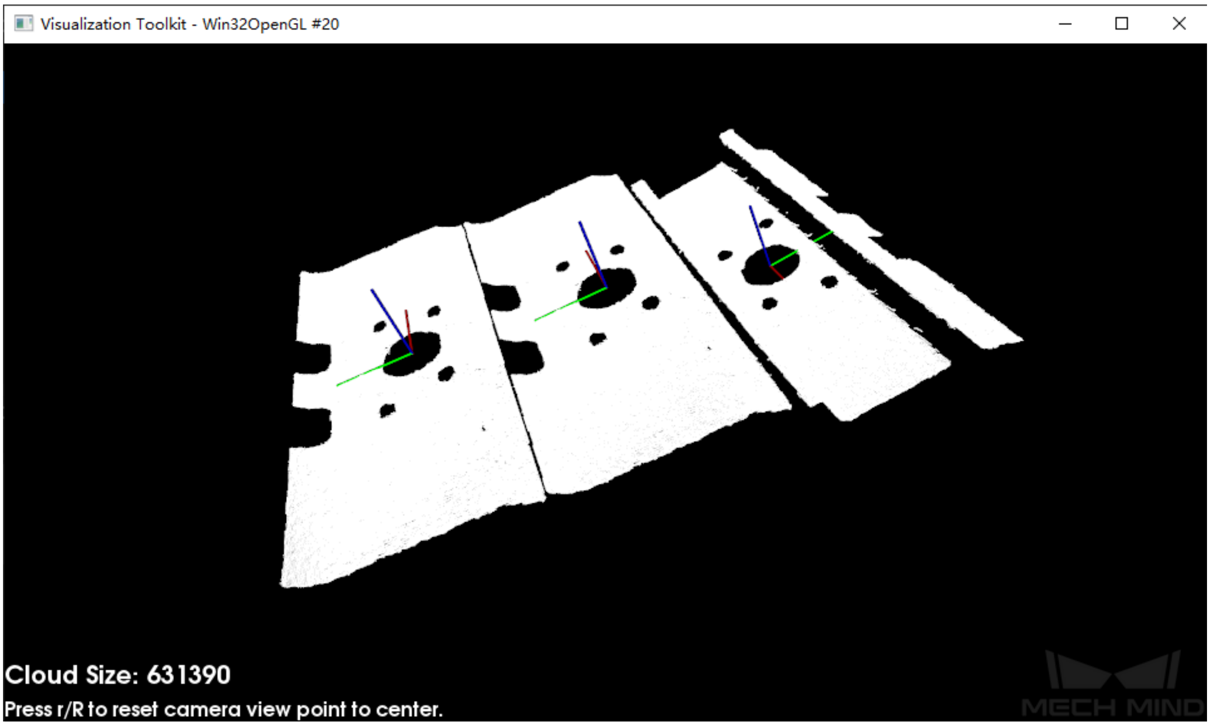


Figure13 3D Meatching Result

5.26 Point Cloud Preprocessing

The main function of this step is to generate the point cloud, remove noises and then obtain the point cloud in the ROI. It usually follows the step *Capture Images from Camera* to accelerate. The combination is as *Figure 1.* shown.

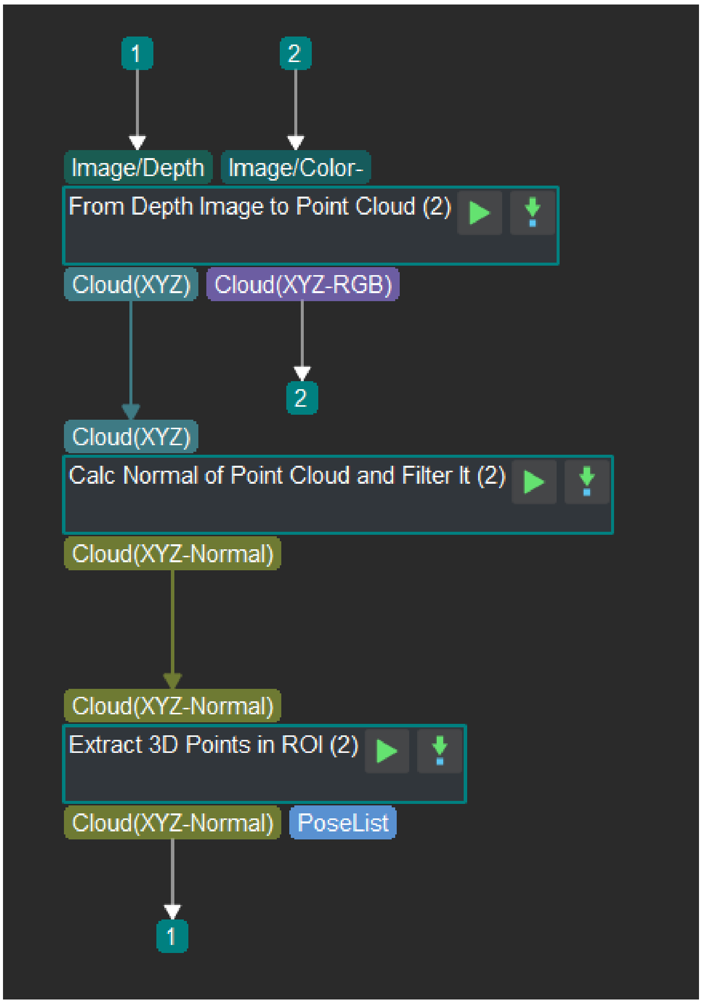


Figure 1. Procedure of point cloud preprocessing

The introduction of each steps is as follows:

1. *From Depth Map to Point Cloud* Input the original colored image and its depth image, and output the point cloud of the current scene.
2. *Calc Normals of Point Cloud and Filter It* Calculate the normal vectors and remove useless points of the point cloud obtained, amd output the filtered point cloud with normal vectors.
3. *Extract 3D Points in 3D ROI* Input the processed point cloud and then extract part of the point cloud in the ROI.

5.27 Predict The Heat Map Of Grasp Probability Of The Pick Points

Predicting the probability distribution of Pick Points is suitable for grabbing unknown objects. Input the depth image and color image, use the deep learning algorithm to distinguish the “graspable’ and “ungraspable” features in the image, and predict the grasp probability, output the heat map of grasp probability. The program of this procedure is shown in *Figure 1*.

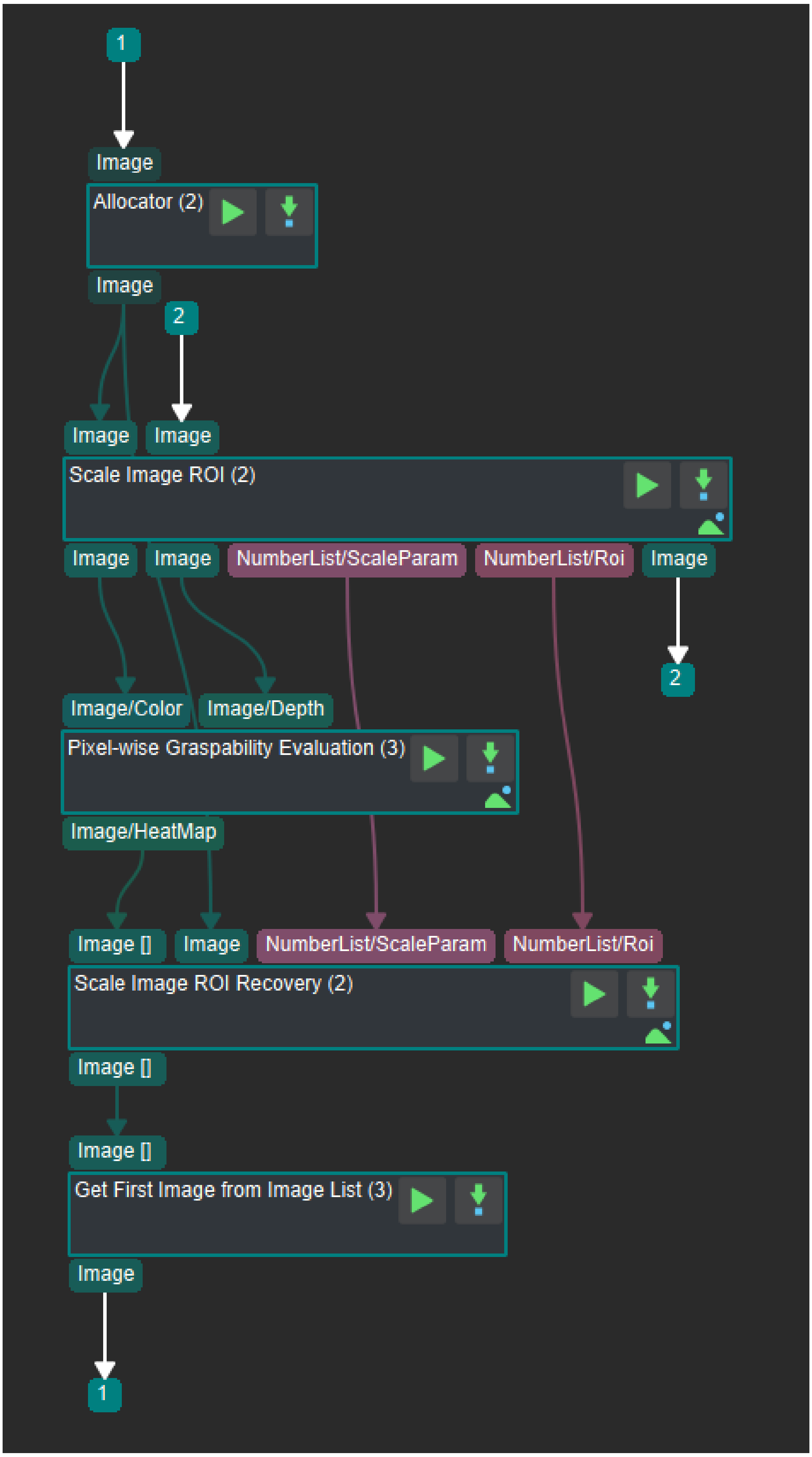


Figure 1 The program of predicting the heat map of grasp probability of the pick points in Mech-Vision

The introduction of each step in the program is as follows :

1. *Scale Image in 2D ROI* : Input the original depth image and color image, intercept the region of interest, zoom to the image size required for deep learning and output the color image and depth image after processing.
2. *Pixel-wise Graspability Evaluation* : Input the result of the previous step, then use the deep learning algorithm to calculate the grasp probability with each pixel as the center of the end effector and obtain the heat map of the grasp probability of the pick points. The color of the pixel from blue to red represents the probability from low to high, that mean the red area is the graspable area predicted by the model, which will be used in the subsequent calculation of the pick points.
3. *Recover Scaled Images in 2D ROI* : Input the result of the previous step and the original color image, restore the scaled heat map of the grasp probability and obtain a list of images.
4. *Get First Image from Image List* : Obtain the image corresponding to the first capture of the heat map of grasp probability.

5.28 Refine The Pick Points By Point Clouds

In the process of grabbing any object, the position and the symmetry-label of the object both need to considered. Therefore, this procedure calculate the relatively precise grabbing position and the direction of the long axis, after adjustment and screening, the pick points which meet the requirements in the region of interest are obtained. The program of this procedure in Mech-Vision is shown in *Figure 1*.

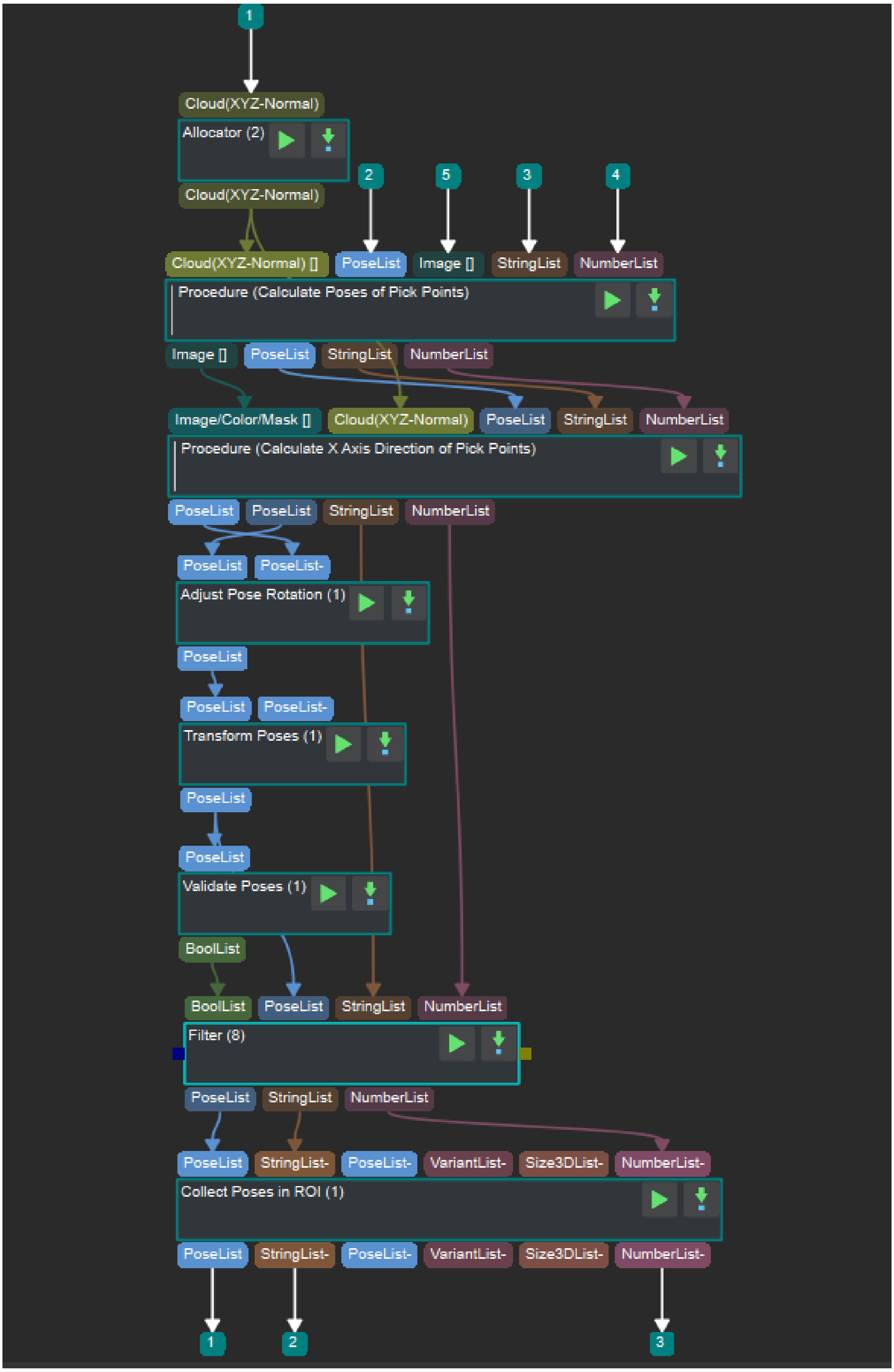


Figure 1 The program of refining the pick points by point clouds in Mech-Vision

The introduction of each step in the program is shown in as follows :

1. *Allocator* : Allocate the inputs to the next few steps.
2. *Calculate The Position Of The Pick Points* : Input the point clouds and the planar pick points list in the region of interest in the scene, calculate the pick points in the space and obtain a relatively precise grabbing position.
3. *Calculate the direction of the X axis* : Input the point clouds and the mask of the objects in the region of interest in the scene, calculate the pick points in the space. The direction of the long axis of the pick points can be determined, since the mask includes the shape information of the object.
4. *Point Axes of Poses to Given Direction* : Input the pick points list obtained in the above two steps, adjust the posture according to the settings, determine the pick points position and the direction of the long axis, then output the pick points list.
5. *Transform Poses* : Input the result of the previous step and obtain the representation fo the pick points in the robot coordinate system.
6. *Validate Poses by Included Angle to Reference Direction* : Input the result of the previous step, then calculate the angle between the Z axis of the pick points and the reference direction, after that compare with the threshold in settings, if it is less then "True" otherwise "False", finally output the list of Boolean values.
7. *Filter* : Input the list of Boolean values, the pick points in the robot coordinate system and the label and the size of the gripper correspond to the pick points, output the result corresponding to "True".
8. *Collect Poses in 3D ROI* : Input the result of the previous step, obtain the pick points, their tags and their confidence level in the region of interest according to the file. This step must run under the robot coordinate system.

5.28.1 Calculate The Position Of The Pick Points

The program of this procedure is shown in *Figure 2* 。

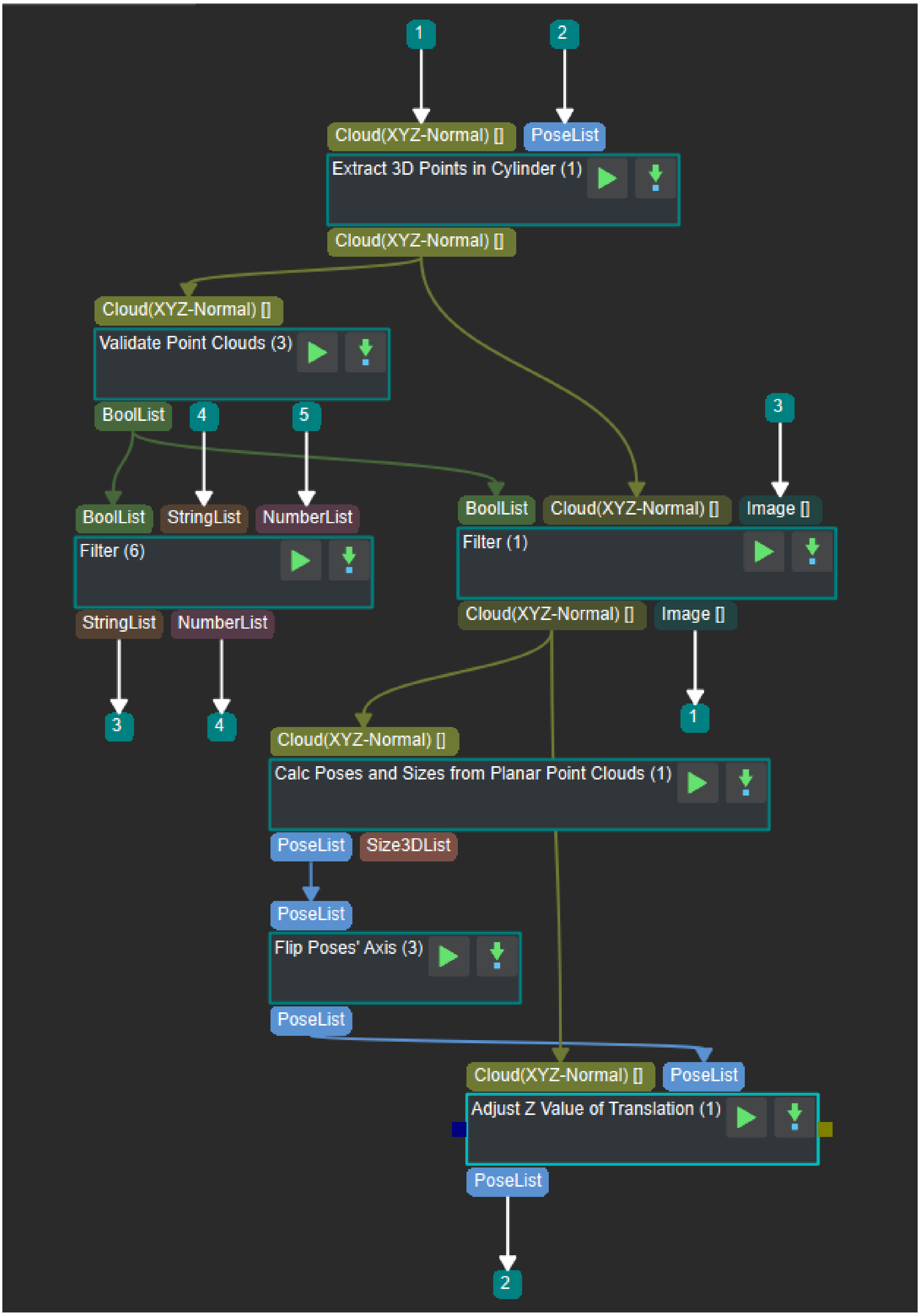


Figure 2 The program of calculating the position of the Pcik Points in Mech-Vision

The introduction of each step in the program is as follows :

1. *Extract 3D Points in Cylinder* : Input the point clouds and the planar pick points list in the region of interest in the scene, extract the point clouds in the set cylinder.
2. *Validate Point Clouds* : Input the result of the previous step, compare the number of points of each point clouds with the set threshold. If the result is within the threshold range it is "True", otherwise it is "False". Output the list of Boolean values.
3. *Filter* : Input the list of Boolean values, the label corresponding to the pick points and the size of the gripper, output the result corresponding to "True". This step can filter out the label and the size of gripper corresponding to the empty point clouds.
4. *Filter* : Input the list of Boolean values, the point clouds in the cylinder and the mask of the object, output the result corresponding to "True". This step can filter out the the empty point clouds and the mask of the object corresponding to the empty point clouds.
5. *Calc Poses and Dimensions from Planar Point Clouds* : Input the point clouds with the required number of points, calculate the spatial Pick Point corresponding to each point cloud and output it.
6. *Flip Poses' Axes* : Input the list of pick points, flip it according to the direction of the specified coordinate axis to make the direction as the same, which is convenient for subsequent grabbing, output the list of pick points after flipping.
7. *Move Poses to Point Cloud Surfaces along Z-Axis* : Input the point clouds in the ROI and the list of pick points after flipping, calculate the distance between the Pick Point and the point clouds area corresponding to it, move the Pick Point to the object surface according to the rules settings.

5.28.2 Calculate the direction of the X axis

The program of this procedure is shown in *Figure 3*.

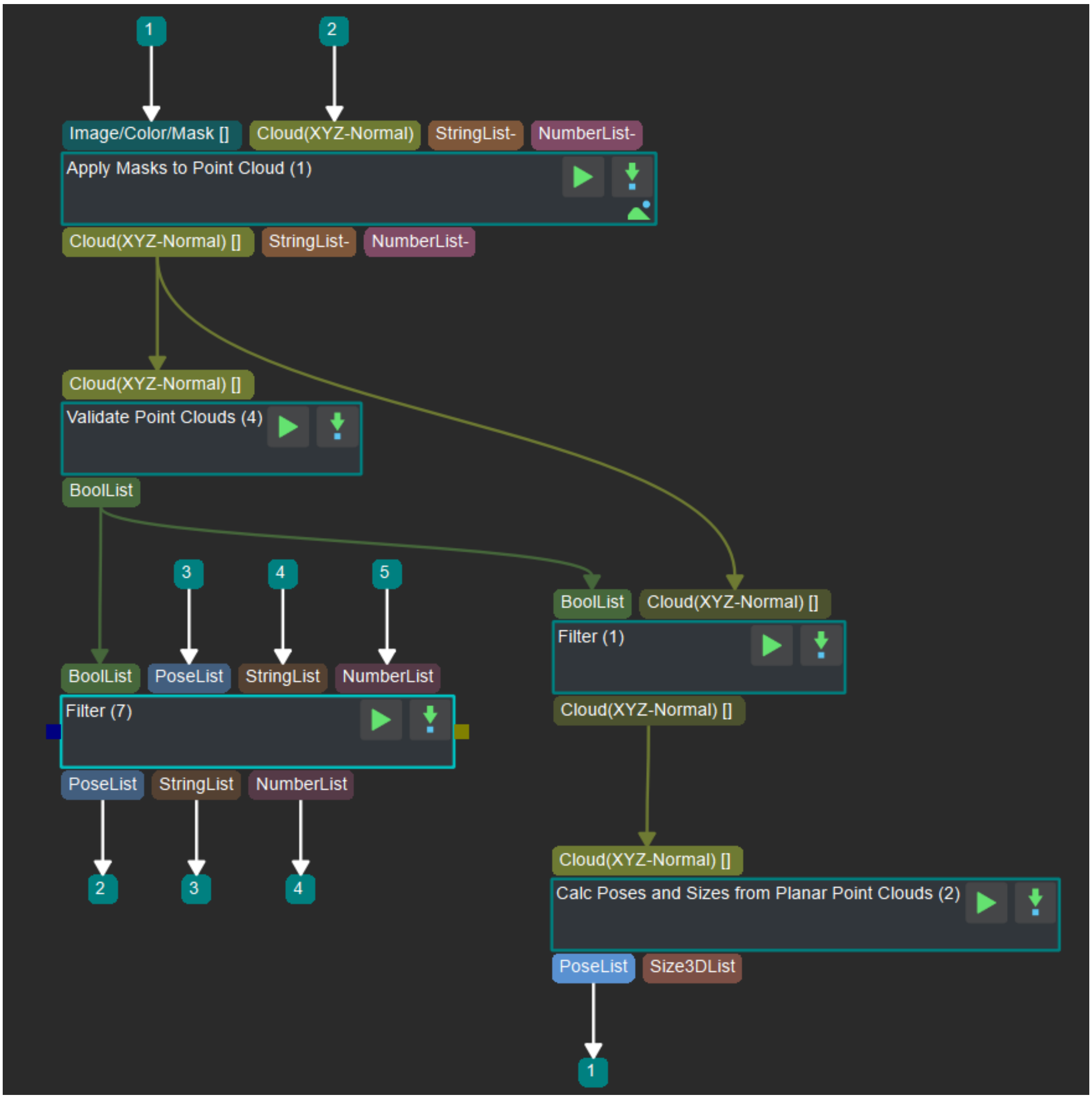


Figure 3 The program of calculating the direction of the X axis in Mech-Vision

The introduction of each step in the program is as follows :

1. *Apply Masks to Point Cloud* : Input the mask of the scene and the point clouds in the region of interest, calculate the point clouds corresponding to the mask and output it.
2. *Validate Point Clouds* : Input the result of the previous step, compare the number of points of each point cloud with the set threshold. If the result is within the threshold range it is "True", otherwise it is "False". Output the list of Boolean values.
3. *Filter* : Input the list of Boolean values, the label and the size of the gripper corresponding to the pick points (with accurate position), output the result corresponding to "True". This step can filter out the label and the size of gripper corresponding to the empty mask.

4. *Filter* : Input the list of Boolean values and the point clouds corresponding to the mask, output the result of "True" . This step can filter out the empty mask.
5. *Calc Poses and Dimensions from Planar Point Clouds* : Input the result of the previous step, calculate the pick points corresponding to each point cloud.

5.29 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 as *Figure 1.* shown.

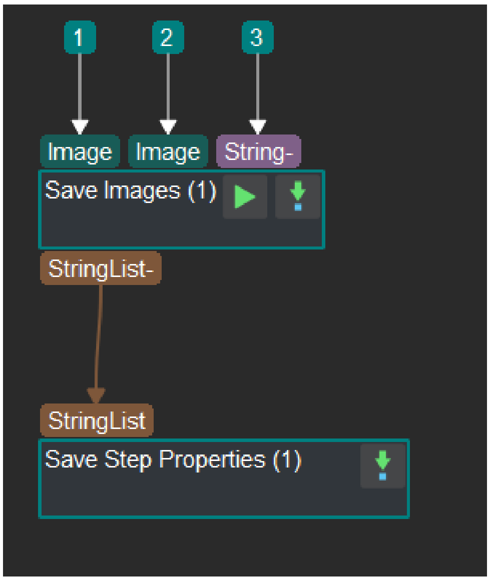


Figure 1. Saving images and step parameters

Input images to be saved, including colored and depth images and others, into step *Save Images*. Users can input multiple images together and add indexes to image names if needed.

The step *Save Step Parameters to File* is optional and users can remove it if not needed.

5.30 Send The Point Clouds To Server

This procedure can merge the point clouds and send it to Mech-Viz for visualizing the trajectory planning. The program of this procedure is shown in *Figure 1.*

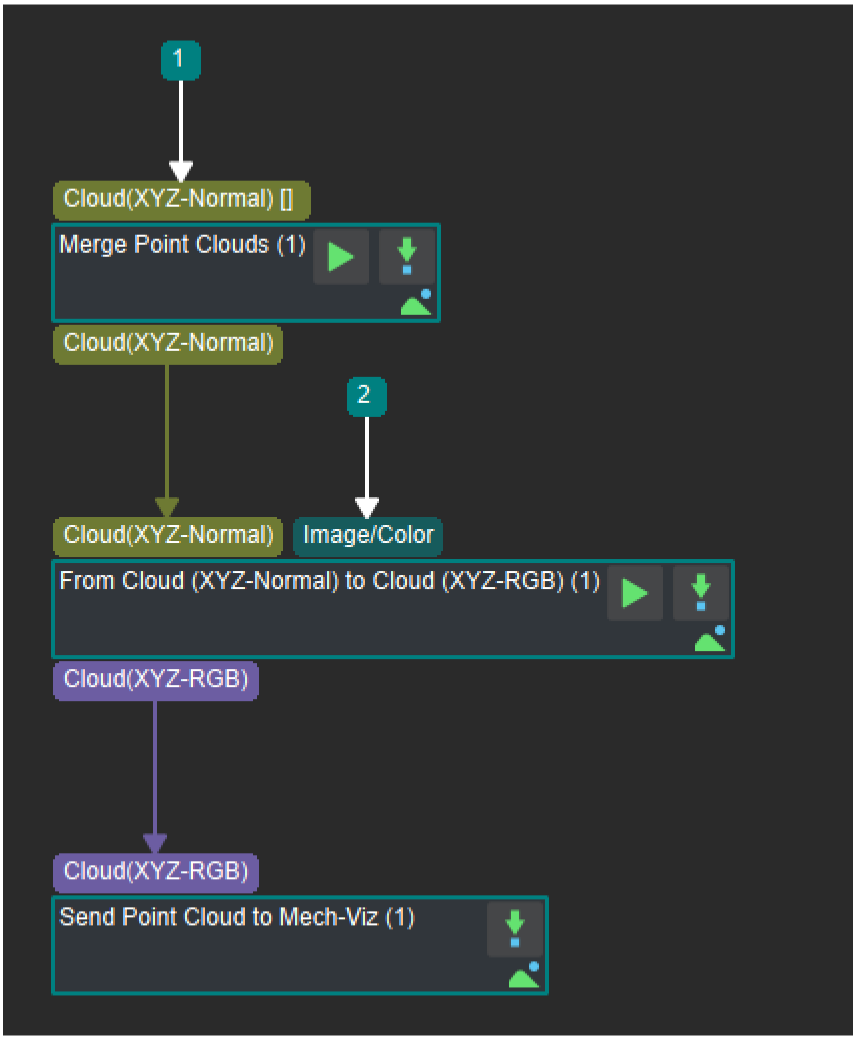


Figure 1 The program of sending the point clouds to server in Mech-Vision

The introduction of each step in the program is as follows :

1. *Merge Point Clouds* : Merge multiple point clouds with normal vector into one and output it.
2. *From Cloud (XYZ-Normal) to Cloud (XYZ-RGB)* : Input the result of the previous step and the corresponding color image, generate color point clouds and output it.
3. *Send Point Cloud to External Service* : Send the result of the previous step to Mech-Viz.

5.31 Sort Pick Points

5.31.1 Functions

This Procedure converts poses under the camera coordinate system to the robot coordinate system, and 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 as shown in *Figure 1*

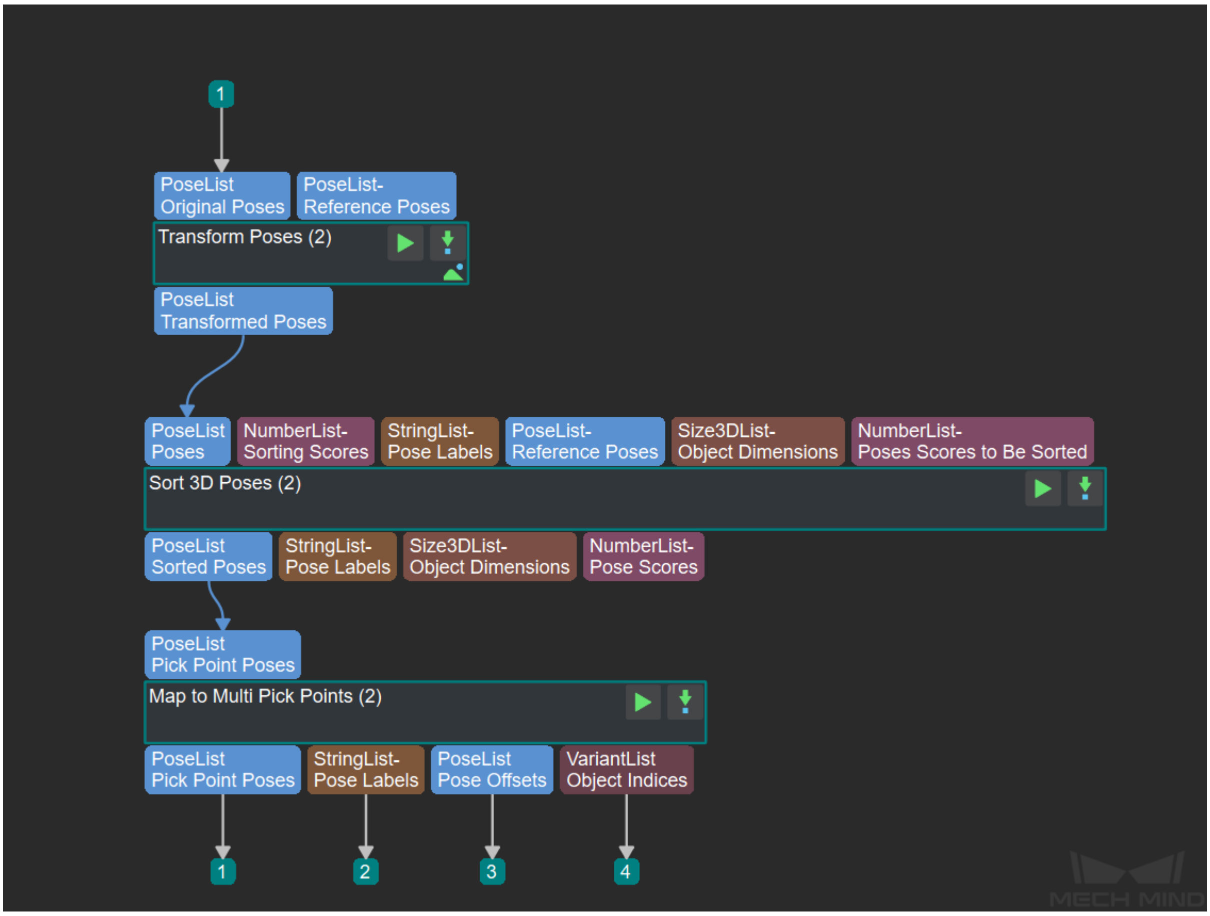


Figure 1. Graphical programming of **Sort Pick Points**

5.31.2 Functions of Involved Steps

The functions of the Steps involved are as follows (please click on the links for detailed descriptions of the Steps):

- *Transform Poses* transforms the input pick point poses under the camera coordinate system to the robot coordinate system 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, sorting by the pose scores in ascending order, etc.
- *Map to Multi Pick Points* maps each sorted pose to multiple pick points based on a pick point model to increase the feasibility of picking.

5.31.3 Usage Example

Figure 2 gives an example of transforming poses under the camera coordinate system to the robot coordinate system, sorting poses by the Z values in descending order, and mapping each pose to multiple pick points for output in the end.

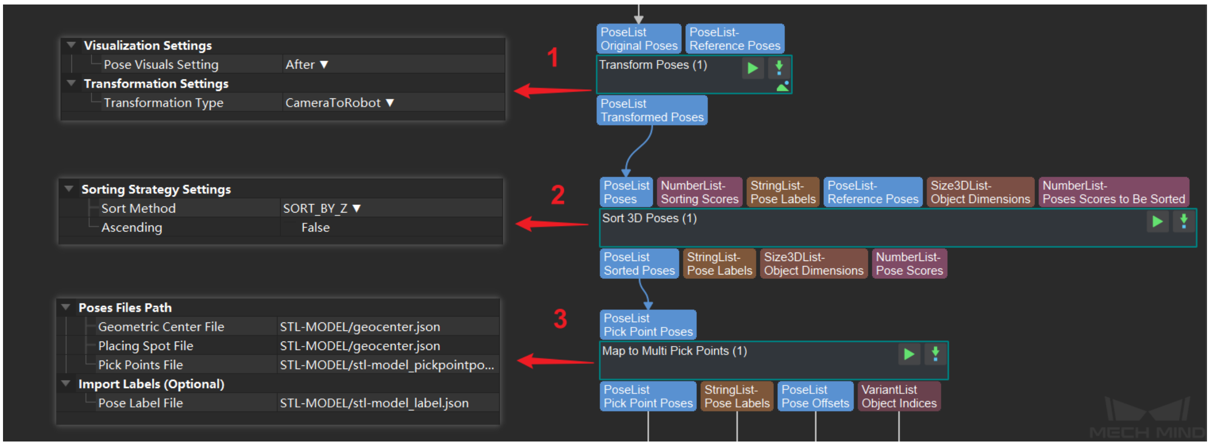


Figure 2. Usage example

1. Transform poses to the robot coordinate system.
2. Sort poses by the Z values in descending order.
3. Map each pose to multiple pick points.

The upper and lower part of *Figure 3* shows the poses and the mapped multiple pick points respectively.

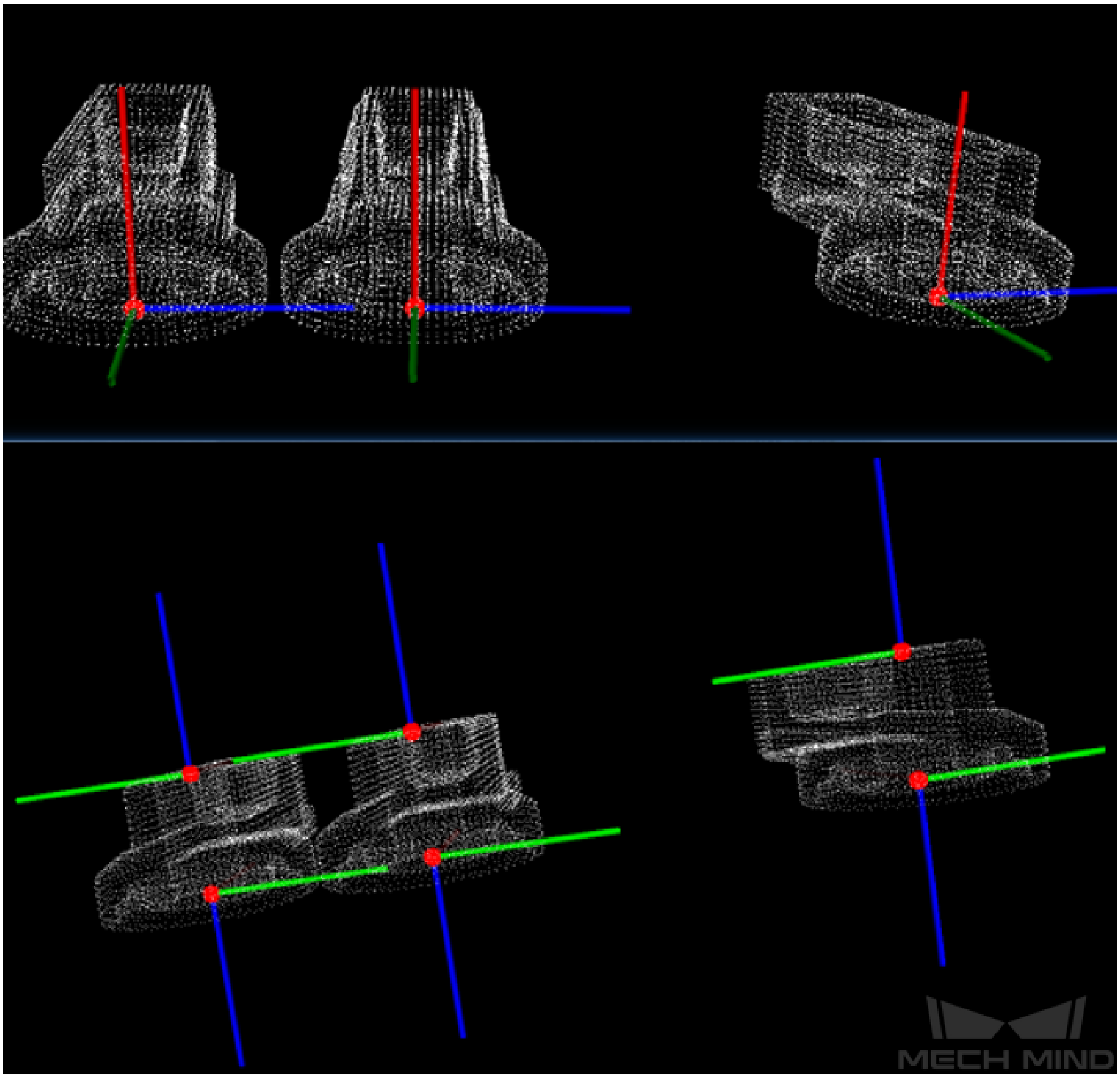


Figure 3. Poses (upper) and mapped multiple pick points (lower)

5.32 Sort The Pick Points (within and between layers)

For a complex situation where a variety of objects are scattered, it’s necessary to consider the height of the object and the probability of successful grasp. To solve this problem, this procedure stratifies the spatial pick points by their height and obtain the corresponding planar pick points. Then sort the pick points of each layer according to the specified rules. After that the index list that meet the requirements is obtained. Finally the procedure sort the spatial pick points by the index list. The program of this procedure is shown in *Figure 1*.

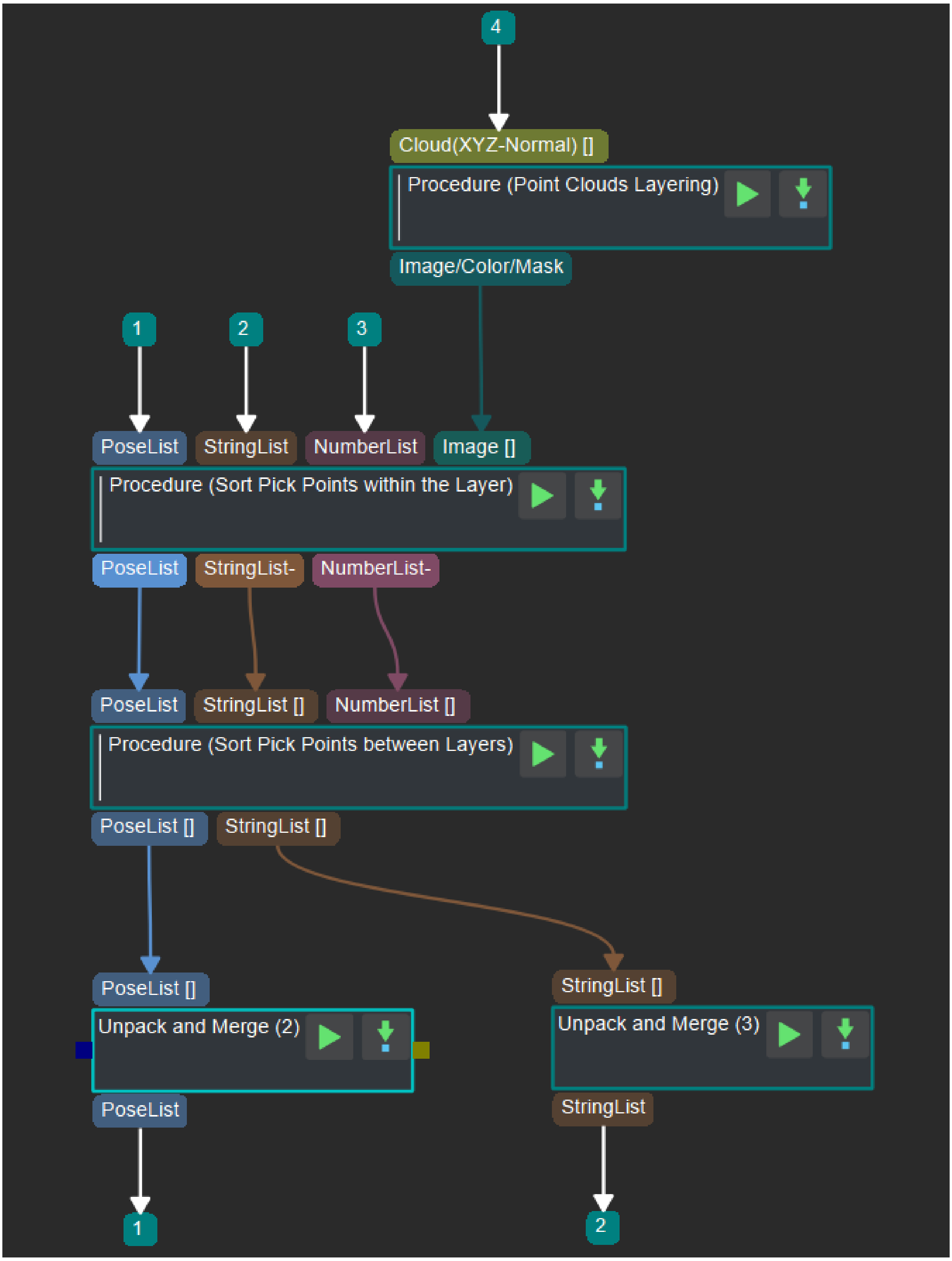


Figure 1 The program of sorting the pick points in Mech-Vision

The introduction of each step in the program is as follows :

1. *Layer Point Clouds* : Input the point clouds in the region of interest, stratity it by their height, project it to the plane and obtain the mask of the objects in different heights.
2. *Sort Pick Points Within The Layer* : Input the result of the previous step, the spatial pick points and their labels and confidence level to obtain the planar pick points which will be put into different groups and sorted inside the group by specified rules. Output the list of the pick points after sorting with their labels and their confidence level.
3. *Sort Pick Points Between Layers* : Input the result of the previous step, extract the Z value of the first pick point in each layer, use the index to sort the different layers by the specified rules. Output the list of the pick points after sorting with their labels and their confidence level.
4. *Unpack and Merge Data* : Input the pick points list (label list) obtained in the previous step, unpack it and merge it into a new pick points list (label list) then output it.

5.32.1 Layer Point Clouds

The program of this procedure is shown in *Figure 2*.

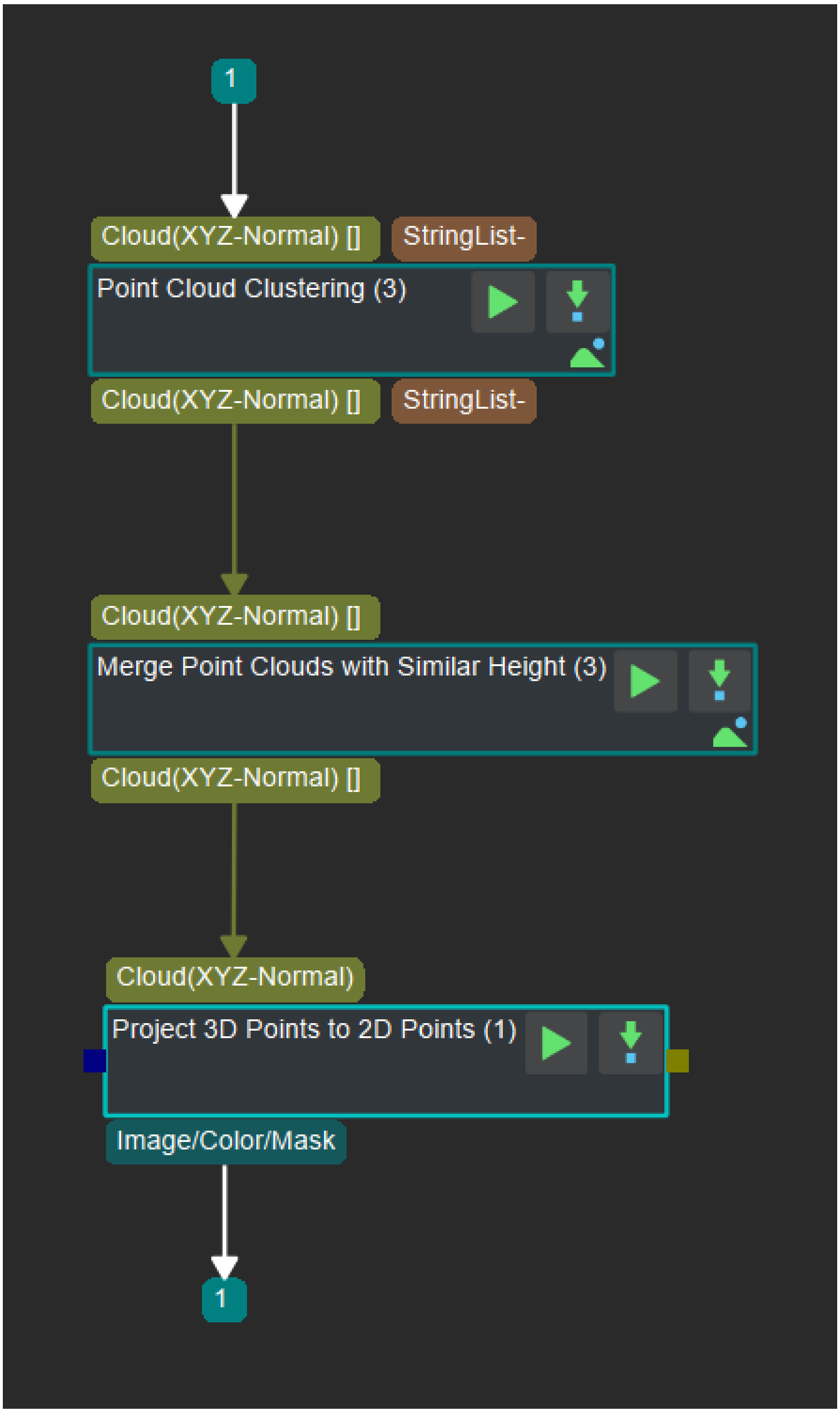


Figure 2 The program of laying point clouds

The introduction of each step in the program is as follows :

1. *Point Cloud Clustering* : Input the point clouds in the region of interest, classify the point clouds which meet the rules into the same category and output the list of the point clouds.
2. *Merge Point Clouds with similar height* : input the result of the previous step, merge multiple point clouds within a certain height range in the specified direction into one point cloud, then output the list of point cloud after merging.
3. *Project 3D Point Cloud to 2D Image* : Input the result of the previous step, calculate the projection of each 3D point and obtain the corresponding 2D image (the mask) then output it.

5.32.2 Sort Pick Points Within The Layer

The program of this procedure is shown in *Figure 3*.

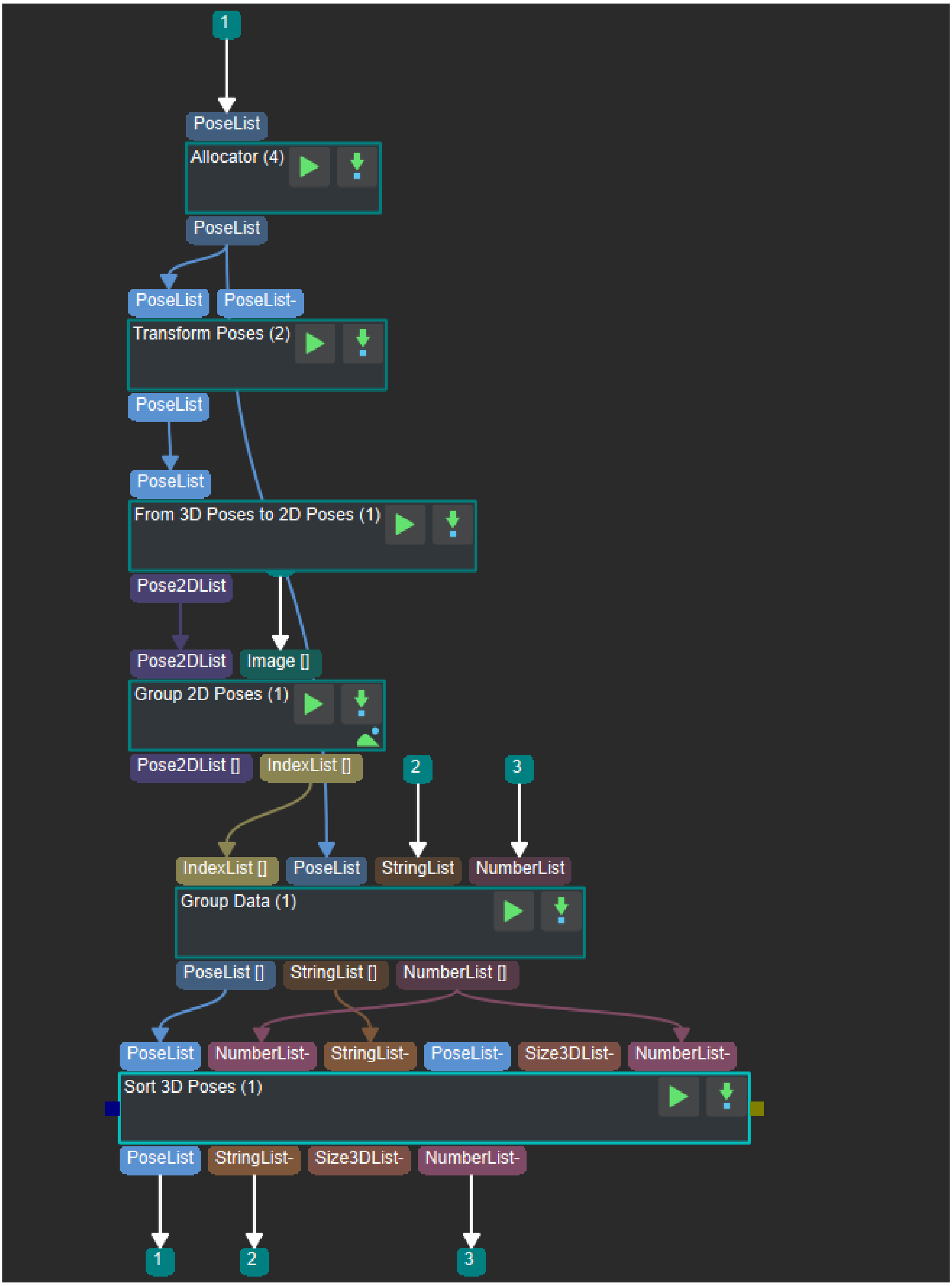


Figure 3 The program of sorting the pick points in the same layer in Mech-Vision

The introduction of each step in the program is as follows :

1. *Allocator* : Allocate the input to the next few steps.
2. *Transform Poses* : Input the pick points list, transform it from the robot coordinate system to the camera coordinate system and output it. This step is for the calculation of the planar pick points.
3. *From 3D Poses to 2D Poses* : Input the result of the previous step, calculate the corresponding 2D pick points and output it.
4. *Group 2D Poses* : Input the 2D pick points list and the mask of the objects at different heights, put the pick points which are in the same mask into the same group and output the corresponding index list.
5. *Group Data* : According to the index list from the previous step, make different groups for the pick points in the robot coordinate system, their confidence level and their labels then output it.
6. *Sort 3D Poses* : Input the result of the previous step, sort the pick points, their confidence level and their labels according to the specified rules and then output it.

5.32.3 Sort Pick Points Between Layers

The program of this procedure is shown in *Figure 4*.

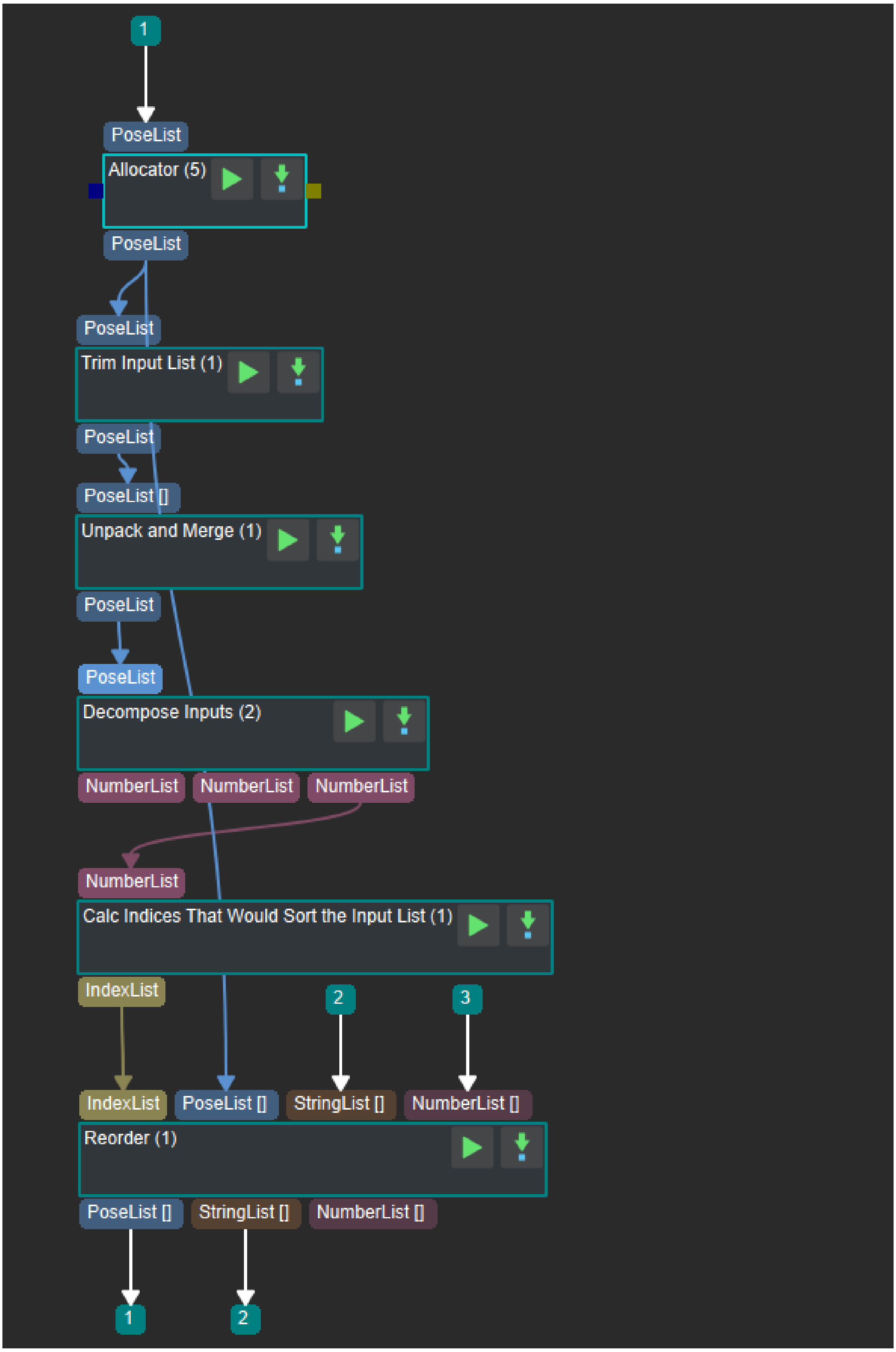


Figure 4 The program of sorting the pick points in different layers in Mech-Vision

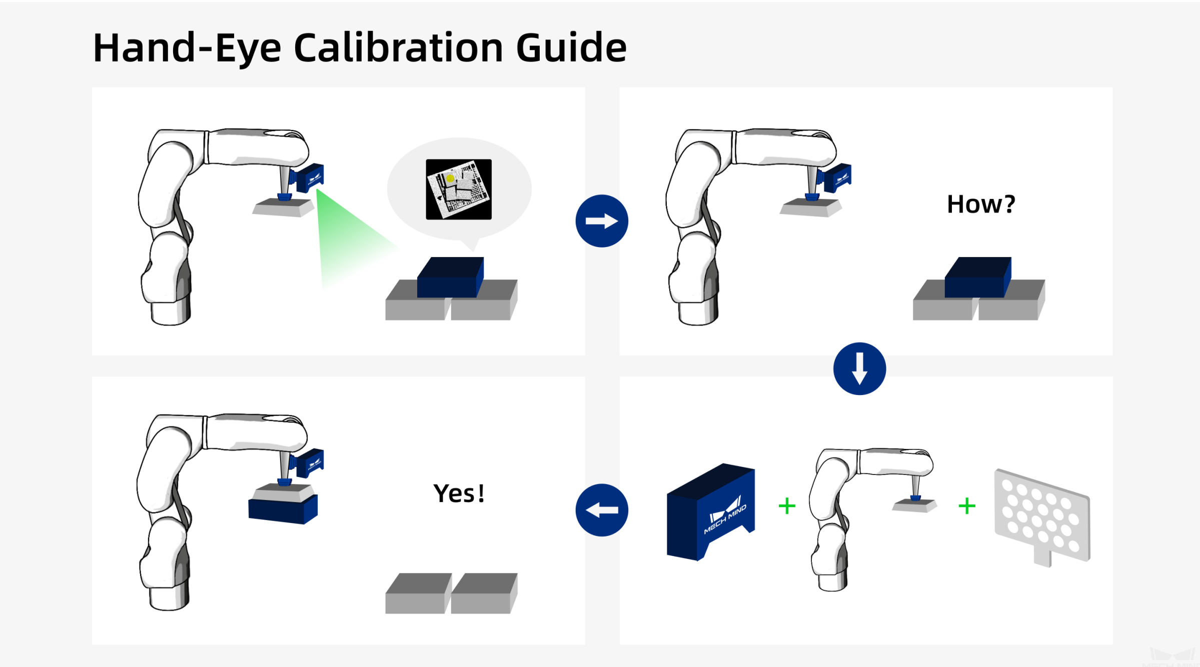
The introduction of each step in the program is as follows :

1. *Allocator* : Allocate the input to the next few steps.
2. *Trim Input List* : Input the pick points list after layering and sorting, output the first few elements in each layer according to the settings.
3. *Unpack and Merge Data* : Input the pick points list obtained in the previous step, unpack it and merge it into a new pick points list then output it.
4. *Decompose Object Dimensions* : Input the result of the previous step, put the X, Y, Z values of each pick point into three different lists and output them.
5. *Sort and Output Index List* : Input the list corresponding to the Z value of each pick point, sort it by the specified rules and then output the corresponding index list.
6. *Reorder by Index List* : Sort the ordered pick points in the same layer, their labels and their confidence level according to the index list and then output them.

CHAPTER

SIX

HAND-EYE CALIBRATION GUIDE



Hand-eye calibration **establishes the transformation relationship between the camera and robot coordinate systems**. With this relationship, the object pose determined by Mech-Vision can be transformed into the robot coordinate system and used to guide the robot to perform its tasks. To achieve the coordinate transformation from the target point on the image to the pick point on the actual object, accurate intrinsic and extrinsic parameters of the camera are indispensable.

What are the 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.

What are the extrinsic parameters?

> The extrinsic parameters describe the pose transformation between the robot coordinate system and the camera coordinate system. 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.

You will need to complete some **necessary preparation** before starting the calibration. See the section below for more information.

Preparation for Calibration

Mech-Vision provides two calibration modes: standard and quick. For the first time, please use the standard mode to perform the calibration.

Calibration - standard mode

If the workstation camera is replaced, you will need to calibrate the new camera. The quick mode can be used to conveniently calibrate a new camera of the same model.

Calibration - quick mode

Once you have completed the calibration, refer to the following section for **evaluating calibration results**.

Inspect and Analyze Calibration Results

Please refer to the following section for solutions to **common problems in calibration**.

FAQ

6.1 Preparation for Calibration

Depending on the relative position between the camera and the robot, the hand-eye calibration can be divided into three types:

1. Eye To Hand (ETH): the camera is installed on a stationary stand independent from the robot.
2. Eye In Hand (EIH): the camera is installed on the flange located at the end of the robot.
3. Eye To Eye (ETE): two cameras are installed on stationary stand(s) independent from the robot.

6.1.1 Install/Place the Calibration Board

1. Choose a calibration board: pick one whose circles are clearly visible and without obvious scratches or deformations.
2. For ETH: mount the robot-specific bracket for calibration board onto the robot flange, and then install the calibration board onto the bracket. Make sure that the calibration board is rigidly attached, and that the board is parallel to the XY plane of the robot's flange coordinate system.

Note: If an undetachable end effector is connected to the robot flange, you can attach the calibration board directly to the end effector.

3. For EIH: place the calibration board in the center of the object plane, where target objects are to be placed.

   Figure 1 below shows the installation/placement of the calibration board for ETH (left) and EIH (right).

Figure 1. Installation/Placement of the calibration board

4. Move the robot to the starting point for calibration after you have installed/placed the calibration board. *Figure 2* shows the starting point for ETH (left) and EIH (right).

   - For ETH: the starting point is the center of the camera's field of view, at the lowest height in the robot's task space (the robot will move up during calibration).
   - For EIH: the starting point is the camera's scanning position, at the lowest height in the robot's task space (the robot will move up during calibration).

Figure 2. Robot starting point for calibration

Note: If you use the TCP touch method to perform calibration, then the calibration board should be placed on the object plane for both ETH and EIH.

6.1.2 Connect to the Robot

The following steps only apply to robots that can be connected through Mech-Center.

1. Connect to the robot according to the process in connect_real_robot.
2. Check the robot’s connection status, that is, check if the camera, robot and IPC are successfully connected via the router. Make sure that the robot can be controlled by Mech-Viz and its pose is consistent with the simulation results of Mech-Viz.
3. Once the robot is successfully connected, an icon will show up in Mech-Center, as shown in *Figure 3* (The displayed robot model should match the actual robot).

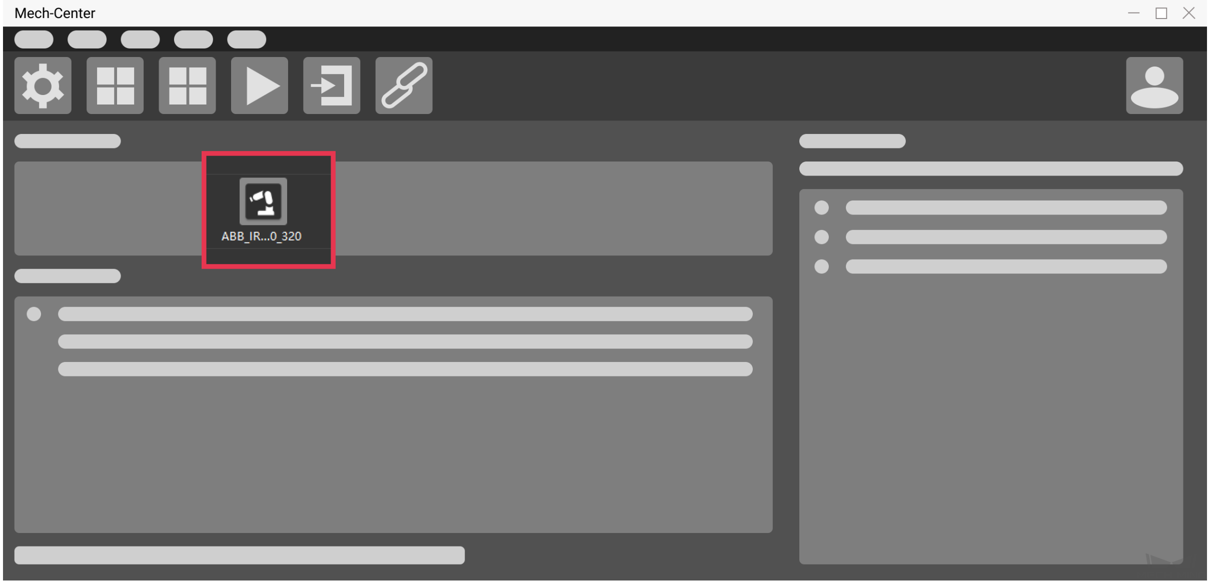


Figure 3. Mech-Center interface after robot connection

6.1.3 Adjust Camera Settings for Point Cloud Generation of the Calibration Board

1. Open Mech-Eye Viewer and adjust camera settings, please refer to Mech-Eye Viewer for detailed instructions.
2. Adjust the settings under *2D Scanning* to make sure that the 2D image of the calibration board is clear and neither overexposed nor underexposed.
3. Adjust the settings under *3D Scanning* to make sure that the point clouds of the circles on the calibration board are complete and have clear contours. For smoother point clouds and more accurate calibration result, we also recommend setting *Cloud Smoothing* and *Outliers Removal* under *Point Cloud Processing* to `Normal`.
4. If the on-site lighting conditions are not ideal and affects the quality of 2D images and point clouds, you can use shading or supplemental light to improve the lighting conditions.
5. Make sure the obtained point clouds of the circles on the calibration board are complete and have clear contours, as shown in *Figure 4*.

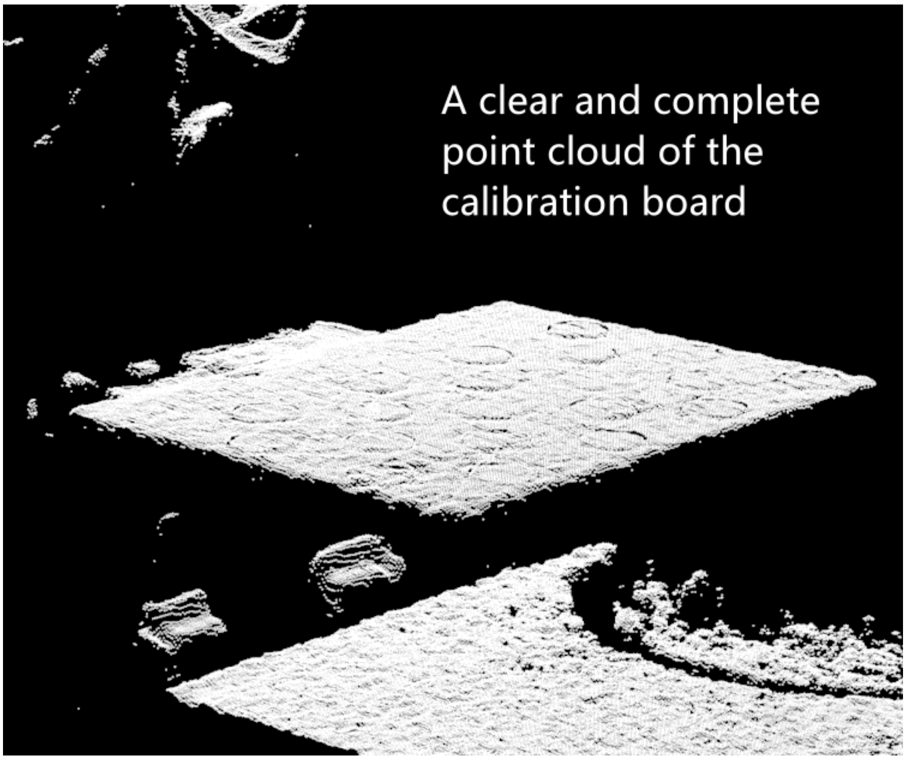
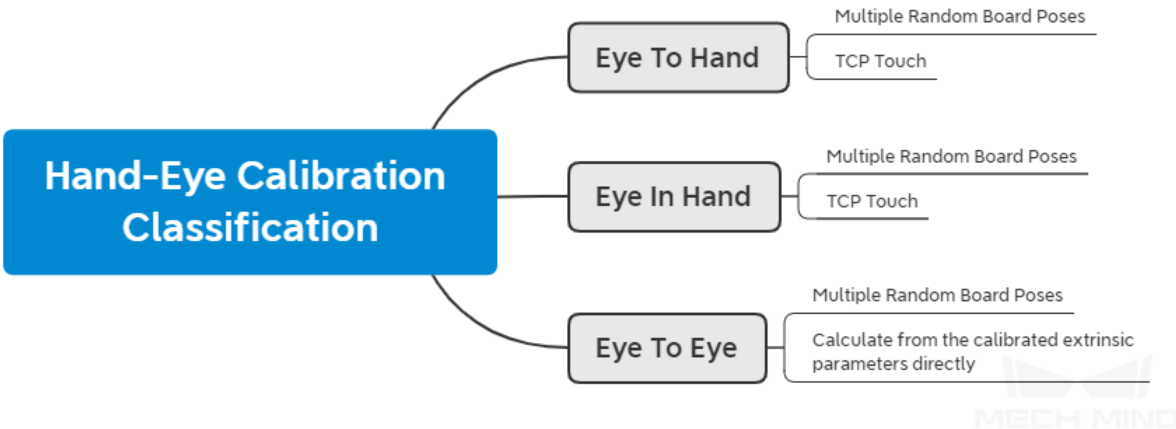


Figure 4. Point cloud of calibration board

6.2 Start Calibration - Standard Mode



The difference between multiple random board poses method and TCP touch method is as follows:

- The multiple random board poses method utilizes multiple poses either generated automatically by Mech-Vision or added manually. Images are captured in each pose and the circles on the calibration board are recognized, therefore establishing the spatial relationship among the calibration board, camera, and robot. It is easy to perform and has a high accuracy.

- The TCP touch method first determines the pose of the calibration board through the three-point method, and then establishes the spatial relationship among the calibration board, camera, and robot. This method is suited for situations where the robot is installed in a limited space and where the calibration board cannot be installed.

You can choose the calibration method that best suits the actual conditions on your site, and refer to the corresponding section below.

6.2.1 Eye To Hand

Calibration Mechanism

Basic Mechanism of ETH Calibration

Calibration Process

Add Calibration Points by Multiple Random Board Poses

Add Calibration Points by TCP Touch

Basic Mechanism of ETH Calibration

When using the multiple random board pose method, a calibration board of known dimensions is installed on the robot flange, and the coordinates of each mark point on the calibration board, denoted by A, in the robot base coordinate system, can be obtained. By taking photos of the calibration board, the coordinates of the camera's optical center relative to each mark point, denoted by B, can be obtained. The transformation relationship between the camera's optical center and the robot base coordinates, 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. 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 *Figure 1* below.

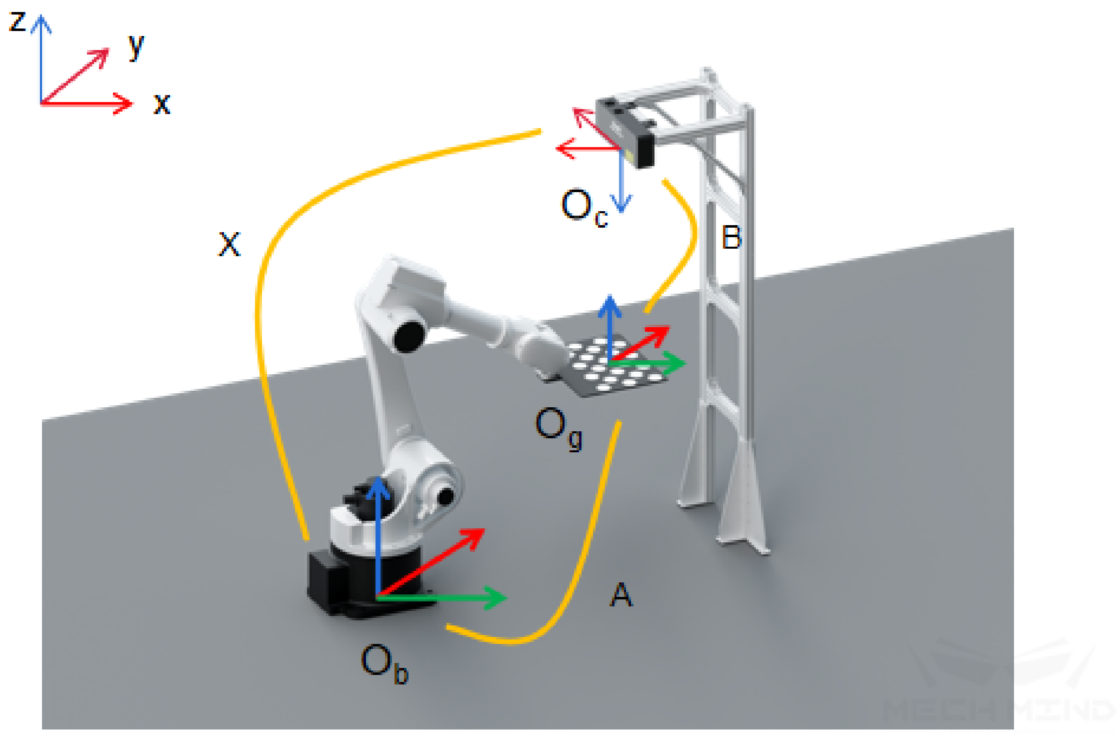


Figure 1. The multiple random board poses method (ETH)

When using the TCP touch method, place the calibration board on the object plane and install a sharp-tipped tool with known dimensions on the robot flange. Then, touch the mark points on the calibration board with the center point of the tool by moving the robot. As shown in *Figure 2*, 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 mark points on the calibration board using a sharp-tipped tool with known TCP coordinates.

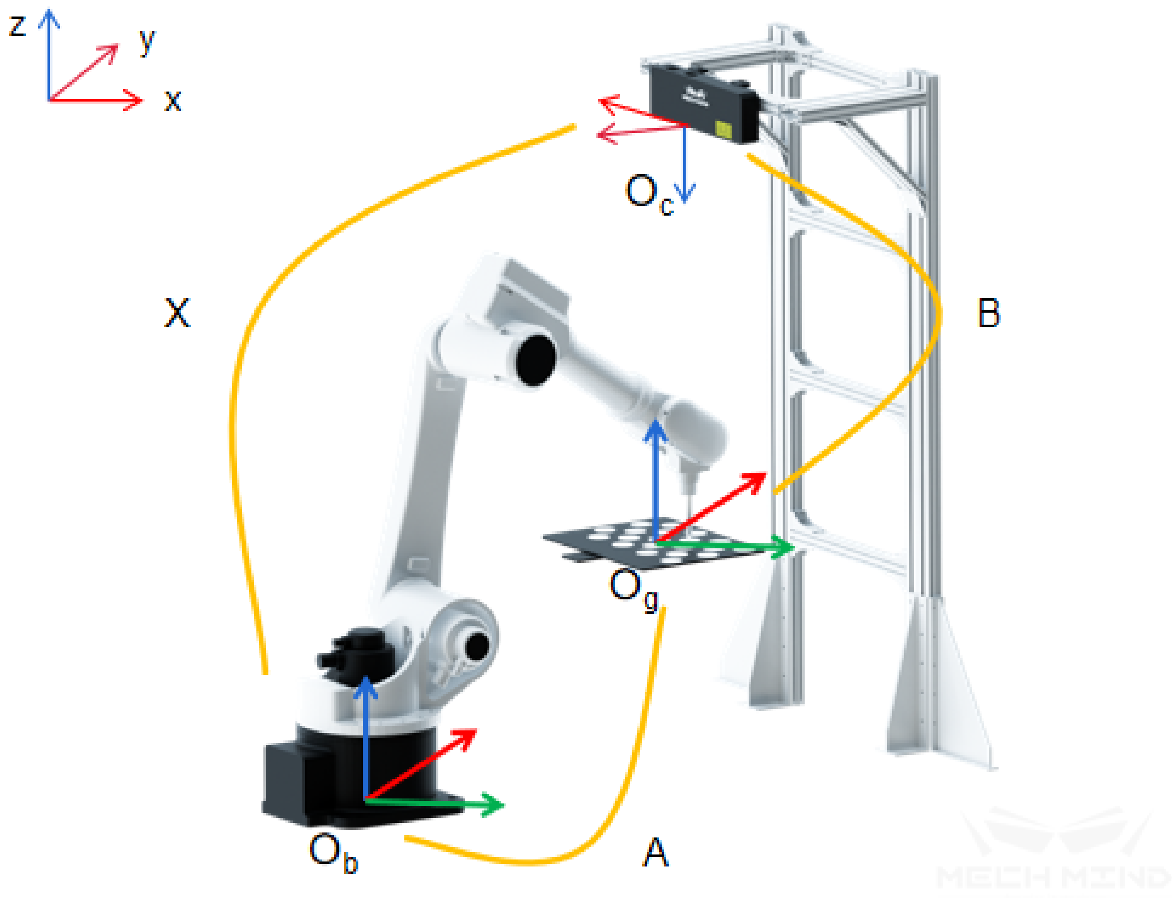


Figure 2. The TCP touch method (ETH)

Hint: Calibration under ETH calibrates the transformation relationship between the coordinate systems of the camera' s optical center and the robot base. If either the robot base or the camera moves, the extrinsic parameters would be altered, and the hand-eye relationship must be calibrated again.

Add Calibration Points by Multiple Random Board Poses

Using the trajectories generated by the software or the multiple poses added manually, capture images of calibration points at each waypoint on the trajectories in order to construct the correct relationship between the calibration board, camera, and robot.

Calibration Preset

- **Select a directory to save calibration results**

Launch the Mech-Vision software and select *Camera* → *Camera Calibration* → Select a directory to save calibration results (if a Mech-Vision project is opened, *Use current project directory* is chosen by default), as shown in *Figure 1*.

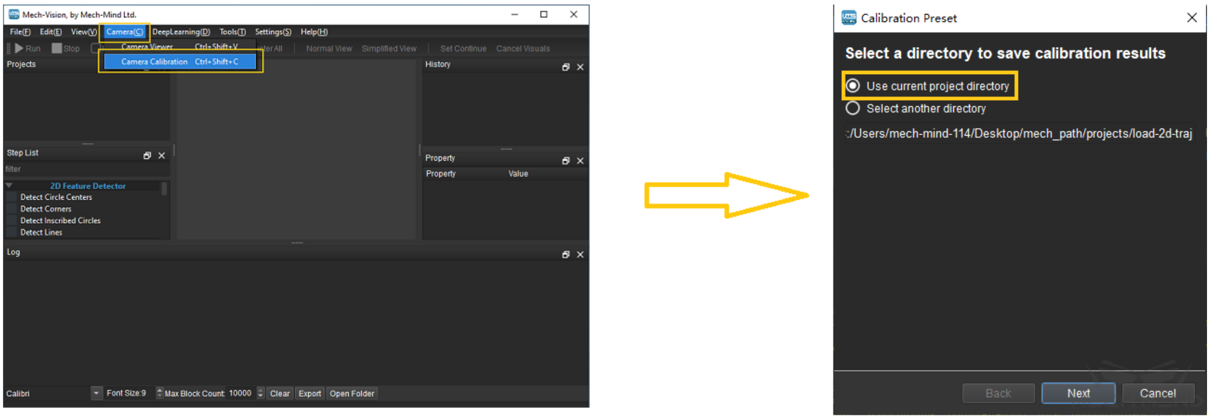


Figure 1 Select a Directory to Save Calibration Results

- **Select a calibration setup**

Select the corresponding calibration type (select *Eye to Hand* for this section), as shown in *Figure 2*.

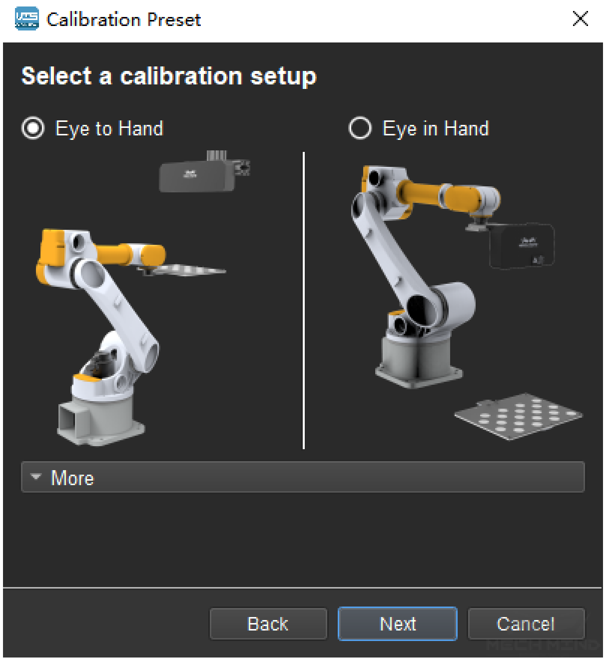


Figure 2 Select the Calibration Type

- **Check the robot connection status and select the calibration method**

If the robot is connected, the window in *Figure 3* will be shown. Select *Move and capture images automatically using robot* and click the *Finish* button to enter the calibration interface.

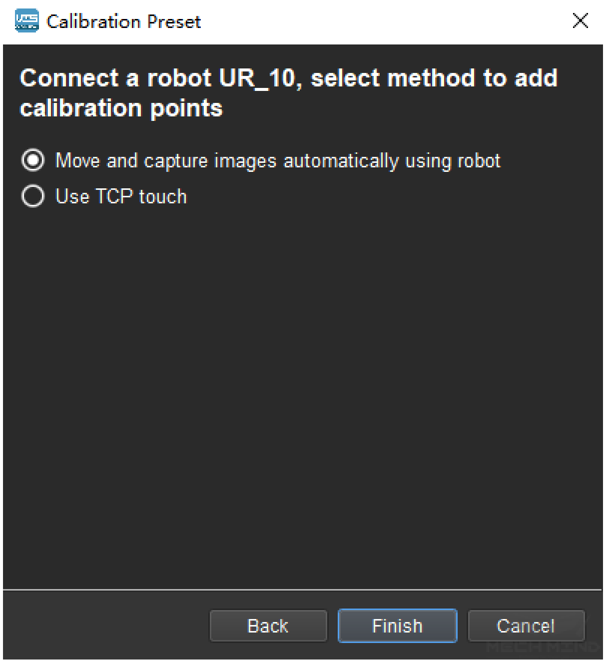


Figure 3 Select the Method to Add Calibration Points

If no robot is connected, select the correct robot type or go back to the previous step and retry after connecting a robot in Mech-Center, as shown in *Figure 4*.

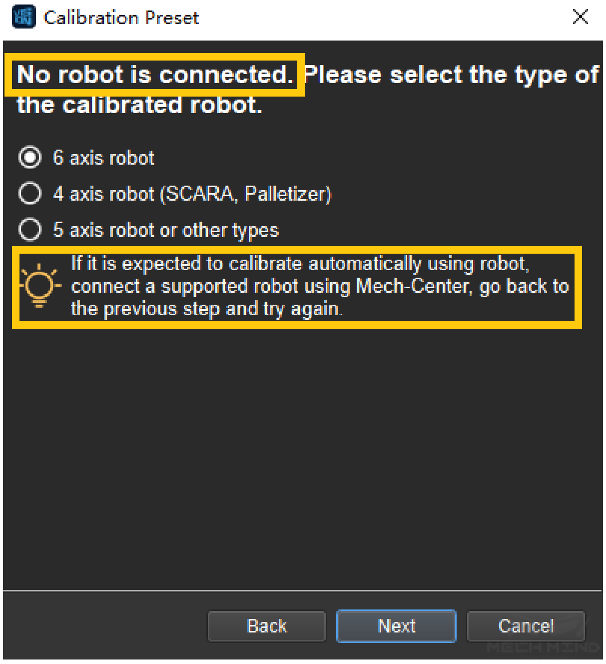


Figure 4 No Robot is Connected

Note: In the following calibration process, a connected 6 axis robot is used as an example. For the details on special types of robots, please refer to *Multiple Random Board Poses*.

Connect to Camera

- **Select the camera**

When the network is well connected and the camera works normally, the camera ID will automatically appear under the "Detected Cameras & Local Parameter Groups" list. After selecting the camera, click *Connect*, and the camera will connect to the Mech-Vision software, as shown in *Figure 5* below.

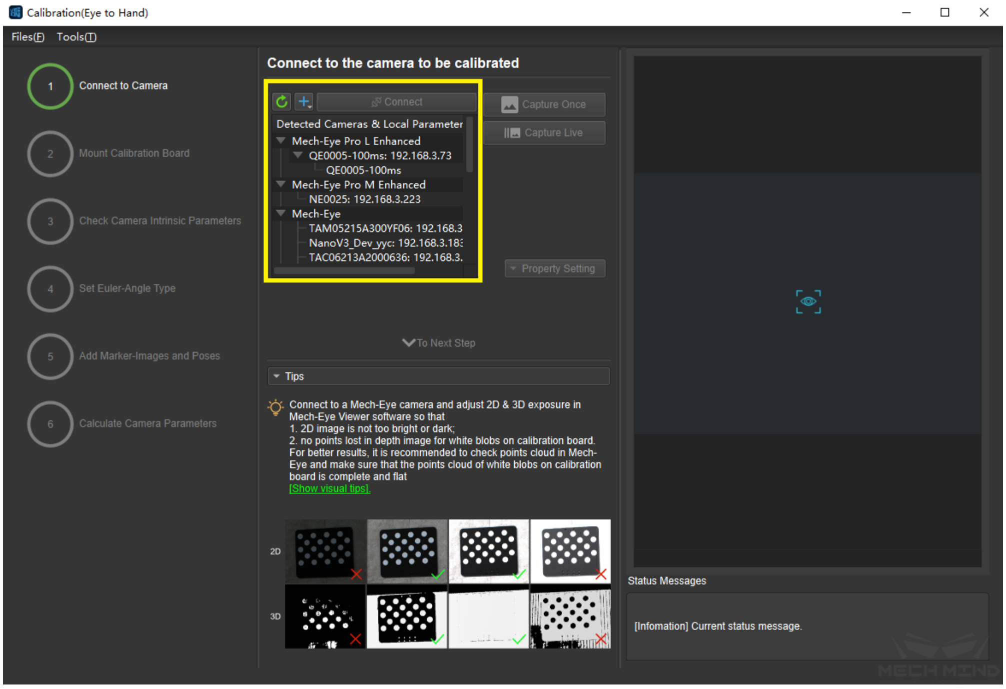


Figure 5 Connect to the Real Camera

- **Capture images**

After connecting to the camera, select *Capture Live* or *Capture Once* to view the captured images on the right side of the interface, as shown in *Figure 6* below (it is suggested to stop capturing images once the images in the view meet the requirements. Otherwise, multiple refreshes may affect the next calibration step).

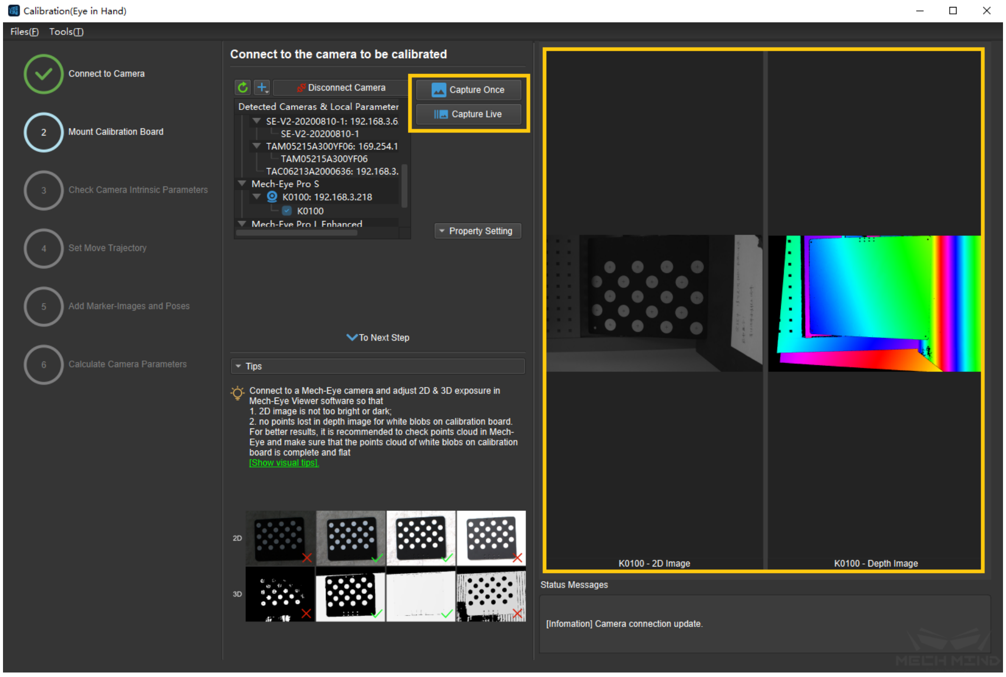


Figure 6 Capture Images

- **Manually add camera**

If the camera is functioning normally but cannot be found under the list of detected cameras, please manually add the camera IP. Click the + icon, and a dialog box will pop up, as shown in *Figure 7* below. Fill in the corresponding camera name and camera IP to add the camera to the detected cameras list. Then, connect the camera following the steps mentioned above.

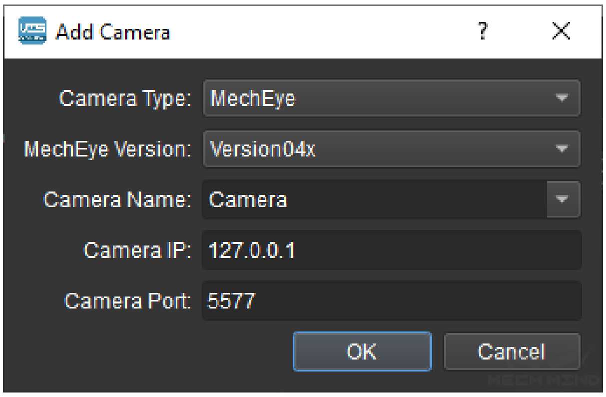


Figure 7 Manually Add Camera

Mount Calibration Board

Select the type of calibration board and put it in the camera' s field of view. The operation interface is shown in *Figure 8* below.

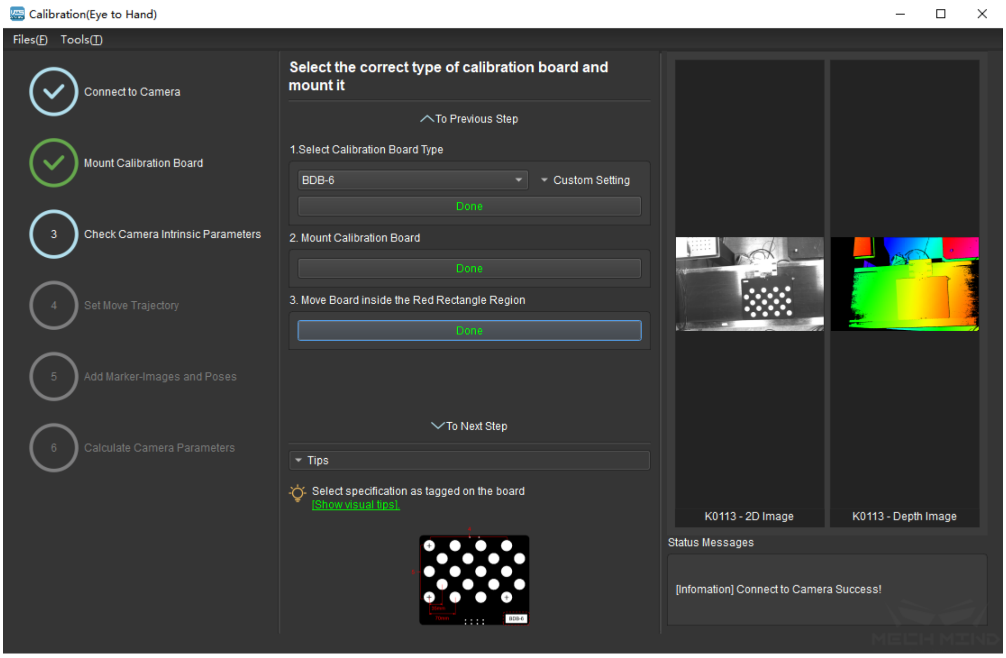


Figure 8 Select and Install the Calibration Board

Specific steps are as follows:

1. Select the corresponding calibration board model (the model nameplate is affixed on the top of the calibration board), and click the *Confirm* button.
2. Confirm that the calibration board is installed on the robot and within the camera' s field of view. Click the *Confirm* button when finished.
3. Make sure the calibration board is within the red rectangle in the 2D image, and click the *Confirm* button.

Check Camera Intrinsic Parameters

- **Check Camera Intrinsic Parameters**

After mounting the calibration board, check if the camera parameters are correct in order to ensure that feature points can be detected during the calibration process. Click the *Check Camera Intrinsic Parameters* button, and the result of the intrinsic parameter checking will be displayed in a pop-up window. *Figure 9* shows a successful intrinsic parameter check.

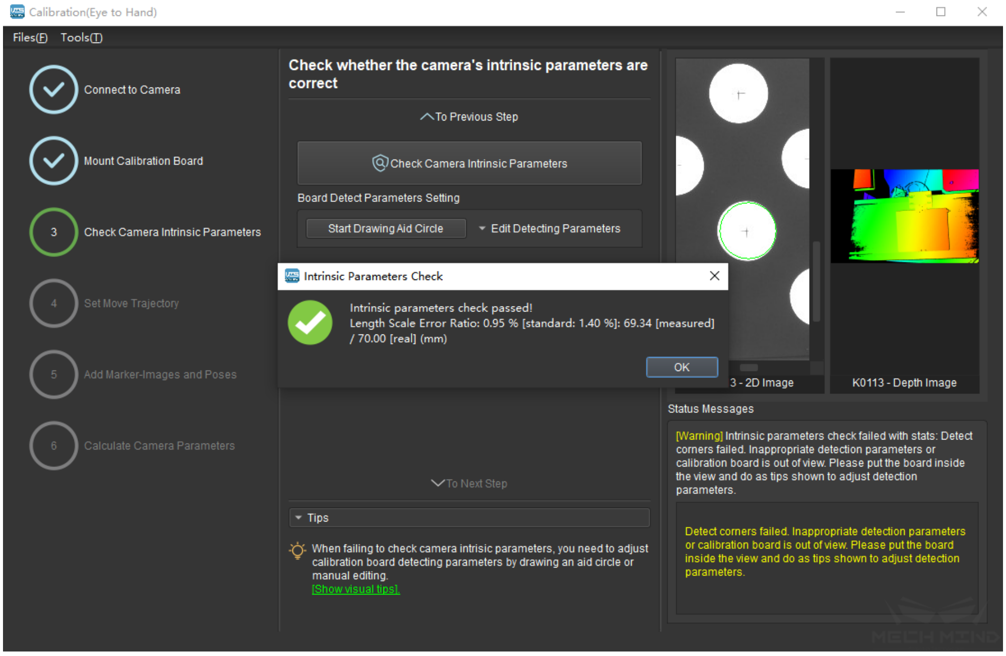


Figure 9 Check Camera Intrinsic Parameters

Generally, when the lighting conditions are good and the camera settings are appropriate, the intrinsic parameter check will pass by simply keeping the default values. *Figure 10* shows a successful detection of the feature points.

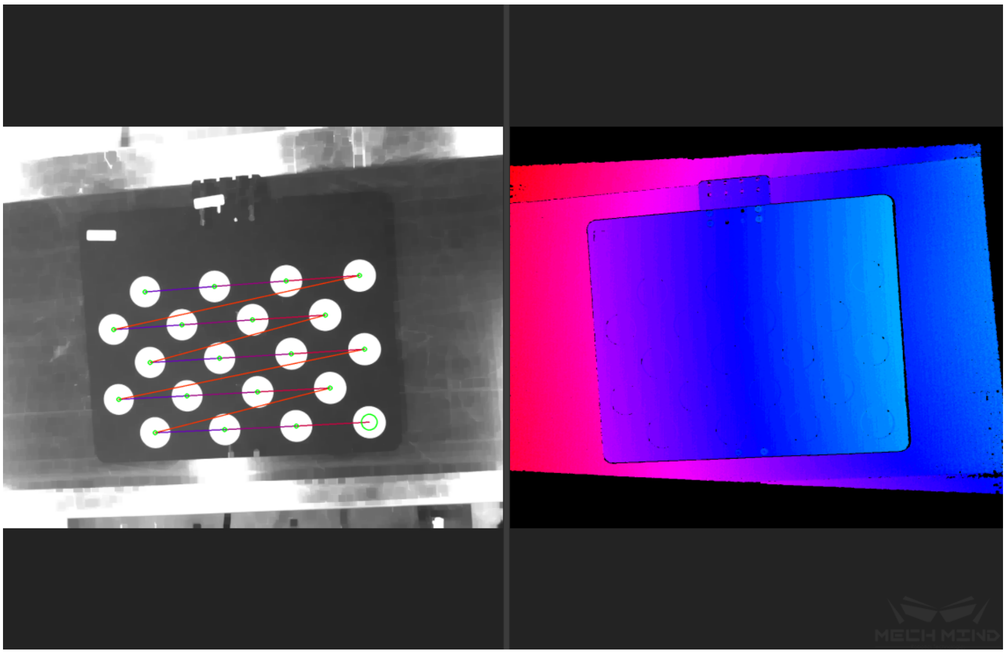


Figure 10 Detected Feature Points

- **Draw Aid Circle**

If the intrinsic parameter check fails, draw an aid circle or manually adjust the calibration board detection parameters. Select the *Start Drawing Aid Circle* button and draw a circle coinciding with a circle in the calibration board. After which, the detecting parameters of the circles should change.

If the user chooses to manually adjust the detecting parameters, simply click *Edit Detecting Parameters* and change the values accordingly.

After completing the above steps, click the *Check Camera Intrinsic Parameters* button again to get a new result. The steps are shown in *Figure 11* below.

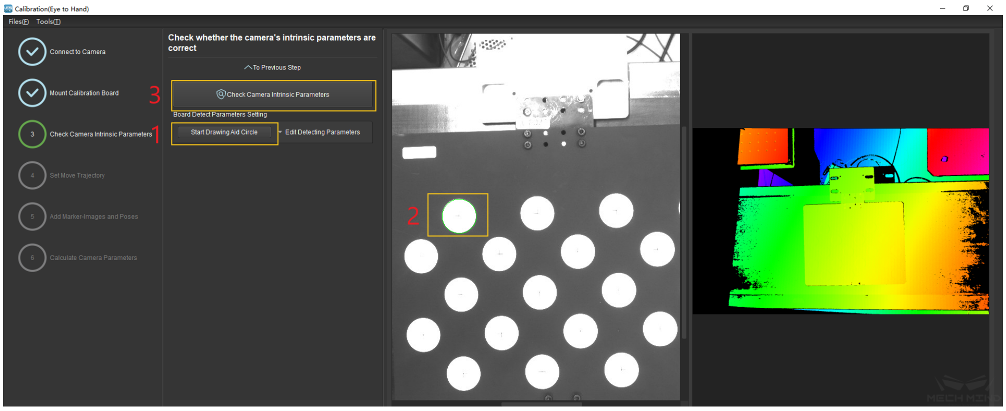


Figure 11 Draw the Aid Circle

Tip: If the calibration circles are too small to be selected easily, the user could right click the 2D image, uncheck *Fit to Window*, and select *Normal Size*. Start drawing aid circles after adjusting the size of the image.

If the feature points are still not detected, please adjust the camera settings according to the on-site operating conditions. Detailed information can be found in Mech-Eye Viewer.

Set Move Trajectory

- **Set Calibration Height Range**

The robot trajectory will be automatically generated. In order to ensure that the robot can successfully complete the movement, it is necessary to set a proper height limit according to the size of the robot' s working space. The details are shown in *Figure 12*.

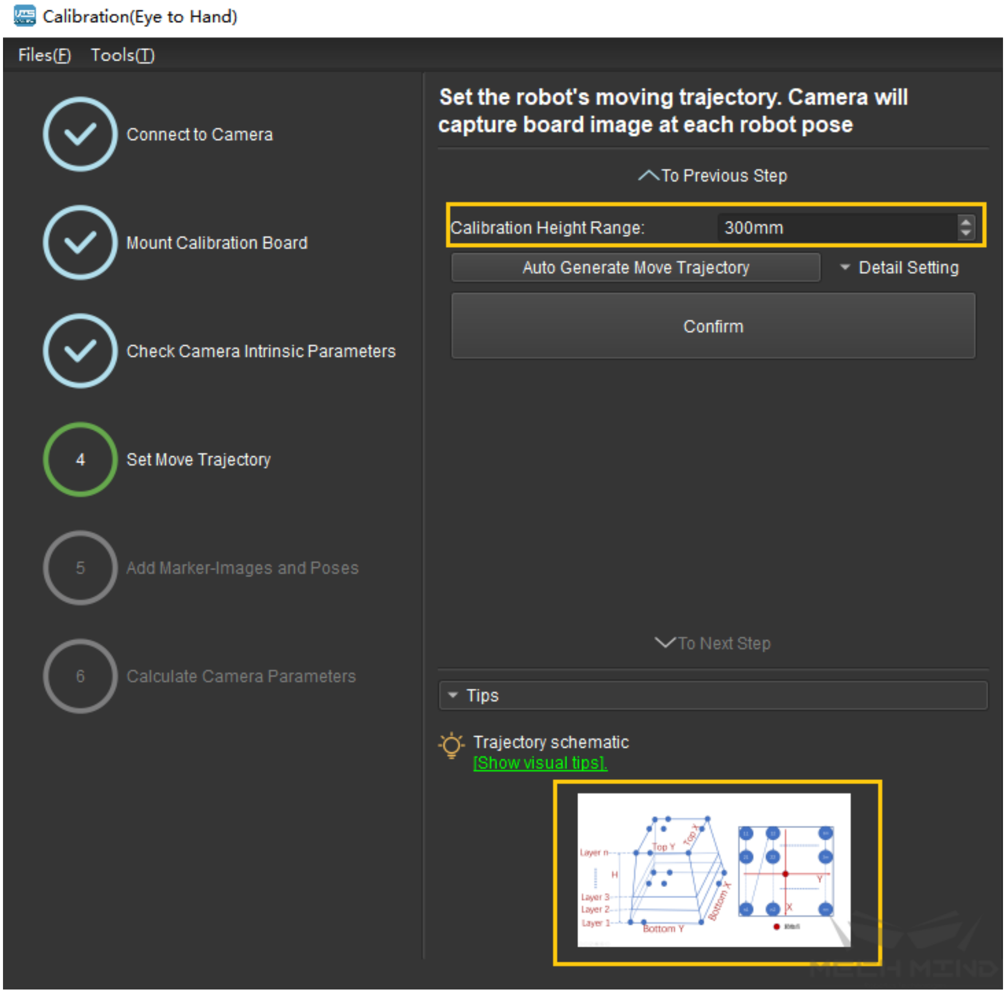


Figure 12 Set the Calibration Height Range

- **Automatically generate move trajectory**

After setting the trajectory range and ensuring that Mech-Viz is connected properly, click the *Detail Setting* button to set detailed parameters such as height, layer numbers, etc. Then, click the *Auto Generate Move Trajectory* button, and the program will automatically generate each waypoint of the trajectory. A dialog box will pop up after completion. The steps mentioned above are shown in *Figure 13* and *Figure 14* below.

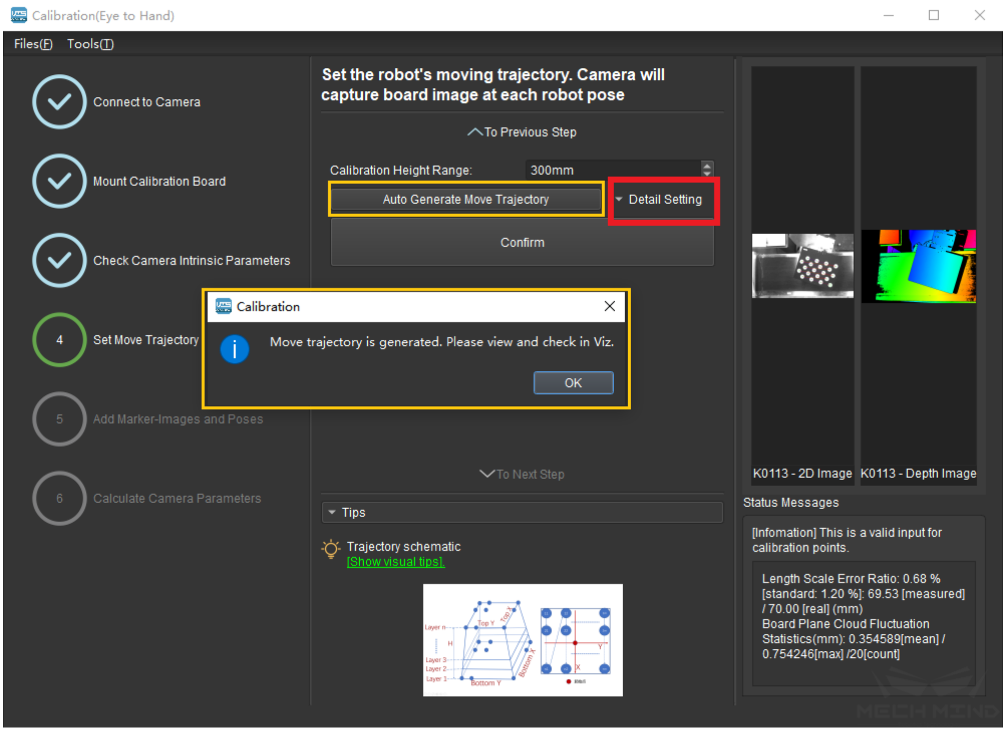


Figure 13 Automatically Generated Trajectory

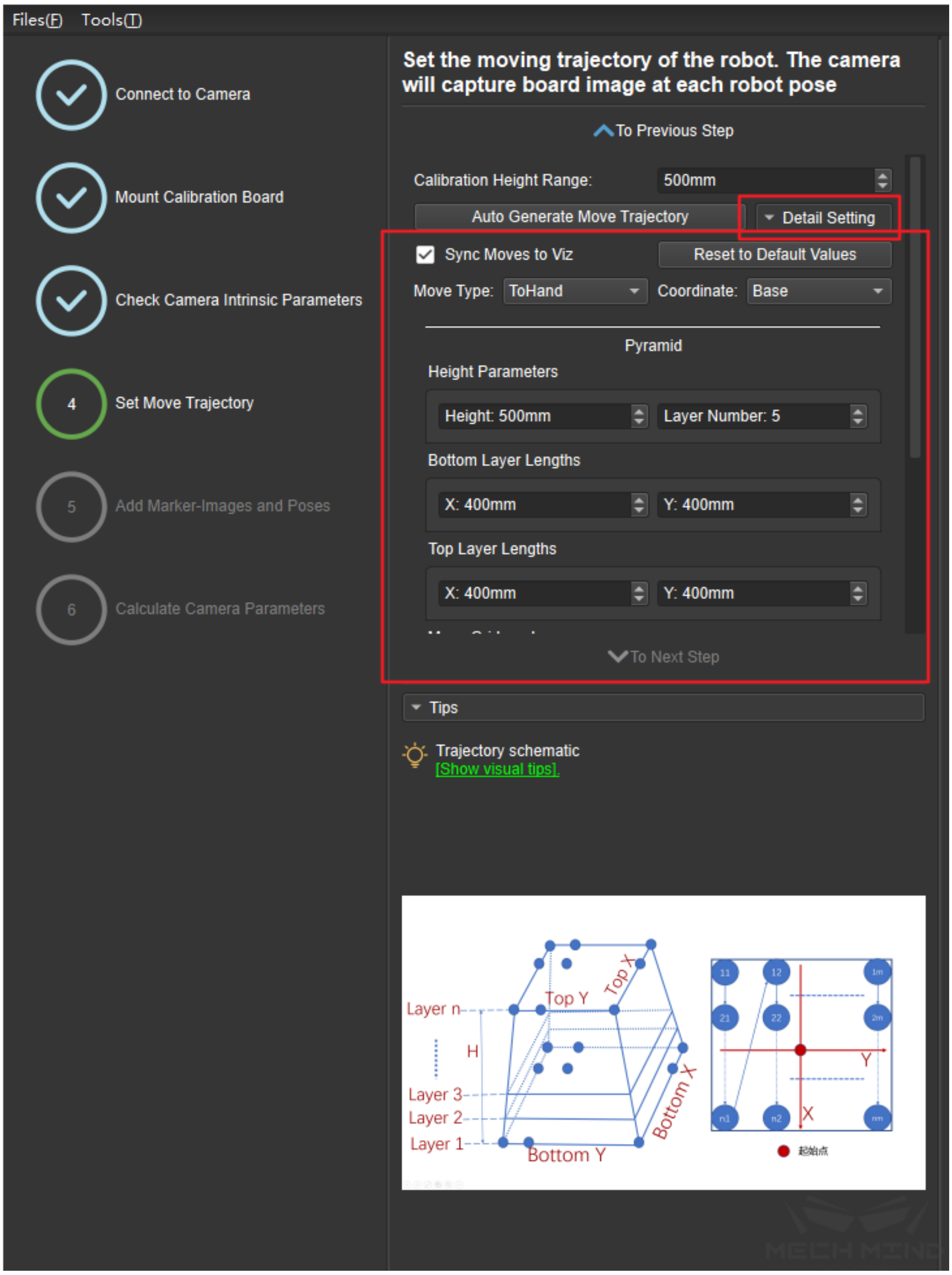


Figure 14 Detail Setting

- **IMPORTANT!!! Check trajectory in Mech-Viz**

In Mech-Viz, click the *robot* tab. Select the current robot type and click the *Sync Robot* button. The distribution of the waypoints in the auto-generated trajectory is displayed, as shown in *Figure 15*. The user must check that the trajectory is reasonable and will not collide with obstacles in the environment!!!

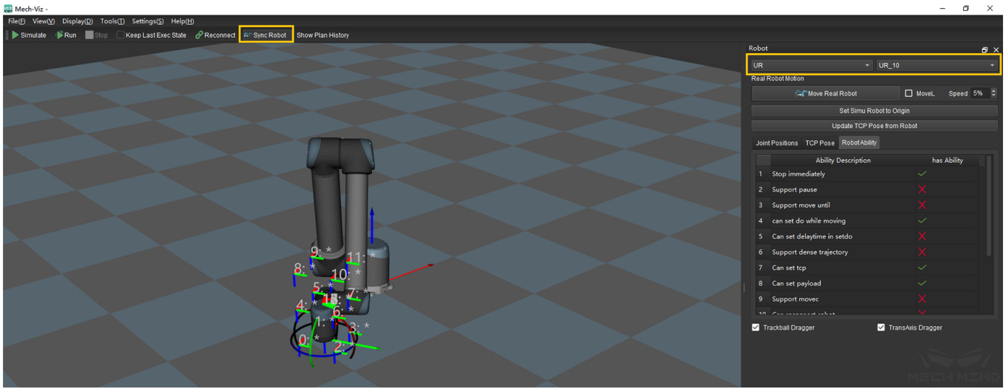


Figure 15 Check the Automatically Generated Trajectory in Mech-Viz

After ensuring that the trajectory is safe, return to Mech-Vision and click the *Done* and *To Next Step* buttons, as shown in *Figure 16* below.

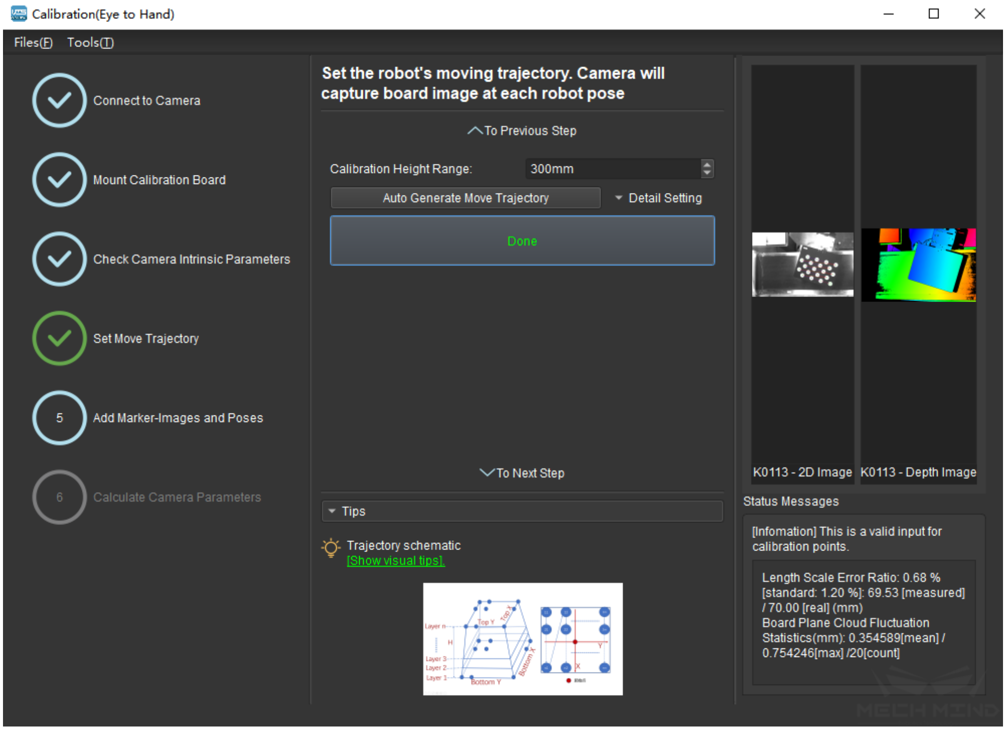


Figure 16 Confirm the Automatically Generated Trajectory

Add Marker-Images and Poses

- **Preset for calibration**

Enter the Add Marker-Images and Poses interface. As shown in *Figure 17*, check the *Save Images* option, choose the corresponding robot name, and click the *Move Robot along Trajectory and Add Board Images* button.

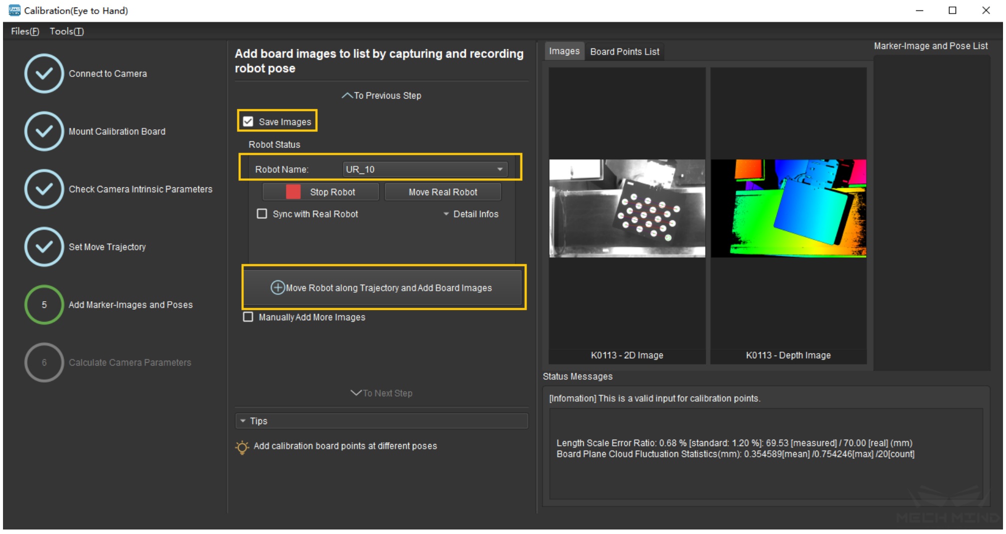


Figure 17 Preset for Calibration

A safety tips will pop up before starting the calibration, as shown in *Figure 18*. Make sure that the operation environment is safe and then click the *OK* button.

Figure 18 Safety Tips before Starting Calibration

- **Calibration process**

Figure 19 show the interface when the calibration process is running. Please stay away from the robot working area to keep safe. The progress bar will load from 0% to 100% and the captured images can be seen in the right viewer at the same time.

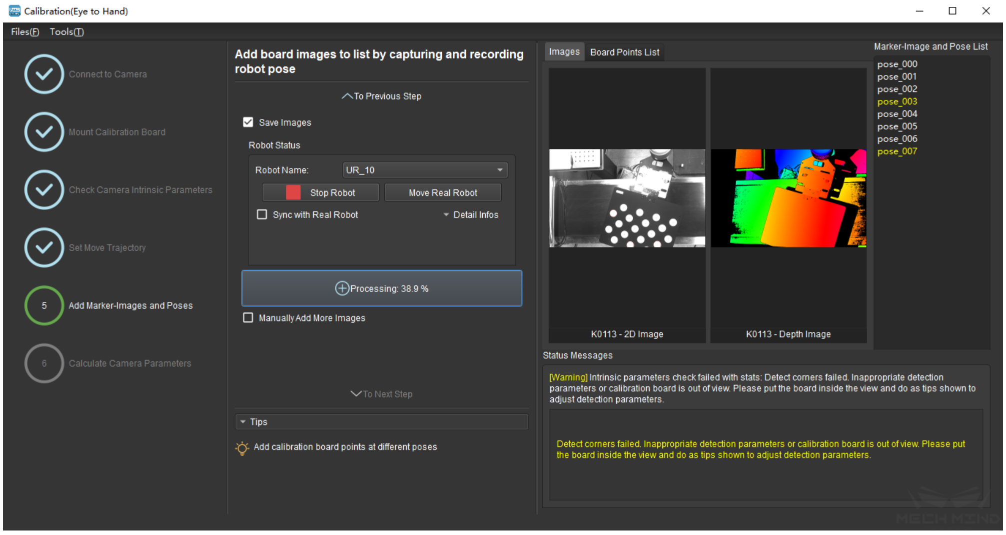


Figure 19 Calibration Process is Running

Attention: 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 press the emergency stop button on the robot teach pendant to stop the robot immediately.

- **Detect Deviation of measurements**

During the calibration process, a prompt box may appear, indicating that a measurement deviation has been detected, as shown in *Figure 20*. It is recommended to recalibrate the camera intrinsic parameters and check whether the robot loses its absolute accuracy. The users could choose to continue this calibration process by clicking the *Ignore* button. However, the calibration accuracy cannot be guaranteed.

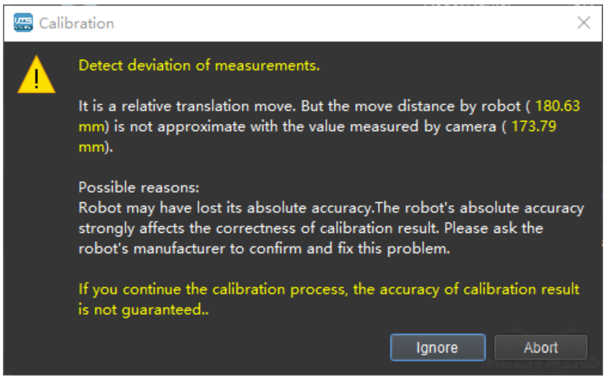


Figure 20 Detect a Measurement Deviation

- **Image acquisition completed**

The interface after the the completion of the image acquisition is shown in *Figure 21* below. It is necessary to determine whether the currently recognized calibration point meets the requirements from Tips. If so, click the *To Next Step* button. If not, please manually add the calibration points by moving the robot through the teach pendant or Mech-Viz software. Check the *Manually Add More Images* button, and click the *Add Single Calibration Board Image And Record Pose* button to collect images again.

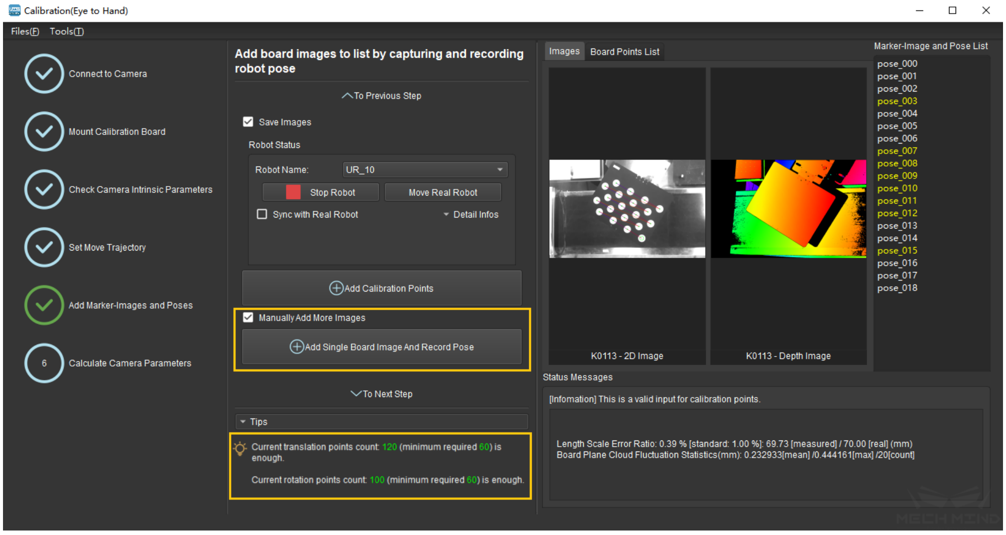


Figure 21 Image Acquisition Completed

Calculate Camera Parameters

- **Calculate Camera Parameters**

In the Calculate Camera Parameters interface, click the *Calculate Camera Parameters Results* button. Confirm and save the calibration data accordingly. The calibration result and the point cloud showing the calibration error will be displayed on the right part of the window, as shown in *Figure 22*.

Figure 22 Camera Calibration Results

- **Display PointCloud in Mech-Viz software**

As shown in *Figure 23* below, click the *Display PointCloud in Mech-Viz* button.

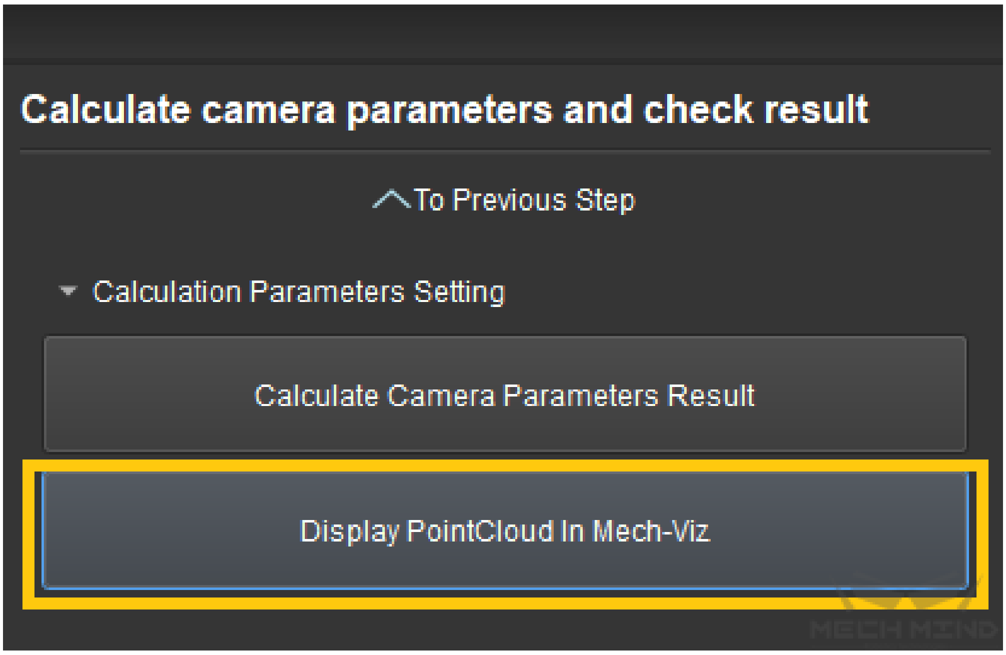


Figure 23 Display Point Cloud in Mech-Viz

Next, in Mech-Viz, select the *robot* tab. Click the *Configure Robot* button, and in the pop-up that appears, choose the correct robot type. After which, click the *Sync Robot* button. The above steps are shown in *Figure 24* below.

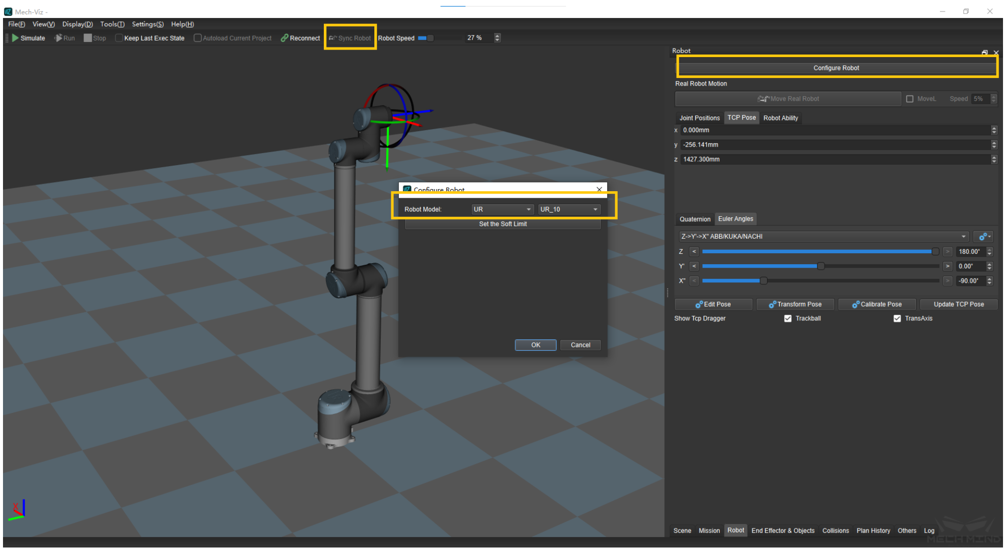


Figure 24 Sync Robot in Mech-Viz

Select the *Scene* tab, and adjust the ground height accordingly, as shown in *Figure 25*.

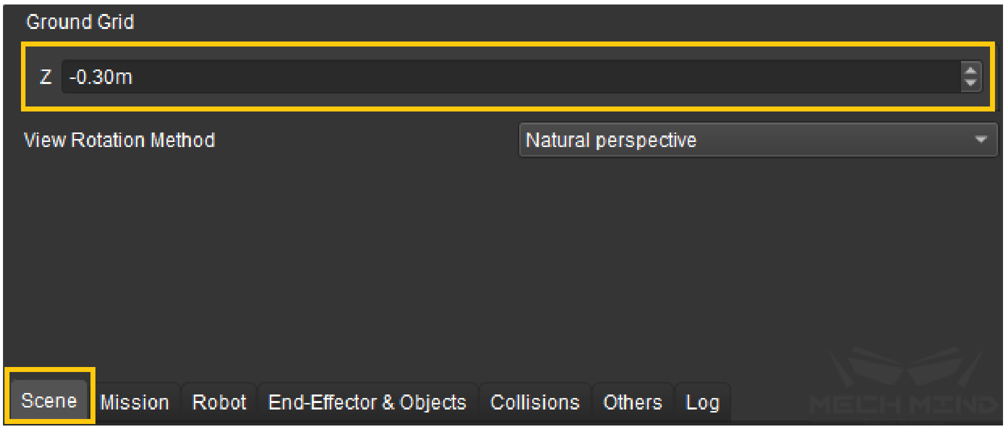


Figure 25 Adjust the Ground Height in Mech-Viz

Finally, the point cloud of the calibration board will be displayed as shown in *Figure 26*.

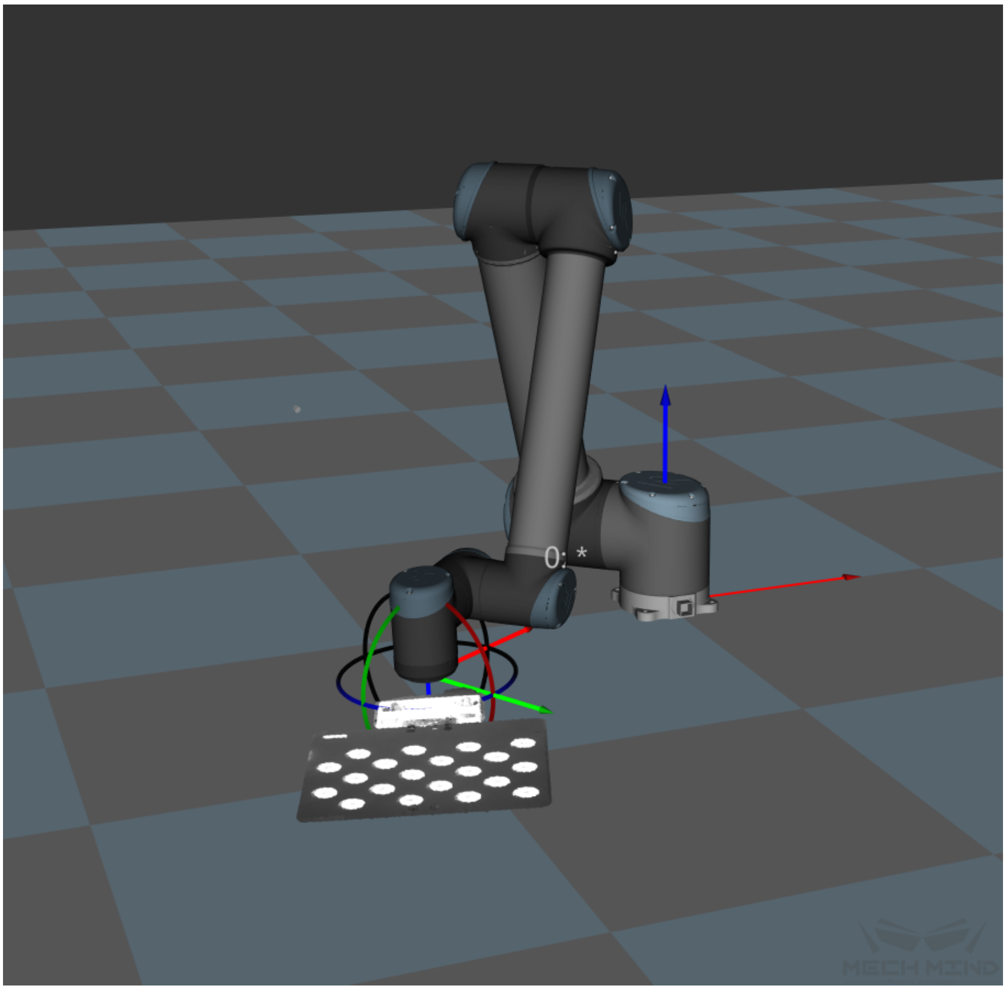


Figure 26 Point Cloud of Calibration Board in Mech-Viz

Add Calibration Points by TCP Touch

In special cases where the robot cannot be controlled by Mech-Viz or the calibration board cannot be installed, TCP touch is recommended for robot hand-eye calibration. Compared to other methods, TCP touch only needs one set of calibration points to accomplish the task.

Calibration Preset

- **Select a directory to save calibration results**

Launch Mech-Vision and click *Camera* → *Camera Calibration* → Select a directory to save calibration results (if a Mech-Vision project is opened, it will use current project directory by default), as shown in *Figure 1*.

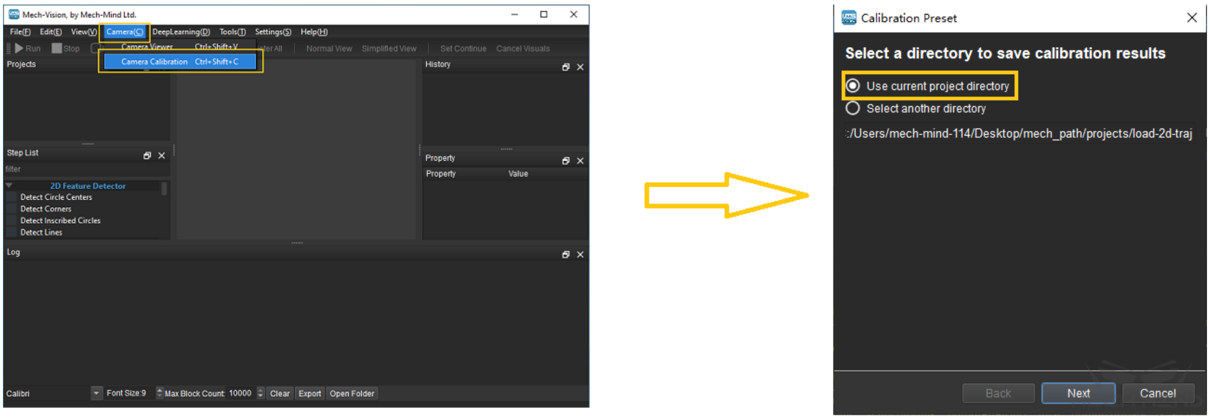


Figure 1 Select a Directory to Save Calibration Results

- **Select a calibration setup**

Select the corresponding calibration type, as shown in *Figure 2*. *Eye to Hand* (ETH) will be used in the following section.

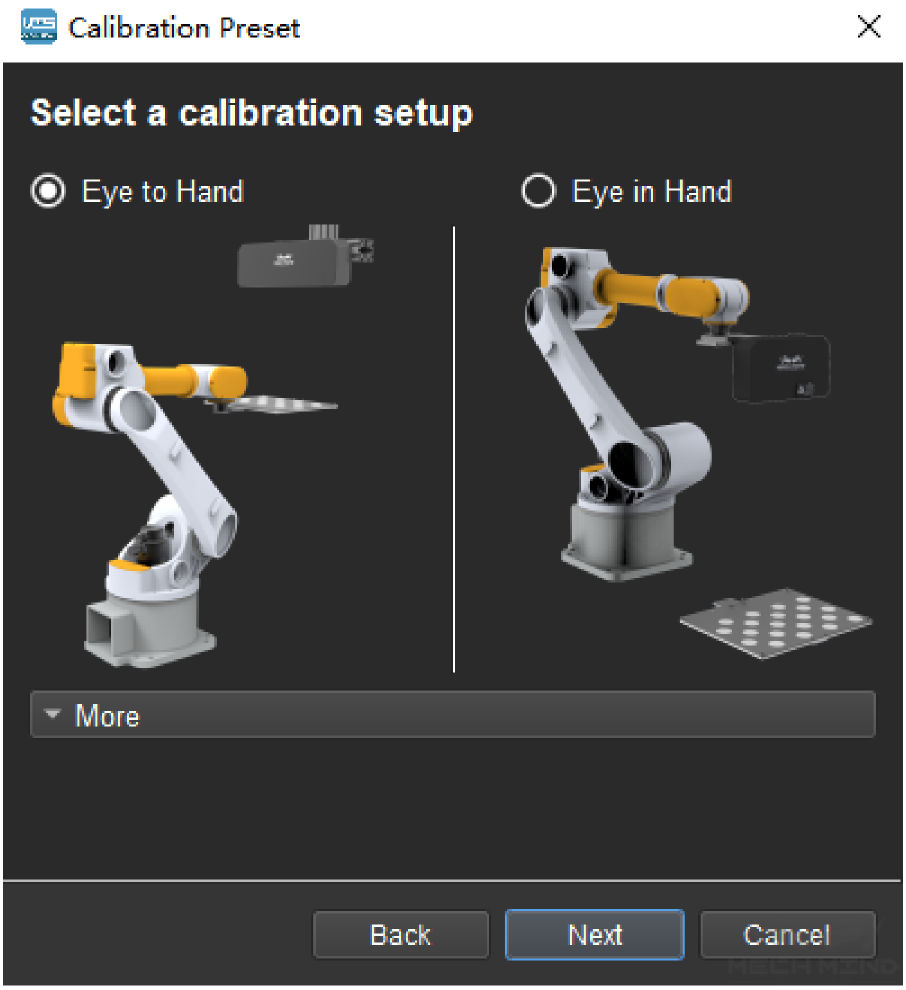


Figure 2 Select ETH Method

- **Check robot connection status and select a calibration method**

If the robot is connected, the window in *Figure 3* will be shown. Select *Use TCP touch* and click the *Finish* button to enter the calibration interface.

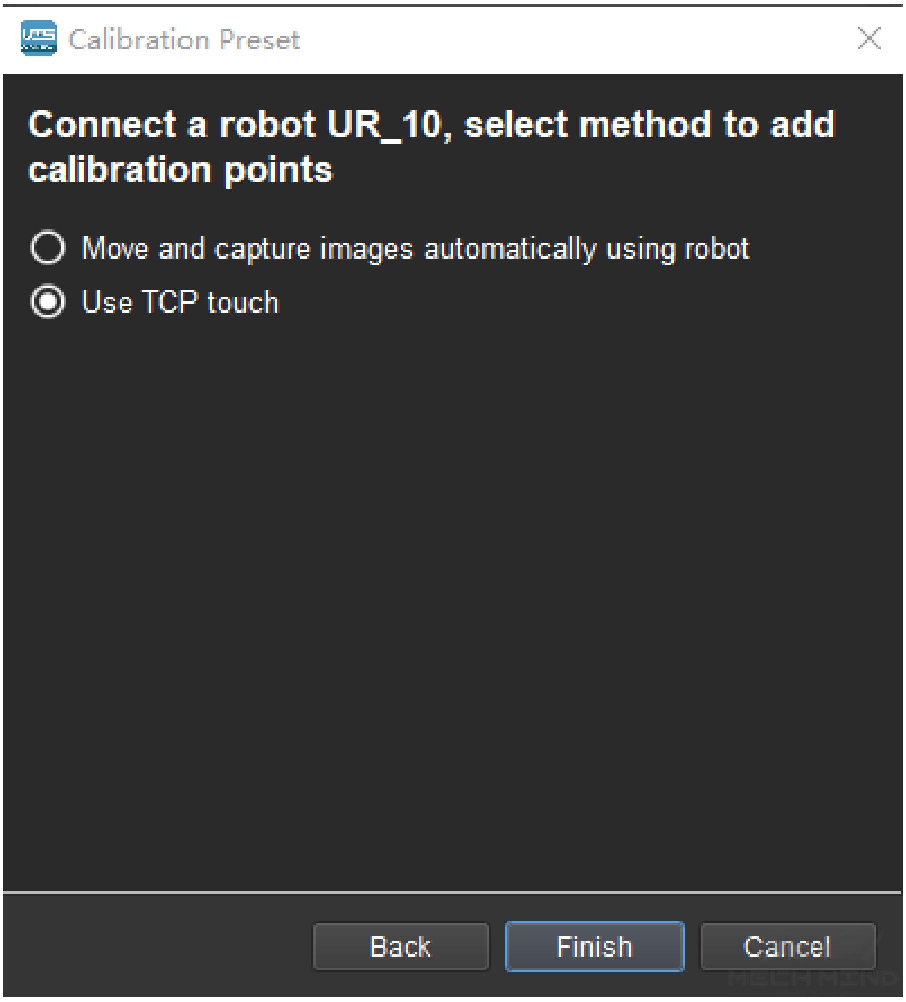


Figure 3 Select a Method to Add Calibration Points

If no robot is connected, select the correct robot type or go back to the previous step and retry after connecting a robot in Mech-Center, as shown in *Figure 4*.

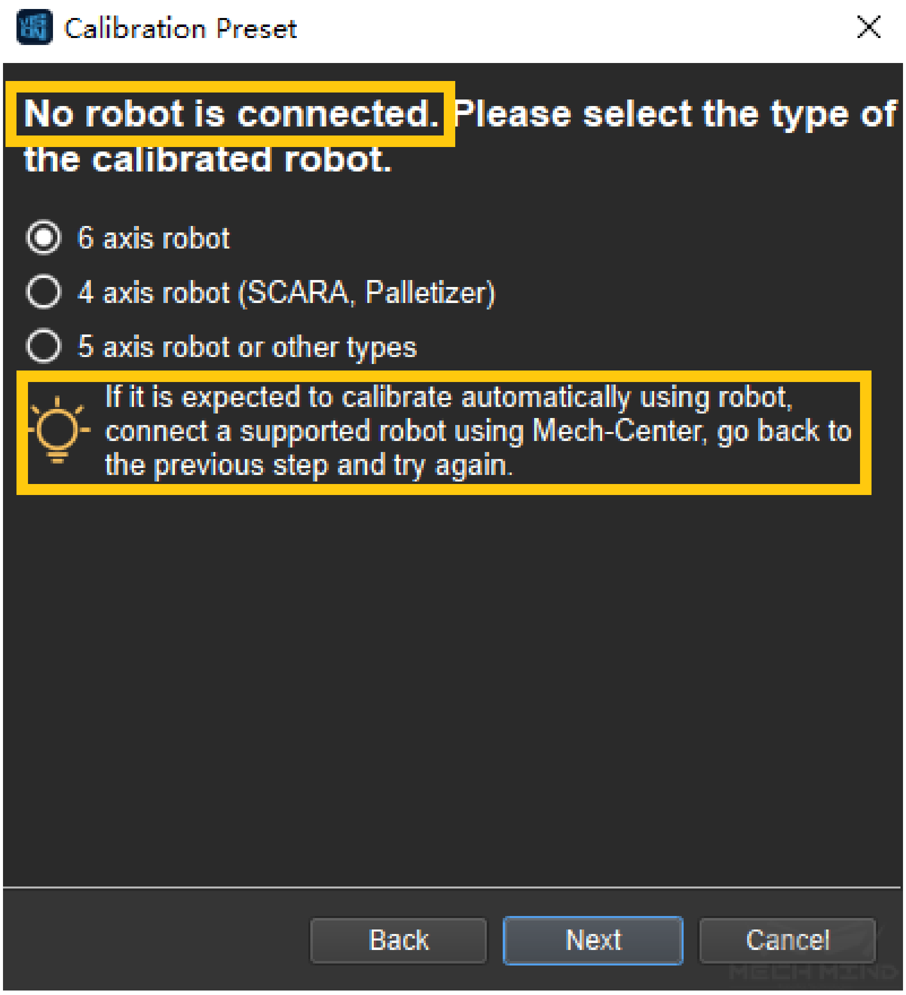


Figure 4 No Robot is Connected

Note: In the following calibration process, a connected 6 axis robot is used as an example. For the details on special types of robots, please refer to *TCP Touch*.

Connect to Camera

- **Select the camera**

When the network is well connected and the camera works normally, the camera ID will automatically appear under the "Detected Cameras & Local Parameter Groups" list. After selecting the camera, click *Connect*, and the camera will connect to the Mech-Vision software, as shown in *Figure 5* below.

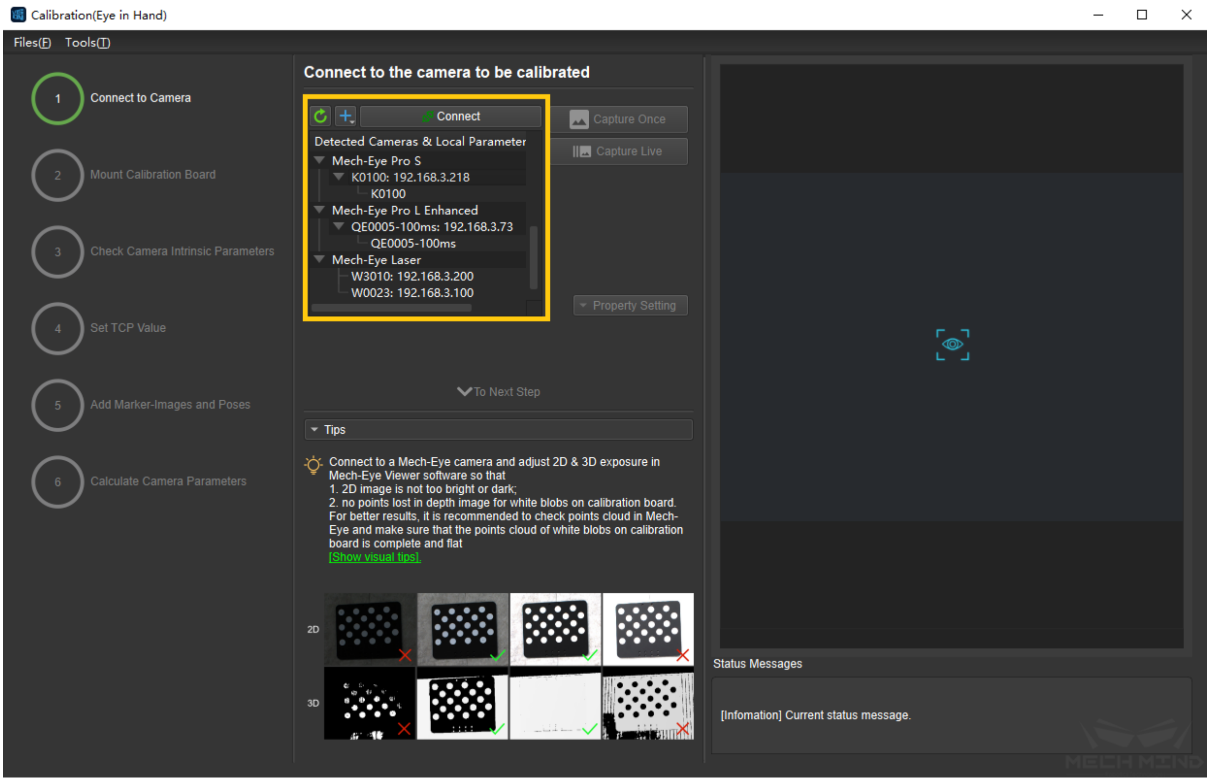


Figure 5 Connect the Camera

- **Capture images**

After connecting to the camera, select *Capture Live* or *Capture Once* to view the captured images on the right side of the interface, as shown in *Figure 6* below (it is suggested to stop capturing images once the images in the view meet the requirements. Otherwise, multiple refreshes may affect the next calibration step).

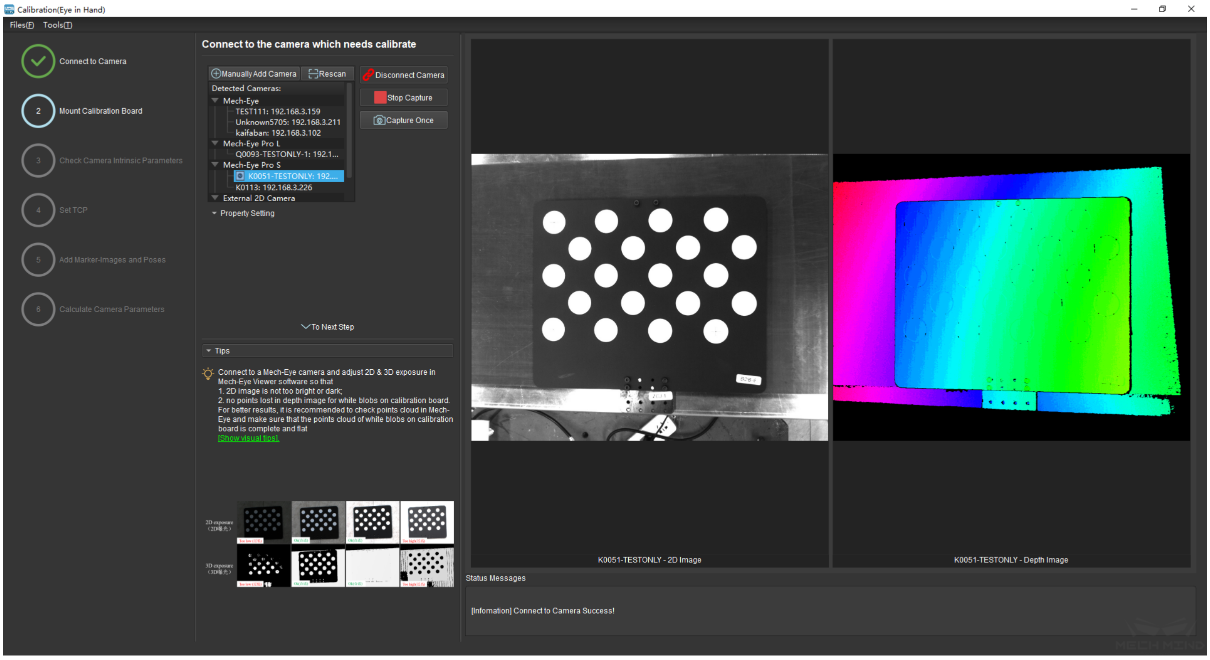


Figure 6 Capture images

- **Manually Add Camera**

If the camera is functioning normally but cannot be found under the list of detected cameras, please manually add the camera IP. Click the + icon, and a dialog box will pop up, as shown in *Figure 7* below. Fill in the corresponding camera name and camera IP to add the camera to the detected cameras list. Then, connect the camera following the steps mentioned above.

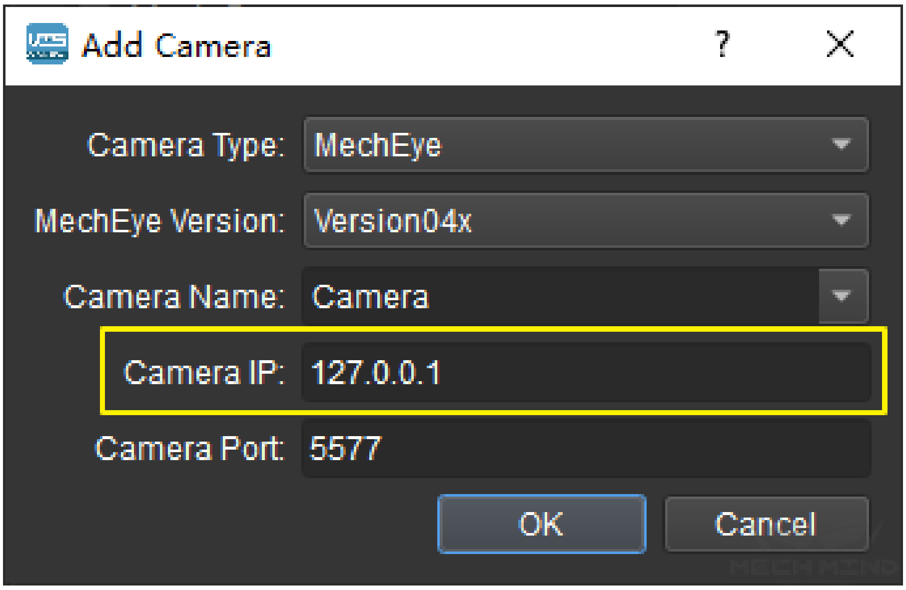


Figure 7 Manually Add Camera

Mount Calibration Board

Select the correct type of calibration board and put it in the camera' s field of view. The operation interface is shown in *Figure 8* below.

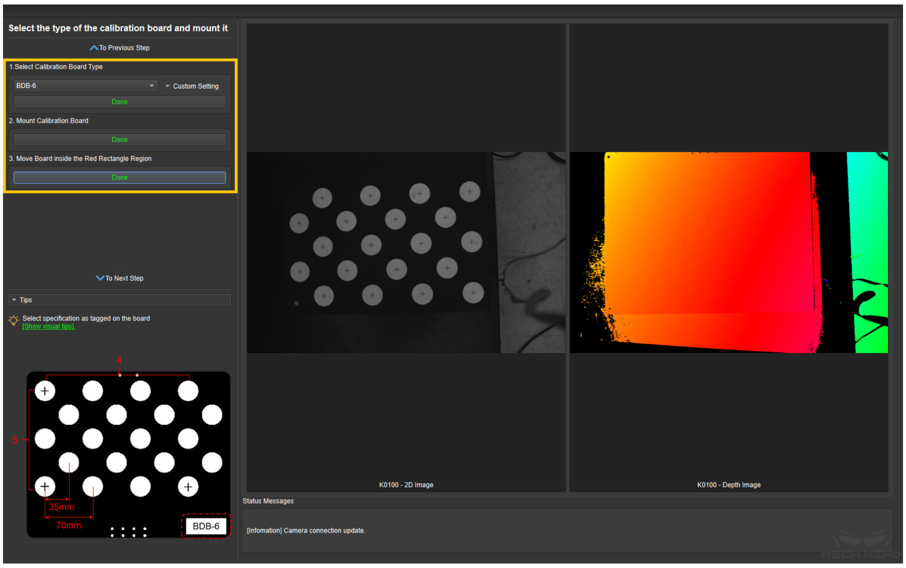


Figure 8 Mount Calibration Board

Specific steps are as follows:

1. Select the corresponding calibration board model (the name of the model is affixed on the top of the calibration board), and click the *Confirm* button.
2. Confirm that the camera is installed on the robot and the calibration board is within the camera' s field of view. Click the *Confirm* button when finished.
3. Make sure the calibration board is within the red rectangle in the 2D image, and click the *Confirm* button when finished.

Check Camera Intrinsic Parameters

- **Check Camera Intrinsic Parameters**

After mounting the calibration board, check if the camera parameters are correct in order to ensure that feature points can be detected during the calibration process. Click the *Check Camera Intrinsic Parameters* button, and the result of the intrinsic parameter checking will be displayed in a pop-up window. *Figure 9* shows a successful intrinsic parameter check.

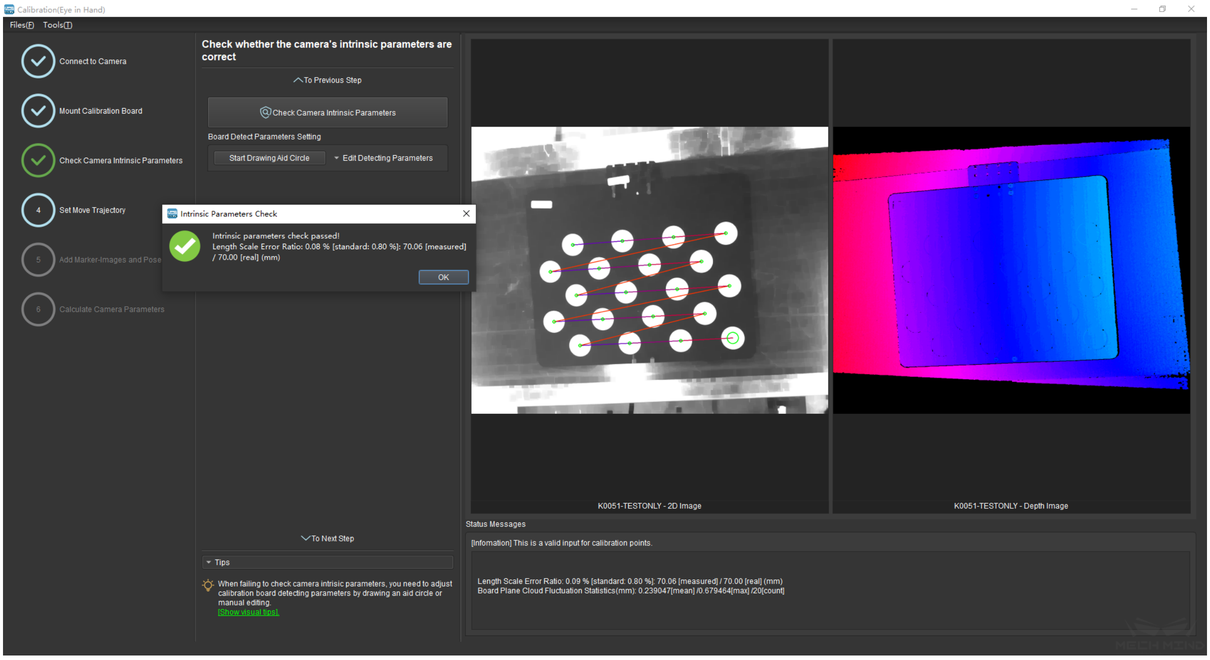


Figure 9 Check Camera Intrinsic Parameters

Generally, when the lighting conditions are good and the camera settings are appropriate, the intrinsic parameter check will pass by simply keeping the default values. *Figure 10* shows a successful detection of the feature points.

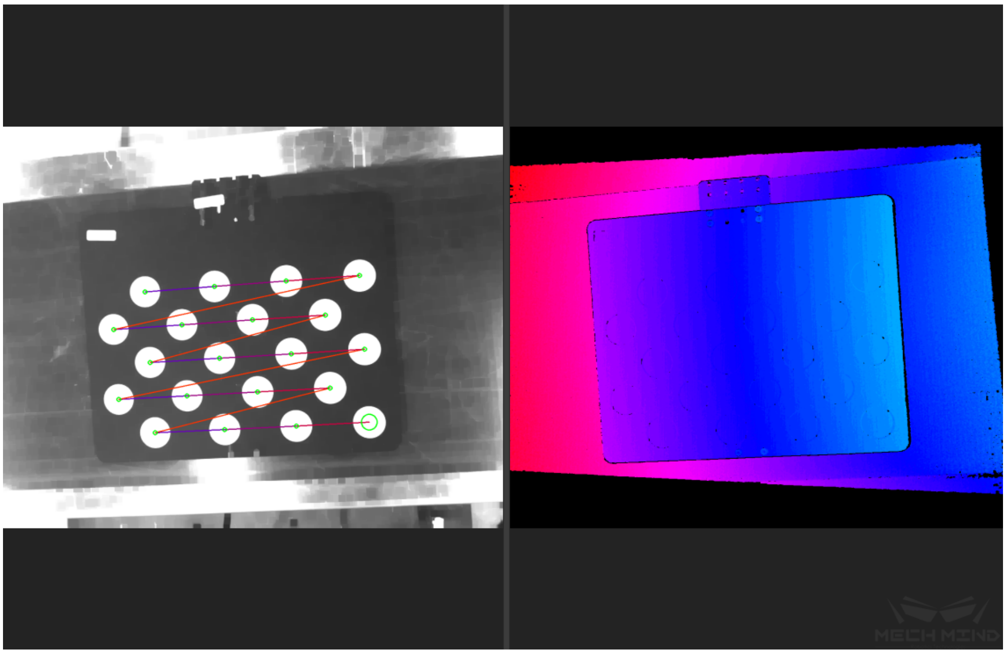


Figure 10 Detected Feature Points

- **Draw Aid Circle**

If the intrinsic parameter check fails, draw an aid circle or manually adjust the calibration board detection parameters. Select the *Start Drawing Aid Circle* button and draw a circle coinciding with a circle in the calibration board. After which, the detecting parameters of the circles should change.

If the user chooses to manually adjust the detecting parameters, simply click *Edit Detecting Parameters* and change the values accordingly.

After completing the above steps, click the *Check Camera Intrinsic Parameters* button again to get a new result. The steps are shown in *Figure 11* below.

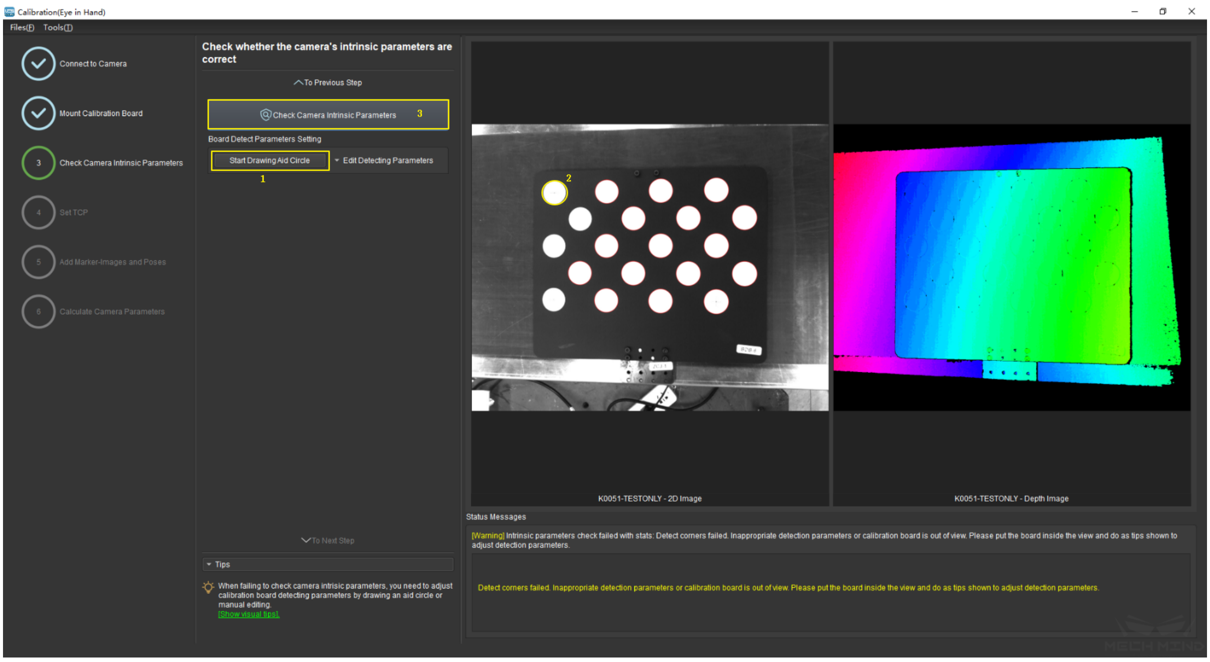


Figure 11 Draw the Aid Circle

Tip: If the calibration circles are too small to be selected easily, the user could right click the 2D image, uncheck *Fit to Window*, and select *Normal Size*. Start drawing aid circles after adjusting the size of the image.

If the feature points are still not detected, please adjust the camera settings according to the on-site operating conditions. Detailed information can be found in Mech-Eye Viewer.

Set TCP

- **Calibrate TCP**

In order to accurately measure the pose of the calibration board in the robot base coordinate, it is necessary to obtain the position of TCP relative to the origin of flange (default direction) first.

The Set TCP Value interface is shown in *Figure 12* below. If the TCP position is known, please check the *Known TCP Value* option and input the TCP value. If the TCP position is unknown, please check the *Unknown TCP Value* option and click the *Calibrate TCP* button.

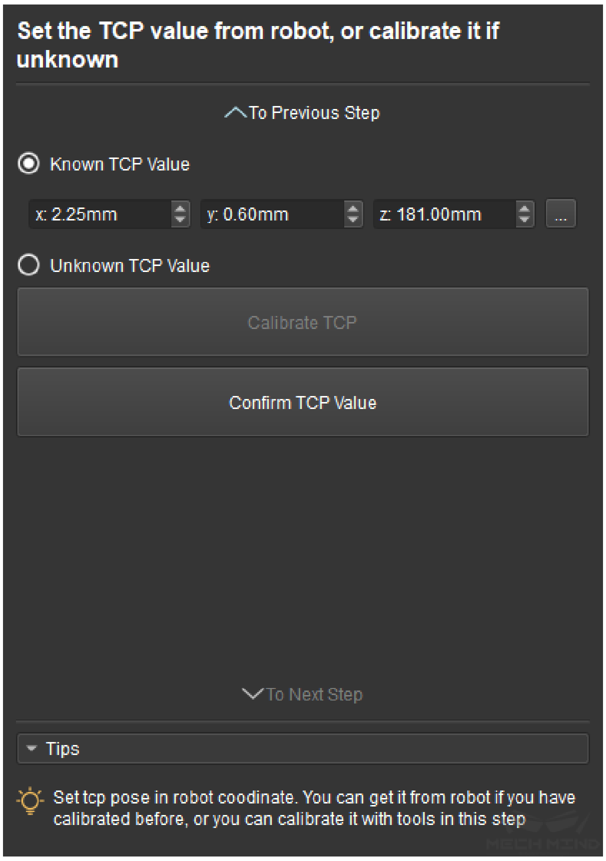


Figure 12 Set TCP

The steps are as follows:

1. Fix a sharp point in the robot work space. Let the robot touch this point from 4 different directions (namely 4 different poses). Click the *Add Pose* button every time the robot touches the point and the pose will be shown in the list.
2. *Delete Pose* will delete the pose which is not needed.
3. After adding 4 poses, click *Calculate TCP* to obtain the position of TCP. The error of calculation is shown in the pop-up window. As shown in *Figure 13* below.
4. After confirming that the TCP position is right, click *Confirm TCP Value* and *To Next Step* to finish the TCP calibration.

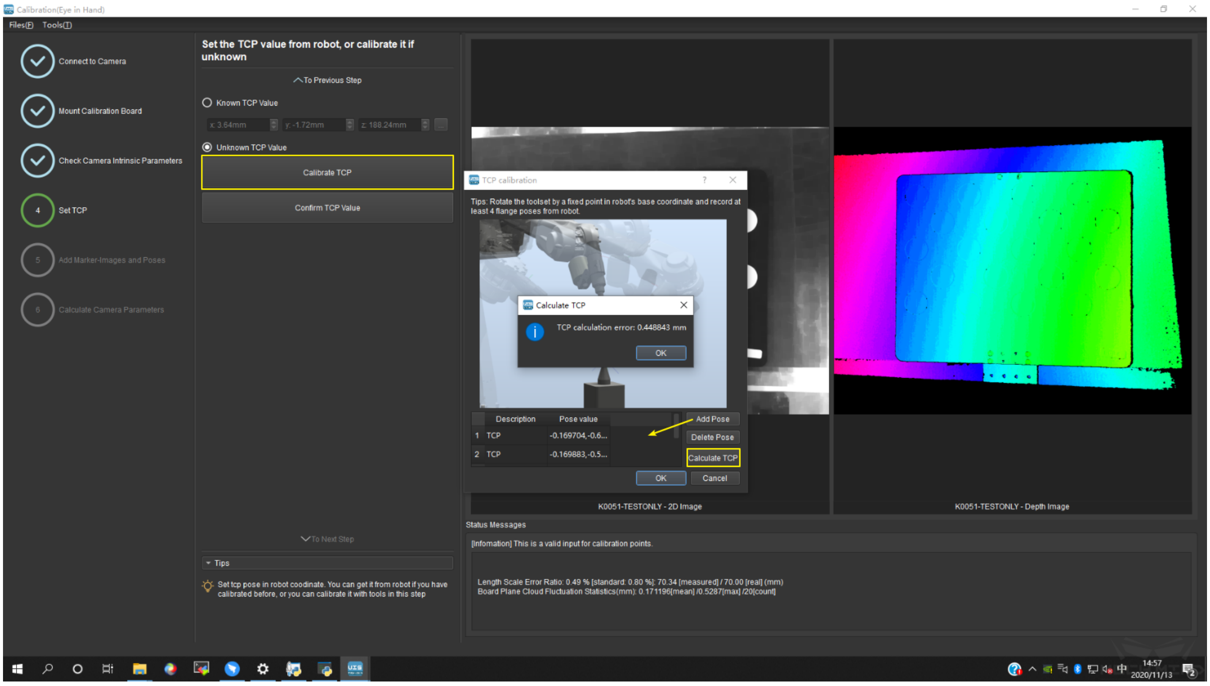


Figure 13 TCP Calculation Result

Tip: Some robots, such as ABB, cannot be controlled by the teach pendant after connecting with Mech-Center. The solution is to disconnect the robot from Mech-Center, move robot to touch the sharp point with the teach pendant, click *Add Pose*, and manually input the pose which is shown on the teach pendant.

Note: Make sure the TCP value set to 0 in the teaching pendant when checking *Known TCP Value*.

Add Marker-Images and Poses

Start calibrating after the value of TCP is set.

Drive the robot to touch P1, P2 and P3 on the calibration board. Everytime the robot touches a point, click *Add Pose* to add the pose into the list. After touching all three points, the three poses will be shown in the Touch Poses Viewer. This is shown in *Figure 14*.

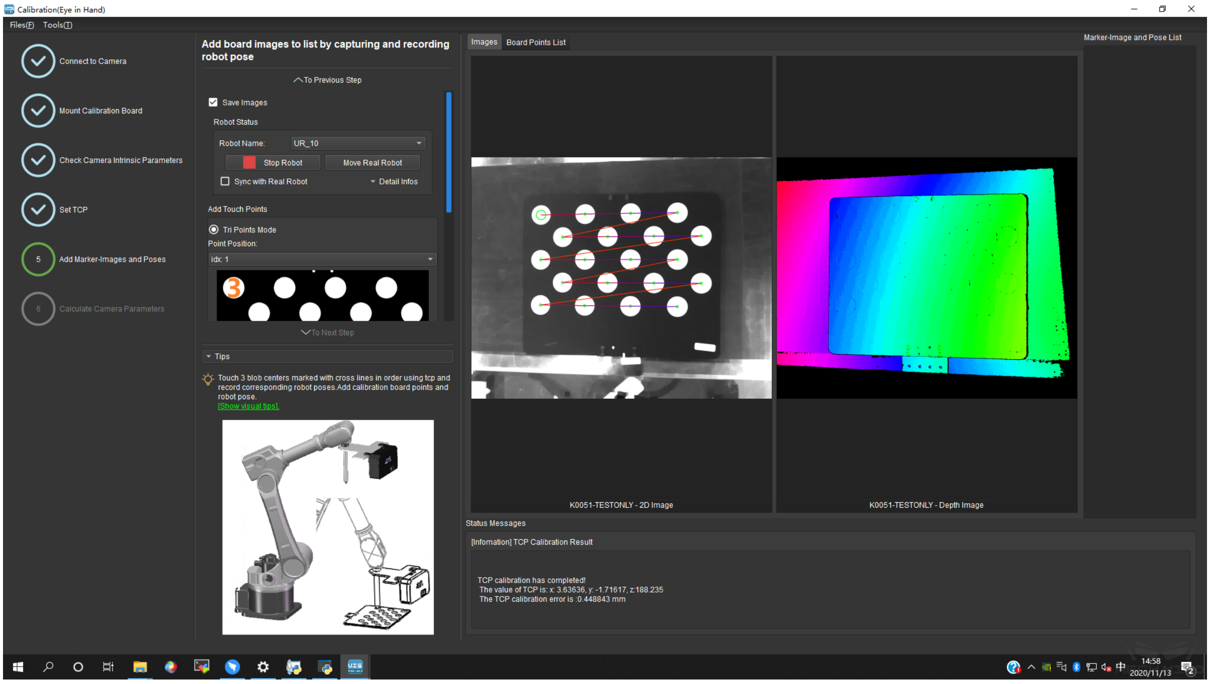


Figure 14 Add Three Poses

The real TCP touch for EIH is shown in *Figure 15* below.

Figure 15 Real TCP touch

Move the robot to a pose where the camera could capture the entire calibration board. Click the button *Add Single Board Image And Record Pose* to make the camera take a photo and detect the circles on the calibration board. Then click *To Next Step* to continue to the Calculate Camera Parameters page. The above steps are shown in *Figure 16* below.

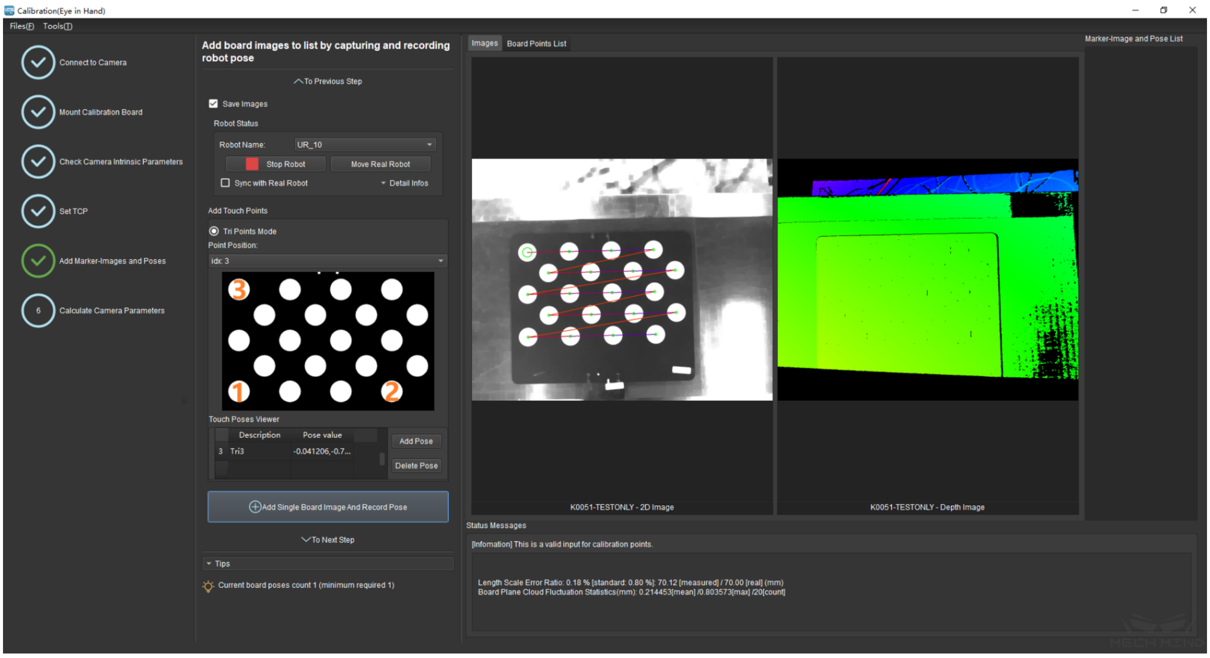


Figure 16 Add Calibration Point

Attention: Make sure the calibration board is stationary in the whole process.

Calculate Camera Parameters

- **Calculate Camera Parameters**

In the Calculate Camera Parameters interface, click the *Calculate Camera Parameters Result* button. The point cloud showing the calibration error and the calibration result will be displayed on the right part of the window, as shown in `Figure 17<#ETH_TCP_pic17>`___ below.

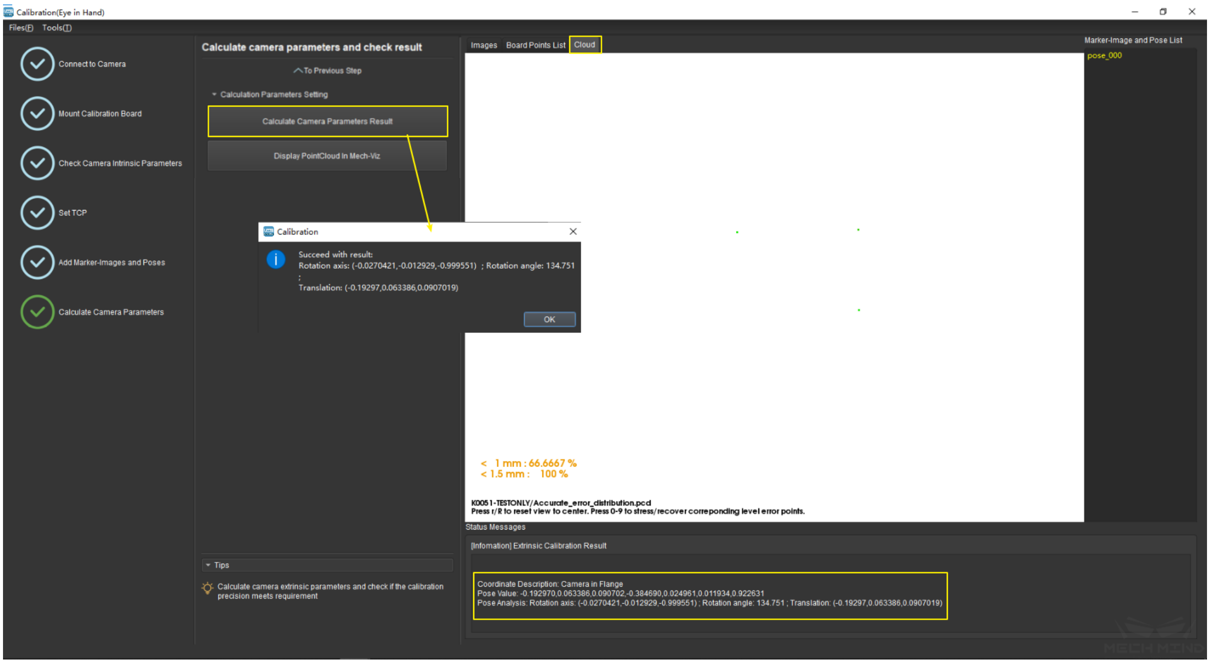


Figure 17 Calculate Camera Parameters

- **Display point cloud in Mech-Viz**

As shown in *Figure 18*, click the button *Display PointCloud in Mech-Viz*.

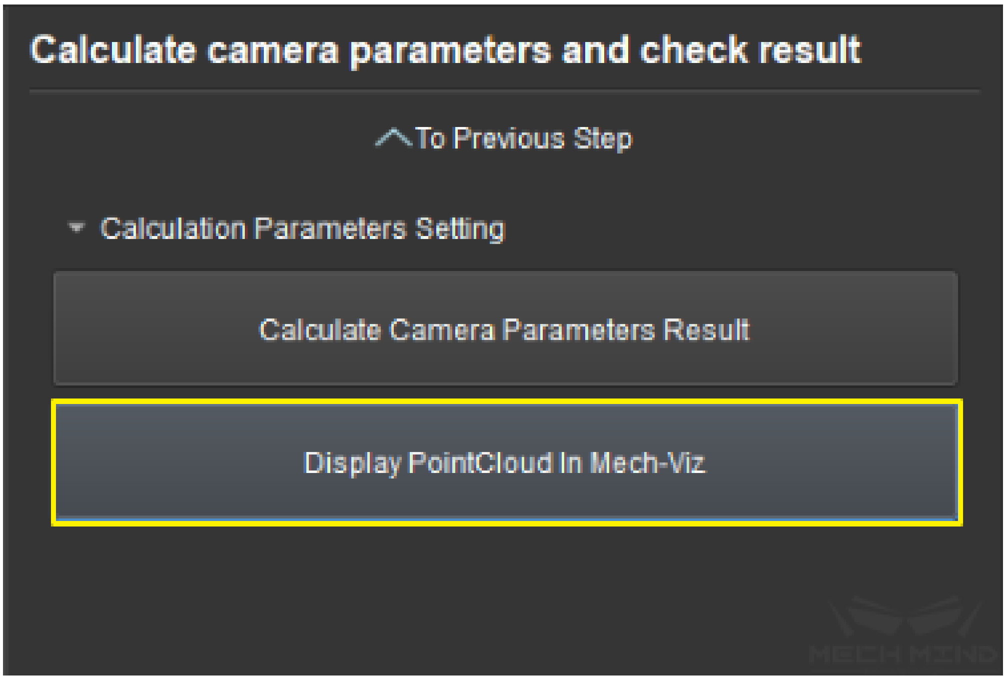


Figure 18 Display Point Cloud in Mech-Viz

Next, in Mech-Viz, select the *robot* tab. Click the *Configure Robot* button, and in the pop-up that appears, choose the correct robot type. After which, click the *Sync Robot* button. The above steps are shown in *Figure 19* below.

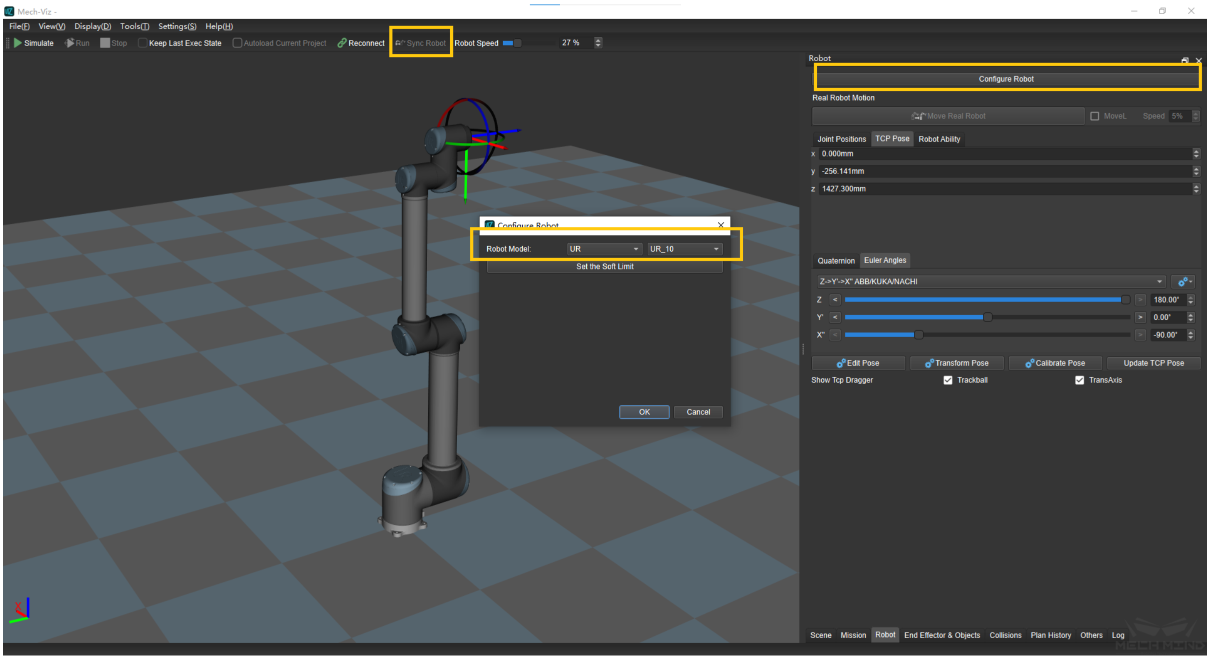


Figure 19 Synchronize Robot in Mech-Viz

Select the *scene* tab and adjust the ground height, as shown in *Figure 20*.

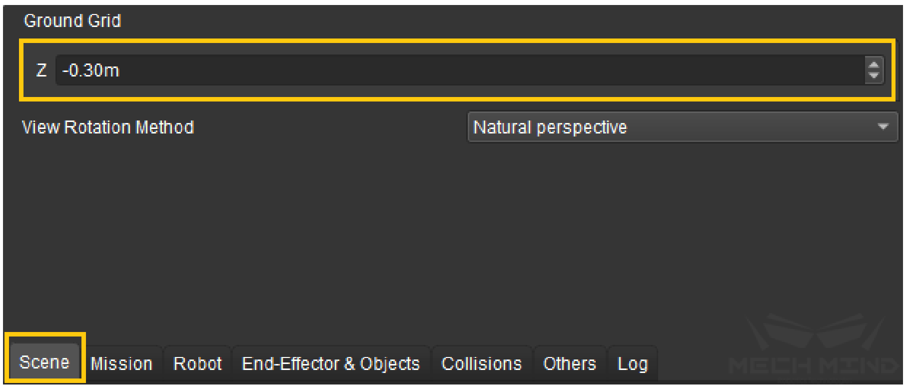


Figure 20 Adjust the Ground Height in Mech-Viz

Finally, the point cloud of the calibration board will be displayed as shown in *Figure 21*.

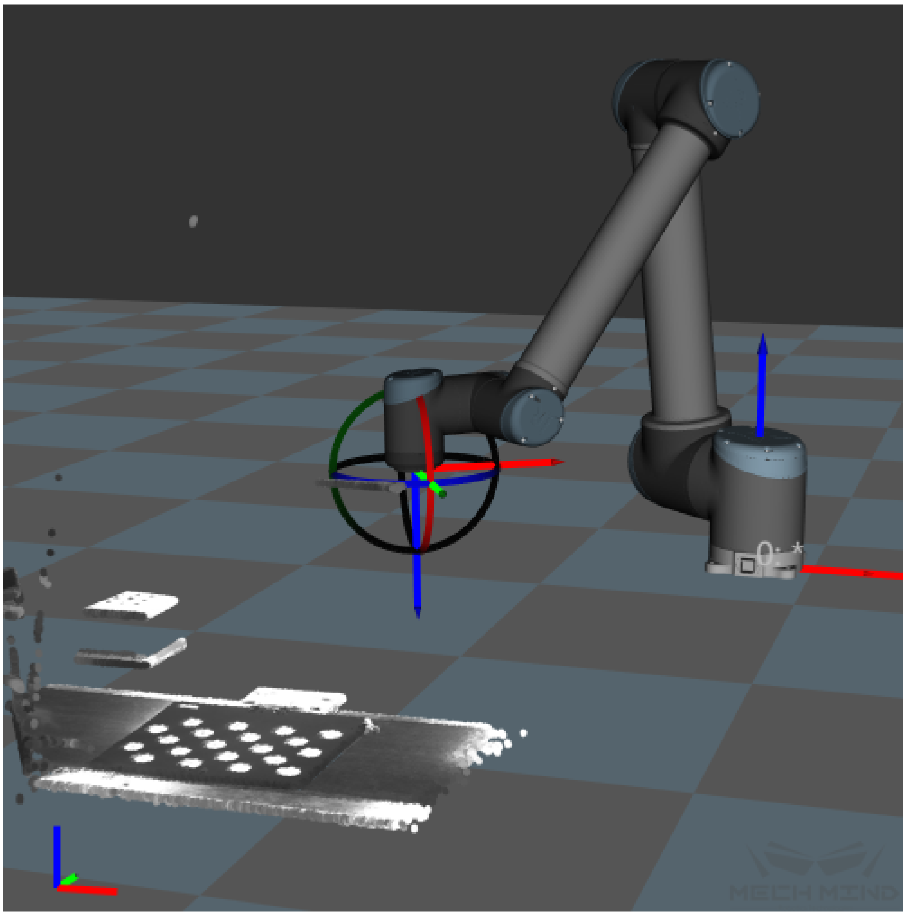


Figure 21 Point Cloud of the Calibration Board in Mech-Viz

6.2.2 Eye In Hand

Calibration Mechanism

Basic Mechanism of Calibration under EIH

Calibration Process

Add Calibration Points by Multiple Random Board Poses

Add Calibration Points by TCP Touch

Basic Mechanism of Calibration under EIH

When using the multiple random board pose method, the camera is mounted on a frame which is fixed at the end of the robot. The calibration board is placed within the camera' s field of view. The pose of the calibration board in the robot base coordinate system, denoted by A, is a measurable constant. The pose of the flange in the robot base coordinate system 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 spatial relationship between the center of the flange and the camera' s optical center, denoted by X, is an unknown constant. The above constants and variables are shown in *Figure 1*. Thus, A, B, C, and X form a closed loop, as shown in the following 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 pictures 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.

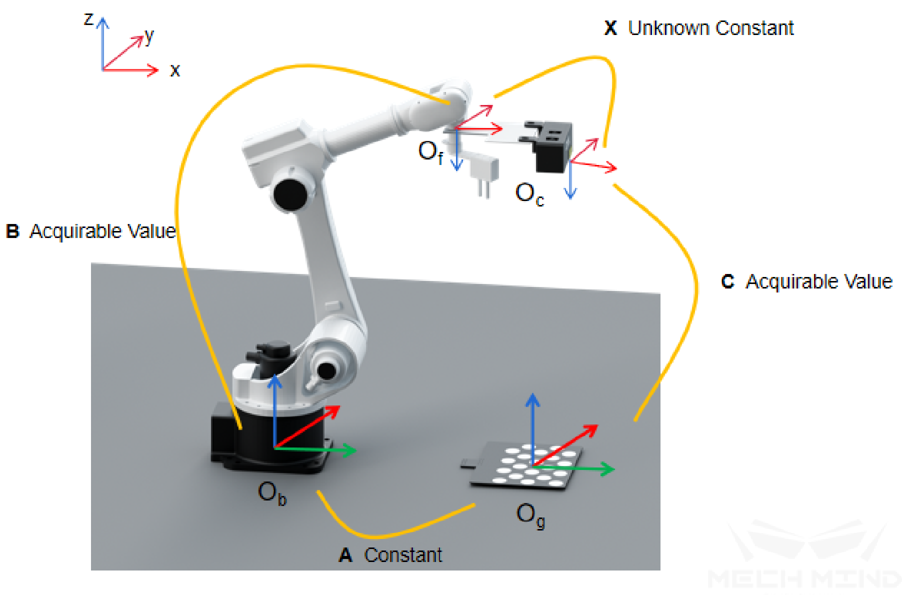


Figure 1. The multiple random board poses method (EIH)

$$B_1 \bullet X \bullet C_1 = A$$

$$B_2 \bullet X \bullet C_2 = A$$

$$B_1 \bullet X \bullet C_1 = B_2 \bullet X \bullet C_2$$

When using the TCP touch method, place the calibration board on the object plane and install a sharp-tipped tool with known dimensions on the robot flange. Then, touch the mark points on the calibration board with the center point of the tool by moving the robot. As shown in *Figure 2*, since the values of A, B and C are known, the value of X can also be obtained.

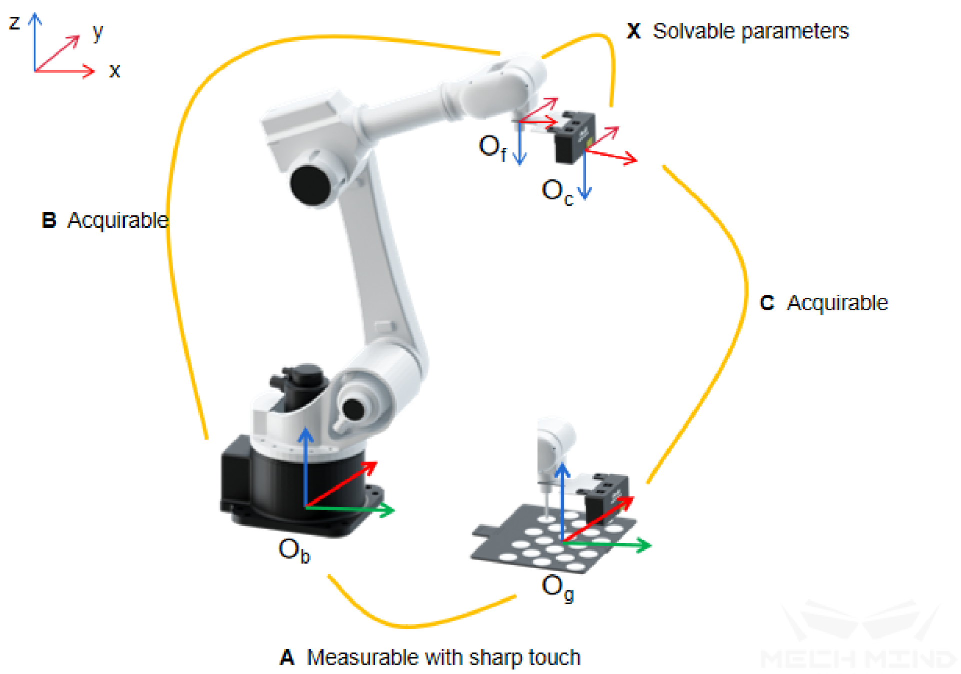


Figure 2. TCP Touch Method (EIH)

Calibration under EIH calibrates the transformation relationship between the coordinate systems of the camera' s optical center and the robot base. If the camera moves relative to the flange, the extrinsic parameters will be altered and must be recalibrated.

Add Calibration Points by Multiple Random Board Poses

Using the trajectories generated by the software or the multiple poses added manually, capture images of calibration points at each waypoint on the trajectories in order to construct the correct relationship between the calibration board, camera, and robot.

Calibration Preset

- **Select a directory to save calibration results**

Launch the Mech-Vision software and select *Camera* → *Camera Calibration* → Select a directory to save calibration results (if a Mech-Vision project is opened, *Use current project directory* is chosen by default), as shown in *Figure 1*.

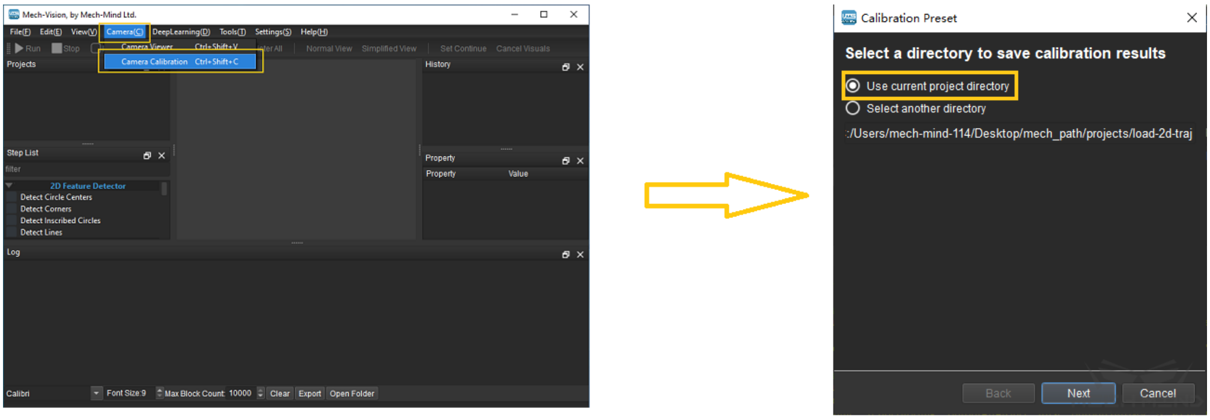


Figure 1 Select a Directory to Save Calibration Results

- **Select a calibration setup**

Select the corresponding calibration type, as shown in the *Figure 2*. *Eye in Hand* (EIH) will be used in the following section.

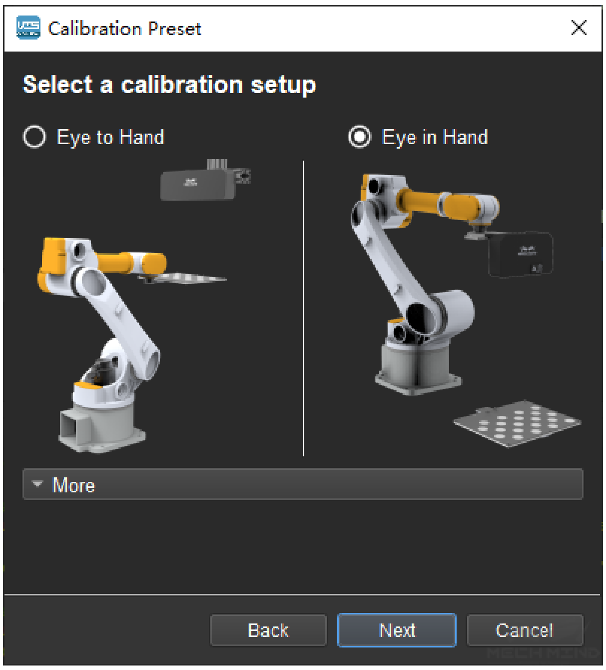


Figure 2 Select the Calibration Type

- **Check the robot connection status and select the calibration method**

If the robot is connected, the window in *Figure 3* will be shown. Select *Move and capture images automatically using robot* and click the *Finish* button to enter the calibration interface.

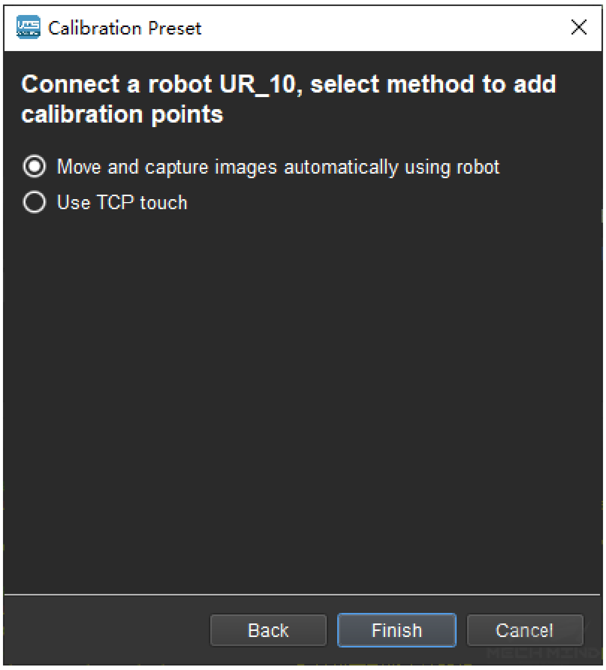


Figure 3 Select the Method to Add Calibration Points

If no robot is connected, select the correct robot type or go back to the previous step and retry after connecting a robot in Mech-Center, as shown in *Figure 4*.

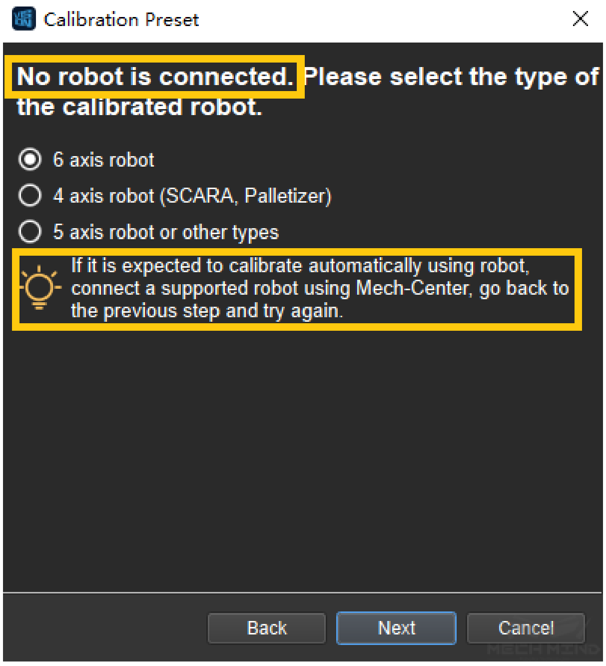


Figure 4 No robot is connected

Note: In the following calibration process, a connected 6 axis robot is used as an example. For the details to calibrate certain types of robots, please refer: *Multiple Random Board Poses*.

Connect to Camera

- **Select the camera**

When the network is well connected and the camera works normally, the camera ID will automatically appear under the "Detected Cameras & Local Parameter Groups" list. After selecting the camera, click *Connect*, and the camera will connect to the Mech-Vision software, as shown in *Figure 5* below.

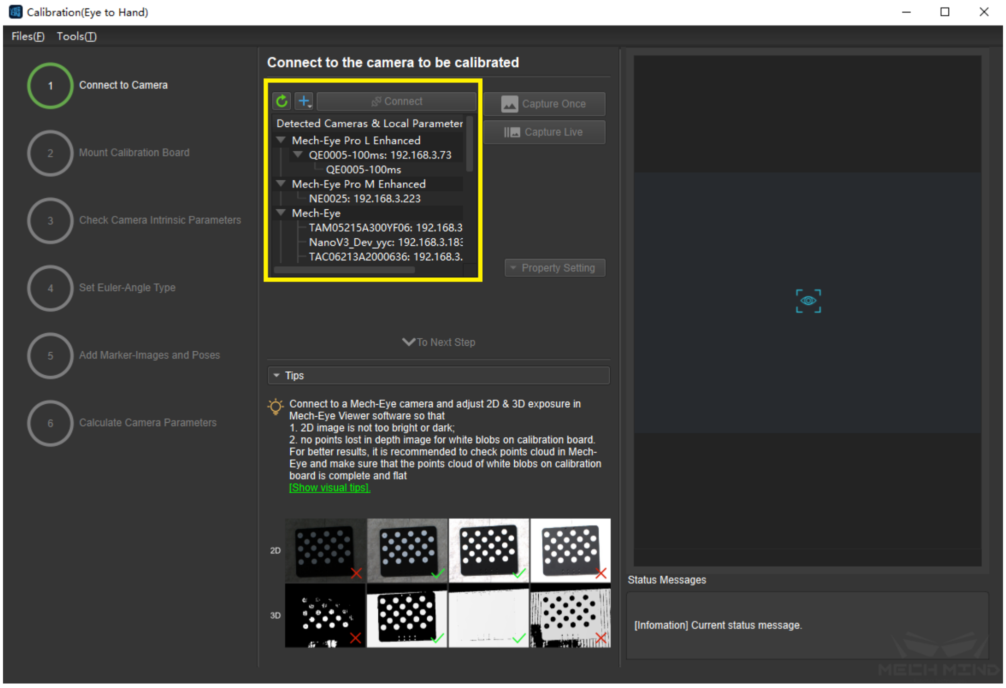


Figure 5 Connect to the Real Camera

- **Capture images**

After connecting to the camera, select *Capture Live* or *Capture Once* to view the captured images on the right side of the interface, as shown in *Figure 6* below (it is suggested to stop capturing images once the images in the view meet the requirements. Otherwise, multiple refreshes may affect the next calibration step).

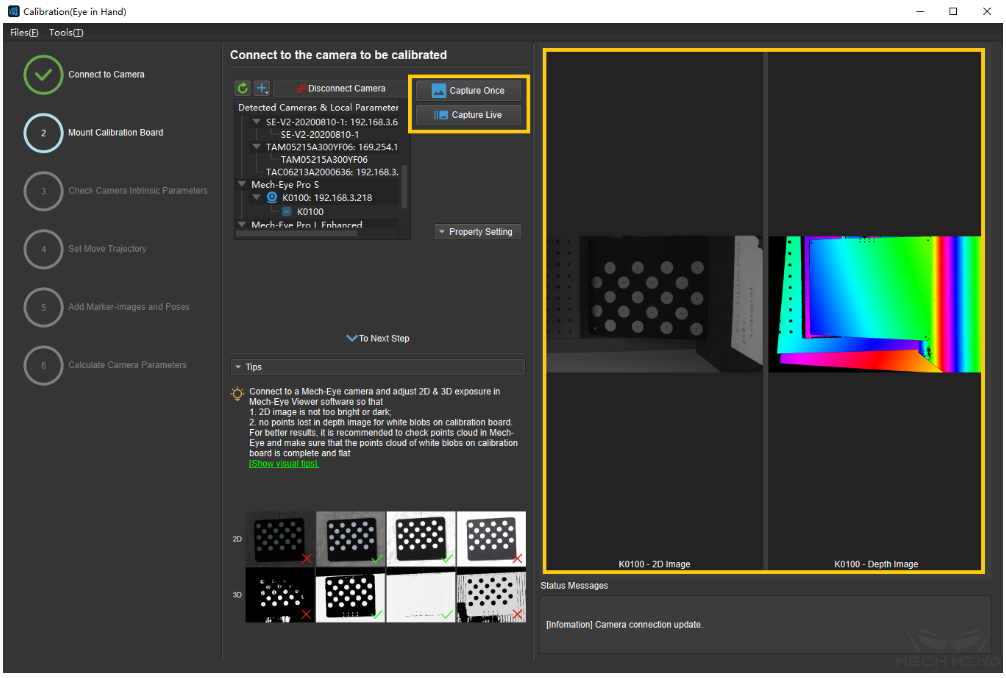


Figure 6 Capture Images

- **Manually add camera**

If the camera is functioning normally but cannot be found under the list of detected cameras, please manually add the camera IP. Click the + icon, and a dialog box will pop up, as shown in *Figure 7* below. Fill in the corresponding camera name and camera IP to add the camera to the detected cameras list. Then, connect the camera following the steps mentioned above.

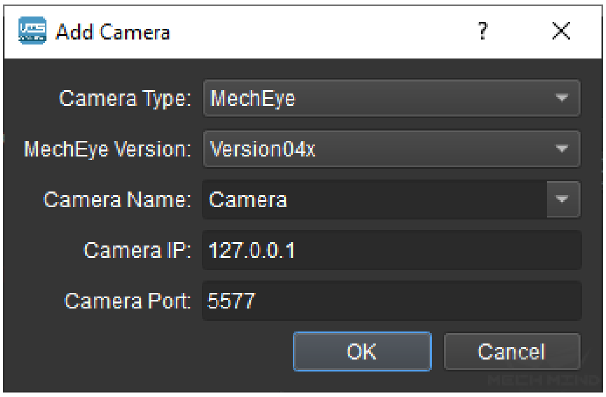


Figure 7 Manually Add Camera

Mount Calibration Board

Select the type of calibration board and put it in the camera’s field of view. The operation interface is shown in *Figure 8* below.

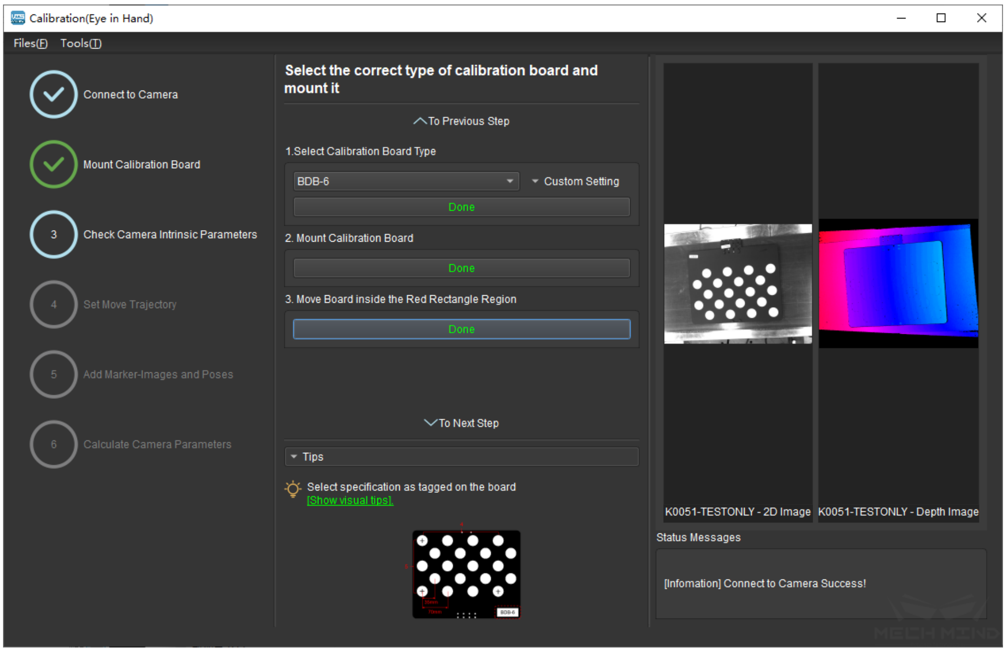


Figure 8 Select and Install the Calibration Board

Specific steps are as follows:

1. Select the corresponding calibration board model (the model nameplate is affixed on the top of the calibration board), and click the *Confirm* button.
2. Confirm that the camera is installed on the robot and the calibration board is within the camera’s field of view. Click the *Confirm* button when finished.
3. Make sure the calibration board is within the red rectangle in the 2D image, and click the *Confirm* button.

Check Camera Intrinsic Parameters

- **Check Camera Intrinsic Parameters**

After mounting the calibration board, check if the camera parameters are correct in order to ensure that feature points can be detected during the calibration process. Click the *Check Camera Intrinsic Parameters* button, and the result of the intrinsic parameter checking will be displayed in a pop-up window. *Figure 9* shows a successful intrinsic parameter check.

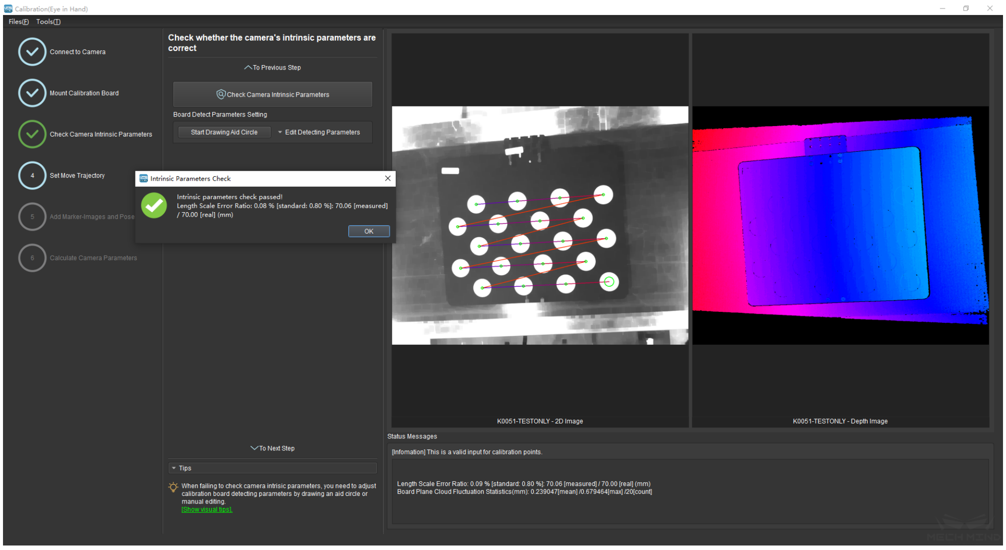


Figure 9 Check Camera Intrinsic Parameters

Generally, when the lighting conditions are good and the camera settings are appropriate, the intrinsic parameter check will pass by simply keeping the default values. *Figure 10* shows a successful detection of the feature points.

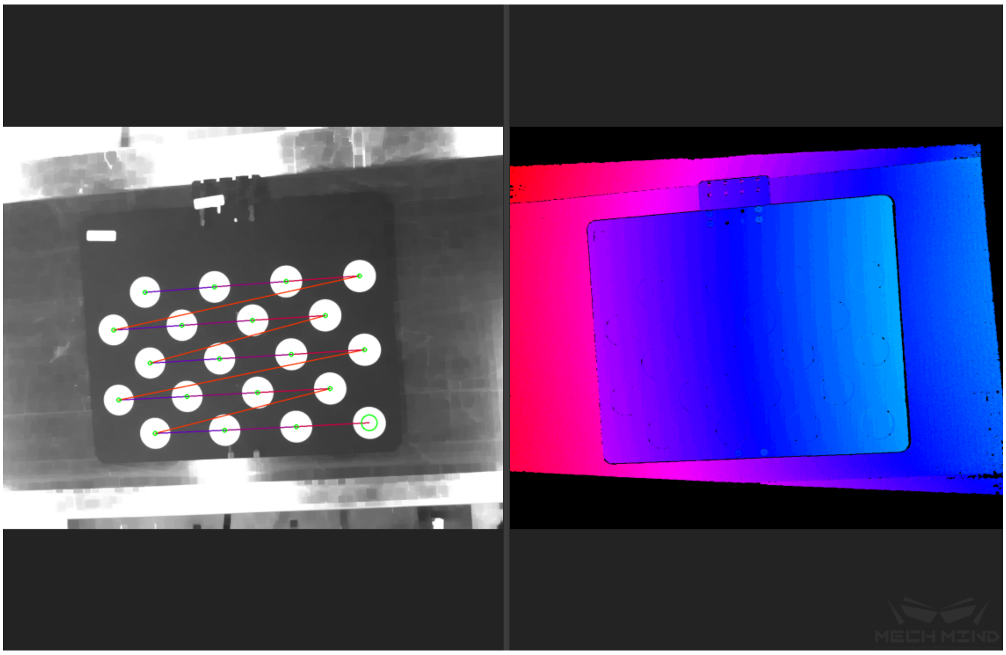


Figure 10 Detected Feature Points

- **Draw Aid Circle**

If the intrinsic parameter check fails, draw an aid circle or manually adjust the calibration board detection parameters. Select the *Start Drawing Aid Circle* button and draw a circle coinciding with a circle in the calibration board. After which, the detecting parameters of the circles should change.

If the user chooses to manually adjust the detecting parameters, simply click *Edit Detecting Parameters* and change the values accordingly.

After completing the above steps, click the *Check Camera Intrinsic Parameters* button again to get a new result. The steps are shown in *Figure 11* below.

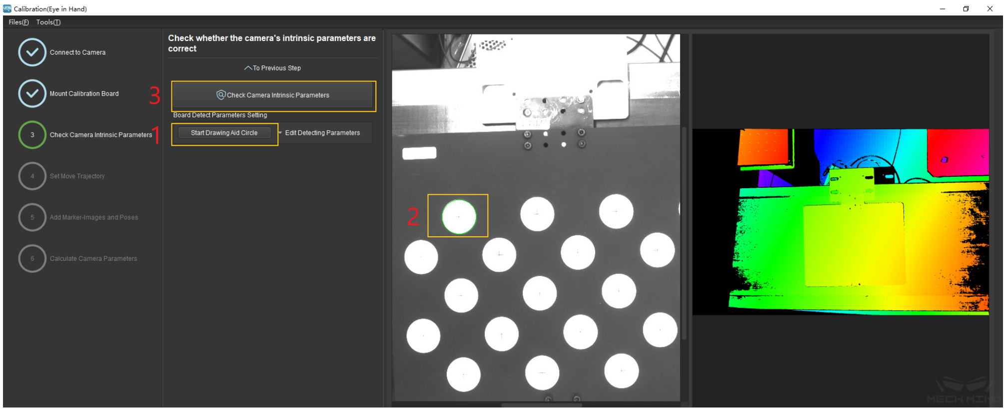


Figure 11 Draw the aid circle

Tip: If the calibration circles are too small to be selected easily, the user could right click the 2D image, uncheck *Fit to Window*, and select *Normal Size*. Start drawing aid circles after adjusting the size of the image.

If the feature points are still not detected, please adjust the camera settings according to the on-site operating conditions. Detailed information can be found in Mech-Eye Viewer.

Set Move Trajectory

- **Set Calibration Height Range**

The robot trajectory will be automatically generated. In order to ensure that the robot can successfully complete the movement, it is necessary to set a proper height according to the size of the robot’s working space. The details are shown in the *Figure 12*.

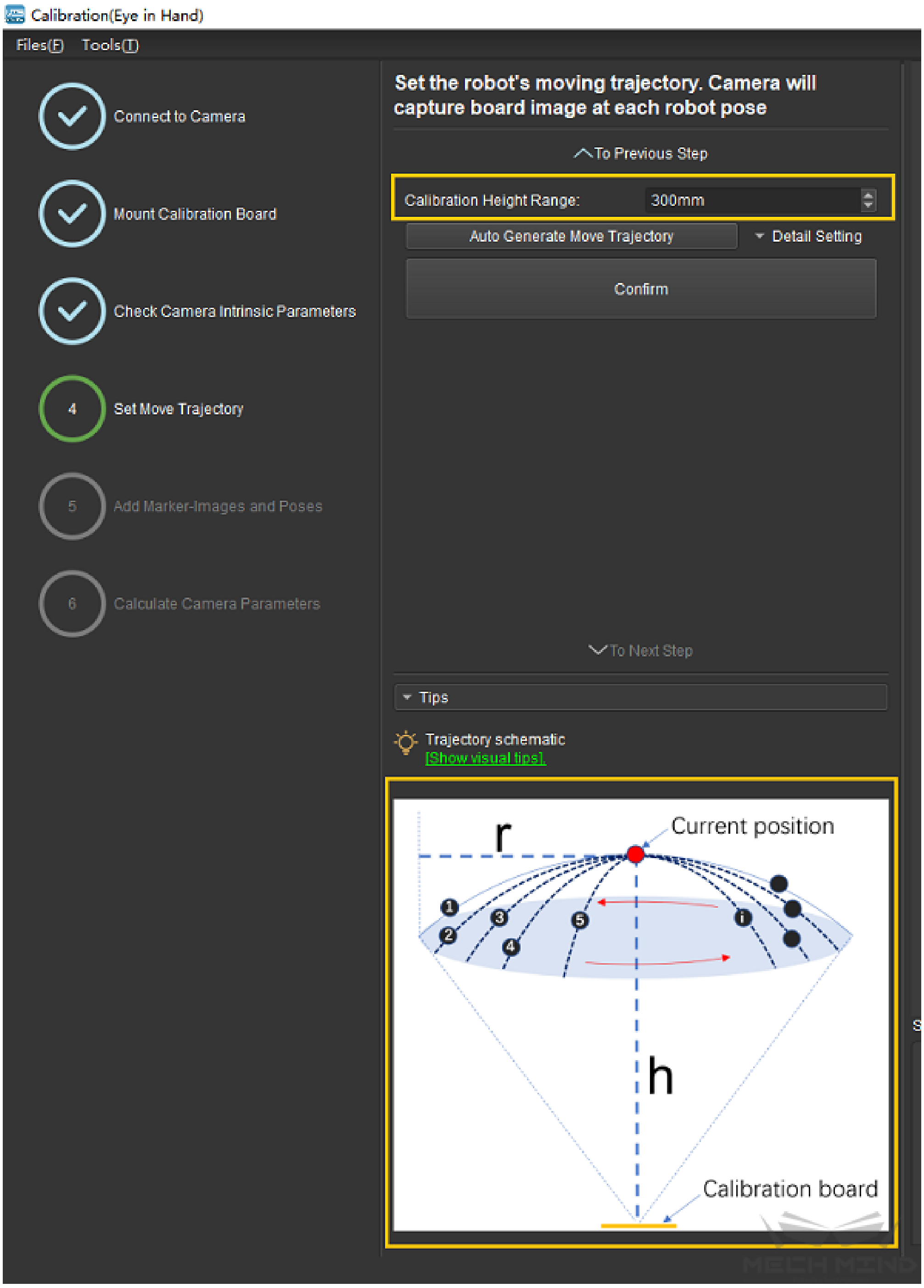


Figure 12 Set the Calibration Height Range

- **Automatically generate move trajectory**

After setting the trajectory range and ensuring that Mech-Viz is connected properly, click the *Detail Setting* button to set detailed parameters such as height, layer numbers, etc. Then, click the *Auto Generate Move Trajectory* button, and the program will automatically generate each waypoint of the trajectory. A dialog box will pop up after completion. The steps mentioned above are shown in *Figure 13* and *Figure 14* below.

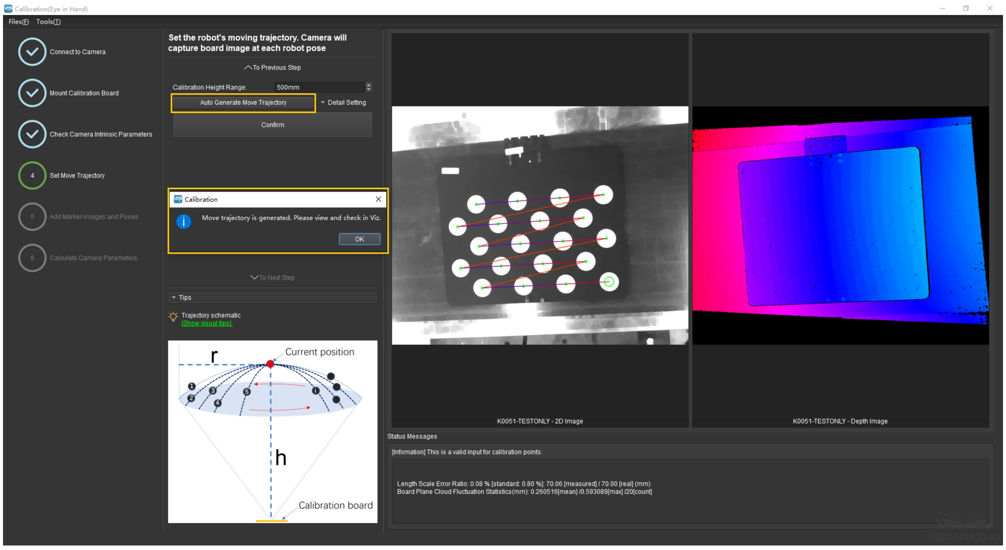


Figure 13 Automatically generated trajectory

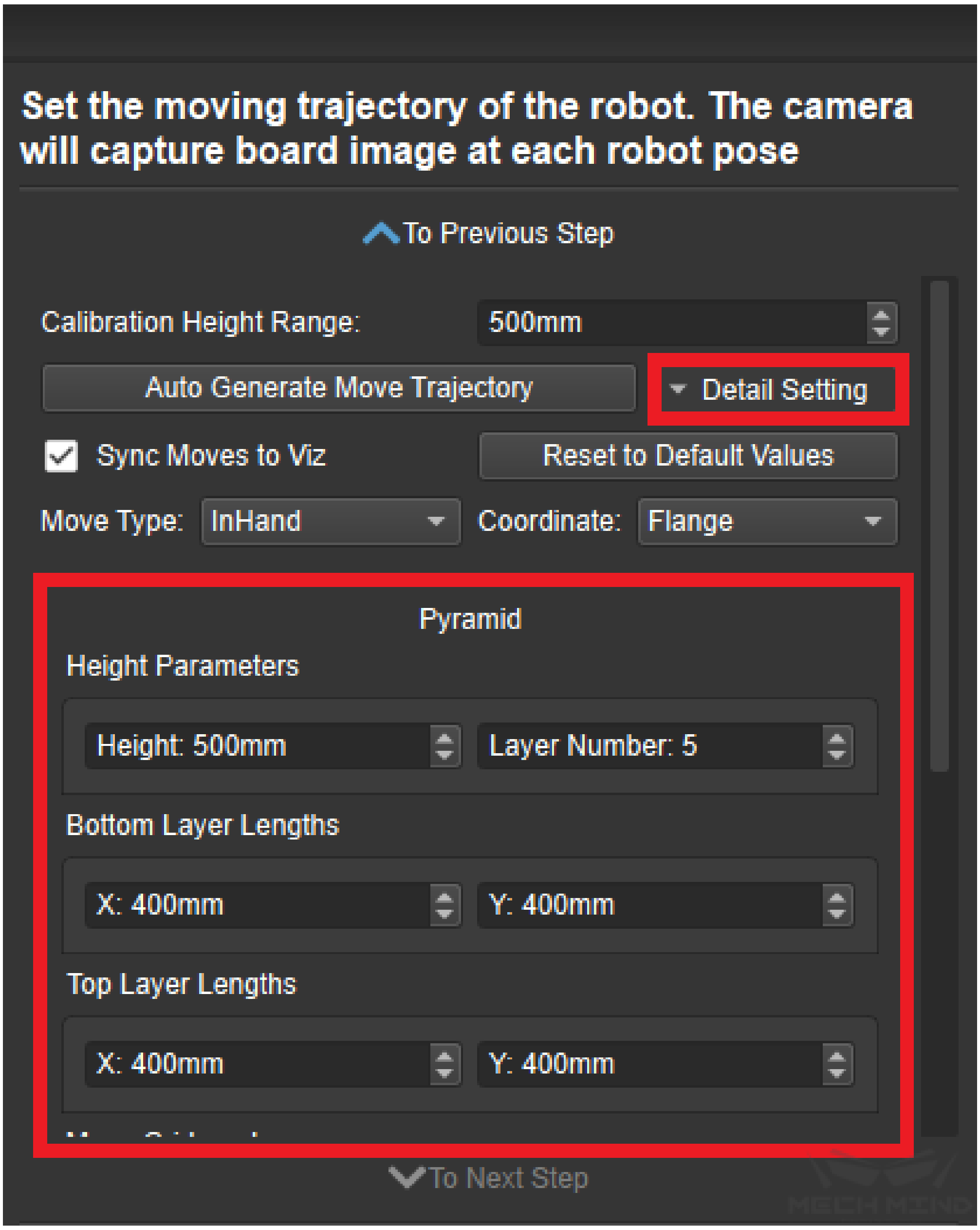


Figure 14 Detail Setting

- **IMPORTANT!!! Check trajectory in Mech-Viz software**

In Mech-Viz, click the *robot* tab. Select the current robot type and click the *Sync Robot* button. The distribution of the waypoints in the auto-generated trajectory is displayed, as shown in *Figure 15*. The user must check that the trajectory is reasonable and will not collide with obstacles in the environment!!!

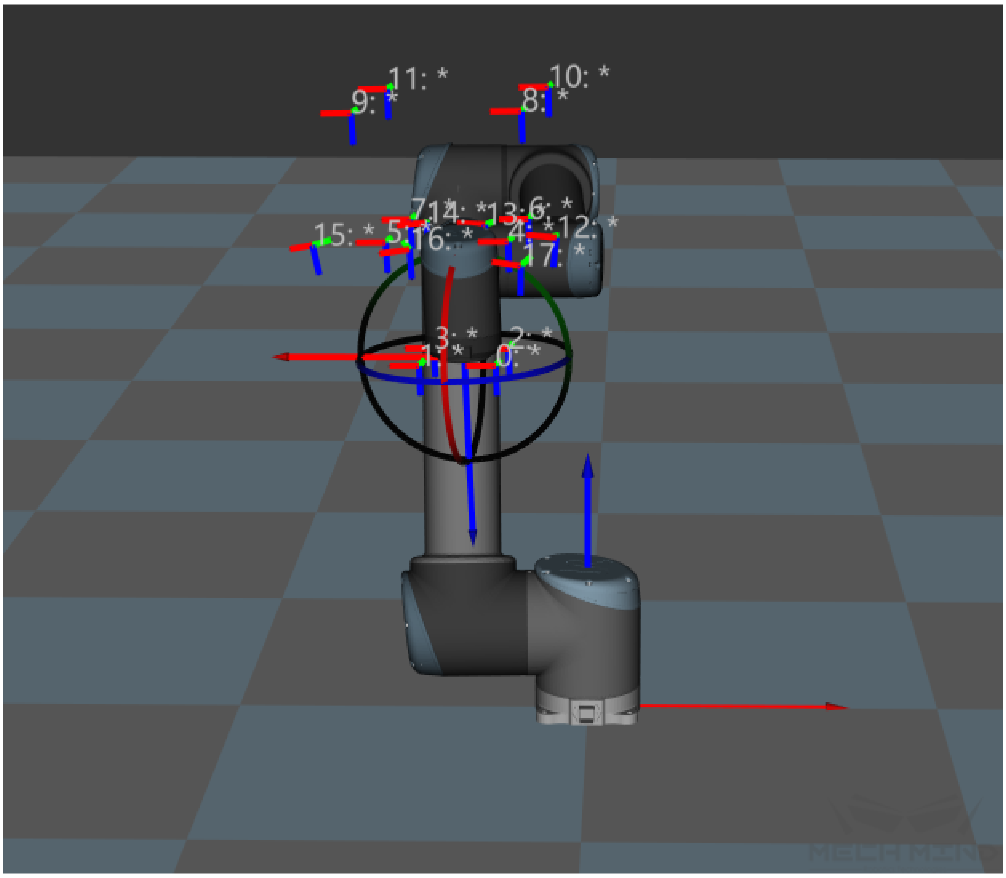


Figure 15 Check the Automatically Generated Trajectory in Mech-Viz

After ensuring that the trajectory is safe, return to Mech-Vision and click the *Done* and *To Next Step* buttons, as shown in *Figure 16* below.

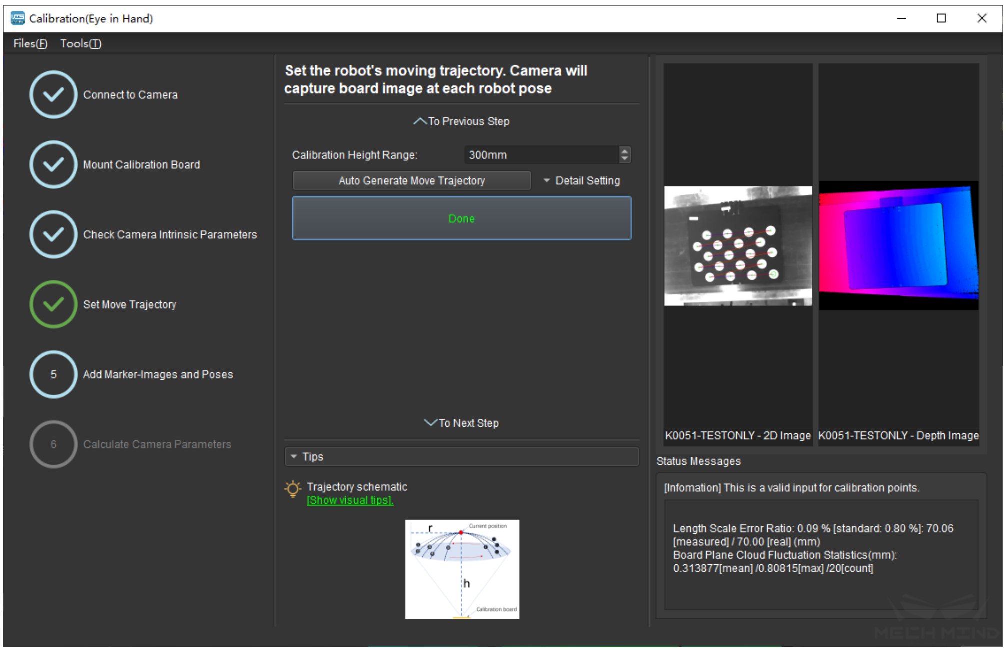


Figure 16 Confirm the Automatically Generated Trajectory

Add Marker-Images and Poses

- **Preset for calibration**

Enter the Add Marker-Images and Poses interface. As shown in the *Figure 17*, check the *Save Images* option, choose the corresponding robot name, and click the *Move Robot along Trajectory and Add Board Images* button.

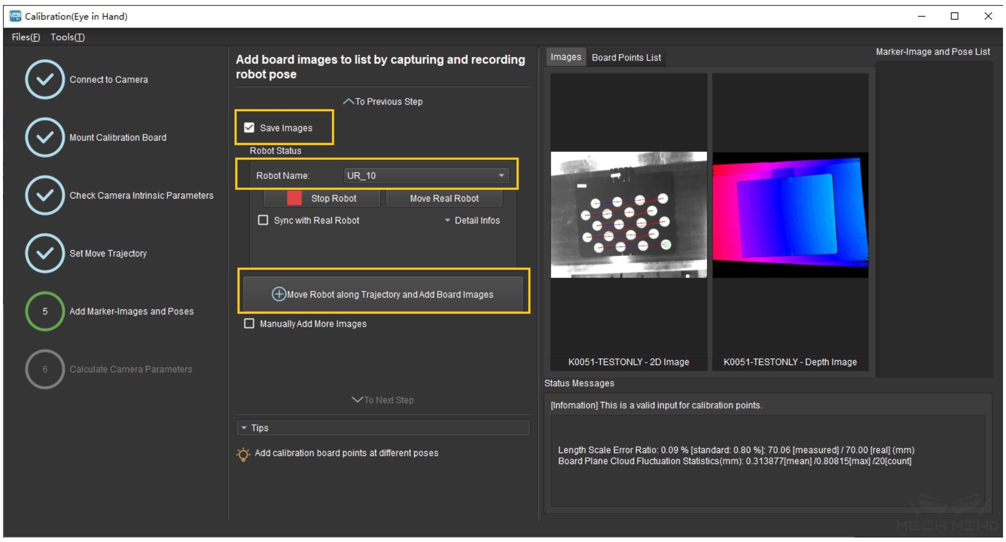


Figure 17 Preset for Calibration

A safety tips will pop up before starting the calibration, as shown in the *Figure 18*. Make sure that the operation environment is safe and then click the *OK* button.

Figure 18 Safety Tips before Starting Calibration

- **Calibration process**

Figure 19 show the interface when the calibration process is running. Please stay away from the robot working area to keep safe. The progress bar will load from 0% to 100% and the captured images can be seen in the right viewer at the same time.

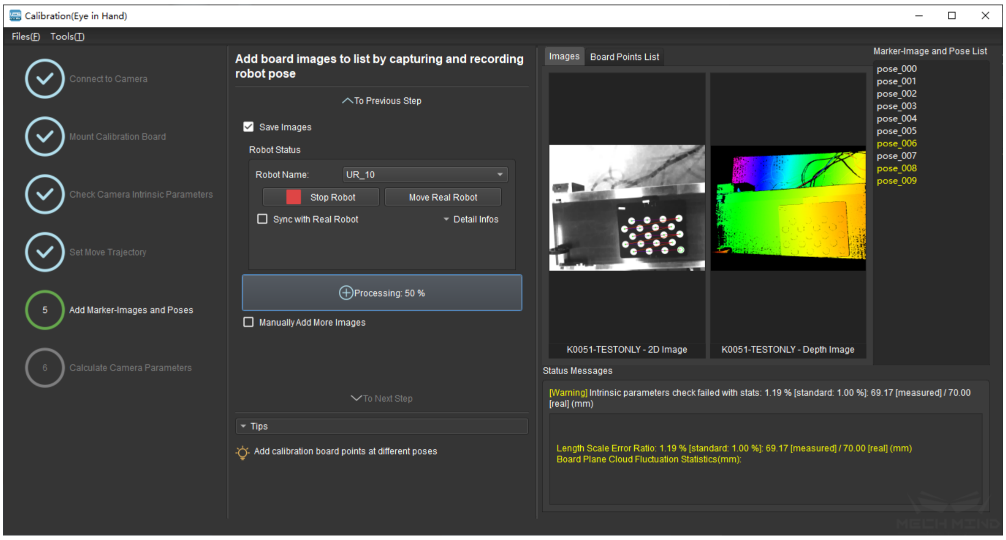


Figure 19 Calibration process is running

Attention: Clicking the *Stop Robot* button can exit the calibration process, but the robot will not stop until it finishes the current waypoints. In case of emergency, please press the emergency stop button on the robot teach pendant to stop the robot immediately.

- **Detect Deviation of measurements**

During the calibration process, a prompt box may appear, indicating that a measurement deviation has been detected, as shown in the *Figure 20*. It is recommended to recalibrate the camera intrinsic parameters and check whether the robot loses its absolute accuracy. The users could choose to continue this calibration process by clicking the *Ignore* button. However, the calibration accuracy cannot be guaranteed.

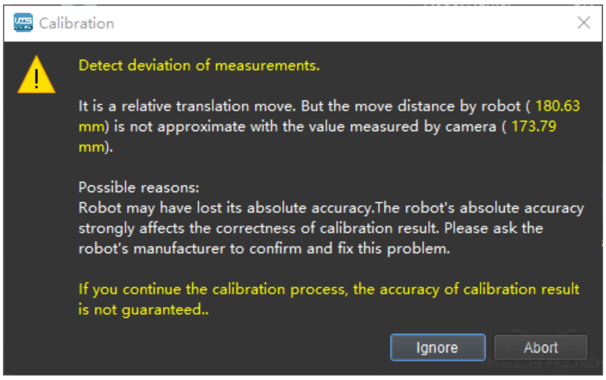


Figure 20 Detect a Measurement Deviation

- **Image acquisition completed**

The interface after the the completion of the image acquisition is shown in *Figure 21* below. It is necessary to determine whether the currently recognized calibration point meets the requirements from Tips. If so, click the *To Next Step* button. If not, please manually add the calibration points by moving the robot through the teach pendant or Mech-Viz software. Check the *Manually Add More Images* button, and click the *Add Single Calibration Board Image And Record Pose* button to collect images again.

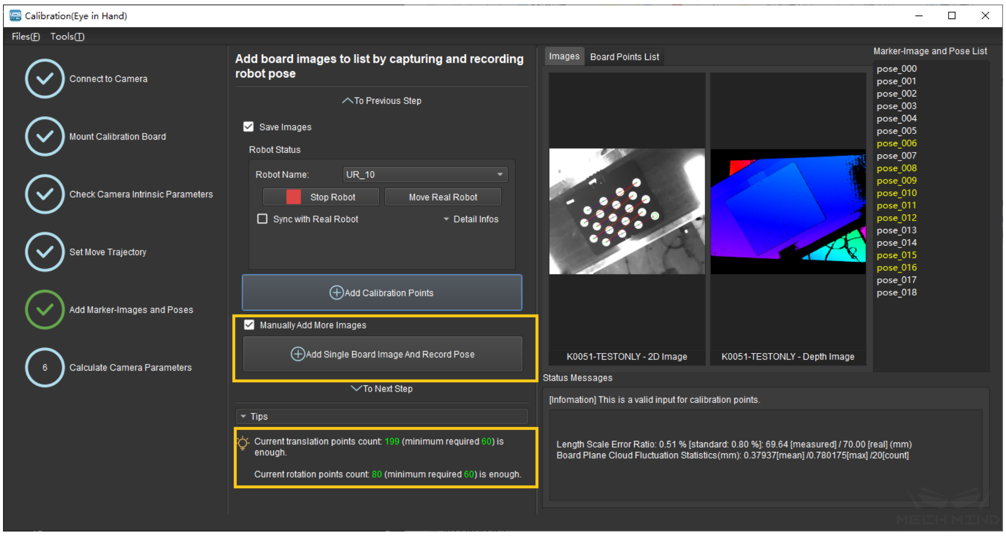


Figure 21 Image acquisition completed

Calculate Camera Parameters

- **Calculate Camera Parameters**

In the Calculate Camera Parameters interface, click the *Calculate Camera Parameters Results* button. Confirm and save the calibration data accordingly. The calibration result and the point cloud showing the calibration error will be displayed on the right part of the window, as shown in *Figure 22*.

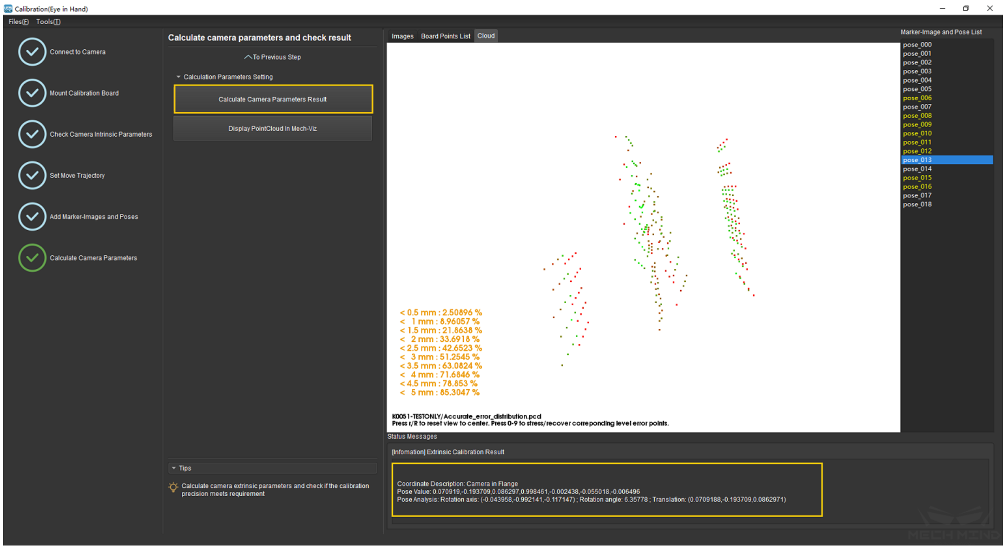


Figure 22 Camera Calibration Results

- **Display PointCloud in Mech-Viz software**

As shown in *Figure 23* below, click the *Display PointCloud in Mech-Viz* button.

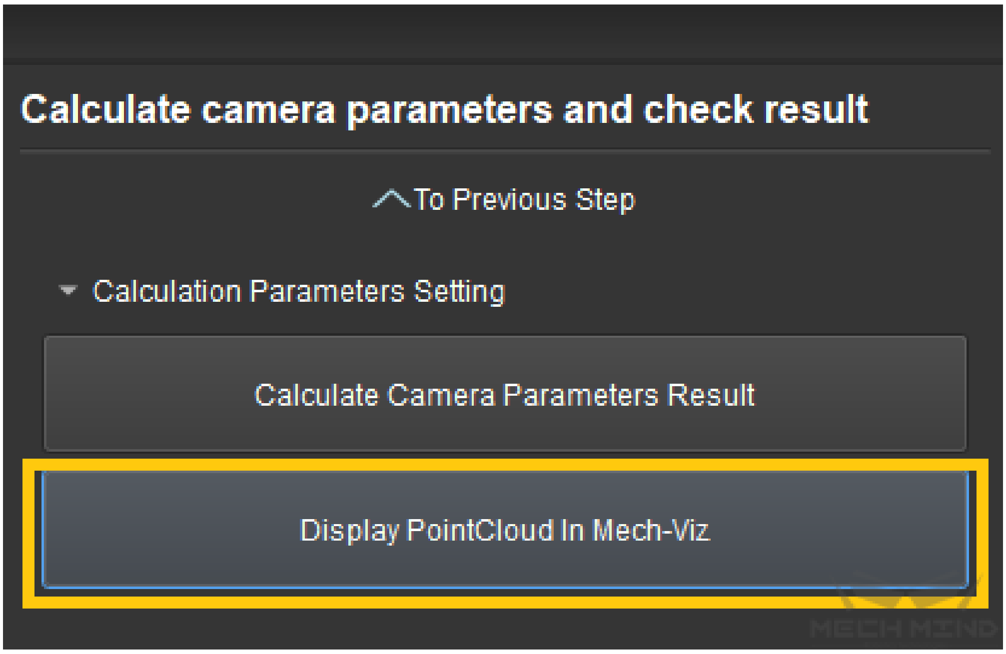


Figure 23 Display Point Cloud in Mech-Viz

Next, in Mech-Viz, select the *robot* tab. Click the *Configure Robot* button, and in the pop-up that appears, choose the correct robot type. After which, click the *Sync Robot* button. The above steps are shown in *Figure 24* below.

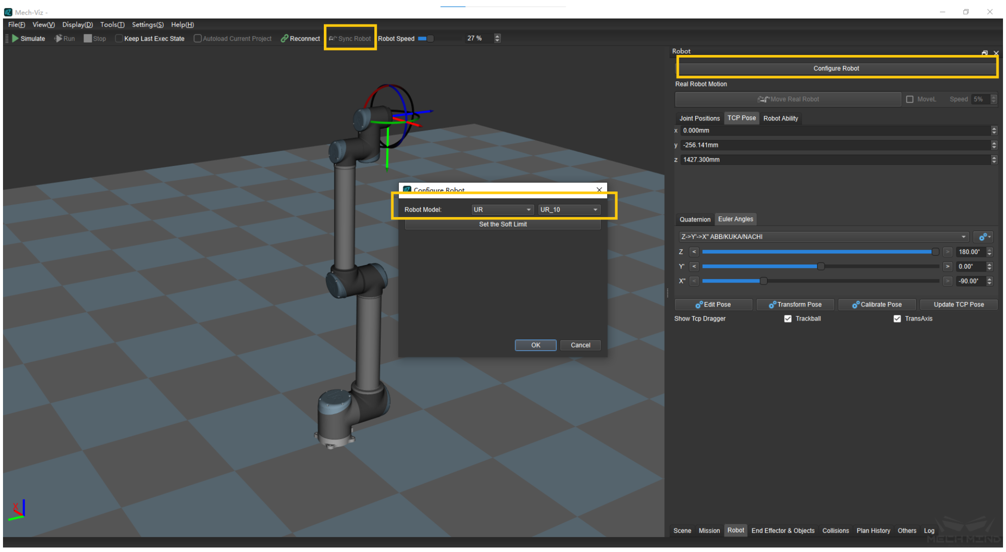


Figure 23 Sync Robot in Mech-Viz

Select the *Scene* tab, and adjust the ground height accordingly, as shown in *Figure 25*.

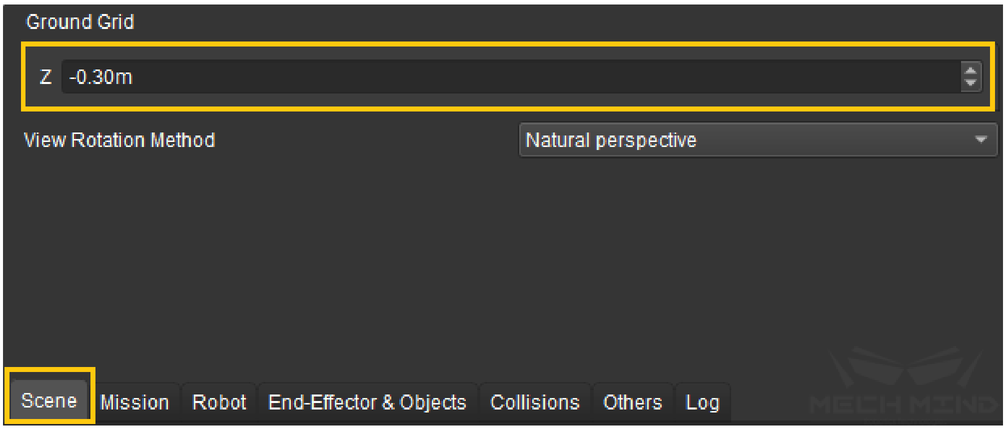


Figure 25 Adjust the Ground Height in Mech-Viz

Finally, the point cloud of the calibration board will be displayed as shown in *Figure 26*.

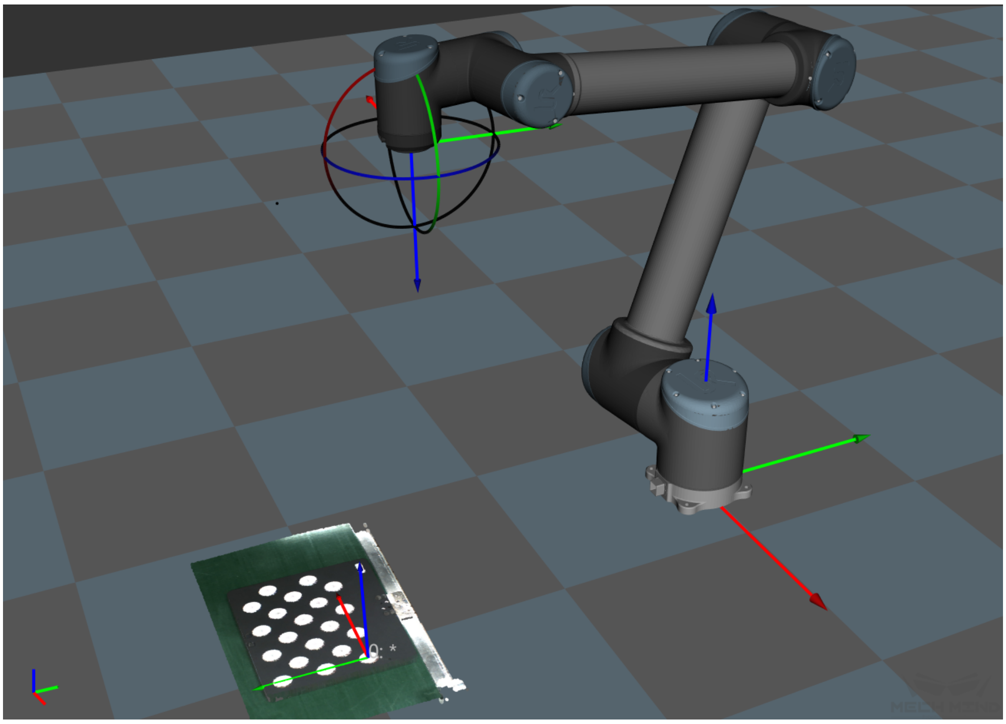


Figure 26 Point Cloud of Calibration Board in Mech-Viz

Add Calibration Points by TCP Touch

In special cases where the robot cannot be controlled by Mech-Viz or the calibration board cannot be installed, TCP touch is recommended for robot hand-eye calibration. Compared to other methods, TCP touch only needs one set of calibration points to accomplish the task.

Calibration Preset

- **Select a directory to save calibration results**

Launch Mech-Vision and click *Camera* → *Camera Calibration* → Select a directory to save calibration results (if a Mech-Vision project is opened, it will use current project directory by default), as shown in *Figure 1*.

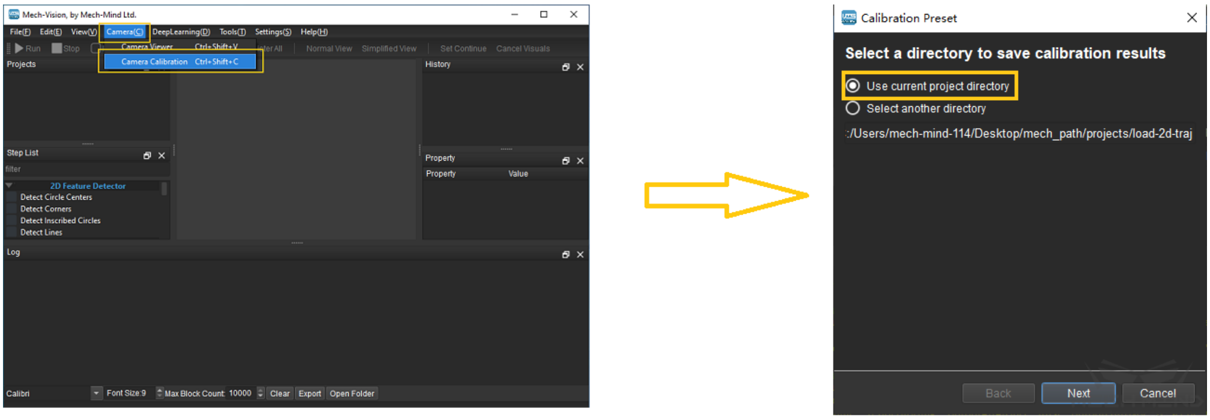


Figure 1 Select a Directory to Save Calibration Results

- **Select a calibration setup**

Select the corresponding calibration type, as shown in Figure 2. *Eye in Hand* (EIH) will be used in the following section.

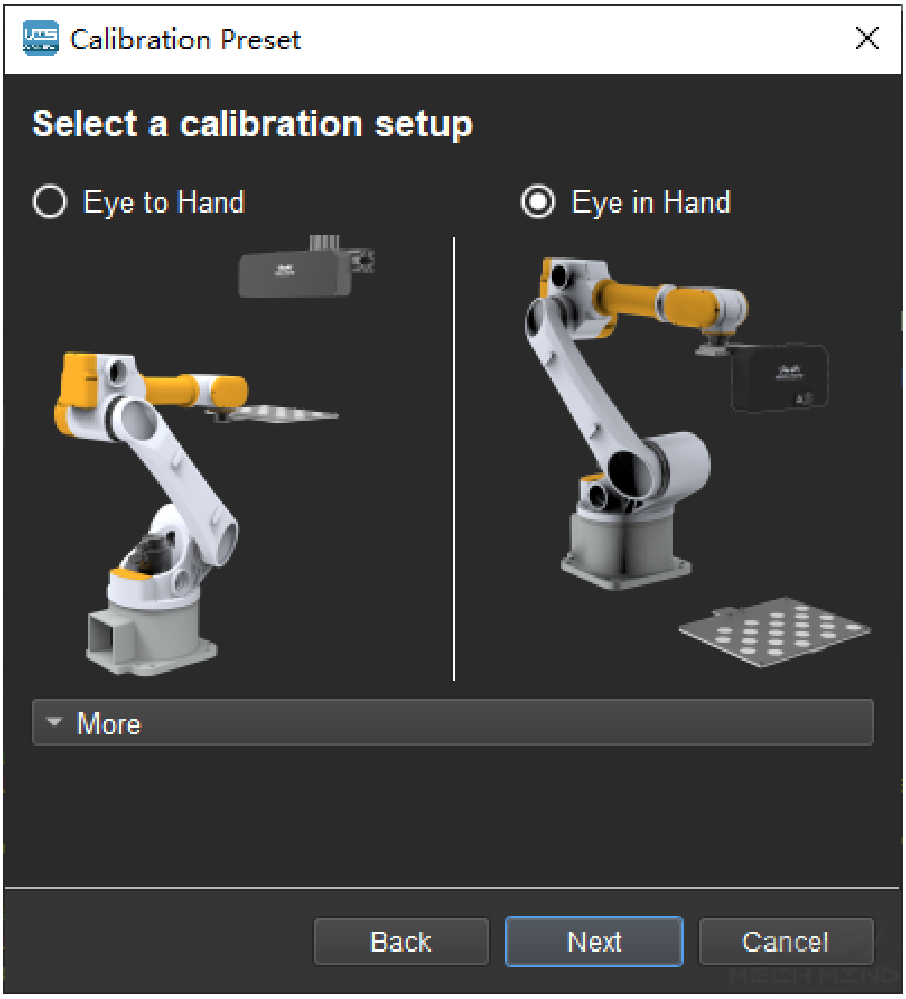


Figure 2 Select EIH Method

- **Check robot connection status and select a calibration method**

If the robot is connected, the window in *Figure 3* will be shown. Select *Use TCP touch* and click the *Finish* button to enter the calibration interface.

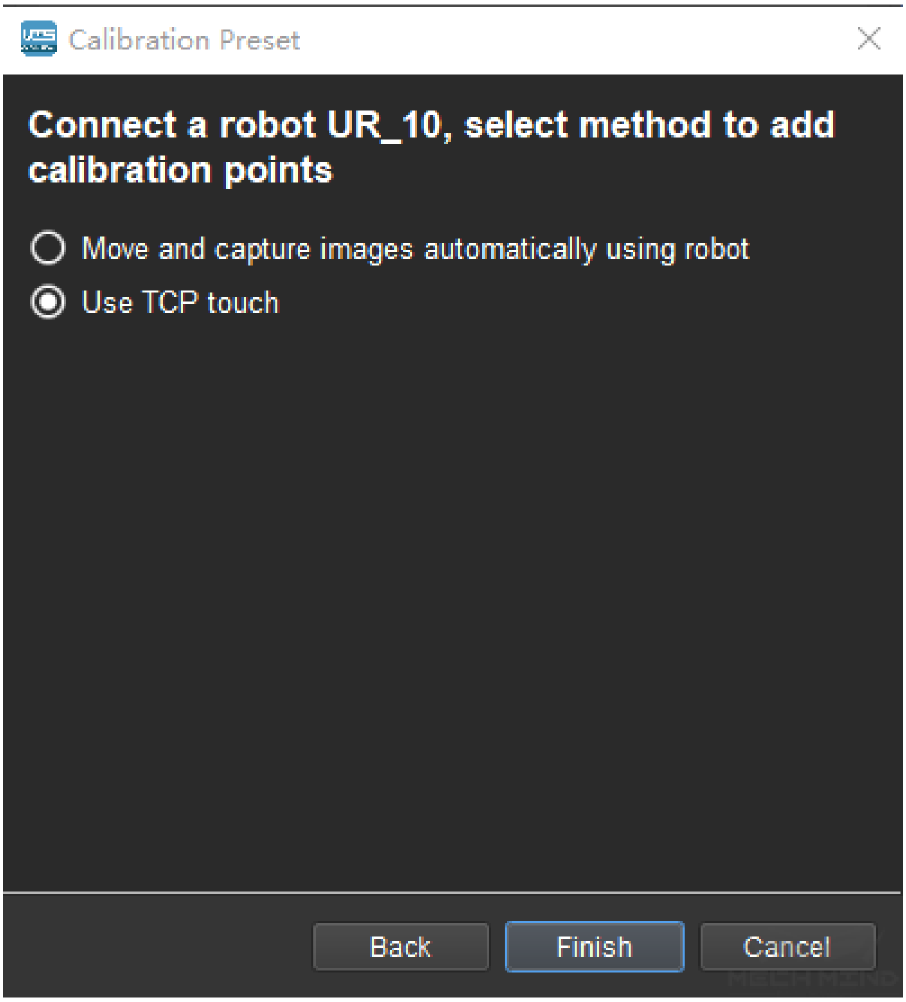


Figure 3 Select a Method to Add Calibration Points

If no robot is connected, select the correct robot type or go back to the previous step and retry after connecting a robot in Mech-Center, as shown in *Figure 4*.

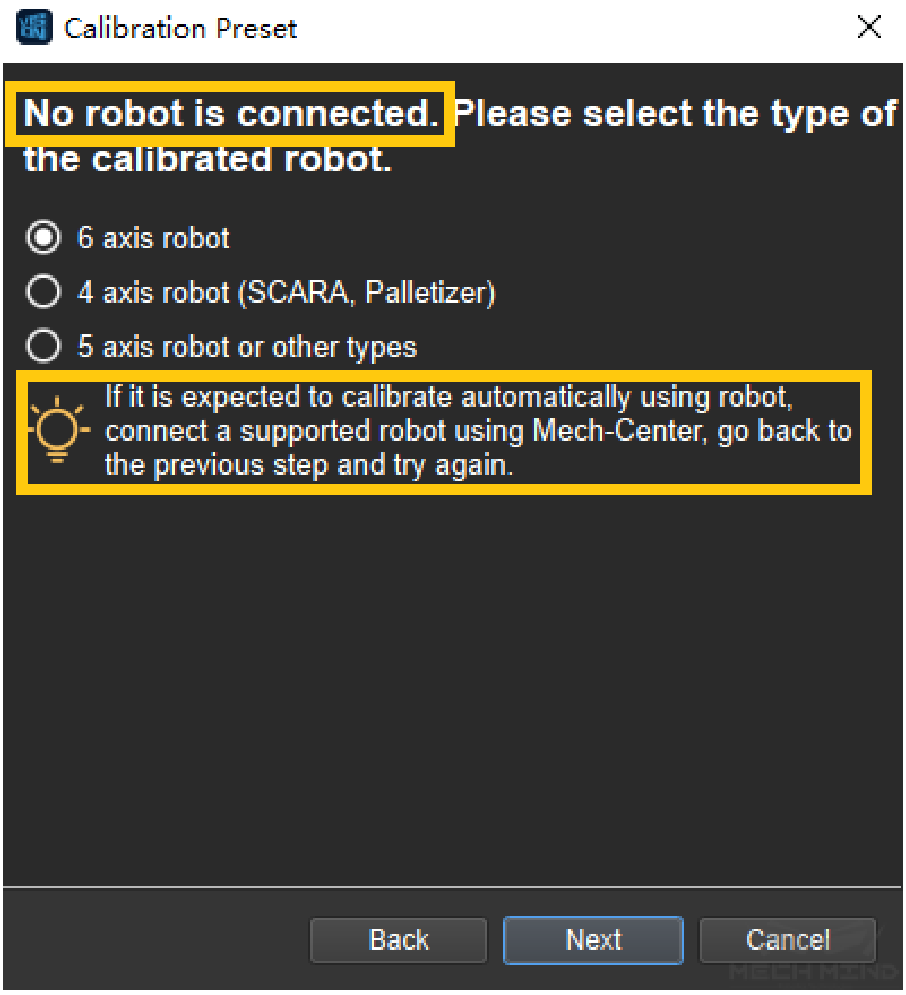


Figure 4 No Robot is Connected

Note: In the following calibration process, a connected 6 axis robot is used as an example. For the details on special types of robots, please refer to *TCP Touch*.

Connect to Camera

- **Select the camera**

When the network is well connected and the camera works normally, the camera ID will automatically appear under the "Detected Cameras & Local Parameter Groups" list. After selecting the camera, click *Connect*, and the camera will connect to the Mech-Vision software, as shown in *Figure 5* below.

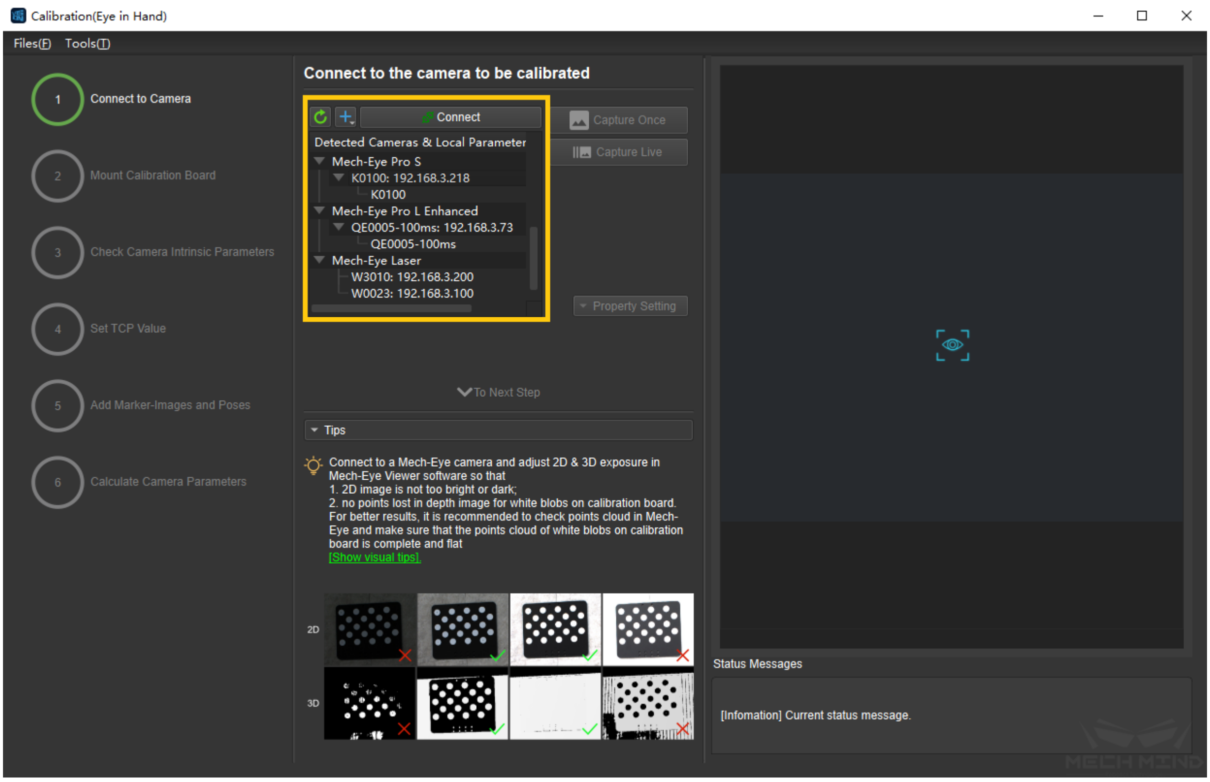


Figure 5 Connect the Camera

- **Capture images**

After connecting to the camera, select *Capture Live* or *Capture Once* to view the captured images on the right side of the interface, as shown in *Figure 6* below (it is suggested to stop capturing images once the images in the view meet the requirements. Otherwise, multiple refreshes may affect the next calibration step).

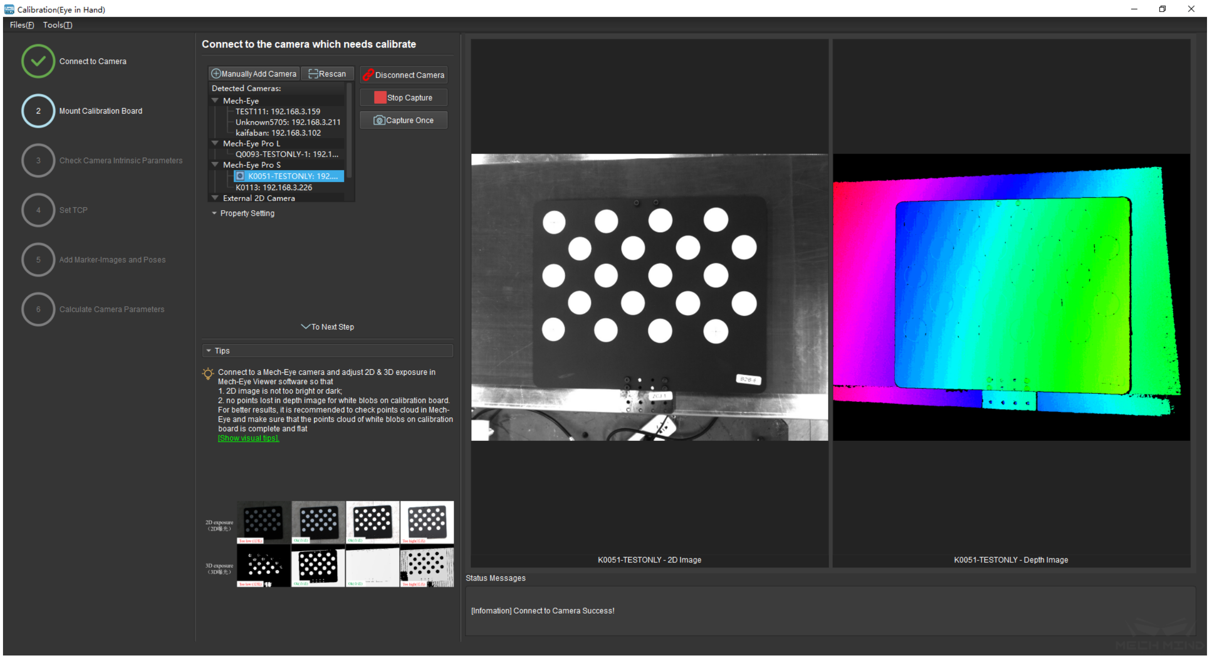


Figure 6 Capture Images

- **Manually Add Camera**

If the camera is functioning normally but cannot be found under the list of detected cameras, please manually add the camera IP. Click the + icon, and a dialog box will pop up, as shown in *Figure 7* below. Fill in the corresponding camera name and camera IP to add the camera to the detected cameras list. Then, connect the camera following the steps mentioned above.

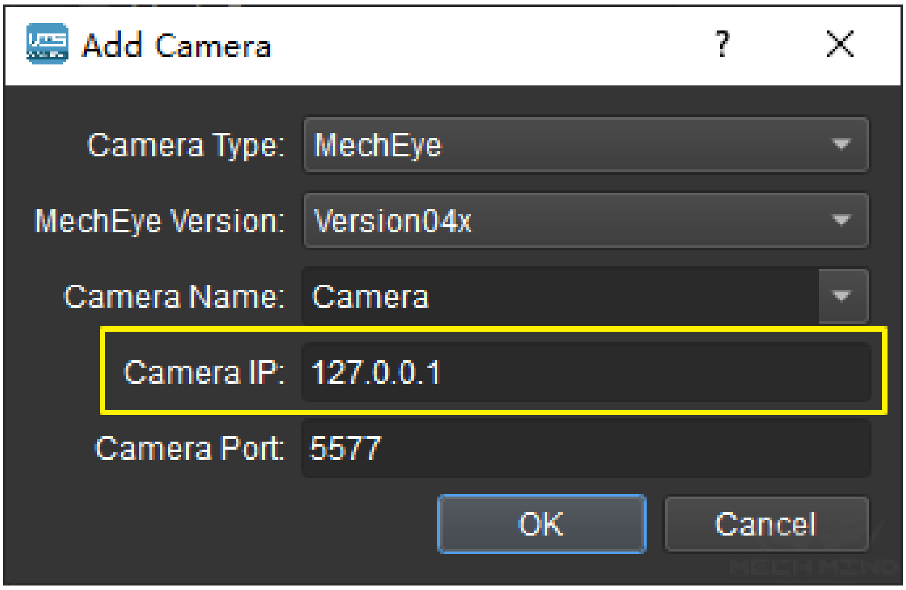


Figure 7 Manually Add Camera

Mount Calibration Board

Select the correct type of calibration board and put it in the camera's field of view. The operation interface is shown in *Figure 8* below.

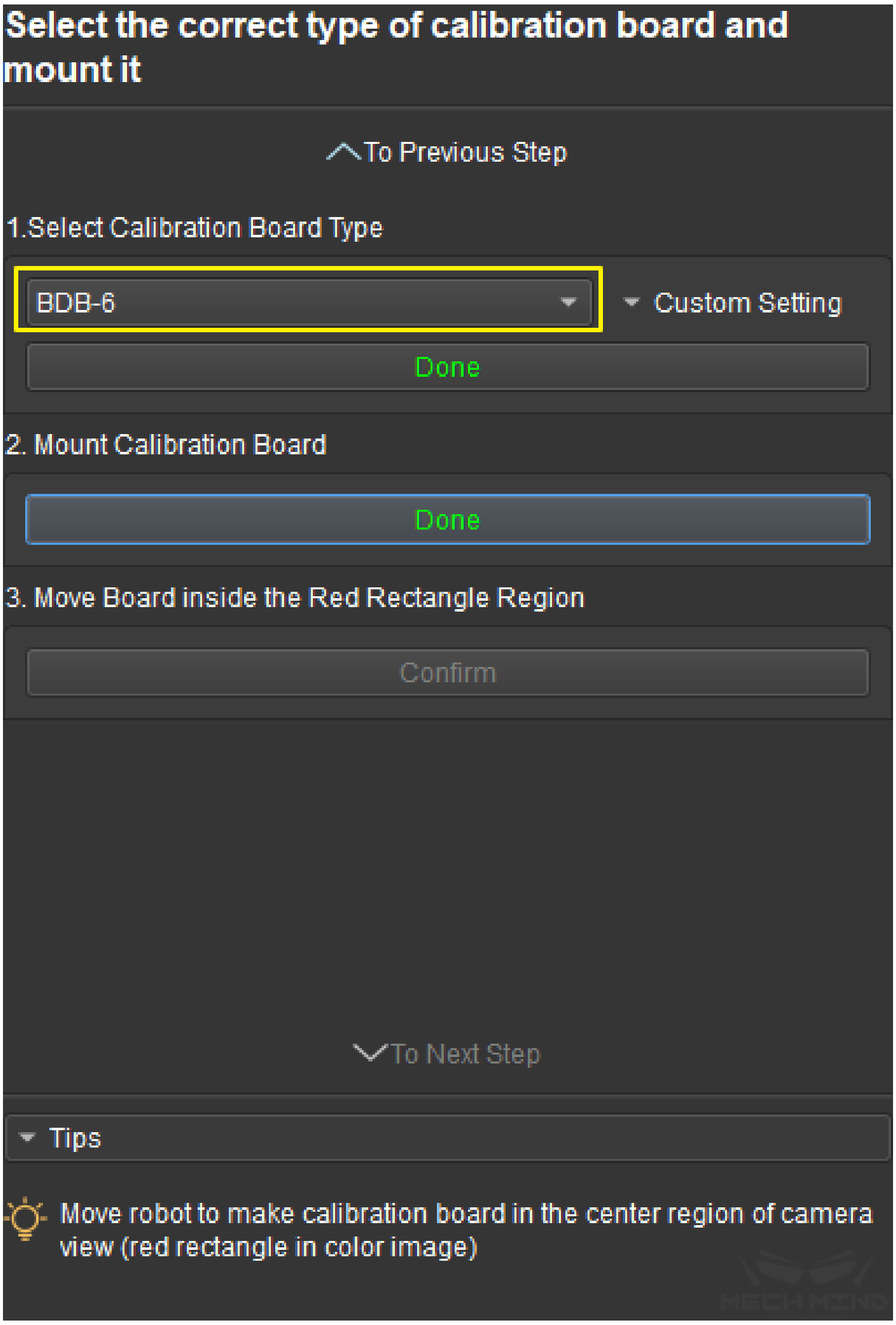


Figure 8 Mount Calibration Board

Specific steps are as follows:

1. Select the corresponding calibration board model (the name of the model is affixed on the top of the calibration board) and click the *Confirm* button.
2. Confirm that the camera is installed on the robot and the calibration board is within the camera' s field of view. Click the *Confirm* button when finished.
3. Make sure the calibration board is within the red rectangle in the 2D image, and click the *Confirm* button when finished.

Check Camera Intrinsic Parameters

- **Check Camera Intrinsic Parameters**

After mounting the calibration board, check if the camera parameters are correct in order to ensure that feature points can be detected during the calibration process. Click the *Check Camera Intrinsic Parameters* button, and the result of the intrinsic parameter checking will be displayed in a pop-up window. *Figure 9* shows a successful intrinsic parameter check.

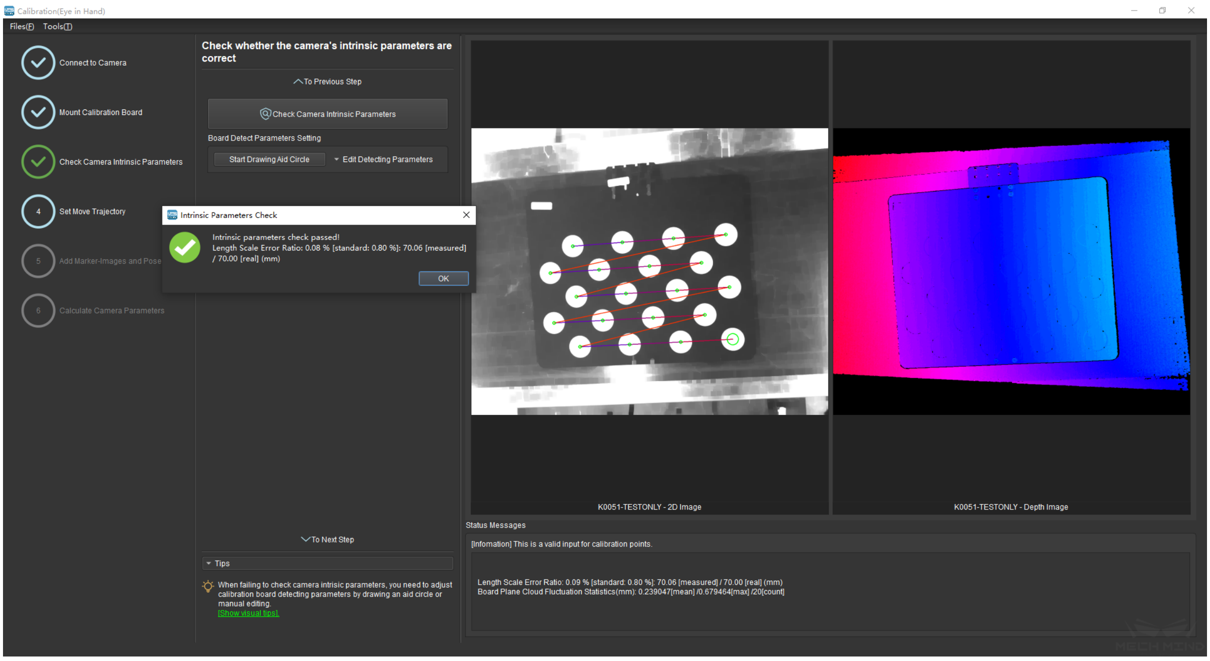


Figure 9 Check Camera Intrinsic Parameters

Generally, when the lighting conditions are good and the camera settings are appropriate, the intrinsic parameter check will pass by simply keeping the default values. *Figure 10* shows a successful detection of the feature points.

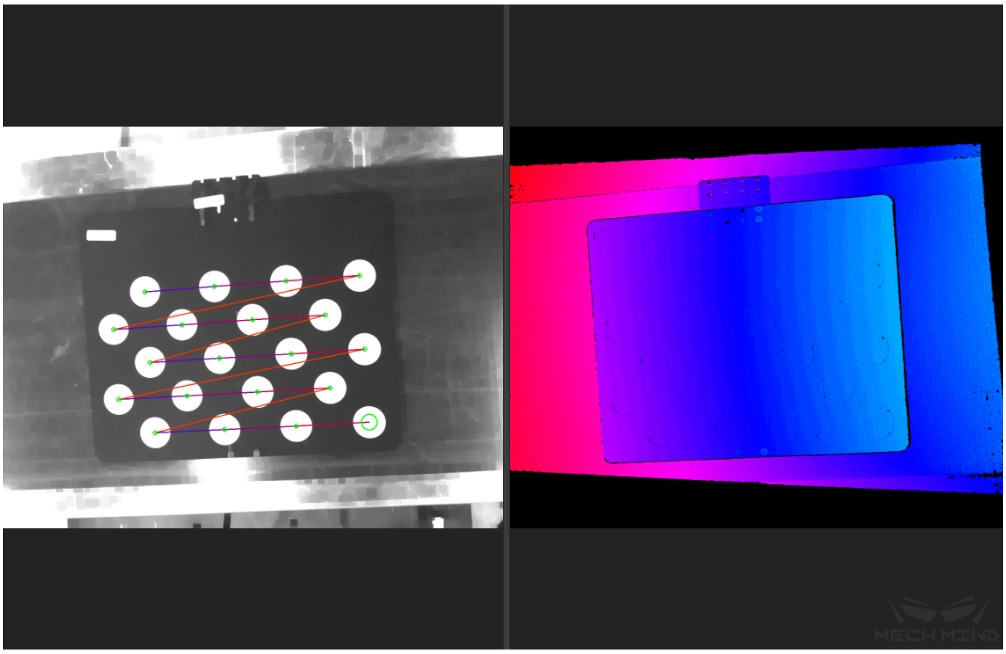


Figure 10 Detected Feature Points

- **Draw Aid Circle**

If the intrinsic parameter check fails, draw an aid circle or manually adjust the calibration board detection parameters. Select the *Start Drawing Aid Circle* button and draw a circle coinciding with a circle in the calibration board. After which, the detecting parameters of the circles should change.

If the user chooses to manually adjust the detecting parameters, simply click *Edit Detecting Parameters* and change the values accordingly.

After completing the above steps, click the *Check Camera Intrinsic Parameters* button again to get a new result. The steps are shown in *Figure 11* below.

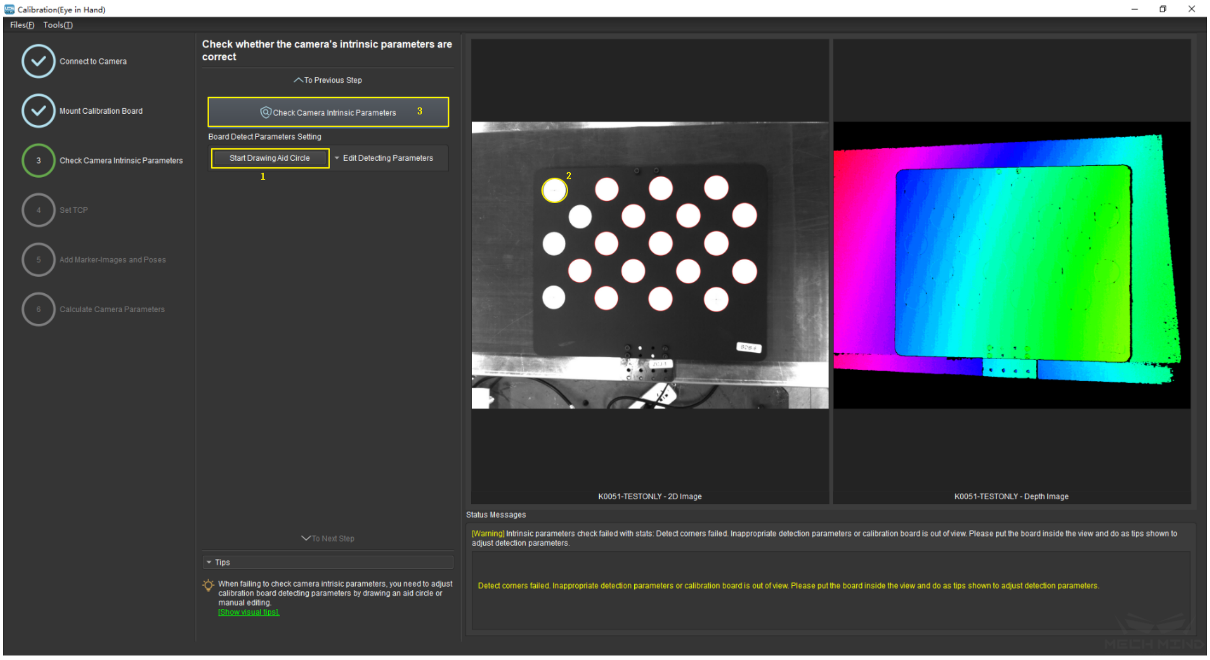


Figure 11 Draw the aid circle

Tip: If the calibration circles are too small to be selected easily, the user could right click the 2D image, uncheck *Fit to Window*, and select *Normal Size*. Start drawing aid circles after adjusting the size of the image.

If the feature points are still not detected, please adjust the camera settings according to the on-site operating conditions. Detailed information can be found in Mech-Eye Viewer.

Set TCP

- **Calibrate TCP**

In order to accurately measure the pose of the calibration board in the robot base coordinate, it is necessary to obtain the position of TCP relative to the origin of flange (default direction) first.

The Set TCP Value interface is shown in *Figure 12* below. If the TCP position is known, please check the *Known TCP Value* option and input the TCP value. If the TCP position is unknown, please check the *Unknown TCP Value* option and click the *Calibrate TCP* button.

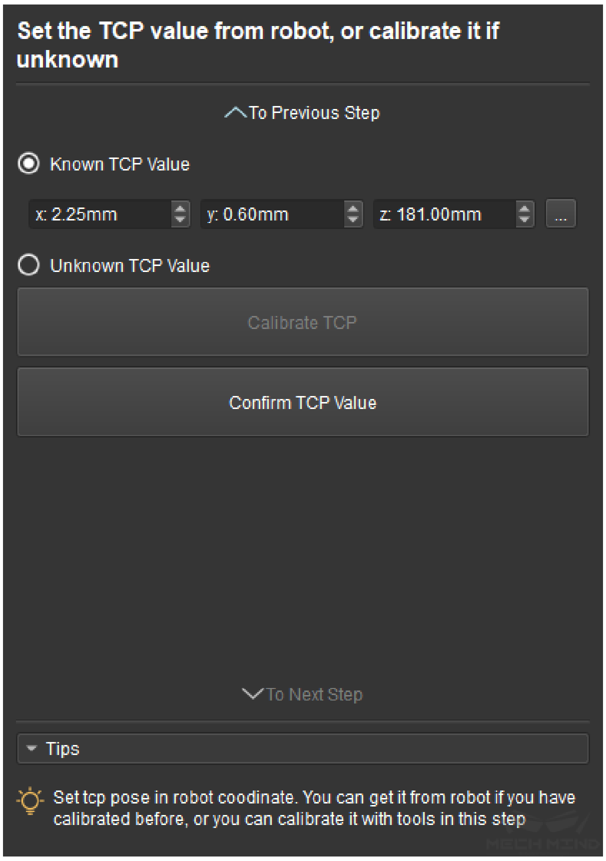


Figure 12 Set TCP

The steps are as follows:

1. Fix a sharp point in the robot work space. Let the robot touch this point from 4 different directions (namely 4 different poses). Click the *Add Pose* button every time the robot touches the point and the pose will be shown in the list.
2. *Delete Pose* will delete the pose which is not needed.
3. After adding 4 poses, click *Calculate TCP* to obtain the position of TCP. The error of calculation is shown in the pop-up window. As shown in *Figure 13* below.
4. After confirming that the TCP position is right, click *Confirm TCP Value* and *To Next Step* to finish the TCP calibration.

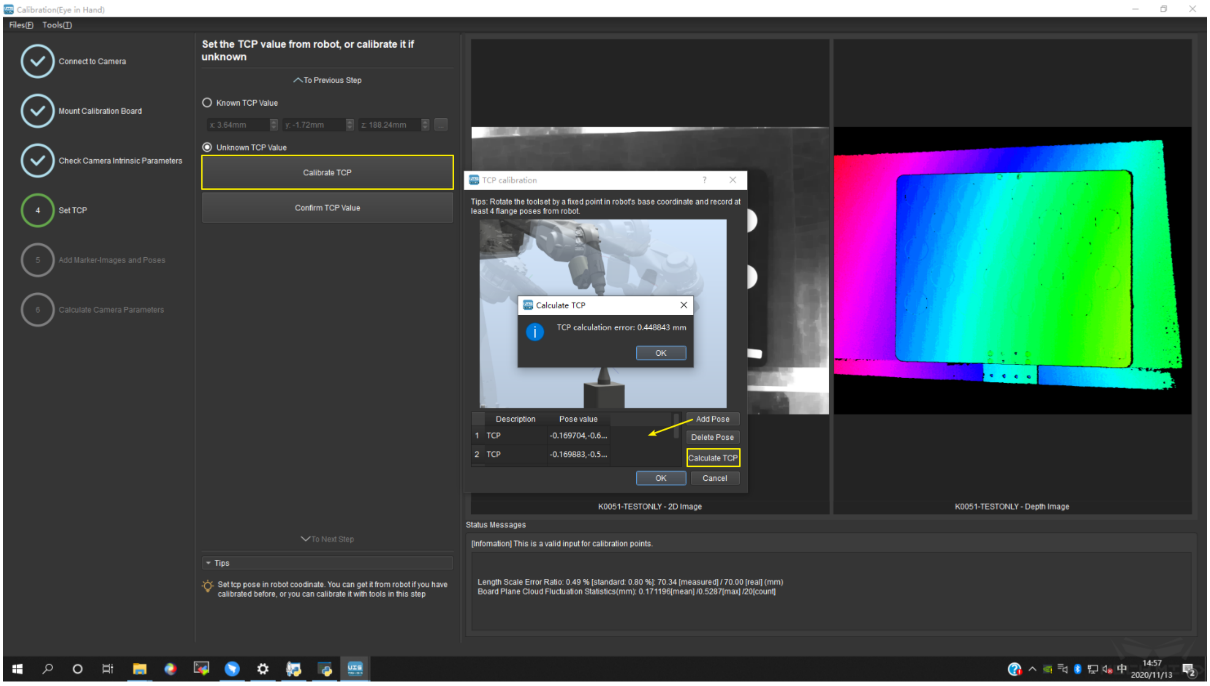


Figure 13 TCP Calculation Result

Tip: Some robots, such as ABB, cannot be controlled by the teach pendant after connecting with Mech-Center. The solution is to disconnect the robot from Mech-Center, move robot to touch the sharp point with the teach pendant, click *Add Pose*, and manually input the pose which is shown on the teach pendant.

Note: Make sure the TCP value set to 0 in the teaching pendant when checking *Known TCP Value*.

Add Marker-Images and Poses

Start calibration after the value of TCP is set.

Drive the robot to touch P1, P2 and P3 on the calibration board. Everytime the robot touches a point, click *Add Pose* to add the pose into the list. After touching all three points, the three poses will be shown in the Touch Poses Viewer. This is shown in *Figure 14*.

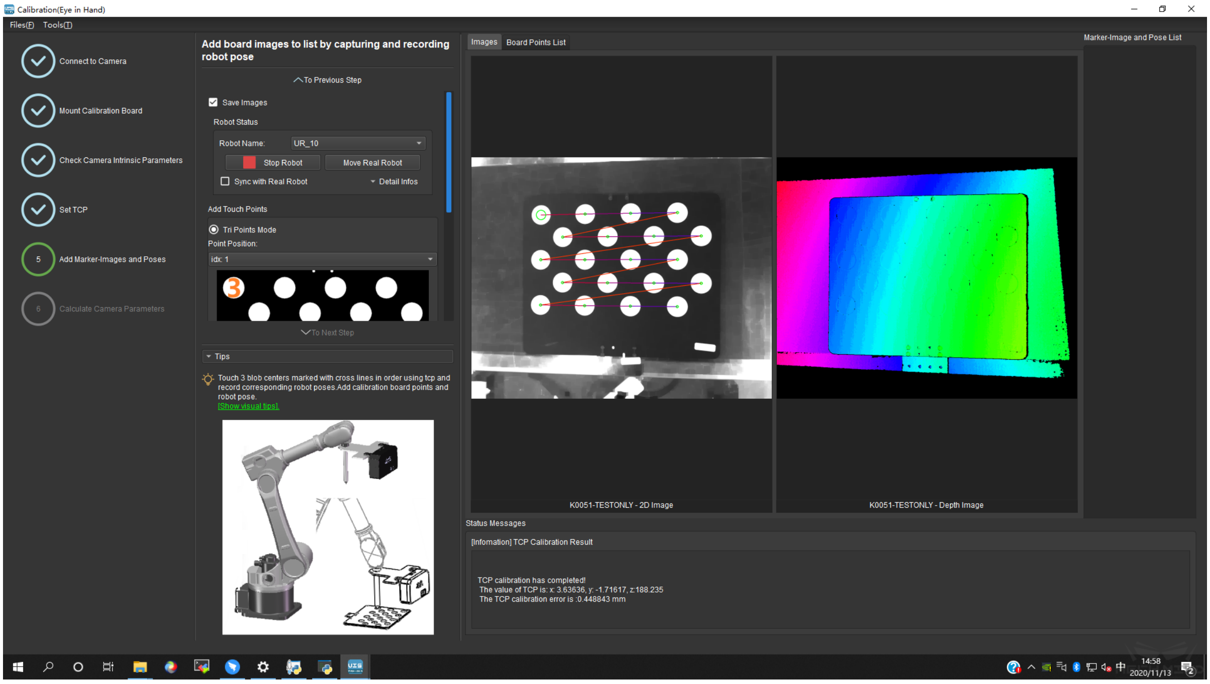


Figure 14 Add Three Poses

The real TCP touching for EIH is shown in *Figure 15* below.

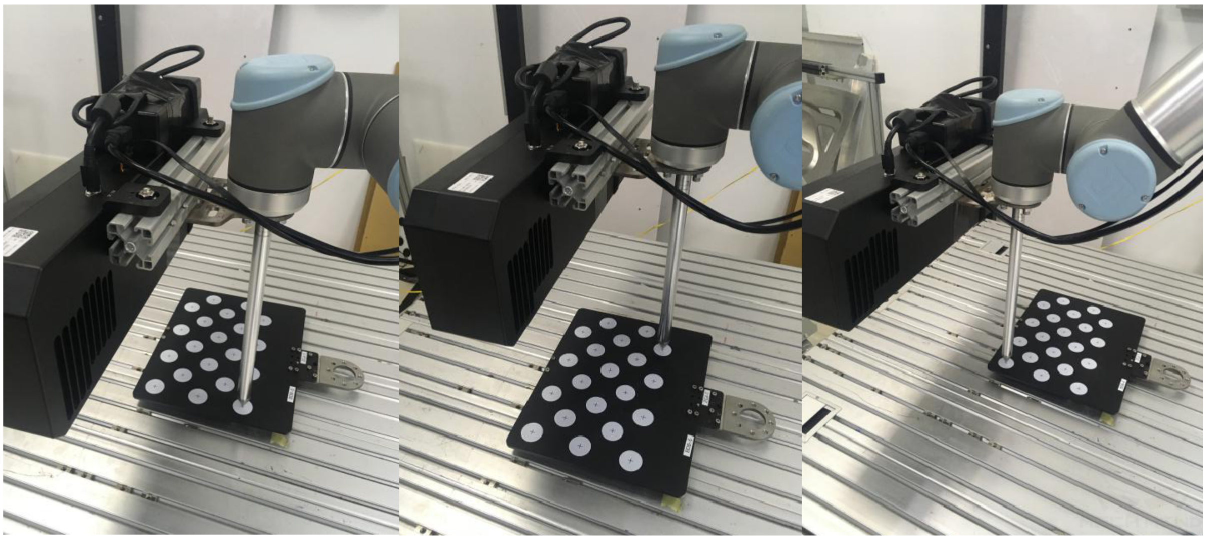


Figure 15 Real TCP Touch

Move the robot to a pose where the camera could capture the entire calibration board. Click the button *Add Single Board Image And Record Pose* to let the camera take a photo and detect the circles on the calibration board. Then click *To Next Step* to continue to the Calculate Camera Parameters page. The above steps are shown in *Figure 16* below.

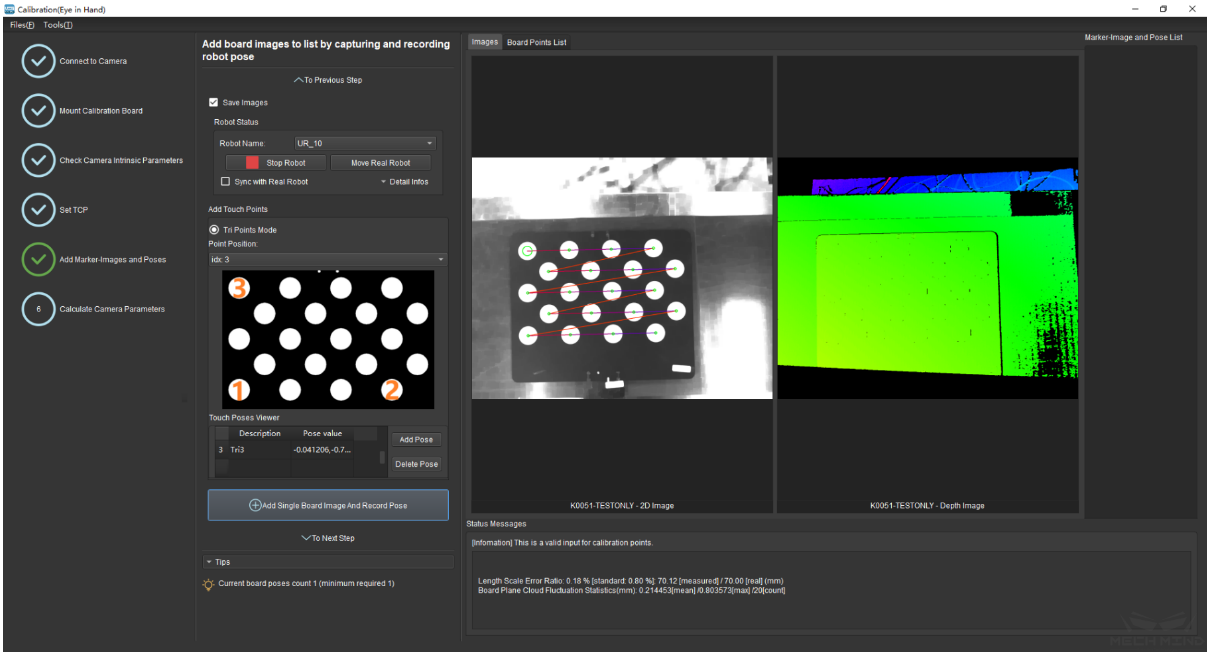


Figure 16 Add Calibration Point

Attention: Make sure the calibration board is stationary during the whole process.

Calculate Camera Parameters

- **Calculate Camera Parameters**

In the Calculate Camera Parameters interface, click the *Calculate Camera Parameters Result* button. The point cloud showing the calibration error and the calibration result will be displayed on the right part of the window, as shown in `Figure 17 <#tcp(EIH)_pic17>`__ below.

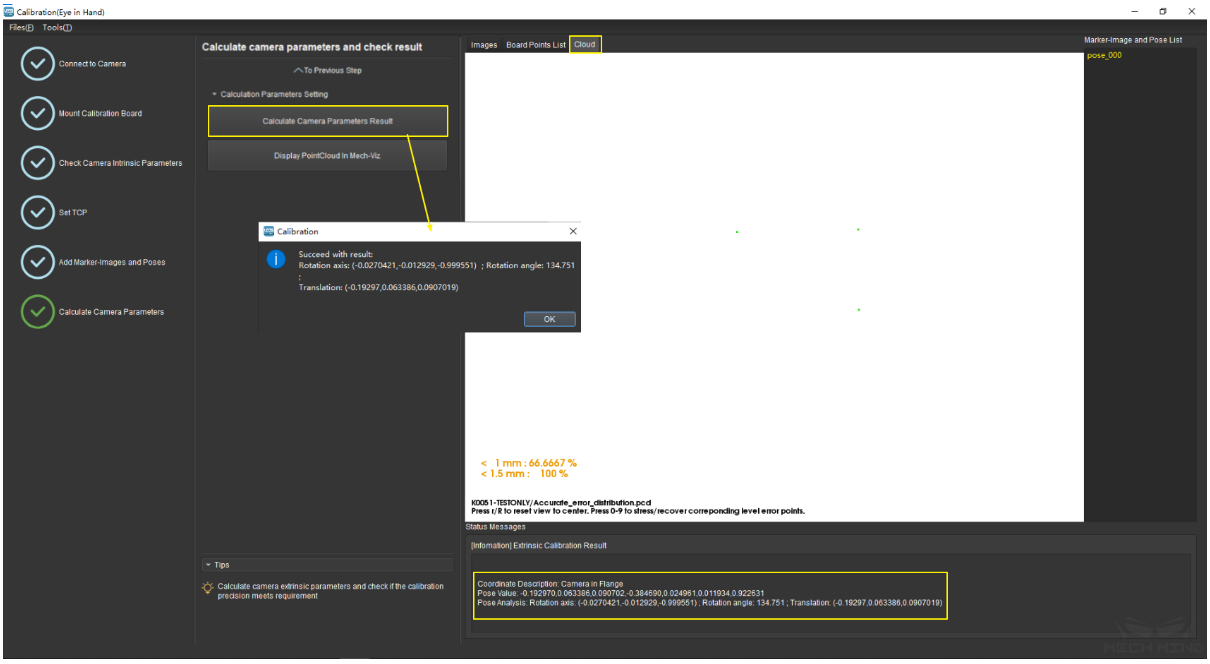


Figure 17 Calculate Camera Parameters

- **Display PointCloud in Mech-Viz**

As shown in *Figure 18*, click the button *Display PointCloud in Mech-Viz*.

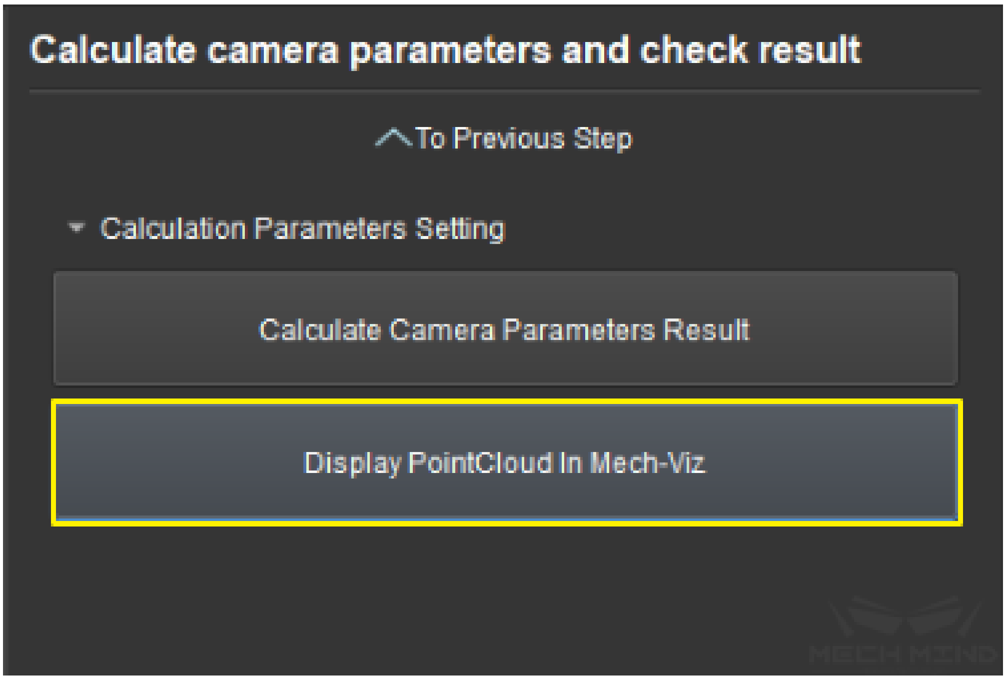


Figure 18 Display Point Cloud in Mech-Viz

Next, in Mech-Viz, select the *robot* tab. Click the *Configure Robot* button, and in the pop-up that appears, choose the correct robot type. After which, click the *Sync Robot* button. The above steps are shown in *Figure 19* below.

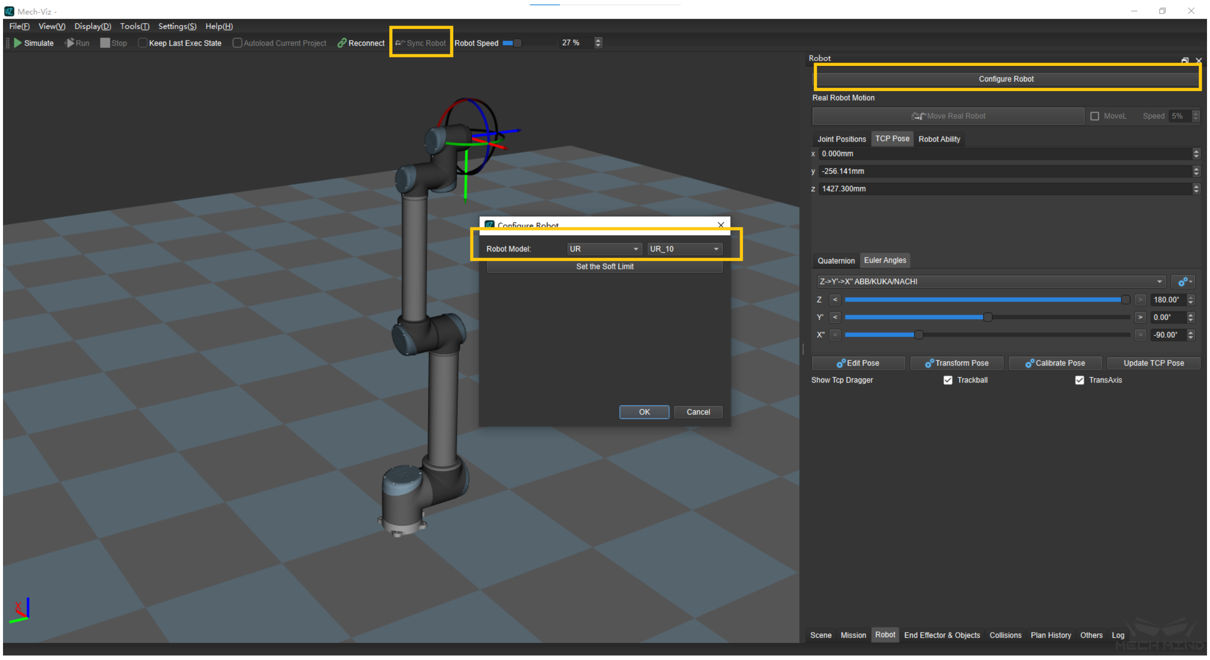


Figure 19 Synchronize Robot in Mech-Viz

Select the *scene* tab and adjust the ground height, as shown in *Figure 20*.

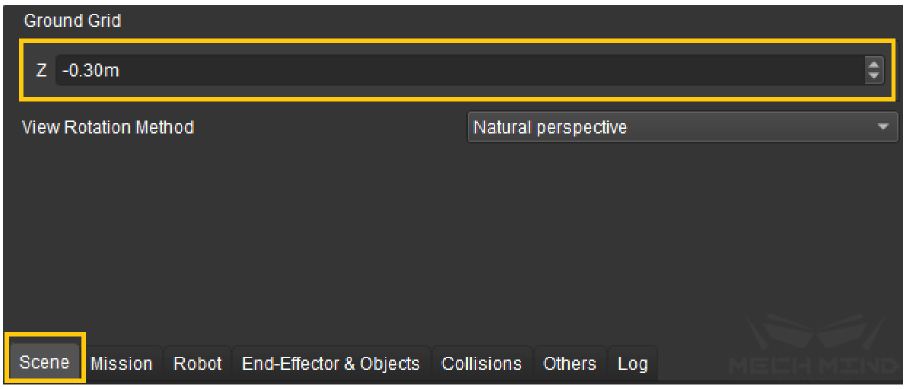


Figure 20 Adjust the the ground height in Mech-Viz software

Finally, the point cloud of the calibration board will be displayed as shown in *Figure 21*.

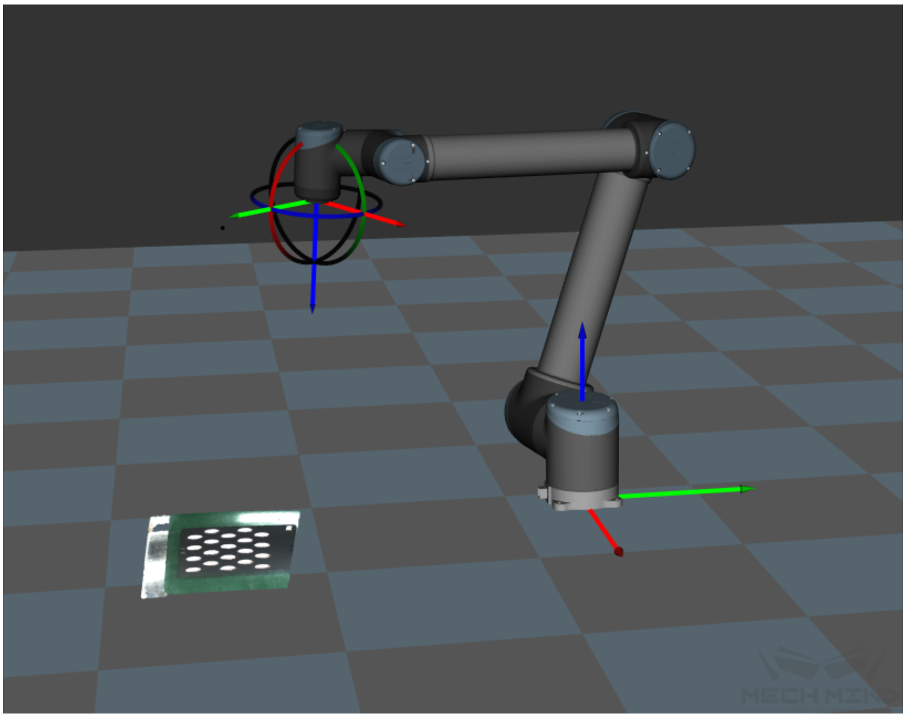


Figure 21 Point cloud of calibration board in Mech-Viz software

6.2.3 Eye To Eye

Double cameras calibration is to calibrate the pose relationship between the two cameras. Currently, ETE is used to achieve the calibration of the double cameras pose relationship. Using double cameras calibration can expand the camera field of view and improve the point cloud’s quality of the two cameras’s overlapping part. As shown in the figure *1*.

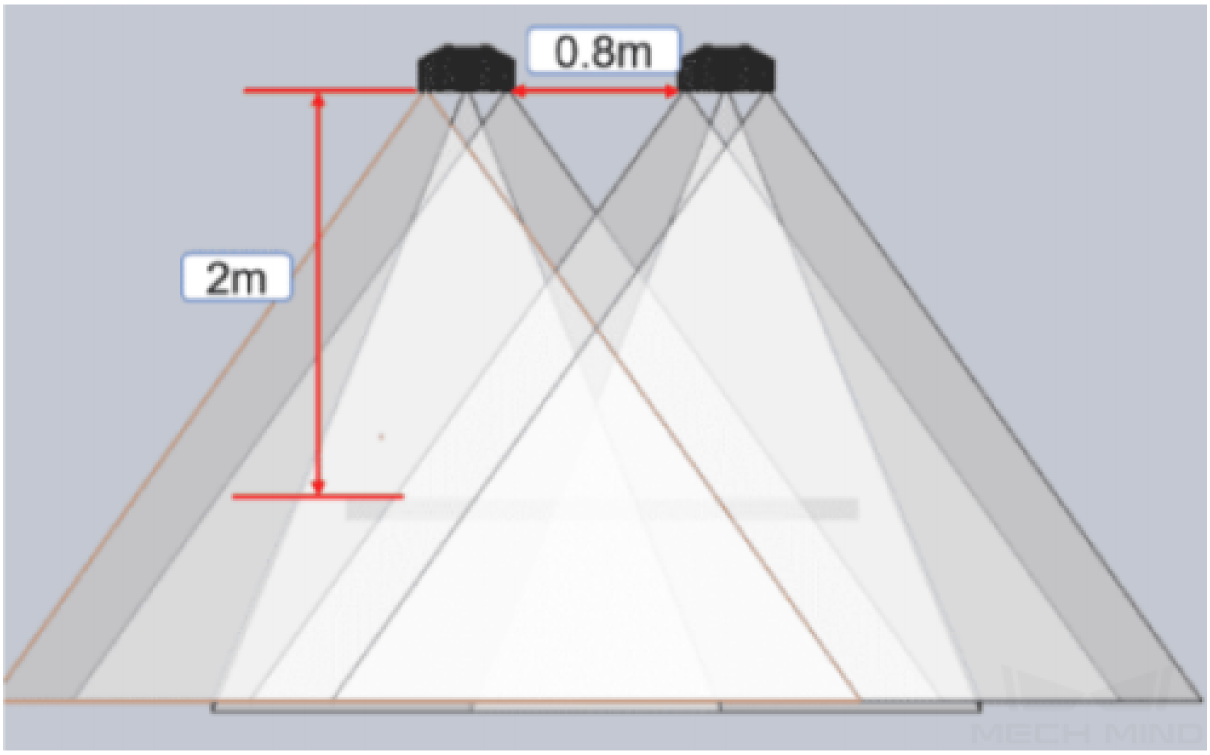


Figure1 Double cameras Field of View

Double Cameras Calibration Method

Multiple random board poses

Calibration premise: The two cameras used have the same resolution. Camera field of view (2D&3D) overlapping area covers the entire working area.

Use Mech-Viz or the control handle to move the robot to place the calibration plate in the field's center of the two cameras's view. At this time, the cameras can be connected to collect images to check the position of the calibration board in the field of view. From *Camera Assistant* → *Camera Calibration* into the calibration pre-configuration. Add calibration point method select "Multiple random calibration board poses", calibration setup select "EyeToEye". As shown in the figure2.

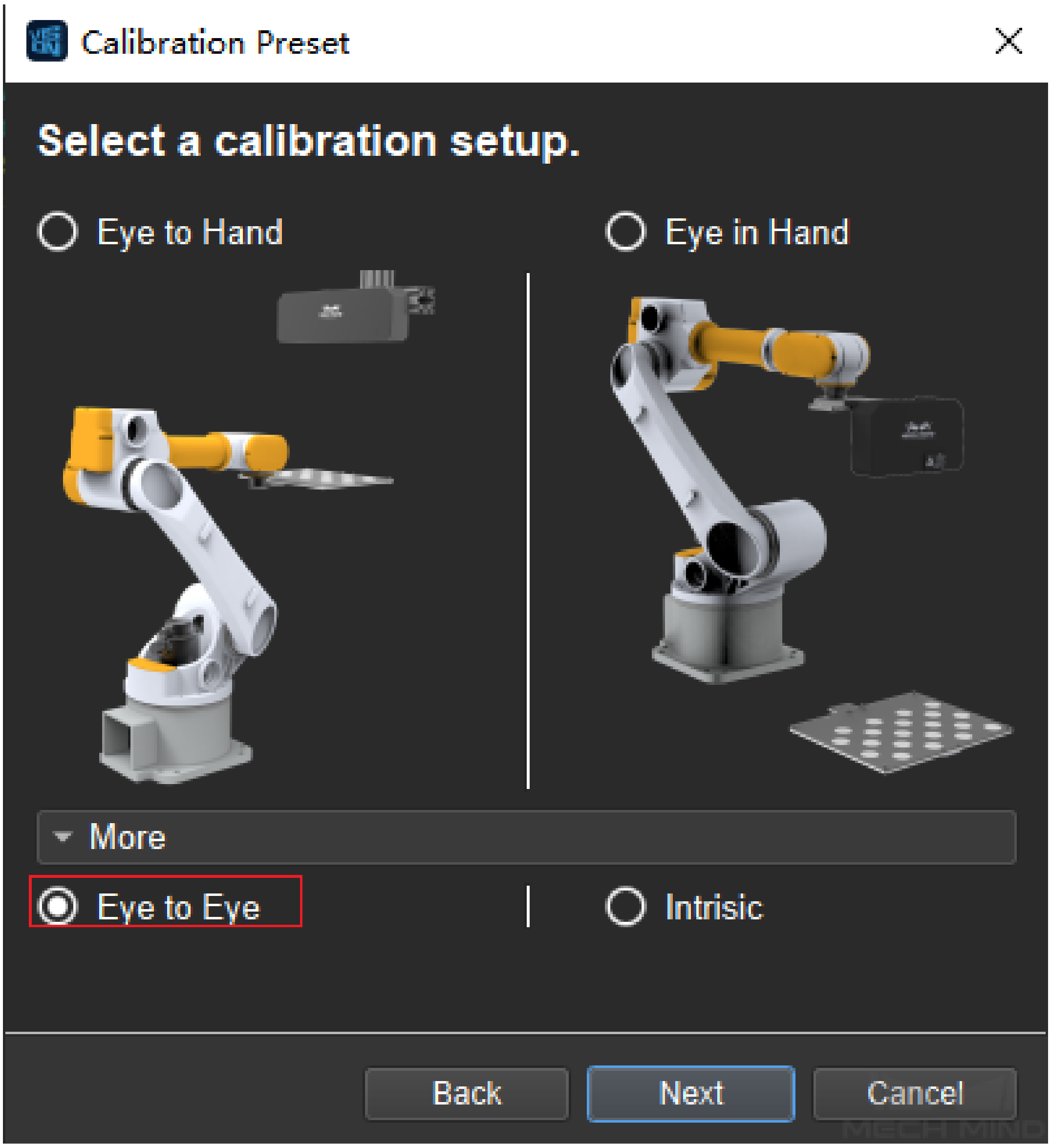


Figure2 Select the Calibration Installation Method

After entering the calibration interface, connect two cameras and choose one of them as the main camera. as shown in the figure 3.

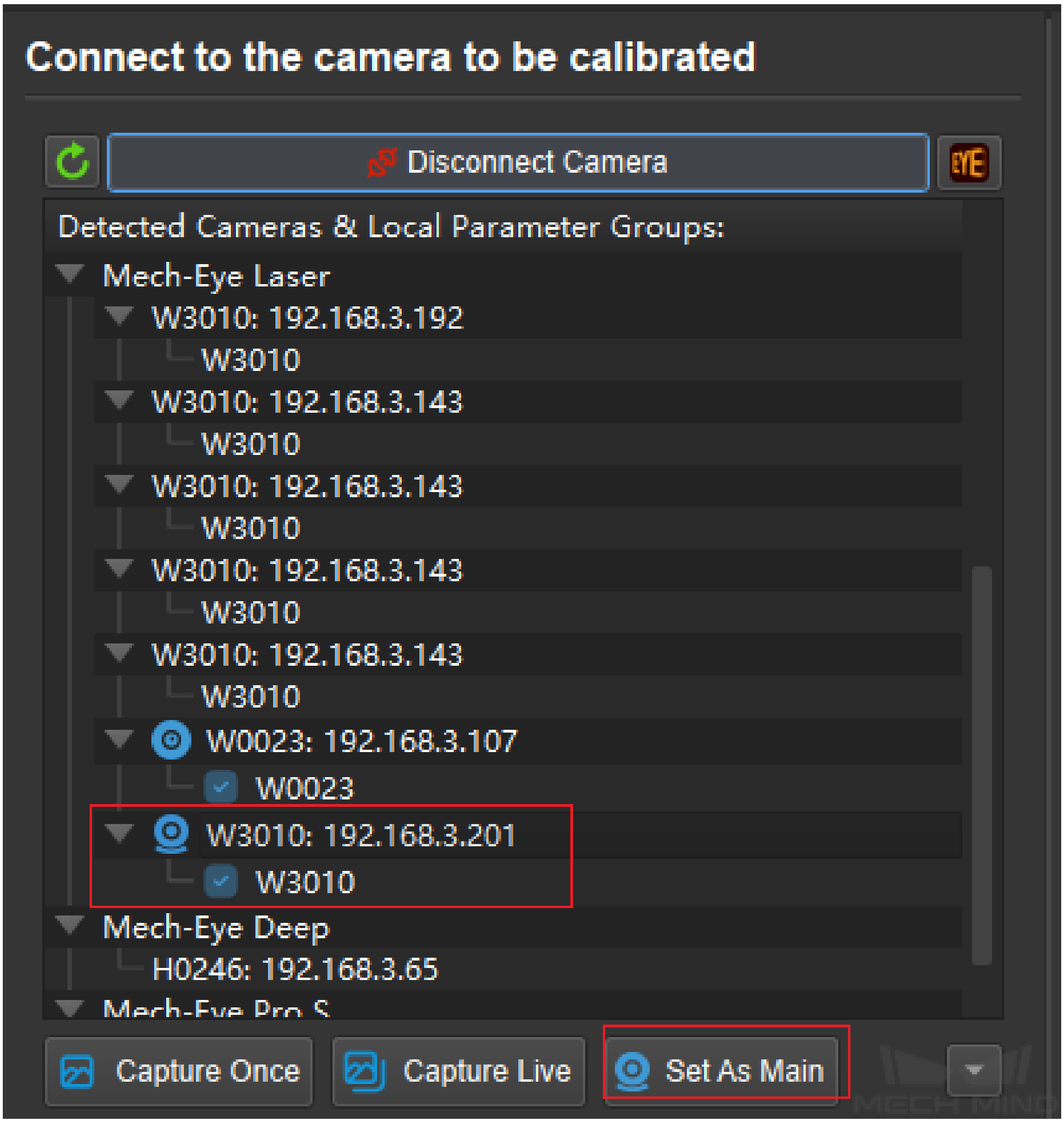


Figure3 Set the main camera

Then calibrate according to the ETH "Multiple random calibration board poses" calibration method. The data displayed on the interface is all the data of the main camera, and the data of the sub-camera will be generated synchronously. After the calibration, the extrinsic parameter data of the two cameras will be generated at the same time. At this time, the ETE calibration is completed.

Calculate from the calibrated extrinsic parameters directly

After calibrating the extrinsic parameters of the two cameras separately by ETH, use ETE to calculate the pose relationship of the two cameras. Open the calibration interface, select ETE for the calibration installation method, enter the calibration interface, In the "Choose the method of adding calibration points", select "Use the calibrated extrinsic parameters to calculate directly", and select the extrinsic parameter data of the last calibration, as shown in the figure 4.

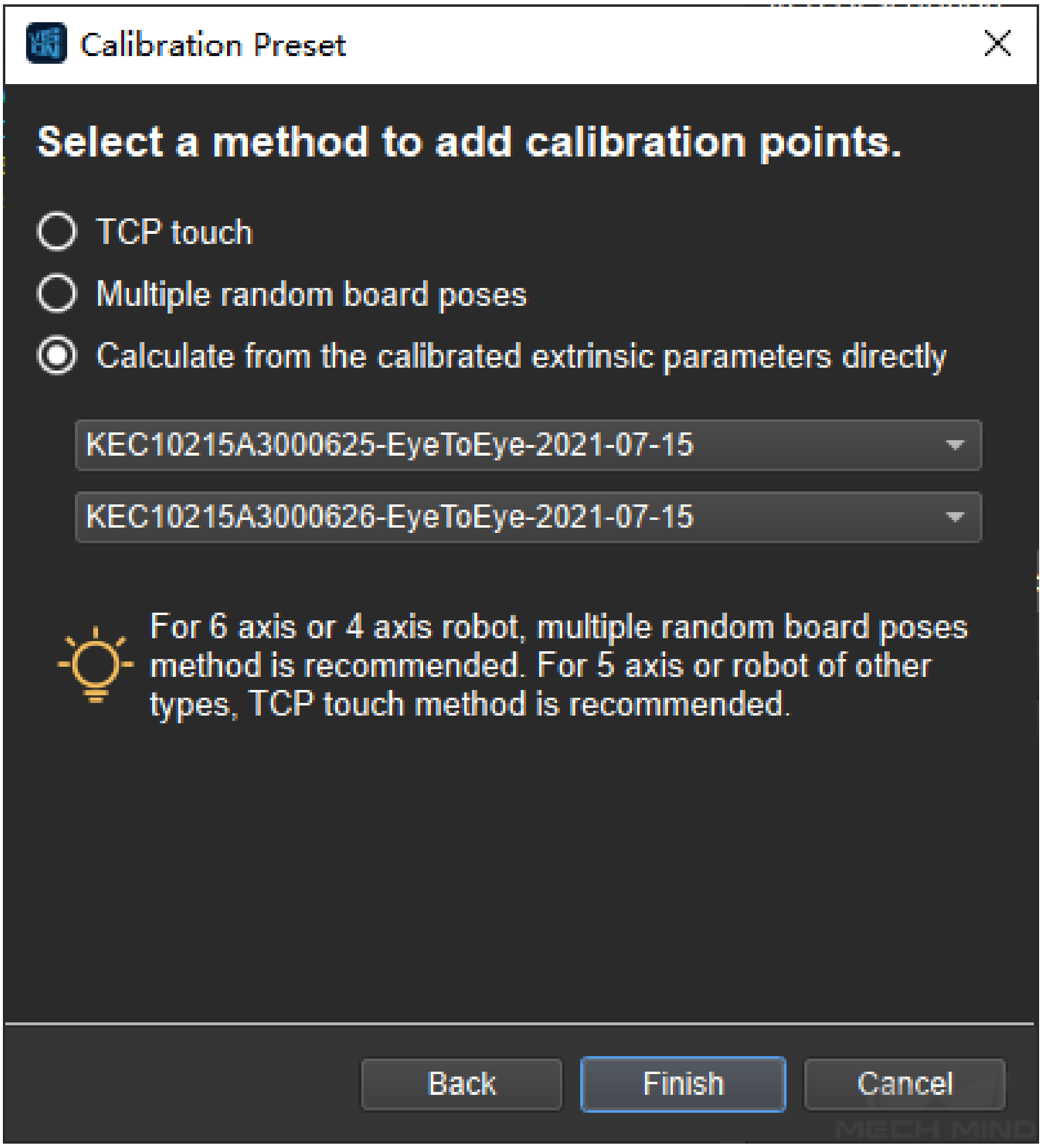


Figure4 Select the Method of Adding Calibration Points

After the setting is completed, enter the calibration interface and no need to connect the camera. Directly enter the "calculate camera parameters". Click *Calculate Camera Parameters Result* to get the pose relationship of the two cameras. Click *Display PointCloud in Mech-Viz* to generate a fused point cloud. If the point cloud fusion effect is not good, the extrinsic parameters of the two cameras need to be re-calibrated. The point cloud is shown in the figure 5.

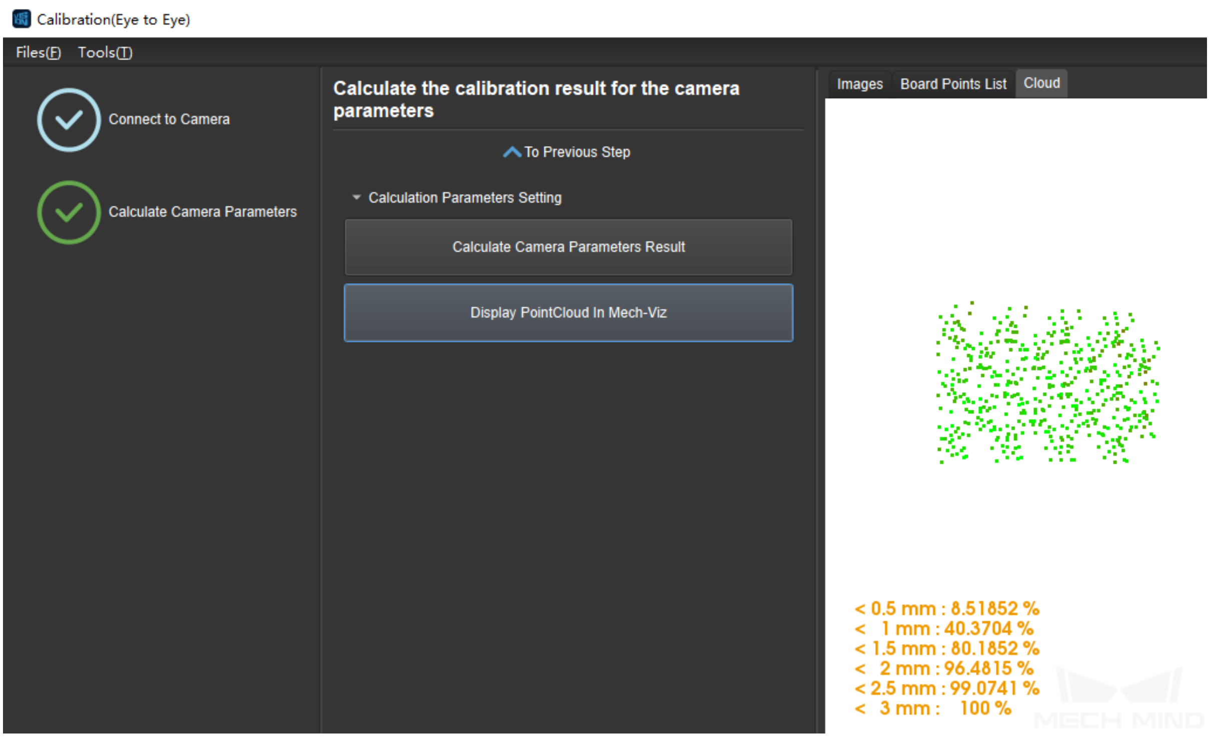


Figure5 View PointCloud

Check the fusion effect of double cameras calibration by building a project in Mech-Vision

The construction of the project is shown in the figure *6*. Pay attention to check the "Trigger Control Flow When No Output" and "Trigger Control Flow Output".

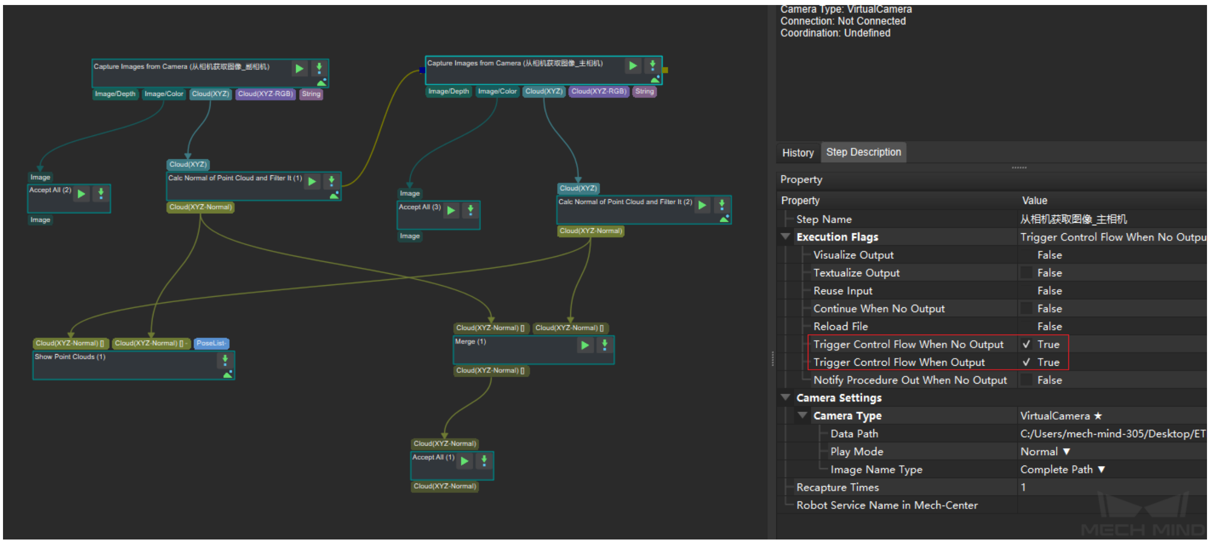


Figure6 Project Construction

Run the "Merge Data" Step to display the merged point cloud. The point cloud output after merging is the fused whole point cloud, as shown in the figure below *7*, and you can click the upper left corner *View as Whole| 1 | 2* to switch points cloud.

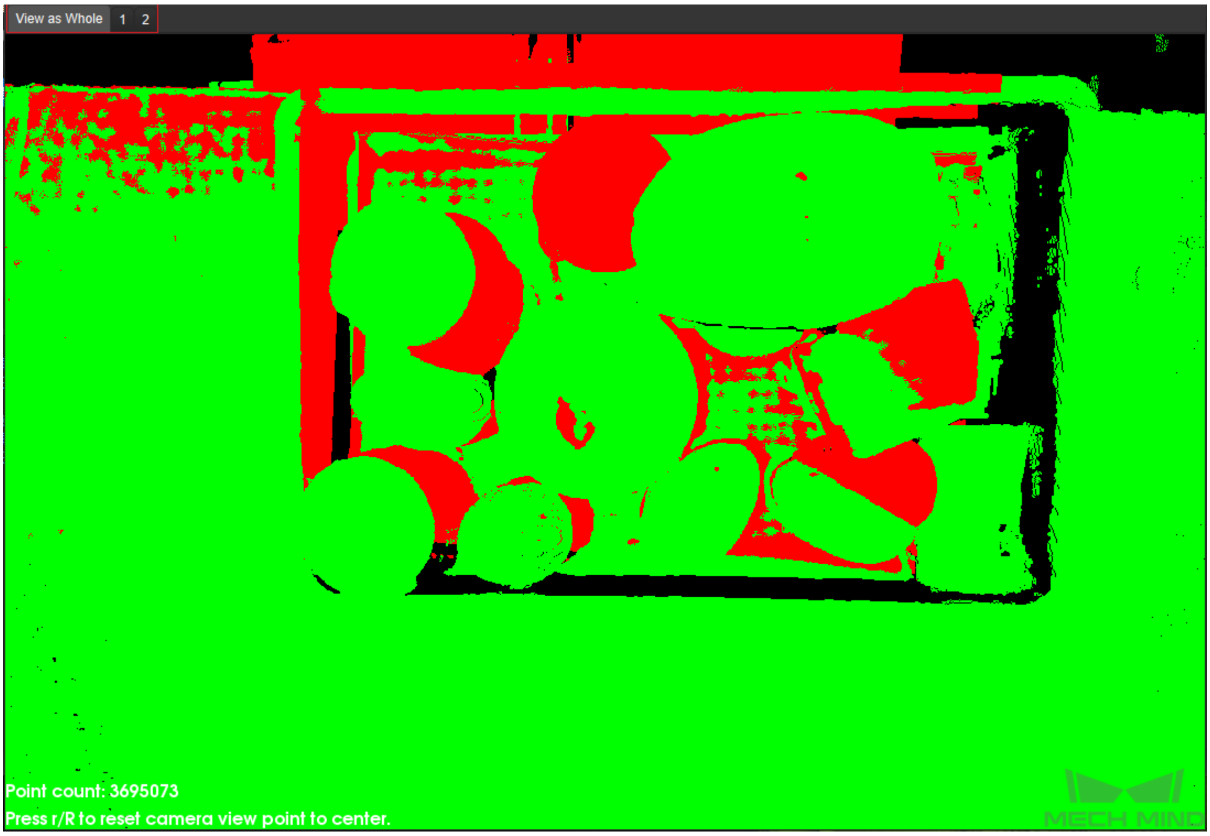


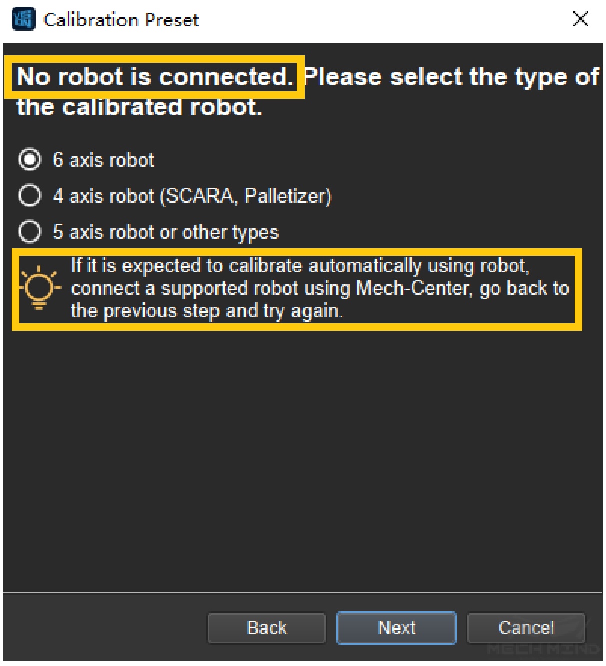
Figure7 Fusion PointCloud

For cases where the robot cannot be controlled by Mech-Viz, you can add poses manually to perform the hand-eye calibration.

6.2.4 Manually Add Calibration Points

This chapter will introduce the method to add calibration points manually in situations where the robot cannot be controlled by Mech-Viz.

First, select the calibration type and robot model according to the real robot in the preset window, as shown in / *Figure 1*/ .



Select Robot Model

Multiple Random Board Poses

If multiple random board poses is chosen to add calibration points, please follow the process below:

Set Euler-Angle Type

The euler angle type need to be set in step 4.

If the type of Euler angle is known, please check *Known Euler-Angle Type* and choose the corresponding type.

If the Euler angle type is unknown, please do as the following steps, which are also shown in *Figure 2*:

1. Click *Unknown Euler-Angle Type* and then *Get Euler-Angle Type*.
2. Install a fixed sharp point in the robot work area, then rotate the tool around this point and record three different robot poses.
3. Click *Get Euler-Angle Type* , the Euler angle type will be calculated.

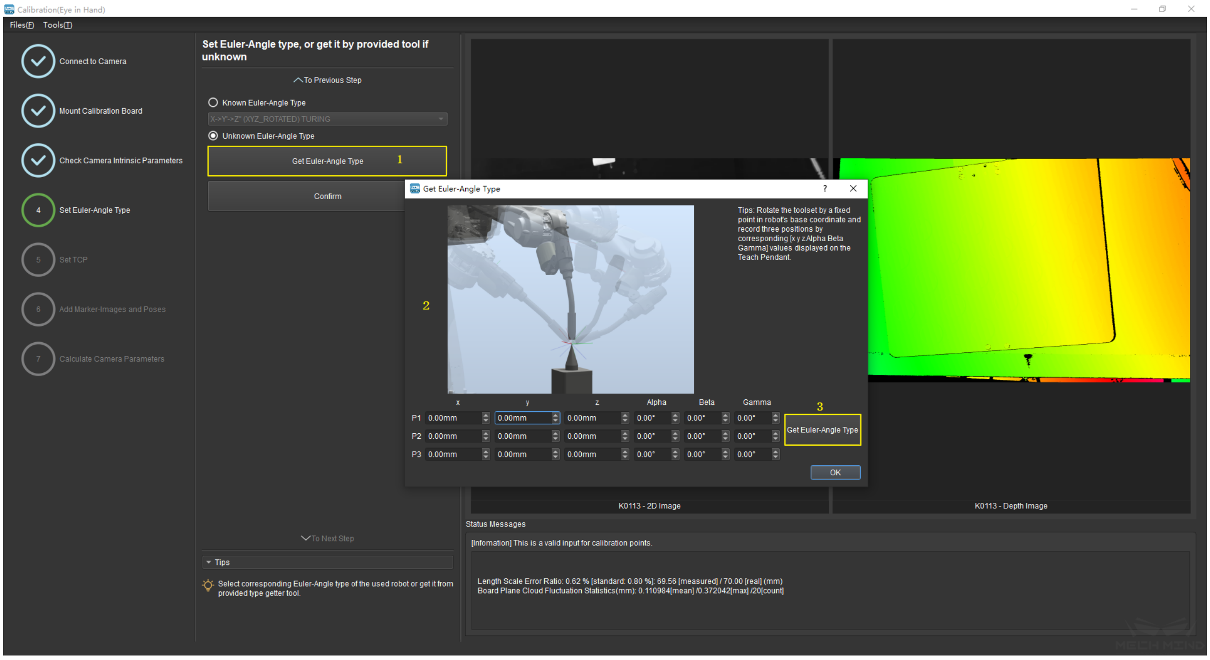
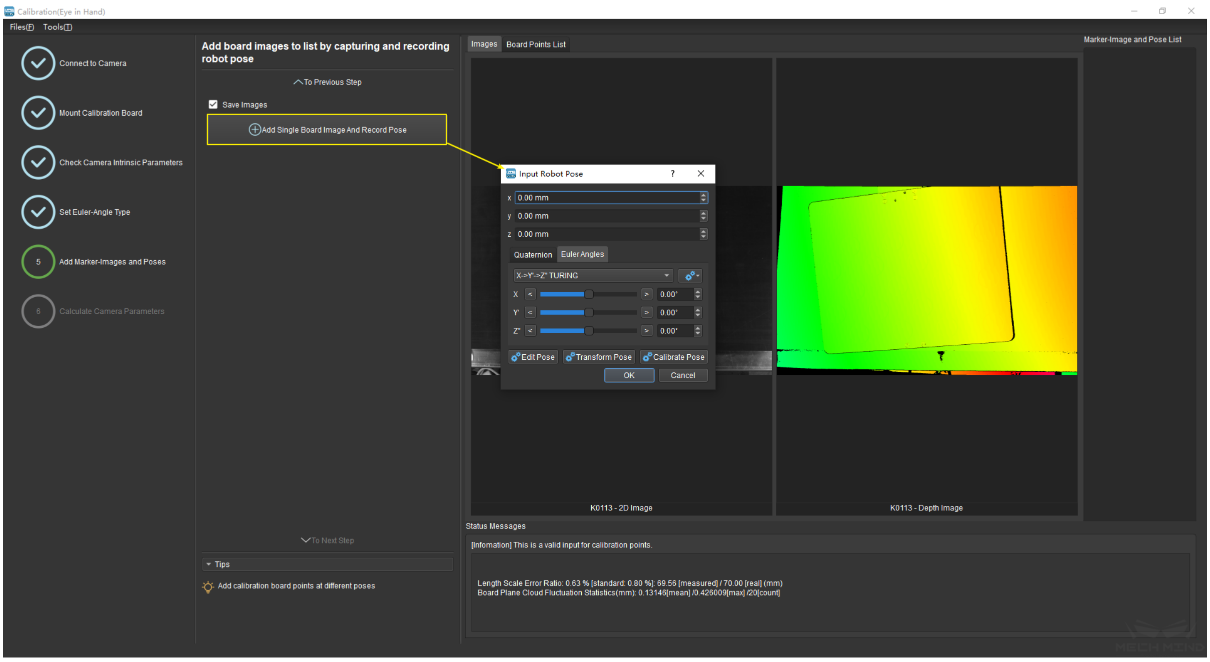


Figure 2 Get Euler-Angle type

Add Marker-Images and Poses

In step 5, everytime an image of the calibration board is added, the user needs to input the current robot pose manually, as shown in *Figure 3*.



Add Poses Manually

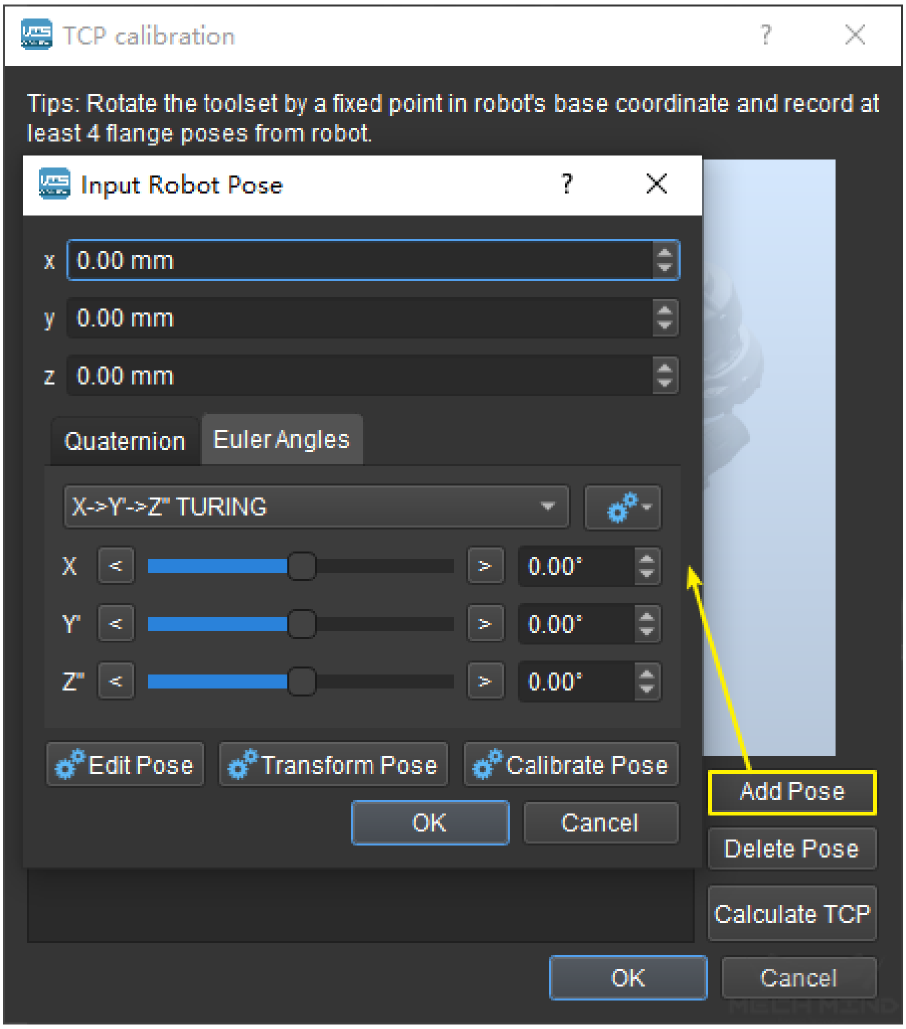
TCP Touch

If TCP touch is used to add calibration points, it is also necessary to set the Euler angle. Please refers to *Set Euler-Angle Type* for the operation.

Set TCP

If the position of TCP relative to the origin of the flange is known, check *Known TCP Value* and input the X, Y, Z value.

If the position is unknown, check *Unknown TCP Value* and click the *Calibrate TCP* button. Touch the fixed point with the sharp point at the end of the robot from at least four different poses. Click *Add Pose* everytime the robot touches the fixed point, and in the pop-up that appears, enter the current **flange pose**, as shown in *Figure 4*. After four sets of poses are recorded, click *Calculate TCP*.

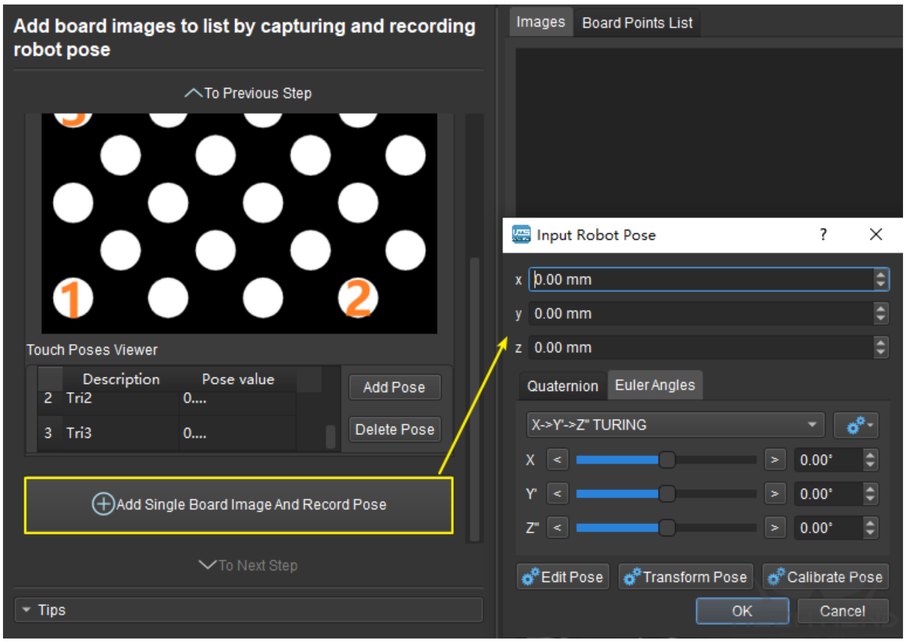


Calibrate TCP

> **Attention:** Make sure the TCP value is set to 0 in the teach pendant when checking the *Known TCP Value* option.

Add Marker-Images and Poses

Drive the robot to touch P1, P2 and P3 on the calibration board. Everytime the robot touches a point, click *Add Pose* and input the current robot pose in the pop-up, as shown in *Figure 5*.



Add Three Poses

Move the robot to a pose where the camera could capture the entire calibration board. Click the button *Add Single Board Image And Record Pose* to let the camera take a photo and detect the circles on the calibration board. If the calibration setup is EIH, it is necessary to input the current robot pose in the pop-up window.

Then, click *To Next Step* to go to the Calculate Camera Parameters page.

> **Attention:** Make sure the calibration board is stationary during the whole process.

6.3 Quick Calibration Mode

In some project sites, it is often necessary to re-calibrate the camera after replacing the camera. For projects with restrictions on time spent on camera replacement, workstation space, or fixture mounting, quick calibration mode can be used to quickly and easily replace the camera with a camera of the same model.

6.3.1 Preparations Before Replacing the Camera

Before using the quick calibration mode to calibrate the camera, preparations listed below should be made:

- The old camera and the camera parameter group that has been calibrated in the standard extrinsic parameter calibration mode.
- Choose a suitable calibration board model for the camera, install and fix the calibration board so that the calibration board is located somewhere convenient within the reach of the camera's field of view, not affecting the normal running of the project (the calibration board cannot be moved or should not be damaged).
- If in Eye In Hand mode, move the robot to a proper pose, record the pose through the Mech-Viz project or a third-party master control program, and ensure that the "pose during data collection by photo-taking" is consistent with the "pose during secondary calibration by photo-taking".

Record Feature Points

Click :menuselection: *Camera –> Camera Calibration –> Quick* to enter the quick calibration mode. If the Mech-Vision project has been opened in Mech-Vision, the recorded feature point data will be saved under the project path; if no project is opened, please select the project path. If the project has no recorded feature points, the interface shown in *Figure 1* will appear.

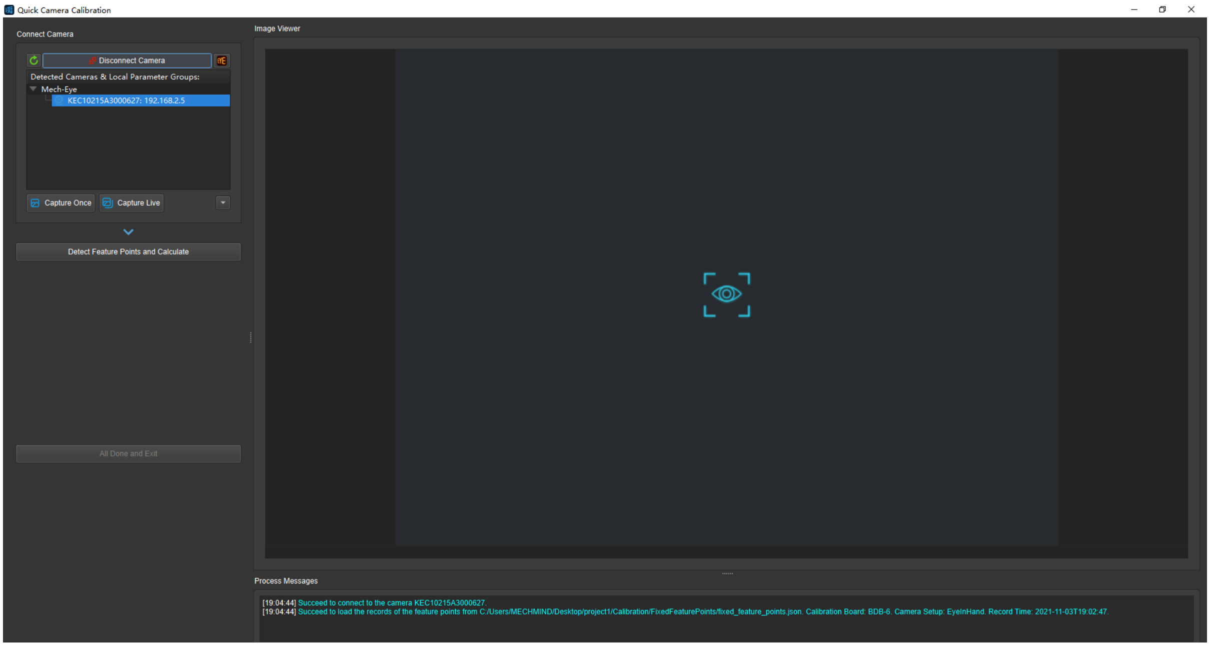


Figure 1. Window of Recording Feature Points

If the project has recorded feature points, the window as shown in *Figure 2* will appear. If the extrinsic parameter group is updated, the position of the calibration board is moved or damaged, and the pose of the robot changes before the quick calibration of the new camera, please select *Re-record the fixed feature points*.

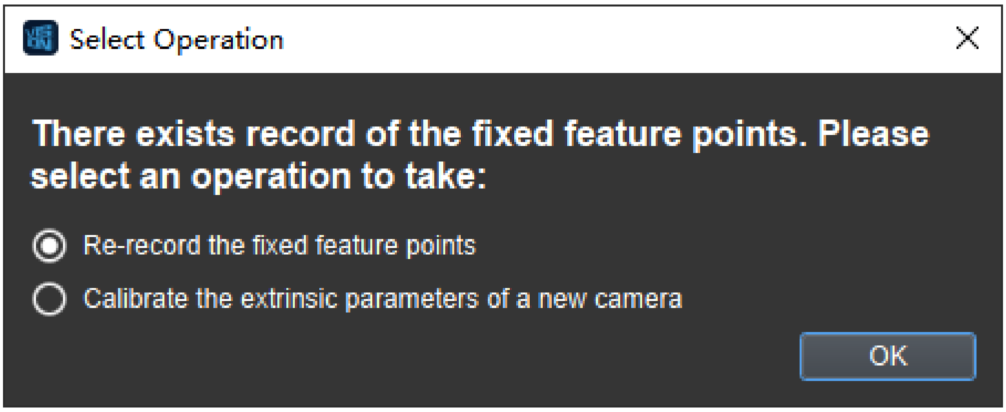


Figure 2. Re-Record Feature Points or Start a New Camera Calibration

After entering the record feature point page, connect the camera and select the corresponding extrinsic parameter group. When the camera has only one parameter group, there is no need to manually select the parameter group.

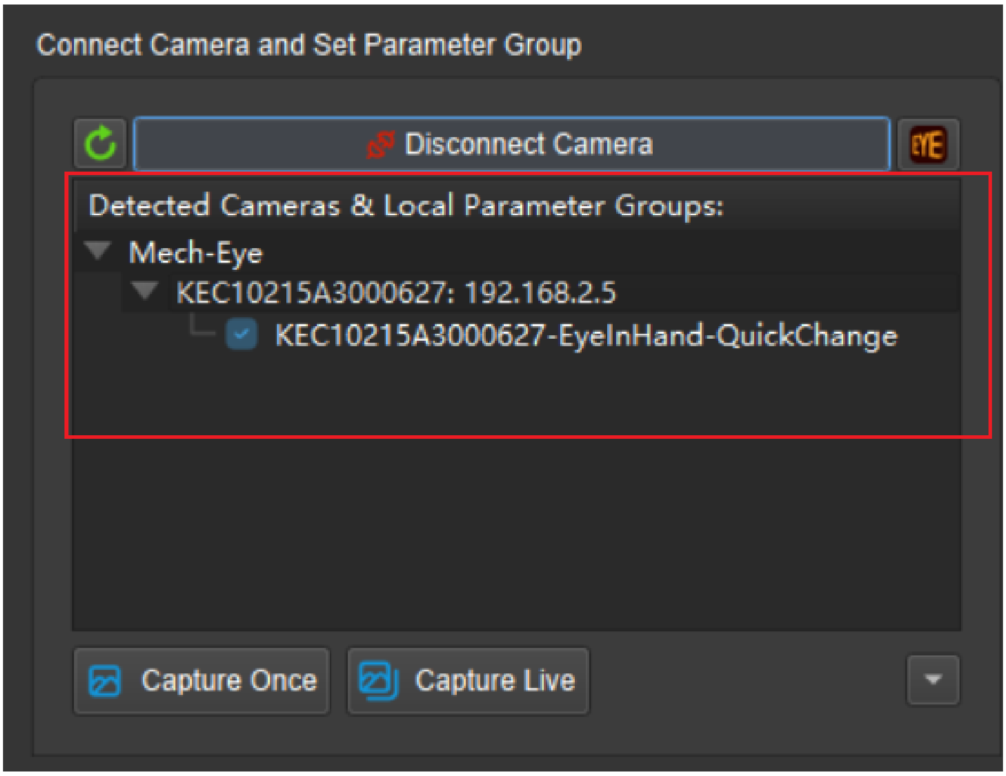


Figure 3. Connect the Camera and Select the Extrinsic Parameter Group

Then select the calibration board type to use. After done setting, click *Detect Feature Points and Calculate* to start recording feature points. If the *Finish and Exit* button is on, the feature points have been recorded. Click *Finish and Exit*.

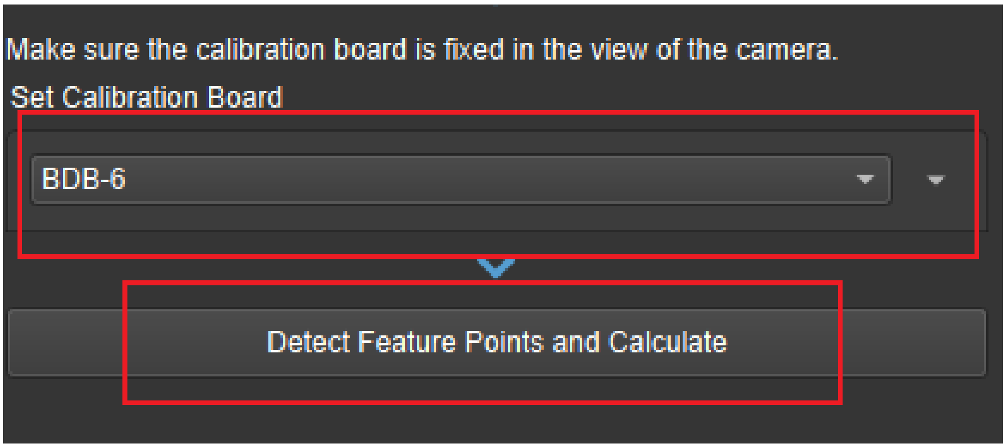


Figure 4. Set Calibration Board Type and Calculate Feature Points

After recording, a feature point folder is generated in the project folder, and the project can be quickly calibrated directly after replacing the new camera.

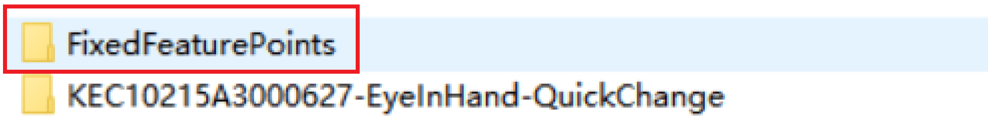


Figure 5. Folder Containing Feature Points

6.3.2 New Camera Quick Calibration

Enter the quick calibration mode, and select *Calibrate the extrinsic parameters of a new camera* in the window shown in *Figure 6*.

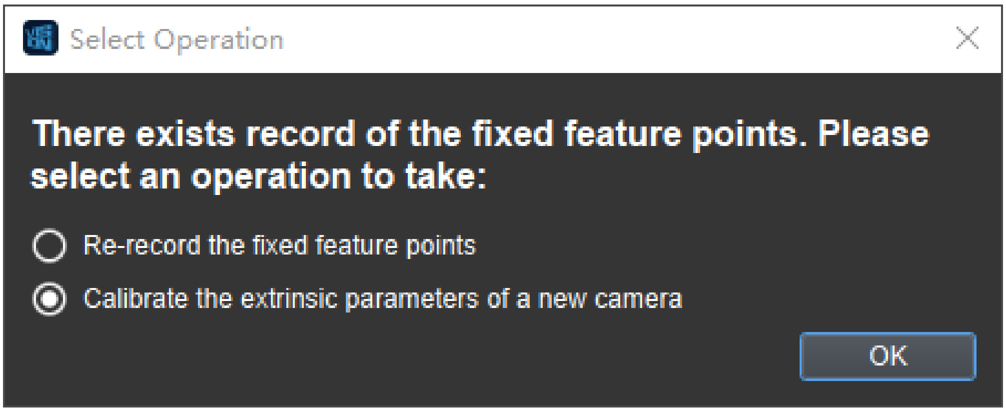


Figure 6. Choose to Re-Record Feature Points or Start a New Camera Calibration

Select *Calibrate the extrinsic parameters of a new camera* and enter the quick calibration page as shown in *Figure 7*. First select and connect the new camera, then click *Detect Feature Points and Calculate*, and click *Finish and Exit*. The operation notification bar will report the current progress in real time.

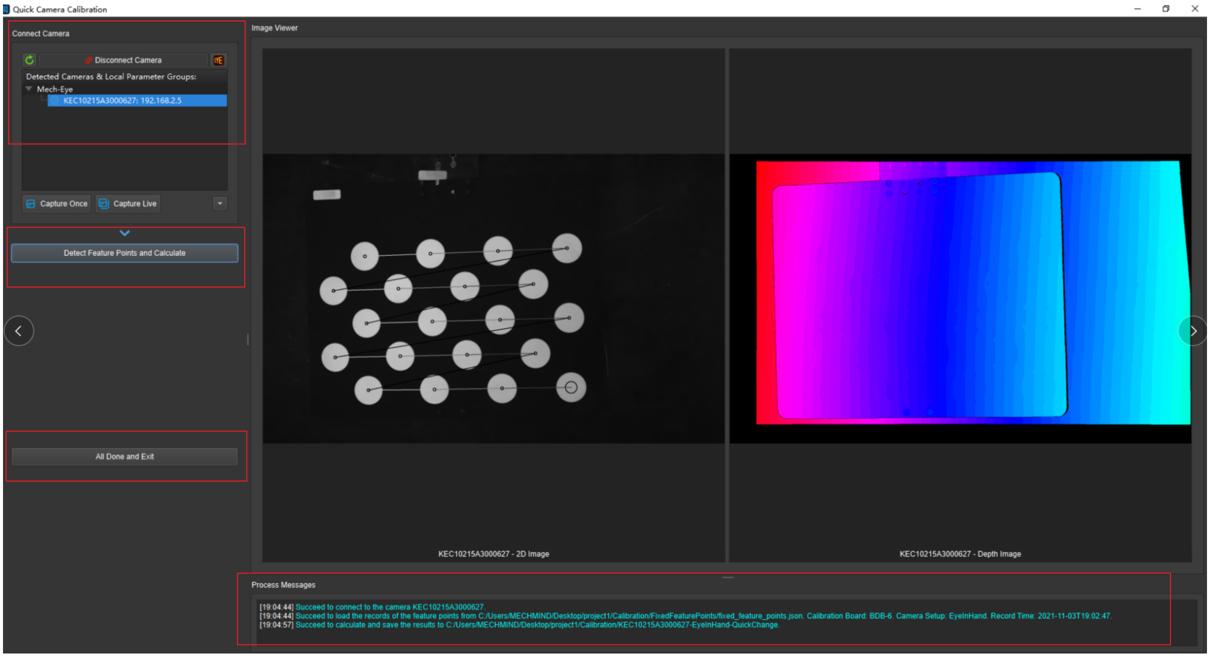


Figure 7. New Camera Calibration

After the calibration is completed, the camera with the parameter group can be selected in the Mech-Vision project. The folder location of the new camera parameter group obtained by the quick calibration is shown in *Figure 8*.

Figure 8. Folder Containing the New Camera Parameter Group

6.4 Inspect and Analyze Calibration Results

For the calibrated extrinsic parameters, it is necessary to check whether the calibration accuracy meets the requirements. If the error fall outside the normal range, the user should diagnose and solve the problems that are responsible for the error before recalibrating to obtain qualified extrinsic parameters.

6.4.1 Inspect Calibration Results

- **Inspect the extrinsic parameters**

After the calibration calculation is complete, the Calculate Camera Parameters interface will display the calibration result and the point cloud showing the calibration error, as shown in *Figure 1* below. The error point cloud shows the deviation between the calculated value and the actual value of the feature points on the calibration board.

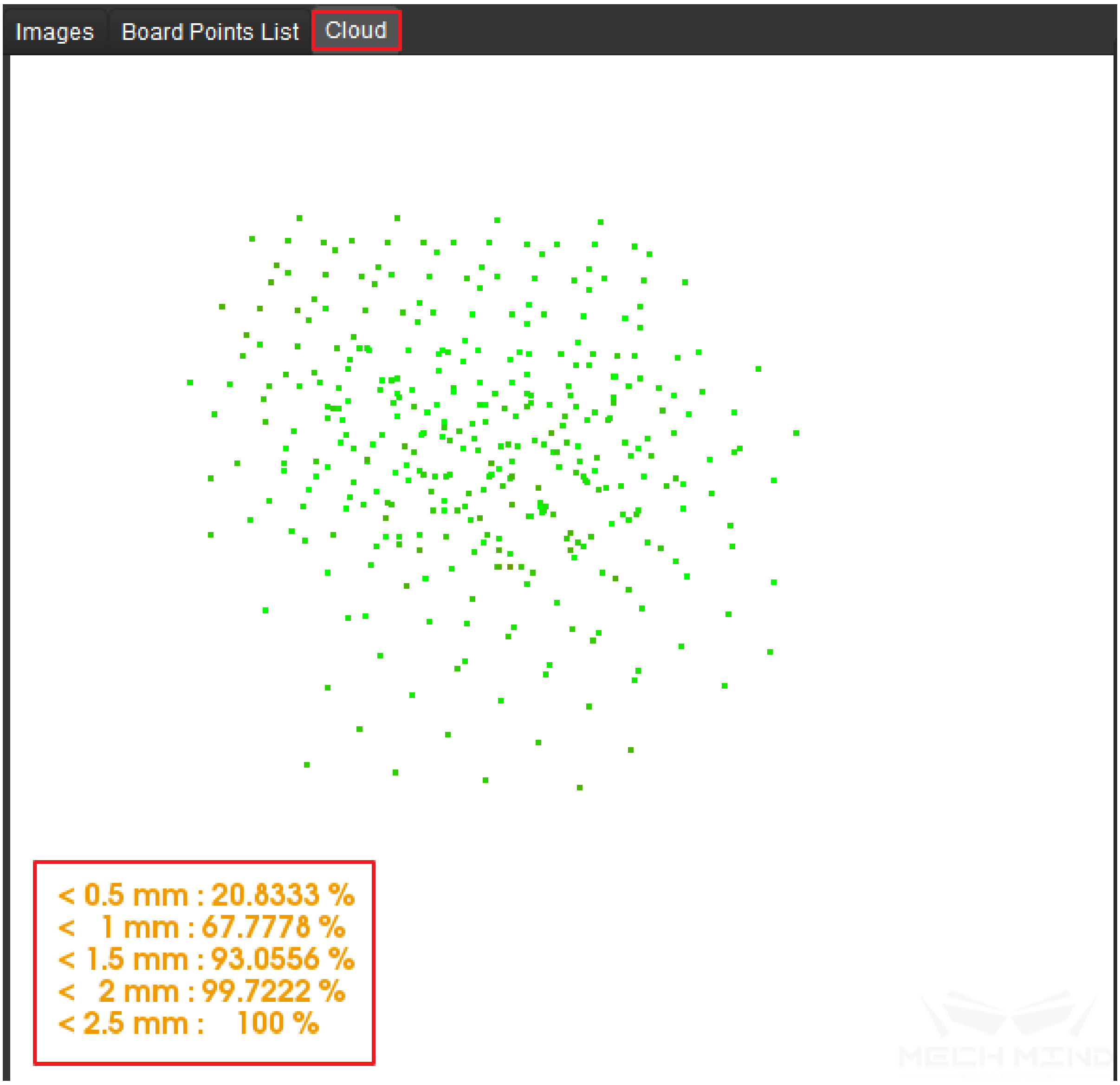


Figure 1 Point Cloud Displaying Calibration Error

The color of the point represents the error level (one error level per 0.5mm of deviation). The darker the color is, the greater the error of the point. Pressing the number keys 0-9 will highlight the points in the corresponding error level (where 0 corresponds to points with error less than 0.5mm, 1 corresponds to points with errors between 0.5mm and 1mm, etc). As shown in *Figure 2*, pressing the number key 0 will highlight all points with error less than 0.5mm.

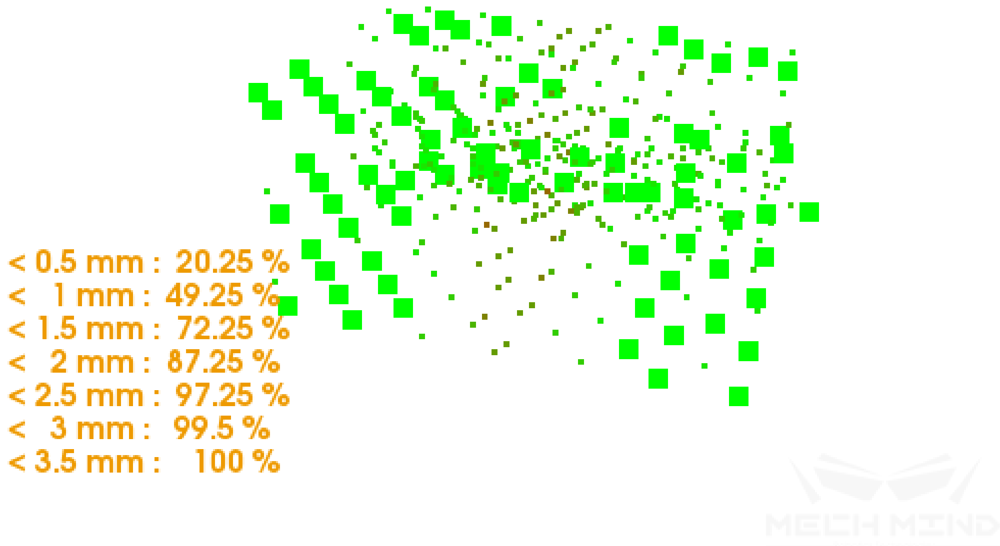


Figure 2 Points with Calibration Error Less than 0.5mm

Similarly, as shown in *Figure 3*, pressing the number key 5 will highlight all points with error between 2.0mm and 2.5mm.

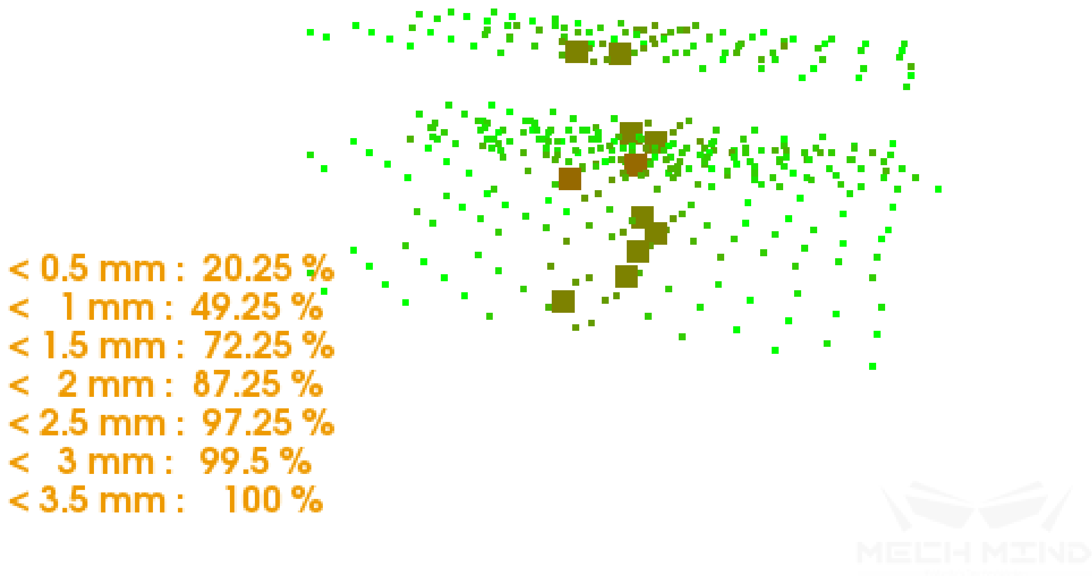


Figure 3 Points with Calibration Error between 2.0mm and 2.5mm

The calibration error will be synchronized in the *Log* window on the bottom of Mech-Vision interface, as shown in *Figure 4* below.

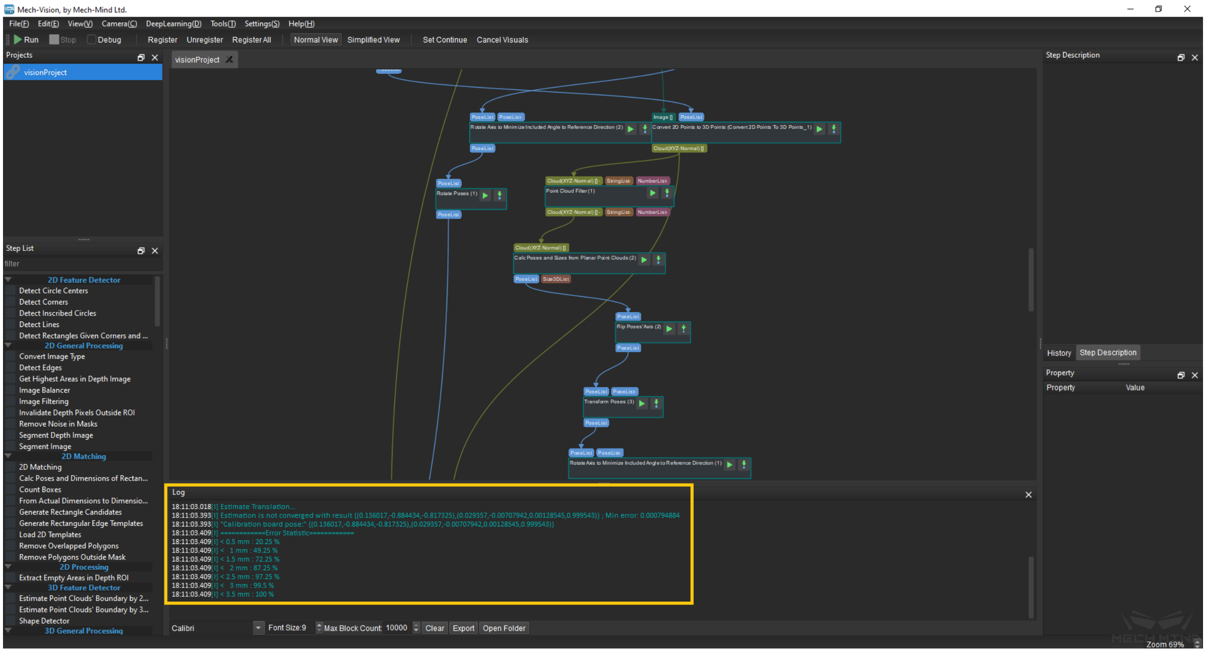


Figure 4 Calibration Error Shown in the Log Window

The point cloud displayed in Mech-Viz can be used to inspect the calibration results roughly.

- **Roughly inspect the extrinsic parameters for ETH**

After obtaining the extrinsic parameters using the ETH method, move the robot into the view of camera and click the *Display PointCloud in Mech-Viz* button. The point cloud should be roughly consistent with the robot model if the calibration accuracy is high, as shown in the *Figure 5* (be aware that the simulation cannot match the real robot completely, and thus cannot be used to fine-tune the extrinsic parameters).

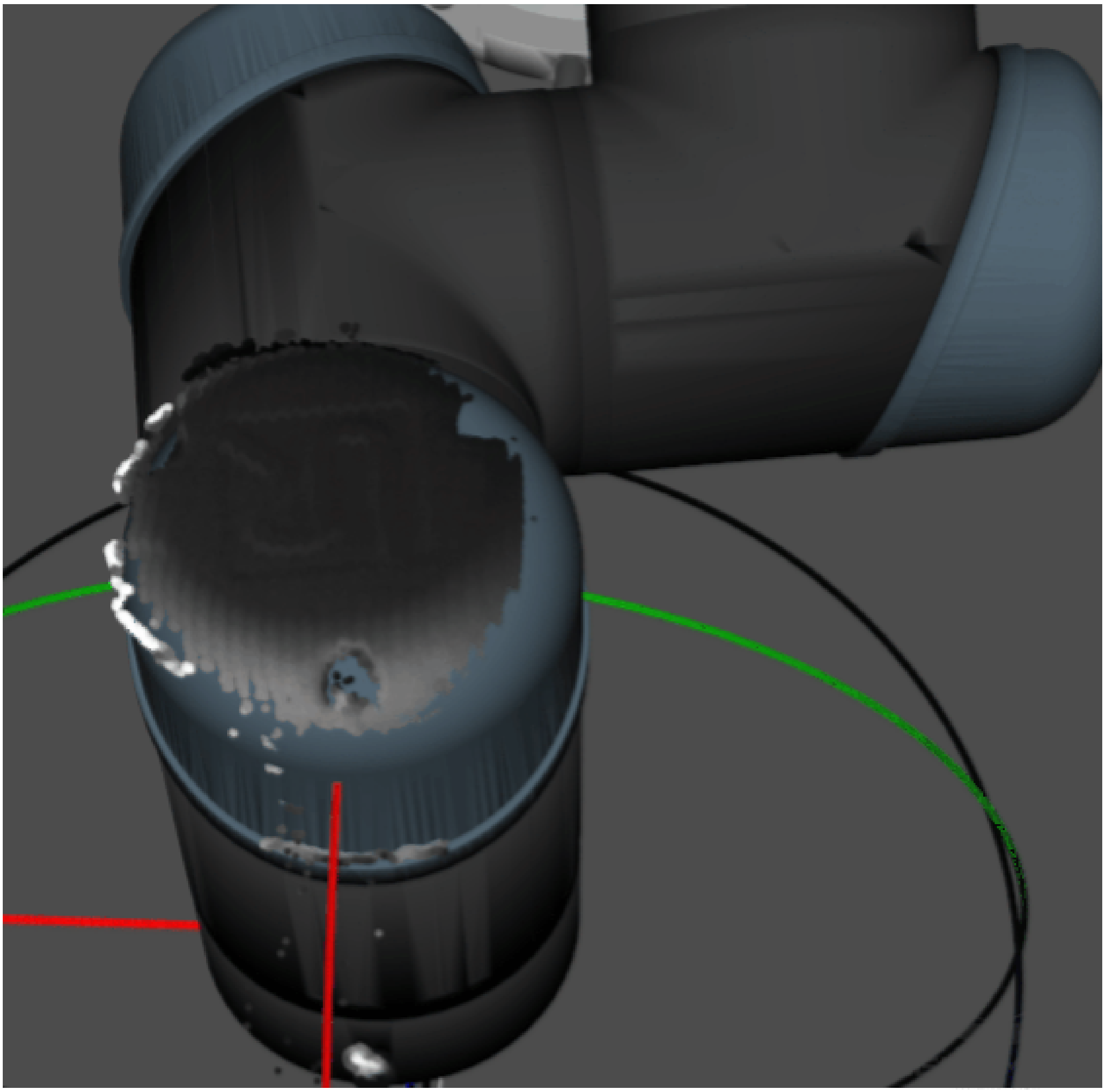


Figure 5 Roughly Inspect the ETH Calibration Results in Mech-Viz

- **Roughly inspect the extrinsic parameters for EIH**

When using the EIH method, place the calibration board on the ground and capture images of the board from multiple different poses. Then, observe if the point cloud of the calibration board deviates noticeably in the robot base coordinate system, as shown in *Figure 6*.

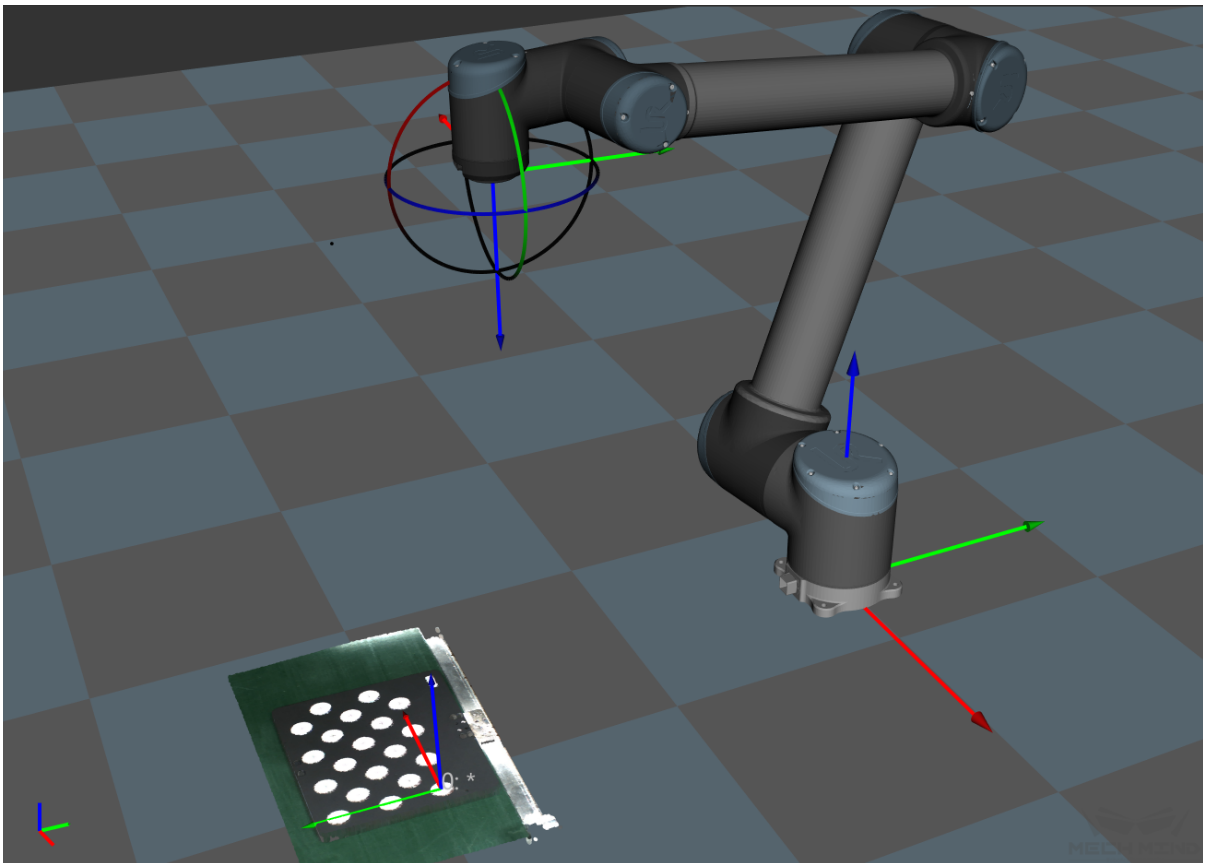


Figure 6 Roughly Inspect the EIH Calibration Results in Mech-Viz

Attention: When using the EIH method, robot poses cannot be obtained in real time if no robot is connected. It is required to enter the current pose before checking the point cloud in Mech-Viz. Do not fine-tune the extrinsic parameter according to the rough inspecting results.

6.4.2 Analyze Calibration Results

- **Evaluate the calibration results**

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 3mm (< 3mm: 100%).

For high-precise projects, all data points should have errors lower than 2mm (< 2mm: 100%).

For palletizing and depalletizing projects, all data points should have errors lower than 5mm (< 5mm: 100%).

The above standards are just listed as references. Please follow the specific accuracy requirement of the real application.

- **Factors affecting the calibration accuracy**

If the calibration accuracy is below standard, the user must follow the following instructions to troubleshoot the source of the errors.

Inspect the Accuracy of Calibration Data

Calibration data refers to the calibration points data generated during the calibration process and is stored in calib_data.json. This file records data such as the flange poses and calibration points' position for every imaging pose. When loading the calibration data, select the *Load the calibration data with a virtual camera* option, and the software will automatically load the data in this file.

Inspect three main aspects of this data:

- **Check if the Euler angles change between different sets calibration points (excluding the rotating sets)**

Click a pose in the Marker-Image and Pose List. Select *Board Points List* to see the flange pose of the current calibration point set, as shown in *Figure 7* below.

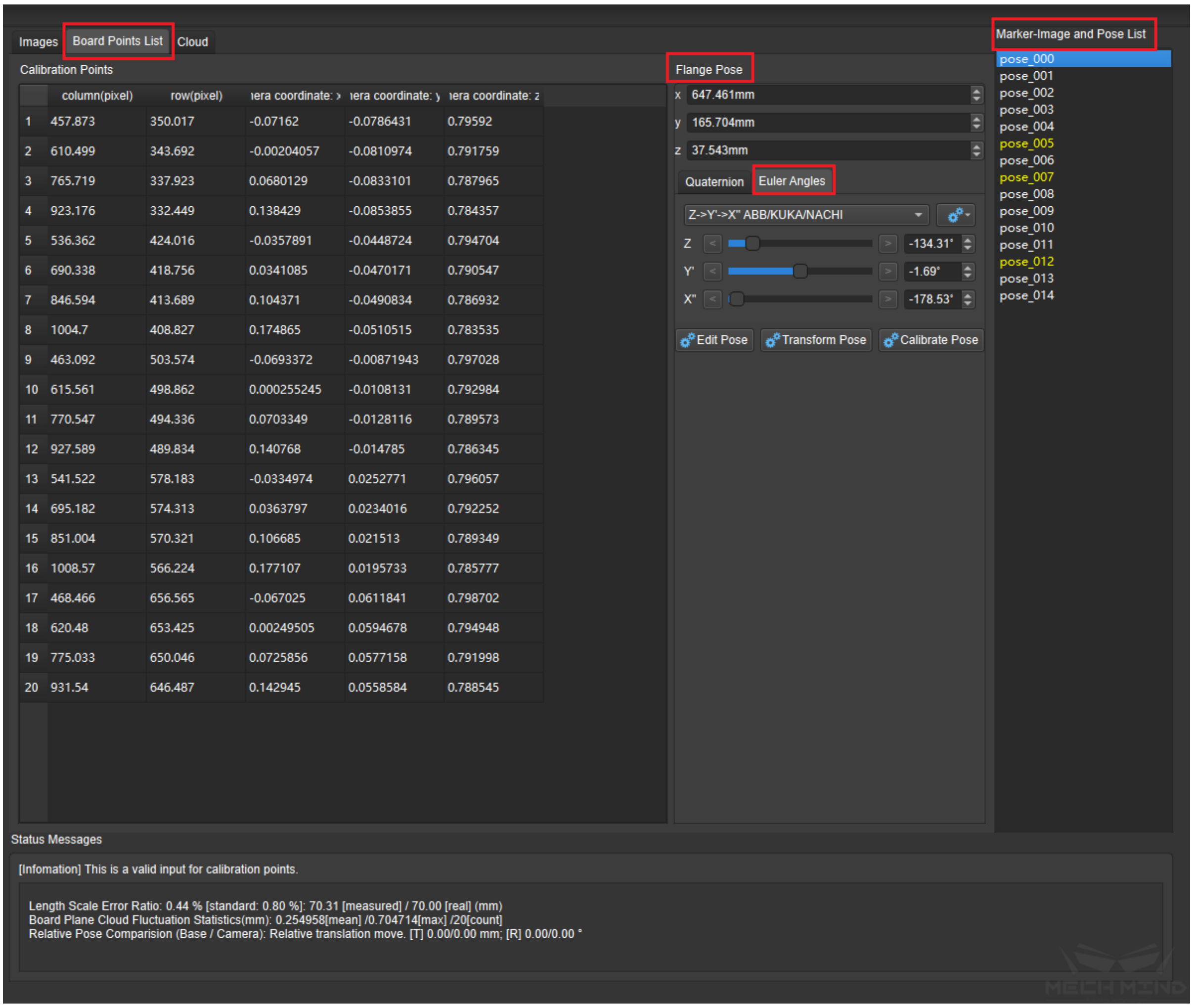


Figure 7 Check Flange Poses of Sets of Calibration Points

Note: In *Figure 7*, column(pixel) and row(pixel) represent the pixel coordinates of the center points of the calibration circles in the current 2D image; camera_coordinate:x(m)/y(m)/z(m) represents the coordinates of the center points relative to the camera coordinate system in the current depth image.

During the calibration process, the robot will move along its base coordinate or flange without rotating, meaning the Euler angle will be consistent throughout the entire procedure. Depending on the precision level of the robot, the Euler angle might fluctuate differently. If the fluctuation exceeds 1 degree, the robot might have lost its zero position or have bad precision.

Solution: in the above situation, hand-eye calibration should not be continued. The user should check the robot’s zero position and fix the imprecision problem before continuing to calibrate.

- **Check if the “Length Scale Error Ratio” of the calibration points sets exceeds the standard values**

After selecting a pose in the Marker-Image and Pose List, the Status Messages box below will display the *Length Scale Error Ratio* of the current image. If it exceeds the standard value, the calibration point will turn to yellow to warn the user. As part of the intrinsic parameters check result, *Length Scale Error Ratio* can partly reflect the current error in the camera intrinsic parameters.

Solution: for projects will 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 error, it is recommended to recalibrate the intrinsic parameters or replace the camera.

Note: Errors in intrinsic parameters will affect the calibration result, but if it is within the normal range, it is generally acceptable.

- **Check if the max value of the “Board Plane Cloud Fluctuation Statistics” exceeds 3mm**

Board Plane Cloud Fluctuation Statistics represents the overall fluctuation of the plane containing the coordinates of all the circles’ center points. Every calibration points set has its own *Board Plane Cloud Fluctuation Statistics*. This value directly influences the accuracy of the extrinsic parameters, and the higher a project’s precision requirements are, the lower the fluctuation value should be. Generally, the max acceptable value for the fluctuation is 3mm. If more than three calibration points sets have fluctuation statistics greater than 3mm, the user should troubleshoot the problems and recalibrate. Potential factors that might cause this are as follows:

1. The 3D camera exposure parameters are not optimized. It is likely that the *Cloud Smoothing* and *Outliers Removal* options are turn off or that the camera gain is turned on.

Solution: Change *Cloud 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.

2. The second source of the fluctuation occurs when multiple random board poses method is used to calibrate for the ETH setup. If the calibration board is not firmly installed on the robot’s flange and the speed of 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.

Check the *Length Scale Error Ratio* and *Board Plane Cloud Fluctuation Statistics* as shown in *Figure 8* below.

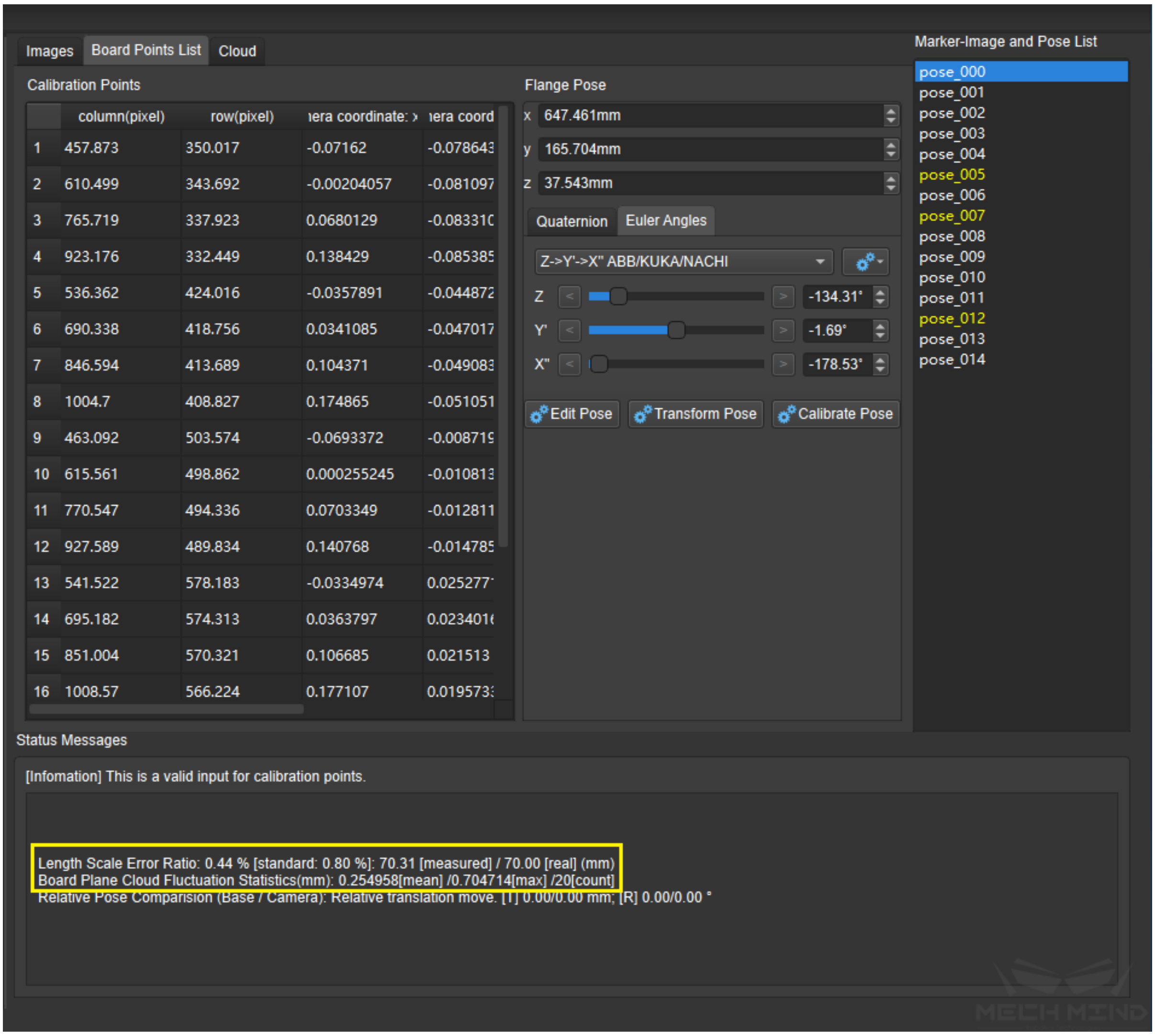


Figure 8 Check Intrinsic Parameters and Point Cloud Fluctuation

Recalculate the Rectification Parameters

After troubleshooting the above problems, please recalibrate the camera and check if the result meets the requirement. If so, finish the calibration process. If not, click the *Calculation Parameters Setting* button and check the *Recalculate Rectification Parameters* option to recalculate the result.

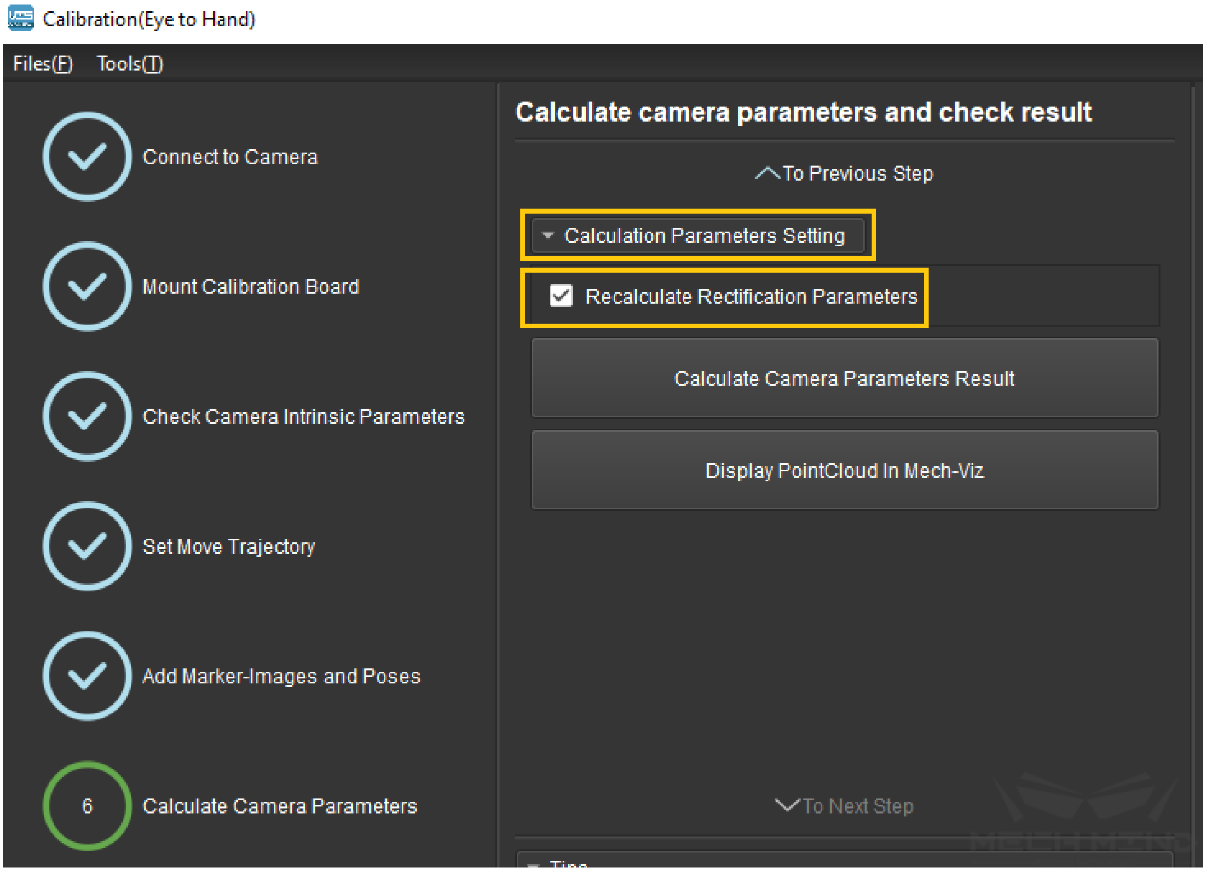


Figure 9 Recalculate Rectification Parameters

Attention: Before checking the *Recalculate Rectification Parameters* option, please create a copy of the extri_param.json file and rename it to factory rectification parameter.

Now, recalibrate the extrinsic parameters until it reaches the required precision.

Note: After calibration, adjust 3D exposure parameters in Mech-Eye Viewer software according to the production requirements.

6.5 FAQ

6.5.1 Explanation for Factory Rectification Parameters

Explanation and Use of "Recalculate Rectification Parameters"

The rectification parameters are stored in the camera extrinsic parameter file extri_param.json, as shown in the following figure *1*:

```
{
    "depthInBase": [
        1.432638082659246,
        1.8377671238758124,
        3.2883363420159943,
        -0.009108137087217968,
        0.7098468663853315,
        -0.7042887897061081,
        0.0034305708025669807
    ],
    "depthToCloud_offset": {
        "Region0": {
            "Layer0": [
                1.0077658936848384,
                0.0012689124404129978,
                0.00043064749099157484,
                2.3869795029440866e-15,
                0.0007093776048349514,
                1.0071560125601289,
                0.0004638496105526385,
                -4.440892098500626e-16,
                0.0017282657096284915,
                0.002712867205123279,
                1.0144842496796416,
                4.440892098500626e-16,
                -0.005293848730093753,
                -0.004644376122671545,
                -0.038295841686403695,
                0.9999999999999942
            ],
            "Layer1": [
                1.0077658936848384,
                0.0012689124404129978,
                0.00043064749099157484,
                2.3869795029440866e-15,
                0.0007093776048349514,
                1.0071560125601289,
                0.0004638496105526385,
                -4.440892098500626e-16,
                0.0017282657096284915,
                0.002712867205123279,
                1.0144842496796416,
                4.440892098500626e-16,
                -0.005293848730093753,
                -0.004644376122671545,
                -0.038295841686403695,
                0.9999999999999942
            ],
```


The rectification parameters will be a unit matrix if the camera does not have rectification parameters, as shown in the figure below *2*:

```
"depthToCloud_offset": {
    "Region0": {
        "Layer0": [
            1,
            0,
            0,
            0,
            0,
            1,
            0,
            0,
            0,
            0,
            1,
            0,
            0,
            0,
            0,
            1
        ],
```

If you check *Recalculate Rectification Parameters* and click ‘calculate’, the optimal rectification parameters will be refitted according to all current calibration points data, and the rectification parameters in the extrinsic parameter file will be updated.

Factory rectification parameters: The factory rectification parameters are based on the focal length of the camera and its maximum workspace. The advantages and disadvantages of using the factory rectification parameters:

Advantages: Using factory rectification parameters, extrinsic parameters that represent the accuracy of the entire working space can be calculated under the condition that the number of calibration points is small and the workspace is not fully marked. The factory rectification parameters are preferred when the TCP touch method is used and the calibration space is limited after the robot is installed with the calibration board and the workspace cannot be fully marked.

Disadvantages: The actual on-site robots and the space of calibration moving trajectory are different. The factory rectification parameters cannot guarantee the best results every time. Using *Recalculate Rectification Parameters* can obtain better results if the workspace is fully marked.

Attention: After obtaining the extrinsic parameters that meet the accuracy requirements with the factory rectification parameters, it is not recommended to check *Recalculate Rectification Parameters* for higher accuracy if the workspace is not fully marked.

Solutions When Factory Rectification Parameters are Invalid

Invalid factory rectification parameters means that the on-site rectification parameters are updated but the factory rectification parameters are not backed up; or the camera is not marked with the factory rectification parameters, etc.

- **Solution when using TCP touch method**

In this case, the problem of inaccurate rectification parameters can be solved by adding multiple sets of touch points and then calculating.

Add multiple sets of touch points: place the calibration board at the bottom middle of the workspace, poke three points with the tip and take a picture to generate a touch point; place the calibration board on the edge of the workspace, poke points again to generate a touch point;

Divide the workspace into different layers and add multiple sets of touch points, as shown above (generally, three layers and two points per layer are enough).

- **Solution when using multiple random calibration board poses**

In this case, the problem of inaccurate rectification parameters can be solved by fully marking the workspace and checking *Recalculate Rectification Parameters*.

6.5.2 Non-six-axis Robot Calibration

Four-axis Robot

Four-axis robots include truss robots, SCARA robots, and palletizing robots, etc. Considering the frequency of use, Mech-Viz is only adapted to a small number of SCARA and palletizing robots (the adaptation range will continue to be expanded in the future).

- **Tip fixation when using TCP touch method**

Mount the sharp tip on the end of the robot and check whether the tip is stably fixed at one point and installed in the center of the flange by rotating the fourth axis. The TCP of the four-axis robot cannot be marked. The XYZ values of the TCP need to be measured manually.

- **How to adjust the Z-direction for four-axis robot calibration**

When using the multiple random calibration board poses method, the four-axis robots' lack of rotational degree of freedom will result in the robot lacking the rotational values during the calibration process. After the calibration is completed, the Z direction of the extrinsic parameters needs to be manually adjusted:

1. Find the base coordinate position of the robot;
2. Place the calibration board on the working plane parallel to the XY plane of the robot base coordinate, usually the ground;
3. Measure the distance from the base coordinate of the robot to the working plane. In Mech-Viz, set the ground height to this distance. As shown in *Figure 3* below:

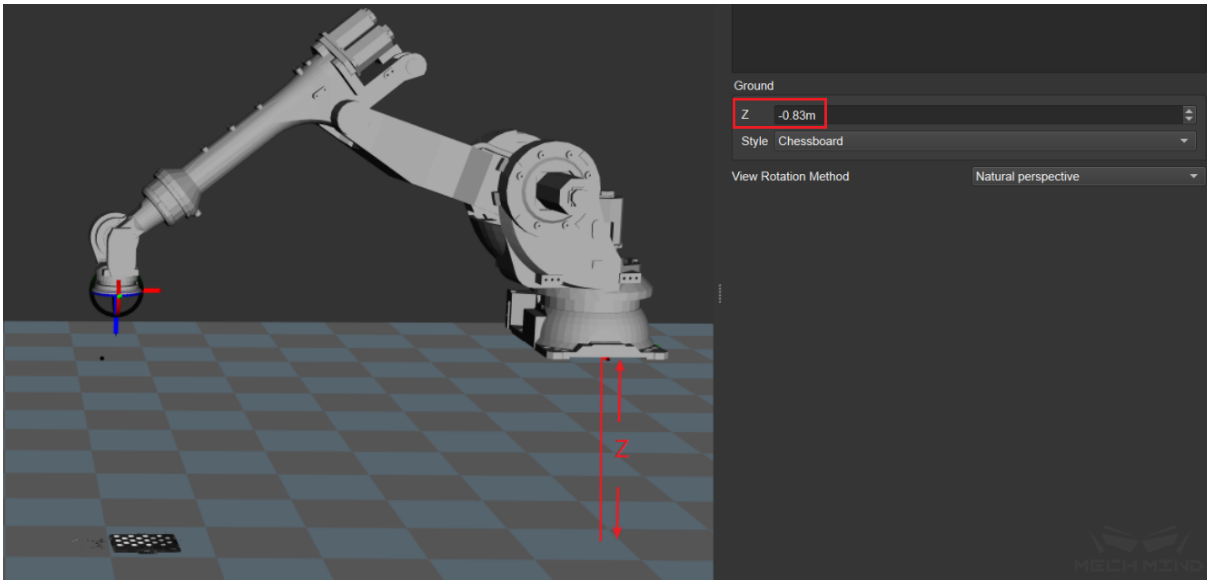


Figure 3 Adjust the Ground Height in Mech-Viz

4. Adjust the Z-direction value of the extrinsic parameters so that the point cloud of the calibration board is exactly on the working plane in Mech-Viz. Then, the Z-direction adjustment is completed.

- **Instructions for truss robots**

Depending on the type of robot and the defined position of the base coordinate of the truss robot, it is difficult to adjust the Z direction based on the base coordinate position. Therefore, only the TCP touch method is recommended for truss robot.

The most common setup for truss robot is to mount the camera on the third axis:

EIH: In this case, the truss robot becomes a three-axis robot as the fourth axis cannot be used. The user can only send fixed angles to the robot through the adapter.

ETH: There is a limited number of fixed scanning points and the offset between each scanning point are known. Add the robot base coordinate offset in the adapter each time a different scanning point is used.

Seven-axis Robot/Six-axis Robot with Slide Rail/Five-axis Robot

Note: 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 seven-axis robots using multiple random calibration board poses method**

When there is no suitable sharp tip on site or the sharp tip cannot be mounted, the multiple random board poses method can be used to calibrate the seven-axis robot. During the calibration process, it is necessary to limit 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.

6.5.3 How to Calibrate When the Euler Angle Type of the Robot is Uncertain

Get Euler Angle Type

When the Euler angle of the robot is unknown, use the *Get Euler Angle Type* under the *Tools* menu bar to get the Euler angle type of the robot, as shown in *Figure 4* below:

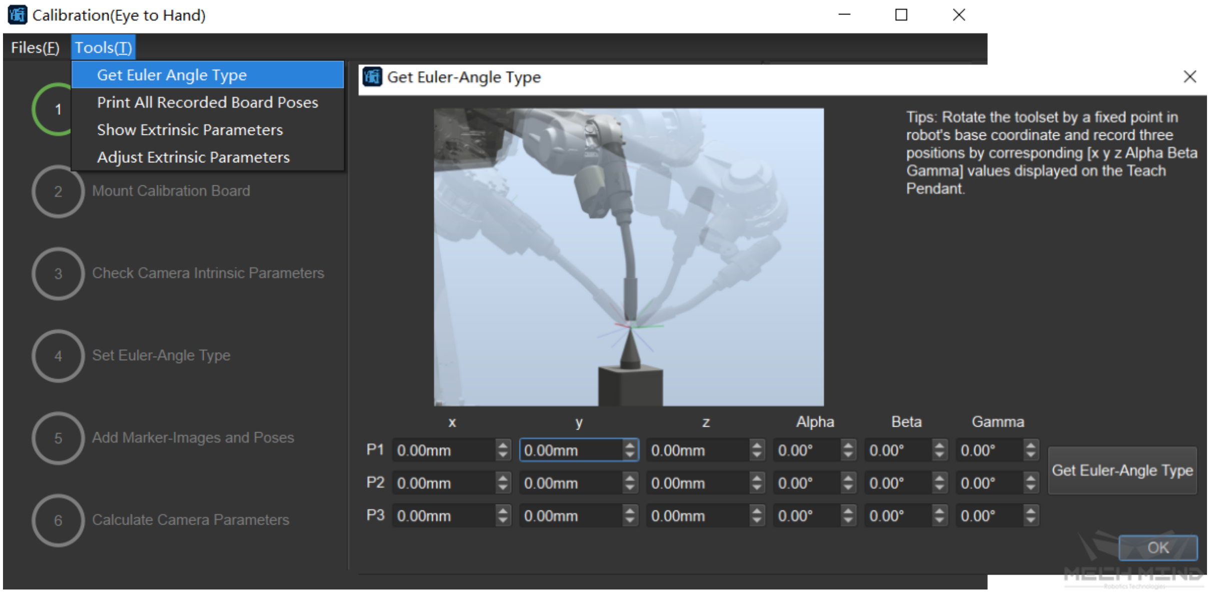


Figure 4 Get Euler Angle Type

Adjust the robot’s pose to obtain three different poses where the tip mounted at the end of the robot and the tip on the desktop touch each other. Fill in the pose of the robot on teach pendant after each touch. After which, click *Get Euler Angle Type* to obtain the recommended Euler angle.

How to Calibrate if There is Only a Sharp Tip Mounted at the End of the Robot with Known TCP but No Suitable Tip Fixed onto the Workspace

Since the accurate Euler angle type of the robot is unknown, the Euler angle cannot be inputted according to the correct type when using the TCP touch method for calibration.

In this case, switch the pose on the robot teach pendant to the TCP pose so the pose of the tip is obtained.

Touch three points in turn, read the values of XYZ on the teach pendant and input. Choose any Euler angle type and enter any constant value. Make sure that the Euler angles of the three input poses are the same.

6.5.4 Solutions for When A Large Number of Calibration Point Cloud Fluctuations are Out of Tolerance During Long-Distance Calibration and Adjusting Camera Parameters Does Not Improve

Check the calibration board to see if there are crosses which may cause large fluctuations in the point cloud, as shown in *Figure 5* below:

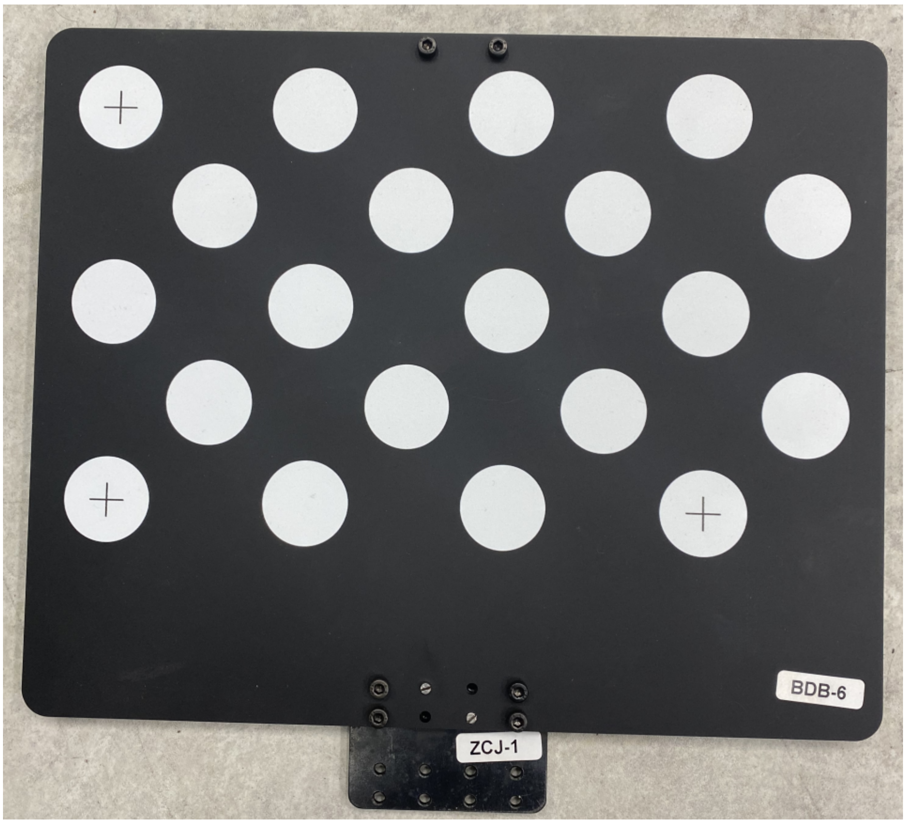


Figure 5 Crosses on the Calibration Board

Find A4 paper to cover the center of each dot (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 problem, try to manually add more poses and delete the poses with severe point cloud fluctuations.

6.5.5 Common Misunderstandings about Calibration

- **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 rectification parameters to calculate, determine the number of points for each layer according to the camera focal length, the size of the calibration board, etc.

When the focal length is 300-2000mm, 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 2000-3500mm, 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 using the factory rectification parameters, calibration should be carried out in layers around the center of focus and around the focal length of the camera. When the factory rectification parameters are not used or when it is impossible to calibrate around the camera’s focal length, it is recommended to calibrate the entire working area.

- **The extrinsic parameters of the calibrated area are accurate, while the extrinsic parameters of excluded area are inaccurate**

The error point cloud generated after calibration shows the extrinsic parameters error of the calibrated area. However, it does not mean that the extrinsic parameters of the unmarked area must be inaccurate. On the contrary, when using the factory rectification parameters, the extrinsic parameters of the unmarked area are also usually accurate.

CHAPTER

SEVEN

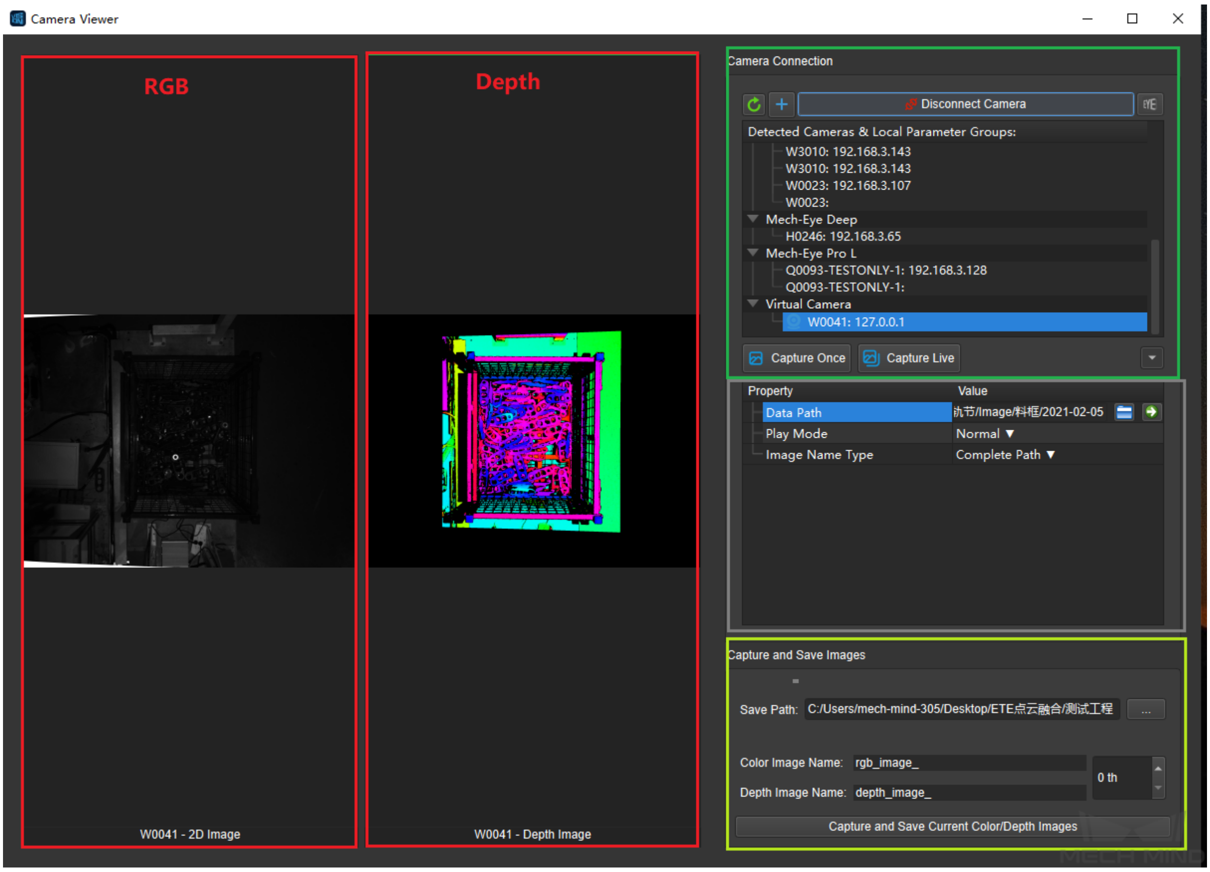
SUPPLEMENTARY TOOLS

This chapter will introduce the tools in Mech-Vision. These tools are used in some special aplication scenarios in order to improve the efficiency and accuracy of vision solutions.

7.1 Camera Viewer

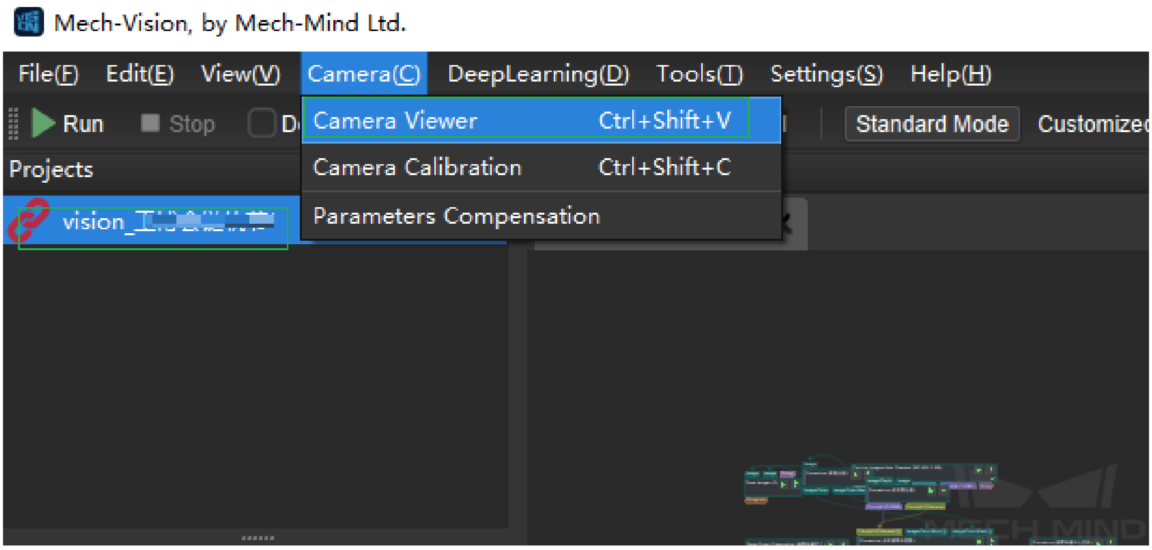
7.1.1 Instruction of Camera Viewer

The functions of camera viewer are: 1. connect and control the camera to take pictures; 2. set the attributes of the camera; 3. display RGB and Depth images; 4. specify the save path and save the image. The function introduction of each part of the camera viewer is shown in the figure below.



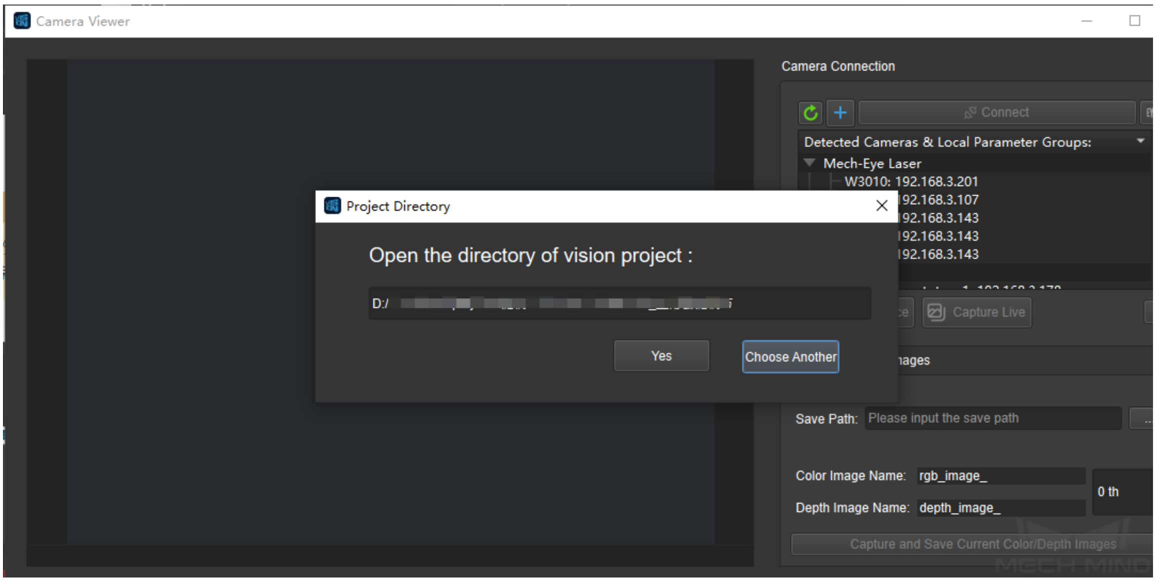
7.1.2 Operation Procedure of Camera Viewer

1. Open the project in Mech-Vision. Enter "Camera Viewer" from *Camera* → *Camera Viewer* .

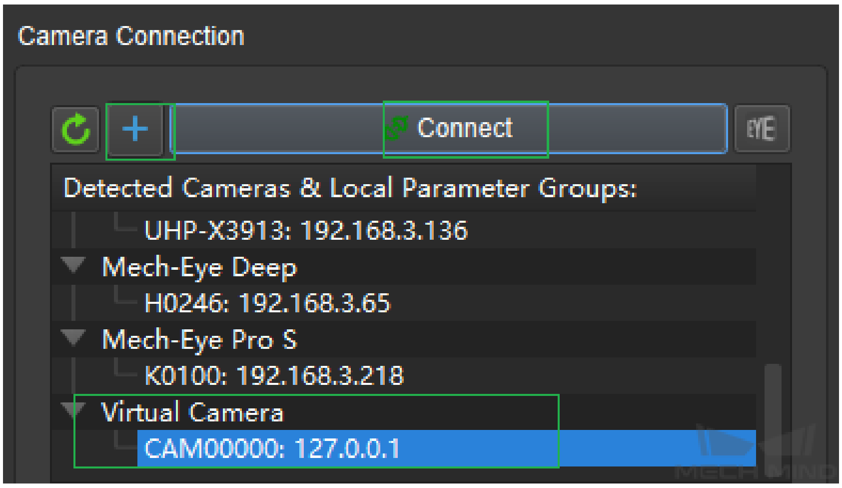


2. Load camera parameters.

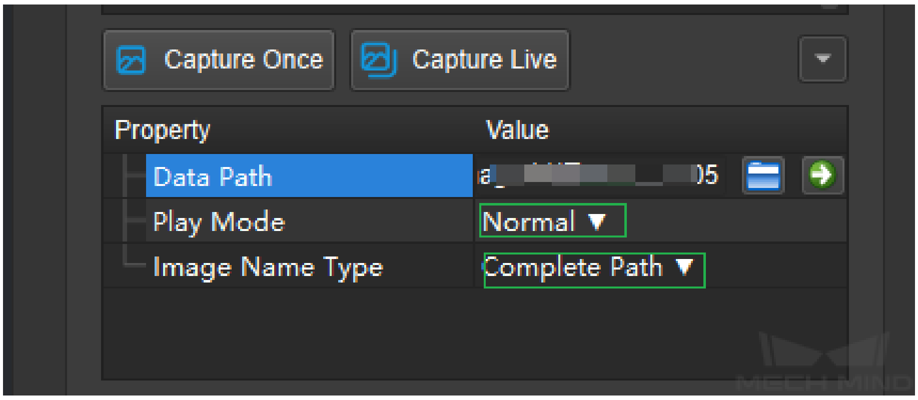
Attention: The camera needs to load the corresponding intrinsic and extrinsic parameters when taking pictures. The intrinsic and extrinsic parameters are saved in the corresponding project directory, so just load the corresponding project directory; you can also enter without loading any directory.



3. If the user choose to use a virtual camera, click + in the camera connection bar to create a virtual camera. Select "CAM0000: 127.0.0.1" to *connect the camera.*

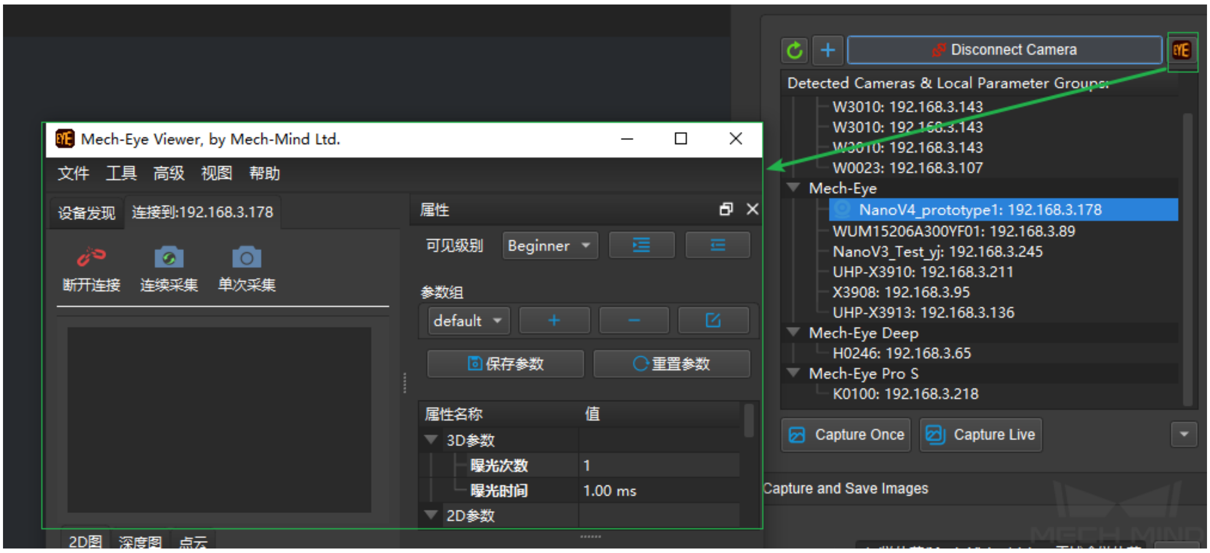


To use the virtual camera, the user need to select the image data path. If the selected image data path is correct, the log window prompts that the intrinsic and extrinsic parameters are successfully loaded. The camera configuration is updated, and the picture can be collected normally. The user can also change the Play Mode (Normal, Repeat One, Rpeat all, Shuffle), and change the output Image Name Type (Complete path, File Name, Base name).

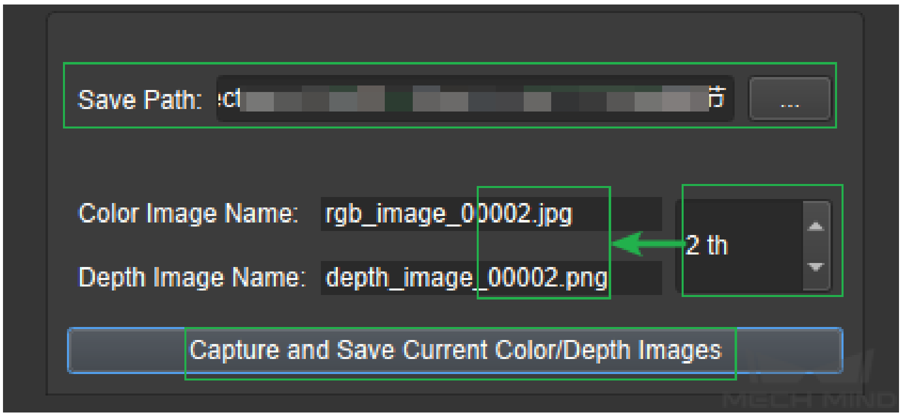


When selecting the image data path ,if the depth image, color image, camera parameters, and flange poses files in the selected folder are not stored in a fixed format, it will trigger the “Virtual Camera Assistant” to assist in selecting image data. Refer to *Capture Images from Camera* for specific usage. After the settings are completed, the images can be captured normally.

4. If it is a real camera, click the single acquisition or continuous acquisition to obtain the color image and the depth image, or click the *Mech-Eye Viewer* in the upper right corner to enter the Mech-Eye Viewer software window for more detailed settings. For the use of Mech-Eye Viewer software, please refer to Mech-Eye Viewer.



5. Set the image save path and the file name. Set the serial number of color image and depth image. Finally, click the *Capture and save Current Color/Depth Images* to complete the operation.

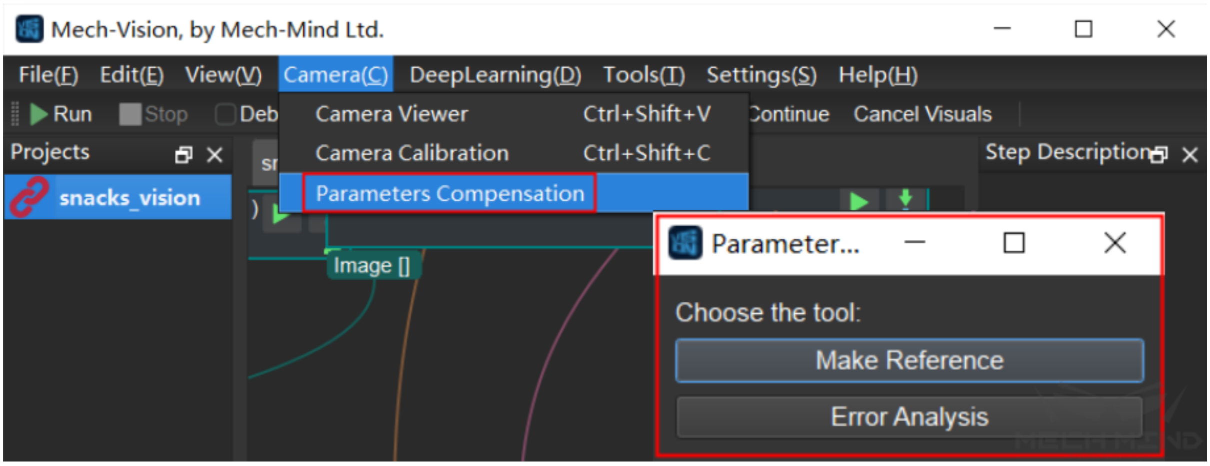


7.2 Parameter Compensation

Long-term operation of the robot and camera may cause slight changes in the intrinsic and extrinsic parameters of the camera, which will affect the accuracy of grasping. If it is not possible to calibrate the camera on site, the parameters compensation method can be used to correct the intrinsic and extrinsic parameters of the camera.

The principle of parameters compensation is similar to *Hand-Eye Calibration Guide*, and it can be regarded as a lighter and faster calibration. As a result, its accuracy is not as high as the camera calibration. When there are serious camera positioning problems (according to the requirements of different tasks), it is recommended to recalibrate.

The parameters compensation tool is under *Camera* → *Parameters Compensation*. Click *Parameters Compensation*, there will be a dialog box as shown in the figure below.

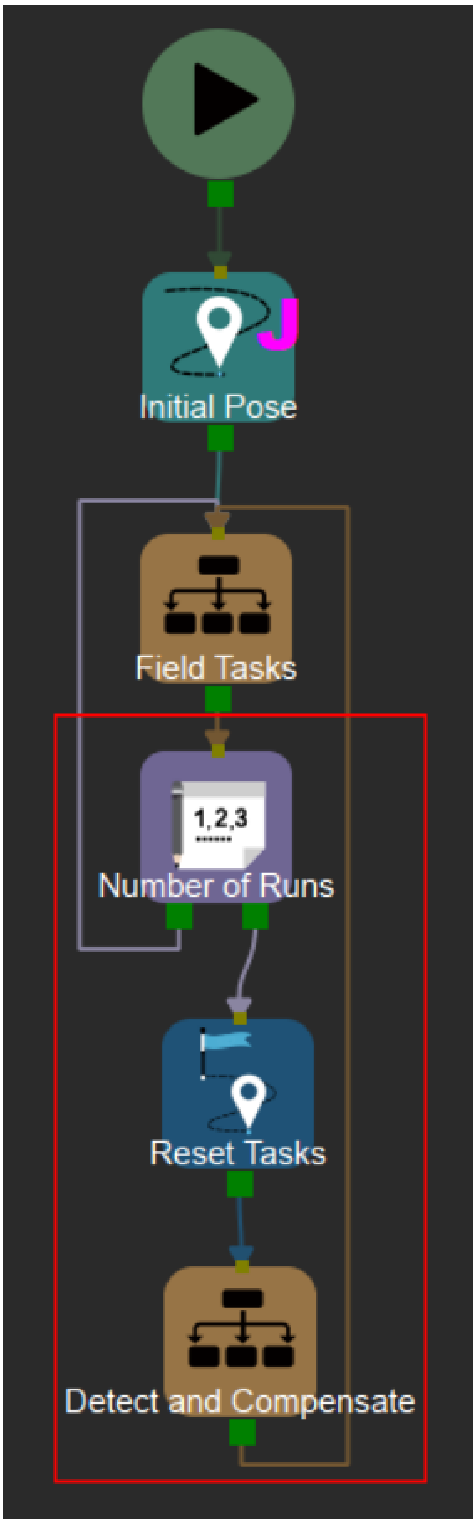


Make Reference is to detect the markers and estimate their poses in camera coordinates through each observation. Compensation is obtained by comparing the difference between the baseline (initial) observation poses of the markers in the scene and the subsequent observation poses.

Error Analysis is a data visualization analysis of the results of parameters compensation.

7.2.1 Procedure of Parameters Compensation

It is required to add a process of detecting markers and compensating parameters after the normal field tasks, and the markers need to be fixed on site. In terms of project, it is necessary to add some Skills to the original Mech-Viz project and add some Mech-Vision projects. The Skills and projects that need to be added is shown in the figure below.


Initial Pose
Field Tasks
1,2,3
Number of Runs
Reset Tasks
Detect and Compensate


marker_detection
error_compensation

Compensation is not required every time a normal project is carried out, so a Counter and a Reset Task are added to set a value and perform compensation when the number of times of normal project operation reaches this value.

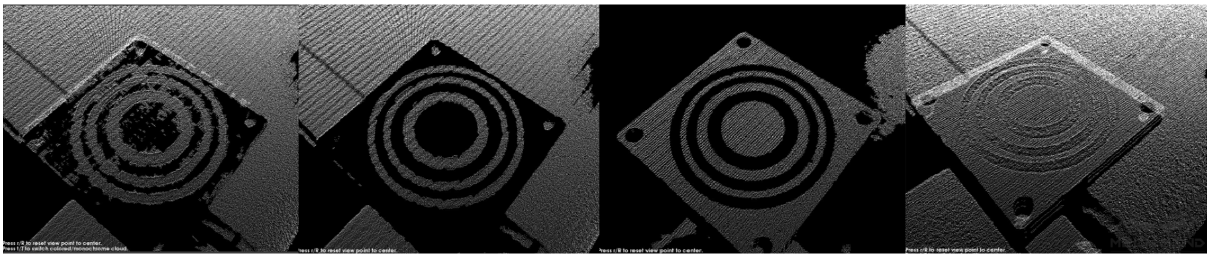
The added content is generated by *make_reference* in Mech-Vision. The usage of this tool will be introduced below.

7.2.2 Make Reference

Set Camera Exposure Parameters

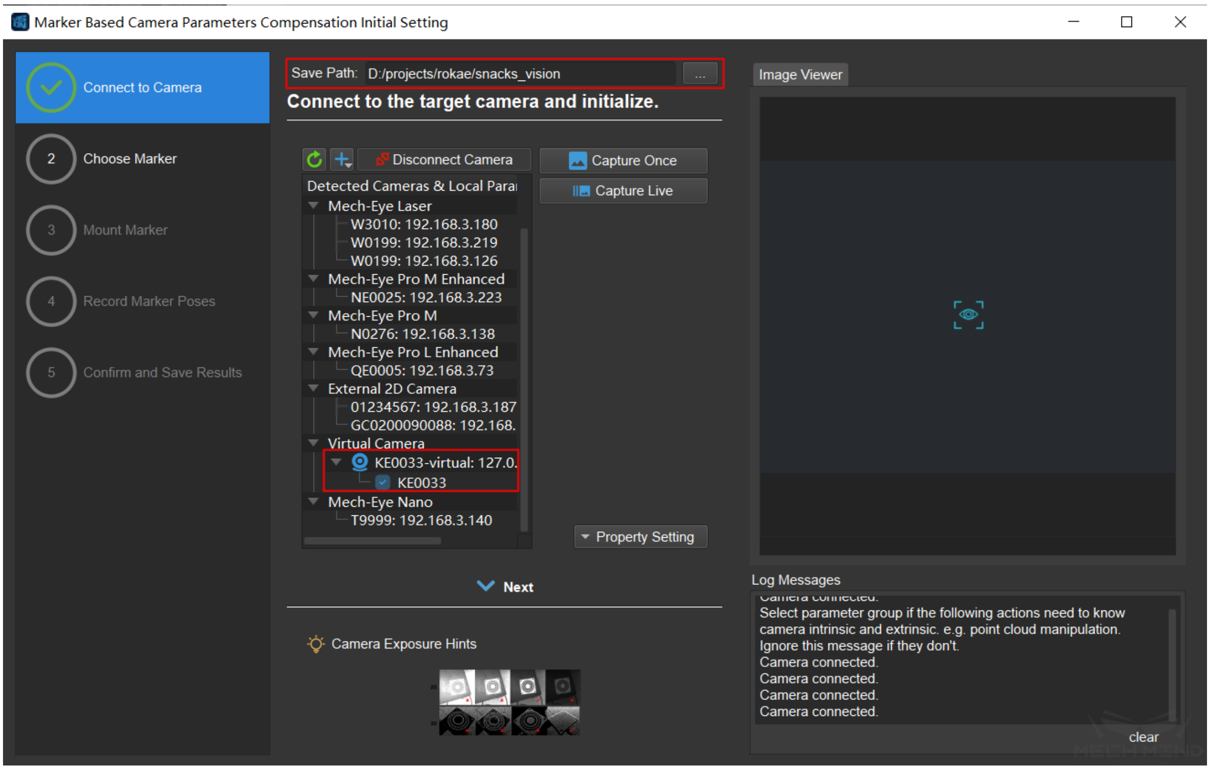
Under the condition of meeting the original project requirements, adjust the 3D exposure parameters in Mech-Eye Viewer until the white area on the surface of the markers has point cloud. The picture below ranges from poor to good. The image on the far left is the worst, while the image on the far right is the best.

Make sure that the markers are within the best field of view of the Mech-Eye camera to ensure smooth point clouds on the surface of the markers. If it cannot be guaranteed due to special reasons, priority should be given to ensuring the smoothness of the point cloud near the key points of the marker (center of a circle, corner points of a square).



Connect to Camera

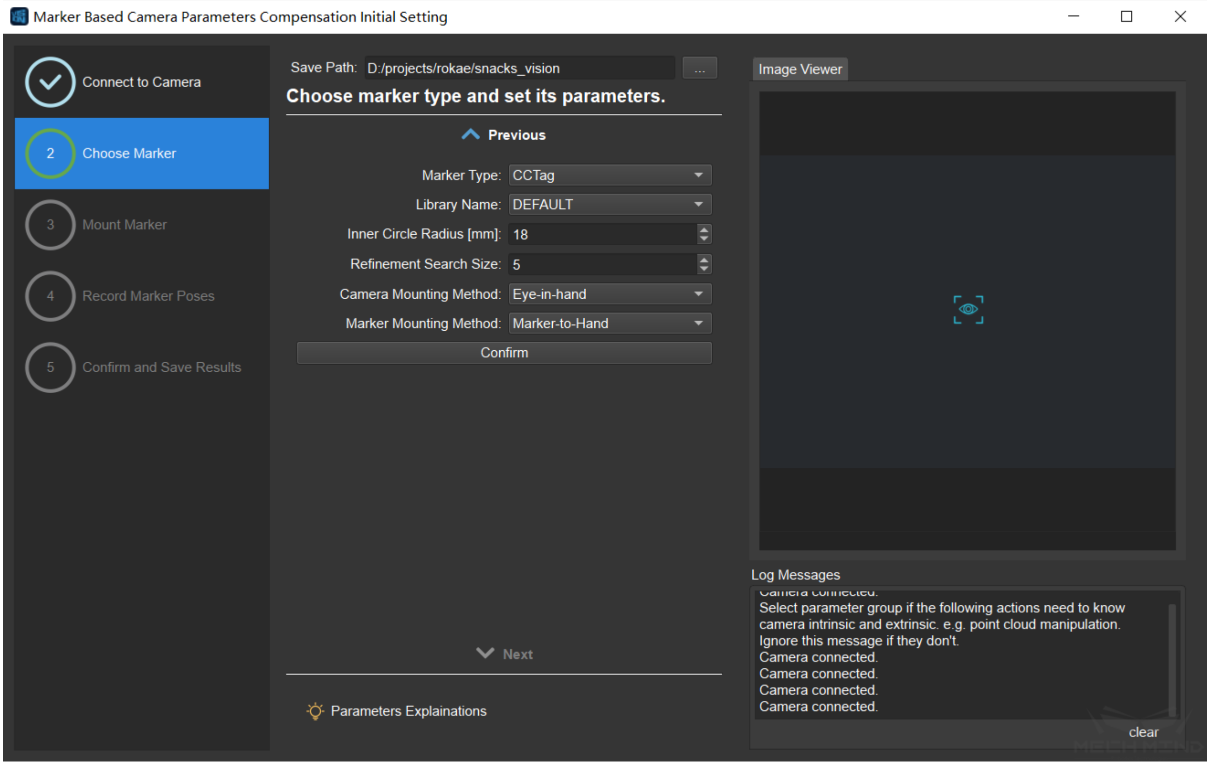
Click *Parameters Compensation* and select *Make Reference*. The interface is shown below.



User needs to set the save path, connect to the camera, and select the existing camera extrinsic parameters group. Click *Capture Once* after connecting to confirm that camera connection is correct. Then click *Next*.

Choose Marker

At present, there are not many options for markers, just choose marker type, camera mounting method, marker mounting method and use default value for other parameters. More parameters and types of markers will be added later to suit different scenarios. It is recommended to use CCTag or STag. In this article, the camera mounting method is EIH, the marker type is CCTag, and the marker mounting method is marker to hand . The detailed settings are shown in the figure below. Click *Confirm* and *Next* to the next step.



Note: If the camera mounting method is ETH, marker mounting method needs to be marker in hand.

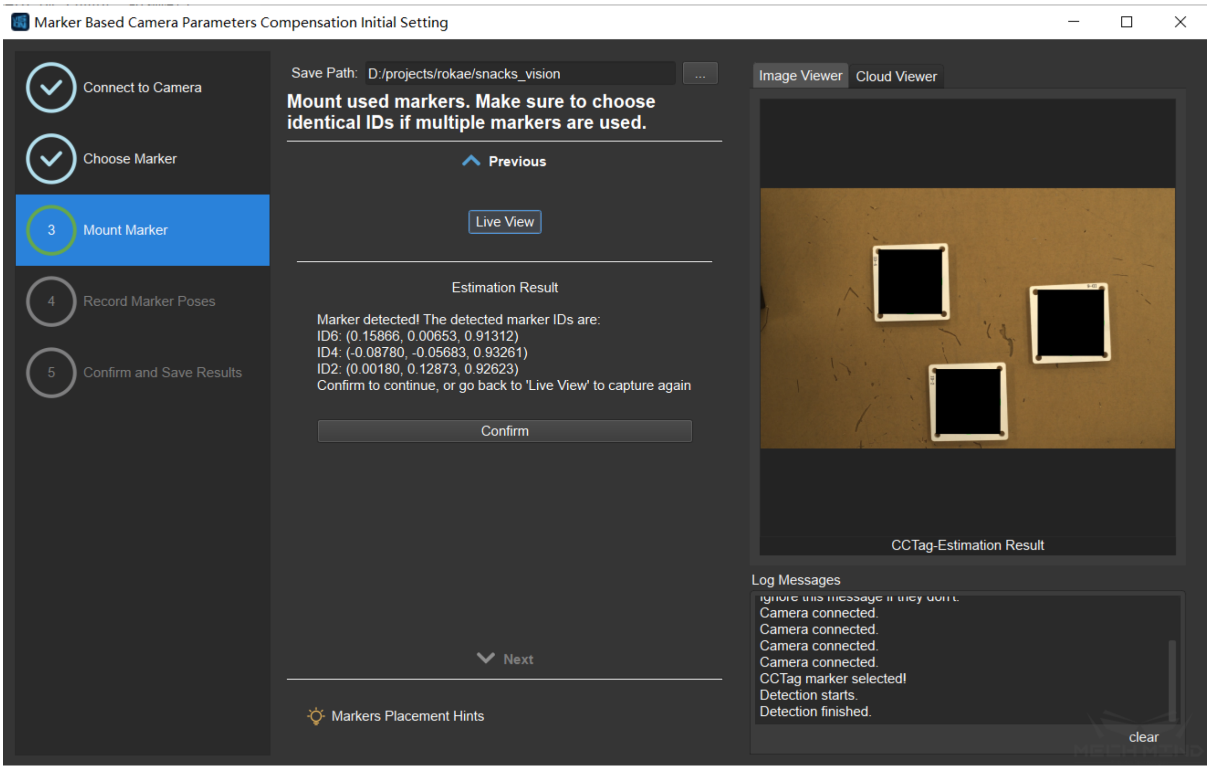
Mount Marker

At this step, the camera will continuously capture images, and the real-time position of the markers can be seen from the image viewer on the right.

The placement rules of the markers are as follows:

- Mount the marker at a suitable position in the work area or on the robot arm. Make sure that the relative position of the marker and the fixed object (ground, robotic arm, etc.) remains unchanged.
- If marker is placed in the work area, it is recommended to use multiple markers with different numbers (ID), and they should be scattered as much as possible in the camera' s field of view.
- If multiple markers are used, try to ensure that any three markers are not in a straight line.

After placing the marker, click *Capture Once*. It will take a few seconds for the software to calculate the estimated position. As shown below.

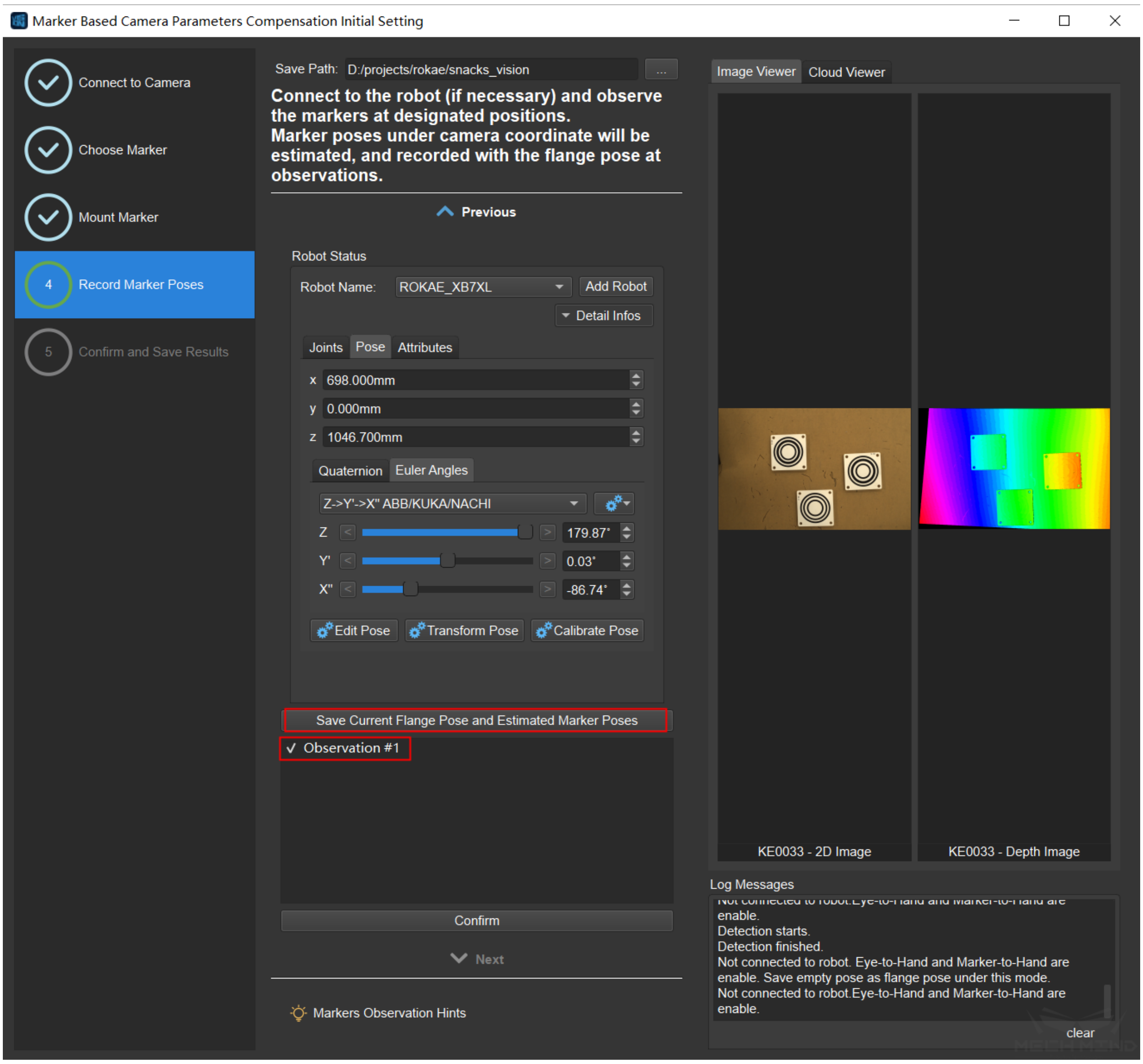


After confirming that the result is correct, click *Confirm* and *Next*.

Record Marker Poses

First, record the robot's initial observation position (i.e. flange pose) and the marker pose at this time

Click the *Save Current Flange Pose and Estimated Marker Poses* button. After completion, it will display "Observation #1". As shown below.

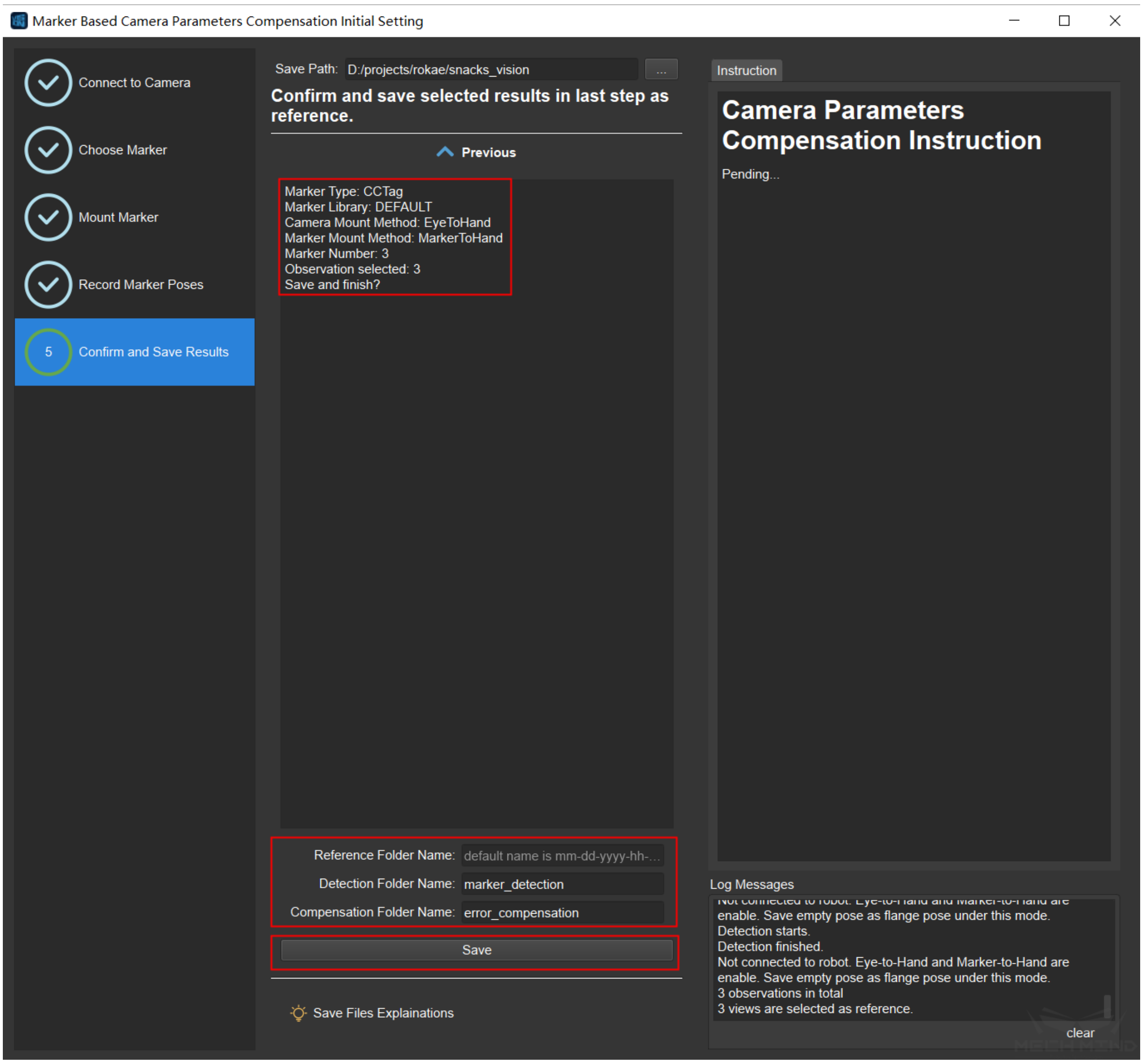


Move the robot to the expected observation position through Mech-Viz or the robot teach pendant and click "Save Current Flange Pose and Estimated Marker Poses" again, one more observation result will be displayed.

Add more observation results and then check whether each result meets the requirements. If there are observation results that do not meet the requirements, uncheck them and they will be discarded. Then click *Confirm* and *Next*.

Confirm and Save Results

The interface is shown in the figure below, and user can modify the name of the output file. Then click *Save*.



Three items are saved:

- Parameters compensation folder: ParamsCompensation.

  Each reference will be saved in a folder named in the form of "date-trial". There can be multiple reference for the same project, which can be selected and used according to requirements.

  flange_*.json records a single flange pose, which is used to send to the robot motion trajectory points later using *Read Poses from File* Step.

  flanges.json records the flange poses of all observation positions. This file is planned to be used to send motion trajectory points to the robot through other Steps, but it is not used at present.

reference.json records all the information and observation results related to parameters compensation. It is used with *Validate and Calc Parameter Compensation* Step to compare observation results and calculate compensation.

observation_*.json records the estimated results of subsequent observations.

offset.json records the compensation data generated by *Validate and Calc Parameter Compensation* Step, which is used to compensate the pick points later using *Read Poses from File* Step.

> **Attention:** The pick points to be compensated must be based on the camera coordinate system. If they are in the robot coordinate system, coordinate transformation should be carried out before compensation.

- Mech-Vision project————Marker detection: marker_detection.

  This Mech-Vision project uses Mech-Viz to call the vision service to detect the position of the marker, and save the estimated result of the position of the detected marker to the file observation_*.json.

- Mech-Vision Project————Error analysis: error_compensation.

  This Mech-Vision project uses Mech-Viz to call the vision service to calculate the compensation. By reading the marker pose in the observation_*.json file, comparing it with the initial marker pose and calculating, the compensation is obtained and saved in the offset.json file.

7.2.3 Use the Generated Projects and Files for Error Compensation

Mech-Vision Projects

In addition to the original project, the two generated projects, marker_detection and error_compensation, need to be added.

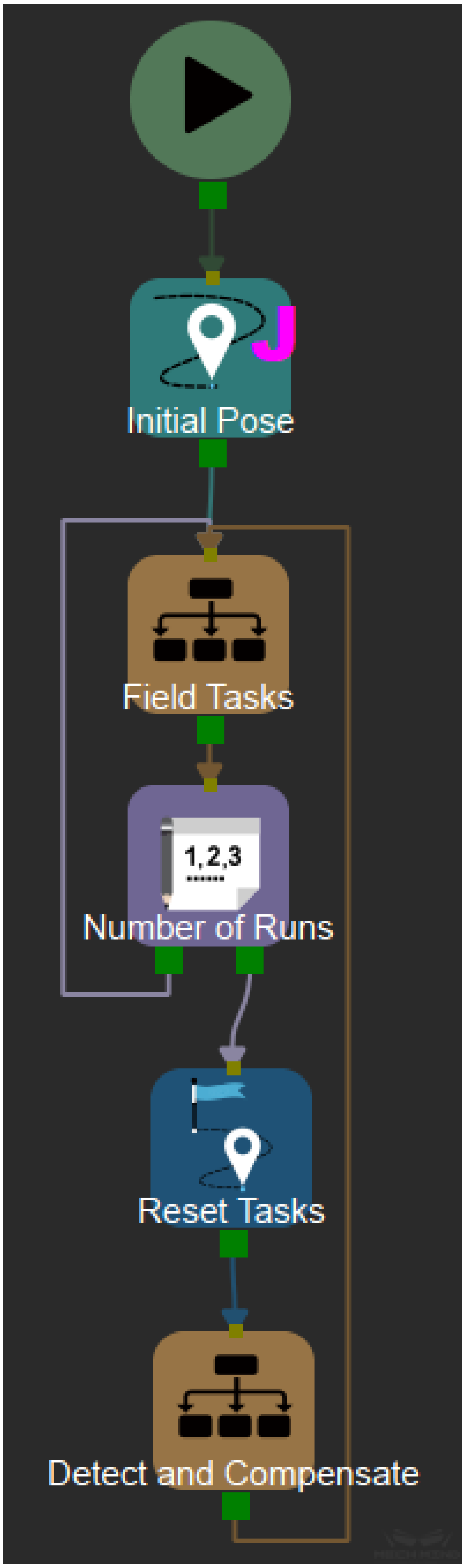
The marker_detection project needs to set the camera IP and camera parameters group, and the path to save the images.

In the *Detect Fiducial Markers* Step, user needs to set the parameters compensation folder ParamsCompensation and the reference target (i.e. the folder named in the form of “date-trial” in the parameter compensation folder).

Similar to :guilabel: *Parameters Compensation* Step, the error_compensation project also needs to set the parameter compensation folder ParamsCompensation and reference target.

Mech-Viz Project

The Mech-Viz project can be set as shown in the figure below. First place the original project in the Field Task mission.

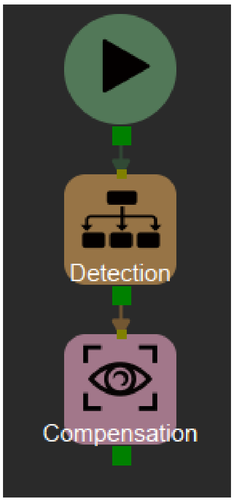

Initial Pose
Field Tasks
1,2,3
Number of Runs
Reset Tasks
Detect and Compensate

Then add counter, reset tasks, and detect and compensate.

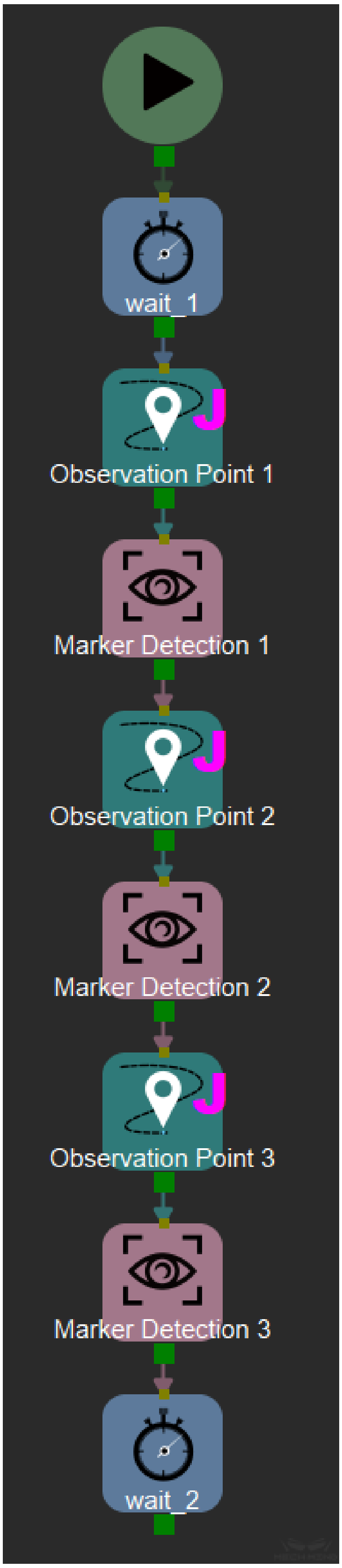
The counter is set according to the actual situation. (That is, how many times the field task runs before compensate once)

Select the counter as the task to be reset for the Reset Tasks Skill.

The content of detect and compensate mission is shown in the figure below. There are two modules of detection and compensation.



The content of detection mission is shown in the figure below.


wait_1
Observation Point 1
Marker Detection 1
Observation Point 2
Marker Detection 2
Observation Point 3
Marker Detection 3
wait_2

Poses at the observation points (Move Skill) are set as the pose at the initial observation point. Pose is recorded in the flange_*.json file mentioned above. The number of observation points depends on the number of markers.

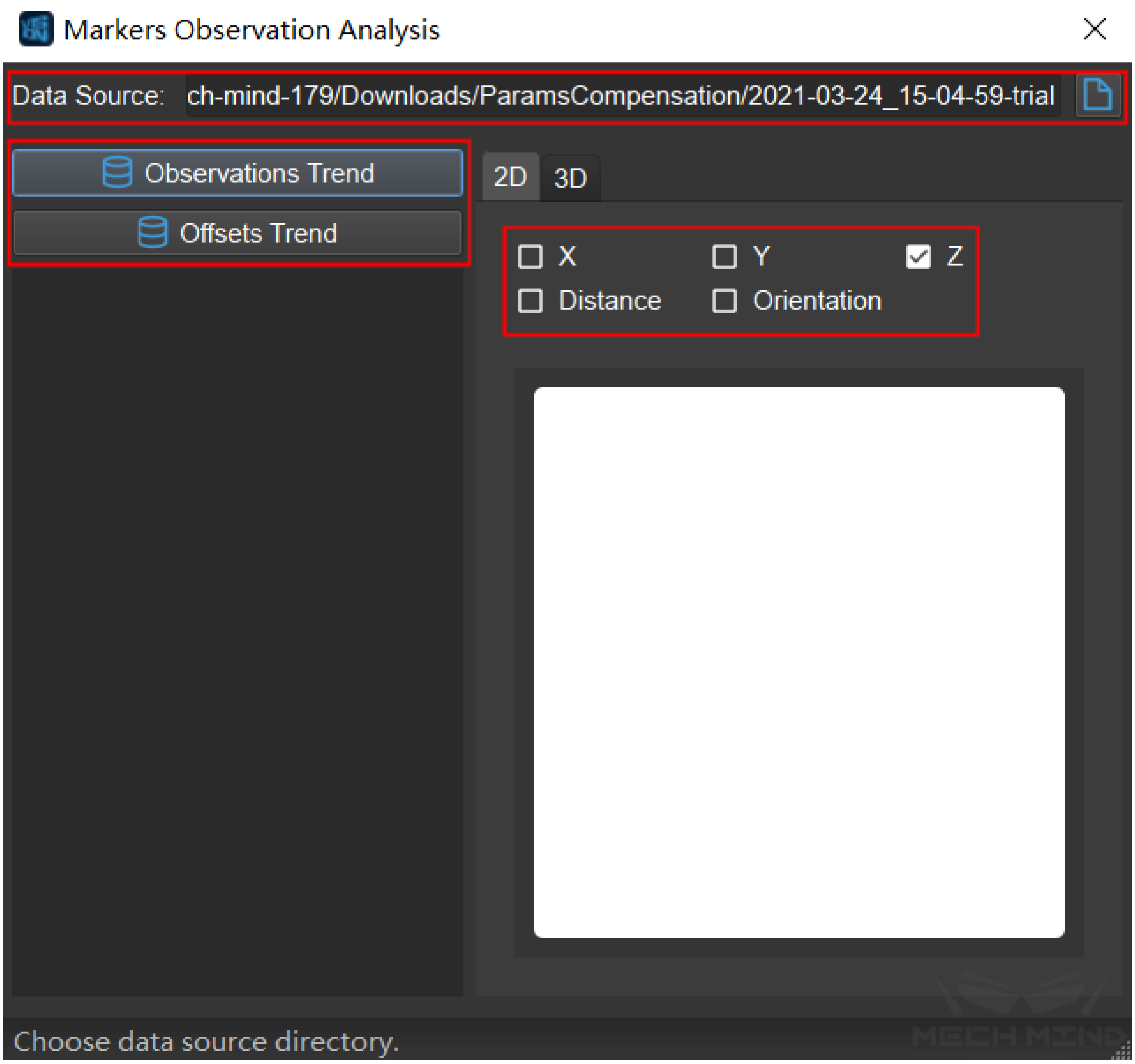
Visual_look Skill needs to call the marker_detection vision service. The number of visual_look Skills also depends on the number of markers.

Compensation (visual_look Skill) needs to call the error_compensation vision service.

Now, run the Mech-Viz project to perform camera parameters compensation in the project.

7.2.4 Error Analysis

Error analysis is to visualize the generated result of camera parameters compensation, and the trend and distribution of the error can be seen through the chart. Select Error Analysis in the parameters compensation tool to enter the error analysis page. As shown below.



First, select data source, which is the folder named in the form of “date-trial”.

- Observations Trend: Observe the trend of the errors of the markers’ poses. The data is from the observation_*.json file in the folder.
- Offsets Trend: The trend of the compensation values. The data is from the offsets.json file.

7.3 Project Assistant

This section will introduce the tools in **Project Assistant**.

Click the section below to learn about **Parameter Recipes**.

Parameter Recipe Configuration

Click the section below to learn about **Data Storage**.

Data Storage

Click the section below to learn about how to select a **Scene Point Cloud**.

Scene Point Cloud

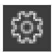
7.3.1 Parameter Recipe Configuration

Application Scenarios

In on-site applications, there are often scenarios of “using the same Mech-Vision project to process different workpieces (items)”.

In scenarios of this type, Mech-Vision project programming can be the same, but some parameters change. In this case, **the parameter recipe can be dynamically modified to adapt the project to different working scenarios**.

Instructions

1. After opening a project, click  in the **Project Assistant** to enter the **Parameter Recipe Editor**.
2. Add and delete recipes.
   - Click *Add Compensation* to add a recipe for each model of the object to pick.
   - Fill in a unique formula name, as shown in *Figure 1*.
   - Click *Delete Recipe* to delete selected recipe (hold Ctrl for multiple selections).

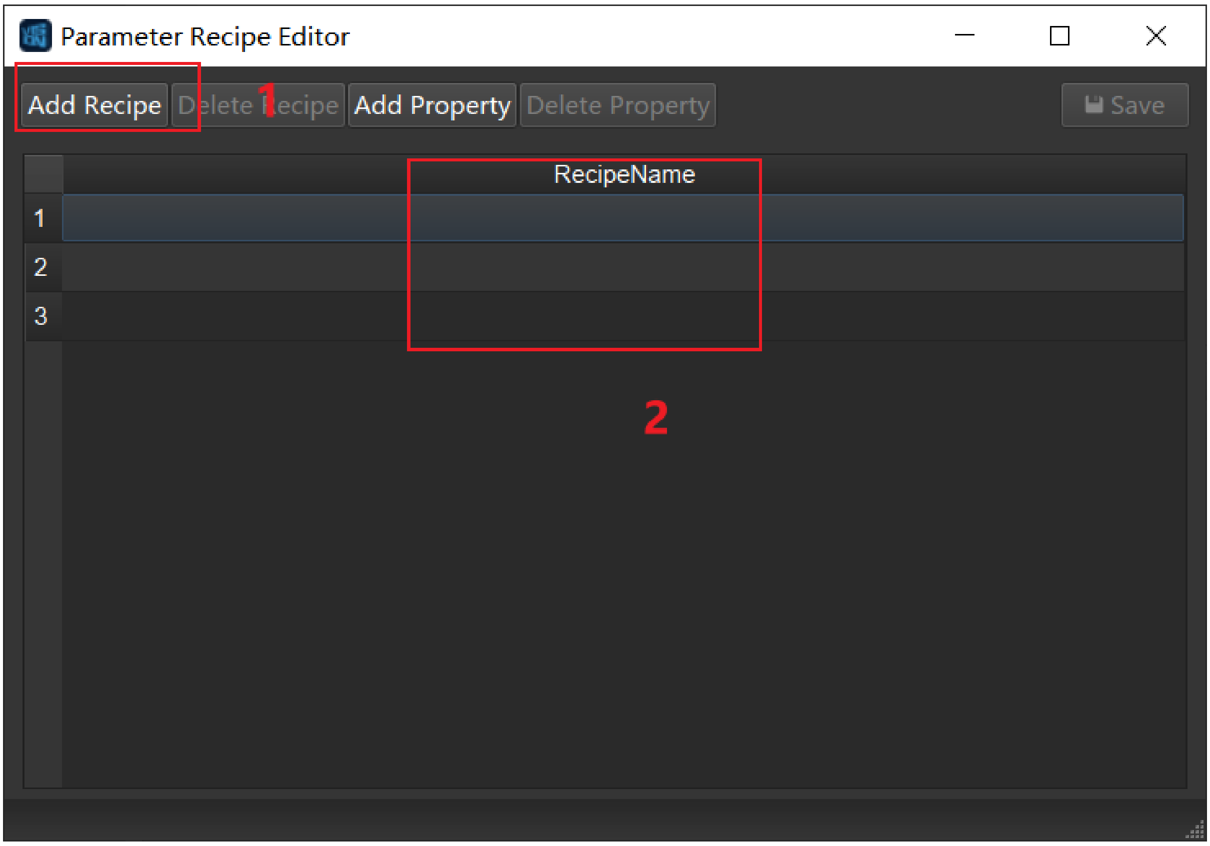


Figure 1. Add Recipe

3. Add, modify, delete properties.
   - Click *Add parameter*.
   - In the pop-up window **Properties Selection**, select the properties needed

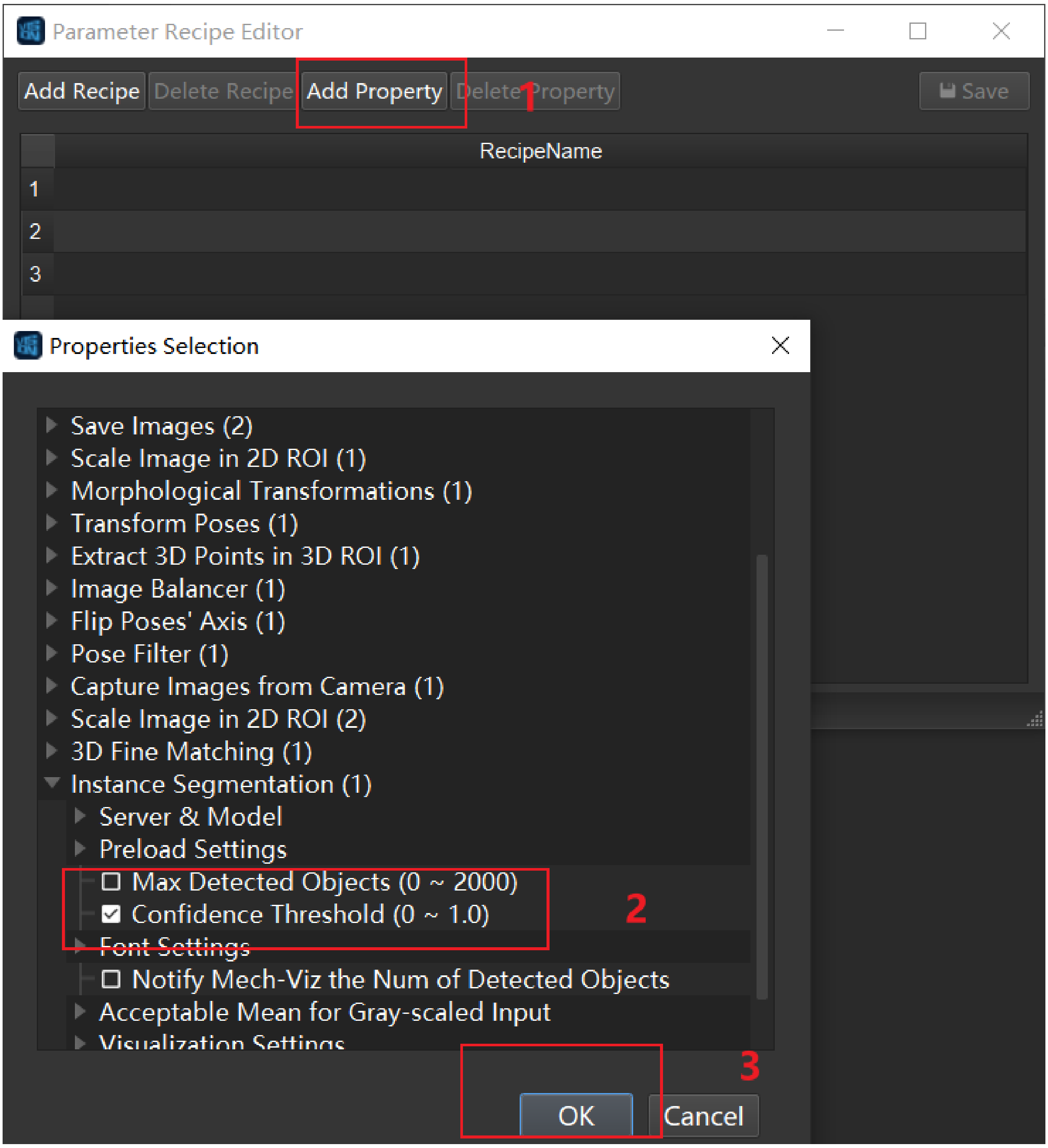


Figure 2. Add Properties

- Click *OK* to automatically obtain the initial status of the added step properties (input parameter type, input form, upper and lower limits, etc.), as shown in *Figure 3*.

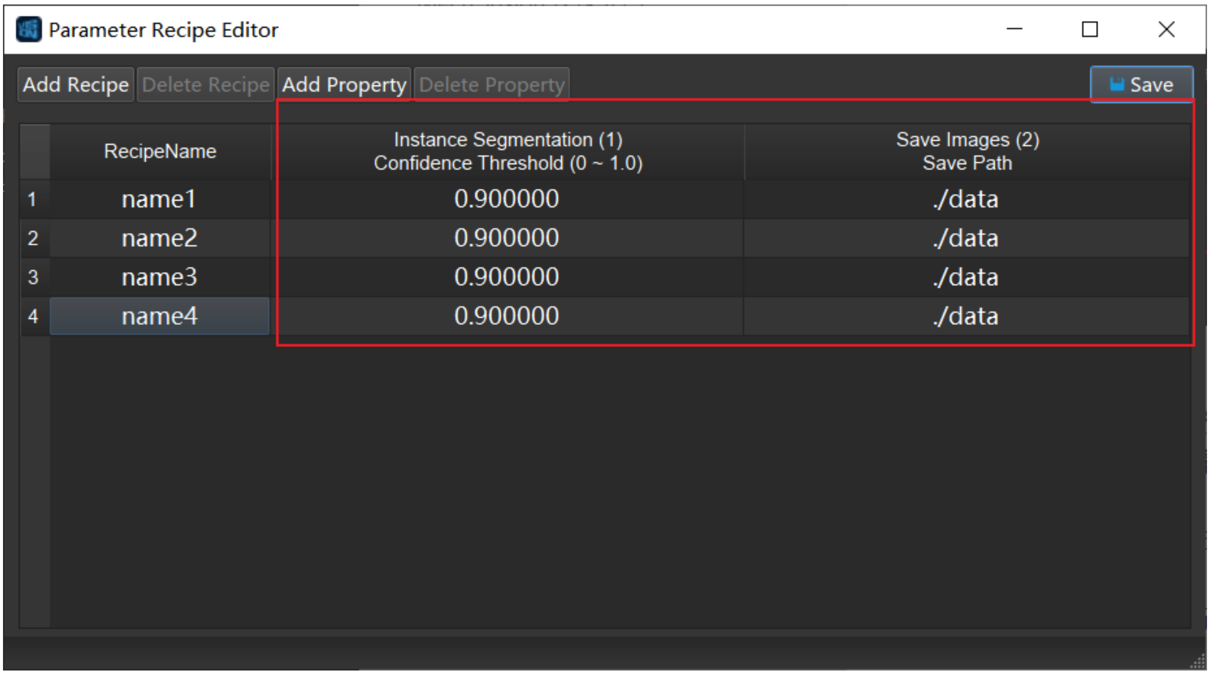


Figure 3. Initial Status Obtained

- Double-click the parameter value in the list to modify the corresponding parameter. The modified parameter value shall not exceed the limits, be of incompatible format, etc. Example: for instance segmentation confidence threshold, the value should be from 1 to 1, and for save path, a valid file path should be entered.
- Click *Delete Parameter* to delete selected parameter (hold Ctrl for multiple selections).

4. Colum copy and row copy.
   - Column copy: *Select the content in a column that needs to be copied to another column (hold Ctrl for multiple selections; click the parameter name to select all in this column) → right-click the selected area → select the target column to copy to.*

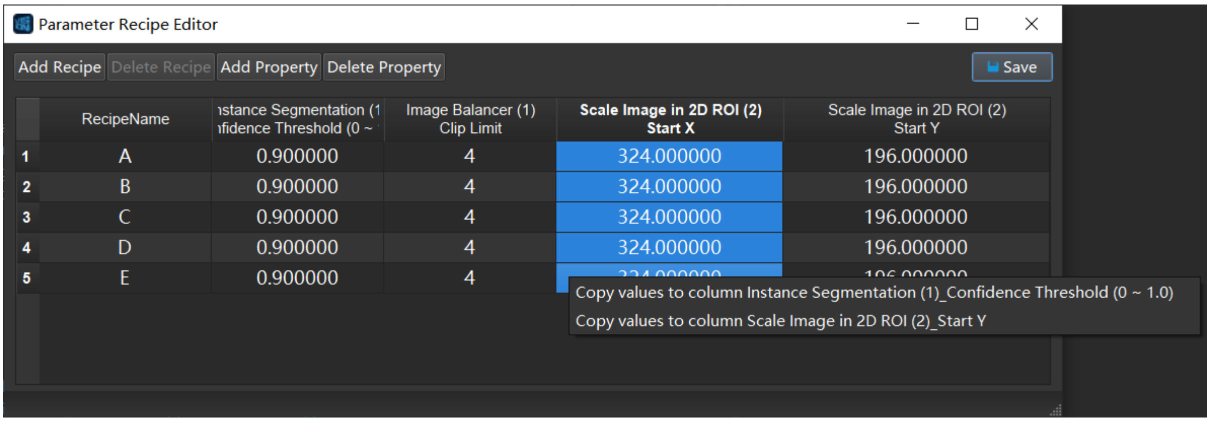


Figure 4. Column Copy

   - Row copy: *Select the content in a row that needs to be copied to another row (hold Ctrl for* *multiple selections; click the number to select all in this row)* → *right-click the selected area* → *select the target row to copy to.*

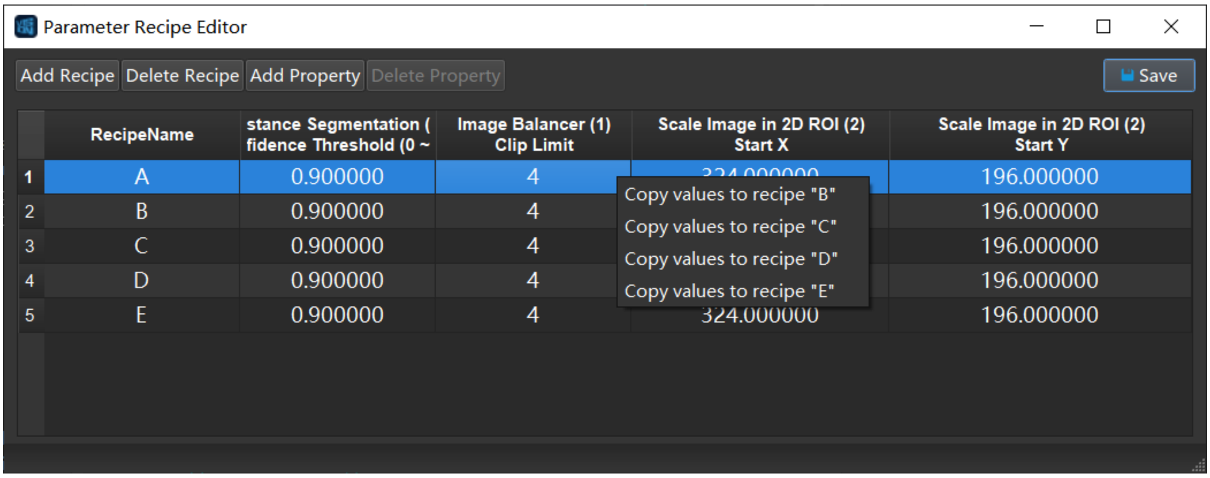


Figure 5. Row Copy

5. Save and switch recipes.
   - Save settings: When done setting parameter recipes, click *Save*. The prompt "Save successfully!" means that all settings have been successfully saved.
   - Switch recipe: select the recipe name of the corresponding model of the object to pick in the **drop-down menu** in **Project Assistant**. When running the project, the set properties will take effect.

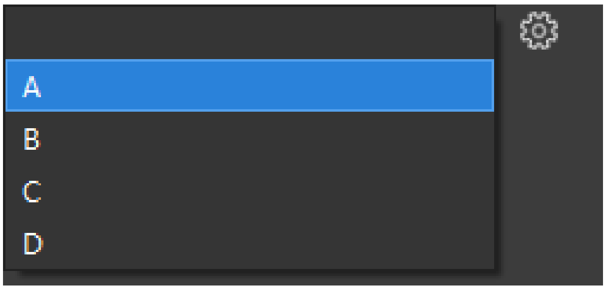


Figure 6. Switch Recipe

Note: For automatic switching of parameter recipe, please start Mech-Interface in Mech-Center and connect by the standard interface. For detailed operation instructions, please refer to mech_interface.

7.3.2 Data Storage

This tool is used to save on-site data for efficient tracing of issues.

Instructions

1. After opening the project, add the step **Save Images and Step Properties** in the project, and connect the step (input port 1 is the depth map, port 2 is the 2D image) as shown in Figure 1.

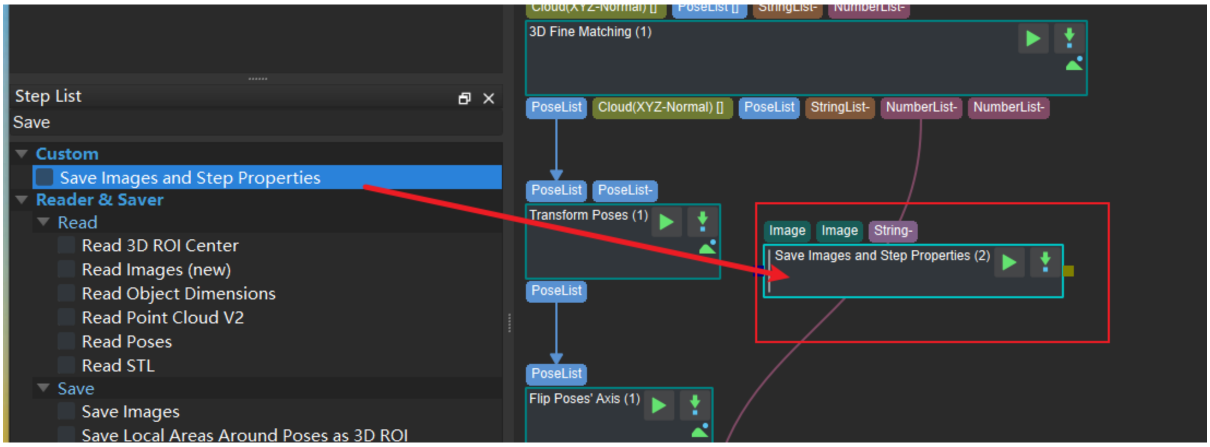


Figure 1. Add Step **Save Images and Step Properties** and Connect

1. Relevant settings can be adjusted in parameters. **It is recommended to use the default settings except for Max Subdirectories' Number**.

   - Directory Settings: The directory path for saving images.
   - Create Subdirectories by Date: After checking, a subdirectory named after the current date will be created.
   - Max Subdirectories' Number: The maximum number of subdirectories (one subdirectory each day).
   - Save Contents Settings: Whether to save the camera parameters, flange pose.
   - Max Images' Number: Set the maximum number of images saved. If the number of images saved exceeds the maximum, the sequence number of the next picture will be reset to 0.

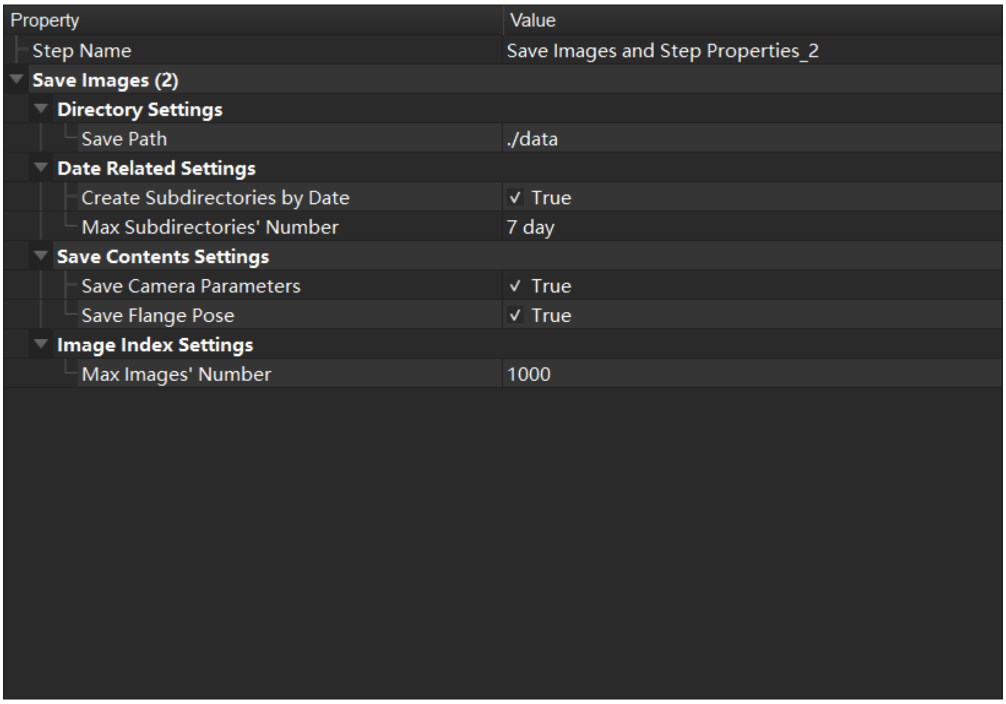


Figure 2 . Set related parameters

2. The *Parameter Recipe* used the running the project is saved by default for data playback. In addition to the parameter recipe, data that needs to be saved can be added by the steps shown in Figure 3.

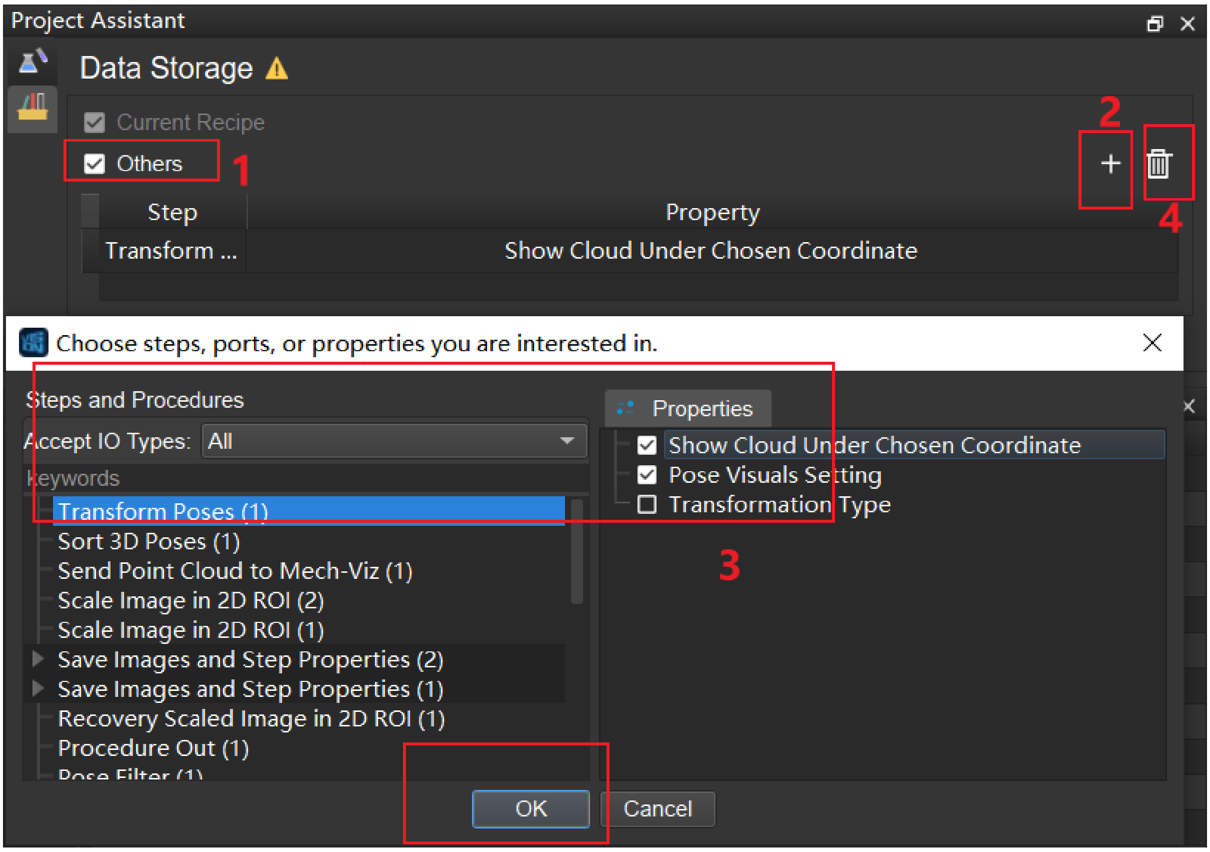


Figure 3. Add Step parameters

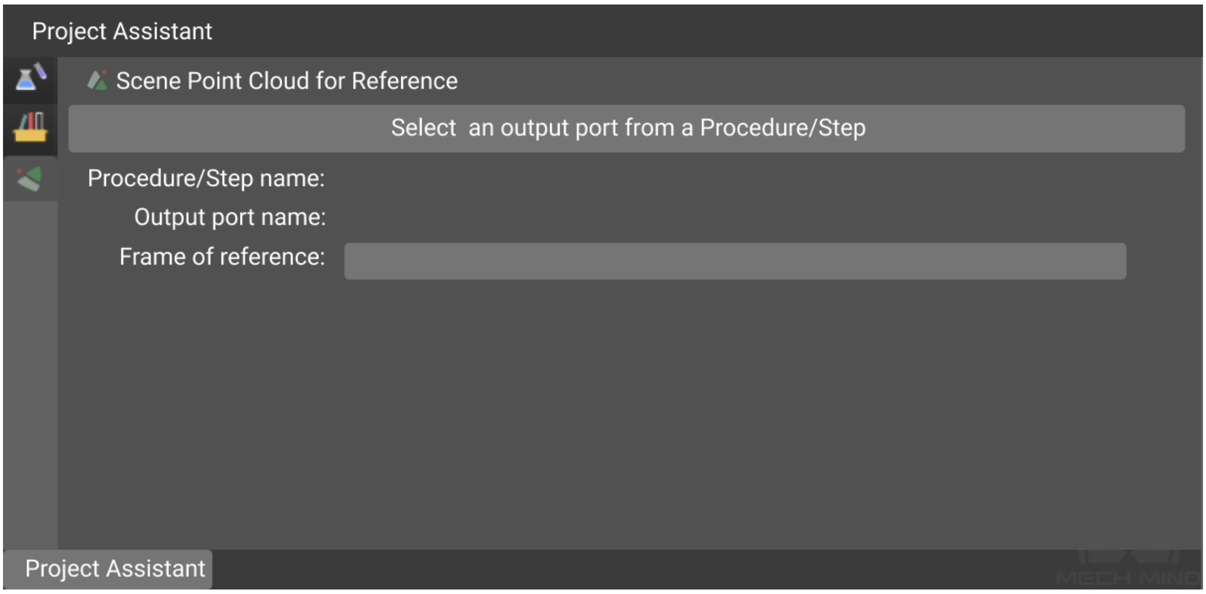
1. In *Project Assistant* → *Data Storage* window, check *Others*.
2. Click + to pop up the window **Choose steps, port, or parameters you are interested in**.
3. Check the parameters that need to be added, and click *OK* to confirm.
4. Click 🗑 to delete a selected parameter.

7.3.3 Scene Point Cloud

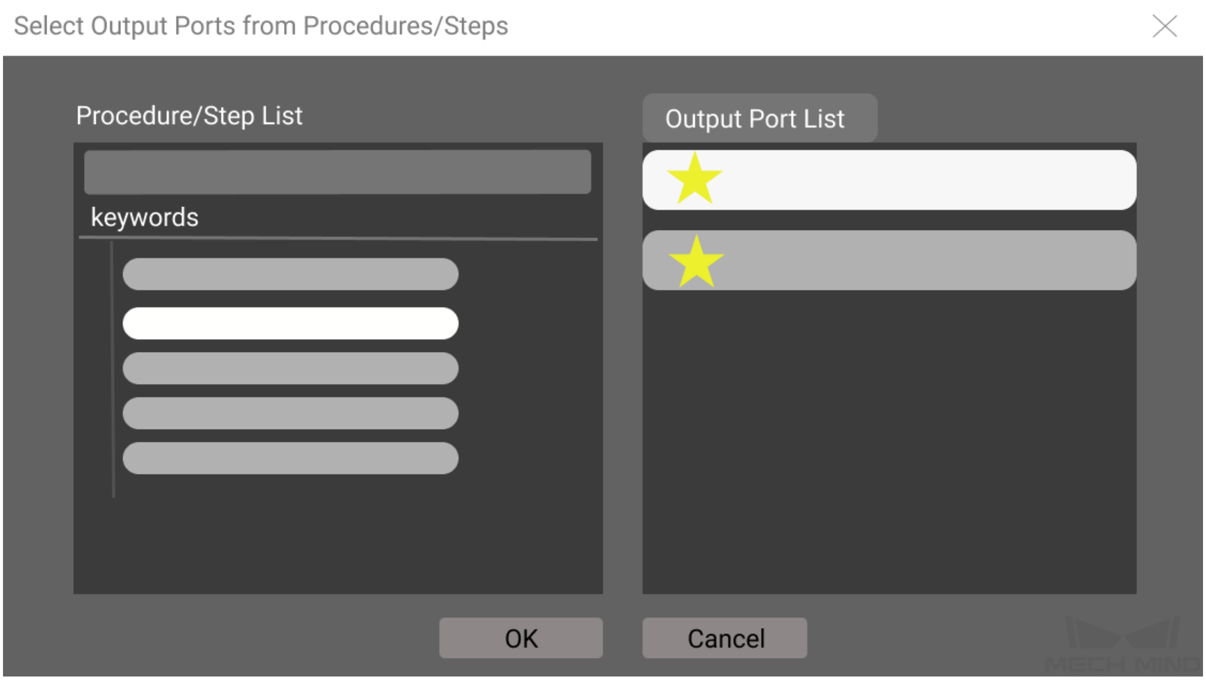
The scene point cloud consists of all points in the camera’ s field of view, and can be used as a reference for setting 3D ROI in relevant Steps. This tool is used to select a scene point cloud before setting a 3D ROI in relevant Steps.

Instructions

1. Open **Project Assistant** and click on *Select an output port from a Procedure/Step* as shown below.



2. A pop-up window will appear. Select the Step which outputs the scene point cloud on the left side, and select the output port on the right side, as shown below. You can also type the keyword above the list to search for the Step you need.



Attention: It is not recommended to select *Capture Images from Camera* as the Step which outputs the scene point cloud, which may slow down the computing speed of the project. Normally, the Step *From Depth Map to Point Cloud* is selected. You can also select other appropriate Step according to actual needs.

Note: The icon  represents that the output port is connected with an input port of a succeeding Step, and the icon  represents that the output port is not connected with a succeeding Step. However, either port can be selected as the output port of the scene point cloud.

1. Click *OK* to finish setting.
2. After seleting the Step and output port, you can also choose the **Frame of reference** according to actual needs in **Project Assistant**.

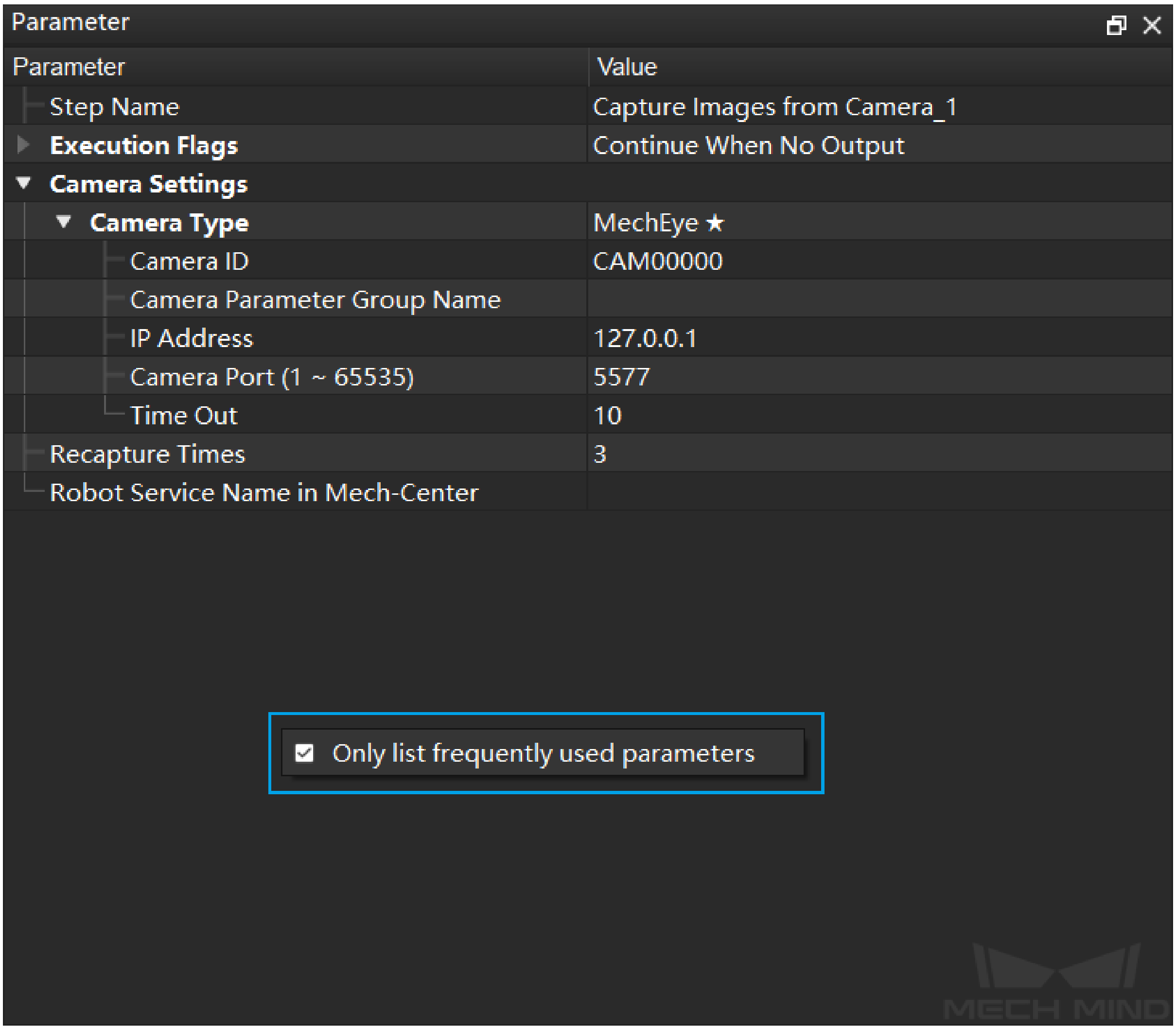
Note: Please refer to *Instructions for Setting 3D ROI* to learn more details about setting up 3D ROI.

7.4 Set Static Background for Project

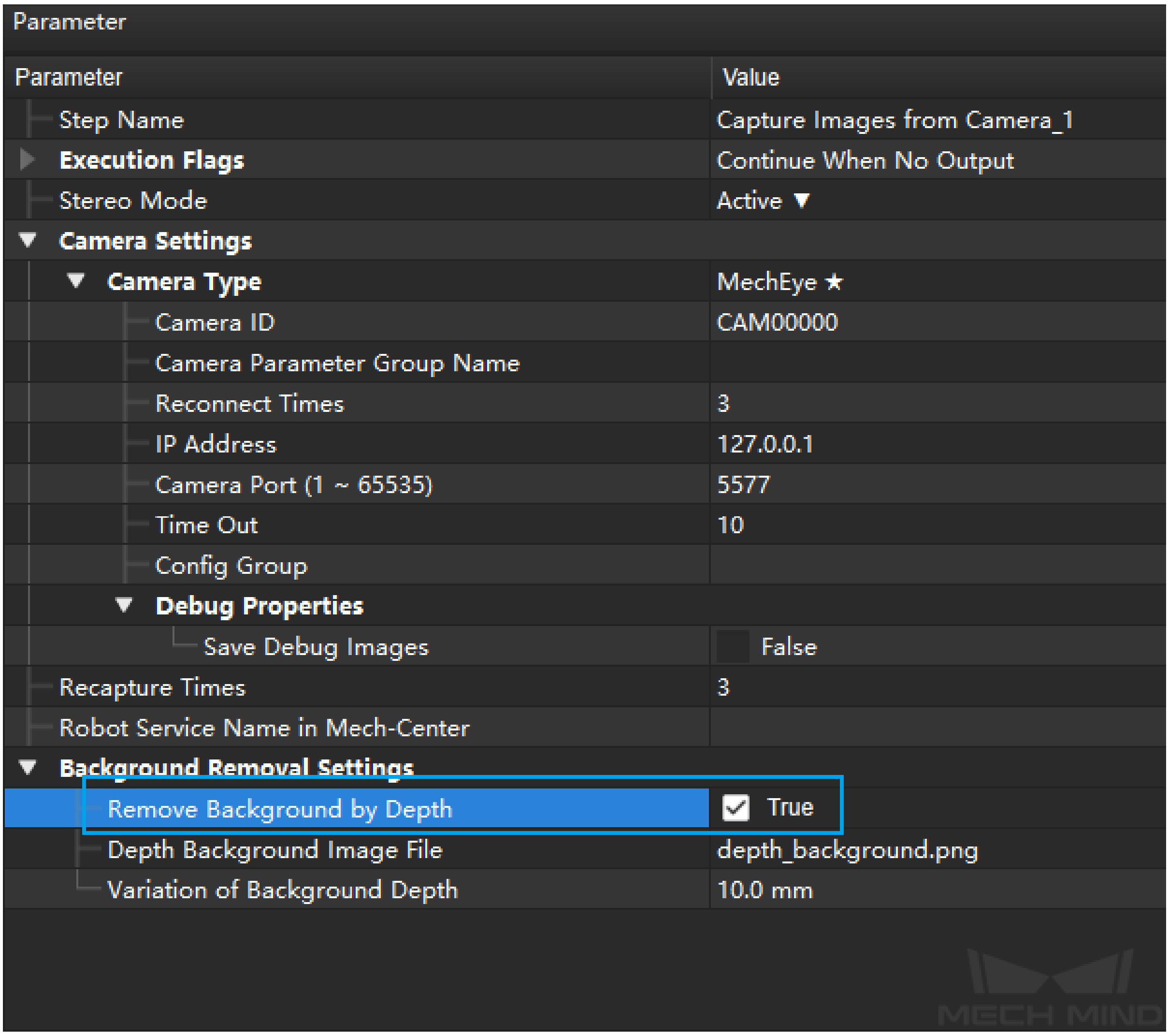
- Setting a static background and obtaining a depth map of it can avoid the interference of the background in following Steps and facilitate the calculation of the object height.

7.4.1 Instructions for Setting Static Background

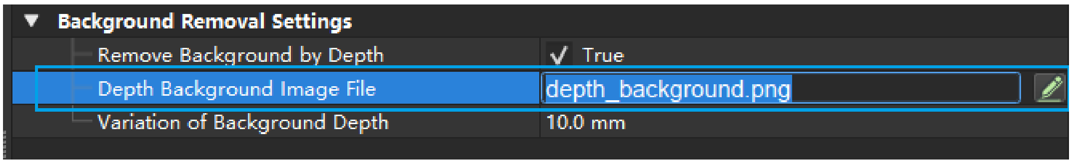
1. Open Mech-Vision project and select the Step *Capture Images from Camera*. Then right-click on the empty space in the Parameter section and uncheck *Only list frequently used parameters* as shown below.



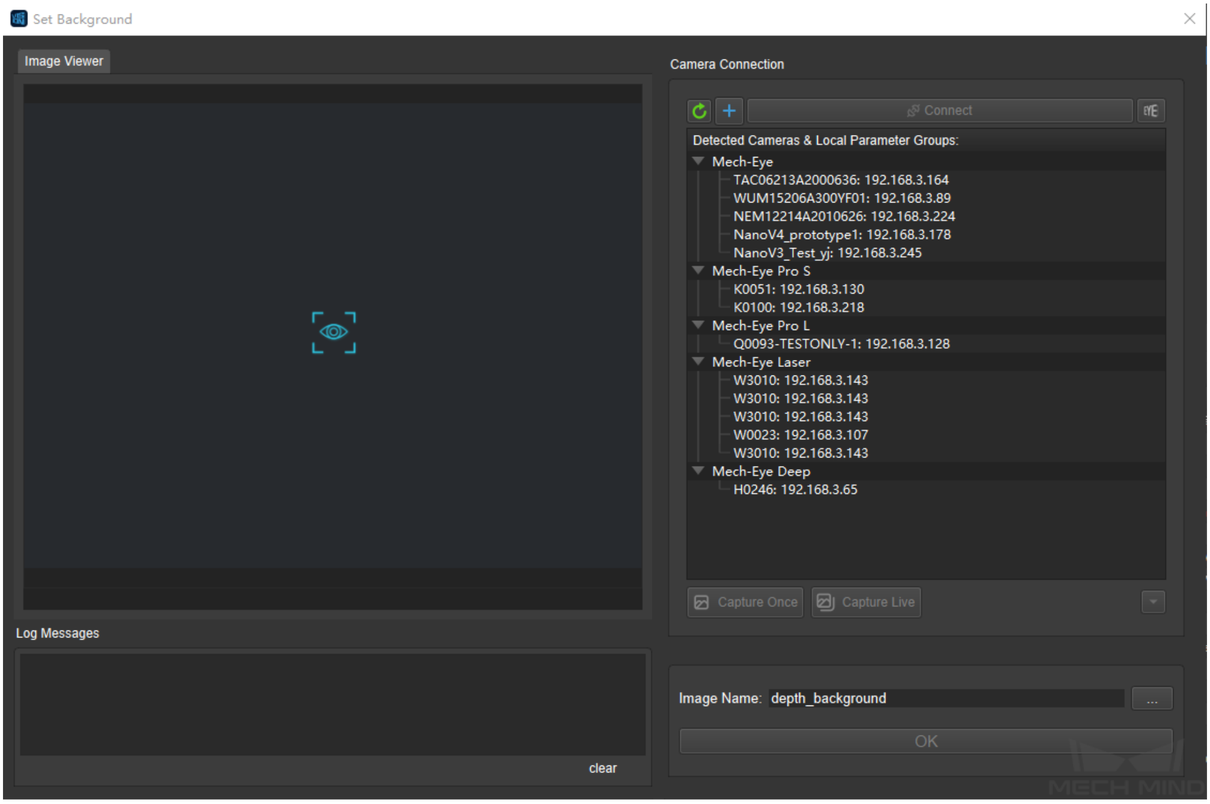
2. Change the value of *Remove Background by Depth* in the *Background Removal Settings* from “False” to “True”, as shown below.



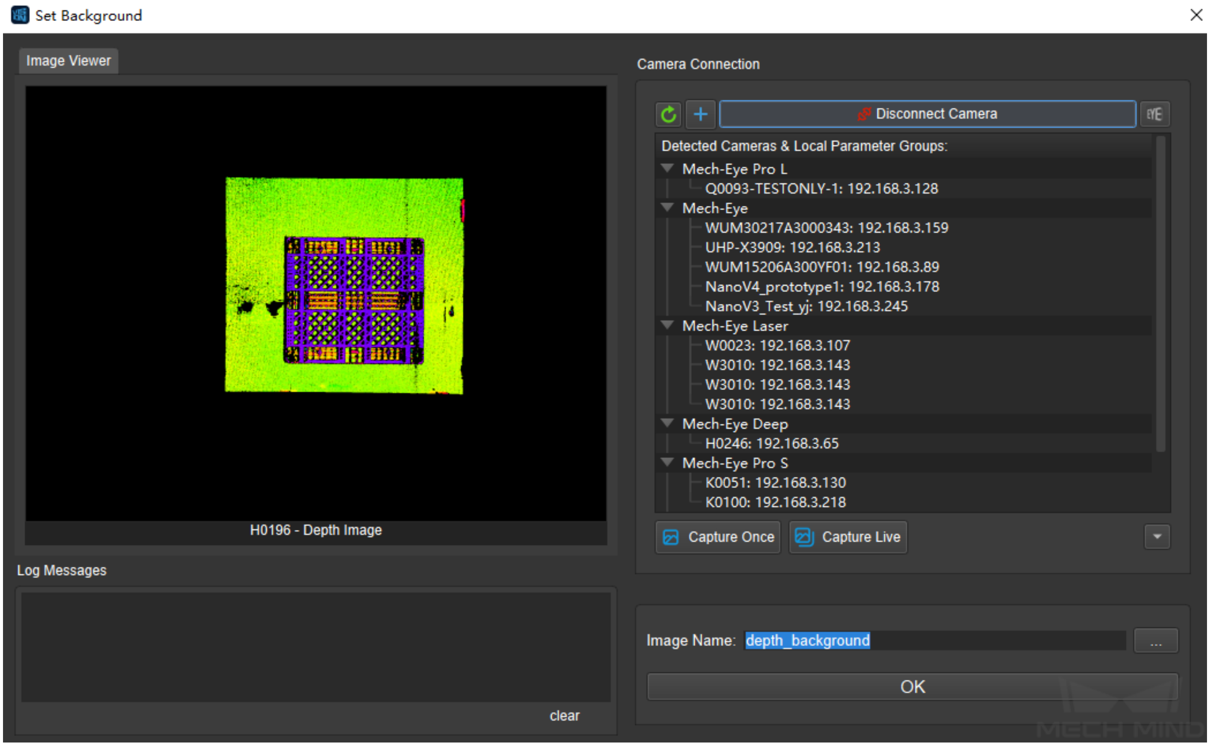
3. Click the edit button next to *Depth Background Image File* to open the Set Background tool, as shown below.



4. The Set Background tool includes 4 sections: Camera Connection, Image Viewer, Log Messages, and Image Saving, as shown below.



5. To set the background, double click the camera number or select the camera and click *Connect Camera* to connect it. Click *Capture Once* or *Capture Live* to take photos after the connection is done, and click *OK* to save the picture after obtaining the background image, as shown below.



7.5 Set ROI for Project

- 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 speed of the entire Mech-Vision project can be increased.

7.5.1 Instructions for Setting 2D ROI

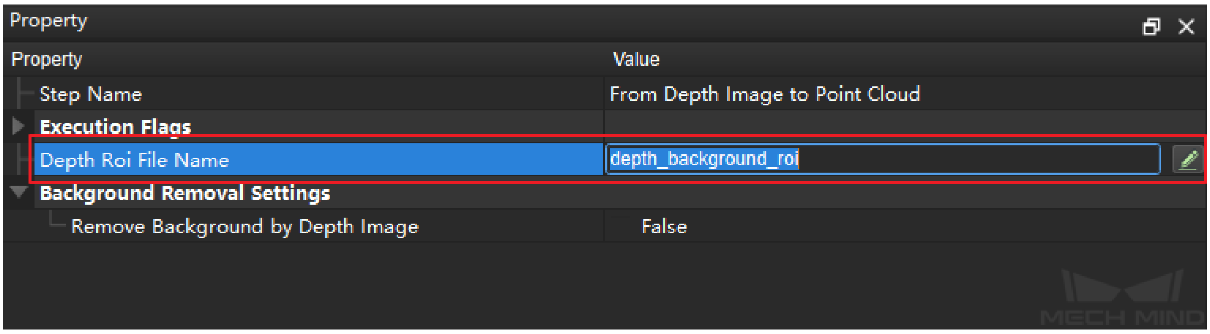
Steps related to 2D ROI include:

Scale Image in 2D ROI

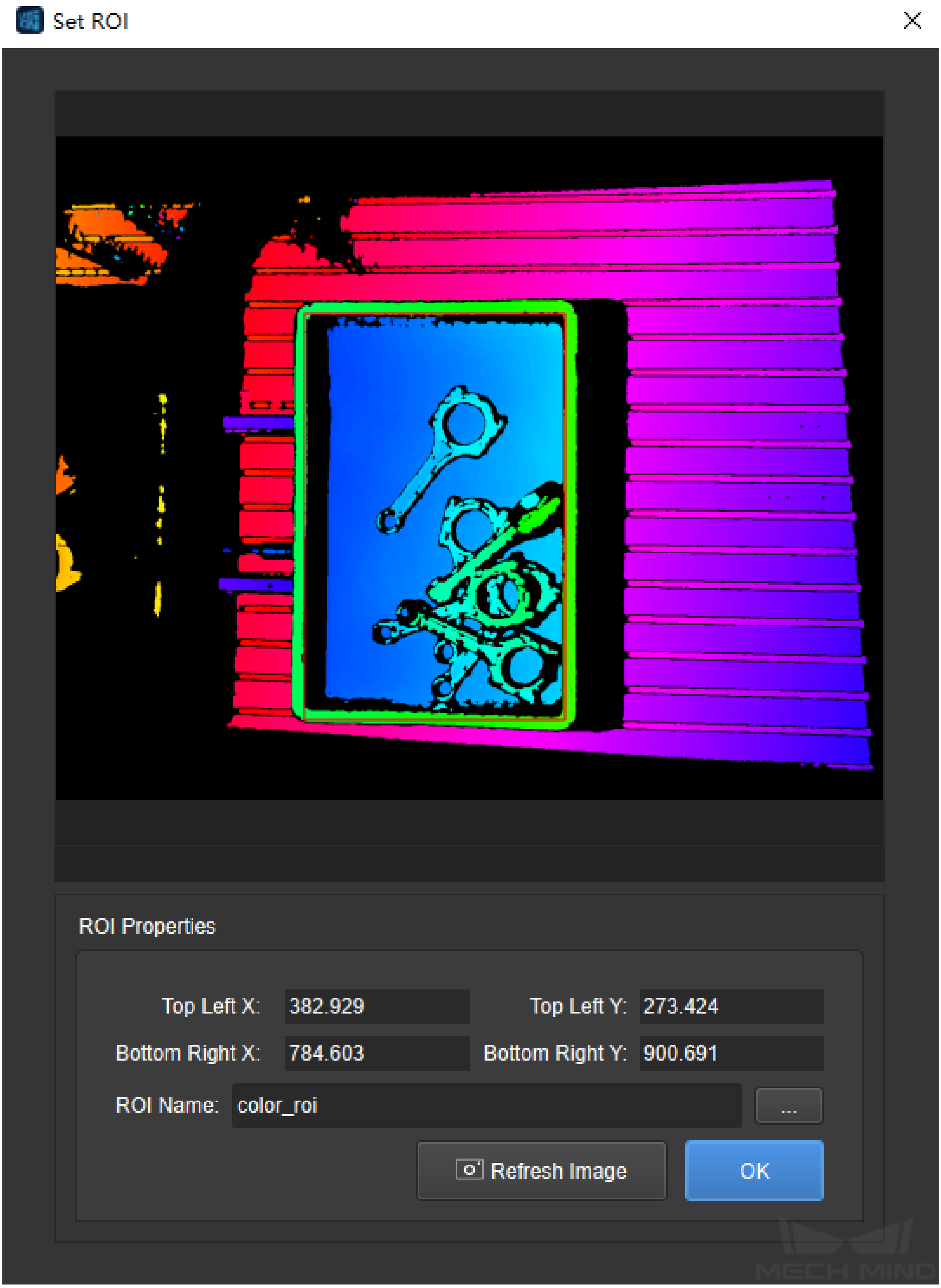
Segment Depth Image

From Depth Map to Point Cloud

1. For the Steps which need to set the 2D ROI, click on the edit button  next to *Depth Roi File* or *Color Roi File* to enter the Set 2D ROI tool, as shown below.



2. In Image Viewer area, hold down the left mouse button and drag to select a ROI of the color or depth image, and click the left mouse button again to complete, as shown below. If you need to re-select a ROI, please click the left mouse button again to re-select.


Set ROI
ROI Properties
Top Left X: 382.929
Top Left Y: 273.424
Bottom Right X: 784.603
Bottom Right Y: 900.691
ROI Name: color_roi
...
Refresh Image
OK

3. The parameters of the selected ROI will be displayed in the ROI Properties section. Click *OK* to save and exit.

7.5.2 Instructions for Setting 3D ROI

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

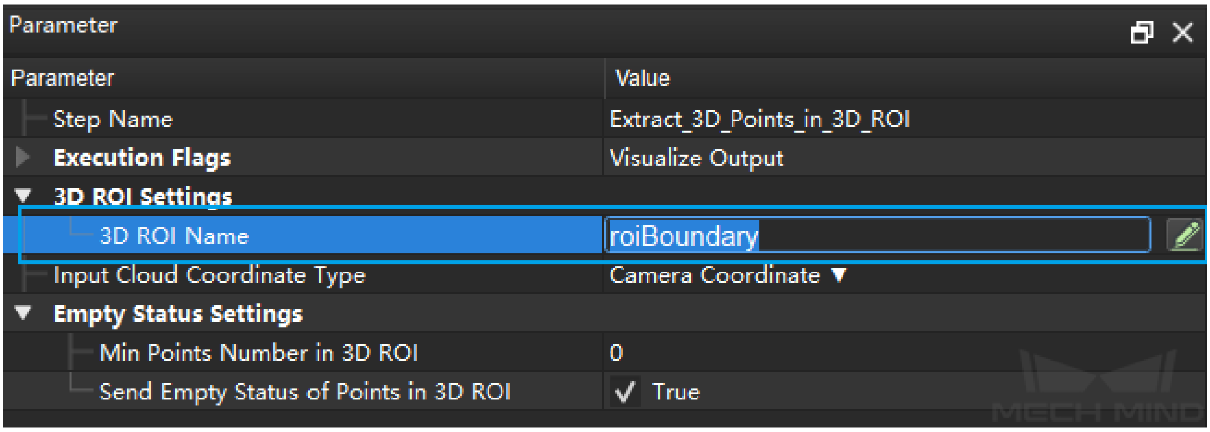
Read 3D ROI Center

Save Local Areas around Poses as 3D ROI

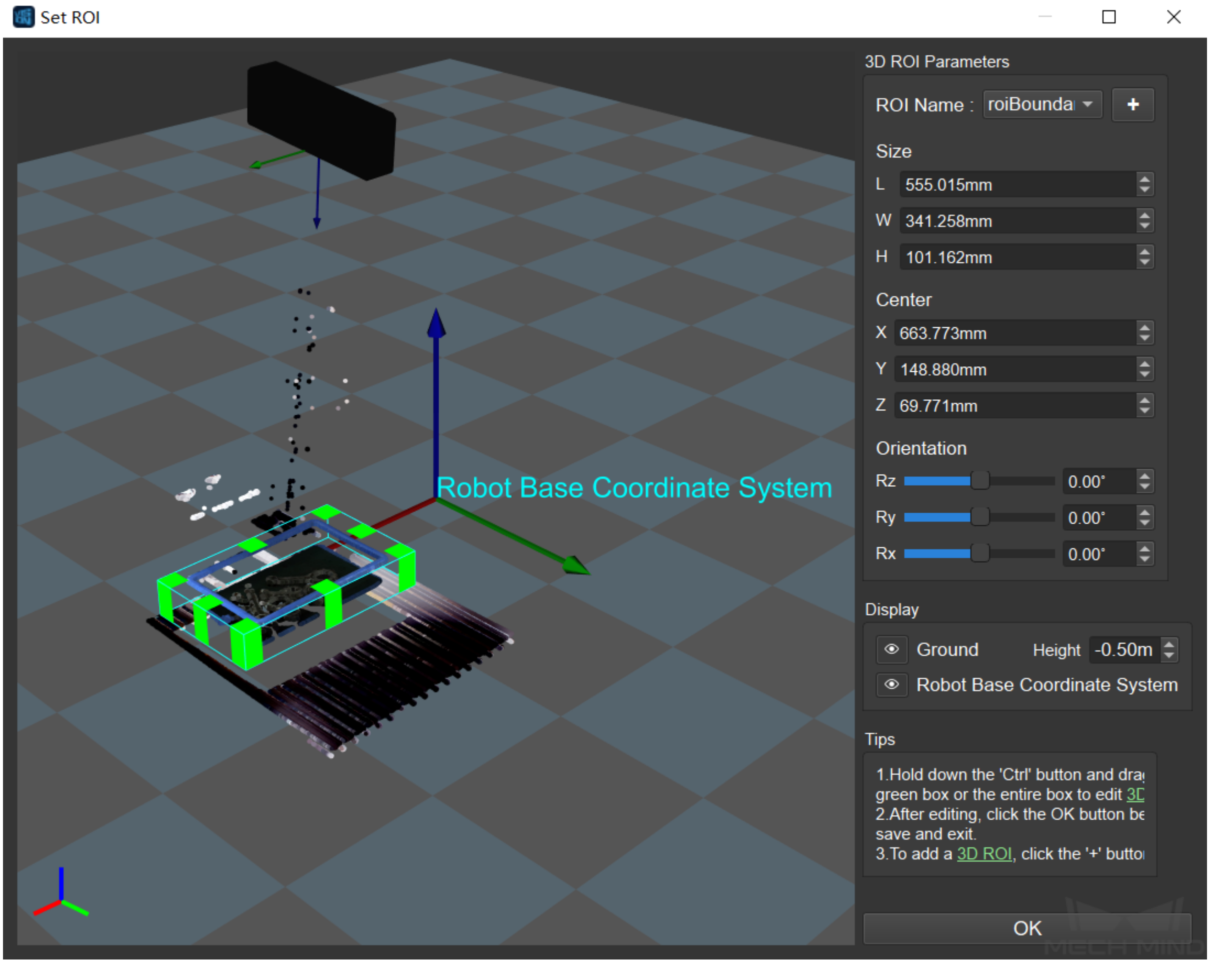
Extract Empty Areas in Depth Image Within 3D ROI

Attention: 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. For detailed settings, please refer to *Scene Point Cloud*.

1. Select the Step which you want to set a 3D ROI, and then go to **Parameter** and click on the edit button next to the 3D ROI Name in the 3D ROI Settings.



2. A window for setting 3D ROI will pop up. It consists of 4 parts: Point Cloud Display, 3D ROI Parameters, Display, and Tips, as shown below.



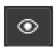
Point Cloud Display Area

- Includes robot base coordinate system, camera coordinate system, 3D ROI, and target point cloud.

3D ROI Parameters

- Includes the name, size, center, and orientation of the 3D ROI.

Display

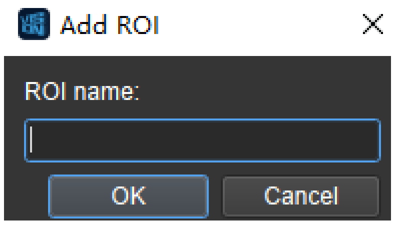
- Adjusts the display status of ground and robot base coordinate system. Click on  in Display section to display or hide the ground and robot base coordinate system. You can also configure the height between the origin of the robot base coordinate system and the ground.

Tips

- Provides tips for 3D ROI setting.

3. You can adjust the default generated 3D ROI by holding down the Ctrl key on the keyboard and drag it with the left mouse botton. 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, a new 3D ROI can be created by clicking + 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. To switch different 3D ROIs, please click on ▾ next to the ROI Name and select the 3D ROI you want to edit in the drop-down list.



Attention: If different 3D ROIs are created, please make sure that the one used in the Step corresponds to the appropriate 3D ROI.

7.6 Matching Model and Pick Point Editor

Location: *Menu Bar* → *Toolkit* →*Matching Model and Pick Point Editor*

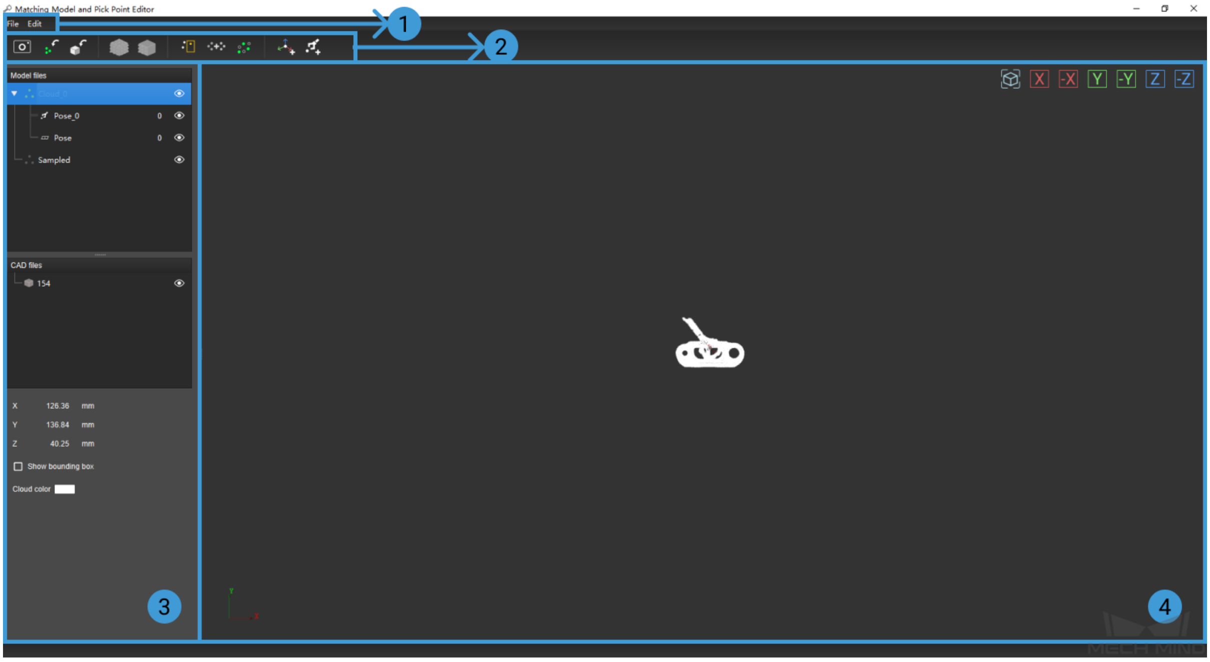
Hint: Matching Model and Pick Point Editor requires data and configurations of a Mech-Vision project to run. Therefore, please open the project for which you need to use Matching Model and Pick Point Editor first.

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.

Point cloud model: A point cloud of the target object, used as a model for matching. Mech-Vision compares the point clouds generated during project execution with this model to generate actual picking poses for the robot.

7.6.1 User Interface

The user interface of Matching Model and Pick Point Editor, as shown below, can be divided into four parts:



1. Menu Bar: including the File and Edit menus for saving, undoing, redoing, etc.

File Menu

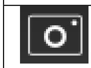
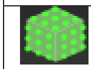
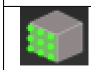
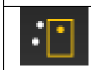
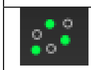
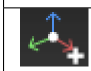
Option	Description	Shortcut
Save	Save all files in the Lists to *Project Folder/resource/model_editor*	Ctrl+S
Exit	Exit Matching Model and Pick Point Editor	None

Edit Menu

Option	Description	Shortcut
Remove selected points	Delete the selected points from the point cloud	Del
Undo	Undo point deletion	Ctrl+Z
Redo	Redo point deletion	Ctrl+Y
Display Settings	Adjust the size of points, font size of pose label, pick point size, size of drag handles, and types of handles displayed	None

2. Toolbar: including buttons used in point cloud model generation, point cloud model editing, and pick point addition.

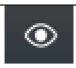
Only brief descriptions of each button is provided here. For detailed instructions of how to use these buttons, please refer to the chapters linked in General Workflow below.

Button	Description
	Generate a point cloud model from image acquired by the camera
	Import an existing point cloud file in PLY format
	Import a CAD model file in STL format
	Generate a point cloud of the CAD model' s surface
	Generate a point cloud of only the part visible in the current view of the CAD model' s surface
	Select points in the point cloud
	Merge point clouds
	Downsample a point cloud
	Add a pick point
	Add a pick point by teaching

3. Left side panel: including the **Model Files** List, **CAD Files** List, and the Parameters and Display pane.

Model Files List

All point cloud model files, pick point files, and geometric center files saved under *Project Folder/resource/model_editor/matching_source* are listed here.

Mouse action	Function
Click on  next to a file name	Hide this file in the Visualizing Space. Clicking again makes the file visible.
Double-click a file name	Rename the file
Drag	1. Adjust the order of files in the list 2. Dragging a pick point/geometric center onto a point cloud model associates the two
Right-click on a point cloud model	Can copy or delete this model
Right-click on a pick point/geometric center	Can copy, delete, or set this point as geometric center and/or pick point

Hint: If a point cloud model only has one pick point, then the "Set as geo. center" option in the right-click menu of this pick point cannot be unchecked.

CAD Files List

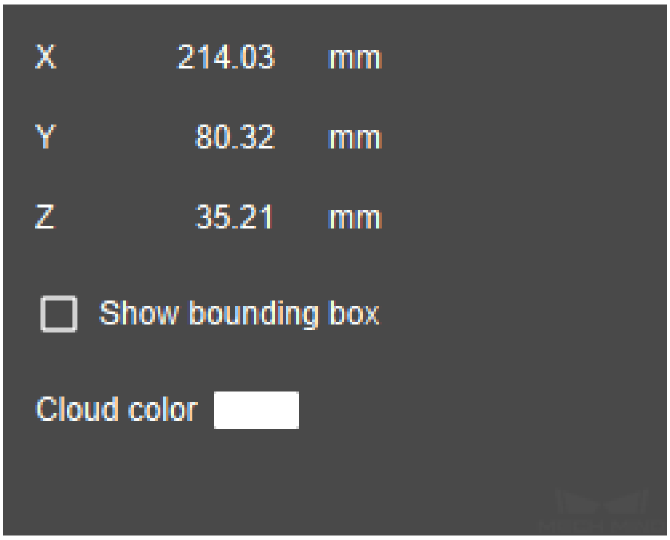
- All CAD model files saved under *Project Folder/resource/model_editor/cad_files* are listed here.
- Similar to the **Model Files** List, clicking on the next to a file name hides the file in the Visualizing Space. Clicking again makes the file visible.
- Right-clicking on a file name gives the option to remove the file from the List, but this will not delete the CAD file save in the above directory.

Parameters and Display Pane

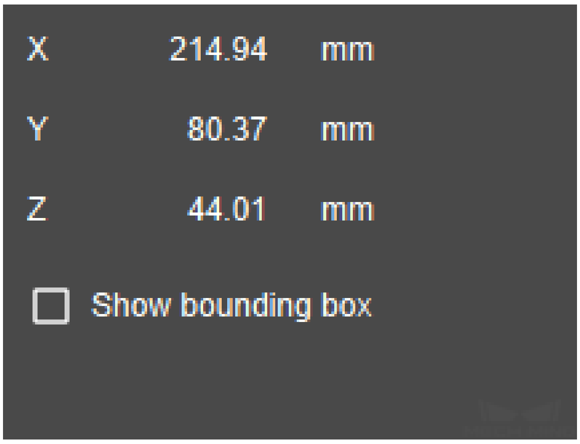
The Parameters and Display pane is the region below the **CAD Files** List. When you select a point cloud model, a pick point/geometric center, or a CAD file, this Pane will display the functions corresponding to the type of file.

1. When a point cloud model is selected: this pane displays the dimensions of the model' s bounding box, and the **Show bounding box** and **Cloud color** options.
   - Check **Show bounding box** to show the bounding box of the selected point cloud model.
   - Double-click on the white rectangle next to **Cloud color** to change the display color of the point cloud model.

   Hint: The display color of an imported color point cloud cannot be changed.

2. When a CAD file is selected: this pane displays the dimensions of the CAD model' s bounding box, and the **Show bounding box** option.
   - Check **Show bounding box** to show the bounding box of the selected CAD model.



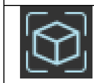
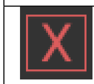
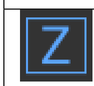
3. When a pick point/geometric center is selected: this pane displays the Pose Editing Widget. For detailed description, please refer to pose_editing_widget.

4. Visualizing Space: displaying the point cloud models and CAD files opened in Matching Model and Pick Point Editor, view adjustable.

You can adjust your view through the following mouse actions.

Mouse action	Function
Left-button drag	Rotate the view
Right-button drag or scroll	Zoom
Drag with the scroll wheel	pan the view

You can also use the buttons in the upper right of the Visualizing Space to achieve the following view adjustments.

Button	Description
	Zoom to fit, and center the selected file in the Visualizing Space
	View along the positive X-axis
	View along the negative X-axis
	View along the positive Y-axis
	View along the negative Y-axis
	View along the positive Z-axis
	View along the negative Z-axis

7.6.2 General Workflows

Depending on the method for adding pick point, there are two workflows for using Matching Model and Pick Point Editor:

1. Obtain pick point by moving the robot with the teach pendant: obtain the pick point first, and then generate the point cloud model, and edit the model last.
2. Add pick point directly in Matching Model and Pick Point Editor: generate the point cloud model first, and then edit the model, and add the pick point last.

The following chapters provide detailed instructions for each process:

Generate Point Cloud Model

There are two ways for you to generate a point cloud model in Matching Model and Pick Point Editor: from images acquired by the camera, or from an imported CAD file.

Generate from Images Acquired by the Camera

This feature first acquires a depth map of the background and one with the target object in it, and then generates a point cloud of the target object through comparing the two depth maps.

Hint: This feature is available only when the opened project contains the *Capture Images from Camera* Step, AND the Camera Settings parameters of this Step are fully configured.

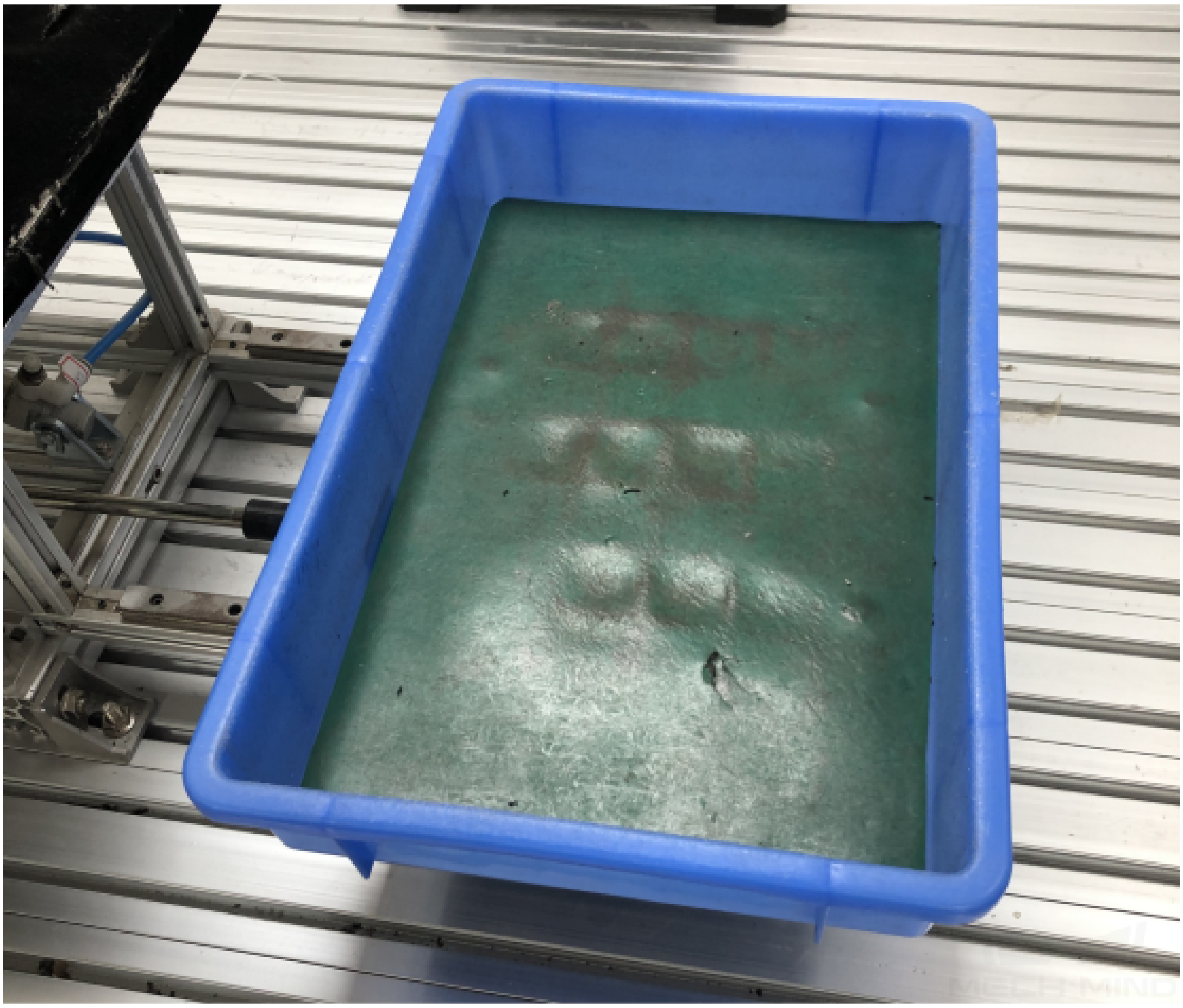
The procedure differs depending on whether you are using a real camera or a virtual camera. Click on the options below for detailed instructions.

Real camera

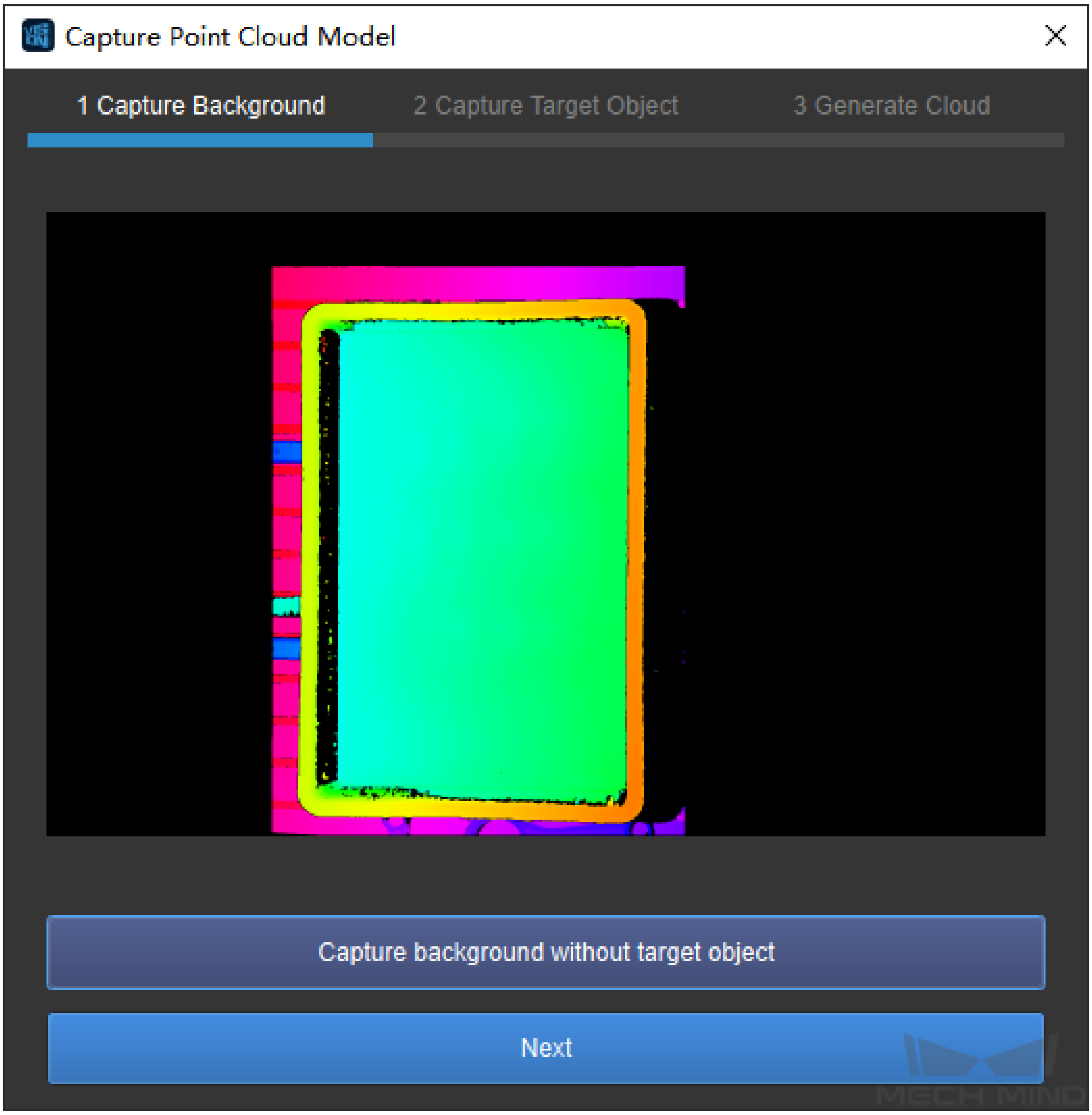
Depending on whether you have added pick points by teaching, the procedure for using a real camera to acquire images for point cloud model generation differs. Click on the option that fits your situation for detailed instructions.

Have NOT added pick point by teaching

1. Remove all target objects in the camera' s field of view and retain the background only. If the robot picks from a bin, then you need to remove all the target objects from the bin but keep the bin in place.

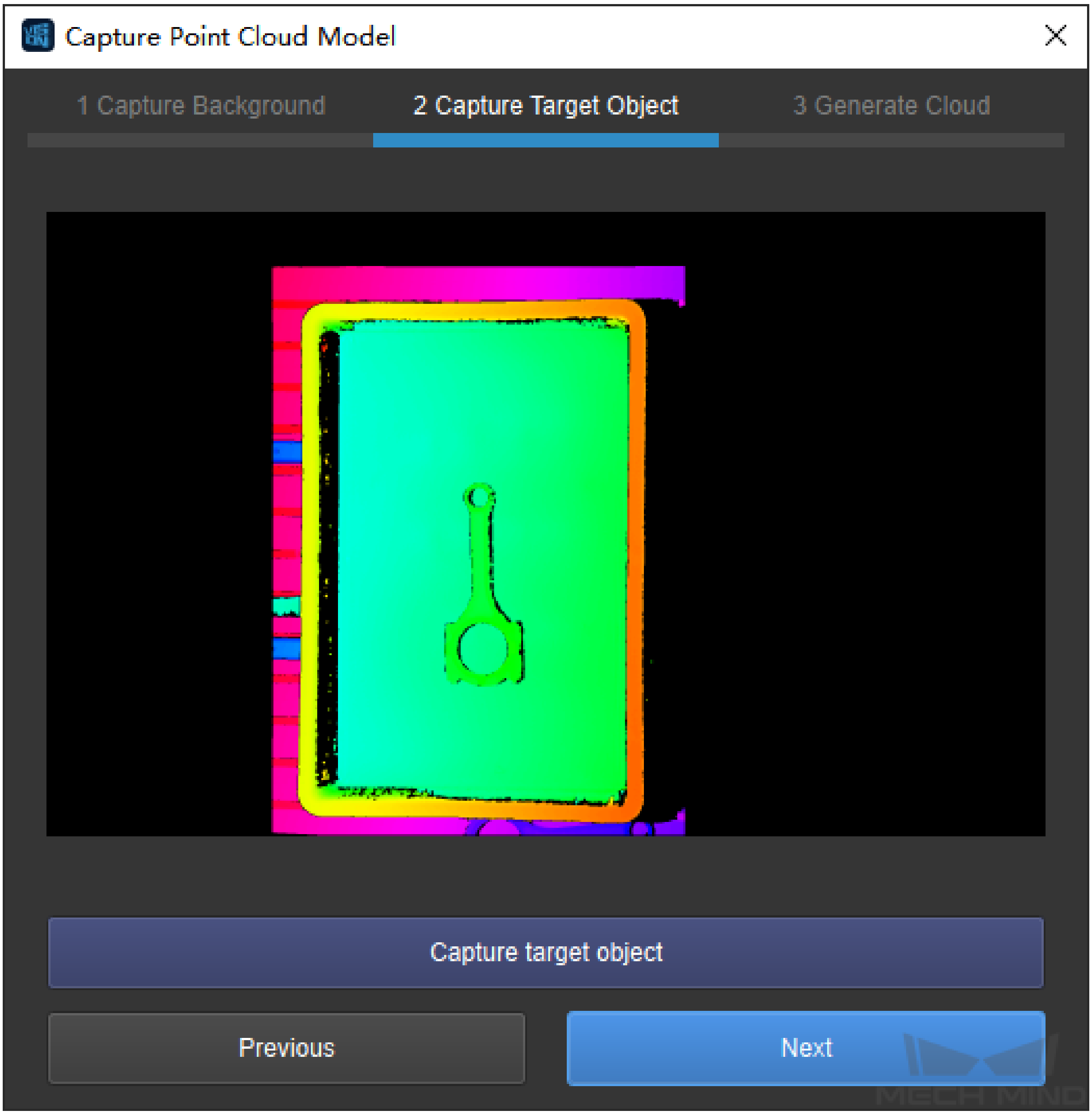


2. Click on [camera icon] in the Toolbar to open the Capture Point Cloud Model Window.
3. Click on *Capture background without target object* to acquire a background depth map, and then click on *Next*.

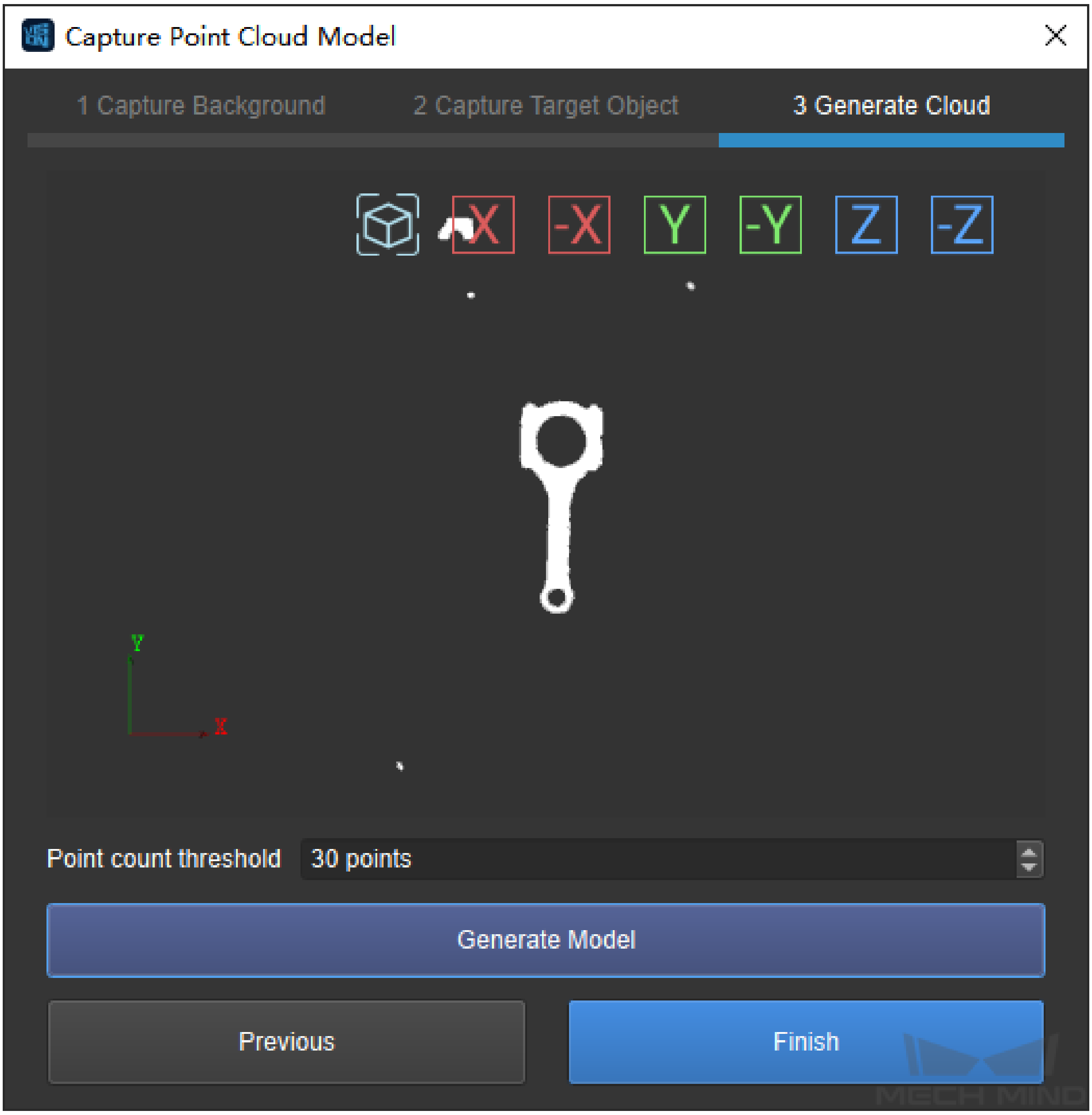


4. Place a target object in the camera’ s field of view (in the bin).

5. Click on *Capture target object* to acquire a depth map of the target object, and then click on *Next*.



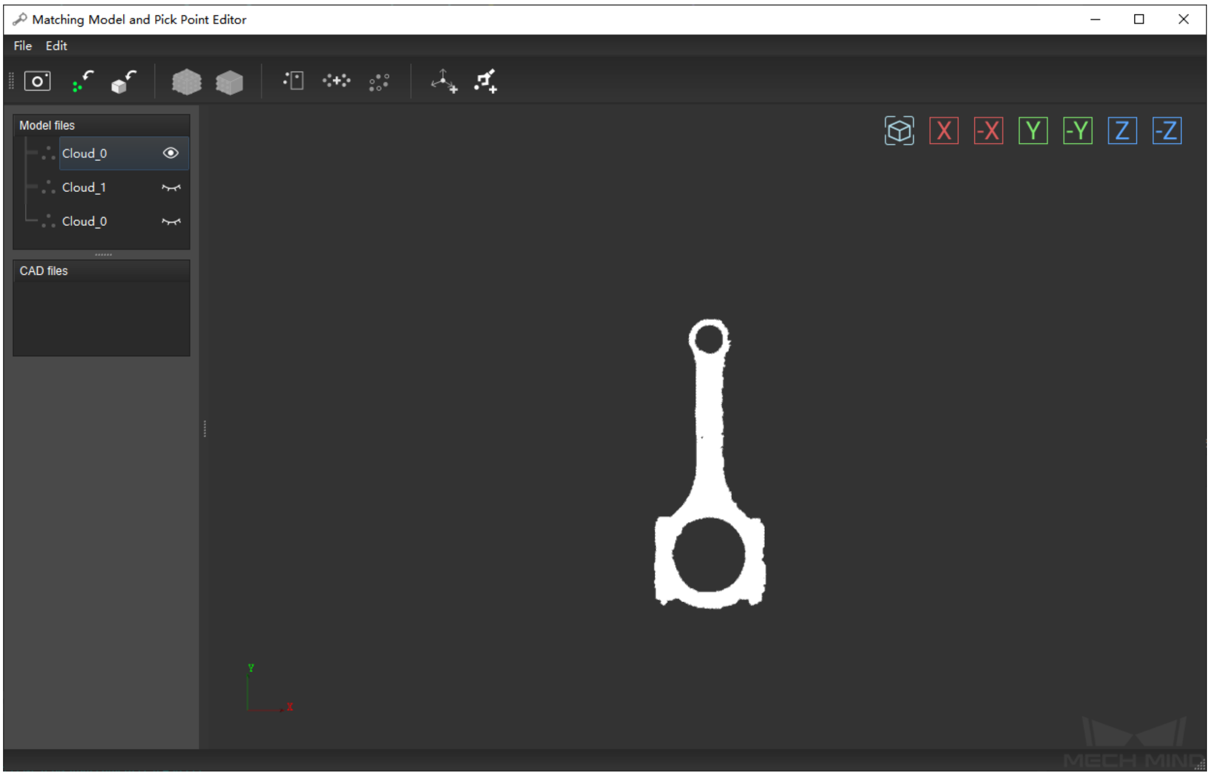
6. Click on *Generate Model* to generate a point cloud model of the target object.



Hint: You can check the point cloud by adjusting your view using the mouse in this page.

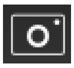
7. Adjust **Point count threshold**:
   - If the target object in the point cloud is not intact, you can increase the value here and then click on *Generate Model* to check the result.
   - If the point cloud contains too many points of the background, you can decrease the value here and then click on *Generate Model* to check the result.
8. Click on *Finish* to exit the window. Now you can see the newly generated point cloud

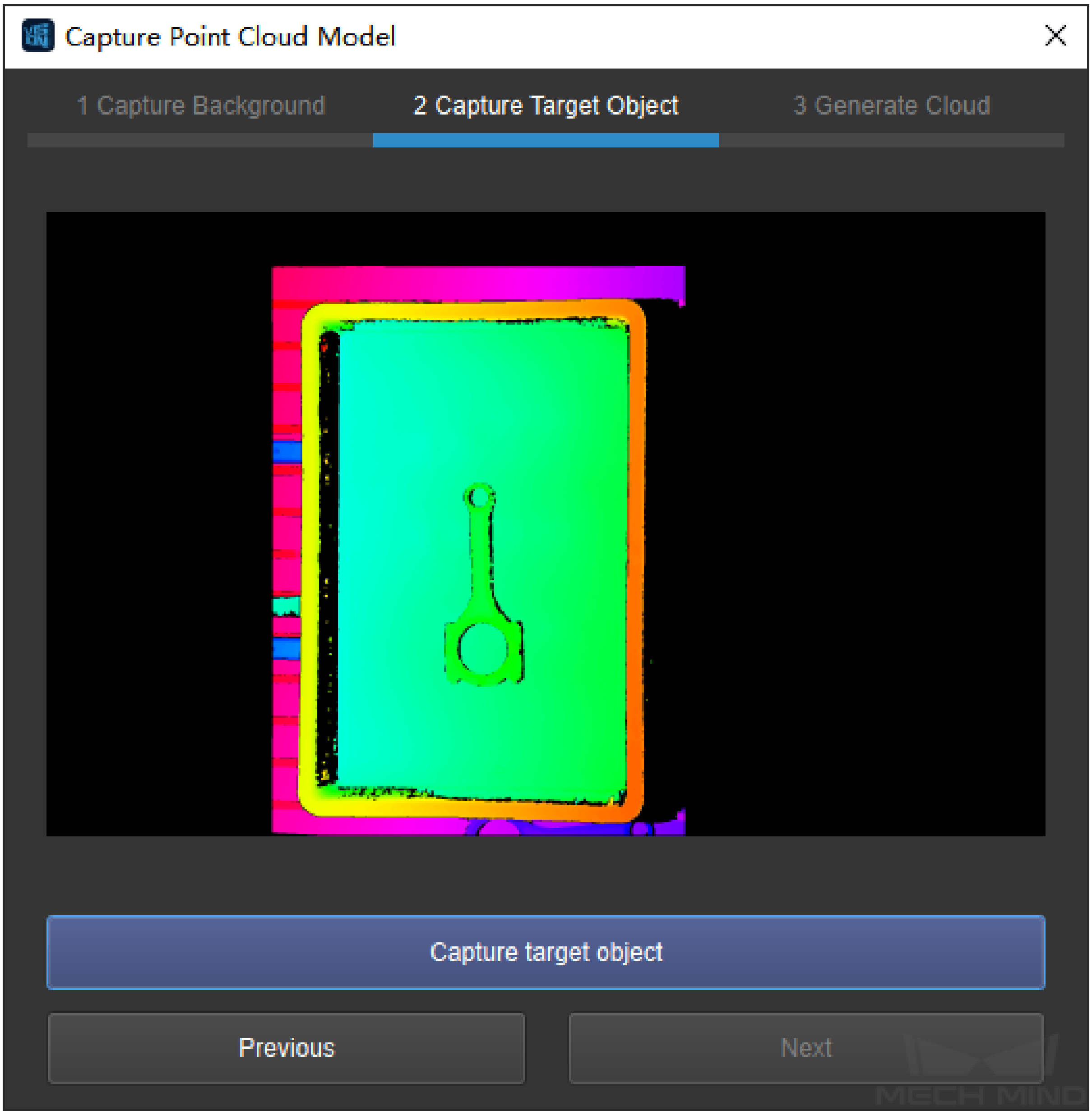
model in the Visualizing Space and in the Model Files List on the left.



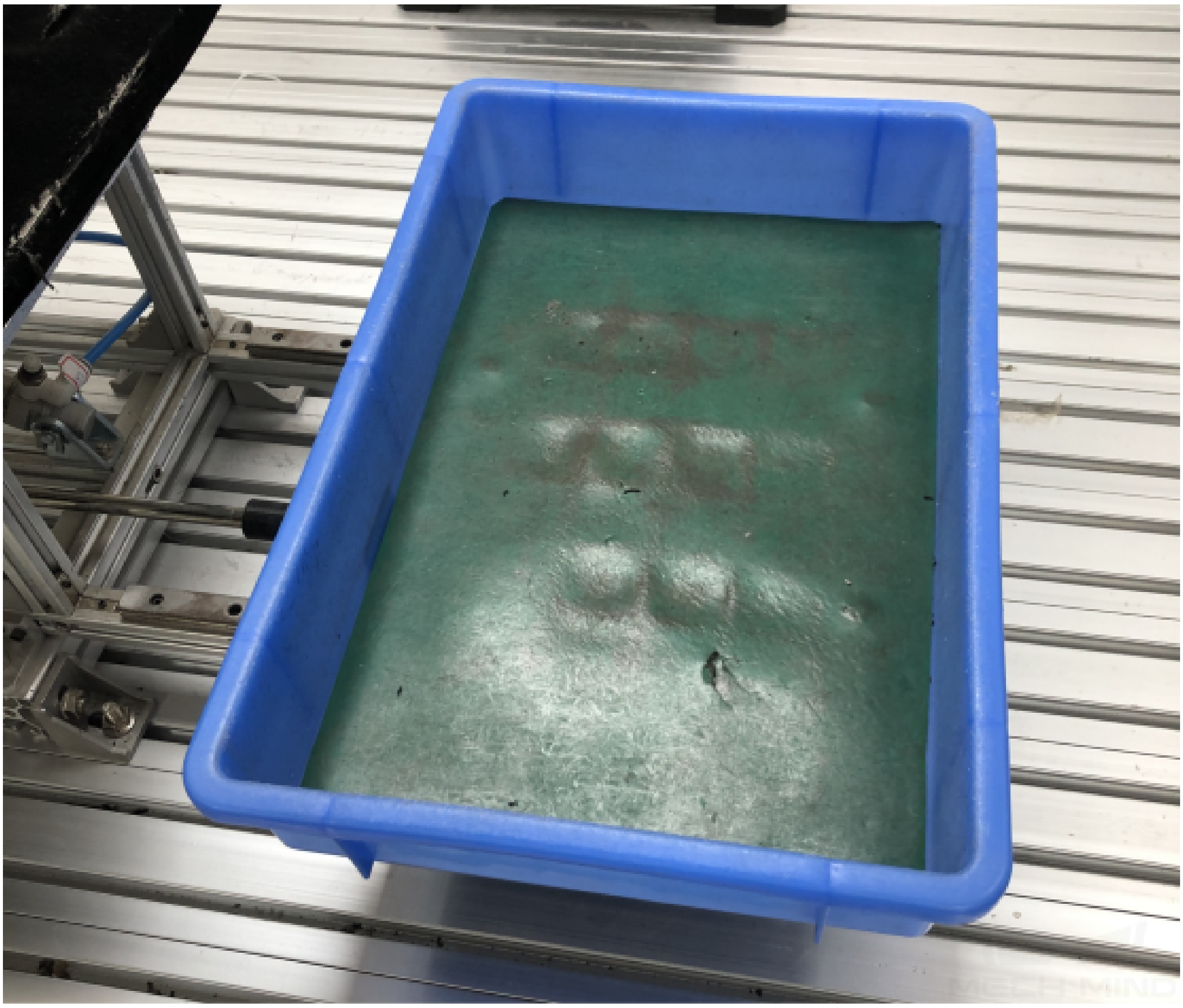
Hint: Double-click on the model name in the **Model Files** List to rename a point cloud model.

Already added pick point by teaching

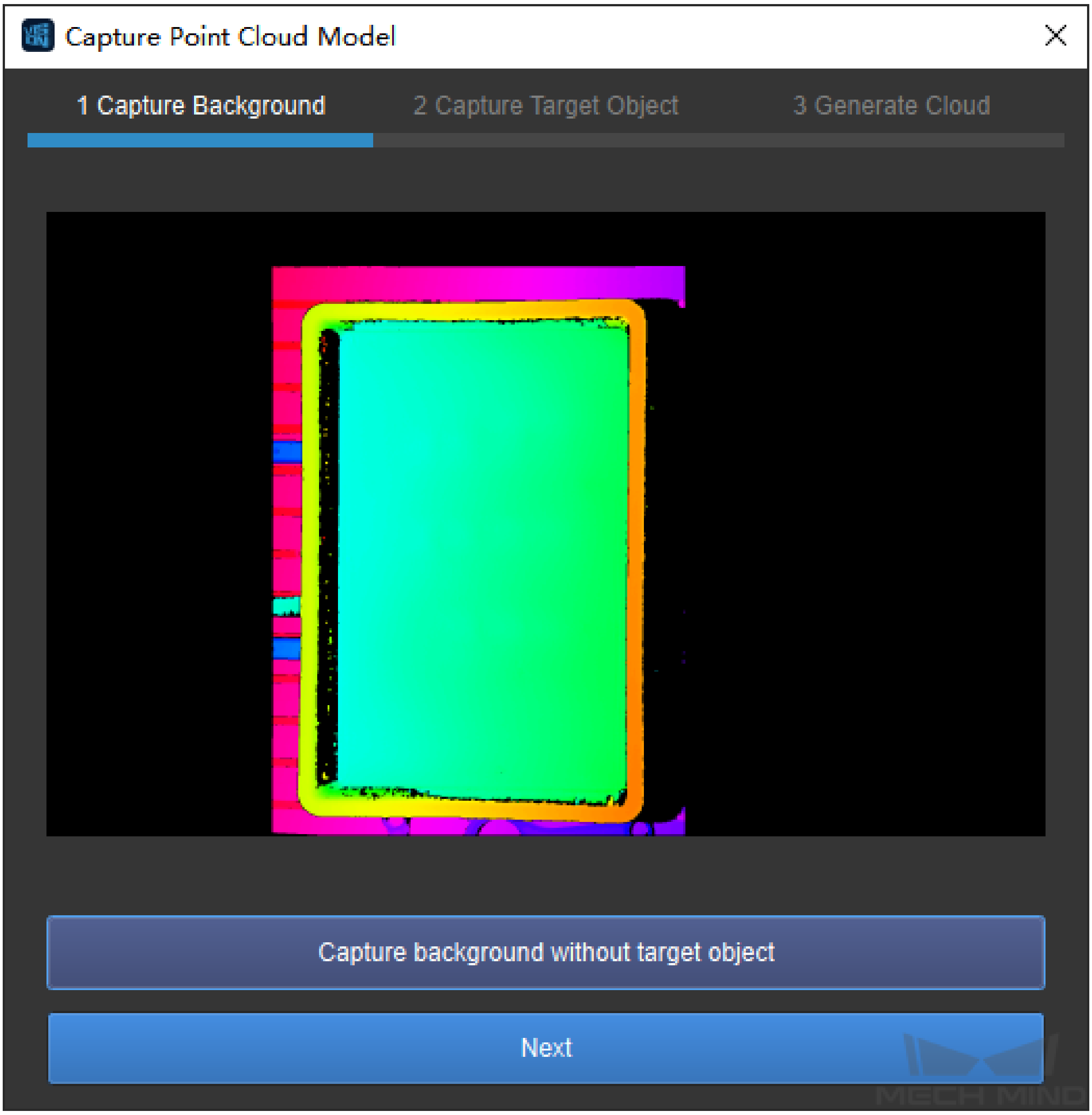
1. Make sure that the target object has not been moved. Click on  in the Toolbar, and then click on *Next* in the window that appears.
2. Click on *Capture target object* to acquire a depth map of the target object, and then click on *Next*.



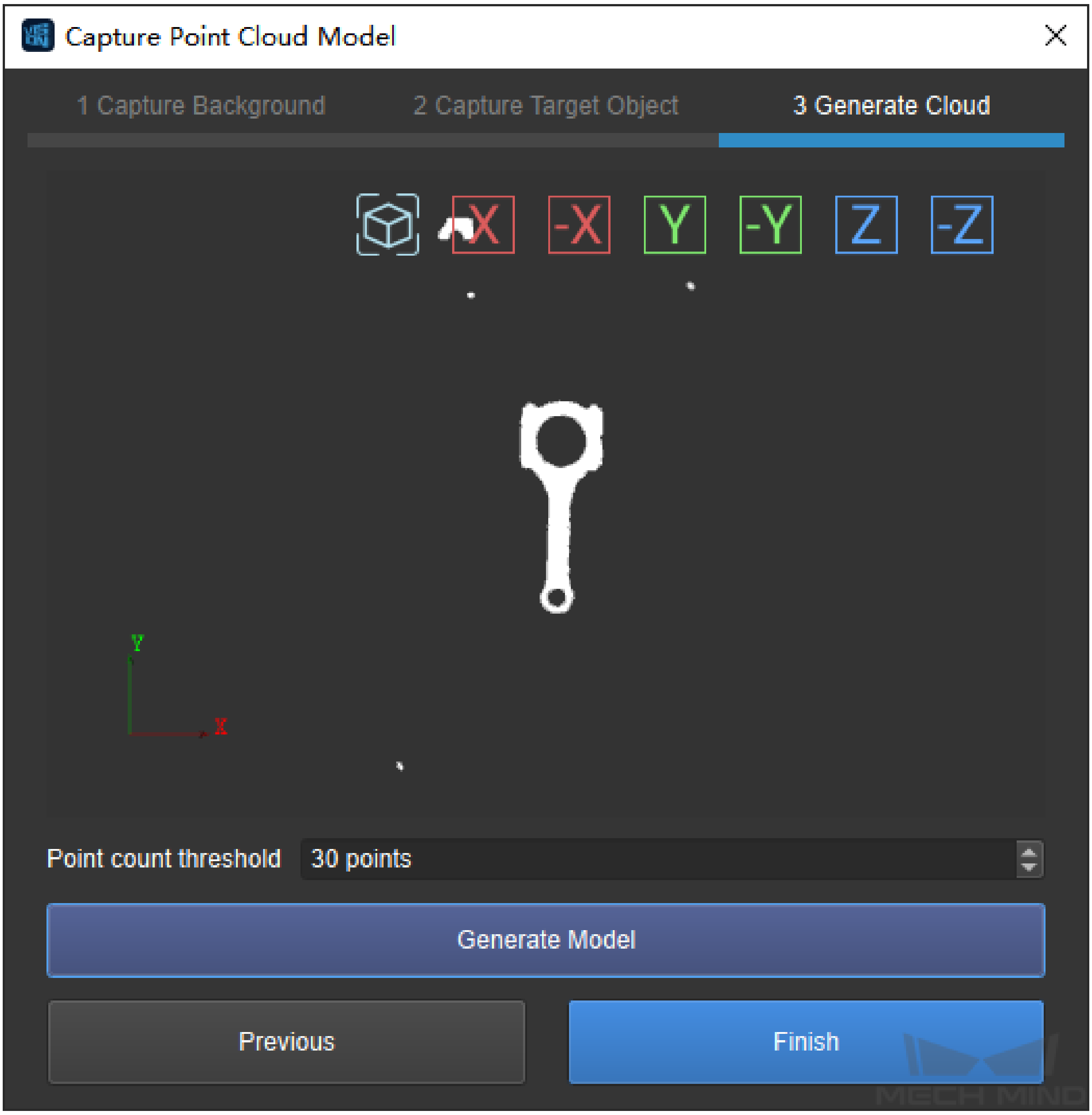
3. Remove all target objects in the camera' s field of view and retain the background only. If the robot picks from a bin, then you need to remove all the target objects from the bin but keep the bin in place.



4. Click on *Capture background without target object* to acquire a background depth map. Then, click on *Next*, and click on *Next* AGAIN to reach the Generate Model page.



5. Click on *Generate Model* to generate a point cloud model of the target object.

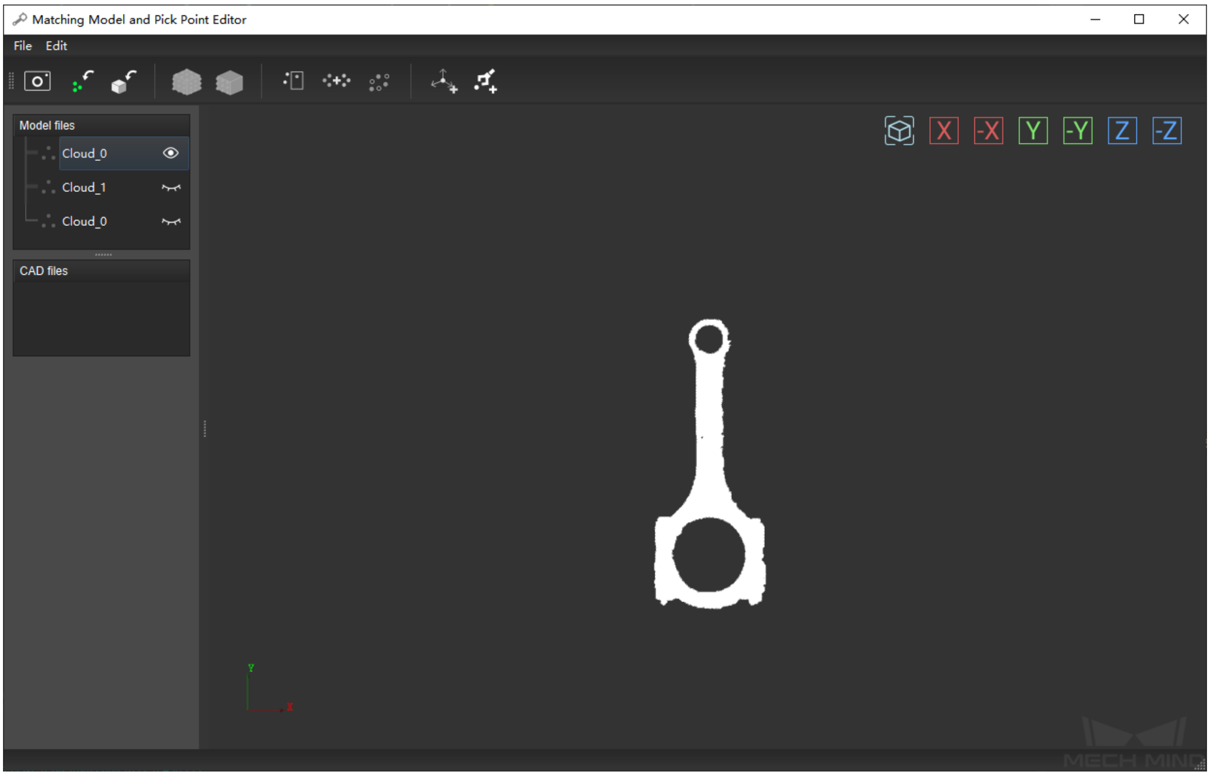


Hint: You can check the point cloud by adjusting your view using the mouse in this page.

---

6. Adjust **Point count threshold**:
   - If the point cloud of the target object is incomplete, you can increase the value here, and then click on *Generate Model* to check the result.
   - If the point cloud contains too many points of the background, you can decrease the value here, and then click on *Generate Model* to check the result.
7. Click on *Finish* to exit the window. Now you can see the newly generated point cloud

model in the Visualizing Space and in the Model Files List on the left.



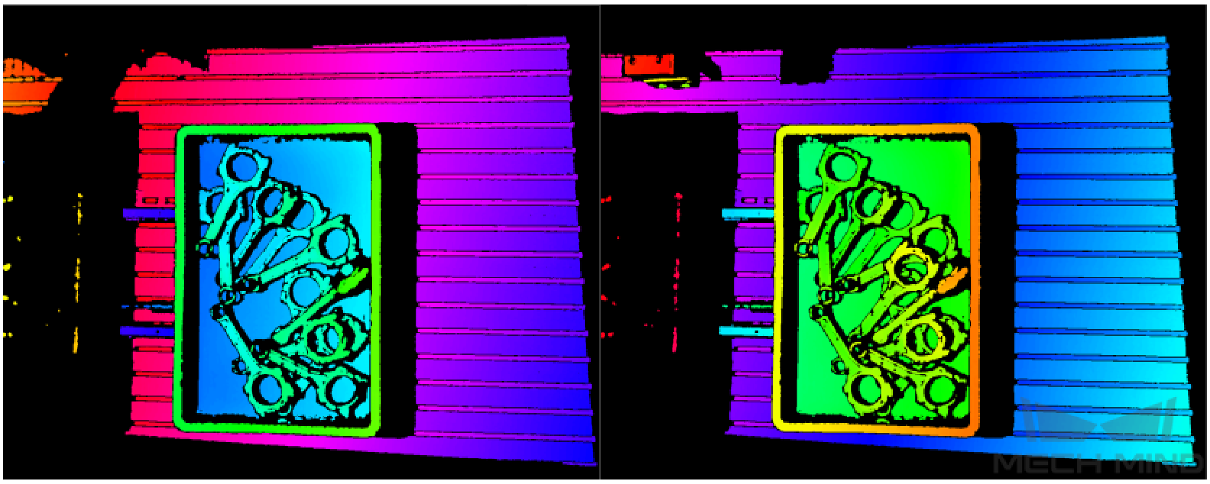
Hint: Double-click on the model name in the **Model Files** List to rename a point cloud model.

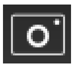
Hint: If you reached this page through clicking on the link in Add Pick Point by Teaching, you can use the Back button on your browser to return and continue reading the instructions of Add Pick Point by Teaching.

Virtual camera

When using a virtual camera, since you will not be able to acquire images in real time, you need to get images collected previously ready for use.

Tip: The background depth map doesn't have to contain no target objects at all; it just needs to contain one fewer target object than the depth map of the target object. For example, in the figure below, the left image can be used as the background depth map, and the right one as the depth map of the target object.

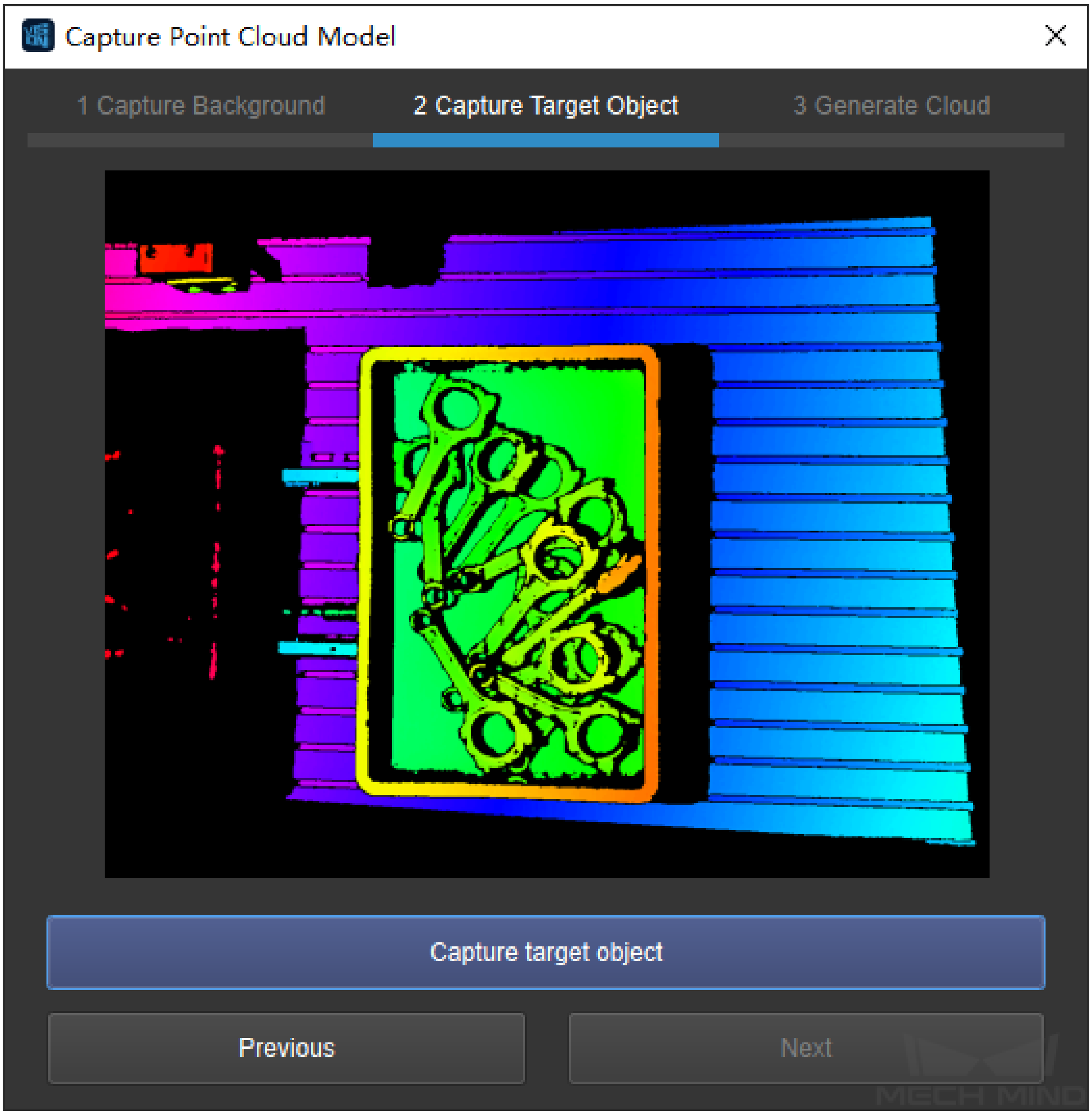


1. In the Graphical Programming Workspace, select the Capture Images from Camera Step, and adjust the following in the Parameter panel to the right:
   - For **Data Path**, select the parent folder of the folder containing the depth maps. For detailed instructions, please see *Capture Images from Camera*.
   - Set **Play Mode** to **Repeat All**.
2. Click on *Toolkit* → *Matching Model and Pick Point Editor* in the Menu Bar to open the Editor, and then click on  in the Toolbar to open the Capture Point Cloud Model Window.

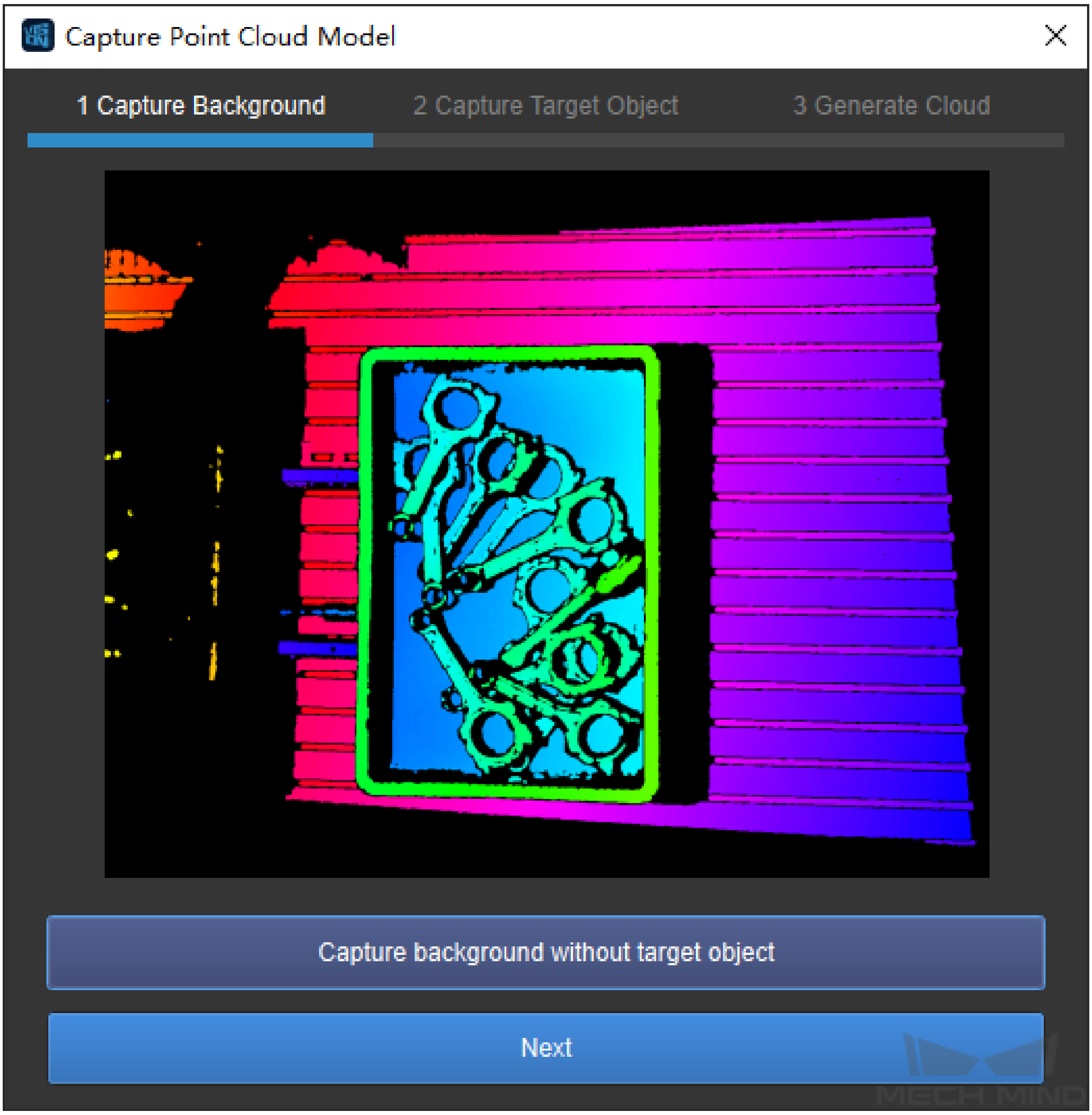
   Note: As saved images are usually acquired during actual picking, the *n* th image usually contains one more target object than the *(n-1)* th image. Therefore, we recommend that you read in the depth map of the target object first, and then use the next image as the background depth map.

3. Click on *Next*, and then click on *Capture target object* to read in a depth map of the target object.

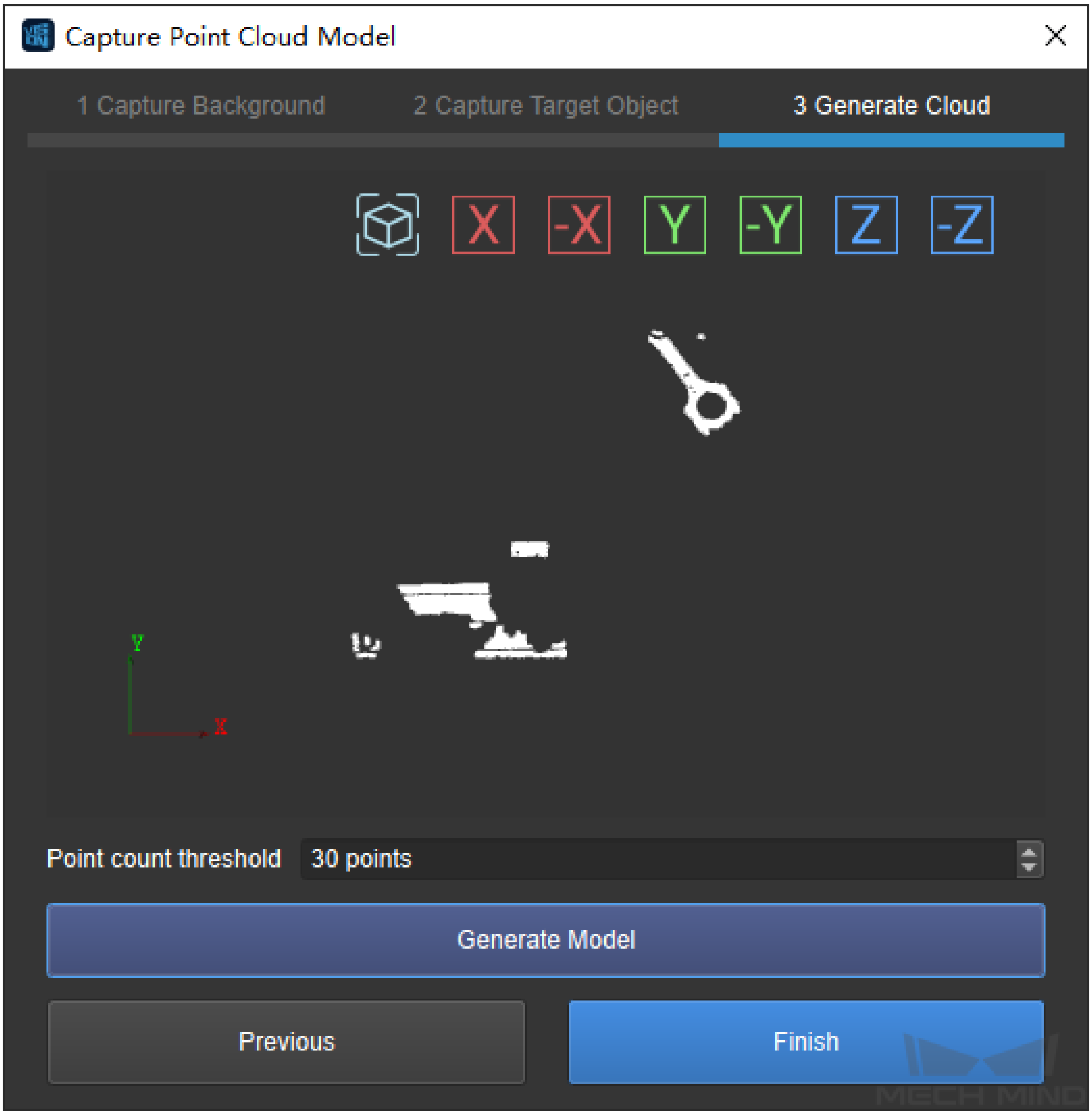
   Click on *Capture target object* again to read in the next image in the folder. Repeat until the displayed depth map meets your needs, and then click on *Previous*.



4. Click on *Capture background without target object* to read in a background depth map. Then, click on *Next*, and click on *Next* AGAIN to reach the Generate Model page.



5. Click on *Generate Model* to generate a point cloud model of the target object.

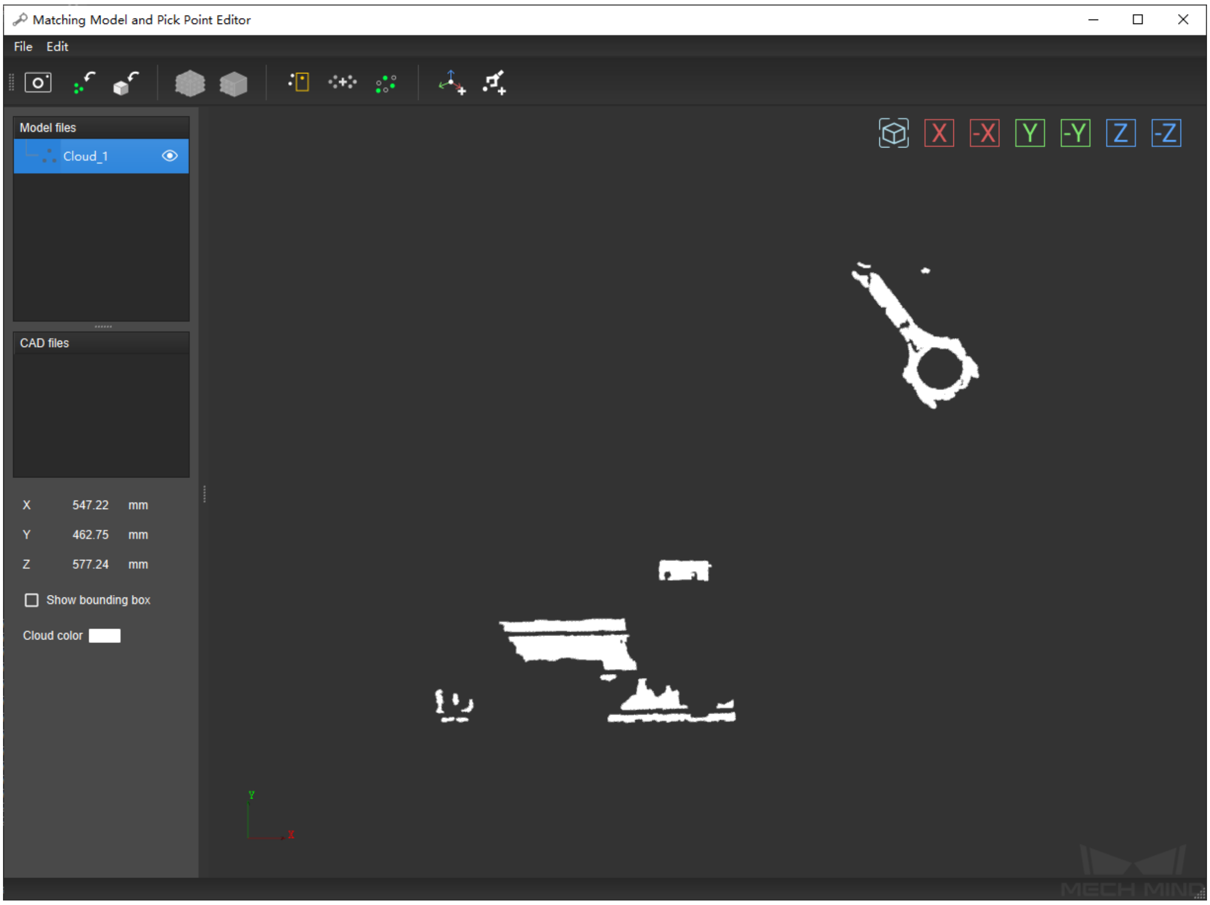


Hint: You can check the point cloud by adjusting your view using the mouse in this page.

---

6. Adjust **Point count threshold**:
   - If the point cloud of the target object is incomplete, you can increase the value here, and then click on *Generate Model* to check the result.
   - If the point cloud contains too many points of the background, you can decrease the value here, and then click on *Generate Model* to check the result.
7. Click on *Finish* to exit the window. Now you can see the newly generated point cloud model in

the Visualizing Space and in the **Model Files** List on the left.



Hint: Double-click on the model name in the **Model Files** List to rename a point cloud model.

Due to factors like lighting variations in the images, the generated point cloud model is likely to contain unwanted points. You can trim the model during *editing*.

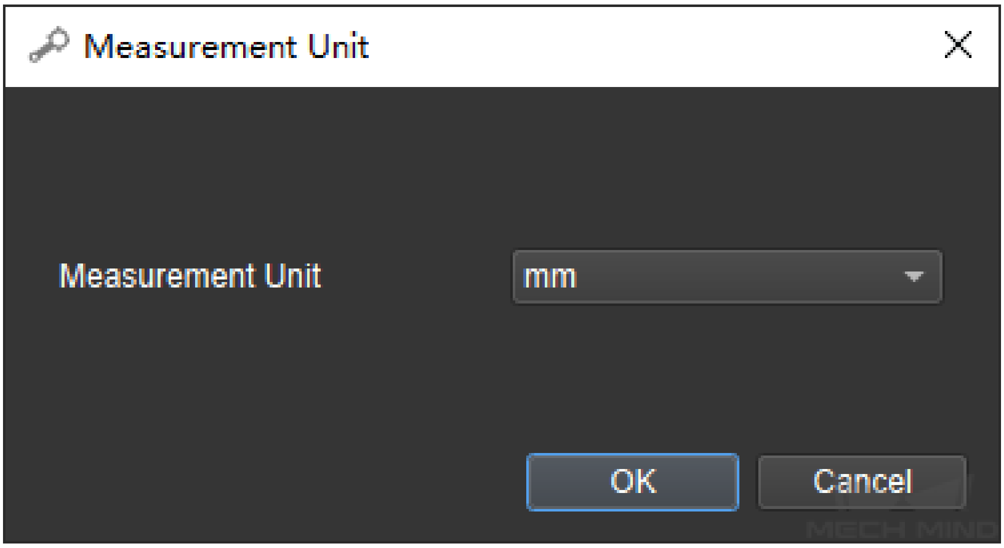
Generate from imported CAD files

If you already have the CAD model file of the target object, you can import this file into Matching Model and Pick Point Editor, and generate a point cloud model from the CAD file.

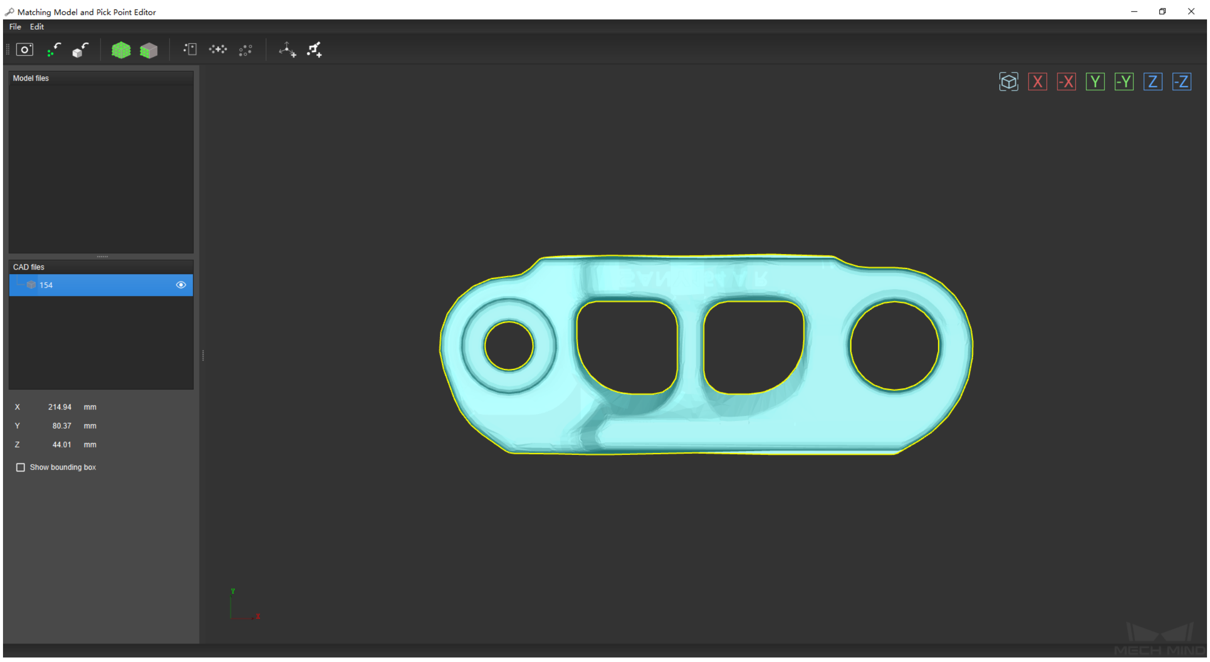
Hint: Currently, only STL format is supported.

Click here for detailed instructions

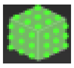
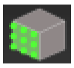
1. Click on [icon] in the Toolbar, locate the CAD file you need, select this file and click on Open.
2. Select the measurement unit of the CAD model in the pop-up window.

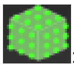


3. The imported CAD file will show up in the **CAD Files** List on the left.

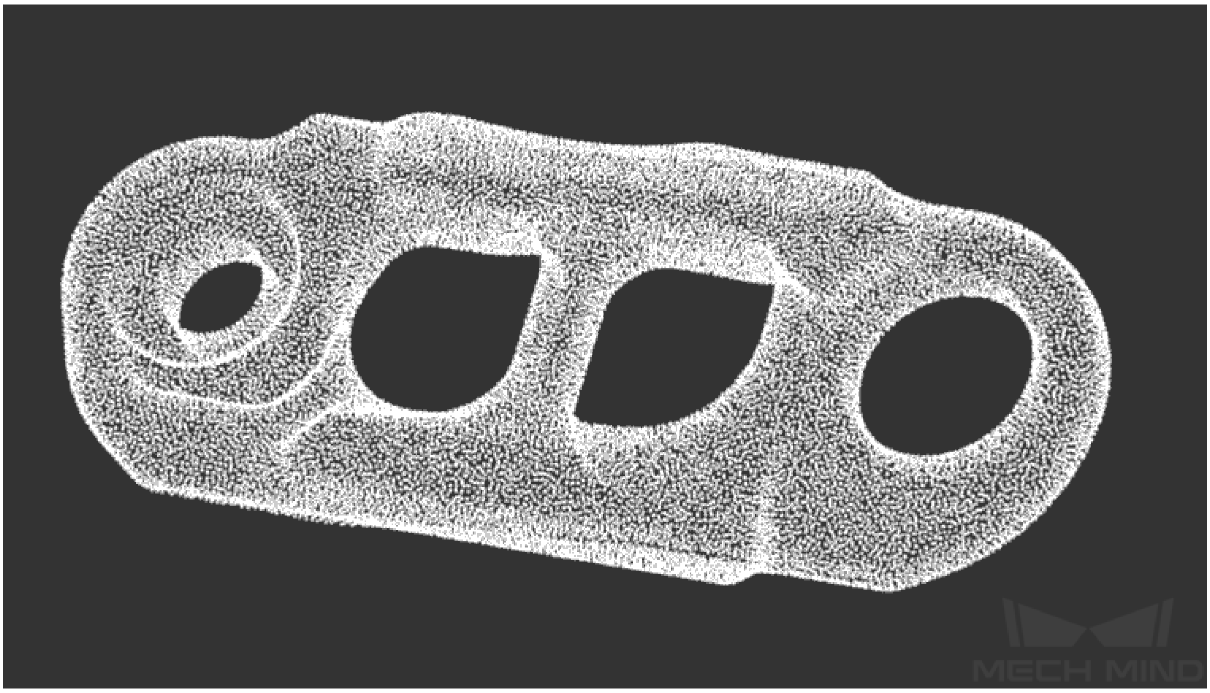


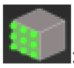
Hint: Only one CAD file can be imported at a time. If you need to import multiple CAD files, please repeat steps 1 to 2.

4. Select the CAD file for point cloud model generation in the **CAD Files** List, and then click on  or  in the Toolbar to generate a point cloud model.

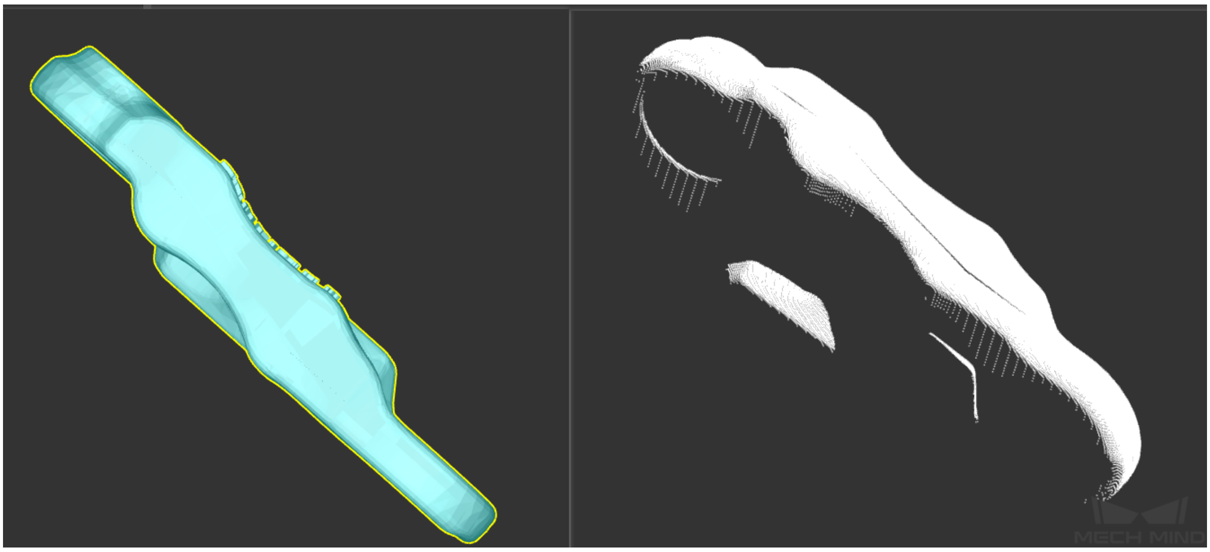
   - : generate a point cloud of the entire surface of the CAD model.

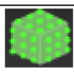
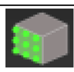
     After clicking, you will need to adjust the sampling interval for the point cloud in the pop-up window, and then click on *OK* to generate the point cloud model.



   - : generate a point cloud of only the part visible in the current view.

     In the figure below, the left image is the view used for point cloud generation, and the right image is the generated point cloud. You can see that the point cloud doesn't contain points from the part that is not visible in the left image.



Hint: Occasionally,  and  may become disabled if you already have point cloud model(s) generated.

In such case, please click on anywhere in the Editor interface to deselect the file in the **CAD Files** List, and then reselect it. The above buttons should become enabled.

Hint: Other than generating point cloud models through the methods described above, you can also import existing point cloud files in PLY format directly by clicking on  in the Toolbar.

The following are some actions you can take once you have generated a point cloud model:

- Save all the files in the Lists: click on *File* → *Save* or use the shortcut Ctrl+S.

  Hint: Files are save to the following directory: Project Folder/resource/model_editor.

- Delete a point cloud model: right-click on the point cloud you wish to delete in the **Model Files** List, and select *Delete* in the pop-up menu.
- Toggle the visibility of a file: click on  to the right of the file name will make the file invisible in the Visualizing Space. Clicking again makes the file visible.
- Show the bounding box: select a file in the Files List, and then check the box in front of **Show bounding box** below the **CAD Files** List.
- Change the display color of a point cloud: select a point cloud model in the **Model Files** List, double-click on the white rectangle next to **Cloud color** below the **CAD Files** List, and then change the color in the pop-up window.

Hint: The display color of an imported color point cloud cannot be changed.

Now that you have a point cloud model, you can start editing it to meet your actual needs. Please click on *Next* in the lower right for a detailed instruction for editing a point cloud model.

Edit Point Cloud Model

Once you have generated a point cloud model using or , or imported a point cloud model using , you may want to edit it so that it can meet your actual needs.

In Matching Model and Pick Point Editor, you can perform the following editing (Click for detailed instructions).

Select points: delete unwanted points

1. Select the point cloud model for editing in the **Model Files** List, adjust your view in the Visualizing Space so that you can see most or all the points you want to delete, and then click on in the Toolbar.

   Hint: Once you've clicked on , the view in the Visualizing Space will be locked. Therefore, please adjust your view first.

2. Select the points to delete in the Visualizing Space. To select multiple areas, hold the Ctrl key while you select. The selected points are shown in orange.

3. Click on *Edit* → *Remove selected points* in the Menu Bar or press the Delete key to delete the selected points.

Tip: If you deleted wanted points by accident, you can undo the deleting by clicking on *Edit* → *Undo* in the Menu Bar or using the shortcut Ctrl+Z.

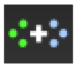
You can also redo deleting by clicking on *Edit* → *Redo* in the Menu Bar or using the shortcut Ctrl+Y.

Tip: The size of the bounding box can indicate if you have deleted all the unwanted points.

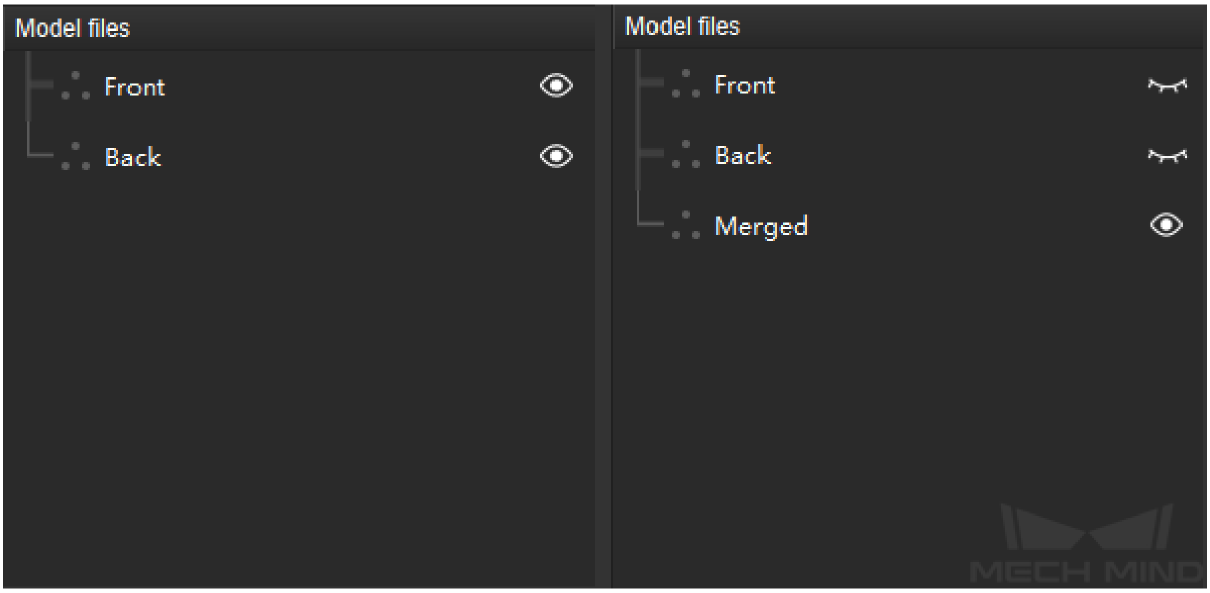
Check the box in front of **Show bounding box** below the **CAD Files** List to show the bounding box.

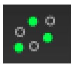
4. Click on  again, or press the Esc key to unlock the view in the Visualizing Space.

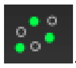
 Merge point cloud: merge multiple point clouds into one

Select the point cloud models for merging in the **Model Files** List, and then click on  in the Toolbar to generate a new point cloud from merging. Meanwhile, the point clouds selected for merging will become invisible automatically.

In the figure below, the left image is the **Model Files** List before point cloud merging, and the right is the **Model Files** List after merging the point clouds Front and Back.



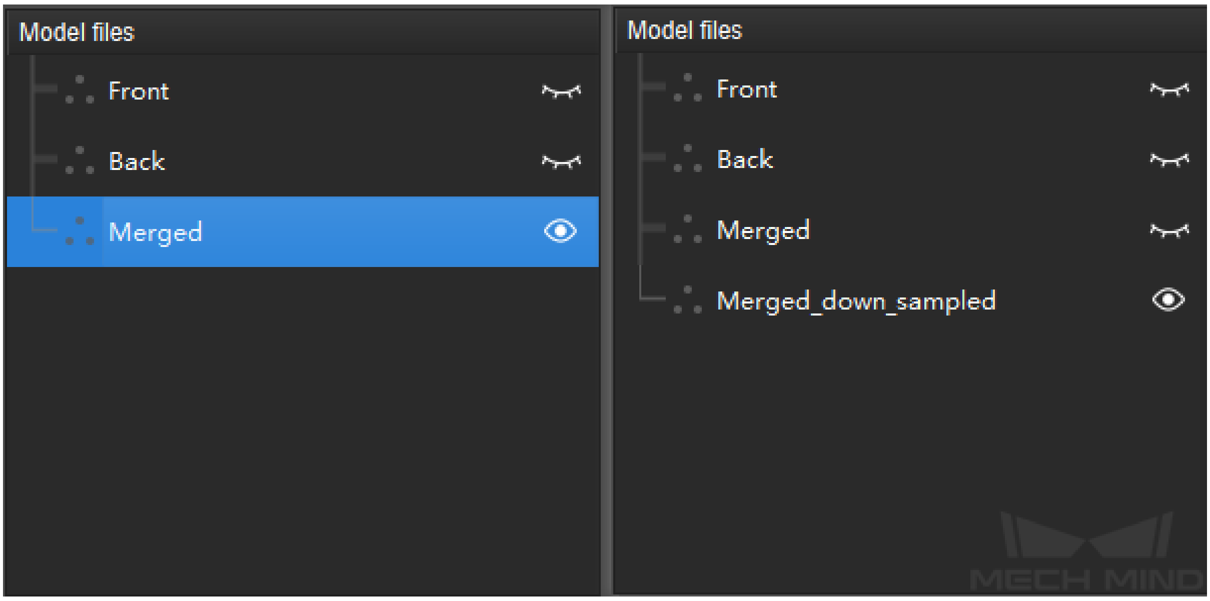
 Downsample: reduce the size of the point cloud model to speed up subsequent processing

1. Select the point cloud model for downsampling in the **Model Files** List, click on , and adjust the sampling interval in the pop-up window.

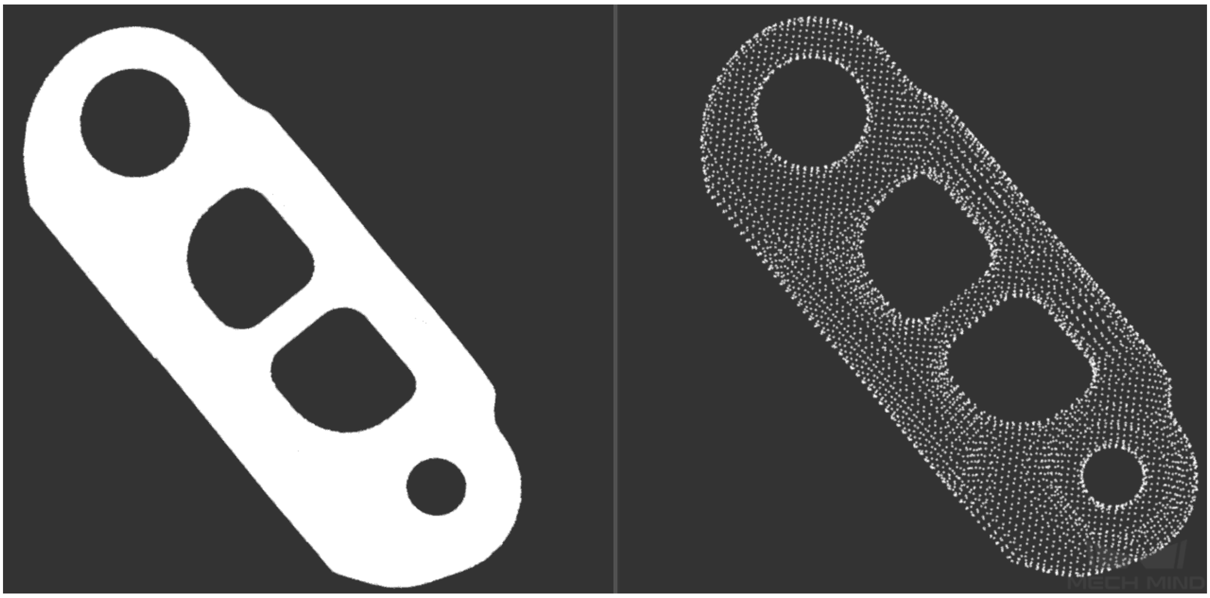
   Note: the larger the sampling interval, the sparser the points in the point cloud model, and therefore the smaller the file size will be.

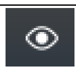
2. Once you finish adjusting the sampling interval, click on *OK* to generate a new, downsampled point cloud. Meanwhile, the point cloud selected for downsampling will become invisible automatically.

   In the figure below, the left image is the **Model Files** List before downsampling Merged, and the right after.



In the figure below, the left image is a point cloud model before downsampling, and the right one is after downsampling with a sampling interval of 3 mm.



Hint: Point cloud models must be visible (i.e., the icon after the file name is ) to be edited.

Add Pick Point

The robot needs a picking pose provided by Mech-Vision to actually perform the picking. The picking pose is transformed from the pick point on the target object. Therefore, pick points need to be added to point cloud models, so that Mech-Vision can generate pick points for target objects through matching, and further output picking poses for the robot.

Note:

- Mech-Vision' s algorithm requires one and only one "geometric center" for every point cloud model.
- The first pick point you add will be automatically set as the geometric center.
- The geometric center can be a pick point at the same time.

There are two ways for you to add pick points in Matching Model and Pick Point Editor:

-  Add pick point:
  - Directly add and adjust a pick point in Matching Model and Pick Point Editor
  - Select a point cloud model first to enable this button
-  Add pick point by teaching:
  - Adjust the robot pose manually through the teach pendant and obtain the pick point. Used in cases that require high picking accuracy or where TCP measurement is difficult
  - Add the pick point first, and then generate the point cloud model through image acquisition by the camera

 Add pick point

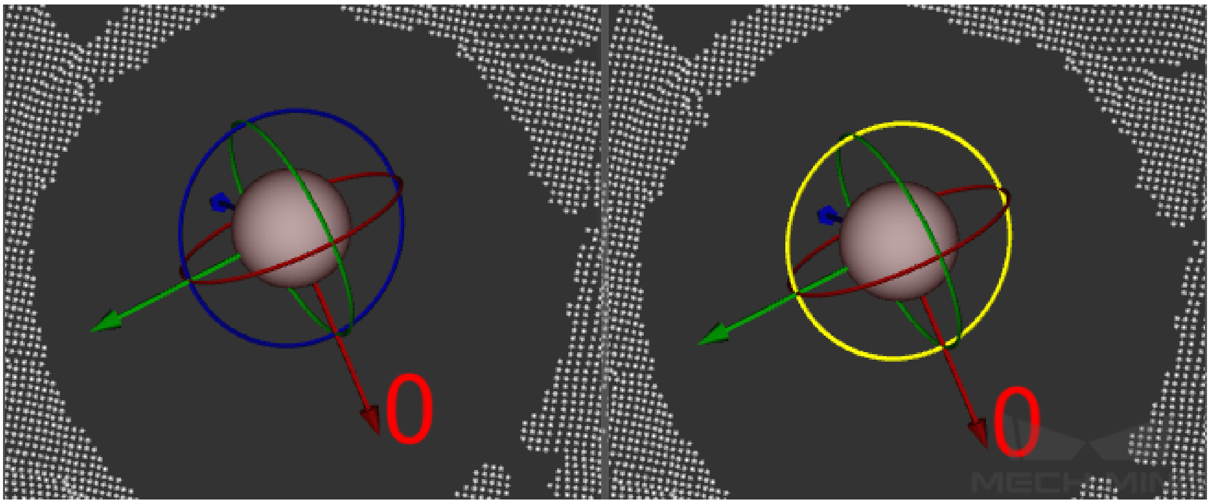
1. Select the point cloud model for adding pick point in the **Model Files** List, and then click on  in the Toolbar to add a pick point.

   Hint: By default, pick points added using  are placed at the geometric center of the point cloud model' s bounding box.

   ---

2. Select this pick point in the **Model Files** List to adjust it. There are two ways you can adjust a pick point:
   - By dragging the drag handles on the pick point: while holding the Ctrl key, click and drag a handle to move the pick point along the axis of this handle.

     Hint: A drag handle is successfully selected if it turns yellow after being clicked. In the figure below, the left image is before selection and the right after.



- By editing the pose data: Pose editing can be done below the **CAD Files** List. Please see pose_editing_widget for instructions.

 Add pick point by teaching

You' ll need to input the TCP manually when adding a pick point this way. Therefore, please have the TCP data ready for use beforehand.

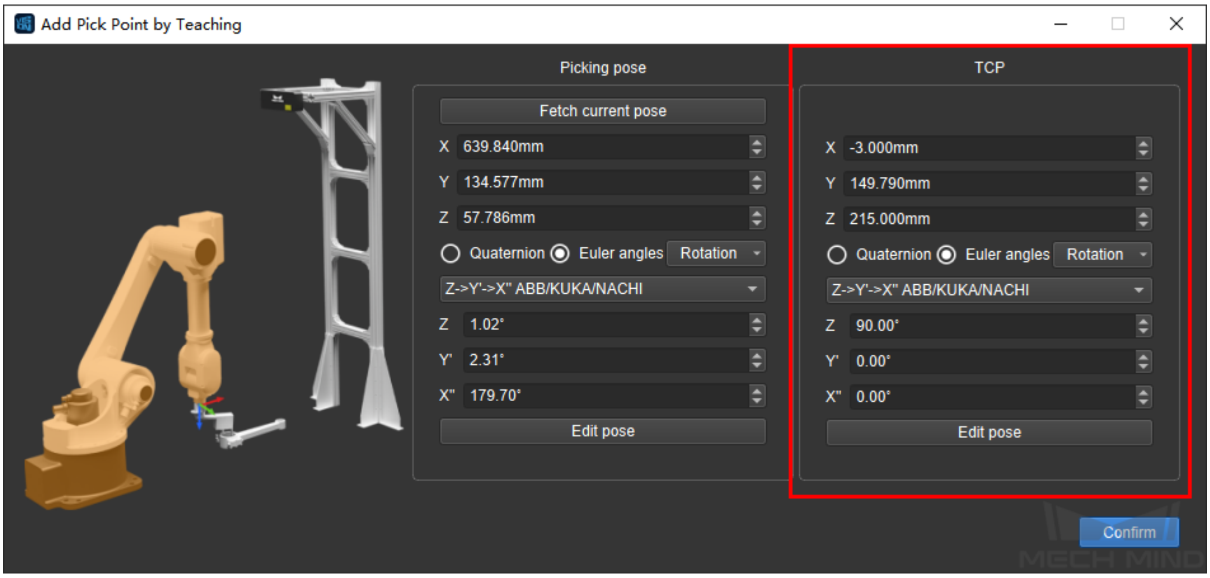
> **Tip:** If you are using Mech-Viz, you can check the TCP by going into *Mech-Viz* → *End Effector & Objects Tab* → *End Effectors* and double-clicking on the corresponding end effector model.

The procedure for adding a pick point by teaching differs depending on how your camera is installed. Click on the options below for detailed instructions for ETH and EIH.

> **Note:**
>
> 1. Mech-Vision automatically determines how the camera is installed based on the extrinsic parameters in the project, and shows the corresponding Add Pick Point by Teaching window.
> 2. If the robot is connected through Mech-Center, the picking pose and the image capturing pose can be obtained automatically. Otherwise, these poses must be input manually.

Add pick point by teaching under ETH

1. Click on  in the Toolbar to open the Add Pick Point by Teaching window.
2. Input the TCP obtained beforehand into the TCP section.



3. Move the robot to the picking pose using the teach pendant. Operate the end effector to perform picking to make sure the picking pose is accurate.

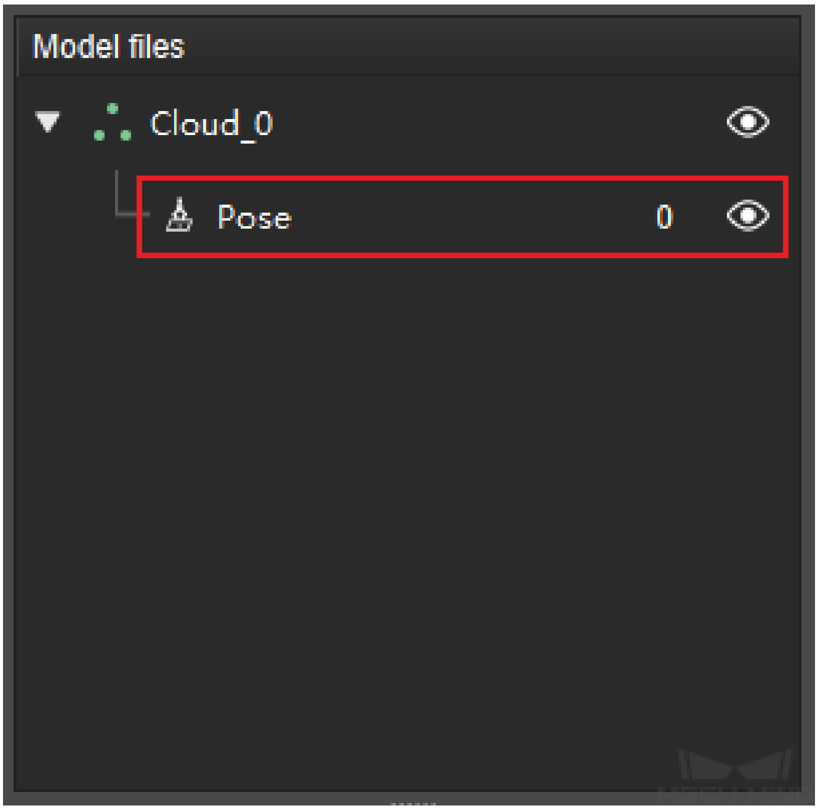
   Hint: If you are using a fingered gripper, you can grasp and drop the target object several times to make sure that the object can be firmly grasped in this picking pose.

4. In the Picking Pose section, click on *Fetch current pose*, or input the pose displayed on the teach pendant manually. Click on *Confirm* to generate a pick point.

   The newly generate pick point will show up in the **Model Files** List.

5. Move the robot outside the camera's field of view. **Be careful not to touch the target object in this process to avoid altering its pose**.
6. Generate the point cloud model of the target object using the connected camera. Please follow the instructions in *Generate Point Cloud Model* of using a real camera.
7. In the **Model Files** List, select the pick point generated in step 4 and drag it onto the point cloud model to associate the pick point with the model.

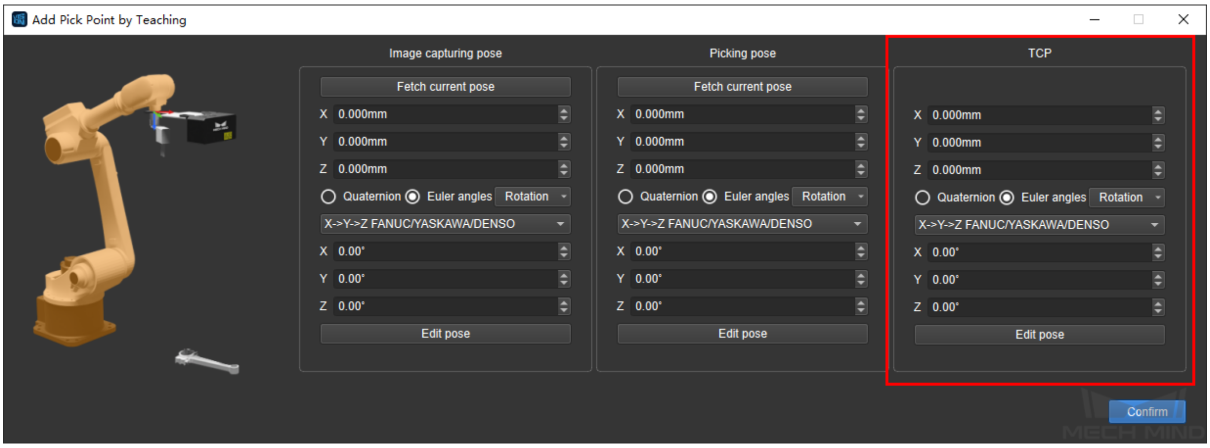
   Successfully associated pick point will be nested below the point cloud model.



Add pick point by teaching under EIH

Under EIH, you also need to obtain the image capturing pose in addition to the picking pose.

1. Click on  in the Toolbar to open the Add Pick Point by Teaching window.
2. Input the TCP obtained beforehand into the TCP section.



3. Move the robot to the picking pose using the teach pendant. Operate the end effector to perform picking to make sure the picking pose is accurate.

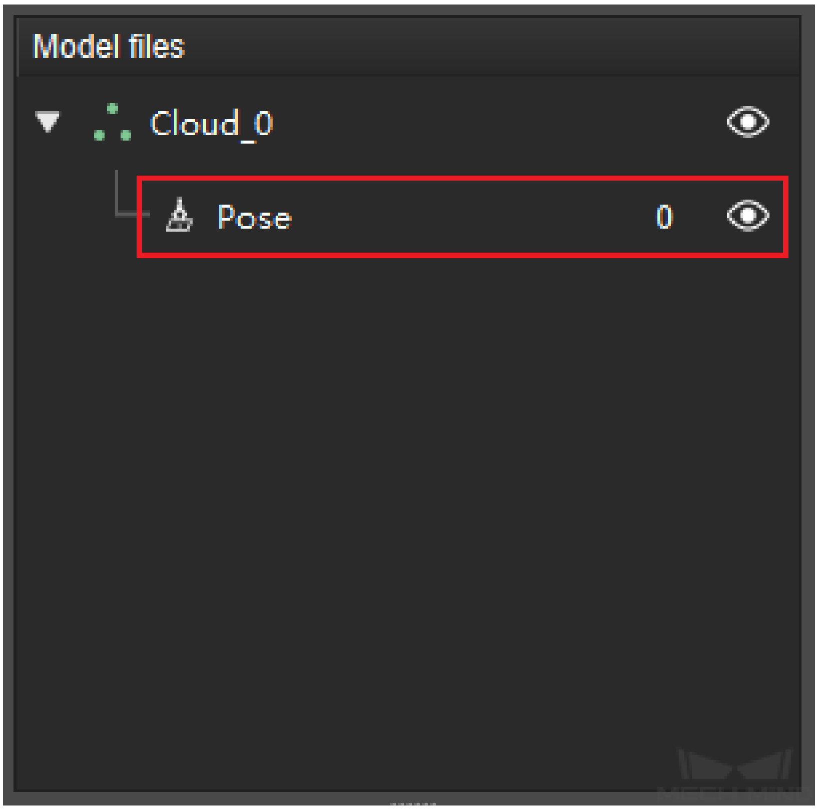
Hint: If you are using a fingered gripper, you can grasp and drop the target object several times to make sure that the object can be firmly grasped in this picking pose.

4. In the Picking Pose section, click on *Fetch current pose*, or input the pose displayed on the teach pendant manually.
5. Move the robot outside the camera’ s field of view. **Be careful not to touch the target object in this process to avoid altering its pose**.
6. Move the robot to the image capturing pose using the teach pendant. Capture an image to check if the pose is accurate.
7. In the Image Capturing Pose section, click on *Fetch current pose*, or input the pose displayed on the teach pendant manually. Click on *Confirm* to generate a pick point.

   The newly generate pick point will show up in the **Model Files** List.

8. Generate the point cloud model of the target object using the connected camera. Please follow the instructions in *Generate Point Cloud Model* of using a real camera.
9. In the **Model Files** List, select the pick point generated in step 7 and drag it onto the point cloud model to associate the pick point with the model.

   Successfully associated pick point will be nested below the point cloud model.



7.6.3 Relevant Steps

Some Steps need the files created in Matching Model and Pick Point Editor to run. You can create point cloud model files, pick point files, and geometric center files in Matching Model and Pick Point Editor.

- Steps that require point cloud model file(s) and pick point file(s): *3D Coarse Matching*, *3D Coarse Matching (Multiple models)*, *3D Fine Matching*, and *3D Fine Matching (Multiple Models)*.
- Step that requires a pick point file and a geometric center file: *Map to Multi Pick Points*.

7.7 Model Editor Tools (old)

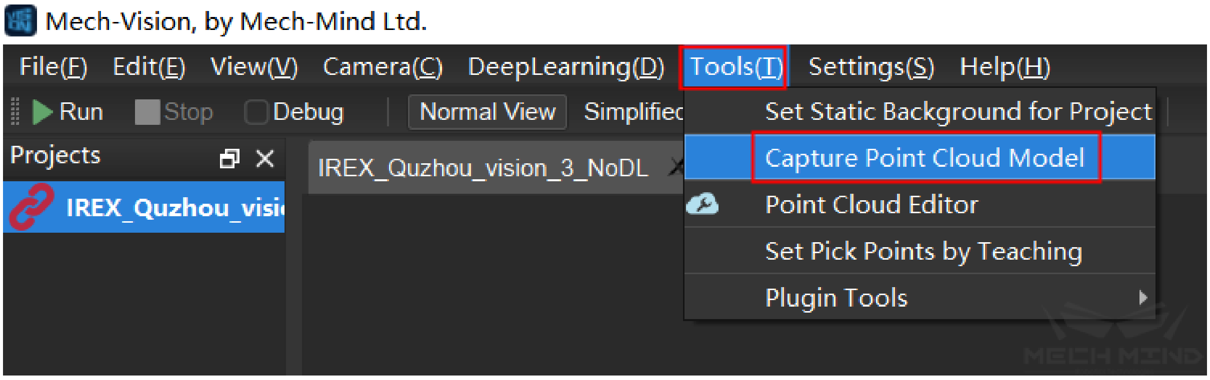
7.7.1 Capture Point Cloud Model

Capture Point Cloud Model Instruction

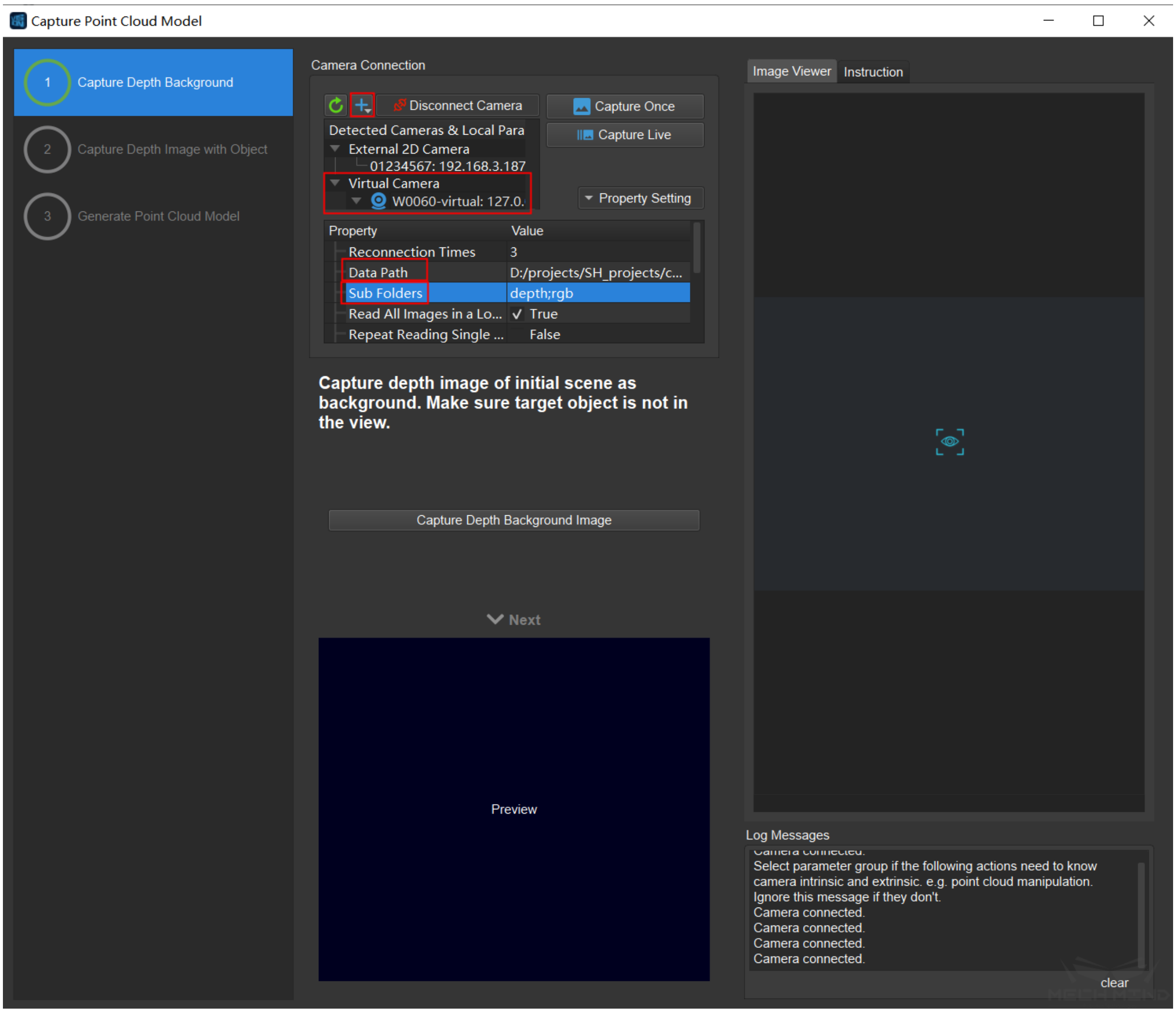
This tool is used to generate and save the target point cloud model files required by Steps such as 3D Coarse Matching and 3D Fine Matching.

Operating Procedure of Capturing Point Cloud Model

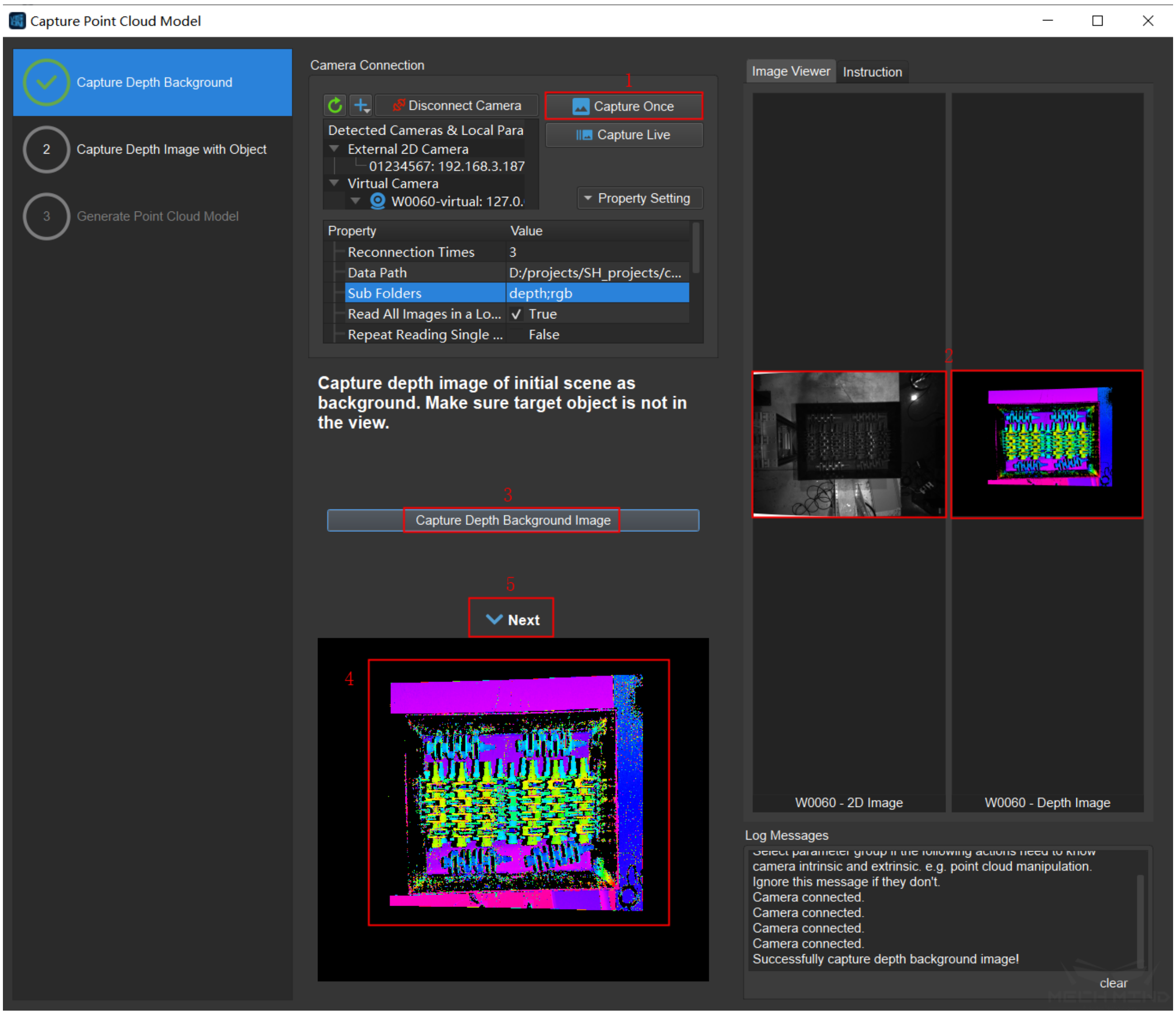
1. Open the Mech-Vision project and click *Tools* -> *Capture Point Cloud Model*.



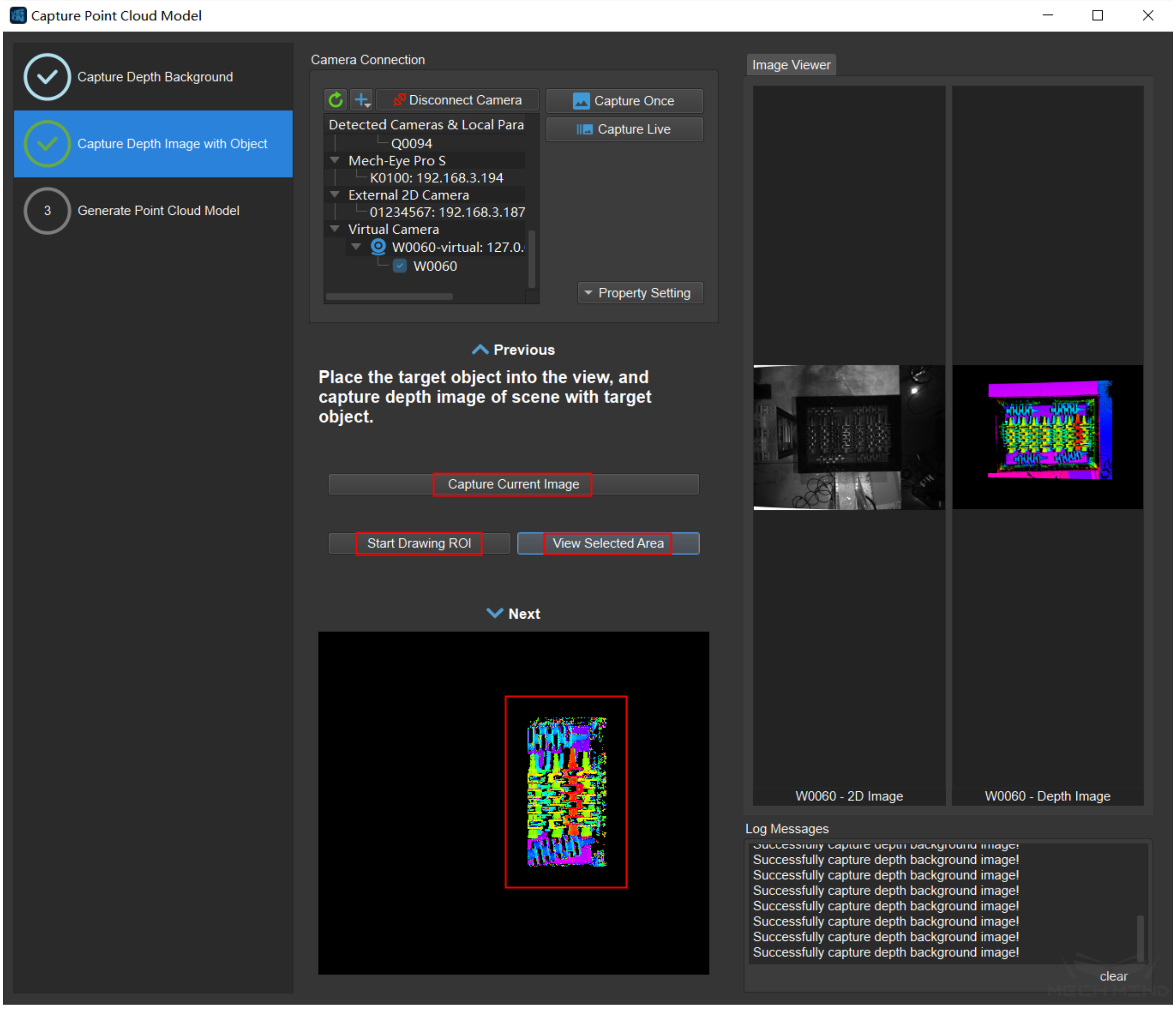
2. Connect to real camera and double-click the camera parameters group. Virtual camera is also available. Click the plus sign above the camera list to add a virtual camera. It is necessary to set the image data path and sub folders when using a virtual camera.



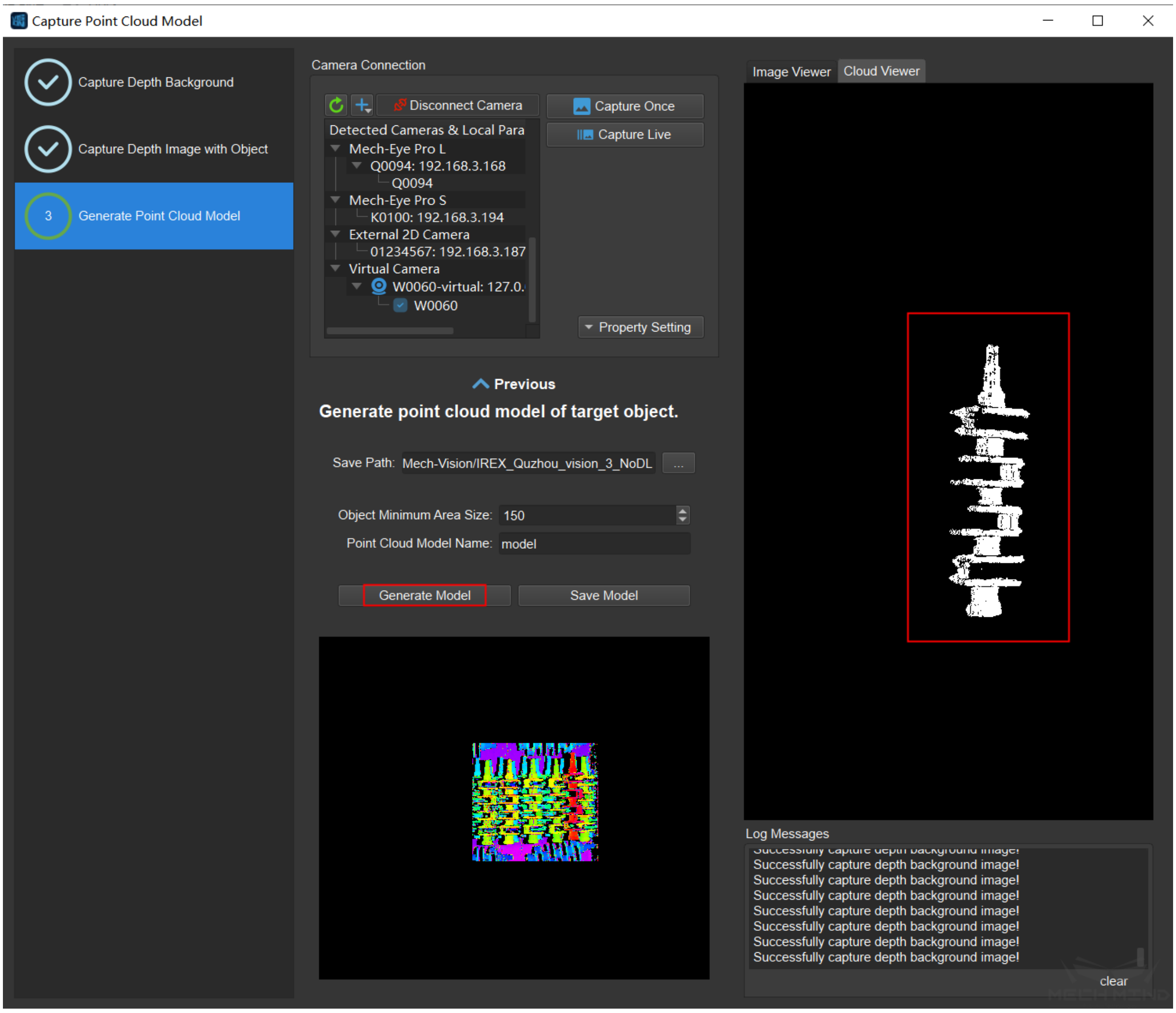
3. Click *Capture Once*, then the color image and depth image will be displayed in the image viewer on the right if the camera settings are correct. Make sure that the target object is not in the field of view, then click *Capture Depth Background Image*, and click *Next* after confirming that the background image in the preview box is correct.



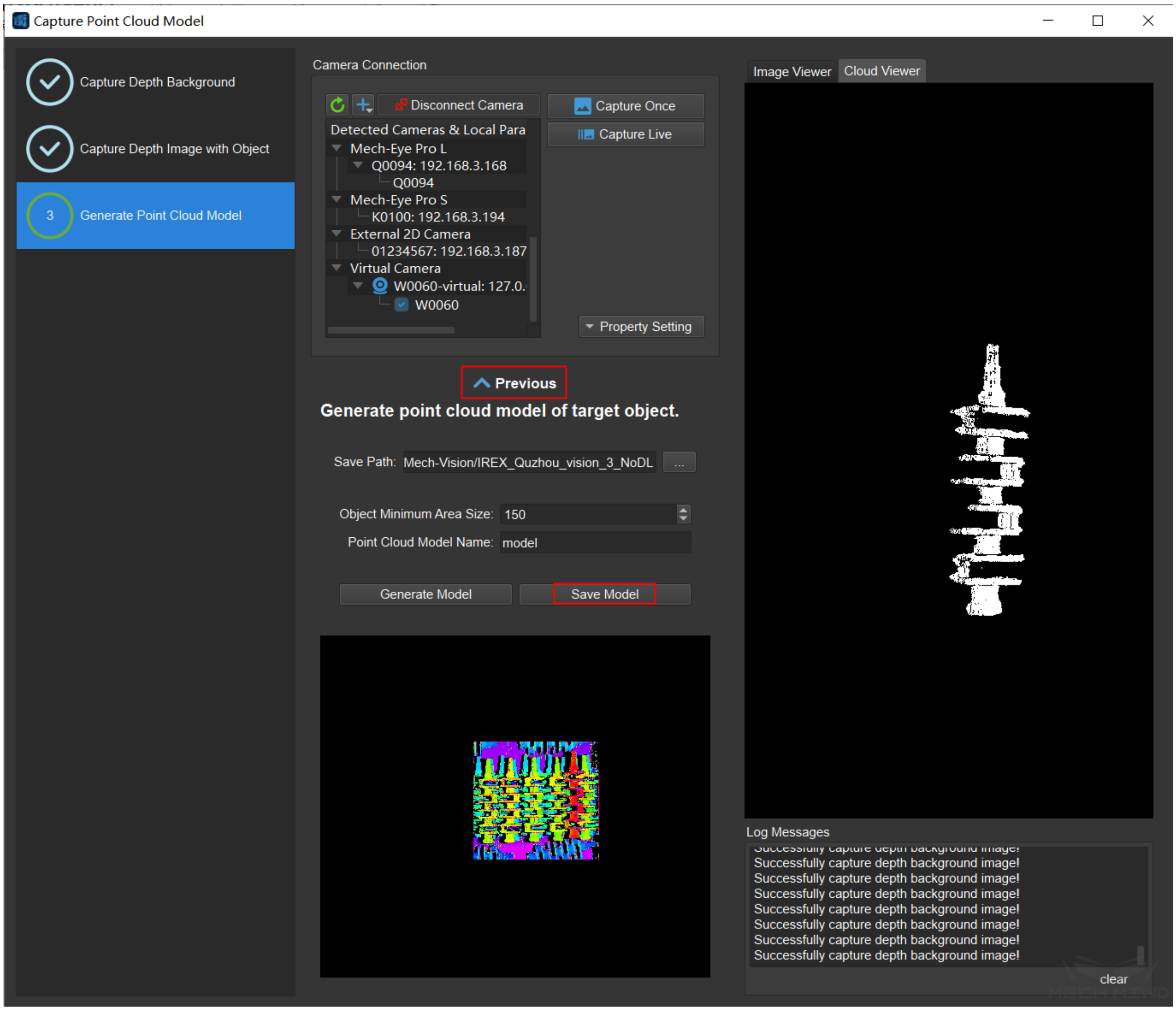
4. Place the target object in the field of view, click *Capture Current Image*, then click *Start Drawing ROI*, and draw the ROI box on the depth image. Click *View Selected Area* to confirm that the target object is in the ROI, and then click *Next* .



5. Modify the save path of the model file, the minimum area size of the object and the name of the point cloud model if needed. Then click *Generate Model*, the generated target object point cloud model will be displayed in the image viewer. Press and hold the left mouse key to rotate the point cloud, scroll the mouse wheel to zoom in and out, and press and hold the mouse wheel to pan the point cloud.



6. If the generated point cloud model is good, click *Save Model*. The model file will be saved to the set path. If the generated point cloud model is bad, click *Previous* to recapture image, draw ROI, and then try to generate model again.



7.7.2 Point Cloud Editor

Point Cloud Editor is for editing and processing point clouds, the point cloud editor mainly has the following functions:

1. Edit point cloud: filter out unnecessary point clouds and get a point cloud template suitable for subsequent processing.
2. Set pick point(s): add one or more pick points to the point cloud template and export a JSON file of pick points.
3. Merge all visible point clouds: merge multiple point cloud templates.
4. Downsampling: Downsampling the point cloud to reduce the workload of subsequent calculations while maintaining the point cloud structure information.
5. Zoom: zoom the point cloud to facilitate subsequent processing.

Interface

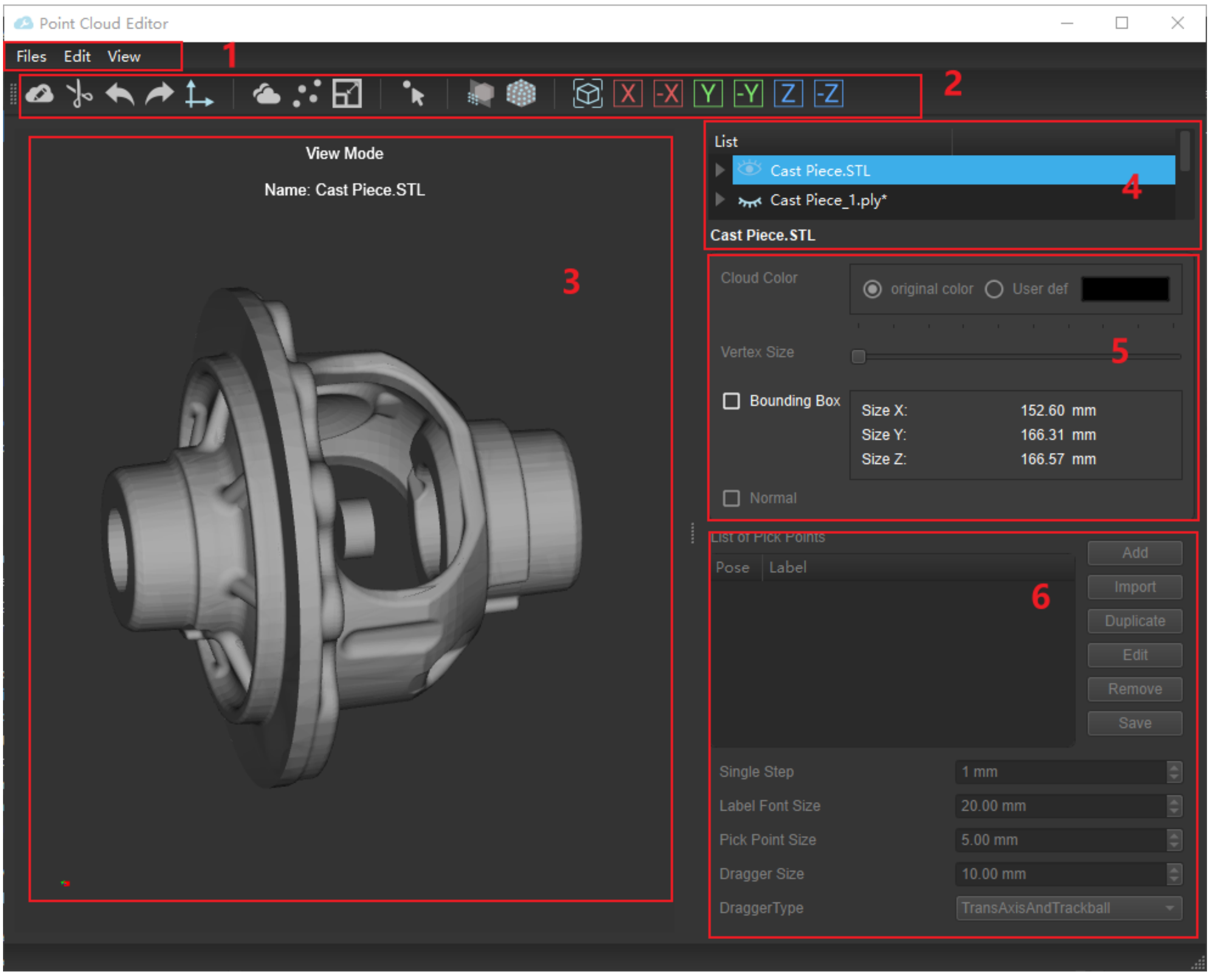


Figure 1. Point Cloud Editor Interface Introduction

The point cloud editor interface includes the following six sections:

Menu Bar	File: Open Cloud, Save Cloud As.
	Edit: Edit Point Cloud, Set Pick Points, Merge Visible Clouds, Downsample, Scale, Sample Cloud By View, Delete, Undo, Redo, Render Axis.
	View: Fit In Window, X View, Negative X View, Y View, Negative Y View, Z View, Negative Z View.
Toolbar	Shortcut buttons for editing tools.
3D simulation	Display point cloud and editing status.
Point cloud list	Display the opened point cloud file and its path information; set the visibility.
Point cloud parameters	Cloud Color: set the display color of the point cloud in the 3D simulation; for color point clouds, the original object color can be displayed.
	Vertex Size
	Bounding Box (check to display)
	Normal (check to display)
List of pick points	List of pick points: display the grab point information of the currently activated point cloud; select an item to activate for editing by selecting.
	Add: add a pick point to the currently active point cloud.
	Import: Load pick points and labels from THE file.
	Duplicate: copy the information of the selected pick point and generate a new pick point.
	Edit: Edit the currently selected pick point.
	Delete: delete the currently selected pick point.
	Save: save the pick point and label information in the pick point list.

Instructions

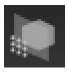
View a Point Cloud

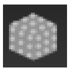
Click *File* → *Open Cloud* (shortcut Ctrl+O) to load an existing point cloud file.

Point cloud files are .stl or .ply files.

Import an .stl file:

Select the .stl file, click *Open*, and select the correct unit in the **Measurement Unit** window that pops up. If the selected unit is wrong, a warning window asking for whether to continue loading will pop up. The loaded .stl file will be added to the **Point cloud list** in the upper right corner, and the point cloud will be displayed in the 3D simulation on the left.

Sample cloud from STL model by view After selecting the .stl file in the point cloud list, click  in the toolbar to generate the point cloud of the part of the surface visible from the current perspective. In the point cloud list, the .ply file of the current point cloud will also be added.

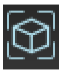
Sample cloud from STL model by out After selecting the .stl file in the point cloud list, click  in the toolbar, and the **Sampling Interval** window will pop up. After completing the settings, a complete external surface point cloud will be generated. In the point cloud list, the .ply file of the current point cloud will also be added.

Import a .ply file::

Select the target file and click *Open*. The loaded .ply file will be added to the **point cloud list**, and the point cloud will be displayed in the 3D simulation on the left.

- **Change the perspective of viewing the point cloud:**

  Drag the point cloud with the left mouse button to view the point cloud objects from different perspectives.

  Click  to fit in the window.

  Click 

 to view the point cloud from the front, back, top, bottom, left, and right respectively.

- **Zoom or enlarge to view the point cloud:**

  Scroll to zoom the point cloud.

- **Point cloud visibility settings:**

  When multiple point clouds are open, this function helps to view each point cloud separately.

  The eye icon to the left of the point cloud name in the point cloud list specifies the visibility of the point cloud. After clicking to make it a closed eye icon, the point cloud will be hidden.

- **Merge all visible point clouds:**

  1. In the point cloud list, set the point clouds to merge as visible, set the ones not to merge as hidden, as shown in *Figure 2*.

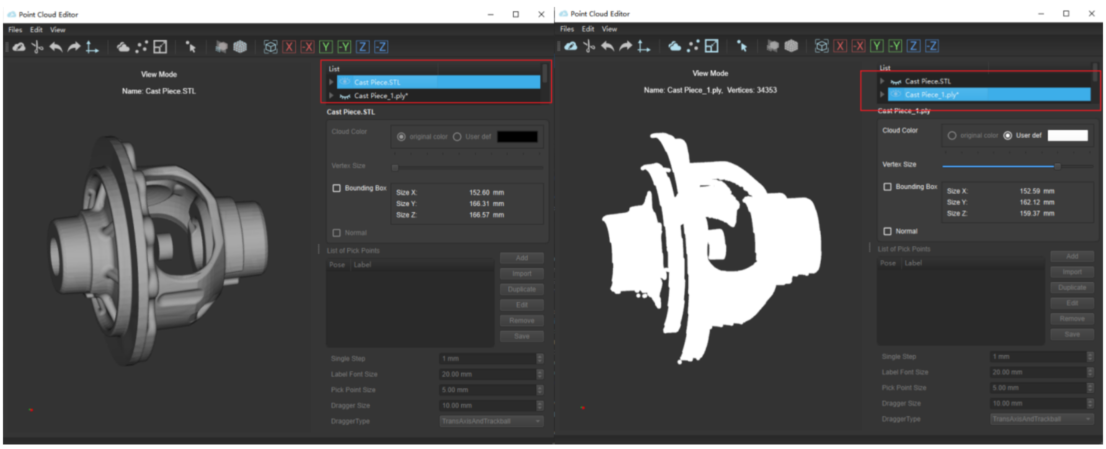
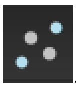


Figure 2. Set Point Cloud as Visible or Hidden

2. Click  to select the save path of the new point cloud.

- **Downsample:**

  1. Select the point cloud to downsample in the point cloud list on the right, and click .
  2. Set the appropriate sampling interval, click *Confirm*, the down-sampled point cloud will be automatically generated and the file name will be displayed in the point cloud list (at this time, the previously selected point cloud will be automatically hidden). The comparison of the point clouds before and after adjusting the sampling interval is shown in *Figure 3*.

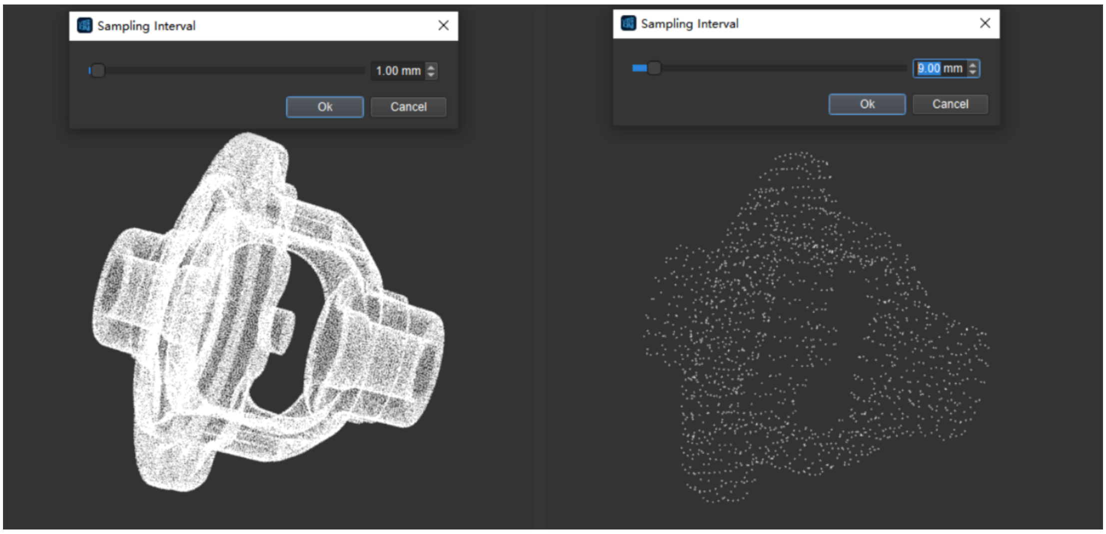
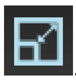


Figure 3. Point Cloud Downsampling

> **Attention:** The point cloud after downsampling will not be saved automatically. Please save it manually.

- **Scale:**

  1. Select the point cloud to scale in the point cloud list on the right, and click .
  2. Set the scale value in the X, Y, and Z directions. By default, the option of equal values for each direction is checked. Click *Confirm* and the zoomed point cloud template will not generate a new point cloud template, only the zoomed effect will be displayed in the current template. The comparison chart before and after scaling is shown in *Figure 4*.

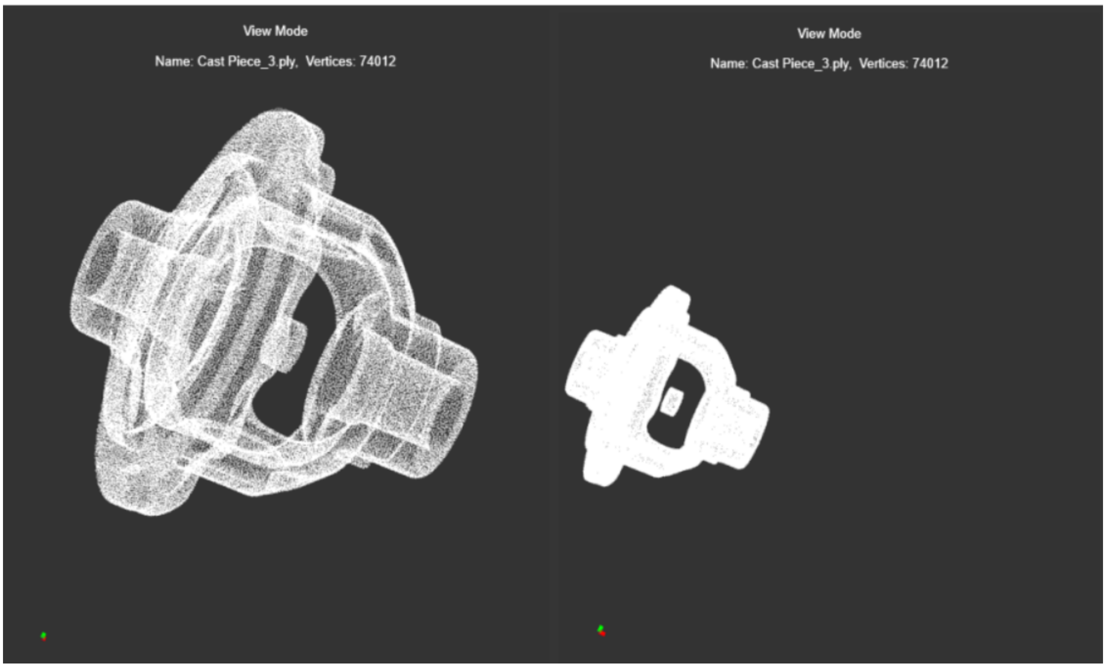


Figure 4. Scale

Attention: The scaled point cloud will not be saved automatically. Please save it manually.

Point Cloud Parameters

1. Customize point cloud color

   Set the display color of the point cloud in the 3D scene. For color point clouds, the original object color can be displayed. The comparison of point clouds before and after customizing color is shown in *Figure 5* .

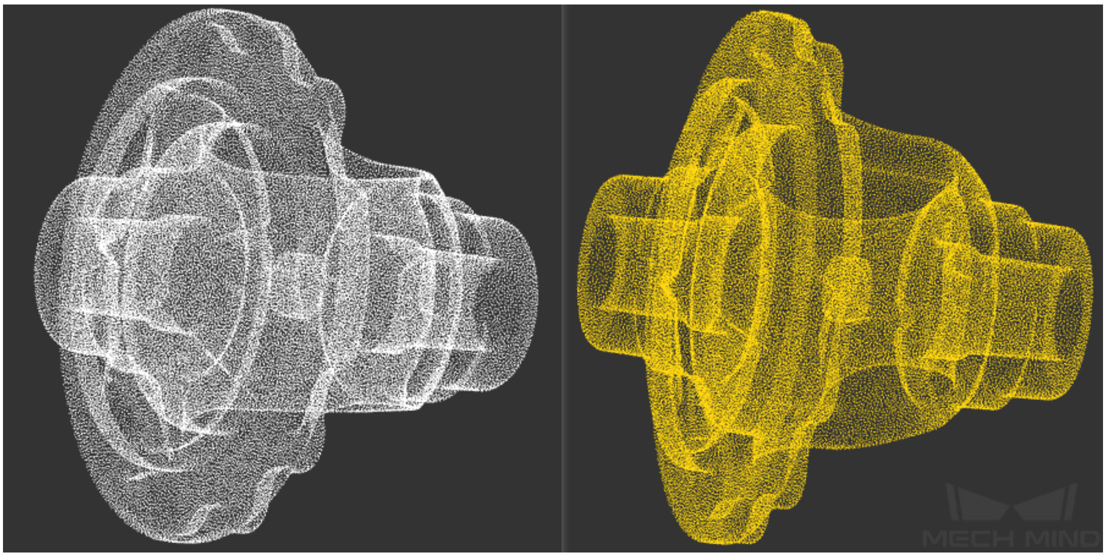


Figure 5. Customize Point Cloud Color

2. Vertex size

   This feature is to change the vertex size of the displayed point cloud. The comparison of point clouds before and after vertex adjustment is shown in *Figure 6* .

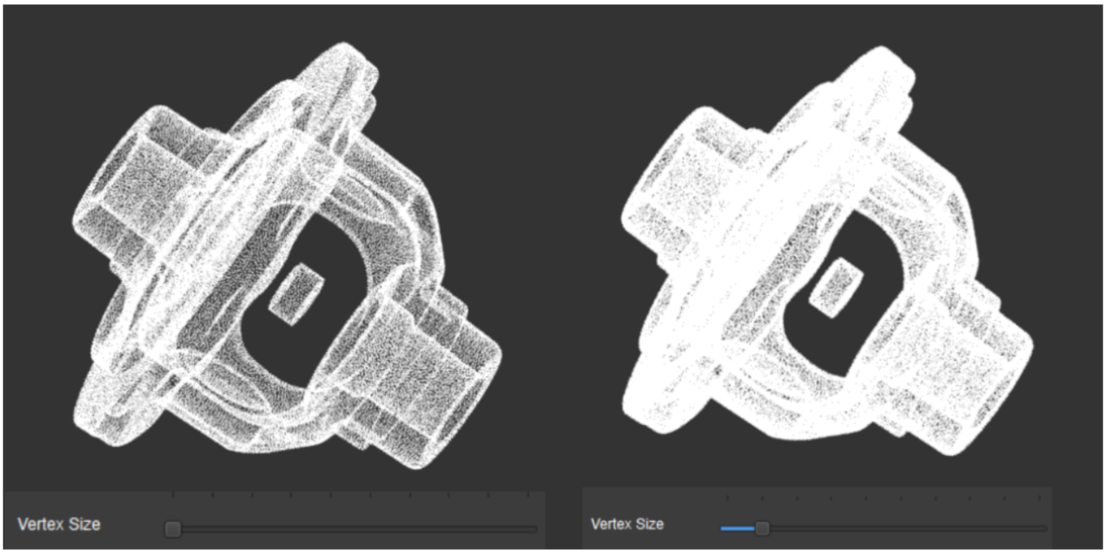


Figure 6. Vertex Size

3. Display bounding box

   The comparison of the display before and after displaying the bounding box is shown in *Figure 7* .

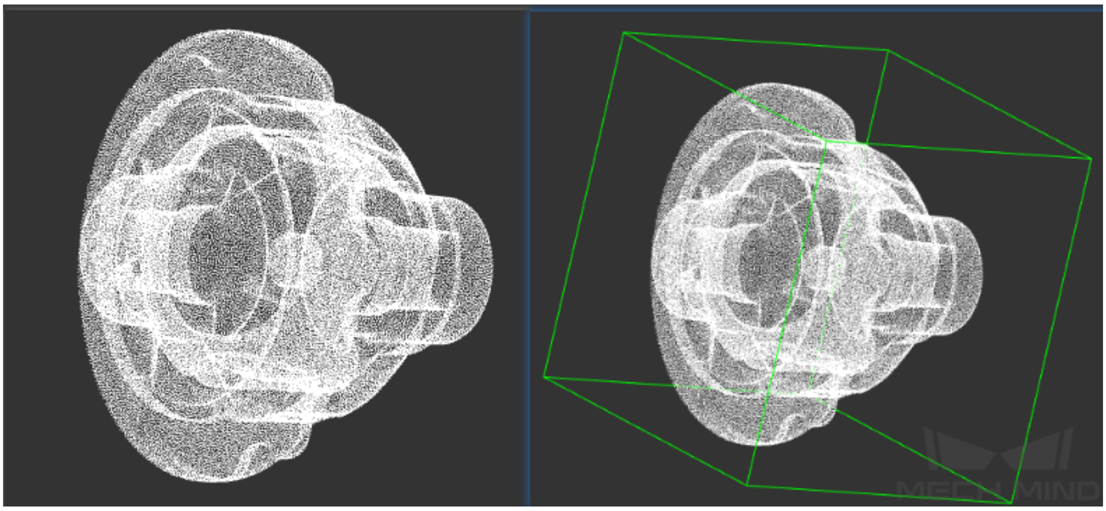


Figure 7. Displaying the Bounding Box

4. Display the normals

   The comparison of the display before and after displaying the normal is shown in *Figure 8* .

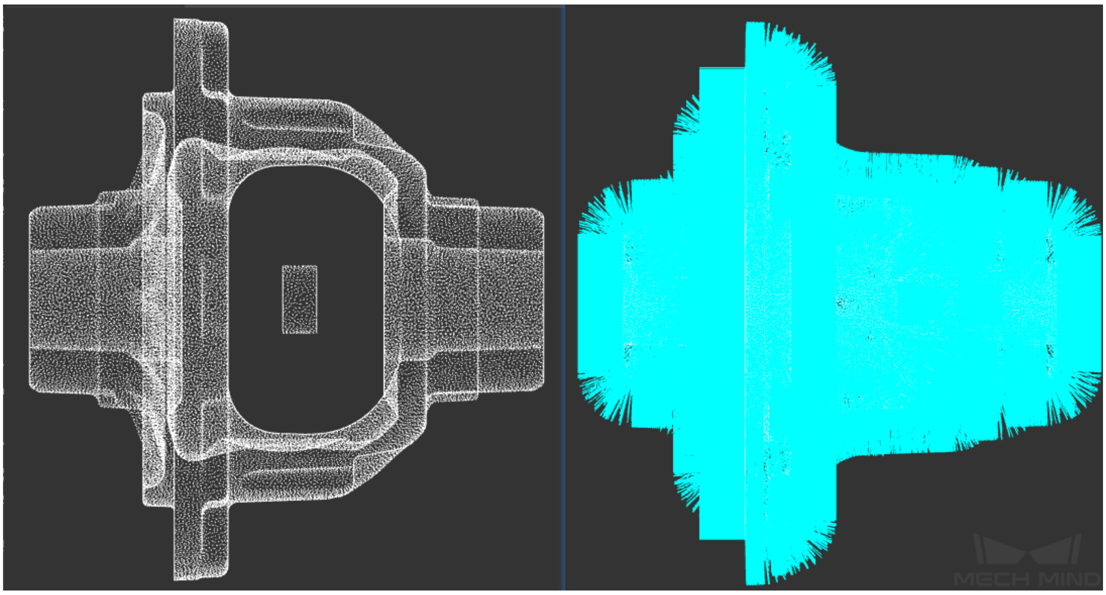


Figure 8. Display the Normals

Make a Point Cloud Template

After a certain point cloud in the point cloud list is selected, the point cloud is active, and all editing operations will only act on the activated point cloud (the point cloud needs to be visible at the same time). If there are multiple point clouds in the list, only the selected point cloud (highlighted) is active.

> **Attention:** Visibility and activation state are two different states of the point cloud. Invisible point clouds can be activated but cannot be edited. Inactive point clouds can be made visible. Only point clouds that are visible and activated at the same time can be edited.

1. After finding a suitable perspective, click  in the toolbar (shortcut Alt+M) to enter the point cloud editing mode.
2. Select the area to delete. If the selected area is not ideal, please reselect it. Hold Ctrl to select multiple different areas at the same time.

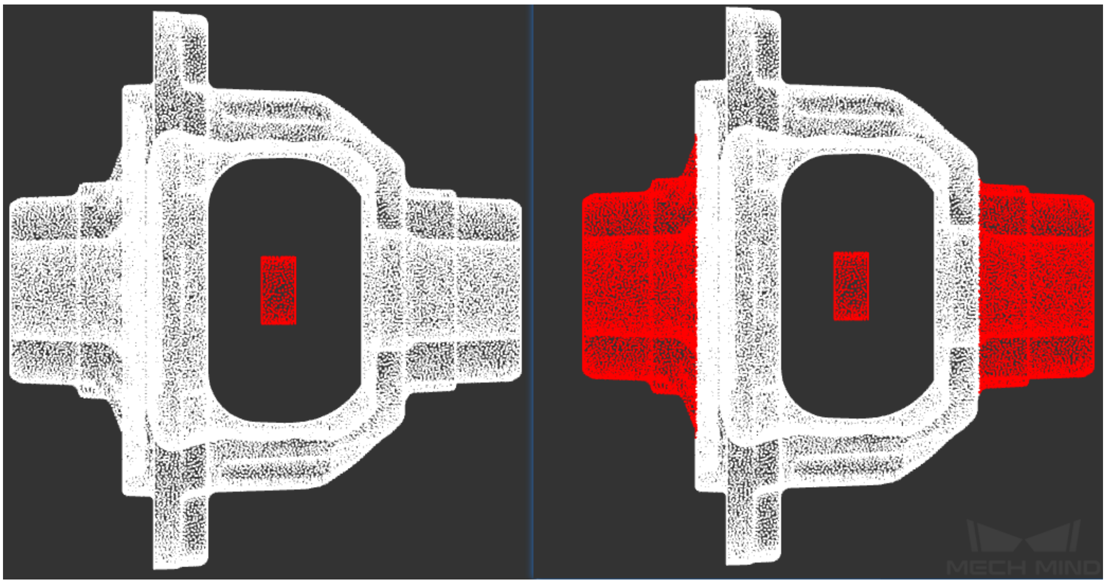


Figure 9. Select Area

3. Click  (shortcut key Delete) to delete the selected points, then the editor will enter the point cloud view mode. If further editing is needed, please re-enter the point cloud editing mode. A delete operation can be undone by clicking .
4. After editing, please save the edited point cloud. Click *File* → *Save loud As* or *Right click the current point cloud file* → *Save Cloud As* (shortcut Ctrl+S) in the point cloud list, select the path, enter the file name and save.
5. Reload the saved edited point cloud file to see that the editing has been saved correctly, as shown in *Figure 10*.

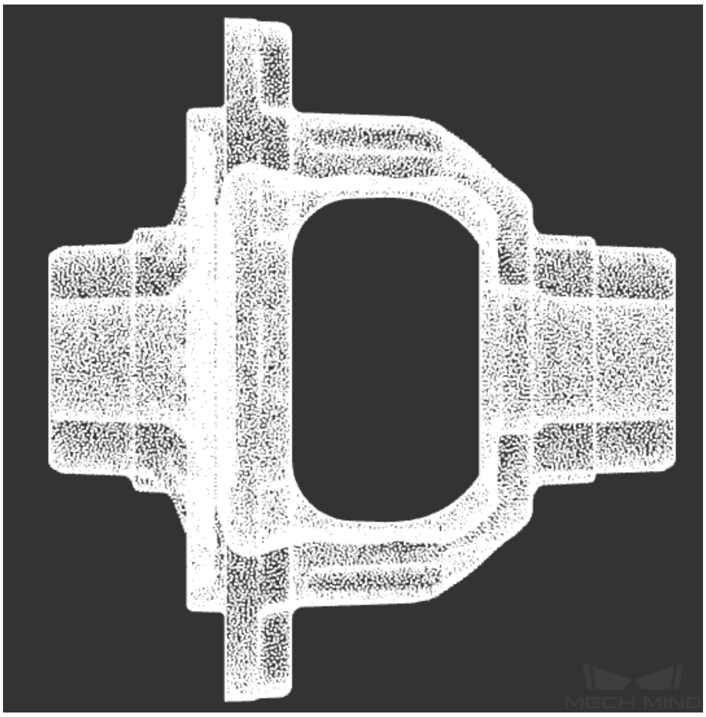


Figure 10. Edited Point Cloud

Set Pick Point

The procedure for setting the pick point is as follows:

1. In the point cloud list on the right, select the point cloud that needs pick point settings, and click .
2. Click *Add* or *Import* on the right to set one or more pick points (when importing, if a pick point has a label, please import the corresponding label file).
3. Or set the pose of a selected pick point by dragging the dragger (trackball or transaxis or both) of the pick point. The properties of the dragger can be set in the lower right corner, as shown in *Figure 11*.
4. Or click *Edit* on the right to set the pick point.

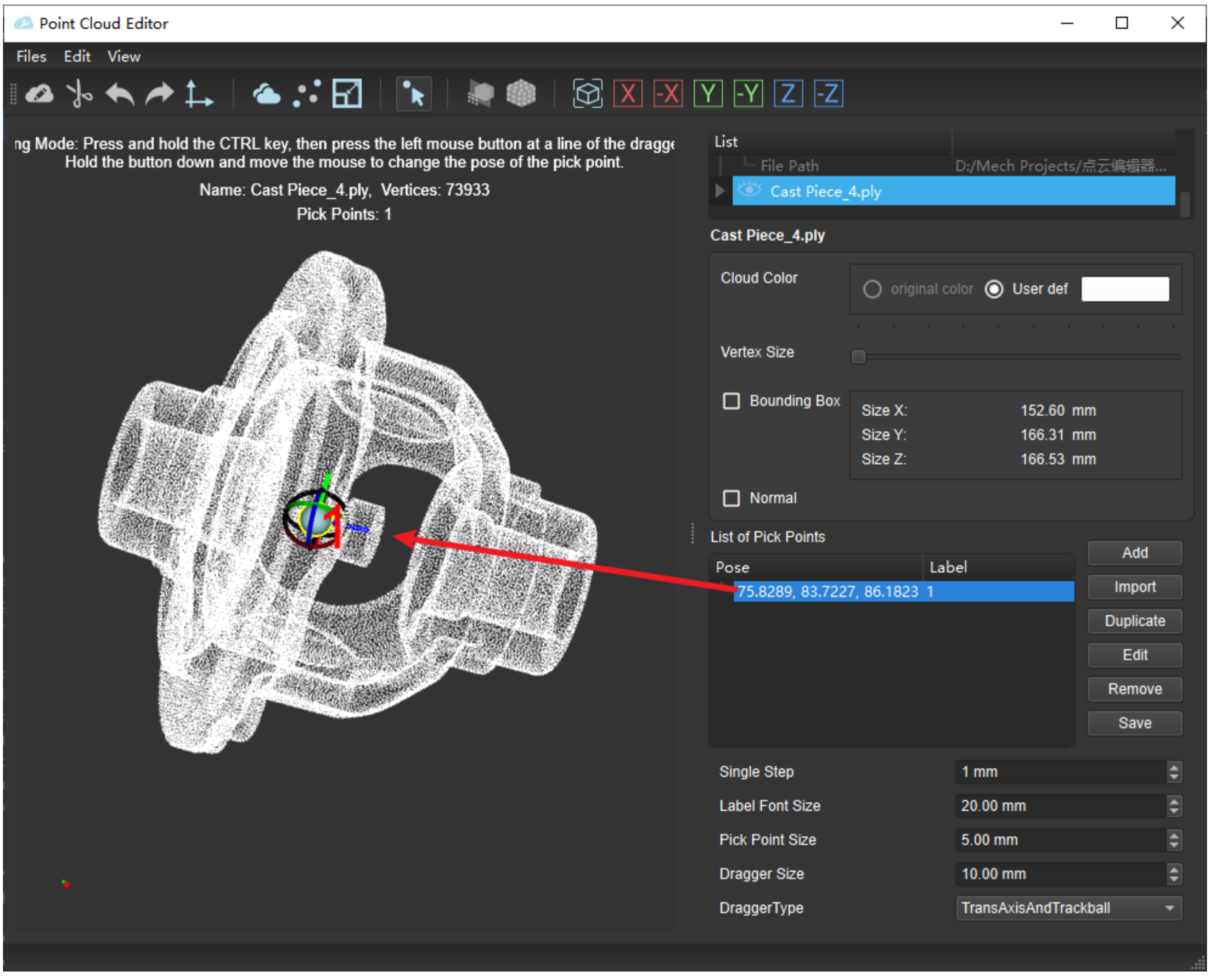


Figure 11. Set Pick Point

Tip: The edit window of a pose in the pick point list can be directly opened by double-clicking on the pose. The single step length, label font size, pick point display size, and dragger type can be edited by actual needs.

5. When done setting the pick points, click *Save*, select the file save path, and finally generate the pick point JSON file.

7.7.3 Set Pick Points by Teaching

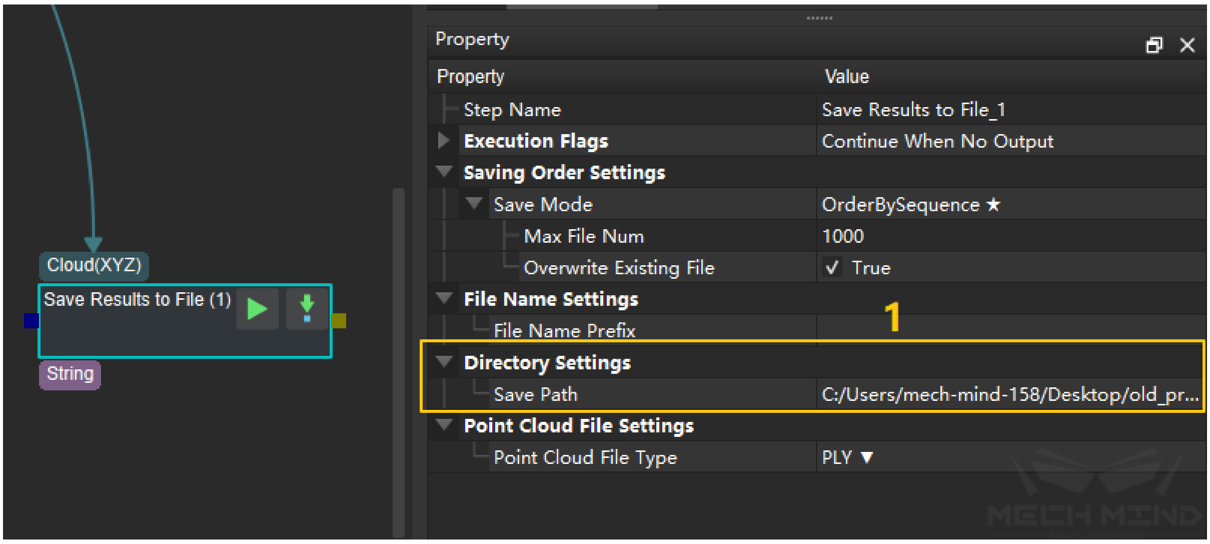
Pick Point Setting Usage Scene (Teaching Method)

1. For the project required high precision of pick point, the precision requires within 0.5mm.
2. The accuracy of TCP pose is not sufficient.
3. The pose of the workpieces are relatively consistent in the entire process, otherwise you need to use the multi-templates.

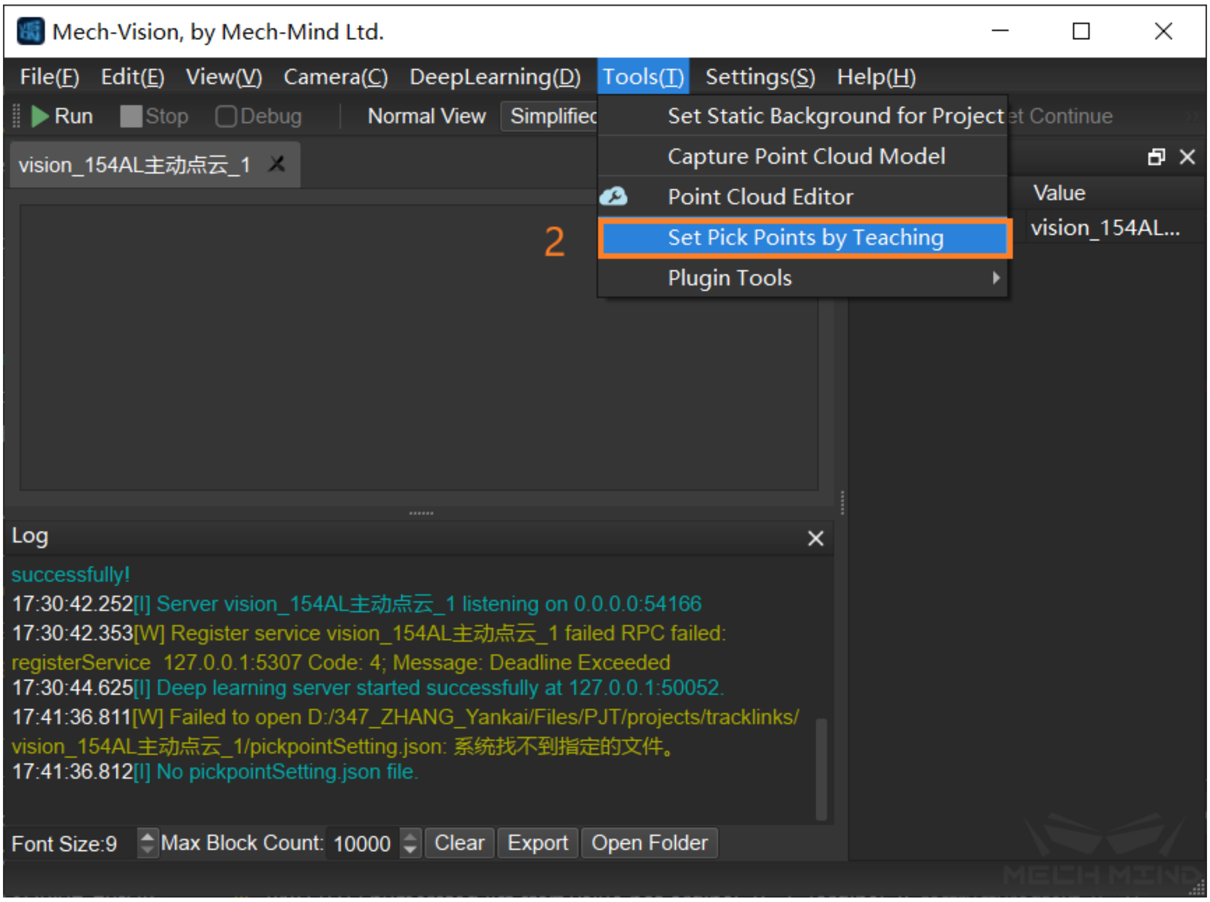
Operating Procedure of Setting Pick Points by Teaching

Method of making template

Release End effector, make sure the position of workpiece has not changed, take a photo at shooting point to have depth image and then generate point cloud template file(.ply) by corresponding Mech-Vision project.

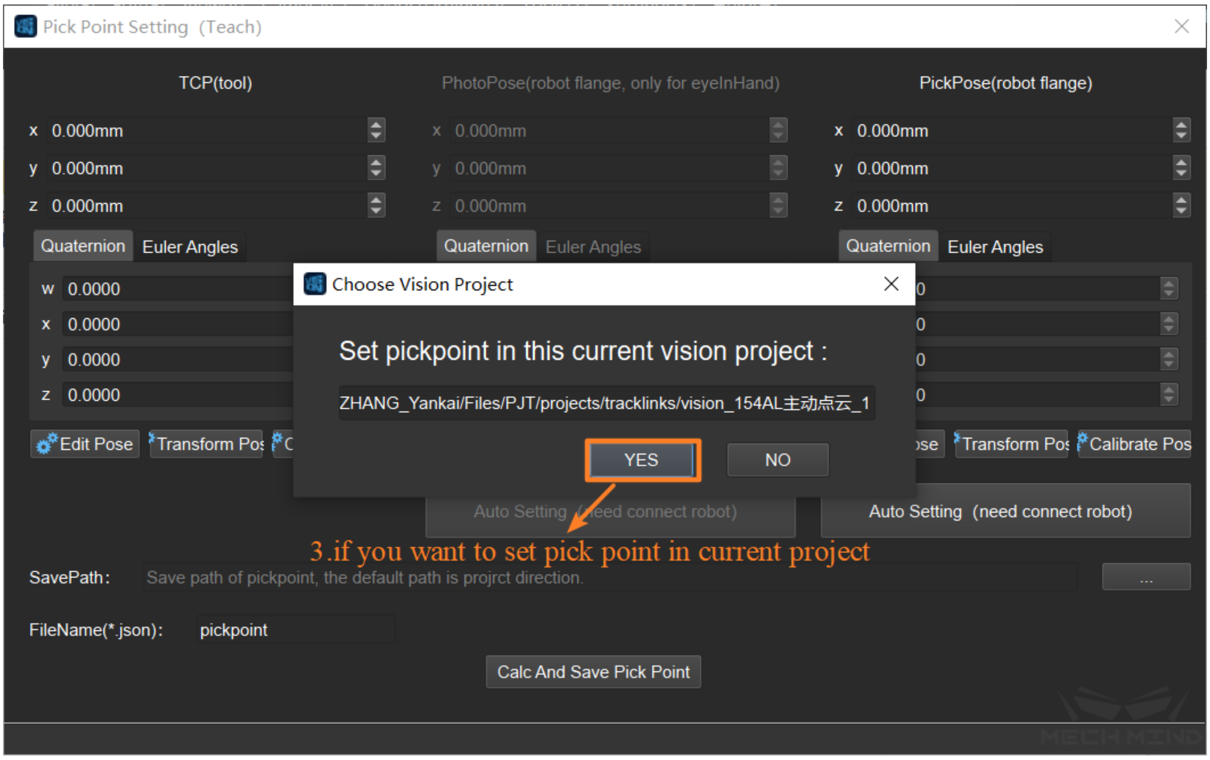


Mech-Vision->Tool(T)->Set Pick Points by Teaching



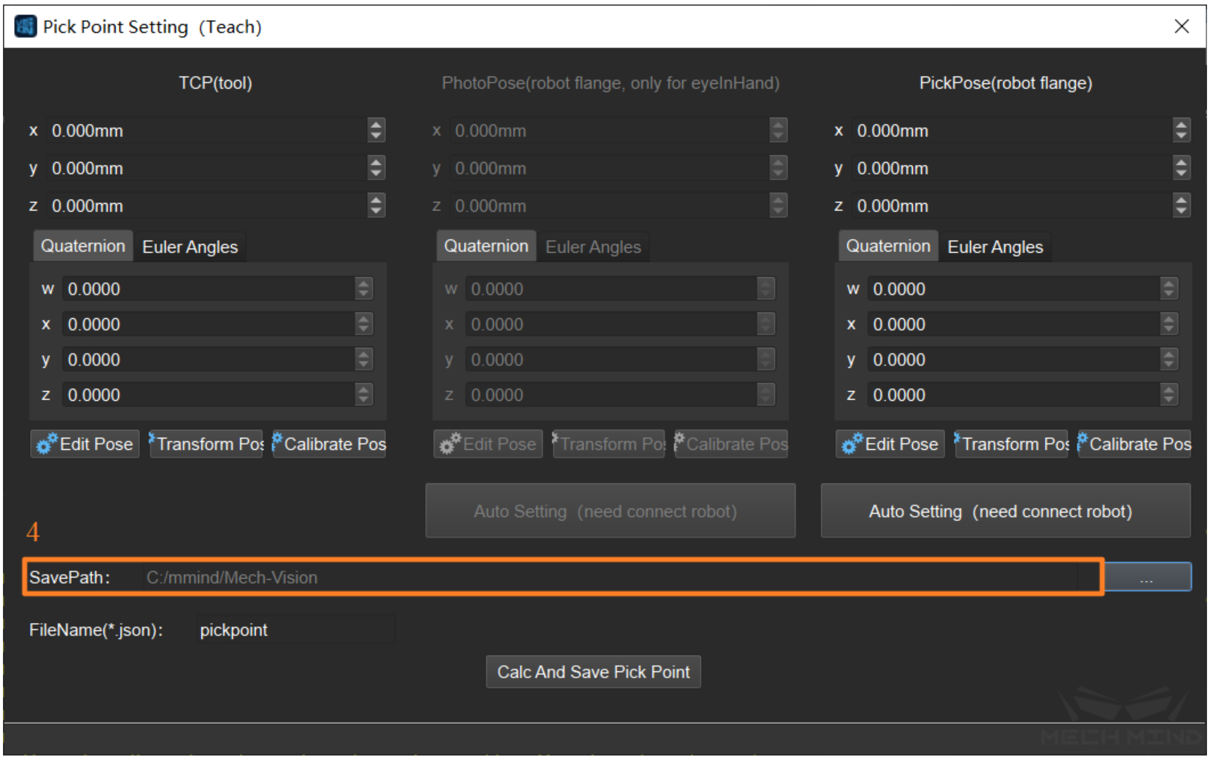
Select the corresponding project for pick point setting

> **Attention:** After updating extrinsic parameters, pick point has to reset, otherwise it will cause errors



Select saving path

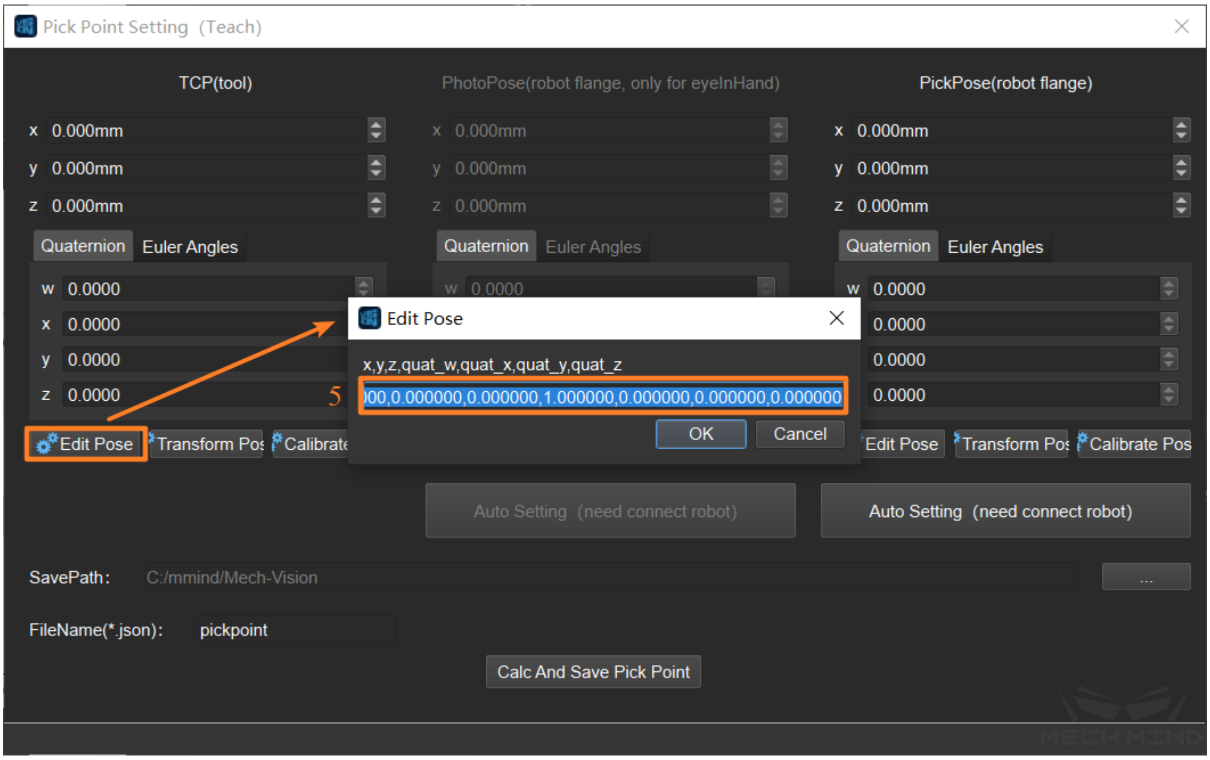
Attention: The default path for saving pick points template is the project directory.



Set TCP Pose

Input TCP coordinate manually

> **Attention:** If it doesn' t set TCP, the default tcp can be (0,0,0,1,0,0,0), which flange center point will be the pick point.



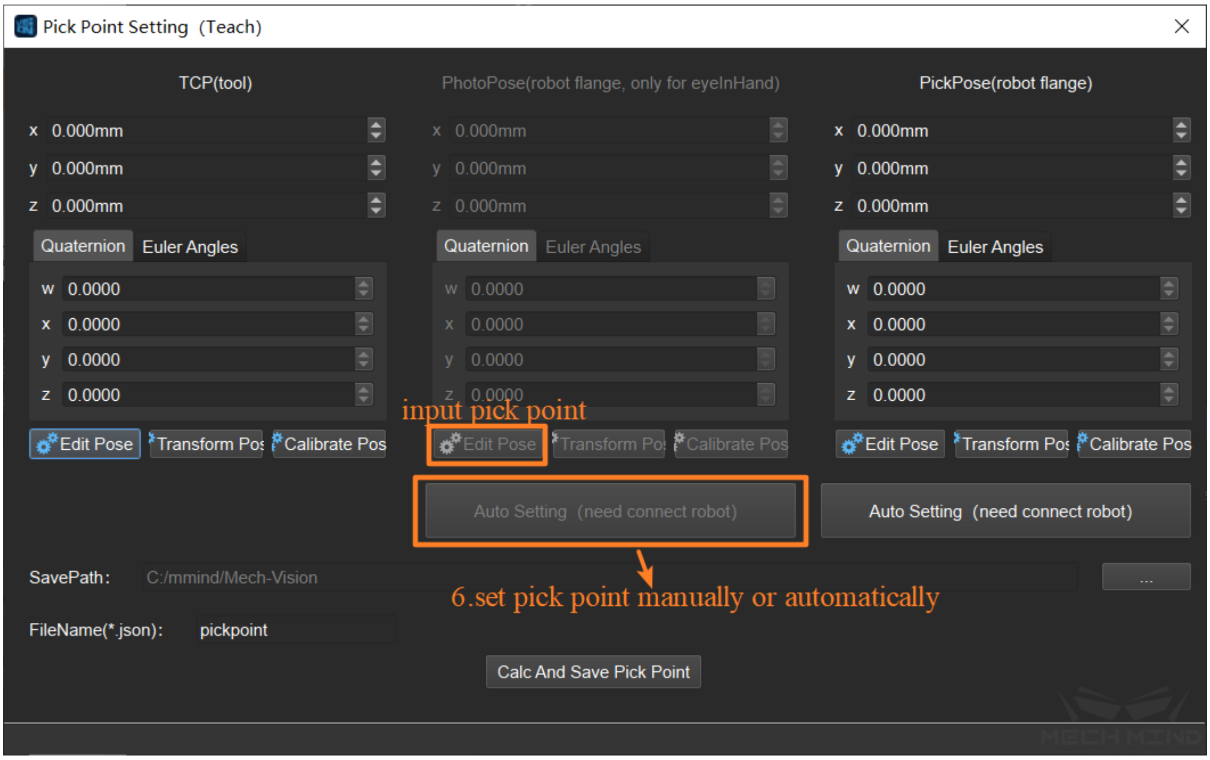
Set photo point (required only in the Eye In Hand situation)

1. Set photo point automatically: connect robot, move the robot to the photo point, click "Auto Setting" to set photo point and it will obtain the flange coordinate of the robot as the photo point automatically.
2. Set photo point manually: connect robot, move the robot to photo point, input flange center pose as photo point manually.

Attention:

A. If the photo point is input manually, the subsequent pick point setting should also be input manually.

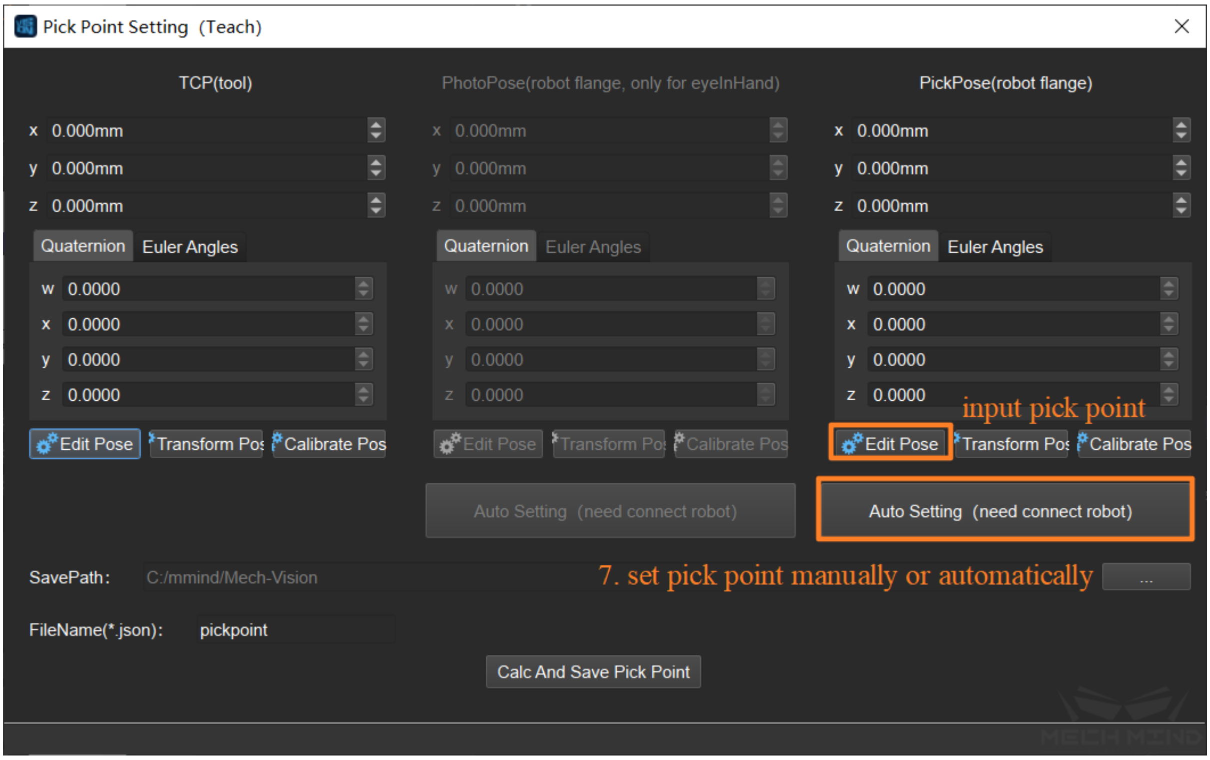
B. It has to input flange center pose, instead of TCP pose.



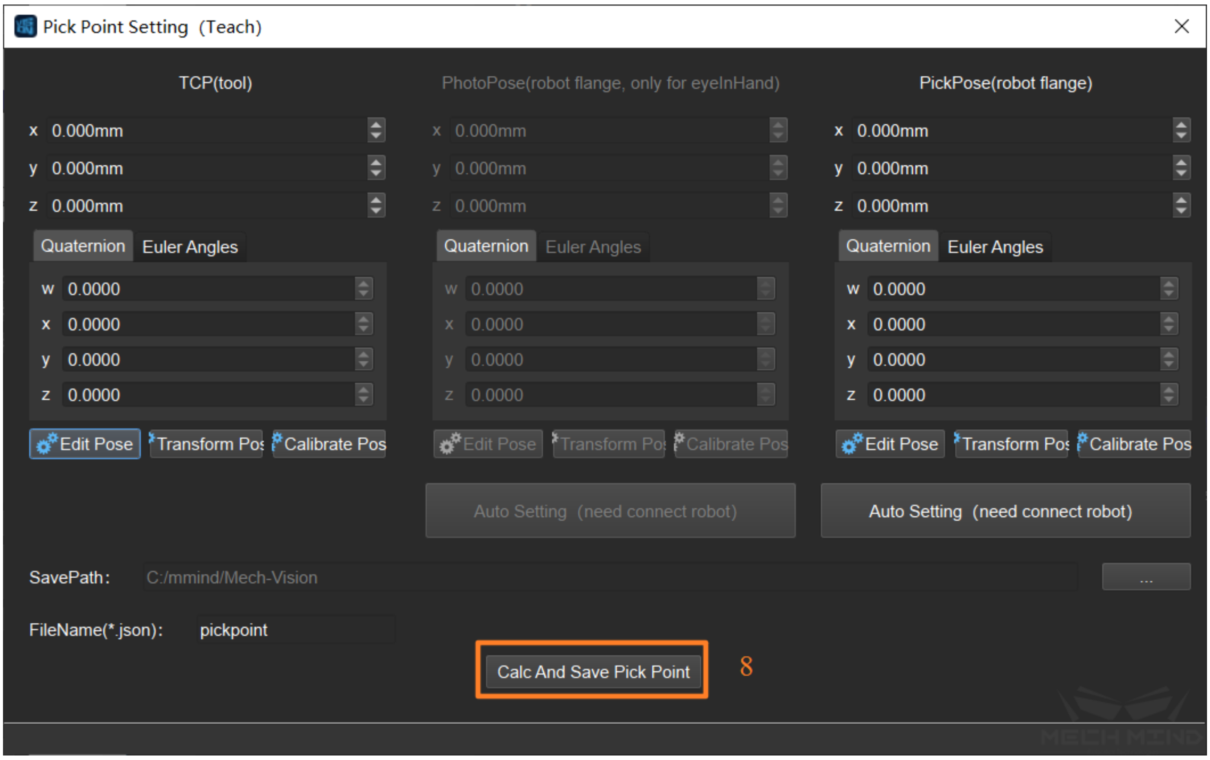
Set pick point

1. Set pick point automatically: connect robot, move the robot to the pick point, pick the object, click "Auto Setting" to set pick point.
2. Set pick point manually: connect the robot, move the robot to the pick point, pick the object, input robot flange center pose manually.

Attention: In order to make sure the pick point and photo point are under the same coordinate system, it has to ensure that the pose of object can not change before or after the end effector is enabled or release. You can clamped and release the object repeatedly, and set the pick point after the object is stable.


Pick Point Setting (Teach)
TCP(tool)
PhotoPose(robot flange, only for eyeInHand)
PickPose(robot flange)
x 0.000mm
y 0.000mm
z 0.000mm
Quaternion
Euler Angles
w 0.0000
x 0.0000
y 0.0000
z 0.0000
input pick point
Edit Pose
Transform Pos
Calibrate Pos
Auto Setting (need connect robot)
SavePath:
C:/mmind/Mech-Vision
7. set pick point manually or automatically
FileName(*.json):
pickpoint
Calc And Save Pick Point

Click "Calc and Save Pick Point" (the file of pick point is saved under the path of corresponding project)

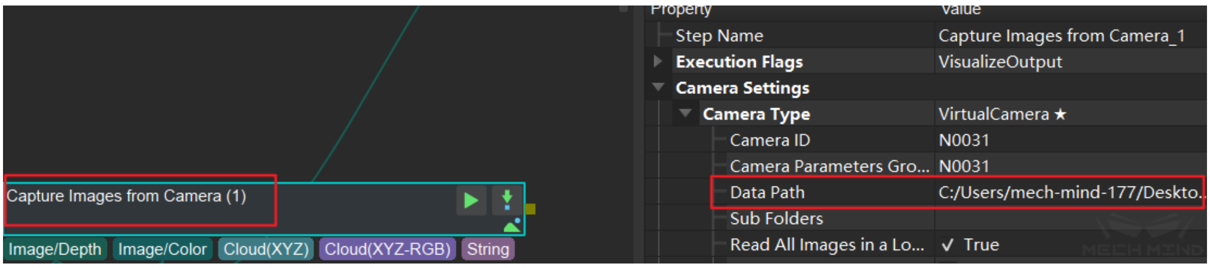


Attention: For the situation that the object is huge, which needs double cameras to recognize: the first step Capture Images from Camera is the secondry camera, and the second step Capture Images from Camera is the main camera. It has to run the whole Mech-Vision project before setting pick points.

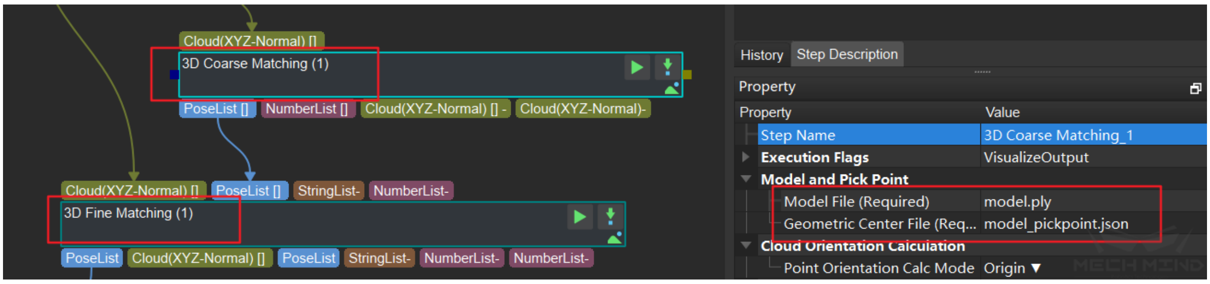
Verify the correctness of pick point setting

Method 1: (Preferable)

1. The directory where the RGB image and depth image for making template can be saved in the "dataPath" of step Capture Images from Camera.

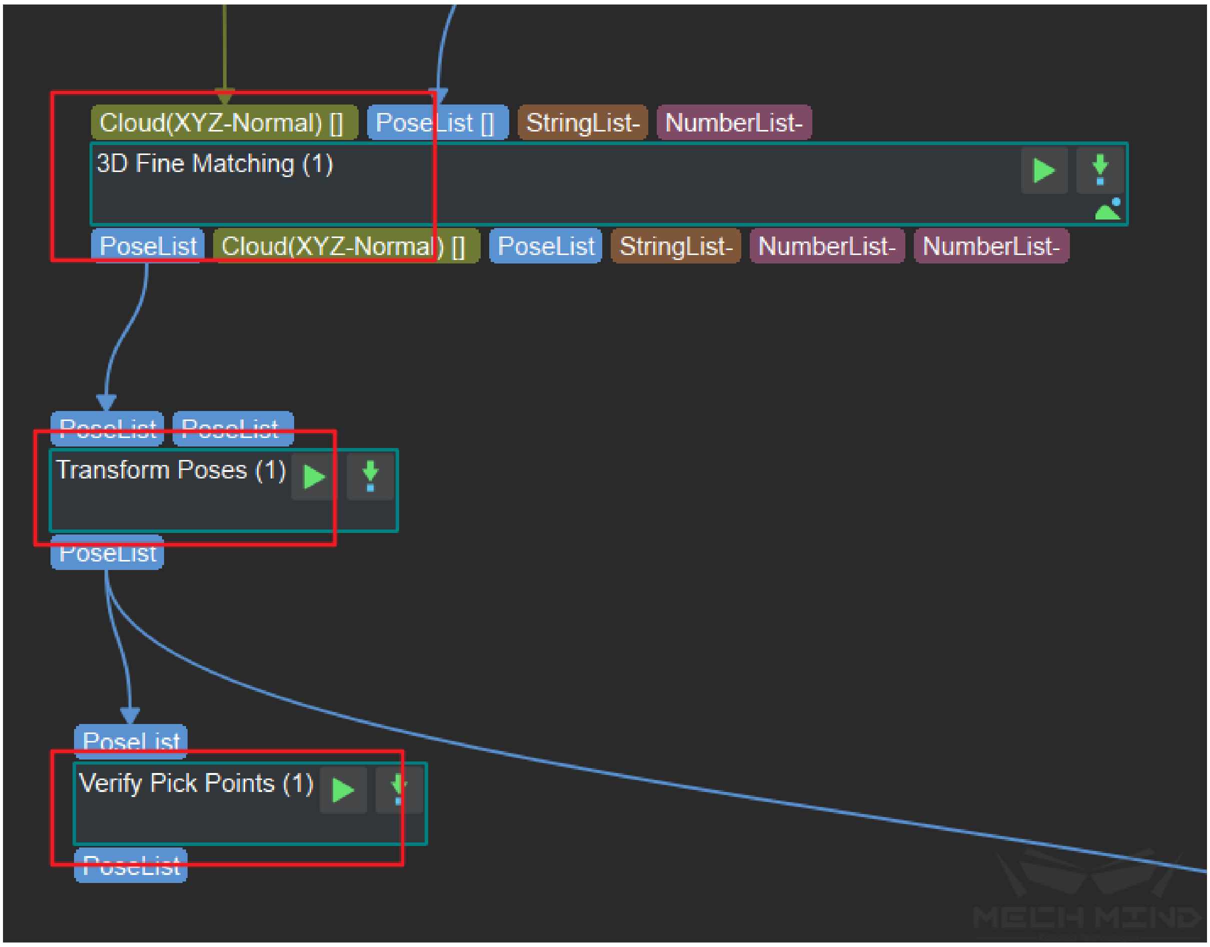


2. Set the template file and pick point file into step 3D Coarse Matching and 3D Fine Matching.

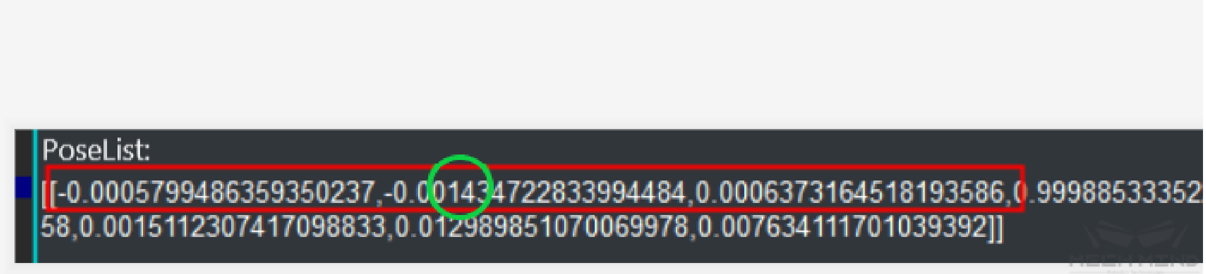


3. Connect the step Verify Pick Points after the step *Transform Poses* in Mech-Vision project, run and then achieve the verification environment of template self-matching scenario.

Attention: During this process, make sure the pose of object does not changed



4. The specific meaning of the pose output from the step Verify Pick Points is the difference between the pick point pose set by user and the pick point pose output from Mech-Vision project. As the figure shown below, observe "TextOutput" of the step Verify Pick Points , the difference along Y axis of pick point pose is 1.4mm, the error is relatively greater.



Attention: It has to make sure that the error of *3D Fine Matching* is very small, otherwise the output error of Verify Pick Points step is bigger, because it is consisted of matching error and pick point error.

Method 2

Load new template and pick points files, run the corresponding project in Mech-Vision. Use Mech-Viz to control the robot to pick the object which is used to set pick point. If it can accurately pick the object, that proves the pick points is set correctly.

Attention: If the setting of pick point has some problems, it may cause collision. If you want to use this method, be cautious to operate robot.

7.8 Data Playback

7.8.1 Data Playback Introduction

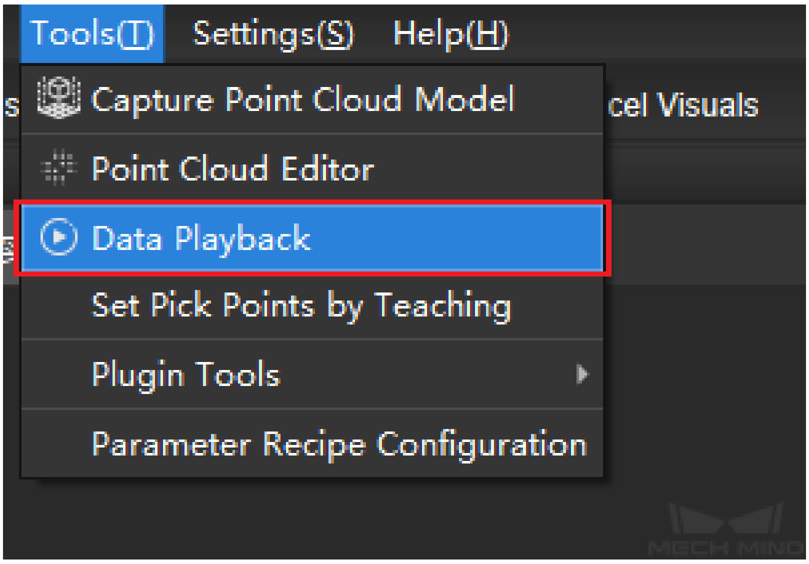
The data playback tool can playback and verify a large amount of data in the project to find and correct small problems, so as to improve the efficiency of the regression test in the final stage of the project.

There are two modes of Data Playback tool:

- *All Data Playback*: Replay a large amount of data quickly to pick out problematic data, or select and save excellent data for archiving.
- *Invalid Data Playback*: Perform playback verification on a specific data group, mostly after adjusting the project.

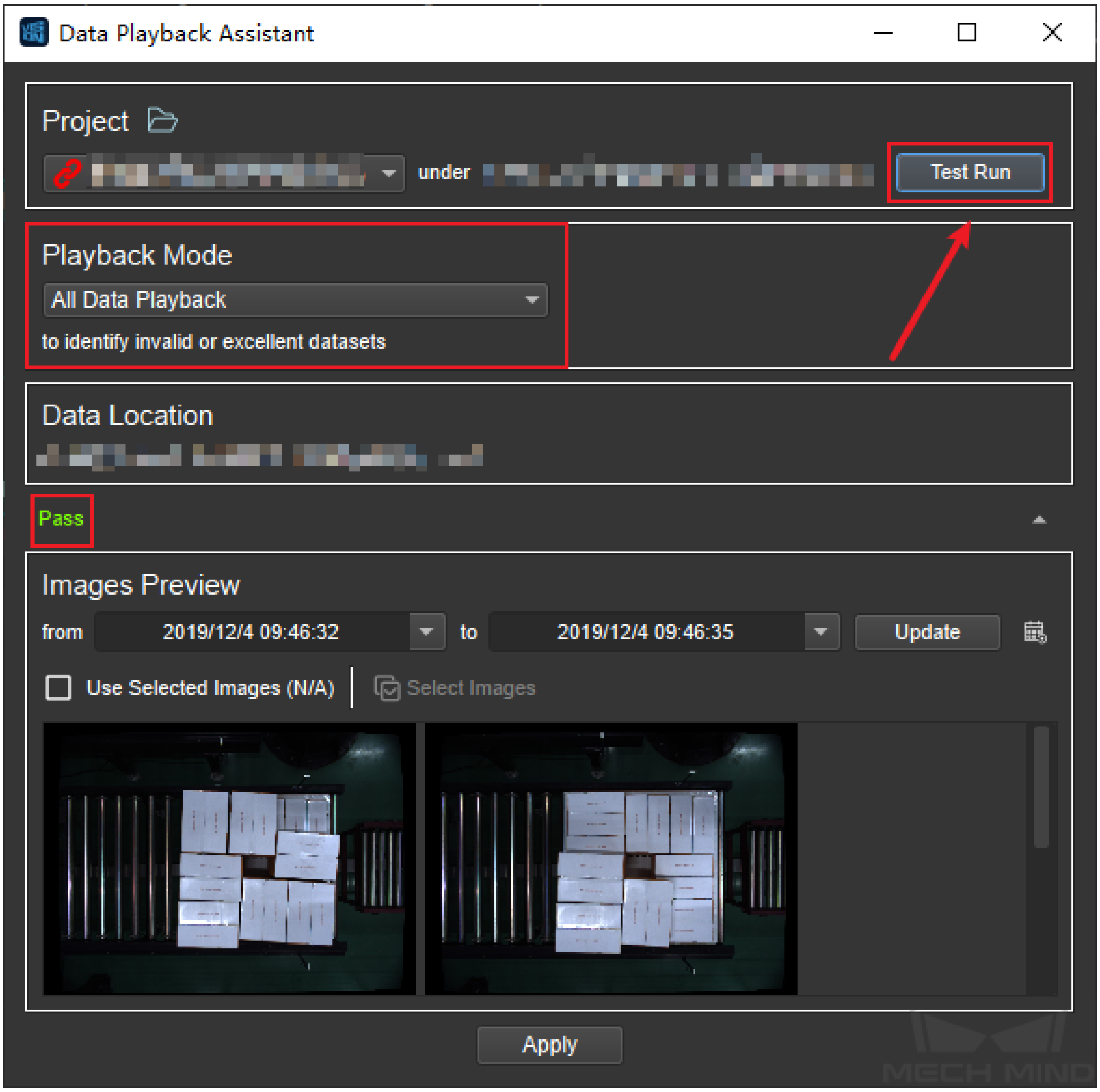
7.8.2 Operation Procedure

1. Selecting the project that needs to be set, and proceed to *Tool* → *Data Playback* .



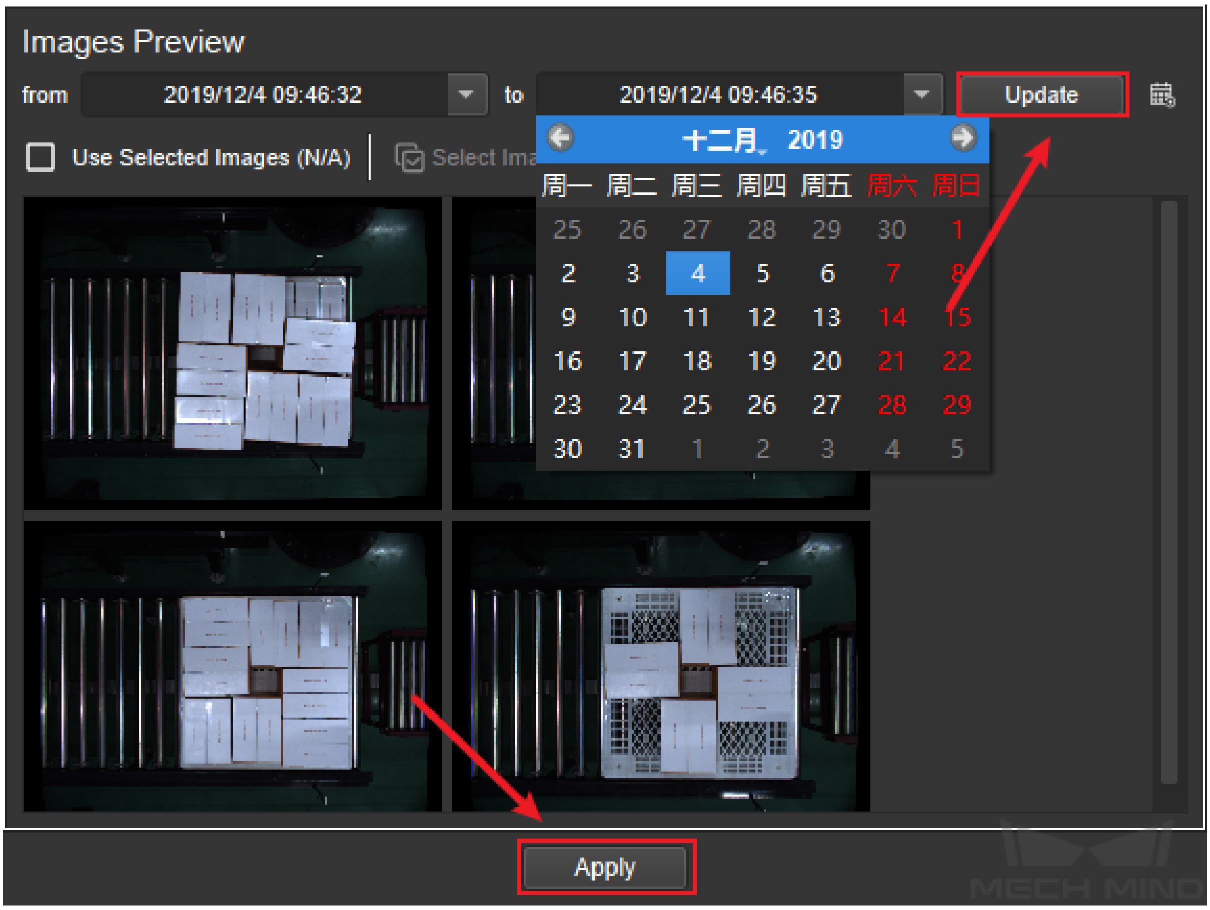
2. Data Playback Assistant window will pop up as shown in the figure.

- Check whether the project information and path are correct, and click *Test Run*. Pass appears if the project runs successfully.
- Choose one playback mode from the drop-down list:

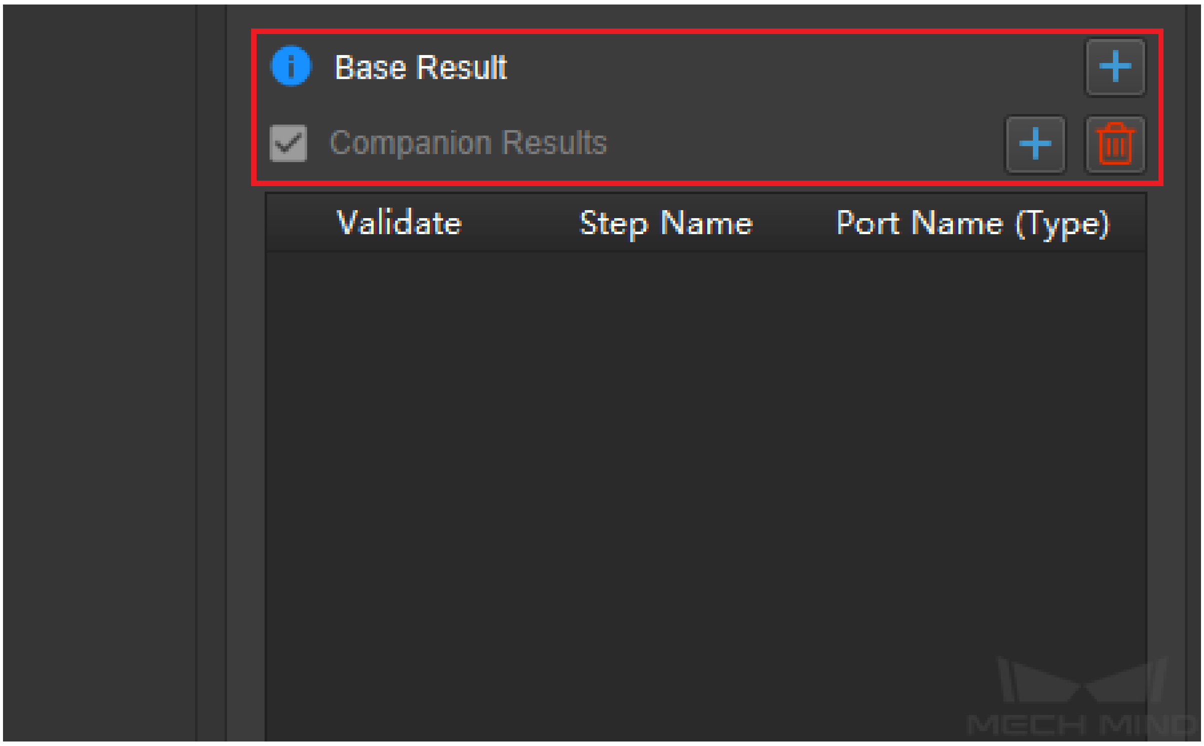

Data Playback Assistant
Project
under
Test Run
Playback Mode
All Data Playback
to identify invalid or excellent datasets
Data Location
Pass
Images Preview
from
2019/12/4 09:46:32
to
2019/12/4 09:46:35
Update
Use Selected Images (N/A)
Select Images
Apply

7.8.3 All Data Playback

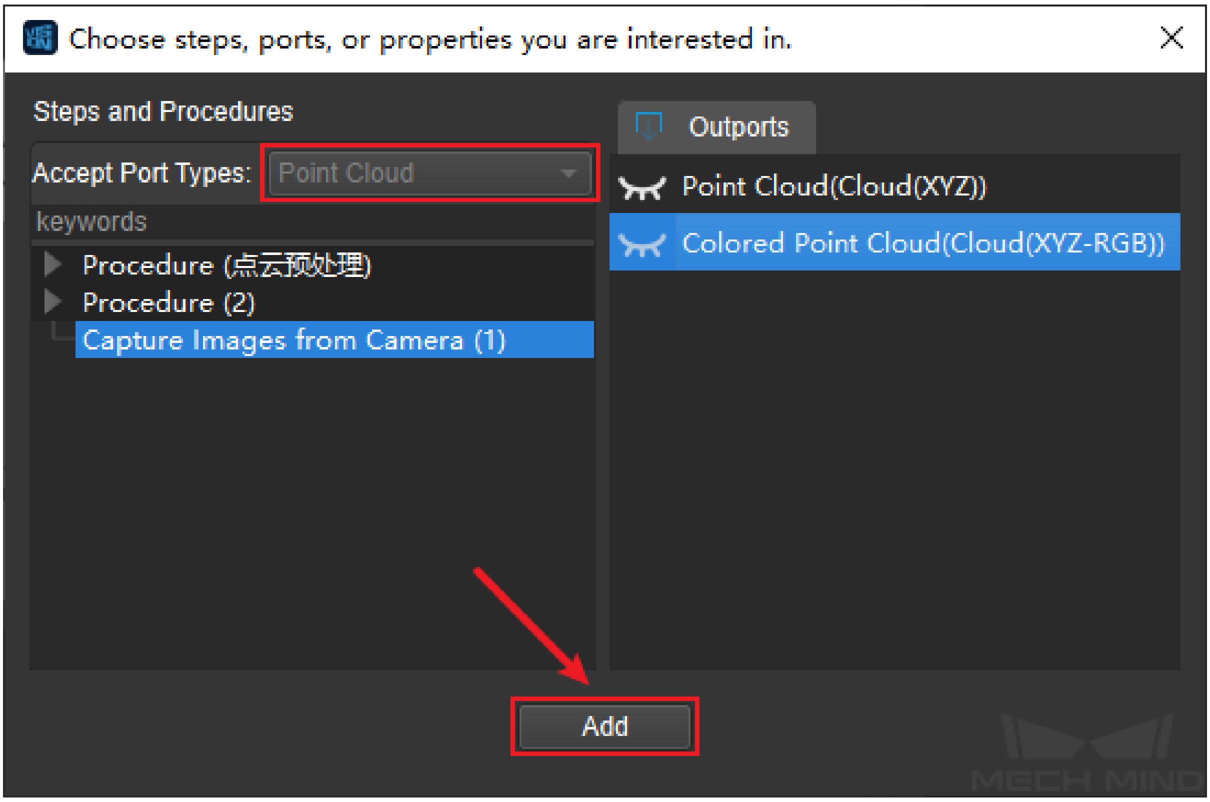
1. Set the time range in the **Image Preview** at the bottom of the interface shown below, and then click *update* to filter out undesired images. Click *apply* to proceed to the next step.



2. Enter the step selection interface, and click + to add **Base Results** and **Companion Results**.

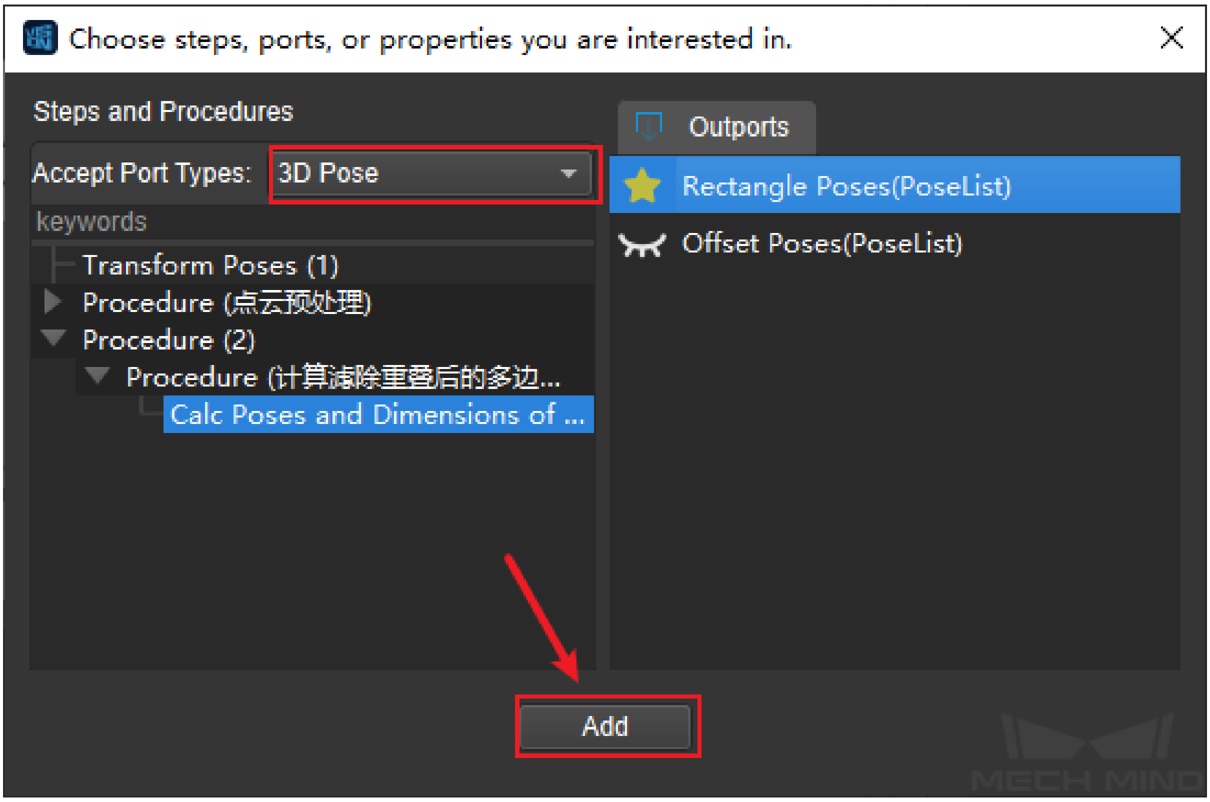


3. Select a **Base Result** that can output point cloud, and click *Add*.

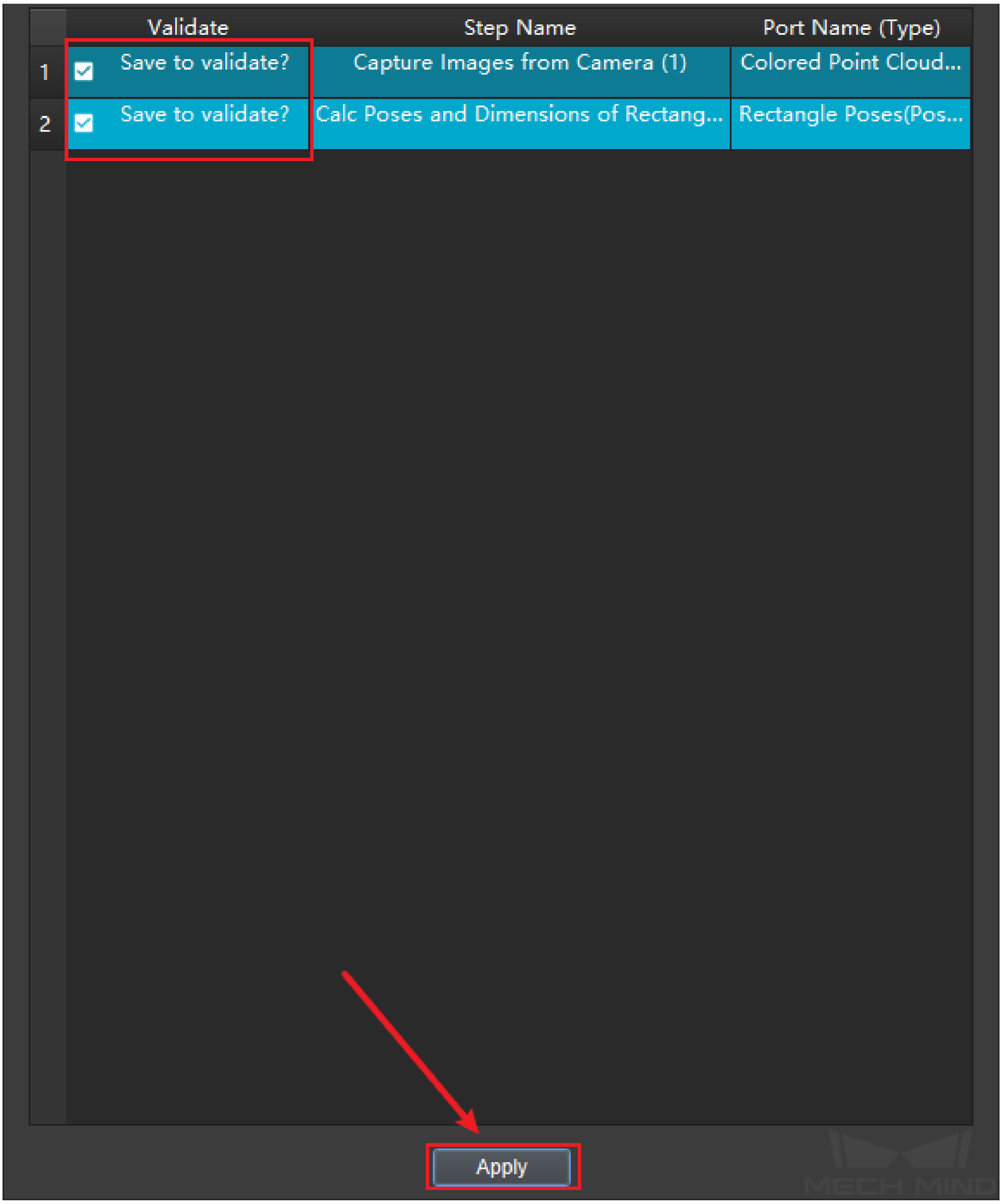


Note: Base Result is a reference for Companion Results. For example, after the background point cloud is selected, it can be compared with the point cloud and pose in Companion Results to judge whether the output is correct.

4. Select one or more **Companion Results** to output point cloud/3D pose, and click *Add*.

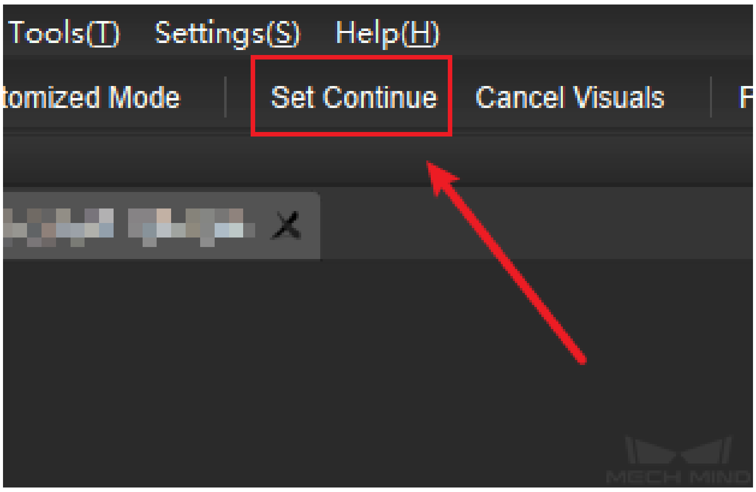


5. The selected options will appear in the list on the right side of the interface. After confirming that they are correct, click *Apply*. If the user want to verify the data after the playback is complete, choose **Save to validate** option. Then, the data will be saved and compared during Invalid Data Playback in the later stage (only available for 3D pose data).

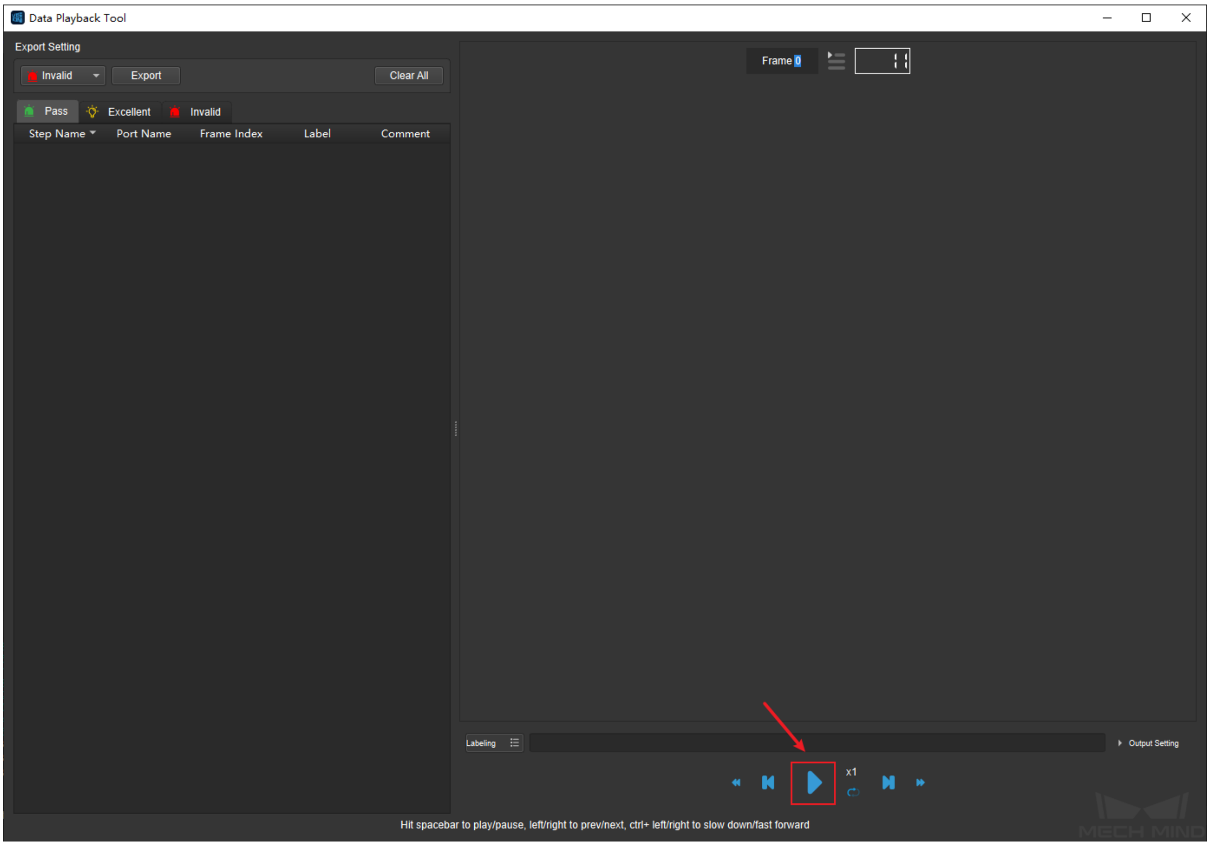


Note: The selected "Base Results" cannot be deleted, but can be replaced; "Companion Results" can be deleted by pressing the delete key.

6. Before operating data playback, click *Set Continue* on the top of the main interface.



7. Finally, enter the data playback interface, and click the play button to start.

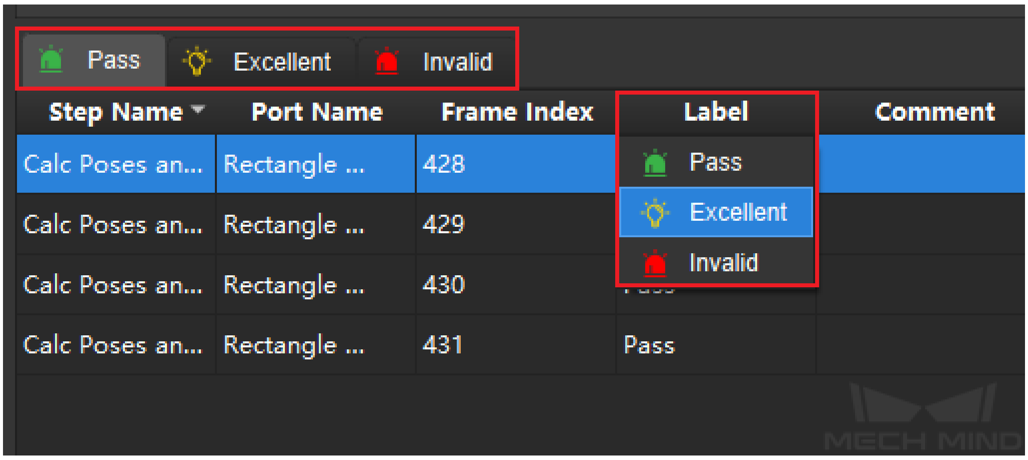


Note:

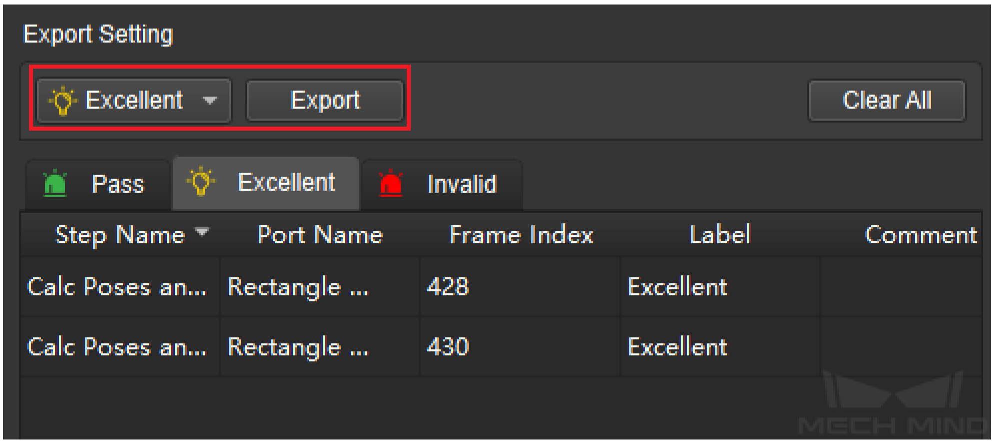
1. Play one frame and pause. Choose an angle where the problem can be found, then continue to play.
2. In order to view the problematic data, double-click the number under *Frame* to jump to the corresponding frame, or play adjacent frames to check.
3. If the project is not completed when clicking Pause, there will be a corresponding prompt.

---

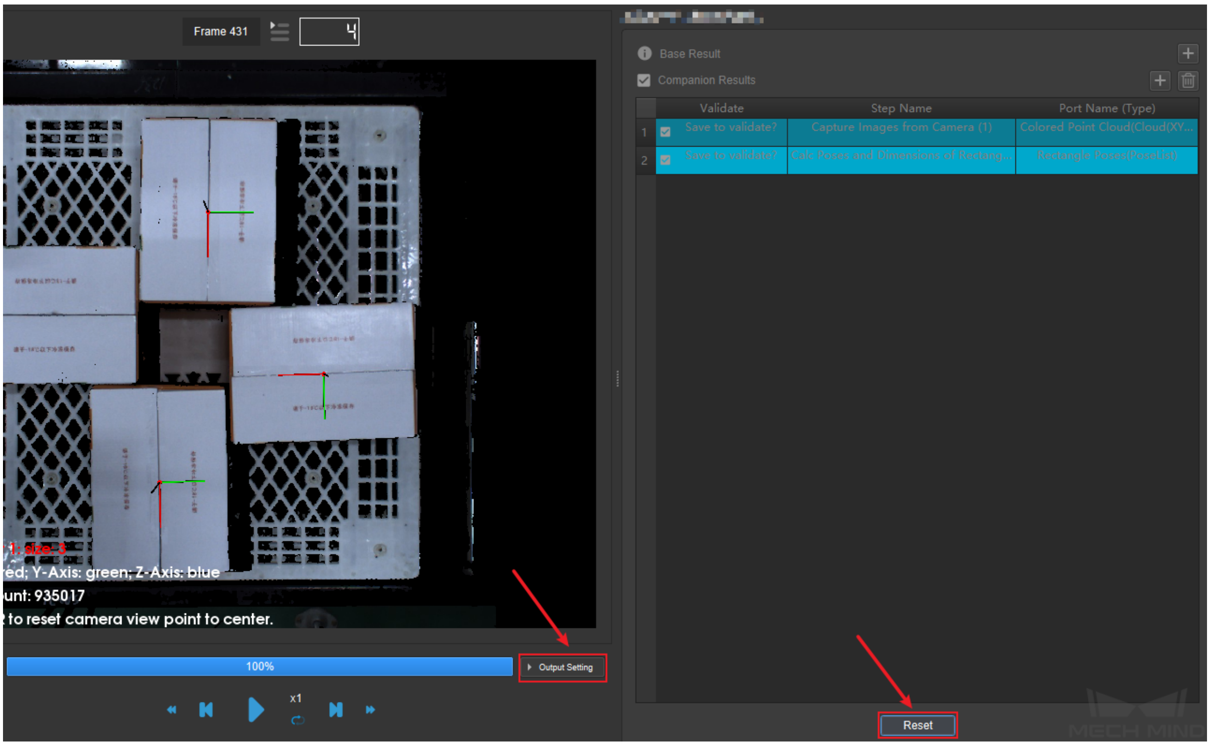
8. In order to classify the data, select **successful, excellent or invalid** under *Label*, and the classified data will be placed in the corresponding list. Also, notes can be added under *Comment*.



9. The filtered **Excellent** or **Invalid** data can be output by clicking the *Output* button.



Note: Results resetting can be achieved by clicking the *Output Setting* and *Reset* in the sidebar.



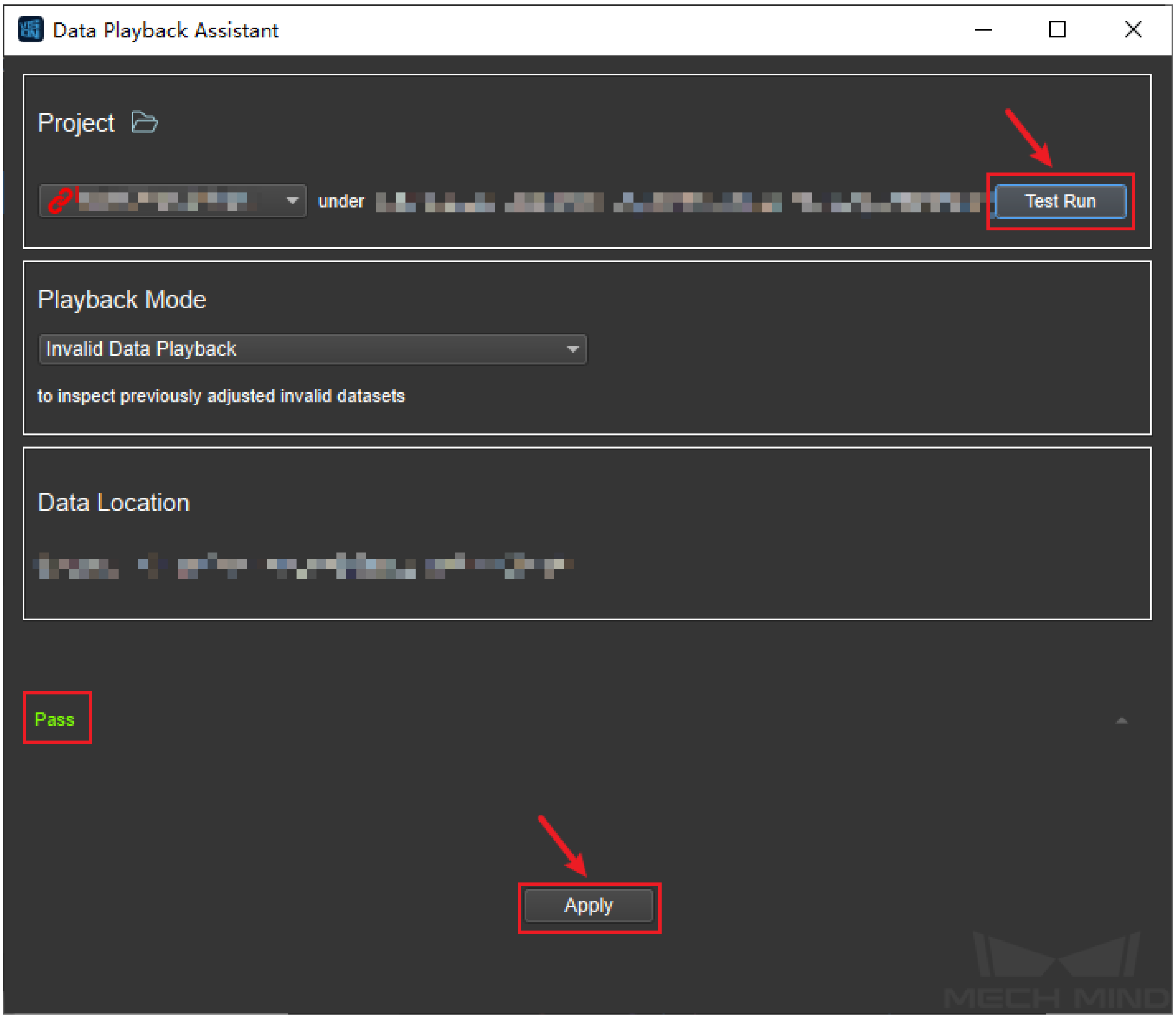
7.8.4 Invalid Data Playback

This mode is suitable for validating the data when the project has run *All Data Playback* mode and "Save to validate" has been chosen. The operation procedure is very similar to All Data Playback mode, but the project needs to be modified before data verification:

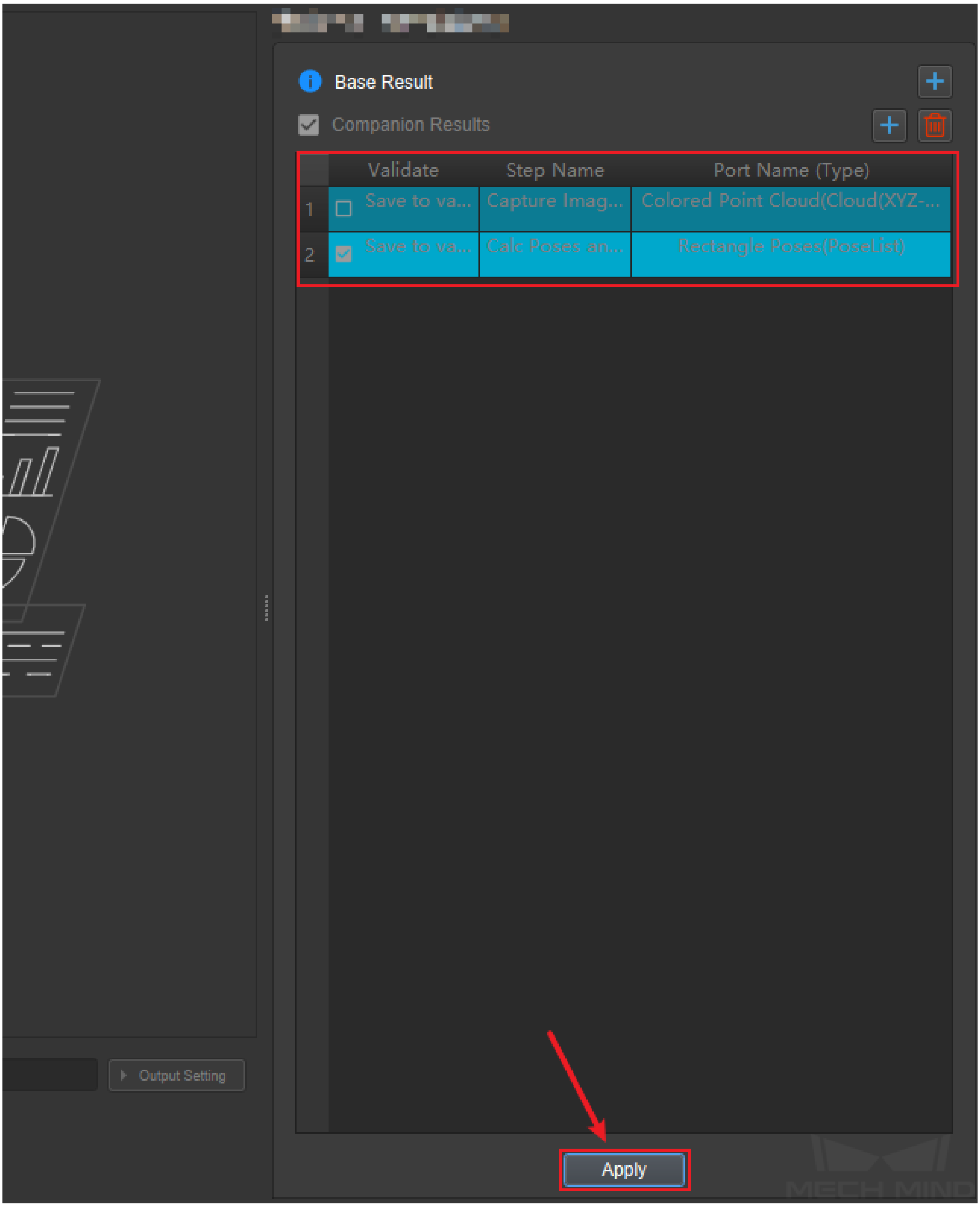
- Export invalid data in All Data Playback mode, and directly use this data to run the project. View and modify the invalid data, and use this mode again for verification after the modification is completed.
- Enter the playback interface of this tool, double-click the frame to view the invalid data, and modify the project directly based on the problem found (no need to close the tool). Then, use this mode for verification after the modification is completed.

Operating procedures

1. Click *Test Run* and see the green Pass, then click *Apply* to proceed to the next step.



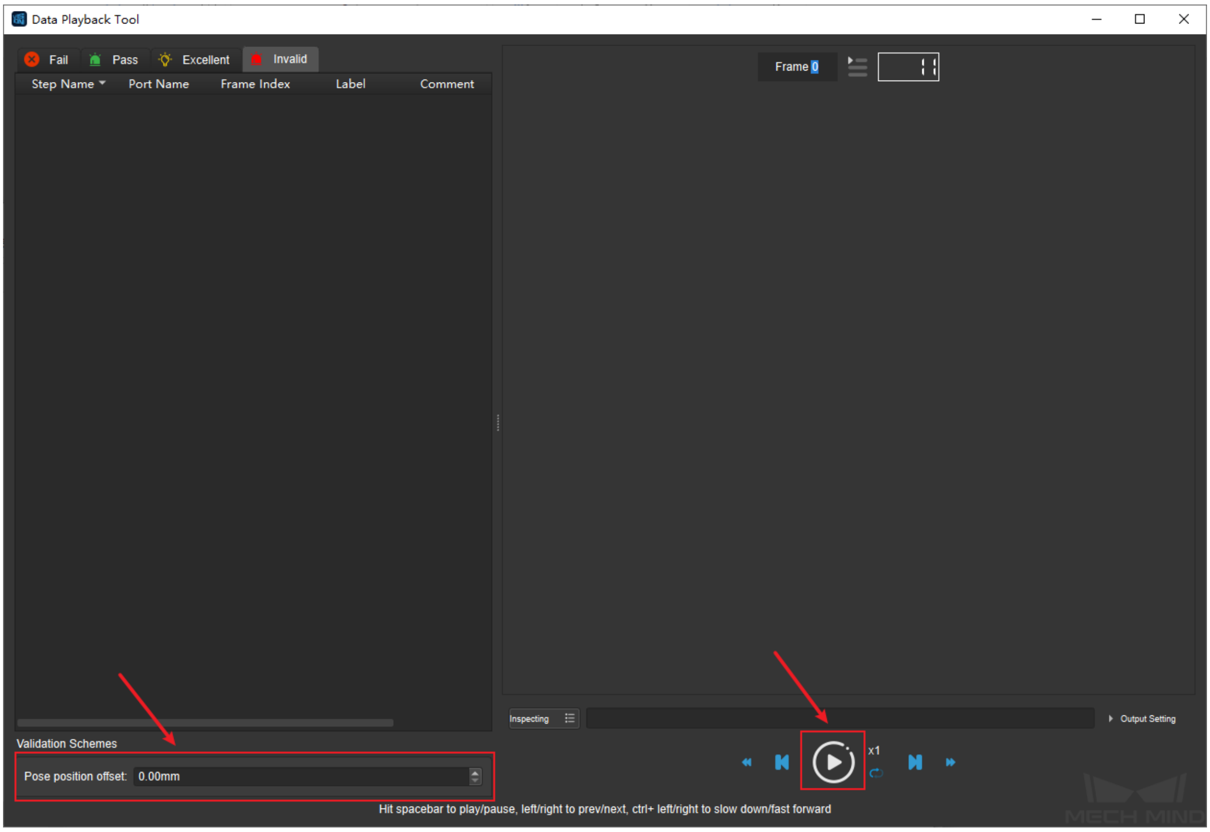
2. Enter the step selection interface. This project has run the data playback mode before, so the operations that have been performed are saved in the list on the right, and can be used directly.



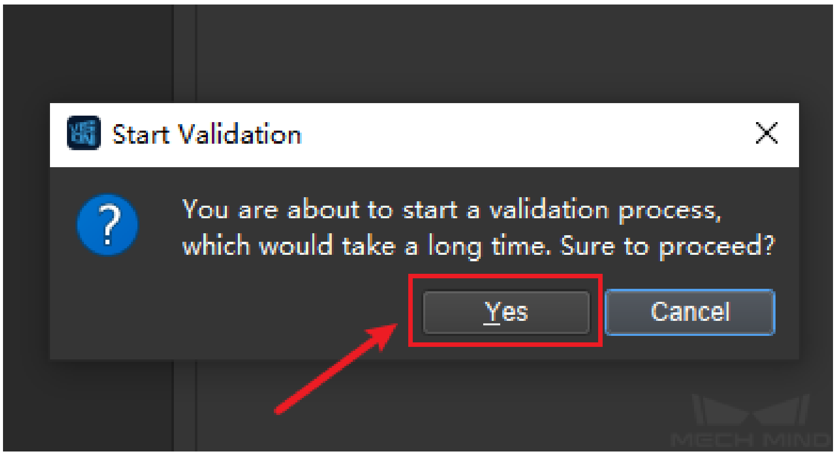
Note: "Save to validate" option cannot be changed here.

3. After confirming that all options are correct, click *Apply*. The data playback interface shown in the

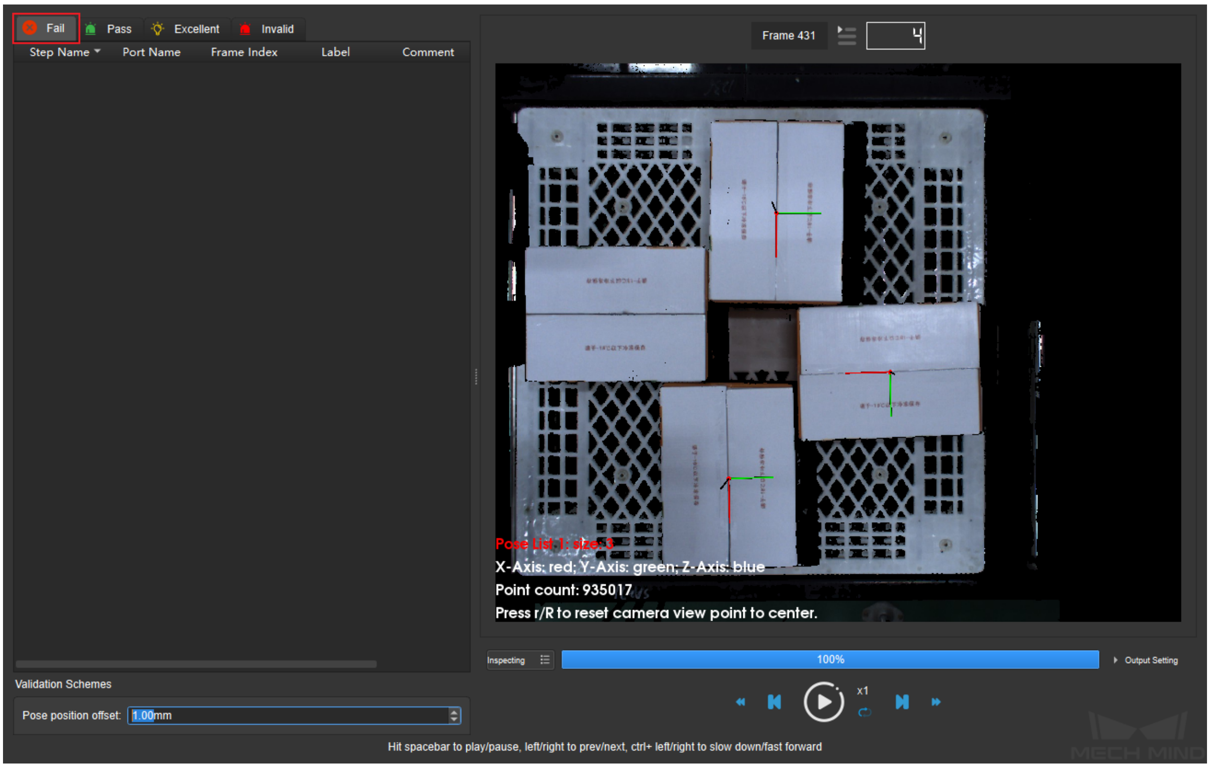
figure below will pop up, and set **Pose position offset** in the lower left corner. If the offset between new and odd data is larger than this value, the corresponding data will be automatically classified as "Fail". Then, click the play button to start the playback.



Note: If this dialog box pops up, select *Yes*.



4. If there is no "Fail" data after the playback has finished, it is proved that the project meets the conditions. Otherwise, the data still needs to be modified.



Note:

1. When Invalid Data Playback starts, it is not possible to pause and resume verification. If it is suspended, the verification needs to be performed again.
2. This mode doesn't allow to reset the base or companion results. If you want to adjust the settings after the first pass of verification, you need to manually switch them to re-verify.

7.9 Glue Wizard

7.9.1 Instruction to the Glue Wizard

Glue wizard is mainly aimed at glue trajectory points generating of frame object (such as car door). It prefers to apply in the scenarios such as: various types of objects, any position of incoming objects, complicated glue trajectorys and high precision requirement.

7.9.2 Use of Glue Tools (take an example as hatch glue)

Preparation

1> Paste white paper on workpiece and draw black lines on the white paper corresponding to the glue trajectory. Make sure black lines that are clear and no shadow, add proper light source if necessary

2> If the workpiece is thin, which causes the object to fit tightly to the table. It requires to elevated properly to the workpiece, which the elevated workpiece should still keep in a horizontal level.

3> Under the folder of saving workpieces (such as "D:/GlueWizard"), create subfolder of data and project, which are "D:/GlueWizard/data" and" D:/GlueWizard/project".

4> Open Mech-Vision Software, load projects "get-init-traj" and "load-2d-traj"

Collect information of workpiece

1> Open Mech-Vision -> Tools -> Plugin Tools -> Glue Wizard, the following figure illustrates information collecting interface and instruction of parameters.

2> Set the folder of saving workpiece information (such as "D:/GlueWizard"); click "New WorkPiece", add workpiece id and label, it will list out on the left bar after adding; select the proper type of camera.

3> Click "Init Trajectory" to collect data of workpiece and then the "get-init-traj" project will be run at background process. After that, it will automatically create a subfolder named by the workpiece ID number under data folder. This folder saves data information of current workpiece, which contains original RGB and depth images, RGB image with red track lines, the backup of project which is used to collect data of this workpiece, entire point cloud and contour point cloud of workpiece, weight point cloud in the inner contour of workpiece, initial 2D track points. It is shown as the following figure. After running this project, workpiece image can be checked, and also check whether the point cloud is complete as expected. Check parameter settings and projects if any problem occurs.

Mark and generate 2D trajectory points

1> After completion of collecting workpiece information, click 'Next' to the mark interface.

Default Param Path	you can set the parameters by loading historic files.
Open Adsorption	if checked, the mark point will be adhered to the nearest red line when marking.
Search Point Radius	
Corner Point	The radius of searching for corner point when opening adsorption.
Line Point	The radius of searching for line point when opening adsorption.
Sample Point	
Sample Distance	Every this sample distance will generate a sample point between two marking points.
Min Distance Ratio	The distance between the last sample point and marking point needs to larger than the length that "Sample Distance" * "Min Distance Ratio", which avoid those two points too close.
Rectify Line	
Gaussian Blur Size	The size of gaussian kernel function which is used to smooth the contour
Search Radius	The radius of searching contour around
Closest Point	the smallest distance of the closest point. It can filter out the sample point which just on the edge of workpiece, ensuring the accuracy of rectification.

2> set parameters according to the parameter instruction above, click 'Start Labeling', use left click of mouse to add the marking point, after marking a whole part, use right click to end this marking, and then re-click 'Start Labeling' for the next part marking。Marked lines can be shown on the 'Labeling Result' of right bar

Attention: During marking, please follow these principles: A. only mark the point in corner, between two marking points, it automatically generates several sampling point based on "sample distance"; B.There should be more points at arc surface, and make sure to keep distance in each point (such as 3-4cm), to avoid the glue gun collides with the workpiece at the arc surface during the actual use of glue. C. At steep place, it must have marking points

3> As for the part that the whole part is straight line, if trajectory didn't appear in the middle of two side contour as expected, it can be rectified (right click -> Rectify Polyline), shown as follows

Attention: This function works out only if the whole part is straight line segment

4> If mark failed or excessive deviation on marking, you can click 'Clear Canvas' and remark it.

Generate 3d Trajectory

1> Click ‘Save Result’ at marking interface, and then click ‘Next’ to the interface of generate 3d trajectory, click ‘generate 3d trajectory’, the “load-2d-traj” project will be run in the background process to generate initial 3D trajectory.

2> Edit and Save 3D trajectory and its parameters. After ‘generating 3D trajectory’ above, it will switch to the edit interface of initial 3D trajectory, which can modify trajectory and set parameters based on the manufacturing needs. After modification, click ‘Save Traj’ to save trajectory and its parameters. Finally click Finish to complete the trajectory generation of this workpiece. Edits and parameters instructions are shown below.

2.1> Global Parameter Setting(Global Setting Tab)

Traj Offset	Offset the whole trajectory (apply to the situation that there are error with the extrinsic parameter of camera)
Angle to rotate	The chosen trajectory will spin on Y axis according to this angle
Show Ground Z	As for huge workpiece, you can uncheck it to observe the trajectory clearer.
Ground Z	The grid position in Z axis
Traj Point Size	The size of trajectory showing on the interface
dragger Size	The dragger size of each trajectory point
Motion Type	Move type of robot: MoveL or MoveJ
Acc	Acceleration of move
Vel	Velocity of move
Blend Radius	Blend radius when moving arc
apply All	if clicked, all these parameters apply to all trajectory (not recommend)

2.2> Speicial Parameters Setting

Glue On/Off Blend Radius	
Glue On	The blend radius before starting glue
Glue Off	The blend radius after finish glue
Vel Setting	
Traj Point	Velocity of trajectory point (except the transition points between different segments)
Mid Point	velocity of the transition points between different segments.
Generate Points From Start	
Interval	The distance between advanced point and start gluing point
Num	Generate multiple advanced points forward every "interval".
Raise Z	To raise the advanced point along Z axis. It avoid the glue gun colliding with workpiece or touching the glue in other segments.
Generate Points After End	
Interval	The distance between extend point and finish gluing point.
Forward Num	Generate multiple extend points backward every "interval".
Blend Radius	The blend radius of extend points
Generate Pull Up Points	
Interval	The distance between extend point and pull up point, and the distance between neighboring pull up points.
Pull Up Num	Generate multiple pull up points every "interval".
Pull Up Angle	The angle between the line consists of gluing points and the line consists of pull up points. It decide the inclination of pulling up.
Generate Middle Points	
Interval	The distance between neighboring middle points.
OffsetZ	To raise the middle point along Z axis (based on the highest height of neighboring point cloud). It avoid the glue gun colliding with workpiece.
Adjust Blend Radius	Search the trajectory points in corner(in the range of setting curve), and then set the blend radius of this point to 5mm, which avoid robot reaching the point in corner not exactly due to the improper blend radius.
Min Curve	if a trajectory point"s curve is between the range of "Min Curve" and "Max Curve", it is considered the point in corner.
Max Curve	if a trajectory point"s curve is between the range of "Min Curve" and "Max Curve", it is considered the point in corner.

2.3> 3D trajectory points list

7.10 Pose Editor

7.10.1 The Introduction of Pose Editor

Th main function of the pose editor is to improve the ease of use of the pose adjustment, provide the visualaztion and reduce the learning cost. The pose editor has the following two main modes :

1. Preset Edit Mode: use the preset functions, coordinate system transformation, adjustment and sorting to quickly edit pose.
2. Customized Edit Mode: add the corresponding step to edit the pose, user can check the edited pose in the scene view.

7.10.2 The Introduction of Pose Editor Main Interface

As shown in *figure 1*, The pose editor interface can divided into four areas:

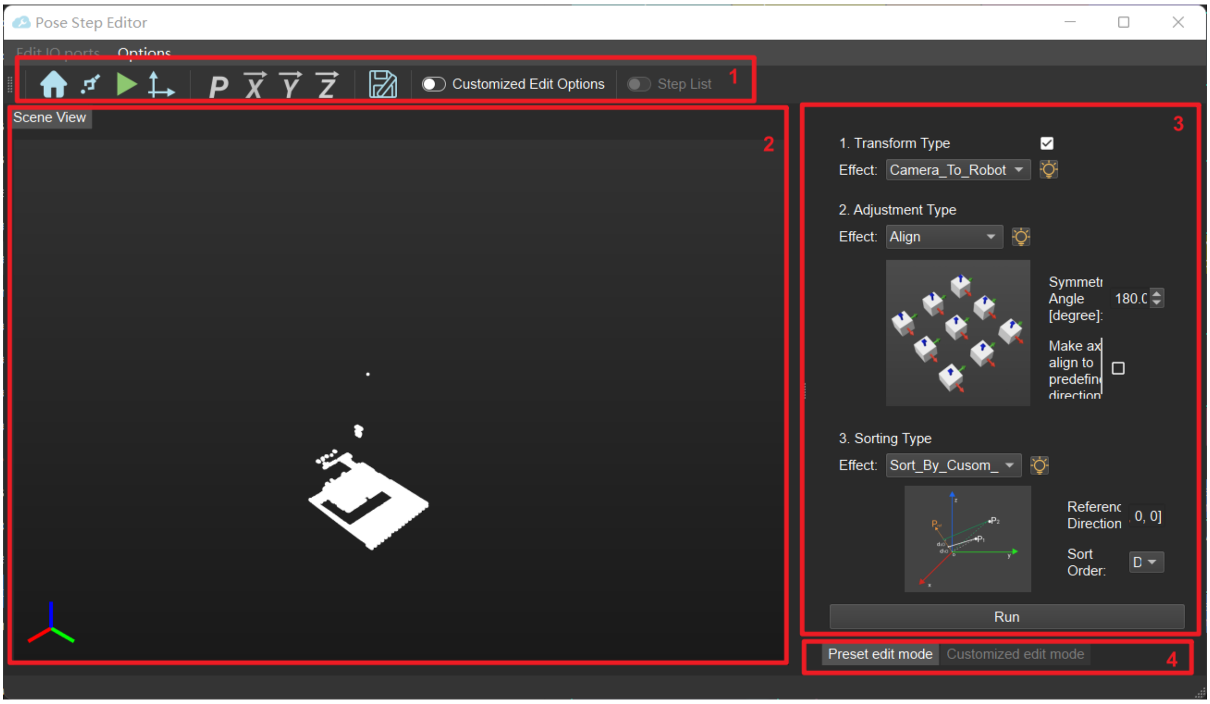


figure 1 Interface of Pose Editor

1. Toolbar
2. Scene View (visualaztion area)
3. Editing Area
4. Edit Mode Selection

Toolbar

The main function of is to adjust the pose more easily by using these tools:

1.  Back to input poses.
2.  Edit a target position or orientation of a pose in the scene.
3.  Run through all existed steps.
4.  Show world coordinate in scene.
5.  Copy the current position.
6.  Copy X-axis vector.
7.  Copy Y-axis vector.
8.  Copy Z-axis vector.
9.  Save configs into a JSON file.
10. 

Show step library（only used in Customized Edit Mode）。

Scene View

The main function is to review the poses after editing and to facilitate the subsequent further adjustments.

Editing Area

The main function is to edit the pose quickly and conveniently. The editing area will be different according to the selected editing mode, as shown in *figure 2*and *figure 3*.

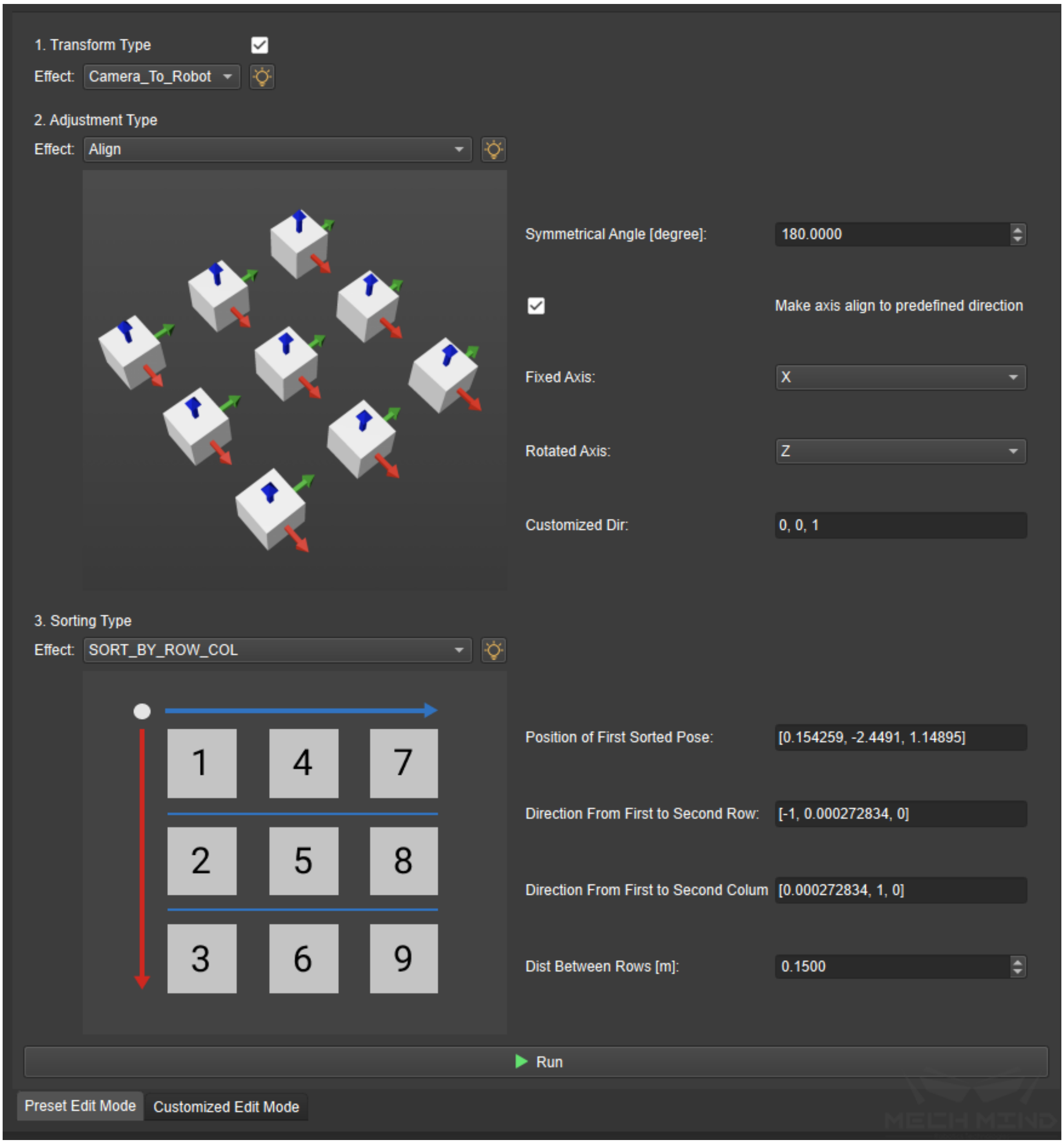


figure 2 Interface of Preset Edit Mode

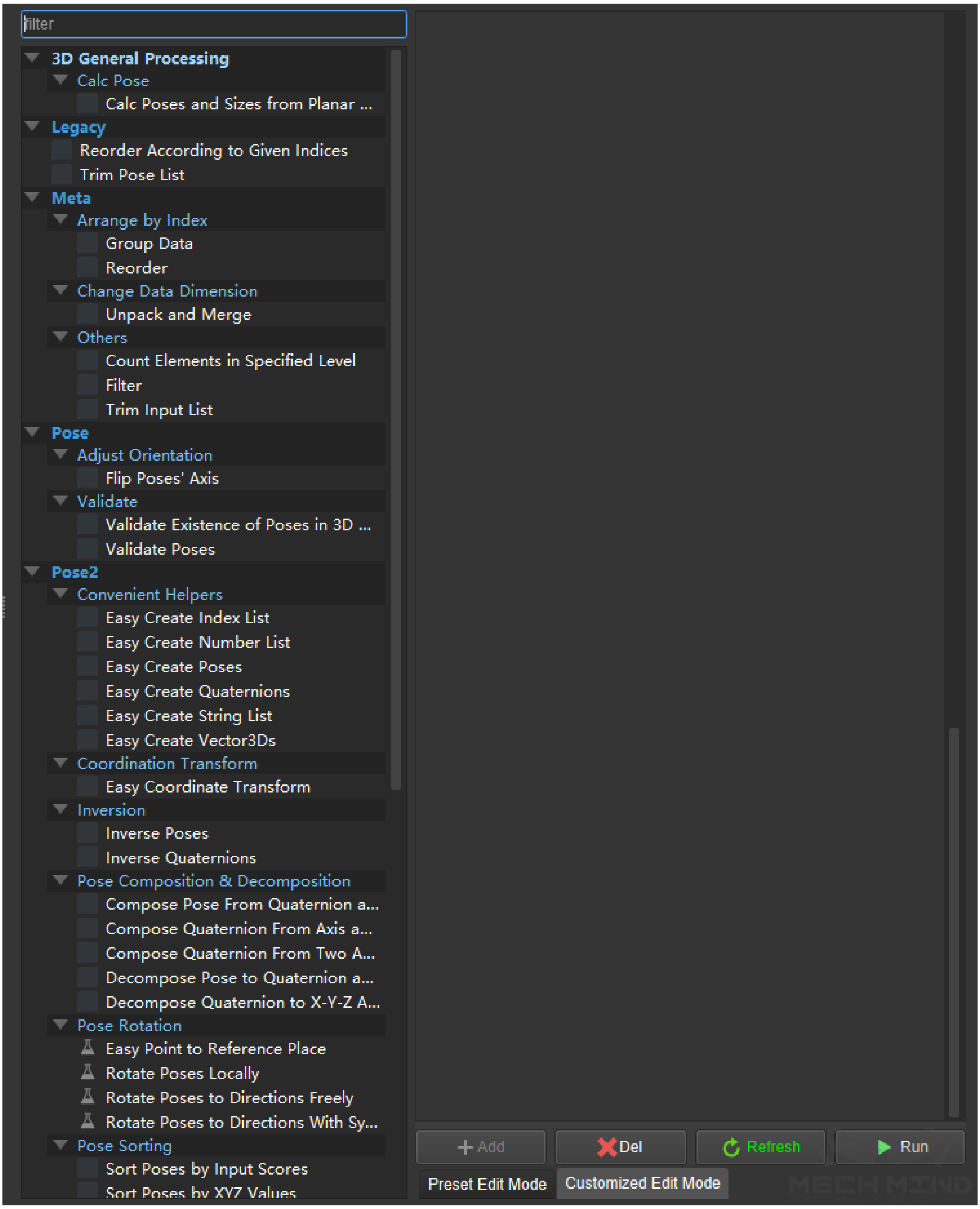


figure 3 Interface of Customized Edit Mode

Edit Mode Selection

Seclect different edit modes to adjuste the pose according to actual needs. It is recommended to mix the two modes to adjust the pose, it will be more conveniently and faster.

7.10.3 The Introduction of Pose Editor Mode

Preset Edit Mode

In the preset edit mode, there are three types of preset editing functions:

Transform Type: Transform the coordinate system of the pose by selecting *camera to robot* or *robot to camera* , as shown in *figure 4*.

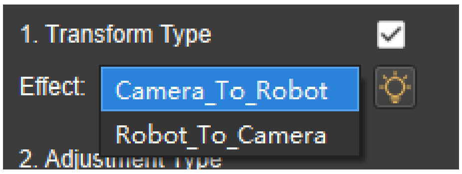


figure 4 Coordinate transformation

Adjustment Type: There are three adjustment types: *Align*, *Outwards* and *Point to* .

Sorting Type: There are five sorting types to choose: *Sort_By_Cusom_Direction*, *Sort_By_Dist_to_Point*, *Sort_By_Dist_to_Point_On_XY_Plane*, *Sort_By_Position_XYZ*, *Sort_By_Row_Col* .

Customized Edit Mode

When using the customized edit mode, you first need to click *Show step library*, after the list is expanded, select the corresponding step and click *Add* , as shown in *figure 5*.

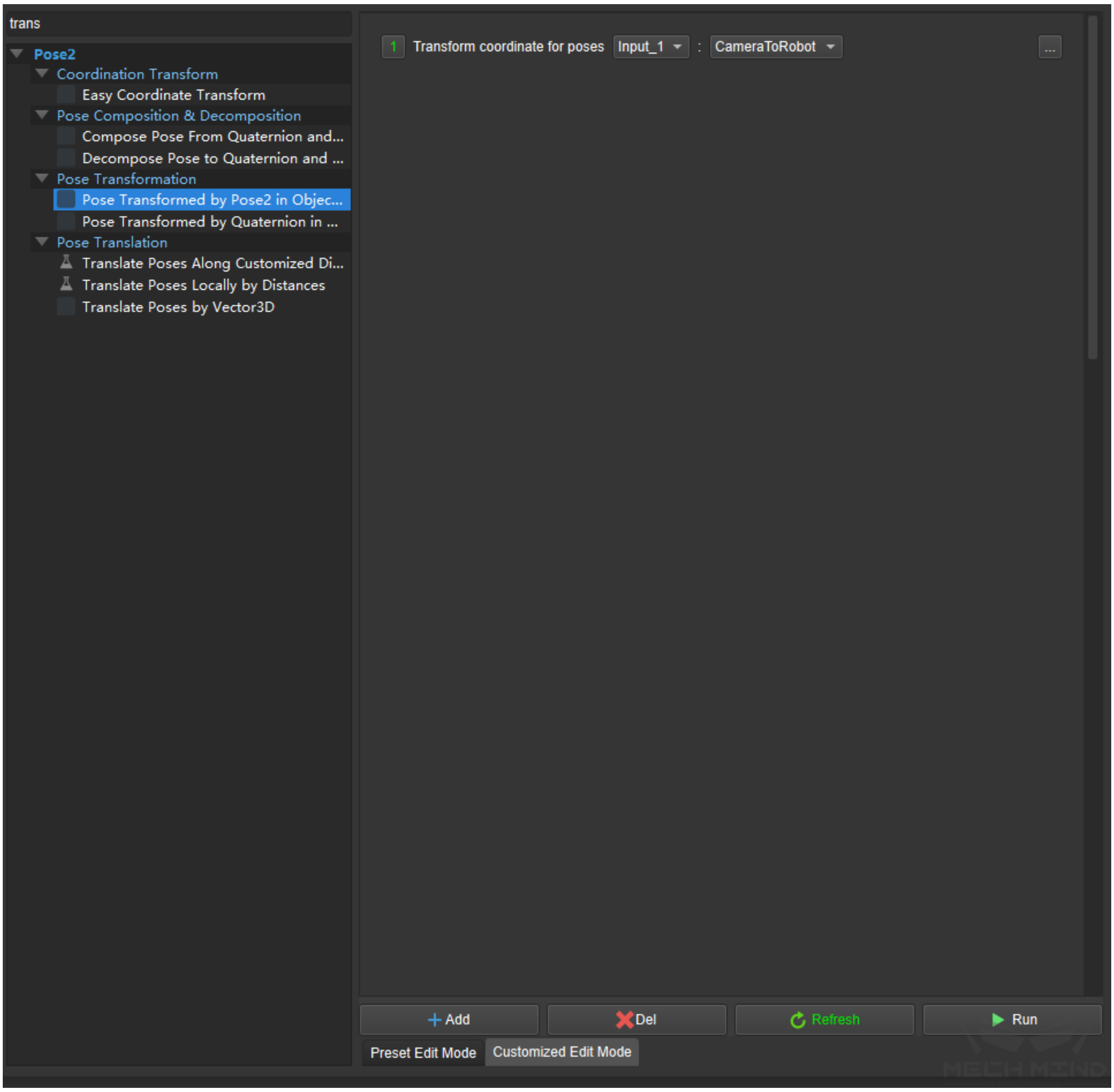


figure 5 Add step in Customized Edit Mode

7.10.4 How to use Pose Editor

Let’ s take a project as an example to introduce the process of using the pose step editor:

Step 1: Firstly, at the place where the pose needs to be edited, add the step “Pose Adjustment Collection” , as shown in *figure 6*.

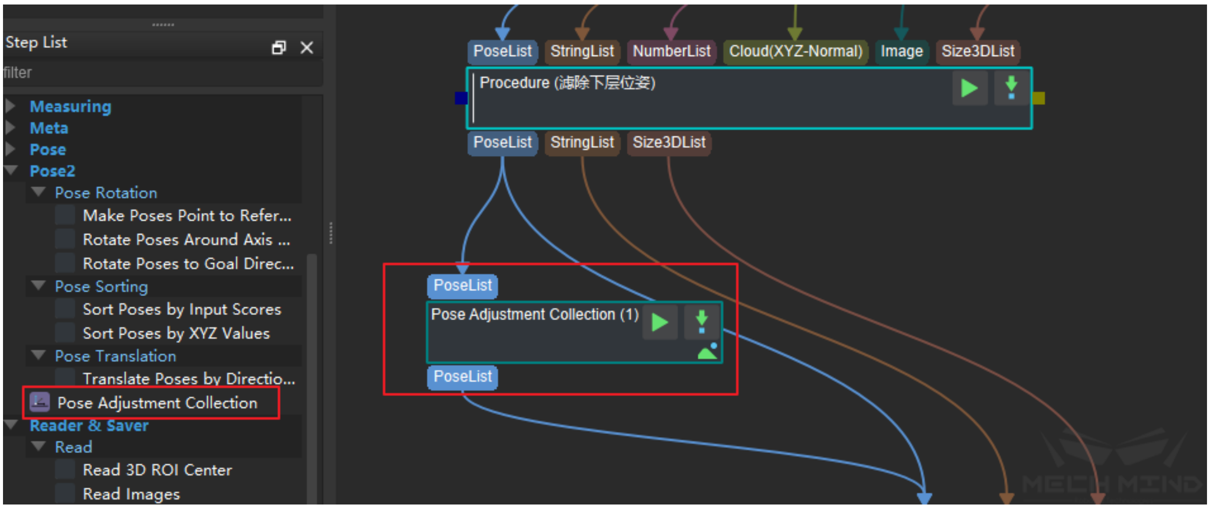


figure 6 Add step “Pose Adjustment Collection”

Step 2. Run the project to get the input pose, and then click the  button in the Operations input box in the property bar to enter the pose editor, as shown in *figure 7*.

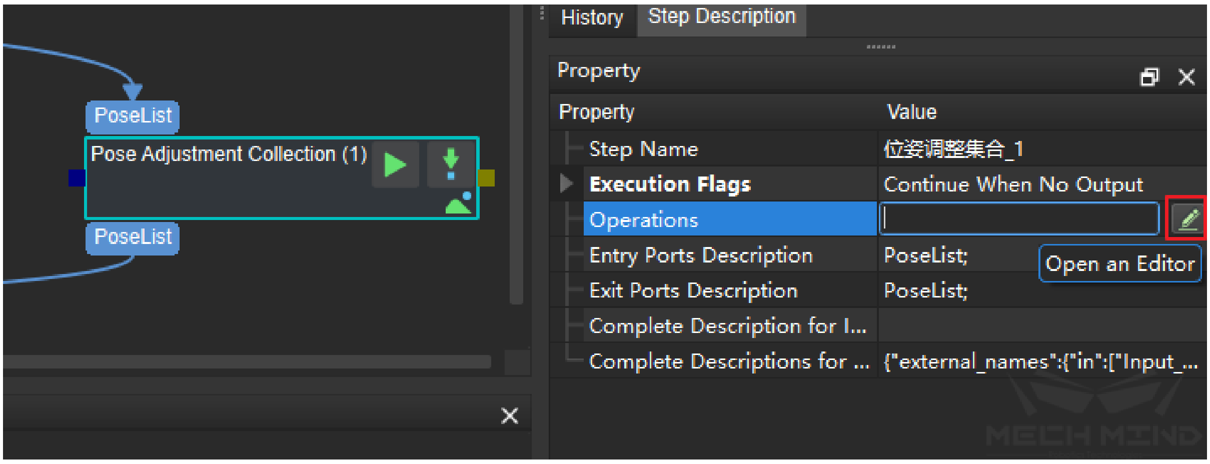


figure 7 Enter the Pose Editor

Step 3. By using the preset editing mode, select the preset function that needs to be used: 1. Transform the pose to the robot coordinate system, 2. Align the pose direction, 3. Sort the pose by row and column, 4. Run the programme to preview the result. The result is shown in *figure 8*.

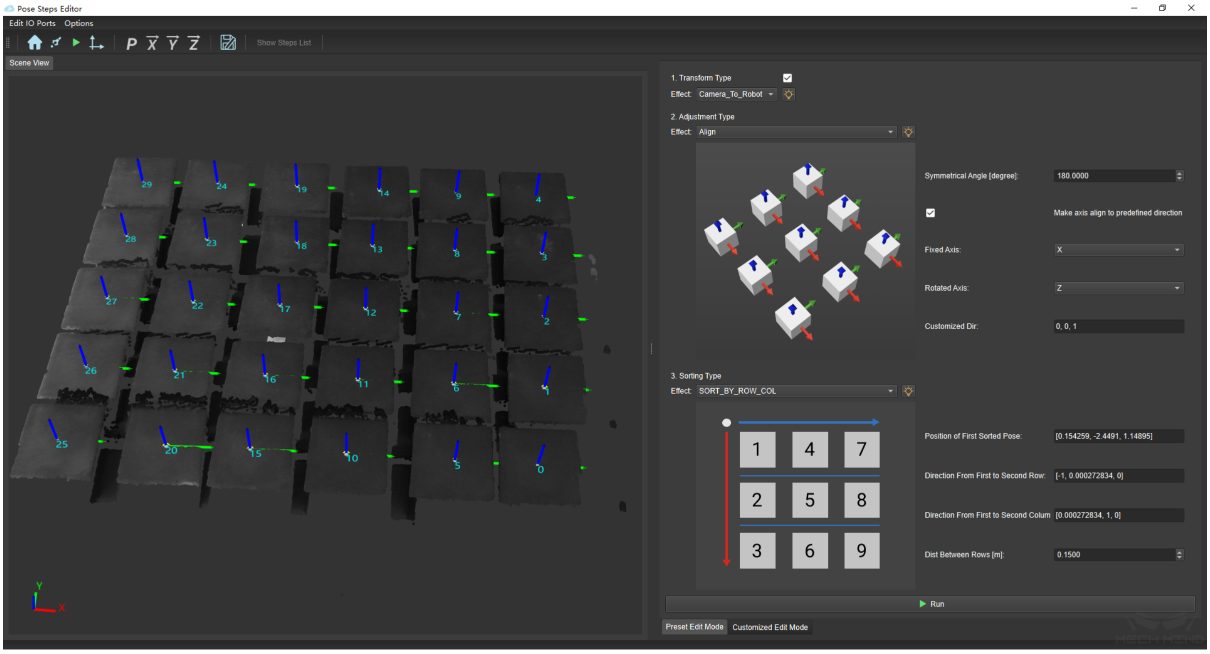


figure 8 Edit pose in Preset Edit Mode

Step 4. By using the production mode (custom), add the output quantity limit: 1. Click and get into the custom editing mode, 2. Display the step library, 3. Add the corresponding step, 4. Select the number of displayed outputs, 5. Fill in the limit number , 6. Run the programme to preview the results. The result is shown in *figure 9*.

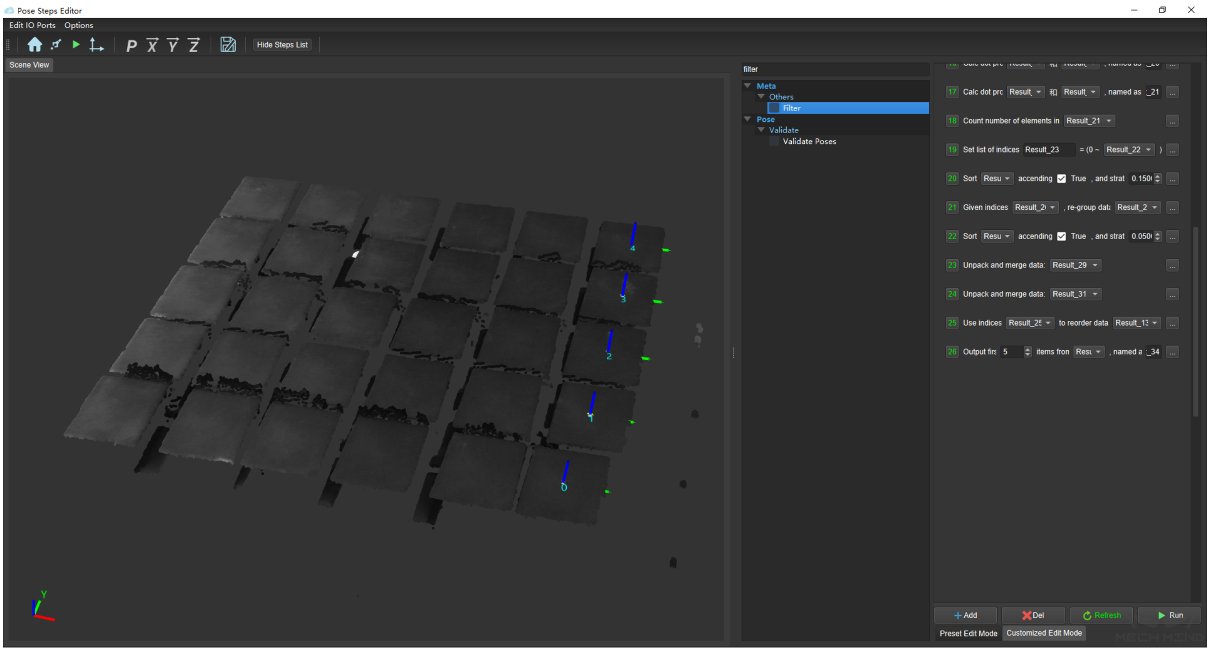


figure 9 Edit pose in Customized Edit Mode

Step 5. After finishing the pose editing, click the exit button, as shown in *figure 10*, to save the edited pose.

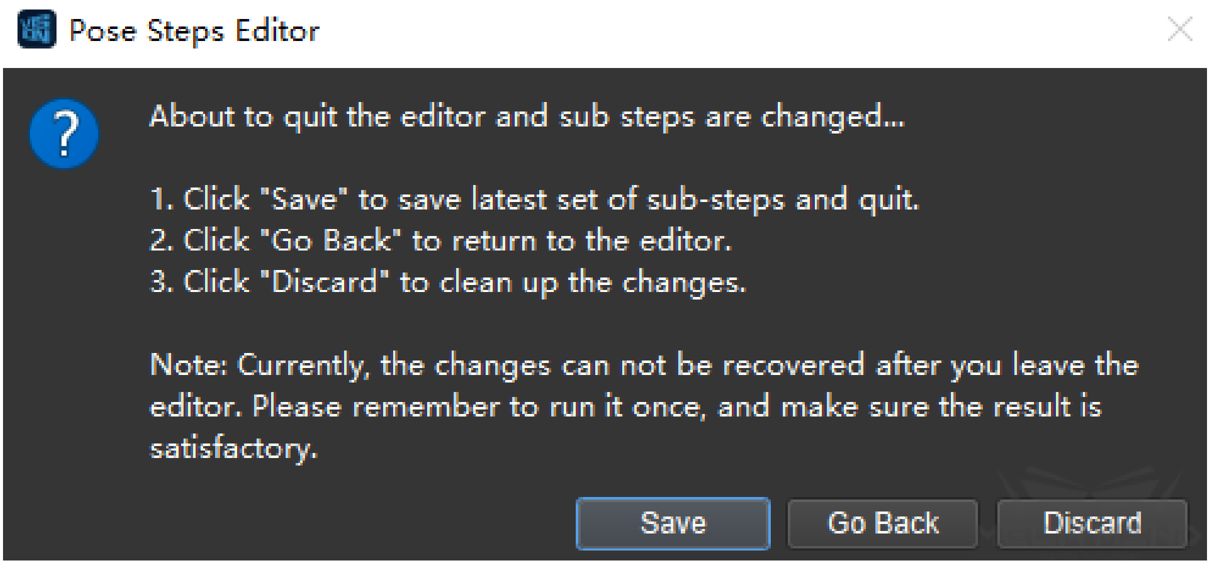


figure 10 Exit and Svae

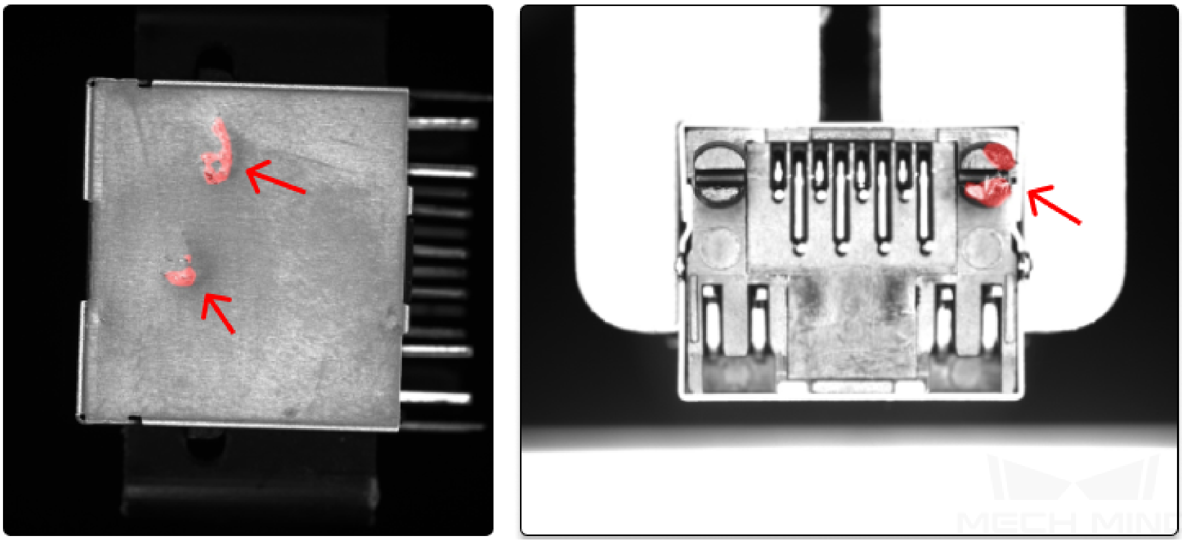
Step 6. Run the project to quickly review the results.

MEASUREMENT MODE GUIDE

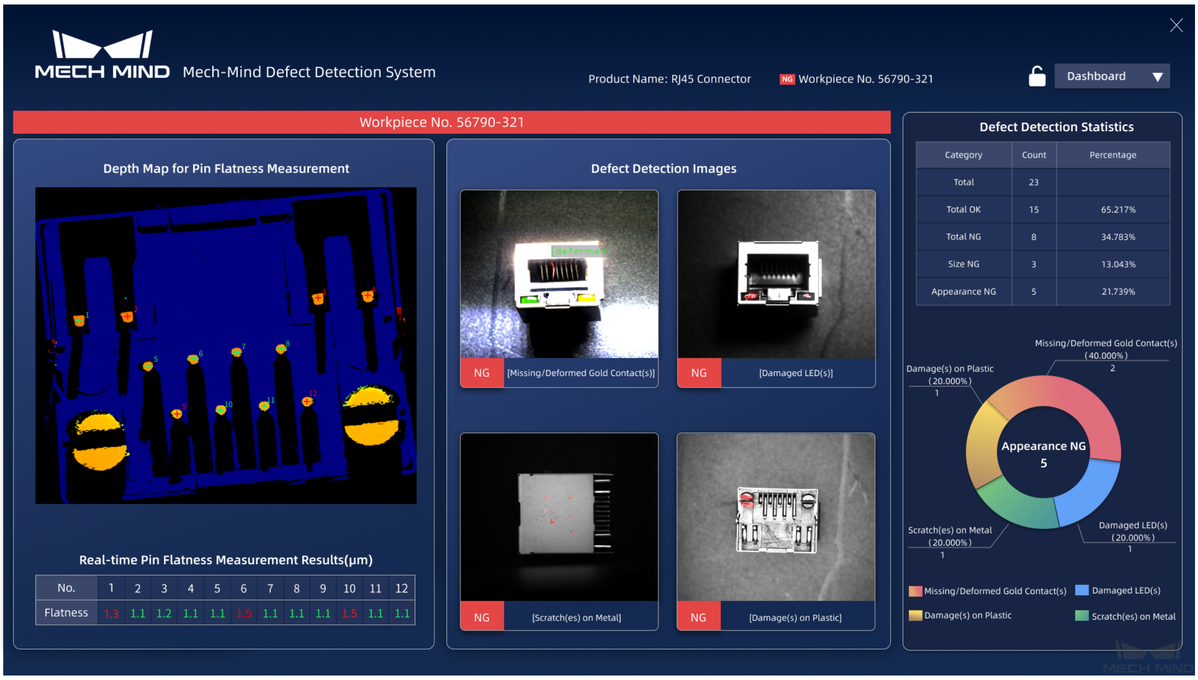
Quick Facts about Measurement Mode

Measurememt, detection, alignment, and recognition are common applications in industrial industry. Therefore, we provide 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



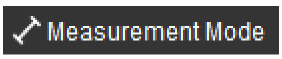
Custom operator interface

Check the section below to **getting started with measurement mode** and complete your first application.

8.1 Getting Started with Measurememt 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.

8.1.1 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 on 

in the toolbar to switch to measurement mode.

8.1.2 Input

The source data can be input via the following Steps:

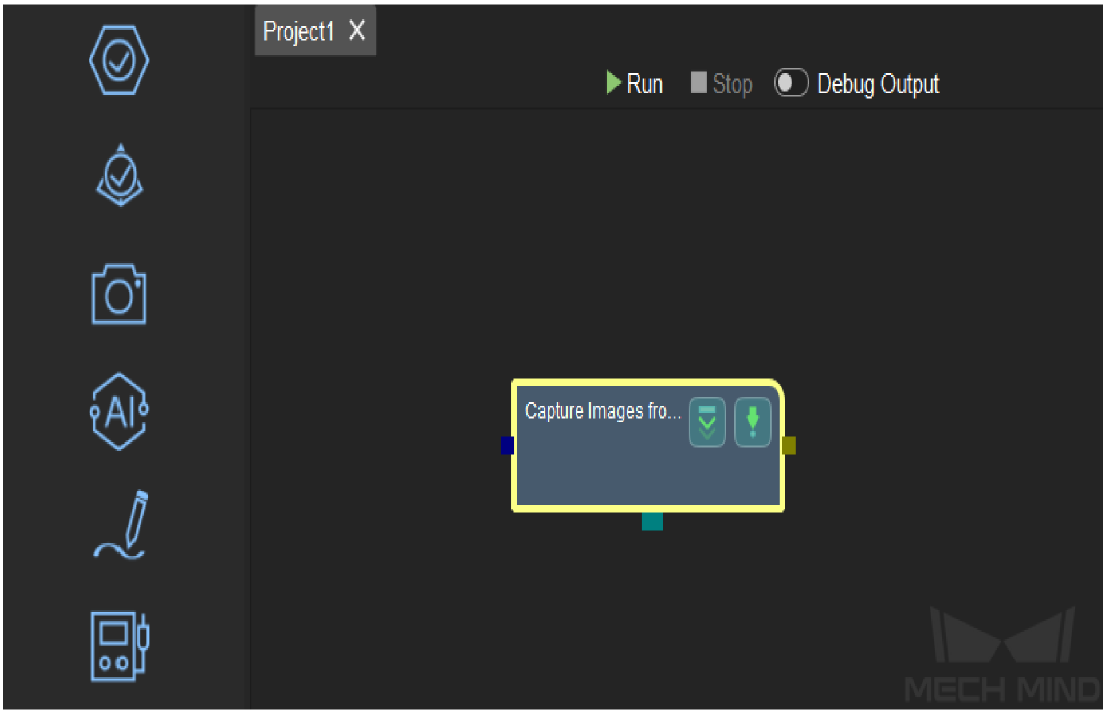
1. *Capture Images from Camera*
2. *Read Point Cloud V2*
3. *Read Images V2*
4. Laser Profiler

Note:

- 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 applictaion below.

---

1. Locate *Capture Images from Camera* in **Step Library** and drag it into the graphical programming workspace.



> **Hint:** You can click on the > 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.

8.1.3 Processing

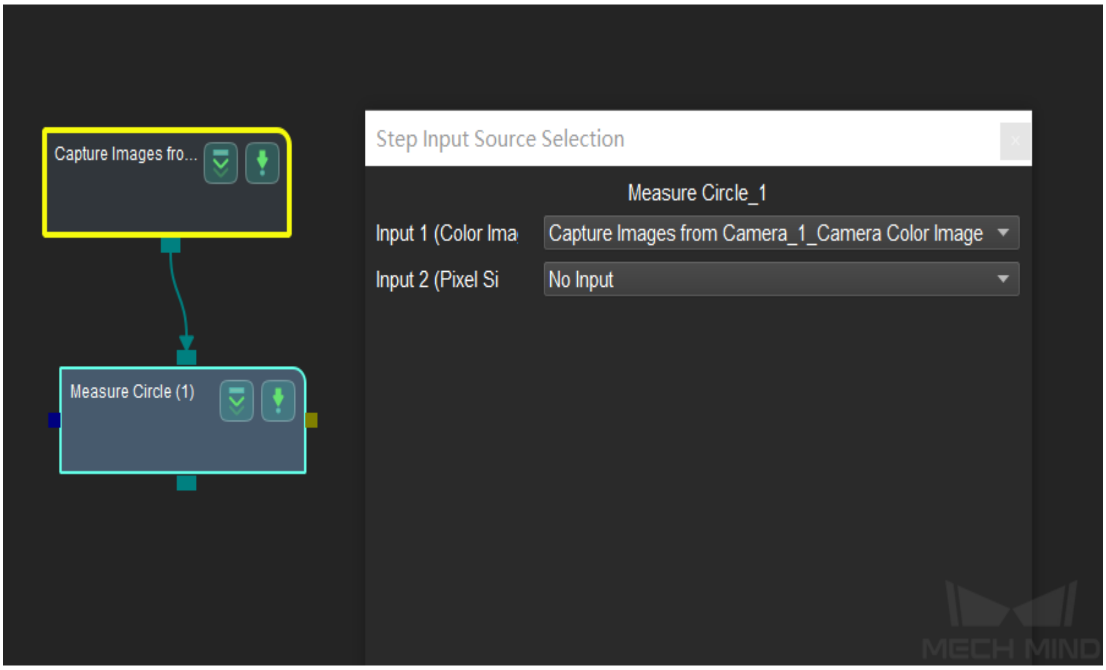
This example aims to check if the circular part of the object conforms to the specification or not, so the Step *Measure Circle* is used.

1. Locate the Step

   Go to *Measuring→ 2D Measuring* in the Step Library, and drag the Step *Measure Circle* into the graphical programming workspace.

2. Connect Steps

   Select *Measure Circle*, and then select **Capture Images from Camera_1_Camera Color Image** as **Input 1 (Color Image)** in the **Step Input Source Selection** panel.



Hint:

- The selected data type should conform to the data type within brackets, or else an error message will pop up.
- If you cannot see the **Step Input Source Selection** panel, please go to *View* and check the box before **Step Input Source Selection**.

3. Run the project

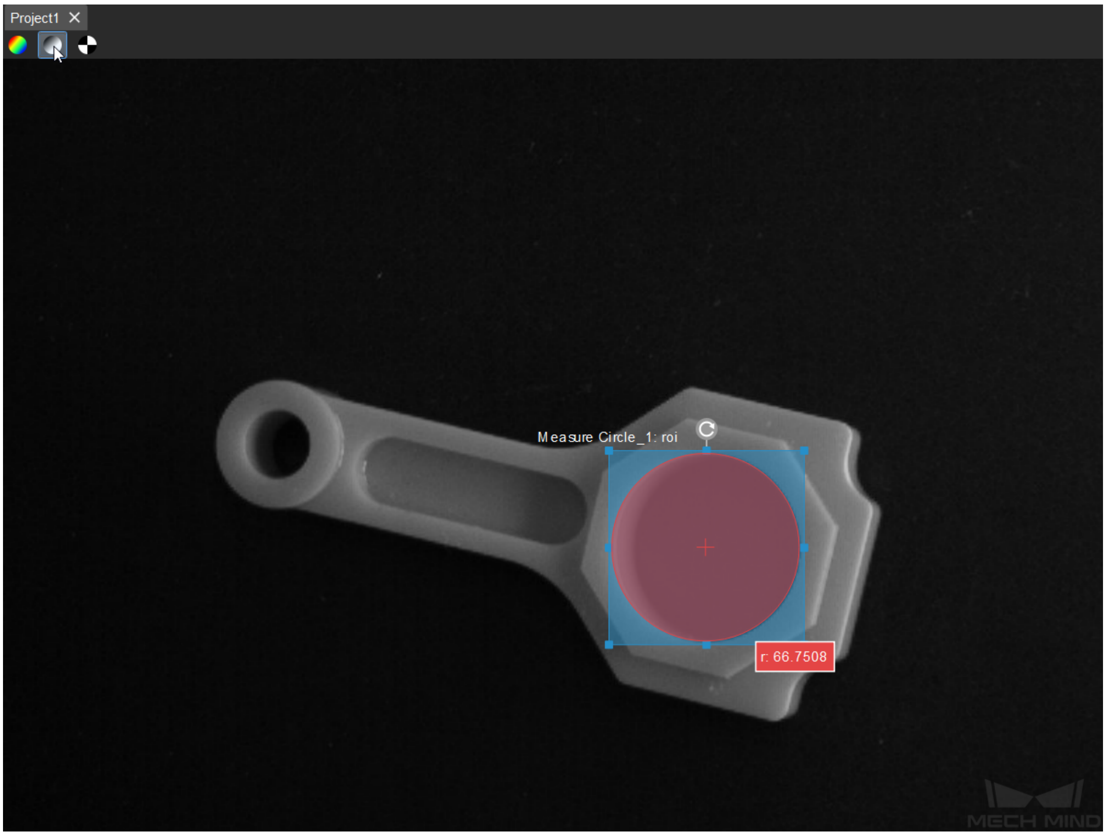
   Click on *Run* in the top middle of the graphical programming workspace or click on  of the Step *Capture Images from Camera* to run the project. Then the color image will appear in the sketchpad area.

4. Select the circular part for detection

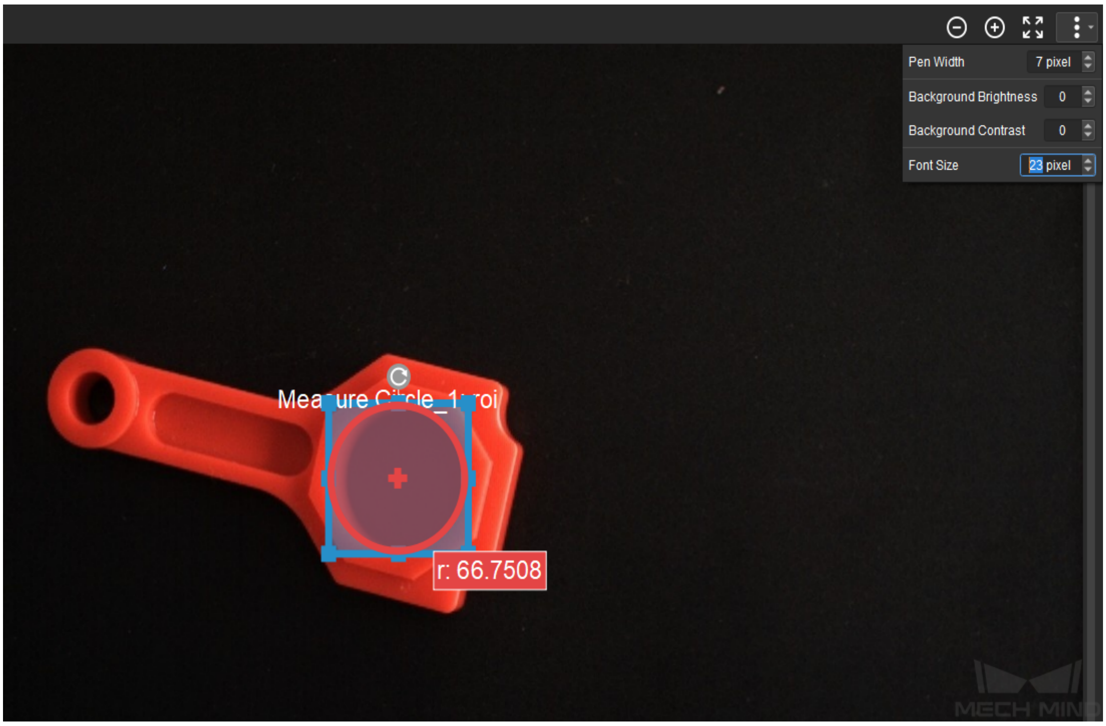
- Click on *Measure Circle* first. The select frame is in the upper left corner of the image. It is recommended to zoom out the image first and then adjust the frame to a proper size. Then you can drag it to the circular part in the image and therefore select an ROI.

Hint: The sketchpad is used for visualization, selecting an ROI, and adding points.

- You can switch the image display type between color image, gray image, and binary image by clicking on the buttons in the upper left corner.



- You can click on the buttons in the upper right corner of the sketchpad to zoom in or out the image. Click on to restore the image scale to 100%. Click on to adjust the **Pen Width**, **Background Brightness**, **Background Contrast** and **Font Size**.



Hint: You can also scroll upwards or downwards to zoom in or out the image.

8.1.4 Output

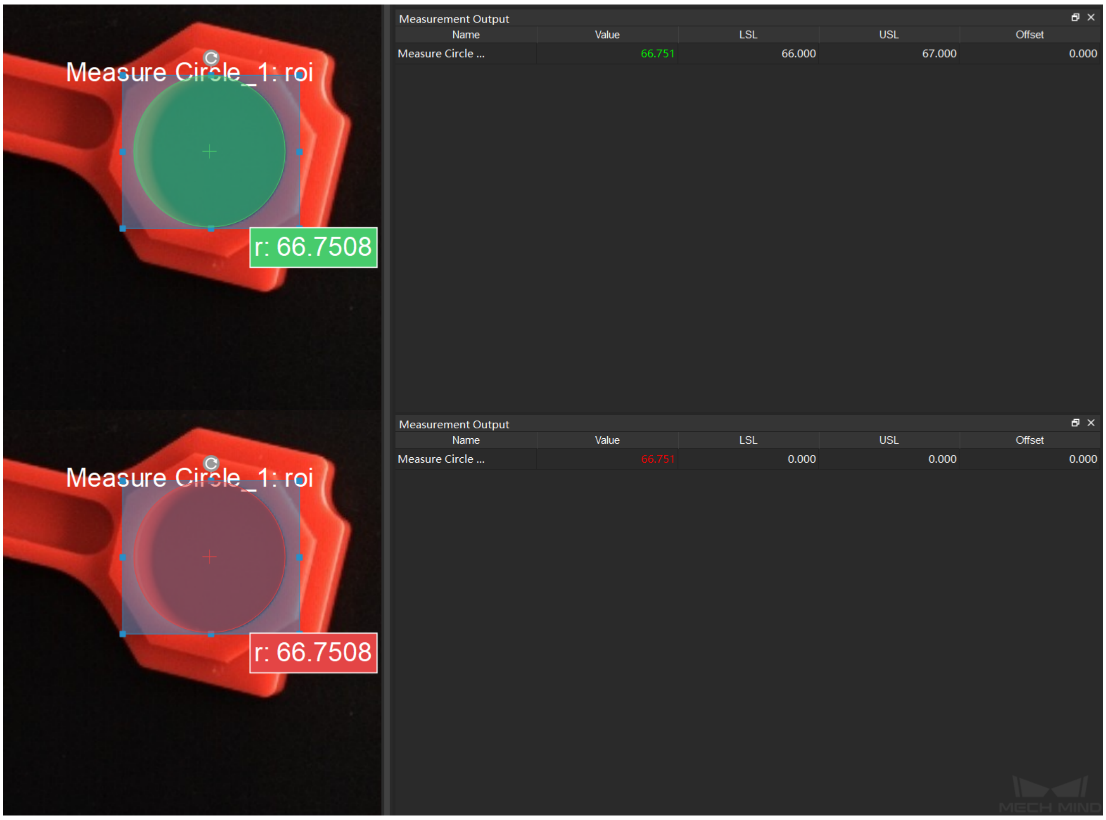
1. Read the measurement

   - 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.

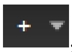
   Hint: If you cannot see the **Measurement Output** panel, please go to *View* and check the box before **Measurement Output**.

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. 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.



- Result View

The **Result View** in Measurement Mode provides a more user-friendly display of the measurement result. Click on , 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.

- Operator Interface

  Click on *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!

8.1.5 General Workflow in Measurement Mode

1. Create a New Project and switch to measurement mode.
2. Connect the Steps to program and configure parameters in Steps.
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.

Check the section below to learn about the **Main Algorithms** in measurement mode.

8.2 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 Circle* , *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 Distance from 3D Points to Plane* , 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.

   Relevant Steps: *Instance Segmentation* , *Defect Detection* , etc.

CHAPTER

NINE

ADVANCED GUIDE

This advanced guide is intended for users who are familiar with the basics of Mech-Vision and provides more information on the software's features and typical application projects.

Read the sections below to learn about **project-related** functions.

9.1 File Structure of a Mech-Vision Project

Figure 1 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.

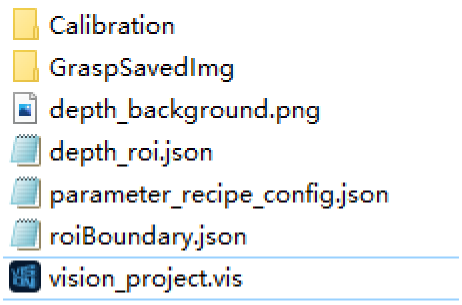


Figure 1. File structure of a Mech-Vision Project

9.2 Create Project

There are two ways to create a new project:

1. Click on *File* → *New Project* on the Menu bar.
2. Use the shortcut Ctrl + N .

It is recommended to set a project path and always save the project in time. Please see *Save Project* for more details.

9.3 Open Project

Open an existing project:

- Click on *File* → *Open Project* (shortcut Ctrl + O). Just select the corresponding folder. There is no need to locate the specific .vis file.
- Or, double-click the .vis file in the project folder to start Mech-Vision and automatically load the project.

Open a recently opened project:

Find it in *File* → *Open Recent*··· and click on it to open.

Tip:

- If Mech-Vision 1.4.0 or above is used to open a project created with a lower version of Mech-Vision, the original .json algorithm file will be automatically replaced by a .vis file after saving, and a backup .bak file will also be generated.
- Click on *File* → *Save Project as JSON* to save the project in .json format if necessary.

Mech-Vision supports opening multiple projects at the same time:

The file names of the opened projects are displayed in the Project panel on the left, as shown in *Figure 2*. Left-click the project name (no. 1 or no. 2 in the figure) to see the editing status of the selected project in the Graphical Programming Workspace.

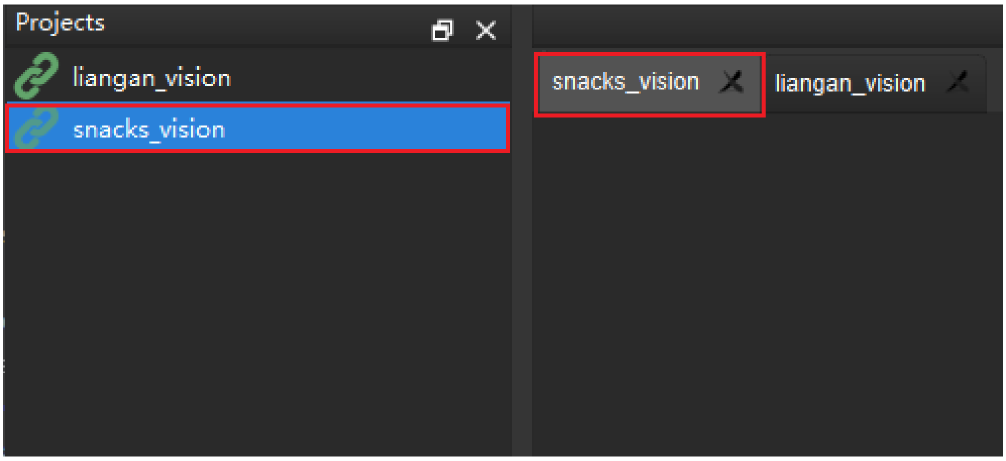


Figure 2. Switching between projects when multiple projects are opened

Attention:

- Please make sure that the folder name is identical to the name of the .vis file, or else the file cannot be opened.
- When using a virtual camera, files of the intrinsic parameters and calibrated extrinsic parameters of the camera should be included in the folder.
- When using a real camera, the file of the calibrated extrinsic parameters needs to be included in the folder. The intrinsic parameter file will be automatically downloaded when the project runs.

9.4 Add and Connect Steps

Select a Step from the Step Library panel on the left, press and hold the left mouse button to drag and release the Step to the Graphical Programming Workspace, and then connect the corresponding Steps based on actual project requirements. Steps can be connected by dragging the output of one Step to the input of another. Only input and output modules with matching formats can be successfully connected. For example, a *Cloud(Normal)* output can only be connected to a *Cloud(Normal)* or *Cloud(Normal)[]* input.

9.5 Run the Project and Debug

Attention:

- Mech-Vision 1.4.0 will perform a version compatibility check when running a project. It is recommended to use Mech-Viz 1.4.0 and Mech-Center 1.4.0 with Mech-Vision 1.4.0.
- If the version of Mech-Viz or Mech-Center used is lower than 1.4.0, a risk alert window will pop up when running a project.

Click on "Run" in the Toolbar or use the shortcut Ctrl + R to run the current Mech-Vision project, as shown in *Figure 3*.

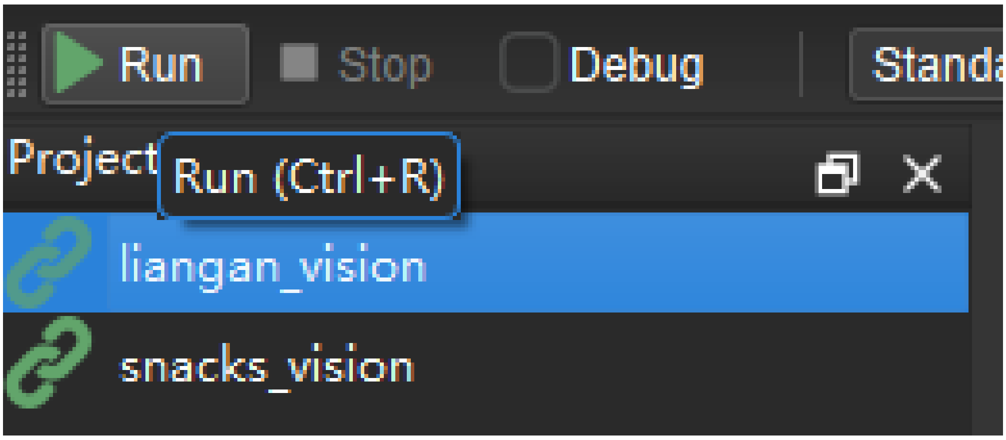


Figure 3. Run the Current Project

When the project 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 Step(s) whose result does not match expectations should be optimized through adjusting the Step parameters. For detailed instructions, please see *Guide to Steps*. For each executed Step, the execution time of the Step is displayed in the lower-left corner of the Step box, as shown in *Figure 4*.

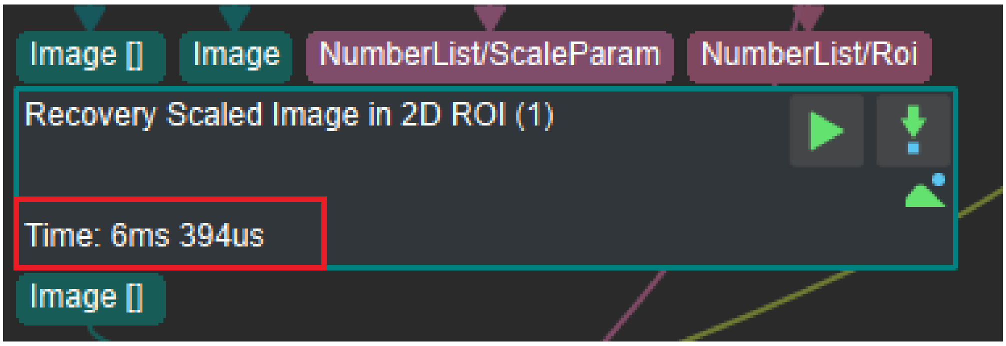
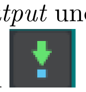


Figure 4. Display of the Running Time of a Step

There are three ways to view the debugging information of each Step:

1. Double-click the data flow (the connection arrow between steps) to view the content of the data flow;
2. Check the global *Debug* on the Toolbar and *VisualizeOutput* under *Execution Flags* in the Property panel, and then run the Step individually by clicking on  (Single Step Execution) to view the image output;
3. Check the global *Debug* on the Toolbar and *TextOutput* under *Execution Flags* in the Property panel, and then run the Step individually by clicking on  to view the text output.

The execution status of the operations performed by the project will be displayed in the Log panel, as shown in *Figure 5*. If the project has an error that cannot be readily located through the above debugging methods, the details of the error can be found in the log.

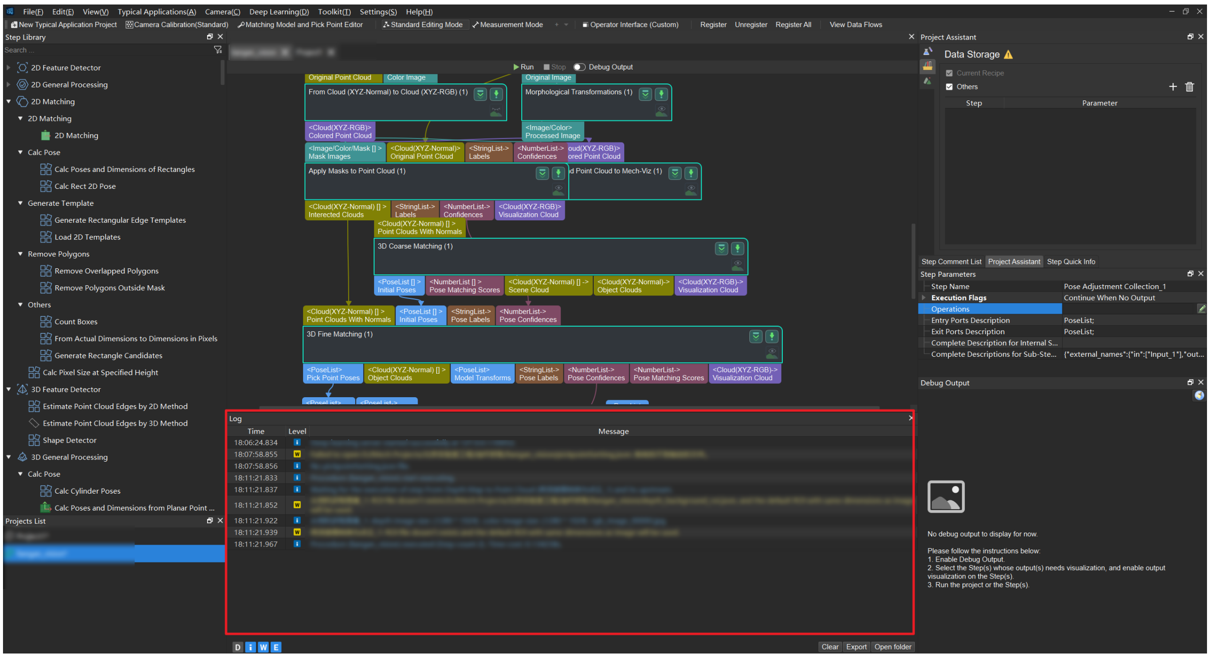


Figure 5. Execution status displayed in Mech-Vision Log

9.6 Save Project

A project can be saved by clicking on *File* → *Save Project* or using the shortcut Ctrl + S.

Please note that a Mech-Vision project has to be saved to an empty folder, which should be created before saving the project and selected as the save location during saving. A .vis file with the same name as the folder will be generated in this folder after saving; double-clicking on this .vis file opens this project in Mech-Vision.

9.7 Autoload Project

Right-click on a project in the Projects panel and check *Autoload Project*, as shown in *Figure 6*. This project will be opened automatically when Mech-Vision is started next time.

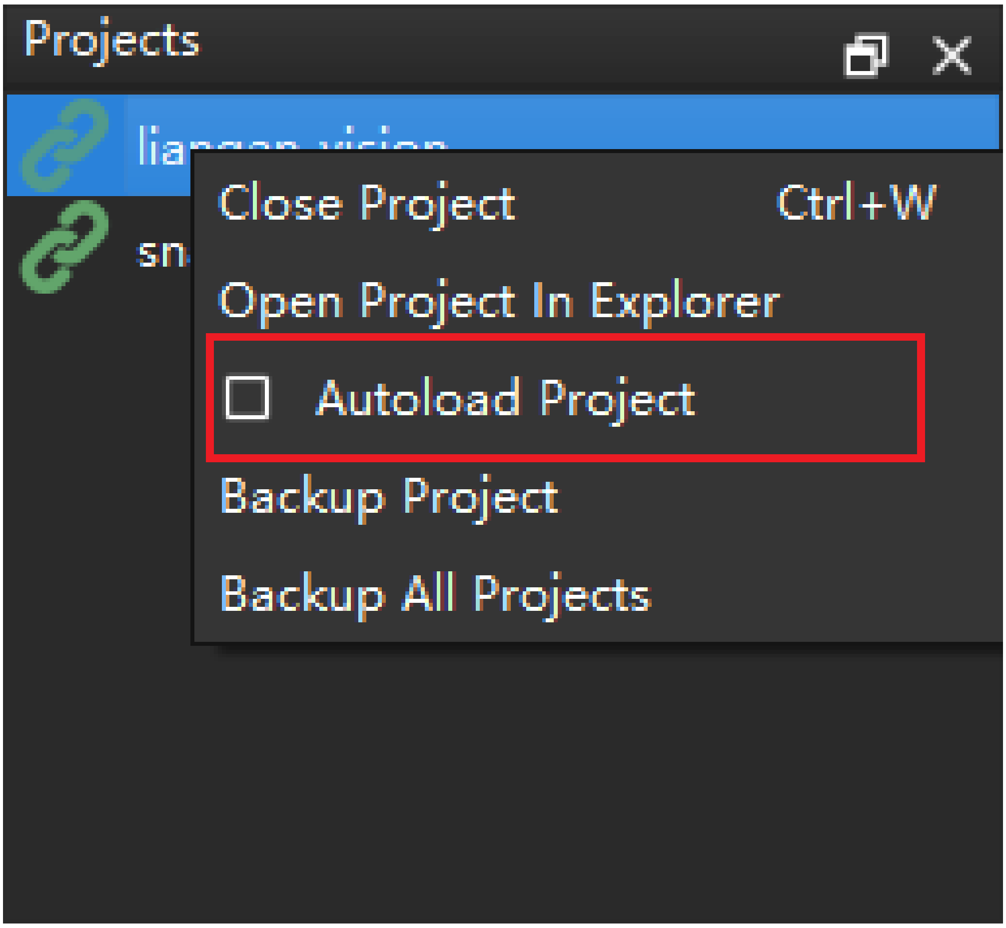


Figure 6. Autoload Project

9.8 Close Project

A Mech-Vision project can be either closed only in the Graphical Programming Workspace or completely exited from Mech-Vision.

1. Only closing a project in the Graphical Programming Workspace: Click on the X sign next to the project name in the Workspace, as shown in *Figure 7*. This operation only exits the project from programming. The project is still opened in Mech-Vision and listed in the Projects panel. Double-clicking on the project name in the Projects panel will reopen it in the Graphical Programming Workspace.

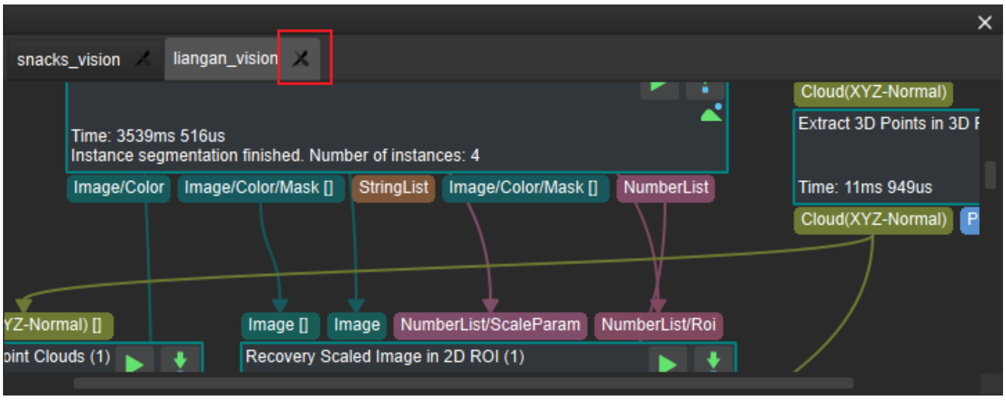


Figure 7. Close a project in the Graphical Programming Workspace

2. Completely exiting a project from Mech-Vision: Right-click on the project name in the Projects panel, and then click on *Close Project*, as shown in *Figure 8*. If there are unsaved contents in the project, a warning window as shown in *Figure 9* will pop up. Please select *Save*, *Discard*, or *Cancel* according to actual needs.

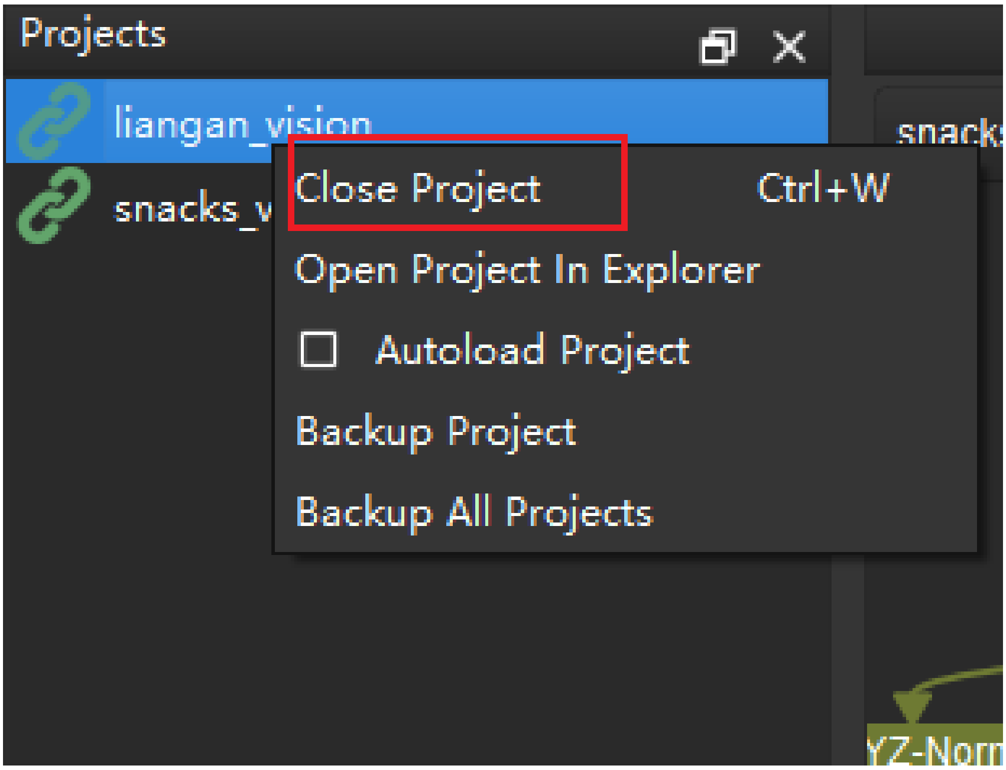


Figure 8. Close a project in the Projects panel

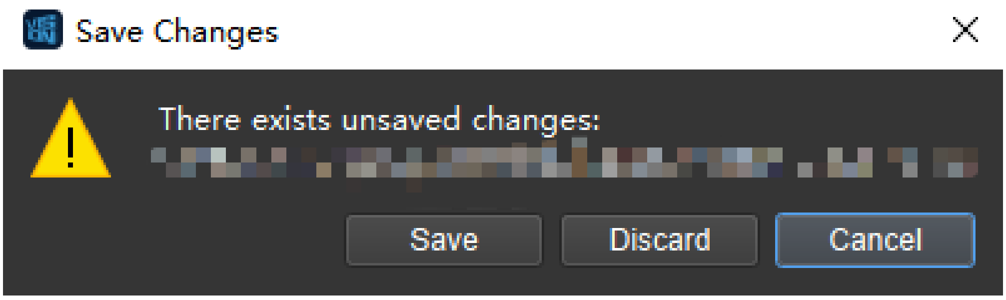


Figure 9. Alert to save the project before closing

Read the sections below to learn more about **workflows** and **constituting Steps** of typical application projects.

9.9 Sacks

This Typical Project adopts deep learning and 3D vision algorithms to recognize sacks and is applicable to depalletizing scenarios of sacks with various surface patterns.

Sack picking tasks are common in the logistics industry. A typical task is moving sacks onto conveyor belts, and the task's basic operation is to pick up a sack and put it at a designated place.

This project involves two sub-tasks:

1. Obtain the pick point pose by recognizing a sack and calculating the sack's pose.
2. Determine the overlapping condition of the sacks. If the target sack is overlapped by other sacks, the gripper on the robot may collide with the overlapping sacks. Therefore, those not blocked by other sacks are prioritized for picking.

9.9.1 Mech-Vision Project Workflow

The project has two concurrent sub-workflows.

1. Generate the scene point cloud, preprocess the point cloud, find the sacks in the 2D image, and extract the pose of each sack.
2. Filter out the poses of sacks that are overlapped by others, and prioritize the poses of sacks not overlapped.

After the two sub-workflows finish, the pick points' poses are sorted.

The project workflow is shown in *Figure 1*.

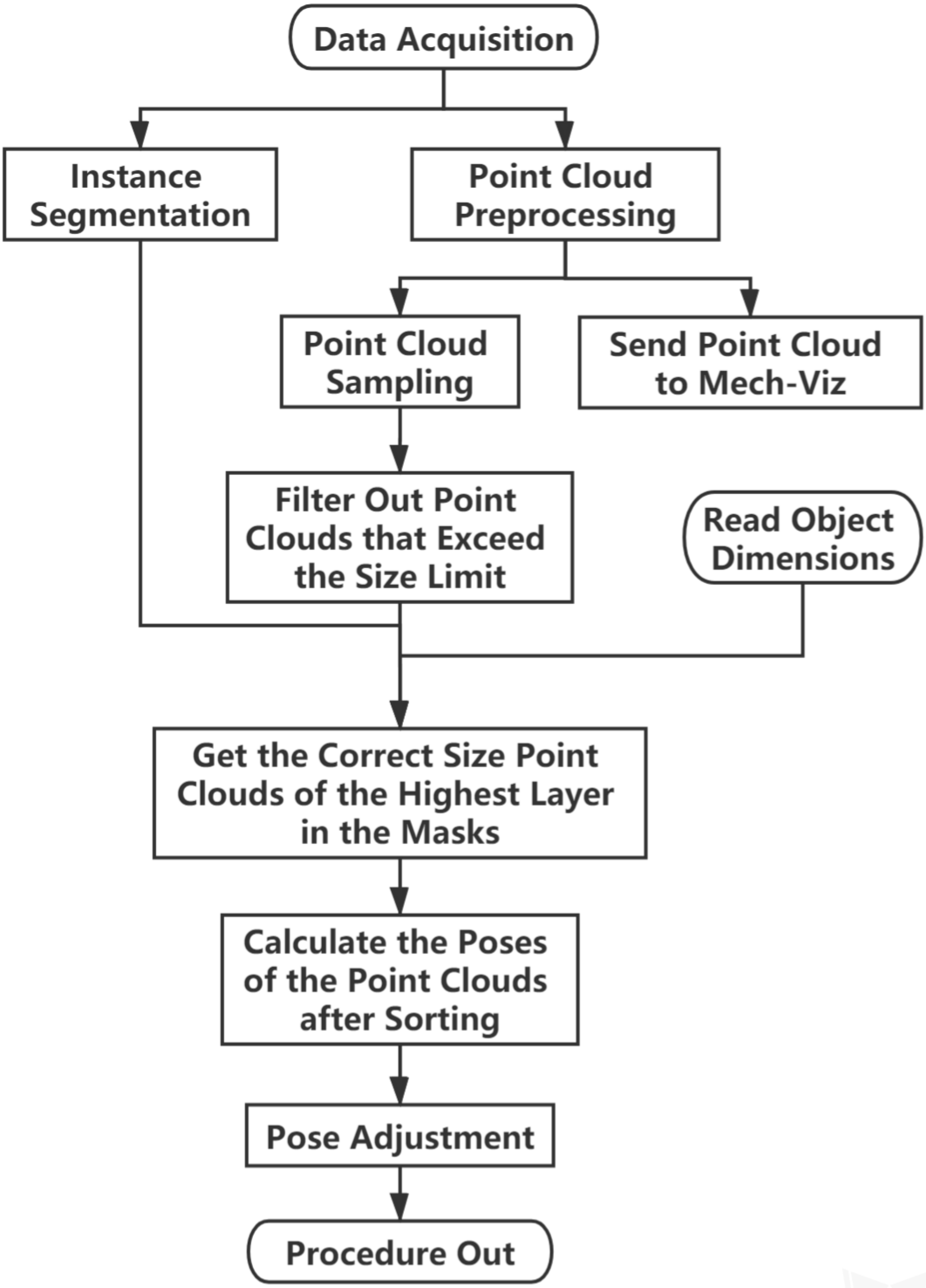


Figure 1. Workflow of sack depalletizing

First, instance segmentation is done by the deep learning model on the 2D image. The instance segmentation results are then used to extract point clouds of individual sacks from the scene point cloud.

To avoid collisions, the poses of sacks not overlapped by or closely fitted with other sacks are prioritized for picking. In the end, the pick point pose for each sack selected for picking is calculated and sent to Mech-Viz.

Sacks are soft and are prone to deformation, so there is no matching model point cloud. Therefore, deep learning is needed for recognizing sacks.

Knowing the dimensions of the sacks is a prerequisite for this project.

The graphical programming of this project is shown in *Figure 2*.

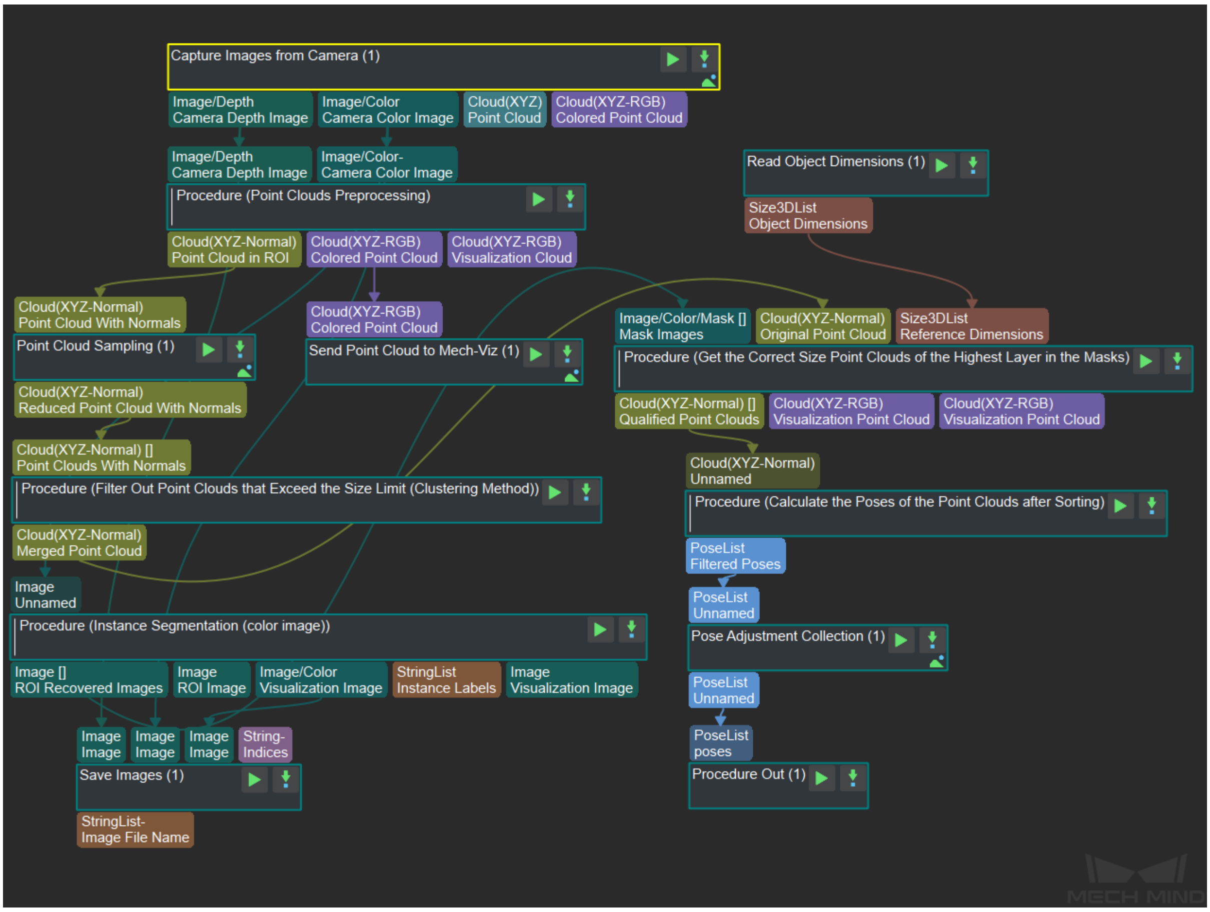


Figure 2. The graphical programming of a Typical Project for Sack Depalletizing

9.9.2 Steps and Procedures

A Procedure is a functional program block that consists of more than one Step.

Instance Segmentation (Color Image)

Instance segmentation by deep learning is used to recognize and generate masks for individual sacks, thus forming the basis for the subsequent task of obtaining point clouds for each mask. A sample result is shown in *Figure 3*.

Please see *Instance Segmentation (Colored Image)* for details about this Procedure.

Figure 3. A sample result of Instance Segmentation (Color Image)

Point Cloud Preprocessing

This Procedure facilitates and shortens the processing time for the subsequent calculations. Point Cloud Preprocessing generates a raw point cloud from the depth map and the color image, deletes the outliers, calculates the normals for the point cloud, and in the end extracts the part of the point cloud within the ROI.

For details about this Procedure, please see *Point Cloud Preprocessing*.

Down-Sample Point Cloud

This Step downsamples the point cloud to reduce its size.

Filter Out Point Clouds That Exceed the Limit

This Step filters out the point clouds with point counts exceeding the limit through point cloud clustering and merging.

A sample result is shown in *Figure 4*.

Please see *Point Cloud Clustering* for details about this Procedure.

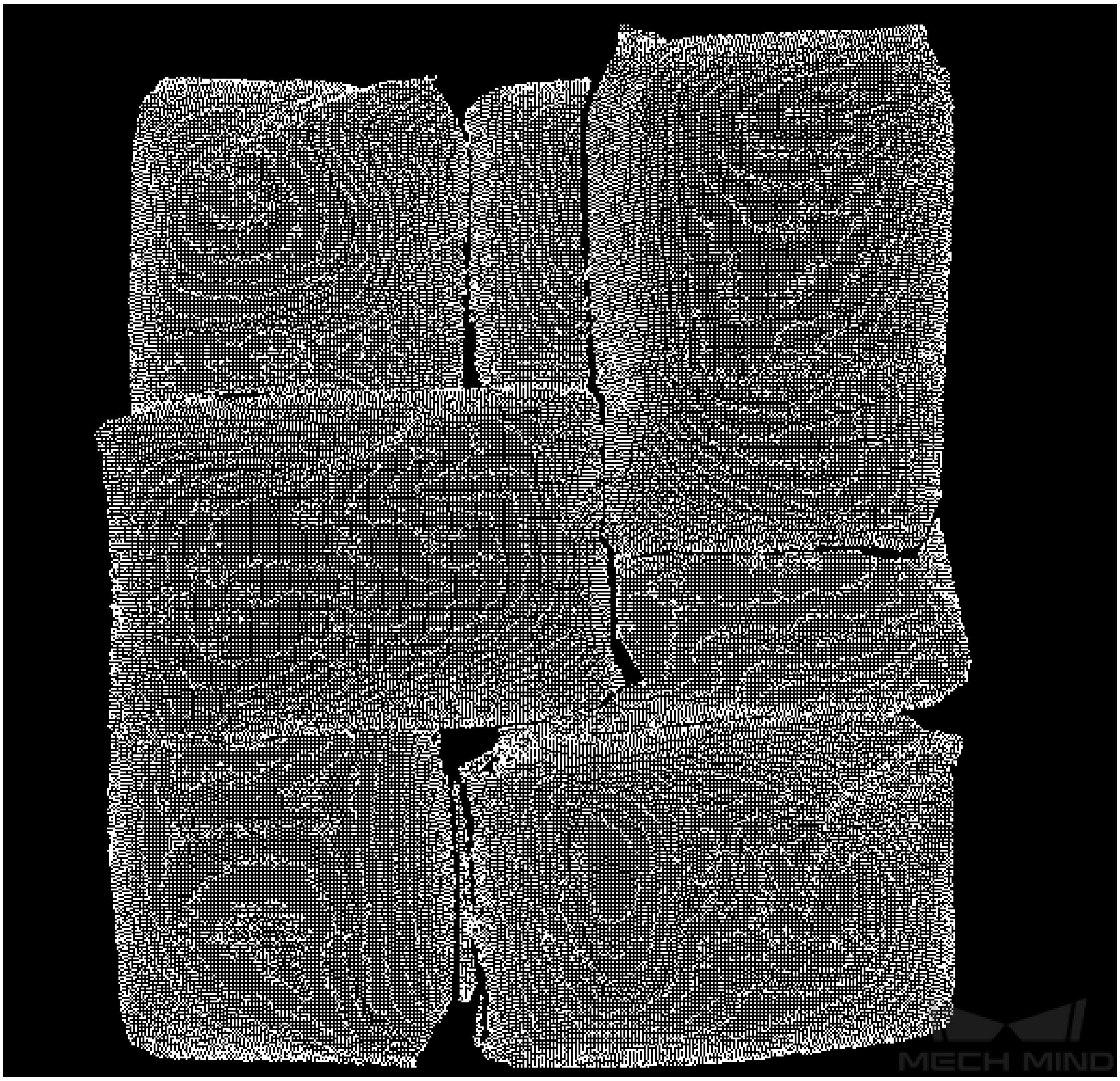


Figure 4. A sample result of Filter Out Point Clouds That Exceed the Limit

Get the Correct Size Point Clouds of the Highest Layer in the Masks

This Procedure obtains the highest layer point clouds that are of the correct size based on the sack positions segmented from the 2D image (under masks), the preprocessed scene point cloud, and the detected dimensions of the sacks.

In this project, this Procedure obtains the point clouds of the sacks on the highest layer.

A sample result is shown in *Figure 5*.

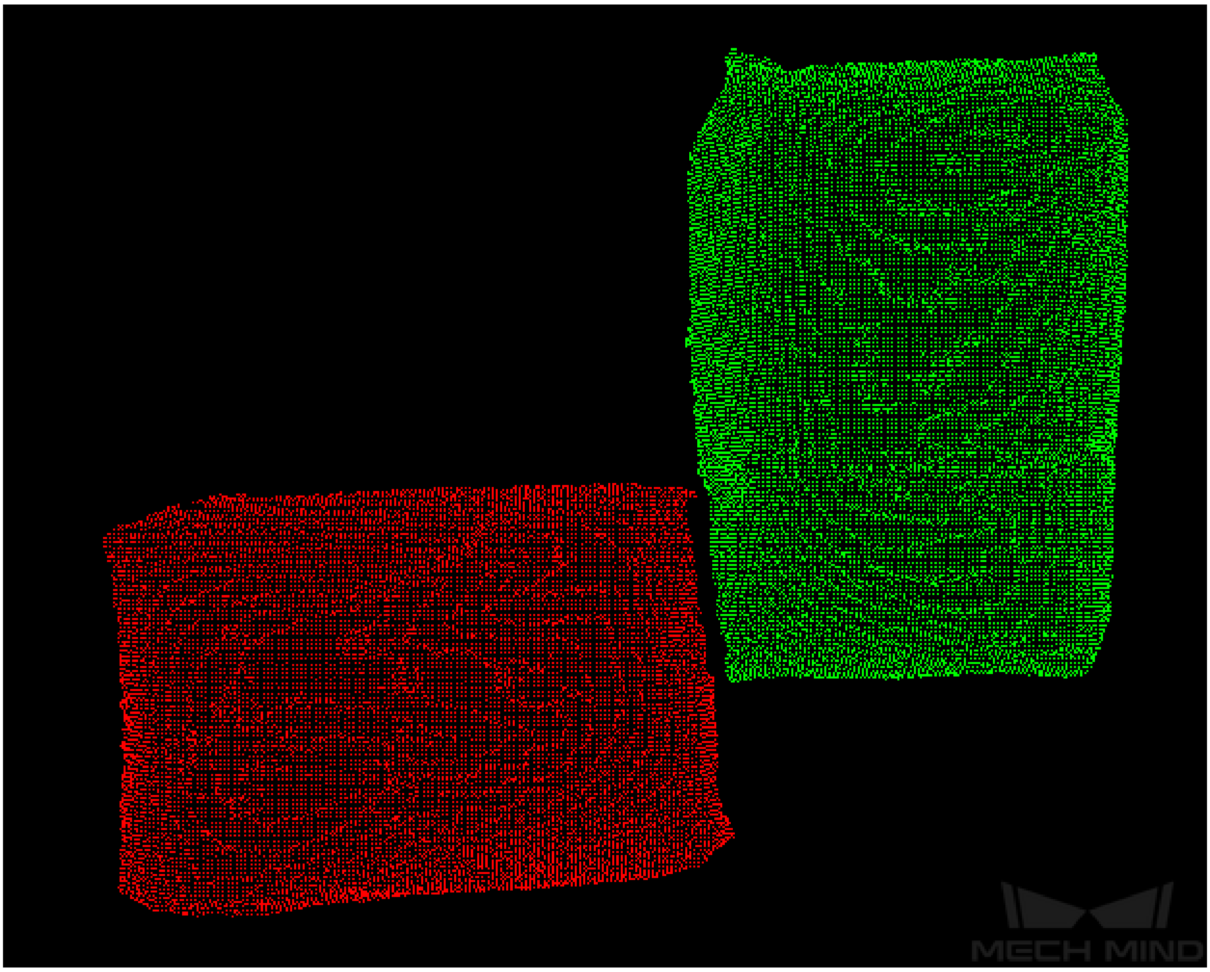


Figure 5. A sample result of Get the Correct Size Point Clouds of the Highest Layer in the Masks

Calculate the Poses of the Point clouds after Sorting

This Procedure sorts the sack point clouds, and performs fitting on the surface to get the sacks' surface poses.

In this project, this Procedure obtains the point clouds of the pickable sacks, i.e., sacks that are not overlapped or closely fitted with other sacks.

Pose Adjustment Collection

This Step is for adjusting the poses in the pose editor.

Please see *Pose Editor* for instructions on adjusting poses.

9.9.3 Debugging

The default parameters of the Steps in Typical Projects are not necessarily applicable to all scenarios.

If the instructions of the Steps and Procedures mentioned above do not suffice to identify the cause of a recognization issue, following the workflow below may help.

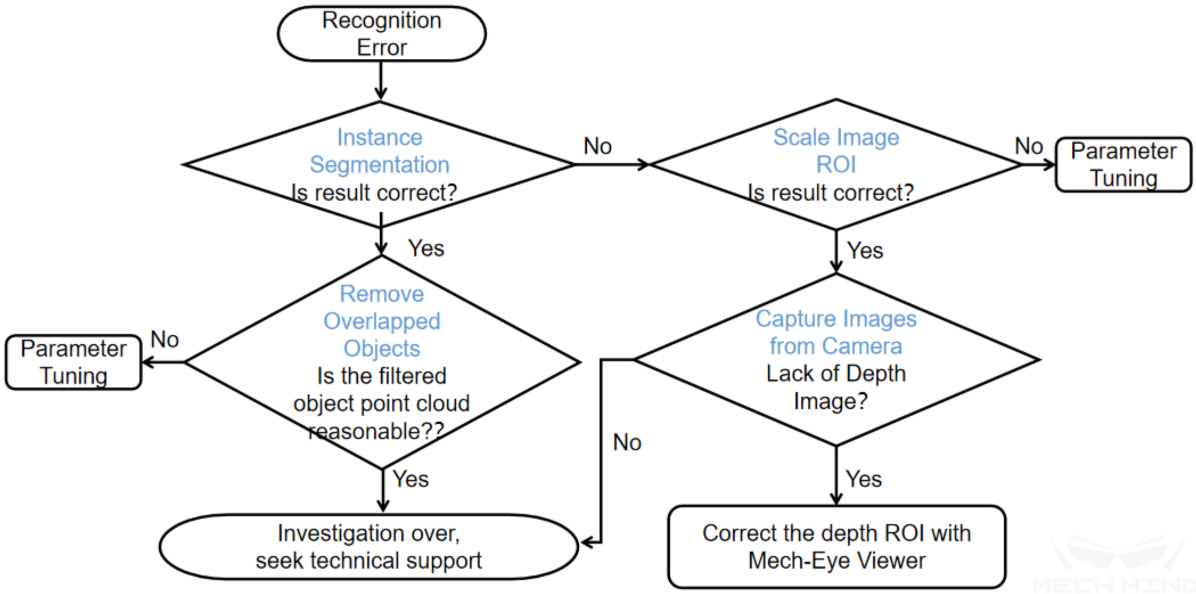


Figure 6. Identify the cause of a recognization issue

For instructions on tuning the parameters involved in the workflow above, please see *Scale Image in 2D ROI*, *Remove Overlapped Objects*.

For *Remove Overlapped Objects*, parameters to adjust are mainly point cloud bounding box resolution and overlap ratio threshold.

- Point cloud bounding box resolution is used to calculate the number of points in the bounding box. After calculating the number of points, the number of overlapping points of these points in other bounding boxes will be calculated.
- An object point cloud' s overlap ratio is obtained by dividing the point count of the overlapped part by the point count of the entire point cloud. The object point cloud is filtered out if its overlap ratio is above the threshold.

9.10 Large Non-Planar Workpiece Loading

9.10.1 Application Scenarios

The project is suitable for picking a small number of larger workpieces, usually when the workpieces have obvious features.

9.10.2 Mech-Vision Project Workflow

As the workpieces have obvious features, pose estimation can be easily done by matching between the object point cloud and a model point cloud. So, no deep learning is involved and the project can run fast. The workflow of a Typical Project for Large Non-Planar Workpiece is shown in *Figure 1*.

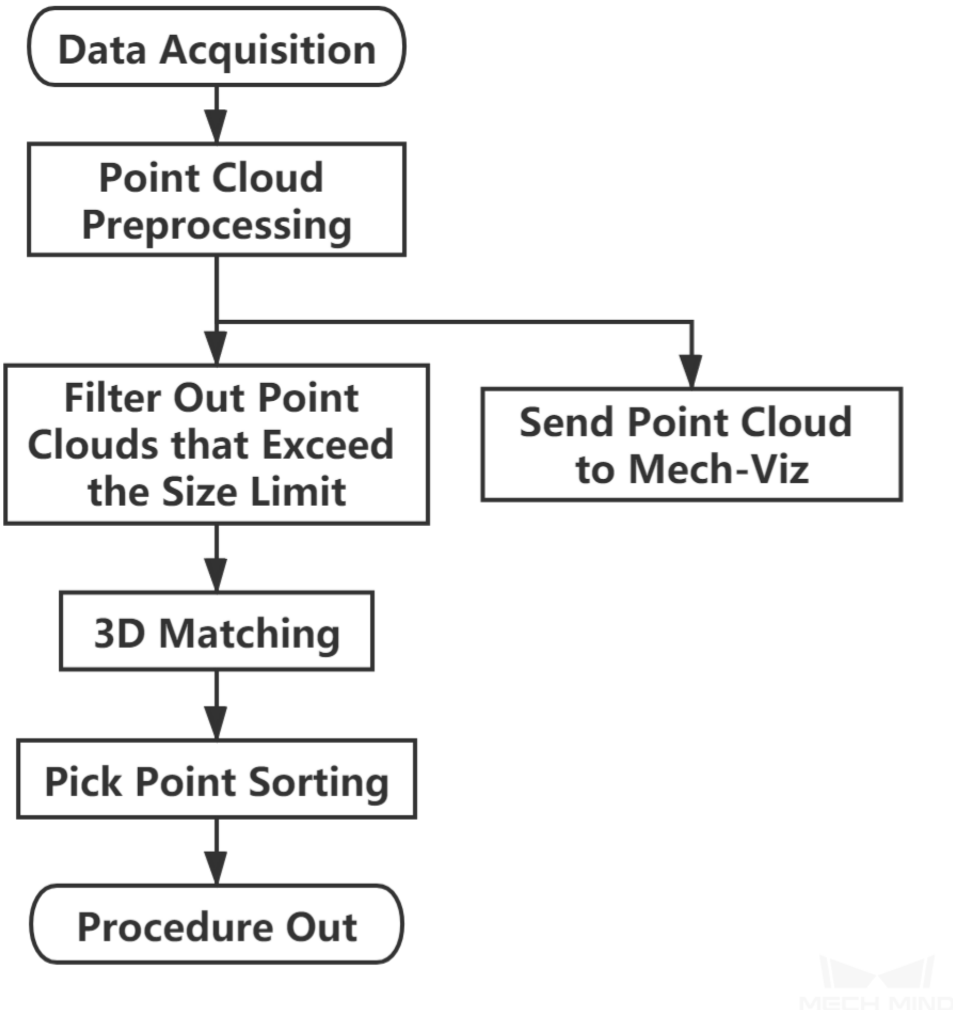


Figure 1. The workflow of a Typical Project for Large Non-Planar Workpiece

Figure 2 is a screenshot of the graphical programming of the project.

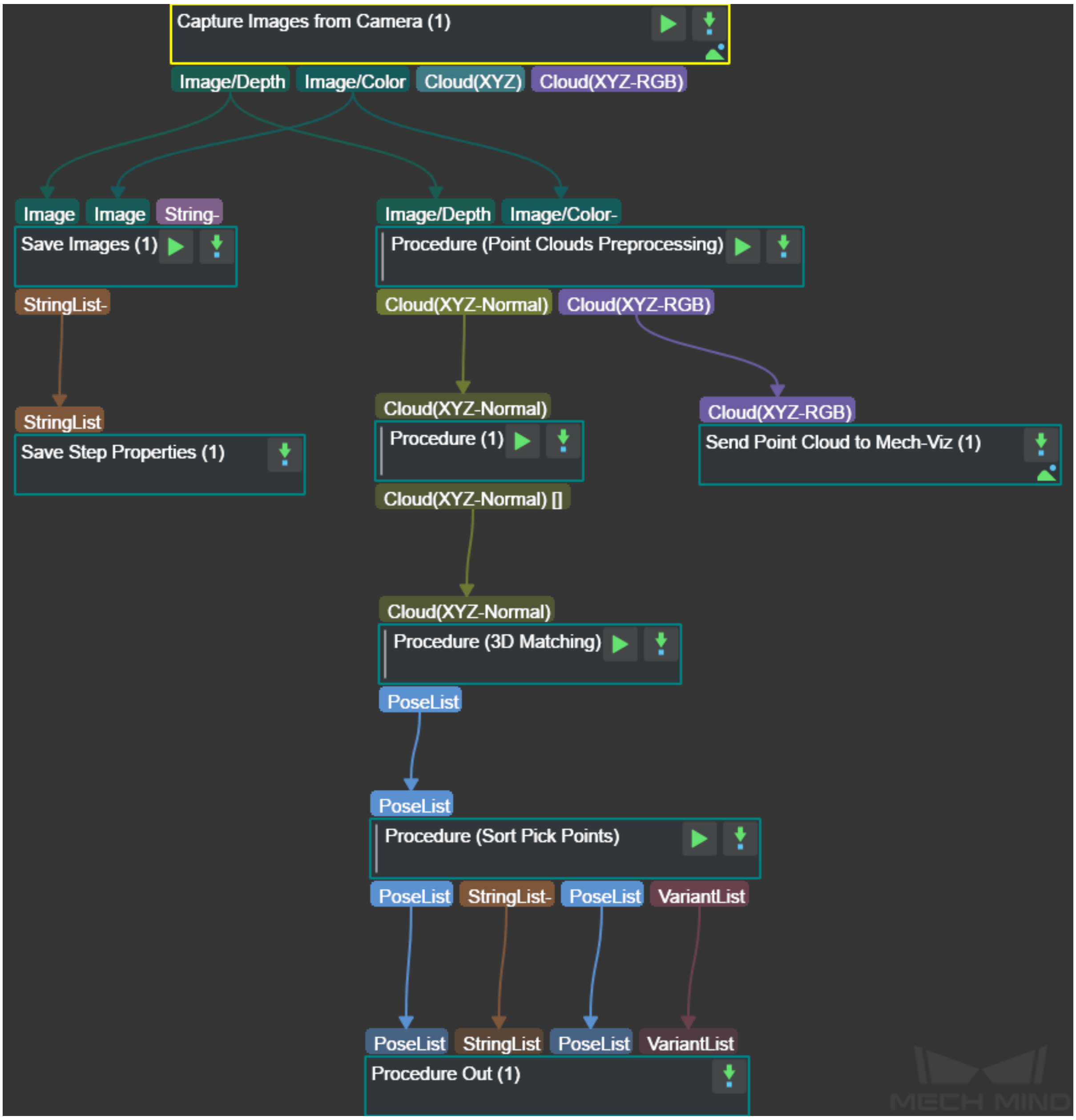


Figure 2. The graphical programming of a Typical Project for Large Non-Planar Workpiece

9.10.3 Steps and Procedures

A Procedure is a functional program block that consists of more than one Step.

Point Cloud Preprocessing

This Procedure facilitates and shortens the processing time for the subsequent calculations. Point Cloud Preprocessing generates a raw point cloud from the depth map and the color image, deletes the outliers, calculates the normals for the point cloud, and in the end extracts the part of the point cloud within the ROI.

For details about this Procedure, please see *Point Cloud Preprocessing*.

A sample result of Point Cloud Preprocessing is shown in *Figure 3*.

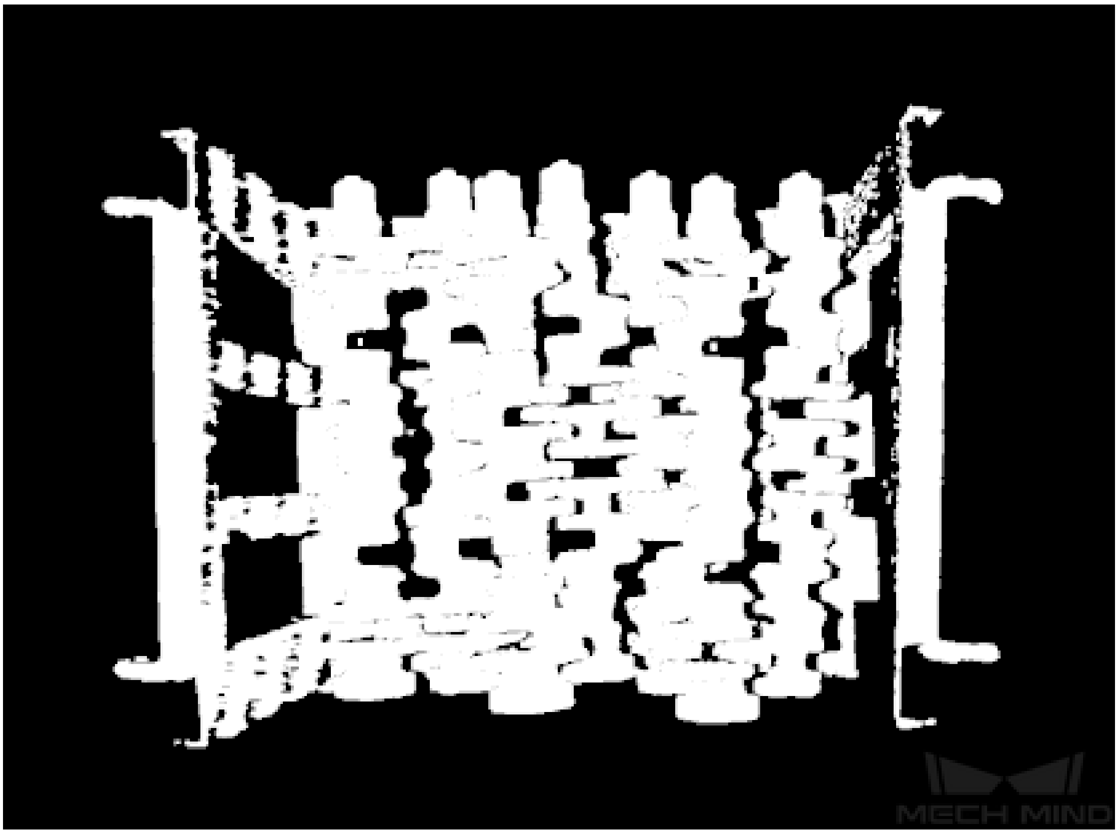


Figure 3. A sample result of Point Cloud Preprocessing

Filter Out Point Clouds That Exceed the Limit

This Procedure filters out point clouds that affect 3D matching to improve the accuracy of matching.

A sample result is shown in *Figure 4*.

Please see *Filter Out Point Clouds That Exceed The Limit* for the detailed description of the Procedure.

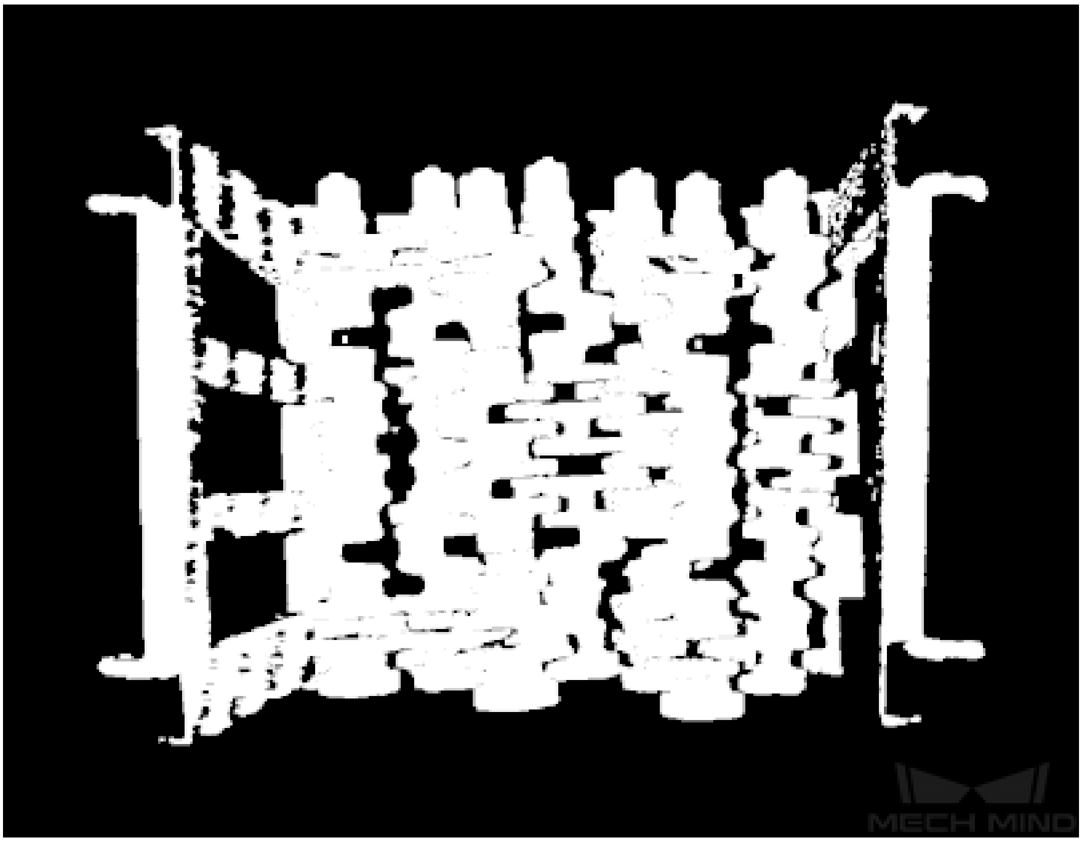


Figure 4. A sample result of Filter Out Point Clouds That Exceed the Limit

3D Matching

3D Matching is performed on the workpiece point cloud to obtain pick points after filtering the point cloud.

A sample result is shown in *Figure 5*. The colored parts of the point clouds match the model point cloud.

Please see *3D Matching* for detailed information about this Procedure.

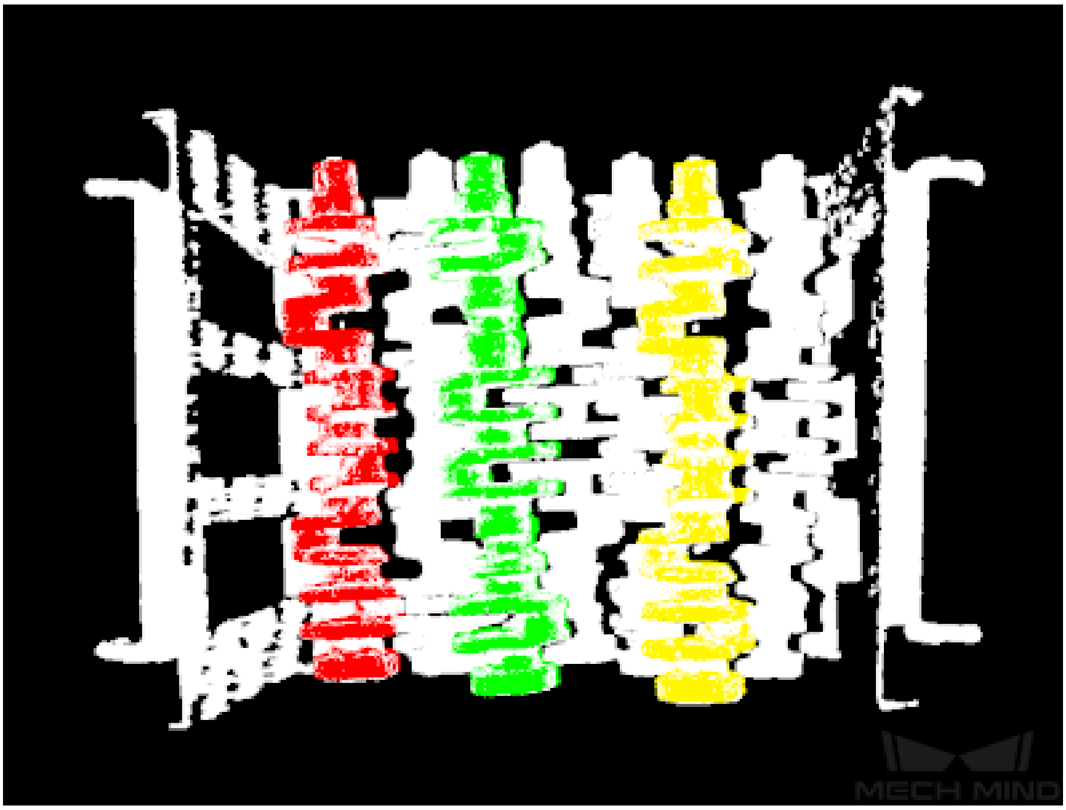


Figure 5. A sample result of 3D Matching

Pose Adjustment Collection

This Step is for adjusting the poses.

Please see *Pose Editor* for instructions on adjusting poses.

Map to Multi Pick Points

This Step takes a sorted pose list and maps them to a workpiece to provide different potential picking poses for the robot.

Please see *Map to Multi Pick Points* for details about this Step.

9.11 Small Non-Planar Workpiece Loading

9.11.1 Application Scenarios

The project is suitable for picking a large number of small workpieces. Because the workpieces come in large quantities, they may easily overlap, which affects feature recognizing.

In addition, the front and back faces of non-planar workpieces need to be distinguished, which is different from picking tasks of flat workpieces whose front and back faces are the same and do not need to be distinguished.

9.11.2 Mech-Vision Project Workflow

Because the small workpieces come in large numbers and may easily overlap, deep learning and 3D matching are needed. The workflow for a Typical Project for Small Non-Planar Workpieces is shown in *Figure 1*.

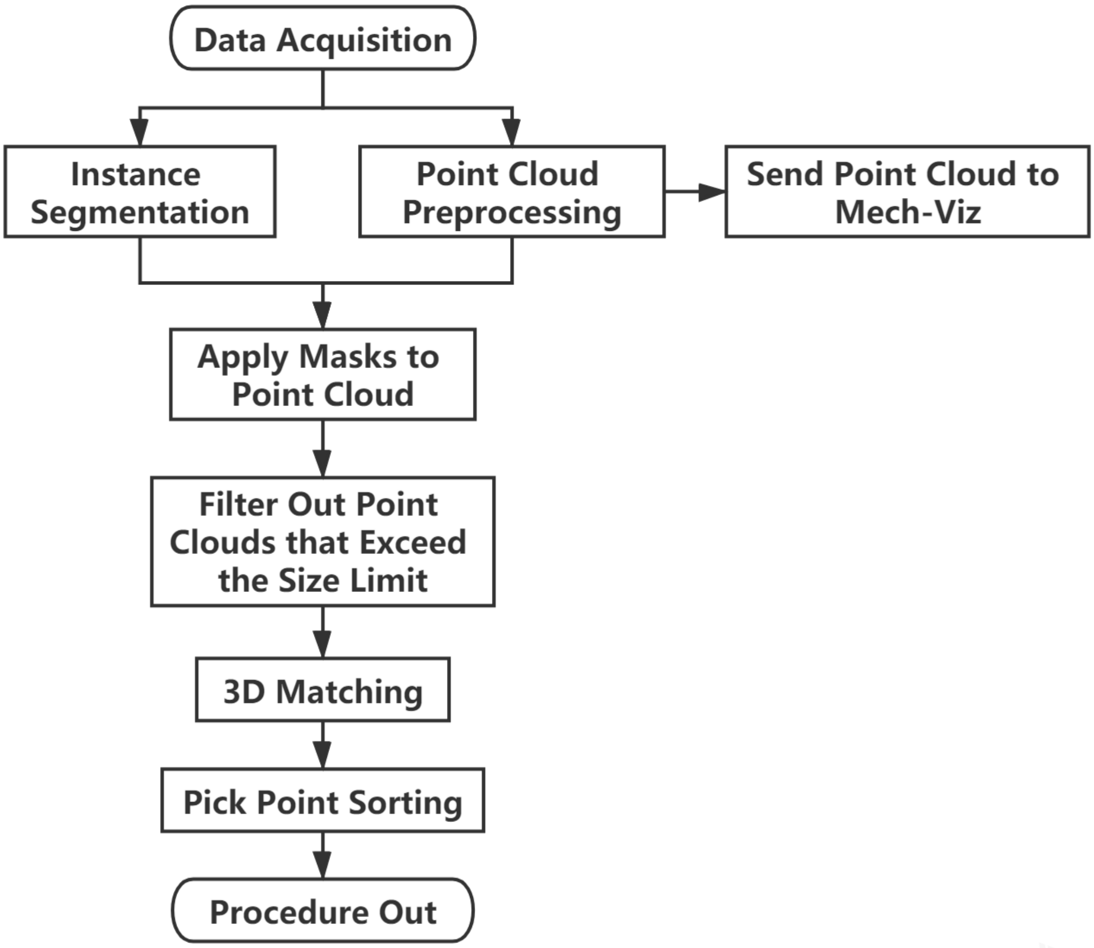


Figure 1. The workflow of a Typical Project for Small Non-Planar Workpieces

*Figure 2*is a screenshot of the graphical programming of the project.

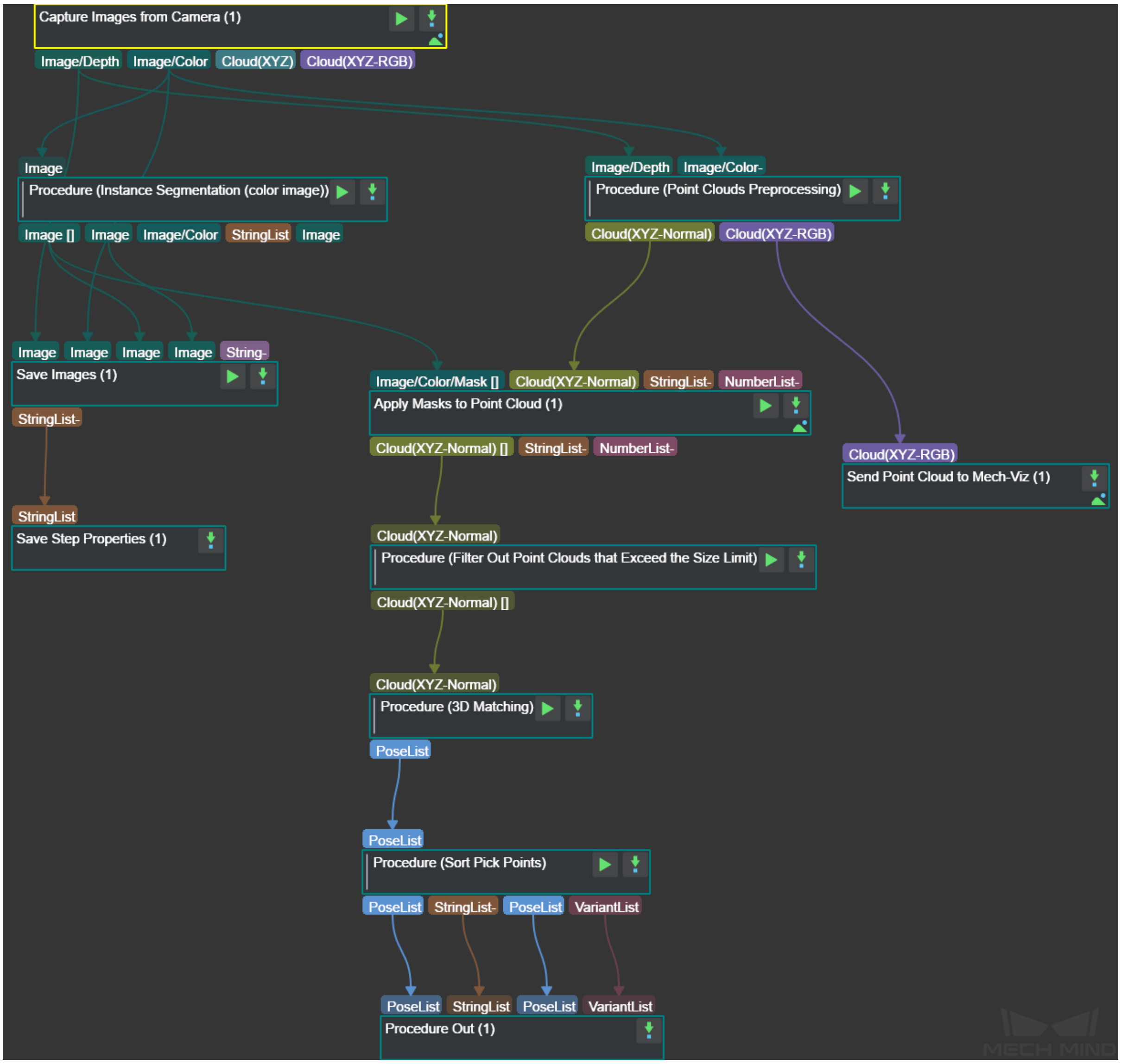


Figure 2. The graphical programming of a Typical Project for Small Non-Planar Workpiece

9.11.3 Steps and Procedures

A Procedure is a functional program block that consists of more than one Step.

Point Cloud Preprocessing

This Procedure facilitates and shortens the processing time for the subsequent calculations. Point Cloud Preprocessing generates a raw point cloud from the depth map and the color image, deletes the outliers, calculates the normals for the point cloud, and in the end extracts the part of the point cloud within the ROI.

For details about this Procedure, please see *Point Cloud Preprocessing*.

A sample result of Point Cloud Preprocessing is shown in *Figure 3*.

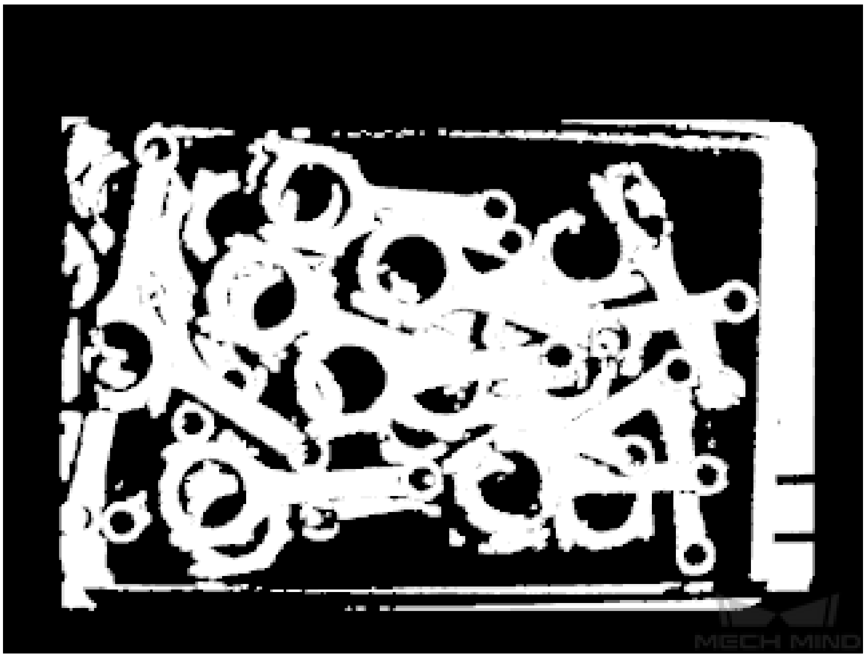


Figure 3. A sample result of Point Cloud Preprocessing

Instance Segmentation (Color Image)

In the Typical Project for Small Non-Planar Workpieces, instance segmentation by deep learning is used to find the planar projection of the workpieces and generate masks, thus forming the basis for the subsequent task of generating the workpiece point clouds. A sample result is shown in *Figure 4*.

Please see *Instance Segmentation (Colored Image)* for details about this Procedure.

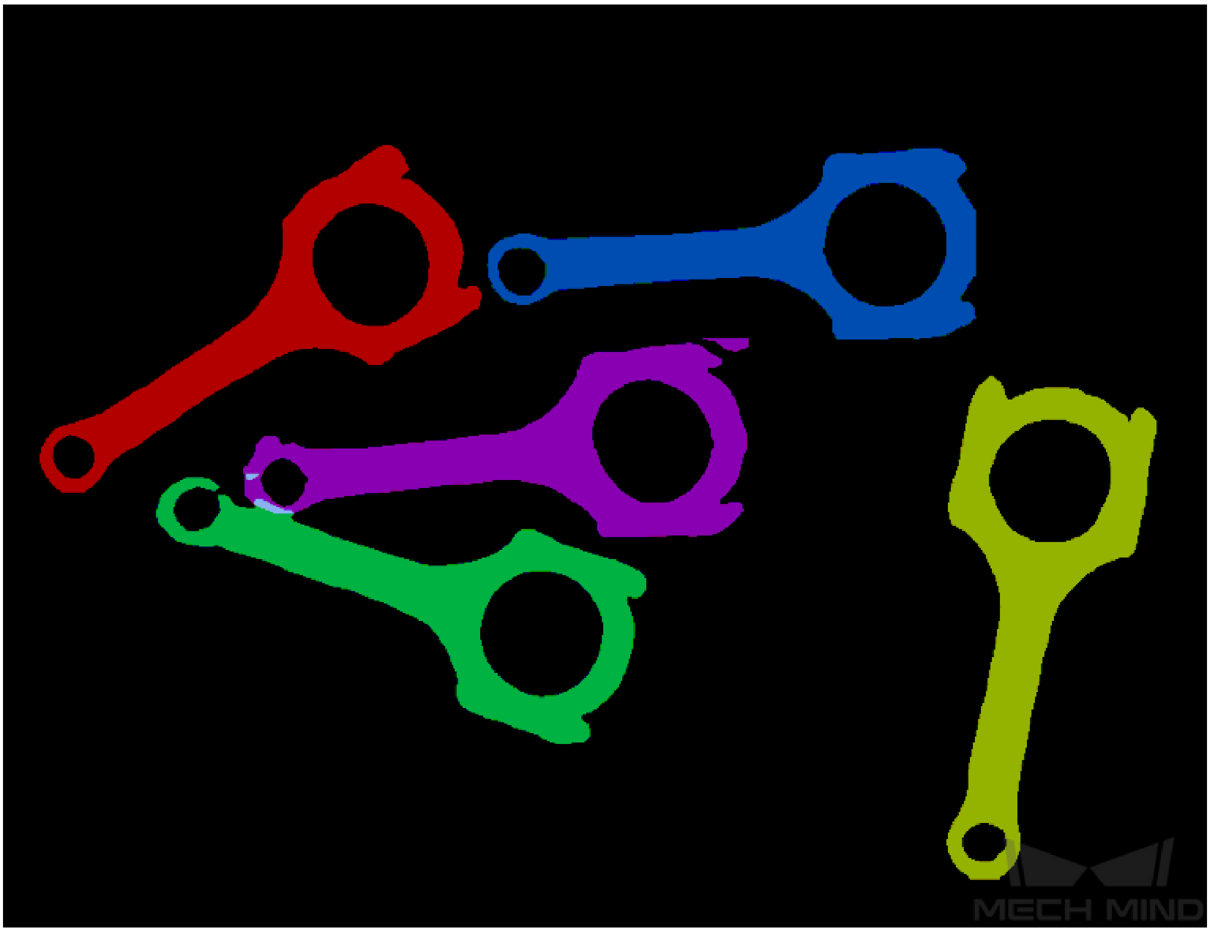


Figure 4. A sample result of Instance Segmentation (Color Image)

Apply Masks to Point Cloud

This Step extracts the part of the point cloud covered by a mask based on the masks and the point cloud of workpieces. A sample result is shown in *Figure 5*.

Please see *Apply Masks to Point Cloud* for detailed information about this Step.

Figure 5. A sample result of Apply Masks to Point Cloud

Filter Out Point Clouds That Exceed the Limit

After the point clouds of individual workpieces are generated, this Procedure filters out those that affect 3D matching to improve the accuracy of matching.

A sample result is shown in *Figure 6*.

Please see *Filter Out Point Clouds That Exceed The Limit* for details about this Procedure.

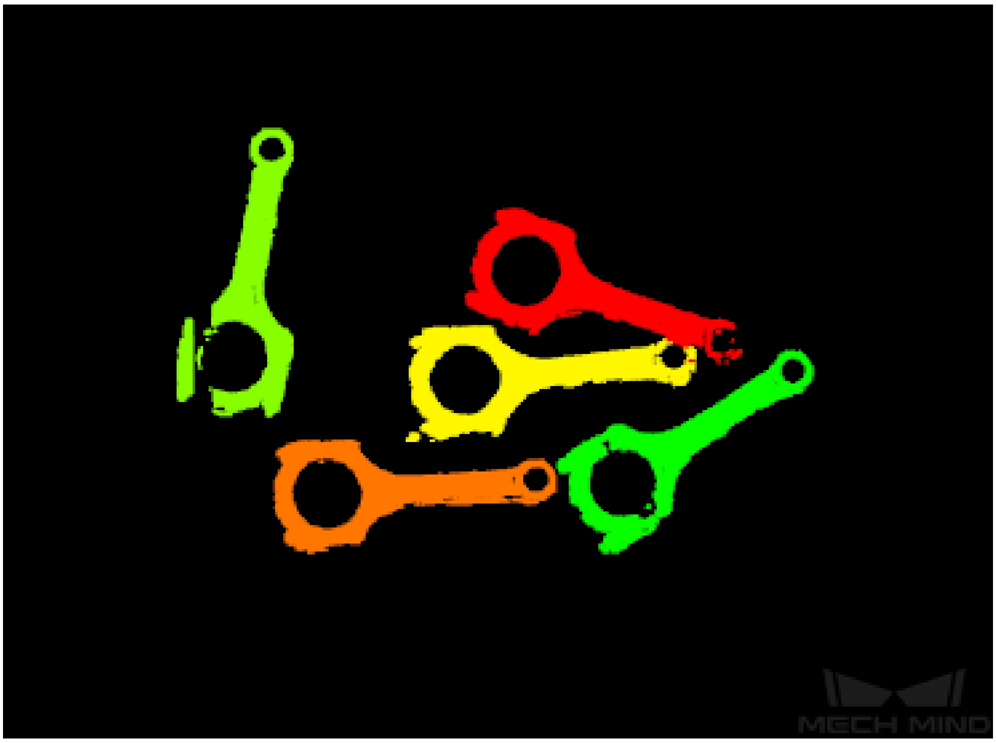


Figure 6. A sample result of Filter Out Point Clouds That Exceed the Limit

3D Matching

3D Matching is performed on the workpiece point clouds to obtain pick points after the point clouds are filtered.

A sample result of 3D Matching is shown in *Figure 7*. The colored parts of the point clouds match the model point cloud.

Please see *3D Matching* for details about this Procedure.

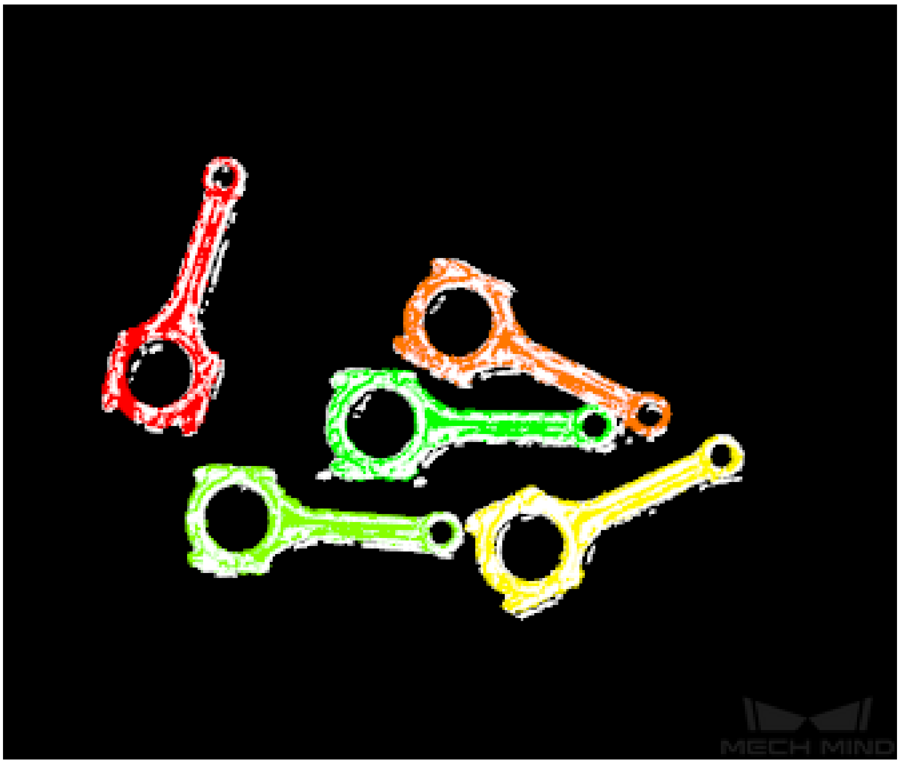


Figure 7. A sample result of 3D Matching

Pose Adjustment Collection

This Step is for adjusting the poses.

Please see *Pose Editor* for instructions on adjusting poses.

Map to Multi Pick Points

This Step takes a sorted pose list and maps them to a workpiece to provide different potential picking poses for the robot.

Please see *Map to Multi Pick Points* for details about this Step.

9.12 Overlapping Planar Workpiece Loading

9.12.1 Application Scenarios

The project is suitable for picking objects that have few features and irregular shapes, are prone to overlaps, and have the same front and back faces which means there is no need to distinguish the two faces.

9.12.2 Mech-Vision Project Workflow

As the workpieces come in large quantities and are prone to overlaps, and as the features are usually at the workpiece edges, only the part of point clouds of the workpiece edges should be extracted for 3D matching after instance segmentation to reduce the interference from the unusable non-edge workpiece features. The workflow is as shown in *Figure 1*.

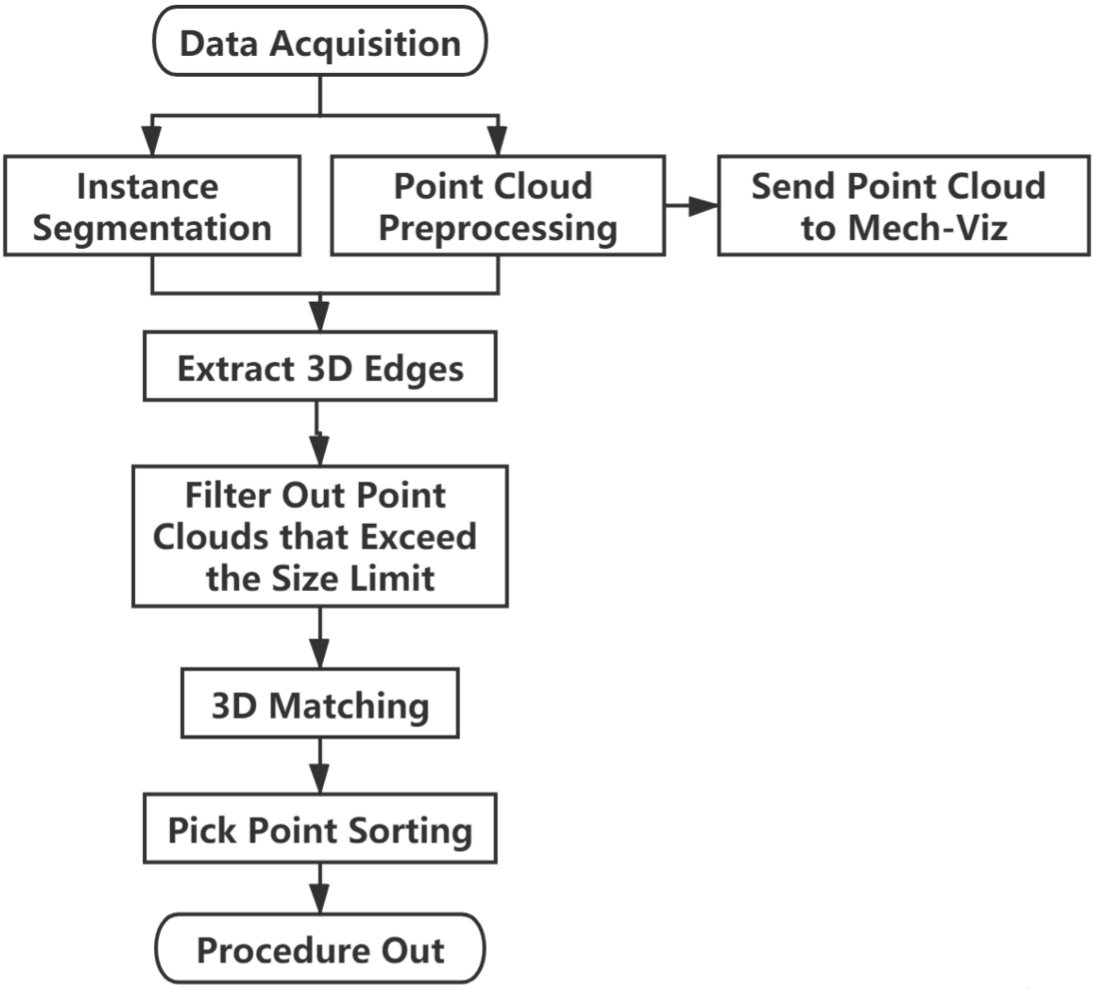


Figure 1. The workflow of a Typical Project for Overlapping Planar Workpieces

*Figure 2*is a screenshot of the graphical programming of the project.

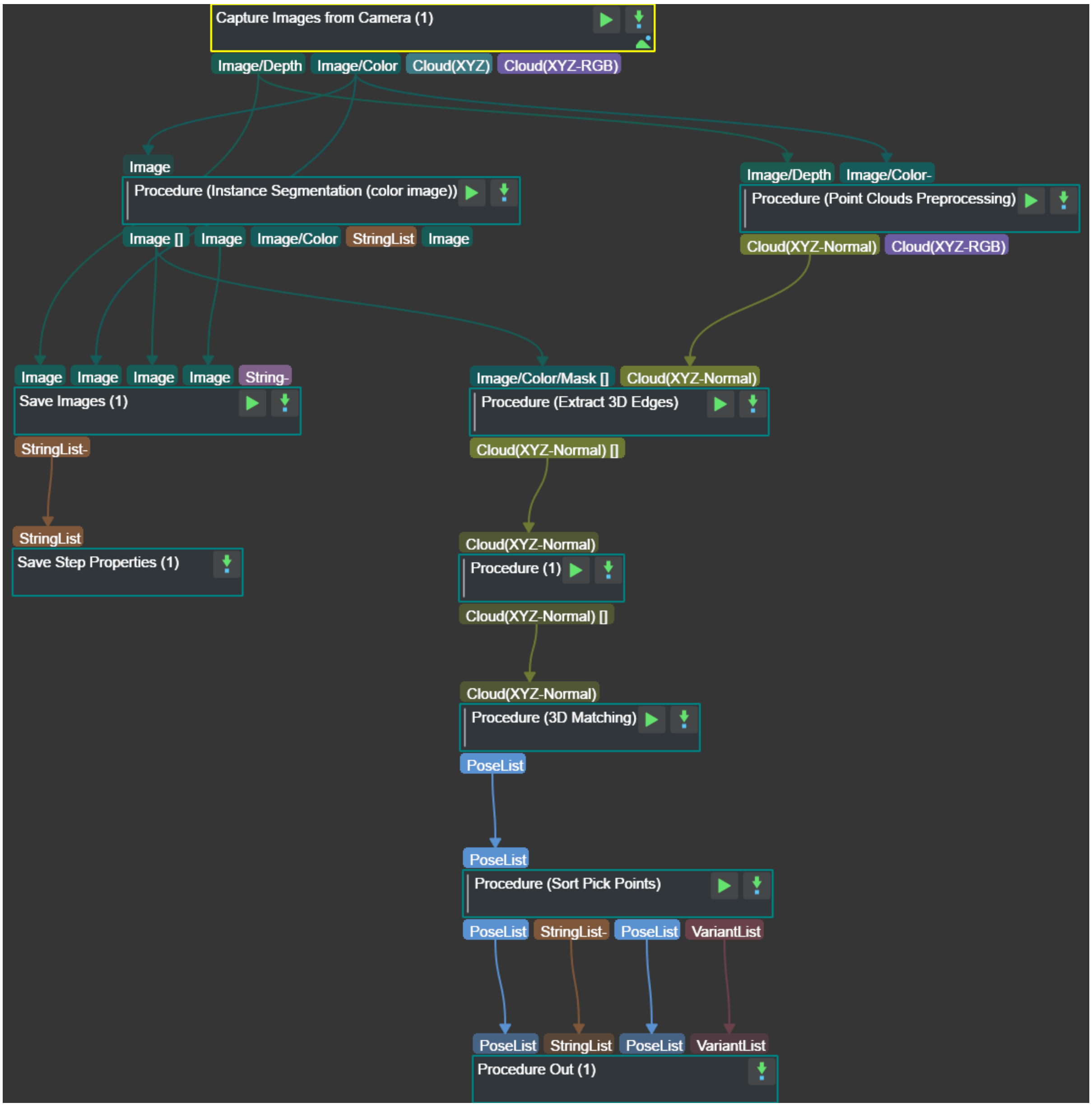


Figure 2. The graphical programming of a Typical Project for Overlapping Planar Workpieces

9.12.3 Steps and Procedures

A Procedure is a functional program block that consists of more than one Step.

Point Cloud Preprocessing

This Procedure facilitates and shortens the processing time for the subsequent calculations. Point Cloud Preprocessing generates a raw point cloud from the depth map and the color image, deletes the outliers, calculates the normals for the point cloud, and in the end extracts the part of the point cloud within the ROI.

For details about this Procedure, please see *Point Cloud Preprocessing*.

A sample result of Point Cloud Preprocessing is shown in *Figure 3*.

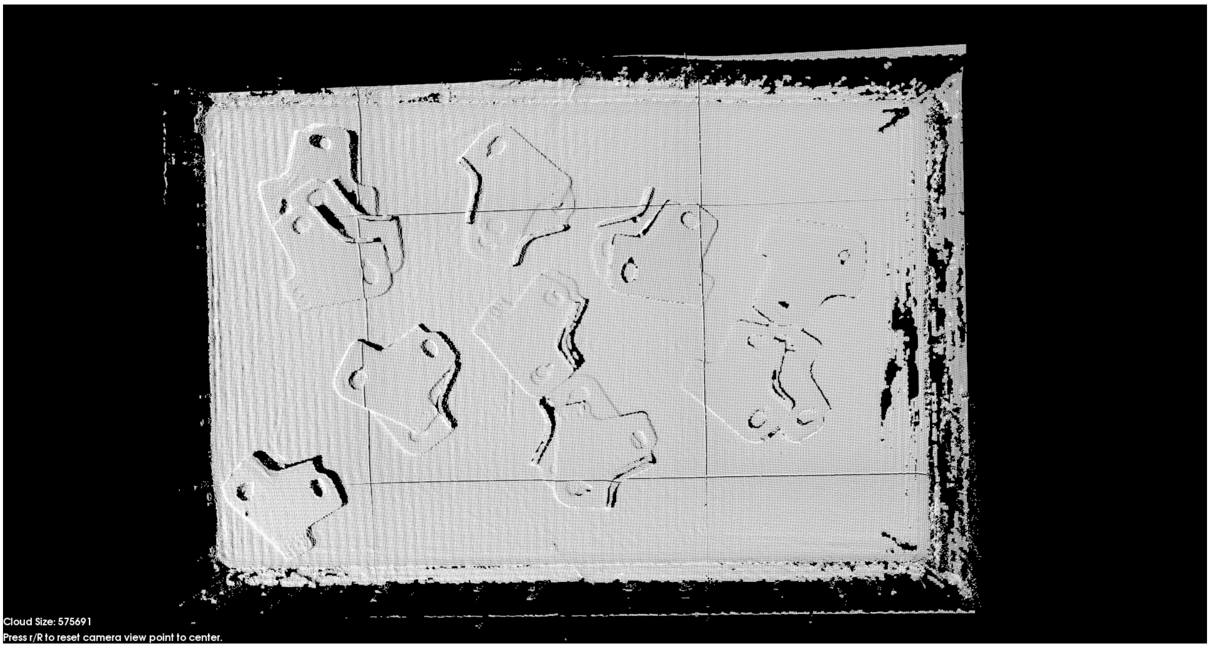


Figure 3. A sample result of Point Cloud Preprocessing

Instance Segmentation (Colored Image)

In this project, instance segmentation by deep learning is used to find the planar projection of the workpieces and generate masks, thus forming the basis for the subsequent task of generating the workpiece point clouds. A sample result is shown in *Figure 4*.

Please see *Instance Segmentation (Colored Image)* for details about this Procedure.

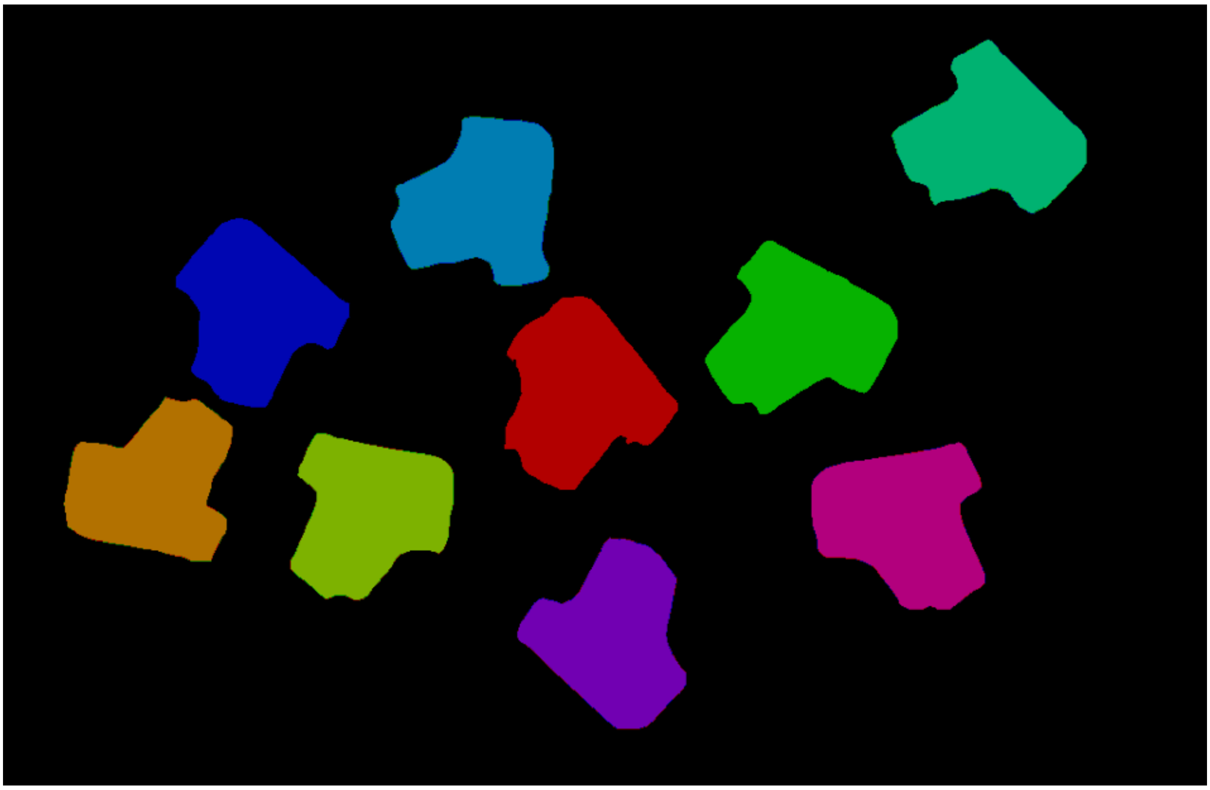


Figure 4. A sample result of Instance Segmentation (Color Image)

3D Boundary Extraction

This Procedure extracts the workpiece edge point clouds after the workpiece point clouds under the masks are obtained. The purpose is to filter out unnecessary point cloud parts to facilitate the subsequent 3D matching.

A sample result is shown in *Figure 5*.

Please see *3D Boundary Extraction* for details about this Procedure.

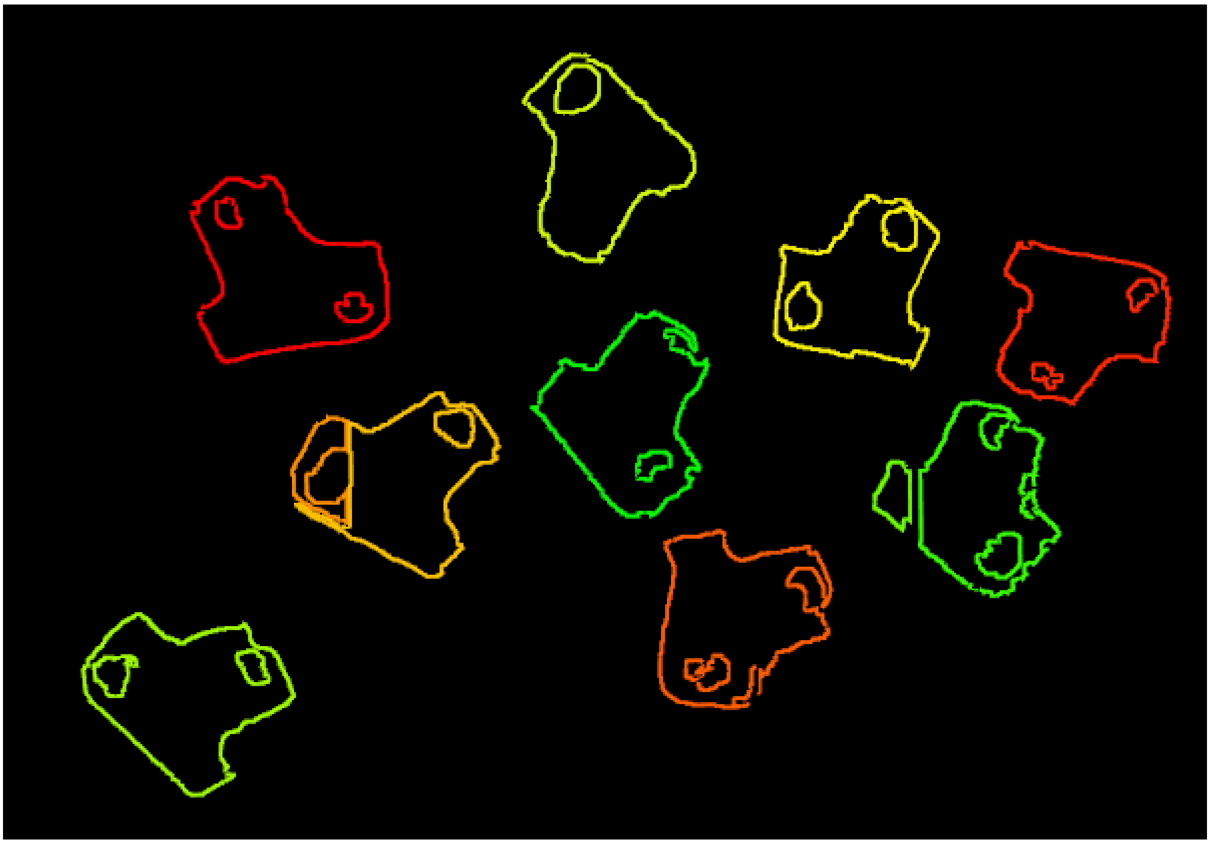


Figure 5. A sample result of 3D Boundary Extraction

Filter Out Point Clouds That Exceed the Limit

After the point clouds of individual workpieces are generated, this Procedure filters out point clouds that have point counts excceding the limit and thus affect 3D matching to improve the accuracy of matching.

A sample result is shown in *Figure 6*.

Please see *Filter Out Point Clouds That Exceed The Limit* for details about this Procedure.

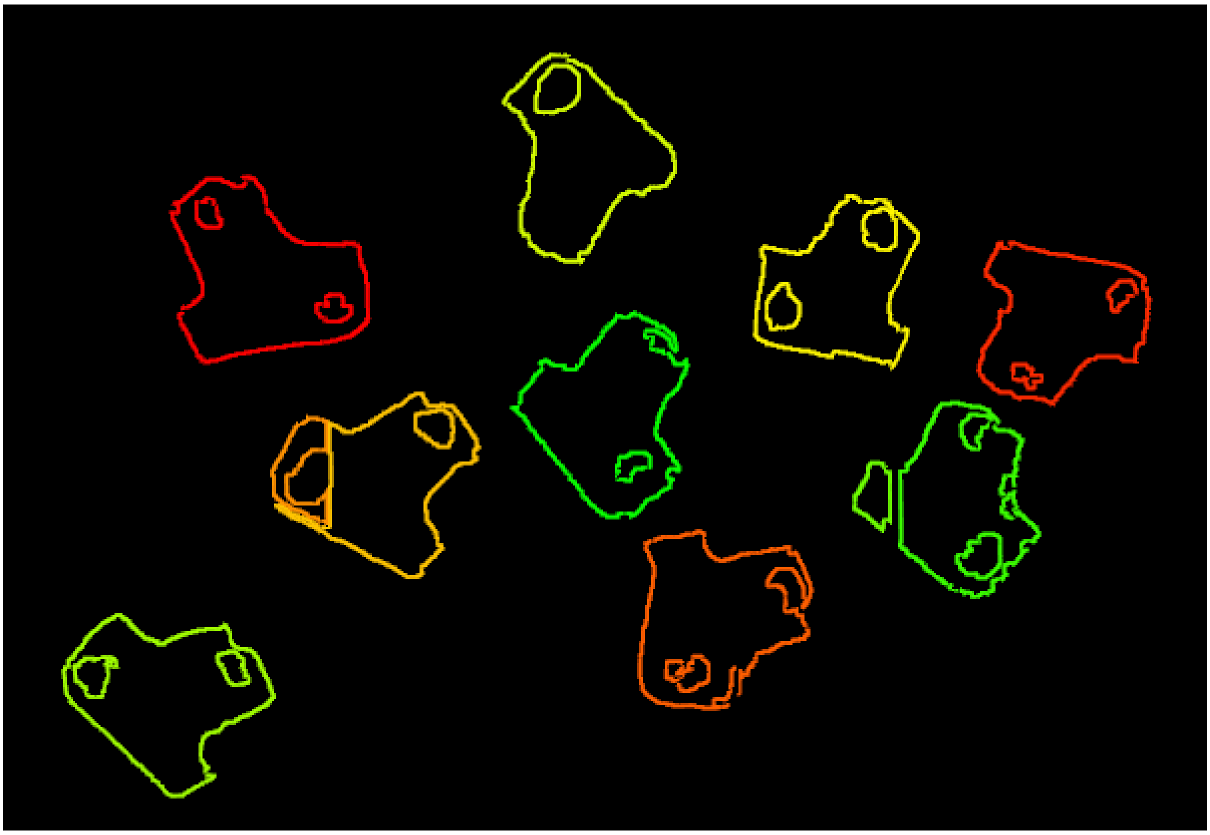


Figure 6. A sample result of Filter Out Point Clouds That Exceed the Limit

3D Matching

3D Matching is performed on the workpiece point clouds to obtain pick points after the point clouds are filtered.

A sample result of 3D Matching is shown in *Figure 7*.

Please see *3D Matching* for details about this Procedure.

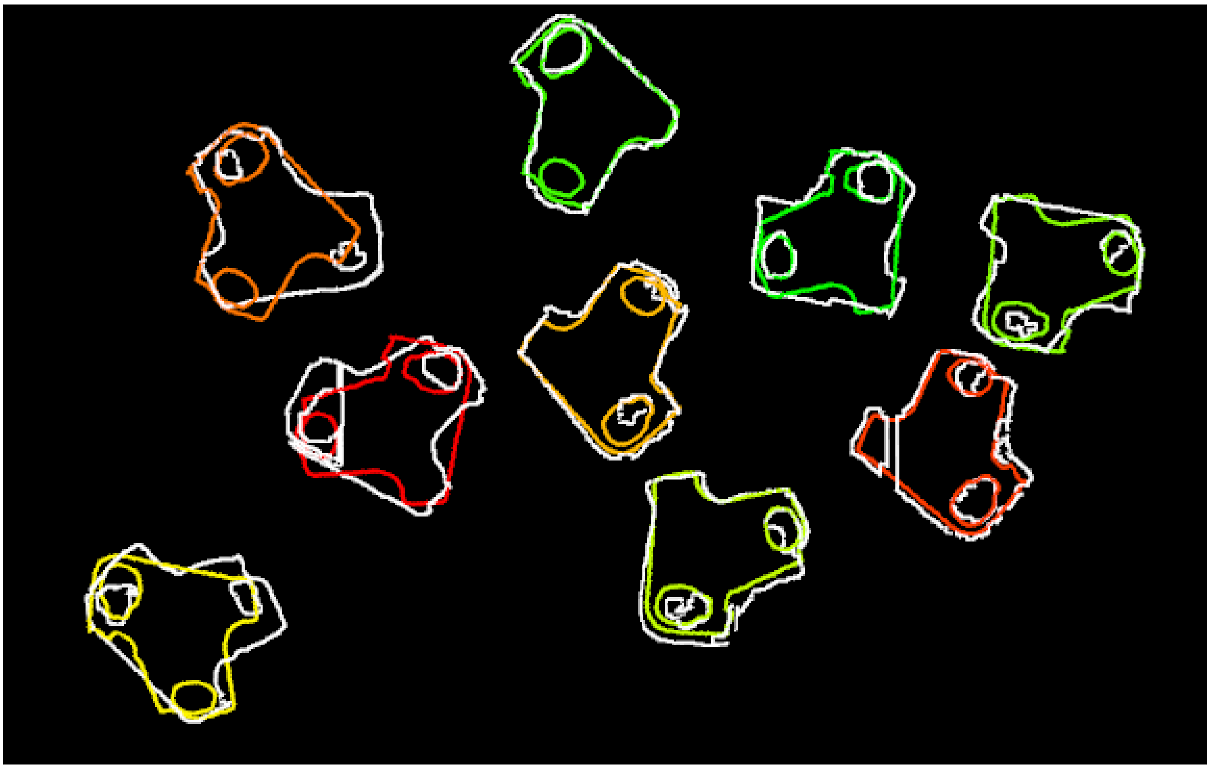


Figure 7. A sample result of 3D Matching

Pose Adjustment Collection

This Step is for adjusting the poses.

Please see *Pose Editor* for instructions on adjusting poses.

Map to Multi Pick Points

This Step takes a sorted pose list and maps them to a workpiece to provide different potential picking poses for the robot.

Please see *Map to Multi Pick Points* for details about this Step.

9.13 Piece Picking (with Bin)

This Typical Project is applicable to scenarios where objects are placed in bins.

9.13.1 Mech-Vision Project Workflow

The target objects are placed in a bin, so the picking task can be more easily completed by obtaining the relative positions of the target objects to the bin.

Therefore, the pose of the bin should be obtained based on the highest layer of the point cloud, i.e., the part of the point cloud for the bin' s upper edge.

Then the positions and grasp poses of the target objects should be obtained.

Since the target objects' shapes differ greatly and no universally applicable model is available, grasp pose estimation is done by deep learning.

The workflow of a Piece Picking Typical Project is shown in *Figure 1*.

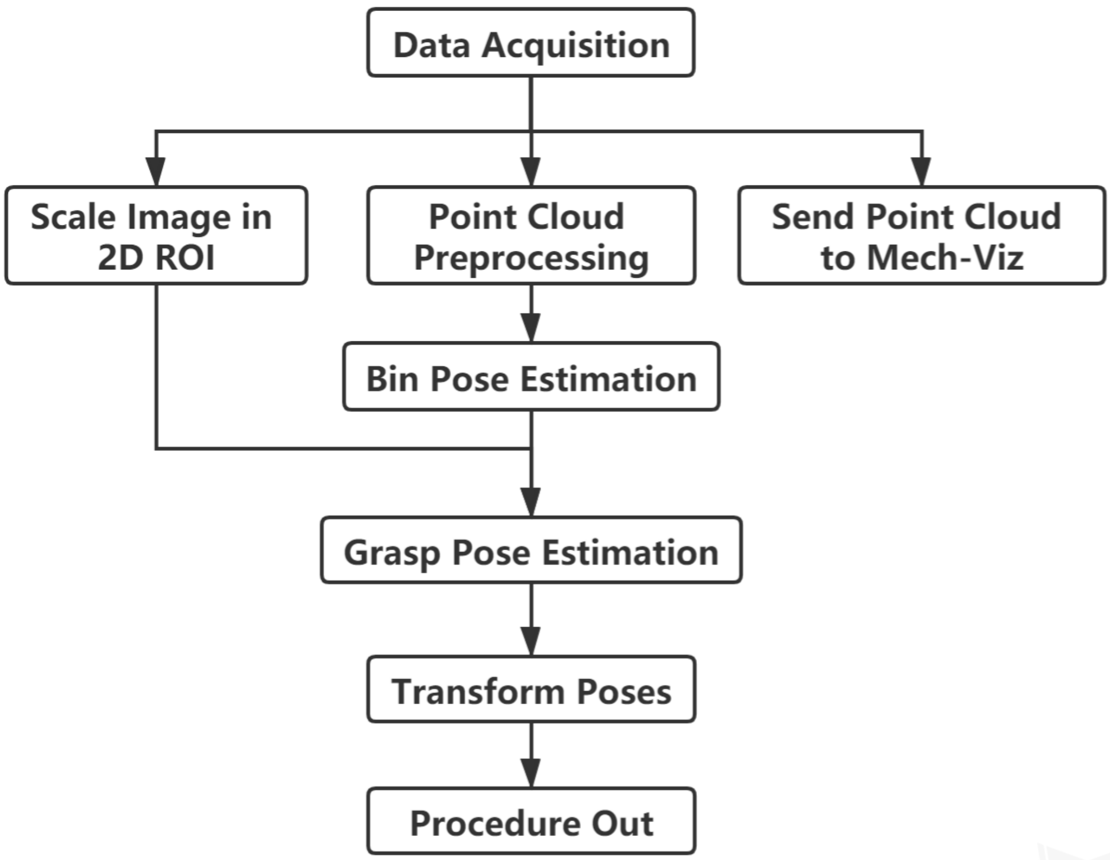


Figure 1. The workflow of a Piece Picking Typical Project

Figure 2 is a screenshot of the graphical programming of the project.

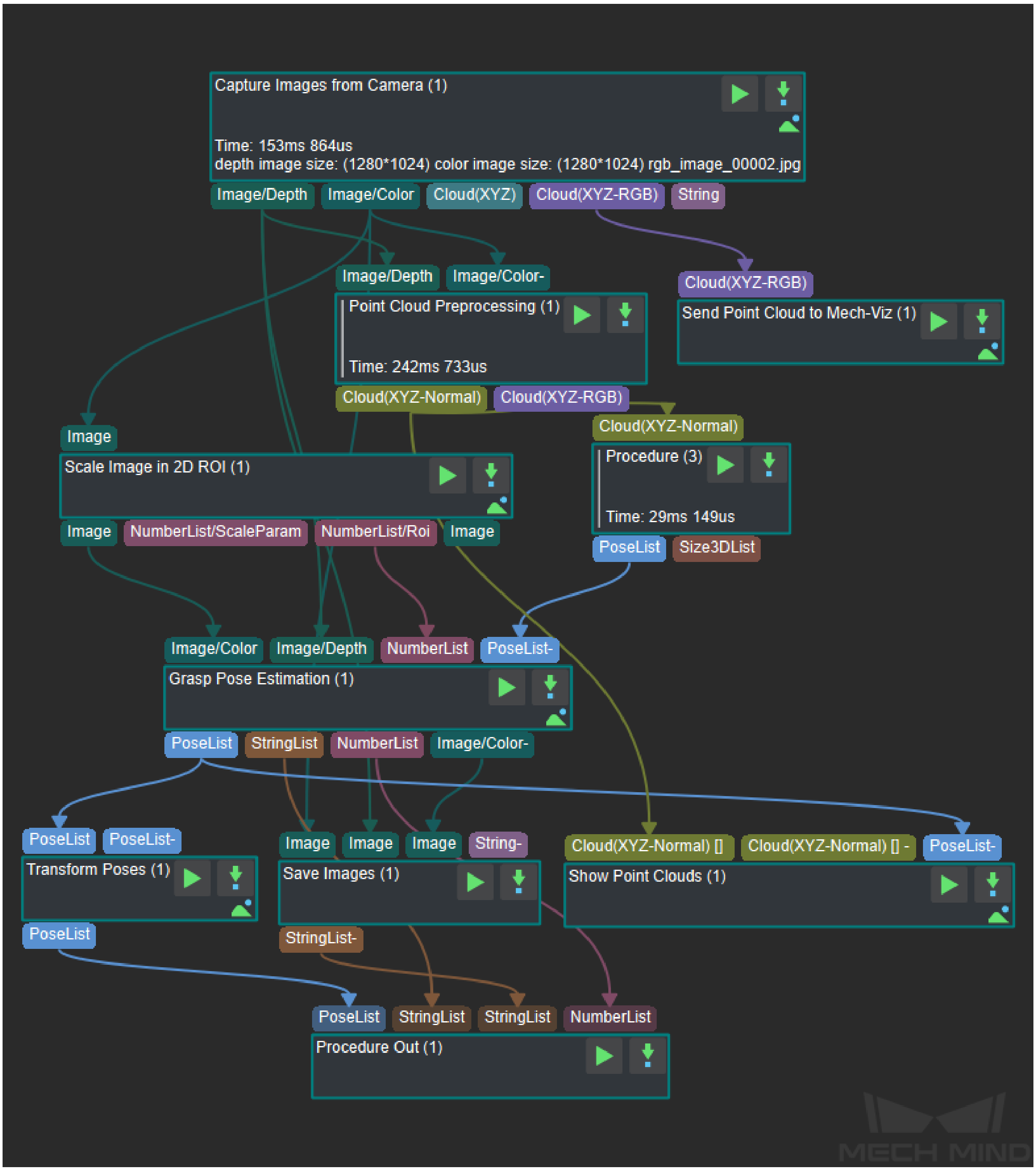


Figure 2. The graphical programming of a Piece Picking Typical Project

9.13.2 Steps and Procedures

A Procedure is a functional program block that consists of more than one Step.

Capture Images from Camera

This Step acquires the color image, depth map, and point cloud from the camera as the input data for subsequent vision calculations.

For details about this Step, please see *Capture Images from Camera*.

Point Cloud Preprocessing

This Procedure facilitates and shortens the processing time for the subsequent calculations. Point Cloud Preprocessing generates a raw point cloud from the depth map and the color image, deletes the outliers, calculates the normals for the point cloud, and in the end extracts the part of the point cloud within the ROI.

For details about this Procedure, please see *Point Cloud Preprocessing*.

A sample result of Point Cloud Preprocessing is shown in *Figure 3*.

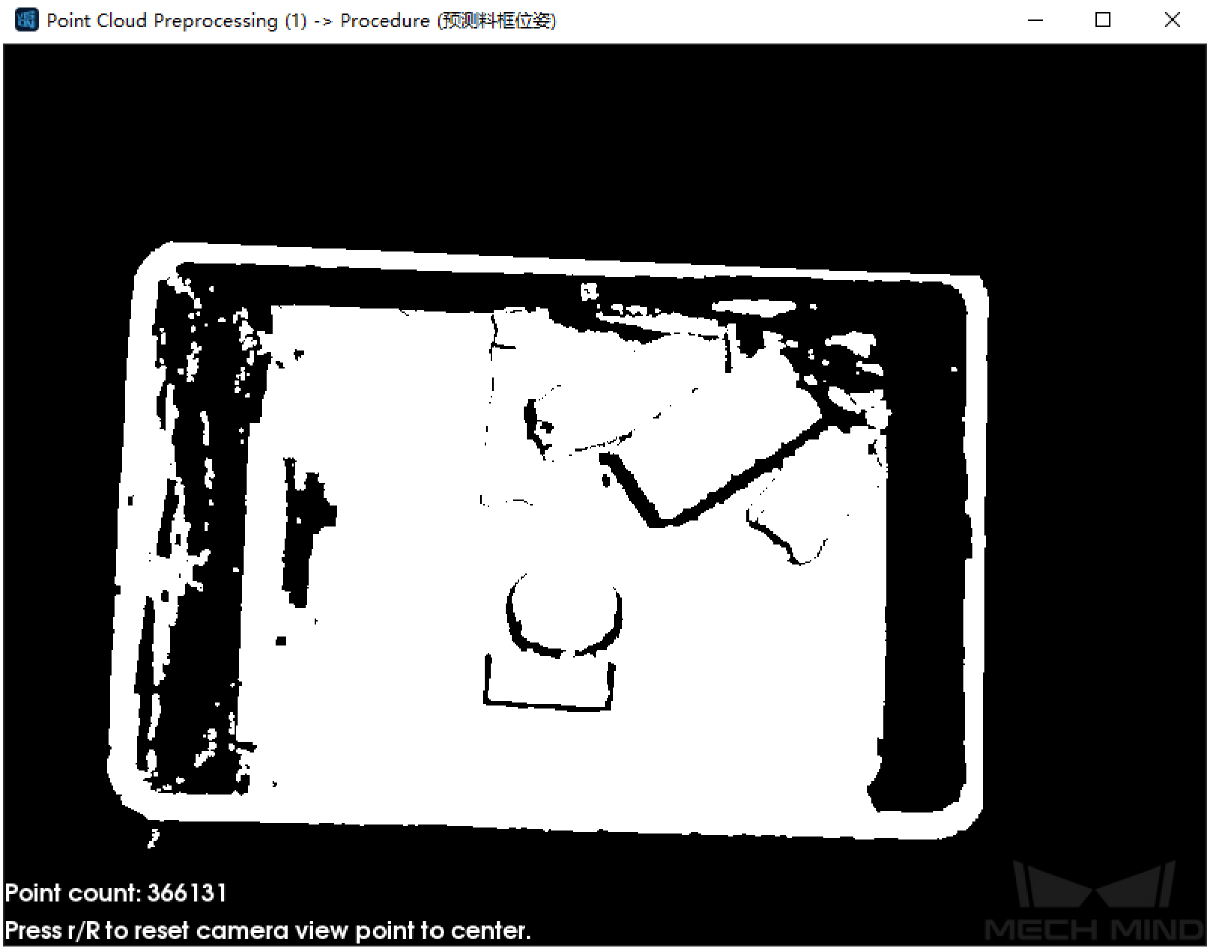


Figure 3. A sample result of Point Cloud Preprocessing

Scale Image in 2D ROI

This Step scales the ROI of the image to a designated size. The result of object pose estimation is largely dependent on the ROI and the scaling ratio; so the parameters of this Step must be adjusted for ideal pose estimation results.

For detailed instructions on adjusting the parameters of this Step, please see *Deep Learning Deployment ROI Settings*.

Bin Pose Estimation

Bin Pose Estimation is Procedure (3) in *Figure 2*.

This procedure estimates the pose of the bin by obtaining the point cloud of the bin’s upper edge. The point cloud is sampled to reduce its size and then clustered. Next, the point cloud of the highest layer, i.e., the point cloud part of the bin’s upper edge, is obtained, the pose and dimensions of the obtained point cloud calculated, and finally, the bin pose is output after flipping the Z-axis.

The bin pose is shown in *Figure 4*.

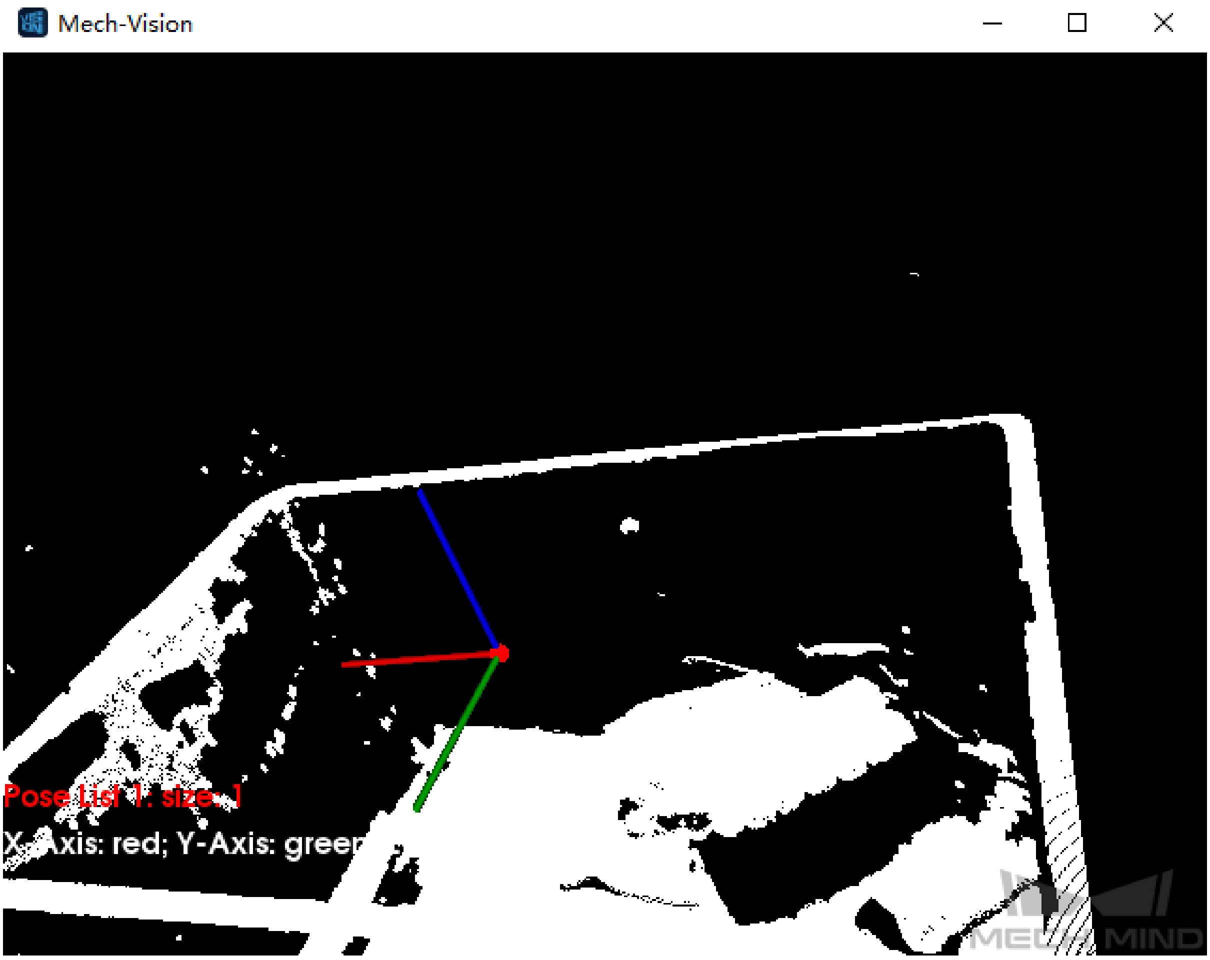


Figure 4. Bin Pose Estimation

Grasp Pose Estimation

Because the target objects' shapes may differ greatly, grasp pose estimation is done by deep learning.

Deep learning model file and configuration file, pose estimation model file, and overlap detection model file and configuration file obtained after deep learning model training should be loaded before running the Step.

Given the color image, depth image, ROI parameters, and bin pose, the deep learning algorithm will generate the following:

- The pose of an object's pickable surface under the camera coordinate system (as shown in *Figure 5*).
- The 3D dimensions of the pickable object.
- The suction cup label corresponding to the object's pickable surface.

Note: The purpose of inputting bin pose is to more accurately determine the position of an object relative to the bin.

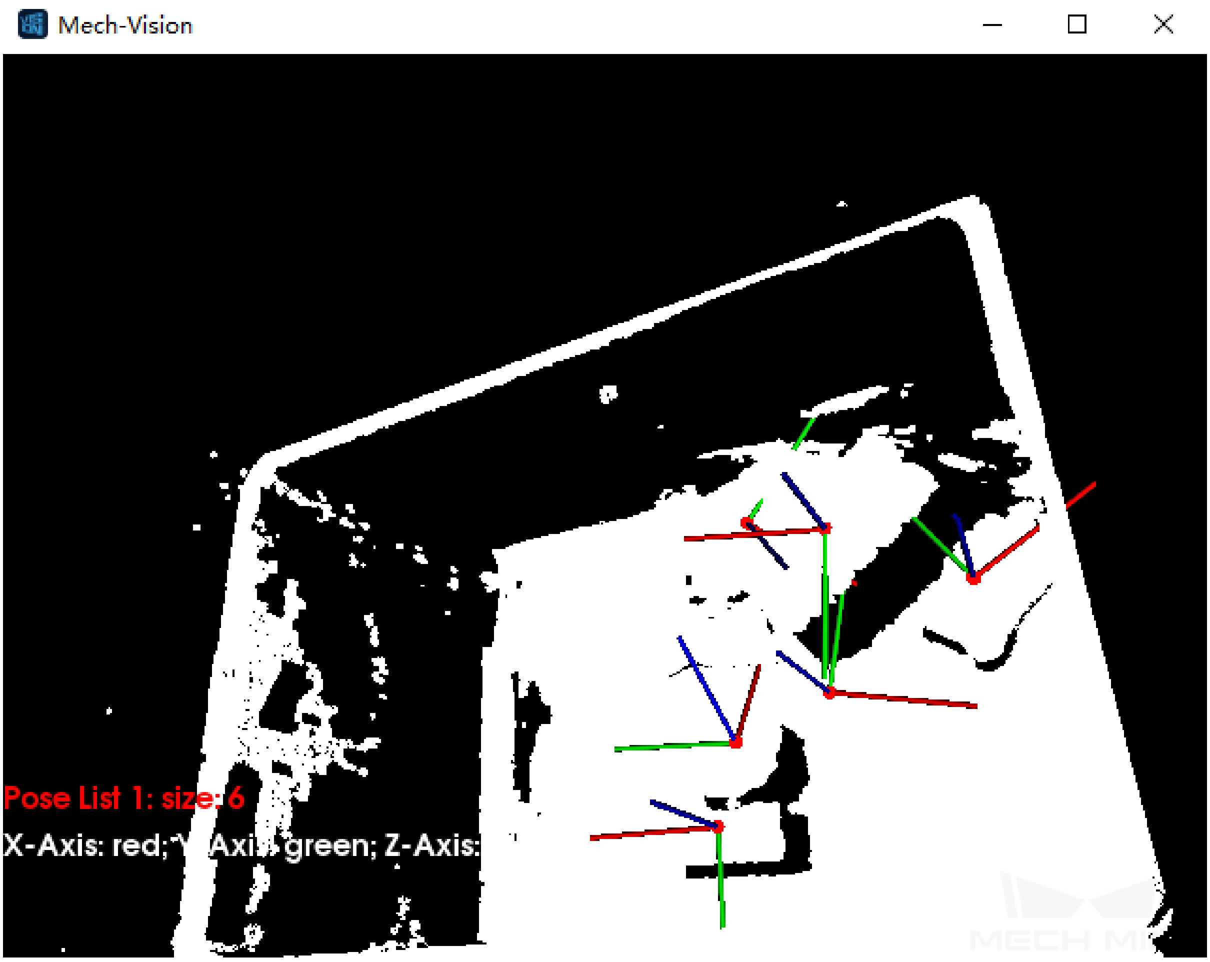


Figure 5. A sample result of Grasp Pose Estimation

This Step can also display a color image with region labels in real-time. A color image displaying pickable regions labeled with suction-cup sizes is shown in *Figure 6*. The type of label displayed can be changed in the Visualization section of parameters.

Note: Enabling visualization will slow down the running of the project.

Figure 6. Visualization option: show suction cup size

Transform Poses & Procedure Out

Transform Poses transforms the list of poses from the camera coordinate system to the robot coordinate system. For details about Transform Poses, please see *Transform Poses*.

Then, **Procedure Out** sends the list of poses under the robot coordinate system to the server. For details about Procedure Out, please see *Procedure Out*.

9.14 Piece Picking (without Bin)

This Typical Project applies to scenarios where objects are not placed in bins.

9.14.1 Mech-Vision Project Workflow

Since the target objects' shapes differ greatly and no universally applicable model is available, grasp pose estimation is done by deep learning.

The workflow of a Piece Picking (without Bin) Typical Project is shown in *Figure 1*.

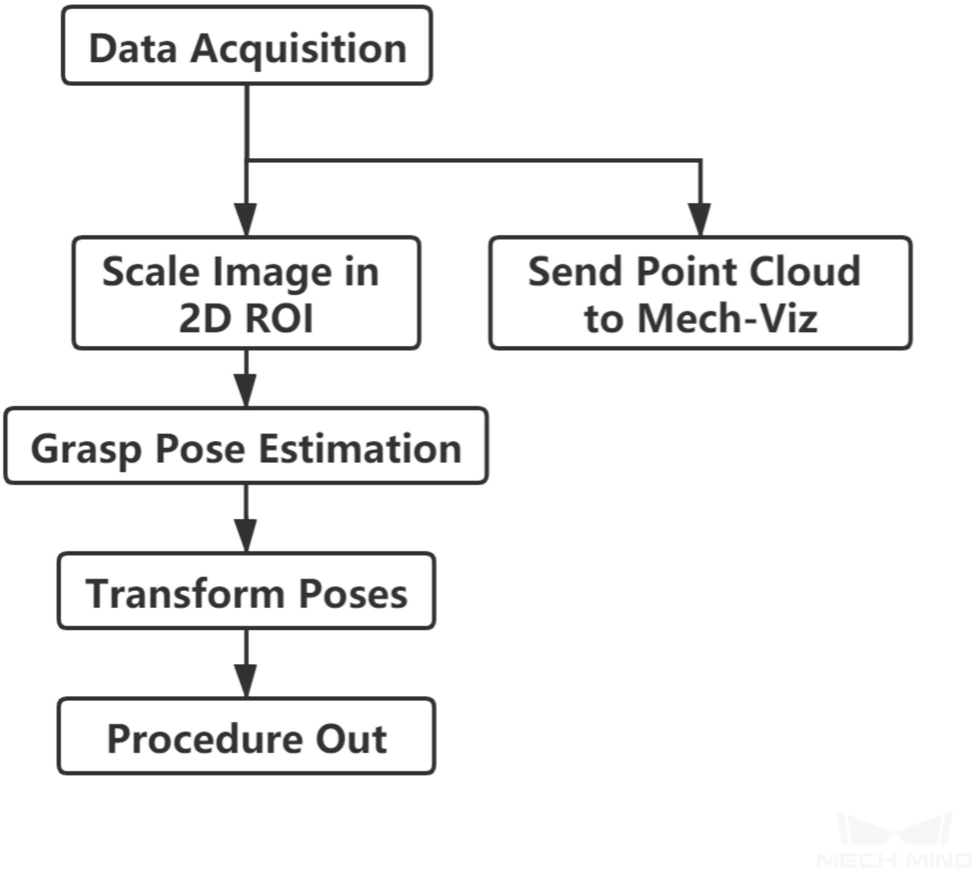


Figure 1. The Workflow of a Piece Picking (without Bin) Typical Project

Figure 2 is a screenshot of the graphical programming of the project.

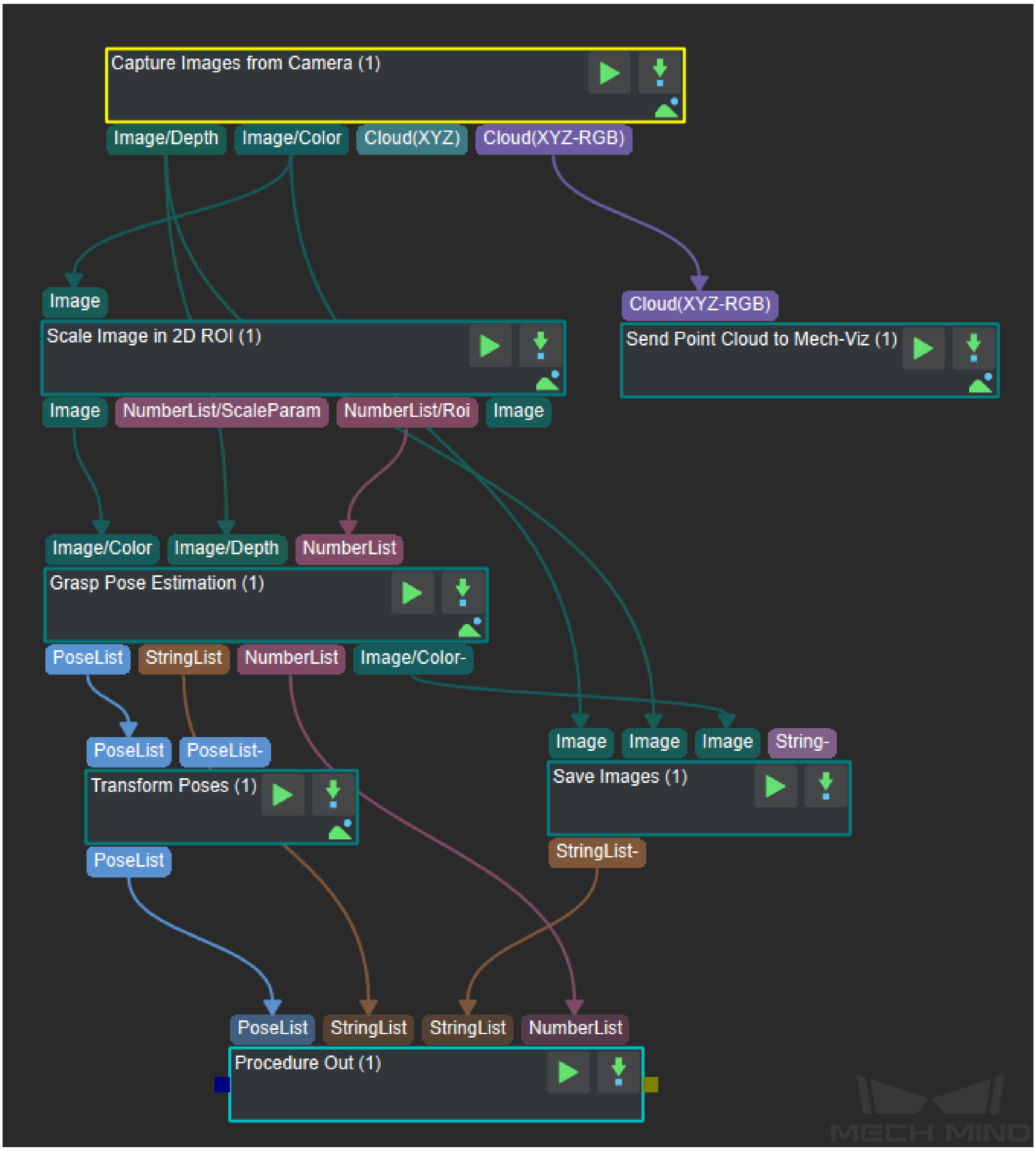


Figure 2. The graphical programming of a Piece Picking (without Bin) Typical Project

9.14.2 Steps and Procedures

A Procedure is a functional program block that consists of more than one Step.

Capture Images from Camera

This Step acquires the color image, depth map, and point cloud from the camera as the input data for subsequent vision calculations.

For details about this Step, please see *Capture Images from Camera*.

Scale Image in 2D ROI

This Step scales the ROI of the image to a designated size. The result of object pose estimation is largely dependent on the ROI and the scaling ratio; so the parameters of this Step must be adjusted for ideal pose estimation results.

For detailed instructions on adjusting the parameters of this Step, please see *Deep Learning Deployment ROI Settings*.

Grasp Pose Estimation

Because the target objects' shapes may differ greatly, grasp pose estimation is done by deep learning.

Deep learning model file and configuration file, pose estimation model file, and overlap detection model file and configuration file obtained after deep learning model training should be loaded before running the Step.

Given the color image, depth image, ROI parameters and bin pose, the deep learning algorithm will generate the following:

- The pose of the object's pickable surface under the camera coordinate system (as shown in *Figure 3*).
- The 3D dimensions of the pickable objects.
- The suction cup labels corresponding to the object's pickable surface.

Figure 3. A sample result of grasp pose estimation

This Step can also display a color image with region labels in real-time. A color image displaying pickable regions labeled with suction cup sizes is shown in *Figure 4*. The type of label displayed can be changed in the Visualization section of the parameter panel.

Note: Enabling visualization will slow down the running of the project.

Figure 4. Visualization option: show suction cup size

Transform Poses & Procedure Out

Transform Poses transforms the list of poses from the camera coordinate system to the robot coordinate system. For details about Transform Poses, please see *Transform Poses*.

Then, **Procedure Out** sends the list of poses under the robot coordinate system to the server. Please see *Procedure Out* for details.

9.15 High Precision Positioning

9.15.1 Application Scenario of High Precision Positioning

In automotive assembly, many processes require high position accuracy, including wheel hub locating, tire tightening, and gearbox assembly, etc.

Traditional automated production processes cannot provide robots with accurate location information of objects and require human intervention, thus having a low efficiency.

To solve this problem, Mech-Mind Robotics proposed a 3D vision positioning technology in which the visual recognition results and model files are matched multiple times to accurately obtain the 3D pose information of objects, thus improving the efficiency of high precision assembly processes.

9.15.2 Mech-Vision Project Workflow

The workflow of a Typical Project for High Precision Positioning is shown in *Figure 1*.

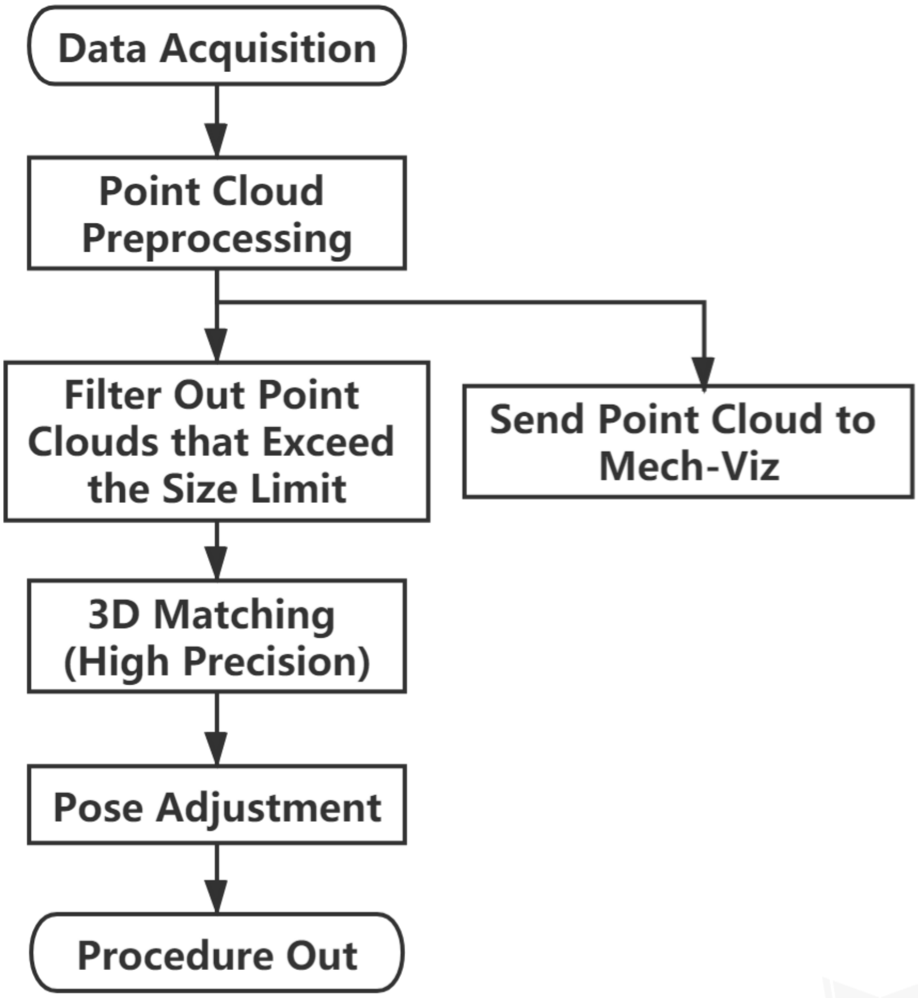


Figure 1. The workflow of High Precision Positioning

The graphical programming of this project is shown in *Figure 2*.

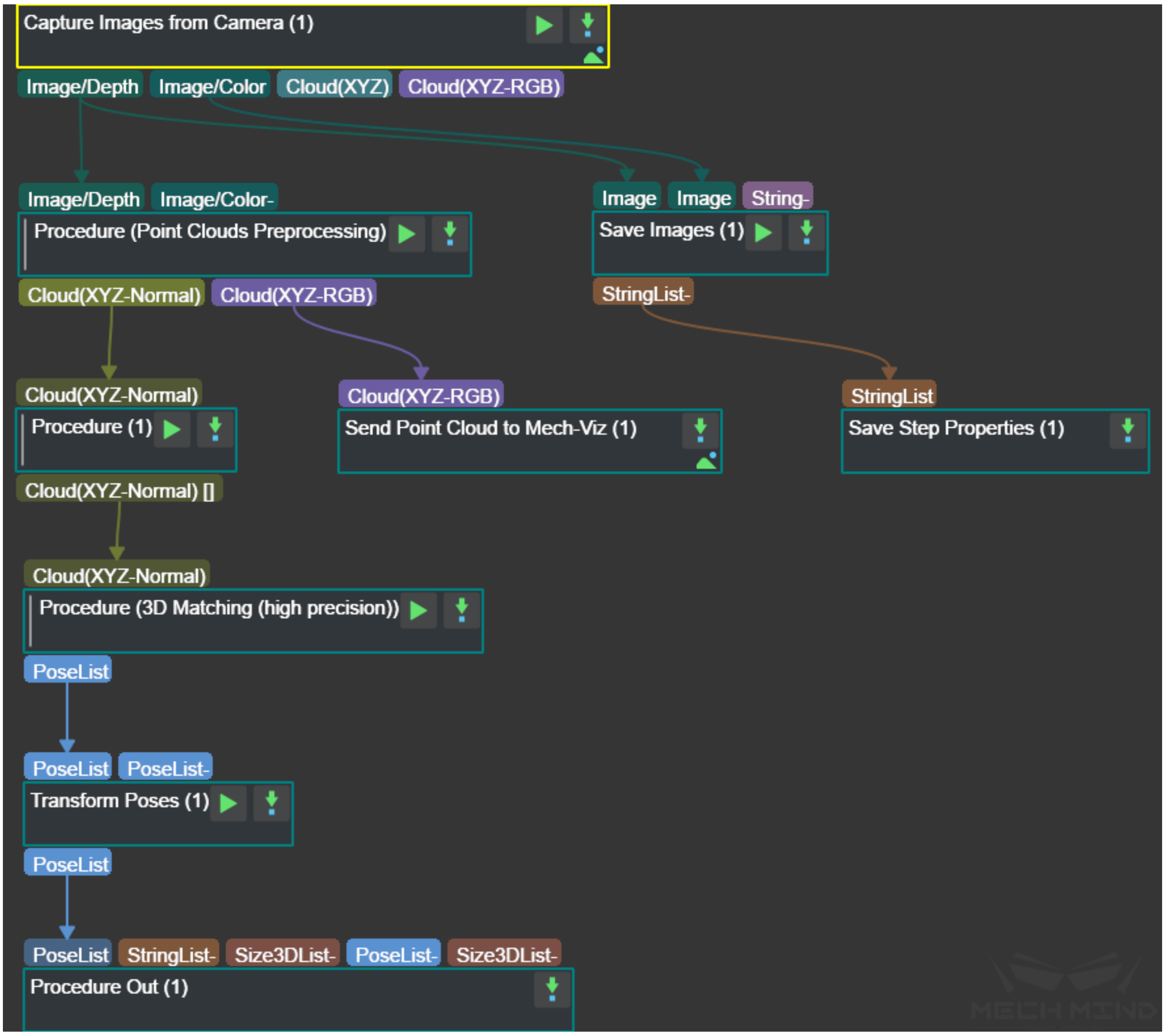


Figure 2. The graphical programming of a Typical Project for High Precision Positioning

9.15.3 Steps and Procedures

A Procedure is a functional program block that consists of more than one Step.

Capture Images from Camera

This Step obtains the color images and depth maps of the scene from the camera and provides data for the subsequent visual calculation.

Please see *Capture Images from Camera* for details about this Step.

Point Cloud Preprocessing

This Procedure facilitates and shortens the processing time for the subsequent calculations. Point Cloud Preprocessing generates a raw point cloud from the depth map and the color image, deletes the outliers, calculates the normals for the point cloud, and in the end extracts the part of the point cloud within the ROI.

For details about this Procedure, please see *Point Cloud Preprocessing*.

A sample result of Point Cloud Preprocessing is shown in *Figure 3*. The original point cloud is on the left and the preprocessed point cloud is on the right.

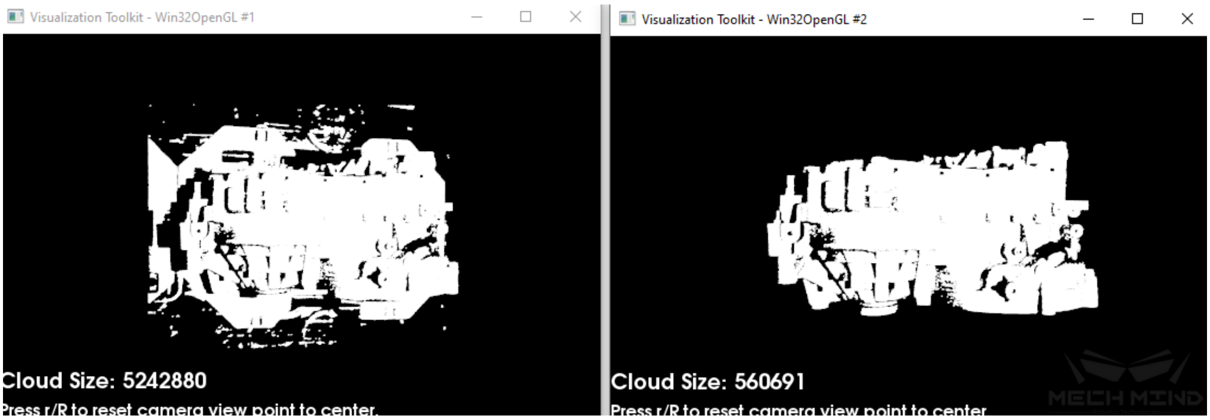


Figure 3. A sample point cloud before and after Point Cloud Preprocessing

Filter Out Point Clouds That Exceed the Limit

This Procedure filters out point clouds that affect 3D matching to improve the accuracy of matching.

Please see *Filter Out Point Clouds That Exceed The Limit* for details about this Procedure.

3D Matching (High Precision)

In this Procedure, 3D Coarse Matching is followed by two rounds of 3D Fine Matching to obtain a list of finely calculated poses.

Please see *3D Matching (High Precision)* for details about this Procedure.

The figures below show sample matching results.

The left figures are the front views and the right figures are the side views. In the point clouds, the red point clouds are the models that are matched with the object point clouds in white.

The coarse matching result is shown in *Figure 4*. If the model and point cloud do not well overlap, the matching has a relatively large error. The final result after two rounds of fine matching is shown in *Figure 5*.

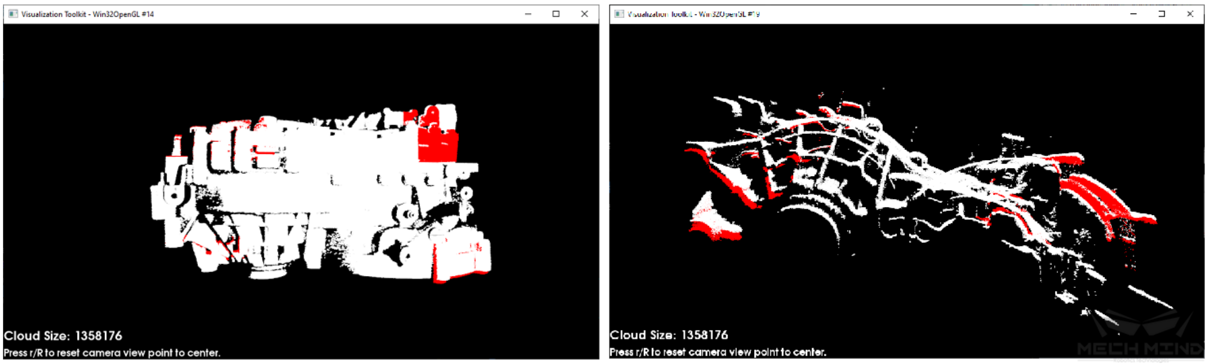


Figure 4. A sample result of 3D Coarse Matching

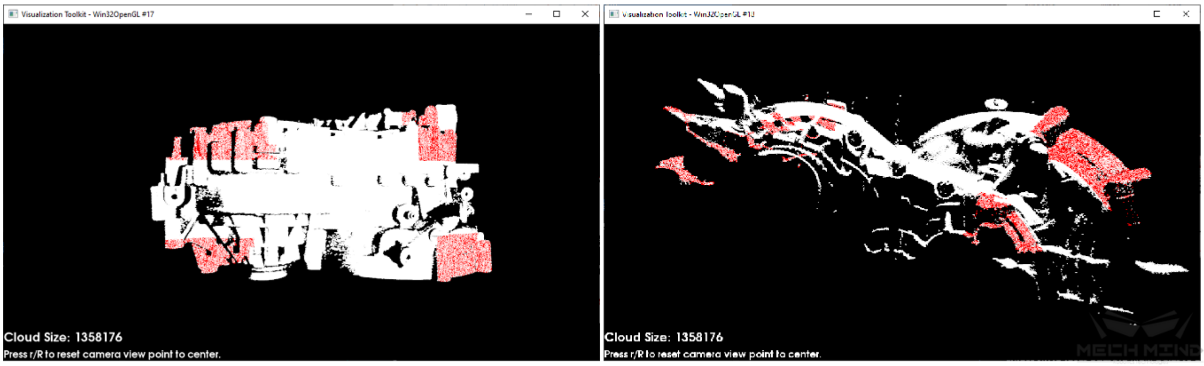


Figure 5. A sample result of 3D Matching (High Precision)

Pose Adjustment Collection

This Step is for adjusting the poses.

Please see *Pose Editor* for instructions on adjusting poses.

CHAPTER

TEN

FAQ

10.1 Common problems about project

10.1.1 Description for common problems about project

This chapter mainly list the common setup issues in the actual use of the software, and provide the corresponding solutions.

10.1.2 Missing of point cloud

The most common problem in the actual use is the missing of points. To solve this problem, it is necessary to understand the process of the point cloud from the generation of point cloud to the final deployment. The process is shown in Figure 1.1, wherein the red part is processed in the camera and the yellow part is processed in Mech-Vision. The missing of point cloud may happen in any step in this process.

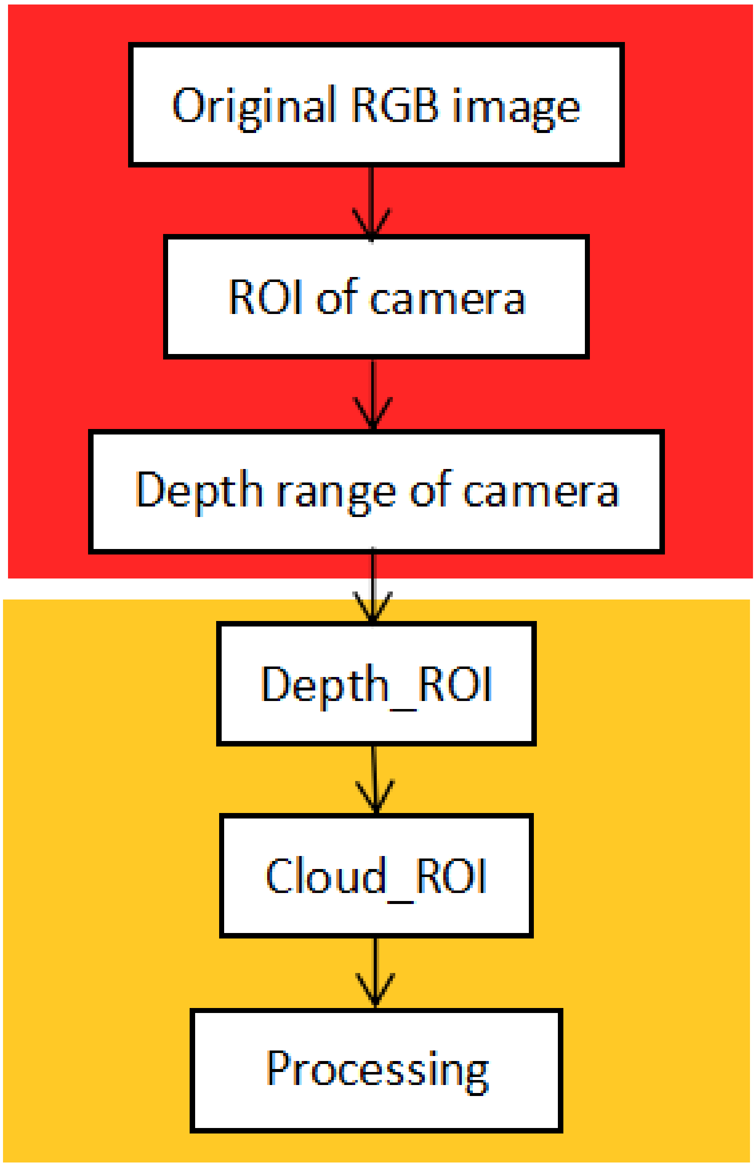


Figure 1.1 Process for getting point cloud

First, it is necessary to confirm whether the point clouds in Mech-Eye Viewer are already missing. If there are, check the config settings in Mech-Eye Viewer. If there is no missing, check whether *removeBackground* is checked in *Capture Images from Camera* or *From Depth Map to Point Cloud*. If it is checked, check *VisualizeOutput* for the corresponding Step.

After running the single Step, the result is shown in Figure 1.2, where the red area is the removed point cloud. If it still doesn' t work well, uncheck the *removeBackground* or reset the background in *Set Static Background for Project*. If this parameter is uncheck , further check the point cloud output by *Capture Images from Camera* or *From Depth Map to Point Cloud* in the point cloud display Step. If the point cloud is missing, it is necessary to reset *2D ROI* (depth_roi), if the point cloud is not missing, then further check the results of *Extract 3D Points in 3D ROI* or Set Zeros for Depth Pixels Oit of ROI. The result of *Extract 3D Points in 3D ROI* is shown in Figure 1.3, wherein the white point cloud is the reserved point cloud, and the red is the removed point cloud. If it is not suitable, reset *3D ROI*. The results of Step Set Zero for Depth Pixels out of ROI are shown in Figure 1.4, wherein the left image is the original depth image and the right image is the processed one.

It should be noted that if it is necessary to update the settings of this step, *Extract 3D Points in 3D ROI* is required for the project, otherwise the settings cannot be performed.

The above order for checking is not absolute, it can be adjusted flexibly according to the actual situation.

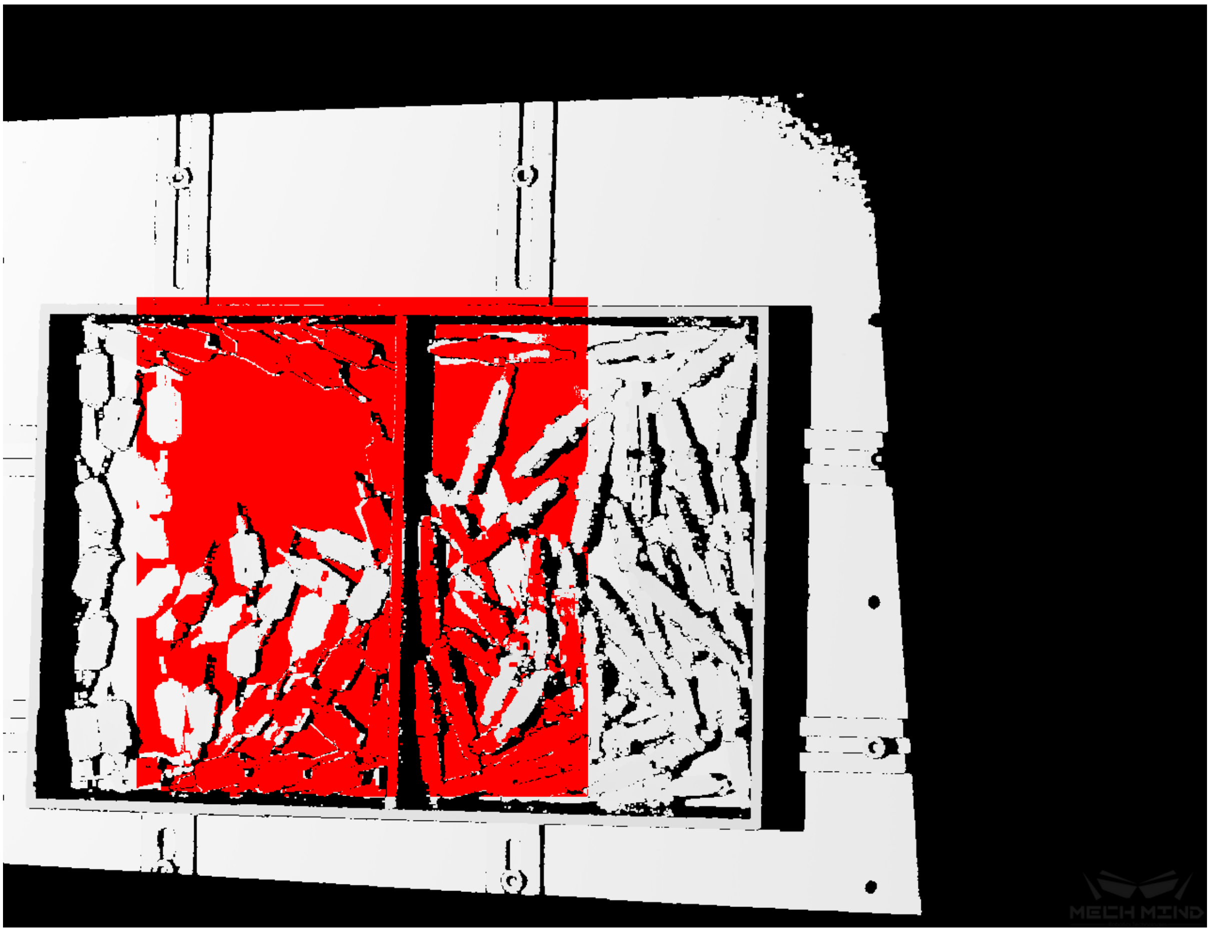


Figure 1.2 Results after removing background

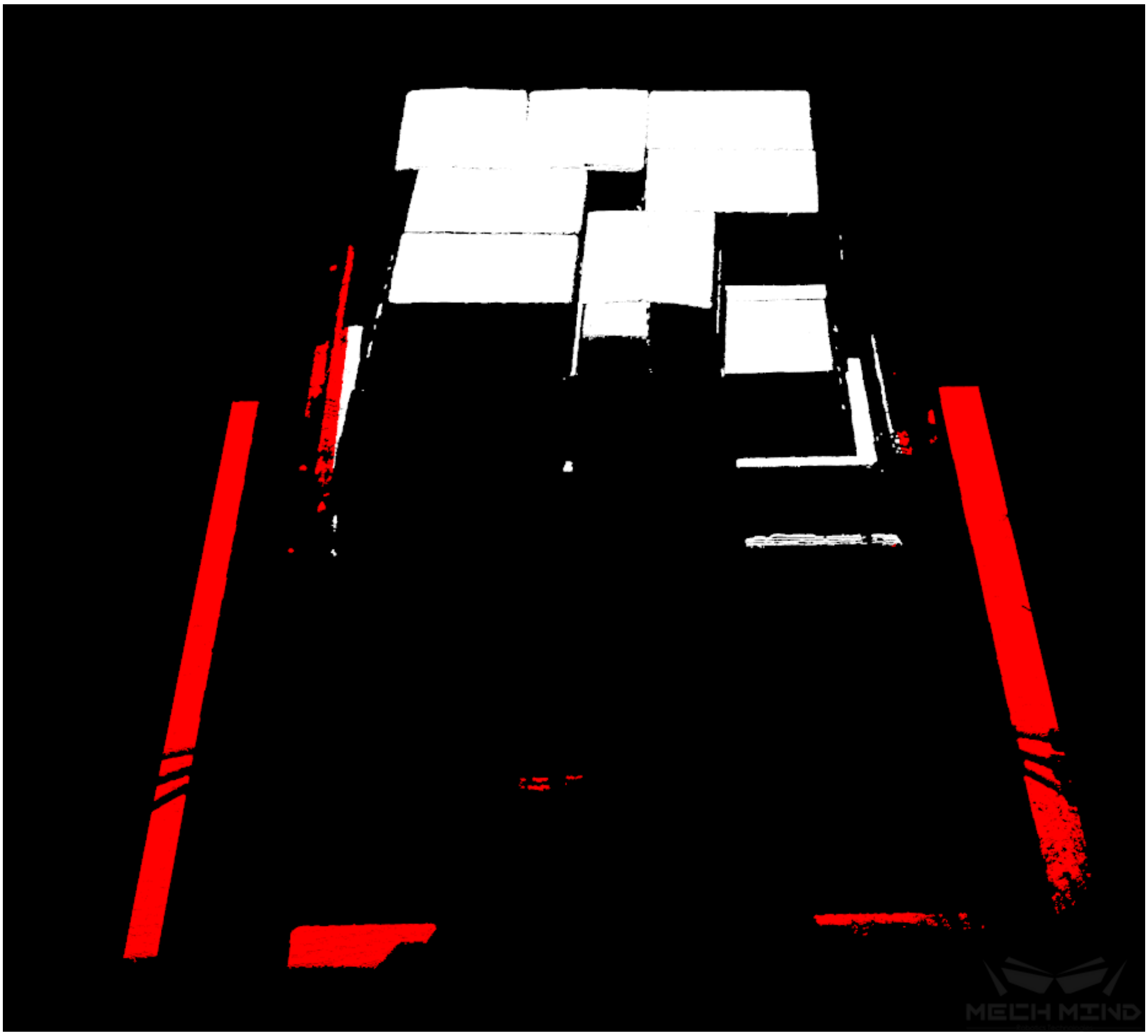


Figure 1.3 Results after setting “Cloud Roi”

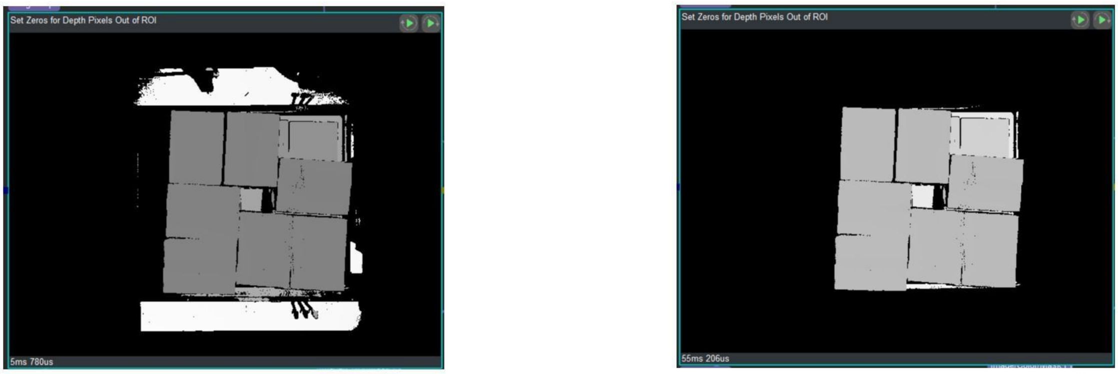


Figure 1.4 Result after setting Zeros for "Depth Pixels Out of ROI"

10.1.3 Cannot find Step or the parameters of Step

Some parameters or Step may not be found during debugging and implementation on site, this is because by default only the frequently used Steps and parameters are displayed. Uncheck these two options *List Only Frequently Used Steps* and *List Only Frequently Used Properties* to display all, as shown in Figure 1.5.

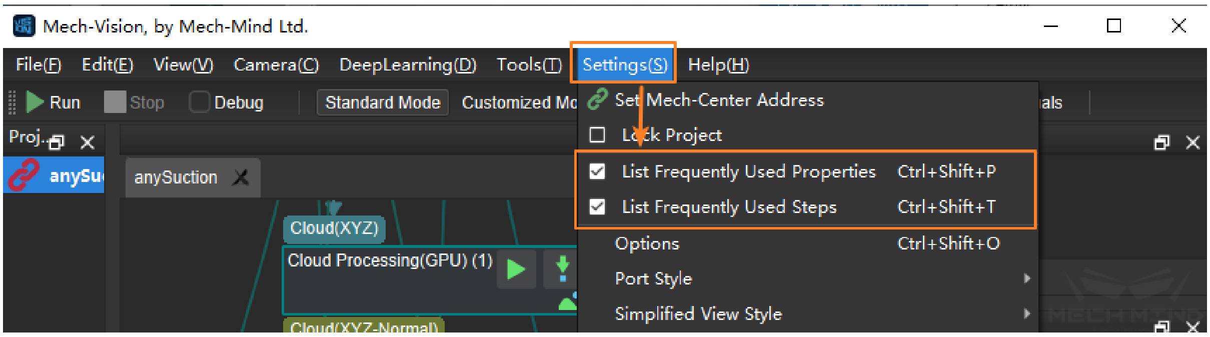


Figure 1.5 Display option for Frequently used parameters and Steps

10.1.4 Files are not updated

In order to avoid the repeated loading of files, currently Step only loads the files for the first time or when the file path changes. If Step has already loaded the file and the content is updated without updating the file name, then the updated content will not be loaded. Check option *ReloadFile* in the menu bar, as shown in Figure 1.6, and then running the Step that needs to be reloaded or run the entire project directly, and in the end uncheck the *ReloadFile* option.

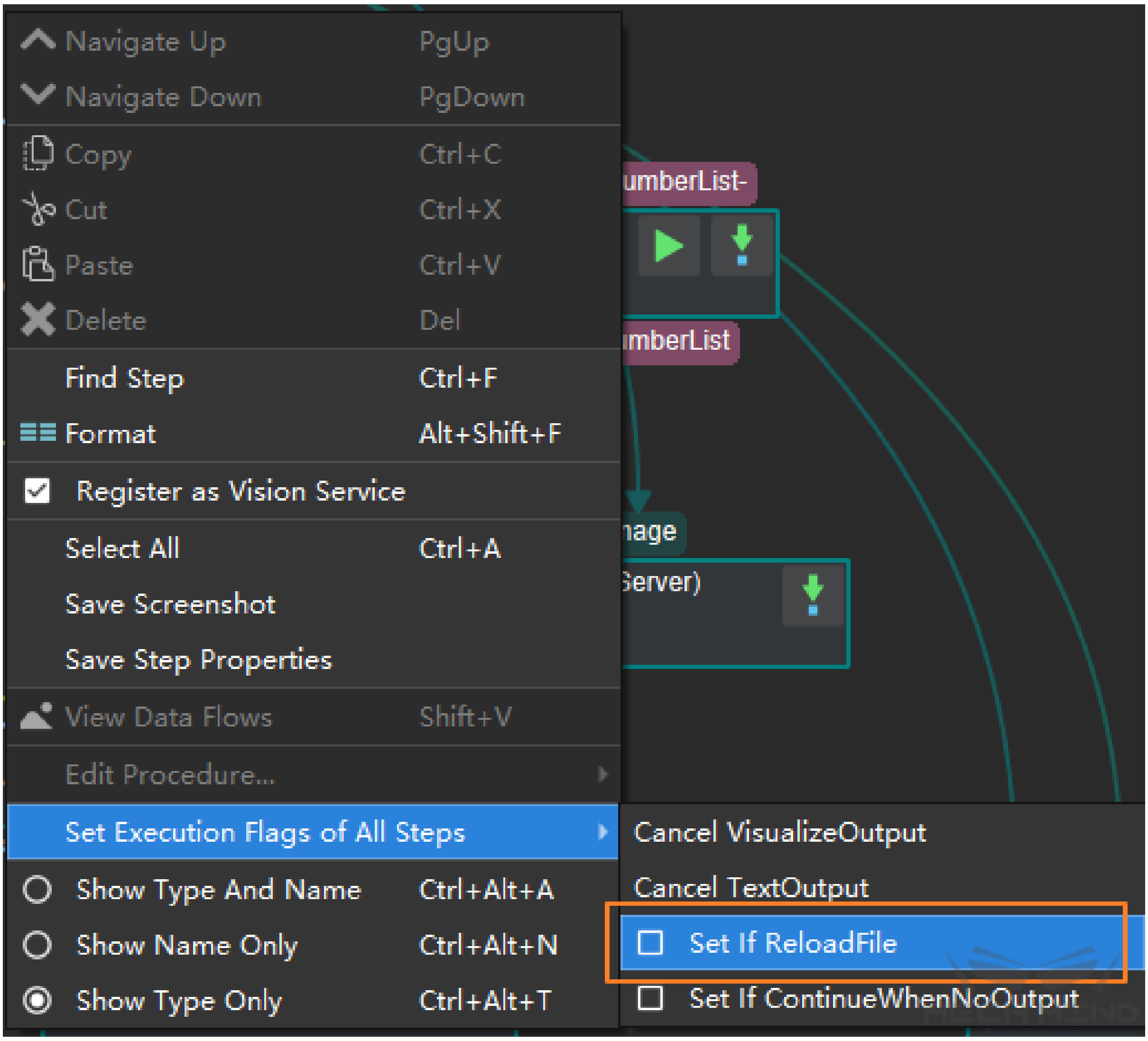


Figure 1.6 Option *ReloadFile* to reload parameter

10.1.5 Template error

Occasionally, the situation that the template cannot be used, mainly because the `Normal` option is not checked when using Meshlab software to save the processed template, so that the saved template has no normal direction and cannot be used. The correct saving process and options are shown in Figure 1.7.

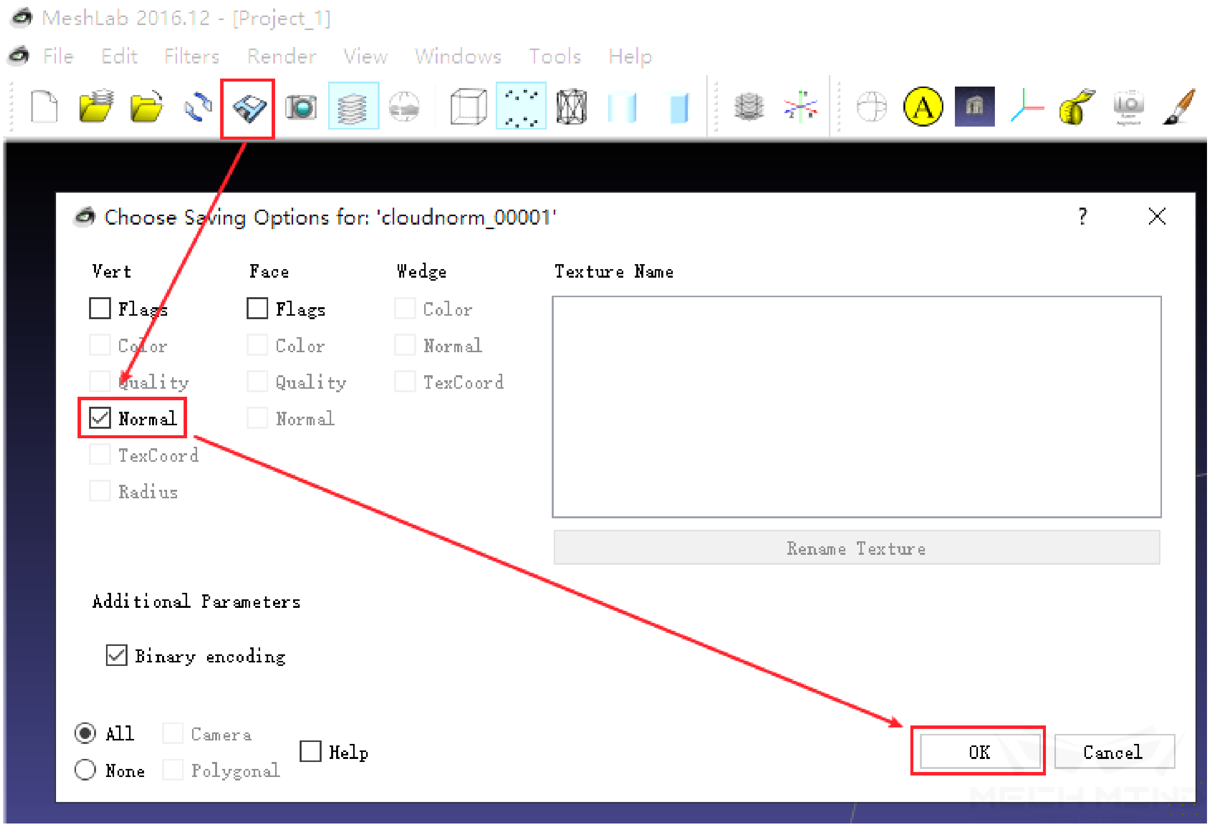


Figure 1.7 Template saving process

10.1.6 Input missing

When the required input of Step is not connected, when running project, the error of missing input will pop up, as shown in Figure 1.8. The error message contains the name of the missing input Step, which can be searched and connected.

Figure 1.8 Process for saving template

10.2 Common problems about vision

10.2.1 Description for common problems about vision

This chapter mainly list the proper methods for recognition of different workpieces and inspection requirements, etc., to improve the recognition accuracy and takt time.

10.2.2 Select features for 3D matching

For a workpiece, such as a sphere, after moving a distance of 1 mm in any direction, the sphere coincides with the original sphere badly. For a plane, after moving 1 mm along the X axis or Y axis, the degree of coincidence between before and after the moving is greater. Therefore the sphere is more restricted than the plane. So for different workpieces, a template can be made according to the degree of restriction, the complete template shall be selected for the more restricted objects, while the edge feature shall be selected for the less restricted objects. Figure 2.1 shows the degree of restriction of the simplified workpiece.

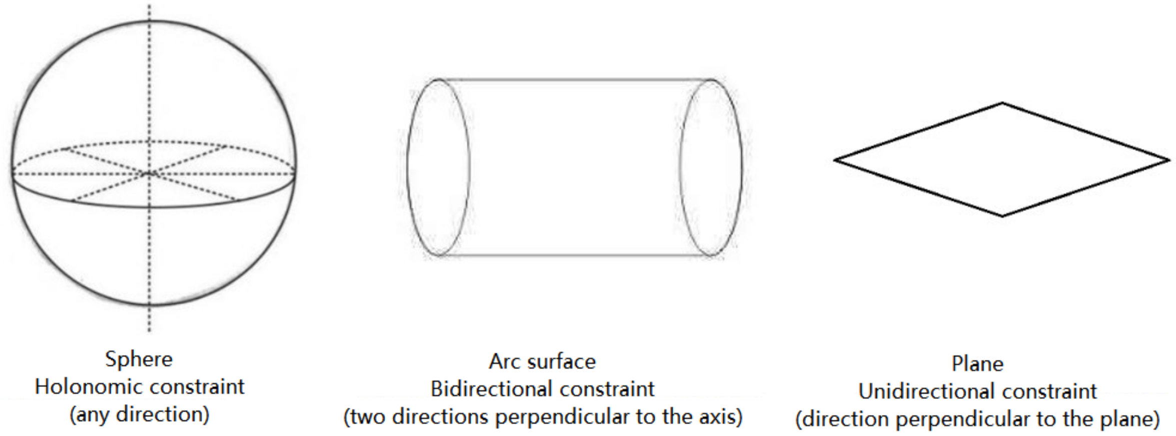


Figure 2.1 Restriction of the simplified workpiece

10.2.3 3D matching for reverse and rotating poses

The shape of the workpiece basically include two types: asymmetry and symmetry. Symmetric workpieces include:

- Axisymmetric (simplified as rectangular);
- Centrosymmetric (simplified as square);
- Continuous symmetry (simplified as circle) etc.

For symmetric workpieces, the matching errors caused by symmetry can be reduced by the following methods:

1> Select the geometric center with symmetry as the grasping point on the point cloud template;

2> Select the axis of symmetry:

- Axisymmetric workpiece: select the coordinate axis along the axial direction as the axis of symmetry, and set the symmetry to 180 degrees in *3D Fine Matching*; as shown in figure 2.2 (left: axis of symmetry; right: parameter settings).
- For Centrosymmetric workpiece: select the coordinate axis along any axis of symmetry as the axis of symmetry, and set the symmetry to 90 degrees in Step (local matching).
- For Continuous symmetric workpiece: select the rotation axis as the symmetry axis, and set the symmetry to any angle in *3D Fine Matching* . If the angle is small, the matching accuracy can be increased, but the matching time will increase; otherwise, the matching accuracy and the matching time may decrease. As shown in figure 2.3 (left: axis of symmetry; right: parameter settings).

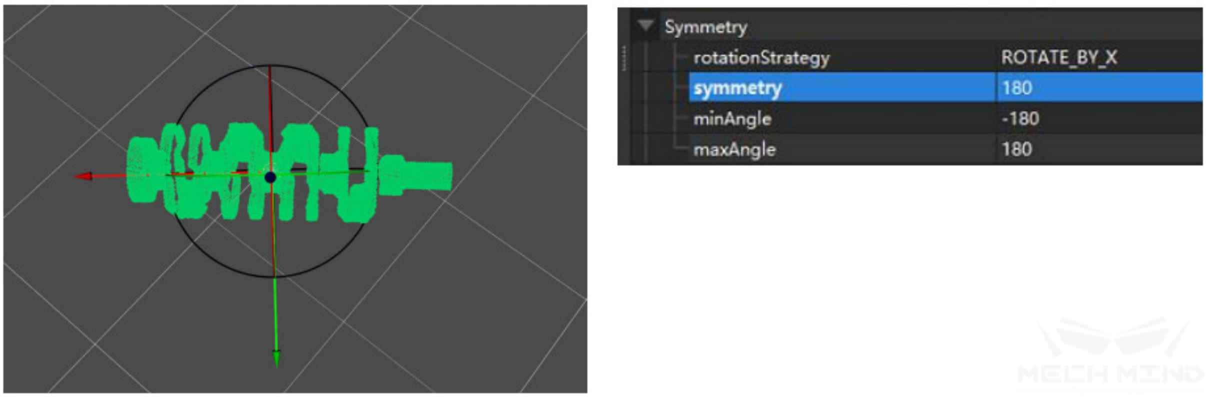


Figure 2.2 Axisymmetry - crankshaft

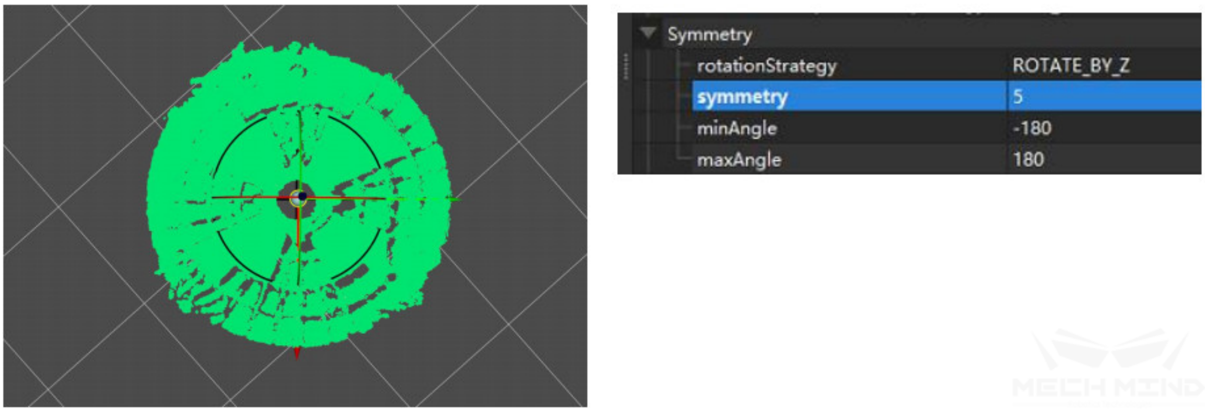


Figure 2.3 Continuous symmetry - roller

3> Set weights:

- For Axisymmetric workpiece: if it is pure axisymmetric, it is unnecessary to set the weight; otherwise for crankshaft-like workpiece, its two ends do not have symmetry. Therefore, in order to avoid mismatches caused by rotation matching, the features of both ends should be treated as weight templates and set in the *3D Fine Matching* weight parameter. The template is shown in figure 2.4 (left: Symmetric matching template; right: Symmetric weight template).
- For Centrosymmetric workpiece: the weight setting is consistent with the axisymmetric workpiece.

- For Continuous symmetric workpiece: the weight setting is consistent with the axisymmetric workpiece. As shown in figure 2.5 (left: Continuous symmetric matching template; right: Continuous symmetric weight template).

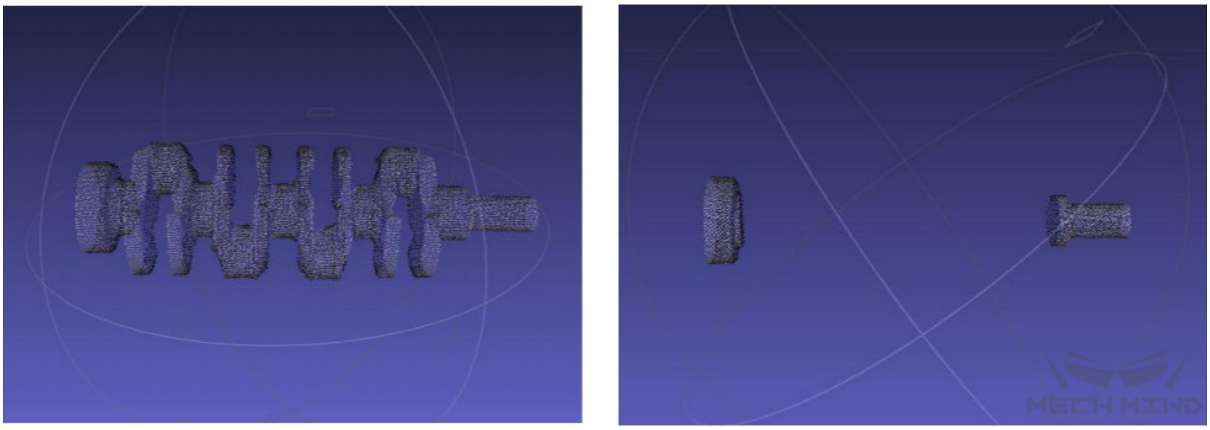


Figure 2.4 Symmetric matching template - weight template

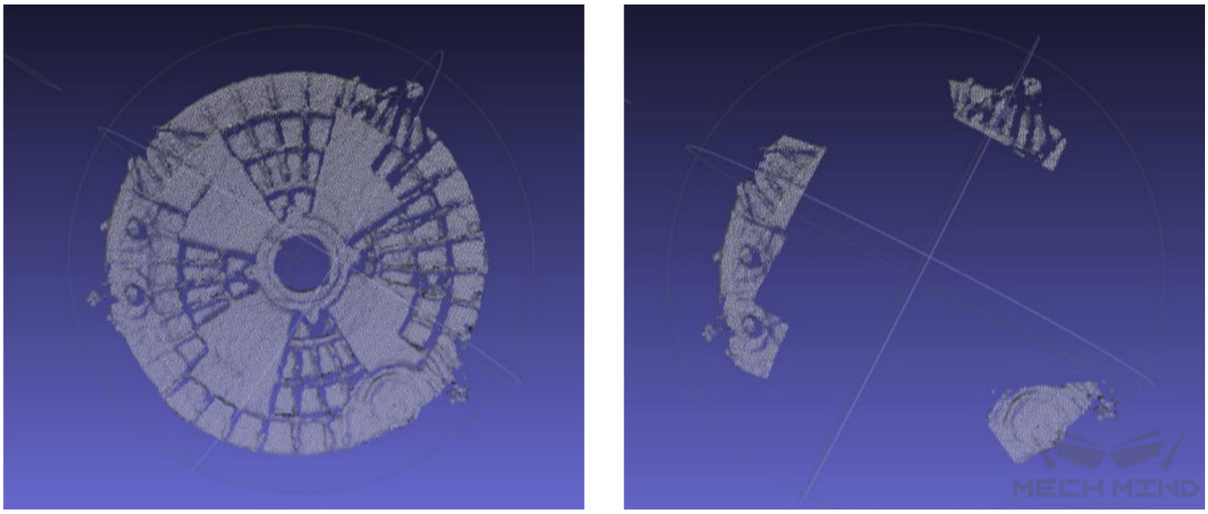


Figure 2.5 Continuous symmetric matching template - weight template

10.2.4 No results for getting highest-score poses in 3D multi-matching

When performing multiple matching (multiple matches are included in a project) to get the highest matching pose, sometimes there will be no results. The solution is as follows:

1> Make sure the value of confidence in *3D Fine Matching* is 0 (To avoid the problem that the number of input parameters in **Get Highest Score Result** does not match due to a certain matching score of 0).

2> Make sure that the number of *3D Coarse Matching* outputs is 1 (Generally it is 1, if there are multiple outputted matching poses, there is a chance that the number of outputted poses in two ways are different).

10.2.5 No results for 2D matching

During 2D matching, sometimes there will be no output results. At this time, check the size of template and check the depth roi and 3d roi;

1> Check the size of template;

- Make sure that the size of template set by Step *Read Object Dimensions* is consistent with the size of the actual workpiece;
- When the size is consistent, increase the threshold *maxMeanDistanceError* of *2D Matching* ;
- Adjust the expansion parameter of the outermost mask to expand the range of 2D matching input ROI;

Setting in Step is shown as in figure 2.6 (left: *Read Object Dimensions*; right: *2D Matching*).

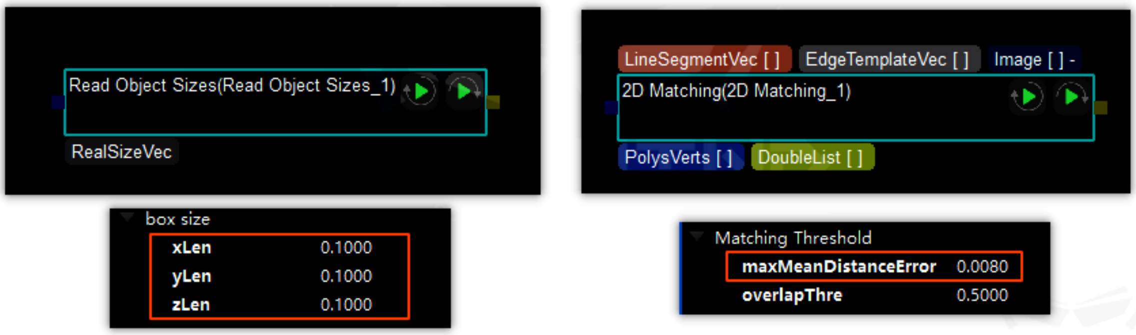


Figure 2.6 Settings of parameters related to 2D matching

2> Check depth roi and 3d roi: mainly to avoid that objects are out of the matching range.

10.3 Common problems about pose transform

10.3.1 Description for common problems about pose transform

Common problems about pose transform mainly list the common problems in the actual use of the software, and provide the corresponding solutions.

10.3.2 Z-direction downward (pointing to the ground)

Problems description: The actual grasping requires that the Z-direction of the object points upward, but sometimes the detected Z-direction of pose points to the ground. As shown in Figure 1.1.

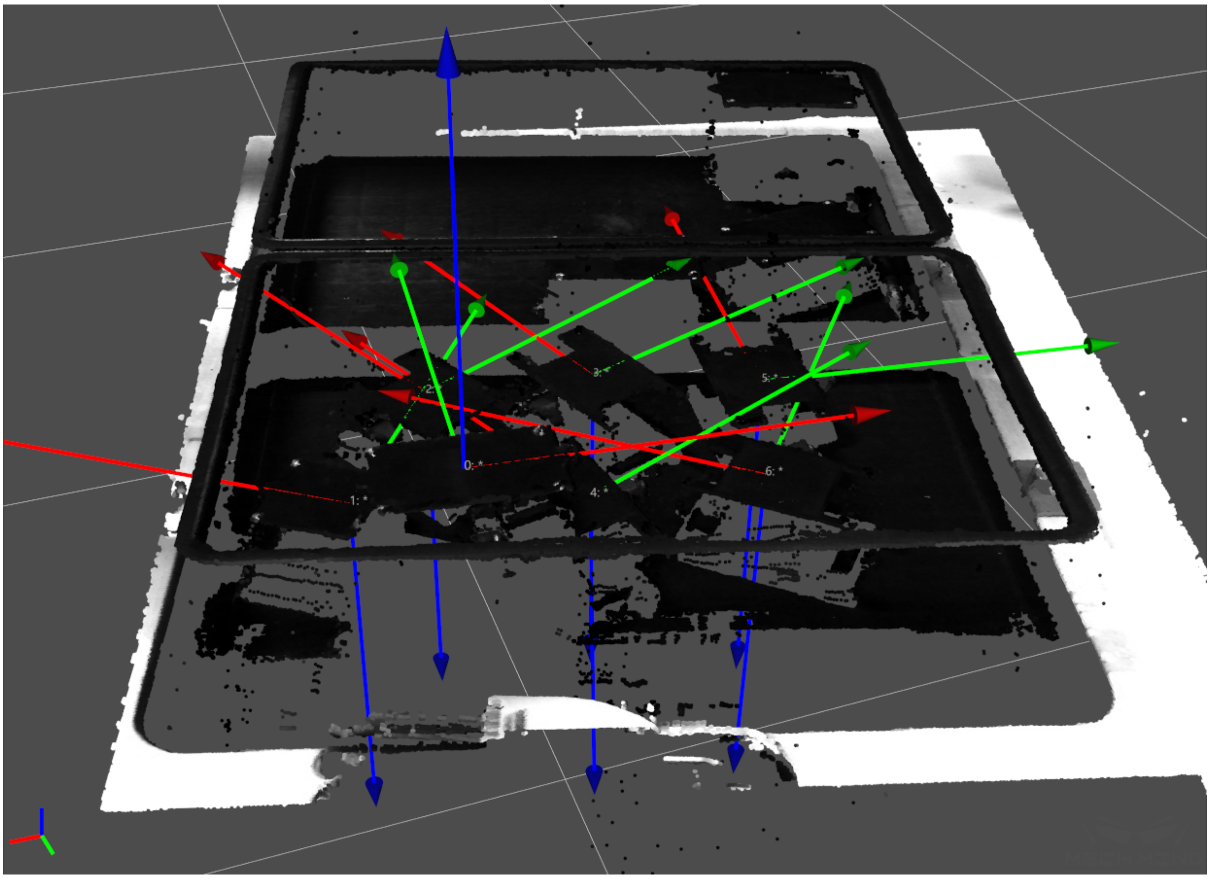


Figure 1.1 Z-direction pointing to the ground

Reason: The orientation is not checked and corrected before outputting the pose.

Solution: Add Step *Flip Poses' Axes* and set the corresponding parameters. For the results with Z-downward , rotate 180° around the x-axis of the detected poses. The parameters are shown in figure 1.2. The recognition result after adding this Step is shown in figure 1.3.

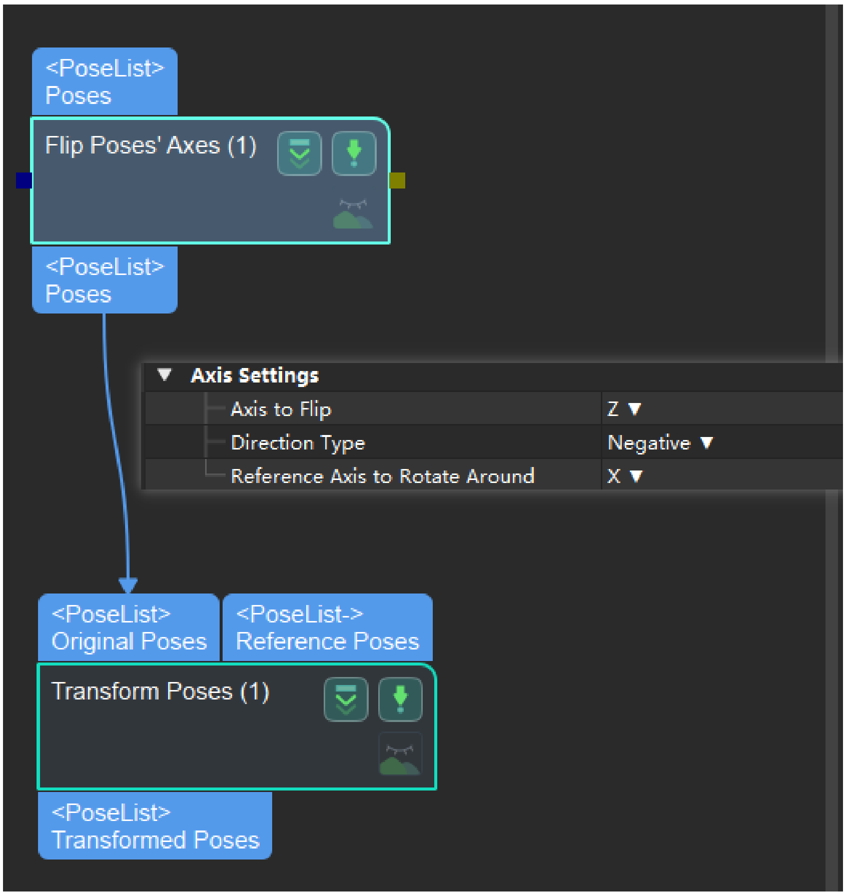


Figure 1.2 Parameters for Step "Flip Axis Direction"

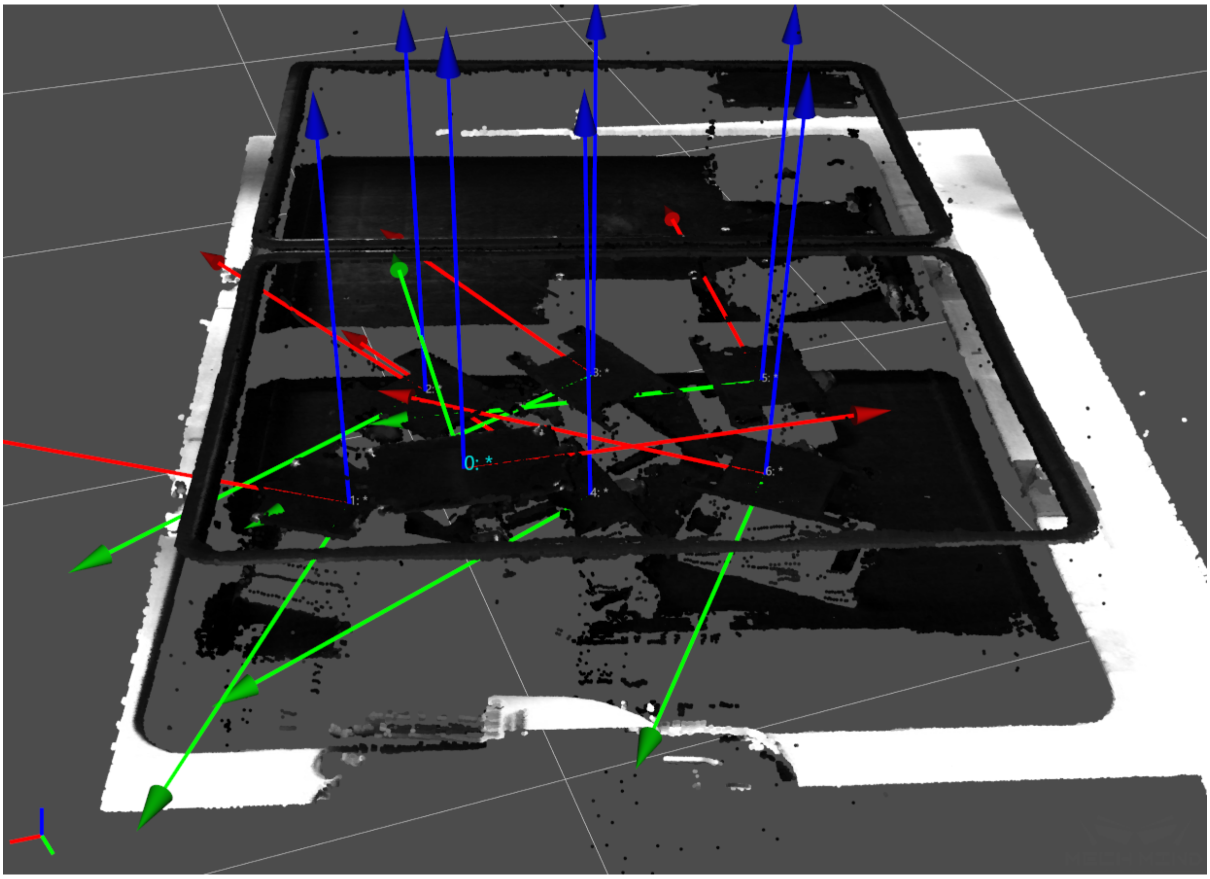


Figure 1.3 Correct recognized result on Z-direction