



Mech-Eye Industrial 3D Camera User Manual

v2.2.2

Table of Contents

1. Welcome	1
2. Getting Started	4
3. Mech-Eye SDK Release Notes	16
4. Mech-Eye SDK Installation Guide	17
5. Mech-Eye Viewer	22
5.1. User Interface	22
5.2. How to Use	26
5.2.1. Set Camera IP Address and Connect to Camera	27
5.2.2. Acquire and View Data	29
5.2.3. Adjust Parameters	37
5.2.4. Save Data	41
5.2.5. Manage Log	42
5.3. Parameter Reference Guide	43
5.3.1. Parameters of NANO	43
5.3.2. Parameters of PRO S and PRO M	59
5.3.3. Parameters of PRO XS	79
5.3.4. Parameters of UHP-140	95
5.3.5. Parameters of DEEP / LSR Series	112
5.3.6. Parameters of V3 Cameras	129
5.4. Tools	147
5.4.1. Intrinsic Parameter Tool	148
5.4.2. 3D Exposure Assistant	152
5.4.3. Depth Analyzer	152
5.4.4. Check and Configure 2D Cameras	153
5.4.5. FOV Calculator	155
5.4.6. Custom Reference Frame	156
5.4.7. Camera Controller	159
5.4.8. Full Pallet Simulator	160
6. Mech-Eye API	163
6.1. Mech-Eye SDK Installation Guide (Ubuntu)	164
6.2. Functional Hierarchy of Classes	166
6.3. Operation Workflow	168
6.4. Migration Guide from Version 2.1.0 to Version 2.2.0	175
6.5. Mech-Eye API Reference	212
6.6. Sample Usage Guide	212
6.6.1. C++ (Windows)	213
6.6.2. C++ (Ubuntu)	221
6.6.3. C#	227
6.6.4. Python (Windows)	230
6.6.5. Python (Ubuntu)	233
6.6.6. VisionPro	237
6.7. Tools	240

6.7.1. Firmware Upgrade Tool	240
6.7.2. IP Configuration Tool	241
6.8. Backward Compatibility	241
6.8.1. Backward Compatibility of Version 2.2.0	241
6.8.2. C++ Parameter Name Changes (2.0.0)	243
6.8.3. C# Method and Parameter Name Changes (2.0.0)	244
7. GenICam Interface	249
7.1. What is GenICam?	249
7.2. HALCON	249
7.2.1. HALCON: Connect to Camera, Perform Image Capturing, and Adjust Parameters	250
7.2.2. HALCON: Hand-Eye Calibration	257
7.2.3. HALCON: Obtain Textured Point Cloud	266
7.2.4. HALCON: Obtain Depth Values Only	267
7.2.5. HALCON: Set IP Address	268
7.3. GenICam Reference	269
7.3.1. Camera Parameters Supported for GenICam	270
7.3.2. Use Mech-Eye Viewer to Adjust Camera Parameters for GenICam Client	274
7.3.3. Obtain HALCON-Readable Point Cloud through Mech-Eye API	276
8. Camera Hardware	277
8.1. Camera Models	277
8.2. Hardware User Manual	279
8.3. Technical Specifications	290
8.3.1. NANO	291
8.3.2. PRO S / PRO M	293
8.3.3. UHP-140	298
8.3.4. DEEP	300
8.3.5. LSR S	303
8.3.6. LSR L	306
8.3.7. Technical Specifications of V3 Cameras	310
9. Extended Reading	312
9.1. Imaging Mechanism of Structured-Light Camera	312
9.2. Built-in Parameter Groups	313
9.3. Set IP Address on Computer	314
9.4. Enable Jumbo Frames	316
10. Support	318
10.1. Troubleshooting	318
10.1.1. Camera Indicator Lights Indicated Abnormality after Power Was Connected	318
10.1.2. Camera Not Found in Mech-Eye SDK	321
10.1.3. Camera Cannot Be Connected in Mech-Eye SDK	324
10.1.4. Camera Did Not Project Light, and Depth Map Was Not Generated	327
10.1.5. Camera Cannot Be Connected in Mech-Eye SDK (Windows Server)	329
10.1.6. Image Acquisition Time Too Long	329
10.2. FAQs	331
10.2.1. How to Read Camera Serial Number?	331

1. Welcome

Introduction of Product and Documentation

Mech-Eye Industrial 3D Camera (“camera”) is the structured-light 3D camera developed by Mech-Mind. Through Mech-Eye SDK or third-party machine vision software, you can obtain 2D image, depth map, and point cloud from the camera.

This documentation provides contents including the software user manual of Mech-Eye SDK, camera hardware user manual, camera technical specifications and troubleshooting guidance.



- If you need to download the PDF hardware user manual and technical specifications, or obtain the camera 3D models, installation drawings, accessory information, product flyers, etc., please visit [Mech-Mind Download Center](#).
- If you encounter any issue while using the camera, please visit [Mech-Mind Online Community](#) to get help (account registration and sign-in required).

Essential Guide for New Users

The following topics provide the beginner’s tutorial for the camera, explanation of the camera’s imaging mechanism, and comparison of different camera models.

Tutorial

Guides you through the entire process from checking package contents to completing your first data acquisition:

[Getting Started](#)

Imaging Mechanism

Learn how the camera generates 3D data of objects:

[Imaging Mechanism of Structured-Light Camera](#)

Camera Models

Learn the characteristics and suitable applications of different camera models:

[Camera Models](#)

Software User Manual

Through Mech-Eye SDK or third-party machine vision software, you can connect to the camera, acquire data, and adjust parameters. Meanwhile, the camera also supports the GenICam standard and provides the GenICam generic programming interface.

Mech-Eye Viewer

Mech-Eye Viewer provides an easy-to-use graphical user interface. You can adjust parameters and check the effects immediately.

The following topic provides instructions on using Mech-Eye Viewer:

[Mech-Eye Viewer User Guide](#)

Mech-Eye API

Mech-Eye API is the application programming interface of the camera, available in C++, C#, and Python. It is applicable on both Windows and Ubuntu.

Based on Mech-Eye API, you can develop your own camera control program:

[Mech-Eye API User Guide](#)

Through Mech-Eye API, you can also use third-party machine vision software (such as VisionPro) to control the camera and obtain data:

- [VisionPro Sample Usage Guide](#)
- LabVIEW Sample Usage Guide (coming soon)

GenICam Interface

Third-party machine vision software compliant with the GenICam standard (such as HALCON) can control the camera through the GenICam interface, which lowers the difficulty of integration into existing systems.

The following topic provides instructions on using Mech-Eye Viewer:

[Control the Camera with HALCON](#)

Reference Information

Camera Materials

The following topics provide the hardware user manual and technical specifications of the camera:

- [Hardware User Manual](#)
- [Technical Specifications](#)

If you need to download the PDF hardware user manual and technical specifications, or obtain the camera 3D models, installation drawings, accessory information, product flyers, etc., please visit Mech-Mind Download Center:

[Download camera materials](#)

Troubleshooting and FAQs

The following topics provide troubleshooting guidance for common issues and answers to frequently asked questions:

- [Troubleshooting](#)
- [FAQs](#)

If you encounter issues and questions not covered by the above topics, please visit Mech-Mind

Online Community to get help (account registration and sign-in required):

- [Post questions on the community](#)
- [Camera usage experience and tips](#)

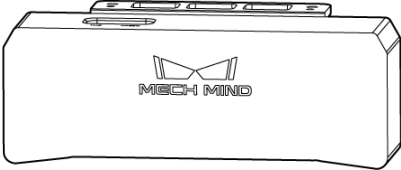


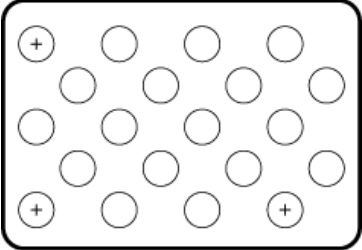
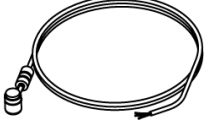
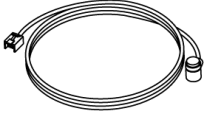
2. Getting Started

This topic will guide you through the entire process from checking package contents to using Mech-Eye Viewer to capture images.

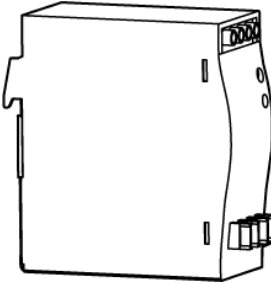
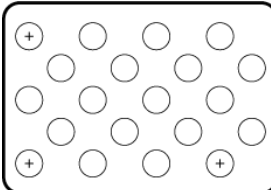
Check Package Contents

1. Make sure that the package is intact when you receive it.
2. Check contents according to the “packing list” in the package to ensure that no devices or accessories are missing or damaged.

 The table below is for reference only. Please take the “packing list” in the package as the final.

Camera	
Accessory bag	
User manual	
Calibration board (UHP-140 only)	
DC power cable (20 m) CBL-PWR-20M-LU	
Ethernet cable (20 m) CBL-ETH-20M-LU	

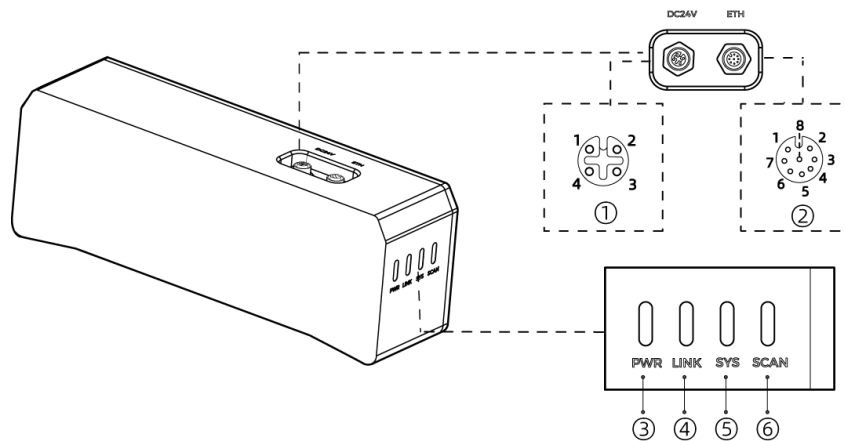
- Optional accessories:

DIN rail power supply	
Calibration board (except for UHP-140)	

Check Ports and Indicator Lights

Please refer to the following figures and tables and check the functions of the ports and indicator lights on the camera.

PRO S / PRO M, DEEP, UHP-140, LSR S, and LSR L

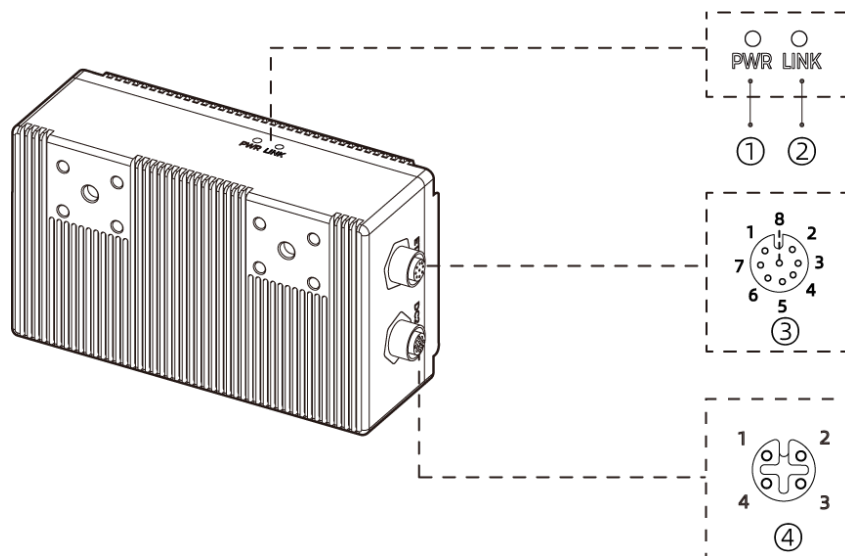


The figure above is for reference only. The actual product may differ.

No.	Name	Function	
①	DC 24V port	1: GND	2: GND
		3: 24 VDC	4: 24 VDC
②	ETH port	1: MD3_P	2: MD2_N
		3: MD2_P	4: MD0_P
		5: MD1_P	6: MD0_N
		7: MD3_N	8: MD1_N

③	PWR indicator light	Off: not connected to power
		Green: normal voltage
		Yellow: abnormal voltage, but system is still functioning
		Red: voltage too high or too low, system power is disconnected
④	LINK indicator light	Off: not connected to network
		Blinking green: data being transmitted
		Solid green: no ongoing data transmission
⑤	SYS indicator light	Off: system not started
		Solid green: system starting up
		Blinking green: system operating properly
		Blinking yellow: system operating with error but still functioning
		Blinking red: system not operating
⑥	SCAN indicator light	Solid: capturing and processing in progress
		Off: not capturing or processing

NANO and PRO XS




The above figure uses NANO as an example.

No.	Name	Function
①	PWR indicator light	Off: not connected to power
		Green: normal voltage
②	LINK indicator light	Off: not connected to network
		Blinking green: data being transmitted
		Solid green: no ongoing data transmission

③	ETH port	1: MD3_P	2: MD2_N
		3: MD2_P	4: MD0_P
		5: MD1_P	6: MD0_N
		7: MD3_N	8: MD1_N
④	DC 24V port	1: GND	2: GND
		3: 24 VDC	4: 24 VDC

Mount the Camera

The camera can be mounted in multiple ways. Please select the mounting method appropriate for your actual situation.

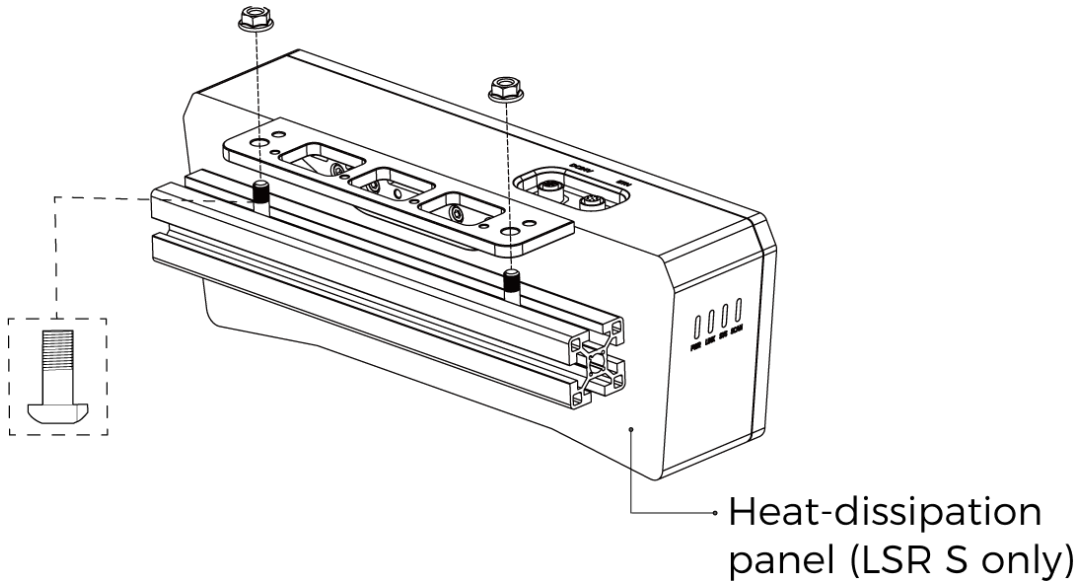
 Please prepare a wrench.

Mount through Camera Bracket

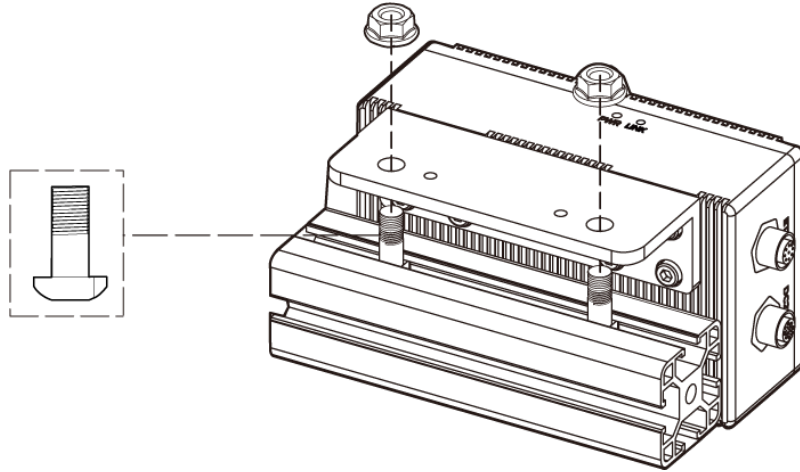
 The camera bracket and the heat-dissipation panel (LSR S only) are attached to the back of the camera when packaged.

As shown below, tighten the two bolts with the wrench.

- PRO S / PRO M, DEEP, UHP-140, LSR S, and LSR L:



- NANO and PRO XS:



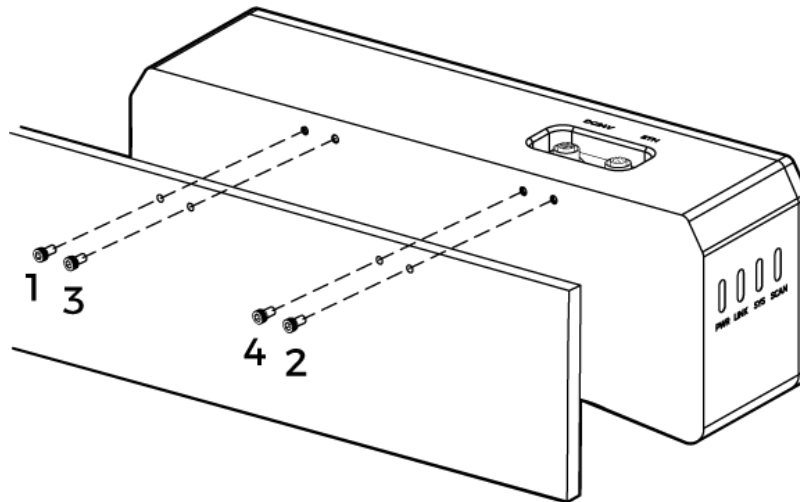
Mount through Threaded Holes on the Back



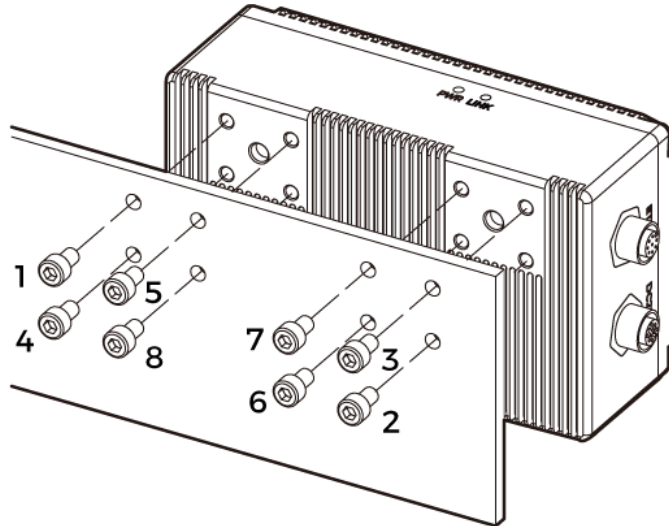
- Please remove the camera bracket and heat-dissipation panel (LSR S only) before mounting.
- When mounting LSR S using this method, ensure that the back of the camera is in close contact with a metal surface for heat dissipation, so that the camera does not overheat and malfunction.

As shown below, use the wrench to loosely screw on the bolts in the specified order, and then fully tighten all the bolts in the specified order.

- PRO S / PRO M, DEEP, UHP-140, LSR S, and LSR L:



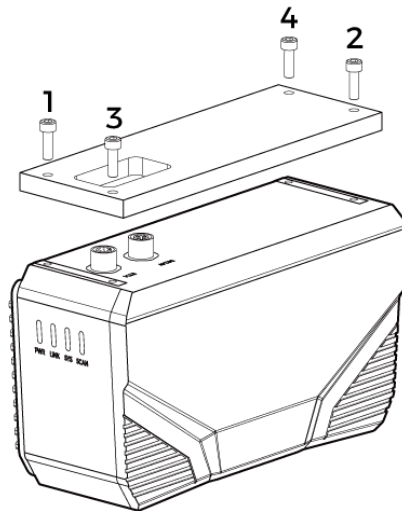
- NANO and PRO XS:



Mount through Threaded Holes at the Top (LSR S and UHP-140)

-  Please remove the camera bracket before mounting.

As shown below, use the wrench to loosely screw on the bolts in the specified order, and then fully tighten all the bolts in the specified order.



Connect Cables

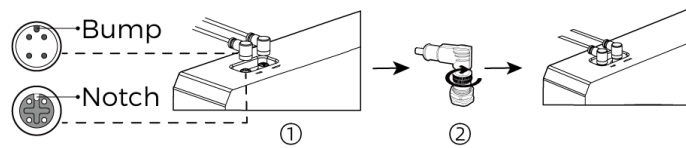
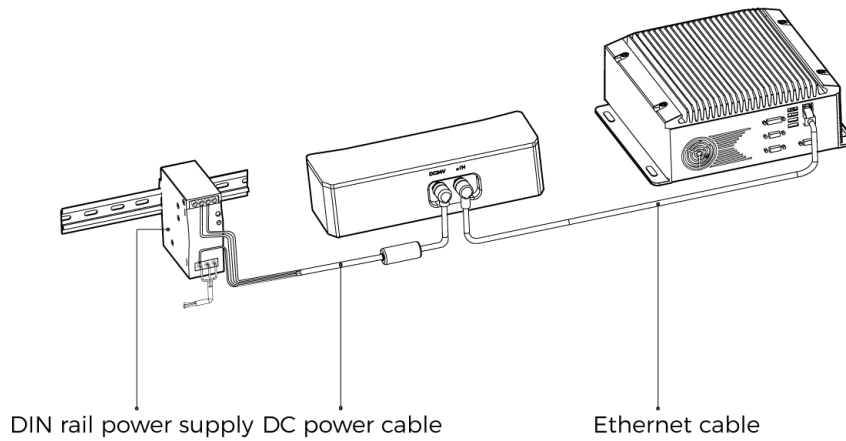
Please follow these steps to connect the Ethernet and power cables of the camera.

Ethernet Cable

As shown below, insert the M12-A connector of the Ethernet cable to the ETH port on the camera, and insert the RJ45 connector to the Ethernet port on the IPC.

DC Power Cable

As shown below, insert the M12-A connector of the DC power cable to the DC 24V port on the camera.



When inserting the Ethernet cable and DC power cable:

1. When inserting, align the bump in the connector with the notch in the port.
2. Tighten the nut. The recommended tightening torque is 0.7 N·m. A gap of about 2 mm remains after the nut is fully tightened.

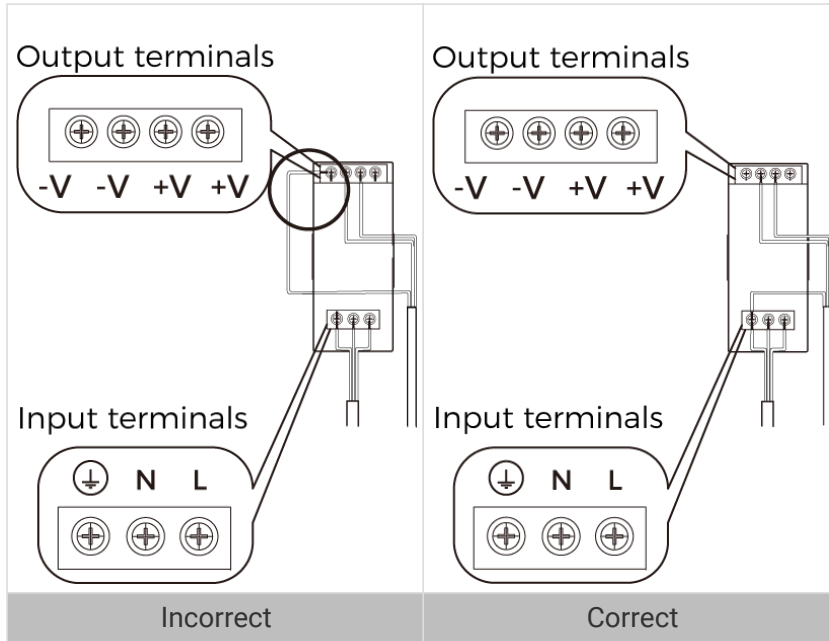


- Network switch can be used to connect the camera and IPC.
- Please connect the power cable last. After power is supplied, the PWR indicator light should be solid green. If not, please contact Mech-Mind.
- When mounting the camera onto a robot arm or other mobile equipment, please fasten the DC power and Ethernet cables on the camera end properly to avoid damaging the cables or connectors due to strain.

DIN Rail Power Supply



- Please prepare the AC power cable.
- The instructions below apply to the DIN rail power supply provided by Mech-Mind. If you use your own DIN rail power supply, please refer to its user manual for connection instructions.



The AC power cable has three wires: L, N, and PE (⊥).

The DC power cable has three wires: +V, -V and PE (⊥).

When wiring the DIN rail power supply, wires of the power cable must be inserted to the corresponding input/output terminals.

⚠ Warning

- Please mount the DIN rail power supply inside a control cabinet.
- The DIN rail power supply or the DIN rail should be reliably grounded. If mounting multiple DIN rail power supplies on the same DIN rail, ensure enough distance in between.
- Please use a single-phase, 3-prong power outlet with protective grounding for the AC power.

The mounting and connection of the camera hardware is completed. The following sections will introduce how to use Mech-Eye Viewer to connect to the camera and control the camera to capture images.

Download and Install Mech-Eye SDK

Please contact Mech-Mind pre-sales or support engineers to obtain the Mech-Eye SDK installation package.


After decompressing the installation package, double-click the installer to install Mech-Eye SDK. For more information, please refer to [Mech-Eye SDK Installation Guide](#).

Set IP Addresses

Before connecting to the camera, please make sure that the following IP address are unique and in the same subnet.

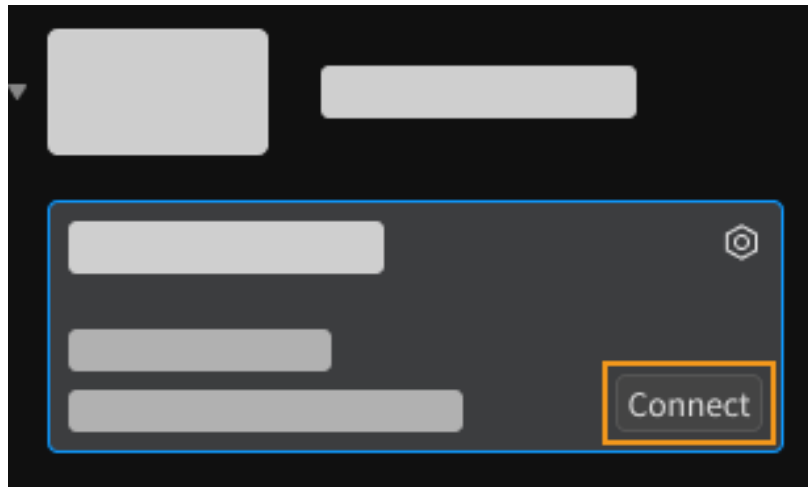
- Camera IP address
- The IP address of the IPC Ethernet port connected to the camera

Follow these steps to set the camera IP address:

1. Double-click to open Mech-Eye Viewer.
2. Find the camera to be connected and click .
3. [Set the camera IP address.](#)

Connect to Camera

1. Find the camera to be connected in Mech-Eye Viewer, and click [**Connect**].




If the software or firmware needs to be upgraded, the [**Upgrade**] button is displayed instead. Please click this button to perform upgrade first, and then connect to the camera.

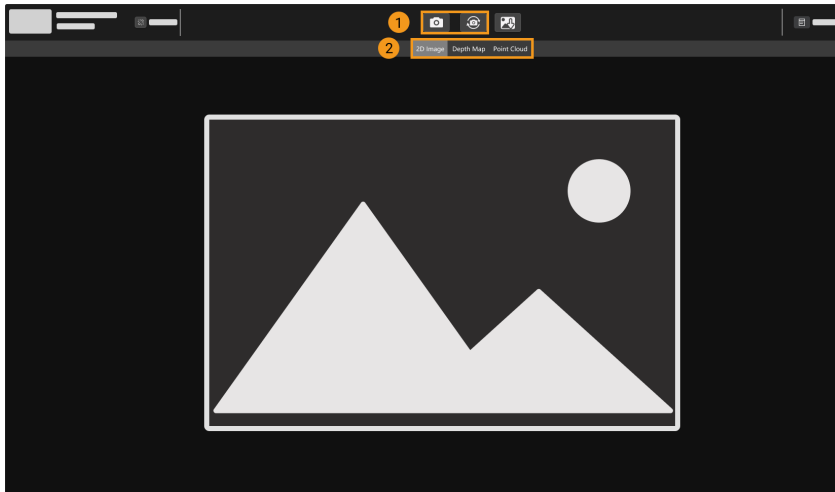
Capture Images

Click  to perform image capturing once.



Clicking  will trigger the camera to repeatedly capture images at predetermined intervals. Click again to stop the image capturing.

Click the data types below the image capturing buttons to check the 2D image, depth map and point cloud obtained from the camera.



For more information on image capturing and data types, please refer to [Acquire and View Data](#).

Adjust Parameters

If the quality of the obtained 2D image, depth map and point cloud is not satisfactory, you can adjust the relevant parameters in the **Parameters** tab on the right of the software to improve data quality.

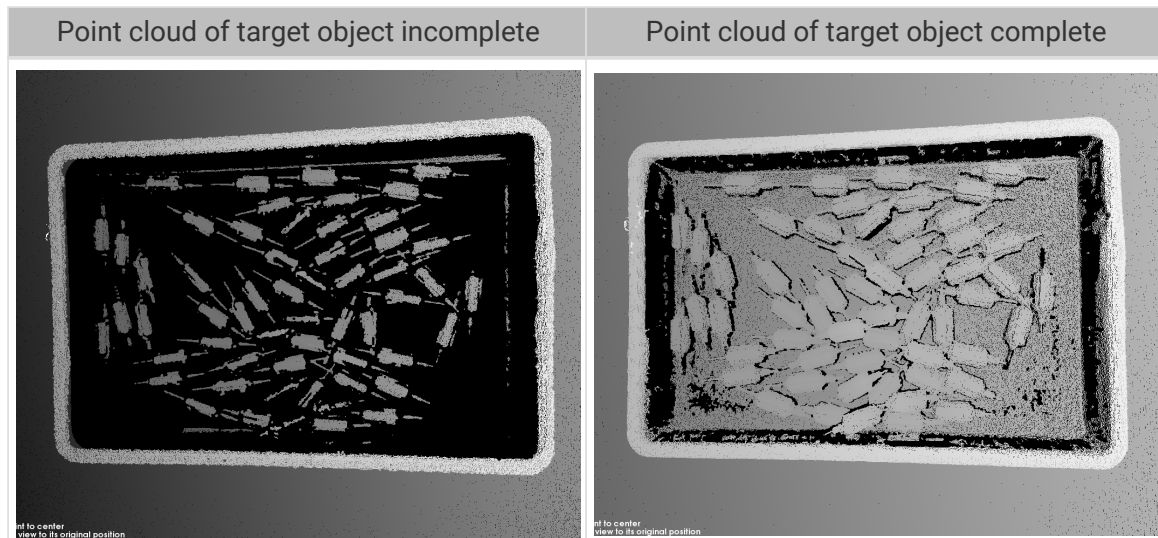
Determine Data Quality

Based on the following criteria, roughly determine the quality of the obtained data.

- 2D image: the 2D image is not too bright or too dark; the surface features of the target object are clearly visible.



- Depth map and point cloud: in the depth map and point cloud, the data corresponding to the target object should be complete. In the following example, the target object is the rotors.



Improve 2D Image Quality

1. Set the **Exposure Mode** parameter in the **2D Parameters** category to **Timed**, and then adjust the **Exposure Time** parameter.
 - If the 2D image is too dark, increase **Exposure Time**.
 - If the 2D image is too bright, decrease **Exposure Time**.



DEEP and LSR series provide two types of 2D images corresponding to different parameters. For detailed information, please refer to [Parameters of DEEP and LSR Series](#).

2. Capture images again and check the quality of the 2D image.

Improve Quality of Depth Map and Point Cloud


1. Adjust the **Exposure Time** parameter in the **3D Parameters** category.
 - If the object is dark-colored or not reflective, increase **Exposure Time**.
 - If the object is light-colored or reflective, decrease **Exposure Time**.
2. Capture images again and check the quality of the depth map and point cloud.



For more explanations of the parameters, please refer to [Parameter Reference Guide](#).

Use Data

The 2D image, depth map and point cloud obtained through Mech-Eye Viewer can be saved locally or can be output to Mech-Vision or third-party machine vision software for subsequent processing and calculation.

- Save the data: Click  in the data acquisition area, set the destination path, check the data types to be saved, and then click [**Save**].
- Use the data in Mech-Vision: Please refer to [Vision System Tutorial](#) and learn how to build the entire vision system that includes Mech-Vision.

- Use the data in third-party machine vision software: The data obtained by the camera can be transmitted to third-party software through [Mech-Eye API](#) or the [GenICam interface](#).

3. Mech-Eye SDK Release Notes

This topic introduces the new features, improvements, and resolved issues of Mech-Eye SDK 2.2.2.

Mech-Eye Viewer

Improved Intrinsic Parameter Tool

Mech-Eye Viewer 2.2.2 adjusted the standards for intrinsic parameter check, more suitable for actual usage scenarios.

Release Notes of Previous Versions

- [Mech-Eye SDK 2.2.1 Release Notes](#)
- [Mech-Eye SDK 2.2.0 Release notes](#)
- [Mech-Eye SDK 2.1.0 Release Notes](#)

4. Mech-Eye SDK Installation Guide

This topic guides you through the downloading, installation, upgrade, uninstallation, repair, and installation modification of Mech-Eye SDK (including Mech-Eye Viewer and Mech-Eye API) on Windows.

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

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



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

Download Mech-Eye SDK Installation Package

You can download the Mech-Eye SDK installation package from [Mech-Mind Download Center](#).

Verify the Integrity of Mech-Eye SDK Installation Package

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



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

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

1. Copy the installation package to a specified directory on the IPC or PC, such as *D:/*.
2. Decompress the installation package. After decompression, you will obtain the software installer (Mech-Eye SDK Installer 2.2.2.exe).
3. Open 7-Zip, and use the address bar to navigate to the directory where the software installation package is located.
4. Select the software installation package, and in the menu bar, select File > CRC > CRC-32 to calculate the CRC-32 value.
5. Check the calculated CRC32 value against the one on the download page. The two values should be identical.
6. Repeat steps 3 to 5 to the software installer obtained after decompression.



If the calculated values do not match those provided on the download page, please download it again.

If you are using Mech-Eye SDK for the first time, refer to [Install Mech-Eye SDK](#) to complete the software installation.

If you have already installed Mech-Eye SDK, refer to [Upgrade Mech-Eye SDK](#) to upgrade the software to the latest version.

Install Mech-Eye SDK

To install Mech-Eye SDK, follow these steps:

1. Double-click the downloaded installer to open the Mech-Eye SDK setup wizard.
2. In the **Welcome** window, browse the software information and click **[Next]**.
3. In the **License Agreement** window, read the license agreement carefully, check the **I accept the terms and conditions in the license agreement** check box, and then click **[Next]**.
4. In the **Select Product** window, select the components to install. If needed, check the **Create desktop shortcut(s)** option, and then click **[Next]**.



- Make sure that the **Add to PATH** option is checked.
- It is recommended to check the **Mech-Eye SDK Docs** option, which allows you to open the user manual without Internet connection.

5. In the **Set Path** window, change the installation path as required, and then click **[Next]**.



Default installation path: *C:/Mech-Mind/Mech-Eye SDK-x.x.x*.

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



It is recommended to restart your computer after the installation. If not restarted, the environmental variable added to PATH may not be effective.

Upgrade Mech-Eye SDK



- If you have Mech-Eye SDK above 1.6.1 installed, use the later version of setup wizard to upgrade.
- If you have Mech-Eye SDK 1.6.1 or below installed, please uninstall the lower version and then [install the new version](#).

To upgrade Mech-Eye SDK, follow these steps:

1. Double-click **Mech-Eye SDK Installer x.x.x.exe** to open the Mech-Eye SDK setup wizard.
2. In the **Upgrade** window, click either **[Upgrade and keep history versions]** or **[Upgrade and delete history versions]**.
3. Upgrade the software according to the prompts in the setup wizard.
4. After the software is upgraded, in the **Finish** window, click **[Finish]** to exit the setup wizard.

Uninstall Mech-Eye SDK

You can uninstall Mech-Eye SDK either with the Mech-Eye SDK setup wizard or the Windows

Control Panel.

Uninstall Using the Setup Wizard

To uninstall Mech-Eye SDK using the setup wizard, follow these steps:


1. Double-click **Mech-Eye SDK Installer x.x.x.exe** to open the Mech-Eye SDK setup wizard.
2. In the **Maintain** window, click [**Uninstall**].
3. In the **Uninstall** window, click either [**Keep user configuration files**] or [**Do not keep user configuration files**].
4. Wait for the uninstallation to complete. After the software is uninstalled, in the **Finish** window, click [**Finish**] to exit the setup wizard.

Uninstall Using the Control Panel

To uninstall Mech-Eye SDK using the Control Panel, follow these steps:

1. Open the **Control Panel** on the computer.
2. Choose **Program > Uninstall or change a program**.
3. Right-click **Mech-Eye SDK**, and then select **Uninstall**. The Mech-Eye SDK setup wizard will be opened.
4. In the **Maintain** window, click [**Uninstall**].
5. In the **Uninstall** window, click either [**Keep user configuration files**] or [**Do not keep user configuration files**].
6. Wait for the uninstallation to complete. After the software is uninstalled, in the **Finish** window, click [**Finish**] to exit the setup wizard.

Repair Mech-Eye SDK

 Only Mech-Eye SDK setup wizard 2.0.0 and above supports this function.

If Mech-Eye SDK failed to work normally, you can repair it using the Mech-Eye SDK setup wizard.

To repair Mech-Eye SDK, follow these steps:

1. Double-click **Mech-Eye SDK Installer x.x.x.exe** to open the Mech-Eye SDK setup wizard.
2. In the **Maintain** window, click [**Repair**].
3. Wait for the repair to complete. After the software is repaired, in the **Finish** window, click [**Finish**] to exit the setup wizard.

Modify Installed Components

 Only Mech-Eye SDK setup wizard 2.0.0 and above supports this function.

After installation, you can use the setup wizard to change the installed components of the software. To modify installed components, follow these steps:

1. Double-click **Mech-Eye SDK Installer x.x.x.exe** to open the Mech-Eye SDK setup wizard.

2. In the **Maintain** window, click [**Modify**].
3. In the **Select Product** window, select the products or components to install, or deselect the products or components to uninstall.
4. Install the software according to the prompts in the setup wizard.
5. After the software is installed, in the **Finish** window, click [**Finish**] to exit the setup wizard.

Software License Agreement

For the license agreement of Mech-Eye SDK, refer to [End-User License Agreement](#).

Troubleshoot Common Installation Problems

Failed to Start the Setup Wizard

Symptom:

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

Possible Causes:

There is not enough space on the system drive.

Troubleshooting Procedure:

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

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

Installation Failed

Symptom:

An error message saying "Installation failed." appeared in the **Finish** window during the installation.

Possible Causes:

- Corrupted installer or missing files.
- The current user does not have administrator access.
- Another Wizard is performing the installation or the Windows system is being upgraded.
- Other reasons.

Troubleshooting Procedure:

1. Obtain a new installer and [check its integrity](#), and then try the installation again.
 - If the problem is resolved, the troubleshooting is completed.

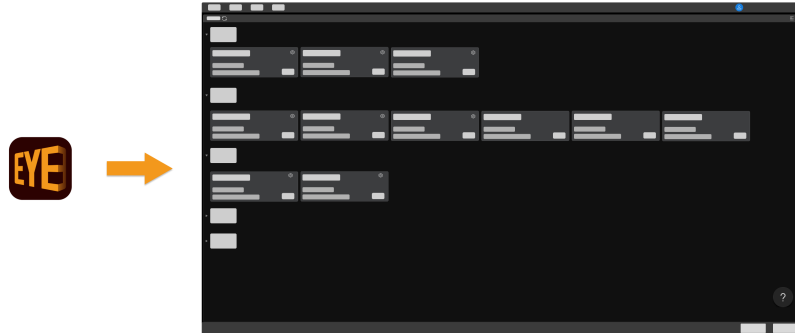
- Otherwise, proceed to step 2.
- 2. Right-click the installer and select **Run as administrator** to reinstall.
 - If the problem is resolved, the troubleshooting is completed.
 - Otherwise, proceed to step 3.
- 3. Try the installation again after the installation of other software completes or after the Windows system upgrade completes.
 - If the problem is resolved, the troubleshooting is completed.
 - Otherwise, proceed to step 4.
- 4. Restart your computer and try the installation again.
 - If the problem is resolved, the troubleshooting is completed.
 - If the problem still exists, click the **installation log** link in the **Finish** window to collect installation logs and send them to Technical Support.

5. Mech-Eye Viewer

Mech-Eye Viewer is a graphical interface software developed based on Mech-Eye API. Based on the characteristics of the target objects, you can adjust the parameters of the camera using Mech-Eye Viewer to conveniently obtain high-quality 2D images, depth maps, and point clouds.



Currently, Mech-Eye Viewer is only available on Windows. Ubuntu users can control the camera with [Mech-Eye API](#).



Introduction of the user interface:

[User Interface](#)

Instructions on connecting to camera, acquire data, and adjust parameters:

[How to Use](#)

Detailed information on parameters:

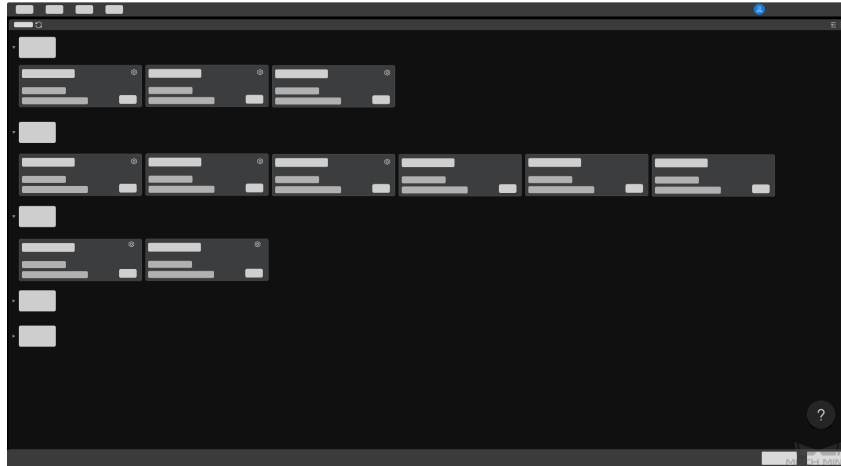
[Parameter Reference Guide](#)

Instructions on using the tools in Mech-Eye Viewer:


[Tools](#)

5.1. User Interface

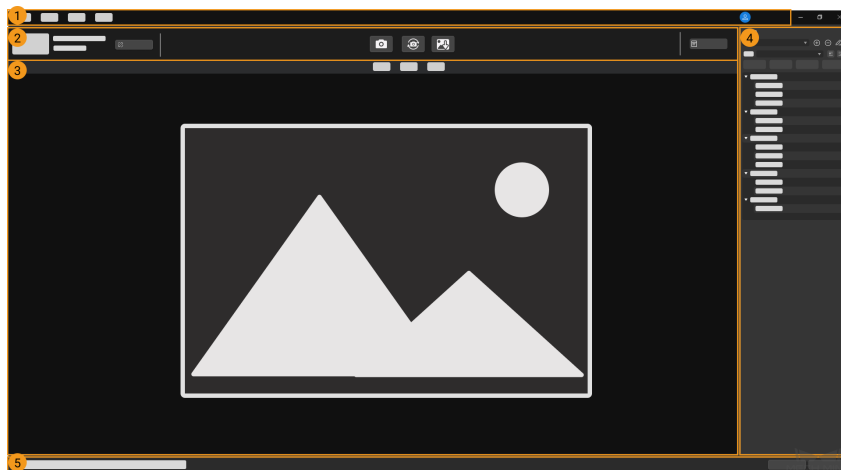
When you open Mech-Eye Viewer, the following interface is displayed.



In this interface, you can view all the available cameras and their information, [set camera IP address and connect to a camera](#).

Hover the cursor on , and then click **Industrial 3D Camera** to view the troubleshooting guidance on [camera connection issues](#).

After the camera is successfully connected, the following interface is displayed.




This interface consists of the following parts:

1. **Menu bar**: provides the **File**, **Tools**, **View**, and **Help** menus.
2. **Data acquisition area**: used to set a custom name, acquire data, save data, disconnect, and view the log.
3. **Data display area**: displays the data obtained from the camera.
4. **Parameter and acquisition info panel**: used to adjust parameters, manage parameter groups, and view the information related to acquisition.
5. **Status bar**: displays the camera operation status. Click [**Previous**] or [**Next**] to switch between messages.

Menu Bar

The menu bar provides the **File**, **Tools**, **View**, **Help** and user switch menus.

Menu	Option	Description
File	Save Raw Data of Camera	Save the raw data in MRAW format for analysis and troubleshooting later on.
	Start Virtual Camera	Load saved raw data and view the parameter values set when the data was saved.
Tool	Provide supplementary tools. For detailed information, please refer to Tools .	
View	Toolbar	Check to display the toolbar. Unselected by default.
	Image Information Box	Check to display the image information box on the 2D image and depth map. Checked by default.
	Point Cloud Exhibit Button	Check to display the [Play] button on the point cloud, used to exhibit the point cloud . Unchecked by default.
	Parameter Description	Check to display the Parameter Description section in the Parameters tab. Checked by default.
	Depth Map Grid	Select whether to display a grid on the depth map and the grid pattern to be displayed.
Help	About	Display software version.
	What's New	Open the Mech-Eye SDK release notes in your browser.
	User Manual	Open the user manual in your browser.
	Online Community	Open Mech-Mind Online Community in the browser.
	Options	Switch the software display language. Change is applied after the software is restarted.
	Click to switch the user account. The default account is Standard . If you need to use the Administrator account, please contact Mech-Mind Technical Support.	

Data Acquisition Area

In the data acquisition area, you can:

- [Set a Custom Name](#)
- [Capture images](#)
- [Save obtained data to local](#)
- [View the log](#)
- Disconnect from the camera

Data Display Area

This area displays the data output from the camera. You can [switch among 2D image, depth map and point cloud](#).

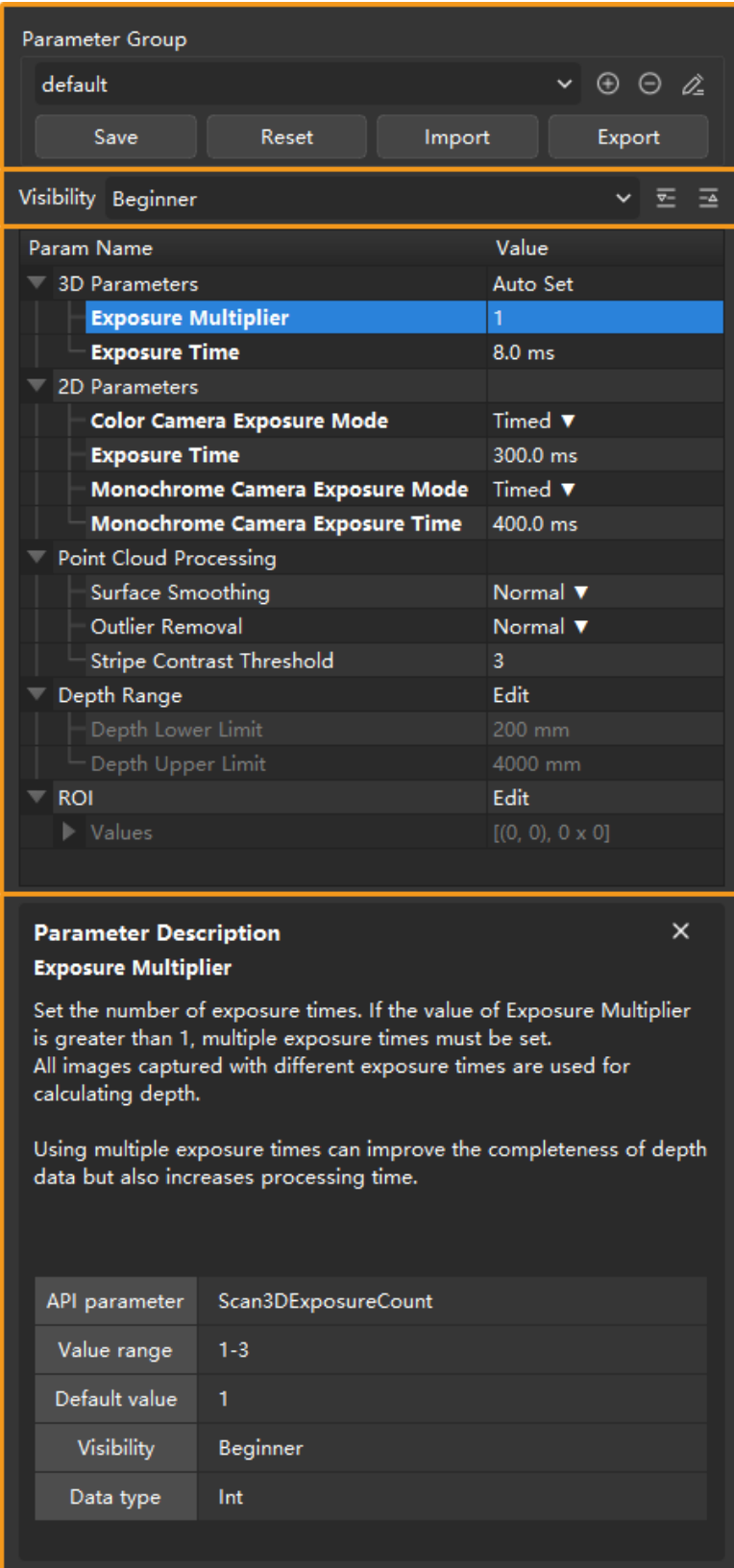
For detailed information on the function of this area, please refer to [Adjust Data Display](#).

Parameter and Acquisition Info Panel

This panel includes two tabs, **Parameters** and **Acquisition info**.

Parameters Tab

The Parameters tab consists of the following sections:



The screenshot shows the Parameters Tab interface with four numbered callouts:

- 1** Points to the **Parameter Group** section, which includes a dropdown menu set to "default", and buttons for **Save**, **Reset**, **Import**, and **Export**.
- 2** Points to the **Visibility** section, which is currently set to **Beginner**.
- 3** Points to the main parameter list table.
- 4** Points to the **Parameter Description** panel for the selected **Exposure Multiplier** parameter.

Param Name	Value
3D Parameters	Auto Set
Exposure Multiplier	1
Exposure Time	8.0 ms
2D Parameters	
Color Camera Exposure Mode	Timed ▼
Exposure Time	300.0 ms
Monochrome Camera Exposure Mode	Timed ▼
Monochrome Camera Exposure Time	400.0 ms
Point Cloud Processing	
Surface Smoothing	Normal ▼
Outlier Removal	Normal ▼
Stripe Contrast Threshold	3
Depth Range	Edit
Depth Lower Limit	200 mm
Depth Upper Limit	4000 mm
ROI	Edit
Values	[(0, 0), 0 x 0]

Parameter Description ✕

Exposure Multiplier

Set the number of exposure times. If the value of Exposure Multiplier is greater than 1, multiple exposure times must be set. All images captured with different exposure times are used for calculating depth.

Using multiple exposure times can improve the completeness of depth data but also increases processing time.

API parameter	Scan3DExposureCount
Value range	1-3
Default value	1
Visibility	Beginner
Data type	Int

1. Parameter group: create a new parameter group, save parameter values to a parameter group, delete a parameter group, rename a parameter group, and import and export parameter

groups.

2. Visibility: [change the visibility of parameters](#).
3. Parameter list: [adjust parameters](#).
4. Parameter description: [view more information of the selected parameter](#).

Acquisition Info Tab

The **Acquisition info** tab displays the following information related to the current data acquisition:

- Camera name and the completion time of the acquisition
- Acquisition time: the total time for acquiring the 2D image, depth map, and point cloud
 - 2D image / 2D image (texture): the time for acquiring the 2D image / 2D image (texture)
 - Depth data: the time for acquiring the depth map and point cloud
- Resolutions: content dependent on the camera model

Model	Displayed content	Description
Models other than LSR L, LSR S, and DEEP	2D image	Resolution of the 2D image
	Depth map	Resolution of the depth map
LSR L, LSR S, and DEEP	2D image (texture)	Resolution of the 2D image (texture)
	2D image (depth source)	Resolution of the 2D images (depth source)
	Depth map	Resolution of the depth map



The resolution of the 2D image (texture) or depth map of LSR L, LSR S, and DEEP can be changed using [Camera Controller](#) in the **Administrator** account.

- Temperature
 - CPU: the temperature of the camera CPU
 - Projector: the temperature of the camera projector

This topic introduced the user interface of Mech-Eye Viewer. Continue reading for instructions on using Mech-Eye Viewer.

5.2. How to Use

This topic provides instructions on using Mech-Eye Viewer to connect to the camera, acquire data, view the data, adjust parameters and save the data.

Before adjusting the parameters, please [check the intrinsic parameters of the camera](#). The coordinates of points are accurate only if the camera intrinsic parameters are accurate.

Instructions on [setting the camera IP address, connecting to the camera, and setting a custom camera name](#):

Set Camera IP Address and Connect to Camera

Instructions on **obtaining and viewing the data**:

[Acquire and View Data](#)

Instructions on **adjusting parameters** to improve data quality:

[Adjust Parameters](#)

Instructions on **saving the acquired data**:

[Save Data](#)

Information on **actions relevant to the log** in the case that you encounter a problem with the software:

[Manage Log](#)

5.2.1. Set Camera IP Address and Connect to Camera

To ensure that Mech-Eye Viewer can connect to the camera successfully, it is recommended to set the IP addresses first. Please make sure that the following IP addresses are in the same subnet and are unique.

- The [IP address of the camera](#)
- The [IP address of the computer Ethernet port connected to the camera](#)

After setting the IP addresses, you can [connect to the camera](#). Once the camera is connected, you can [set a custom name](#) for it for easy recognition of different devices.



If the camera to be connected is not displayed in Mech-Eye Viewer, please refer to [Camera Not Found in Mech-Eye SDK](#).

Set Camera IP Address


You can set a static IP address for the camera, or dynamically assign the IP address.

Set Static IP address



Record the set camera IP address for future reference.

Follow these steps to set a static IP address for the camera:

1. Open Mech-Eye Viewer, select the info card of the camera to be connected, and click  in the upper right of the card to open the **IP Configuration** window.
2. Select **Set as Static IP**.


- Based on the IP address and subnet mask displayed in **Computer IP Configuration**, select **IP Address Class**, and enter the camera IP address and subnet mask. Then, click [**Apply**] to apply the changes.



- The IP address must be unique.
- The **Computer IP Configuration** section displays the IP address and subnet mask of the computer Ethernet port connected to the camera.

Using dynamic IP address

Follow these steps to dynamically assign an IP address to the camera:

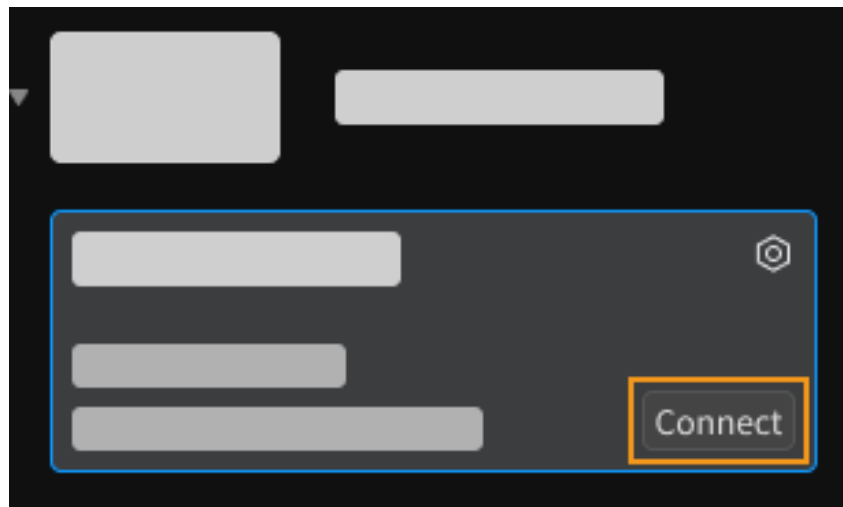
- Open Mech-Eye Viewer, select the info card of the camera to be connected, and click  in the upper right of the card to open the **IP Configuration** window.
- Select **Set via DHCP** and click [**Apply**].



If you want to set the IP address of the computer Ethernet port connected to the camera, please refer to [Set IP Address on Computer](#).

Connect to Camera


When the camera info card displays the [**Connect**] button, click this button to connect to the camera.



For the very first time that the camera is connected to the computer, it may take up to a few minutes for the camera to show up in Mech-Eye Viewer.


If other content is displayed, please complete the corresponding action according to the following table, and then connect to the camera.

Displayed content	Description
[Upgrade] button	Click this button to upgrade the software or the camera firmware first.
Connected	The camera is connected to a GenICam client. Please disconnect the camera from the GenICam client first.

Displayed content	Description
	Hover the cursor here to check the cause and solution, and resolve the issue according to these.

Set a Custom Name

The camera name can be used to distinguish different devices, clarifying the usages and locations of the devices, and more. Please set the custom name according to your actual needs. The custom name will be displayed in the **Device List**.

After connecting to the camera, on the left of the data acquisition area, click  or **Set device name** to set a custom name.

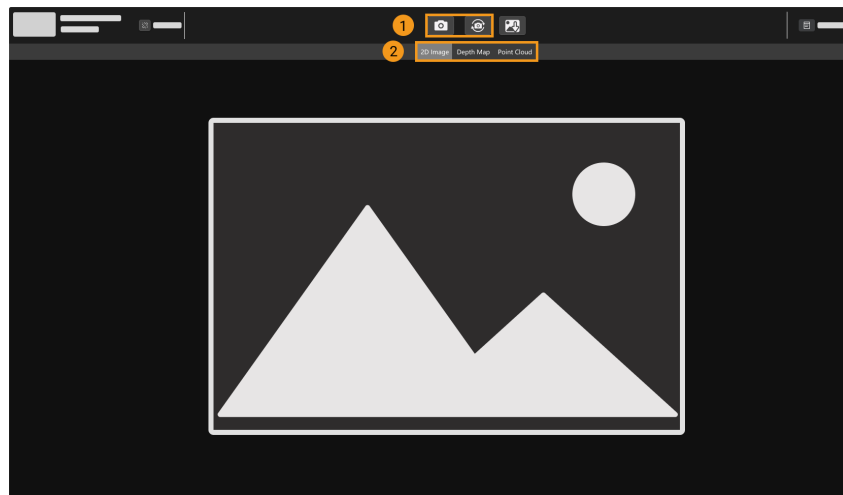


- The following characters cannot be used: \ / : * ? " < > | .
- Click the custom name again to modify it.
- The name of a virtual camera cannot be customized.

Once the camera is connected, you can start acquiring data. The next topic introduces the actions for acquiring data and viewing the obtained data.

5.2.2. Acquire and View Data



Once the camera is successfully connected, the following interface is displayed. In this interface, you can acquire and view the data.



1. Capture images
2. Switch Data Type

Capture images

There are two ways to acquire data:

-  : Acquire data once.
-  : Acquire data continuously at predetermined intervals. Click again to stop data acquisition.



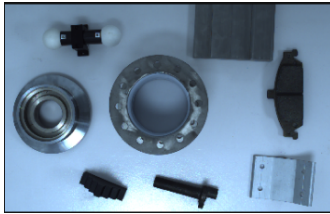
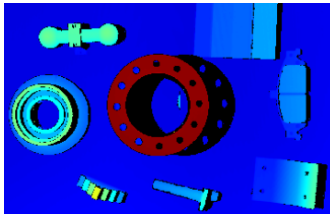
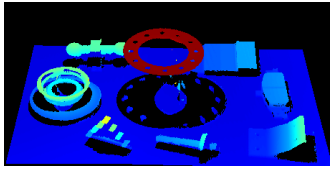
For an explanation of the data acquisition and output process of structured-light cameras, please refer to [Imaging Mechanism of Structured-Light Camera](#).

Switch Data Type


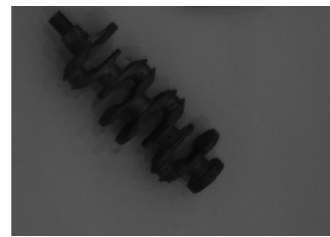
The camera outputs multiple types of data to Mech-Eye Viewer. Click the corresponding button to switch the displayed data type.



The types of data output by different camera models differ.

- The data output by models other than the DEEP (V4) and LSR (V4) series include the following types:

Data type	Description	Example
2D image	The 2D image used for texturing the point cloud	
Depth map	A 2D image containing depth information	
Point cloud	A collection of data points that represent the 3D shape of the object surface	

- The data output by the DEEP (V4) and LSR (V4) series include the following types:

Data type	Description	Example
2D image (texture)	The 2D image used for texturing the point cloud	
2D images (depth source)	The 2D images captured by the 2D cameras that acquire the depth data	

Data type	Description	Example
Depth map	A 2D image containing depth information	
Point cloud	A collection of data points that represent the 3D shape of the object surface	



The 2D images (depth source) are used in hand-eye calibration, checking intrinsic parameters and setting ROI.

Adjust Data Display

In the data display area, you can adjust the display of the data through mouse actions, keyboard, and buttons and menus in the software interface.

The instructions below are divided by data types.

Adjust 2D Image Display

In the 2D image display, you can perform the following actions:

- **Zoom:** While the cursor is on the 2D image, scroll the mouse wheel to zoom the 2D image.
- **Move:** While the 2D image is zoomed in, hold and drag to move the 2D image.
- **Reset size:** Right-click the 2D image and select **Reset View** to fit it to the window.
- **Check pixel position and RGB/grayscale values:** In the image information box at the lower right of the 2D image, you can check the pixel position and RGB/grayscale values of the pixel where the cursor is located.



- The pixel position of the upper-left corner of the image is (0, 0).
- To hide the image information box, uncheck **Image Information Box** in the **View** menu.

Adjust Depth Map Display

In the depth map display, you can perform the following actions:

- **Zoom:** While the cursor is on the depth map, scroll the mouse wheel to zoom the depth map.
- **Move:** While the depth map is zoomed in, hold and drag to move the depth map.
- **Reset size:** Right-click the depth map and select **Reset View** to fit it to the window.
- **Check pixel position and coordinates:** In the image information box at the lower right of the depth map, you can check the pixel position and coordinates in the camera reference frame of the pixel where the cursor is located.



- The pixel position of the upper-left corner of the image is (0, 0).
- To hide the image information box, uncheck **Image Information Box** in the **View** menu.

menu.

- **Adjust depth data display:** The display of the depth data can be adjusted in the left panel. For detailed instructions, please refer to [Adjust Depth Data Display](#).



A grid can be displayed on the depth map to help you locate the center of the FOV. Through the **Depth Map Grid** option in the **View** menu, select to display a 2×2 or 3×3 grid.

Adjust Point Cloud Display

In the point cloud display, you can perform the following actions:

- **Zoom:** While the cursor is on the point cloud, scroll the mouse wheel to zoom the point cloud.
- **Rotate:**
 - Rotate in 3D space: Hold the left mouse button and drag to rotate the point cloud.
 - Rotate in 2D plane: While holding the **Ctrl** key, hold the left mouse button and drag to rotate the point cloud.
- **Pan:** Hold the mouse wheel and drag to pan the point cloud.
- **Reset the view:**
 - Press the **R** key to reset the panning of the view, and adjust the zoom of the point cloud to a default level.
 - Press the **Backspace** key to reset the panning and rotation of the view, and adjust the zoom of the point cloud to a default level.
- **Display view cone:** Right-click and select **Show Frustrum** to display the FOV of the 2D cameras and projector in the form of view cones.
- **Display camera reference frame:** Right-click and select **Show Axes** to display the axes of the camera reference frame.
- **Adjust depth data display:** The display of the depth data can be adjusted in the left panel. For detailed instructions, please refer to [Adjust Depth Data Display](#).
- **Point Cloud Exhibit:** This function moves the point cloud along a predetermined trajectory to display it from various angles. For detailed instructions, please refer to [Point Cloud Exhibit](#).

Adjust Depth Data Display

In the left panel of the depth map or point cloud display, you can adjust the display reference frame, color rendering, and range of the depth data.

Change Reference Frame

In the **Reference Frame** menu at the top of the left panel, you can select the reference frame used for displaying the depth data.

- **Device** (default): displays the depth data in the camera reference frame.
- **Custom:** displays the depth data in the [custom reference frame defined by the user](#).

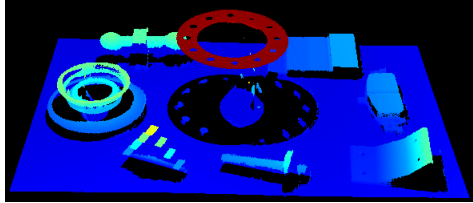
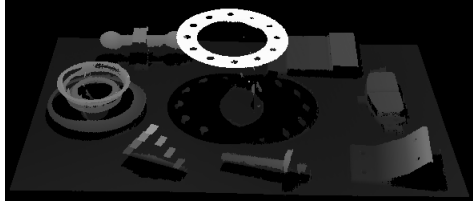
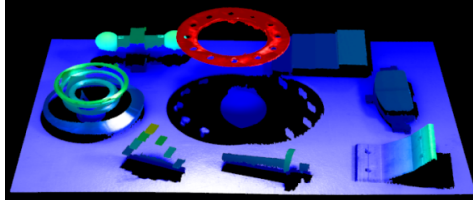


The reference frame setting is simultaneously applied to both depth map and point cloud.


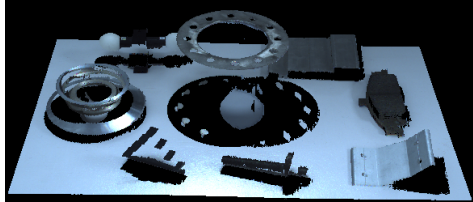
Change Color Rendering

In the **Color** menu in the left panel, you can choose the color rendering scheme of the depth data.

The **Jet**, **Grayscale**, and **Jet + intensity** options are available for both the depth map and point cloud. When one of these three options is selected, the [depth range of the color rendering can be adjusted](#).

Option	Description	Example
Jet	Depth values are displayed in different colors of the jet color scheme.	
Grayscale	Depth values are displayed in different shades of gray.	
Jet + intensity	The intensity information of the object surface is overlaid on the jet color scheme. Suitable for checking the object surface features and depth changes at the same time.	

The **Untextured** and **Textured** options are only available for the point cloud.

Option	Description	Example
Untextured	Displays the untextured, white point cloud.	
Textured	Display the point cloud textured with the 2D image.	



When **Untextured** or **Textured** is selected, if you switch to the depth map display and then switch back to the point cloud display, the **Color** setting will be automatically changed to the option selected in the depth map display.

Adjust Depth Range of Color Rendering

When the depth map and point cloud are displayed in **Jet**, **Grayscale** or **Jet + intensity**, every time data acquisition is performed, Mech-Eye Viewer automatically adjusts the color rendering

according to the range of the obtained depth data.

With the functions in the **Range** section of the left panel, you can adjust the depth range in which the color is rendered, in order to conveniently observe the depth variation in a specific range.



The depth range of color rendering is simultaneously applied to both depth map and point cloud.

Examples of suitable scenarios:

- Observe the depth variation of an area on the workpiece surface to check if this area is flat enough.
- Accentuate the display of a certain workpiece.

Follow these steps to adjust the color rendering range of the depth data:

1. Adjust the **Depth Range** parameter in the **Parameters** tab: **Min** and **Max** of the slider in the left panel corresponds to the **Depth Range** parameter. You can first [adjust this parameter to an appropriate range](#) to eliminate the unneeded depth data.
2. Check the rough depth: On the depth map, move the cursor to the area to be observed, and check the third value of **Coordinates** in the image information box in the lower right. This is the depth value.
3. Adjust depth range of color rendering: Move the slider handles or enter the depth values on the right of the slider to adjust the depth range of color rendering according to your needs.



- When using the UHP series, you can change the unit to μm on the right of the slider.
- After adjusting the range, the depth values outside this range are displayed in the colors corresponding to the minimum/maximum depth values.

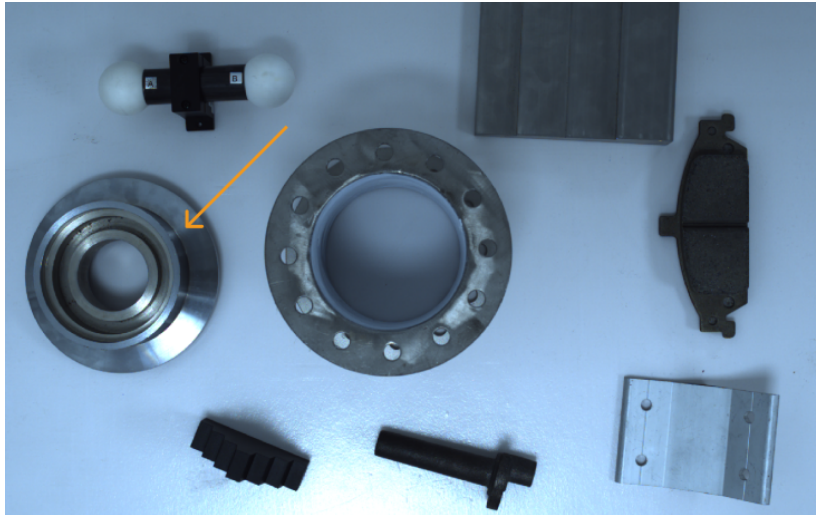
4. Lock the range: If you need to compare the depth variations in the same range across multiple acquisitions, you can check the **Lock** option above the slider. When you acquire data again, the depth range of color rendering manually adjusted will be used.



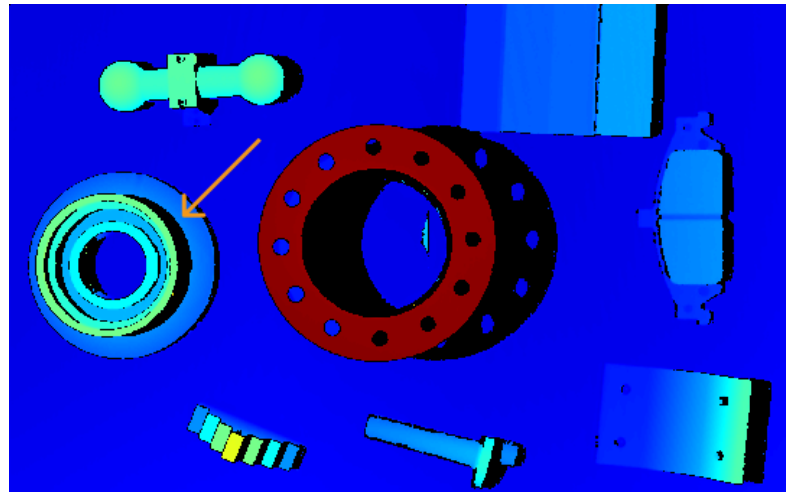
To change back to the automatically adjusted range, uncheck **Lock** and click [**Reset**] to the upper right.

▼ Adjustment example for suitable scenario 1

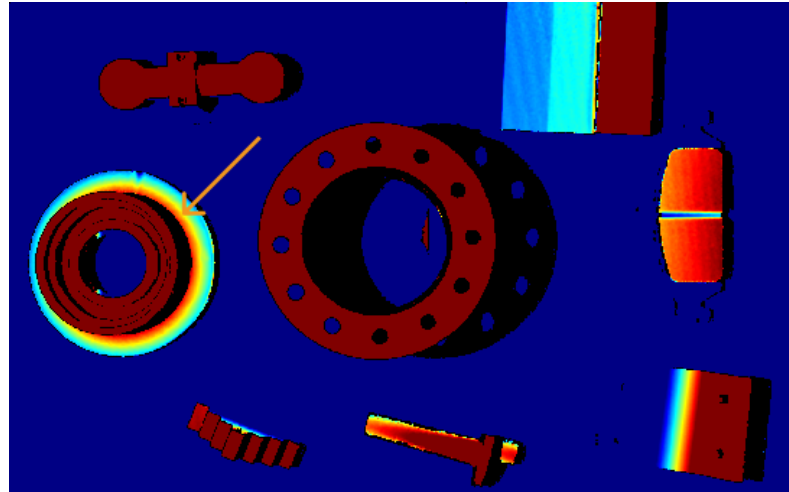
Goal: Check whether the surface pointed by the orange arrow is flat enough.



1. In the depth map, this surface appears to be roughly the same color. It is hard to tell if the surface is perfectly flat or not.

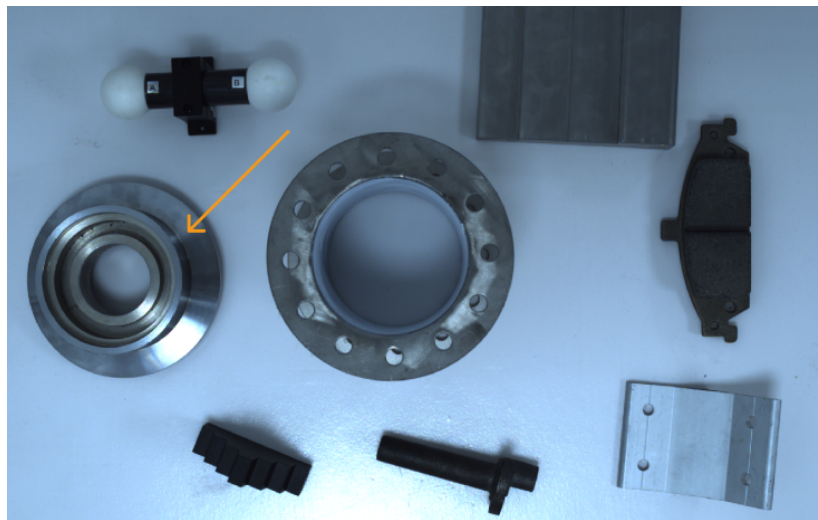


2. Move the cursor onto the surface and check the depth value in the image information box in the lower right (about 0.829 m).
3. Move the slider handles or enter the depth values to adjust the depth range of color rendering to 825–835 mm.
4. Check the depth map again. Now the depth variation of this surface can be easily observed with the color variation.

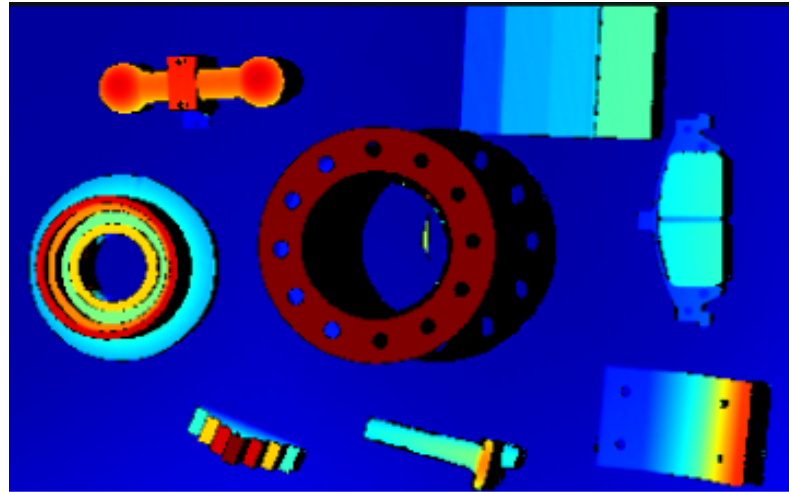


▼ Adjustment example for suitable scenario 2

Goal: Accentuate the display of the workpiece pointed by the orange arrow.



1. On the depth map, move the cursor to the highest and lowest part of the workpiece, and check the depth values in the image information box in the lower right (about 0.795 m and 0.829 m).
2. Move the slider handles or enter the depth values to adjust the depth range of color rendering to 790–850 mm.
3. Check the depth map again. Now the workpiece is displayed in more colors, which makes it stand out.



Point Cloud Exhibit

This function moves the point cloud along a predetermined trajectory to display it from various angles, suitable for scenarios such as exhibits.

Follow these steps to use the **Point Cloud Exhibit** function:

1. Check **Point Cloud Exhibit** in the **View** menu, and a **[Play]** button will appear in the upper right of the point cloud display.
2. Click **[Play]**, and the point cloud will start moving along the predetermined trajectory.
3. While the point cloud is moving, the **[Play]** button is changed to **[Stop]**. Click **[Stop]**, and the point cloud will stop moving.

This topic introduced data acquisition and data types. The next topic will introduce actions related to parameters.

5.2.3. Adjust Parameters

This topic introduces the actions related to parameters, including the general workflow of adjusting parameters, managing parameter groups, changing parameter visibility, and checking parameter descriptions.


General Workflow of Adjusting Parameters

When adjusting the parameters, you usually need to acquire the data again to check the result of parameter adjustment. The general workflow of adjusting parameters is as follows:

1. At the top of the **Parameters** tab, [create a parameter group](#), or select a parameter group for saving the parameter values.



You can select a built-in parameter group based on your actual scenario to simplify the process of parameter adjustment. For detailed descriptions, please refer to [Built-in Parameter Groups](#).

2. Click  to acquire data and check the data quality.
 - 2D image: not too bright or too dark; the surface features of the target object are clearly

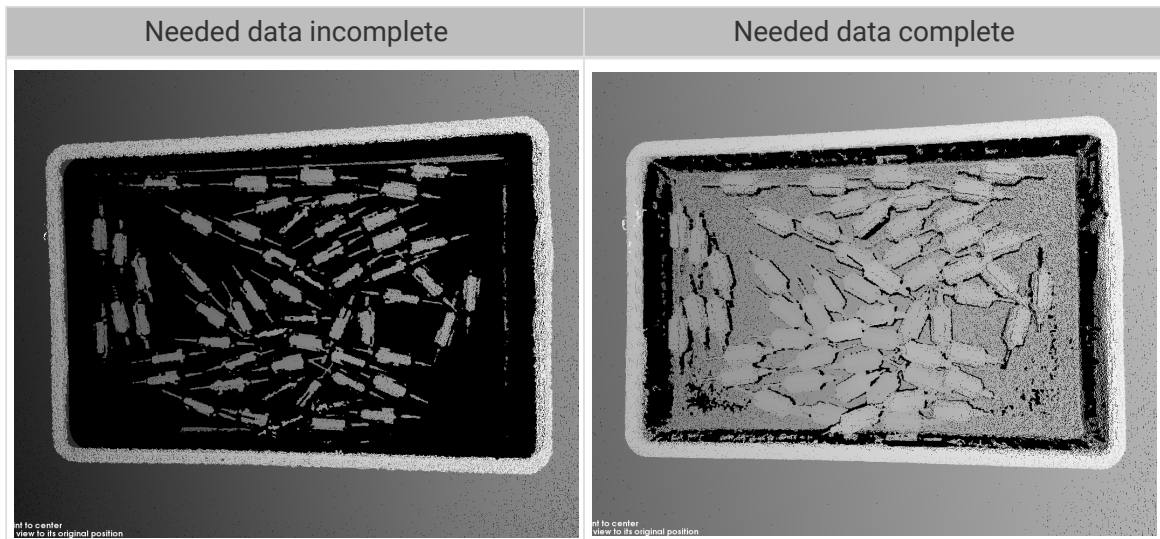
visible.



- Depth map and point cloud: the needed data is complete. In the following example, the needed data is the rim of the bin and the rotors inside the bin.



Determine which part of the data is needed according to your actual needs. For example, if you need to pick metal bowls with opening facing up by the rim, usually you only need to make sure that the data of the bowl rims is complete.



3. If the data quality is not good, adjust the corresponding parameters in the **Parameters** tab on the right.
 - If the quality of the 2D image is not good, adjust the parameters in the **2D Parameters** category.
 - If the quality of the depth map is not good, adjust the parameters in the **3D Parameters**, **Depth Range**, and **ROI** categories.
 - If the quality of the point cloud is not good, adjust the parameters in the **3D Parameters**, **Point Cloud Processing**, **Depth Range**, and **ROI** categories.



- When adjusting the parameters, you can refer to the [parameter description](#) at the bottom of the **Parameters** tab.
- For the detailed descriptions of each parameter, please refer to [Parameter Reference Guide](#).

4. Repeat steps 2 to 3 until the data quality meets the requirements, and then click **[Save]** at the top of the **Parameters** tab.



- Adjustment of the parameters in the **Point Cloud Processing**, **Depth Range**, and **ROI**

categories takes effect immediately. You do not need to acquire the data again.

- If the data quality is still not good, you can [change the visibility](#) and adjust the newly displayed parameters.


Manage Parameter Groups

Parameter groups are used to save parameter values for different applications, projects, workpieces, and so on. When the application, project, workpiece, etc., changes, switching the parameter group in use can quickly apply the corresponding parameter values.

In Mech-Eye Viewer, you can create a new parameter group, save parameter values to a parameter group, delete a parameter group, rename a parameter group, and import and export parameter groups.

Create a parameter group

Please follow these steps to create a parameter groups using the current parameter settings:

1. On the right of the Parameter Group section, click .
2. In the pop-up window, enter the name for the parameter group, and click [**Confirm**].



The parameter group name cannot be longer than 32 characters.

3. After the parameter group is created, it will be displayed in the drop-down menu in the Parameter Group section.

Save Parameters to Parameter Group

Follow these steps to save parameters to a parameter group:


1. Click the drop-down menu in the Parameter Group section, and select the parameter group used for saving parameter values.
2. [Adjust parameters](#).
3. In the Parameter Group section, click [**Save**] or use the shortcut **Ctrl** + **S** to save the adjusted parameter values to the currently selected parameter group.



An asterisk (*) next to a parameter name indicates that the value of this parameter has not been saved to the parameter group. Unsaved changes will be discarded after the camera is disconnected from power.

Delete a parameter group

To delete a no longer needed parameter group, follow these steps:


1. Click the drop-down menu in the Parameter Group section, and select the parameter group to be deleted.
2. Click , and then click [**Confirm**] in the pop-up window to delete the parameter group.



The built-in parameter groups cannot be deleted.

Rename Parameter Group

Follow these steps to rename a parameter group:

1. Click the drop-down menu in the Parameter Group section, and select the parameter group to be renamed.
2. Click , enter the new name in the pop-up window, and then click [**Confirm**].



The parameter group name cannot be longer than 32 characters.

3. After the change, the new name will be displayed in the drop-down menu of the Parameter Group section.



The built-in parameter groups cannot be renamed.

Export parameter groups

You can export all the parameter groups saved in the current camera to a **JSON** file. Follow these steps to export parameter groups:

1. In the Parameter Group section, click [**Export**], select the file path, and then click [**Save**].
2. If the parameter groups are successfully exported, the **Successfully exported parameter groups** window will pop-up. Click [**Confirm**] in the window to close this window.

Import Parameter Groups

You can import parameter groups from a **JSON** file to replace all the parameter groups saved in the current camera. Follow these steps to import parameter groups:

1. In the Parameter Group section, click [**Import**], find the **JSON** file that stores all the parameter group information, and click [**Open**].
2. If the parameter groups are successfully imported, the **Successfully imported parameter groups** window will pop-up. Click [**Confirm**] in the window to close this window.

Reset Parameter Group

To reset the parameter values saved in a parameter group to default values, follow these steps:

1. Click the drop-down menu in the Parameter Group section, and select the parameter group to be reset.
2. Click [**Reset**], and then click [**Yes**] in the pop-up window to reset the parameter values to those stored in the **default** parameter group.

Change Visibility

The visibility of parameters in Mech-Eye Viewer includes three levels: **Beginner**, **Expert**, and **Guru**. Select from the drop-down menu of **Visibility** to change the visibility.

- The **Beginner** level includes the most frequently used basic parameters.
- The **Expert** level includes all parameters in the **Beginner** level and provides more parameters.
- The **Guru** level includes all parameters in the **Expert** level and provides more parameters.

Usually, adjusting parameters in the **Beginner** level is enough to meet your needs. If the data quality is still not satisfactory, you can adjust the parameters in the **Expert** level. In rare cases, the parameters in the **Guru** level need to be adjusted to meet special needs.



The **Guru** level is available in the **Administrator** account. If needed, please contact Technical

Support.

Check Parameter Description

After you select a parameter in the parameter list, the bottom of the **Parameters** tab displays more information about this parameter, include parameter explanation, API parameter name, value range, default value, visibility and data type. When you adjust parameters, you can use the information here as a reference.


- If the Parameter Description section blocks the parameter list, click [×] at the upper right of this section to close it.
- To display the Parameter Description section again, check the **Parameter Description** option in the **View** menu.

5.2.4. Save Data

Through this function, you can save the acquired 2D image, depth map, and point cloud to local.

Instructions

Please follow these steps to save data:

1. Click  in the data acquisition area to open the **Save Data** window.
2. Select the data types to be saved under **Data type**. For detailed descriptions, please refer to [Data Types and Formats](#).
3. (Optional) change **File index**: If the selected **Save to** path already contains data saved through Mech-Eye Viewer, you can change the file index to avoid files with the same names being replaced.



If you need to replace the files with the same names, check the **Replace file with the same name** option.

4. Click [...] to the right of **Save to** to select the folder for saving the data.
5. Click [**Save**], and the **Data saved successfully** window will pop up. Click [**Show in folder**] in this window to open the folder where the data is saved and check the data.

Data Types and Formats

When saving the data, you can select the file format to be saved, as well as the data structure and color of the point cloud.

- 2D image: can be saved in the **PNG** (.png) or **JPEG** (.jpg) format.
- Depth map: can be saved in the **PNG** (.png) or **TIFF** (.tiff) format.
- Point cloud: can select the file format, data structure, and color to be saved.

Format	<ul style="list-style-type: none"> • PLY (.ply) • PCD (.pcd) • CSV (.csv)
--------	---

Data structure	<ul style="list-style-type: none"> • Organized: The point cloud resembles a two-dimensional matrix, which corresponds to the actual spacial relationship between points. • Unorganized: The point cloud resembles a one-dimensional list.
Color	<ul style="list-style-type: none"> • Jet • Grayscale • Jet + intensity • Untextured • Textured

5.2.5. Manage Log

If you encounter a problem when using the camera, you can view the log or export the log and send it to Technical Support to facilitate the troubleshooting process.

View Log

Please follow these steps to view the log:

1. Click [**Show Log**] in the data acquisition area to open the log saved in the camera.
2. Click [**Show Log Entries**] in the lower right to display the full log list.
3. Select a log entry in **Log Entries** to view its content on the left.



The title of the log entry represents the time when it was generated. For example, the log entry titled **20221117171503_887** was generated on November 17, 2022 at 17:15:03.

Export Log

Please follow these steps to export the log:

1. Click [**Show Log**] in the data acquisition area to open the log saved in the camera.
2. Click [**Show Log Entries**] in the lower right to display the full log list.
3. Select the log entry to be exported, click [**Export**], select the folder for saving the log in the pop-up window, and then click [**Select Folder**].
4. If you need to export multiple log entries, repeat step 3.

Log Levels

Logs are categorized into four levels:

- i: INFO, descriptive information of execution processes.
- W: WARNING, warning messages of potential issues.
- C: CRITICAL, errors that may not stop the device from operating.
- F: FATAL, errors that severely affects the operation of the device.

Other Actions

You can also perform the following actions in the log window:

- Auto Refresh: automatically refresh the log when enabled.
- Clear: clear the log content displayed in the software.
- Clear Server Logs: delete the log saved in the camera. This action clears the log content displayed in the software at the same time.

5.3. Parameter Reference Guide

This topic provides detailed descriptions of each parameter.

Detailed descriptions of the parameters by camera models:

- [Parameters of NANO](#)
- [Parameters of PRO S and PRO M](#)
- [Parameters of PRO XS](#)
- [Parameters of UHP-140](#)
- [Parameters of DEEP / LSR Series](#)
- [Parameters of V3 Cameras](#)

5.3.1. Parameters of NANO

This topic introduces the parameters of NANO. The parameters are divided into parameters for the 2D image and parameters for the depth map / point cloud, based on the data types that they affect.

Parameters for the 2D Image

The 2D image should not be too bright or too dark, and the surface features of the target object should be clearly visible in the image.

The quality of the 2D image is affected by parameters in the **2D Parameters** category and the **Gain** parameter in the **3D Parameters** category.



If you are using a color camera, and the color of the image differ significantly from that of the objects, you can adjust the **white balance** setting. For detailed instructions, please refer to [Adjust White Balance](#).

2D Parameters

Exposure Mode

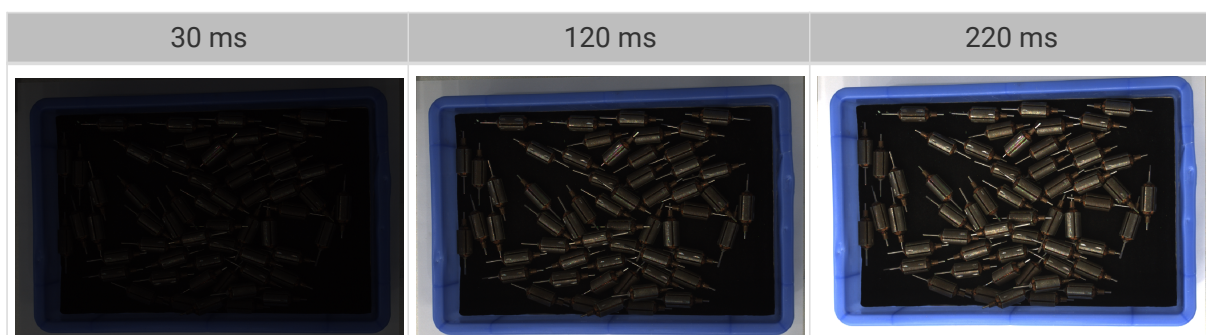
Description	Set the exposure mode for capturing the 2D image.
Visibility	Beginner, Expert, Guru

Values	<ul style="list-style-type: none"> • Timed: Set a single exposure time. Usually used in stable lighting conditions. • Auto: The exposure time is automatically adjusted. Usually used in varying lighting conditions. • HDR: Set multiple exposure times and merge the images. Usually used for objects with various colors or textures. • Flash: Use the projector for supplemental light. Usually used in dark environments.
Instructions	<p>After selecting different options, different parameters are displayed in the 2D Parameters category for adjustment:</p> <ul style="list-style-type: none"> • Timed: displays Exposure Time. • Auto: displays Gray Value and Auto-Exposure ROI. • HDR: displays Tone Mapping and Exposure Sequence. • Flash: no other parameters need to be adjusted. The projector projects supplemental light automatically during image capturing.

Timed: Exposure Time

Description	<p>This parameter affects the brightness of the image.</p> <ul style="list-style-type: none"> • Long exposure time increases the image brightness. • Short exposure time decreases the image brightness.
Visibility	Beginner, Expert, Guru
Values	0.1 to 999 ms
Instructions	<p>Adjust this parameter based on the quality of the 2D image. The 2D image should not be too bright or too dark, and the surface features of the target object should be clearly visible in the image.</p> <ul style="list-style-type: none"> • Long exposure time is usually used in dark environments. • Short exposure time is usually used in bright environments.

2D images obtained with different **Exposure Time** values (all other conditions identical):

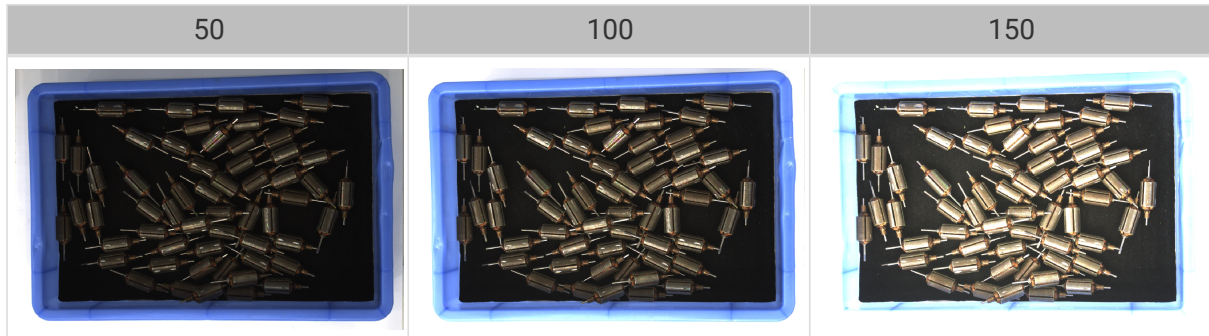


Auto: Gray Value

Description	This parameter affects the brightness of the 2D image. Increase the value if the 2D image is too dark and decrease if too bright.
-------------	---

Visibility	Beginner, Expert, Guru
Values	0 to 255
Instruction	N/A

2D images obtained with different **Gray Value** values (all other conditions identical):



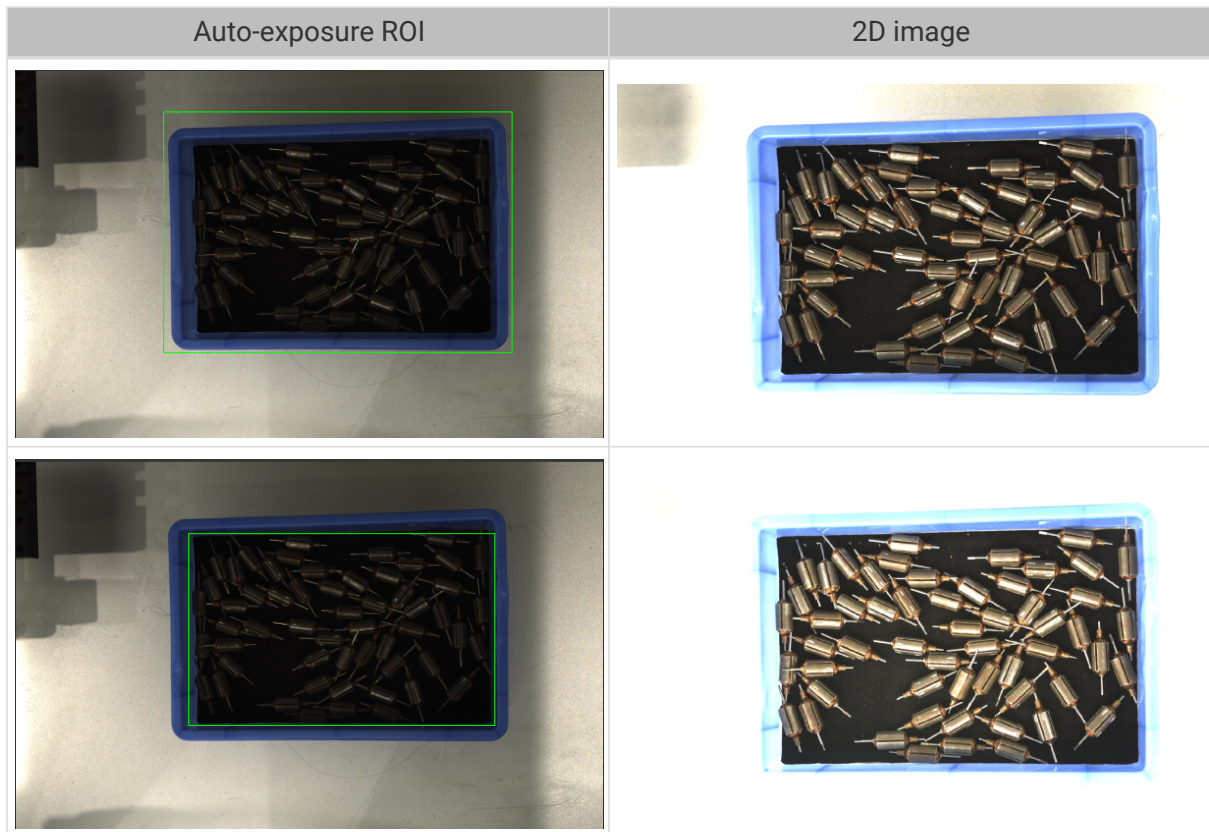
The **Gray Value** of a monochrome image is equivalent to the image brightness; the **Gray Value** of a color image is equivalent to the brightness of each color channel.

Auto: Auto-Exposure ROI

Description	<ul style="list-style-type: none"> The exposure time is automatically adjusted based on the lighting, object colors, etc., in the selected region. When an auto-exposure ROI is not set, the exposure time is automatically adjusted based on the entire camera FOV.
Visibility	Beginner, Expert, Guru
Values	N/A
Instruction	For detailed instructions, refer to Set Auto-Exposure ROI below.

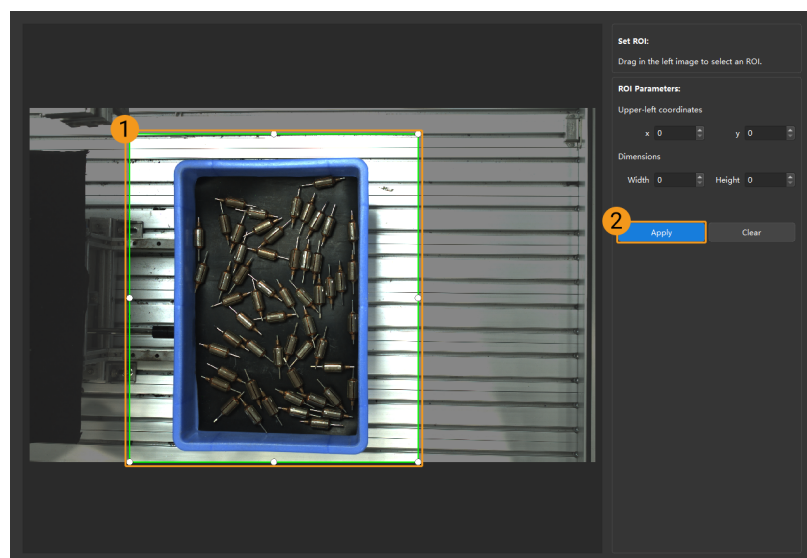
2D images obtained with different **Auto-Exposure ROI** values (all other conditions identical):





Set Auto-Exposure ROI

1. Double-click [**Edit**] to the right of **Auto-Exposure ROI** to open the **Set ROI** window.
2. Select and adjust the ROI on the left. Drag the selection box to move it. Drag the handles on the box to adjust the box size.
3. Click [**Apply**] to apply the set auto-exposure ROI.



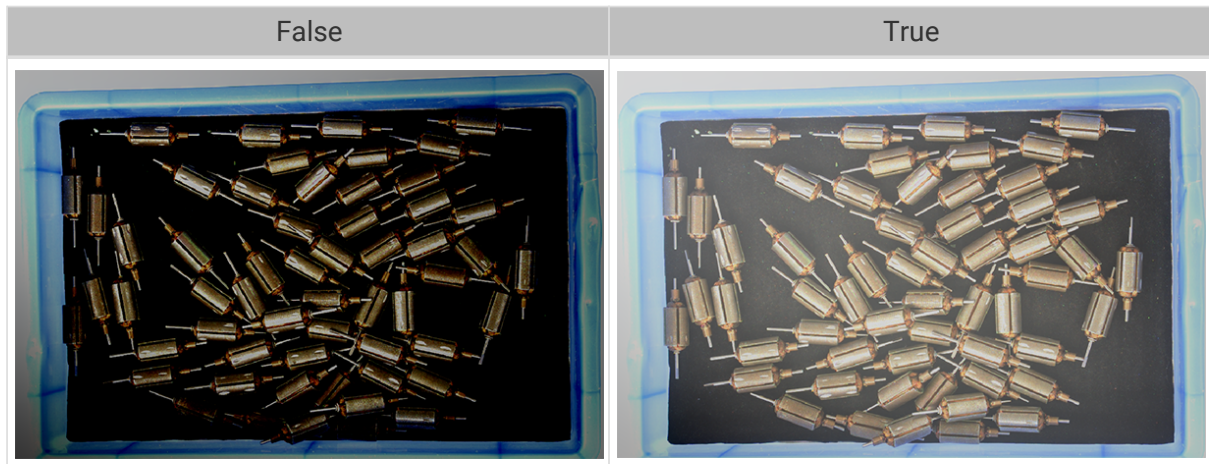
-  Click [**Clear**] to remove the current auto-exposure ROI.

4. Acquire data again and view the 2D image to check the result of automatic exposure adjustment.


HDR: Tone Mapping

Description	This function can make the image look more natural. If the 2D image appears very different from the actual objects, please enable this function.
Visibility	Beginner, Expert, Guru
Values	<ul style="list-style-type: none"> • False • True
Instructions	Check to enable Tone Mapping .

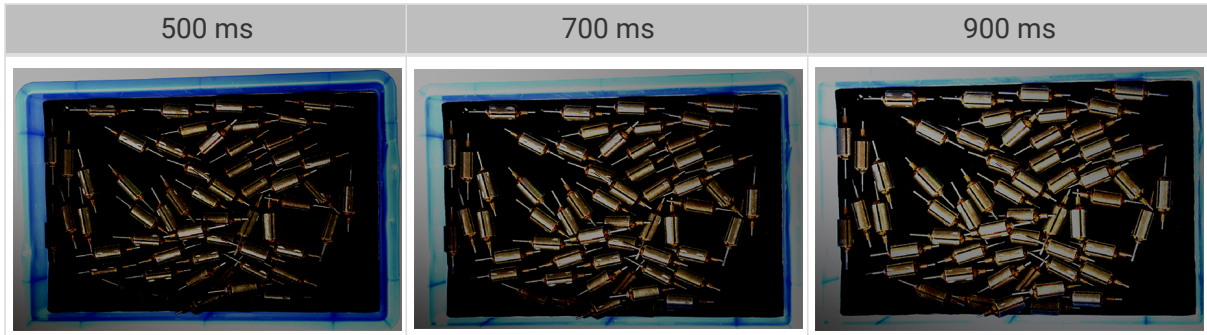
2D images obtained with **Tone Mapping** disabled or enabled (all other conditions identical):



HDR: Exposure Sequence

Description	Set multiple exposure times, and the captured images are merged to generate a 2D image that retains more details in the highlights and shadows.
Visibility	Beginner, Expert, Guru
Values	N/A
Instructions	<ol style="list-style-type: none"> 1. Double-click [Edit] to the right of Exposure Sequence to open the editing window of the exposure sequence. 2. Click [+] to add and set a new exposure time. 3. To delete an exposure time, select it and click [-]. 4. After you finish adding the exposure times, click [Apply] to apply the exposure sequence. <div style="display: flex; align-items: center; margin-top: 10px;"> <div style="margin-right: 10px;">  </div> <ul style="list-style-type: none"> ◦ [Cancel]: do not save the current settings and exit this window. ◦ [Reset]: clear all the exposure times. </div>

2D images captured with single exposure times (all other conditions identical):




2D images captured with different exposure sequences consisting of the above exposure times (all other conditions identical):



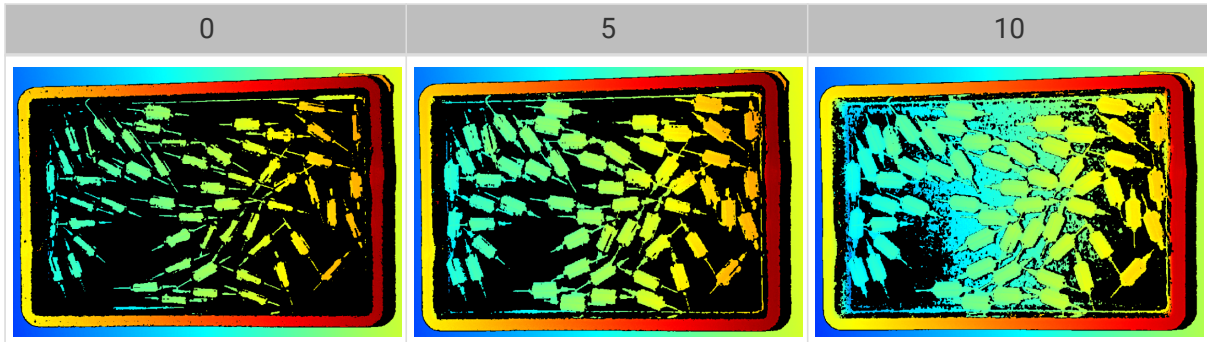
3D Parameters

Gain

Description	This parameter affects the brightness of the captured image. At the same time, more noise is introduced.
	 This parameter affects the quality of the depth map and point cloud by changing the brightness of the images used for calculating depth data .
Visibility	Expert, Guru
Values	0 to 16 dB
Instructions	When the expected brightness cannot be reached by adjusting Exposure Time , adjust this parameter.

2D images and depth maps obtained with different **Gain** values (all other conditions identical):





Parameters for the Depth Map / Point Cloud

In the depth map and point cloud, the needed data should be complete.



Determine which part of the data is needed according to your actual needs. For example, if you need to pick metal bowls with opening facing up by the rim, usually you only need to make sure that the data of the bowl rims is complete.

Parameters in the following categories affect the quality of the depth map and point cloud.

Parameter category	Depth map	Point cloud
3D Parameters	✓	✓
Point Cloud Processing		✓
Depth Range	✓	✓
ROI	✓	✓

3D Parameters

The parameters in this category affect the images used for calculating depth data, thus affecting the quality of the depth map and point cloud.

Use **Exposure Assistant** can obtain the recommended exposure parameter values. Double-click [Auto Set] to the right of **3D Parameters** to open **Exposure Assistant**.

Exposure Multiplier

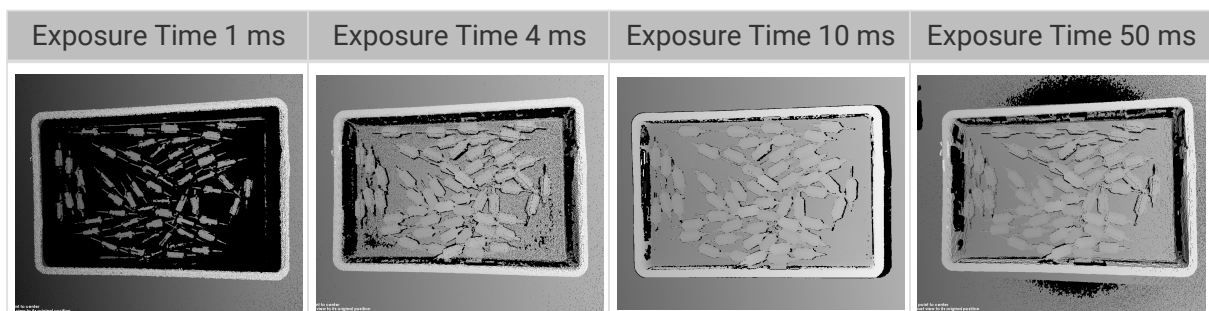
Description	Set the number of Exposure Time .
Visibility	Beginner, Expert, Guru
Values	1 to 3

Instruction	<ul style="list-style-type: none"> When the value of Exposure Multiplier is greater than 1, multiple Exposure Time parameters must be set. All images captured with different exposure times are used for calculating depth. Using multiple exposure times can improve the completeness of depth data but also increases processing time. With multiple exposure times, the amount of time for obtaining the depth map and point cloud increases. Please try to minimize the number of exposure times while ensuring the data quality.
-------------	---

Exposure Time


Description	Set the exposure time for acquiring depth information. The number of Exposure Time to be set depends on the value of Exposure Multiplier .
Visibility	Beginner, Expert, Guru
Values	<ul style="list-style-type: none"> Value range: 0.1 ms to 99 ms
Instructions	<ul style="list-style-type: none"> Usually, long exposure time is used for dark objects, and short exposure time is used for light objects. Exposure times too long or too short both lead to loss of depth information. The exposure time of a laser camera must be a multiple of 4. If the entered value is not a multiple of 4, it will be automatically adjusted. The minimum value that can be set is 4 ms for the Laser series and 8 ms for other laser cameras.

Point clouds obtained under different **Exposure Time** values (all other conditions identical):



The black regions lack depth information to generate corresponding point cloud.

Gain

Description	<p>This parameter affects the brightness of the captured image. At the same time, more noise is introduced.</p> <div style="display: flex; align-items: center;">  <p>This parameter affects the quality of the depth map and point cloud by changing the brightness of the images used for calculating depth data.</p> </div>
Visibility	Expert, Guru
Values	0 to 16 dB
Instructions	When the expected brightness cannot be reached by adjusting Exposure Time , adjust this parameter.

2D images and depth maps obtained with different **Gain** values (all other conditions identical):



Projector

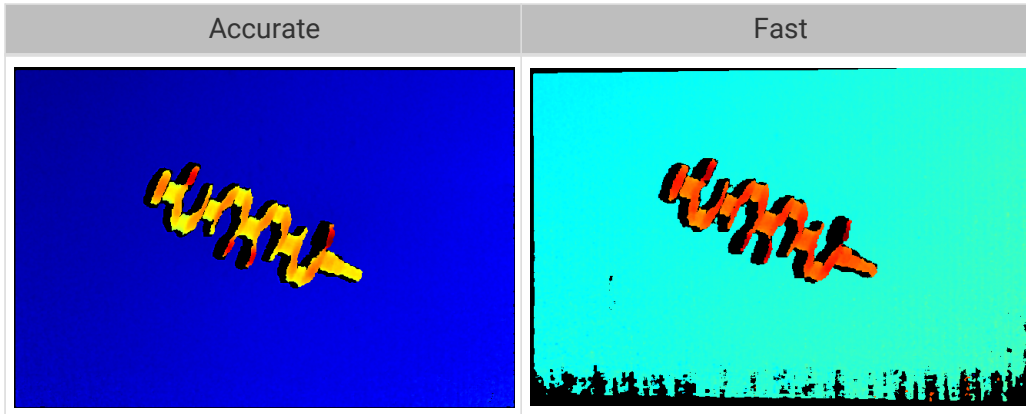
Light Brightness

Description	This parameter affects the brightness of the projected structured light.
Visibility	Expert, Guru
Values	<ul style="list-style-type: none"> • High: high brightness, suitable for dark objects. • Normal: medium brightness, suitable for regular objects. • Low: low brightness, suitable for reflective objects.
Instructions	Adjust this parameter based on the object type.

Fringe Coding Mode

Description	Select the pattern of the structured light to be projected.
Visibility	Expert, Guru
Values	<ul style="list-style-type: none"> • Fast: suitable for opaque objects, provides fast acquisition speed, but the depth data quality is lower. • Accurate: suitable for opaque objects, provides high-quality depth data, but the acquisition speed is slower.
Instructions	Adjust this parameter according to the type of target objects and your actual needs for data quality and data acquisition speed.

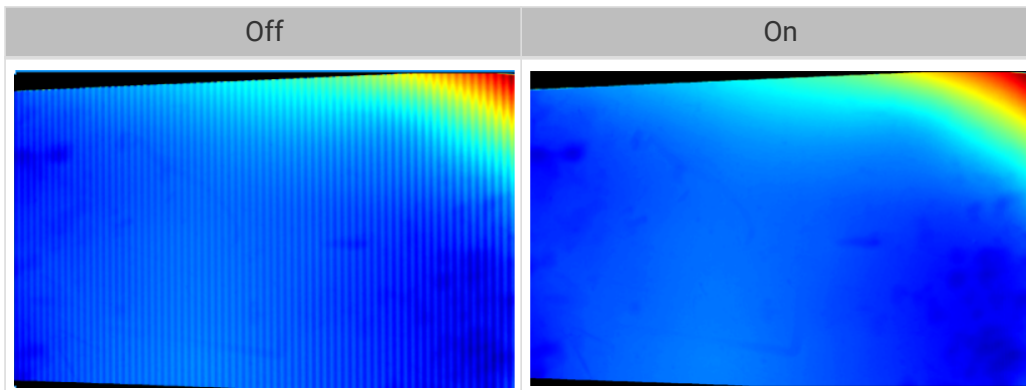
When the target object is not translucent, the point clouds obtained with **Fringe Coding Mode** set to **Accurate** and **Fast** are as follows (all other conditions identical):



Anti-Flicker Mode

Description	Flicker refers to the rapid and periodical change in the intensity of artificial light. This phenomenon can cause fluctuations in the depth data. Such fluctuations can be reduced by adjusting the projection frequency of the structured light.
Visibility	Expert, Guru
Values	<ul style="list-style-type: none"> • Off • AC50Hz • AC60Hz
Instructions	Please select the option that corresponds to the frequency of the power supply in your location. The AC frequency in most countries is 50Hz. The AC frequency in the U.S. and some Asian countries is 60Hz.

Depth maps obtained with **Anti-Flicker Mode** off or on (all other conditions identical):



Point Cloud Processing

Adjusting the parameters in the **Point Cloud Processing** category can improve the quality of the point cloud.

Principles for Adjusting Point Cloud Processing Parameters

When adjusting the **Point Cloud Processing** parameters, follow these principles to reduce the camera acquisition time and cycle time:

1. Adjust **Outlier Removal** first. The calculation time for different intensities of this parameter is

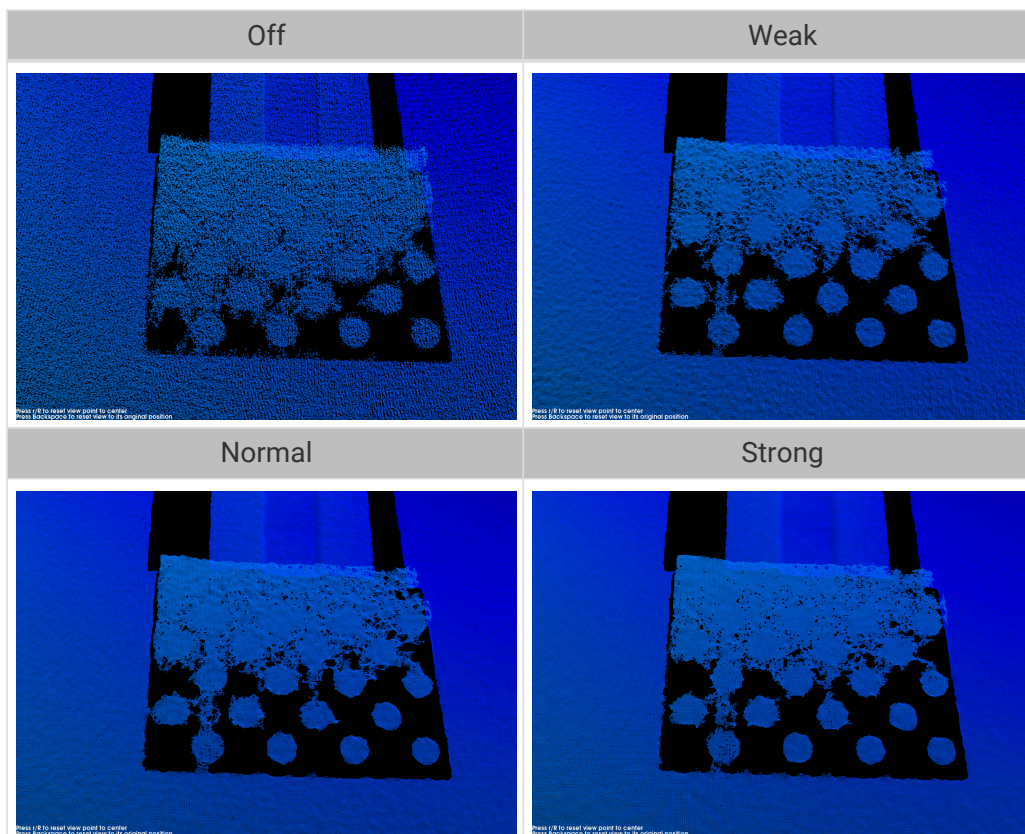
about the same. Using a higher intensity will not add to the calculation time too much.

- It is recommended to use low intensities of **Surface Smoothing** and **Noise Removal**. The higher the intensities of these two parameters, the longer the calculation time.

Surface Smoothing

Description	This parameter reduces the depth fluctuation in the point cloud and improve its resemblance to the actual object surface.
Visibility	Beginner, Expert, Guru
Values	<ul style="list-style-type: none"> • Off • Weak • Normal • Strong
Instructions	<ul style="list-style-type: none"> • Surface Smoothing causes loss of object surface details. Higher intensity causes more object details to be lost. Lower intensity causes fewer object details to be lost. • Higher intensity results in slower calculation, and lower intensity results in faster calculation.

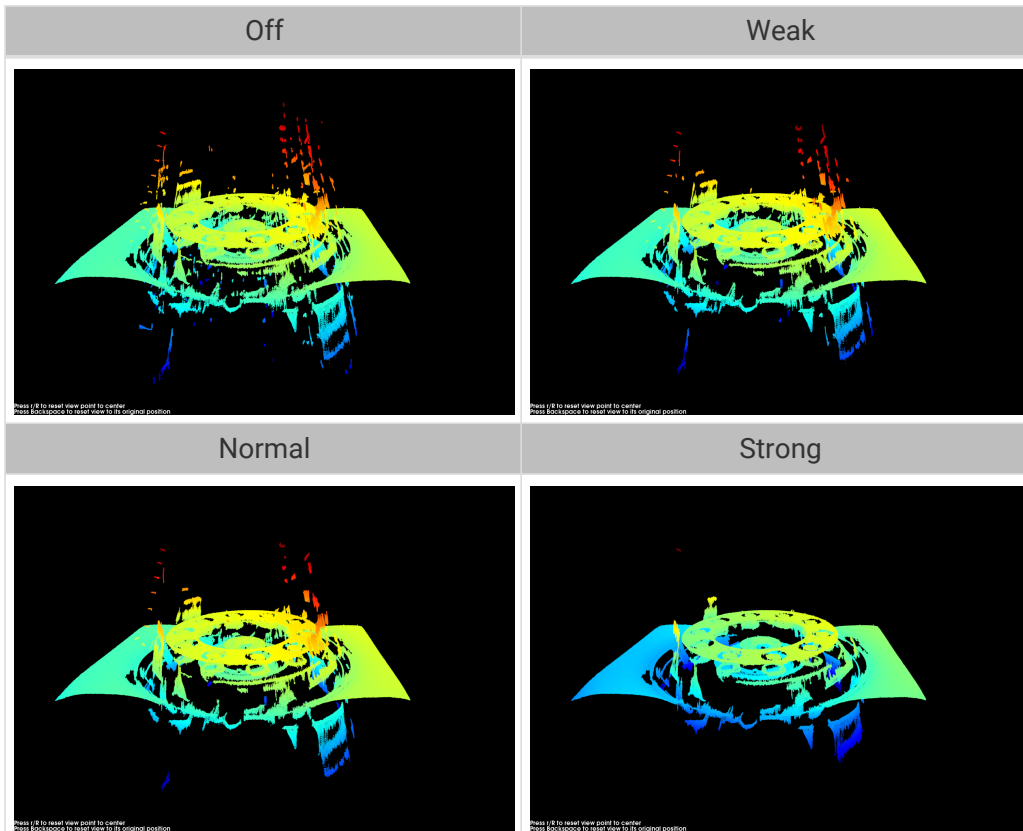
Point clouds obtained with different **Surface Smoothing** values (all other conditions identical):



Outlier Removal

Description	This parameter removes the outliers in the point cloud. Outliers are clustered points away from the object point cloud.
Visibility	Beginner, Expert, Guru
Values	<ul style="list-style-type: none"> • Off • Weak • Normal • Strong
Instructions	<ul style="list-style-type: none"> • Higher intensity removes more outliers, and lower intensity removes fewer outliers. • If the object point cloud contains multiple parts, high intensities of outlier removal might remove some parts. For example, if the object is a cup, outlier removal might remove the point cloud of the handle part.

Point clouds obtained with different **Outlier Removal** values (all other conditions identical):

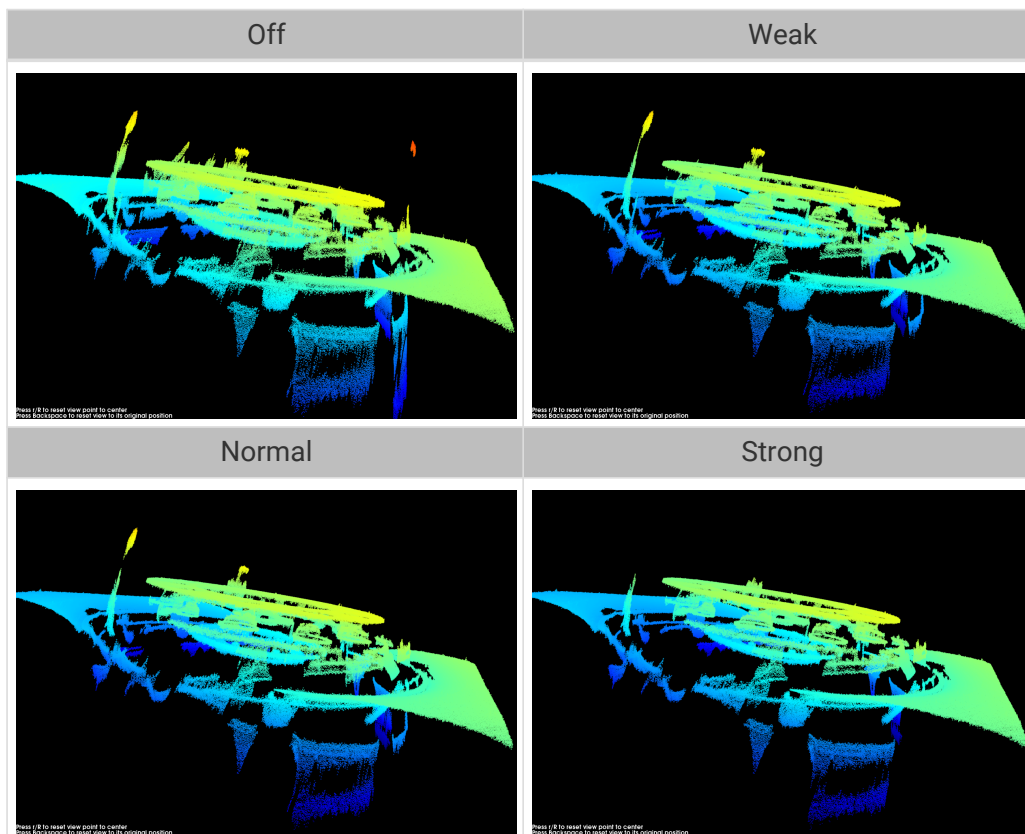


Noise Removal

Description	This parameter removes the noise in the point cloud, thus reducing the impact on the precision and accuracy of subsequent calculation. Noise is the scattered points close to the object surface.
Visibility	Beginner, Expert, Guru

Values	<ul style="list-style-type: none"> • Off • Weak • Normal • Strong
Instructions	<ul style="list-style-type: none"> • Higher intensity removes more noise but may also remove some object features. Lower intensity removes less noise but also preserves the object features better. • Higher intensity results in slower calculation, and lower intensity results in faster calculation.

Point clouds obtained with different **Noise Removal** values (all other conditions identical):



If this function removes the needed object features, please reduce the intensity. However, more noise will be retained.

Edge Preservation

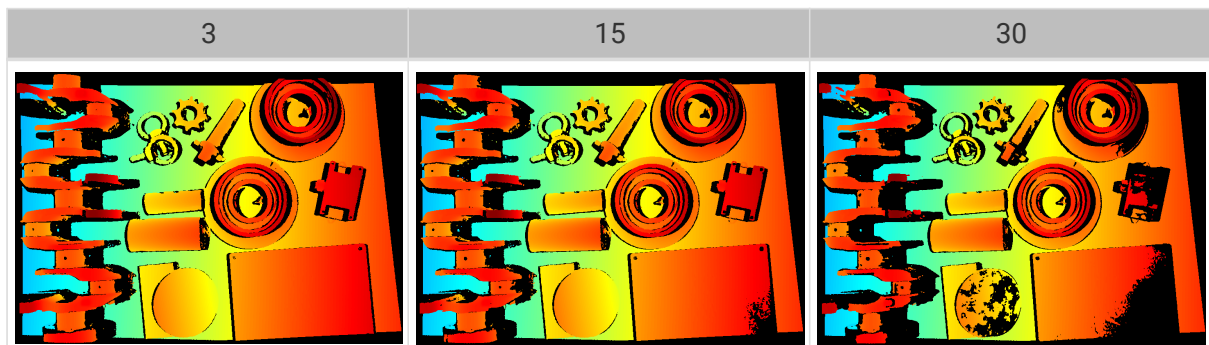
Description	This parameter preserves the sharpness of object edges during surface smoothing.
Visibility	Guru
Values	<ul style="list-style-type: none"> • Sharp: Preserves the sharpness of object edges as much as possible. However, the effect of surface smoothing will be reduced. • Normal: Balances between edge preservation and surface smoothing. • Smooth: Does not preserve the edges. The object surface will be well smoothed, but the object edges will be distorted.

Instructions	Adjust based on your needs for preserving the edges of workpieces.
--------------	--

Stripe Contrast Threshold

Description	This parameter is used to remove noise in the point cloud. Adjust this parameter if the level of noise in the point cloud is still high after adjusting Outlier Removal and Noise Removal .
Visibility	Beginner, Expert, Guru
Values	1 to 100
Instructions	<ul style="list-style-type: none"> • Greater value of this parameter results in more points being removed. Smaller value results in fewer points being removed. • Increase the value of this parameter can remove the noise in the point cloud, but may also cause the points of dark objects to be lost.

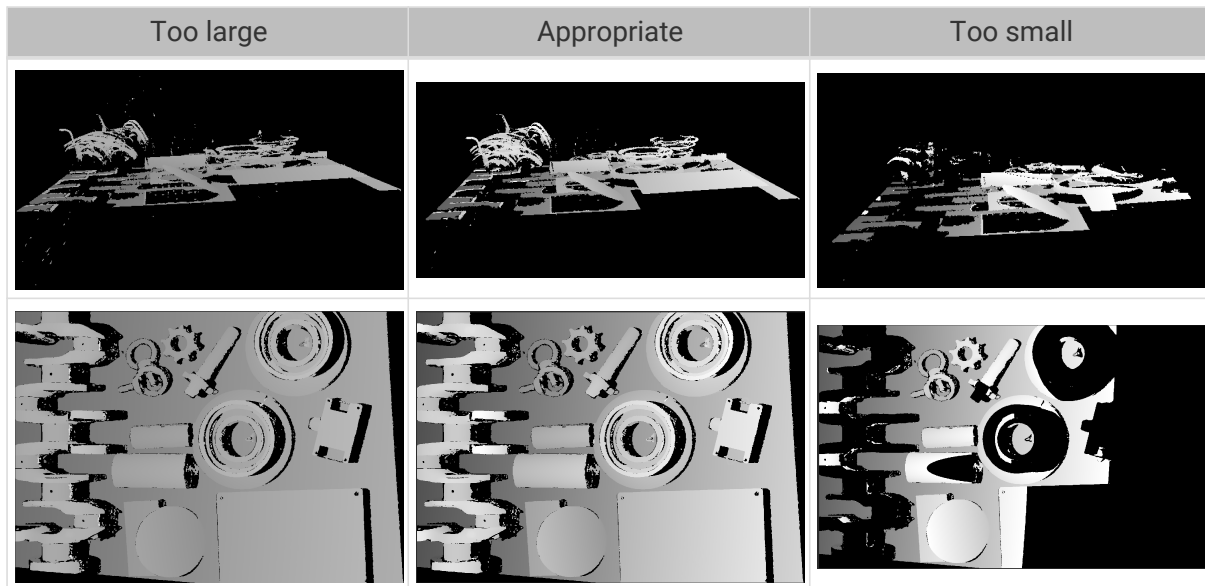
Point clouds obtained with different **Stripe Contrast Threshold** values (all other conditions identical):



Depth Range

Description	Set a Z-direction ROI for the depth map and point cloud. Depth Range can be set within the camera working distance, and the data outside the range is removed from the depth map and point cloud.
Visibility	Beginner, Expert, Guru
Values	<ul style="list-style-type: none"> • Lower Limit: 1 to 4000 mm • Upper Limit: 1 to 5000 mm
Instructions	<ul style="list-style-type: none"> • Depth Range should be adjusted to an appropriate range to ensure the completeness of depth data in the depth map and point cloud. Too large a range may introduce interference, and too small a range may remove the data of key features. • For detailed instructions, please refer to Set Depth Range below.

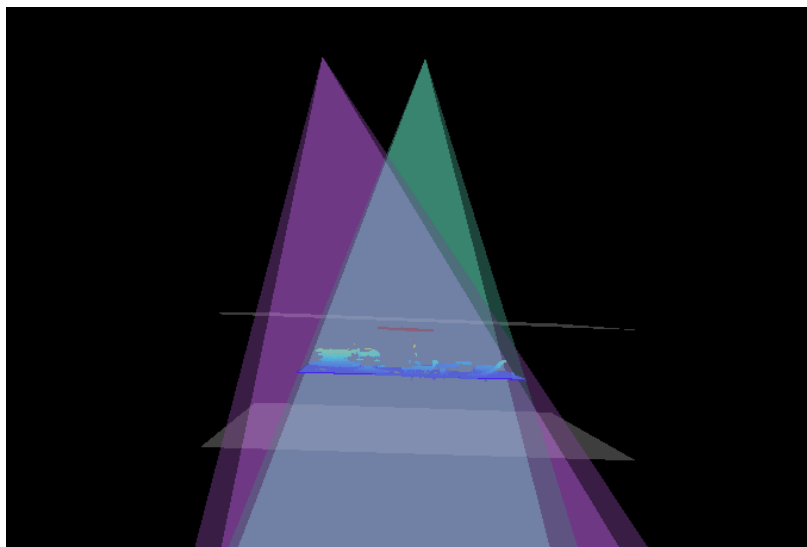
The same point cloud with different **Depth Range** values (all other conditions identical):



Set Depth Range

Follow these steps to adjust Depth Range:

1. Double-click [**Edit**] to the right of **Depth Range** to open the **Set Depth Range** window.
2. Click [**Refresh point cloud**] at the top of the right panel to obtain the newest point cloud.
3. **Adjust the position of the point cloud**: adjust the position until you can see the two gray rectangles that represents the upper and lower limits of **Depth Range**.



4. Adjust the depth range: drag the handles on the slider to adjust **Depth Range** roughly. Then, enter specific values on the right to fine-tune **Depth Range**.



Judge if the depth range is appropriate: All needed object features are located between the two gray rectangles, and most noise and outliers are located outside.

5. Click [**Save**] to save the depth range settings.



- Click [**Recommended range**] to adjust **Depth Range** to the recommended working distance of the current camera.

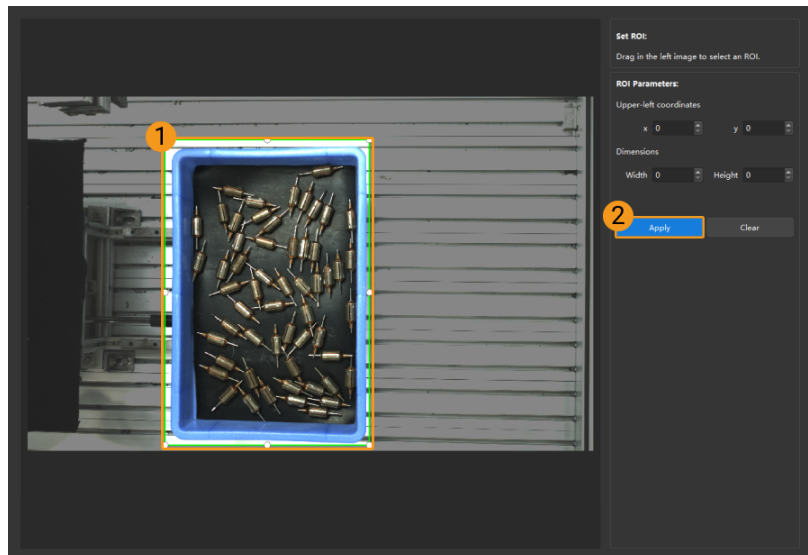
- Click [**Reset**] to restore **Depth Range** to the last saved values.

ROI

Description	Set the ROI on the XY plane for the depth map and point cloud. Points outside the selected region are removed.
Visibility	Beginner, Expert, Guru
Values	N/A
Instructions	For detailed instructions, refer to Set ROI below.

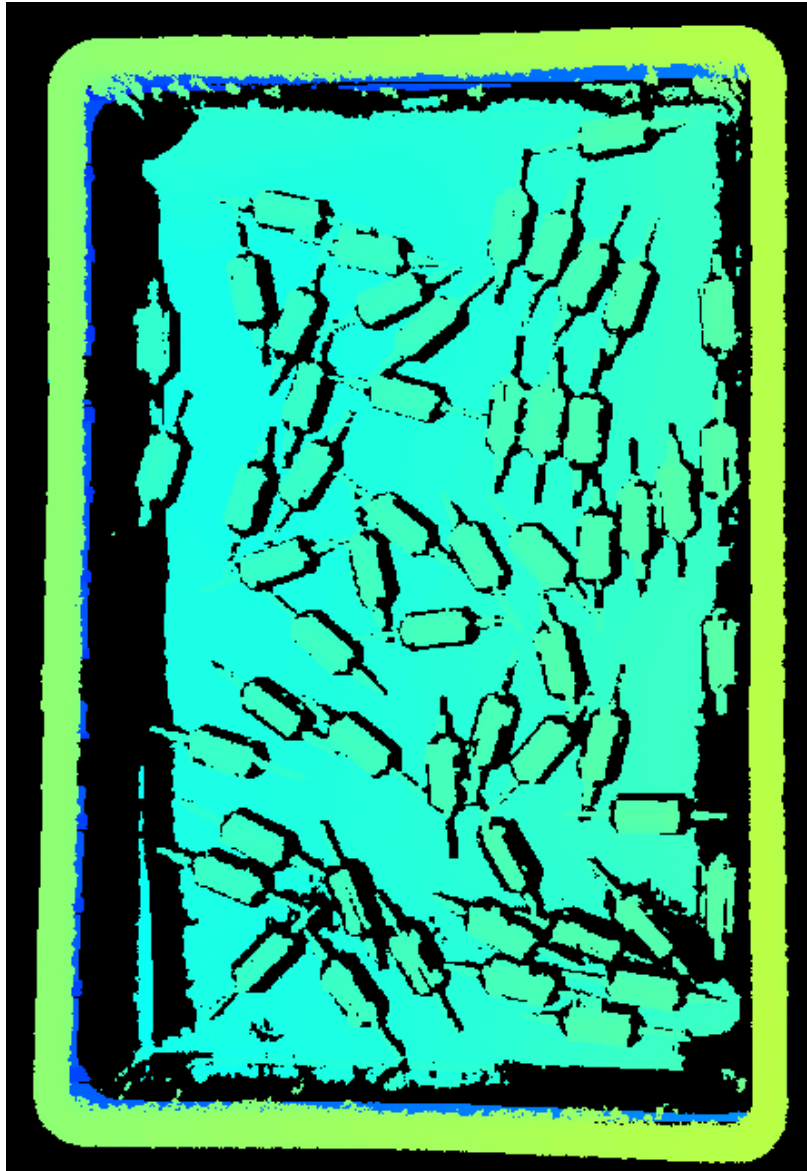
Set ROI

1. Double-click [**Edit**] to the right of **ROI** to open the **Set ROI** window.
2. Select and adjust the ROI on the left. Drag the selection box to move it. Drag the handles on the box to adjust the box size.
3. Click [**Apply**] to apply the set ROI.



- Click [**Clear**] to remove the current ROI.
- For the DEEP (V4) and LSR (V4) series, the 2D image (depth source) is displayed in this window. If the image is too dark or too bright, please adjust **2D Image (Depth Source) Exposure Mode**.

4. Acquire data again and view the depth map or point cloud to check the result of setting the ROI.



5.3.2. Parameters of PRO S and PRO M

This topic introduces the parameters of PRO S and PRO M. The parameters are divided into parameters for the 2D image and parameters for the depth map / point cloud, based on the data types that they affect.

Parameters for the 2D Image

The 2D image should not be too bright or too dark, and the surface features of the target object should be clearly visible in the image.

The quality of the 2D image is affected by parameters in the **2D Parameters** category and the **Gain** parameter in the **3D Parameters** category.



If you are using a color camera, and the color of the image differ significantly from that of the objects, you can adjust the **white balance** setting. For detailed instructions, please refer to [Adjust White Balance](#).

2D Parameters

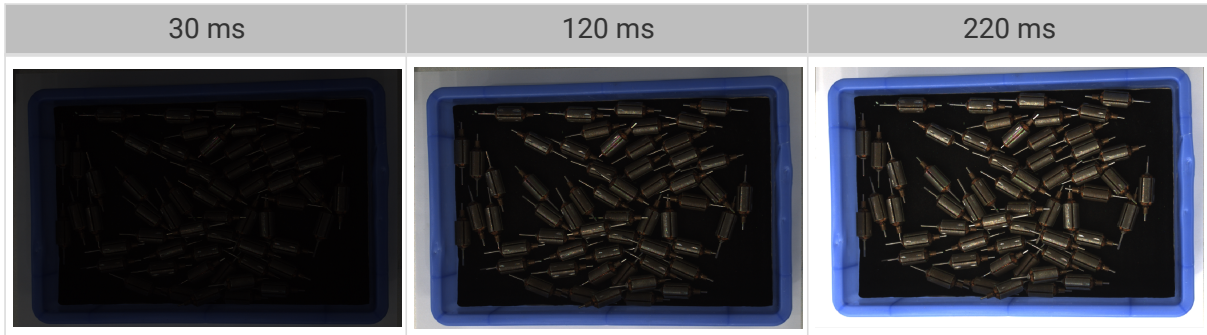
Exposure Mode

Description	Set the exposure mode for capturing the 2D image.
Visibility	Beginner, Expert, Guru
Values	<ul style="list-style-type: none"> • Timed: Set a single exposure time. Usually used in stable lighting conditions. • Auto: The exposure time is automatically adjusted. Usually used in varying lighting conditions. • HDR: Set multiple exposure times and merge the images. Usually used for objects with various colors or textures. • Flash: Use the projector for supplemental light. Usually used in dark environments.
Instructions	<p>After selecting different options, different parameters are displayed in the 2D Parameters category for adjustment:</p> <ul style="list-style-type: none"> • Timed: displays Exposure Time. • Auto: displays Gray Value and Auto-Exposure ROI. • HDR: displays Tone Mapping and Exposure Sequence. • Flash: no other parameters need to be adjusted. The projector projects supplemental light automatically during image capturing.

Timed: Exposure Time

Description	<p>This parameter affects the brightness of the image.</p> <ul style="list-style-type: none"> • Long exposure time increases the image brightness. • Short exposure time decreases the image brightness.
Visibility	Beginner, Expert, Guru
Values	0.1 to 999 ms
Instructions	<p>Adjust this parameter based on the quality of the 2D image. The 2D image should not be too bright or too dark, and the surface features of the target object should be clearly visible in the image.</p> <ul style="list-style-type: none"> • Long exposure time is usually used in dark environments. • Short exposure time is usually used in bright environments.

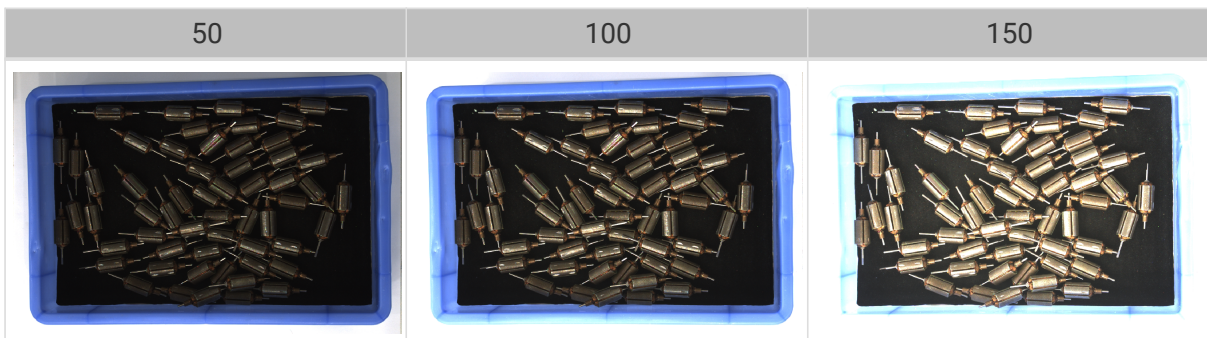
2D images obtained with different **Exposure Time** values (all other conditions identical):



Auto: Gray Value

Description	This parameter affects the brightness of the 2D image. Increase the value if the 2D image is too dark and decrease if too bright.
Visibility	Beginner, Expert, Guru
Values	0 to 255
Instruction	N/A

2D images obtained with different **Gray Value** values (all other conditions identical):

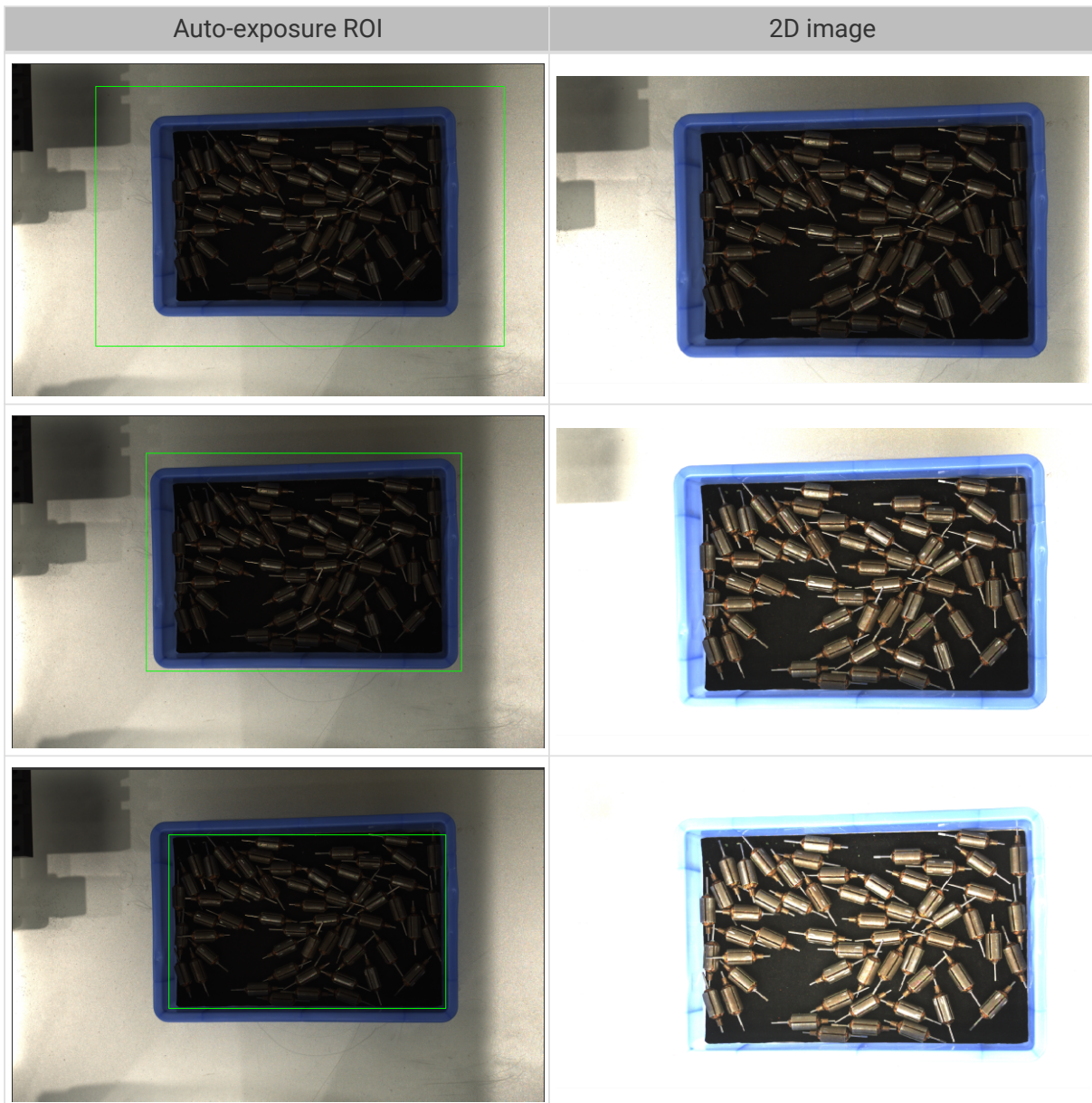


The **Gray Value** of a monochrome image is equivalent to the image brightness; the **Gray Value** of a color image is equivalent to the brightness of each color channel.

Auto: Auto-Exposure ROI

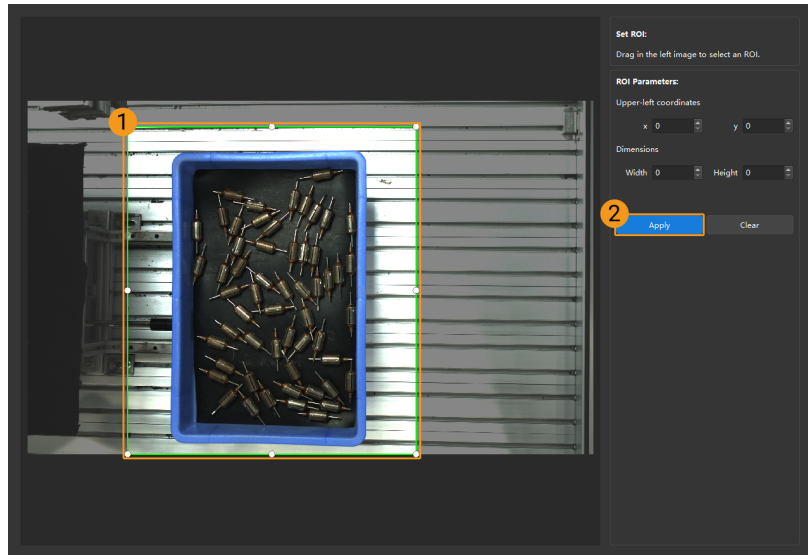
Description	<ul style="list-style-type: none"> The exposure time is automatically adjusted based on the lighting, object colors, etc., in the selected region. When an auto-exposure ROI is not set, the exposure time is automatically adjusted based on the entire camera FOV.
Visibility	Beginner, Expert, Guru
Values	N/A
Instruction	For detailed instructions, refer to Set Auto-Exposure ROI below.

2D images obtained with different **Auto-Exposure ROI** values (all other conditions identical):



Set Auto-Exposure ROI

1. Double-click [**Edit**] to the right of **Auto-Exposure ROI** to open the **Set ROI** window.
2. Select and adjust the ROI on the left. Drag the selection box to move it. Drag the handles on the box to adjust the box size.
3. Click [**Apply**] to apply the set auto-exposure ROI.



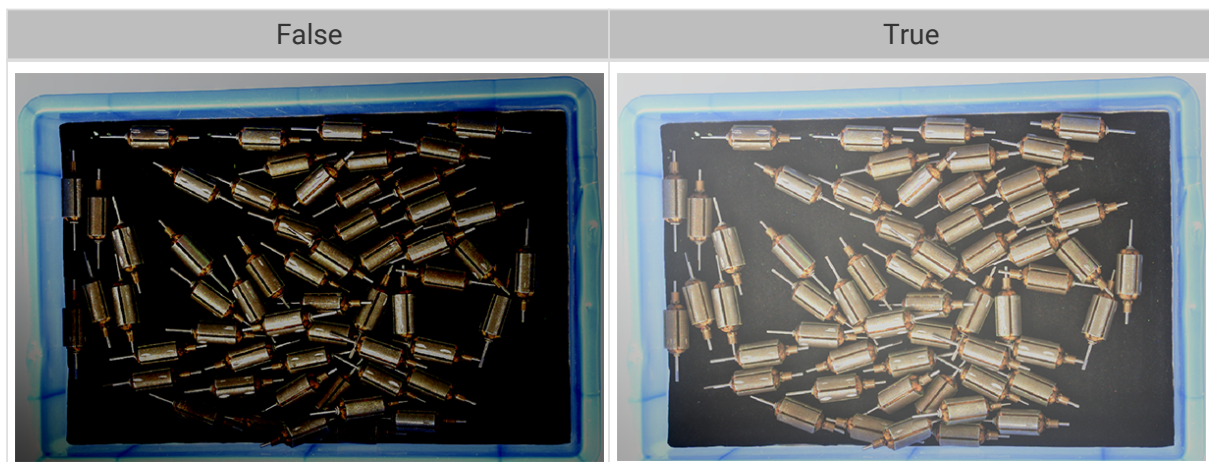
 Click [Clear] to remove the current auto-exposure ROI.

- Acquire data again and view the 2D image to check the result of automatic exposure adjustment.


HDR: Tone Mapping

Description	This function can make the image look more natural. If the 2D image appears very different from the actual objects, please enable this function.
Visibility	Beginner, Expert, Guru
Values	<ul style="list-style-type: none"> • False • True
Instructions	Check to enable Tone Mapping .

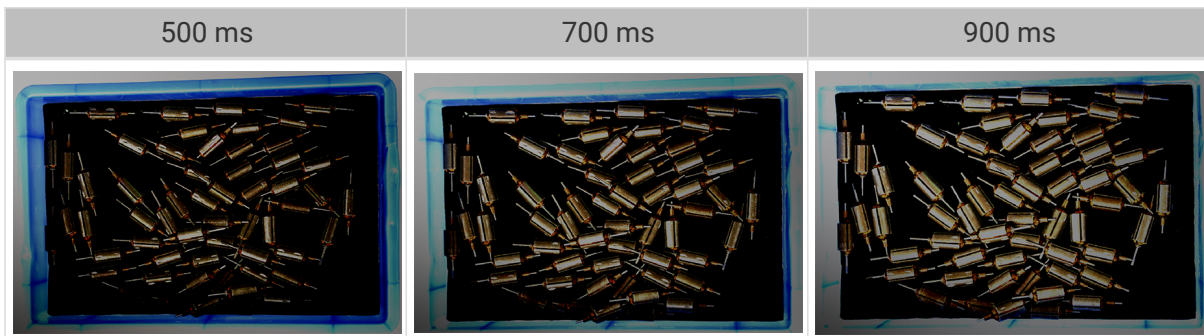
2D images obtained with **Tone Mapping** disabled or enabled (all other conditions identical):



HDR: Exposure Sequence

Description	Set multiple exposure times, and the captured images are merged to generate a 2D image that retains more details in the highlights and shadows.
Visibility	Beginner, Expert, Guru
Values	N/A
Instructions	<ol style="list-style-type: none"> 1. Double-click [Edit] to the right of Exposure Sequence to open the editing window of the exposure sequence. 2. Click [+] to add and set a new exposure time. 3. To delete an exposure time, select it and click [-]. 4. After you finish adding the exposure times, click [Apply] to apply the exposure sequence. <div style="display: flex; align-items: center; margin-top: 10px;"> <div style="margin-right: 10px;">  </div> <ul style="list-style-type: none"> ◦ [Cancel]: do not save the current settings and exit this window. ◦ [Reset]: clear all the exposure times. </div>

2D images captured with single exposure times (all other conditions identical):




2D images captured with different exposure sequences consisting of the above exposure times (all other conditions identical):



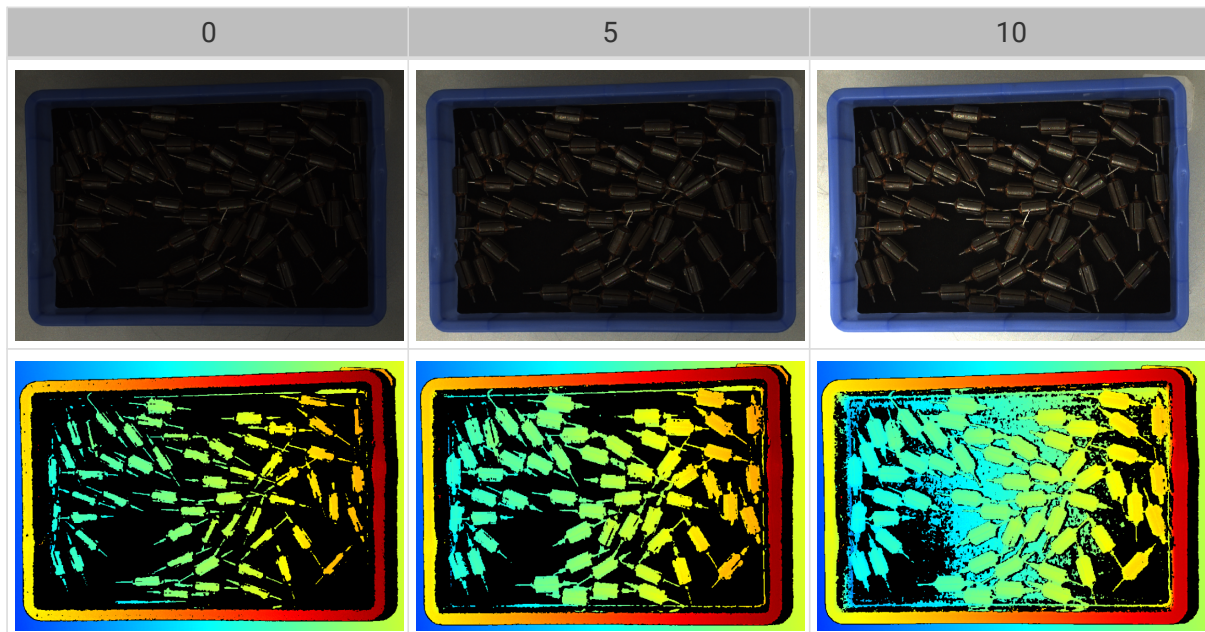
3D Parameters

Gain

Description	<p>This parameter affects the brightness of the captured image. At the same time, more noise is introduced.</p> <div style="display: flex; align-items: center; margin-top: 10px;"> <div style="margin-right: 10px;">  </div> <p>This parameter affects the quality of the depth map and point cloud by changing the brightness of the images used for calculating depth data.</p> </div>
Visibility	Expert, Guru

Values	0 to 16 dB
Instructions	When the expected brightness cannot be reached by adjusting Exposure Time , adjust this parameter.

2D images and depth maps obtained with different **Gain** values (all other conditions identical):



Parameters for the Depth Map / Point Cloud

In the depth map and point cloud, the needed data should be complete.



Determine which part of the data is needed according to your actual needs. For example, if you need to pick metal bowls with opening facing up by the rim, usually you only need to make sure that the data of the bowl rims is complete.

Parameters in the following categories affect the quality of the depth map and point cloud.

Parameter category	Depth map	Point cloud
3D Parameters	✓	✓
Point Cloud Processing		✓
Depth Range	✓	✓
ROI	✓	✓

3D Parameters

The parameters in this category affect the images used for calculating depth data, thus affecting the quality of the depth map and point cloud.

Use **Exposure Assistant** can obtain the recommended exposure parameter values. Double-click [**Auto Set**] to the right of **3D Parameters** to open **Exposure Assistant**.

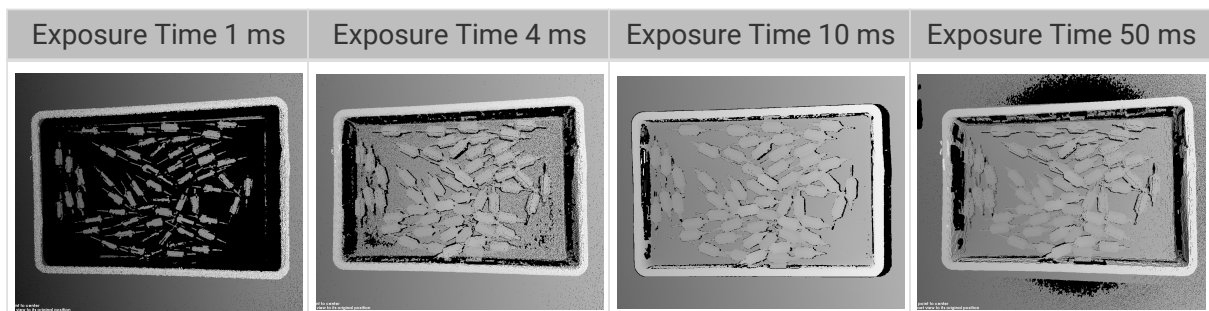
Exposure Multiplier

Description	Set the number of Exposure Time .
Visibility	Beginner, Expert, Guru
Values	1 to 3
Instruction	<ul style="list-style-type: none"> When the value of Exposure Multiplier is greater than 1, multiple Exposure Time parameters must be set. All images captured with different exposure times are used for calculating depth. Using multiple exposure times can improve the completeness of depth data but also increases processing time. With multiple exposure times, the amount of time for obtaining the depth map and point cloud increases. Please try to minimize the number of exposure times while ensuring the data quality.

Exposure Time


Description	Set the exposure time for acquiring depth information. The number of Exposure Time to be set depends on the value of Exposure Multiplier .
Visibility	Beginner, Expert, Guru
Values	<ul style="list-style-type: none"> Value range: 0.1 ms to 99 ms
Instructions	<ul style="list-style-type: none"> Usually, long exposure time is used for dark objects, and short exposure time is used for light objects. Exposure times too long or too short both lead to loss of depth information. The exposure time of a laser camera must be a multiple of 4. If the entered value is not a multiple of 4, it will be automatically adjusted. The minimum value that can be set is 4 ms for the Laser series and 8 ms for other laser cameras.

Point clouds obtained under different **Exposure Time** values (all other conditions identical):



The black regions lack depth information to generate corresponding point cloud.

Gain

Description	<p>This parameter affects the brightness of the captured image. At the same time, more noise is introduced.</p> <div style="display: flex; align-items: center;">  <p>This parameter affects the quality of the depth map and point cloud by changing the brightness of the images used for calculating depth data.</p> </div>
-------------	---

Visibility	Expert, Guru
Values	0 to 16 dB
Instructions	When the expected brightness cannot be reached by adjusting Exposure Time , adjust this parameter.

2D images and depth maps obtained with different **Gain** values (all other conditions identical):



Projector

Light Brightness

Description	This parameter affects the brightness of the projected structured light.
Visibility	Expert, Guru
Values	<ul style="list-style-type: none"> • High: high brightness, suitable for dark objects. • Normal: medium brightness, suitable for regular objects. • Low: low brightness, suitable for reflective objects.
Instructions	Adjust this parameter based on the object type.

Fringe Coding Mode

Description	Select the pattern of the structured light to be projected.
Visibility	Expert, Guru

Values	<ul style="list-style-type: none"> • Fast: suitable for opaque objects, provides fast acquisition speed, but the depth data quality is lower. • Accurate: suitable for opaque objects, provides high-quality depth data, but the acquisition speed is slower. • Translucent: suitable for translucent objects, provides high-quality depth data, but the acquisition speed is slower.
Instructions	Adjust this parameter according to the type of target objects and your actual needs for data quality and data acquisition speed.



- When **Translucent** is selected, the following tool and parameters are unavailable:
 - [3D Exposure Assistant](#) tool
 - [Anti-Flicker Mode](#) in the **3D Parameters** category
 - [Edge Preservation](#) and [Distortion Correction](#) in the **Point Cloud Processing** category
- When **Translucent** is selected, you can fill in the missing points in the point cloud by adjusting [Gap Filling](#) in the **Point Cloud Processing** category.

The suitable scenarios of the Translucent option

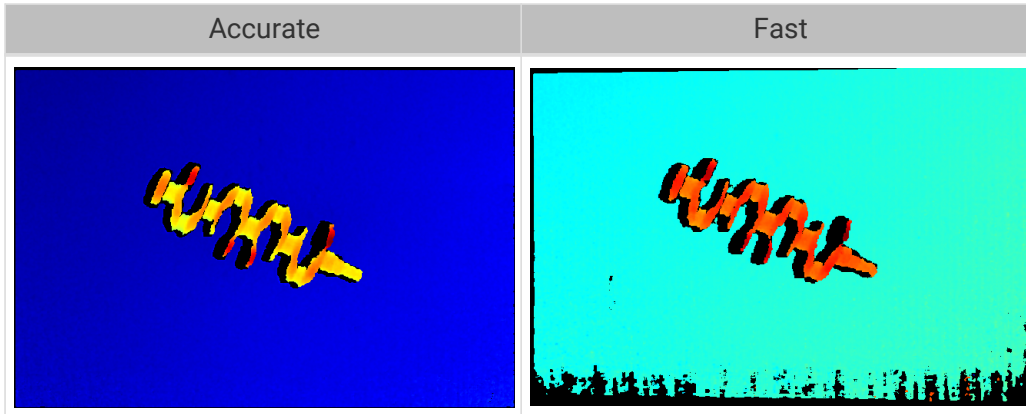
The **Translucent** option can greatly improve the quality of depth data for translucent objects. The following are some typical suitable objects:

- IV fluid bags
- Plastic test tubes (frosted surface)
- Plastic packaging (bags, bottles, cans, boxes, etc.)

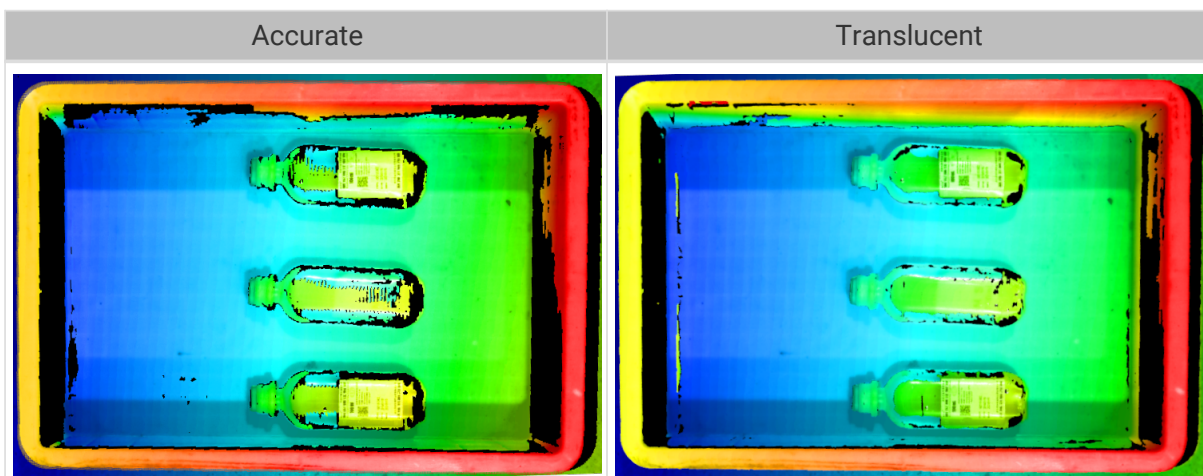
Meanwhile, the improving effect of the **Transparent** option has the following limit:

- For highly transparent objects (such as clear glass test tubes), the improving effect is not significant.
- For largely curved areas on the objects, the improving effect is not significant.
- When translucent objects overlap each other, the improving effect is not significant.
- When the background is light-colored or highly reflective, the improving effect is not significant.
- When the ambient light is strong or varies frequently, the improving effect is not significant.
- When using a camera whose 2D image is monochrome, because the projected structured light is blue, if the translucent objects contain blue objects inside, the improving effect is not significant.

When the target object is not translucent, the point clouds obtained with **Fringe Coding Mode** set to **Accurate** and **Fast** are as follows (all other conditions identical):



When the target object is translucent, the point clouds obtained with **Fringe Coding Mode** set to **Accurate** and **Translucent** are as follows (all other conditions identical):



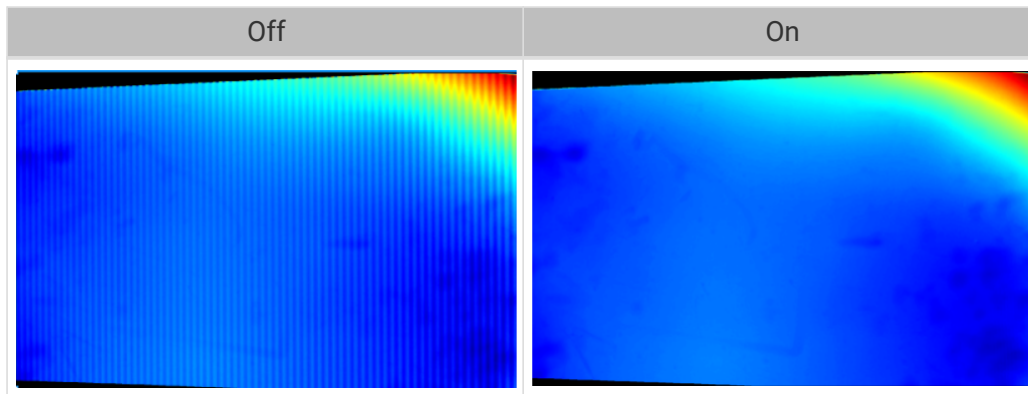
Anti-Flicker Mode

Description	Flicker refers to the rapid and periodical change in the intensity of artificial light. This phenomenon can cause fluctuations in the depth data. Such fluctuations can be reduced by adjusting the projection frequency of the structured light.
Visibility	Expert, Guru
Values	<ul style="list-style-type: none"> • Off • AC50Hz • AC60Hz
Instructions	Please select the option that corresponds to the frequency of the power supply in your location. The AC frequency in most countries is 50Hz. The AC frequency in the U.S. and some Asian countries is 60Hz.



Anti-Flicker Mode is unavailable when **Fringe Coding Mode** in the **3D Parameters** category is set to **Translucent**.

Depth maps obtained with **Anti-Flicker Mode** off or on (all other conditions identical):



Point Cloud Processing

Adjusting the parameters in the **Point Cloud Processing** category can improve the quality of the point cloud.

Principles for Adjusting Point Cloud Processing Parameters

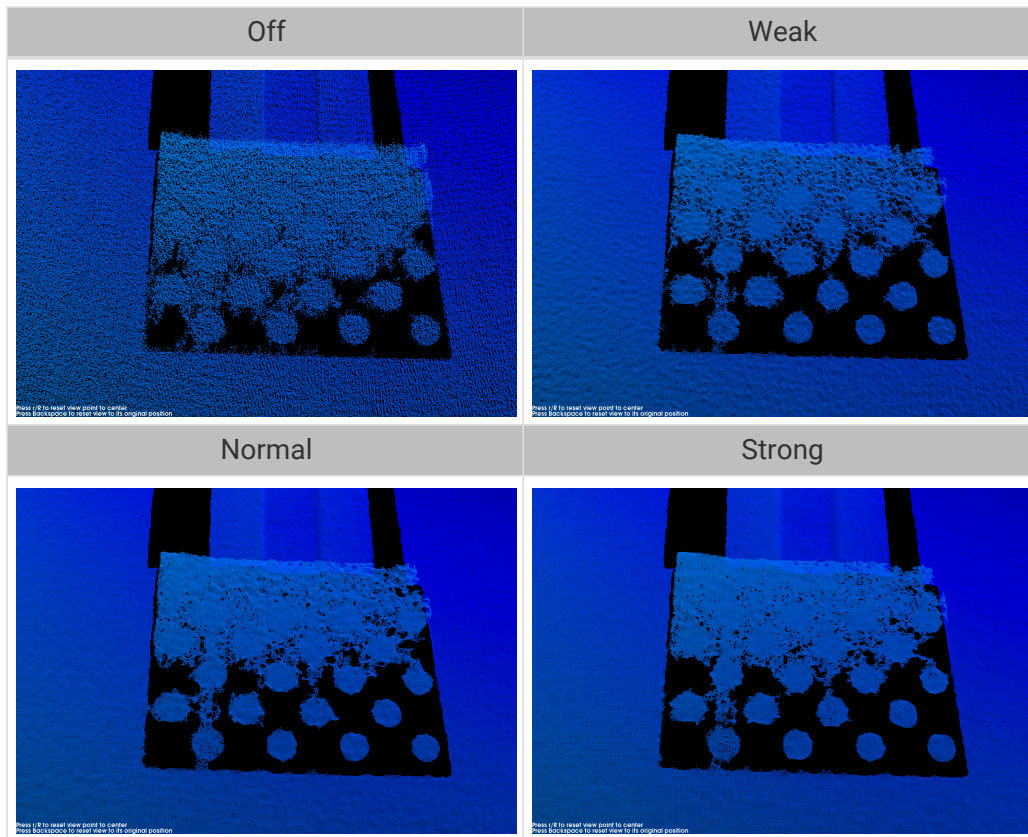
When adjusting the **Point Cloud Processing** parameters, follow these principles to reduce the camera acquisition time and cycle time:

1. Adjust **Outlier Removal** first. The calculation time for different intensities of this parameter is about the same. Using a higher intensity will not add to the calculation time too much.
2. It is recommended to use low intensities of **Surface Smoothing** and **Noise Removal**. The higher the intensities of these two parameters, the longer the calculation time.

Surface Smoothing

Description	This parameter reduces the depth fluctuation in the point cloud and improve its resemblance to the actual object surface.
Visibility	Beginner, Expert, Guru
Values	<ul style="list-style-type: none"> • Off • Weak • Normal • Strong
Instructions	<ul style="list-style-type: none"> • Surface Smoothing causes loss of object surface details. Higher intensity causes more object details to be lost. Lower intensity causes fewer object details to be lost. • Higher intensity results in slower calculation, and lower intensity results in faster calculation.

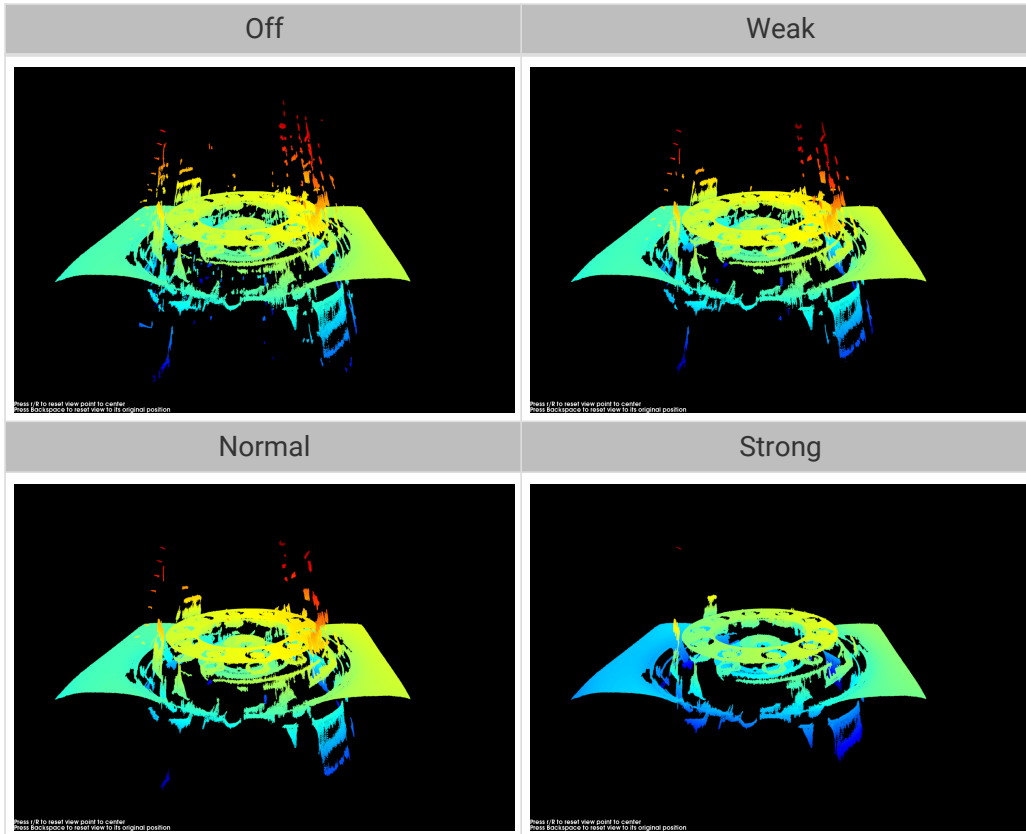
Point clouds obtained with different **Surface Smoothing** values (all other conditions identical):



Outlier Removal

Description	This parameter removes the outliers in the point cloud. Outliers are clustered points away from the object point cloud.
Visibility	Beginner, Expert, Guru
Values	<ul style="list-style-type: none"> • Off • Weak • Normal • Strong
Instructions	<ul style="list-style-type: none"> • Higher intensity removes more outliers, and lower intensity removes fewer outliers. • If the object point cloud contains multiple parts, high intensities of outlier removal might remove some parts. For example, if the object is a cup, outlier removal might remove the point cloud of the handle part.

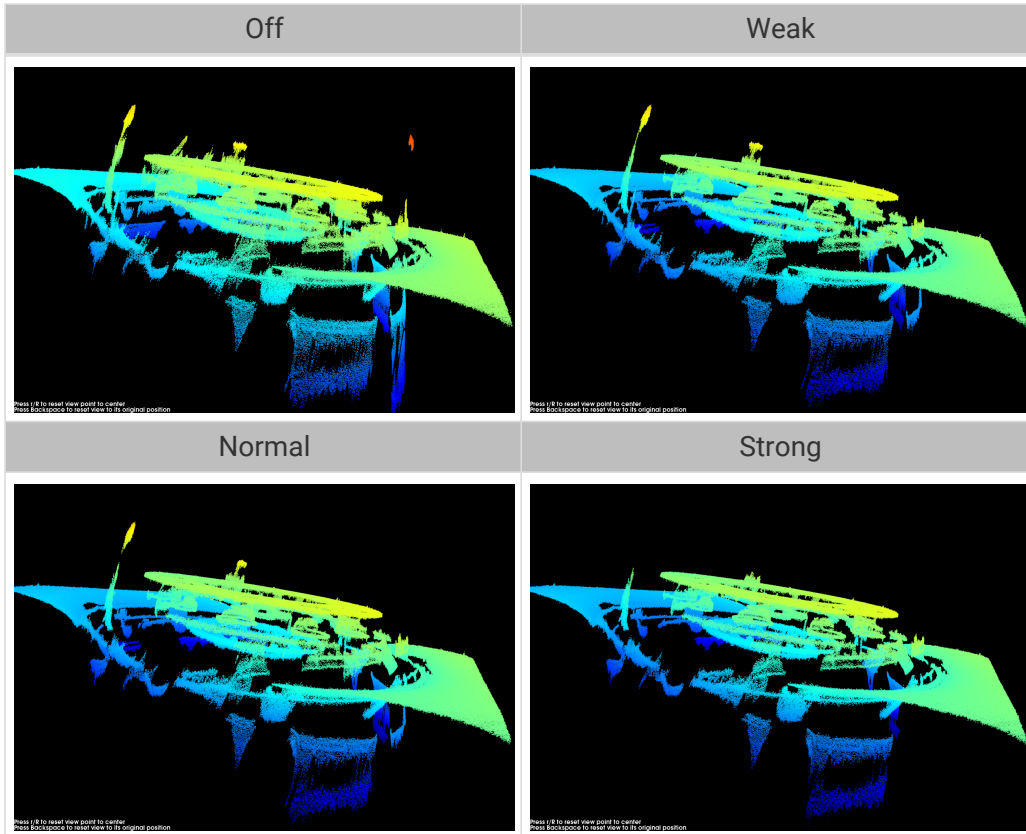
Point clouds obtained with different **Outlier Removal** values (all other conditions identical):



Noise Removal

Description	This parameter removes the noise in the point cloud, thus reducing the impact on the precision and accuracy of subsequent calculation. Noise is the scattered points close to the object surface.
Visibility	Beginner, Expert, Guru
Values	<ul style="list-style-type: none"> • Off • Weak • Normal • Strong
Instructions	<ul style="list-style-type: none"> • Higher intensity removes more noise but may also remove some object features. Lower intensity removes less noise but also preserves the object features better. • Higher intensity results in slower calculation, and lower intensity results in faster calculation.

Point clouds obtained with different **Noise Removal** values (all other conditions identical):



If this function removes the needed object features, please reduce the intensity. However, more noise will be retained.

Distortion Correction

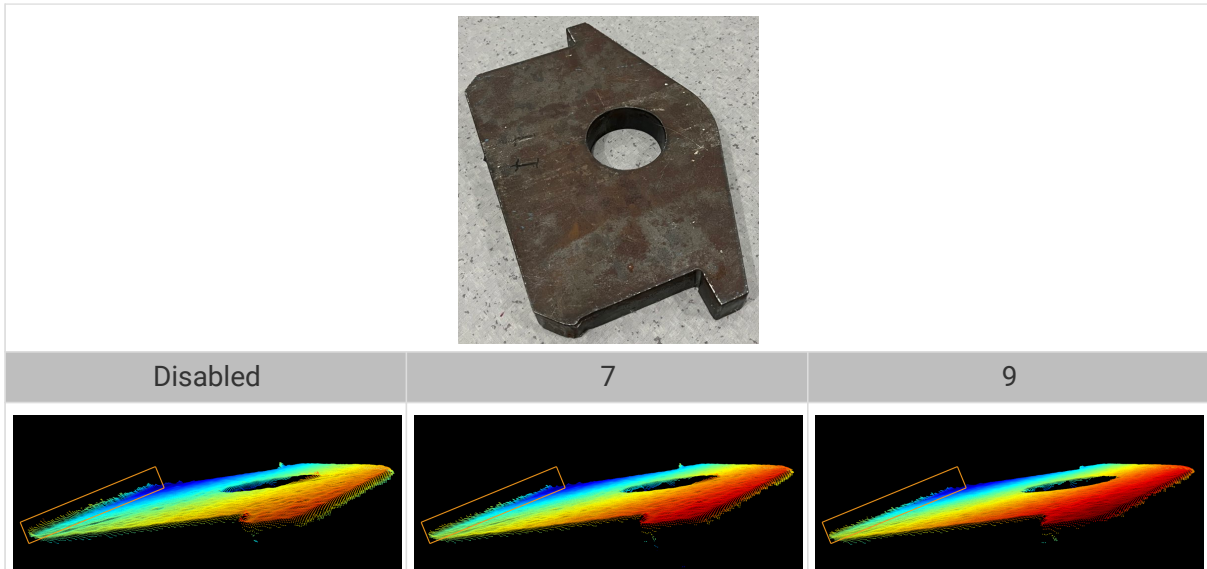
Description	Correct the distortion in the depth data caused by sudden changes in object color or height. Check Enable Distortion Correction first, and then adjust Current Value .
Visibility	Expert, Guru
Values	1 to 10
Instruction	Adjust Current Value based on the degree of distortion in the depth map or point cloud. Acquire data again to see the effect.



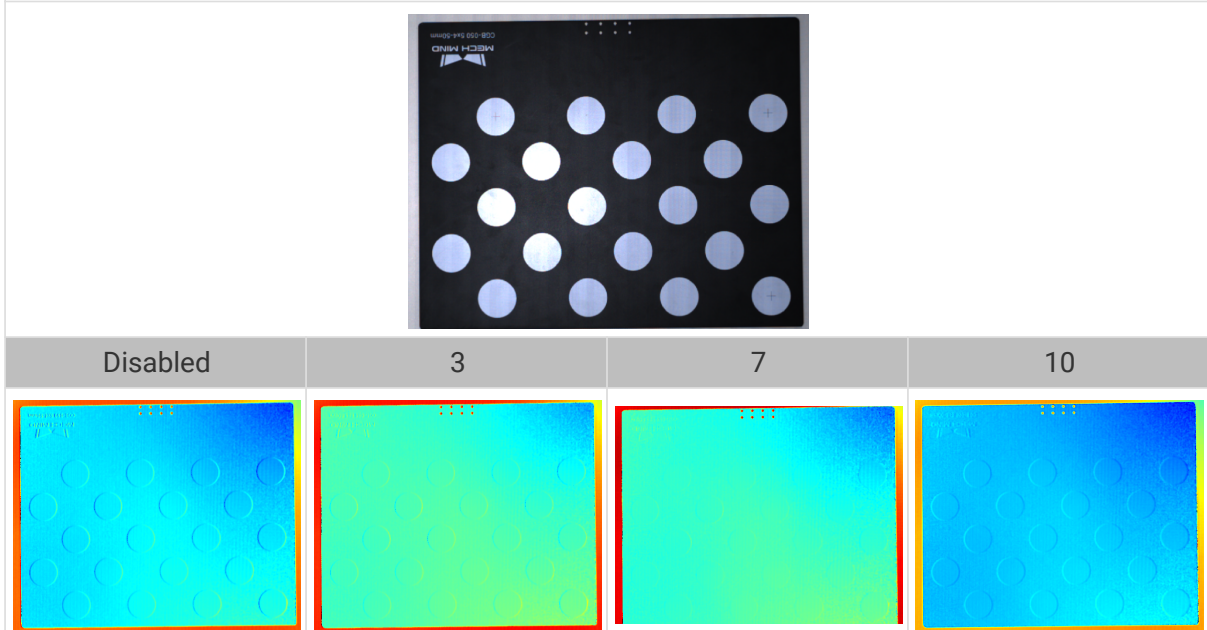
- Enabling distortion correction lowers the acquisition speed.
- A **Current Value** too large may result in adverse effects. Please acquire data again after adjusting the parameter to check its influence on the depth map and point cloud.
- **Distortion Correction** is unavailable when **Fringe Coding Mode** in the **3D Parameters** category is set to **Translucent**.

Depth maps obtained with different **Current Values** of **Distortion Correction** (all other conditions identical):

The target object is a metal workpiece with a smooth surface and sudden height change at the edges.



The target object is a calibration board with a smooth surface and sudden color change from black to white.



Gap Filling

Description	Fill in the gaps in the point cloud so that the object's surface features are more complete.
Visibility	Expert, Guru
Values	<ul style="list-style-type: none"> • Off • Weak • Normal • Strong
Instruction	Adjust this parameter based on the amount of missing points in the point cloud.



- This parameter is only available when [Fringe Coding Mode](#) in the 3D Parameters category is set to **Translucent**.
- More intense gap filling fills more missing points but may also distort object edges.

Edge Preservation

Description	This parameter preserves the sharpness of object edges during surface smoothing.
Visibility	Guru
Values	<ul style="list-style-type: none"> • Sharp: Preserves the sharpness of object edges as much as possible. However, the effect of surface smoothing will be reduced. • Normal: Balances between edge preservation and surface smoothing. • Smooth: Does not preserve the edges. The object surface will be well smoothed, but the object edges will be distorted.
Instructions	Adjust based on your needs for preserving the edges of workpieces.

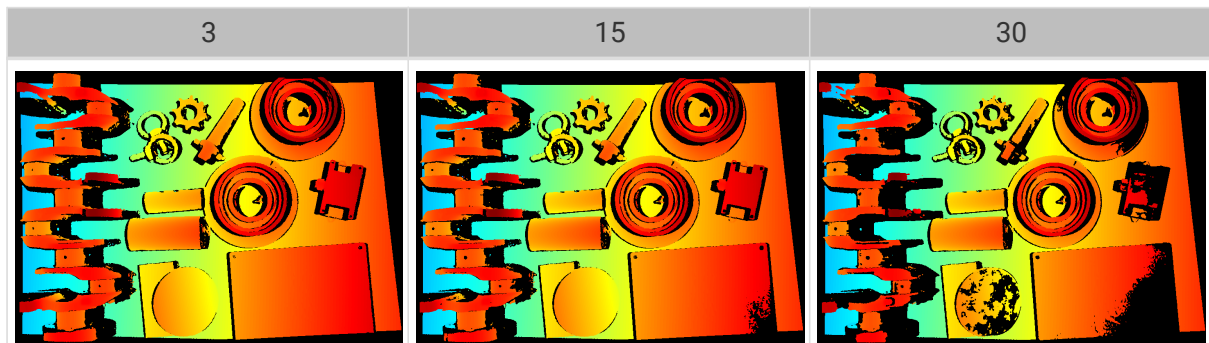


Edge Preservation is unavailable when [Fringe Coding Mode](#) in the 3D Parameters category is set to **Translucent**.

Stripe Contrast Threshold

Description	This parameter is used to remove noise in the point cloud. Adjust this parameter if the level of noise in the point cloud is still high after adjusting Outlier Removal and Noise Removal .
Visibility	Beginner, Expert, Guru
Values	1 to 100
Instructions	<ul style="list-style-type: none"> • Greater value of this parameter results in more points being removed. Smaller value results in fewer points being removed. • Increase the value of this parameter can remove the noise in the point cloud, but may also cause the points of dark objects to be lost.

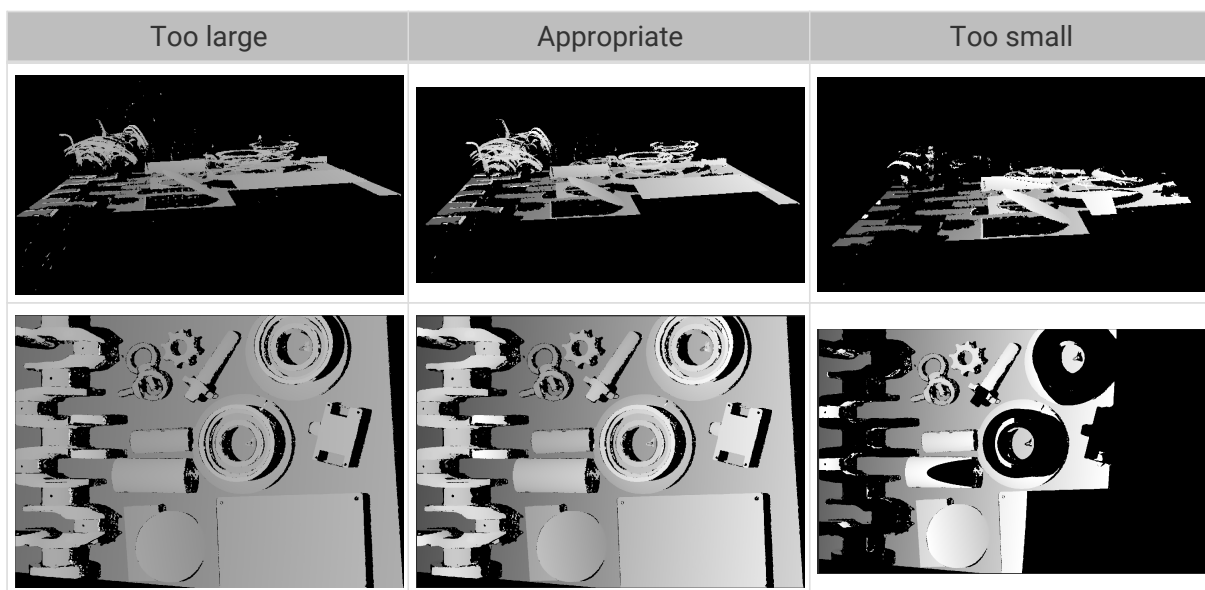
Point clouds obtained with different **Stripe Contrast Threshold** values (all other conditions identical):



Depth Range

Description	Set a Z-direction ROI for the depth map and point cloud. Depth Range can be set within the camera working distance, and the data outside the range is removed from the depth map and point cloud.
Visibility	Beginner, Expert, Guru
Values	<ul style="list-style-type: none"> • Lower Limit: 1 to 4000 mm • Upper Limit: 1 to 5000 mm
Instructions	<ul style="list-style-type: none"> • Depth Range should be adjusted to an appropriate range to ensure the completeness of depth data in the depth map and point cloud. Too large a range may introduce interference, and too small a range may remove the data of key features. • For detailed instructions, please refer to Set Depth Range below.

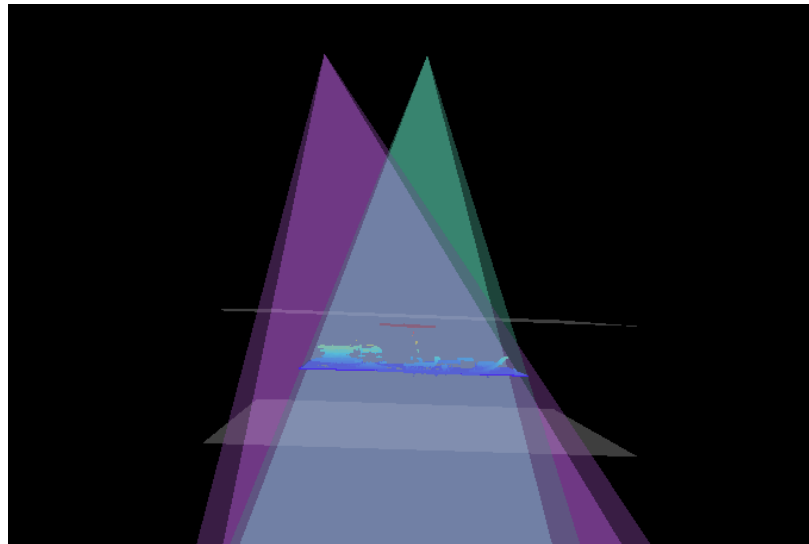
The same point cloud with different **Depth Range** values (all other conditions identical):



Set Depth Range

Follow these steps to adjust **Depth Range**:

1. Double-click [**Edit**] to the right of **Depth Range** to open the **Set Depth Range** window.
2. Click [**Refresh point cloud**] at the top of the right panel to obtain the newest point cloud.
3. [Adjust the position of the point cloud](#): adjust the position until you can see the two gray rectangles that represents the upper and lower limits of **Depth Range**.



- Adjust the depth range: drag the handles on the slider to adjust **Depth Range** roughly. Then, enter specific values on the right to fine-tune **Depth Range**.



Judge if the depth range is appropriate: All needed object features are located between the two gray rectangles, and most noise and outliers are located outside.

- Click [**Save**] to save the depth range settings.



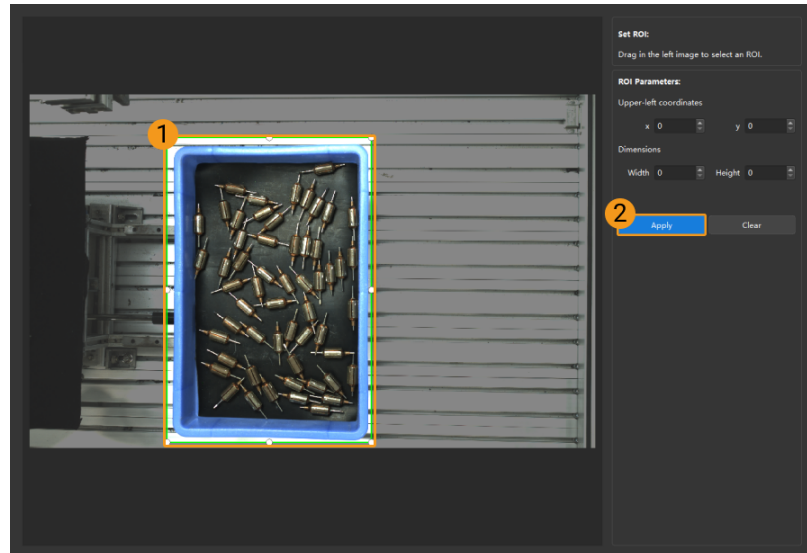
- Click [**Recommended range**] to adjust **Depth Range** to the recommended working distance of the current camera.
- Click [**Reset**] to restore **Depth Range** to the last saved values.

ROI

Description	Set the ROI on the XY plane for the depth map and point cloud. Points outside the selected region are removed.
Visibility	Beginner, Expert, Guru
Values	N/A
Instructions	For detailed instructions, refer to Set ROI below.

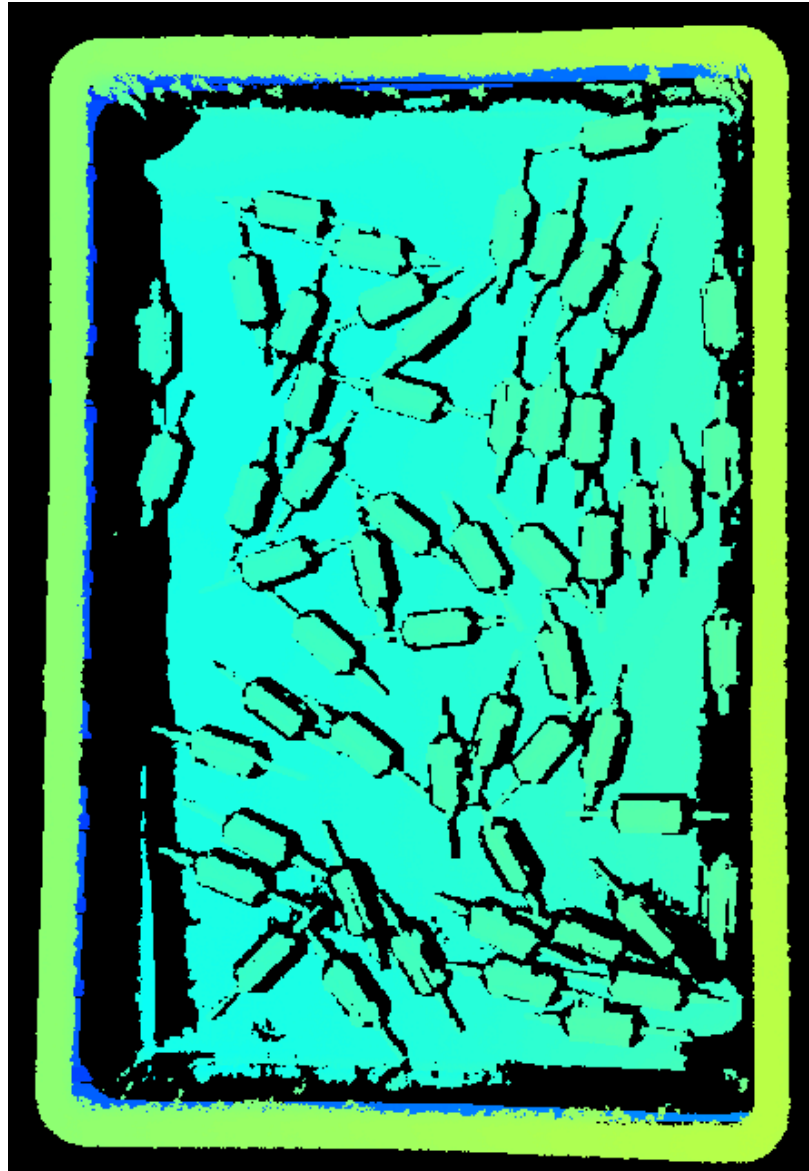
Set ROI

- Double-click [**Edit**] to the right of **ROI** to open the **Set ROI** window.
- Select and adjust the ROI on the left. Drag the selection box to move it. Drag the handles on the box to adjust the box size.
- Click [**Apply**] to apply the set ROI.



- Click [Clear] to remove the current ROI.
- For the DEEP (V4) and LSR (V4) series, the 2D image (depth source) is displayed in this window. If the image is too dark or too bright, please adjust **2D Image (Depth Source) Exposure Mode**.

4. Acquire data again and view the depth map or point cloud to check the result of setting the ROI.



5.3.3. Parameters of PRO XS

This topic introduces the parameters of PRO XS. The parameters are divided into parameters for the 2D image and parameters for the depth map / point cloud, based on the data types that they affect.

Parameters for the 2D Image

The 2D image should not be too bright or too dark, and the surface features of the target object should be clearly visible in the image.

The quality of the 2D image is affected by parameters in the **2D Parameters** category and the **Gain** parameter in the **3D Parameters** category.



If you are using a color camera, and the color of the image differ significantly from that of the objects, you can adjust the **white balance** setting. For detailed instructions, please refer to [Adjust White Balance](#).

2D Parameters

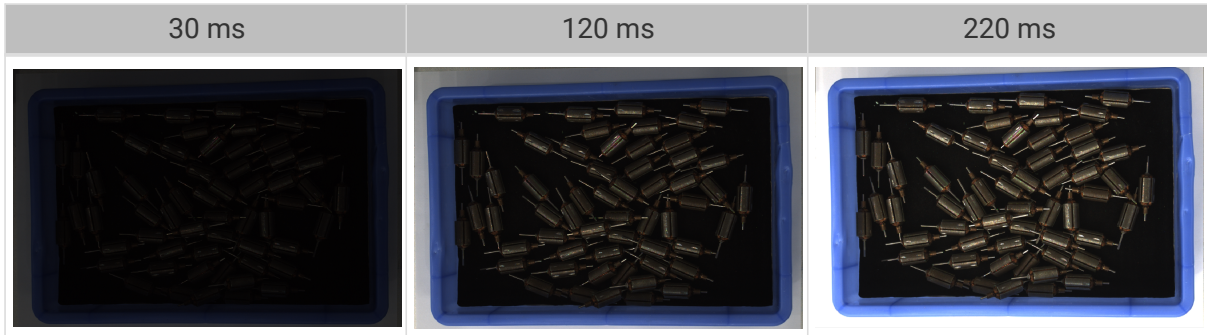
Exposure Mode

Description	Set the exposure mode for capturing the 2D image.
Visibility	Beginner, Expert, Guru
Values	<ul style="list-style-type: none"> • Timed: Set a single exposure time. Usually used in stable lighting conditions. • Auto: The exposure time is automatically adjusted. Usually used in varying lighting conditions. • HDR: Set multiple exposure times and merge the images. Usually used for objects with various colors or textures. • Flash: Use the projector for supplemental light. Usually used in dark environments.
Instructions	<p>After selecting different options, different parameters are displayed in the 2D Parameters category for adjustment:</p> <ul style="list-style-type: none"> • Timed: displays Exposure Time. • Auto: displays Gray Value and Auto-Exposure ROI. • HDR: displays Tone Mapping and Exposure Sequence. • Flash: no other parameters need to be adjusted. The projector projects supplemental light automatically during image capturing.

Timed: Exposure Time

Description	<p>This parameter affects the brightness of the image.</p> <ul style="list-style-type: none"> • Long exposure time increases the image brightness. • Short exposure time decreases the image brightness.
Visibility	Beginner, Expert, Guru
Values	0.1 to 999 ms
Instructions	<p>Adjust this parameter based on the quality of the 2D image. The 2D image should not be too bright or too dark, and the surface features of the target object should be clearly visible in the image.</p> <ul style="list-style-type: none"> • Long exposure time is usually used in dark environments. • Short exposure time is usually used in bright environments.

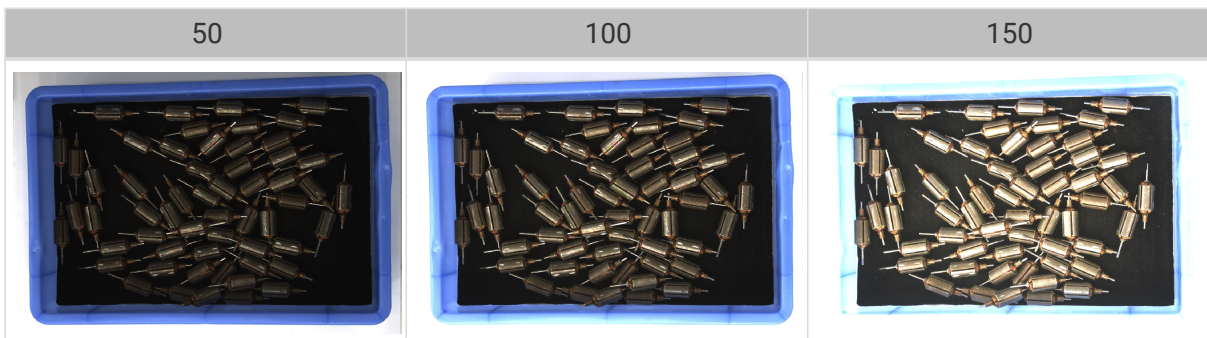
2D images obtained with different **Exposure Time** values (all other conditions identical):



Auto: Gray Value

Description	This parameter affects the brightness of the 2D image. Increase the value if the 2D image is too dark and decrease if too bright.
Visibility	Beginner, Expert, Guru
Values	0 to 255
Instruction	N/A

2D images obtained with different **Gray Value** values (all other conditions identical):

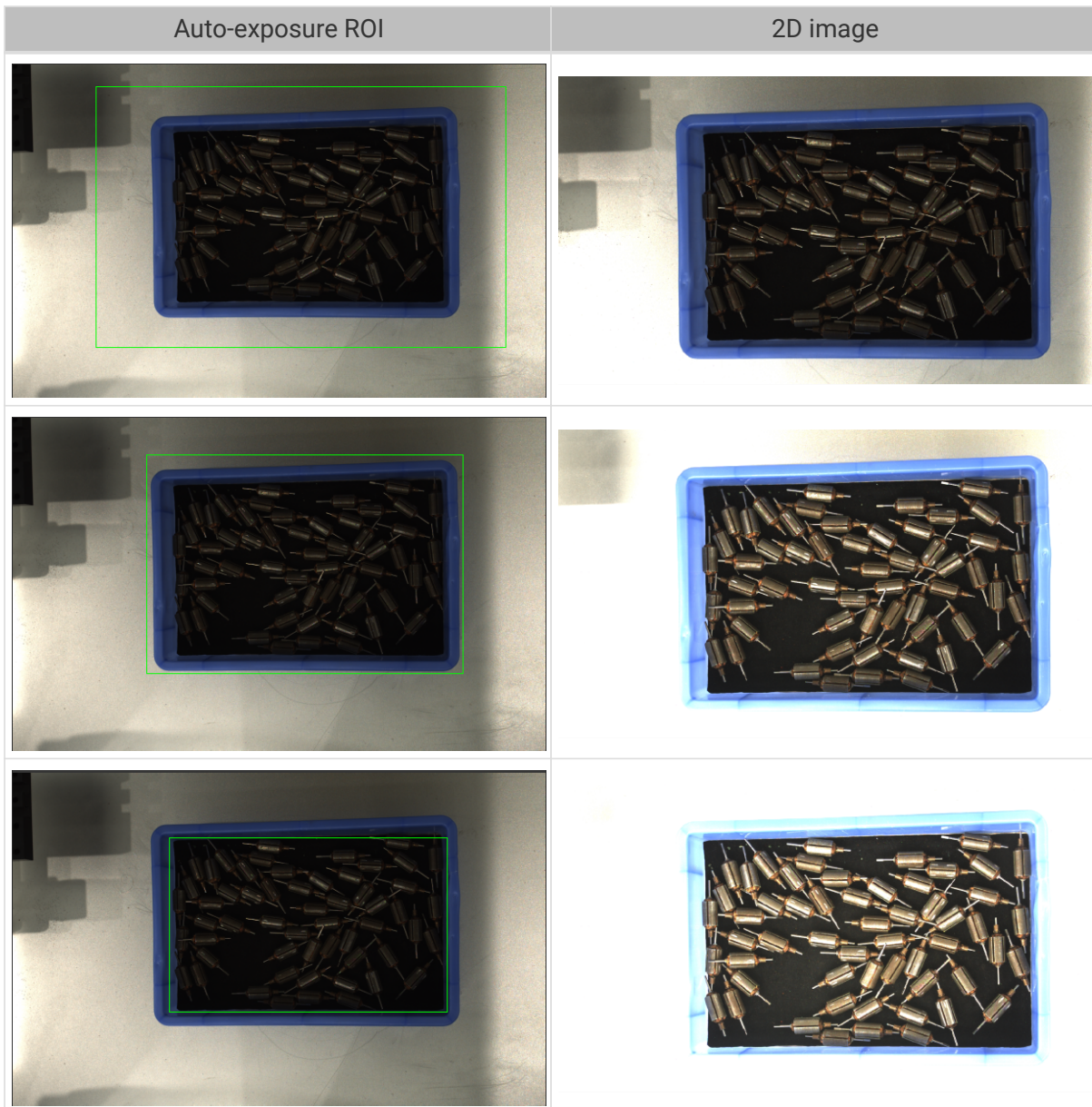


The **Gray Value** of a monochrome image is equivalent to the image brightness; the **Gray Value** of a color image is equivalent to the brightness of each color channel.

Auto: Auto-Exposure ROI

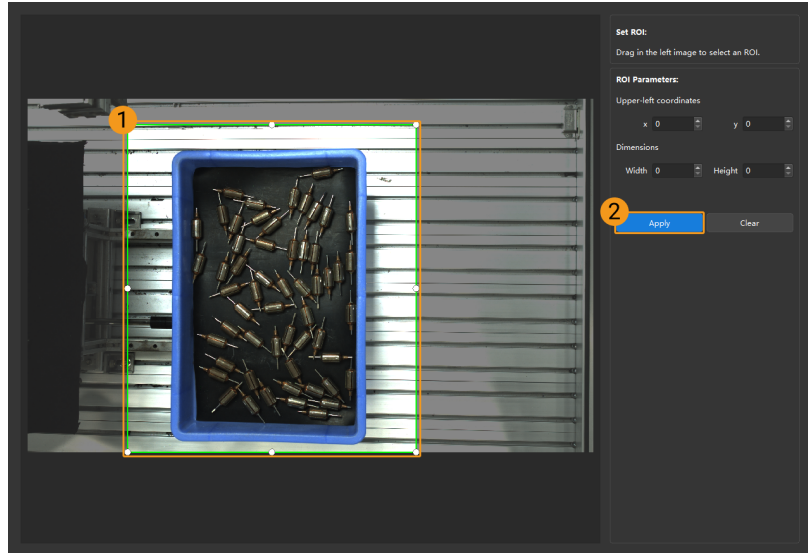
Description	<ul style="list-style-type: none"> The exposure time is automatically adjusted based on the lighting, object colors, etc., in the selected region. When an auto-exposure ROI is not set, the exposure time is automatically adjusted based on the entire camera FOV.
Visibility	Beginner, Expert, Guru
Values	N/A
Instruction	For detailed instructions, refer to Set Auto-Exposure ROI below.

2D images obtained with different **Auto-Exposure ROI** values (all other conditions identical):



Set Auto-Exposure ROI

1. Double-click [**Edit**] to the right of **Auto-Exposure ROI** to open the **Set ROI** window.
2. Select and adjust the ROI on the left. Drag the selection box to move it. Drag the handles on the box to adjust the box size.
3. Click [**Apply**] to apply the set auto-exposure ROI.



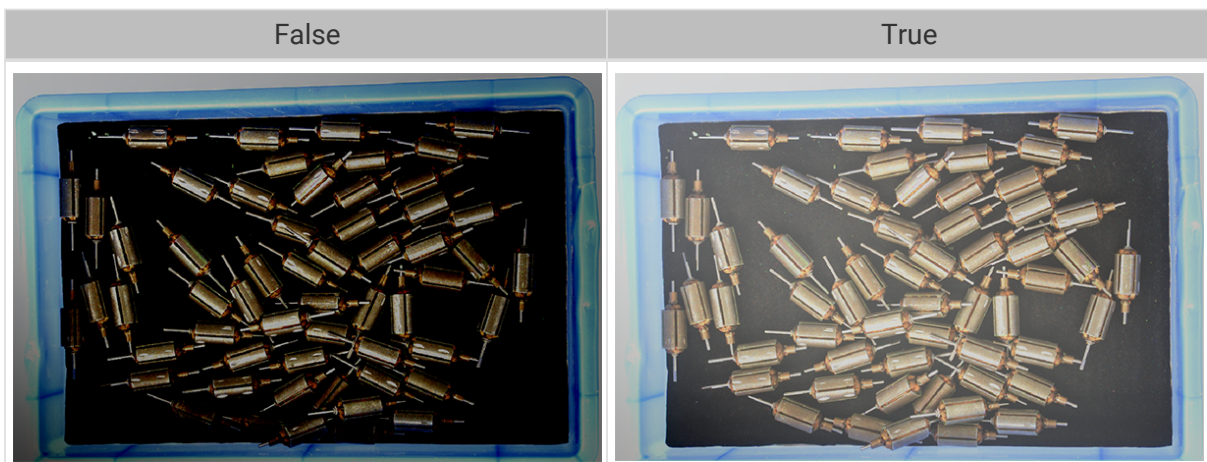
 Click [Clear] to remove the current auto-exposure ROI.

- Acquire data again and view the 2D image to check the result of automatic exposure adjustment.


HDR: Tone Mapping

Description	This function can make the image look more natural. If the 2D image appears very different from the actual objects, please enable this function.
Visibility	Beginner, Expert, Guru
Values	<ul style="list-style-type: none"> • False • True
Instructions	Check to enable Tone Mapping .

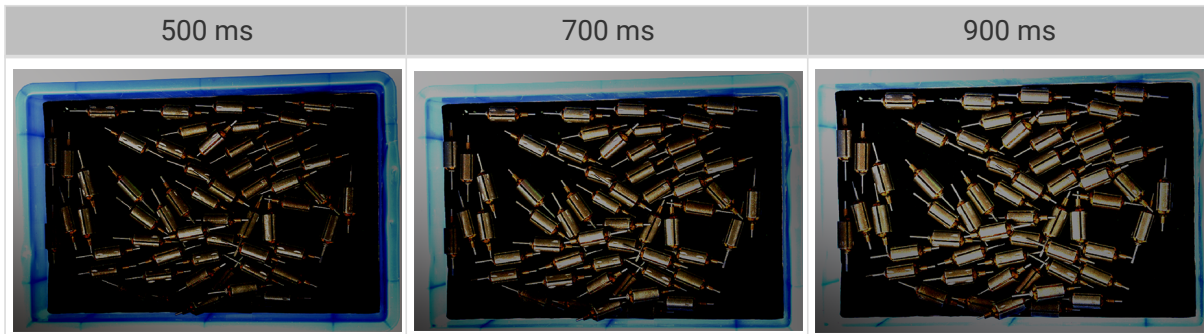
2D images obtained with **Tone Mapping** disabled or enabled (all other conditions identical):



HDR: Exposure Sequence

Description	Set multiple exposure times, and the captured images are merged to generate a 2D image that retains more details in the highlights and shadows.
Visibility	Beginner, Expert, Guru
Values	N/A
Instructions	<ol style="list-style-type: none"> 1. Double-click [Edit] to the right of Exposure Sequence to open the editing window of the exposure sequence. 2. Click [+] to add and set a new exposure time. 3. To delete an exposure time, select it and click [-]. 4. After you finish adding the exposure times, click [Apply] to apply the exposure sequence. <div style="display: flex; align-items: center; margin-top: 10px;"> <div style="margin-right: 10px;">  </div> <ul style="list-style-type: none"> ◦ [Cancel]: do not save the current settings and exit this window. ◦ [Reset]: clear all the exposure times. </div>

2D images captured with single exposure times (all other conditions identical):




2D images captured with different exposure sequences consisting of the above exposure times (all other conditions identical):



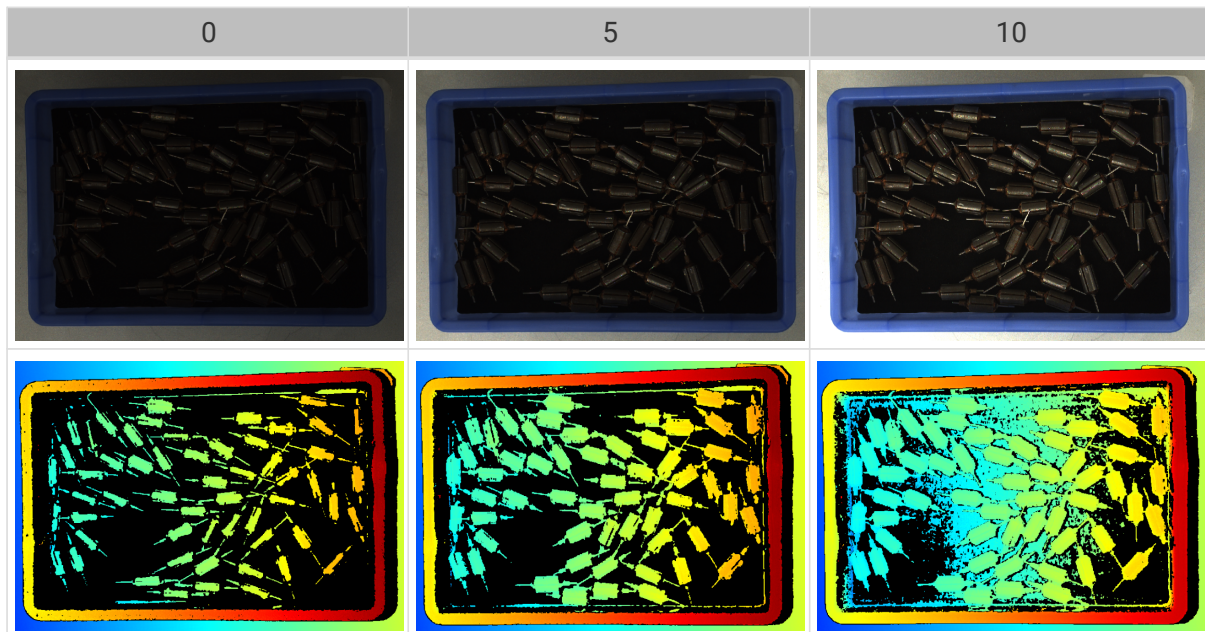
3D Parameters

Gain

Description	<p>This parameter affects the brightness of the captured image. At the same time, more noise is introduced.</p> <div style="display: flex; align-items: center; margin-top: 10px;"> <div style="margin-right: 10px;">  </div> <p>This parameter affects the quality of the depth map and point cloud by changing the brightness of the images used for calculating depth data.</p> </div>
Visibility	Expert, Guru

Values	0 to 16 dB
Instructions	When the expected brightness cannot be reached by adjusting Exposure Time , adjust this parameter.

2D images and depth maps obtained with different **Gain** values (all other conditions identical):



Parameters for the Depth Map / Point Cloud

In the depth map and point cloud, the needed data should be complete.



Determine which part of the data is needed according to your actual needs. For example, if you need to pick metal bowls with opening facing up by the rim, usually you only need to make sure that the data of the bowl rims is complete.

Parameters in the following categories affect the quality of the depth map and point cloud.

Parameter category	Depth map	Point cloud
3D Parameters	✓	✓
Point Cloud Processing		✓
Depth Range	✓	✓
ROI	✓	✓

3D Parameters

The parameters in this category affect the images used for calculating depth data, thus affecting the quality of the depth map and point cloud.

Use **Exposure Assistant** can obtain the recommended exposure parameter values. Double-click [**Auto Set**] to the right of **3D Parameters** to open **Exposure Assistant**.

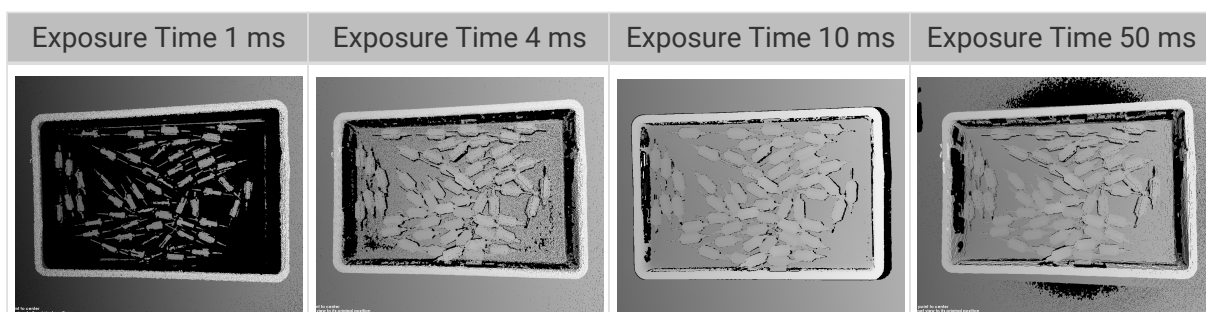
Exposure Multiplier

Description	Set the number of Exposure Time .
Visibility	Beginner, Expert, Guru
Values	1 to 3
Instruction	<ul style="list-style-type: none"> When the value of Exposure Multiplier is greater than 1, multiple Exposure Time parameters must be set. All images captured with different exposure times are used for calculating depth. Using multiple exposure times can improve the completeness of depth data but also increases processing time. With multiple exposure times, the amount of time for obtaining the depth map and point cloud increases. Please try to minimize the number of exposure times while ensuring the data quality.

Exposure Time


Description	Set the exposure time for acquiring depth information. The number of Exposure Time to be set depends on the value of Exposure Multiplier .
Visibility	Beginner, Expert, Guru
Values	<ul style="list-style-type: none"> Value range: 0.1 ms to 99 ms
Instructions	<ul style="list-style-type: none"> Usually, long exposure time is used for dark objects, and short exposure time is used for light objects. Exposure times too long or too short both lead to loss of depth information. The exposure time of a laser camera must be a multiple of 4. If the entered value is not a multiple of 4, it will be automatically adjusted. The minimum value that can be set is 4 ms for the Laser series and 8 ms for other laser cameras.

Point clouds obtained under different **Exposure Time** values (all other conditions identical):



The black regions lack depth information to generate corresponding point cloud.

Gain

Description	<p>This parameter affects the brightness of the captured image. At the same time, more noise is introduced.</p> <p> This parameter affects the quality of the depth map and point cloud by changing the brightness of the images used for calculating depth data.</p>
-------------	--

Visibility	Expert, Guru
Values	0 to 16 dB
Instructions	When the expected brightness cannot be reached by adjusting Exposure Time , adjust this parameter.

2D images and depth maps obtained with different **Gain** values (all other conditions identical):



Projector

Light Brightness

Description	This parameter affects the brightness of the projected structured light.
Visibility	Expert, Guru
Values	<ul style="list-style-type: none"> • High: high brightness, suitable for dark objects. • Normal: medium brightness, suitable for regular objects. • Low: low brightness, suitable for reflective objects.
Instructions	Adjust this parameter based on the object type.

Fringe Coding Mode

Description	Select the pattern of the structured light to be projected.
Visibility	Expert, Guru
Values	<ul style="list-style-type: none"> • Fast: suitable for opaque objects, provides fast acquisition speed, but the depth data quality is lower. • Accurate: suitable for opaque objects, provides high-quality depth data, but the acquisition speed is slower.

Instructions	Adjust this parameter according to the type of target objects and your actual needs for data quality and data acquisition speed.
--------------	--

When the target object is not translucent, the point clouds obtained with **Fringe Coding Mode** set to **Accurate** and **Fast** are as follows (all other conditions identical):



Point Cloud Processing

Adjusting the parameters in the **Point Cloud Processing** category can improve the quality of the point cloud.

Principles for Adjusting Point Cloud Processing Parameters

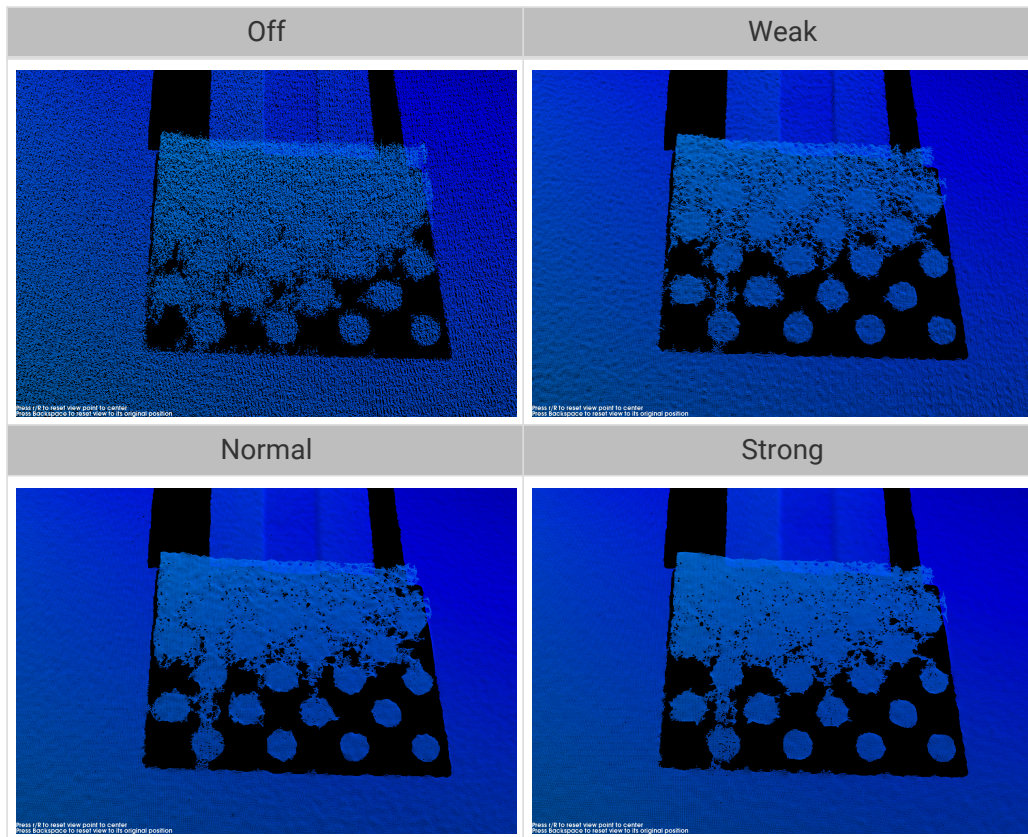
When adjusting the **Point Cloud Processing** parameters, follow these principles to reduce the camera acquisition time and cycle time:

1. Adjust **Outlier Removal** first. The calculation time for different intensities of this parameter is about the same. Using a higher intensity will not add to the calculation time too much.
2. It is recommended to use low intensities of **Surface Smoothing** and **Noise Removal**. The higher the intensities of these two parameters, the longer the calculation time.

Surface Smoothing

Description	This parameter reduces the depth fluctuation in the point cloud and improve its resemblance to the actual object surface.
Visibility	Beginner, Expert, Guru
Values	<ul style="list-style-type: none"> • Off • Weak • Normal • Strong
Instructions	<ul style="list-style-type: none"> • Surface Smoothing causes loss of object surface details. Higher intensity causes more object details to be lost. Lower intensity causes fewer object details to be lost. • Higher intensity results in slower calculation, and lower intensity results in faster calculation.

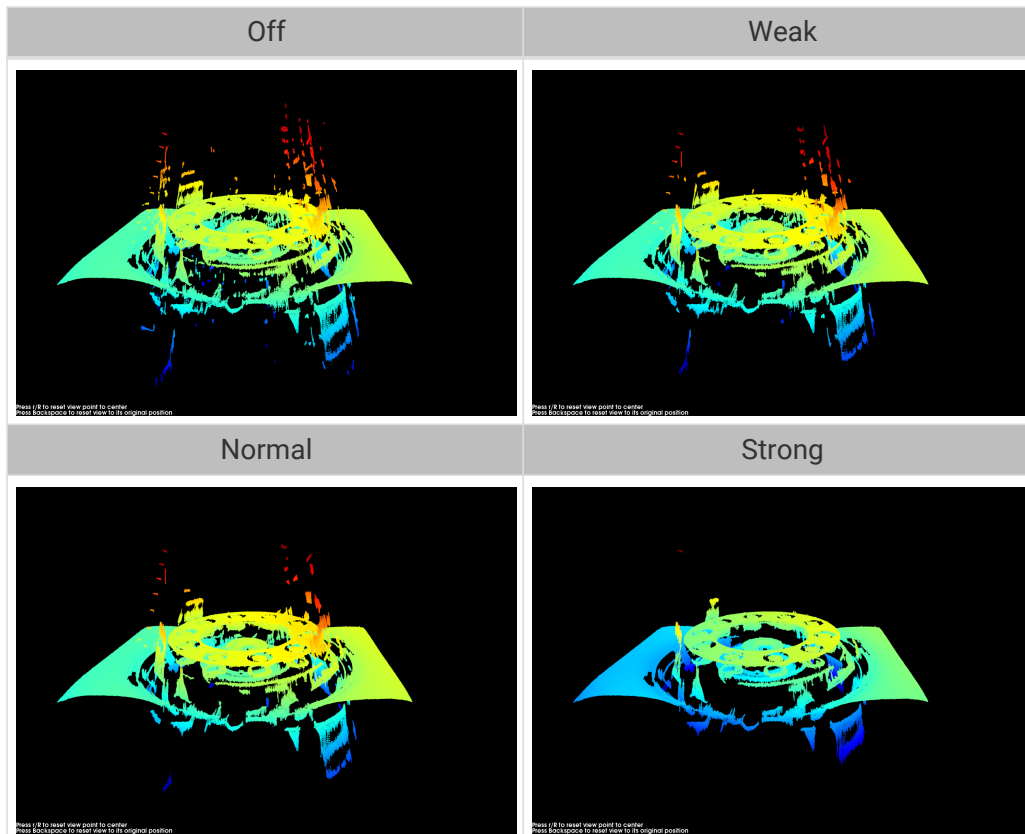
Point clouds obtained with different **Surface Smoothing** values (all other conditions identical):



Outlier Removal

Description	This parameter removes the outliers in the point cloud. Outliers are clustered points away from the object point cloud.
Visibility	Beginner, Expert, Guru
Values	<ul style="list-style-type: none"> • Off • Weak • Normal • Strong
Instructions	<ul style="list-style-type: none"> • Higher intensity removes more outliers, and lower intensity removes fewer outliers. • If the object point cloud contains multiple parts, high intensities of outlier removal might remove some parts. For example, if the object is a cup, outlier removal might remove the point cloud of the handle part.

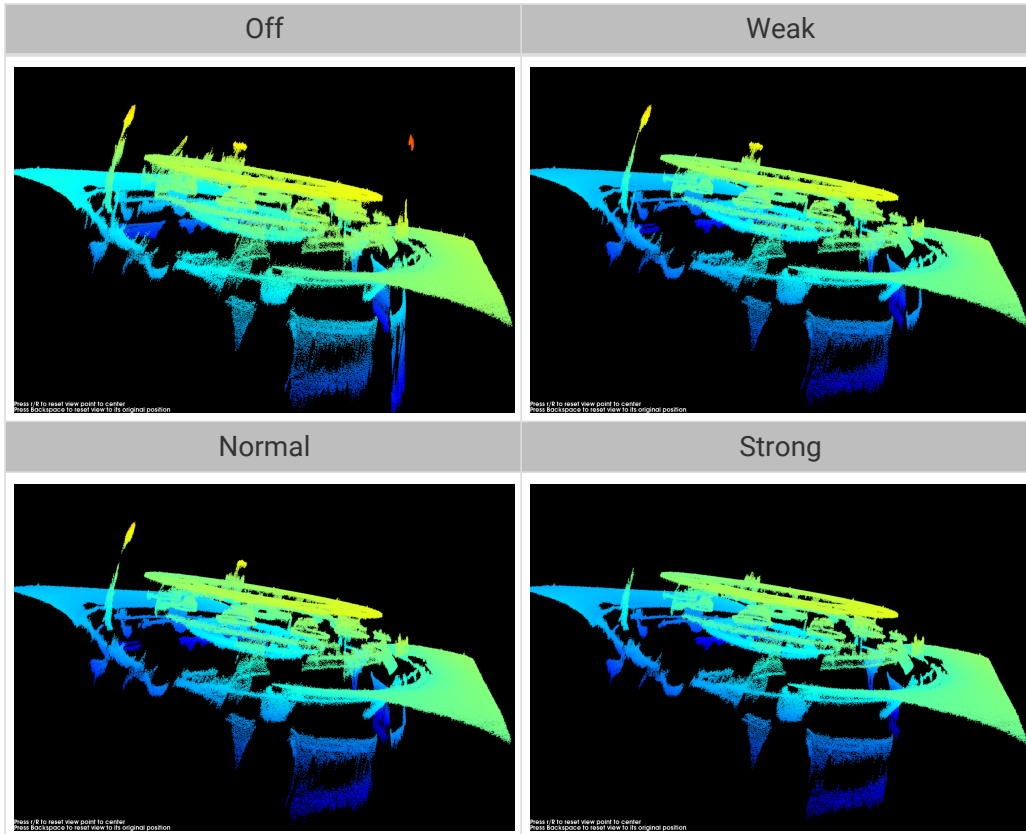
Point clouds obtained with different **Outlier Removal** values (all other conditions identical):



Noise Removal

Description	This parameter removes the noise in the point cloud, thus reducing the impact on the precision and accuracy of subsequent calculation. Noise is the scattered points close to the object surface.
Visibility	Beginner, Expert, Guru
Values	<ul style="list-style-type: none"> • Off • Weak • Normal • Strong
Instructions	<ul style="list-style-type: none"> • Higher intensity removes more noise but may also remove some object features. Lower intensity removes less noise but also preserves the object features better. • Higher intensity results in slower calculation, and lower intensity results in faster calculation.

Point clouds obtained with different **Noise Removal** values (all other conditions identical):



If this function removes the needed object features, please reduce the intensity. However, more noise will be retained.

Edge Preservation

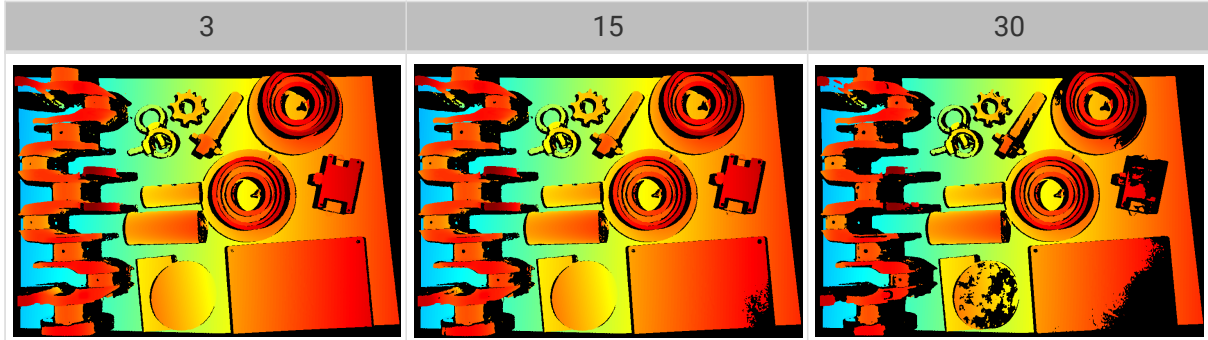
Description	This parameter preserves the sharpness of object edges during surface smoothing.
Visibility	Guru
Values	<ul style="list-style-type: none"> • Sharp: Preserves the sharpness of object edges as much as possible. However, the effect of surface smoothing will be reduced. • Normal: Balances between edge preservation and surface smoothing. • Smooth: Does not preserve the edges. The object surface will be well smoothed, but the object edges will be distorted.
Instructions	Adjust based on your needs for preserving the edges of workpieces.

Stripe Contrast Threshold

Description	This parameter is used to remove noise in the point cloud. Adjust this parameter if the level of noise in the point cloud is still high after adjusting Outlier Removal and Noise Removal .
Visibility	Beginner, Expert, Guru
Values	1 to 100

Instructions	<ul style="list-style-type: none"> • Greater value of this parameter results in more points being removed. Smaller value results in fewer points being removed. • Increase the value of this parameter can remove the noise in the point cloud, but may also cause the points of dark objects to be lost.
--------------	---

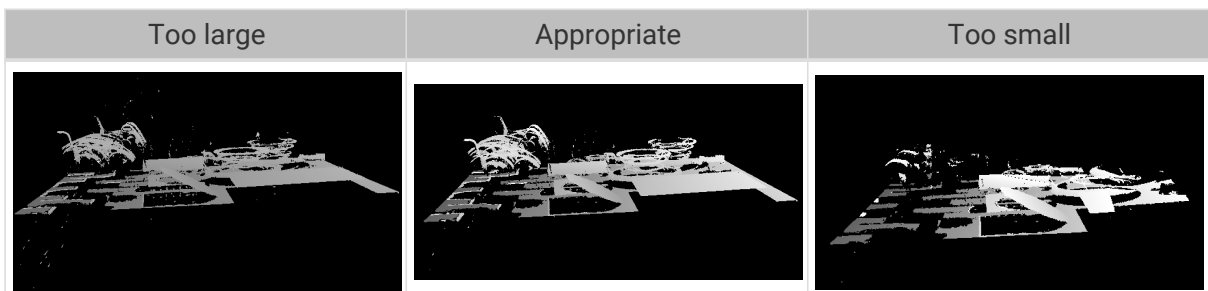
Point clouds obtained with different **Stripe Contrast Threshold** values (all other conditions identical):

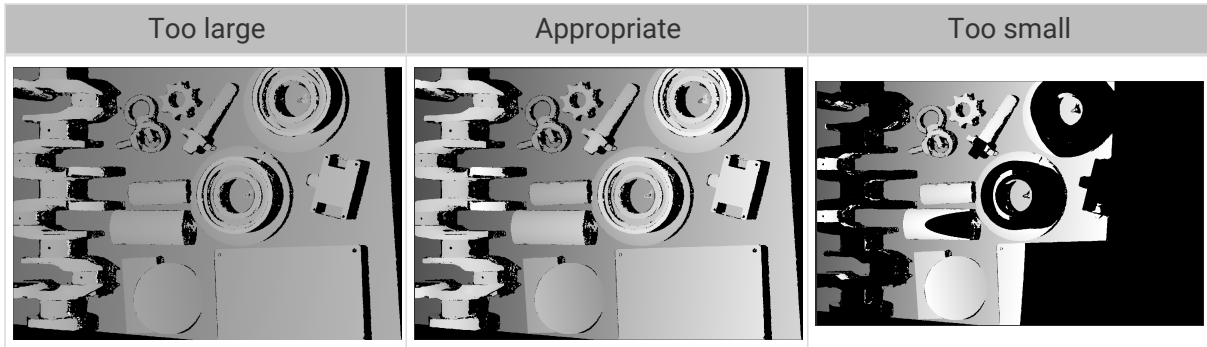


Depth Range

Description	Set a Z-direction ROI for the depth map and point cloud. Depth Range can be set within the camera working distance, and the data outside the range is removed from the depth map and point cloud.
Visibility	Beginner, Expert, Guru
Values	<ul style="list-style-type: none"> • Lower Limit: 1 to 4000 mm • Upper Limit: 1 to 5000 mm
Instructions	<ul style="list-style-type: none"> • Depth Range should be adjusted to an appropriate range to ensure the completeness of depth data in the depth map and point cloud. Too large a range may introduce interference, and too small a range may remove the data of key features. • For detailed instructions, please refer to Set Depth Range below.

The same point cloud with different **Depth Range** values (all other conditions identical):

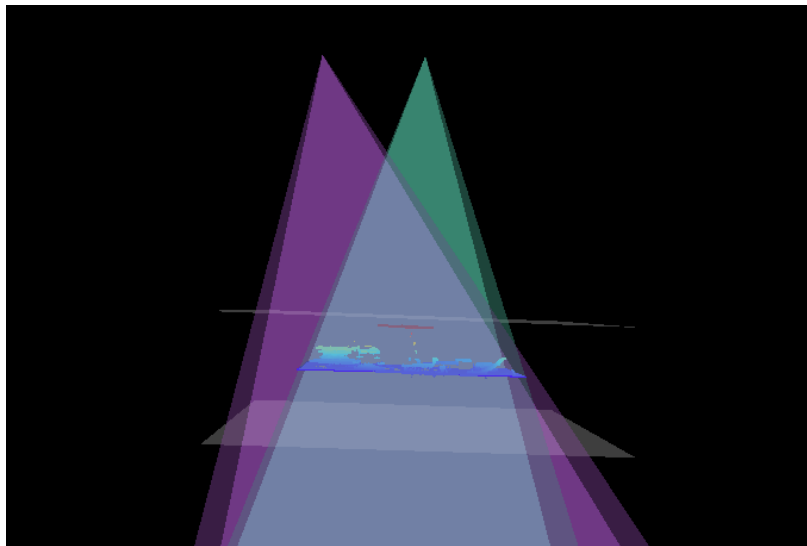




Set Depth Range

Follow these steps to adjust **Depth Range**:

1. Double-click [**Edit**] to the right of **Depth Range** to open the **Set Depth Range** window.
2. Click [**Refresh point cloud**] at the top of the right panel to obtain the newest point cloud.
3. [Adjust the position of the point cloud](#): adjust the position until you can see the two gray rectangles that represents the upper and lower limits of **Depth Range**.



4. Adjust the depth range: drag the handles on the slider to adjust **Depth Range** roughly. Then, enter specific values on the right to fine-tune **Depth Range**.



Judge if the depth range is appropriate: All needed object features are located between the two gray rectangles, and most noise and outliers are located outside.

5. Click [**Save**] to save the depth range settings.



- Click [**Recommended range**] to adjust **Depth Range** to the recommended working distance of the current camera.
- Click [**Reset**] to restore **Depth Range** to the last saved values.

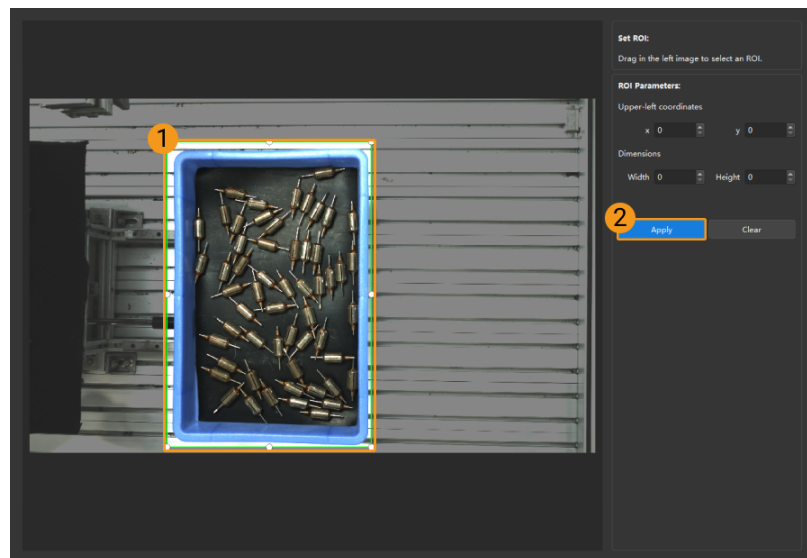
ROI

Description	Set the ROI on the XY plane for the depth map and point cloud. Points outside the selected region are removed.
-------------	--

Visibility	Beginner, Expert, Guru
Values	N/A
Instructions	For detailed instructions, refer to Set ROI below.

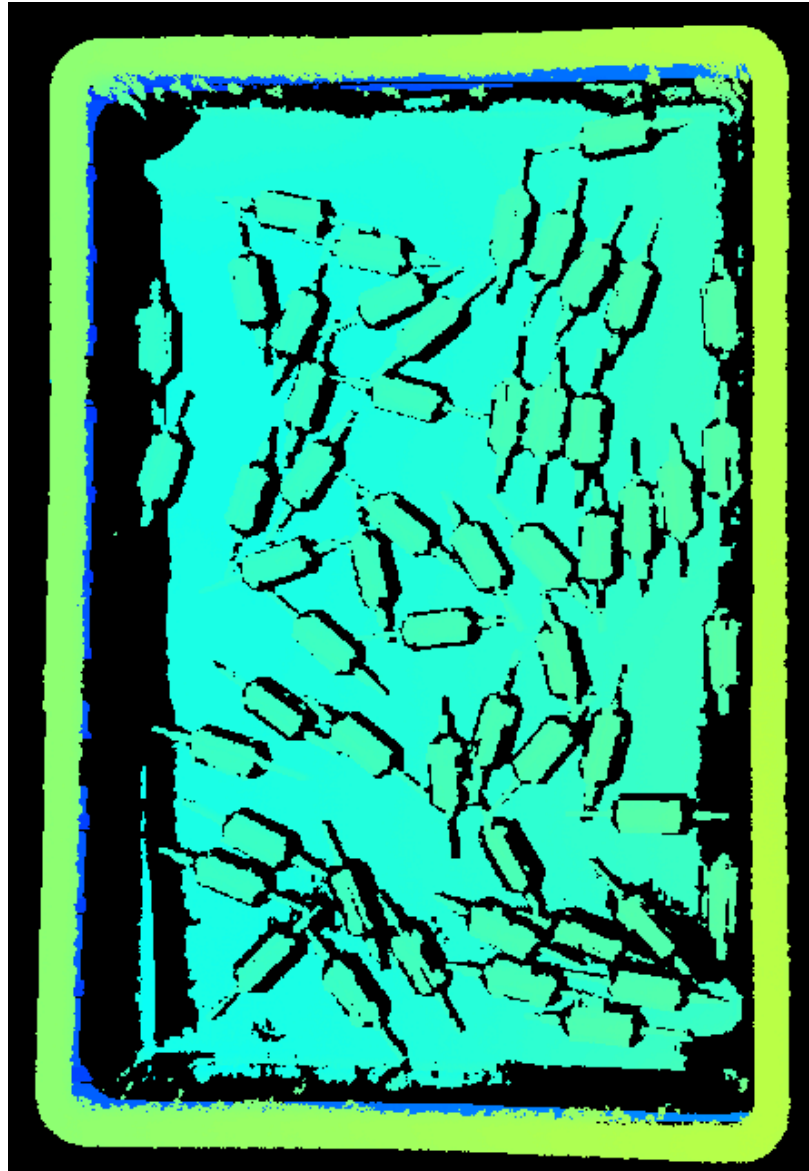
Set ROI

1. Double-click [**Edit**] to the right of **ROI** to open the **Set ROI** window.
2. Select and adjust the ROI on the left. Drag the selection box to move it. Drag the handles on the box to adjust the box size.
3. Click [**Apply**] to apply the set ROI.



- Click [**Clear**] to remove the current ROI.
- For the DEEP (V4) and LSR (V4) series, the 2D image (depth source) is displayed in this window. If the image is too dark or too bright, please adjust **2D Image (Depth Source) Exposure Mode**.

4. Acquire data again and view the depth map or point cloud to check the result of setting the ROI.



5.3.4. Parameters of UHP-140

This topic introduces the parameters of UHP-140. The parameters are divided into parameters for the 2D image and parameters for the depth map / point cloud, based on the data types that they affect.

Parameters for the 2D Image

The 2D image should not be too bright or too dark, and the surface features of the target object should be clearly visible in the image.

The quality of the 2D image is affected by parameters in the **2D Parameters** category and the **Gain** parameter in the **3D Parameters** category.



If you are using a color camera, and the color of the image differ significantly from that of the objects, you can adjust the **white balance** setting. For detailed instructions, please refer to [Adjust White Balance](#).

2D Parameters

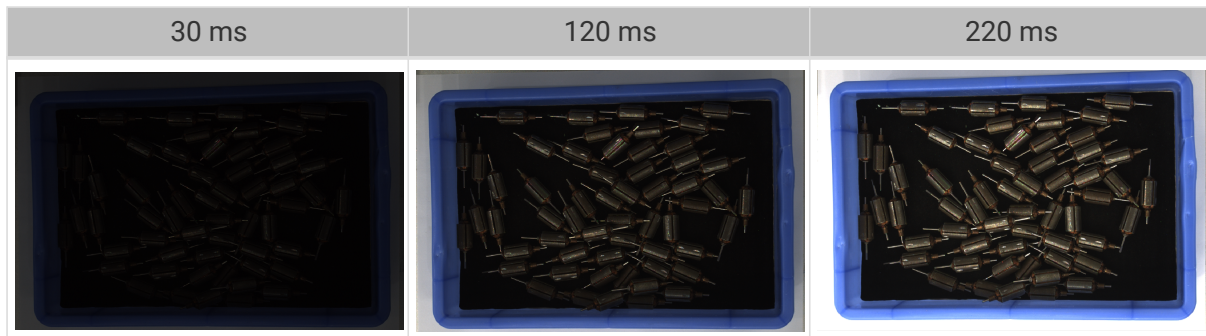
Exposure Mode

Description	Set the exposure mode for capturing the 2D image.
Visibility	Beginner, Expert, Guru
Values	<ul style="list-style-type: none"> • Timed: Set a single exposure time. Usually used in stable lighting conditions. • Auto: The exposure time is automatically adjusted. Usually used in varying lighting conditions. • HDR: Set multiple exposure times and merge the images. Usually used for objects with various colors or textures. • Flash: Use the projector for supplemental light. Usually used in dark environments.
Instructions	<p>After selecting different options, different parameters are displayed in the 2D Parameters category for adjustment:</p> <ul style="list-style-type: none"> • Timed: displays Exposure Time. • Auto: displays Gray Value and Auto-Exposure ROI. • HDR: displays Tone Mapping and Exposure Sequence. • Flash: no other parameters need to be adjusted. The projector projects supplemental light automatically during image capturing.

Timed: Exposure Time

Description	<p>This parameter affects the brightness of the image.</p> <ul style="list-style-type: none"> • Long exposure time increases the image brightness. • Short exposure time decreases the image brightness.
Visibility	Beginner, Expert, Guru
Values	0.1 to 999 ms
Instructions	<p>Adjust this parameter based on the quality of the 2D image. The 2D image should not be too bright or too dark, and the surface features of the target object should be clearly visible in the image.</p> <ul style="list-style-type: none"> • Long exposure time is usually used in dark environments. • Short exposure time is usually used in bright environments.

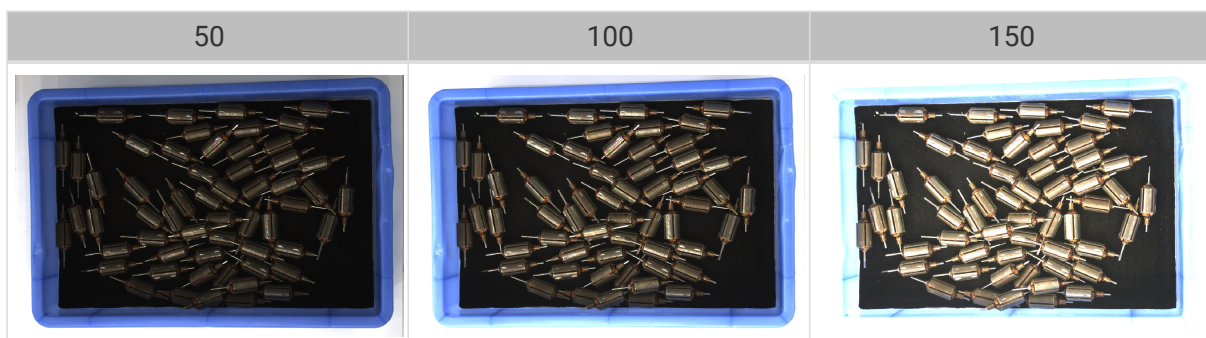
2D images obtained with different **Exposure Time** values (all other conditions identical):



Auto: Gray Value

Description	This parameter affects the brightness of the 2D image. Increase the value if the 2D image is too dark and decrease if too bright.
Visibility	Beginner, Expert, Guru
Values	0 to 255
Instruction	N/A

2D images obtained with different **Gray Value** values (all other conditions identical):

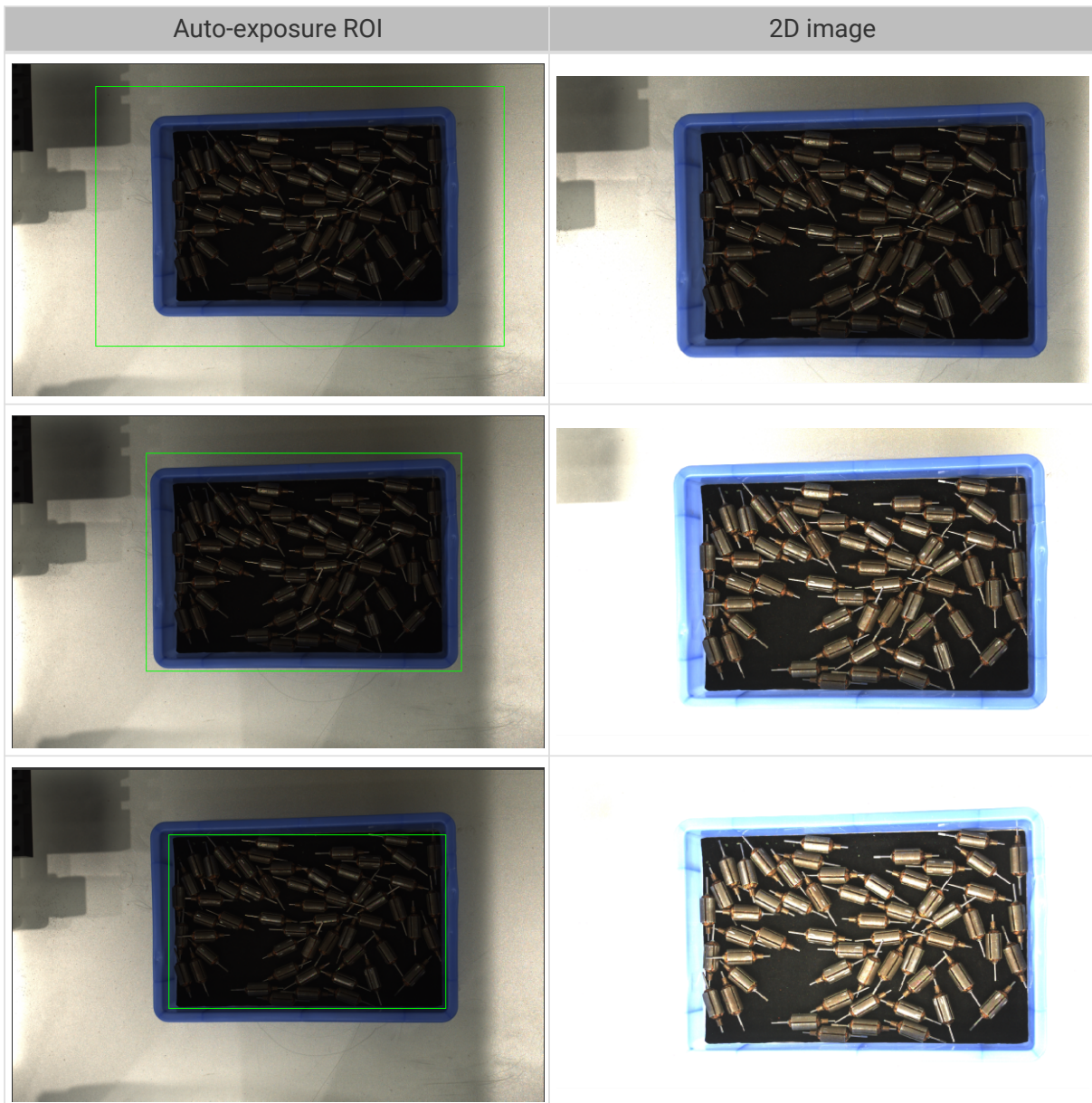


The **Gray Value** of a monochrome image is equivalent to the image brightness; the **Gray Value** of a color image is equivalent to the brightness of each color channel.

Auto: Auto-Exposure ROI

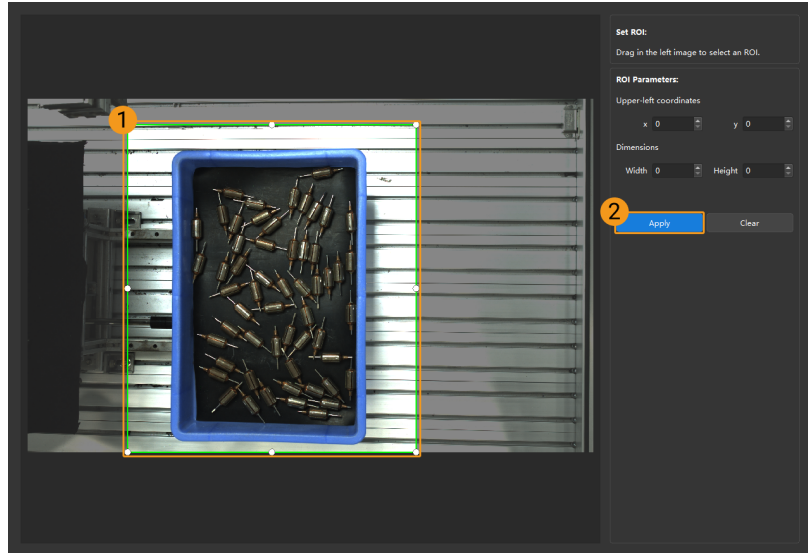
Description	<ul style="list-style-type: none"> The exposure time is automatically adjusted based on the lighting, object colors, etc., in the selected region. When an auto-exposure ROI is not set, the exposure time is automatically adjusted based on the entire camera FOV.
Visibility	Beginner, Expert, Guru
Values	N/A
Instruction	For detailed instructions, refer to Set Auto-Exposure ROI below.

2D images obtained with different **Auto-Exposure ROI** values (all other conditions identical):



Set Auto-Exposure ROI

1. Double-click [**Edit**] to the right of **Auto-Exposure ROI** to open the **Set ROI** window.
2. Select and adjust the ROI on the left. Drag the selection box to move it. Drag the handles on the box to adjust the box size.
3. Click [**Apply**] to apply the set auto-exposure ROI.



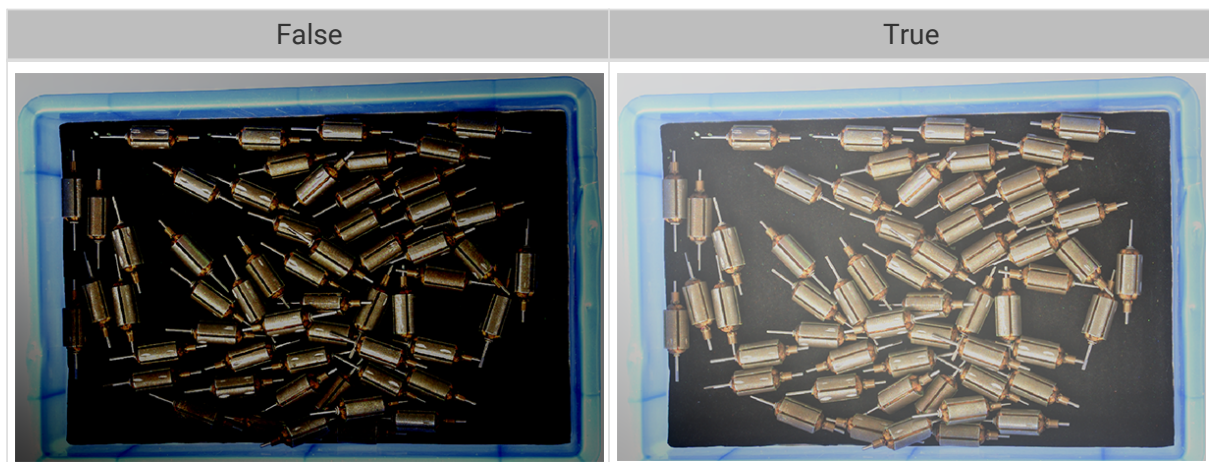
 Click [Clear] to remove the current auto-exposure ROI.

- Acquire data again and view the 2D image to check the result of automatic exposure adjustment.


HDR: Tone Mapping

Description	This function can make the image look more natural. If the 2D image appears very different from the actual objects, please enable this function.
Visibility	Beginner, Expert, Guru
Values	<ul style="list-style-type: none"> • False • True
Instructions	Check to enable Tone Mapping .

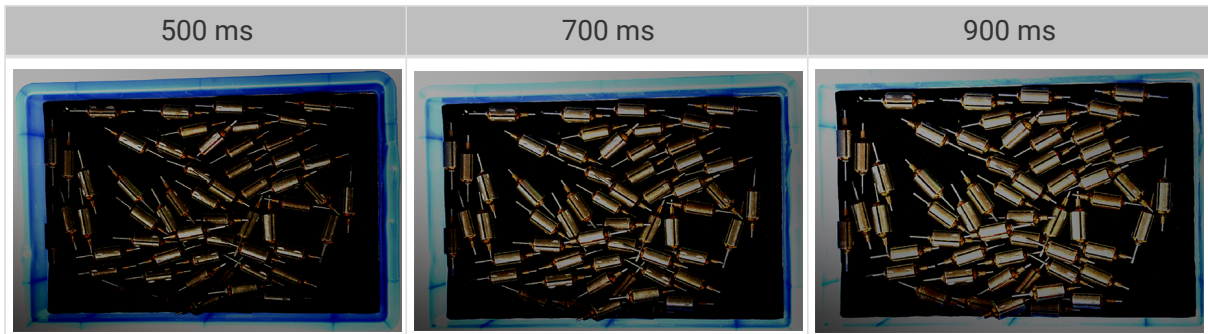
2D images obtained with **Tone Mapping** disabled or enabled (all other conditions identical):



HDR: Exposure Sequence

Description	Set multiple exposure times, and the captured images are merged to generate a 2D image that retains more details in the highlights and shadows.
Visibility	Beginner, Expert, Guru
Values	N/A
Instructions	<ol style="list-style-type: none"> 1. Double-click [Edit] to the right of Exposure Sequence to open the editing window of the exposure sequence. 2. Click [+] to add and set a new exposure time. 3. To delete an exposure time, select it and click [-]. 4. After you finish adding the exposure times, click [Apply] to apply the exposure sequence. <div style="display: flex; align-items: center; margin-top: 10px;"> <div style="margin-right: 10px;">  </div> <ul style="list-style-type: none"> ◦ [Cancel]: do not save the current settings and exit this window. ◦ [Reset]: clear all the exposure times. </div>

2D images captured with single exposure times (all other conditions identical):




2D images captured with different exposure sequences consisting of the above exposure times (all other conditions identical):



3D Parameters

Gain

Description	<p>This parameter affects the brightness of the captured image. At the same time, more noise is introduced.</p> <div style="display: flex; align-items: center; margin-top: 10px;"> <div style="margin-right: 10px;">  </div> <p>This parameter affects the quality of the depth map and point cloud by changing the brightness of the images used for calculating depth data.</p> </div>
Visibility	Expert, Guru

Values	0 to 16 dB
Instructions	When the expected brightness cannot be reached by adjusting Exposure Time , adjust this parameter.

2D images and depth maps obtained with different **Gain** values (all other conditions identical):



Parameters for the Depth Map / Point Cloud

In the depth map and point cloud, the needed data should be complete.



Determine which part of the data is needed according to your actual needs. For example, if you need to pick metal bowls with opening facing up by the rim, usually you only need to make sure that the data of the bowl rims is complete.

Parameters in the following categories affect the quality of the depth map and point cloud.

Parameter category	Depth map	Point cloud
3D Parameters	✓	✓
Point Cloud Processing		✓
Depth Range	✓	✓
ROI	✓	✓

3D Parameters

The parameters in this category affect the images used for calculating depth data, thus affecting the quality of the depth map and point cloud.

Use **Exposure Assistant** can obtain the recommended exposure parameter values. Double-click [**Auto Set**] to the right of **3D Parameters** to open **Exposure Assistant**.

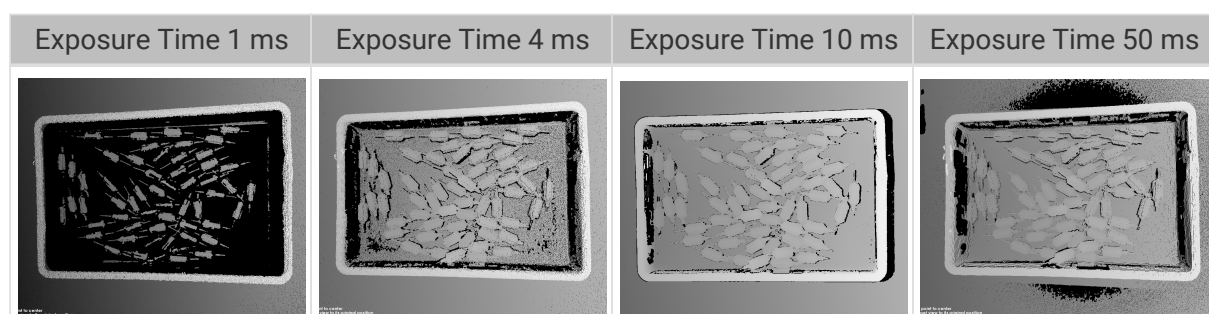
Exposure Multiplier

Description	Set the number of Exposure Time .
Visibility	Beginner, Expert, Guru
Values	1 to 3
Instruction	<ul style="list-style-type: none"> When the value of Exposure Multiplier is greater than 1, multiple Exposure Time parameters must be set. All images captured with different exposure times are used for calculating depth. Using multiple exposure times can improve the completeness of depth data but also increases processing time. With multiple exposure times, the amount of time for obtaining the depth map and point cloud increases. Please try to minimize the number of exposure times while ensuring the data quality.

Exposure Time


Description	Set the exposure time for acquiring depth information. The number of Exposure Time to be set depends on the value of Exposure Multiplier .
Visibility	Beginner, Expert, Guru
Values	<ul style="list-style-type: none"> Value range: 0.1 ms to 99 ms
Instructions	<ul style="list-style-type: none"> Usually, long exposure time is used for dark objects, and short exposure time is used for light objects. Exposure times too long or too short both lead to loss of depth information. The exposure time of a laser camera must be a multiple of 4. If the entered value is not a multiple of 4, it will be automatically adjusted. The minimum value that can be set is 4 ms for the Laser series and 8 ms for other laser cameras.

Point clouds obtained under different **Exposure Time** values (all other conditions identical):



The black regions lack depth information to generate corresponding point cloud.

Gain

Description	<p>This parameter affects the brightness of the captured image. At the same time, more noise is introduced.</p> <p> This parameter affects the quality of the depth map and point cloud by changing the brightness of the images used for calculating depth data.</p>
-------------	--

Visibility	Expert, Guru
Values	0 to 16 dB
Instructions	When the expected brightness cannot be reached by adjusting Exposure Time , adjust this parameter.

2D images and depth maps obtained with different **Gain** values (all other conditions identical):



Projector

Light Brightness

Description	This parameter affects the brightness of the projected structured light.
Visibility	Expert, Guru
Values	<ul style="list-style-type: none"> • High: high brightness, suitable for dark objects. • Normal: medium brightness, suitable for regular objects. • Low: low brightness, suitable for reflective objects.
Instructions	Adjust this parameter based on the object type.

Fringe Coding Mode

Description	Select the pattern of the structured light to be projected.
Visibility	Expert, Guru
Values	<ul style="list-style-type: none"> • Fast: suitable for opaque objects, provides fast acquisition speed, but the depth data quality is lower. • Accurate: suitable for opaque objects, provides high-quality depth data, but the acquisition speed is slower.

Instructions	Adjust this parameter according to the type of target objects and your actual needs for data quality and data acquisition speed.
--------------	--

When the target object is not translucent, the point clouds obtained with **Fringe Coding Mode** set to **Accurate** and **Fast** are as follows (all other conditions identical):



Point Cloud Processing

Adjusting the parameters in the **Point Cloud Processing** category can improve the quality of the point cloud.

Principles for Adjusting Point Cloud Processing Parameters

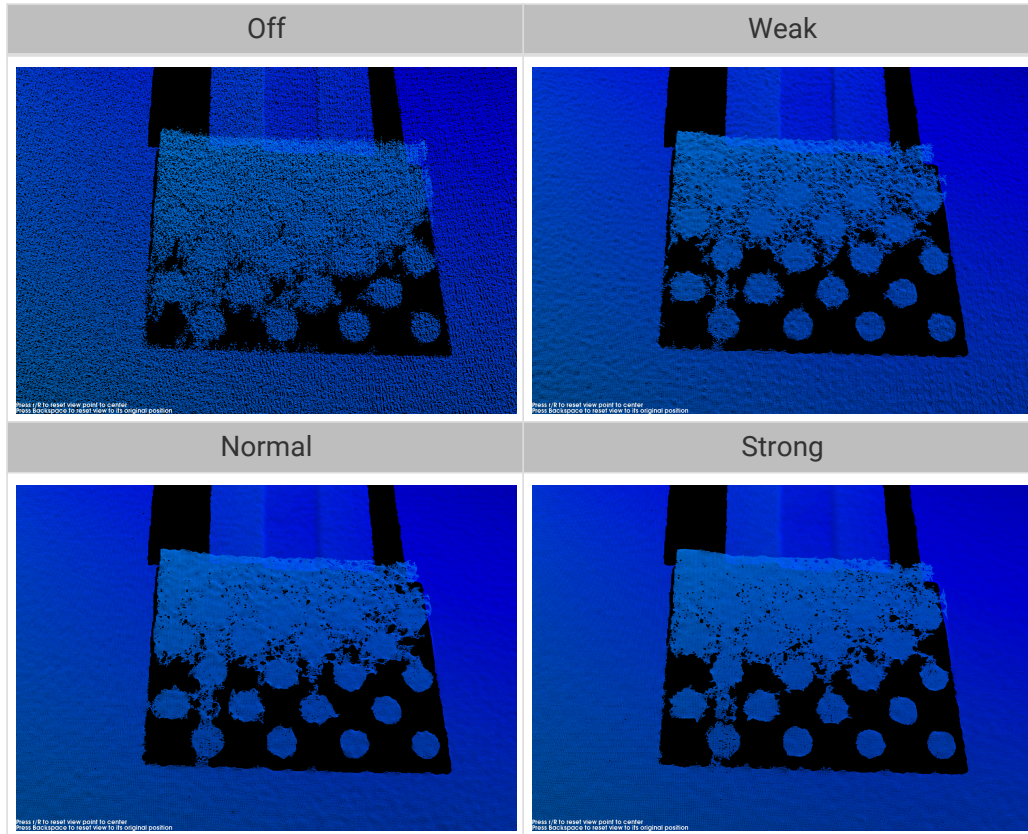
When adjusting the **Point Cloud Processing** parameters, follow these principles to reduce the camera acquisition time and cycle time:

1. Adjust **Outlier Removal** first. The calculation time for different intensities of this parameter is about the same. Using a higher intensity will not add to the calculation time too much.
2. It is recommended to use low intensities of **Surface Smoothing** and **Noise Removal**. The higher the intensities of these two parameters, the longer the calculation time.

Surface Smoothing

Description	This parameter reduces the depth fluctuation in the point cloud and improve its resemblance to the actual object surface.
Visibility	Beginner, Expert, Guru
Values	<ul style="list-style-type: none"> • Off • Weak • Normal • Strong
Instructions	<ul style="list-style-type: none"> • Surface Smoothing causes loss of object surface details. Higher intensity causes more object details to be lost. Lower intensity causes fewer object details to be lost. • Higher intensity results in slower calculation, and lower intensity results in faster calculation.

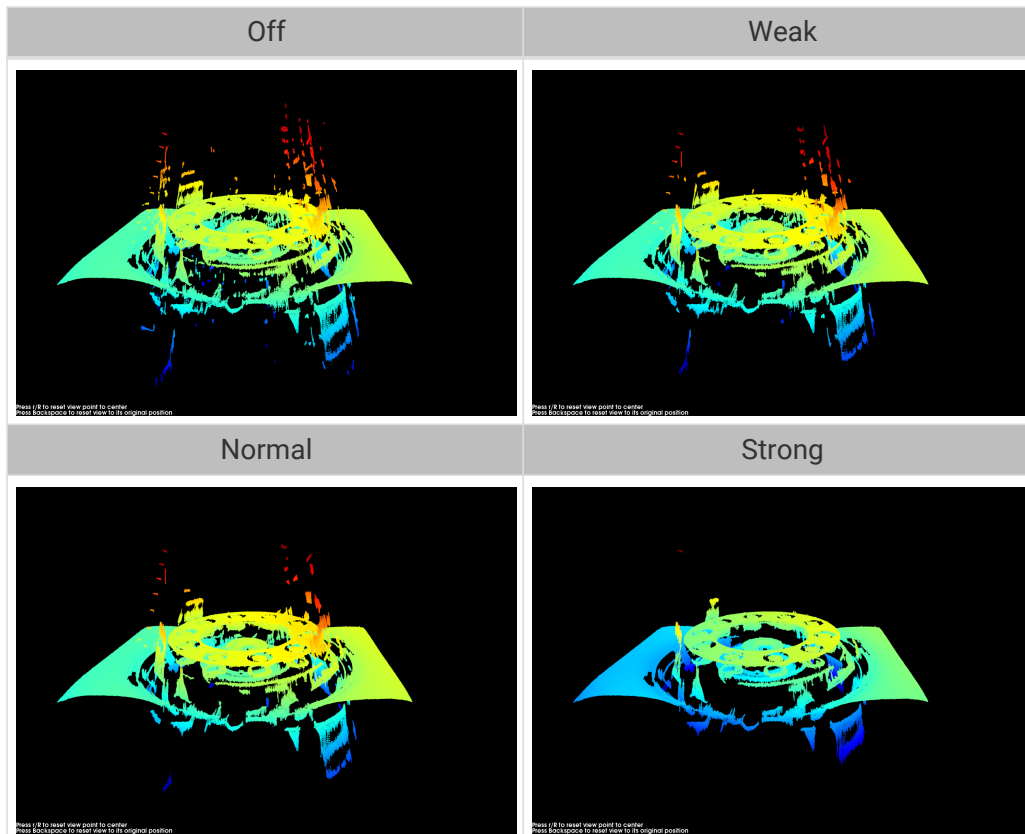
Point clouds obtained with different **Surface Smoothing** values (all other conditions identical):



Outlier Removal

Description	This parameter removes the outliers in the point cloud. Outliers are clustered points away from the object point cloud.
Visibility	Beginner, Expert, Guru
Values	<ul style="list-style-type: none"> • Off • Weak • Normal • Strong
Instructions	<ul style="list-style-type: none"> • Higher intensity removes more outliers, and lower intensity removes fewer outliers. • If the object point cloud contains multiple parts, high intensities of outlier removal might remove some parts. For example, if the object is a cup, outlier removal might remove the point cloud of the handle part.

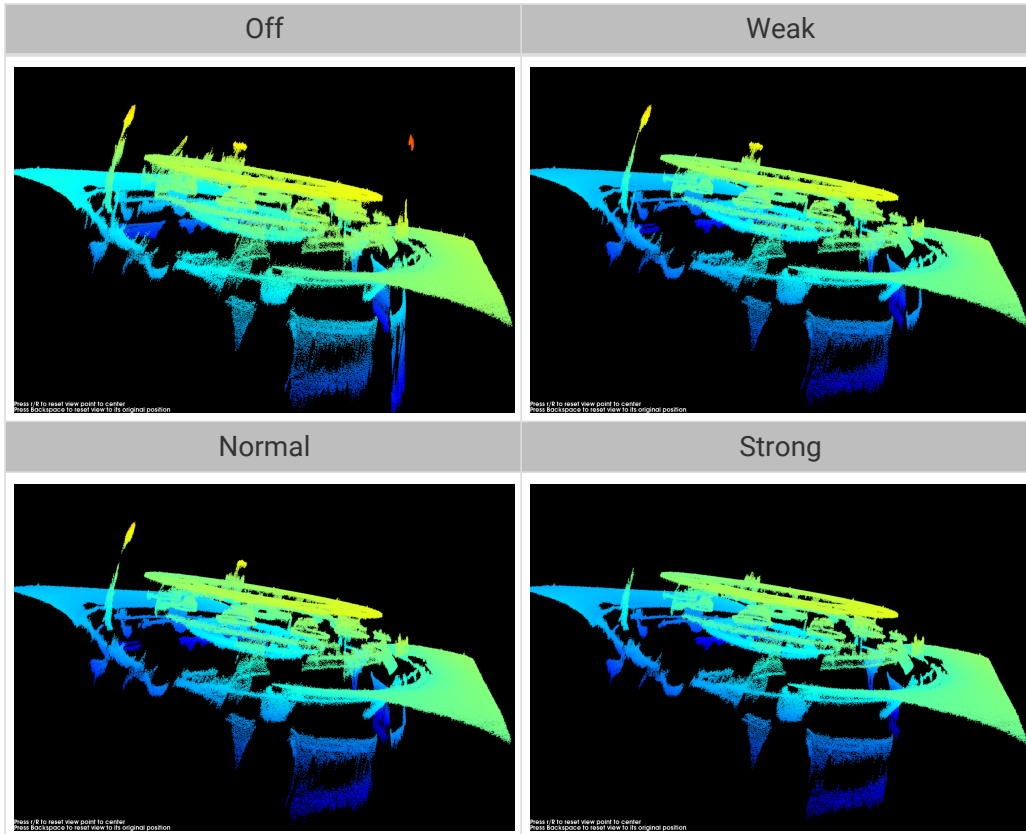
Point clouds obtained with different **Outlier Removal** values (all other conditions identical):



Noise Removal

Description	This parameter removes the noise in the point cloud, thus reducing the impact on the precision and accuracy of subsequent calculation. Noise is the scattered points close to the object surface.
Visibility	Beginner, Expert, Guru
Values	<ul style="list-style-type: none"> • Off • Weak • Normal • Strong
Instructions	<ul style="list-style-type: none"> • Higher intensity removes more noise but may also remove some object features. Lower intensity removes less noise but also preserves the object features better. • Higher intensity results in slower calculation, and lower intensity results in faster calculation.

Point clouds obtained with different **Noise Removal** values (all other conditions identical):



If this function removes the needed object features, please reduce the intensity. However, more noise will be retained.

Distortion Correction

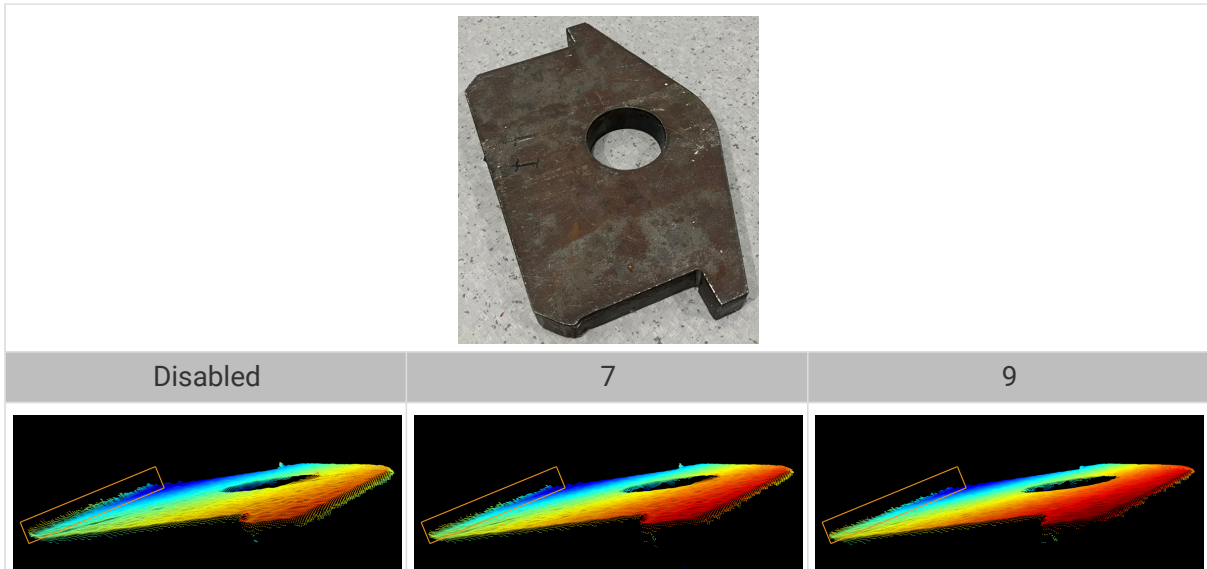
Description	Correct the distortion in the depth data caused by sudden changes in object color or height. Check Enable Distortion Correction first, and then adjust Current Value .
Visibility	Expert, Guru
Values	1 to 10
Instruction	Adjust Current Value based on the degree of distortion in the depth map or point cloud. Acquire data again to see the effect.



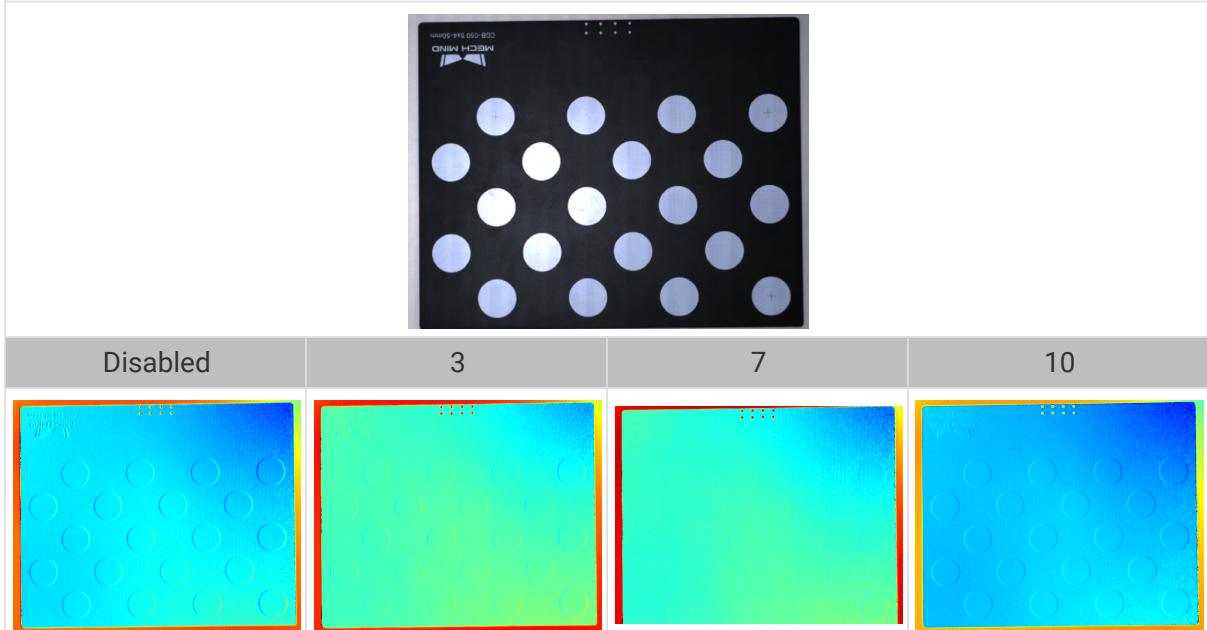
- Enabling distortion correction lowers the acquisition speed.
- A **Current Value** too large may result in adverse effects. Please acquire data again after adjusting the parameter to check its influence on the depth map and point cloud.

Depth maps obtained with different **Current Values** of **Distortion Correction** (all other conditions identical):

The target object is a metal workpiece with a smooth surface and sudden height change at the edges.



The target object is a calibration board with a smooth surface and sudden color change from black to white.



Edge Preservation

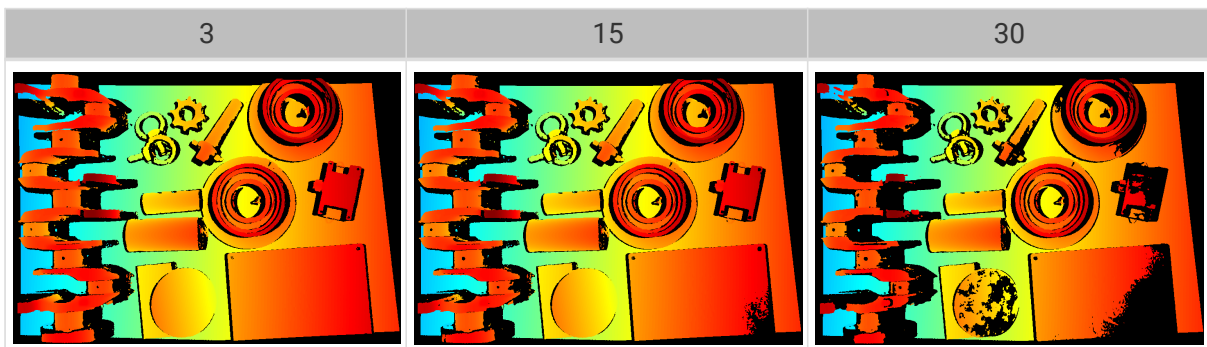
Description	This parameter preserves the sharpness of object edges during surface smoothing.
Visibility	Guru
Values	<ul style="list-style-type: none"> • Sharp: Preserves the sharpness of object edges as much as possible. However, the effect of surface smoothing will be reduced. • Normal: Balances between edge preservation and surface smoothing. • Smooth: Does not preserve the edges. The object surface will be well smoothed, but the object edges will be distorted.

Instructions	Adjust based on your needs for preserving the edges of workpieces.
--------------	--

Stripe Contrast Threshold

Description	This parameter is used to remove noise in the point cloud. Adjust this parameter if the level of noise in the point cloud is still high after adjusting Outlier Removal and Noise Removal .
Visibility	Beginner, Expert, Guru
Values	1 to 100
Instructions	<ul style="list-style-type: none"> • Greater value of this parameter results in more points being removed. Smaller value results in fewer points being removed. • Increase the value of this parameter can remove the noise in the point cloud, but may also cause the points of dark objects to be lost.

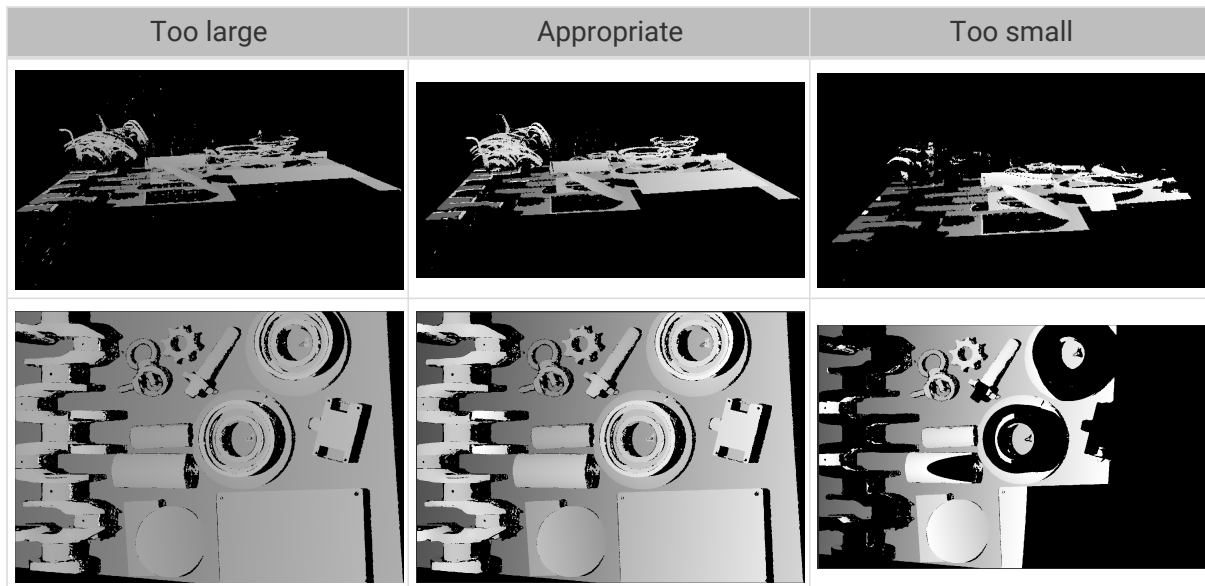
Point clouds obtained with different **Stripe Contrast Threshold** values (all other conditions identical):



Depth Range

Description	Set a Z-direction ROI for the depth map and point cloud. Depth Range can be set within the camera working distance, and the data outside the range is removed from the depth map and point cloud.
Visibility	Beginner, Expert, Guru
Values	<ul style="list-style-type: none"> • Lower Limit: 1 to 4000 mm • Upper Limit: 1 to 5000 mm
Instructions	<ul style="list-style-type: none"> • Depth Range should be adjusted to an appropriate range to ensure the completeness of depth data in the depth map and point cloud. Too large a range may introduce interference, and too small a range may remove the data of key features. • For detailed instructions, please refer to Set Depth Range below.

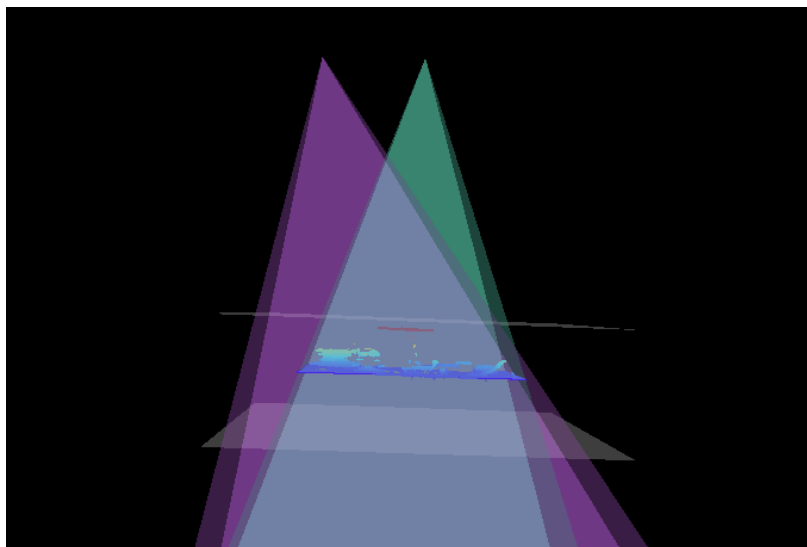
The same point cloud with different **Depth Range** values (all other conditions identical):



Set Depth Range

Follow these steps to adjust **Depth Range**:

1. Double-click [**Edit**] to the right of **Depth Range** to open the **Set Depth Range** window.
2. Click [**Refresh point cloud**] at the top of the right panel to obtain the newest point cloud.
3. **Adjust the position of the point cloud**: adjust the position until you can see the two gray rectangles that represents the upper and lower limits of **Depth Range**.



4. Adjust the depth range: drag the handles on the slider to adjust **Depth Range** roughly. Then, enter specific values on the right to fine-tune **Depth Range**.



Judge if the depth range is appropriate: All needed object features are located between the two gray rectangles, and most noise and outliers are located outside.

5. Click [**Save**] to save the depth range settings.



- Click [**Recommended range**] to adjust **Depth Range** to the recommended working distance of the current camera.

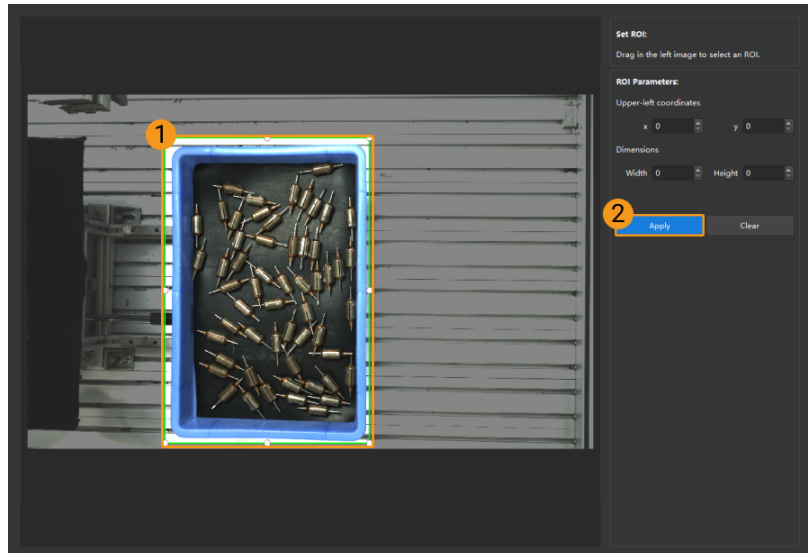
- Click [**Reset**] to restore **Depth Range** to the last saved values.

ROI

Description	Set the ROI on the XY plane for the depth map and point cloud. Points outside the selected region are removed.
Visibility	Beginner, Expert, Guru
Values	N/A
Instructions	For detailed instructions, refer to Set ROI below.

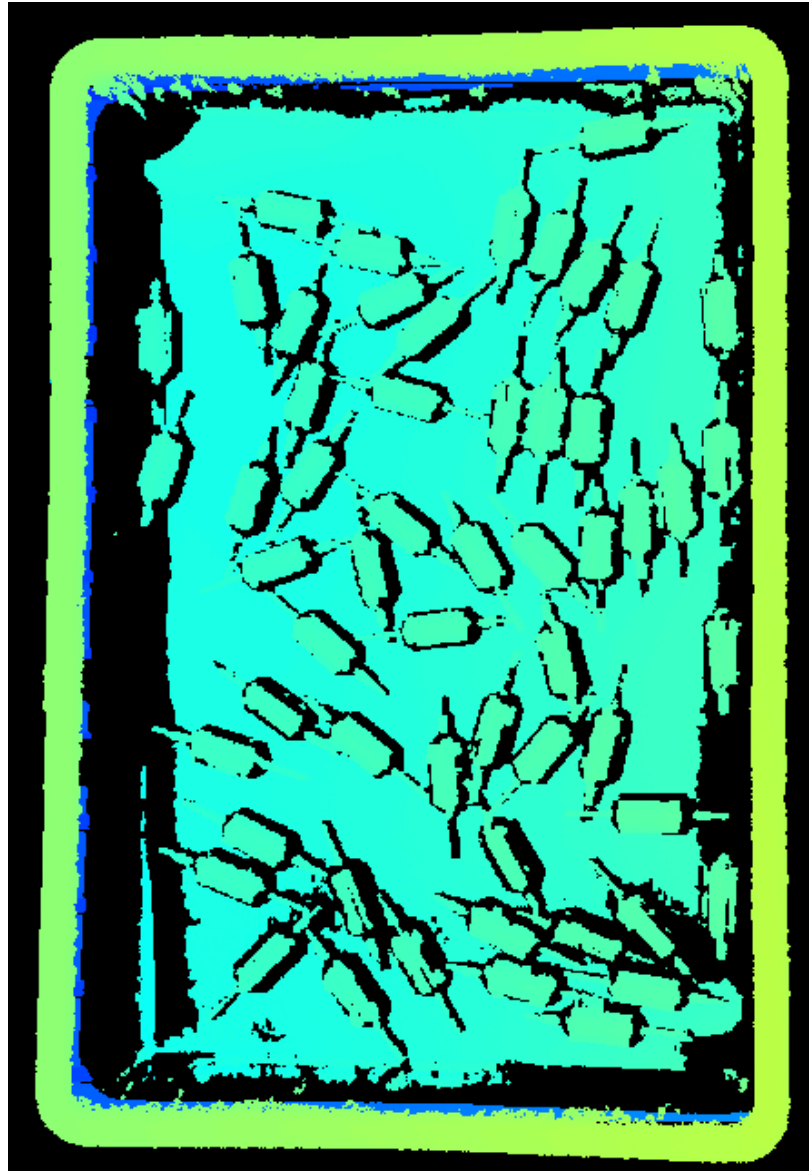
Set ROI

1. Double-click [**Edit**] to the right of **ROI** to open the **Set ROI** window.
2. Select and adjust the ROI on the left. Drag the selection box to move it. Drag the handles on the box to adjust the box size.
3. Click [**Apply**] to apply the set ROI.



- Click [**Clear**] to remove the current ROI.
- For the DEEP (V4) and LSR (V4) series, the 2D image (depth source) is displayed in this window. If the image is too dark or too bright, please adjust **2D Image (Depth Source) Exposure Mode**.

4. Acquire data again and view the depth map or point cloud to check the result of setting the ROI.



5.3.5. Parameters of DEEP / LSR Series

This topic introduces the parameters of the DEEP / LSR series. The parameters are divided into parameters for the 2D image and parameters for the depth map / point cloud, based on the data types that they affect.

Parameters for the 2D Image

The DEEP and LSR series provide two types of 2D images: 2D image (texture) and 2D images (depth source). The usage of each type is as follows:

2D image type	Usage
2D image (texture)	Texture the point cloud
2D images (depth source)	Check intrinsic parameters
	Set ROI
	Perform hand-eye calibration

The 2D image should not be too bright or too dark, and the surface features of the target object should be clearly visible in the image.

The quality of the 2D image is affected by parameters in the **2D Parameters** category and the **Gain** parameter in the **3D Parameters** category.



If you are using a color camera, and the color of the image differ significantly from that of the objects, you can adjust the **white balance** setting. For detailed instructions, please refer to [Adjust White Balance](#).

2D Parameters

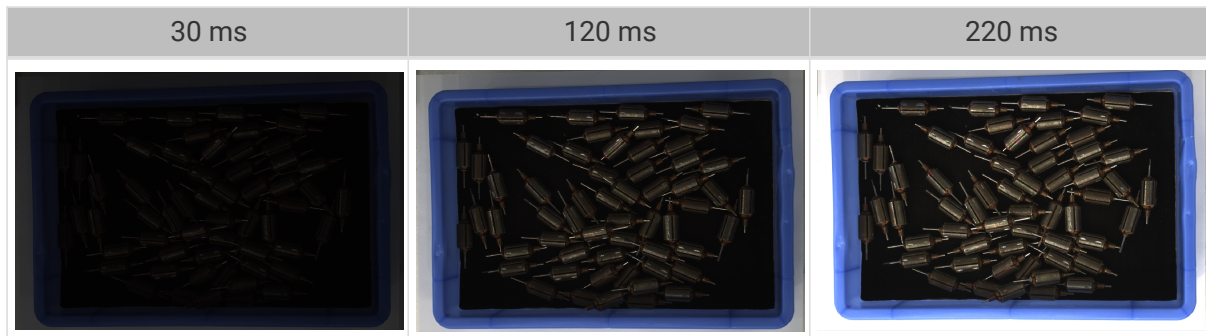
2D Image (Texture) Exposure Mode

Parameter Description	Set the exposure mode for capturing the 2D image (texture).
Visibility	Beginner, Expert, Guru
Values	<ul style="list-style-type: none"> • Timed: Set a single exposure time. Usually used in stable lighting conditions. • Auto: The exposure time is automatically adjusted. Usually used in varying lighting conditions. • HDR: Set multiple exposure times and merge the images. Usually used for objects with various colors or textures.
Instruction	After selecting different options, different parameters are displayed in the 2D Parameters category for adjustment: <ul style="list-style-type: none"> • Timed: displays Exposure Time. • Auto: displays Gray Value and Auto-Exposure ROI. • HDR: displays Tone Mapping and Exposure Sequence.

Timed: Exposure Time

Description	This parameter affects the brightness of the image. <ul style="list-style-type: none"> • Long exposure time increases the image brightness. • Short exposure time decreases the image brightness.
Visibility	Beginner, Expert, Guru
Values	0.1 to 999 ms
Instructions	Adjust this parameter based on the quality of the 2D image. The 2D image should not be too bright or too dark, and the surface features of the target object should be clearly visible in the image. <ul style="list-style-type: none"> • Long exposure time is usually used in dark environments. • Short exposure time is usually used in bright environments.

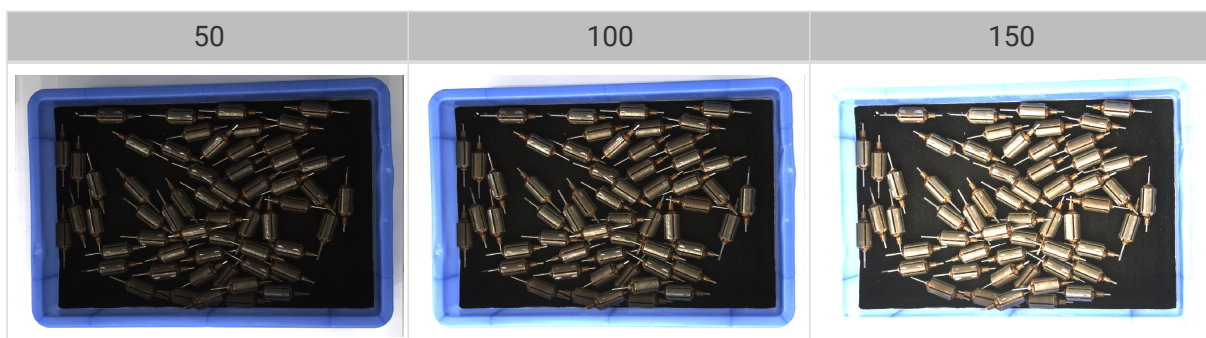
2D images obtained with different **Exposure Time** values (all other conditions identical):



Auto: Gray Value

Description	This parameter affects the brightness of the 2D image. Increase the value if the 2D image is too dark and decrease if too bright.
Visibility	Beginner, Expert, Guru
Values	0 to 255
Instruction	N/A

2D images obtained with different **Gray Value** values (all other conditions identical):

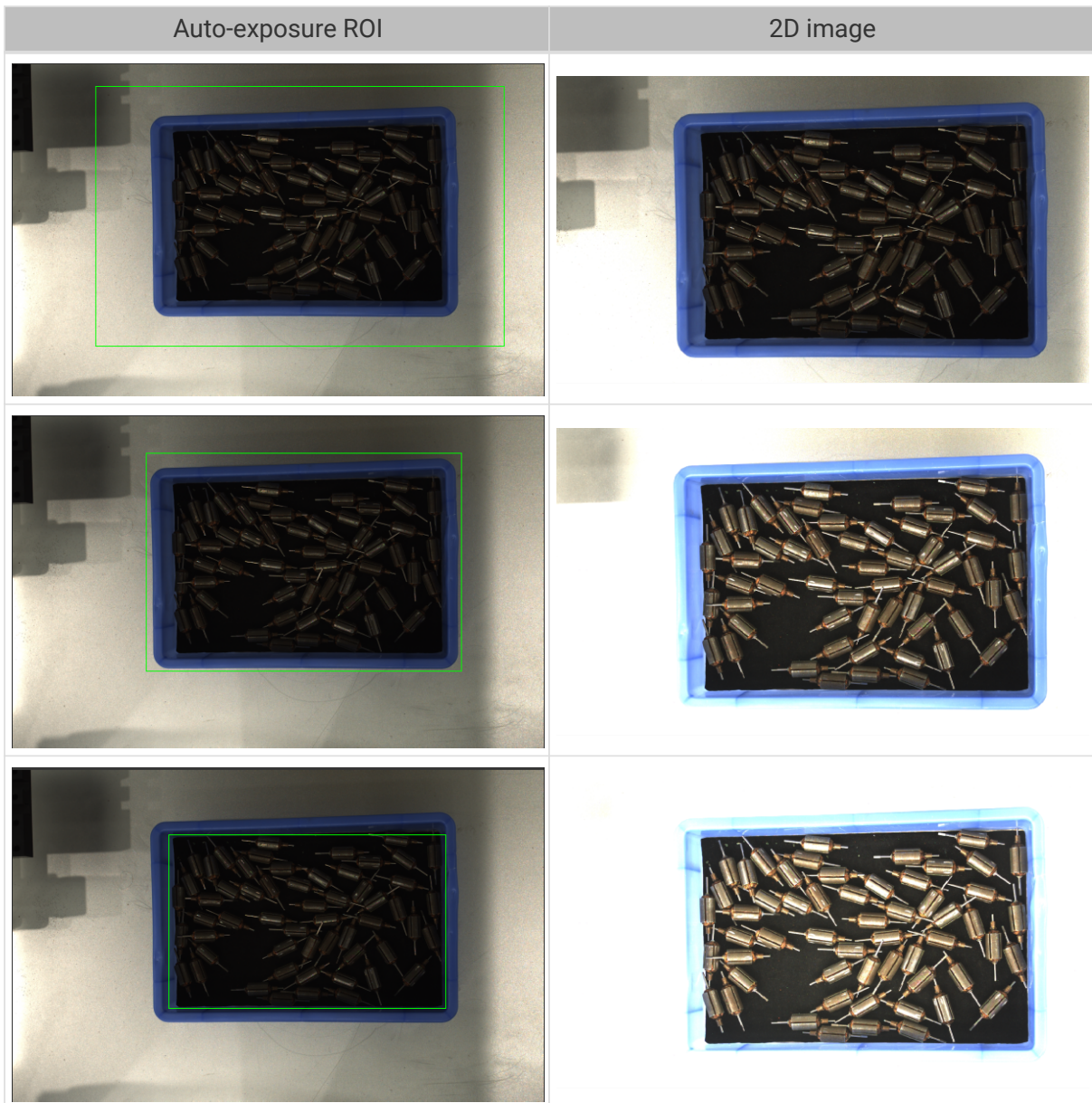


The **Gray Value** of a monochrome image is equivalent to the image brightness; the **Gray Value** of a color image is equivalent to the brightness of each color channel.

Auto: Auto-Exposure ROI

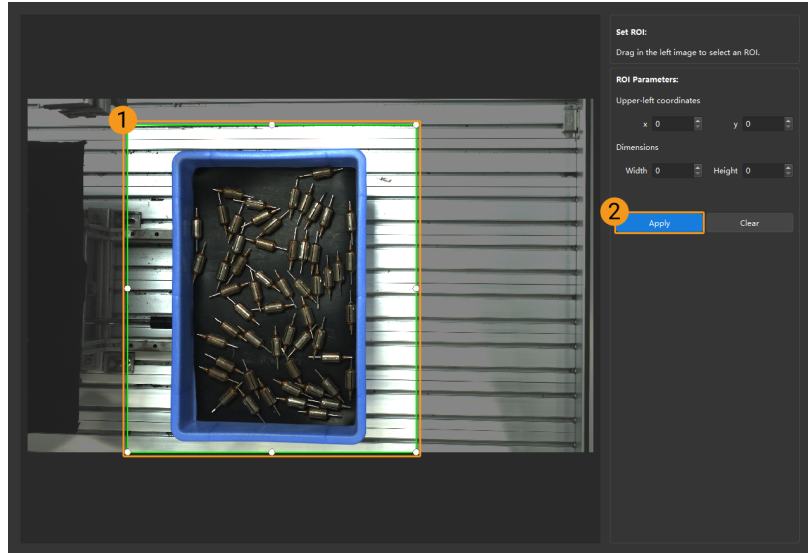
Description	<ul style="list-style-type: none"> The exposure time is automatically adjusted based on the lighting, object colors, etc., in the selected region. When an auto-exposure ROI is not set, the exposure time is automatically adjusted based on the entire camera FOV.
Visibility	Beginner, Expert, Guru
Values	N/A
Instruction	For detailed instructions, refer to Set Auto-Exposure ROI below.

2D images obtained with different **Auto-Exposure ROI** values (all other conditions identical):



Set Auto-Exposure ROI

1. Double-click [**Edit**] to the right of **Auto-Exposure ROI** to open the **Set ROI** window.
2. Select and adjust the ROI on the left. Drag the selection box to move it. Drag the handles on the box to adjust the box size.
3. Click [**Apply**] to apply the set auto-exposure ROI.



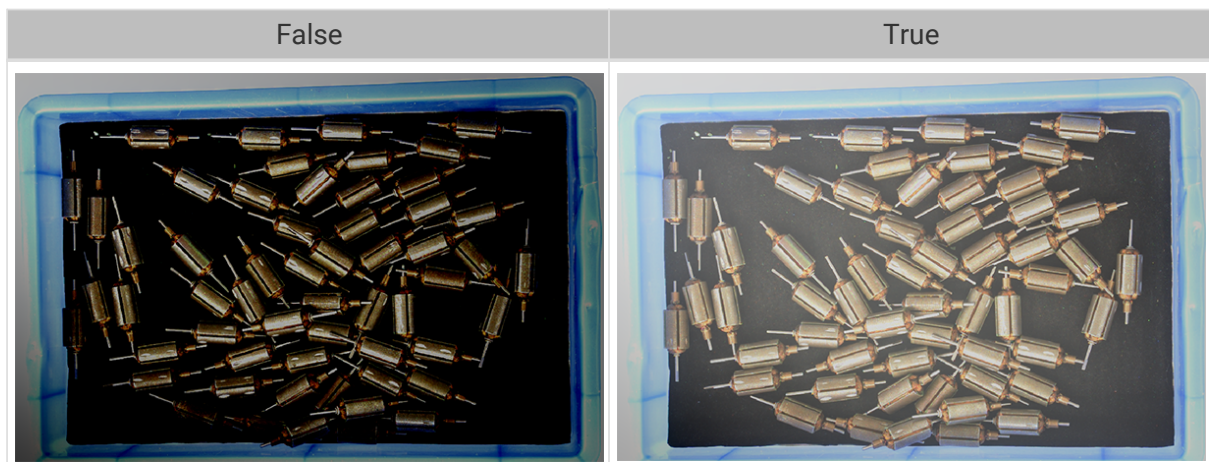
 Click [Clear] to remove the current auto-exposure ROI.

- Acquire data again and view the 2D image to check the result of automatic exposure adjustment.


HDR: Tone Mapping

Description	This function can make the image look more natural. If the 2D image appears very different from the actual objects, please enable this function.
Visibility	Beginner, Expert, Guru
Values	<ul style="list-style-type: none"> • False • True
Instructions	Check to enable Tone Mapping .

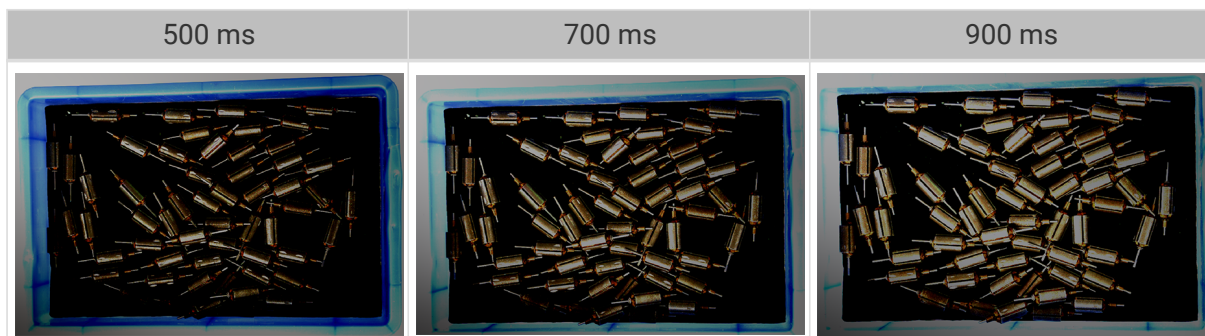
2D images obtained with **Tone Mapping** disabled or enabled (all other conditions identical):



HDR: Exposure Sequence

Description	Set multiple exposure times, and the captured images are merged to generate a 2D image that retains more details in the highlights and shadows.
Visibility	Beginner, Expert, Guru
Values	N/A
Instructions	<ol style="list-style-type: none"> 1. Double-click [Edit] to the right of Exposure Sequence to open the editing window of the exposure sequence. 2. Click [+] to add and set a new exposure time. 3. To delete an exposure time, select it and click [-]. 4. After you finish adding the exposure times, click [Apply] to apply the exposure sequence. <div style="display: flex; align-items: center; margin-top: 10px;"> <div style="margin-right: 10px;">  </div> <ul style="list-style-type: none"> ◦ [Cancel]: do not save the current settings and exit this window. ◦ [Reset]: clear all the exposure times. </div>

2D images captured with single exposure times (all other conditions identical):



2D images captured with different exposure sequences consisting of the above exposure times (all other conditions identical):



2D Image (Depth Source) Exposure Mode

Parameter Description	Set the exposure mode for capturing the 2D images (depth source).
Visibility	Beginner, Expert, Guru
Values	<ul style="list-style-type: none"> • Timed: Set a single exposure time. Usually used in stable lighting conditions. • Flash: Use the projector for supplemental light. Usually used in dark environments.

Instructions	After selecting different options, different parameters are displayed in the 2D Parameters category for adjustment: <ul style="list-style-type: none"> • Timed: displays 2D Image (Depth Source) Exposure Time. • Flash: no other parameters need to be adjusted. The projector projects supplemental light automatically during image capturing.
--------------	---

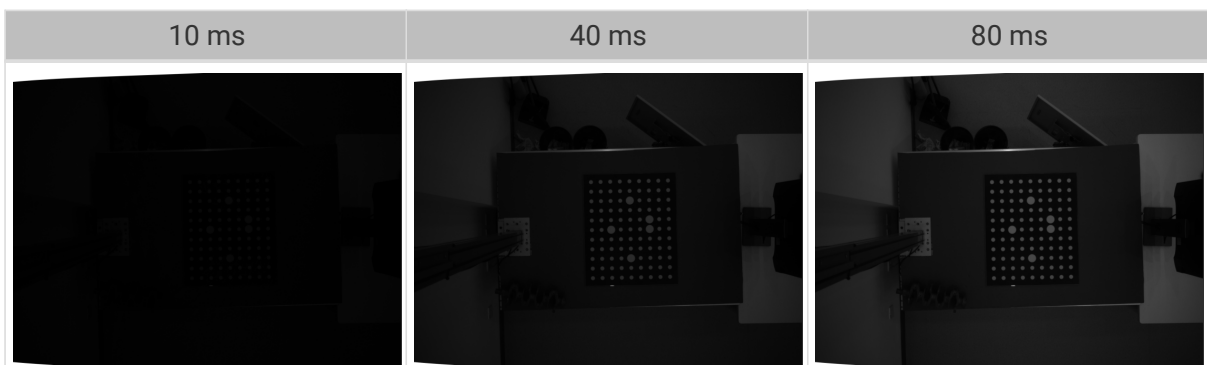


When using a LSR or DEEP series camera with Mech-Vision, the “Color Image” port in Mech-Vision outputs the 2D image (depth source). If you need to adjust the “Color Image” in Mech-Vision, please adjust **2D Image (Depth Source) Exposure Mode**.

Timed: 2D Image (Depth Source) Exposure Time


Description	This parameter affects the brightness of the image. <ul style="list-style-type: none"> • Long exposure time increases the image brightness. • Short exposure time decreases the image brightness.
Visibility	Beginner, Expert, Guru
Values	0.1 to 999 ms
Instructions	Adjust this parameter based on the quality of the 2D image. The 2D image should not be too bright or too dark, and the surface features of the target object should be clearly visible in the image. <ul style="list-style-type: none"> • Long exposure time is usually used in dark environments. • Short exposure time is usually used in bright environments.

2D images obtained with different **2D Image (Depth Source) Exposure Time** values (all other conditions identical):



3D Parameters

Gain

Description	This parameter affects the brightness of the captured image. At the same time, more noise is introduced. <div style="display: flex; align-items: center; margin-top: 10px;">  This parameter affects the quality of the depth map and point cloud by changing the brightness of the images used for calculating depth data. </div>
Visibility	Expert, Guru

Values	0 to 16 dB
Instructions	When the expected brightness cannot be reached by adjusting Exposure Time , adjust this parameter.

2D images and depth maps obtained with different **Gain** values (all other conditions identical):



Parameters for the Depth Map / Point Cloud

In the depth map and point cloud, the needed data should be complete.



Determine which part of the data is needed according to your actual needs. For example, if you need to pick metal bowls with opening facing up by the rim, usually you only need to make sure that the data of the bowl rims is complete.

Parameters in the following categories affect the quality of the depth map and point cloud.

Parameter category	Depth map	Point cloud
3D Parameters	✓	✓
Point Cloud Processing		✓
Depth Range	✓	✓
ROI	✓	✓

3D Parameters

The parameters in this category affect the images used for calculating depth data, thus affecting the quality of the depth map and point cloud.

Use **Exposure Assistant** can obtain the recommended exposure parameter values. Double-click [**Auto Set**] to the right of **3D Parameters** to open **Exposure Assistant**.

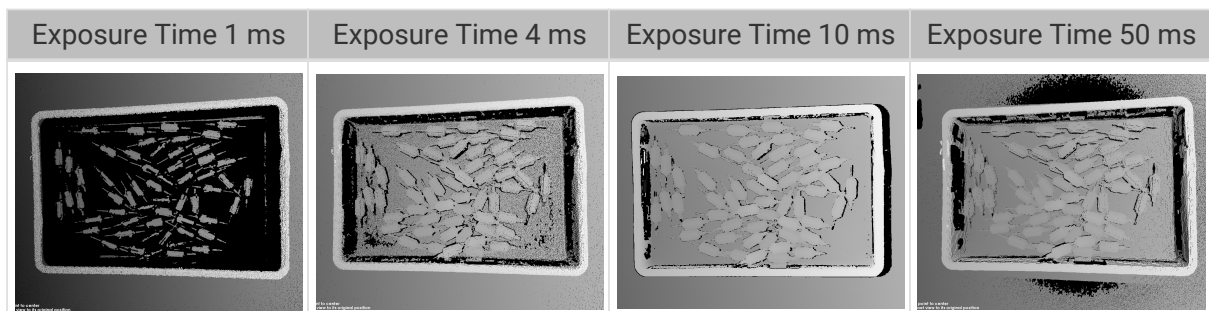
Exposure Multiplier

Description	Set the number of Exposure Time .
Visibility	Beginner, Expert, Guru
Values	1 to 3
Instruction	<ul style="list-style-type: none"> When the value of Exposure Multiplier is greater than 1, multiple Exposure Time parameters must be set. All images captured with different exposure times are used for calculating depth. Using multiple exposure times can improve the completeness of depth data but also increases processing time. With multiple exposure times, the amount of time for obtaining the depth map and point cloud increases. Please try to minimize the number of exposure times while ensuring the data quality.

Exposure Time


Description	Set the exposure time for acquiring depth information. The number of Exposure Time to be set depends on the value of Exposure Multiplier .
Visibility	Beginner, Expert, Guru
Values	<ul style="list-style-type: none"> Value range: 0.1 ms to 99 ms
Instructions	<ul style="list-style-type: none"> Usually, long exposure time is used for dark objects, and short exposure time is used for light objects. Exposure times too long or too short both lead to loss of depth information. The exposure time of a laser camera must be a multiple of 4. If the entered value is not a multiple of 4, it will be automatically adjusted. The minimum value that can be set is 4 ms for the Laser series and 8 ms for other laser cameras.

Point clouds obtained under different **Exposure Time** values (all other conditions identical):



The black regions lack depth information to generate corresponding point cloud.

Gain

Description	<p>This parameter affects the brightness of the captured image. At the same time, more noise is introduced.</p> <p> This parameter affects the quality of the depth map and point cloud by changing the brightness of the images used for calculating depth data.</p>
-------------	--

Visibility	Expert, Guru
Values	0 to 16 dB
Instructions	When the expected brightness cannot be reached by adjusting Exposure Time , adjust this parameter.

2D images and depth maps obtained with different **Gain** values (all other conditions identical):

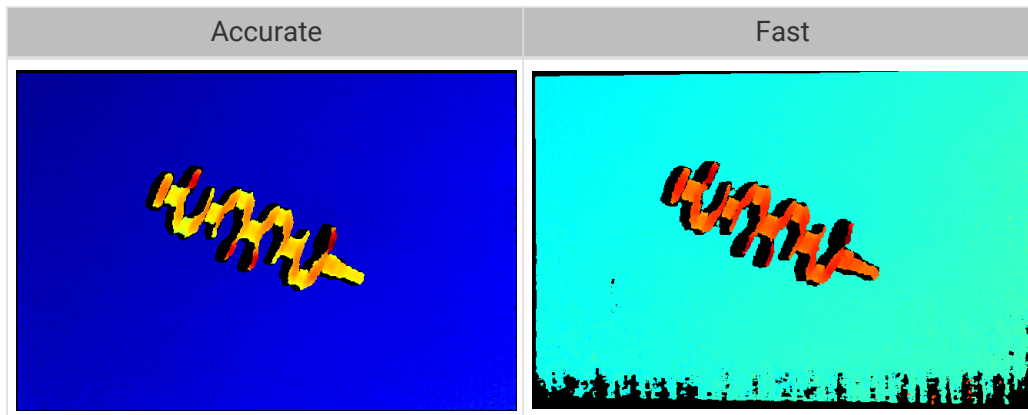


Laser

Fringe Coding Mode

Parameter Description	Change the structured light patterns used, which affects the data acquisition speed and depth data quality.
Visibility	Expert, Guru
Values	<ul style="list-style-type: none"> • Fast: faster data acquisition, but the quality of the depth map and point cloud is lower. • Accurate: slower data acquisition, but the quality of the depth map and point cloud is higher.
Instructions	Adjust this parameter according to your actual needs for data quality and data acquisition speed.

Point clouds obtained with different **Fringe Coding Mode** values (all other conditions identical):



Laser Power

Description	Set the power of the laser projector, which affects the brightness of the projected structured light.
Visibility	Expert, Guru
Values	50 to 100%
Instruction	Usually, using the default value is sufficient. <ul style="list-style-type: none"> • Greater value results in higher structured light brightness; smaller value results in lower structured light brightness. • Greater values are suitable for dark objects, and smaller values are suitable for reflective objects.

Point Cloud Processing

Adjusting the parameters in the **Point Cloud Processing** category can improve the quality of the point cloud.

Principles for Adjusting Point Cloud Processing Parameters

When adjusting the **Point Cloud Processing** parameters, follow these principles to reduce the camera acquisition time and cycle time:

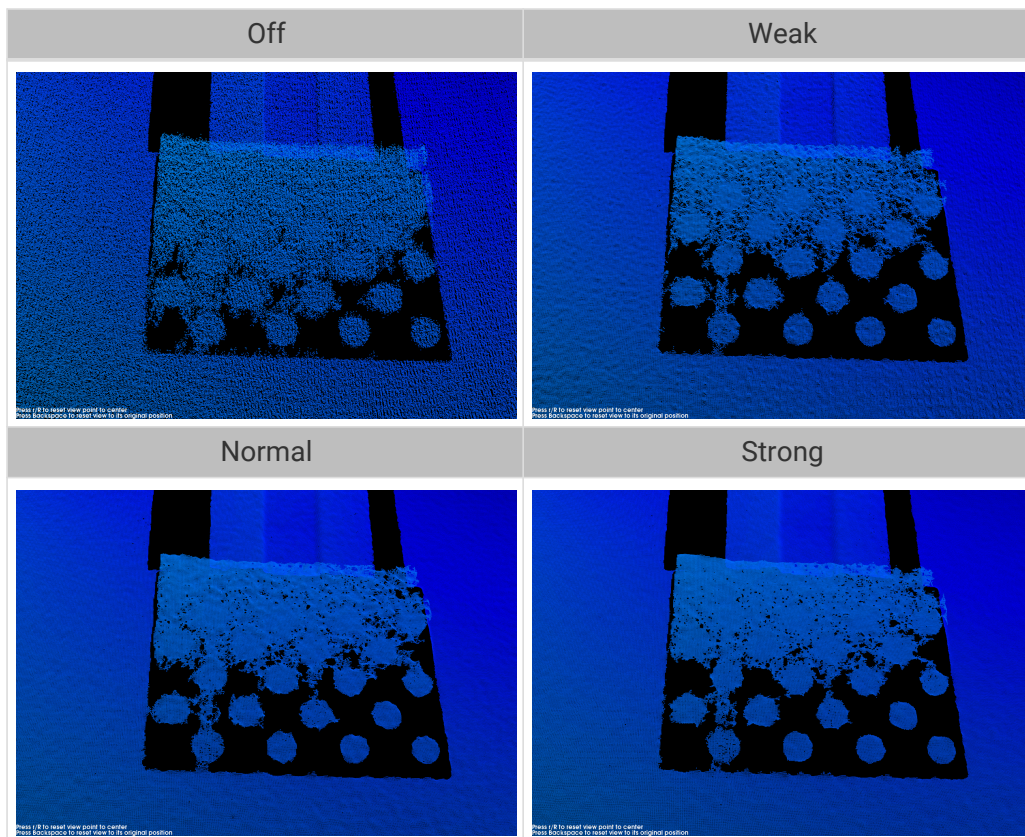
1. Adjust **Outlier Removal** first. The calculation time for different intensities of this parameter is about the same. Using a higher intensity will not add to the calculation time too much.
2. It is recommended to use low intensities of **Surface Smoothing** and **Noise Removal**. The higher the intensities of these two parameters, the longer the calculation time.

Surface Smoothing

Description	This parameter reduces the depth fluctuation in the point cloud and improve its resemblance to the actual object surface.
Visibility	Beginner, Expert, Guru

Values	<ul style="list-style-type: none"> • Off • Weak • Normal • Strong
Instructions	<ul style="list-style-type: none"> • Surface Smoothing causes loss of object surface details. Higher intensity causes more object details to be lost. Lower intensity causes fewer object details to be lost. • Higher intensity results in slower calculation, and lower intensity results in faster calculation.

Point clouds obtained with different **Surface Smoothing** values (all other conditions identical):

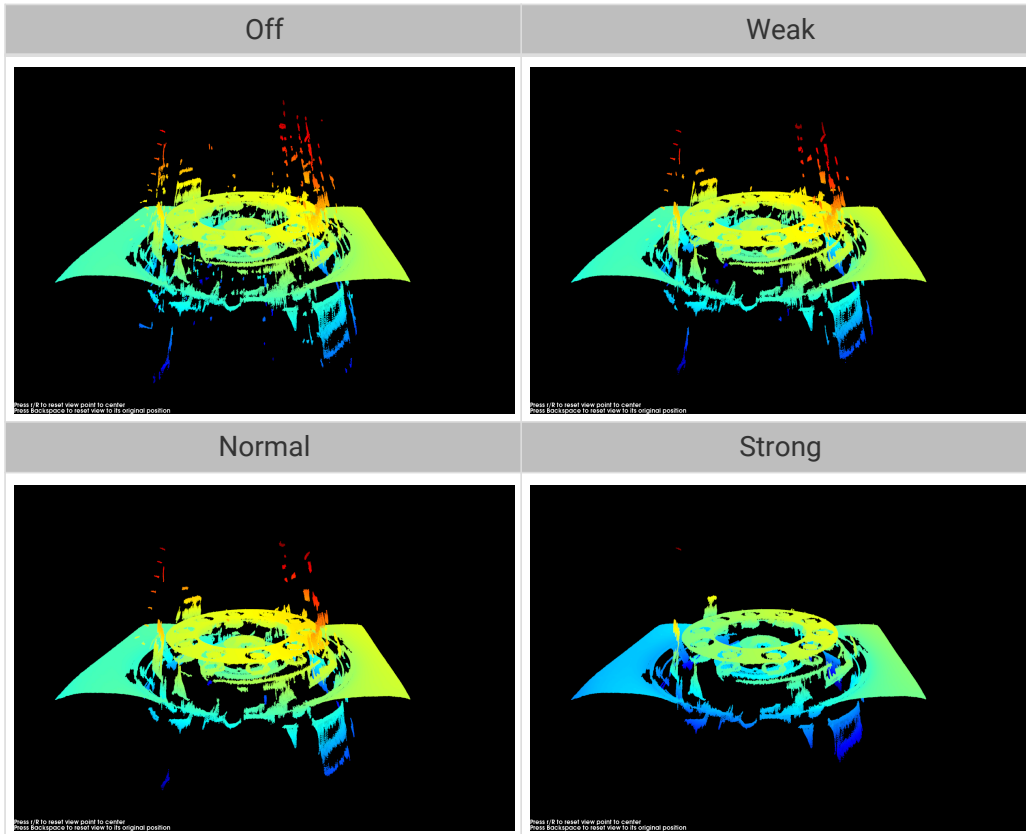


Outlier Removal

Description	This parameter removes the outliers in the point cloud. Outliers are clustered points away from the object point cloud.
Visibility	Beginner, Expert, Guru
Values	<ul style="list-style-type: none"> • Off • Weak • Normal • Strong

Instruction s	<ul style="list-style-type: none"> Higher intensity removes more outliers, and lower intensity removes fewer outliers. If the object point cloud contains multiple parts, high intensities of outlier removal might remove some parts. For example, if the object is a cup, outlier removal might remove the point cloud of the handle part.
------------------	--

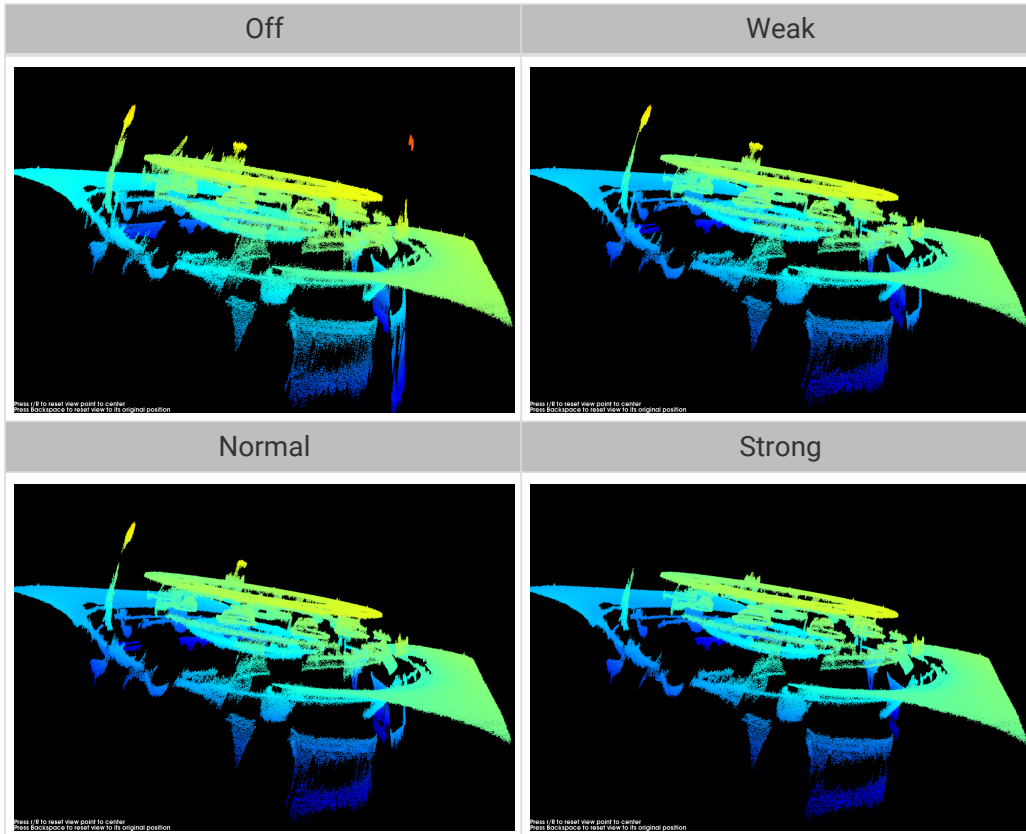
Point clouds obtained with different **Outlier Removal** values (all other conditions identical):



Noise Removal

Description	This parameter removes the noise in the point cloud, thus reducing the impact on the precision and accuracy of subsequent calculation. Noise is the scattered points close to the object surface.
Visibility	Beginner, Expert, Guru
Values	<ul style="list-style-type: none"> Off Weak Normal Strong
Instructions	<ul style="list-style-type: none"> Higher intensity removes more noise but may also remove some object features. Lower intensity removes less noise but also preserves the object features better. Higher intensity results in slower calculation, and lower intensity results in faster calculation.

Point clouds obtained with different **Noise Removal** values (all other conditions identical):



If this function removes the needed object features, please reduce the intensity. However, more noise will be retained.

Edge Preservation

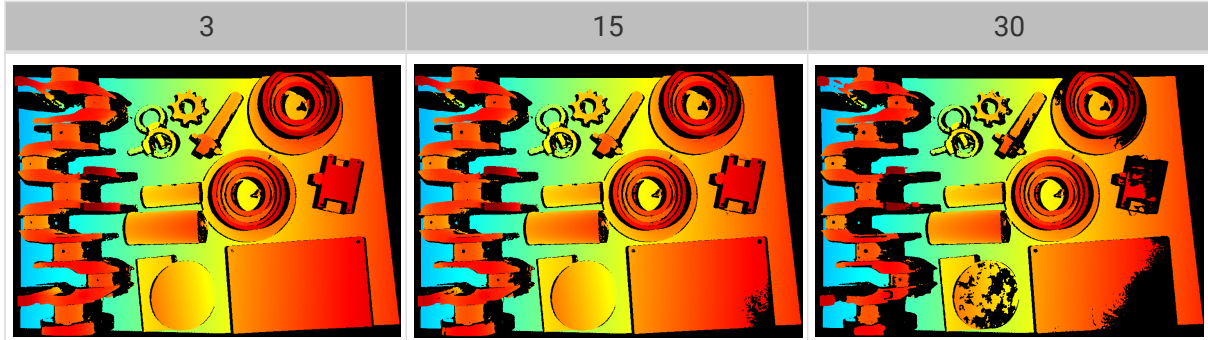
Description	This parameter preserves the sharpness of object edges during surface smoothing.
Visibility	Guru
Values	<ul style="list-style-type: none"> • Sharp: Preserves the sharpness of object edges as much as possible. However, the effect of surface smoothing will be reduced. • Normal: Balances between edge preservation and surface smoothing. • Smooth: Does not preserve the edges. The object surface will be well smoothed, but the object edges will be distorted.
Instructions	Adjust based on your needs for preserving the edges of workpieces.

Stripe Contrast Threshold

Description	This parameter is used to remove noise in the point cloud. Adjust this parameter if the level of noise in the point cloud is still high after adjusting Outlier Removal and Noise Removal .
Visibility	Beginner, Expert, Guru
Values	1 to 100

Instructions	<ul style="list-style-type: none"> • Greater value of this parameter results in more points being removed. Smaller value results in fewer points being removed. • Increase the value of this parameter can remove the noise in the point cloud, but may also cause the points of dark objects to be lost.
--------------	---

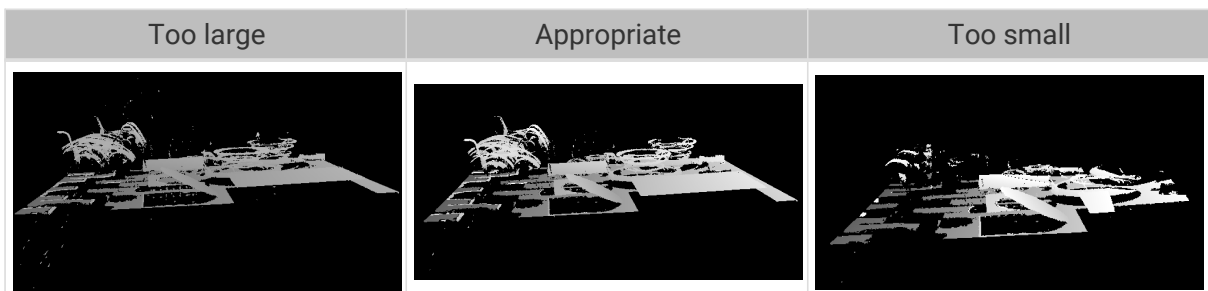
Point clouds obtained with different **Stripe Contrast Threshold** values (all other conditions identical):

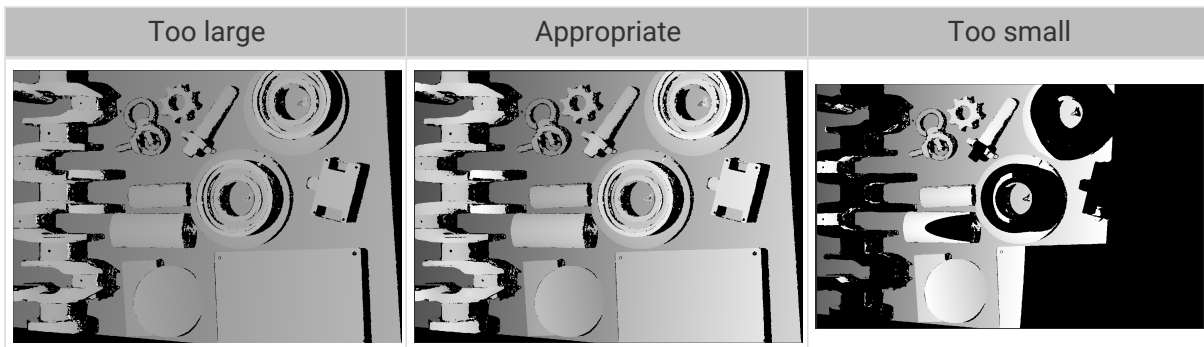


Depth Range

Description	Set a Z-direction ROI for the depth map and point cloud. Depth Range can be set within the camera working distance, and the data outside the range is removed from the depth map and point cloud.
Visibility	Beginner, Expert, Guru
Values	<ul style="list-style-type: none"> • Lower Limit: 1 to 4000 mm • Upper Limit: 1 to 5000 mm
Instructions	<ul style="list-style-type: none"> • Depth Range should be adjusted to an appropriate range to ensure the completeness of depth data in the depth map and point cloud. Too large a range may introduce interference, and too small a range may remove the data of key features. • For detailed instructions, please refer to Set Depth Range below.

The same point cloud with different **Depth Range** values (all other conditions identical):

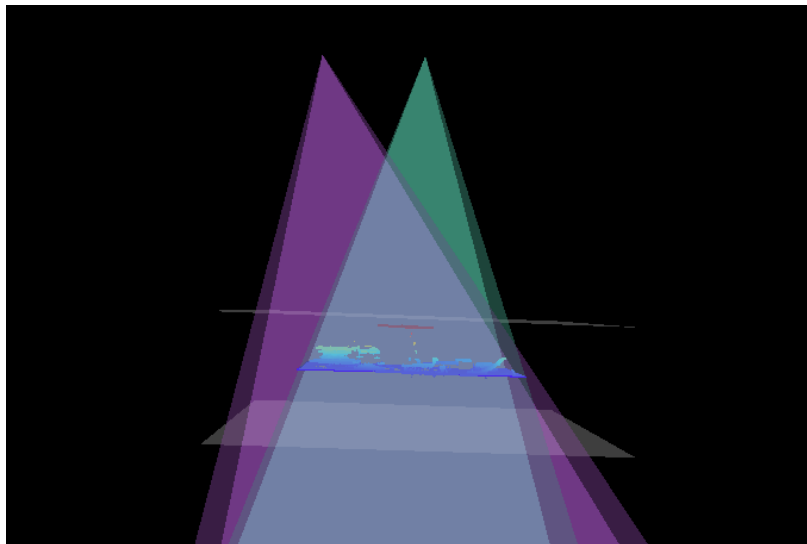




Set Depth Range

Follow these steps to adjust **Depth Range**:

1. Double-click [**Edit**] to the right of **Depth Range** to open the **Set Depth Range** window.
2. Click [**Refresh point cloud**] at the top of the right panel to obtain the newest point cloud.
3. [Adjust the position of the point cloud](#): adjust the position until you can see the two gray rectangles that represents the upper and lower limits of **Depth Range**.



4. Adjust the depth range: drag the handles on the slider to adjust **Depth Range** roughly. Then, enter specific values on the right to fine-tune **Depth Range**.



Judge if the depth range is appropriate: All needed object features are located between the two gray rectangles, and most noise and outliers are located outside.

5. Click [**Save**] to save the depth range settings.



- Click [**Recommended range**] to adjust **Depth Range** to the recommended working distance of the current camera.
- Click [**Reset**] to restore **Depth Range** to the last saved values.

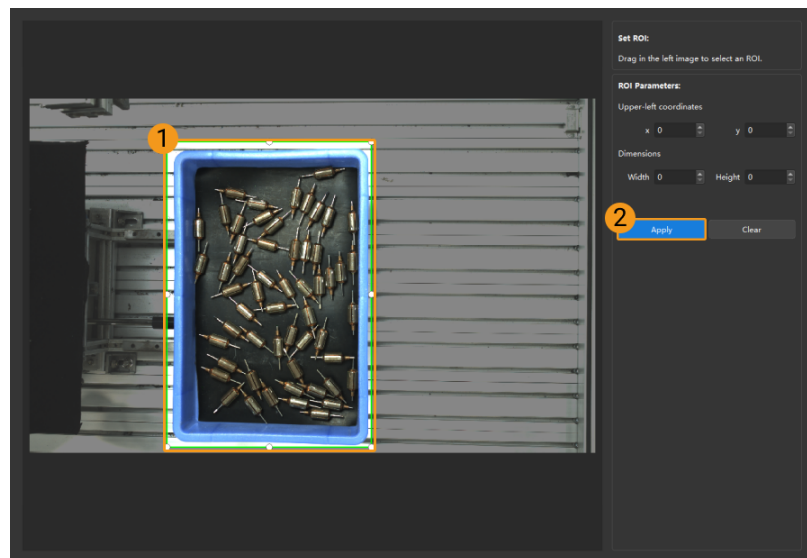
ROI

Description	Set the ROI on the XY plane for the depth map and point cloud. Points outside the selected region are removed.
-------------	--

Visibility	Beginner, Expert, Guru
Values	N/A
Instructions	For detailed instructions, refer to Set ROI below.

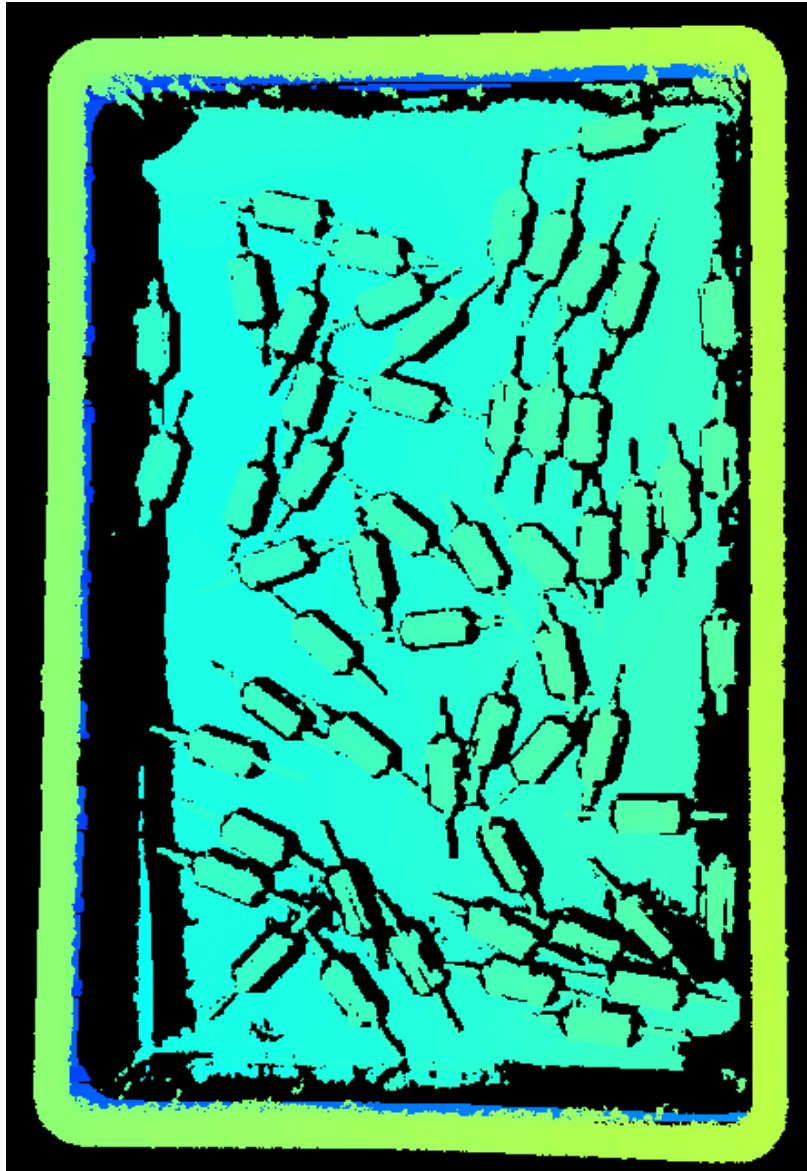
Set ROI

1. Double-click [**Edit**] to the right of **ROI** to open the **Set ROI** window.
2. Select and adjust the ROI on the left. Drag the selection box to move it. Drag the handles on the box to adjust the box size.
3. Click [**Apply**] to apply the set ROI.



- Click [**Clear**] to remove the current ROI.
- For the DEEP (V4) and LSR (V4) series, the 2D image (depth source) is displayed in this window. If the image is too dark or too bright, please adjust **2D Image (Depth Source) Exposure Mode**.

4. Acquire data again and view the depth map or point cloud to check the result of setting the ROI.



5.3.6. Parameters of V3 Cameras

This topic introduces the parameters of the V3 Cameras. The parameters are divided into parameters for the 2D image and parameters for the depth map / point cloud, based on the data types that they affect.

Parameters for the 2D Image

The 2D image should not be too bright or too dark, and the surface features of the target object should be clearly visible in the image.

The quality of the 2D image is affected by parameters in the **2D Parameters** category and the **Gain** parameter in the **3D Parameters** category.



If you are using a color camera, and the color of the image differ significantly from that of the objects, you can adjust the **white balance** setting. For detailed instructions, please refer to [Adjust White Balance](#).

2D Parameters

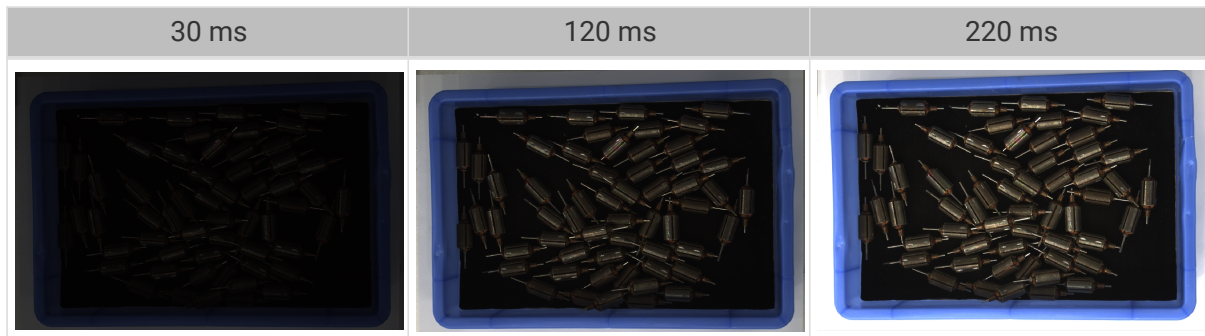
Exposure Mode

Description	Set the exposure mode for capturing the 2D image.
Visibility	Beginner, Expert, Guru
Values	<ul style="list-style-type: none"> • Timed: Set a single exposure time. Usually used in stable lighting conditions. • Auto: The exposure time is automatically adjusted. Usually used in varying lighting conditions. • HDR: Set multiple exposure times and merge the images. Usually used for objects with various colors or textures. • Flash: Use the projector for supplemental light. Usually used in dark environments.
Instructions	<p>After selecting different options, different parameters are displayed in the 2D Parameters category for adjustment:</p> <ul style="list-style-type: none"> • Timed: displays Exposure Time. • Auto: displays Gray Value and Auto-Exposure ROI. • HDR: displays Tone Mapping and Exposure Sequence. • Flash: no other parameters need to be adjusted. The projector projects supplemental light automatically during image capturing.

Timed: Exposure Time

Description	<p>This parameter affects the brightness of the image.</p> <ul style="list-style-type: none"> • Long exposure time increases the image brightness. • Short exposure time decreases the image brightness.
Visibility	Beginner, Expert, Guru
Values	0.1 to 999 ms
Instructions	<p>Adjust this parameter based on the quality of the 2D image. The 2D image should not be too bright or too dark, and the surface features of the target object should be clearly visible in the image.</p> <ul style="list-style-type: none"> • Long exposure time is usually used in dark environments. • Short exposure time is usually used in bright environments.

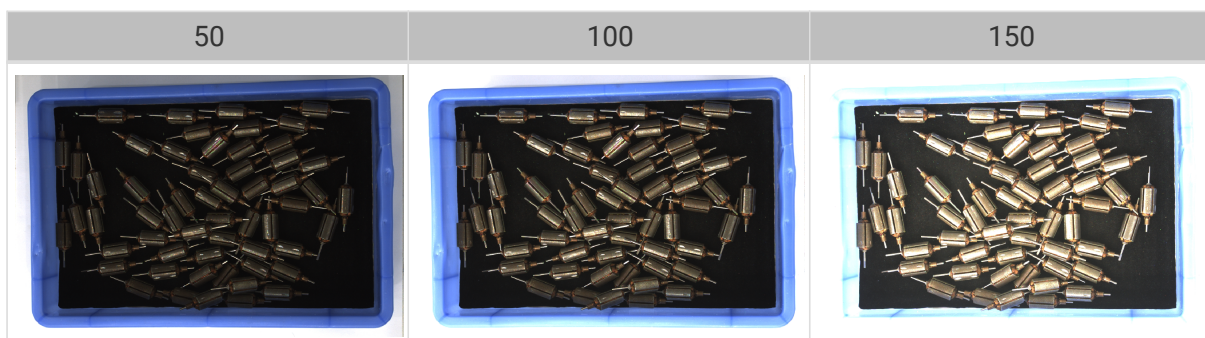
2D images obtained with different **Exposure Time** values (all other conditions identical):



Auto: Gray Value

Description	This parameter affects the brightness of the 2D image. Increase the value if the 2D image is too dark and decrease if too bright.
Visibility	Beginner, Expert, Guru
Values	0 to 255
Instruction	N/A

2D images obtained with different **Gray Value** values (all other conditions identical):

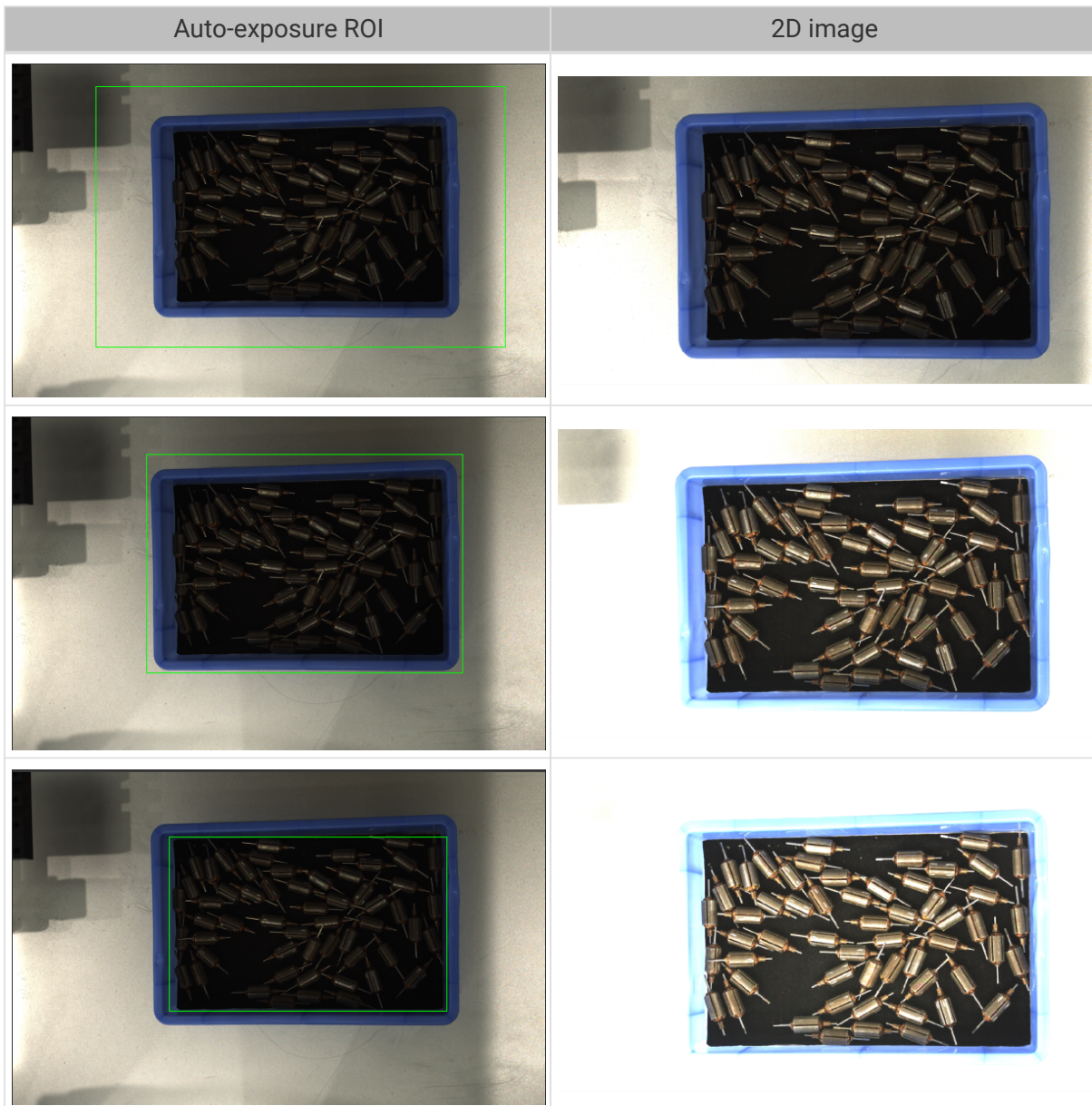


The **Gray Value** of a monochrome image is equivalent to the image brightness; the **Gray Value** of a color image is equivalent to the brightness of each color channel.

Auto: Auto-Exposure ROI

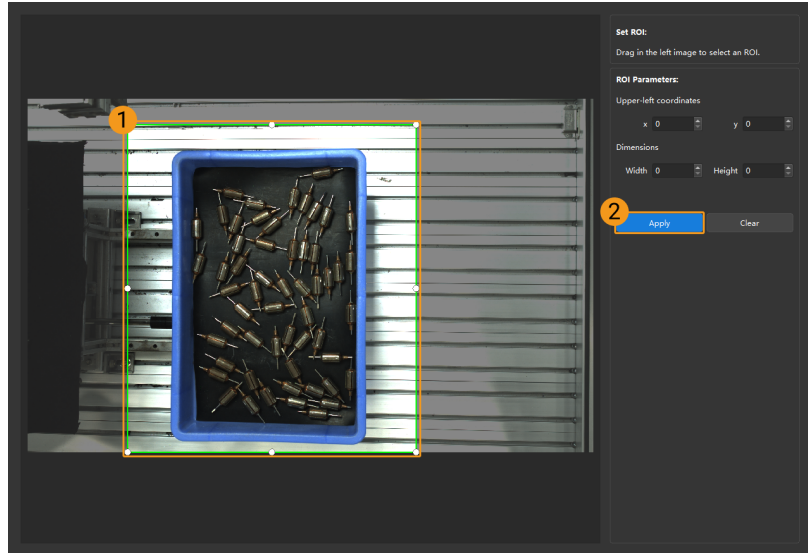
Description	<ul style="list-style-type: none"> The exposure time is automatically adjusted based on the lighting, object colors, etc., in the selected region. When an auto-exposure ROI is not set, the exposure time is automatically adjusted based on the entire camera FOV.
Visibility	Beginner, Expert, Guru
Values	N/A
Instruction	For detailed instructions, refer to Set Auto-Exposure ROI below.

2D images obtained with different **Auto-Exposure ROI** values (all other conditions identical):



Set Auto-Exposure ROI

1. Double-click [**Edit**] to the right of **Auto-Exposure ROI** to open the **Set ROI** window.
2. Select and adjust the ROI on the left. Drag the selection box to move it. Drag the handles on the box to adjust the box size.
3. Click [**Apply**] to apply the set auto-exposure ROI.



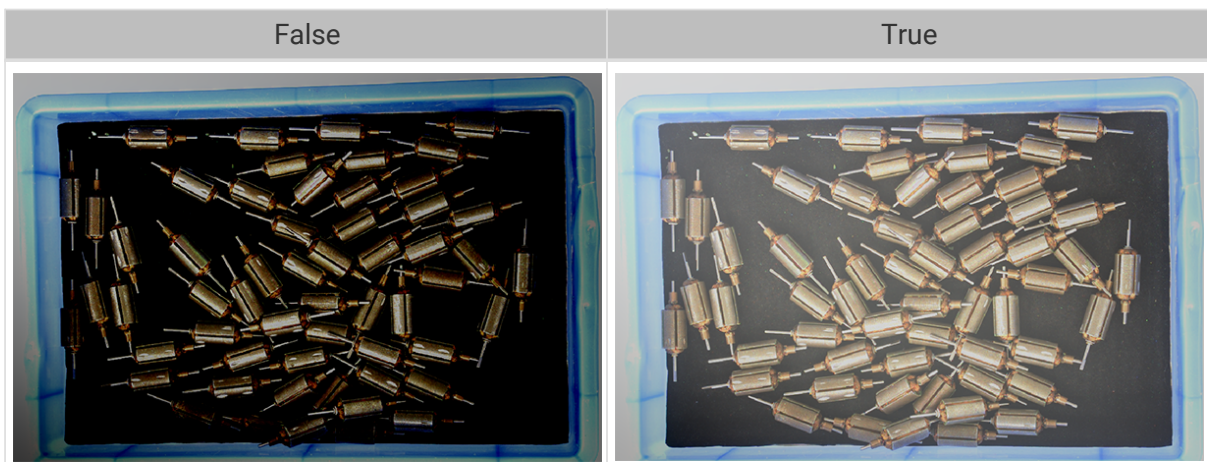
 Click [Clear] to remove the current auto-exposure ROI.

- Acquire data again and view the 2D image to check the result of automatic exposure adjustment.


HDR: Tone Mapping

Description	This function can make the image look more natural. If the 2D image appears very different from the actual objects, please enable this function.
Visibility	Beginner, Expert, Guru
Values	<ul style="list-style-type: none"> • False • True
Instructions	Check to enable Tone Mapping .

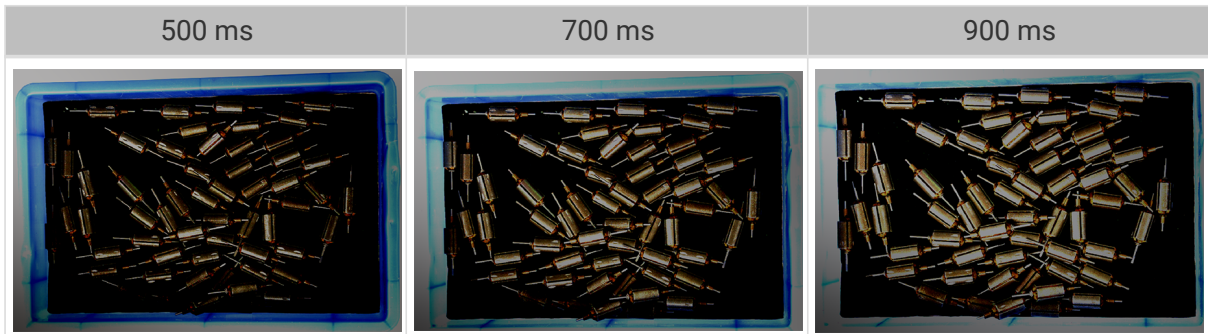
2D images obtained with **Tone Mapping** disabled or enabled (all other conditions identical):



HDR: Exposure Sequence

Description	Set multiple exposure times, and the captured images are merged to generate a 2D image that retains more details in the highlights and shadows.
Visibility	Beginner, Expert, Guru
Values	N/A
Instructions	<ol style="list-style-type: none"> 1. Double-click [Edit] to the right of Exposure Sequence to open the editing window of the exposure sequence. 2. Click [+] to add and set a new exposure time. 3. To delete an exposure time, select it and click [-]. 4. After you finish adding the exposure times, click [Apply] to apply the exposure sequence. <div style="display: flex; align-items: center; margin-top: 10px;"> <div style="margin-right: 10px;">  </div> <ul style="list-style-type: none"> ◦ [Cancel]: do not save the current settings and exit this window. ◦ [Reset]: clear all the exposure times. </div>

2D images captured with single exposure times (all other conditions identical):




2D images captured with different exposure sequences consisting of the above exposure times (all other conditions identical):



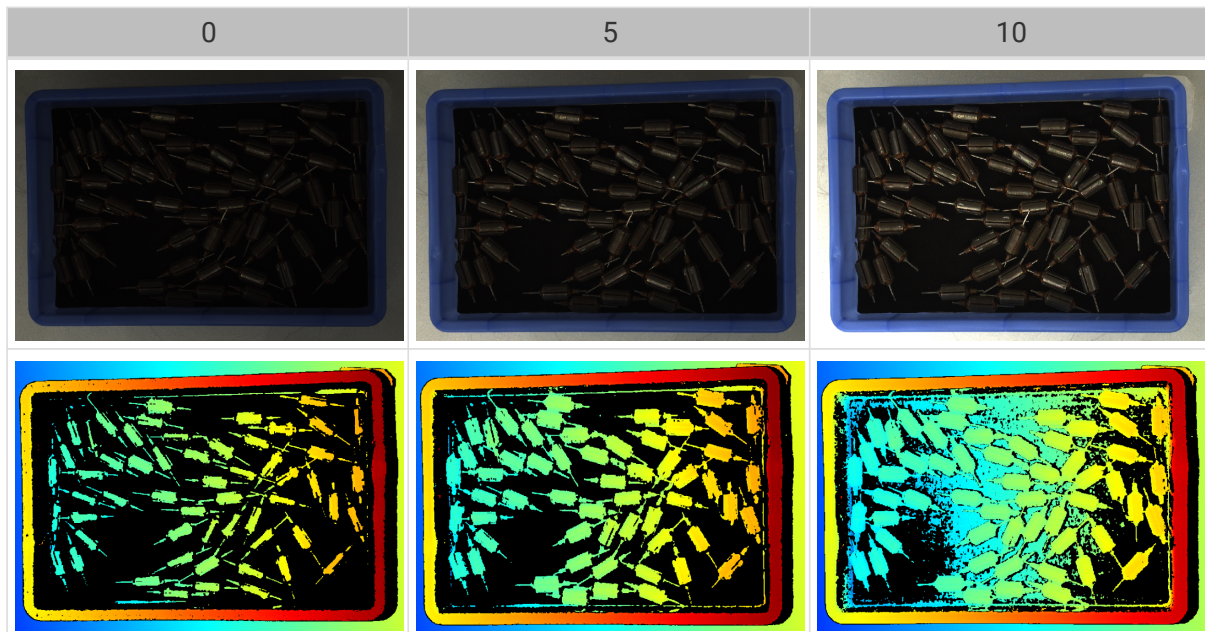
3D Parameters

Gain

Description	<p>This parameter affects the brightness of the captured image. At the same time, more noise is introduced.</p> <div style="display: flex; align-items: center; margin-top: 10px;"> <div style="margin-right: 10px;">  </div> <p>This parameter affects the quality of the depth map and point cloud by changing the brightness of the images used for calculating depth data.</p> </div>
Visibility	Expert, Guru

Values	0 to 16 dB
Instructions	When the expected brightness cannot be reached by adjusting Exposure Time , adjust this parameter.

2D images and depth maps obtained with different **Gain** values (all other conditions identical):



Parameters for the Depth Map / Point Cloud

In the depth map and point cloud, the needed data should be complete.



Determine which part of the data is needed according to your actual needs. For example, if you need to pick metal bowls with opening facing up by the rim, usually you only need to make sure that the data of the bowl rims is complete.

Parameters in the following categories affect the quality of the depth map and point cloud.

Parameter category	Depth map	Point cloud
3D Parameters	✓	✓
Point Cloud Processing		✓
Depth Range	✓	✓
ROI	✓	✓

3D Parameters

The parameters in this category affect the images used for calculating depth data, thus affecting the quality of the depth map and point cloud.

Use **Exposure Assistant** can obtain the recommended exposure parameter values. Double-click [**Auto Set**] to the right of **3D Parameters** to open **Exposure Assistant**.

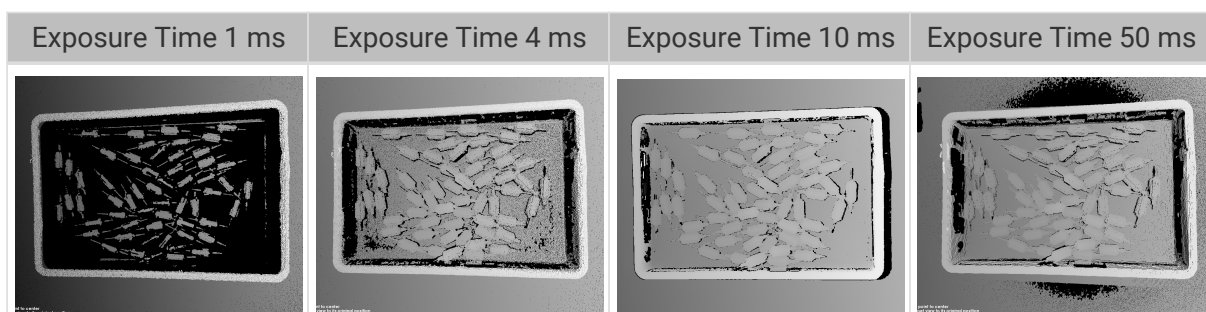
Exposure Multiplier

Description	Set the number of Exposure Time .
Visibility	Beginner, Expert, Guru
Values	1 to 3
Instruction	<ul style="list-style-type: none"> When the value of Exposure Multiplier is greater than 1, multiple Exposure Time parameters must be set. All images captured with different exposure times are used for calculating depth. Using multiple exposure times can improve the completeness of depth data but also increases processing time. With multiple exposure times, the amount of time for obtaining the depth map and point cloud increases. Please try to minimize the number of exposure times while ensuring the data quality.

Exposure Time


Description	Set the exposure time for acquiring depth information. The number of Exposure Time to be set depends on the value of Exposure Multiplier .
Visibility	Beginner, Expert, Guru
Values	<ul style="list-style-type: none"> Value range: 0.1 ms to 99 ms
Instructions	<ul style="list-style-type: none"> Usually, long exposure time is used for dark objects, and short exposure time is used for light objects. Exposure times too long or too short both lead to loss of depth information. The exposure time of a laser camera must be a multiple of 4. If the entered value is not a multiple of 4, it will be automatically adjusted. The minimum value that can be set is 4 ms for the Laser series and 8 ms for other laser cameras.

Point clouds obtained under different **Exposure Time** values (all other conditions identical):



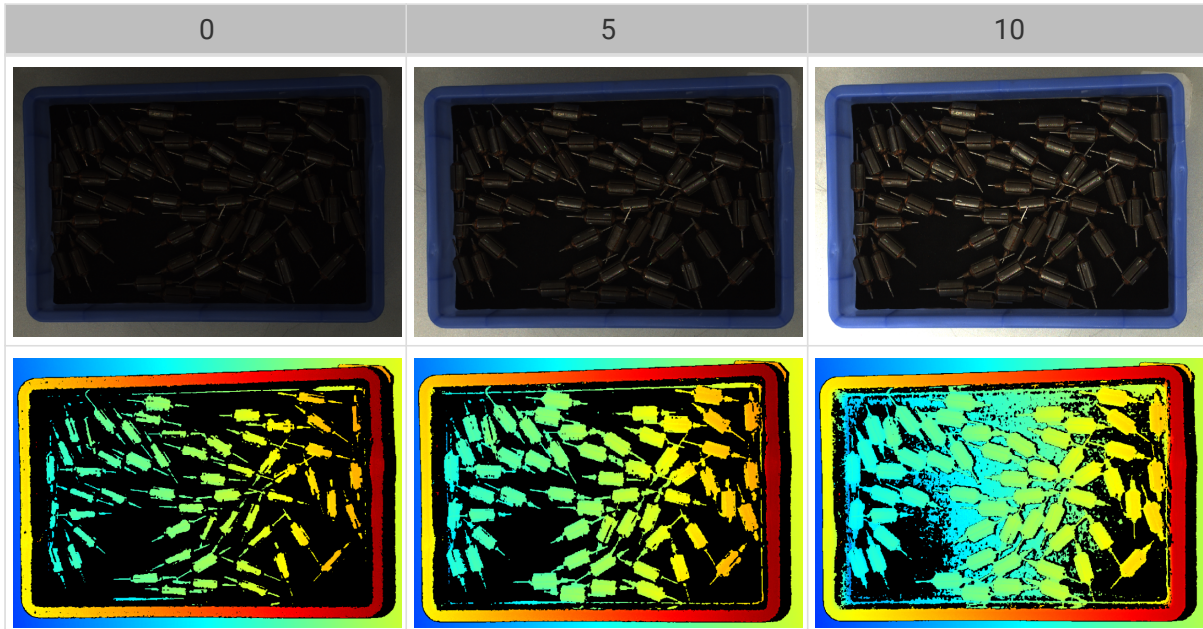
The black regions lack depth information to generate corresponding point cloud.

Gain

Description	<p>This parameter affects the brightness of the captured image. At the same time, more noise is introduced.</p> <p> This parameter affects the quality of the depth map and point cloud by changing the brightness of the images used for calculating depth data.</p>
-------------	--

Visibility	Expert, Guru
Values	0 to 16 dB
Instructions	When the expected brightness cannot be reached by adjusting Exposure Time , adjust this parameter.

2D images and depth maps obtained with different **Gain** values (all other conditions identical):



Projector

Available for: Nano (V3), Pro (V3) and Log (V3) series.

Light Brightness

Description	This parameter affects the brightness of the projected structured light.
Visibility	Expert, Guru
Values	<ul style="list-style-type: none"> • High: high brightness, suitable for dark objects. • Normal: medium brightness, suitable for regular objects. • Low: low brightness, suitable for reflective objects.
Instructions	Adjust this parameter based on the object type.

Fringe Coding Mode

Available for: Nano (V3) and Pro XS (V3).

Description	Select the pattern of the structured light to be projected.
Visibility	Expert, Guru

Values	<ul style="list-style-type: none"> • Fast: suitable for opaque objects, provides fast acquisition speed, but the depth data quality is lower. • Accurate: suitable for opaque objects, provides high-quality depth data, but the acquisition speed is slower.
Instructions	Adjust this parameter according to the type of target objects and your actual needs for data quality and data acquisition speed.

When the target object is not translucent, the point clouds obtained with **Fringe Coding Mode** set to **Accurate** and **Fast** are as follows (all other conditions identical):

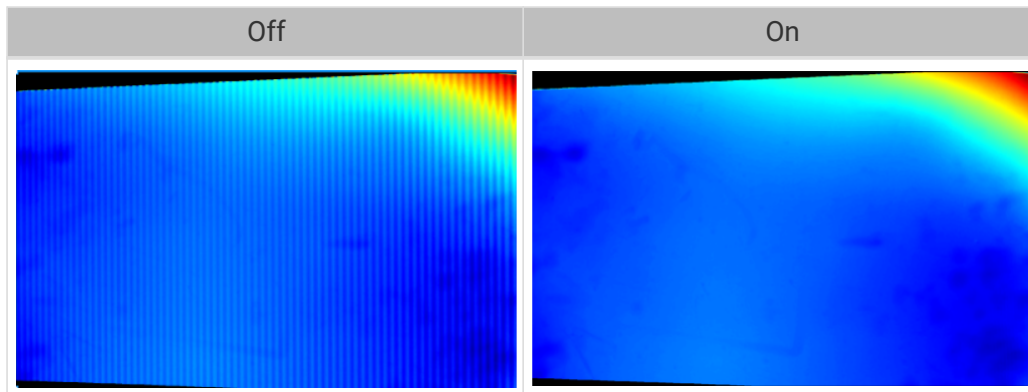


Anti-Flicker Mode

Available for: Nano (V3).

Description	Flicker refers to the rapid and periodical change in the intensity of artificial light. This phenomenon can cause fluctuations in the depth data. Such fluctuations can be reduced by adjusting the projection frequency of the structured light.
Visibility	Expert, Guru
Values	<ul style="list-style-type: none"> • Off • AC50Hz • AC60Hz
Instructions	Please select the option that corresponds to the frequency of the power supply in your location. The AC frequency in most countries is 50Hz. The AC frequency in the U.S. and some Asian countries is 60Hz.

Depth maps obtained with **Anti-Flicker Mode** off or on (all other conditions identical):



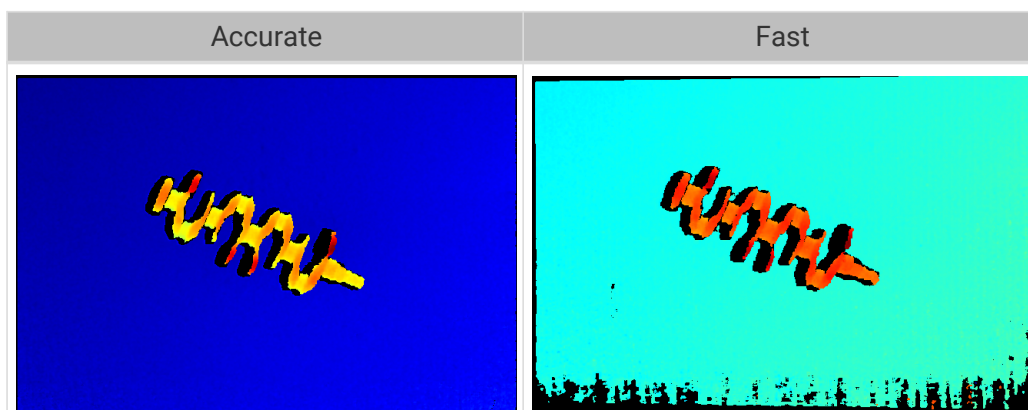
Laser

Available for: Laser (V3) series.

Fringe Coding Mode

Parameter Description	Change the structured light patterns used, which affects the data acquisition speed and depth data quality.
Visibility	Expert, Guru
Values	<ul style="list-style-type: none"> • Fast: faster data acquisition, but the quality of the depth map and point cloud is lower. • Accurate: slower data acquisition, but the quality of the depth map and point cloud is higher.
Instructions	Adjust this parameter according to your actual needs for data quality and data acquisition speed.

Point clouds obtained with different **Fringe Coding Mode** values (all other conditions identical):



Laser Power

Description	Set the power of the laser projector, which affects the brightness of the projected structured light.
Visibility	Expert, Guru
Values	50 to 100%

Instruction	<p>Usually, using the default value is sufficient.</p> <ul style="list-style-type: none"> • Greater value results in higher structured light brightness; smaller value results in lower structured light brightness. • Greater values are suitable for dark objects, and smaller values are suitable for reflective objects.
--------------------	--

Point Cloud Processing

Adjusting the parameters in the **Point Cloud Processing** category can improve the quality of the point cloud.

Principles for Adjusting Point Cloud Processing Parameters

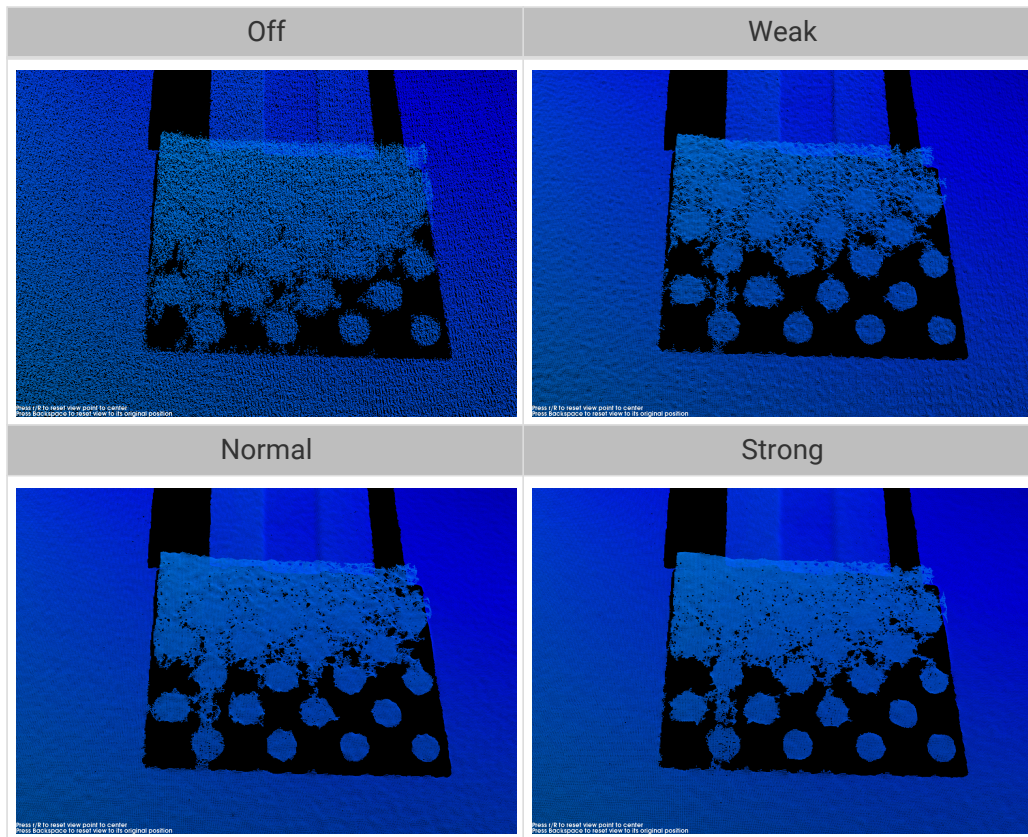
When adjusting the **Point Cloud Processing** parameters, follow these principles to reduce the camera acquisition time and cycle time:

1. Adjust **Outlier Removal** first. The calculation time for different intensities of this parameter is about the same. Using a higher intensity will not add to the calculation time too much.
2. It is recommended to use low intensities of **Surface Smoothing** and **Noise Removal**. The higher the intensities of these two parameters, the longer the calculation time.

Surface Smoothing

Description	This parameter reduces the depth fluctuation in the point cloud and improve its resemblance to the actual object surface.
Visibility	Beginner, Expert, Guru
Values	<ul style="list-style-type: none"> • Off • Weak • Normal • Strong
Instructions	<ul style="list-style-type: none"> • Surface Smoothing causes loss of object surface details. Higher intensity causes more object details to be lost. Lower intensity causes fewer object details to be lost. • Higher intensity results in slower calculation, and lower intensity results in faster calculation.

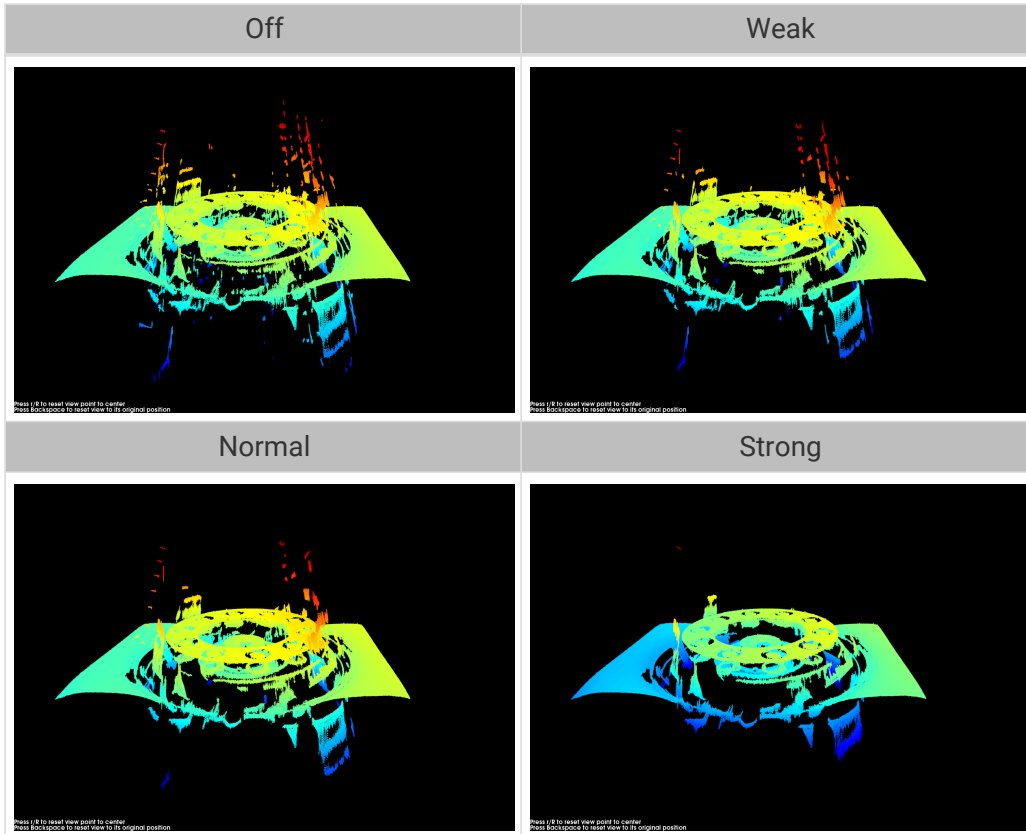
Point clouds obtained with different **Surface Smoothing** values (all other conditions identical):



Outlier Removal

Description	This parameter removes the outliers in the point cloud. Outliers are clustered points away from the object point cloud.
Visibility	Beginner, Expert, Guru
Values	<ul style="list-style-type: none"> • Off • Weak • Normal • Strong
Instructions	<ul style="list-style-type: none"> • Higher intensity removes more outliers, and lower intensity removes fewer outliers. • If the object point cloud contains multiple parts, high intensities of outlier removal might remove some parts. For example, if the object is a cup, outlier removal might remove the point cloud of the handle part.

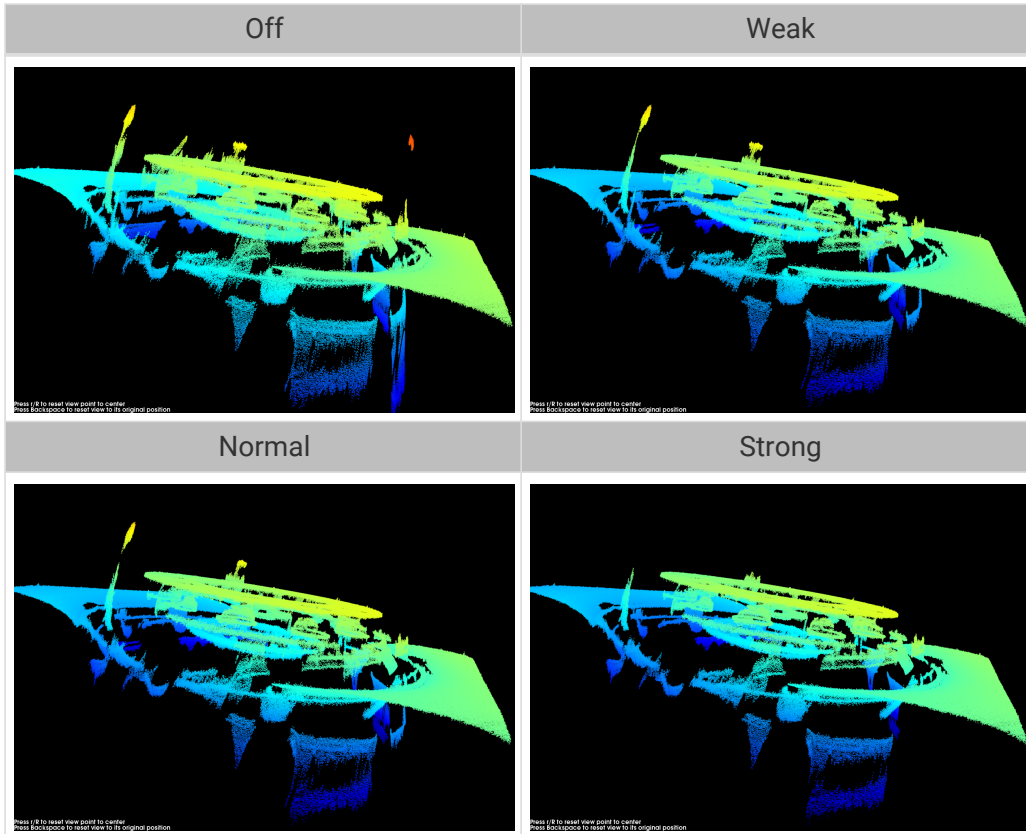
Point clouds obtained with different **Outlier Removal** values (all other conditions identical):



Noise Removal

Description	This parameter removes the noise in the point cloud, thus reducing the impact on the precision and accuracy of subsequent calculation. Noise is the scattered points close to the object surface.
Visibility	Beginner, Expert, Guru
Values	<ul style="list-style-type: none"> • Off • Weak • Normal • Strong
Instructions	<ul style="list-style-type: none"> • Higher intensity removes more noise but may also remove some object features. Lower intensity removes less noise but also preserves the object features better. • Higher intensity results in slower calculation, and lower intensity results in faster calculation.

Point clouds obtained with different **Noise Removal** values (all other conditions identical):



If this function removes the needed object features, please reduce the intensity. However, more noise will be retained.

Edge Preservation

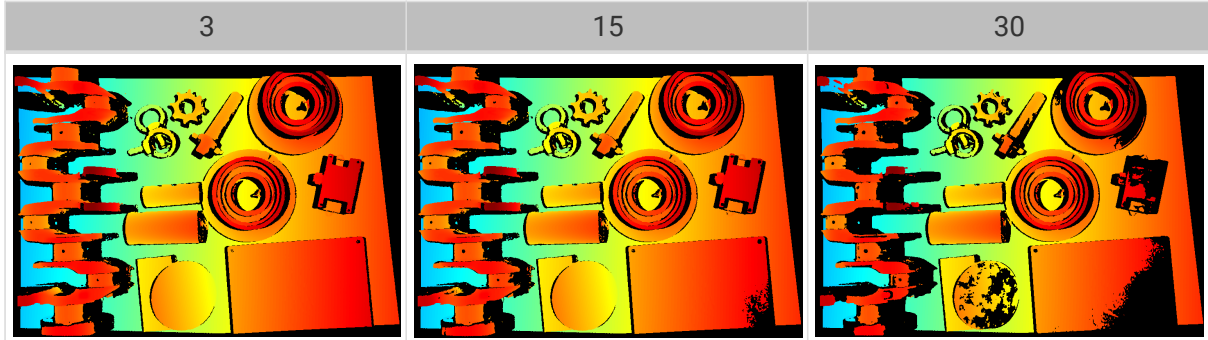
Description	This parameter preserves the sharpness of object edges during surface smoothing.
Visibility	Guru
Values	<ul style="list-style-type: none"> • Sharp: Preserves the sharpness of object edges as much as possible. However, the effect of surface smoothing will be reduced. • Normal: Balances between edge preservation and surface smoothing. • Smooth: Does not preserve the edges. The object surface will be well smoothed, but the object edges will be distorted.
Instructions	Adjust based on your needs for preserving the edges of workpieces.

Stripe Contrast Threshold

Description	This parameter is used to remove noise in the point cloud. Adjust this parameter if the level of noise in the point cloud is still high after adjusting Outlier Removal and Noise Removal .
Visibility	Beginner, Expert, Guru
Values	1 to 100

Instructions	<ul style="list-style-type: none"> • Greater value of this parameter results in more points being removed. Smaller value results in fewer points being removed. • Increase the value of this parameter can remove the noise in the point cloud, but may also cause the points of dark objects to be lost.
--------------	---

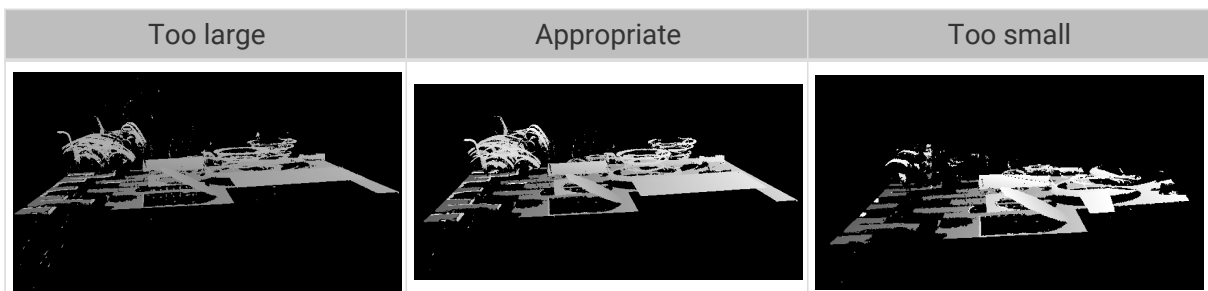
Point clouds obtained with different **Stripe Contrast Threshold** values (all other conditions identical):

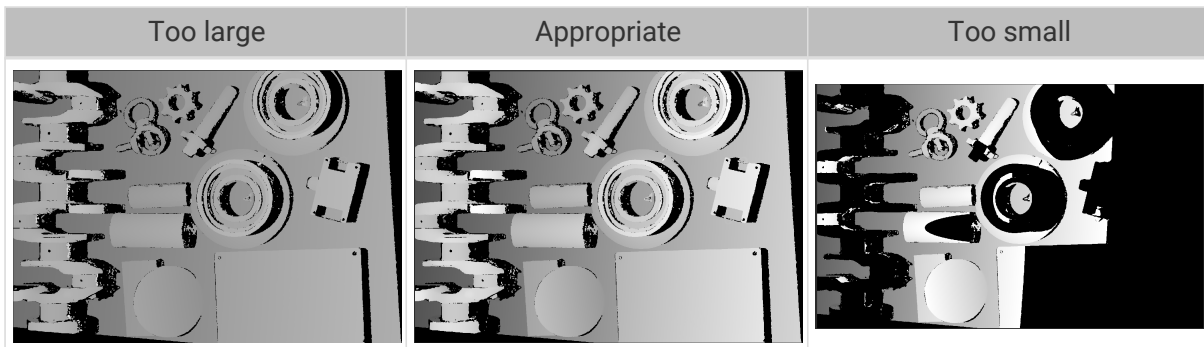


Depth Range

Description	Set a Z-direction ROI for the depth map and point cloud. Depth Range can be set within the camera working distance, and the data outside the range is removed from the depth map and point cloud.
Visibility	Beginner, Expert, Guru
Values	<ul style="list-style-type: none"> • Lower Limit: 1 to 4000 mm • Upper Limit: 1 to 5000 mm
Instructions	<ul style="list-style-type: none"> • Depth Range should be adjusted to an appropriate range to ensure the completeness of depth data in the depth map and point cloud. Too large a range may introduce interference, and too small a range may remove the data of key features. • For detailed instructions, please refer to Set Depth Range below.

The same point cloud with different **Depth Range** values (all other conditions identical):

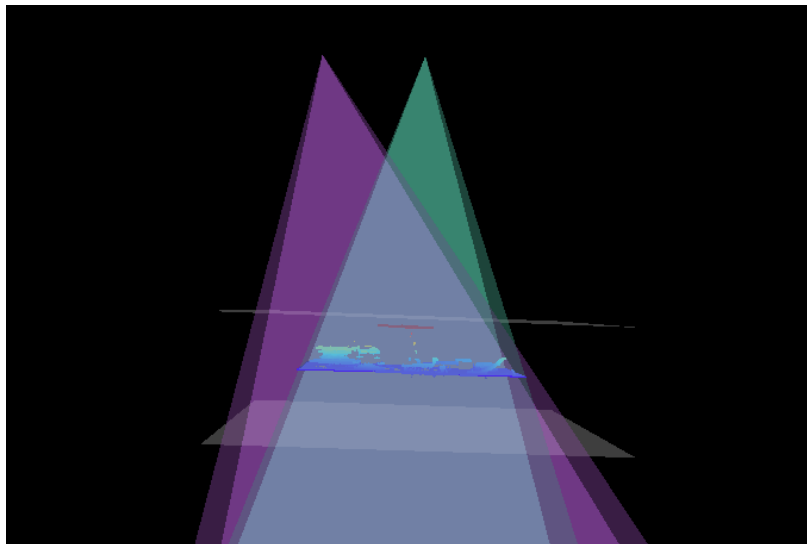




Set Depth Range

Follow these steps to adjust **Depth Range**:

1. Double-click [**Edit**] to the right of **Depth Range** to open the **Set Depth Range** window.
2. Click [**Refresh point cloud**] at the top of the right panel to obtain the newest point cloud.
3. [Adjust the position of the point cloud](#): adjust the position until you can see the two gray rectangles that represents the upper and lower limits of **Depth Range**.



4. Adjust the depth range: drag the handles on the slider to adjust **Depth Range** roughly. Then, enter specific values on the right to fine-tune **Depth Range**.



Judge if the depth range is appropriate: All needed object features are located between the two gray rectangles, and most noise and outliers are located outside.

5. Click [**Save**] to save the depth range settings.



- Click [**Recommended range**] to adjust **Depth Range** to the recommended working distance of the current camera.
- Click [**Reset**] to restore **Depth Range** to the last saved values.

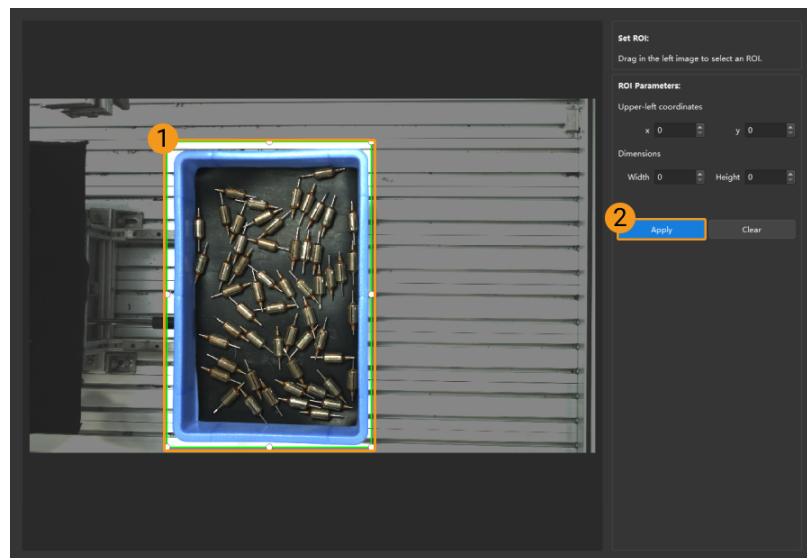
ROI

Description	Set the ROI on the XY plane for the depth map and point cloud. Points outside the selected region are removed.
-------------	--

Visibility	Beginner, Expert, Guru
Values	N/A
Instructions	For detailed instructions, refer to Set ROI below.

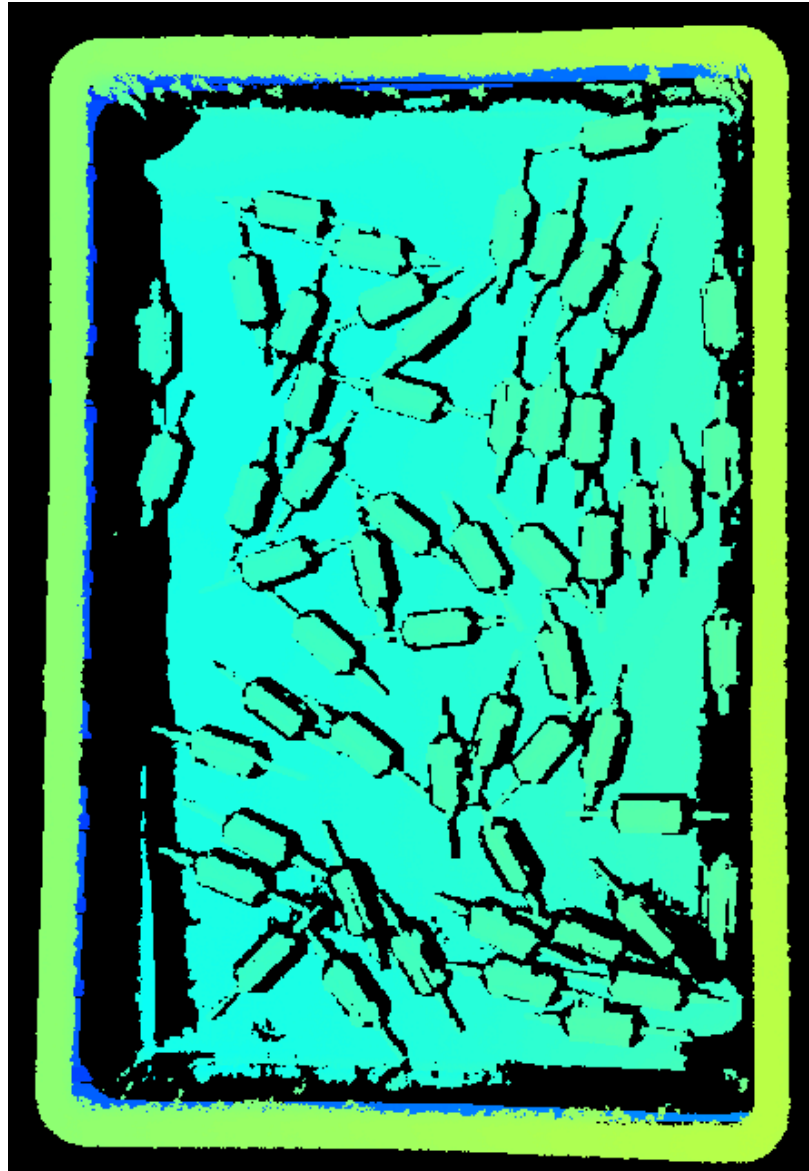
Set ROI

1. Double-click [**Edit**] to the right of **ROI** to open the **Set ROI** window.
2. Select and adjust the ROI on the left. Drag the selection box to move it. Drag the handles on the box to adjust the box size.
3. Click [**Apply**] to apply the set ROI.



- Click [**Clear**] to remove the current ROI.
- For the DEEP (V4) and LSR (V4) series, the 2D image (depth source) is displayed in this window. If the image is too dark or too bright, please adjust **2D Image (Depth Source) Exposure Mode**.

4. Acquire data again and view the depth map or point cloud to check the result of setting the ROI.



5.4. Tools

Mech-Eye Viewer provides a series of tools that facilitates you in using the camera. This topic introduces the functions and usages of the tools and provides detailed instructions.

Tool	Function
Intrinsic Parameter Tool	Check if the camera intrinsic parameters are accurate, correct the intrinsic parameters, and reset to factory intrinsic parameters.
3D Exposure Assistant	Obtain the recommended combination of exposure parameter values in 3D Parameters .
Depth Analyzer	Check the quality of the depth data.
Check and Configure 2D Cameras	Check and configure the 2D cameras.
FOV Calculator	Calculate camera FOV based on the working distance.

Tool	Function
Custom Reference Frame	Define a custom reference frame used for displaying the depth map and point cloud.
Camera Controller	Check the project firmware version, time, and temperature of the camera and change the image resolution.
Full Pallet Simulator	Check if the top layer of a full load on a pallet is entirely located within the camera FOV.

Check the **Toolbar** option in the **View** menu to display the toolbar that provide quick access to the tools.

Click [**More Tools**] in the toolbar to select which tools to be displayed. Click [**OK**] to save the changes.

5.4.1. Intrinsic Parameter Tool

This tool is used to check if the camera intrinsic parameters are accurate, to correct the intrinsic parameters, and to reset to factory intrinsic parameters. The coordinates of points are accurate only if the intrinsic parameters are accurate.

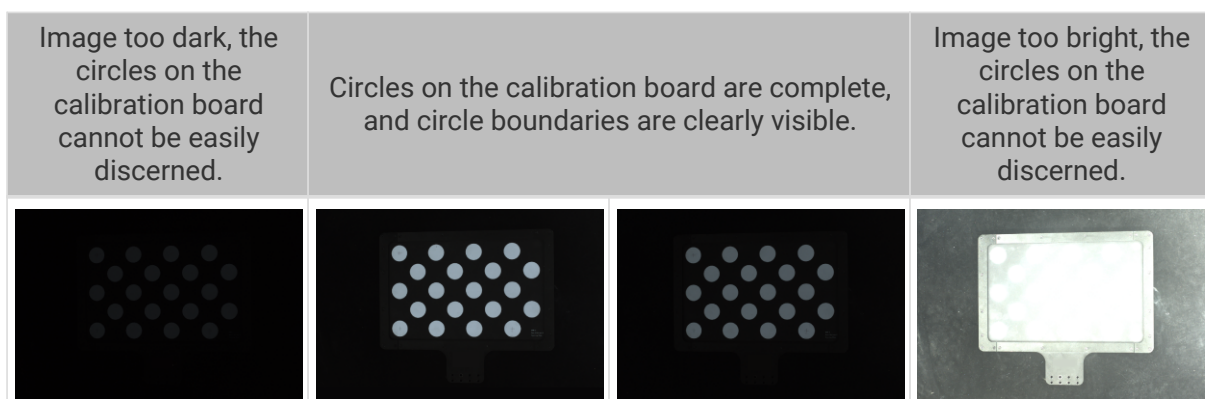
Preparation

Before checking the intrinsic parameters, please prepare the calibration board that came with the camera.

Place the calibration board within the camera FOV and obtain the 2D image and depth map of the calibration board. Make sure the obtained 2D image and depth map satisfy the following requirements:

- The region containing the circles on the calibration board is completely captured.
- The 2D image is not too bright or too dark, and the circles on the calibration board are clear and complete.
- The circles on the calibration board in the depth map are complete.

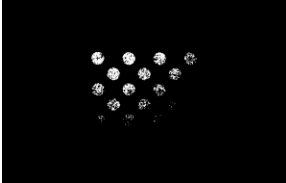
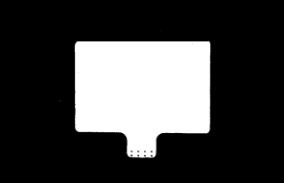
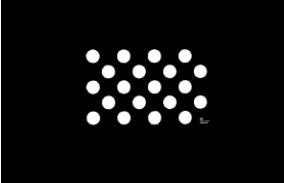
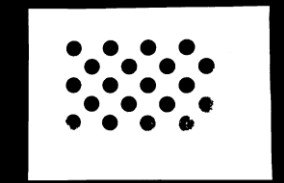
Examples of 2D images of the calibration board:



If the obtained 2D image does not satisfy the requirements, please adjust the [parameters for the](#)

2D image.

Examples of depth maps of the calibration board:

Exposure time too short, the circles on the calibration board are incomplete.	Circles on the calibration board / the entire calibration board are complete.		Exposure time too long, some or all the circles on the calibration board are lost.
			

If the obtained depth map does not satisfy the requirements, please adjust the [parameters for the depth map](#).

Check Intrinsic Parameters

Click the **Tool** menu in the menu bar and select **Intrinsic Parameter Tool** to open the **Intrinsic Parameter Tool** window.

To use **Intrinsic Parameter Tool** to check the camera intrinsic parameters, please follow these steps.

1. In **1. Place calibration board and check image quality**, click [**Acquire images**] to obtain the 2D image and depth map of the calibration board. Check if the images satisfy the requirements.
2. In **2. Select model of placed calibration board**, set the specifications of the calibration board:
 - If using a standard calibration board, select the model from the drop-down menu.
 - If using a customized calibration board, select **Customized** first, and then set the model, rows, columns and spacing according to the information provided by the engineer.
3. In **3. Check intrinsic parameters**, click [**Check intrinsic parameters**] to start checking the camera intrinsic parameters. When the check is completed, a window of the check results will pop up.



If an error message is displayed, please troubleshoot the issue according to the next section.

Troubleshoot Common Issues

The following issues may be encountered while checking the intrinsic parameters.

Circles on the Calibration Board not Detected

Symptom:

The intrinsic parameter check result window displays this message: **Circles on the calibration board not detected**.

Solution:

Please try the following solutions:

1. Check if the calibration board specifications are correct.
 - If incorrect, enter the correct specifications. Then, check the intrinsic parameters again.
 - If correct, proceed to the next step.
2. Check if the 2D image and depth map meet the image requirements. Please refer to the [image requirements](#).
 - If the 2D image does not satisfy the requirements, please adjust the [parameters for the 2D image](#). If the depth map does not satisfy the requirements, please adjust the [parameters for the depth map](#). Then, check the intrinsic parameters again.
 - If the images satisfy the requirements, proceed to the next step.
3. Use the **Draw aid circle** to facilitate the detection. For detailed instructions, please refer to [Draw Aid Circle](#).
4. In **3. Check Intrinsic Parameters**, click [**Check intrinsic parameters**] to check the intrinsic parameters again.
 - If the check succeeded, the troubleshooting has been completed.
 - If the intrinsic parameter check result window still displays **Circles on the calibration board not detected**, proceed to the next step.
5. Use the **Edit Feature Detection Parameters** function to facilitate the detection. For detailed instructions, please refer to [Edit Feature Detection Parameters](#).
6. In **3. Check Intrinsic Parameters**, click [**Check intrinsic parameters**] to check the intrinsic parameters again.
 - If the check succeeded, the troubleshooting has been completed.
 - If the intrinsic parameter check result window still displays **Circles on the calibration board not detected**, please contact Technical Support.

Valid Feature Points Not Found in Depth Map

Symptom:

The intrinsic parameter check result window displays this message: **Valid feature points not found in depth map**.

Solution:

Check if the depth map satisfies the requirements. Please refer to the [image requirements](#).

- If the depth map does not satisfy the requirements, please adjust the [parameters for the depth map](#). Then, check the intrinsic parameters again.
- If the depth map satisfies the requirements, please contact Technical Support.

Supplementary Functions

Intrinsic Parameter Tool provides the following functions that assist you in resolving the common issues.

Correct Intrinsic Parameters

This function can assist you in resolving the issue of large errors in the intrinsic parameters.

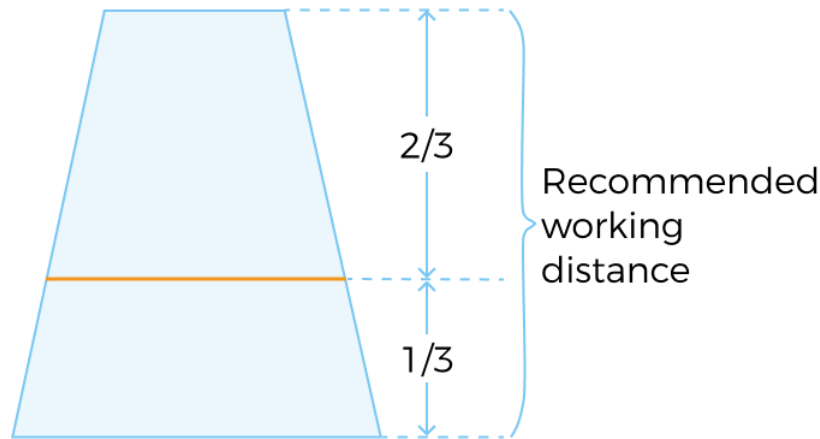


The UHP series cannot use the **Correct Intrinsic Parameters** function. If the errors in the

intrinsic parameters are large, please contact Technical Support.

When using the **Correct Intrinsic Parameters** function, place the calibration board according to the following requirements:

- Place the calibration board at the height of the orange line in the figure below:



- Acquire at least 5 sets of data where the calibration board is located at the center and the four corners of the FOV.



Follow these steps to use the **Correct Intrinsic Parameters** function:

- Place the calibration board according to the above requirements.
- Check the **Correct Intrinsic Parameters** option, and then click [**Acquire data**].
- Move the calibration board, and click [**Acquire data**] again.
- After adding the data, click [**Correct intrinsic parameters**]. When the correction is completed, a window of the correction results will pop up.

Draw Aid Circle

This function can assist you in resolving the issue of [circles on the calibration board not detected](#).

Follow these steps to use the **Draw aid circle** function:

- In **3. Check Intrinsic Parameters**, click [**Draw aid circle**].
- Find a complete circle on the calibration board in the 2D image. Click while holding the **Ctrl** key and then move diagonally up or down to adjust the size of the aid circle. Click again to finish drawing the aid circle.



- Only one aid circle can be drawn. The aid circle should be as close possible to the boundary of the circle on the calibration board.
- If you need to re-draw the aid circle, click [**Draw aid circle**] again.

- To zoom the 2D image, move the cursor onto the 2D image and scroll the mouse wheel.

Edit Feature Detection Parameters (Advanced)


This function can assist you in resolving the issue of [circles on the calibration board not detected](#).

Follow these steps to use the **Edit Feature Detection Parameters** function:

1. In **3. Check Intrinsic Parameters**, click **Edit Feature Detection Parameters (Advanced)**.
2. Adjust the displayed feature detection parameters according to the actual situation.
3. If not all the circles on the calibration board can be captured, check **Allow absence of some circles**, and adjust the newly displayed feature detection parameters.
4. In **3. Check Intrinsic Parameters**, click [**Check intrinsic parameters**] to check the intrinsic parameters again.

5.4.2. 3D Exposure Assistant

This tool is used for obtaining the recommended values for the **Exposure Time** parameter in the **3D Parameters** category.


 Currently, only one exposure time can be calculated.

Follow these steps to use **3D Exposure Assistant**:

1. Check the values of the **Depth Range** and **ROI** parameters in the **Parameters** tab: Make sure that the target object is in the set depth range and ROI.
2. Double-click [**Auto Set**] to the right of **3D Parameters** in the **Parameters** tab, or click **3D Exposure Assistant** in the **Tools** menu in the menu bar to open **3D Exposure Assistant**.
3. Adjust the ROI. **3D Exposure Assistant** will calculate the exposure time automatically based on this ROI. Please select the area where the target objects are located and avoid including irrelevant objects as much as possible. Then, click [**Next Step**] to apply the ROI.

 Drag the selection box to move it. Drag the handles on the box to adjust its size.

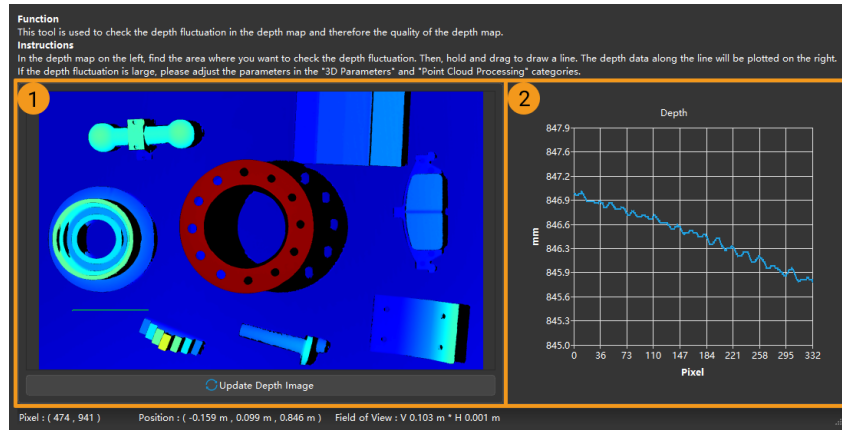
4. Click [**Calculate exposure time**] to calculate the time of a single exposure. The result will be displayed in the **Calculation result** area.
5. Check the quality of the depth map and point cloud in the image area on the left:
 - If the quality satisfies your requirements, click [**Apply**] to apply the calculation result to the **3D Parameters** category.

 If the **Exposure Multiplier** parameter is set to a value greater than 1, its value will be changed to 1 after the calculation result is applied.

- If the quality does not satisfy your requirements, click [**Back**] and repeat steps 2 to 4.

5.4.3. Depth Analyzer

This tool is used to check the depth fluctuation in the depth map and therefore the quality of the depth map.

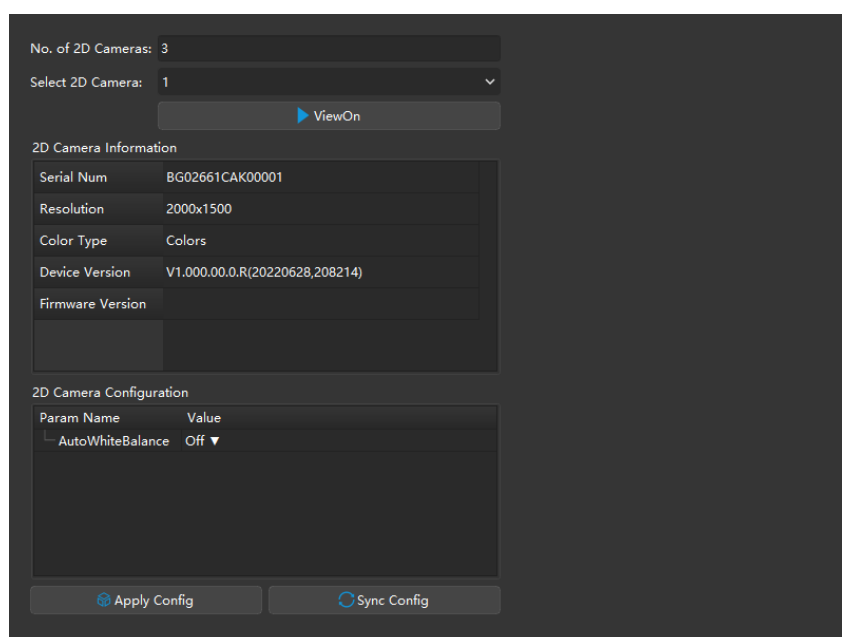


Follow these steps to check the depth fluctuation in the depth map:

1. Click Tools > Depth Analyzer to open **Depth Analyzer**.
2. In the depth map on the left (① in the above figure), find the area where you want to check the depth fluctuation.
3. Hold the left mouse button and drag to draw a line. The depth data along the line will be plotted on the right (② in the above figure).
4. Check the fluctuation of the depth values in the plot:
 - If the fluctuation is small compared to the actual depth of the scene, then the quality of the depth data is good.
 - If the fluctuation is large compared to the actual depth of the scene, then the quality of the depth data is poor. Please adjust the parameters in the **3D Parameters** and **Point Cloud Processing** categories to improve the quality of the depth data.

5.4.4. Check and Configure 2D Cameras

This tool is used for checking and configuring the 2D cameras of a Mech-Eye Industrial 3D Camera.



Check 2D Camera Information

In the **2D Camera Information** section, you can check the serial number, resolution, color type, device version, and firmware version of a 2D camera.

To check the above information of a 2D camera, select the 2D camera in the **Select 2D Camera** menu first.

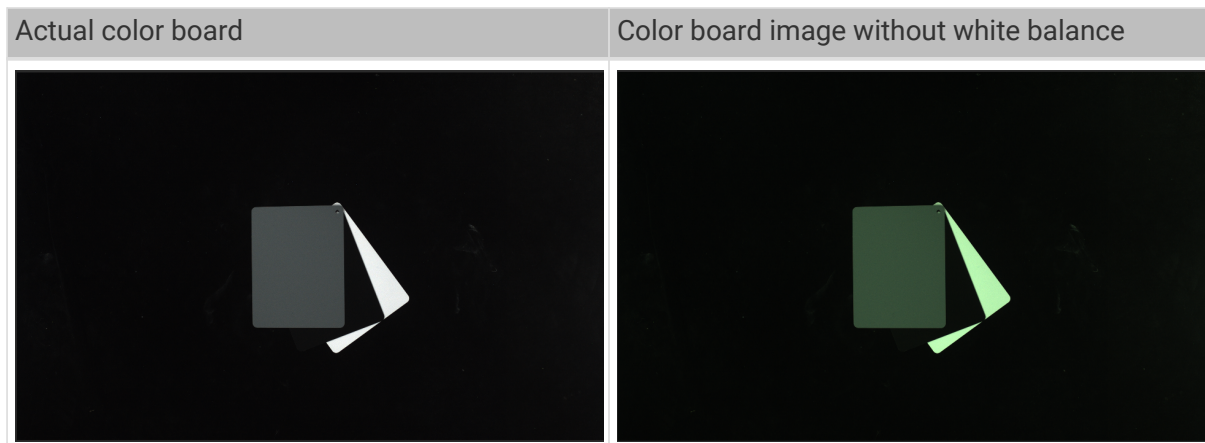
Adjust White Balance



Only available for color 2D cameras. Select a 2D camera and then check the **Color Type** in **2D Camera Information** to see if it is a color 2D camera.

If the color of the obtained 2D image differs significantly from that of the objects, you can adjust the white balance setting. Otherwise, distorted color of the 2D image will result in distorted color in the textured point cloud and affect subsequent processing. For deep learning, if 2D images with distorted color are used for model training, the distortions will be extracted and trained as the object features, thus affecting the model performance on object recognition.

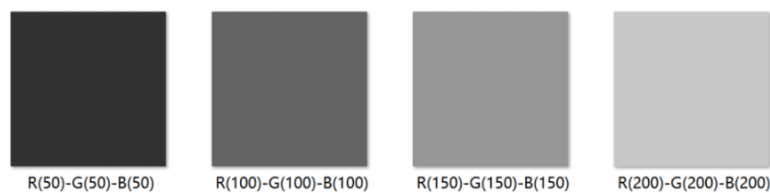
Comparison Example



Preparation

Before adjusting the white balance, please complete the following preparation:

1. Make sure that the 2D image is not over- or under-exposed. If the 2D image is over- or under-exposed, please adjust the [parameters for the 2D image](#).
2. Prepare a neutral-gray color board. Refer to the following image for examples of neutral-gray colors. Then, place the color board close to the camera and acquire data to ensure, as much as possible, that only the color board is captured.



Instructions

Please follow these steps to adjust the white balance of a color 2D camera:

1. Select the 2D camera whose white balance needs to be adjusted, and click [**ViewOn**]. The camera will acquire data continuously.
2. In **2D Camera Configuration**, set **AutoWhiteBalance** to **Once** or **Continuous**, and the camera will start to adjust the white balance.
 - If the lighting conditions remain relatively unchanged, it is recommended to use **Once**.
 - If the lighting conditions vary, it is recommended to use **Continuous**.
3. Check the color of the gray color board in the image.
 - If the color still appears distorted, move the board until the color appears consistent to the actual color board.
 - If the color appears consistent to the actual color board, set **AutoWhiteBalance** to **Off**, and click [**ViewOff**] to finish adjusting the white balance.
4. Click [**Apply Config**] to save the configuration to the camera.

5.4.5. FOV Calculator

This tool is used for helping determine the mounting height of the camera based on your needs for camera FOV. Based on the entered working distance, this tool calculates the FOV height and width at this distance for the camera.

The calculation result is for reference only.

Calculate Camera FOV

Please follow these steps calculate the camera FOV:

1. Select the camera model in **Select camera model**.



If the model in use is not listed in the drop-down menu, click **Other models** at the bottom of the menu to view more models.

2. Enter the working distance in the **Enter working distance** field. The **FOV length** and **FOV width** are calculated based on the entered working distance. If the calculated FOV does not meet your needs, proceed to the next step.



- The range of the working distance is the recommended working distance of the selected camera model. Refer to [Camera Working Distance](#).
- The entered working distance must be within the above range. Otherwise, the entered value will be automatically adjusted.

3. Click the arrow buttons to the right or scroll the mouse wheel to adjust the working distance, until the calculated FOV meets your needs. Use this working distance to estimate the appropriate mounting height of the camera.

Camera Working Distance

The following table lists the default values and ranges of camera working distances (unit: m).

Camera model	Working distance	
	Default value	Value range
DEEP	3	1.2–3.5
LSR L	2.5	1.2–3
LSR S	1	0.5–1.5
Log M	2	0.8–2
Log S	1	0.5–1
NANO	0.5	0.3–0.6
PRO M	2	1–2
PRO S	1	0.5–1
PRO XS	0.5	0.3–0.6
UHP-140	0.3	0.28–0.32
Deep	3	1.2–3.5
Laser L	2.5	1.5–3
Laser L Enhanced	2.5	1.5–3
Pro M Enhanced	2	0.8–2
Pro S Enhanced	1	0.5–1

5.4.6. Custom Reference Frame

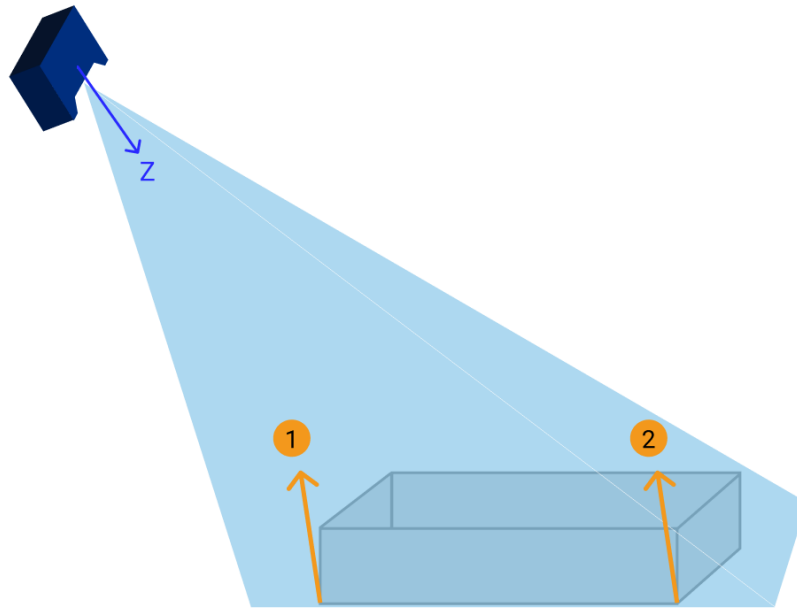
This tool is used to define a custom reference frame for display. The depth map and point cloud can be displayed in the custom reference frame.



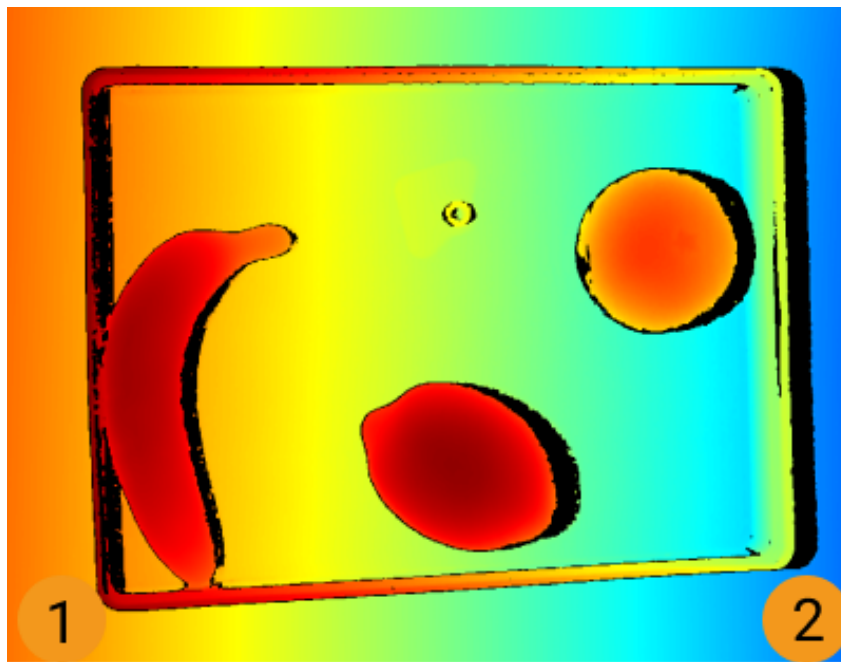
The custom reference frame only affects the display in Mech-Eye Viewer and does not affect the actual data in the depth map and point cloud.

Suitable Scenarios

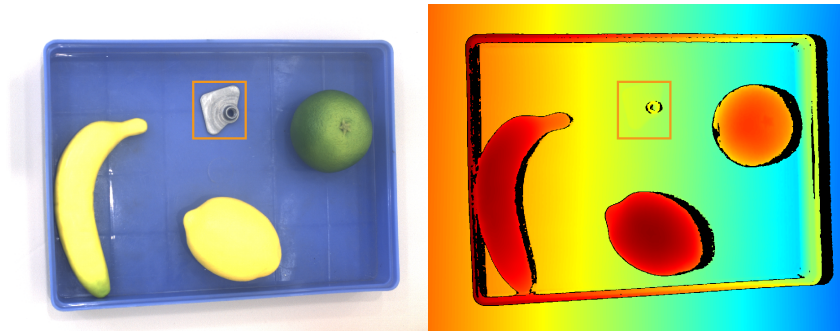
A custom reference frame is suited for scenarios where the camera is not mounted directly above the target object, as shown below.



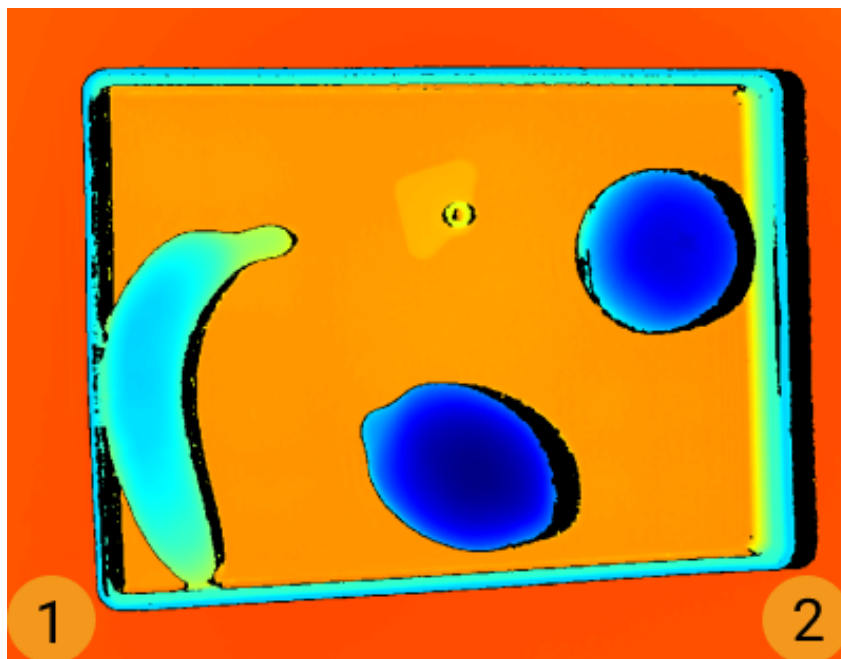
In the above figure, Points ① and ② are at the same height in the eye of an observer. However, since the Z-axis of the camera reference frame is not perpendicular to the plane where the target object is located, the depth values of the two points differ in the obtained depth map and point cloud.



In addition, in the camera reference frame, the thickness of a flat, thin object is “subsumed” by the depth change of the surface where the object is located, making it difficult to observe the object itself.



By defining a custom reference frame, you can adjust the display of the depth map and point cloud, making the depth values of points ① and ② the same and the thin target object easily observable, as shown below.



Instructions

Follow these steps to define a custom reference frame:

1. [Adjust the position of the point cloud](#) and find the surface to be used as the XY-plane of the custom reference frame.
2. Hold the **Shift** key and click on the point cloud to select 3 points. The 3 points serve the following functions:
 - The first point is the origin of the reference frame.
 - The second point determines the positive direction of the X-axis.
 - The third point determines the positive direction of the Y-axis.
3. After the 3 points are selected, the reference frame is generated according to the right-hand rule. Rotate and zoom the point cloud to check if the reference frame meets your needs:
 - Check if the three points are located on the same surface.
 - Check if the orientation of the Z-axis is correct.

If not, Click [**Reset**] in the lower right, and then repeat step 2.

4. Click [**OK**] to save the custom reference frame.

Use the Custom Reference Frame

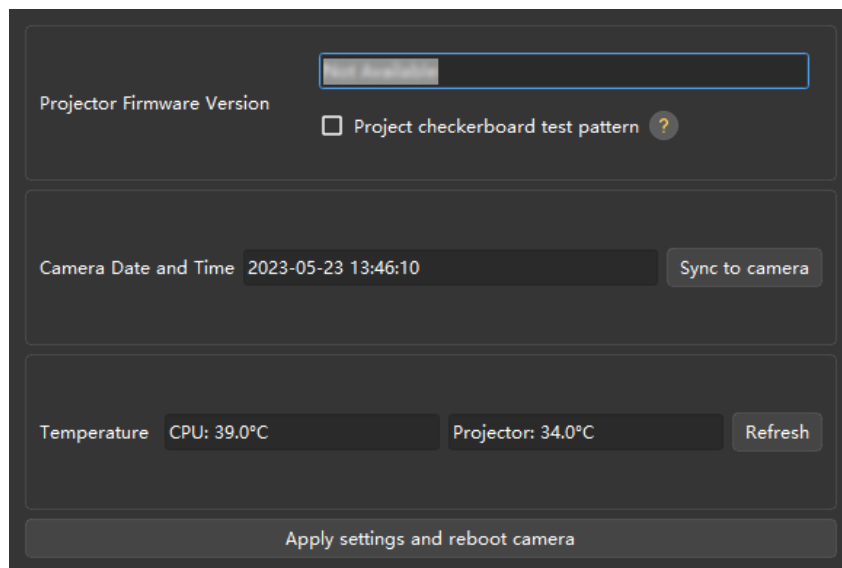
To display the depth map and point cloud in the custom reference frame, follow these steps:

1. In the data display area, switch to **Depth Map** or **Point Cloud**.
2. At the top of the left panel, click the **Reference Frame** drop-down menu and select **Custom**. The depth map and point cloud will be displayed in the custom reference frame.

 The reference frame setting is simultaneously applied to both depth map and point cloud.

5.4.7. Camera Controller

This tool is used for checking the project firmware version, time, and temperature of the camera and change the image resolution.



Check Camera Information

You can check the following information in **Camera Controller**.

- Projector firmware version: the projector firmware version of the currently connected camera.
 - Project chessboard test pattern: Check this option to project a chessboard pattern used for checking the connection and focus of the projector.

 This option is only available for DLP cameras.

- If the pattern is successfully projected, the project is connected.
 - If the pattern is clear and not blurry, the projector is in focus.
 - If the pattern is not projected or is blurry, please contact Technical Support.
- Camera Date and Time: the date and time stored in the camera.
 - Click [**Sync to camera**] to sync the computer's date and time to the camera.

- Temperature: the temperatures of the CPU and projector.

Click [**Refresh**] to obtain the latest temperature data.

Change Image Resolution



 This function is available in the **Administrator** account.

The resolution of the 2D image or depth map is adjustable for some camera models. Using the lower resolution can speed up data acquisition effectively and thus better cope with applications with short cycle time.

- LSR (V4) series: The resolution of the 2D image can be either 4000 × 3000 or 2000 × 1500.
- DEEP (V4) series: The resolution of the depth map can be either 2048 × 1536 or 1024 × 768.

Please follow these steps to change the image resolution of the camera:


1. Switch to the **Administrator** account and open **Camera Controller**.
2. In the **Resolution** section, select the needed resolution from the drop-down menu of **2D image** or **Depth map**.
3. Click [**Apply settings and reboot camera**] at the bottom, and then click [**Confirm**] in the pop-up window to restart the camera.

 The camera restart may take several minutes. Please wait until it completes. Click  in the upper left to update the list.

4. After the camera is displayed in the list again, connect to the camera. The 2D image or depth map that you acquire now will have the selected resolution.

5.4.8. Full Pallet Simulator

This tool is used for checking if the top layer of a full load on a pallet is entirely located within the camera FOV. The camera mounting height can be adjusted according to the result.

- 
- This tool can only be used for applications where the camera is mounted perpendicularly above the pallet.
 - The displayed result is for reference only.

Preparation

Before using **Full Pallet Simulator**, please complete the following preparation:

- Check the dimensions of the full load.
- Check if any limitations on the camera mounting location exist, such as the maximum mounting height allowed.
- Prepare and place the pallet within the camera FOV, acquire the point cloud of the pallet, and check the point cloud quality.
- Adjust the position of the pallet so that the edges of the pallet point cloud are parallel to the sides of the camera FOV.
- Set an **ROI** to retain only the point cloud of the pallet.

Instructions

1. Click Tools > Full Pallet Simulator to open **Full Pallet Simulator**.
2. [Adjust the position of the point cloud](#) until you can see the entire point cloud from the side.
3. Click [**Create cuboid**] to add a cuboid that represents the full load.



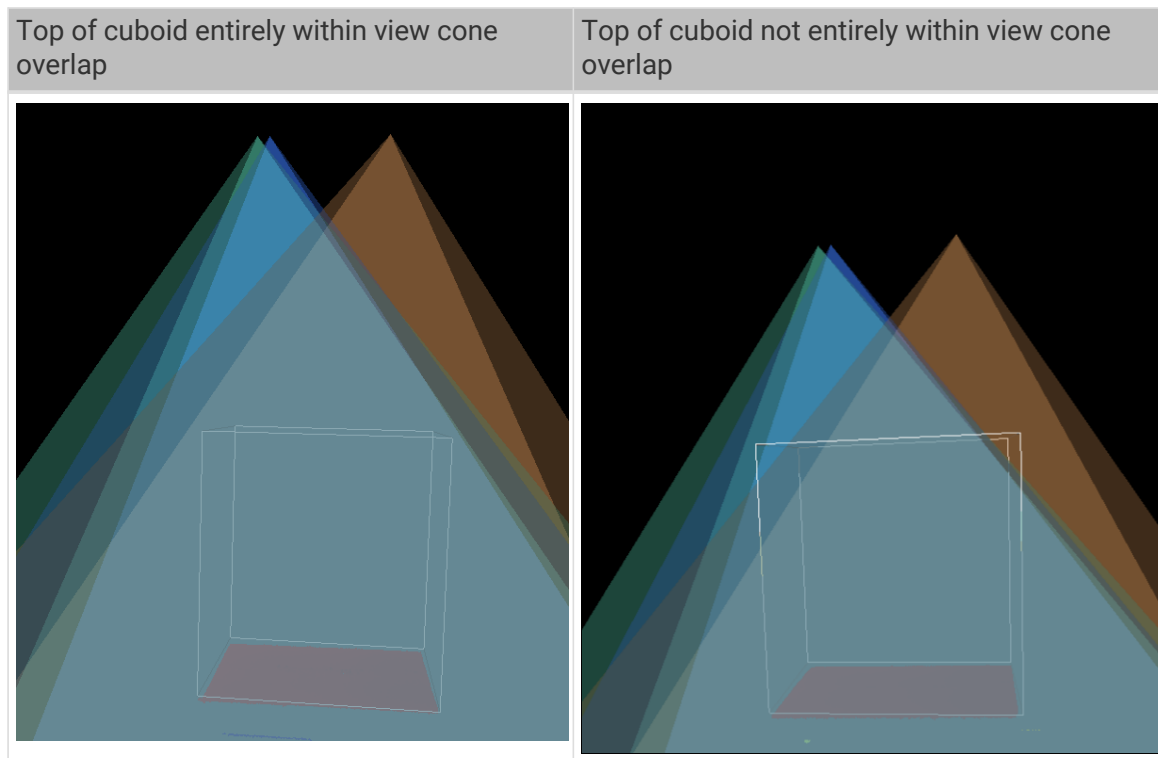
The cuboid cannot be rotated. If the edges of the cuboid and those of the pallet point cloud are not aligned, please exit **Full Pallet Simulator**, adjust the position of the pallet, and acquire data again.

4. In **Cuboid Dimensions**, enter the actual dimensions of the full load.
5. (Optional) Adjust the position of the cuboid: if the bottom of the cuboid is not on the pallet point cloud, please adjust the **z** value in **Center of Cuboid Bottom**, until the cuboid bottom is on the pallet point cloud.



Please record the amount of **z** adjustment made.

6. Rotate the point cloud and check if the top of the cuboid is confined to the space where the view cones overlap. Use the examples below as a reference.



7. Evaluate camera mounting height:
 - If the top of the cuboid is entirely confined to the view cone overlap, the camera mounting height is appropriate.
 - If the top of the cuboid is not entirely confined to the view cone overlap, the camera is mounted too low. Please proceed to the next step.
8. Increase the **z** value in **Center of Cuboid Bottom**, until the top of the cuboid is entirely confined to the view cone overlap.



Please record the amount of **z** adjustment made.

9. Calculate the amount of adjustment for the camera mounting height: the camera mounting height should be raised by (the z adjustment in step 8 - the z adjustment in step 5).



Modifications in **Full Pallet Simulator** are not saved. Please make sure you recorded the z adjustments.

6. Mech-Eye API

You can use Mech-Eye API to connect to the camera, adjust parameters, and obtain and save 2D images, depth maps, and point clouds.

The C++ and C# Mech-Eye API are included in the Mech-Eye SDK installation package. The Python Mech-Eye API need to be downloaded and installed separately.



Mech-Eye API 2.2.0 has been restructured to provide a clearer structure while keeping all the previously available functions. New functions have also been added.

- If you wish to use the 2.2.0 version of Mech-Eye API in your existing client program, please refer to [Migration Guide](#) and modify your client program.
- If you need to upgrade the camera firmware to 2.2.0 but need to keep using the 2.1.0 version of Mech-Eye API in your client program, please refer to [Backward Compatibility](#) and complete the corresponding actions for the language in use.

Instructions on **installing Mech-Eye SDK**:

[Mech-Eye SDK Installation Guide \(Windows\)](#)

[Mech-Eye SDK Installation Guide \(Ubuntu\)](#)

Introduction of the **overall structure of Mech-Eye API**:

[Functional Hierarchy of Classes](#)

Introduction of the **workflow of using Mech-Eye API to control the camera**:

[Operation Workflow](#)

The Mech-Eye API reference guide:

[Mech-Eye API Reference](#)

Instructions on **installing, configuring, building, and running Mech-Eye API samples**:

[Sample Usage Guide](#)

Instructions on **using the supplementary tools for Mech-Eye API**:

[Firmware Upgrade Tool](#)

IP Configuration Tool

List of the **methods and parameters renamed in version 2.0.0**:


[C++ Parameter Name Changes \(2.0.0\)](#)

[C# Method and Parameter Name Changes \(2.0.0\)](#)

 Not applicable to Mech-Eye API below 2.0.0.

6.1. Mech-Eye SDK Installation Guide (Ubuntu)

This topic guides you through the downloading, installation, upgrade, and uninstallation of Mech-Eye SDK on Ubuntu.

 If you are using the Windows operating system, please refer to [Mech-Eye SDK Installation Guide](#).

Download Mech-Eye SDK


Mech-Eye SDK installation packages for the AMD64 and ARM64 architectures are provided:

- AMD64: **Mech-Eye_API_2.2.2_amd64.deb**
- ARM64: **Mech-Eye_API_2.2.2_arm64.deb**

Please download the corresponding [Mech-Eye SDK installation package](#) according to your needs.

 Use the `arch` command to check the system architecture.

Install Mech-Eye SDK

 If you have installed Mech-Eye SDK already, please [upgrade Mech-Eye SDK](#) instead.

Follow these steps to install Mech-Eye SDK on Ubuntu:

1. Enter the path where the installation package is stored and execute the following command to install Mech-Eye SDK:
 - If the system architecture is AMD64, execute the following command:

```
sudo dpkg -i 'Mech-Eye_API_2.2.2_amd64.deb'
```

- If the system architecture is ARM64, execute the following command:

```
sudo dpkg -i 'Mech-Eye_API_2.2.2_arm64.deb'
```

2. Execute the following command to check if the installation succeeded. If the Mech-Eye SDK version is printed, the installation succeeded.

```
dpkg -l | grep mecheyeapi
```

Upgrade Mech-Eye SDK

If you have installed Mech-Eye SDK already, follow these steps to upgrade Mech-Eye SDK:

1. Execute the following command to check if Mech-Eye SDK has been installed:

```
dpkg -l | grep mecheyeapi
```

2. Execute the following command to uninstall Mech-Eye SDK:

```
sudo dpkg -P mecheyeapi
```

3. Enter the path where the installation package is stored and execute the following command to install Mech-Eye SDK:

- If the system architecture is AMD64, execute the following command:

```
sudo dpkg -i 'Mech-Eye_API_2.2.2_amd64.deb'
```

- If the system architecture is ARM64, execute the following command:

```
sudo dpkg -i 'Mech-Eye_API_2.2.2_arm64.deb'
```

4. Execute the following command to check if the installation succeeded. If the Mech-Eye SDK version is printed, the installation succeeded.

```
dpkg -l | grep mecheyeapi
```

Uninstall Mech-Eye SDK

Follow these steps to uninstall Mech-Eye SDK:

1. Execute the following command to uninstall Mech-Eye SDK:

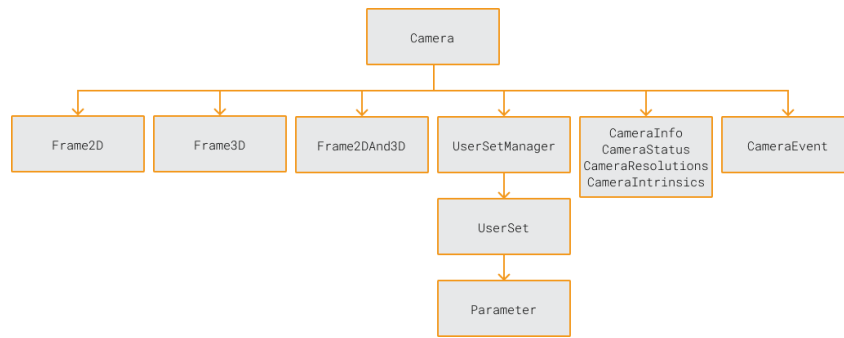
```
sudo dpkg -P mecheyeapi
```

2. Execute the following command to check if the uninstallation succeeded. If no messages are printed, the uninstallation succeeded.


```
dpkg -l | grep mecheyeapi
```

6.2. Functional Hierarchy of Classes

This topic introduces the functional hierarchy of the classes in Mech-Eye API. In the following figure, the classes higher up in the hierarchy provide methods for accessing the objects of the lower classes. Therefore, the higher classes should be instantiated earlier in the client application.



The following sections introduce the functions of each class in the above figure.

Camera

The **Camera** class represents a camera. It is the highest class in the functional hierarchy in Mech-Eye API. In the client application, this class should be instantiated first. Through the **Camera** class, you can access the objects of other classes.

The methods provided by the **Camera** class mainly include:

- Discover cameras
- Connect to a camera
- Acquire data
- Obtain the camera information
- Access the objects of the **UserSetManager** and **UserSet** classes

Frame2D

The **Frame2D** class is used to store the 2D data. Using the data in this class, you can generate the 2D image.

Frame3D

The **Frame3D** class is used to store the 3D data. Using the data in this class, you can generate the depth map and untextured point cloud.

Frame2DAnd3D

The **Frame2DAnd3D** class is used to store the 2D and 3D data together. Using the data in this class, you can generate the 2D image, depth map, and textured point cloud.

UserSetManager

The **UserSetManager** class provides methods for **managing parameter groups**, which mainly include:

- Check all available parameter groups
- Select a parameter group
- Create a parameter group
- Delete a parameter group
- Import parameter groups
- Export parameter groups
- Access the object of the **UserSet** class

UserSet

The **UserSet** class represents a parameter group and provides methods relevant to the currently selected parameter group. These methods mainly include:

- Obtain all parameters in the current parameter group
- Reset the current parameter group
- Save parameters to the current parameter group
- Obtain parameter information
- Obtain and adjust parameter values
- Rename the current parameter group

Parameter

The **Parameter** class represents a single parameter and provides methods relevant to individual parameters. These methods mainly include:

- Obtain parameter information (including parameter name, parameter description, data type, maximum value, minimum value, minimum increment, read and write permissions, etc.)
- Obtain and adjust parameter values

CameraInfo, etc.

The following classes are used to store the information and properties of the camera:

- **CameraInfo**: model, serial number, hardware and firmware versions, IP configurations, and more
- **CameraStatus**: the temperatures of the camera projector and CPU
- **CameraResolutions**: the resolutions of the 2D image and depth map
- **CameraIntrinsics**: camera intrinsic parameters

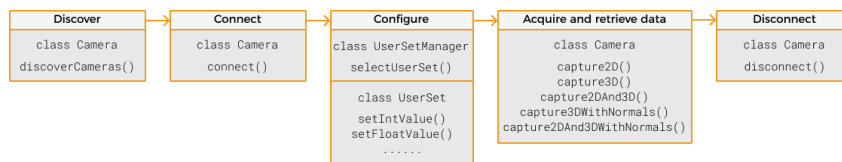
CameraEvent

The `CameraEvent` class provides methods for detecting events related to the camera. Currently, camera disconnection events can be detected.

This topic introduced the functional hierarchy of the classes in Mech-Eye API. The next topic will introduce the basic workflow of using Mech-Eye API to control a camera.

6.3. Operation Workflow

This topic introduces the basic workflow of using Mech-Eye API to control a camera. The figure and descriptions in this topic use the class and method names of C++ Mech-Eye API. For the corresponding names of different languages, please refer to the example codes.



Discover

Call the `discoverCameras()` method in the `Camera` class to enumerate all currently connectable cameras and obtain the information of each camera.

C++

```
std::vector<mmind::eye::CameraInfo> cameraInfoList = mmind::eye::
Camera::discoverCameras();
```

C#

```
var cameraInfoList = Camera.DiscoverCameras();
```

Python

```
camera_info_list = Camera.discover_cameras()
```

Connect

After instantiating the `Camera` class, call the `connect()` method in this class to connect to the corresponding camera using the IP address or the device information obtained by the `discoverCameras()` method.

C++

```
mmind::eye::Camera camera;
mmind::eye::ErrorStatus status = camera.connect(cameraInfoList
[inputIndex]);
```

```
mmind::eye::Camera cameras;
mmind::eye::ErrorStatus status = camera.connect("192.168.0.10");
```

C#

```
var camera = new Camera();
var status = camera.Connect(cameraInfoList[inputIndex]);
```

```
var camera = new Camera();
var status = camera.Connect("192.168.0.10");
```

Python

```
camera = Camera()
camera.connect("192.168.0.10")
```

Configure

Call the methods in the `UserSetManager` and `UserSet` classes to select the parameter group on the camera and adjust the parameters.

1. Call the `selectUserSet` method in the `UserSetManager` class to select the parameter group to be used.



You can obtain all the available parameter groups on the camera through the `getAllUserSetNames()` method in this class.

C++

```
mmind::eye::UserSetManager userSetManager = camera.userSetManager();
std::vector<std::string> userSets;
auto status = userSetManager.getAllUserSetNames(userSets);
status = userSetManager.selectUserSet(userSets.front());
```

C#

```
var userSetManager = camera.UserSetManager();
List<string> userSets = new List<string>();
var status = userSetManager.GetAllUserSetNames(ref userSets);
status = userSetManager.SelectUserSet(userSets[0]);
```

Python

```
user_set_manager = camera.userSetManager()
error, user_sets = user_set_manager.get_all_user_set_names()
user_set_manager.select_user_set(user_sets[0])
```

- Call the `getAvailableParameters()` method in the `UserSet` class to obtain the information of all the parameters in the current parameter group and check the data type of each parameter.

C++

```
mmind::eye::UserSet currentUserSet = camera.currentUserSet();
std::vector<Parameter*> parameters = currentUserSet
    .getAvailableParameters();
```

C#

```
var currentUserSet = camera.CurrentUserSet();
var parameters = currentUserSet.GetAvailableParameters();
```

Python

```
current_user_set = camera.current_user_set()
arameters = current_user_set.get_available_parameters()
```

- Call the `getIntValue()` and other similar methods in the `UserSet` class to obtain the current value of a parameter.

C++

```
int fringeContrastThreshold = 0;
currentUserSet.getIntValue(mmind::eye::pointcloud_processing_setting
    ::FringeContrastThreshold::name, fringeContrastThreshold);
```

C#

```
var fringeContrastThreshold = new int();
camera.CurrentUserSet().GetIntValue(MMind.Eye.PointCloudProcessingSetting.FringeContrastThreshold.Name, ref fringeContrastThreshold);
```

Python

```
error, fringe_contrast_threshold = current_user_set.get_int_value
(PointCloudFringeContrastThreshold.name)
```

4. Call the `setIntValue()` and other similar methods in the `UserSet` class to set the value of a parameter.

C++

```
auto status = currentUserSet.setIntValue(mmind::eye
::pointcloud_processing_setting::FringeContrastThreshold::name, 15);
```

C#

```
var status = currentUserSet.SetIntValue(MMind.Eye
.PointCloudProcessingSetting.FringeContrastThreshold.Name, 15);
```

Python

```
status = current_user_set.set_int_value
(PointCloudFringeContrastThreshold.name, 15)
```

5. Call the `saveAllParametersToDevice()` method in the `UserSet` class to save the set parameter values to the camera.

C++

```
auto status = currentUserSet.SaveAllParametersToDevice();
```

C#

```
var status = currentUserSet.SaveAllParametersToDevice();
```

Python

```
status = current_user_set.save_all_parameters_to_device()
```

Acquire and Retrieve Data

Mech-Eye API provides various methods for acquiring and retrieving different types of data. Using these methods, you can obtain the 2D and 3D data separately or together.

- Acquire and retrieve the data for generating the 2D image:

C++

```
mmind::eye::Frame2D frame2D;
camera.capture2D(frame2D);
mmind::eye::Color2DImage color = frame2D.getColorImage();
mmind::eye::GrayScale2DImage gray= frame2D.getGrayScaleImage();
```

C#

```
var frame = new Frame2D();
camera.Capture2D(ref frame);
var color = frame.GetColorImage();
var gray = frame.GetGrayScaleImage();
```

Python

```
frame_2d = Frame2D()
camera.capture_2d(frame_2d)
color = frame.get_color_image()
gray = frame.get_gray_scale_image()
```

- Acquire and retrieve the data for generating the depth map and untextured point cloud:

C++

```
mmind::eye::Frame3D frame3D;
camera.capture3D(frame3D);
mmind::eye::DepthMap depth = frame3D.getDepthMap();
mmind::eye::PointCloud cloud= frame3D.getUntexturedPointCloud();
```

C#

```
var frame = new Frame3D();
camera.Capture3D(ref frame);
var depth = frame.GetDepthMap();
var cloud = frame.GetUntexturedPointCloud();
```

Python

```

frame_3d = Frame3D()
camera.capture_3d(frame_3d)
depth = frame.get_depth_map()
cloud = frame_3d.get_untextured_point_cloud()
    
```

- Acquire and retrieve the data for generating the 2D image, depth map, and textured point cloud:

C++

```

mmind::eye::Frame2DAnd3D frame2DAnd3D;
camera.capture2DAnd3D(frame2DAnd3D);
mmind::eye::Color2DImage color = frame2DAnd3D.frame2D
().getColorImage();
mmind::eye::DepthMap depth = frame2DAnd3D.frame3D().getDepthMap();
mmind::eye::TexturedPointCloud cloud = frame2DAnd3D
.getTexturedPointCloud();
    
```

C#

```

var frame = new Frame2DAnd3D();
camera.Capture2DAnd3D(ref frame);
var color = frame2DAnd3D.Frame2D().GetColorImage();
var depth = frame2DAnd3D.Frame3D().GetDepthMap();
var cloud = frame2DAnd3D.GetTexturedPointCloud();
    
```

Python

```

frame = Frame2DAnd3D()
camera.capture_2d_and_3d(frame)
color = frame.frame_2d().get_color_image()
depth = frame.frame_3d().get_depth_image()
cloud = frame.get_textured_point_cloud()
    
```

- Acquire and retrieve the data for generating the depth map and untextured point cloud with normals:

C++

```

mmind::eye::Frame3D frame3D;
camera.capture3DWithNormal(frame3D);
mmind::eye::DepthMap depth = frame3D.getDepthMap();
    
```



```
PointCloudWithNormals pointCloud = frame3D
    .getUntexturedPointCloudWithNormals();
```

C#

```
var frame3D = new Frame3D();
camera.Capture3DWithNormal(ref frame3D);
var depth = frame3D.GetDepthMap();
var pointCloud = frame3D.GetUntexturedPointCloudWithNormals();
```

Python

```
frame_3d = Frame3D()
camera.capture_3d_with_normal(frame_3d)
depth = frame_3d.get_depth_map()
point_cloud = frame_3d.get_untextured_point_cloud_with_normals()
```

- Acquire and retrieve the data for generating the 2D image, depth map, and textured point cloud with normals:

C++

```
mmind::eye::Frame2DAnd3D frame2DAnd3D;
camera.capture2DAnd3DWithNormal(frame2DAnd3D);
mmind::eye::Color2DImage color = frame2DAnd3D.frame2D
    ().getColorImage();
mmind::eye::DepthMap depth = frame2DAnd3D.frame3D().getDepthMap();
TexturedPointCloudWithNormals pointCloud = frame2DAnd3D
    .getTexturedPointCloudWithNormals();
```

C#

```
var frame2DAnd3D = new Frame2DAnd3D();
camera.Capture2DAnd3DWithNormal(ref frame2DAnd3D);
var color = frame2DAnd3D.Frame2D().GetColorImage();
var depth = frame2DAnd3D.Frame3D().GetDepthMap();
var pointCloud = frame2DAnd3D.GetTexturedPointCloudWithNormals();
```

Python

```
frame_2d_and_3d = Frame2DAnd3D()
camera.capture_2d_and_3d_with_normal(frame_2d_and_3d)
color = frame_2d_and_3d.frame_2d().get_color_image()
depth = frame_2d_and_3d.frame_3d().get_depth_map()
point_cloud = frame_2d_and_3d.
```

```
get_textured_point_cloud_with_normals()
```

Disconnect

Call the `disconnect()` method in the `Camera` class to disconnect from the current camera.

C++

```
camera.disconnect();
```

C#

```
camera.Disconnect();
```

Python

```
camera.disconnect()
```

This topic introduced the basic workflow of using Mech-Eye API to control a camera. The next topic provides the reference manual for Mech-Eye API.

6.4. Migration Guide from Version 2.1.0 to Version 2.2.0

Mech-Eye API 2.2.0 has been restructured to provide a clearer structure while keeping all the previously available functions. New functions have also been added.

This topic lists the main changes of Mech-Eye API 2.2.0 compared to Mech-Eye API 2.1.0. If you wish to use the 2.2.0 version of Mech-Eye API in your existing client program, you can refer to this topic and modify your client program.



- For the overall structure of Mech-Eye API 2.2.0, please refer to [Functional Hierarchy of Classes](#).
- For the basic workflow of using Mech-Eye API 2.2.0 to control the camera, please refer to [Functional Hierarchy of Classes](#).

Import Modules

In Mech-Eye API 2.2.0, the following changes have been made to the statement for importing modules:

2.1.0	2.2.0
<p>C++</p> <pre>#include "MechEyeApi.h" using namespace mmind::api;</pre> <p>C#</p> <pre>using mmind.apiSharp;</pre> <p>Python</p> <pre>from MechEye import Device</pre>	<p>C++</p> <pre>#include "Camera.h" using namespace mmind::eye;</pre> <p>C#</p> <pre>using MMind.Eye;</pre> <p>Python</p> <pre>from mecheye.shared import * from mecheye.area_scan_3d_camera import * from mecheye.area_scan_3d_camera_u tils import *</pre>

Create Camera Object

In Mech-Eye API 2.2.0, the following changes have been made to the statement for creating an object representing the cameras:

2.1.0	2.2.0
<p>C++</p> <pre>mmind::api::MechEyeDevice device;</pre> <p>C#</p> <pre>var device = new MechEyeDevice();</pre> <p>Python</p> <pre>device = Device()</pre>	<p>C++</p> <pre>mmind::eye::Camera camera;</pre> <p>C#</p> <pre>var camera = new Camera();</pre> <p>Python</p> <pre>camera = Camera()</pre>

Discover and Connect to Camera

In Mech-Eye API 2.2.0, the following changes have been made to the statement for discovering and connecting to a camera:

2.1.0	2.2.0
<p>C++</p> <pre>std::vector<mmind::api::MechEyeDeviceInfo> deviceInfoList = mmind::api::MechEyeDevice::enumerateMechEyeDeviceList(); mmind::api::MechEyeDevice device; device.connect(deviceInfoList[0]);</pre> <p>C#</p> <pre>var deviceInfoList = MechEyeDevice.EnumerateMechEyeDeviceList(); var device = new MechEyeDevice(); device.Connect(deviceInfoList[0]);</pre> <p>Python</p> <pre>device_list = Device.get_device_list() device = Device() device.connect(device_list[0])</pre>	<p>C++</p> <pre>std::vector<mmind::eye::CameraInfo> cameraInfoList = mmind::eye::Camera::discoverCameras(); mmind::eye::Camera camera; camera.connect(cameraInfoList[0]);</pre> <p>C#</p> <pre>var cameraInfoList = Camera.DiscoverCameras(); var camera = new Camera(); camera.Connect(cameraInfoList[0]);</pre> <p>Python</p> <pre>camera_infos = Camera.discover_cameras() camera = Camera() camera.connect(camera_infos[0])</pre>

Manage Parameter Groups

Mech-Eye API 2.2.0 uses classes to organize the functions related to parameter groups and parameters. For details, please refer to [Functional Hierarchy of Classes](#). The changes to frequently used functions are listed here.

2.1.0

C++

```

std::vector<std::string>
userSets;
device.getAllUserSets(userSets);

std::string
currentUserSetName;
device.getCurrentUserSet(currentUserSet);

device.setCurrentUserSet(userSets.front());

device.addUserSet("NewSetting");

device.saveAllSettingsToUserSets();
    
```

C#

```

var userSets = new List<string>();
device.GetAllUserSets(ref userSets);

std::string
currentUserSetName = "";
device.GetCurrentUserSet(ref currentUserSetName);

device.SetCurrentUserSet(userSets[0])

device.AddUserSet("NewSetting");

device.SaveAllSettingsToUserSets();
    
```

Python

```
# Obtain the names of all
```

2.2.0

C++

```

std::vector<std::string>
userSets;
mmind::eye::UserSetManager&
userSetManager = camera
.userSetManager();
userSetManager.getAllUserSetNames(userSets);

mmind::eye::UserSet&
curSettings = userSetManager
.currentUserSet();
std::string currentName;
curSettings.getName(currentName);

userSetManager.selectUserSet(
userSets.front());

userSetManager.addUserSet("NewSetting");

camera.currentUserSet().saveAllParametersToDevice();
    
```

C#

```

var userSetManager = camera
.UserSetManager();
var userSets = new List<string>();
userSetManager.GetAllUserSetNames(ref userSets);

var currentName =
userSetManager.CurrentUserSet().GetName();

userSetManager.SelectUserSet(
userSets[0]);

userSetManager.AddUserSet("NewSetting");
    
```

Set and Obtain Parameter Values

Mech-Eye API 2.2.0 uses classes to organize the functions related to parameter groups and parameters. For details, please refer to [Functional Hierarchy of Classes](#).

Meanwhile, Mech-Eye API 2.2.0 provides methods for setting and obtaining parameters according to the data types of the parameters.

The following data types of parameters are distinguished in Mech-Eye API 2.2.0:

- `_Int`
- `_Float`
- `_Bool`
- `_Enum`
- `_Roi`
- `_Range`
- `_FloatArray`

The following sections provide examples of each data type.

`_Int` Type

The **Stripe Contrast Threshold** parameter in the **Point Cloud Processing** category is used as an example.

2.1.0

C++

```
device.setFringeContrastThres
hold(15);

int fringeContrastThreshold =
0;
device.getFringeContrastThres
hold(fringeContrastThreshold)
;
```

C#

```
device.SetFringeContrastThres
hold(15);

var fringeContrastThreshold =
new int();
device.GetFringeContrastThres
hold(ref
fringeContrastThreshold);
```

Python

```
device.set_fringe_contrast_th
reshold(15)

fringe_contrast_threshold =
device.get_fringe_contrast_th
reshold()
```

2.2.0

C++

```
#include
"area_scan_3d_camera/paramete
rs/PointCloudProcessing.h"

camera.currentUserSet().setIn
tValue(mmind::eye::pointcloud
_processing_setting::FringeCo
ntrastThreshold::name, 15);

int fringeContrastThreshold =
0;
camera.currentUserSet().getIn
tValue(mmind::eye::pointcloud
_processing_setting::FringeCo
ntrastThreshold::name,
fringeContrastThreshold);
```

C#

```
camera.CurrentUserSet().SetIn
tValue(MMind.Eye.PointCloudPr
ocessingSetting.FringeContras
tThreshold.Name, 15);

var fringeContrastThreshold =
new int();
camera.CurrentUserSet().GetIn
tValue(MMind.Eye.PointCloudPr
ocessingSetting.FringeContras
tThreshold.Name, ref
fringeContrastThreshold);
```

Python

```
camera.current_user_set().set
_int_value(PointCloudFringeCo
ntrastThreshold.name, 15)

error,
fringe_contrast_threshold =
camera.current_user_set().get
_int_value(PointCloudFringeCo
```

_Float Type

The **Exposure Time** parameter in the **2D Parameters** category is used as an example.

2.1.0

C++

```
device.setScan2DExposureTime(
    100);

double scan2DExposureTime =
    0;
device.getScan2DExposureTime(
    scan2DExposureTime);
```

C#

```
device.setScan2DExposureTime(
    100);

var scan2DExposureTime = new
    double();
device.GetScan2DExposureTime(
    ref scan2DExposureTime);
```

Python

```
device.set_scan_2d_exposure_t
ime(100)

scan_2d_exposure_time =
device.get_scan_2d_exposure_t
ime()
```

2.2.0

C++

```
#include
"area_scan_3d_camera/paramete
rs/Scanning2D.h"

camera.currentUserSet().setFl
oatValue(mmind::eye::scanning
2d_setting::ExposureTime::nam
e, 100);

double scan2DExposureTime =
    0;
camera.currentUserSet().getFl
oatValue(mmind::eye::scanning
2d_setting::ExposureTime::nam
e, scan2DExposureTime);
```

C#

```
camera.CurrentUserSet().SetFl
oatValue(MMind.Eye.Scanning2D
Setting.ExposureTime.Name,
true);

var scan2DExposureTime = new
    double();
camera.CurrentUserSet().GetFl
oatValue(MMind.Eye.Scanning2D
Setting.ExposureTime.Name,
ref scan2DExposureTime);
```

Python

```
camera.current_user_set().set
_float_value(Scanning2DExposu
reTime.name, 100)

error, scan_2d_exposure_time
= camera.current_user_set
().get_float_value(Scanning2D
ExposureTime.name)
```

_Bool Type

The **Tone Mapping** parameter in the **2D Parameters** category is used as an example.

2.1.0

C++

```
device.setScan2DToneMappingEnable(true);

bool toneMappingEnable = false;
device.getScan2DToneMappingEnable(toneMappingEnable);
```

C#

```
device.SetScan2DToneMappingEnable(true);

var toneMappingEnable = new bool();
device.GetScan2DToneMappingEnable(ref toneMappingEnable);
```

Python

```
device.set_scan_2d_tone_mapping_enable(True)

tone_mapping_enable = device.get_scan_2d_tone_mapping_enable()
```

2.2.0

C++

```
#include "area_scan_3d_camera/parameters/Scanning2D.h"

camera.currentUserSet().setBoolValue(mmind::eye::scanning2d_setting::ToneMappingEnable::name, true);
```

```
bool toneMappingEnable = false;
camera.currentUserSet().getBoolValue(mmind::eye::scanning2d_setting::ToneMappingEnable::name, toneMappingEnable);
```

C#

```
camera.CurrentUserSet().SetBoolValue(MMind.Eye.Scanning2DSetting.ToneMappingEnable.Name, true);
```

```
var toneMappingEnable = new bool();
camera.CurrentUserSet().GetBoolValue(MMind.Eye.Scanning2DSetting.ToneMappingEnable.Name, ref toneMappingEnable);
```

Python

```
camera.current_user_set().set_bool_value(Scanning2DToneMappingEnable.name, True)

error, tone_mapping_enable = camera.current_user_set().get_bool_value(Scanning2DToneMappingEnable.name)
```

_Enum Type

The **Surface Smoothing** parameter in the **Point Cloud Processing** category is used as an example.

2.1.0

C++

```
device.setCloudSurfaceSmoothingMode(mmind::api::PointCloudProcessingSettings::PointCloudSurfaceSmoothing::Normal);

mmind::api::PointCloudProcessingSettings::PointCloudSurfaceSmoothing surfaceSmoothing;
device.getCloudSurfaceSmoothingMode(surfaceSmoothing);
```

C#

```
device.SetCloudSurfaceSmoothingMode(PointCloudSurfaceSmoothing.Normal);

var surfaceSmoothingMode =
    new
    PointCloudSurfaceSmoothing();
device.GetCloudSurfaceSmoothingMode(ref
    surfaceSmoothingMode);
```

Python

```
device.set_cloud_surface_smoothing_mode("Normal")
cloud_surface_smoothing_mode
= device
.get_cloud_surface_smoothing_mode()
```

2.2.0

C++

```
#include
"area_scan_3d_camera/parameters/PointCloudProcessing.h"

camera.currentUserSet().setEnumValue(mmind::eye::pointcloud_processing_setting::SurfaceSmoothing::name,
    static_cast<int>
    (mmind::eye::pointcloud_processing_setting::SurfaceSmoothing::Value::Normal));
```

```
int surfaceSmoothing = 0;
camera.currentUserSet().currentUserSet.getEnumValue(mmind::eye::pointcloud_processing_setting::SurfaceSmoothing::name, surfaceSmoothing);
```

C#

```
camera.currentUserSet().SetEnumValue(MMind.Eye.PointCloudProcessingSetting.SurfaceSmoothing.Name, (int)MMind.Eye.PointCloudProcessingSetting.SurfaceSmoothing.Value.Normal);
```

```
var surfaceSmoothingMode =
    new int();
camera.currentUserSet().GetEnumValue(MMind.Eye.PointCloudProcessingSetting.SurfaceSmoothing.Name, ref
    surfaceSmoothingMode);
```

Python

```
camera.current_user_set().set_enum_value(PointCloudSurfaceSmoothing.name,
```


_Roi Type

The ROI parameter is used as an example.

2.1.0	2.2.0
<p>C++</p> <pre>device.setScan3DROI(mmind::api::ROI(0, 0, 500, 500)); mmind::api::ROI roi; device.getScan3DROI(roi);</pre> <p>C#</p> <pre>device.SetScan3DROI(new ROI(0, 0, 500, 500)); var roi = new ROI(); device.GetScan3DROI(ref roi);</pre> <p>Python</p> <pre>device.set_scan_3d_roi(0, 0, 500, 500) roi = device.get_scan_3d_roi()</pre>	<p>C++</p> <pre>#include "area_scan_3d_camera/parameters/Scanning3D.h" camera.currentUserSet().setRoiValue(mmind::eye::scanning3d_setting::ROI::name, mmind::eye::ROI(0, 0, 500, 500)); mmind::eye::ROI roi; camera.currentUserSet().getRoiValue(mmind::eye::scanning3d_setting::ROI::name, roi);</pre> <p>C#</p> <pre>camera.CurrentUserSet().SetRoiValue(MMind.Eye.Scanning3DSetting.ROI.Name, new ROI(0, 0, 500, 500)); var roi = new ROI(); camera.CurrentUserSet().GetRoiValue(MMind.Eye.Scanning3DSetting.ROI.Name, ref roi);</pre> <p>Python</p> <pre>camera.current_user_set().set_roi_value(Scanning3DROI.name, ROI(0, 0, 500, 500)) error, roi = camera.current_user_set().get_roi_value(Scanning3DROI.name)</pre>

_Range Type

The **Depth Range** parameter is used as an example.

2.1.0

C++

```
device.setDepthRange(mmind::api::DepthRange(100, 2000));

mmind::api::DepthRange
depthRange;
device.getDepthRange(depthRange);
```

C#

```
device.SetDepthRange(new
DepthRange(100, 2000));

val depthRange = new
DepthRange();
device.GetDepthRange(ref
depthRange);
```

Python

```
device.set_depth_range(100,
2000)

depth_range = device
.get_depth_range()
```

2.2.0

C++

```
#include
"area_scan_3d_camera/parameters/Scanning3D.h"

camera.currentUserSet().setRangeValue(mmind::eye::scanning3d_setting::DepthRange::name, mmind::eye::Range<int>{100, 2000});

mmind::eye::Range<int>
rangeValue;
camera.currentUserSet().getRangeValue(mmind::eye::scanning3d_setting::DepthRange::name, rangeValue);
```

C#

```
camera.CurrentUserSet().SetRangeValue(MMind.Eye.Scanning3DSetting.DepthRange.Name, new
IntRange(100, 2000));

var range = new IntRange();
camera.CurrentUserSet().GetRangeValue(MMind.Eye.Scanning3DSetting.DepthRange.Name, ref
range);
```

Python

```
camera.current_user_set().set_range_value(Scanning3DDepthRange.name, RangeInt(100, 1000))

error, range = camera
.current_user_set().get_range_value(Scanning3DDepthRange.name)
```

`_FloatArray` Type

The `Exposure Time` parameters in the `3D Parameters` category is used as an example.

2.1.0

C++

```
device.setScan3DExposure(std:
:vector<double>{5, 10});

std::vector<double>
exposureSequence;
device.getScan3DExposure(expo
sureSequence);
```

C#

```
device.SetScan3DExposure(new
List<double> { 5, 10 });

var exposureSequence = new
List<double>();
device.GetScan3DExposure(ref
exposureSequence);
```

Python

```
device.set_scan_3d_exposure([
5.0, 10.0])

exposure_sequence = device
.get_scan_3d_exposure()
```

2.2.0

C++

```
#include
"area_scan_3d_camera/paramete
rs/Scanning3D.h"

camera.currentUserSet().setFl
oatArrayValue(mmind::eye::sca
nning3d_setting::ExposureSequ
ence::name, std::vector
<double>{5, 10});

std::vector<double>
exposureSequence;
camera.currentUserSet().getFl
oatArrayValue(mmind::eye::sca
nning3d_setting::ExposureSequ
ence::name,
exposureSequence);
```

C#

```
camera.CurrentUserSet().SetFl
oatArrayValue(MMind.Eye.Scann
ing3DSetting.ExposureSequ
equence.Name, new List<double> { 5,
10 });

var exposureSequence = new
List<double>();
camera.CurrentUserSet().GetFl
oatArrayValue(MMind.Eye.Scann
ing3DSetting.ExposureSequ
equence.Name, ref exposureSequence);
```

Python

```
camera.current_user_set().set
_float_array_value(Scanning3D
ExposureSequence.name, [5,
10])

error, exposure_sequence =
camera.current_user_set().get
_float_array_value(Scanning3D
```

Acquire data

In Mech-Eye API 2.2.0, the following changes have been made to the statements for acquiring data:

Acquire 2D Image

2.1.0	2.2.0
<p>C++</p> <pre data-bbox="272 555 756 725">mmind::api::ColorMap color; device.captureColorMap(color) ;</pre> <p>C#</p> <pre data-bbox="272 801 756 972">var color = new ColorMap(); device.CaptureColorMap(ref color);</pre> <p>Python</p> <pre data-bbox="272 1048 756 1173">color_map = device .capture_color()</pre>	<p>C++</p> <pre data-bbox="879 555 1362 801">mmind::eye::Frame2D frame2D; camera.capture2D(frame2D); mmind::eye::Color2DImage color = frame3D. getColorImage();</pre> <p>C#</p> <pre data-bbox="879 878 1362 1084">var frame = new Frame2D(); camera.Capture2D(ref frame); var color = frame .GetColorImage();</pre> <p>Python</p> <pre data-bbox="879 1160 1362 1366">frame_2d = Frame2D() camera.capture_2d(frame_2d) color = frame. get_color_image()</pre>

Acquire Depth Map

2.1.0	2.2.0
<p>C++</p> <pre>mmind::api::DepthMap depth; device.captureDepthMap(depth) ;</pre> <p>C#</p> <pre>var depth = new DepthMap(); device.CaptureDepthMap(ref depth);</pre> <p>Python</p> <pre>depth_map = device .capture_depth()</pre>	<p>C++</p> <pre>mmind::eye::Frame3D frame3D; camera.capture3D(frame3D); mmind::eye::DepthMap depth = frame3D.getDepthMap();</pre> <p>C#</p> <pre>var frame = new Frame3D(); camera.Capture3D(ref frame); var depth = frame. GetDepthMap();</pre> <p>Python</p> <pre>frame_3d = Frame3D() camera.capture_3d(frame_3d) depth = frame_3d .get_depth_map()</pre>

Acquire Untextured Point Cloud

2.1.0	2.2.0
<p>C++</p> <pre data-bbox="272 327 756 533">mmind::api::PointXYZMap pointXYZMap; device.capturePointXYZMap(poi ntXYZMap);</pre> <p>C#</p> <pre data-bbox="272 607 756 813">var pointXYZMap = new PointXYZMap(); device.CapturePointXYZMap(ref pointXYZMap);</pre> <p>Python</p> <pre data-bbox="272 891 756 1021">points_xyz = device .capture_point_xyz()</pre>	<p>C++</p> <pre data-bbox="879 327 1362 568">mmind::eye::Frame3D frame3D; camera.capture3D(frame3D); mmind::eye::PointCloud cloud = frame3D .getUntexturedPointCloud();</pre> <p>C#</p> <pre data-bbox="879 645 1362 851">var frame = new Frame3D(); camera.Capture3D(ref frame); var cloud = frame .GetUntexturedPointCloud();</pre> <p>Python</p> <pre data-bbox="879 927 1362 1133">frame_3d = Frame3D() camera.capture_3d(frame_3d) cloud = frame_3d .get_untextured_point_cloud()</pre>

Acquire Textured Point Cloud

2.1.0	2.2.0
<p>C++</p> <pre data-bbox="272 327 756 533">mmind::api::PointXYZBGRMap pointXYZBGRMap; device.capturePointXYZBGRMap(pointXYZBGRMap);</pre> <p>C#</p> <pre data-bbox="272 609 756 815">var pointXYZBGRMap = new PointXYZBGRMap(); device.CapturePointXYZBGRMap(ref pointXYZBGRMap)</pre> <p>Python</p> <pre data-bbox="272 891 756 1021">points_xyz_bgr = device .capture_point_xyz_bgr()</pre>	<p>C++</p> <pre data-bbox="880 327 1364 645">mmind::eye::Frame2DAnd3D frame2DAnd3D; camera.capture2DAnd3D(frame2D And3D); mmind::eye::TexturedPointClou d cloud = frame2DAnd3D .getTexturedPointCloud();</pre> <p>C#</p> <pre data-bbox="880 721 1364 1003">var frame = new Frame2DAnd3D(); camera.Capture2DAnd3D(ref frame); var cloud = frame .GetTexturedPointCloud();</pre> <p>Python</p> <pre data-bbox="880 1079 1364 1326">frame = Frame2DAnd3D() camera.capture_2d_and_3d(fram e) cloud = frame .get_textured_point_cloud()</pre>

Obtain Camera Information and Properties

Mech-Eye API 2.2.0 uses classes to organize the information and properties of the camera. For details, please refer to [Functional Hierarchy of Classes](#).

Obtain Camera Information

2.1.0	2.2.0
<p>C++</p> <pre> mmind::api::MechEyeDeviceInfo deviceInfo; device.getDeviceInfo(deviceIn fo); std::string model = deviceInfo.model; std::string serialNumber = deviceInfo.id; std::string firmwareVersion = deviceInfo.firmwareVersion; </pre> <p>C#</p> <pre> var deviceInfo = new MechEyeDeviceInfo(); device.GetDeviceInfo(ref deviceInfo); var model = deviceInfo.model; var serialNumber = deviceInfo.id; var firmwareVersion = deviceInfo.firmwareVersion; </pre> <p>Python</p> <pre> device_info = device .get_device_info() </pre>	<p>C++</p> <pre> mmind::eye::CameraInfo cameraInfo; camera.getCameraInfo(cameraIn fo); std::string model = cameraInfo.model; std::string serialNumber = cameraInfo.serialNumber; std::string firmwareVersion = cameraInfo.firmwareVersion.to String(); </pre> <p>C#</p> <pre> var cameraInfo = new CameraInfo(); camera.GetCameraInfo(ref cameraInfo); var model = cameraInfo.Model; var serialNumber = cameraInfo.SerialNumber; var firmwareVersion = cameraInfo.FirmwareVersion.To String(); </pre> <p>Python</p> <pre> camera_info = CameraInfo() camera.get_camera_info(camera _info) </pre>

Obtain Camera Temperatures

2.1.0

C++

```

mmind::api::DeviceTemperature
devicetemperature;
device.getDeviceTemperature(d
evicetemperature);
    
```

```

float cpu =
devicetemperature.cpuTemperat
ure;
float projector =
devicetemperature.projectorMo
duleTemperature;
    
```

C#

```

var temperature = new
DeviceTemperature();
device.GetDeviceTemperature(r
ef temperature);
    
```

```

var cpu = temperature.cpu;
var projector = temperature
.projectorModule;
    
```

Python

```

device_temperature = device
.get_device_temperature()

cpu = device_temperature
.cpu_temperature()
projector =
device_temperature.projector_
module_temperature()
    
```

2.2.0

C++

```

mmind::eye::CameraStatus
cameraStatus;
camera.getCameraStatus(camera
Status);
    
```

```

float cpu = cameraStatus
.temperature.cpuTemperature;
float projector =
cameraStatus.temperature.proj
ectorTemperature;
    
```

C#

```

var cameraStatus = new
CameraStatus();
camera.GetCameraStatus(ref
cameraStatus);
    
```

```

var cpu = cameraStatus
.Temperature.CpuTemperature;
var projector = cameraStatus
.Temperature.ProjectorModuleT
emperature;
    
```

Python

```

camera_status =
CameraStatus()
camera.get_camera_status(came
ra_status)

cpu = camera_status
.temperature.cpu_temperature
projector = camera_status
.temperature.projector_module
_temperature
    
```

Obtain Image Resolutions

2.1.0

C++

```

mmind::api::DeviceResolution
deviceResolution;
device.getDeviceResolution(de
viceResolution);

unsigned int textureWidth =
deviceResolution.colorMapWidt
h;
unsigned int textureHeight =
deviceResolution.colorMapHeig
ht;
unsigned int depthWidth =
deviceResolution.depthMapWidt
h;
unsigned int depthHeight =
deviceResolution.depthMapHeig
ht;
    
```

C#

```

var deviceResolution = new
DeviceResolution();
device.GetDeviceResolution(re
f deviceResolution);

var textureWidth =
deviceResolution.colorMapWidt
h;
var textureHeight =
deviceResolution.colorMapHeig
ht;
var depthWidth =
deviceResolution.depthMapWidt
h;
var depthHeight =
deviceResolution.depthMapHeig
ht;
    
```

Python

```

device_resolution = device
.get_device_resolution()
    
```

2.2.0

C++

```

mmind::eye::CameraResolutions
cameraResolutions;
camera.getCameraResolutions(c
ameraResolutions);

unsigned int textureWidth =
cameraResolutions.texture.wid
th;
unsigned int textureHeight =
cameraResolutions.texture.hei
ght;
unsigned int depthWidth =
cameraResolutions.depth.width
;
unsigned int depthHeight =
cameraResolutions.depth.heigh
t;
    
```

C#

```

var cameraResolutions = new
CameraResolutions();
camera.GetCameraResolutions(r
ef cameraResolutions);

var textureWidth =
cameraResolutions.Texture.Wid
th;
var textureHeight =
cameraResolutions.Texture.Hei
ght;
var depthWidth =
cameraResolutions.Depth.Width
;
var depthHeight =
cameraResolutions.Depth.Heigh
t;
    
```

Python

```

camera_resolutions =
CameraResolutions()
camera.get_camera_resolutions
    
```

Obtain Camera Intrinsic Parameters

2.1.0

C++

```

mmind::api::DeviceIntri
deviceIntri;
device.getDeviceIntri(deviceI
ntri);

const mmind::eye::CameraIntri
textureIntri = deviceIntri
.textureCameraIntri;

const double fx =
textureIntri.cameraMatrix[0];
const double fy =
textureIntri.cameraMatrix[1];
const double cx =
textureIntri.cameraMatrix[2];
const double cy =
textureIntri.cameraMatrix[3];

const double k1 =
textureIntri.distortion[0];
const double k2 =
textureIntri.distortion[1];
const double p1 =
textureIntri.distortion[2];
const double p2 =
textureIntri.distortion[3];
const double k3 =
textureIntri.distortion[4];
    
```

C#

```

var deviceIntri = new
DeviceIntri();
device.GetDeviceIntri(ref
deviceIntri);

var textureCameraIntri =
deviceIntri.textureCameraIntri;

const double fx =
textureCameraIntri.fx;
const double fy =
    
```

2.2.0

C++

```

mmind::eye::CameraIntrinsics
intrinsics;
camera.getCameraIntrinsics(in
trinsics);

const mmind::eye
::Intrinsics2dCamera
textureIntrinsics =
intrinsics.texture;
const mmind::eye
::CameraMatrix cameraMatrix =
textureIntrinsics.cameraMatri
x;
const double fx =
cameraMatrix.fx;
const double fy =
cameraMatrix.fy;
const double cx =
cameraMatrix.cx;
const double cy =
cameraMatrix.cy;

const mmind::eye
::CameraDistortion distCoeffs
= intrinsics.texture
.cameraDistortion;
const double k1 = distCoeffs
.k1;
const double k2 = distCoeffs
.k2;
const double p1 = distCoeffs
.p1;
const double p2 = distCoeffs
.p2;
const double k3 = distCoeffs
.k3;
    
```

C#

```

var intrinsics = new
CameraIntrinsics();
camera.GetCameraIntrinsics(re
f intrinsics);
    
```

New Functions

Multiple convenient new functions have been added to Mech-Eye API 2.2.0.



If you want to use the following new functions, please modify your client program according to this topic to use the 2.2.0 version of Mech-Eye API.

Compute Point Cloud Normals

By calling the following methods you can now obtain 3D data containing normals directly, reducing the workload of subsequent data processing.

Obtain Untextured Point Cloud with Normals

C++

```
mmind::eye::Frame3D frame3D;  
camera.capture3DWithNormal(frame3D);  
  
PointCloudWithNormals pointCloud =  
frame3D.getUntexturedPointCloudWithNormals();
```

C#

```
var frame3D = new Frame3D();  
camera.Capture3DWithNormal(ref frame3D);  
  
var pointCloud = frame3D.GetUntexturedPointCloudWithNormals();
```

Python

```
frame_3d = Frame3D()  
camera.capture_3d_with_normal(frame_3d)  
  
point_cloud = frame_3d.get_untextured_point_cloud_with_normals()
```

Obtain Textured Point Cloud with Normals

C++

```
mmind::eye::Frame2DAnd3D frame2DAnd3D;  
camera.capture2DAnd3DWithNormal(frame2DAnd3D);  
  
TexturedPointCloudWithNormals pointCloud =  
frame2DAnd3D.getTexturedPointCloudWithNormals();
```

C#

```
var frame2DAnd3D = new Frame2DAnd3D();
camera.Capture2DAnd3DWithNormal(ref frame2DAnd3D);

var pointCloud = frame2DAnd3D.GetTexturedPointCloudWithNormals();
```

Python

```
frame_2d_and_3d = Frame2DAnd3D()
camera.capture_2d_and_3d_with_normal(frame_2d_and_3d)

point_cloud = frame_2d_and_3d.get_textured_point_cloud_with_normals()
```

Save Point Cloud

By calling the following methods, you can now save the point cloud in PLY, PCD or CSV format directly, without the need to rely on any third-party software libraries.

Save Untextured Point Cloud

C++

```
mmind::eye::Frame3D frame3D;
camera.capture3D(frame3D);

frame3D.saveUntexturedPointCloud(mmind::eye::FileFormat::PLY,
"PointCloud.ply");
```

C#

```
var frame3D = new frame3D();
camera.Capture3D(ref frame3D);

frame3D.SaveUntexturedPointCloud(FileFormat.PLY, "PointCloud.ply");
```

Python

```
frame_3d = Frame3D()
camera.capture_3d(frame_3d)

frame_3d.save_untextured_point_cloud(FileFormat_PLY, "PointCloud.ply")
```

Save Textured Point Cloud

C++

```
mmind::eye::Frame2DAnd3D frame2DAnd3D;  
camera.capture2DAnd3D(frame2DAnd3D);  
  
frame2DAnd3D.saveTexturedPointCloud(mmind::eye::FileFormat::PLY,  
"TexturedPointCloud.ply");
```

C#

```
var frame2DAnd3D = new Frame2DAnd3D();  
camera.Capture2DAnd3D(ref frame2DAnd3D);  
  
frame2DAnd3D.SaveTexturedPointCloud(FileFormat.PLY,  
"TexturedPointCloud.ply");
```

Python

```
frame_2d_and_3d = Frame2DAnd3D()  
camera.capture_2d_and_3d(frame_2d_and_3d)  
  
frame_2d_and_3d.save_textured_point_cloud(FileFormat_PLY,  
"TexturedPointCloud.ply")
```

Import and Export Parameter Groups

By calling the following methods, you can now import and replace all parameter groups from a JSON file, or save all parameter groups to a JSON file.

Import parameter groups

C++

```
camera.userSetManager().loadFromFile("camera_config.json");
```

C#

```
camera.UserSetManager().LoadFromFile("camera_config.json");
```

Python

```
camera.user_set_manager().load_from_file("camera_config.json")
```

C++

```
camera.userSetManager().saveToFile("camera_config.json");
```

C#

```
camera.UserSetManager().SaveToFile("camera_config.json");
```

Python

```
camera.user_set_manager().save_to_file("camera_config.json")
```

Check Camera Connection Status

The new heartbeat mechanism can be used to detect device disconnections.

Set Heartbeat Interval

Use the following method to set the interval of the heartbeat signals:

C++

```
camera.setHeartbeatInterval(1000);
```

C#

```
camera.SetHeartbeatInterval(1000);
```

Python

```
camera.set_heartbeat_interval(1000)
```

Register Callback Function

Use the following method to register the callback function that automatically detects a camera disconnection and reports error:

C++

```
mmind::eye::CameraEvent::EventCallback callback =  
[](mmind::eye::CameraEvent::Event event, void* pUser) {  
    std::cout << "A camera event has occurred. The event ID is " <<  
event << "." << std::endl;
```

```
};

mmind::eye::CameraEvent::registerCameraEventCallback(camera, callback,
nullptr, mmind::eye::CameraEvent::CAMERA_EVENT_DISCONNECTED);
```

C#

```
private static void CallbackFunc(CameraEvent.Event cameraEvent, IntPtr
pUser)
{
    Console.WriteLine("A camera event has occurred. The event ID is
{0}.", cameraEvent);
}

Utils.ShowError(CameraEvent.RegisterCameraEventCallback(ref camera,
CallbackFunc, IntPtr.Zero,
(uint)CameraEvent.Event.CAMERA_EVENT_DISCONNECTED));
```

Python

```
class CustomCallback(EventCallbackBase):
    def __init__(self):
        super().__init__()

    def run(self, event):
        print("A camera event has occurred. The event ID is {0}.".
format(event))

camera_event = CameraEvent()
callback = CustomCallback()
show_error(camera_event.register_camera_event_callback(camera, callback,
CameraEvent.CAMERA_EVENT_ALL))
```

6.5. Mech-Eye API Reference

Thank you for using Mech-Eye API! Please refer to [Mech-Eye API Reference](#) for detailed information.

6.6. Sample Usage Guide

This topic provides the instructions on using the samples for different platforms and languages, as well as the samples of third-party software calling Mech-Eye API.

- [C++ \(Windows\)](#)
- [C++ \(Ubuntu\)](#)

- [C#](#)
- [Python \(Windows\)](#)
- [Python \(Ubuntu\)](#)
- [VisionPro](#)

6.6.1. C++ (Windows)

This topic introduces how to configure the Mech-Eye API C++ samples with CMake and then build the samples with Visual Studio on Windows.

Sample List

Samples are divided into the following categories: **Basic**, **Advanced**, **Util**, **Calibration**, **Pcl** and **Halcon**.

- **Basic** samples: Connect and acquire data.
- **Advanced** samples: Acquire data in more complicated manners and set model-specific parameters.
- **Util** samples: Obtain camera information and set common parameters.
- **Calibration** samples: perform hand-eye calibration through Mech-Eye API.
- **Pcl** samples: Use the PCL library to convert data structure and visualize data.
- **Halcon**: obtain HALCON-readable point clouds through Mech-Eye API.

The samples in each category and their descriptions are provided below.

▼ Basic

- **ConnectToCamera**: Connect to a camera.
- **ConnectAndCaptureImages**: Connect to a camera and obtain the 2D image, depth map, and point cloud data.
- **Capture2DImage**: Obtain and save the 2D image.
- **CaptureDepthMap**: Obtain and save the depth map.
- **CapturePointCloud**: Obtain and save the untextured and textured point clouds.
- **CapturePointCloudHDR**: Set multiple exposure times, and then obtain and save the point cloud.
- **CapturePointCloudWithNormals**: Calculate normals and save the point cloud with normals.

▼ Advanced

- **ConvertDepthMapToPointCloud**: Generate a point cloud from the depth map and save the point cloud.
- **MultipleCamerasCaptureSequentially**: Obtain and save 2D images, depth maps, and point clouds sequentially from multiple cameras.
- **MultipleCamerasCaptureSimultaneously**: Obtain and save 2D images, depth maps, and point clouds simultaneously from multiple cameras.
- **CapturePeriodically**: Obtain and save 2D images, depth maps, and point clouds periodically for the specified duration from a camera.
- **Mapping2DImageToDepthMap**: Generate untextured and textured point clouds from a

masked 2D image and a depth map.

- **SetParametersOfLaserCameras:** Set the parameters specific to laser cameras (the DEEP and LSR series).
- **SetParametersOfUHPCameras:** Set the parameters specific to the UHP series.
- **RegisterCameraEvent:** Define and register the callback function for monitoring the camera connection status.
- **CaptureStereo2DImages:** Obtain the 2D images from both 2D cameras of Deep (V3), Laser L Enhanced (V3), PRO XS (V4), LSR L (V4), LSR S (V4), and DEEP (V4).

▼ Util

- **GetCameraIntrinsics:** Obtain and print the camera intrinsic parameters.
- **PrintCameraInfo:** Obtain and print the camera information, such as model, serial number, firmware version, and temperatures.
- **SetScanningParameters:** Set the parameters in the **3D Parameters**, **2D Parameters**, and **ROI** categories.
- **SetDepthRange:** Set the **Depth Range** parameter.
- **SetPointCloudProcessingParameters:** Set the **Point Cloud Processing** parameters.
- **ManageUserSets:** Manage parameter groups, such as obtaining the names of all parameter groups, adding a parameter group, switching the parameter group, and saving parameter settings to the parameter group.
- **SaveAndLoadUserSet:** Import and replace all parameter groups from a JSON file, and save all parameter groups to a JSON file.

▼ Calibration

- **HandEyeCalibration:** Perform hand-eye calibration.

▼ Pcl

- **ConvertPointCloudToPcl:** Obtain the point cloud data from the camera and convert it to the PCL data structure.
- **ConvertPointCloudWithNormalsToPcl:** Obtain the point cloud data with normals from the camera and convert it to the PCL data structure.

▼ Halcon

- **ConvertPointCloudToObjectModel3D:** Obtain the point cloud data from a camera, and then transform and save the point clouds using the HALCON C++ interface.

Prerequisites

In order to use the C++ samples of Mech-Eye API, the following prerequisites must be satisfied:

- [Connect the camera and computer correctly.](#)
- Obtain the samples.
- Install the required software.
- (Optional) Install the software libraries on which samples depend and add relevant environment variables.

Obtain Samples

C++ samples are included in the installation path of Mech-Eye SDK or can be obtained from [GitHub](#) by cloning. The samples in the installation path are the version at the time of Mech-Eye SDK release. The version on GitHub may contain the latest changes.

- The samples in the installation path are located in `xxx/Mech-Eye SDK-2.2.2/API/samples/cpp/area_scan_3d_camera`.
- The samples cloned from GitHub are located in `xxx/mecheye_cpp_samples/area_scan_3d_camera`.

Install Required Software

In order to use the C++ samples of Mech-Eye API, Mech-Eye SDK, CMake and Visual Studio must be installed.

Install the Latest Version of Mech-Eye SDK

Please install or upgrade Mech-Eye SDK according to [Mech-Eye SDK Installation Guide](#).

Install CMake (Version 3.2 or Above)

1. Download [CMake](#): download the installer to the right of **Windows x64 Installer**.

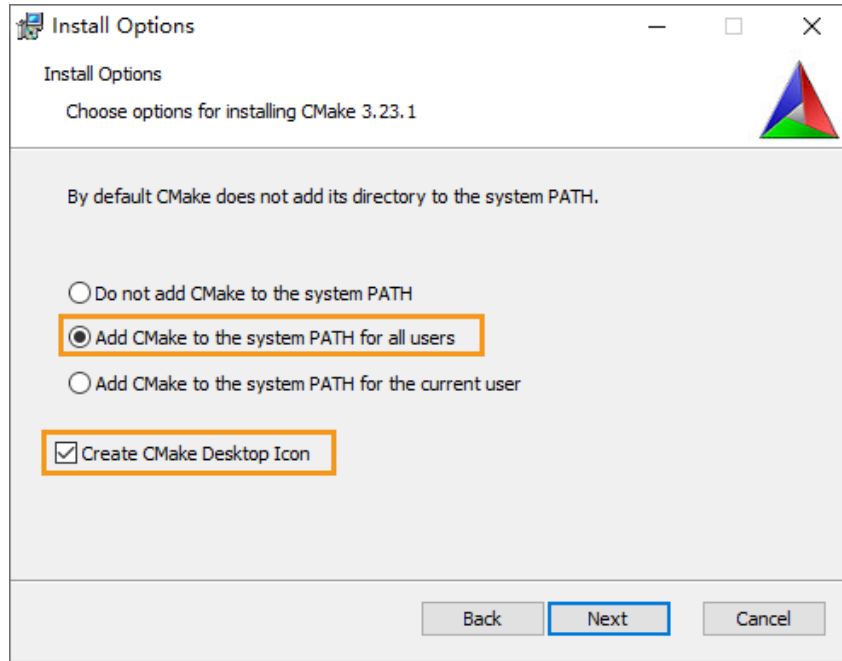
Source distributions:

Platform	Files
Unix/Linux Source (has \n line feeds)	cmake-3.23.1.tar.gz
Windows Source (has \r\n line feeds)	cmake-3.23.1.zip

Binary distributions:

Platform	Files
Windows x64 Installer	cmake-3.23.1-windows-x86_64.msi
Windows x64 ZIP	cmake-3.23.1-windows-x86_64.zip
Windows i386 Installer	cmake-3.23.1-windows-i386.msi
Windows i386 ZIP	cmake-3.23.1-windows-i386.zip

2. When installing, select the following two options to add CMake to environment variables and create a desktop shortcut for CMake.
 - Add CMake to the system PATH for all users
 - Create Cmake Desktop Icon



Install Visual Studio (Version 2017 or Above)

1. Download the [Visual Studio installer](#).
2. When installing, select the following two workloads in the **Desktop & Mobile** category, and then click [**Install**].
 - Desktop development with C++
 - Universal Windows Platform development



Please restart your computer after installing Visual Studio. If you still need to install the optional software libraries, you can restart your computer after [adding environment variables](#).

(Optional) Install Dependent Software Libraries

Some samples contain functions that depend on the following third-party software libraries. If you need to use the samples listed in the table below, you must install the software libraries on which the samples depend.

- OpenCV: version 3.4.5 or above
- PCL: version 1.8.1 or above
- HALCON: version 20.11 or above



HALCON versions below 20.11 are not fully tested.



If any of the optional software libraries is not installed, then you must complete the steps in [Disable Unneeded Samples \(Optional\)](#) before configuring the samples.

Refer to the following table for the samples that depend on third-party software libraries.

Sample	OpenCV	PCL	Halcon
Capture2DImage	√		
CaptureDepthMap	√		

Sample	OpenCV	PCL	Halcon
CapturePeriodically	√		
CaptureStereo2DImages	√		
ConvertPointCloudToObjectModel3D			√
ConvertPointCloudToPCL		√	
ConvertPointCloudWithNormalsToPCL		√	
HandEyeCalibration	√		
MultipleCamerasCaptureSequentially	√		
MultipleCamerasCaptureSimultaneously	√		

Install OpenCV

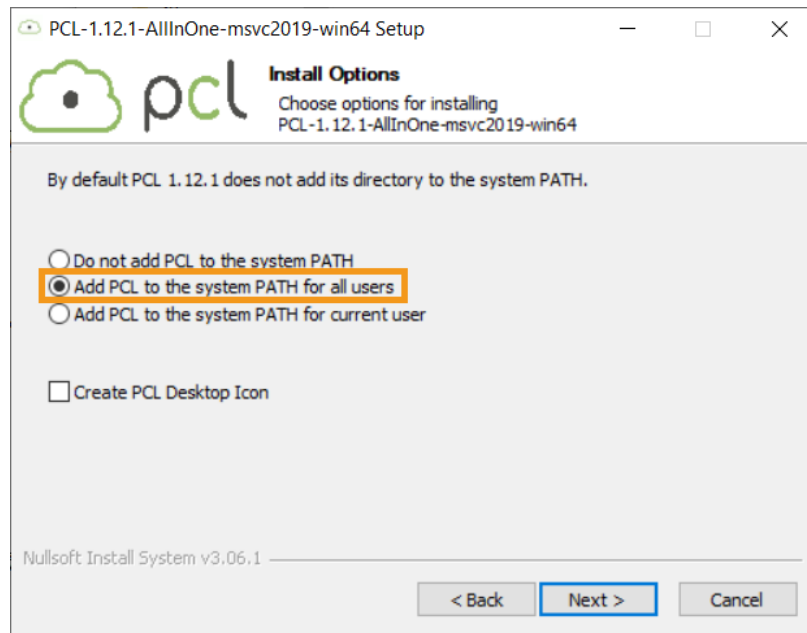
1. [Download and install OpenCV](#). Note down the installation path of OpenCV.
2. Add OpenCV to environment variables. Refer to [Add Environment Variables](#) for instructions.

Install PCL

1. Refer to the following table and determine the version of PCL to install based on the version of Visual Studio.

Visual Studio version	Supported PCL versions
2017	1.8.1–1.9.1
2019	1.8.1–1.12.1
2022	1.8.1 and above

2. Open the [PCL download page](#), click on **Assets** under the corresponding version, and download the EXE installer.
3. Run the installer to install PCL. When installing, select **Add PCL to the system PATH for all users** to add PCL to environment variables.



The following error message might pop up during installation:



Warning! PATH too long installer unable to modify PATH!

In this case, you need to add PCL to environment variables manually. Refer to [Add Environment Variables](#) for instructions.

4. Add OpenNI2 on which PCL depends to environment variables. Refer to [Add Environment Variables](#) for instructions.

Install HALCON

Install [HALCON](#) 20.11 or above. Please confirm that your HALCON license is still valid.

Add Environment Variables

If you need to use samples that depend on OpenCV or PCL, please add the relevant environment variables according to the following steps.

1. Right-click **This PC** on the desktop and select **Properties**.
2. Click **Advanced system settings**, and in the pop-up **System Properties** window, click **[Environment Variables]** to open the **Environment Variables** window.
3. Double-click **Path** in the user variables to enter the **Edit environment variable** page. Click **[New]** in the upper-right and add the following paths one by one. Then, click **[OK]** to exit the window.
 - For PCL: *C:/Program Files/OpenNI2/Tools*
 - For OpenCV:
 - *xxx/OpenCV/build/x64/vc14/bin*
 - *xxx/OpenCV/build/x64/vc14/lib*
 - If you encountered the error message while installing PCL, please add the following path as well:
 - *xxx/PCL 1.12.1/bin*

- `xxx/PCL 1.12.1/3rdParty/VTK/bin`



It is recommended to restart your computer after adding the environmental variables. If not restarted, the environmental variables added may not be effective.

Build and Run Samples

You can build all samples at once or build a single sample individually.



The paths provided in this section correspond to the samples included in the installation path. If you are using the samples cloned from GitHub, please modify the paths correspondingly.

Disable Unneeded Samples (Optional)



If you did not install some of the optional software libraries and still want to build all samples at once, then you must complete the steps in this section. Otherwise, an error will occur during the configuration of samples using CMake.

If you do not need the samples that depend on OpenCV, PCL or HALCON, please edit the CMakeLists file to disable these samples before building all samples at once.

1. Open **CMakeLists.txt** in `xxx/Mech-Eye SDK-2.2.2/API/samples/cpp/area_scan_3d_camera`.
2. Edit lines containing **options**: find the lines starting with **options**, change **ON** at the end of the line to **OFF** to disable the samples that depend on the software library mentioned in that line.

Example: If you do not need to build the sample that depends on HALCON, change **ON** at the end of the **option(USE_HALCON...)** line to **OFF**.

3. Save the changes and close **CMakeLists.txt**.

Configure Samples in CMake

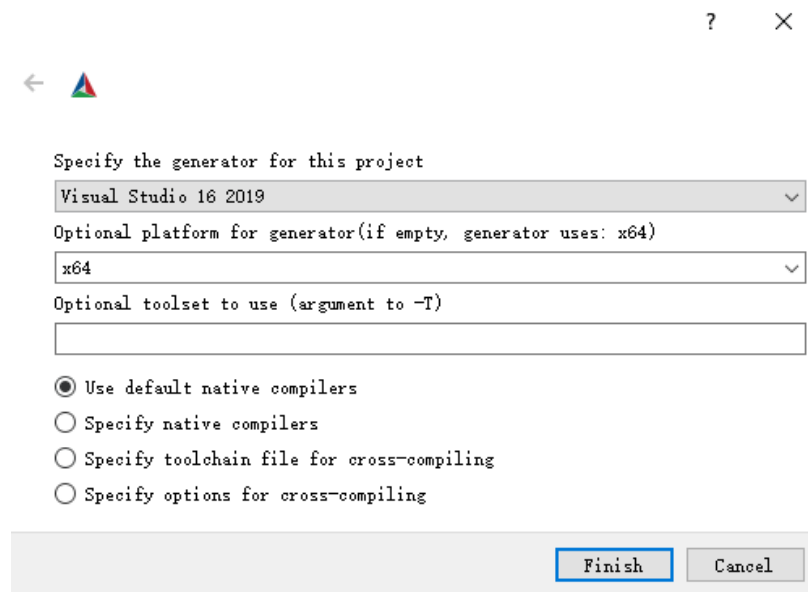
1. Right-click CMake and select **Run as administrator**.
2. Enter the paths of source code and binaries.
 - To build all samples at once, enter the following paths.

Where is the source code	<code>xxx/Mech-Eye SDK-2.2.2/API/samples/cpp/area_scan_3d_camera</code>
Where to build the binaries	<code>xxx/Mech-Eye SDK-2.2.2/API/samples/cpp/area_scan_3d_camera/build</code>

- To build one sample individually, enter the following paths. Replace **Category** with the category name of the sample and **SampleName** with the name of the sample.

Where is the source code	<code>xxx/Mech-Eye SDK-2.2.2/API/samples/cpp/area_scan_3d_camera/Category/SampleName</code>
Where to build the binaries	<code>xxx/Mech-Eye SDK-2.2.2/API/samples/cpp/area_scan_3d_camera/Category/SampleName/build</code>

- Click [**Configure**] in the bottom to open the configuration page. Select Visual Studio version, and set the platform to **x64**. Then, click [**Finish**]. When the configuration completes, the log will display **Configuring done**.



- Click [**Generate**] to generate Visual Studio solution. When the generation completes, the log will display **Generating done**. Then, click [**Open Project**] to open the solution in Visual Studio.

Build Samples in Visual Studio

- In Visual Studio toolbar, change the solution configuration from **Debug** to **Release**.
- In the menu bar, select Build > Build Solution. An executable file (.exe) is generated for each sample. The executable files are saved to the **Release** folder, located in the **Where to build the binaries** directory that you entered in CMake.

Run Samples

You can run the samples in Visual Studio after building them, or run the samples by double-clicking the executable files.

Run a Sample in Visual Studio

- In the **Solution Explorer** panel, right-click a sample, and select **Set as Startup Project**.
- Click [**Local Windows Debugger**] in the toolbar to run the sample.

While the sample is running, input the **index** of the camera to which you want to connect according to the instruction, and press **Enter** to connect to the camera.

- If image and/or point cloud files are saved by the sample, you can find the files in the **build** folder (the **Where to build the binaries** directory that you entered in CMake).

Run the Executable File of a Sample

- Navigate to the **Where to build the binaries** directory that you entered in CMake, and open the **Release** folder.

2. Run the executable file with the same name as the sample.

While the sample is running, input the `index` of the camera to which you want to connect according to the instruction, and press `Enter` to connect to the camera.

3. If image and/or point cloud files are saved by the sample, you can find the files in the **Release** folder.

6.6.2. C++ (Ubuntu)

This topic introduces how to configure the Mech-Eye API C++ samples with CMake and then build the samples with the make command on Ubuntu.

Sample List

Samples are divided into the following categories: **Basic**, **Advanced**, **Util**, **Calibration**, **Pcl** and **Halcon**.

- **Basic** samples: Connect and acquire data.
- **Advanced** samples: Acquire data in more complicated manners and set model-specific parameters.
- **Util** samples: Obtain camera information and set common parameters.
- **Calibration** samples: perform hand-eye calibration through Mech-Eye API.
- **Pcl** samples: Use the PCL library to convert data structure and visualize data.
- **Halcon**: obtain HALCON-readable point clouds through Mech-Eye API.

The samples in each category and their descriptions are provided below.

▼ Basic

- **ConnectToCamera**: Connect to a camera.
- **ConnectAndCaptureImages**: Connect to a camera and obtain the 2D image, depth map, and point cloud data.
- **Capture2DImage**: Obtain and save the 2D image.
- **CaptureDepthMap**: Obtain and save the depth map.
- **CapturePointCloud**: Obtain and save the untextured and textured point clouds.
- **CapturePointCloudHDR**: Set multiple exposure times, and then obtain and save the point cloud.
- **CapturePointCloudWithNormals**: Calculate normals and save the point cloud with normals.

▼ Advanced

- **ConvertDepthMapToPointCloud**: Generate a point cloud from the depth map and save the point cloud.
- **MultipleCamerasCaptureSequentially**: Obtain and save 2D images, depth maps, and point clouds sequentially from multiple cameras.
- **MultipleCamerasCaptureSimultaneously**: Obtain and save 2D images, depth maps, and point clouds simultaneously from multiple cameras.
- **CapturePeriodically**: Obtain and save 2D images, depth maps, and point clouds periodically for the specified duration from a camera.

- **Mapping2DImageToDepthMap**: Generate untextured and textured point clouds from a masked 2D image and a depth map.
- **SetParametersOfLaserCameras**: Set the parameters specific to laser cameras (the DEEP and LSR series).
- **SetParametersOfUHPCameras**: Set the parameters specific to the UHP series.
- **RegisterCameraEvent**: Define and register the callback function for monitoring the camera connection status.
- **CaptureStereo2DImages**: Obtain the 2D images from both 2D cameras of Deep (V3), Laser L Enhanced (V3), PRO XS (V4), LSR L (V4), LSR S (V4), and DEEP (V4).

▼ Util

- **GetCameraIntrinsics**: Obtain and print the camera intrinsic parameters.
- **PrintCameraInfo**: Obtain and print the camera information, such as model, serial number, firmware version, and temperatures.
- **SetScanningParameters**: Set the parameters in the **3D Parameters**, **2D Parameters**, and **ROI** categories.
- **SetDepthRange**: Set the **Depth Range** parameter.
- **SetPointCloudProcessingParameters**: Set the **Point Cloud Processing** parameters.
- **ManageUserSets**: Manage parameter groups, such as obtaining the names of all parameter groups, adding a parameter group, switching the parameter group, and saving parameter settings to the parameter group.
- **SaveAndLoadUserSet**: Import and replace all parameter groups from a JSON file, and save all parameter groups to a JSON file.

▼ Calibration

- **HandEyeCalibration**: Perform hand-eye calibration.

▼ Pcl

- **ConvertPointCloudToPcl**: Obtain the point cloud data from the camera and convert it to the PCL data structure.
- **ConvertPointCloudWithNormalsToPcl**: Obtain the point cloud data with normals from the camera and convert it to the PCL data structure.

▼ Halcon

- **ConvertPointCloudToObjectModel3D**: Obtain the point cloud data from a camera, and then transform and save the point clouds using the HALCON C++ interface.



This sample is not available on platforms based on the ARM64 architecture.

Prerequisites

In order to use the C++ samples of Mech-Eye API, the following prerequisites must be satisfied:

- [Connect the camera and computer correctly.](#)
- The version of the Ubuntu operating system is 18 or above.
- Obtain the samples.
- Install the required software.

- (Optional) Install software libraries on which the samples depend.



If using a Ubuntu virtual machine, please ensure that the available disk space is greater than 20 GB. Otherwise, installation of the software may fail.

Obtain Samples

C++ samples are included in the installation path of Mech-Eye SDK or can be obtained from [GitHub](#) by cloning. The samples in the installation path are the version at the time of Mech-Eye SDK release. The version on GitHub may contain the latest changes.

- The samples in the installation directory are located in `/opt/mech-mind/mech-eye-sdk/samples/cpp/area_scan_3d_camera`.
- The samples cloned from GitHub are located in `xxx/mecheye_cpp_samples/area_scan_3d_camera`.

Install Required Software

In order to use the C++ samples of Mech-Eye API, the software source list must be updated, and Mech-Eye SDK and dependencies must be installed.

1. Update software source list:

```
sudo apt-get update
```

2. Install dependencies:

```
sudo apt-get install -y build-essential pkg-config cmake
```

3. Install Mech-Eye SDK: Please refer to [Mech-Eye SDK Installation Guide \(Ubuntu\)](#).

(Optional) Install Dependent Software Libraries

Some samples contain functions that depend on the following third-party software libraries. If you need to use the samples listed in the table below, you must install the software libraries on which the samples depend.

- OpenCV: latest version recommended
- PCL: latest version recommended
- HALCON: version 20.11 or above



HALCON versions below 20.11 are not fully tested.



If any of the optional software libraries is not installed, then you must complete the steps in [Disable Unneeded Samples \(Optional\)](#) before configuring the samples.

Refer to the following table for the samples that depend on third-party software libraries.

Sample	OpenCV	PCL	Halcon
Capture2DImage	√		

Sample	OpenCV	PCL	Halcon
CaptureDepthMap	√		
CapturePeriodically	√		
CaptureStereo2DImages	√		
ConvertPointCloudToObjectModel3D			√
ConvertPointCloudToPCL		√	
ConvertPointCloudWithNormalsToPCL		√	
HandEyeCalibration	√		
MultipleCamerasCaptureSequentially	√		
MultipleCamerasCaptureSimultaneously	√		

Install OpenCV

Execute the following command to install the latest version of OpenCV:

```
sudo apt update && sudo apt install -y unzip
wget -O opencv.zip https://github.com/opencv/opencv/archive/4.x.zip
unzip opencv.zip
mkdir build && cd build
cmake ../opencv-4.x
cmake --build .
sudo make install
```

Install PCL

Execute the following command to install PCL:

```
sudo apt-get install libpcl-dev
```



On different versions of Ubuntu, this command installs different versions of PCL. On Ubuntu 18.04, PCL 1.8.1 is installed; on Ubuntu 20.04, PCL 1.10.0 is installed.

Install HALCON

Follow these steps to install HALCON:

1. After downloading the HALCON installer, execute the following command to install HALCON (taking version 20.11 as an example):

```
tar zxvfHALCON-20.11.3.0-linux.tar.gz
sudo sh install-linux.sh #Note down the installation directory ofHALCON.
```

2. Add HALCON-relevant environment variables: open */etc/profile* in an editor (such as vi) and

paste the following lines to the end of the file. Replace `/opt/halcon` with the actual installation directory of HALCON.

```
HALCONARCH=x64-linux; exportHALCONARCH
HALCONROOT="/opt/halcon"; exportHALCONROOT
HALCONEXAMPLES=${HALCONROOT}/examples; exportHALCONEXAMPLES
HALCONIMAGES=${HALCONROOT}/examples/images; exportHALCONIMAGES
PATH=${HALCONROOT}/bin/${HALCONARCH}:${PATH}; export PATH
if [ ${LD_LIBRARY_PATH} ]; then
    LD_LIBRARY_PATH=${HALCONROOT}/lib/${HALCONARCH}:${LD_LIBRARY_PATH};
export LD_LIBRARY_PATH
else
    LD_LIBRARY_PATH=${HALCONROOT}/lib/${HALCONARCH}; export LD_LIBRARY_PATH
fi
```



- The changes are applied when you log in again. Or, you can run the `source /etc/profile/` command before you configure and build the samples.
- For more information, please refer to HALCON's installation guide.

Build and Run Samples

You can build all samples at once or build a single sample individually.



The paths provided in this section correspond to the samples included in the installation path. If you are using the samples cloned from GitHub, please modify the paths correspondingly.

Disable Unneeded Samples (Optional)



If you did not install some of the optional software libraries and still want to build all samples at once, then you must complete the steps in this section. Otherwise, an error will occur during the configuration of samples using CMake.

If you do not need the samples that depend on OpenCV, PCL or HALCON, please edit the CMakeLists file to disable these samples before building all samples at once.

1. Open `/opt/mech-mind/mech-eye-sdk/samples/cpp/area_scan_3d_camera/CMakeLists.txt` in an editor (such as vi).
2. Edit lines containing `options`: find the lines starting with `options`, change `ON` at the end of the line to `OFF` to disable the samples that depend on the software library mentioned in that line.

Example: If you do not need to build the sample that depends on HALCON, change `ON` at the end of the `option(USE_HALCON...)` line to `OFF`.

3. Save the changes in the editor and close the CMakeLists file.

Build the sample

Follow these steps to run the sample:

1. Enter the directory of the sample(s):

- If you want to build all samples at once, use the following command.

```
cd /opt/mech-mind/mech-eye-sdk/samples/cpp/area_scan_3d_camera/
```

- If you want to build a single sample individually, use the following command. Replace **Category** with the category name of the sample and **SampleName** with the name of the sample.

```
cd /opt/mech-mind/mech-eye-  
sdk/samples/cpp/area_scan_3d_camera/Category/SampleName/
```

2. Create a **build** folder in the directory for saving the built samples and so on.

```
sudo mkdir build && cd build
```

3. Configure and build the sample(s):

```
sudo cmake ..  
sudo make
```



If the source code files in the sample folder are altered, please delete the **build** folder and repeat steps 2 and 3 to rebuild the samples.

Run Samples

After building the samples, follow these steps to run the sample:

1. Enter the **build** folder in the directory of the sample(s) (Skip this step if you are running the sample(s) directly after building the sample(s):

- If all samples were built at once, use the following command.

```
cd /opt/mech-mind/mech-eye-sdk/samples/cpp/area_scan_3d_camera/build/
```

- If a single sample was built individually, use the following command. Replace **Category** with the category name of the sample and **SampleName** with the name of the sample.

```
cd /opt/mech-mind/mech-eye-  
sdk/samples/cpp/area_scan_3d_camera/Category/SampleName/build/
```

2. Run the sample: replace **SampleName** with the name of the sample. While the sample is running, input the index of the camera to which you want to connect according to the instruction, and press **Enter** to connect to the camera.

```
sudo ./SampleName
```

3. If image and/or point cloud files are saved by the sample, you can find the files in the **build** folder.

6.6.3. C#

This topic introduces how to build the Mech-Eye API C# samples with Visual Studio on Windows.

Sample List

Samples are divided into the following categories: **Basic**, **Advanced**, and **Util**.

- **Basic** samples: Connect and acquire data.
- **Advanced** samples: Acquire data in more complicated manners and set model-specific parameters.
- **Util** samples: Obtain camera information and set common parameters.

The samples in each category and their descriptions are provided below.

▼ Basic

- **ConnectToCamera**: Connect to a camera.
- **ConnectAndCaptureImages**: Connect to a camera and obtain the 2D image, depth map, and point cloud data.
- **Capture2DImage**: Obtain and save the 2D image.
- **CaptureDepthMap**: Obtain and save the depth map.
- **CapturePointCloud**: Obtain and save the untextured and textured point clouds.
- **CapturePointCloudHDR**: Set multiple exposure times, and then obtain and save the point cloud.
- **CapturePointCloudWithNormals**: Calculate normals and save the point cloud with normals.

▼ Advanced

- **ConvertDepthMapToPointCloud**: Generate a point cloud from the depth map and save the point cloud.
- **MultipleCamerasCaptureSequentially**: Obtain and save 2D images, depth maps, and point clouds sequentially from multiple cameras.
- **MultipleCamerasCaptureSimultaneously**: Obtain and save 2D images, depth maps, and point clouds simultaneously from multiple cameras.
- **CapturePeriodically**: Obtain and save 2D images, depth maps, and point clouds periodically for the specified duration from a camera.
- **Mapping2DImageToDepthMap**: Generate untextured and textured point clouds from a masked 2D image and a depth map.
- **SetParametersOfLaserCameras**: Set the parameters specific to laser cameras (the DEEP and LSR series).
- **SetParametersOfUHPCameras**: Set the parameters specific to the UHP series.

- **RegisterCameraEvent**: Define and register the callback function for monitoring the camera connection status.
- **CaptureStereo2DImages**: Obtain the 2D images from both 2D cameras of Deep (V3), Laser L Enhanced (V3), PRO XS (V4), LSR L (V4), LSR S (V4), and DEEP (V4).

▼ Util

- **GetCameraIntrinsics**: Obtain and print the camera intrinsic parameters.
- **PrintCameraInfo**: Obtain and print the camera information, such as model, serial number, firmware version, and temperatures.
- **SetScanningParameters**: Set the parameters in the **3D Parameters**, **2D Parameters**, and **ROI** categories.
- **SetDepthRange**: Set the **Depth Range** parameter.
- **SetPointCloudProcessingParameters**: Set the **Point Cloud Processing** parameters.
- **ManageUserSets**: Manage parameter groups, such as obtaining the names of all parameter groups, adding a parameter group, switching the parameter group, and saving parameter settings to the parameter group.
- **SaveAndLoadUserSet**: Import and replace all parameter groups from a JSON file, and save all parameter groups to a JSON file.

Prerequisites

In order to use the C# samples of Mech-Eye API, the following prerequisites must be satisfied:

- [Connect the camera and computer correctly.](#)
- Obtain the samples.
- Install the required software.

Obtain Samples

C# samples are included in the installation path of Mech-Eye SDK or can be obtained from [GitHub](#) by cloning. The samples in the installation path are the version at the time of Mech-Eye SDK release. The version on GitHub may contain the latest changes.

- The samples in the installation path are located in `xxx/Mech-Eye SDK-2.2.2/API/samples/csharp/area_scan_3d_camera`.
- The samples cloned from GitHub are located in `xxx/mecheye_csharp_samples/area_scan_3d_camera`.

Install Required Software

In order to use the C++ samples of Mech-Eye API, Mech-Eye SDK, Visual Studio and Emgu CV must be installed.

Install the Latest Version of Mech-Eye SDK

Please install or upgrade Mech-Eye SDK according to [Mech-Eye SDK Installation Guide](#).

Install Visual Studio (Version 2019 or Above)

1. Download the [Visual Studio installer](#).
2. When installing, select the following workloads and individual component, and then click [Install].
 - Workloads in the **Desktop & Mobile** category:
 - **.NET desktop development**
 - **Desktop development with C++**
 - **Universal Windows Platform development**
 - Individual component: **.NET Framework 4.8 targeting pack**



C# Mech-Eye API is developed based on .NET Framework 4.8. If .NET Framework 4.8 is not installed, the samples cannot be built.

Install Emgu CV

The **CaptureDepthMap** sample contains functions that depend on the OpenCV software library. Therefore, Emgu CV (the .NET wrapper for OpenCV) must be installed through NuGet Package Manager in Visual Studio.

For detailed instructions, refer to [the guide provided by Microsoft](#).

Build and Run Samples

After building the samples in Visual Studio, select the appropriate way to run the samples.

Build Samples in Visual Studio

1. Double-click **MechEyeCSharpSamples.sln** in the **area_scan_3d_camera** folder.
2. In Visual Studio toolbar, change the solution configuration from **Debug** to **Release**.
3. In the menu bar, select **Build > Build Solution**. An executable file (.exe) is generated for each sample. The executable files are saved to the **Build** folder, located in the **area_scan_3d_camera** folder.

Run Samples

You can run the samples in Visual Studio after building them, or run the samples by double-clicking the executable files.

Run a Sample in Visual Studio

1. In the **Solution Explorer** panel, right-click a sample, and select **Set as Startup Project**.
2. Click [**Local Windows Debugger**] in the toolbar to run the sample.

While the sample is running, input the index of the camera to which you want to connect according to the instruction, and press **Enter** to connect to the camera.

3. If image and/or point cloud files are saved by the sample, you can find the files in the **Build** folder.

Run the Executable File of a Sample

1. Enter the **Build** folder in the **area_scan_3d_camera** folder.
2. Run the executable file with the same name as the sample.

While the sample is running, input the index of the camera to which you want to connect according to the instruction, and press **Enter** to connect to the camera.

3. If image and/or point cloud files are saved by the sample, you can find the files in the **Build** folder.

6.6.4. Python (Windows)

This topic introduces how to obtain and run the Mech-Eye API Python samples on Windows.

Sample List

Samples are divided into the following categories: **basic**, **advanced**, and **util**.

- **basic** samples: Connect and acquire data.
- **advanced** samples: Acquire data in more complicated manners and set model-specific parameters.
- **util** samples: Obtain camera information and set common parameters.

The samples in each category and their descriptions are provided below.

▼ basic

- **connect_to_camera**: Connect to a camera.
- **connect_and_capture_images**: Connect to a camera and obtain the 2D image, depth map, and point cloud data.
- **capture_2d_image**: Obtain and save the 2D image.
- **capture_depth_map**: Obtain and save the depth map.
- **capture_point_cloud**: Obtain and save the untextured and textured point clouds.
- **capture_point_cloud_hdr**: Set multiple exposure times, and then obtain and save the point cloud.
- **capture_point_cloud_with_normals**: Calculate normals and save the point cloud with normals.

▼ advanced

- **convert_depth_map_to_point_cloud**: Generate a point cloud from the depth map and save the point cloud.
- **multiple_cameras_capture_sequentially**: Obtain and save 2D images, depth maps, and point clouds sequentially from multiple cameras.
- **multiple_cameras_capture_simultaneously**: Obtain and save 2D images, depth maps, and point clouds simultaneously from multiple cameras.
- **capture_periodically**: Obtain and save 2D images, depth maps, and point clouds periodically for the specified duration from a camera.
- **mapping_2d_image_to_depth_map**: Generate untextured and textured point clouds from a masked 2D image and a depth map.

- **set_parameters_of_laser_cameras**: Set the parameters specific to laser cameras (the DEEP and LSR series).
- **set_parameters_of_uhp_cameras**: Set the parameters specific to the UHP series.
- **register_camera_event**: Define and register the callback function for monitoring the camera connection status.
- **capture_stereo_2d_images**: Obtain the 2D images from both 2D cameras of Deep (V3), Laser L Enhanced (V3), PRO XS (V4), LSR L (V4), LSR S (V4), and DEEP (V4).

▼ util

- **get_camera_intrinsics**: Obtain and print the camera intrinsic parameters.
- **print_camera_info**: Obtain and print the camera information, such as model, serial number, firmware version, and temperatures.
- **set_scanning_parameters**: Set the parameters in the **3D Parameters**, **2D Parameters**, and **ROI** categories.
- **set_depth_range**: Set the **Depth Range** parameter.
- **set_point_cloud_processing_parameters**: Set the **Point Cloud Processing** parameters.
- **manage_user_sets**: Manage parameter groups, such as obtaining the names of all parameter groups, adding a parameter group, switching the parameter group, and saving parameter settings to the parameter group.
- **save_and_load_user_set**: Import and replace all parameter groups from a JSON file, and save all parameter groups to a JSON file.

Prerequisites

In order to use the Python samples of Mech-Eye API, the following prerequisites must be satisfied:

- [Connect the camera and computer correctly.](#)
- Make sure that the variant of Python installed is 64-bit, and that the version is between 3.7 and 3.11.



If **Add Python to PATH** was not selected during installation, please add the following two paths to the **Path** environment variable:

- *xxx/AppData/LocalProgramsPython/Python36/*
- *xxx/AppData/LocalProgramsPython/Python36/Scripts/*

- Obtain the samples.
- Install the required software.
- (Optional) Install software libraries on which the samples depend.

Obtain Samples

Python samples are included in the installation path of Mech-Eye SDK or can be obtained from [GitHub](#) by cloning. The samples in the installation path are the version at the time of Mech-Eye SDK release. The version on GitHub may contain the latest changes.

- The samples in the installation path are located in *xxx/Mech-Eye SDK-2.2.2/API/samples/python/area_scan_3d_camera*.
- The samples cloned from GitHub are located in

`xxx/mecheye_python_samples/area_scan_3d_camera.`

Install Required Software

In order to use the Python samples of Mech-Eye API, Mech-Eye SDK and Python Mech-Eye API must be installed.

Install the Latest Version of Mech-Eye SDK

Please install or upgrade Mech-Eye SDK according to [Mech-Eye SDK Installation Guide](#).

Install the Latest Version of Python Mech-Eye API

Follow these steps to install the latest version of Python Mech-Eye API:

1. Press Win + R on the keyboard to open Run box. Type "PowerShell" and then press Ctrl+Shift+Enter to run PowerShell as Administrator.
2. Execute the following command to install Python Mech-Eye API:

```
pip install MechEyeAPI
```

(Optional) Install Dependent Software Libraries

Some samples contain functions that depend on OpenCV. If you need to use the samples listed in the table below, OpenCV must be installed (latest version recommended).

- `capture_2d_image`
- `capture_depth_map`
- `capture_periodically`
- `capture_stereo_2d_images`
- `multiple_cameras_capture_sequentially`
- `multiple_cameras_capture_simultaneously`

Execute the following command to install the latest version of OpenCV:

```
pip install opencv-python
```

Run Samples



The paths provided in this section correspond to the samples included in the installation path. If you are using the samples cloned from GitHub, please modify the paths correspondingly.

Follow these steps to run the samples:

1. Navigate to the folder where a sample is located. Replace **category** with the category name of the sample.

```
cd xxx/Mech-Eye_SDK-2.2.2/API/samples/python/area_scan_3d_camera/category
```

2. Run the sample: replace **sample_name** with the name of the sample. While the sample is running, input the index of the camera to which you want to connect according to the instruction, and press **Enter** to connect to the camera.

```
python sample_name.py
```

3. If image and/or point cloud files are saved by the sample, you can find the files in the folder where the sample is located.

6.6.5. Python (Ubuntu)

This topic introduces how to obtain and run the Mech-Eye API Python samples on Ubuntu.

Sample List

Samples are divided into the following categories: **basic**, **advanced**, and **util**.

- **basic** samples: Connect and acquire data.
- **advanced** samples: Acquire data in more complicated manners and set model-specific parameters.
- **util** samples: Obtain camera information and set common parameters.

The samples in each category and their descriptions are provided below.

▼ basic

- **connect_to_camera**: Connect to a camera.
- **connect_and_capture_images**: Connect to a camera and obtain the 2D image, depth map, and point cloud data.
- **capture_2d_image**: Obtain and save the 2D image.
- **capture_depth_map**: Obtain and save the depth map.
- **capture_point_cloud**: Obtain and save the untextured and textured point clouds.
- **capture_point_cloud_hdr**: Set multiple exposure times, and then obtain and save the point cloud.
- **capture_point_cloud_with_normals**: Calculate normals and save the point cloud with normals.

▼ advanced

- **convert_depth_map_to_point_cloud**: Generate a point cloud from the depth map and save the point cloud.
- **multiple_cameras_capture_sequentially**: Obtain and save 2D images, depth maps, and point clouds sequentially from multiple cameras.
- **multiple_cameras_capture_simultaneously**: Obtain and save 2D images, depth maps, and point clouds simultaneously from multiple cameras.

- **capture_periodically**: Obtain and save 2D images, depth maps, and point clouds periodically for the specified duration from a camera.
- **mapping_2d_image_to_depth_map**: Generate untextured and textured point clouds from a masked 2D image and a depth map.
- **set_parameters_of_laser_cameras**: Set the parameters specific to laser cameras (the DEEP and LSR series).
- **set_parameters_of_uhp_cameras**: Set the parameters specific to the UHP series.
- **register_camera_event**: Define and register the callback function for monitoring the camera connection status.
- **capture_stereo_2d_images**: Obtain the 2D images from both 2D cameras of Deep (V3), Laser L Enhanced (V3), PRO XS (V4), LSR L (V4), LSR S (V4), and DEEP (V4).

▼ util

- **get_camera_intrinsics**: Obtain and print the camera intrinsic parameters.
- **print_camera_info**: Obtain and print the camera information, such as model, serial number, firmware version, and temperatures.
- **set_scanning_parameters**: Set the parameters in the **3D Parameters**, **2D Parameters**, and **ROI** categories.
- **set_depth_range**: Set the **Depth Range** parameter.
- **set_point_cloud_processing_parameters**: Set the **Point Cloud Processing** parameters.
- **manage_user_sets**: Manage parameter groups, such as obtaining the names of all parameter groups, adding a parameter group, switching the parameter group, and saving parameter settings to the parameter group.
- **save_and_load_user_set**: Import and replace all parameter groups from a JSON file, and save all parameter groups to a JSON file.

Prerequisites

In order to use the Python samples of Mech-Eye API, the following prerequisites must be satisfied:

- [Connect the camera and computer correctly.](#)
- Make sure that the variant of Python installed is 64-bit, and that the version is between 3.7 and 3.11.
- Obtain the samples.
- Install the required software.
- (Optional) Install software libraries on which the samples depend.

Obtain Samples

Python samples are included in the installation path of Mech-Eye SDK or can be obtained from [GitHub](#) by cloning. The samples in the installation path are the version at the time of Mech-Eye SDK release. The version on GitHub may contain the latest changes.

- The samples in the installation directory are located in `/opt/mech-mind/mech-eye-sdk/samples/python/area_scan_3d_camera`.
- The samples cloned from GitHub are located in `xxx/mecheye_python_samples/area_scan_3d_camera`.

Execute the following command to clone the Python samples from GitHub:

```
cd ~
git clone https://github.com/MechMindRobotics/mecheye_python_samples.git
```

Install Required Software

In order to use the Python samples of Mech-Eye API, dependency libraries, Mech-Eye SDK, and Python Mech-Eye API must be installed.

1. Install Mech-Eye SDK: Please refer to [Mech-Eye SDK Installation Guide \(Ubuntu\)](#).
2. Install pip3 or upgrade pip3 to ensure its version is 20.3.1 or above.

▼ Instructions

- Execute the following command to check if pip3 is installed and its version:

```
pip3 -V
```

- Execute the following command to install pip3:

```
sudo apt install python3-pip
```

- Execute the following command to upgrade pip3:

```
python3 -m pip install --upgrade pip
```

3. Upgrade g++ to ensure that its version is 12 or above.



If the g++ version is lower than 12, `import` will fail.

▼ Instructions

1. Execute the following command to install a later version g++ (using g++ 13 as an example):

```
sudo add-apt-repository ppa:ubuntu-toolchain-r/test
sudo apt-get update
sudo apt install g++-13
```

2. Use the `ls` command to check the installed versions of g++:

```
ls usr/bin/g++*
```

- Execute the following command to add all installed versions of g++ as alternatives (using g++ 9 and g++ 13 as examples):



The number at the end of the command is priority. The larger the number, the higher the priority.

```
sudo update-alternatives --install /usr/bin/g++ g++ /usr/bin/g++-9 10
sudo update-alternatives --install /usr/bin/g++ g++ /usr/bin/g++-13 20
```

- Execute the following command to select the g++ version. Enter the number corresponding to the later version g++ to select this version.

```
sudo update-alternatives --config g++
```

- Execute the following command to check if the later version g++ is successfully selected:

```
g++ --version
```

- Install Python Mech-Eye API.

```
sudo pip3 install MechEyeApi
```

(Optional) Install Dependent Software Libraries

Some samples contain functions that depend on OpenCV. If you need to use the samples listed in the table below, you must install OpenCV (latest version recommended).

- capture_2d_image
- capture_depth_map
- capture_periodically
- capture_stereo_2d_images
- multiple_cameras_capture_sequentially
- multiple_cameras_capture_simultaneously

Execute the following command to install the latest version of OpenCV:

```
sudo apt-get install libopencv-dev
sudo apt-get install python3-opencv
```

Run Samples



The paths provided in this section correspond to the samples included in the installation path. If you are using the samples cloned from GitHub, please modify the paths correspondingly.

Follow these steps to run the samples:

1. Navigate to the folder where a sample is located. Replace **category** with the category name of the sample.

```
cd ~/opt/mech-mind/mech-eye-sdk/samples/python/area_scan_3d_camera/category
```

2. Run the sample: replace **sample_name** with the name of the sample. While the sample is running, input the index of the camera to which you want to connect according to the instruction, and press **Enter** to connect to the camera.

```
sudo python3 sample_name.py
```

3. If image and/or point cloud files are saved by the sample, you can find the files in the folder where the sample is located.

When writing your own program, use the import command to import Mech-Eye API first:



```
python3
import MechEye
import open3d
```

6.6.6. VisionPro

This topic introduces how to use the VisionPro sample provided by Mech-Mind to connect to the camera and obtain 2D and 3D data through C# Mech-Eye API.

Sample List

The following sample is currently provided:

AcquireColorAndRangeImages: Obtain 2D and depth data and generate the Cognex-format 2D image (CogImage24PlanarColor) and depth map (CogImage16Range).

Prerequisites

In order to use the VisionPro sample, the following prerequisites must be satisfied:

- [Connect the camera and computer correctly.](#)
- Obtain the sample.
- Install the required software.

Obtain Sample

VisionPro sample is included in the installation path of Mech-Eye SDK or can be obtained from [GitHub](#) by cloning. The sample in the installation path is the version at the time of Mech-Eye SDK release. The version on GitHub may contain the latest changes.

- The sample in the installation path is located in `xxx/Mech-Eye SDK-2.2.2/API/samples/visionpro/area_scan_3d_camera`.
- The sample cloned from GitHub is located in `xxx/mecheye_visionpro_samples/area_scan_3d_camera`.

Install Required Software

In order to use the VisionPro sample, Mech-Eye SDK and VisionPro must be installed.

- Mech-Eye SDK: Install the latest version or upgrade according to [Mech-Eye SDK Installation Guide](#).
- Cognex VisionPro: Version 9.8 SR1 has been tested.

Configure and Run Sample

The sample can be run after the necessary configurations are completed.

Copy Dynamic-Link Libraries

To call Mech-Eye API in VisionPro, the dynamic-link libraries of Mech-Eye SDK must be copied to the installation path of VisionPro first.

Follow these steps to copy the dynamic-link libraries:

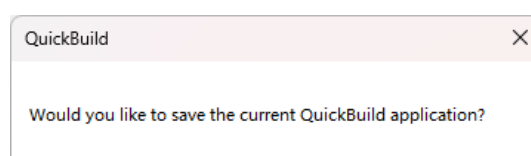
1. Go to the installation path of Mech-Eye SDK and open the following folder: `xxx/Mech-Eye SDK-2.2.2/API/dll`.
2. Copy the following dynamic-link library files (.dll) in the above folder:
 - **MechEyeApi**
 - **MechEyeAPiNet**
 - **MechEyeApiWrapper**
3. Go to the installation path of VisionPro, open the **bin** folder, and paste the copied dynamic-link library files.

Configure Referenced Assembly of Script

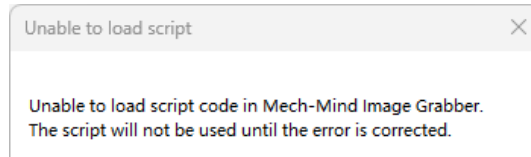
After copying the dynamic-link libraries, you need to configure the path of the referenced assembly in the script of the sample.



Follow these steps to configure the referenced assembly:

1. Open VisionPro QuickBuild and select **Open QuickBuild Application** in the **File** menu.
2. In the pop-up window, select the VisionPro sample to be used, and click the **[Open]** button.
3. The following window will pop-up. Click the **[No]** button in the window to close it.



4. The following window may pop-up. If it pops up, click the **[OK]** button in the window to close it.



5. In the left panel of VisionPro QuickBuild, double-click **Mech-Eye Industrial 3D Camera**.
6. In the left panel of the popped up Job Editor window, double-click **Image Acquisition**.
7. In the toolbar of the popped up **Image Acquisition** window, click the  button.
8. In the toolbar of the popped up **Image Acquisition Script** window, click the  button.
9. In the popped up **Add / Remove Referenced Assemblies** window, scroll down to the bottom and double-click the **MechEyeApiNet.dll** row.
10. In the popped up **Enter Assembly Name** window, click the [**Browse.....**] button, navigate to the folder where the dynamic-link libraries are located in the Mech-Eye SDK installation path (*xxx/Mech-Eye SDK-2.2.2/API/dll*), and select the **MechEyeApiNet** dynamic-link library file.
11. Click the [**OK**] button in the **Enter Assembly Name** window to close it, and then click the [**OK**] button in the **Add / Remove Referenced Assemblies** window to close it.

Set Camera IP Address

Before running the sample, you need to set the IP address of the camera in the script, so that the camera can be successfully connected.




You can modify the IP address of the camera using [Mech-Eye Viewer](#) or the [IP configuration tool](#).

Follow these steps to set the IP address of the camera:


1. In the **Image Acquisition Script** window, locate the following line and change the IP address in it to the actual IP address of the camera to be connected.

```
status = camera.Connect("192.168.20.15", 10000)
```

2. Click the  button in the toolbar to make the change take effect.
3. Close the **Image Acquisition Script** and **Image Acquisition** windows to return to the Job Editor window.

Run Sample

Follow these steps to run the sample and view the obtained 2D image and depth map:

1. In the toolbar of the Job Editor window, click the  button to run the sample.
2. In the left panel, double-click the corresponding tool to view the obtained 2D image or depth map:
 - 2D image: Double-click **CogAffineTransformTool1**.
 - Depth map: Double-click **Cog3DVisionDataReRenderTool1**.
3. If the obtained images do not meet your requirements, refer to the next section and adjust the camera parameters to improve the quality of the data.

Adjust Camera Parameters

The camera parameters can be adjusted in Mech-Eye Viewer or in the sample through C# Mech-Eye API. It is recommended to use Mech-Eye Viewer to adjust parameters, as it provides a graphical user interface where you can acquire data immediately after adjusting the parameters to see the effect.

- Adjust parameters using Mech-Eye Viewer: Please refer to [How to Use](#) for instructions on connecting to the camera, acquiring data, viewing current parameter values, and adjusting parameters in Mech-Eye Viewer.
- Adjust parameters in the sample through C# Mech-Eye API: To adjust the camera parameters in the sample, you need to add the corresponding methods of C# Mech-Eye API to the script of the sample. Please refer to the following contents and edit the script of the sample.
 - C# Mech-Eye API samples: Refer to [C#](#) for an overview of the available samples, where to obtain them, and how to use them.
 - [Mech-Eye API Reference](#): Explanations of the methods in C++ Mech-Eye API are provided here.

6.7. Tools

6.7.1. Firmware Upgrade Tool

With this tool, you can upgrade the camera firmware on Ubuntu directly, simplifying the processing of firmware upgrade.

The firmware upgrade tool can be obtained from [Mech-Mind Download Center](#).

The firmware upgrade tool consists of two parts: an executable file named `firmware_upgrader` and a compressed file named `mmind_eye_v2.2.2`.

Follow these steps to upgrade the firmware:

1. Execute the following command to give execute permission to the firmware upgrade tool.

```
chmod 777 xxx/firmware_upgrader
```

2. Execute the following command to upgrade the firmware.

```
sudo xxx/firmware_upgrader xxx/mmind_eye_v2.2.2.zip
```



- Enter the path of `firmware_upgrader` first, followed by the path of `mmind_eye_v2.2.2.zip`. Otherwise, the firmware upgrade tool cannot be executed.
- The two paths must be separated by a space. Otherwise, the firmware upgrade tool cannot be executed.

3. Input the index of the camera whose firmware needs to be upgraded according to the instruction, and press **Enter** to connect to the camera. If **Upgrade Completed** is displayed, the firmware was successfully upgraded.



If the firmware upgrade failed, please retry or contact Technical Support.

6.7.2. IP Configuration Tool

With this tool, you can modify the IP configuration of the camera on Ubuntu directly, simplifying the processing of IP configuration.

Prerequisites

In order to use the IP configuration tool, the following prerequisites must be satisfied:

[Download and install Mech-Eye SDK 2.2.0 or above](#) to obtain the IP configuration tool.

The IP configuration tool is an executable file named `ip_configurator`, located in `/opt/mech-mind/mech-eye-sdk/bin`.

The IP configuration tool is independent from Mech-Eye SDK. You can move this executable file to another location and run it on its own.

Instructions

Follow these steps to modify the IP configuration of the camera:

1. Execute the following command to run the IP configuration tool: Replace the path with the actual path where the `ip_configurator` executable file is located.

```
sudo /opt/mech-mind/mech-eye-sdk/bin/ip_configurator
```

2. Input the index of the camera whose IP configuration needs to be modified according to the instruction, and press **Enter** to connect to the camera.
3. Select the IP address assignment method based on the prompt: `0` is to assign the IP address dynamically, and `1` is to set a static IP address.
 - To assign an IP address dynamically, proceed to step 6.
 - To set a static IP address, proceed to the next step.
4. Enter the IP address to be set and press the **Enter** key.
5. Enter the subnet mask to be set and press the **Enter** key. To set the subnet mask to `255.255.255.0`, you can enter `0` and then press the **Enter** key.
6. The old and new IP configuration of the camera will be printed in the window. Please check if the new IP configuration is correct:
 - If correct, enter `1` and press the **Enter** key to complete the IP configuration.
 - If incorrect, enter `0` and press the **Enter** key. The program will list all the available cameras again. Please go back to step 2 and repeat the process.

6.8. Backward Compatibility

6.8.1. Backward Compatibility of Version 2.2.0

Mech-Eye API 2.2.0 has been restructured to provide a clearer structure while keeping all the previously available functions. New functions have also been added.

If you need to upgrade the camera firmware to 2.2.0 but need to keep using the 2.1.0 version of Mech-Eye API in your client program, please refer to this topic and complete the corresponding actions for the language in use.



If you wish to use the 2.2.0 version of Mech-Eye API in your existing client program, please refer to [Migration Guide](#) and modify your client program.

C++

To keep using the 2.1.0 version of C++ Mech-Eye API, please complete the corresponding actions based on the operating system in use.

Windows

The 2.1.0 version C++ Mech-Eye API is included in Mech-Eye SDK 2.2.0.

Please [upgrade Mech-Eye SDK](#) and upgrade the camera firmware using Mech-Eye Viewer 2.2.0. After upgrading, please build the client program again.

Ubuntu

The 2.1.0 version C++ Mech-Eye API is included in Mech-Eye SDK 2.2.0.

Please [upgrade Mech-Eye SDK](#) and upgrade the camera firmware with the [firmware upgrade tool](#). After upgrading, please build the client program again.

C#

To keep using the 2.1.0 version of C# Mech-Eye API in your client program, please pay attention to the following:

- While [upgrading Mech-Eye SDK](#), please select [**Upgrade and keep history versions**], and uncheck **Add to PATH**.
- Upgrade the camera firmware with Mech-Eye Viewer 2.2.0.

Python

To keep using the 2.1.0 version of Python Mech-Eye API, please complete the corresponding actions based on the operating system in use.

Windows

If you are using the Windows operating system, please pay attention to the following:

- Do not install the latest version of Python Mech-Eye API using `pip`.
- While [upgrading Mech-Eye SDK](#), please select [**Upgrade and keep history versions**], and uncheck **Add to PATH**.
- Upgrade the camera firmware with Mech-Eye Viewer 2.2.0.

Ubuntu

If you are using the Ubuntu operating system, please pay attention to the following:

- Do not install the latest version of Python Mech-Eye API using [pip3](#).
- Upgrade the camera firmware with the [firmware upgrade tool](#).

6.8.2. C++ Parameter Name Changes (2.0.0)

Mech-Eye API 2.0.0 changed some C++ parameter names. If you have installed Mech-Eye SDK 2.0.0 but still use programs written before, please modify the parameter names in your programs according to the following table.

Before 2.0.0	2.0.0
Scanning3DSettings::ExposureSequence	Scanning3DSettings::exposureSequence
Scanning3DSettings::Gain	Scanning3DSettings::gain
Scanning3DSettings::Scan3DROI	Scanning3DSettings::scan3DROI
Scanning3DSettings::DepthRange	Scanning3DSettings::depthRange
Scanning2DSettings::Scan2DExposureMode::ExposureMode	Scanning2DSettings::Scan2DExposureMode::exposureMode
Scanning2DSettings::ExposureTime	Scanning2DSettings::exposureTime
Scanning2DSettings::SharpenFactor	Scanning2DSettings::sharpenFactor
Scanning2DSettings::ExpectedGrayValue	Scanning2DSettings::expectedGrayValue
Scanning2DSettings::Scan2DROI	Scanning2DSettings::scan2DROI
Scanning2DSettings::ToneMappingEnable	Scanning2DSettings::toneMappingEnable
Scanning2DSettings::HDRExposureSequence	Scanning2DSettings::hdrExposureSequence
PointCloudProcessingSettings::FringeContrastThreshold	PointCloudProcessingSettings::fringeContrastThreshold
PointCloudProcessingSettings::FringeMinThreshold	PointCloudProcessingSettings::fringeMinThreshold
PointCloudProcessingSettings::CloudOutlierFilterMode::OutlierFilterMode	PointCloudProcessingSettings::CloudOutlierFilterMode::outlierFilterMode
PointCloudProcessingSettings::CloudSmoothMode::SmoothMode	PointCloudProcessingSettings::CloudSmoothMode::smoothMode
LaserSettings::LaserFringeCodingMode::FringeCodingMode	LaserSettings::LaserFringeCodingMode::fringeCodingMode
LaserSettings::FrameRangeStart	LaserSettings::frameRangeStart
LaserSettings::FrameRangeEnd	LaserSettings::frameRangeEnd
LaserSettings::FramePartitionCount	LaserSettings::framePartitionCount
LaserSettings::PowerLevel	LaserSettings::powerLevel
UhpSettings::UhpCaptureMode::CaptureMode	UhpSettings::UhpCaptureMode::captureMode
UhpSettings::UhpFringeCodingMode::FringeCodingMode	UhpSettings::UhpFringeCodingMode::fringeCodingMode

6.8.3. C# Method and Parameter Name Changes (2.0.0)

Mech-Eye API 2.0.0 changed C# method and parameter names. If you have installed Mech-Eye SDK 2.0.0 but still use programs written before, please modify the method and parameter names in your programs according to the following tables.

Method Name Changes

Before 2.0.0	2.0.0
ColorMap.at(uint, uint)	ColorMap.At(uint, uint)
ColorMap.data()	ColorMap.Data()
ColorMap.empty()	ColorMap.Empty()
ColorMap.height()	ColorMap.Height()
ColorMap.release()	ColorMap.Release()
ColorMap.resize(uint, uint)	ColorMap.Resize(uint, uint)
ColorMap.width()	ColorMap.Width()
DepthMap.at(uint, uint)	DepthMap.At(uint, uint)
DepthMap.data()	DepthMap.Data()
DepthMap.empty()	DepthMap.Empty()
DepthMap.height()	DepthMap.Height()
DepthMap.release()	DepthMap.Release()
DepthMap.resize(uint, uint)	DepthMap.Resize(uint, uint)
DepthMap.width()	DepthMap.Width()
MechEyeDevice.addUserSet(string)	MechEyeDevice.AddUserSet(string)
MechEyeDevice.captureColorMap(mmind.apiSharp.ColorMap)	MechEyeDevice.CaptureColorMap(mmind.apiSharp.ColorMap)
MechEyeDevice.captureDepthMap(mmind.apiSharp.DepthMap)	MechEyeDevice.CaptureDepthMap(mmind.apiSharp.DepthMap)
MechEyeDevice.capturePointXYZBGRMap(mmind.apiSharp.PointXYZBGRMap)	MechEyeDevice.CapturePointXYZBGRMap(mmind.apiSharp.PointXYZBGRMap)
MechEyeDevice.capturePointXYZMap(mmind.apiSharp.PointXYZMap)	MechEyeDevice.CapturePointXYZMap(mmind.apiSharp.PointXYZMap)
MechEyeDevice.connect(mmind.apiSharp.MechEyeDeviceInfo)	MechEyeDevice.Connect(mmind.apiSharp.MechEyeDeviceInfo)
MechEyeDevice.connect(string)	MechEyeDevice.Connect(string)
MechEyeDevice.connect(string, ushort)	MechEyeDevice.Connect(string, ushort)
MechEyeDevice.deleteUserSet(string)	MechEyeDevice.DeleteUserSet(string)
MechEyeDevice.disconnect()	MechEyeDevice.Disconnect()
MechEyeDevice.enumerateMechEyeDeviceList()	MechEyeDevice.EnumerateMechEyeDeviceList()

Before 2.0.0	2.0.0
MechEyeDevice.getAllUserSets(System.Collections.Generic.List<string>)	MechEyeDevice.GetAllUserSets(System.Collections.Generic.List<string>)
MechEyeDevice.getApiInformation()	MechEyeDevice.GetApiInformation()
MechEyeDevice.getCloudOutlierFilterMode(mmind.apiSharp.CloudOutlierFilterMode)	MechEyeDevice.GetCloudOutlierFilterMode(mmind.apiSharp.CloudOutlierFilterMode)
MechEyeDevice.getCloudSmoothMode(mmind.apiSharp.CloudSmoothMode)	MechEyeDevice.GetCloudSmoothMode(mmind.apiSharp.CloudSmoothMode)
MechEyeDevice.getCurrentUserSet(string)	MechEyeDevice.GetCurrentUserSet(string)
MechEyeDevice.getDepthRange(mmind.apiSharp.DepthRange)	MechEyeDevice.GetDepthRange(mmind.apiSharp.DepthRange)
MechEyeDevice.getDeviceInfo(mmind.apiSharp.MechEyeDeviceInfo)	MechEyeDevice.GetDeviceInfo(mmind.apiSharp.MechEyeDeviceInfo)
MechEyeDevice.getDeviceIntri(mmind.apiSharp.DeviceIntri)	MechEyeDevice.GetDeviceIntri(mmind.apiSharp.DeviceIntri)
MechEyeDevice.getDeviceResolution(mmind.apiSharp.DeviceResolution)	MechEyeDevice.GetDeviceResolution(mmind.apiSharp.DeviceResolution)
MechEyeDevice.getFringeContrastThreshold(int)	MechEyeDevice.GetFringeContrastThreshold(int)
MechEyeDevice.getFringeMinThreshold(int)	MechEyeDevice.GetFringeMinThreshold(int)
MechEyeDevice.getLaserSettings(mmind.apiSharp.LaserSettings)	MechEyeDevice.GetLaserSettings(mmind.apiSharp.LaserSettings)
MechEyeDevice.getScan2DExpectedGrayValue(int)	MechEyeDevice.GetScan2DExpectedGrayValue(int)
MechEyeDevice.getScan2DExposureMode(mmind.apiSharp.Scan2DExposureMode)	MechEyeDevice.GetScan2DExposureMode(mmind.apiSharp.Scan2DExposureMode)
MechEyeDevice.getScan2DExposureTime(double)	MechEyeDevice.GetScan2DExposureTime(double)
MechEyeDevice.getScan2DHDRExposureSequence(System.Collections.Generic.List<double>)	MechEyeDevice.GetScan2DHDRExposureSequence(System.Collections.Generic.List<double>)
MechEyeDevice.getScan2DROI(mmind.apiSharp.ROI)	MechEyeDevice.GetScan2DROI(mmind.apiSharp.ROI)
MechEyeDevice.getScan2DSharpenFactor(double)	MechEyeDevice.GetScan2DSharpenFactor(double)
MechEyeDevice.getScan2DToneMappingEnable(bool)	MechEyeDevice.GetScan2DToneMappingEnable(bool)
MechEyeDevice.getScan3DExposure(System.Collections.Generic.List<double>)	MechEyeDevice.GetScan3DExposure(System.Collections.Generic.List<double>)
MechEyeDevice.getScan3DGain(double)	MechEyeDevice.GetScan3DGain(double)
MechEyeDevice.getScan3DROI(mmind.apiSharp.ROI)	MechEyeDevice.GetScan3DROI(mmind.apiSharp.ROI)
MechEyeDevice.getUhpCaptureMode(mmind.apiSharp.UhpCaptureMode)	MechEyeDevice.GetUhpCaptureMode(mmind.apiSharp.UhpCaptureMode)

Before 2.0.0	2.0.0
MechEyeDevice.getUhpFringeCodingMode(mm mind.apiSharp.UhpFringeCodingMode)	MechEyeDevice.GetUhpFringeCodingMode(mm ind.apiSharp.UhpFringeCodingMode)
MechEyeDevice.getUhpSettings(mm mind.apiShar p.UhpSettings)	MechEyeDevice.GetUhpSettings(mm mind.apiShar p.UhpSettings)
MechEyeDevice.saveAllSettingsToUserSets()	MechEyeDevice.SaveAllSettingsToUserSets()
MechEyeDevice.setCloudOutlierFilterMode(mm ind.apiSharp.CloudOutlierFilterMode)	MechEyeDevice.SetCloudOutlierFilterMode(mm ind.apiSharp.CloudOutlierFilterMode)
MechEyeDevice.setCloudSmoothMode(mm mind. apiSharp.CloudSmoothMode)	MechEyeDevice.SetCloudSmoothMode(mm mind. apiSharp.CloudSmoothMode)
MechEyeDevice.setCurrentUserSet(string)	MechEyeDevice.SetCurrentUserSet(string)
MechEyeDevice.setDepthRange(mm mind.apiShar p.DepthRange)	MechEyeDevice.SetDepthRange(mm mind.apiShar p.DepthRange)
MechEyeDevice.setFringeContrastThreshold(int)	MechEyeDevice.SetFringeContrastThreshold(int)
MechEyeDevice.setFringeMinThreshold(int)	MechEyeDevice.SetFringeMinThreshold(int)
MechEyeDevice.setLaserSettings(mm mind.apiSh arp.LaserSettings)	MechEyeDevice.SetLaserSettings(mm mind.apiSh arp.LaserSettings)
MechEyeDevice.setScan2DExpectedGrayValue(i nt)	MechEyeDevice.SetScan2DExpectedGrayValue(i nt)
MechEyeDevice.setScan2DExposureMode(mm ind.apiSharp.Scan2DExposureMode)	MechEyeDevice.SetScan2DExposureMode(mm ind.apiSharp.Scan2DExposureMode)
MechEyeDevice.setScan2DExposureTime(doubl e)	MechEyeDevice.SetScan2DExposureTime(doubl e)
MechEyeDevice.setScan2DHDRExposureSeque nce(System.Collections.Generic.List<double>)	MechEyeDevice.SetScan2DHDRExposureSeque nce(System.Collections.Generic.List<double>)
MechEyeDevice.setScan2DROI(mm mind.apiShar p.ROI)	MechEyeDevice.SetScan2DROI(mm mind.apiShar p.ROI)
MechEyeDevice.setScan2DSharpenFactor(doubl e)	MechEyeDevice.SetScan2DSharpenFactor(doubl e)
MechEyeDevice.setScan2DToneMappingEnable (bool)	MechEyeDevice.SetScan2DToneMappingEnable (bool)
MechEyeDevice.setScan3DExposure(System.C ollections.Generic.List<double>)	MechEyeDevice.SetScan3DExposure(System.C ollections.Generic.List<double>)
MechEyeDevice.setScan3DGain(double)	MechEyeDevice.SetScan3DGain(double)
MechEyeDevice.setScan3DROI(mm mind.apiShar p.ROI)	MechEyeDevice.SetScan3DROI(mm mind.apiShar p.ROI)
MechEyeDevice.setUhpCaptureMode(mm mind.ap iSharp.UhpCaptureMode)	MechEyeDevice.SetUhpCaptureMode(mm mind.ap iSharp.UhpCaptureMode)
MechEyeDevice.setUhpFringeCodingMode(mm ind.apiSharp.UhpFringeCodingMode)	MechEyeDevice.SetUhpFringeCodingMode(mm ind.apiSharp.UhpFringeCodingMode)
MechEyeDevice.setUhpSettings(mm mind.apiShar p.UhpSettings)	MechEyeDevice.SetUhpSettings(mm mind.apiShar p.UhpSettings)

Before 2.0.0	2.0.0
PointXYZBGRMap.at(uint, uint)	PointXYZBGRMap.At(uint, uint)
PointXYZBGRMap.data()	PointXYZBGRMap.Data()
PointXYZBGRMap.empty()	PointXYZBGRMap.Empty()
PointXYZBGRMap.height()	PointXYZBGRMap.Height()
PointXYZBGRMap.release()	PointXYZBGRMap.Release()
PointXYZBGRMap.resize(uint, uint)	PointXYZBGRMap.Resize(uint, uint)
PointXYZBGRMap.width()	PointXYZBGRMap.Width()
PointXYZMap.at(uint, uint)	PointXYZMap.At(uint, uint)
PointXYZMap.data()	PointXYZMap.Data()
PointXYZMap.empty()	PointXYZMap.Empty()
PointXYZMap.height()	PointXYZMap.Height()
PointXYZMap.release()	PointXYZMap.Release()
PointXYZMap.resize(uint, uint)	PointXYZMap.Resize(uint, uint)
PointXYZMap.width()	PointXYZMap.Width()

Parameter Name Changes

Before 2.0.0	2.0.0
LaserSettings.FramePartitionCount	LaserSettings.framePartitionCount
LaserSettings.FrameRangeEnd	LaserSettings.frameRangeEnd
LaserSettings.FrameRangeStart	LaserSettings.frameRangeStart
LaserSettings.FringeCodingMode	LaserSettings.fringeCodingMode
LaserSettings.PowerLevel	LaserSettings.powerLevel
PointCloudProcessingSettings.CloudSmoothMode	PointCloudProcessingSettings.cloudSmoothMode
PointCloudProcessingSettings.FringeContrastThreshold	PointCloudProcessingSettings.fringeContrastThreshold
PointCloudProcessingSettings.FringeMinThreshold	PointCloudProcessingSettings.fringeMinThreshold
PointCloudProcessingSettings.OutlierFilterMode	PointCloudProcessingSettings.outlierFilterMode
Scanning2DSettings.ExpectedGrayValue	Scanning2DSettings.expectedGrayValue
Scanning2DSettings.ExposureMode	Scanning2DSettings.exposureMode
Scanning2DSettings.ExposureTime	Scanning2DSettings.exposureTime
Scanning2DSettings.Scan2DROI	Scanning2DSettings.scan2DROI
Scanning2DSettings.SharpenFactor	Scanning2DSettings.sharpenFactor
Scanning2DSettings.ToneMappingEnable	Scanning2DSettings.toneMappingEnable

Before 2.0.0	2.0.0
Scanning3DSettings.DepthRange	Scanning3DSettings.depthRange
Scanning3DSettings.ExposureSequence	Scanning3DSettings.exposureSequence
Scanning3DSettings.Gain	Scanning3DSettings.gain
Scanning3DSettings.Scan3DROI	Scanning3DSettings.scan3DROI
UhpSettings.CaptureMode	UhpSettings.captureMode
UhpSettings.FringeCodingMode	UhpSettings.fringeCodingMode

7. GenICam Interface

7.1. What is GenICam?

GenICam is a common machine vision industry standard developed by the European Machine Vision Association (EMVA), which allows the use of a unified programming interface to control machine vision cameras.

GenICam provides a widely-applicable programming interface that provides an end-to-end configuration interface for standard interfaces including GigE Vision, USB 3.0 Vision, Camera Link, and IEEE 1394.

For cameras compliant with GenICam, GenICam-compliant machine vision software (such as HALCON) can be used to conveniently connect and control cameras without camera-specific configuration.

Processing Principles

1. A GenICam-compliant camera provides an XML descriptor file, which describes the camera parameters and indicates the features and functions provided by the camera. A GenICam-compatible machine vision software converts the XML file into the GenAPI application programming interfaces or graphical user interface elements.
2. A GenICam-compatible machine vision software converts the XML file into the GenAPI application programming interfaces or graphical user interface elements.
3. By using the application programming interfaces or graphical user interface on the machine vision software, you can easily access and use the features and functions provided by the camera.

For more information about GenICam, please visit: [GenICam](#).

Camera's Support for GenICam Standard

After upgrading the camera firmware to 2.0.0 or above, the camera is compatible with the GenICam/GigE Vision standards, and you can control cameras with third-party machine vision software such as HALCON.

For camera parameters provided for GenICam by the camera and their description, refer to [Camera Parameters Supported for GenICam](#).

If you want to connect and manage the camera with the HALCON software, refer to [HALCON: Connect to Camera, Perform Image Capturing, and Adjust Parameters](#).

7.2. HALCON

Using the HALCON samples provided by Mech-Mind, you can complete procedures including camera connection, data acquisition, and hand-eye calibration.

HALCON samples are included in the installation path of Mech-Eye SDK or can be obtained from [GitHub](#) by cloning. The samples in the installation path are the version at the time of Mech-Eye SDK release. The version on GitHub may contain the latest changes.

- The samples in the installation path are located in `xxx/Mech-Eye SDK-2.2.2/API/samples/halcon/area_scan_3d_camera`.
- The samples cloned from GitHub are located in `xxx/mecheye_halcon_samples/area_scan_3d_camera`.

The following topics provide instructions on using each sample:

- [HALCON](#)
- [HALCON: Hand-Eye Calibration](#)
- [HALCON: Obtain Textured Point Cloud](#)
- [HALCON: Obtain Depth Values Only](#)
- [HALCON: Set IP Address](#)
- Obtain Point Cloud with Normals (coming soon)



If you encounter any issue while using the camera, please visit [Mech-Mind Online Community](#) to get help.

7.2.1. HALCON: Connect to Camera, Perform Image Capturing, and Adjust Parameters

This topic provides instructions on connecting to a Mech-Eye Industrial 3D Camera, perform image capturing, and adjust camera parameters of the camera with HALCON image acquisition assistant and the HALCON sample provided by Mech-Mind.

Prerequisites

- [The camera and computer are properly connected.](#)



For Nano (V3) and Pro XS (V3), it is recommended to connect the camera directly to the computer without using a network switch.

- [HALCON 20.11](#) or above has been installed on computer.



HALCON versions below 20.11 are not fully tested.

- Mech-Eye SDK 2.0.0 or above has been installed on computer.



The version of the camera firmware must be consistent with that of Mech-Eye SDK. If not, please [upgrade the camera firmware](#).

- The IP addresses of the camera and computer are in the same subnet.



It is recommended to use static IP addresses. For instructions on setting the camera IP address, please refer to [Set Camera IP Address with Mech-Eye Viewer](#).

In HDevelop, you can connect to the camera with the image acquisition assistant or the sample provided by Mech-Mind.

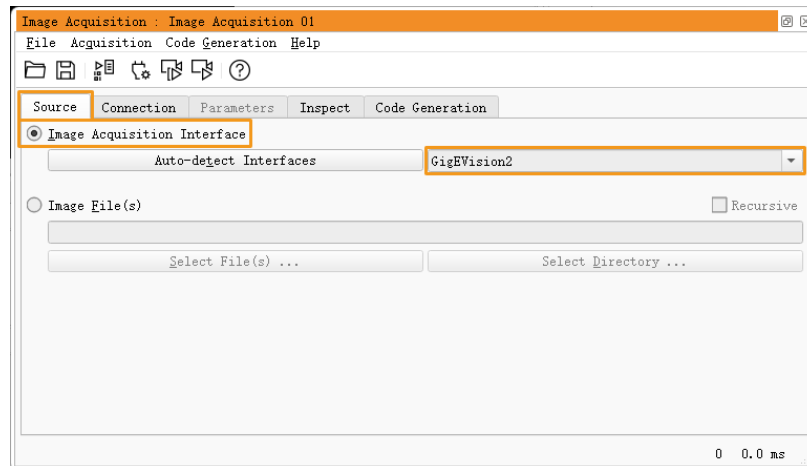
Use Image Acquisition Assistant

Using the image acquisition assistant provided by HDevelop, you can quickly connect to the camera, perform image capturing and adjust camera parameters.

To open the image acquisition assistant: open HDevelop, and in the menu bar, select Assistants > Open New Image Acquisition.

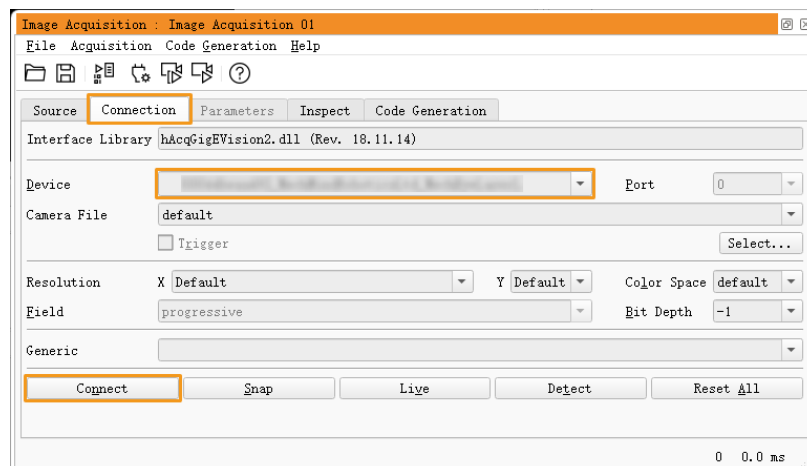
Connect to the Camera

1. In the Image Acquisition window, under the **Source** tab, select **Image Acquisition Interface**, and select **GigEVision2** from the drop-down menu.



If the GigEVision2 option is unavailable, the GigEVision2 image acquisition interface is not installed. Please refer to HALCON's Installation Guide and install the interface through MVTec Software Manager (SOM).

2. Under the **Connection** tab, from the drop-down menu of **Device**, select the device you would like to connect. Then, click **[Connect]** in the lower left.



- If the camera is successfully connected, the **[Connect]** button changes to **[Disconnect]**. If this change does not occur, the camera connection failed. To disconnect from the camera in HDevelop, click **[Disconnect]**.
- If the camera is already connected by another client, it cannot be connected by HDevelop. Please disconnect the camera from the other client first.
- The camera name displayed in **Device** can be customized in Mech-Eye Viewer for easier distinction. Connect to the camera in Mech-Eye Viewer, and then [customize the camera name](#). Avoid using special characters such as letters with accent marks, as these characters cannot be displayed in HDevelop.

Perform Image Capturing

To perform image capturing once: under the **Connection** tab, click [**Snap**].



If the image capturing process takes too long, you can increase the MTU size of the camera, and [enable jumbo frames](#) on your computer.

You can also perform image capturing multiple times or continuously. To do this, you need to adjust the **AcquisitionMode** parameter first.

- To perform image capturing multiple times:
 - Switch to the **Parameters** tab, set the value of the **AcquisitionMode** parameter to **MultiFrame**.
 - Click [**Refresh**] in the upper right, and then set the number of times for image capturing in the **AcquisitionFrameCount** parameter.
 - Switch to the **Connection** tab and click [**Live**] to perform image capturing.
 - After the set times of image capturing are completed, the [**Live**] button becomes [**Stop**]. Click [**Stop**] to stop image capturing.
- To perform image capturing continuously:
 - Switch to the **Parameters** tab, set the value of the **AcquisitionMode** parameter to **Continuous**.
 - Switch to the **Connection** tab and click [**Live**] to perform image capturing.
 - The [**Live**] button becomes [**Stop**]. Click [**Stop**] to stop image capturing.



- You can still use [**Snap**] to perform single-time image capturing when **AcquisitionMode** is set to **MultiFrame** or **Continuous**.
- When **AcquisitionMode** is set to **SingleFrame**, you can only perform single-time image capturing.

Select Data Type

After the camera is connected, the 2D image is obtained by default. You can select whether to obtain 2D image or depth map by adjusting the **DeviceScanType** parameter.

1. Click the **Parameters** tab and find the **DeviceScanType** parameter. Set its value according to your needs. The parameter values are explained below:

Value	Data type
Areascan	2D image
Areascan3D	Depth map (an image containing depth information)

2. Perform image capturing to obtain the data type that you selected.



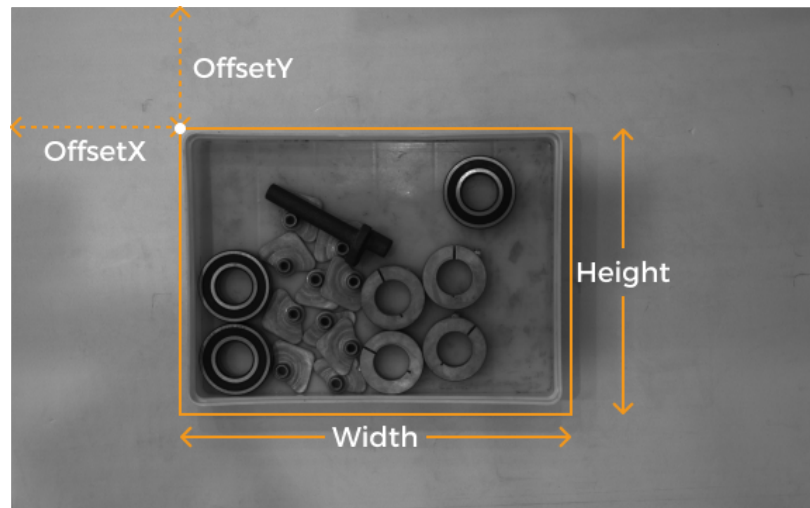
If **Update Image** in the upper right of the **Parameters** tab is checked, the image in the **Canvas** window is automatically updated as you adjust the parameters.

Set Capture Region

After the camera is connected, if you need to trim the obtained image, you can set a capture region by adjusting the **Height**, **Width**, **OffsetX** and **OffsetY** parameters.

To set a capture region, follow these steps:

1. Select the data type for setting a capture region.
2. Perform image capturing once with [**Snap**] to check the current image.
3. Switch to the **Parameters** tab and adjust the **Height**, **Width**, **OffsetX** and **OffsetY** parameters. The following figure shows the four parameters and the capture region defined (orange box) relative to the original image.



- **Width**: the width of the capture region
- **Height**: the height of the capture region
- **OffsetX**: the x-coordinate of the upper-left corner of the capture region (the upper-left corner of the original image being (0, 0))
- **OffsetY**: the y-coordinate of the upper-left corner of the capture region



The above four parameters must satisfy the following requirements:

- $(\text{Width} + \text{OffsetX})$ not greater than the width of the original image
- $(\text{Height} + \text{OffsetY})$ not greater than the height of the original image

The width and height of the original image are displayed in the **WidthMax** and **HeightMax** parameters under **Read-only parameters** (**Visibility** level must be set to **Expert** or higher).

4. Perform image capturing again to see the trimming result.



If **Update Image** in the upper right of the **Parameters** tab is checked, the image in the **Canvas** window is automatically updated as you adjust the parameters.

1. Switch to the **Code Generation** tab, click [**Insert Code**] to generate the corresponding code.
2. If you need to set a capture region for the other data type:
 - a. Disconnect from the camera in the current image acquisition assistant.
 - b. Open a new image acquisition assistant, and connect to the camera.
 - c. Select the other data type and repeat the above steps.



- The values of the four capture region parameters are not saved to parameter groups. If the camera is powered off, their values are reset. Please generate the corresponding

code and save the code for future reference.

- The values of these parameters are also reset if the values of the **DeviceScanType** and **Scan3DBinningEnable** parameters are changed.

Comparison of Capture Region and Scan3DROI

Mech-Eye Industrial 3D Camera provides another set of parameters for setting an ROI: **Scan3DROILeft**, **Scan3DROITop**, **Scan3DROIHeight**, and **Scan3DROIWidth** (collectively referred to as "Scan3DROI").

The differences between the capture region parameters and Scan3DROI parameters are summarized below. Please select the set of parameters that suit your needs.

Capture region	Scan3DROI
Not saved to parameter groups, reset if camera powered off	Can be saved to parameter groups
Applicable to 2D image and depth map	Not applicable to 2D image
Image is trimmed	Image not trimmed
Can only be set in HDevelop	Can be set with visualized tool in Mech-Eye Viewer

Adjust Parameters

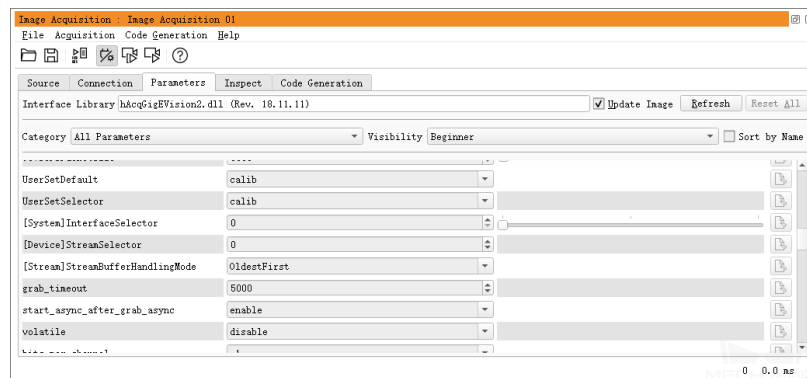
If the quality of the obtained data is unsatisfactory, you can adjust the camera parameters under the **Parameters** tab.



- The camera supports the [parameter group](#) function. By selecting a different parameter group, you can quickly change the parameter configuration for the camera. When adjusting parameters, please select a parameter group for saving the parameter values first.
- You can add or delete a parameter group in Mech-Eye Viewer. After configuring the parameter group in Mech-Eye Viewer, click the **[Refresh]** button in the upper-right corner of the **Parameters** tab in the image acquisition assistant to obtain the latest camera configuration.

To adjust the camera parameters, follow these steps:

1. After the camera is connected, click the **Parameters** tab, set the **UserSetSelector** parameter to the parameter group that you want to modify.




2. Find the **UserSetLoad** parameter, and click **[Apply]** to the right to read in the configuration.

 If parameter values are not updated after you click [**Apply**], please click it again.

3. Find the parameter that you want to adjust and change its value.
4. Find the **UserSetSave** parameter, and click [**Apply**] to the right to save the configuration.
5. Switch to the **Code Generation** tab, click [**Insert Code**] to generate the corresponding code.


References

- Camera parameters supported for GenICam generally correspond with those provided in Mech-Eye Viewer. For the parameter correspondence, descriptions and explanations of camera parameters supported for GenICam, please refer to [Camera Parameters Supported for GenICam](#).
- Setting camera parameters such as **Auto-Exposure ROI**, **Depth Range** and **ROI** usually requires the assistance of visualized tools for several rounds of fine tuning. Since HALCON does not provide visualized tools, you can set these parameters using the visualized tools provided by Mech-Eye Viewer. For detailed information, refer to [Use Mech-Eye Viewer to Adjust Camera Parameters for GenICam Client](#).

 You must disconnect from the camera in HDevelop before you can connect to it in Mech-Eye Viewer and adjust parameters. If you fail to connect the camera in Mech-Eye Viewer, close HDevelop and try again.


Use HALCON Sample

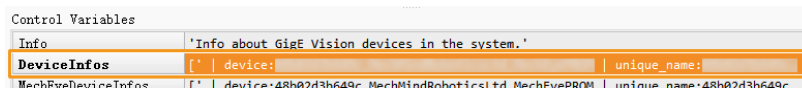
A series of [HALCON samples](#) are provided on GitHub. This section introduces how to use the **connect_to_camera_and_capture_images** sample to connect to a camera, perform image capturing, adjust camera parameters, and save the data. You can run the sample as is or modify it to better suit your needs.

 The HALCON sample is written in C++ and can be executed in the HDevelop integrated development environment directly.

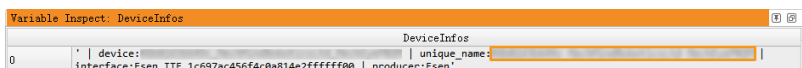
Run the Sample

Please follow these steps to run the sample:

1. [Download the connect_to_camera_and_capture_images sample](#): select Code > Download ZIP.
2. Open the sample in HDevelop: Open HDevelop, and drag the sample into HDevelop.
3. Select the line containing the **info_framegrabber** operator in **Program Windows**, and click  in the toolbar or press the **F6** key to run this line.
4. In the **Control Variables** area, double-click **DeviceInfos** to display a list of all the available cameras.



5. In the list, double-click the camera that you want to connect, and copy the camera name after **unique_name:** or **user_name:**.




 **user_name** is the custom camera name. You can [customize the camera name in Mech-](#)


Eye Viewer.

- Locate the following line, and replace **MechEye** with the copied **unique_name** or **user_name**.

```
DeviceInfo := 'MechEye'
```

- Run the sample by clicking  in the toolbar or pressing the **F5** key.
- If the sample is successfully executed, you can find the saved 2D image and point cloud in the folder where the sample is located. The default file names are **image2d.bmp** and **PointCloud.ply**.



- After executing the entire sample, please click  in the toolbar to reset program execution. Otherwise, the camera cannot be connected by Mech-Eye Viewer.
- When the point cloud is displayed in the **Canvas** window, please click the orange **[Continue]** button in this window to continue the program execution. Otherwise, the program is stuck in the **visualize_object_model_3d** operator and will not proceed.
- If the image capturing process takes too long, you can increase the MTU size of the camera, and [enable jumbo frames](#) on your computer.
- If the network connection is poor, you can reduce data loss by increasing the maximum number of allowed packet gaps with the following operator. Replace **ParameterValues** with the number of allowed packet gaps.

```
set_framegrabber_param (AcqHandle,
 '[Stream]GevStreamMaxPacketGaps', ParameterValues)
```

Adjust Parameters

If the quality of the obtained data is unsatisfactory, you can adjust the camera parameters under the Parameters tab.



- The camera supports the parameter group function; by selecting a different parameter group, you can quickly change the parameter configuration for the camera. Camera parameters must be set and saved under a parameter group.
- You can add or delete a parameter group in Mech-Eye Viewer.

To adjust the camera parameters, follow these steps:

- Use the following operator to obtain a list of available parameter groups.

```
get_framegrabber_param (AcqHandle, 'UserSetSelector_values',
 ParameterValues)
```

- View the list of parameter groups in the **ParameterValues** variable in the **Control Variables** area.
- Use the following operator to select the parameter group to modify. **UserSetSelector** and **UserSetLoad** are the camera parameters used to select and load a parameter group, respectively. Replace **ParameterGroupName** with the actual parameter group name.

```
set_framegrabber_param (AcqHandle, 'UserSetSelector', 'ParameterGroupName')
set_framegrabber_param (AcqHandle, 'UserSetLoad', 'ParameterGroupName')
```

- Use the following operator to obtain the value of the specified parameter. Replace **ParameterName** with the actual camera parameter name. **ParameterValues** is the variable used to save the parameter value, and you can change it according to actual needs. Note that this variable does not need to be included in parentheses.

```
get_framegrabber_param (AcqHandle, 'ParameterName', ParameterValues)
```

- View the parameter value in the **ParameterValues** variable in the **Control Variables** area.
- Use the following operator to adjust the setting of the specified parameter. Replace **ParameterName** with the actual camera parameter name and **NewParameterValue** with the new parameter value.

```
set_framegrabber_param (AcqHandle, 'ParameterName', 'NewParameterValue')
```

- Use the following operator to save the parameter value modification into the parameter group. **UserSetSave** is the camera parameter used to save the parameter group. Replace **ParameterGroupName** with the actual parameter group name.

```
set_framegrabber_param (AcqHandle, 'UserSetSave', 'ParameterGroupName')
```

References

- Camera parameters supported for GenICam generally correspond with those provided in Mech-Eye Viewer. For the parameter correspondence, descriptions and explanations of camera parameters supported for GenICam, please refer to [Camera Parameters Supported for GenICam](#).
- Setting camera parameters such as **Auto-Exposure ROI**, **Depth Range** and **ROI** usually requires the assistance of visualized tools for several rounds of fine tuning. Since HALCON does not provide visualized tools, you can set these parameters using the visualized tools provided by Mech-Eye Viewer. For detailed information, refer to [Use Mech-Eye Viewer to Adjust Camera Parameters for GenICam Client](#).



You must disconnect from the camera in HDevelop before you can connect to it in Mech-Eye Viewer and adjust parameters. If you fail to connect the camera in Mech-Eye Viewer, close HDevelop and try again.

7.2.2. HALCON: Hand-Eye Calibration

This topic introduces how to perform hand-eye calibration in Windows using the HALCON samples provided by Mech-Mind:

- determine_calibration_poses**: used to determine the calibration poses used during the hand-eye calibration.

- `perform_hand_eye_calibration`: used to perform hand-eye calibration.



- Currently, the samples only support hand-eye calibration of 6-axis robots.
- For UHP series, to perform the hand-eye calibration, the **Capture Mode** parameter must be set to **Camera1**.

Prerequisites

Before performing the hand-eye calibration, please make sure that the following requirements are satisfied:

1. The accuracy of the robot is good enough, and the robot can operate properly.
2. Prepare the calibration board that came with the camera, and [mount the calibration board](#).
3. [Download and install the latest version of Mech-Eye SDK](#).
4. Make sure that the quality of the 2D image and depth map [meet the requirements](#).
5. Make sure that the [camera intrinsic parameters are accurate](#).
6. Download the [hand_eye_calibration samples](#): select Code > Download ZIP.

Edit the Samples

After determining the calibration poses with the `determine_calibration_poses` sample, you will use the `perform_hand_eye_calibration` sample to perform the hand-eye calibration.


The following information must be identical in the two samples:

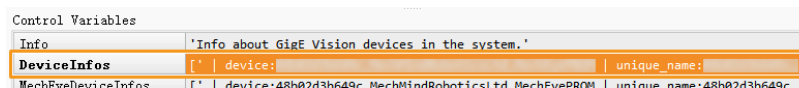
- The camera to be connected
- Calibration board model

Besides, the Euler angle convention in the `robot_pose` JSON file should be set before determining the calibration poses.

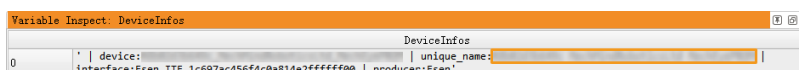
Select the Camera

Before running the samples, you need to set the camera to be connected in the sample to make sure the two samples connect to the same camera. Please follow these steps:

1. Open the sample in HDevelop: Open HDevelop, and drag the sample into HDevelop.
2. Select the line containing the `info_framegrabber` operator in **Program Windows**, and click  in the toolbar or press the **F6** key to run this line.
3. In the **Control Variables** area, double-click `DeviceInfos` to display a list of all the available cameras.



4. In the list, double-click the camera that you want to connect, and copy the camera name after `unique_name:` or `user_name:`.





`user_name` is the custom camera name. You can [customize the camera name in Mech-Eye Viewer](#).

5. Locate the following line, and replace **MechEye** with the copied `unique_name` or `user_name`.

```
DeviceInfo := 'MechEye'
```

Set Calibration Board Model

Please follow these steps to set the calibration board model:

1. Open the sample in HDevelop: Open HDevelop, and drag the sample into HDevelop.
2. Set the calibration board model: The default model is BDB-5. If you are using another model, locate the following operator, and change **BDB-5** to the [calibration board model](#) in use.

```
set_framegrabber_param (AcqHandle, 'BoardType', 'BDB-5')
```

Set Euler Angle Convention and Unit

The calibration poses determined using the **determine_calibration_poses** sample need to be entered into the **robot_pose** JSON file. The default Euler angle convention in this file is **sxyz**, and the default unit is degree. Please follow these steps to set the Euler angle convention:

1. Open the **robot_pose** JSON file.
2. Set the Euler angle convention: locate the following line, and replace **sxyz** with the Euler angle convention of the robot in use. For the Euler angle conventions already supported in the sample, please refer to [Supported Euler Angle Conventions](#).

```
"EulerType" : "sxyz"
```

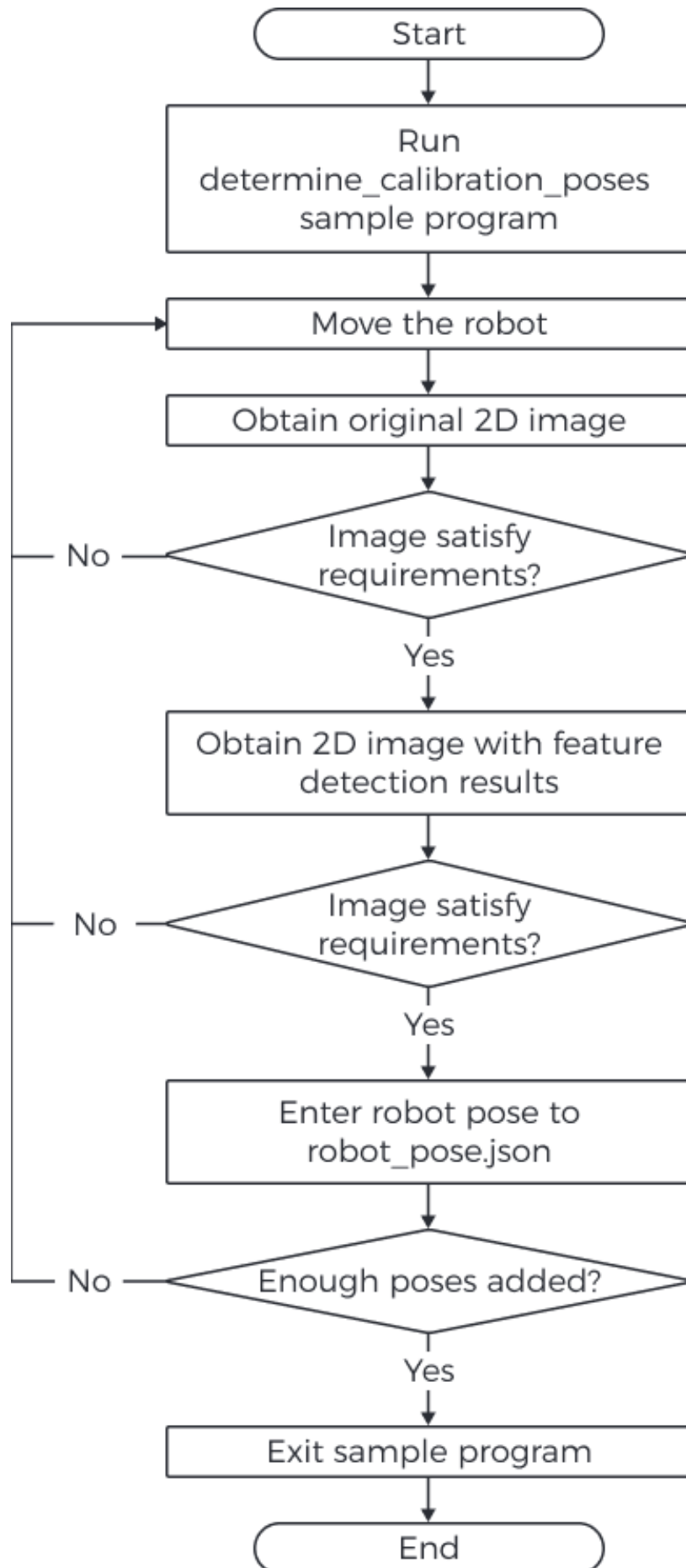
3. Set the Euler angle unit: if you need to enter Euler angles in radians, locate the following line, and replace **true** with **false**.

```
"FromDegree" : true
```

4. Save the **robot_pose** JSON file.


Determine Calibration Poses

Workflow



Instructions

Before performing the hand-eye calibration, you need to determine at least 15 calibration poses. Please follow these steps to determine the calibration poses:

1. Run the sample by clicking  in the toolbar or pressing the **F5** key. When the sample runs to the line **read_char (WindowHandle, Char, ReCode)**, the sample will pause and wait for command input in the following steps.



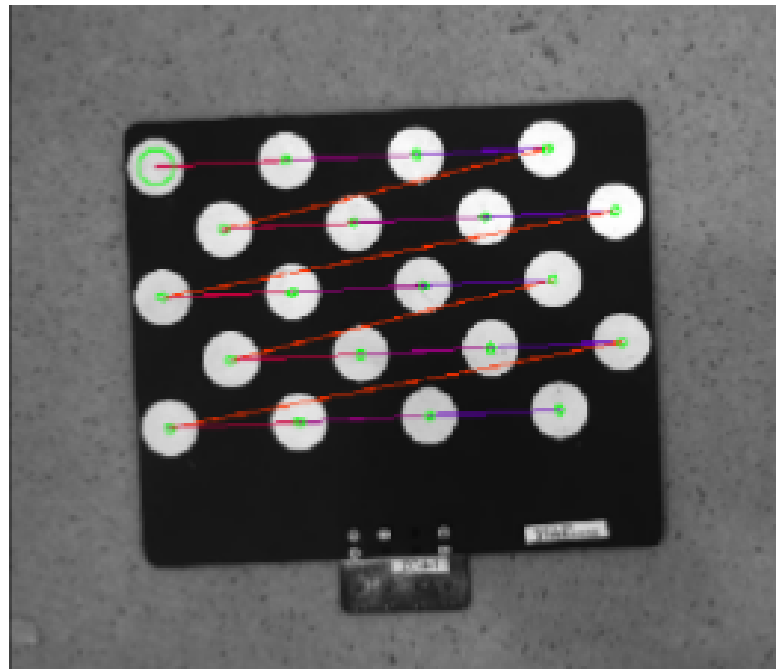
If the camera cannot be connected, please check if it is already connected by Mech-Eye Viewer or another GenICam client.

2. Move the robot to a candidate calibration pose with the teach pendant.



Calibration poses need to satisfy certain requirements. The HALCON Reference provides some guidance: click Help > HALCON Reference in HDevelop and check the **Calibration** chapter.

3. Press **P** to obtain the 2D image from the camera.
 - If the calibration board is not entirely captured, please move the robot and obtain the 2D image again.
 - If the entire calibration board is captured, proceed to the next step.
4. Press **T** to obtain the 2D image containing the feature detection results.
 - If the camera did not detect the circles on the calibration board, no image will be displayed in HDevelop. In this case, please move the robot and obtain the original 2D image and the 2D image containing the feature detection results again.
 - If the camera can detect the circles on the calibration board, a 2D image containing the feature detection results (as shown below) will be displayed in HDevelop. Please proceed to the next step.



5. Check the robot teach pendant, and enter the current robot pose to the **robot_pose** JSON file. The translational component of the pose should be in mm, and the unit of the rotational component (Euler angles) should match the setting in the **robot_pose** JSON file.



It is recommended to save the pose on the teach pendant. When performing the hand-eye calibration, you can use the saved poses to move the robot.

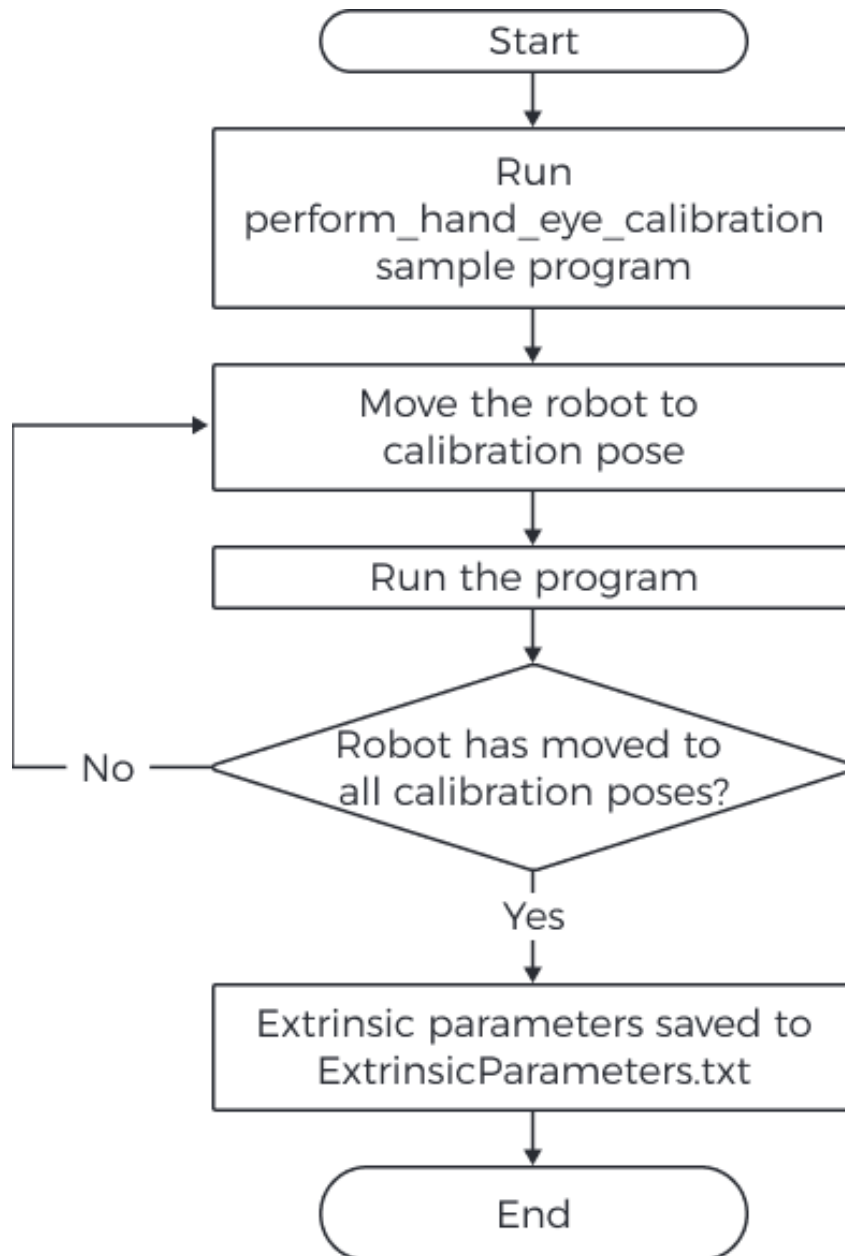
6. Repeat steps 2 to 5 to determine more calibration poses.
7. After determining at least 15 calibration poses, press **Q** to exit the sample.
8. Set the number of calibration poses in the **robot_pose** JSON file: Open the **robot_pose** JSON file, and locate the following line. Replace **15** with the actual number of calibration poses determined.

```
"pose_count" : 15
```

Perform hand-eye calibration

After determining the robot calibration poses, you can perform the hand-eye calibration by running the **perform_hand_eye_calibration** sample.

Workflow



Set Camera Mounting Method

Before performing the hand-eye calibration, please set the camera mounting method in the sample.

The default mounting method is Eye in Hand. If your camera is mounted in the Eye to Hand mode, please locate the following operator, and replace **EyeInHand** with **EyeToHand**.

```
set_framegrabber_param (AcqHandle, 'CalibrationType', 'EyeInHand')
```

Switch Reference Frame


The sample contains an operator that can switch the reference frame in which the point cloud is output from the camera.

The default setting is not to switch the reference frame. If you would like to switch to the robot reference frame, please locate the following operator in the `captureTransformedPointCloud` procedure, and replace `false` with `true`.

```
set_framegrabber_param (AcqHandle, 'Scan3dCoordinateTransformEnable', false)
```


Instructions

Please follow these steps to perform the hand-eye calibration:

1. Run the sample by clicking  in the toolbar or pressing the **F5** key. The sample will stop when it runs to the stop operator.
2. Move the robot to the calibration pose in the `robot_pose` JSON file.



Please move the robot to the calibration poses in the exact order as they are written in the JSON file. Otherwise, the subsequent calculation of the extrinsic parameters will fail.

3. Run the sample by clicking  in the toolbar or pressing the **F5** key. The camera will perform image capturing.
4. In the **Control Variables** area, check the value of **CollectResult**.
 - If **SUCCESS** is displayed, please proceed to the next step.
 - If an error is displayed, please troubleshoot according to the displayed [error code](#), and then determine the calibration poses again.
5. A message saying **Move the robot to the next calibration pose** will be displayed on the image. Please repeat steps 2 to 3.



After the robot has moved to all the calibration poses in the `robot_pose` JSON file, when you run the sample again, the extrinsic parameters will be calculated.

6. In the **Control Variables** area, check the value of **CalibResult**.
 - If **SUCCESS** is displayed, then the hand-eye calibration has succeeded. In the folder where the sample is stored, you can find the **ExtrinsicParameters** TXT file that contains the extrinsic parameters and the obtained point cloud.
 - If an error is displayed, please troubleshoot according to the displayed [error code](#), and then determine the calibration poses again.

Supported Euler Angle Conventions

The conversion from the following Euler angle conventions to quaternions is already supported in the sample.

Common representation	Euler angle convention	Robot brand
Z-Y'-Z''/OAT	rzyz	Kawasaki
Z-Y'-X''/yaw, pitch, roll	rzyx	ABB
		KUKA

Common representation	Euler angle convention	Robot brand
X-Y-Z/WPR	sxyz	FANUC
		YASKAWA
		ROKAE
		UR
X-Y'-Z''	rxyz	/
Z-X'-Z''	rxzx	/



- Even if using the same Euler angle convention, different robot brands might display the Euler angles in different orders. Please input the Euler angles in the order specified in the above table.
- If the Euler angle convention used by your robot is not listed in the above table, you need to add the conversion from this Euler angle convention to quaternions. Please refer to the code of the `euler_to_quad` procedure in the `perform_hand_eye_calibration` sample and add your own conversion.

Hand-Eye Calibration Parameters

This section introduces the parameters used during hand-eye calibration.

BoardType

This parameter is used to set the model of the calibration board in use.

Value list and descriptions:

Value	Description
BDB-5	The recommended distance from the calibration board to the camera is < 0.6 m
BDB-6	The recommended distance from the calibration board to the camera is 0.6–1.5 m
BDB-7	The recommended distance from the calibration board to the camera is > 1.5 m
OCB-005	Only used for Eye-to-Hand projects with high accuracy requirements
OCB-010	
OCB-015	
OCB-020	
CGB-020	The recommended distance from the calibration board to the camera is < 0.6 m
CGB-035	The recommended distance from the calibration board to the camera is 0.6–1.5 m
CGB-050	The recommended distance from the calibration board to the camera is > 1.5 m

ExtrinErrCode

This read-only parameter is used to check the status codes returned during the hand-eye calibration process.

Status code	Description
SUCCESS	Successfully executed.
POSE_INVALID	The pose format is incorrect. Please input quaternions.
IMAGE2D_EMPTY	The 2D image is invalid.
FIND_CORNERS_FAIL	Feature detection of the 2D image failed. Please adjust the parameters for the 2D image .
DEPTH_EMPTY	The depth map is invalid.
CORNERS_3D_INVALID	Feature detection of the depth map failed. Please adjust the parameters for the depth map .
POSES_INSUFFICIENT	Not enough poses are input. Please input at least 15 calibration poses.

7.2.3. HALCON: Obtain Textured Point Cloud

This topic introduces how to obtain the textured point cloud in Windows using the HALCON samples provided by Mech-Mind.

Data Types

This sample obtains the data in the Range and Intensity channels. The data in these channels are described below.

- Range: a 2D image containing the X, Y and Z values of points. This channel cannot be disabled.
- Intensity: The color or monochrome 2D image used to texture the point cloud.


The pixels of the images in the two channels are perfectly aligned and can be used to generate textured point cloud directly.

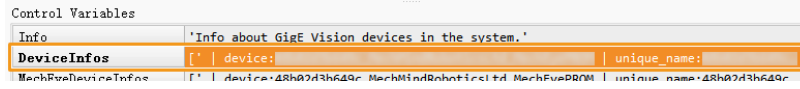


If you need to set capture regions (**Width**, **Height**, **OffsetX** and **OffsetY** parameters) for the images in the two channels, make sure that the parameter values are identical. Otherwise, the pixels of the two images will not be aligned, and the images cannot be used to generate textured point cloud directly.

Obtain Textured Point Cloud

Please follow these steps to obtain the textured point cloud:

1. Download the [obtain_textured_point_cloud sample](#): select Code > Download ZIP.
2. Open the sample in HDevelop: Open HDevelop, and drag the sample into HDevelop.
3. Select the line containing the **info_framegrabber** operator in **Program Windows**, and click  in the toolbar or press the **F6** key to run this line.
4. In the **Control Variables** area, double-click **DeviceInfos** to display a list of all the available cameras.



- In the list, double-click the camera that you want to connect, and copy the camera name after `unique_name:` or `user_name:`.



`user_name` is the custom camera name. You can [customize the camera name in Mech-Eye Viewer](#).

- Locate the following line, and replace `MechEye` with the copied `unique_name` or `user_name`.

```
DeviceInfo := 'MechEye'
```

- When the point cloud is displayed in the **Canvas** window, please click the orange **[Continue]** button in this window to continue the program execution.
- The textured point cloud is stored in the **ObjectModel3D** variable.



For DEEP (V4) and LSR (V4) series, the 2D image (texture) is obtained for texturing the point cloud by default. If you want to use the 2D image (depth source) instead, please locate the following line and activate (uncomment) it.


```
set_framegrabber_param (AcqHandle, 'SourceSelector',  
'Monochrome')
```

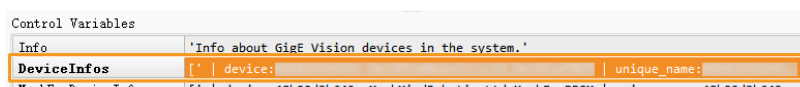
7.2.4. HALCON: Obtain Depth Values Only

This topic introduces how to obtain the depth map that contains only the object depth information in Windows using the HALCON samples provided by Mech-Mind. Obtaining only the depth information can speed up the data transmission and improve efficiency.

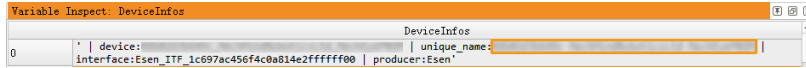
Obtain Depth Map

Please follow these steps to obtain the depth map:

- Download the [obtain_depth_map](#) sample: select Code > Download ZIP.
- Open the sample in HDevelop: Open HDevelop, and drag the sample into HDevelop.
- Select the line containing the `info_framegrabber` operator in **Program Windows**, and click  in the toolbar or press the **F6** key to run this line.
- In the **Control Variables** area, double-click **DeviceInfos** to display a list of all the available cameras.



- In the list, double-click the camera that you want to connect, and copy the camera name after `unique_name:` or `user_name:`.



user_name is the custom camera name. You can [customize the camera name in Mech-Eye Viewer](#).

- Locate the following line, and replace **MechEye** with the copied **unique_name** or **user_name**.

```
DeviceInfo := 'MechEye'
```


- The obtained depth map is stored in the **DepthInM** variable.

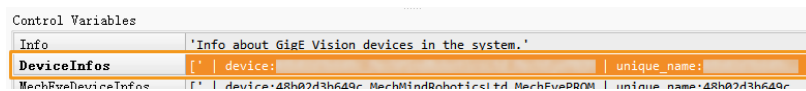
7.2.5. HALCON: Set IP Address

This topic introduces how to obtain and modify the camera's current IP address, subnet mask, and gateway in Windows using the HALCON samples provided by Mech-Mind.

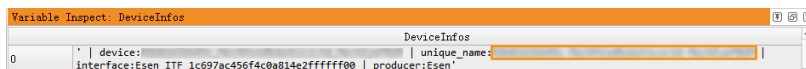
Obtain Sample and Select Camera

Before obtaining or modifying the IP address, you need to download the sample and set the camera to be connected in the sample.

- Download the [configure_camera_ip_address](#) sample: select Code > Download ZIP.
- Open the sample in HDevelop: Open HDevelop, and drag the sample into HDevelop.
- Select the line containing the **info_framegrabber** operator in **Program Windows**, and click  in the toolbar or press the **F6** key to run this line.
- In the **Control Variables** area, double-click **DeviceInfos** to display a list of all the available cameras.



- In the list, double-click the camera that you want to connect, and copy the camera name after **unique_name:** or **user_name:**.




user_name is the custom camera name. You can [customize the camera name in Mech-Eye Viewer](#).

- Locate the following line, and replace **MechEye** with the copied **unique_name** or **user_name**.

```
DeviceInfo := 'MechEye'
```

Obtain Camera IP Address, Subnet Mask and Gateway

Please follow these steps to obtain the current IP address, subnet mask and gateway of the camera:

Run the sample by clicking  in the toolbar or pressing the **F5** key. In the **Control Variables** area, check the following variables for the current IP address, subnet mask and gateway of the camera.


- CurrentIPAddressString: IP address
- CurrentSubnetMaskString: subnet mask
- CurrentDefaultGatewayString: gateway




If the camera firmware version is 2.0.2 or below, the above information cannot be obtained, and the error message will be recorded in the **Exception** variable.

Set Static IP address

This sample sets a static IP address by default. Please follow these steps to modify the IP address, subnet mask and gateway of the camera:


1. Click  in the toolbar to reset program execution.
2. Locate the following line, and change the IP address, subnet mask and gateway to the values to be set:

```
IPAddressString := '192.168.1.100'
SubnetMaskString := '255.255.255.0'
DefaultGatewayString := '192.168.1.1'
```


3. Run the sample by clicking  in the toolbar or pressing the **F5** key.
4. Restart the camera to make the new settings take effect.

Set Dynamic IP Address

Please follow these steps to dynamically assign an IP address to the camera:

1. Click  in the toolbar to reset program execution.
2. Locate the following line, and replace **true** with **false**.

```
set_framegrabber_param (AcqHandle, 'GevCurrentIPConfigurationPersistentIP',
true)
```

3. Run the sample by clicking  in the toolbar or pressing the **F5** key.
4. Restart the camera to make the new settings take effect.



When the camera is connected to the computer directly, the IP address of the camera cannot be dynamically assigned.

7.3. GenICam Reference

This topic provides reference information helpful for using a GenICam client to control the camera.

Camera parameters supported for GenICam clients and their descriptions:

[Camera Parameters Supported for GenICam](#)

Instructions on using Mech-Eye Viewer to adjust camera parameters for a GenICam client:

[Use Mech-Eye Viewer to Adjust Camera Parameters for GenICam Client](#)

Instructions on Obtaining HALCON-readable point clouds through Mech-Eye API:

[Obtain HALCON-Readable Point Cloud through Mech-Eye API](#)

7.3.1. Camera Parameters Supported for GenICam

This topic describes the camera parameters provided by the camera for GenICam, which indicates the features and functions provided by the camera.

Correspondence Between Camera Parameters in GenICam Client and Mech-Eye Viewer



Camera parameters supported for GenICam generally correspond with those provided in Mech-Eye Viewer. Mech-Eye Viewer provides more detailed information such as parameter descriptions and parameter values, which can be helpful for adjusting parameters in GenICam clients.

The following table provides the correspondence between camera parameters in GenICam clients and Mech-Eye Viewer:

Parameter in GenICam Client	Parameter in Mech-Eye Viewer	Visibility	Notes
Scan2DROIHeight	2D Parameters • Auto-Exposure ROI	Beginner	
Scan2DROILeft			
Scan2DROITop			
Scan2DROIWidth			
Scan2DExposureMode	2D Parameters • Exposure Mode • 2D Image (Texture) Exposure Mode	Beginner	
Scan2DExposureTime	2D Parameters • Exposure Time	Beginner	
Scan2DExpectedGrayValue	2D Parameters • Gray Value	Beginner	

Parameter in GenlCam Client	Parameter in Mech-Eye Viewer	Visibility	Notes
Scan2DHDRExposureSequence	2D Parameters <ul style="list-style-type: none"> Exposure Sequence 	Beginner	
Scan2DToneMappingEnable	2D Parameters <ul style="list-style-type: none"> Tone Mapping 	Beginner	
Scan2DSharpenFactor	2D Parameters <ul style="list-style-type: none"> Sharpen Factor 	Guru	
Scan2DPatternRoleExposureMode	2D Parameters <ul style="list-style-type: none"> 2D Image (Depth Source) Exposure Mode 	Beginner	LSR L (V4) DEEP (V4)
Scan2DPatternRoleExposureTime	2D Parameters <ul style="list-style-type: none"> 2D Image (Depth Source) Exposure Time 	Beginner	
Scan3DROIHeight	ROI	Beginner	
Scan3DROILeft			
Scan3DROITop			
Scan3DROIWidth			
ProjectorLightColor	3D Parameters <ul style="list-style-type: none"> Projector <ul style="list-style-type: none"> Light Color 	Guru	Pro M Enhanced (V3) Pro S Enhanced (V3) Log M (V3) Log S (V3)
ProjectorPowerLevel	3D Parameters <ul style="list-style-type: none"> Projector <ul style="list-style-type: none"> Light Brightness 	Expert	DLP cameras
ProjectorFringeCodingMode	3D Parameters <ul style="list-style-type: none"> Projector <ul style="list-style-type: none"> Fringe Coding Mode 	Expert	Nano (V3) Pro XS (V3) PRO M and PRO S (V4) UHP-140
AntiFlickerMode	3D Parameters <ul style="list-style-type: none"> Projector <ul style="list-style-type: none"> Anti-Flicker Mode 	Expert	PRO M and PRO S (V4) NANO (V4)

Parameter in GenICam Client	Parameter in Mech-Eye Viewer	Visibility	Notes
ProjectorSelectionMode	3D Parameters <ul style="list-style-type: none"> • Projector <ul style="list-style-type: none"> ◦ Projector Selection Mode 	Guru	Deep (V3)
ProjectorSelector	3D Parameters <ul style="list-style-type: none"> • Projector <ul style="list-style-type: none"> ◦ Projector Selector 	Guru	Pro L Enhanced (V3)
LaserFringeCodingMode	3D Parameters <ul style="list-style-type: none"> • Laser <ul style="list-style-type: none"> ◦ Fringe Coding Mode 	Expert	
LaserPowerLevel	3D Parameters <ul style="list-style-type: none"> • Laser <ul style="list-style-type: none"> ◦ Laser Power 	Expert	
LaserFrameAmplitude	3D Parameters <ul style="list-style-type: none"> • Laser Projection Frame Control <ul style="list-style-type: none"> ◦ Laser Frame Amplitude 	Guru	Laser L (V3) Laser L Enhanced (V3)
LaserFrameOffset	3D Parameters <ul style="list-style-type: none"> • Laser Projection Frame Control <ul style="list-style-type: none"> ◦ Laser Frame Offset 	Guru	LSR L (V4) DEEP (V4)
LaserFramePartitionCount	3D Parameters <ul style="list-style-type: none"> • Laser Projection Frame Control <ul style="list-style-type: none"> ◦ Laser Frame Partition Count 	Guru	
Scan3DBinningEnable	3D Parameters <ul style="list-style-type: none"> • 3D Binning 	Guru	Pro L Enhanced (V3) Laser L Enhanced (V3)
Scan3DExposureCount	3D Parameters <ul style="list-style-type: none"> • Exposure Multiplier 	Beginner	
Scan3DExposureTime	3D Parameters <ul style="list-style-type: none"> • Exposure Time 	Beginner	

Parameter in GenICam Client	Parameter in Mech-Eye Viewer	Visibility	Notes
Scan3DExposureTime2	3D Parameters • Exposure Time 2	Beginner	
Scan3DExposureTime3	3D Parameters • Exposure Time 3	Beginner	
Scan3DGain	3D Parameters • Gain	Expert	
UhpCaptureMode	3D Parameters • UHP ◦ Capture Mode	Beginner	UHP-140
AcquisitionMode	 and 	-	
DepthLowerLimit	Depth Range	Beginner	
DepthUpperLimit			
DeviceScanType	-	-	
PointCloudSurfaceSmoothing	Point Cloud Processing • Surface Smoothing	Beginner	Available for camera firmware 2.1.0 and above
PointCloudOutlierRemoval	Point Cloud Processing • Outlier Removal	Beginner	
PointCloudNoiseRemoval	Point Cloud Processing • Noise Removal	Expert	
PointCloudEdgePreservation	Point Cloud Processing • Edge Preservation	Guru	
CloudOutlierFilterMode	Point Cloud Processing • Noise Removal	Beginner	Available for camera firmware 2.0.2 and below
CloudSmoothMode	Point Cloud Processing • Point Cloud Smoothing	Beginner	
FringeContrastThreshold	Point Cloud Processing • Stripe Contrast Threshold	Beginner	
FringeMinThreshold	Point Cloud Processing • Minimum Fringe Intensity Threshold	Guru	
UserSetDefault	-	-	

Parameter in GenlCam Client	Parameter in Mech-Eye Viewer	Visibility	Notes
UserSetSelector	Parameter Group drop-down menu	-	

7.3.2. Use Mech-Eye Viewer to Adjust Camera Parameters for GenlCam Client

Setting camera parameters such as **Auto-Exposure ROI**, **Depth Range** and **ROI** usually requires the assistance of visualized tools for several rounds of fine tuning. Since GenlCam-compliant third-party machine vision software (GenlCam clients) does not provide visualized tools, you can set these parameters by using the visualized tools provided in Mech-Eye Viewer.

After setting these parameters using Mech-Eye Viewer, you can read the latest camera configuration in a GenlCam client directly.

Preparation

Before you adjust parameters in Mech-Eye Viewer, complete the following steps:

1. Save the changes that you made in the GenlCam client.
2. Disconnect from the camera in the GenlCam client.
3. [Connect to the camera in Mech-Eye Viewer.](#)
4. In the **Parameters** tab on the right of Mech-Eye Viewer, select the parameter group to which you want to save the parameter values.



A GenlCam client can read parameter groups but not add or delete parameter groups. For instructions on adding/deleting a parameter group, please refer to [Configure Parameter Group in Mech-Eye Viewer.](#)

Adjust Parameters

Below you can find instructions on setting **Auto-Exposure ROI**, **Depth Range** and **ROI**, respectively.

Set Auto-Exposure ROI

To set **Auto-Exposure ROI**, follow these steps:

1. In the **Parameters** tab on the right of Mech-Eye Viewer, locate the **2D Parameters** category.
2. From the drop-down menu of **Exposure Mode**, select **Auto**.
3. Double-click [**Edit**] to the right of **Auto-Exposure ROI** to open the **Set ROI** window.
4. Set **Auto-Exposure ROI** in the pop-up window. For detailed instructions, please refer to [Set Auto-Exposure ROI](#). The correspondence between **Auto-Exposure ROI** parameters in a GenlCam client and those in Mech-Eye Viewer is as follows:

Parameter in GenlCam Client	Parameter in Mech-Eye Viewer
Scan2DROILeft	Upper-left coordinates: x
Scan2DROITop	Upper-left coordinates: y
Scan2DROIHeight	Dimensions: Height
Scan2DROIWidth	Dimensions: Width

Set Depth Range

To set **Depth Range**, follow these steps:

1. In the **Parameters** tab on the right of Mech-Eye Viewer, locate the **Depth Range** category.
2. Double-click [**Edit**] to the right of **Depth Range** to open the **Set Depth Range** window.
3. Set **Depth Range** in the pop-up window. For detailed instructions, please refer to [Set Depth Range](#).

The correspondence between **Depth Range** parameters in a GenICam client and those in Mech-Eye Viewer is as follows:

Parameter in GenICam Client	Parameter in Mech-Eye Viewer
DepthLowerLimit	Depth Range: Depth Lower Limit
DepthUpperLimit	Depth Range: Depth Upper Limit

Set ROI

To set **ROI**, follow these steps:

1. In the **Parameters** tab on the right of Mech-Eye Viewer, locate the **ROI** category.
2. Double-click [**Edit**] to the right of **ROI** to open the **Set ROI** window.
3. Set **ROI** in the pop-up window. For detailed instructions, please refer to [Set ROI](#).

The correspondence between **ROI** parameters in a GenICam client and those in Mech-Eye Viewer is as follows:

Parameter in GenICam Client	Parameter in Mech-Eye Viewer
Scan3DROILeft	Upper-left coordinates: x
Scan3DROITop	Upper-left coordinates: y
Scan3DROIHeight	Dimensions: Height
Scan3DROIWidth	Dimensions: Width

Read Altered Values into GenICam Client

After setting **Auto-Exposure ROI**, **Depth Range** and **ROI** in Mech-Eye Viewer, complete the following steps to ensure that the GenICam client can read the modified parameter values.

1. In the **Parameters** tab on the right of Mech-Eye Viewer, click [**Save**] at the top or press **Ctrl** + **S** on the keyboard to save the parameter values into the parameter group.
2. Disconnect from the camera in Mech-Eye Viewer.
3. Connect to the camera in the GenICam client.
4. In the GenICam client, select and load the parameter group with the **UserSetSelector** and **UserSetLoad** parameters.

7.3.3. Obtain HALCON-Readable Point Cloud through Mech-Eye API

Through Mech-Eye API, you can obtain point clouds faster than through HALCON, and it is possible to obtain textured point clouds directly. Mech-Eye API provides a [C++ sample named CaptureHalconPointCloud](#). By running this sample, you can obtain untextured and textured point clouds through Mech-Eye API and convert the data to HALCON-readable format. Then, you can read the point clouds into HALCON and perform further processing.

This sample can be used on Windows and Ubuntu operating system.



This sample depends on the HALCON/C++ interface. Please confirm that your HALCON license is still valid before using.

Use Sample on Windows

In order to use this sample on Windows, the following software must be installed:

- Mech-Eye SDK
- Cmake
- Visual Studio
- Halcon

For instructions on installing the above software, as well as instructions on building and running this sample individually, please refer to [Instructions on Using Mech-Eye API C++ Samples \(Windows\)](#).

After running the sample, you can read the obtained point clouds into HALCON with the `read_object_model_3d` operator.

Use Sample on Ubuntu

In order to use this sample on Ubuntu, the following software must be installed:

- Mech-Eye SDK
- Cmake
- PCL
- Halcon

For instructions on installing the above software, as well as instructions on building and running this sample individually, please refer to [Instructions on Using Mech-Eye API C++ Samples \(Ubuntu\)](#).

After running the sample, you can read the obtained point clouds into HALCON with the `read_object_model_3d` operator.

8. Camera Hardware

8.1. Camera Models

This topic introduces different camera models, as well as the characteristics and suitable applications of each model.

Basic Information

The currently available camera models are divided into the V3 and V4 generations. The following tables list the basic information of the two generations of models.



- You can use [FOV Calculator](#) to calculate the FOV of the models at different working distances.
- If you need to download the PDF hardware user manual and technical specifications, or obtain the camera 3D models, installation drawings, accessory information, product flyers, etc., please visit [Mech-Mind Download Center](#).

V4 Cameras

Model	2D image color	Material code	Object focal distance (mm)	Recommended working distance (mm)
UHP-140	Monochrome	UHP-140-MP30D300M	300	280–320
NANO	Monochrome	NANO-V4D350M	350	300–450
		NANO-V4D550M	550	450–600
	Color	NANO-V4D350C	350	300–450
		NANO-V4D550C	550	450–600
PRO S	Monochrome	PRO S-V4D500M	500	500–600
		PRO S-V4D700M	700	600–800
		PRO S-V4D1000M	1000	800–1000
	Color	PRO S-V4D500C	500	500–600
		PRO S-V4D700C	700	600–800
		PRO S-V4D1000C	1000	800–1000
PRO M	Monochrome	PRO M-V4D1200M	1200	1000–1300
		PRO M-V4D2000M	2000	1300–2000
	Color	PRO M-V4D1200C	1200	1000–1300
		PRO M-V4D2000C	2000	1300–2000
LSR L	Color ⁽¹⁾	LSR L-V4D1500A	1500	1200–1800
		LSR L-V4D3000A	3000	1800–3000

Model	2D image color	Material code	Object focal distance (mm)	Recommended working distance (mm)
LSR S	Color ⁽¹⁾	LSR S-V4D800A	800	500–900
		LSR S-V4D1400A	1400	900–1500
DEEP	Color	DEEP-V4D3000A	3000	1200–3500

(1) Refers to the color of the 2D image (texture). For detailed descriptions, please refer to [Data Types](#).



For more technical specifications, please refer to [Technical Specifications](#).

V3 Cameras

Model	2D image color	Object focal distance (mm)	Recommended working distance (mm)
Pro XS	Monochrome	350	300–450
		550	450–650
Log S	Color	700	500–750
		1000	750–1000
Log M	Color	1200	1000–1300
		2000	1300–2000
Laser L Enhanced	Monochrome	1500	1200–1700
		3000	1700–3000



For more technical specifications, please refer to [Technical Specifications of V3 Cameras](#).

Characteristics and Suitable Applications

The following table summarizes the characteristics and suitable applications of different camera models.

Model	Characteristics	Suitable applications
UHP-140 (V4)	Micron-level accuracy, independently developed image fusion algorithm. High performance for reflective objects.	Inspection/measurement applications of positional error, gap, and flush in industries such as automotive part manufacturing and assembly.
LSR L (V4) Laser L Enhanced (V3)	High accuracy, large field of view, high resistance to ambient light.	Manufacturing plants and other environments with ambient light interference.
NANO (V4) Pro XS (V3)	Small volume, high accuracy, high resistance to ambient light.	On-robot mounting and high-accuracy applications such as locating and assembly and high-precision picking.

Model	Characteristics	Suitable applications
DEEP (V4)	Large field of view, deep depth of field, fast data acquisition, comprehensive, detailed, precise and color-accurate point cloud.	Logistics applications such as depalletizing and palletizing.
PRO S (V4)	High accuracy, fast data acquisition, high resistance to ambient light, color variant available.	Medium-range applications with high accuracy requirements, such as random picking, positioning, assembly, academic research.
PRO M (V4)		
Log M (V3)	Fast data acquisition, short to medium range.	Designed for logistics applications, suited for piece picking and parcel feeding.
Log S (V3)		

8.2. Hardware User Manual

Safety Instructions

- To ensure safe use, please do not use the product before reading this manual and becoming familiar with the correct usage. Improper use and maintenance may damage the product or cause other hazards. Mech-Mind shall not be liable for any injury or damage brought upon the user or any third party due to improper use and maintenance.
- Following the instructions and warnings in this manual can lower risks, but cannot eliminate risks.
- Every step has been inspected during the drafting of this manual. Please do not hesitate to contact Mech-Mind if you find any mistakes in the manual or have any questions.
- The product is to be mounted, connected, used, and maintained by trained adults only. To ensure safe operation, the product should be transported, stored, mounted, connected, used, and maintained properly.
- Laser is hazardous; please familiarize yourself with hazard prevention before using the camera.

Precautions for Mounting and Using the Camera

- It is PROHIBITED to place any explosive or flammable substances near the camera. Do not expose the product to open fire or high temperature. Do not place the product in fire or crush the product mechanically. Doing so may cause explosion.
- Do not throw, drop, or collide the camera with other objects. The camera may be damaged by strong shock or vibration. Do not repair or disassemble the camera by yourself. Mech-Mind shall not be liable for any damage caused by repair or disassembly not performed by Mech-Mind.
- Do not allow any foreign object, such as metal pieces, dust, paper, and wood chips, to enter the camera. Doing so may lead to fire, electric shock, malfunction, etc.
- Do not use the camera at extremely high or low temperature. For the operating temperature of the camera, please refer to Technical Specifications.
- Please use the camera indoors (except for LSR S).
- Please use the camera at elevations below 4000 meters.
- Do not stare into the light beam when using the camera.

- Please install the camera in a well-circulated, open place.

Check the Camera

- Before using, please check the camera carefully for damage, signs of water entry, suspicious odor, smoke, loose or damaged bolts, etc., and make sure that the camera is in proper working conditions. If any of the above abnormalities occur, please disconnect the power immediately.
- High temperature will age the power cable. Please check the power cable regularly for signs of aging. If the cable has aged, please contact Mech-Mind to acquire a replacement.

Usage of Adapter / DIN Rail Power Supply

- Do not use the product if the power port, power adapter / DIN rail power supply, or power socket is wet.
- Do not heat or put the adapter / DIN rail power supply and power cable in fire.
- Please use a 24 V isolated adapter / DIN rail power supply with an output power of 36 W (NANO and PRO XS) or 90 W (other models).
- Please use the specified voltage. Failure to do so may lead to fire, electric shock, malfunction, etc. The adapter / DIN rail power supply and power cable should be properly grounded. It is recommended to use the isolated adapter / DIN rail power supply provided by Mech-Mind.
- Power plugs should be properly grounded. Do not place the adapter / DIN rail power supply in a place that is hard to disconnect the power.
- Please install the DIN rail power supply inside a control cabinet.
- Use copper conductors only.

Precautions for Using Laser Cameras

- DO NOT look into the laser beam or the reflected laser beam directly. Do not look into the laser beam with optical instruments. Doing so may lead to eye injuries. DO NOT direct the laser at other people.
- The laser beam must be lower or higher than and never at eye level.
- Be cautious of the path of the laser beam. If there is a possibility that personnel may be exposed to the specular or diffuse reflections, block the beam by installing a protective enclosure.
- Do not enter the areas that the laser / reflected laser output reaches. Do not place any reflective light object (mirror, glass, metal, etc.) in the path of the laser beam.
- After the product is powered on, wait at least 30 minutes before using the product. Otherwise, data quality may be unstable. While adjusting the parameters, do not turn off the power. Doing so may lead to loss of some or all parameter adjustments.
- LASER ENERGY - EXPOSURE NEAR APERTURE MAY CAUSE BURNS.
- NOTE: The risk of skin injury is only likely for highly divergent beams for exposure close to the aperture.

Notice for Disposal

- Please comply with local laws and regulations when disposing of the camera to avoid polluting the environment. Do not dispose of the old battery in domestic waste. Do not dispose of the product irresponsibly. Improper disposal may pollute the environment.

- The pollution degree of the product is 3.

Certifications

Mech-Eye Industrial 3D Camera is compliant with the following standards and assessment requirements. Please note that the certification statuses may be updated. For more information, please contact the local sales agents.

Compliant with the following requirements and standards:

- European Electromagnetic Compatibility Standards.
- U.S. ANSI C63.4 and 47 CFR PART 15B
- Canada ICES-003
- Japan VCCI-CISPR 32:2016
- South Korea KS C 9832 and KS C 9835

Safety of Laser Products

The laser classification is implemented based on IEC 60825-1:2014 in accordance with the requirements of Laser Notice No. 56 of the FDA (CDRH).

CE



The full text of the EU Declaration of Conformity is available at <https://downloads.mech-mind.com/?tab=tab-eu-dec>

European Electromagnetic Compatibility Standards

- EN 55032:2015+A1:2020+A11:2020
- EN IEC 61000-3-2:2019+A1:2021
- EN 61000-3-3:2013+A1:2019+A2:2021
- EN 55035:2017+A11:2020



All products bearing this symbol are waste electrical and electronic equipment (WEEE as in directive 2012/19/EU) which should not be mixed with unsorted household waste. Instead, you should protect human health and the environment by handing over your waste equipment to a

designated collection point for the recycling of waste electrical and electronic equipment, appointed by the government or local authorities. Correct disposal and recycling will help prevent potential negative consequences to the environment and human health. Please contact the local authorities for more information about the location as well as terms and conditions of such collection points.

FCC



NOTE: This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference in which case the user will be required to correct the interference at his own expense.

This device complies with part 15 of the FCC Rules. Operation is subject to the following two conditions: (1) This device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation of the device.

“Le présent appareil est conforme aux CNR d’Industrie Canada applicables aux appareils radio exempts de licence. L’exploitation est autorisée aux deux conditions suivantes : (1) l’appareil ne doit pas produire de brouillage, et (2) l’utilisateur de l’appareil doit accepter tout brouillage radioélectrique subi, même si le brouillage est susceptible d’en compromettre le fonctionnement.”

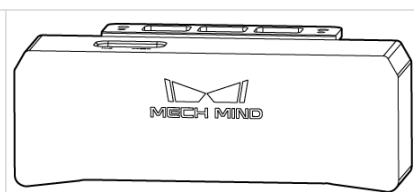
VCCI


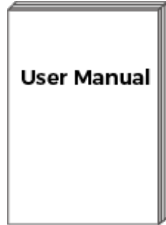
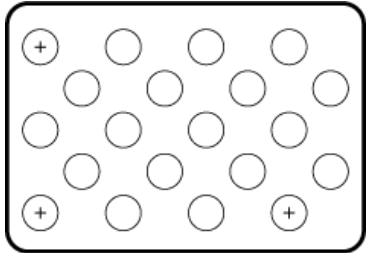




This is a Class A equipment. Operation of this equipment in a residential environment could cause radio interference. In such a case, the user may be required to take corrective actions.

VCCI-A

Package Contents

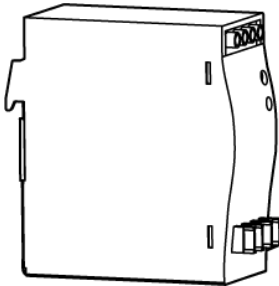
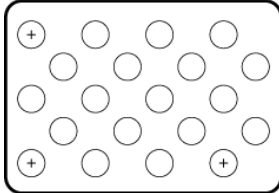
Camera	
--------	--

<p>Accessory bag</p>	
<p>User manual</p>	
<p>Calibration board (UHP-140 only)</p>	
<p>DC power cable (20 m) CBL-PWR-20M-LU</p>	
<p>Ethernet cable (20 m) CBL-ETH-20M-LU</p>	



- Before using, please make sure that the package is intact, the camera is not damaged, and no accessories are missing. Please contact Mech-Mind if any items are damaged or missing.
- If you need cables of other lengths, please contact Mech-Mind.
- For the technical specifications of the cables, please contact Mech-Mind.
- The calibration board is an optional accessory for models other than UHP-140.

Optional Accessories

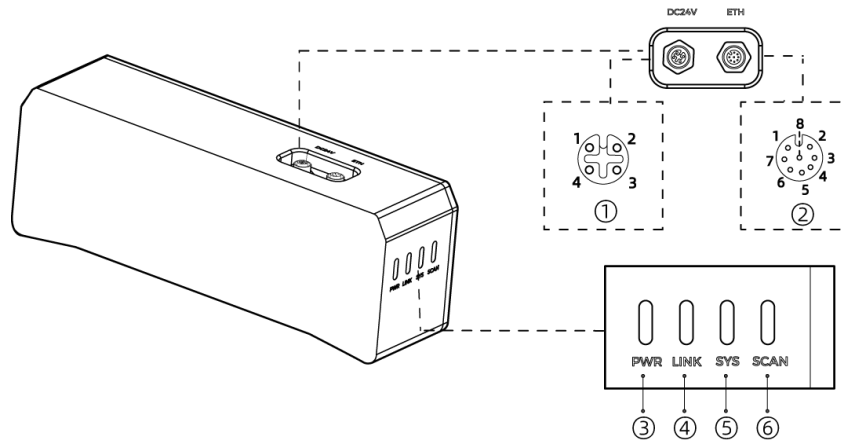
<p>DIN rail power supply</p>	
<p>Calibration board (except for UHP-140)</p>	



For the technical specifications of the DIN rail power supply or calibration board, please contact Mech-Mind.

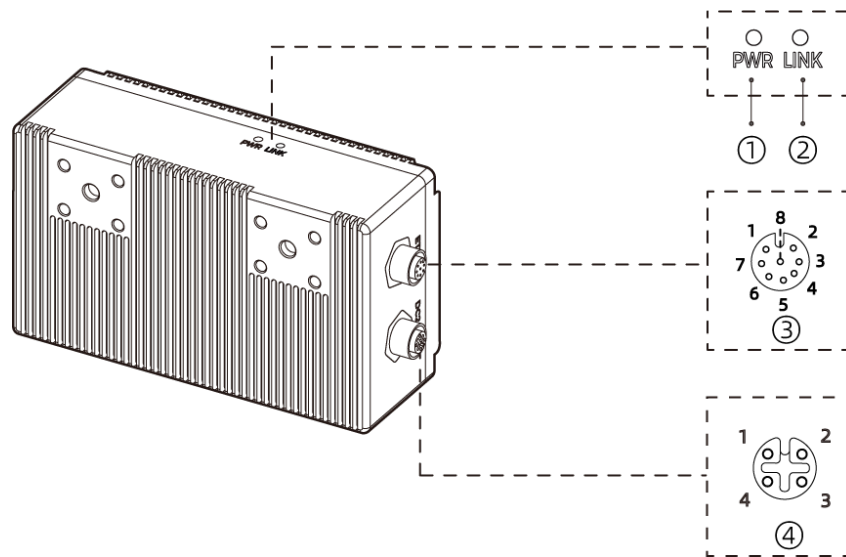
Functional Diagrams


PRO S / PRO M, DEEP, UHP-140, LSR S, and LSR L



No.	Name	Function	
①	DC 24V port	1: GND	2: GND
		3: 24 VDC	4: 24 VDC
②	ETH port	1: MD3_P	2: MD2_N
		3: MD2_P	4: MD0_P
		5: MD1_P	6: MD0_N
		7: MD3_N	8: MD1_N
③	PWR indicator light	Off: not connected to power	
		Green: normal voltage	
		Yellow: abnormal voltage, but system is still functioning	
		Red: voltage too high or too low, system power is disconnected	
④	LINK indicator light	Off: not connected to network	
		Blinking green: data being transmitted	
		Solid green: no ongoing data transmission	
⑤	SYS indicator light	Off: system not started	
		Solid green: system starting up	
		Blinking green: system operating properly	
		Blinking yellow: system operating with error but still functioning	
⑥	SCAN indicator light	Blinking red: system not operating	
		Solid: capturing and processing in progress	
		Off: not capturing or processing	

NANO and PRO XS



 The above figure uses NANO as an example.

No.	Name	Function
①	PWR indicator light	Off: not connected to power Green: normal voltage
②	LINK indicator light	Off: not connected to network Blinking green: data being transmitted Solid green: no ongoing data transmission
③	ETH port	1: MD3_P 2: MD2_N 3: MD2_P 4: MD0_P 5: MD1_P 6: MD0_N 7: MD3_N 8: MD1_N
④	DC 24V port	1: GND 2: GND 3: 24 VDC 4: 24 VDC

Mounting the Camera

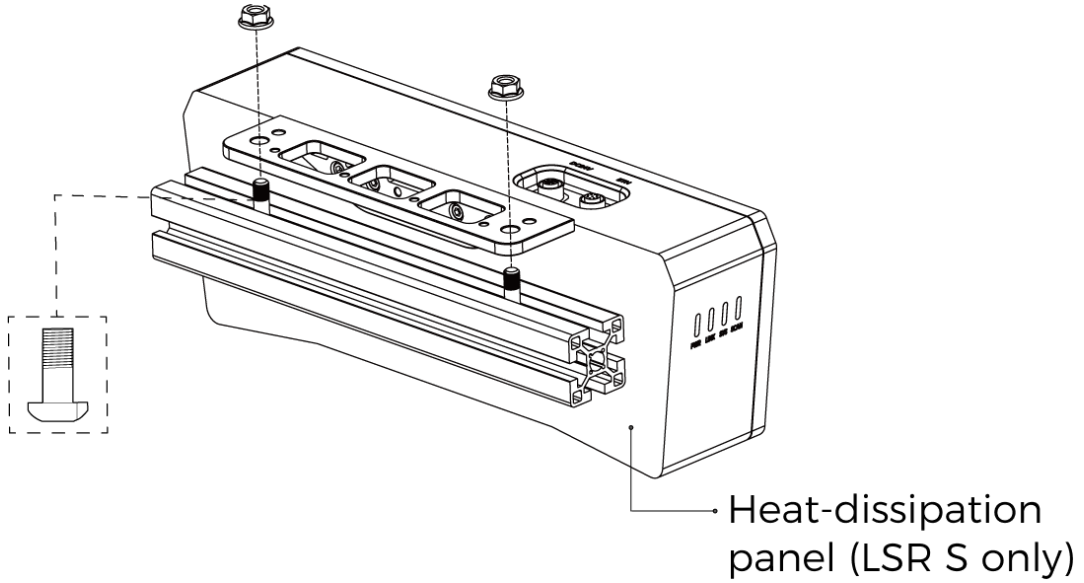
 For camera dimensions, please refer to [Technical Specifications](#). Please prepare a wrench.

Mount through the Camera Bracket

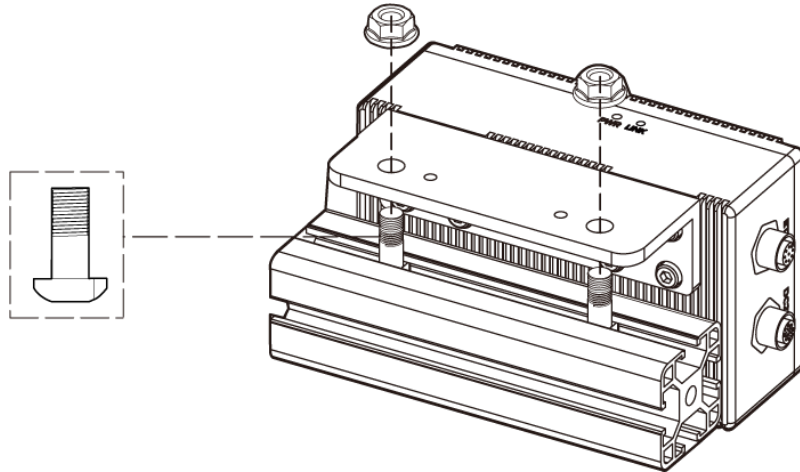
 The camera bracket and the heat-dissipation panel (LSR S only) are attached to the back of the camera when packaged.

As shown below, tighten the two bolts with the wrench.

- PRO S / PRO M, DEEP, UHP-140, LSR S, and LSR L:



- NANO and PRO XS:



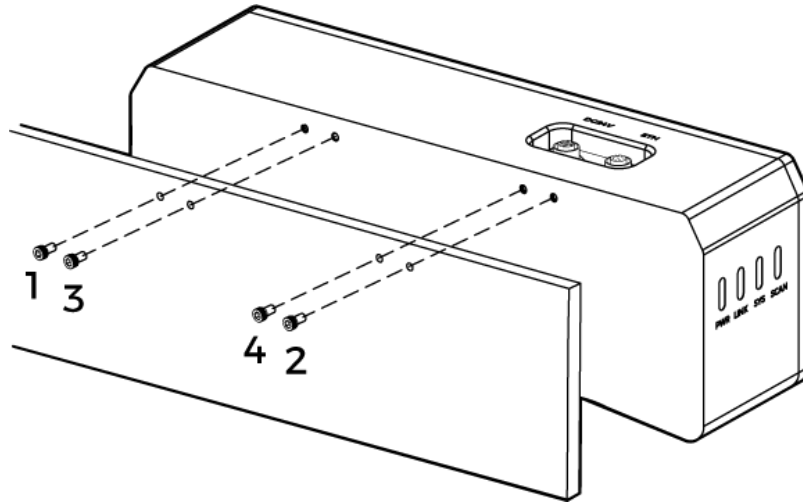
Mount through Threaded Holes on the Back



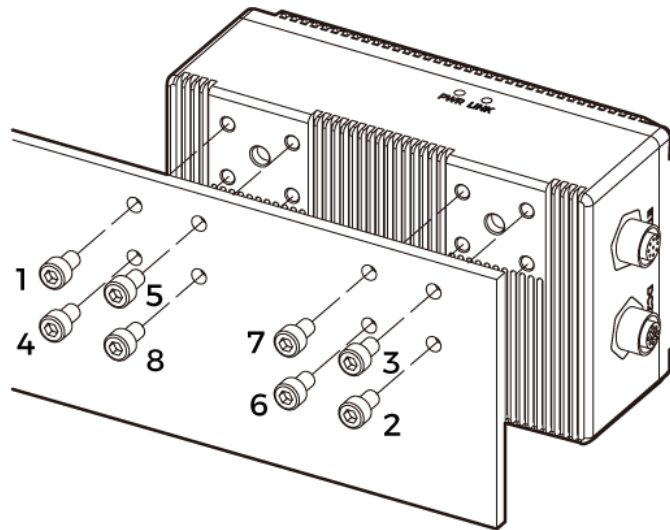
- Please remove the camera bracket and heat-dissipation panel (LSR S only) before mounting.
- When mounting LSR S using this method, ensure that the back of the camera is in close contact with a metal surface for heat dissipation, so that the camera does not overheat and malfunction.

As shown below, use the wrench to loosely screw on the bolts in the specified order, and then fully tighten all the bolts in the specified order.


- PRO S / PRO M, DEEP, UHP-140, LSR S, and LSR L:



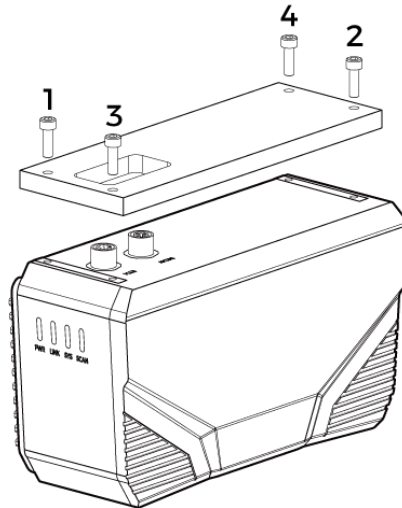
- NANO and PRO XS:



Mount through Threaded Holes at the Top (LSR S and UHP-140)

-  Please remove the camera bracket before mounting.

As shown below, use the wrench to loosely screw on the bolts in the specified order, and then fully tighten all the bolts in the specified order.



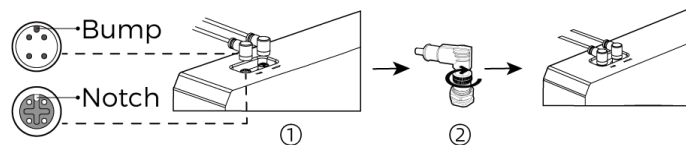
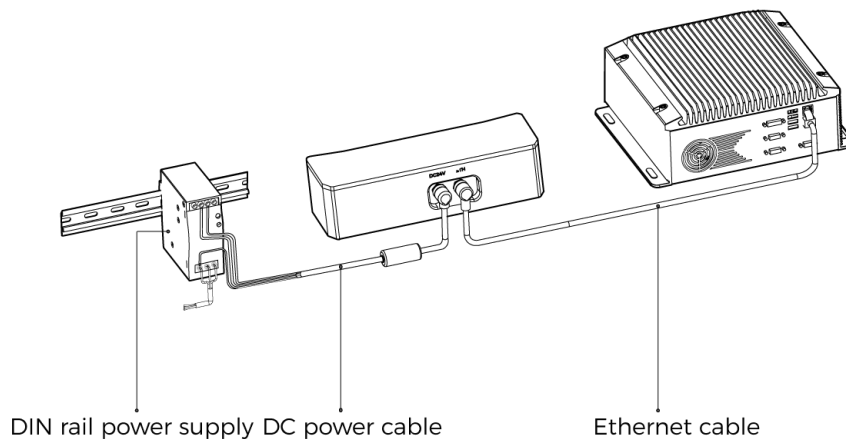
Connect the Camera

Ethernet Cable

As shown below, insert the M12-A connector of the Ethernet cable into the ETH port on the camera, and insert the RJ45 connector into the Ethernet port on the IPC.

DC Power Cable

As shown below, insert the M12-A connector of the DC power cable into the DC 24V port on the camera.



When inserting the Ethernet cable and DC power cable:

1. Align the bump in the connector with the notch in the port.
2. Tighten the nut. The recommended tightening torque is 0.7 N·m. A gap of about 2 mm remains after the nut is fully tightened.



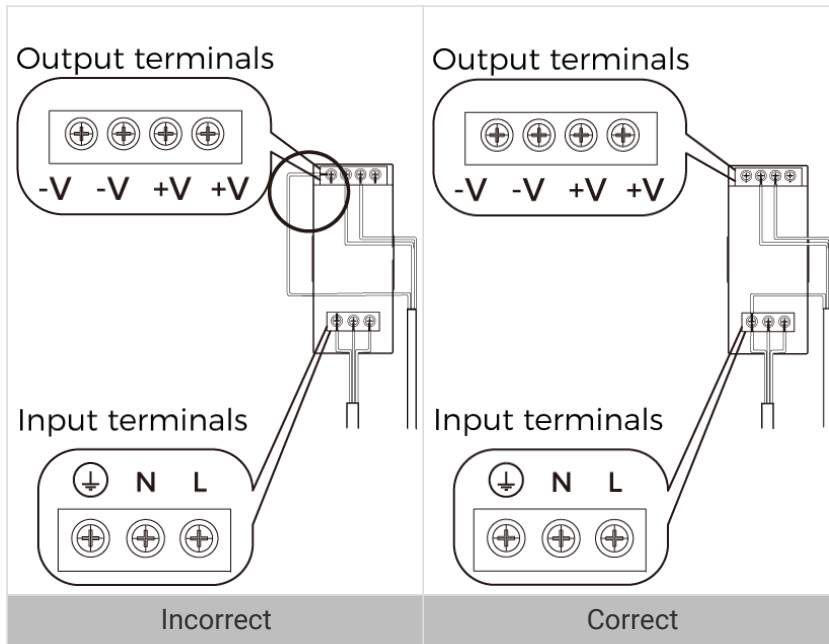
- Network switch can be used to connect the camera and IPC.

- Supply power after all cables are fully connected. After power is supplied, the PWR indicator light should be solid green. If not, please contact Mech-Mind.
- When mounting the camera onto a robot arm or other mobile equipment, please fasten the DC power and Ethernet cables on the camera end properly to avoid damaging the cables or connectors due to strain.

DIN Rail Power Supply



- Please prepare the AC power cable.
- The instructions below apply to the DIN rail power supply provided by Mech-Mind. If you use your own DIN rail power supply, please refer to its user manual for connection instructions.



The AC power cable has three wires: L, N, and PE (⊥).

The DC power cable has three wires: +V, -V and PE (⊥).

When wiring the DIN rail power supply, wires of the power cable must be inserted to the corresponding input/output terminals.

- Please mount the DIN rail power supply inside a control cabinet.
- The DIN rail power supply or the DIN rail should be reliably grounded. If mounting multiple DIN rail power supplies on the same DIN rail, ensure enough distance in between.
- Please use a single-phase, 3-prong power outlet with protective grounding for the AC power.

Maintenance

Cleaning

When cleaning the surface of the product, please use a clean soft cloth to gently wipe off the dust and debris. When cleaning the lenses, to avoid scratching, you can use a clean, soft lint-free cloth

with lens cleaner or glass cleaner to carefully wipe the lenses.



- Do not clean the camera with corrosive or volatile solvents such as alcohol, gasoline and kerosene. These substances may damage the exterior and internal structure of the camera.
- Do not use a pressure washer gun or hose to spray and wash the product. Mech-Mind shall not be liable for any damage or loss caused by liquid damage to the product.

Storage

- LSR S: The product is rated as IP67. The enclosure of the product can prevent dust and water from entering and affecting the functions of the product. Please avoid soaking the product in water or placing it outdoors for an extended period of time. When not using, please store the product in an indoor, dry, cool, and well ventilated place. The storage temperature of the product is -20–60°C.
- Models other than LSR S: The product is rated as IP65. The enclosure of the product can prevent dust from entering and affecting the functions of the product. Please avoid soaking the product in water, installing it in a highly humid environment, or placing it outdoors for an extended period of time. When not using, please store the product in an indoor, dry, cool, and well ventilated place. The storage temperature of the product is -20–60°C.



- Disconnect the camera from power when storing to avoid fire.
- Do not point the lens directly at the sun or other intense light source. Intense light may cause the image sensor to deteriorate and produce white blur in images.

Disclaimer

It is strongly recommended to use the power supply and cables provided by Mech-Mind to ensure compliance with the safety and EMC standards. Mech-Mind shall not be liable for any issues caused by using the power supply and cables provided by a third party.

Trademark and Legal Statement

Mech-Mind and Mech-Mind series logos including  are registered trademarks of Mech-Mind Robotics Technologies Ltd. and other related entities.

© Copyright 2024, Mech-Mind Robotics Technologies Ltd.

Unless authorized in writing in advance by Mech-Mind Robotics Technologies Ltd. (“Mech-Mind”), no part of the trademarks shall be used, reproduced, modified, transmitted, transcribed, or used or sold with other products as a bundle by any entity or individual in any form for any reason.

Any infringement of Mech-Mind’s trademark rights will be punished in accordance with the law.

Mech-Mind reserves all rights regarding this user manual. According to copyright laws, unless authorized by Mech-Mind, this user manual shall not be reproduced, modified, or issued in part or in its entirety by any entity or individual. Users who purchased and use the product may download, print, or copy the user manual for personal use or use inside the belonging organization. Unless authorized by Mech-Mind, the contents of the user manual may not be used for any other purposes.

8.3. Technical Specifications

The following topics provide the technical specifications of the V4 models:

- [NANO](#)
- [PRO S / PRO M](#)
- [UHP-140](#)
- [DEEP](#)
- [LSR S](#)
- [LSR L](#)

For the technical specifications of V3 models, please refer to [Technical Specifications of V3 Cameras](#).

8.3.1. NANO

Technical Specifications

Product name	Mech-Eye Industrial 3D Camera	
Model	NANO	
Object focal distance ⁽¹⁾	350 mm	550 mm
Recommended working distance	300–450 mm	450–600 mm
FOV (near)	220 × 150 mm @ 0.3 m	
FOV (far)	440 × 300 mm @ 0.6 m	
Resolution	1280 × 1024	
Pixels	1.3 MP	
Point Z-value repeatability (σ) ⁽²⁾	0.1 mm @ 0.5 m	
Measurement accuracy (VDI/VDE) ⁽³⁾	0.1 mm @ 0.5 m	
Weight	0.7 kg	
Baseline	68 mm	
Dimensions	145 × 51 × 85 mm	
Typical capture time	0.6–1.1 s	
2D image color	Monochrome	Color
Light source	Blue LED (459 nm, RG2)	White LED (RG2)
Operating temperature	0–45°C	
Communication interface	Gigabit Ethernet	
Input	24 VDC, 1.5 A	
Safety and EMC	CE/FCC/VCCI/KC/ISED/NRTL	
IP rating ⁽⁴⁾	IP65	
Cooling	Passive	

(1) The camera is available in two object focal distances, each corresponding to a different range of recommended working distance.

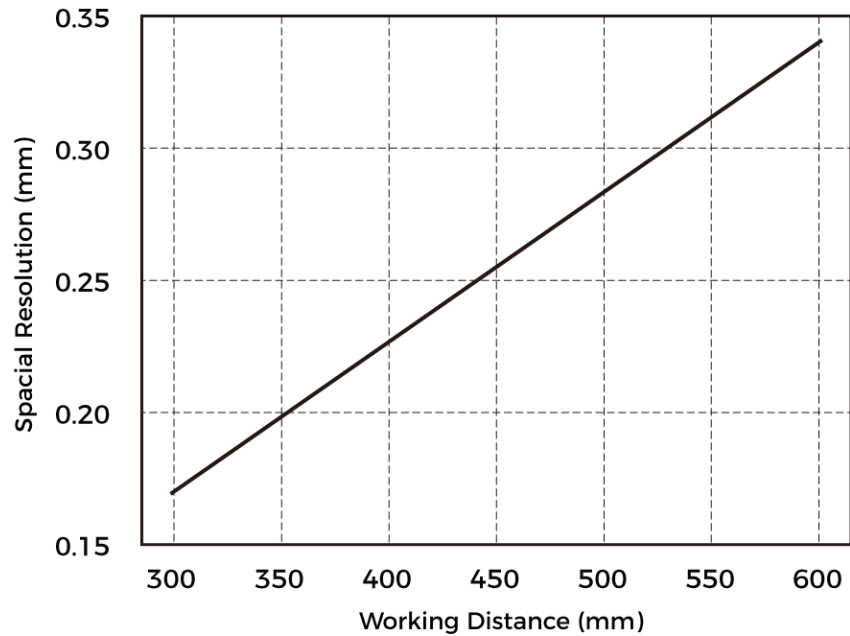
(2) One standard deviation of 100 Z-value measurements of the same point. The measurement

target was a ceramic plate.

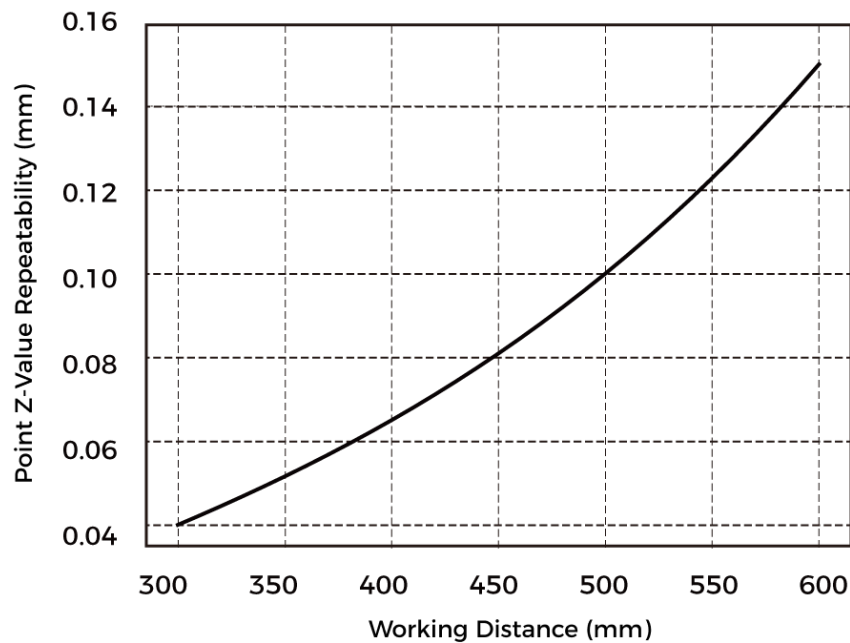
(3) According to VDI/VDE 2634 Part II.

(4) Test implemented based on IEC 60529. 6: dust-tight; 5: protected against water jets.

Spatial Resolution

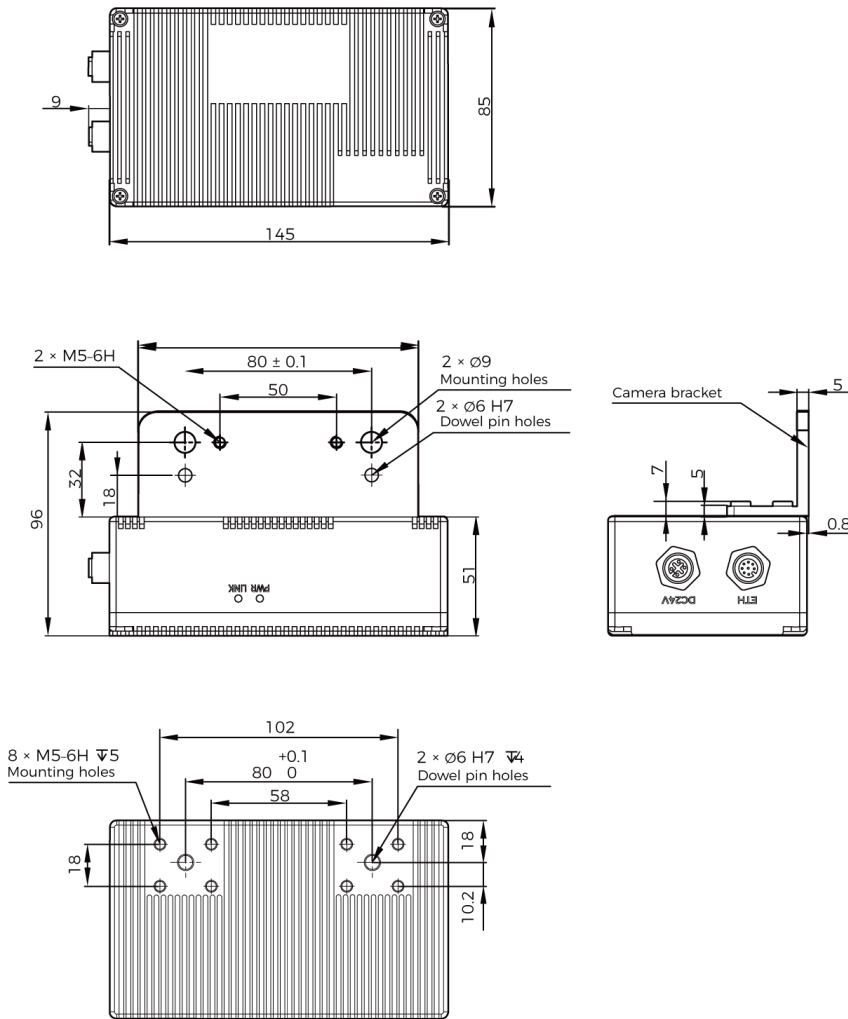


Point Z-Value Repeatability



Dimensions

Unit: mm



CE

Hereby [Mech-Mind Robotics Technologies Ltd.] declares that [NANO] is in compliance with the Electromagnetic Compatibility Directive 2014/30/EU.

The full text of the EU Declaration of Conformity is available at <https://downloads.mech-mind.com/?tab=tab-eu-dec>

8.3.2. PRO S / PRO M

Technical Specifications

Product name	Mech-Eye Industrial 3D Camera		
Model	PRO S		
Object focal distance ⁽¹⁾	500 mm	700 mm	1000 mm
Recommended working distance	500–600 mm	600–800 mm	800–1000 mm
FOV (near)	370 × 240 mm @ 0.5 m		
FOV (far)	800 × 450 mm @ 1 m		
Resolution	1920 × 1200		

Pixels	2.3MP	
Point Z-value repeatability (σ) ⁽²⁾	0.05 mm @ 1 m	
Measurement accuracy (VDI/VDE) ⁽³⁾	0.1 mm @ 1 m	
Weight	1.6 kg	
Baseline	180 mm	
Dimensions	265 × 57 × 100 mm	
Typical capture time	0.3–0.6 s	
2D image color	Monochrome	Color
Light source	Blue LED (459 nm, RG2)	White LED (RG2)
Operating temperature	0–45°C	
Communication interface	Gigabit Ethernet	
Input	24 VDC, 3.75 A	
Safety and EMC	CE/FCC/ISED/VCCI/UKCA/KC	
IP rating ⁽⁴⁾	IP65	
Cooling	Passive	

(1) The camera is available in three object focal distances, each corresponding to a different range of recommended working distance.

(2) One standard deviation of 100 Z-value measurements of the same point. The measurement target was a ceramic plate.

(3) According to VDI/VDE 2634 Part II.

(4) Test implemented based on IEC 60529. 6: dust-tight; 5: protected against water jets.

Product name	Mech-Eye Industrial 3D Camera	
Model	PRO M	
Object focal distance ⁽¹⁾	1200 mm	2000 mm
Recommended working distance	1000–1300 mm	1300–2000 mm
FOV (near)	800 × 450 mm @ 1 m	
FOV (far)	1500 × 890 mm @ 2 m	
Resolution	1920 × 1200	
Pixels	2.3MP	
Point Z-value repeatability (σ) ⁽²⁾	0.2 mm @ 2 m	
Measurement accuracy (VDI/VDE) ⁽³⁾	0.2 mm @ 2 m	
Weight	1.9 kg	
Baseline	270 mm	
Dimensions	353 × 57 × 100 mm	
Typical capture time	0.3–0.6 s	

2D image color	Monochrome	Color
Light source	Blue LED (459 nm, RG2)	White LED (RG2)
Operating temperature	0–45°C	
Communication interface	Gigabit Ethernet	
Input	24 VDC, 3.75 A	
Safety and EMC	CE/FCC/VCCI/KC/ISED/NRTL	
IP rating ⁽⁴⁾	IP65	
Cooling	Passive	

(1) The camera is available in two object focal distances, each corresponding to a different range of recommended working distance.

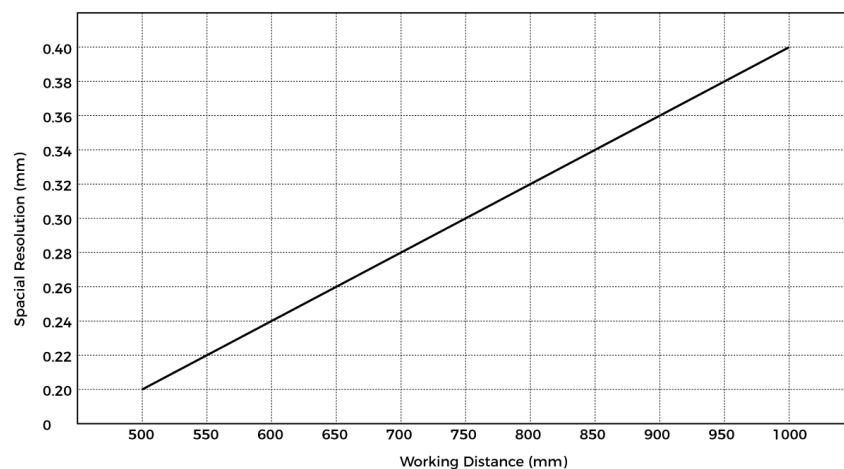
(2) One standard deviation of 100 Z-value measurements of the same point. The measurement target was a ceramic plate.

(3) According to VDI/VDE 2634 Part II.

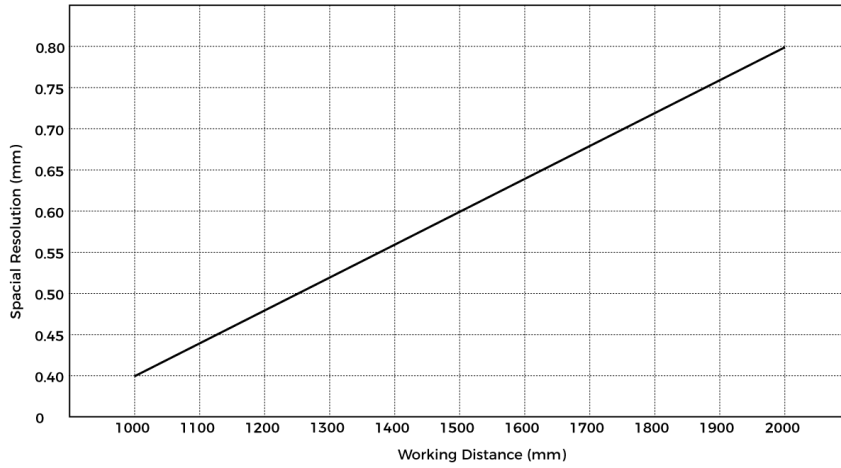
(4) Test implemented based on IEC 60529. 6: dust-tight; 5: protected against water jets.

Spatial Resolution

PRO S

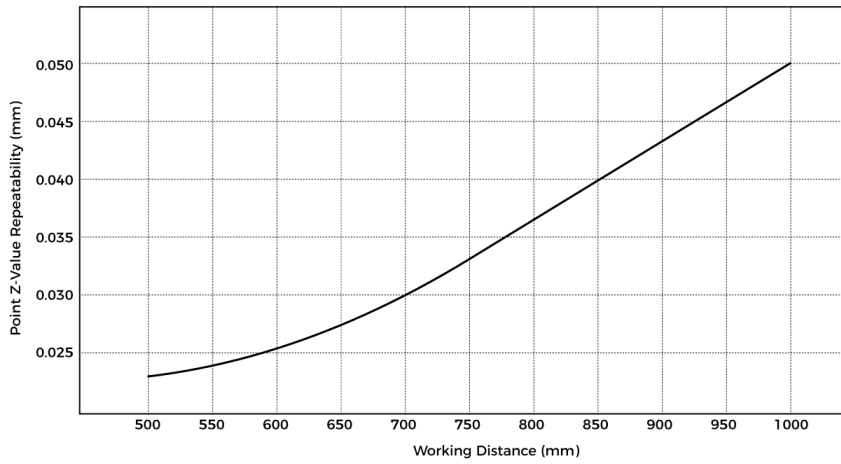


PRO M

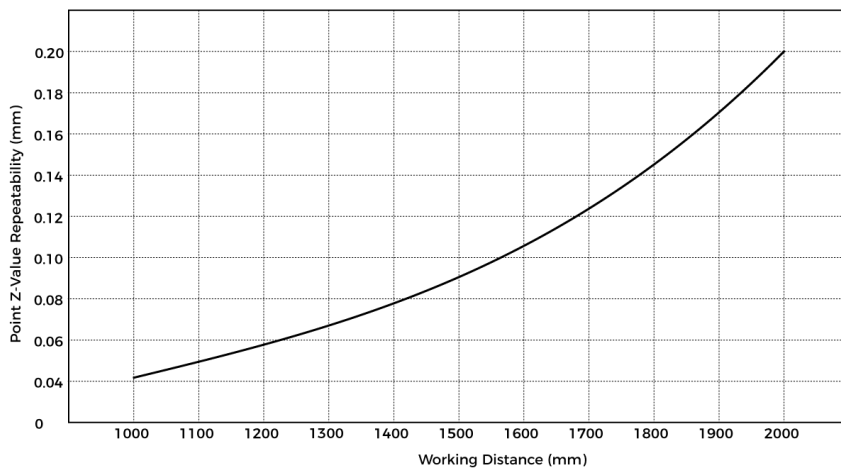


Point Z-Value Repeatability

PRO S



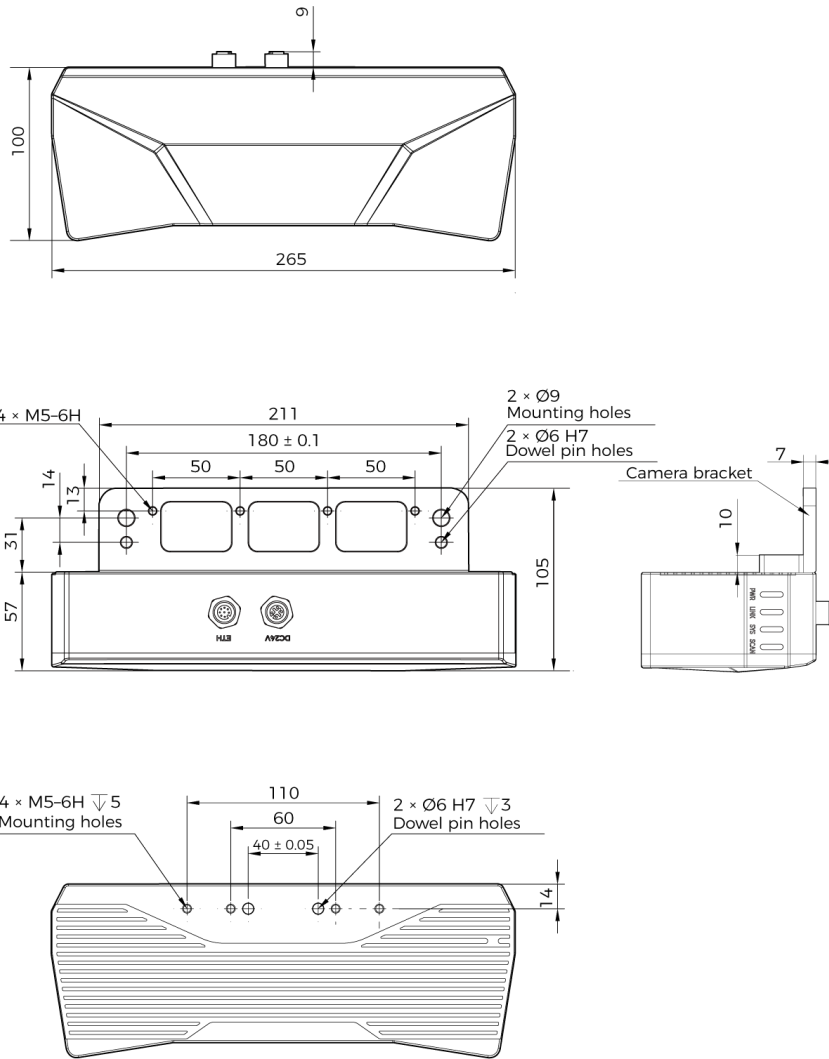
PRO M



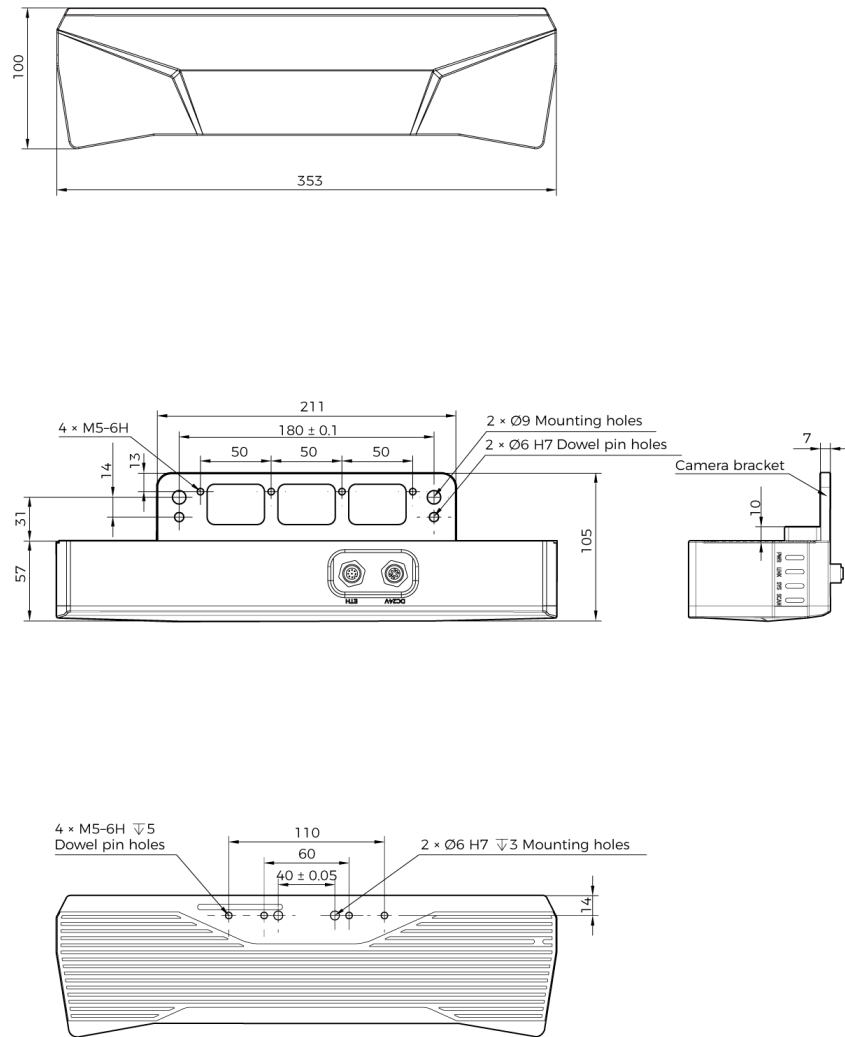
Dimensions

Unit: mm

PRO S



PRO M



CE

Hereby [Mech-Mind Robotics Technologies Ltd.] declares that [PRO S] and [PRO M] are in compliance with the Electromagnetic Compatibility Directive 2014/30/EU.

The full text of the EU Declaration of Conformity is available at <https://downloads.mech-mind.com/?tab=tab-eu-dec>

8.3.3. UHP-140

Technical Specifications

Product name	Mech-Eye Industrial 3D Camera
Model	UHP-140
Object focal distance	300 mm
Recommended working distance	280–320 mm
FOV (near)	135 × 90 mm @ 280 mm
FOV (far)	150 × 100 mm @ 320 mm
Resolution	2048 × 1536

Product name	Mech-Eye Industrial 3D Camera
Pixels	3MP
Point Z-value repeatability (σ) ⁽¹⁾	2.6 μm @ 0.3 m
Regional Z-value repeatability (σ) ⁽²⁾	0.09 μm @ 0.3 m
Measurement accuracy (VDI/VDE) ⁽³⁾	0.03 mm @ 0.3 m
Typical capture time	0.6–0.9 s
Weight	Approx. 1.9 kg
Baseline	Approx. 80 mm
Dimensions	Approx. 260 × 65 × 142 mm
2D image color	Monochrome
Light source	Blue LED (459 nm, RG2)
Operating temperature	0–45°C
Communication interface	Gigabit Ethernet
Input	24 VDC, 3.75 A
Safety and EMC	CE/FCC/VCCI/KC/ISED/NRTL
IP rating ⁽⁴⁾	IP65
Cooling	Passive

(1) One standard deviation of 100 Z-value measurements of the same point. The measurement target was a ceramic plate.

(2) One standard deviation of 100 measurements of the difference between the Z-value means of two same-sized regions. The measurement target was a ceramic plate.

(3) According to VDI/VDE 2634 Part II.

(4) Test implemented based on IEC 60529. 6: dust-tight; 5: protected against water jets.

Dimensions

Unit: mm

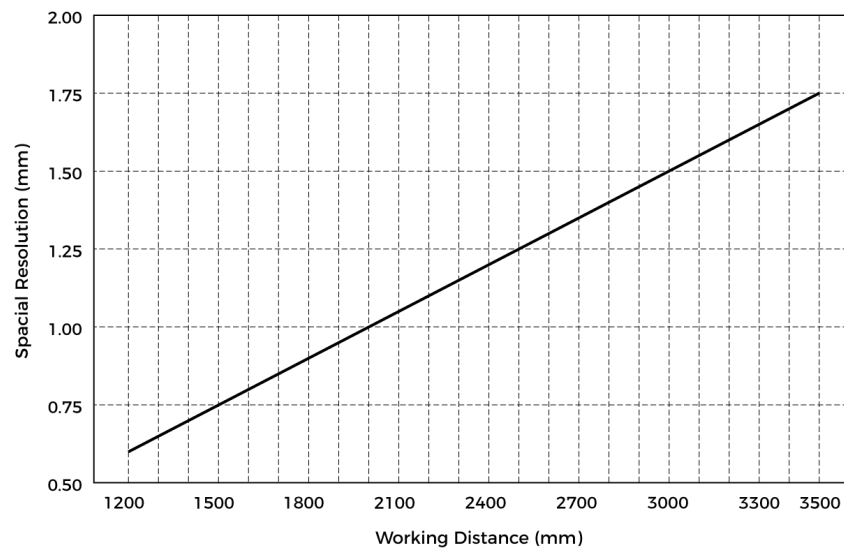
RGB resolution	2000 × 1500
Point Z-value repeatability (σ) ⁽¹⁾	1.0 mm @ 3 m
Measurement accuracy (VDI/VDE) ⁽²⁾	3.0 mm @ 3 m
Weight	2.4 kg
Baseline	300 mm
Dimensions	366 × 77 × 92 mm
Typical capture time	0.5–0.9 s
Light source	Red laser (638 nm, Class 2)
Operating temperature	-10–45°C
Communication interface	Gigabit Ethernet
Input	24 VDC, 3.75 A
Safety and EMC	CE/FCC/VCCI/KC/ISED/NRTL
IP rating ⁽³⁾	IP65
Cooling	Passive

(1) One standard deviation of 100 Z-value measurements of the same point. The measurement target was a ceramic plate.

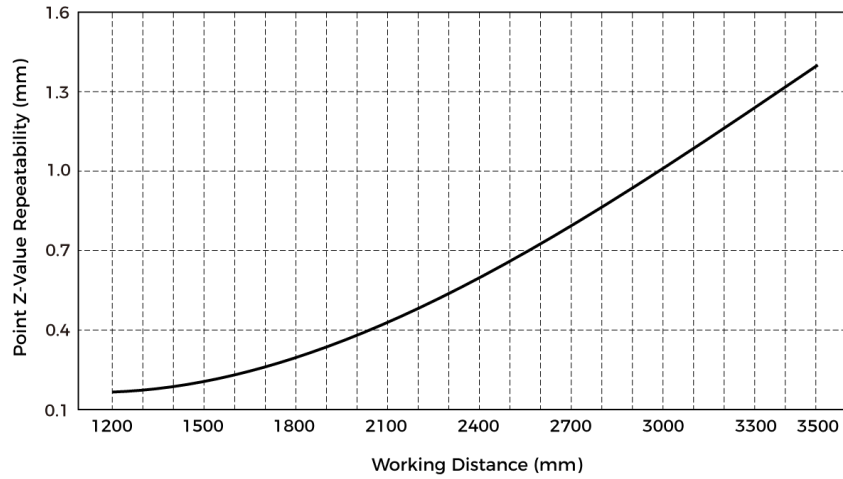
(2) According to VDI/VDE 2634 Part II.

(3) Test implemented based on IEC 60529. 6: dust-tight; 5: protected against water jets.

Spatial Resolution

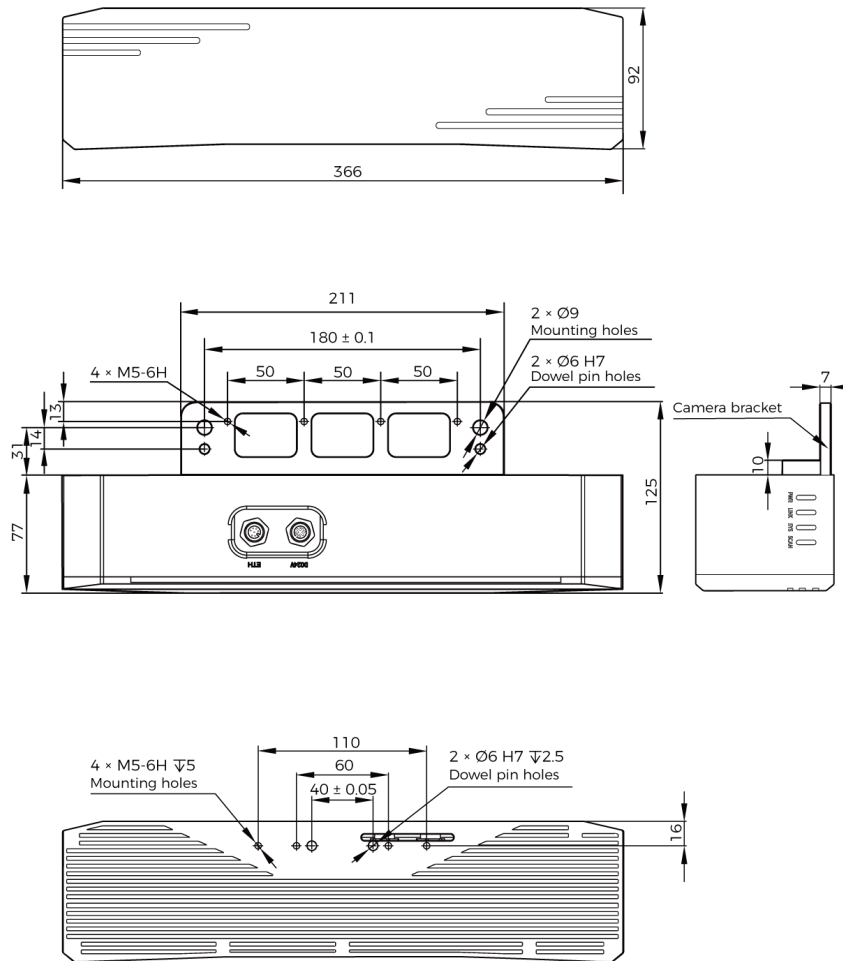


Point Z-Value Repeatability



Dimensions

Unit: mm



Safety of Laser Products

The laser classification is implemented based on IEC 60825-1:2014 in accordance with the requirements of Laser Notice No. 56 of the FDA (CDRH).

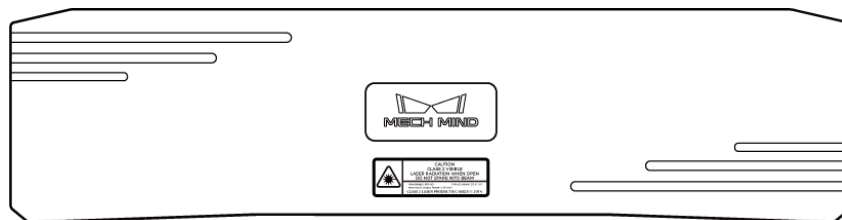
Model	Wavelength	Max. output power	Laser class
-------	------------	-------------------	-------------

DEEP	638 nm	2.46 mW	Class 2
------	--------	---------	---------

Label



Warning Label Attachment



CE

Hereby [Mech-Mind Robotics Technologies Ltd.] declares that [DEEP] is in compliance with the Electromagnetic Compatibility Directive 2014/30/EU.

The full text of the EU Declaration of Conformity is available at <https://downloads.mech-mind.com/?tab=tab-eu-dec>

8.3.5. LSR S

Technical Specifications

Product name	Mech-Eye Industrial 3D Camera	
Model	LSR S	
Object focal distance ⁽¹⁾	800 mm	1400 mm
Recommended working distance	500–900 mm	900–1500 mm
FOV (near)	480 × 360 mm @ 0.5 m	
FOV (far)	1500 × 1200 mm @ 1.5 m	
Depth map resolution	2048 × 1536	
RGB resolution	4000 × 3000 / 2000 × 1500	
Point Z-value repeatability (σ) ⁽²⁾	0.2 mm @ 1.5 m	
Measurement accuracy (VDI/VDE) ⁽³⁾	1.0 mm @ 1.5 m	
Weight ⁽⁴⁾	1.9 kg	
Baseline	140 mm	

Dimensions ⁽⁴⁾	228 × 77 × 126 mm
Typical capture time	0.5–0.9 s
Light source	Red laser (638 nm, Class 2)
Operating temperature ⁽⁵⁾	-10–45°C
Communication interface	Gigabit Ethernet
Input	24 VDC, 3.75 A
Safety and EMC	CE/FCC/VCCI/KC/ISED/NRTL
IP rating ⁽⁶⁾	IP67
Cooling	Passive

(1) The camera is available in two object focal distances, each corresponding to a different range of recommended working distance.

(2) One standard deviation of 100 Z-value measurements of the same point. The measurement target was a ceramic plate.

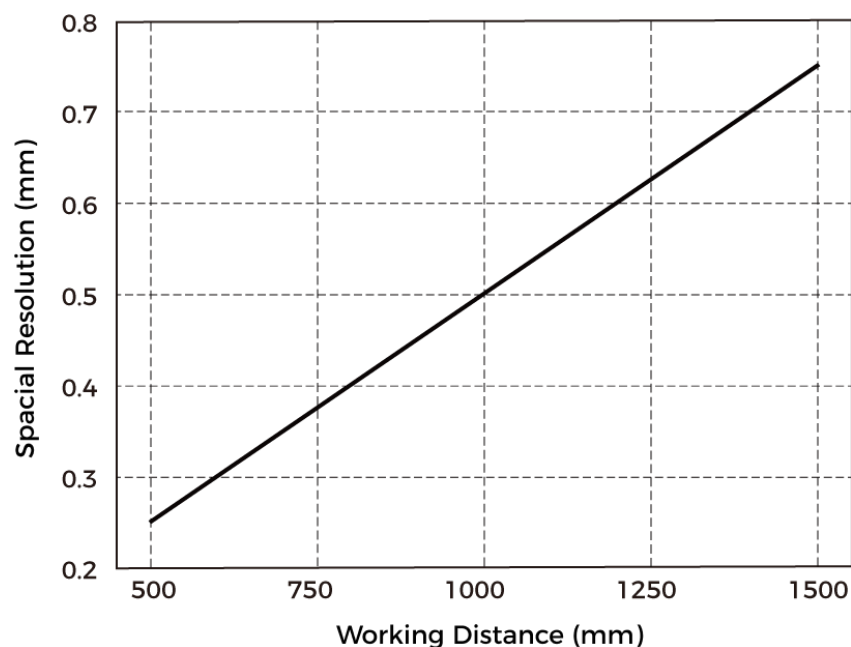
(3) According to VDI/VDE 2634 Part II.

(4) Excluding the heat-dissipation panel and camera bracket.

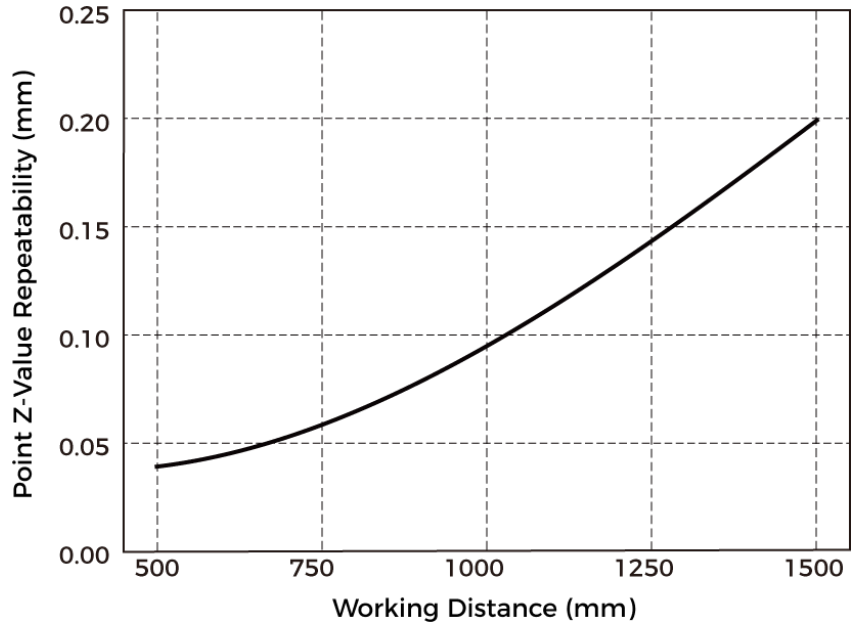
(5) The operating temperature of the camera without the heat-dissipation panel.

(6) Test implemented based on IEC 60529. 6: dust-tight; 5: protected against water jets.

Spatial Resolution

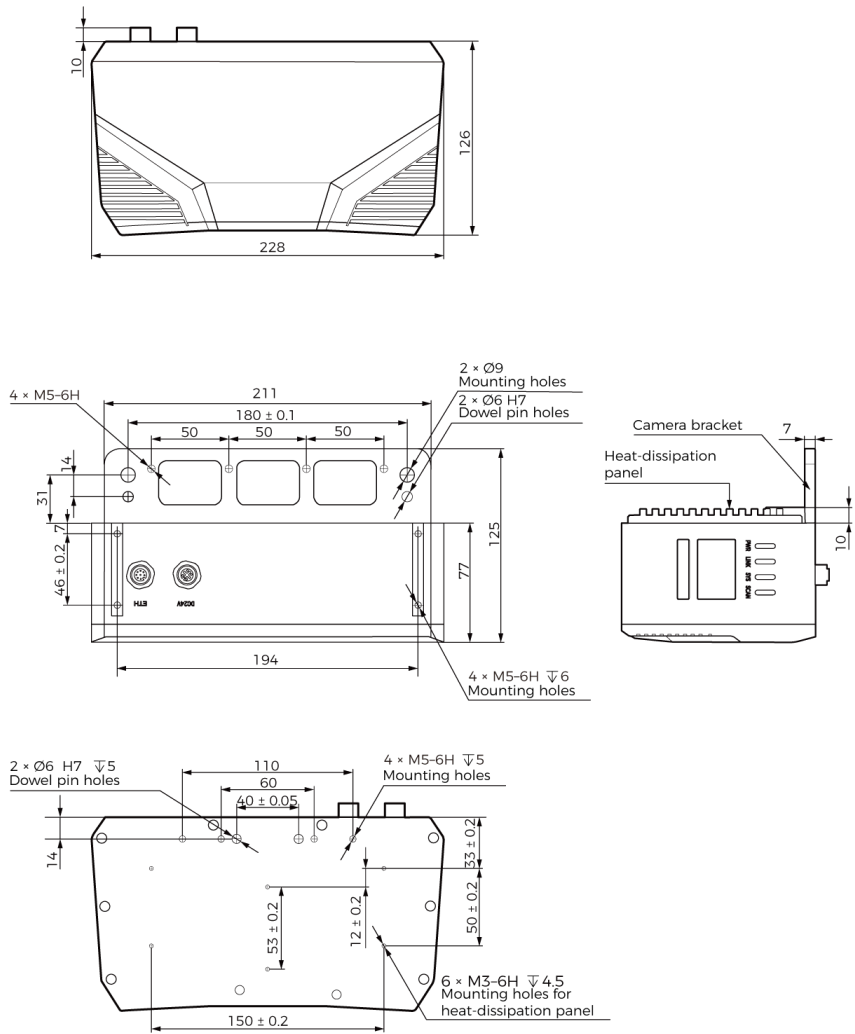


Point Z-Value Repeatability



Dimensions

Unit: mm



Safety of Laser Products

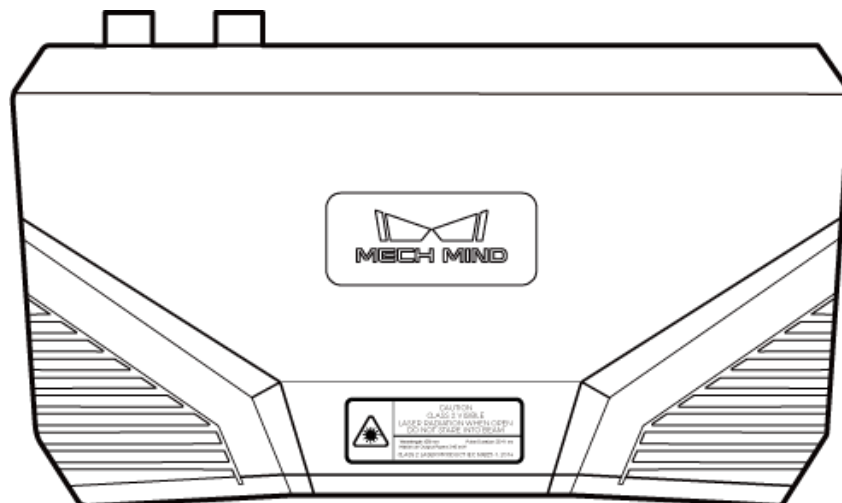
The laser classification is implemented based on IEC 60825-1:2014 in accordance with the requirements of Laser Notice No. 56 of the FDA (CDRH).

Model	Wavelength	Max. output power	Laser class
LSR S	638 nm	2.46 mW	Class 2

Label



Warning Label Attachment



CE

Hereby [Mech-Mind Robotics Technologies Ltd.] declares that [LSR S] is in compliance with the Electromagnetic Compatibility Directive 2014/30/EU.

The full text of the EU Declaration of Conformity is available at <https://downloads.mech-mind.com/?tab=tab-eu-dec>

8.3.6. LSR L

Technical Specifications

Product name	Mech-Eye Industrial 3D Camera
Model	LSR L

Object focal distance ⁽¹⁾	1500 mm	3000 mm
Recommended working distance	1200–1800 mm	1800–3000 mm
FOV (near)	1200 × 1000 mm @ 1.2 m	
FOV (far)	3000 × 2400 mm @ 3.0 m	
Depth map resolution	2048 × 1536	
RGB resolution	4000 × 3000 / 2000 × 1500	
Point Z-value repeatability (σ) ⁽²⁾	0.5 mm @ 3 m	
Measurement accuracy (VDI/VDE) ⁽³⁾	1.0 mm @ 3 m	
Weight	2.9 kg	
Baseline	380 mm	
Dimensions	459 × 77 × 86 mm	
Typical capture time	0.5–0.9 s	
Light source	Red laser (638 nm, Class 2)	
Operating temperature	-10–45°C	
Communication interface	Gigabit Ethernet	
Input	24 VDC, 3.75 A	
Safety and EMC	CE/FCC/VCCI/KC/ISED/NRTL	
IP rating ⁽⁴⁾	IP65	
Cooling	Passive	

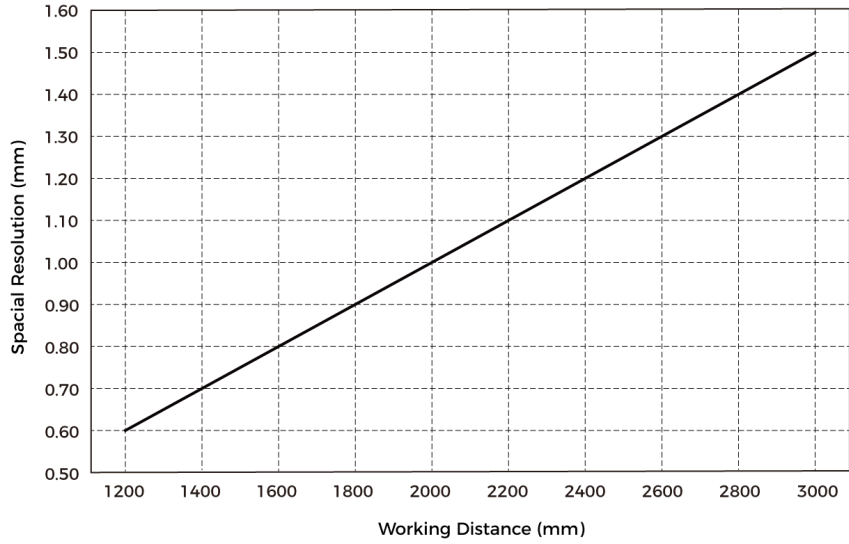
(1) The camera is available in two object focal distances, each corresponding to a different range of recommended working distance.

(2) One standard deviation of 100 Z-value measurements of the same point. The measurement target was a ceramic plate.

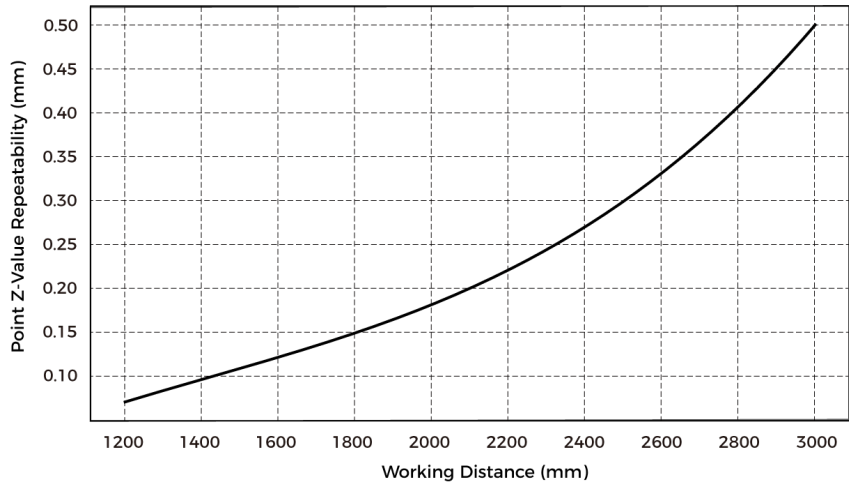
(3) According to VDI/VDE 2634 Part II.

(4) Test implemented based on IEC 60529. 6: dust-tight; 5: protected against water jets.

Spatial Resolution

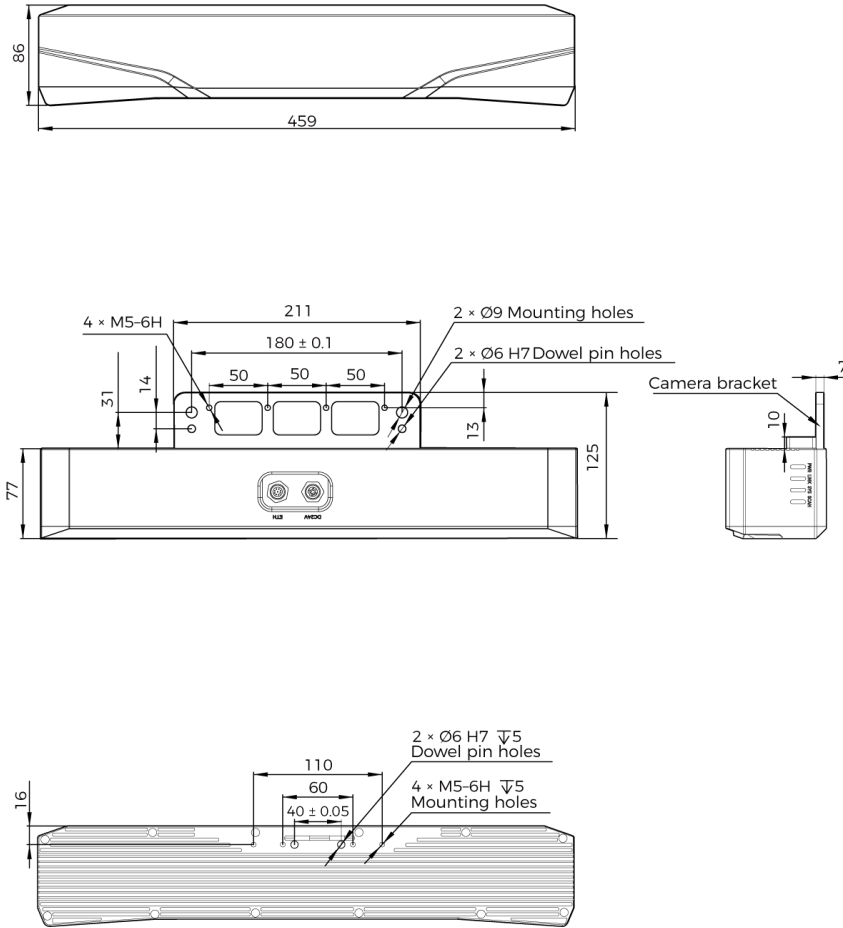


Point Z-Value Repeatability



Dimensions

Unit: mm



Safety of Laser Products

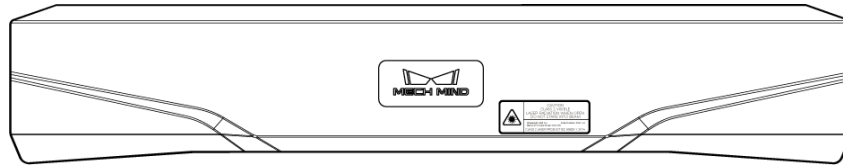
The laser classification is implemented based on IEC 60825-1:2014 in accordance with the requirements of Laser Notice No. 56 of the FDA (CDRH).

Model	Wavelength	Max. output power	Laser class
LSR L	638 nm	2.46 mW	Class 2

Label



Warning Label Attachment



CE

Hereby [Mech-Mind Robotics Technologies Ltd.] declares that [LSR L] is in compliance with the Electromagnetic Compatibility Directive 2014/30/EU.

The full text of the EU Declaration of Conformity is available at <https://downloads.mech-mind.com/?tab=tab-eu-dec>

8.3.7. Technical Specifications of V3 Cameras

Technical Specifications

Product name	Mech-Eye Industrial 3D Camera			
Model	Pro XS	Log S	Log M	Laser L Enhanced
Object focal distance	350	700	1200	1500
	550	1000	2000	3000
Recommended working distance	300–450 mm	500–750 mm	1000–1300 mm	1200–1700 mm
	450–650 mm	750–1000 mm	1300–2000 mm	1700–3000 mm
FOV (near)	220 × 160 mm @ 0.3 m	360 × 250 mm @ 0.5 m	520 × 390 mm @ 0.8 m	1500 × 1200 mm @ 1.5 m
FOV (far)	430 × 320 mm @ 0.6 m	710 × 490 mm @ 1.0 m	1410 × 960 mm @ 2.0 m	3000 × 2400 mm @ 3.0 m
Resolution	1280 × 1024			4096 × 3000
Pixels	1.3 MP			12.0 MP
Point Z-value repeatability (σ)	0.1 mm @ 0.5 m	0.1 mm @ 1 m	0.3 mm @ 2 m	0.5 mm @ 3 m
Measurement accuracy	0.1 mm @ 0.5 m	0.2 mm @ 1 m	0.3 mm @ 2 m	0.5 mm @ 3 m
Typical capture time	0.7–1.1 s	0.3–0.5 s	0.3–0.5 s	1.4–1.7 s
Baseline	About 93 mm	About 150 mm	About 280 mm	About 400 mm
Dimensions	About 160 × 52 × 87 mm	About 270 × 72 × 130 mm	About 387 × 72 × 130 mm	About 459 × 89 × 145 mm
Weight	About 0.8 kg	About 2.2 kg	About 2.4 kg	About 3.9 kg
Operating temperature	0–45°C			-10–45°C
Communication interface	Ethernet			
Operating voltage	24 VDC			
Safety and EMC	CE/FCC/VCCI			

IP rating	IP65
Cooling	Passive

9. Extended Reading

Here you can find helpful and informative topics that can aid you with using the camera.

A brief introduction of the imaging mechanism of structured light cameras:

[Structured-Light 3D Scanning Mechanism](#)

Descriptions of the built-in parameter groups of the camera:

[Built-in Parameter Groups](#)

The following topics provide instructions for operations that you may need while using the camera.

- [Set IP Address on Computer](#)
- [Enable Jumbo Frames](#)
- **

9.1. Imaging Mechanism of Structured-Light Camera

As a structured-light 3D camera, Mech-Eye Industrial 3D Camera (“3D camera”) consists of a projector and 2D camera(s). The projector projects structured light (light of pre-designed, specific patterns) onto the objects, and the shapes of the objects cause the light patterns to deform. Based on the deformation of the patterns, the 3D camera can calculate the depth data of the object surfaces.

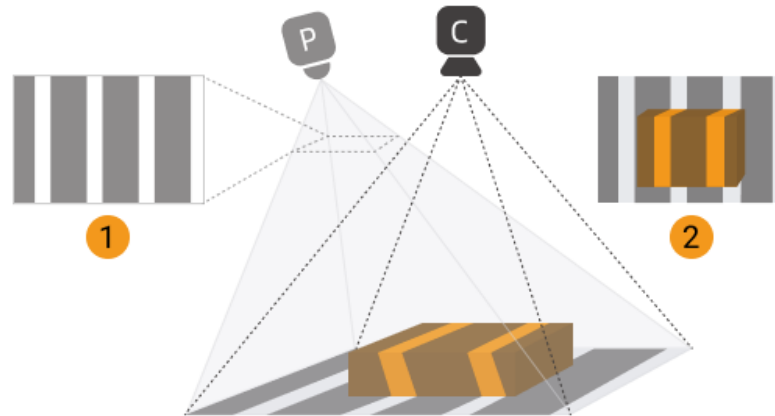
The 3D camera can also obtain the 2D image of the objects, just like a regular 2D camera.

The entire scanning process of the 3D camera can be divided into three phases: image capturing, calculation and processing, and output.

Image Capturing

The 3D camera is triggered to capture images by Mech-Eye SDK. During image capturing, the following processes take place:

1. The projector (P in the figure below) projects stripe patterned light onto the objects. Due to the shapes and structures of the objects, the stripe patterns deform.
2. The 2D camera(s) (C in the figure below) captures images that record the deformation of the stripe patterns.



In the meantime, the 3D camera also captures the regular 2D image of the objects while the structured light is not projected.

Calculation and Processing

The 3D camera calculates the depth data of the object surfaces based on the deformation of the stripe patterns. Using the depth data, the 3D camera then generates the depth map and point cloud.

Output

The 3D camera outputs the generated data to Mech-Eye SDK. Now you can see the 2D image, depth map and point cloud of the objects!

- 2D image: the image captured without structured light projection.
- Depth map: an image containing depth information.
- Point cloud: a collection of data points that contain the 3D coordinates (XYZ) of the object surface.

9.2. Built-in Parameter Groups

After the firmware of V4 cameras is upgraded to 2.2.0, built-in parameter groups suitable for different scenarios or workpieces are added to the camera. After selecting the parameter group that corresponds to your actual situation, you only need to adjust the parameter values slightly to acquire satisfactory data, simplifying the process of parameter adjustment.

This topic introduces the suitable scenario or workpiece of the built-in parameter groups for each model.

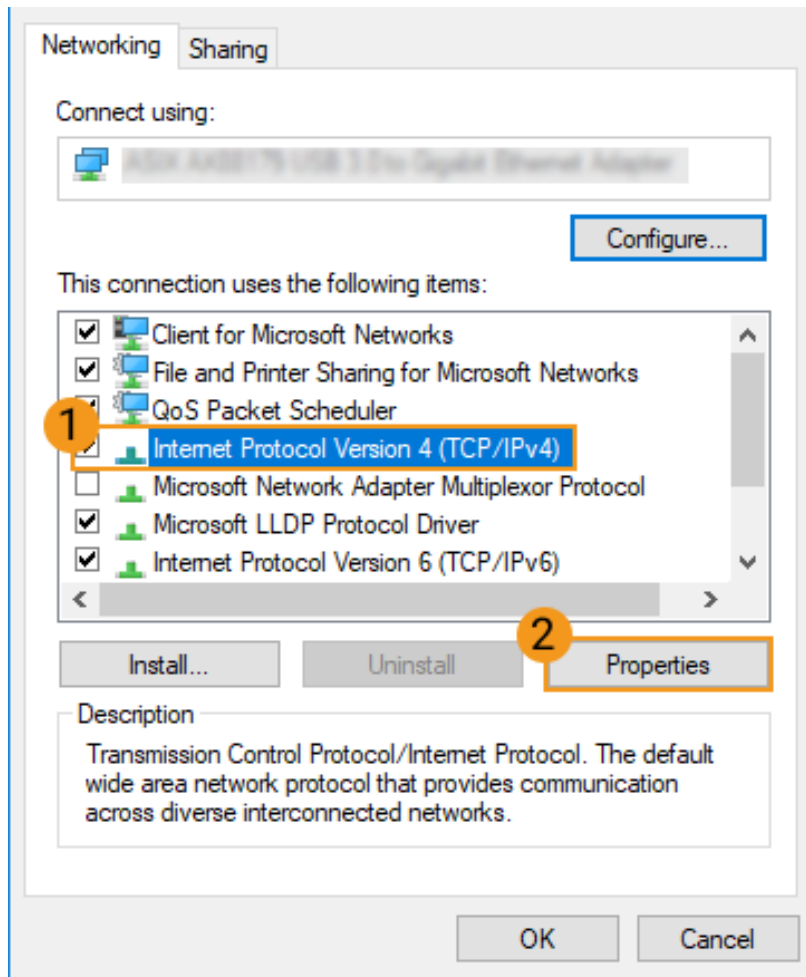
Model	Parameter group	Suitable scenario/workpiece
PRO M	Reflective + unreflective	Workpieces that contain both reflective and unreflective parts
	Reflective metal	Metal workpieces that are reflective
	Carton	Cartons
	Translucent object	Translucent objects

Model	Parameter group	Suitable scenario/workpiece
PRO S	Reflective rod	Rod-shaped workpieces that are reflective
	Small workpieces	Small workpieces with complicated shapes, such as gears
	Threaded rod	Rod-shaped workpieces with threaded surfaces
	Translucent object	Translucent objects
LSR L	Reflective metal	Metal workpieces that are reflective
	Unreflective metal	Metal workpieces that are unreflective
DEEP	Sack	Sack
	Cartons	Cartons
	Plastic turnover boxes	Plastic turnover boxes
UHP-140	Automotive frame	Automotive frames
	Magnetron	Magnetron
	Unreflective aluminum	Aluminum materials that are unreflective
NANO	Metal welding	Metal welding
	Charging port locating	Locating of the charging ports on EVs

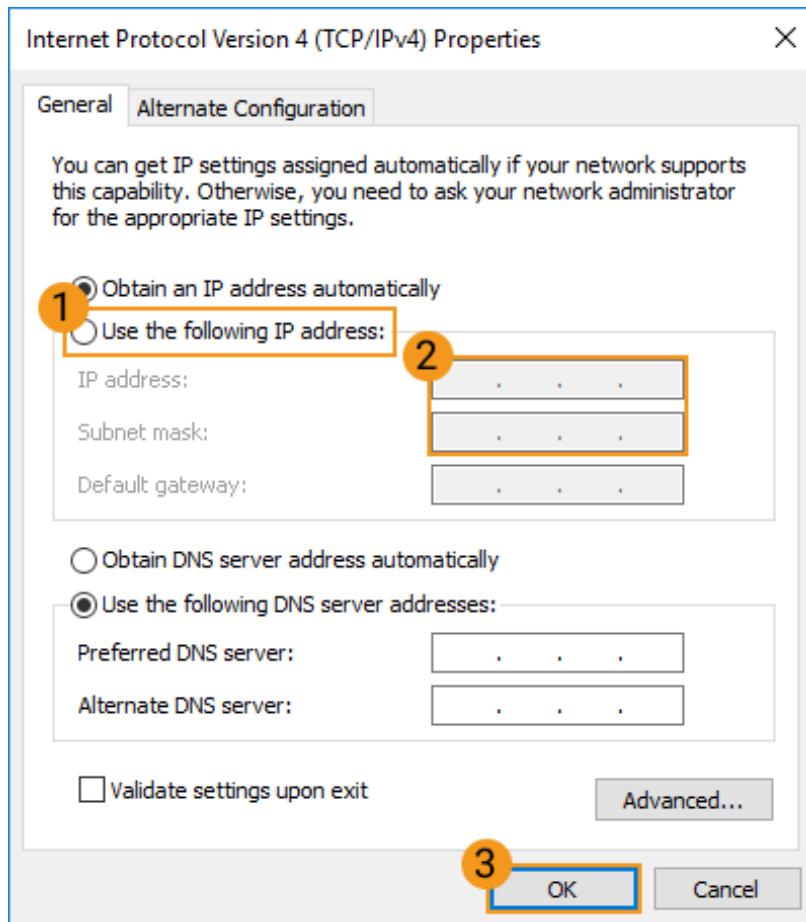
9.3. Set IP Address on Computer

This topic provides instructions on setting the IP address of the computer Ethernet port connected to the camera.

1. On the computer, select Control Panel > Network and Internet > Network and Sharing Center > Change adapter settings. The **Network Connections** page will be displayed.
2. Right-click the Ethernet port connected to the camera, and select **Properties** to enter the **Ethernet Properties** page.
3. Select **Internet Protocol Version 4 (TCP/IPv4)**, and then click [**Property**] to enter the **Internet Protocol Version 4 (TCP/IPv4) Properties** page.



4. Select **Use the following IP address**, and set the **IP address** and **Subnet mask** according to your actual needs. Click [**OK**] to save the changes.



To have the IP address automatically assigned, please select **Obtain an IP address automatically**.

9.4. Enable Jumbo Frames

Standard Ethernet frames carry up to about 1500 bytes of data. Jumbo frames are Ethernet frames that can carry up to about 9000 bytes of data. Using jumbo frames can reduce CPU processing load and speed up data transmission.

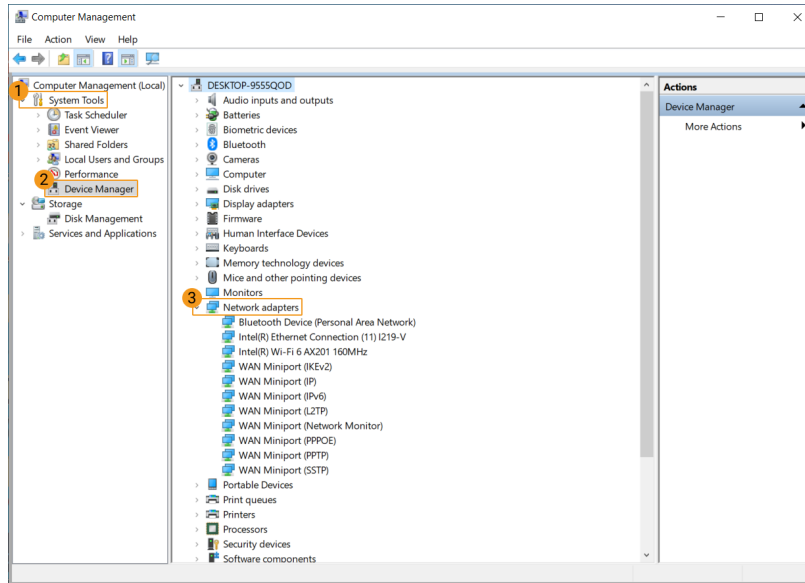
If the camera is directly connected to the computer or connected through a network switch, and the camera MTU size is set to 9000, please enable the jumbo frame function on the computer to improve data transmission speed.



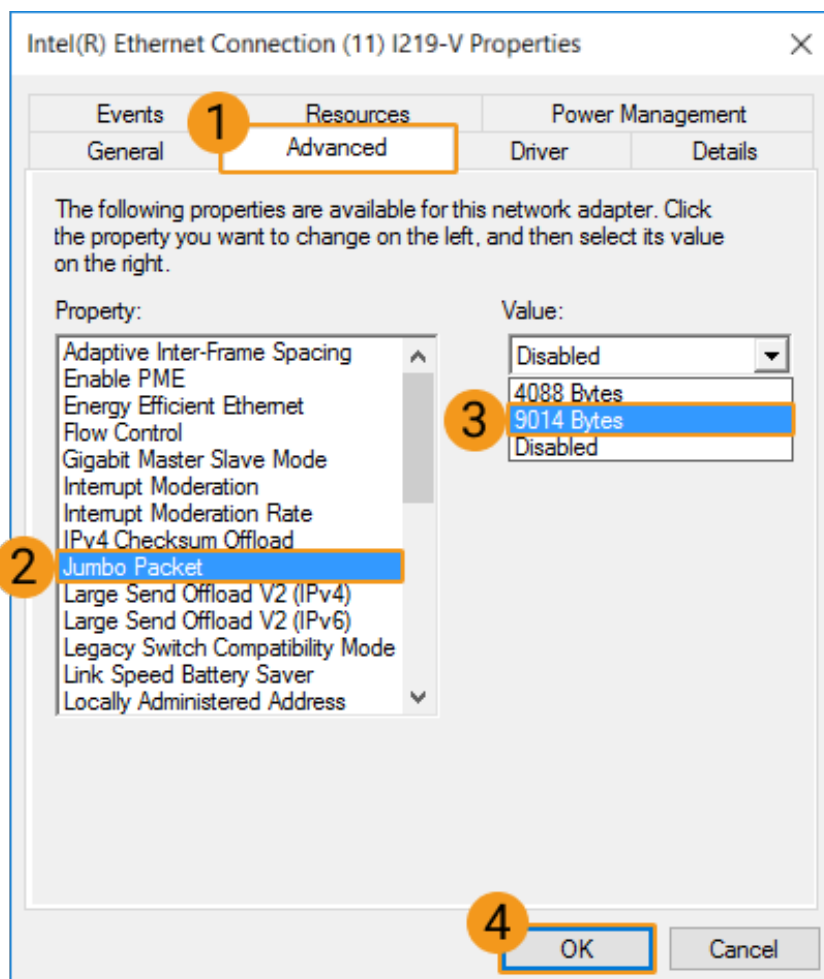
The camera MTU size can be set in the **Administrator** account. If you need to change it, please contact Technical Support.

Follow these steps to enable jumbo frames:

1. Right-click **This PC** and select **Manage** to open the **Computer Management** window.
2. In the left panel, select **System Tools > Device Manager**. Then in the middle panel, select to expand **Network adapters**.



3. Right-click the network interface card connected to the camera and select **Properties**.
4. Select the **Advanced** tab. Next, select **Jumbo Packet** in the **Property** box, and then select **9014 Bytes** for **Value** to the right. Click [**OK**] to save the change.



10. Support

10.1. Troubleshooting

This topic provides troubleshooting guidance for commonly encountered issues.



If you encounter an issue not covered by the following topics, please visit [Mech-Mind Online Community](#) to get help.

Camera Connection

- [Camera Indicator Lights Indicated Abnormality after Power Was Connected](#)
- [Camera Not Found in Mech-Eye SDK](#)
- [Camera Cannot Be Connected in Mech-Eye SDK](#)
- [Camera Cannot Be Connected in Mech-Eye SDK \(Windows Server\)](#)

Data Acquisition

- [Camera Did Not Project Light, and Depth Map Was Not Generated](#)
- [Image Acquisition Time Too Long](#)

10.1.1. Camera Indicator Lights Indicated Abnormality after Power Was Connected

After the camera is connected to power, you can check the indicator lights on the camera to see if its in proper working condition.

This topic provides the troubleshooting guidance for three issues that can be discovered through the indicator lights.



- This topic is applicable for V4 cameras. If you encounter similar issues with V3 cameras, please contact Technical Support.
- For the display of the indicator lights in working condition, please refer to [Camera Hardware User Manual](#).

All Indicator Lights Off

Symptom

After the camera was connected to power, none of the four indicator lights on the camera were lit.

Possible Causes

- The camera cables are not connected correctly.
- The voltage of the power supply is abnormal.
- The DIN rail power supply is not grounded.

Solution

Please follow these steps to troubleshoot the issue:

1. Refer to [Camera Hardware User Manual](#) and check if the camera cables are correctly connected and the connector nuts are fully tightened.
 - If the camera cables are not correctly connected, please correct the connection and check indicator lights on the camera again.
 - If the indicator lights on the camera indicate proper working condition, the troubleshooting is completed.
 - If the indicator lights still indicate the presence of issues, proceed to step 2.
 - If the camera cables are correctly connected, proceed to step 2.
2. Disconnect the camera DC power cable from the camera, and use a multimeter to check the voltage of the M-12 A connector and see if it reaches 24 V.
 - If the voltage does not reach 24 V, check aspects such as the connection of other cables in the power supply line and make corresponding corrections. Then, connect the DC power cable to the camera again and check the indicator lights on the camera.
 - If the indicator lights on the camera indicate proper working condition, the troubleshooting is completed.
 - If the indicator lights still indicate the presence of issues, proceed to step 3.
 - If the voltage reaches 24 V, proceed to step 3.
3. Check if the power supply is correctly grounded: if a DIN rail power supply is used to supply power to the camera, refer to [Camera Hardware User Manual](#) to check if the PE wire of the DIN rail power supply is correctly connected. If a power adaptor is used to supply power to the camera, check if the power outlet to which the 3-prong connector of the adaptor AC power cable is connected is correctly grounded.
 - If the power supply is not correctly grounded, the camera might have been damaged. Please contact Technical Support.
 - If the power supply is correctly grounded but the symptom is still present, please contact Technical Support.

PWR Indicator Light Solid Red

Symptom

After the camera was connected to power, the PWR indicator light was solid red.

Possible Causes

The voltage of the power supply is abnormal.

Solution

Please follow these steps to troubleshoot the issue:

1. Refer to [Camera Hardware User Manual](#) and check if the camera DC power cable is correctly connected and the connector nut is fully tightened.
 - If the camera DC power cable is not correctly connected, please correct the connection and check indicator lights on the camera again.
 - If the indicator lights on the camera indicate proper working condition, the troubleshooting is completed.
 - If the indicator lights still indicate the presence of issues, proceed to step 2.

- If the camera DC power cable is correctly connected, proceed to step 2.
- 2. Disconnect the camera DC power cable from the camera, and use a multimeter to check the voltage of the M-12 A connector and see if it reaches 24 V.
 - If the voltage does not reach 24 V, check aspects such as the connection of other cables in the power supply line and make corresponding corrections. Then, connect the camera DC power cable to the camera again and check the indicator lights on the camera.
 - If the indicator lights on the camera indicate proper working condition, the troubleshooting is completed.
 - If the indicator lights still indicate the presence of issues, please contact Technical Support.

PWR Indicator Light Solid Green but LINK Indicator Light Off

Symptom

After the camera was connected to power, , the PWR indicator light was solid green, but the LINK indicator light was not lit.

Possible Causes

The network connection is abnormal.

Solution

Please follow these steps to troubleshoot the issue:

1. Refer to [Camera Hardware User Manual](#) and check if the camera Ethernet cable is correctly connected and the connector nut is fully tightened.
 - If the camera Ethernet cable is not correctly connected, please correct the connection and check indicator lights on the camera again.
 - If the indicator lights on the camera indicate proper working condition, the troubleshooting is completed.
 - If the indicator lights still indicate the presence of issues, proceed to step 2.
 - If the camera Ethernet cable is correctly connected, proceed to step 2.
2. Replace the camera Ethernet cable, and then check the indicator lights on the camera again.
 - If the indicator lights on the camera indicate proper working condition, the troubleshooting is completed.
 - If the indicator lights still indicate the presence of issues, please contact Technical Support.

Related Issues

The following issues might be caused by the same reasons. If you also encountered these issues, after resolving the issue in this topic, you can check if these other issues have been resolved as well.

- [Camera Did Not Project Light, and Depth Map Was Not Generated](#)
- [Image Acquisition Time Too Long](#)

- [Camera Disconnected Occasionally](#)

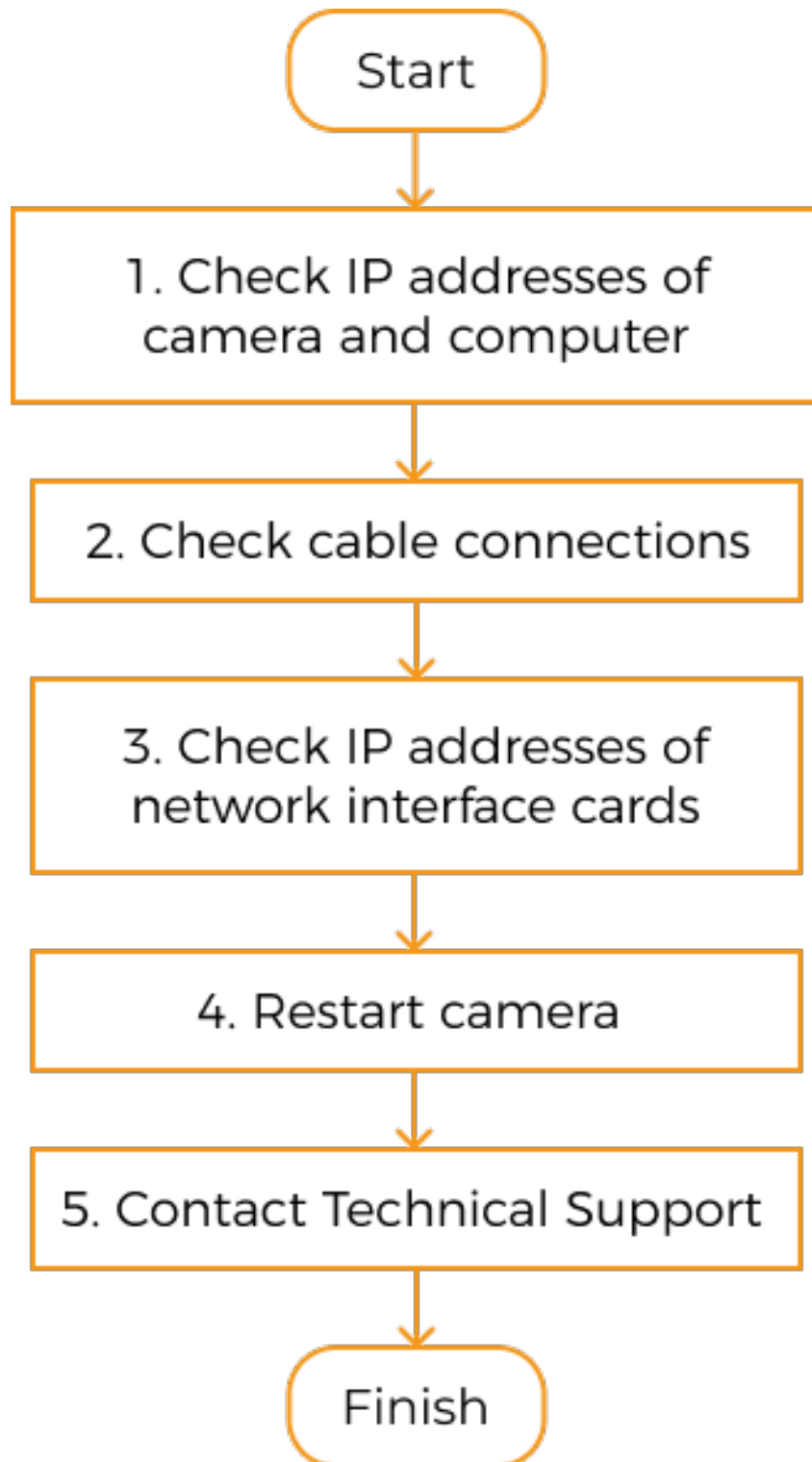
10.1.2. Camera Not Found in Mech-Eye SDK

Symptom: In Mech-Eye SDK, the camera to which you want to connect does not show up in the list of available cameras. This topic provides the possible causes, troubleshooting guidelines and detailed troubleshooting procedures for this problem.

Possible Causes

- IP address conflict between the camera and computer is present.
- Some of the cables are not properly connected.
- Multiple Ethernet ports of the computer have IP addresses in the same subnet.
- Other causes.

Troubleshooting Guidelines



When the camera to which you want to connect does not show up in Mech-Eye SDK, follow these guidelines to troubleshoot:

1. If Mech-Eye SDK can only discover one camera, **check the IP addresses of this camera and the computer.**
2. Next, **check cable connections.** If the camera still does not show up in Mech-Eye SDK, proceed to the next step.
3. Make sure that **the IP addresses of multiple Ethernet ports are not in the same subnet.**

4. If the camera still does not show up in Mech-Eye SDK, proceed to the next step.
5. **Restart the camera.** If the camera still does not show up, please contact Technical Support.

Troubleshooting Procedures

Check IP Addresses of Camera and Computer

If multiple cameras are connected to the computer, but only one Mech-Eye SDK can be discovered in camera, follow these steps to troubleshoot:

1. Check if the following two IP addresses are identical:
 - The IP address of this camera
 - The IP address of the computer Ethernet port connected to the camera
2. If the two IP addresses are identical, modify one of them:
 - Modify the IP address of the camera: Use [Mech-Eye Viewer](#) or the [IP configuration tool](#).
 - Modify the IP address of the computer Ethernet port: Please refer to [Set IP Address on Computer](#).
3. Open Mech-Eye SDK and search for the camera.
 - If all cameras could be discovered, the troubleshooting is completed.
 - If still not all cameras could be discovered, proceed to the next section.

Check Cable Connections

Device	Cable	Description of proper cable connection
Camera	Power cable	PWR indicator light is solid green.
	Ethernet cable	V3 cameras: ETH indicator light is either green (gigabit network) or yellow (megabit network). V4 cameras: LINK indicator light is blinking or solid green .
Computer	Ethernet cable	The indicator light of the Ethernet port is lit.
Network switch	Ethernet cable	Cable is connected to an Ethernet port (router)/access port (network switch), and the indicator light of the port is lit.



- If the camera and computer are connected directly, you do not need to check for the cable connection of the network switch.
- If the LINK/ETH indicator light of the camera blinks several times and goes off, and this phenomenon recurs, please check the connection of Ethernet cable(s). This phenomenon is usually caused by Ethernet cables not fully inserted into ports.

IP Addresses of Multiple Ethernet Ports in the Same Subnet

Procedure:

1. Disable Ethernet ports not connected to the camera.

On the computer, select Control Panel › Network and Internet › Network and Sharing Center ›

Change adapter settings. The **Network Connections** page will be displayed. Right-click an Ethernet port and select **Disable**.

2. Open Mech-Eye SDK and search for the camera.
3. If the Mech-Eye SDK is successfully discovered, one or more of the disabled Ethernet ports have IP address in the same subnet as the port connected to the Mech-Eye SDK. Enable the other ports one at a time and connect to the camera in Mech-Eye SDK each time. The camera connection will fail when the Ethernet port with the problematic IP address is enabled. Please change the IP address of this Ethernet port to a different subnet.



If the camera still cannot be found, disable the Ethernet port connected to the camera and enable it again. If the camera still cannot be found afterward, connect the camera to a different Ethernet port.

Restart the Camera

If the camera still does not show up after the above procedures have been performed, please try restarting the camera.

Restart procedure:

1. Unplug the power cable from the camera.
2. Wait for about 20 seconds, and plug the power cable into the camera.

Contact Technical Support

If the camera still does not show up after the above procedures have been performed, please contact Technical Support.

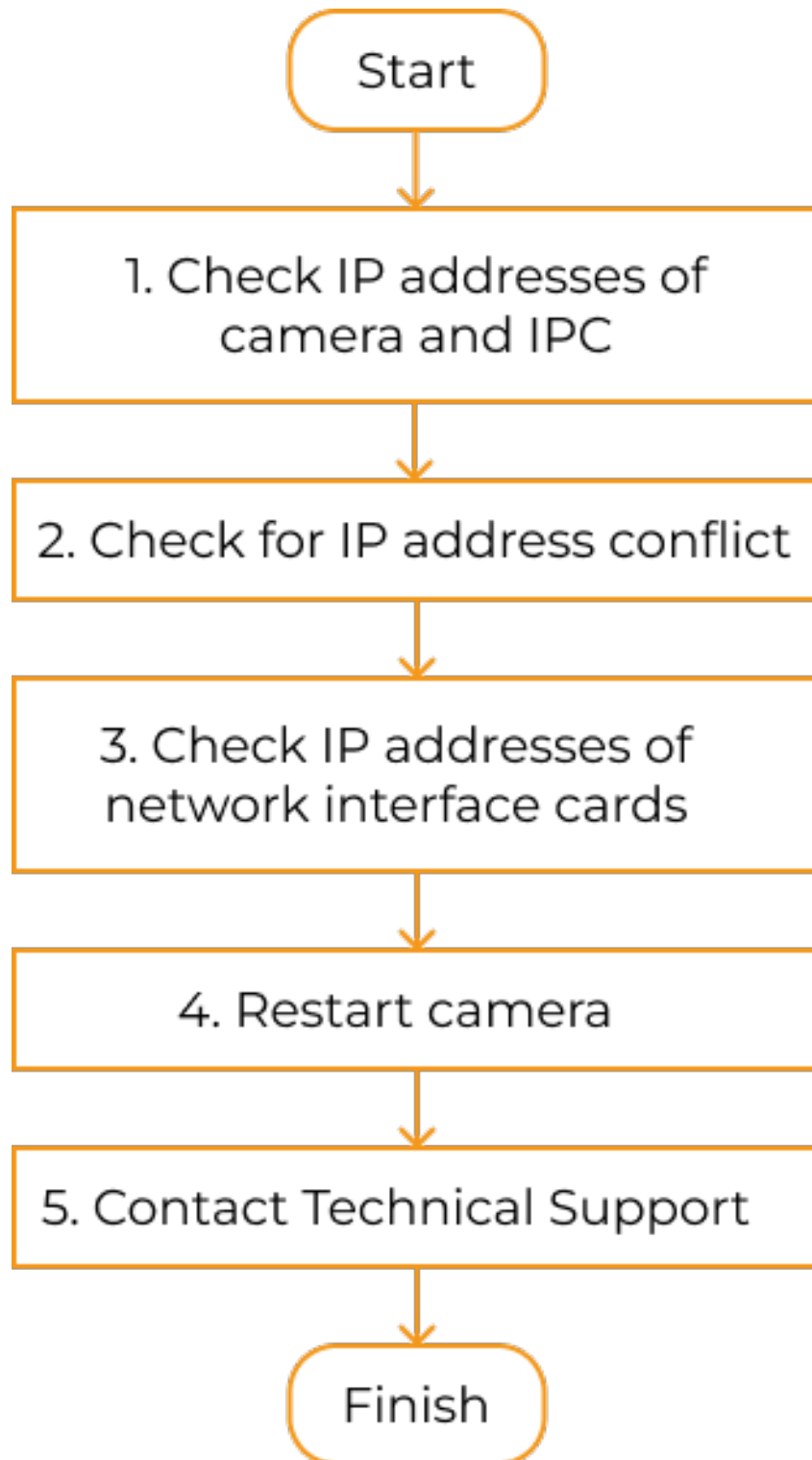
10.1.3. Camera Cannot Be Connected in Mech-Eye SDK

Symptom: The camera shows up in Mech-Eye SDK but cannot be connected. This topic provides the possible causes, troubleshooting guidelines and detailed troubleshooting procedures for this problem.

Possible Causes

- The IP addresses of the camera and computer are not in the same subnet.
- IP address conflict is present.
- Multiple Ethernet ports of the computer have IP addresses in the same subnet.
- Other causes.

Troubleshooting Guidelines



When you cannot connect to the camera in Mech-Eye SDK, follow these guidelines to troubleshoot:

1. Make sure that the IP addresses of the camera and computer are in the same subnet. If the camera still cannot be connected, proceed to the next step.
2. Make sure that no IP address conflicts are present. If the camera still cannot be connected, proceed to the next step.
3. Make sure that the IP addresses of multiple Ethernet ports are not in the same subnet.

4. If the camera still cannot be connected, proceed to the next step.
5. **Restart the camera.** If the camera still cannot be connected, please contact Technical Support.

Troubleshooting Procedures

Check IP Addresses of Camera and computer

Procedure:

1. Check the IP address of the computer. For instructions, please refer to [Set IP Address on Computer](#).
2. Check the IP address of the camera. For instructions, please refer to [Set Camera IP Address](#).
3. Make sure that the IP addresses of the camera and computer are in the same subnet. If not, please [change the camera IP address](#), so that it is in the same subnet as the IP address of the computer.
4. Open Mech-Eye SDK and connect to the camera.

Check for IP Address Conflict

IP conflict is present among the devices connected to the computer and network switch. IP conflict is where two or more devices have identical IP addresses.

Procedure:

1. Disconnect all devices other than the camera from the computer and/or network switch.
2. Open Mech-Eye SDK and connect to the camera.
3. If the camera is successfully connected, one or more other devices have IP addresses identical to the camera. Please reconnect the other devices one at a time and connect to the camera in camera each time. The camera connection will fail when the device(s) with identical IP address is reconnected. Please change the IP address of this device or the camera, so that the two are no longer identical.

IP Addresses of Multiple Ethernet Ports in the Same Subnet

Procedure:

1. Disable Ethernet ports not connected to the camera.

On the computer, select Control Panel > Network and Internet > Network and Sharing Center > Change adapter settings. The **Network Connections** page will be displayed. Right-click an Ethernet port and select **Disable**.

2. Open Mech-Eye SDK and connect to the camera.
3. If the camera is successfully connected, one or more of the disabled Ethernet ports have IP address in the same subnet as the port connected to the camera. Enable the other ports one at a time and connect to the camera in Mech-Eye SDK each time. The camera connection will fail when the Ethernet port with the problematic IP address is enabled. Please change the IP address of this Ethernet port to a different subnet.



If the camera still cannot be connected, disable the Ethernet port connected to the camera and enable it again. If the camera still cannot be connected afterward, connect the camera to a different Ethernet port.

Restart the Camera

If the camera still cannot be connected after the above procedures have been performed, please try restarting the camera.

Restart procedure:

1. Unplug the power cable from the camera.
2. Wait for about 20 seconds, and plug the power cable into the camera.

Contact Technical Support

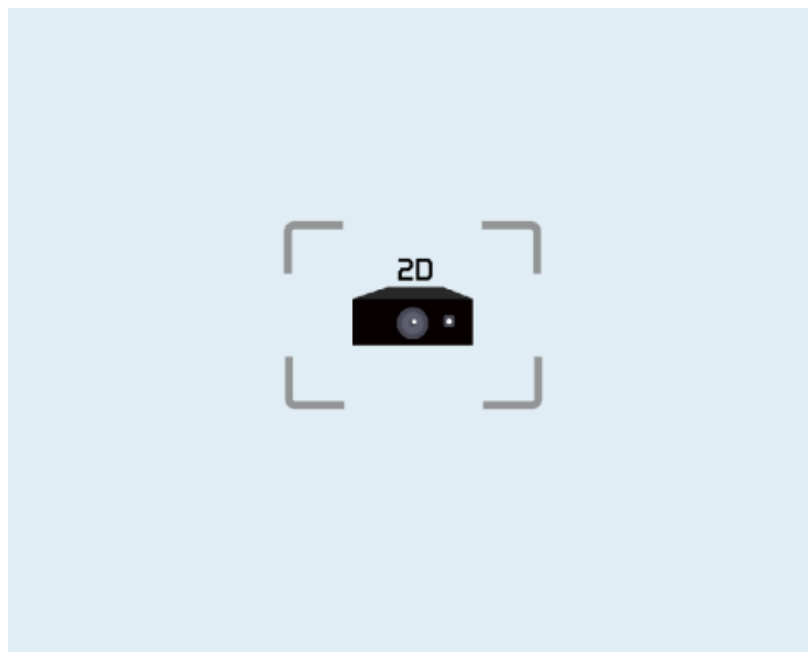
If the camera still cannot be connected after the above procedures have been performed, please contact Technical Support.

10.1.4. Camera Did Not Project Light, and Depth Map Was Not Generated

Symptoms

After controlling the camera to acquire images with Mech-Eye SDK, the camera did not project light, and the software did not receive the depth map.

- Mech-Eye Viewer: The depth map display interface only displayed the following image.



- Mech-Eye API: After the image acquisition method had been called, the error code **-6** was returned.
- GenICam interface: a timeout error might have occurred.

Possible Causes

- The camera cables are not connected correctly.
- The DC power cable of the DIN rail power supply is too long.
- The voltage of the power supply is insufficient.

Solution



The instructions in this section are only applicable to the Windows operating system. If you are using the Ubuntu operating system, please contact Technical Support.

1. Based on the time when the issue occurred, in Mech-Eye Viewer, [open the last log generated before the issue occurred](#).
2. In this log, based on the time on the left, find the message recorded when the issue occurred. Based on the error code in the message, check the corresponding solution below.

Error Code 0x0105, 0x0107 or 0x0109

Please follow these steps to troubleshoot the issue:

1. Refer to [Camera Hardware User Manual](#) and [Guidelines for Mounting Camera Cables](#) and check if the camera cables are correctly connected and mounted, without issues such as not fully tightened connector nut or tangling with other cables.
 - If the camera cables are not correctly connected and mounted, please correct the connection and mounting, and then check if the symptoms are still present.
 - If not present, the troubleshooting is completed.
 - If still present, proceed to step 2.
 - If the camera cables are correctly connected and mounted, proceed to step 2.
2. Based on the power supply of the camera, proceed to the corresponding step:
 - If a DIN rail power supply or power adaptor is used to supply power to the camera, proceed to step 3.
 - If an automated guided vehicle (AGV) is used to supply power to the camera, proceed to step 4.
3. Check if the power supply is correctly grounded: if a DIN rail power supply is used to supply power to the camera, refer to [Camera Hardware User Manual](#) to check if the PE wire of the DIN rail power supply is correctly connected. If a power adaptor is used to supply power to the camera, check if the power outlet to which the 3-prong connector of the adaptor AC power cable is connected is correctly grounded.
 - If the power supply is not correctly grounded, correct the grounding and then check if the symptoms are still present.
 - If not present, the troubleshooting is completed.
 - If still present, please contact Technical Support.
 - If the power supply is correctly grounded but the symptoms are still present, please contact Technical Support.
4. Check if a delay relay is used on the AGV to ensure stable voltage.
 - If a delay relay is not used, please add one and then check if the symptoms are still present.
 - If not present, the troubleshooting is completed.
 - If still present, please contact Technical Support.
 - If a delay relay is already used, please contact Technical Support.

Error Code 0x020c

Please follow these steps to troubleshoot the issue:

Check if the DC power cable of the DIN rail power supply is longer than 20 m.

- If the cable is longer than 20 m, please adjust the voltage of the DIN rail power supply so that the voltage supplied to the camera is increase by 2 V.
- If the cable is not longer than 20 m, please contact Technical Support.

Other Error Codes

Please contact Technical Support.

Related Issues

The following issues might be caused by the same reasons. If you also encountered these issues, after resolving the issue in this topic, you can check if these other issues have been resolved as well.

- [Camera Indicator Lights Indicated Abnormality after Power Was Connected](#)
- [Image Acquisition Time Too Long](#)
- [Camera Disconnected Occasionally](#)

10.1.5. Camera Cannot Be Connected in Mech-Eye SDK (Windows Server)

Symptom: The camera shows up in Mech-Eye SDK but cannot be connected. The status of the IP address displayed in the camera info card in Mech-Eye Viewer is Windows Server. This topic provides the possible causes, troubleshooting guidelines and detailed troubleshooting procedures for this problem.

Possible Causes

This is a bug in Mech-Eye SDK 1.6.1 and below.

Solution

If this problem occurs, it is recommended to upgrade Mech-Eye SDK to the latest version. For instructions, please refer to [Upgrade Mech-Eye SDK](#).

If you cannot upgrade Mech-Eye SDK, please contact Technical Support.

10.1.6. Image Acquisition Time Too Long

Symptoms

After controlling the camera to acquire images with Mech-Eye SDK, the image acquisition time was significantly longer than normal. When using Mech-Eye Viewer to connect to the camera and check the data transmission speed in the upper-left corner of the interface, the data transmission speed was slower than 700 Mbps (slower than 500 Mbps for NANO and Pro XS).

Possible Causes

- The network switch or Ethernet cables in use does not support gigabit bandwidth.
- The driver for the computer Ethernet port requires an update or the settings of the Ethernet port are incorrect.
- The camera Ethernet cable is incorrectly connected or needs to be replaced.

Solution

Please follow these steps to troubleshoot the issue:

1. Check if the network switch or Ethernet cables in use supports gigabit bandwidth.
 - If gigabit bandwidth is not supported, please replace the network switch or Ethernet cables with those that support gigabit bandwidth instead. Megabit bandwidth cannot guarantee the stability of data transmission. After replacing the equipment, check if the symptoms are still present.
 - If not present, the troubleshooting is completed.
 - If still present, proceed to step 2.
 - If gigabit bandwidth is supported, proceed to step 2.
2. Check if the camera Ethernet cable is worn, broken, or damaged otherwise.
 - If the camera Ethernet cable is damaged, contact Technical Support to acquire a replacement. Then, check if the symptoms are still present.
 - If not present, the troubleshooting is completed.
 - If still present, proceed to step 3.
 - If the camera Ethernet cable is not damaged, proceed to step 3.
3. Refer to [Camera Hardware User Manual](#) and [Guidelines for Mounting Camera Cables](#) and check if the camera Ethernet cable is correctly connected and mounted, without issues such as not fully tightened connector nut or tangling with other cables.
 - If the camera Ethernet cable is not correctly connected and mounted, please correct the connection and mounting, and then check if the symptoms are still present.
 - If not present, the troubleshooting is completed.
 - If still present, proceed to step 4.
 - If the camera Ethernet cable is correctly connected and mounted, proceed to step 4.
4. Check the computer Ethernet port connected to the camera and see if its driver requires an update or if its settings require modification (For example, a speed limit on data transmission may need to be lifted).
 - If driver update or modification of the settings is required, please do so and then check if the symptoms are still present.
 - If not present, the troubleshooting is completed.
 - If still present, proceed to step 5.
 - If driver update or modification of the settings is not required, proceed to step 5.
5. Try using a different computer Ethernet port to connect to the camera, [set the IP address of the computer Ethernet port to a unique subnet not used by other ports](#), or disable other computer Ethernet ports not in use. Then, check if the symptoms are still present.

- If not present, the troubleshooting is completed.
- If still present, please contact Technical Support.

Related Issues

The following issues might be caused by the same reasons. If you also encountered these issues, after resolving the issue in this topic, you can check if these other issues have been resolved as well.

- [Camera Indicator Lights Indicated Abnormality after Power Was Connected](#)
- [Camera Did Not Project Light, and Depth Map Was Not Generated](#)
- [Camera Disconnected Occasionally](#)

10.2. FAQs

This topic provides answers to frequently asked questions.

- [How to Read Camera Serial Number?](#)

10.2.1. How to Read Camera Serial Number?

The camera serial number (SN) is printed on the label attached to the camera. It is the unique identification number of each camera.

The serial number has 16 characters, and it contains information such as the camera model, 2D camera type, object focal distance, and production date.



1. The first and second characters represent the camera model:

Number	Model
QA	Deep (V3) or DEEP (V4)
NA/KA	Log M (V3) or Log S (V3)
NE/KE	Pro M Enhanced (V3), Pro S Enhanced (V3), PRO S (V4), or PRO M (V4)
TA	Nano (V3) or NANO (V4)
RA	Pro XS (V3)
WA	Laser L (V3) or LSR L (V4)
VA	LSR S (V4)
WU	Laser L Enhanced-12MP (V3)

QE	Pro L Enhanced (V3)
UA	UHP-140 (V4)

2. The third character represents the type of the 2D camera(s) inside the product:

Code	2D camera type
C	Color 2D camera(s)
M	Monochrome 2D camera(s)
A	Both color and monochrome 2D cameras

3. The fourth and fifth characters represent the object focal distance of the camera:

Model	Object focal distance (mm)
UHP-140 (V4)	Code × 10
Nano (V3) or NANO (V4)	
Pro XS (V3)	
Other models	Code × 100

4. The sixth and seventh characters represent the camera production year: For example, “23” represents that the production year is 2023.
5. The eighth character represents the camera production month:

Code	Month
1 to 9	January to September
A	October
B	November
C	December

6. The ninth to sixteenth characters are the production code of the camera.