

---

# **Mech-Eye Industrial 3D Cameras**

**Mech-Mind**

**Feb 16, 2023**

## GETTING STARTED

1	Mech-Eye Industrial 3D Camera User Manual	5
2	Mech-Eye SDK 2.0 Release Notes	44
3	Mech-Eye SDK Installation Guide	54
4	Mech-Eye Viewer Quick Start Guide	59
5	Mech-Eye Viewer Quick Facts	62
6	User Interface of Mech-Eye Viewer	64
7	Using Mech-Eye Viewer	68
8	Built-In Tools	109
9	Mech-Eye API Quick Facts	132
10	Capture Tutorial	134
11	Samples	144
12	Mech-Eye API Reference	175
13	GenICam Quick Facts	176
14	GenICam Basics	177
15	Use HALCON to Control the Camera	179
16	GenICam Reference	191
17	Troubleshooting	197
18	Extended Reading	207

Thank you for using Mech-Eye Industrial 3D Camera! In conjunction with Mech-Eye SDK, Mech-Eye Industrial 3D Camera can be used to obtain and adjust 2D images, depth maps, and point clouds.

Mech-Eye SDK includes Mech-Eye API and Mech-Eye Viewer, a camera configuration and data visualization software developed using Mech-Eye API.

Mech-Eye Industrial 3D Camera can also be controlled by third-party software compatible with the GenI-Cam standard, such as HALCON (by MVTec) and VisionPro (by Cognex).

This documentation includes the following contents:

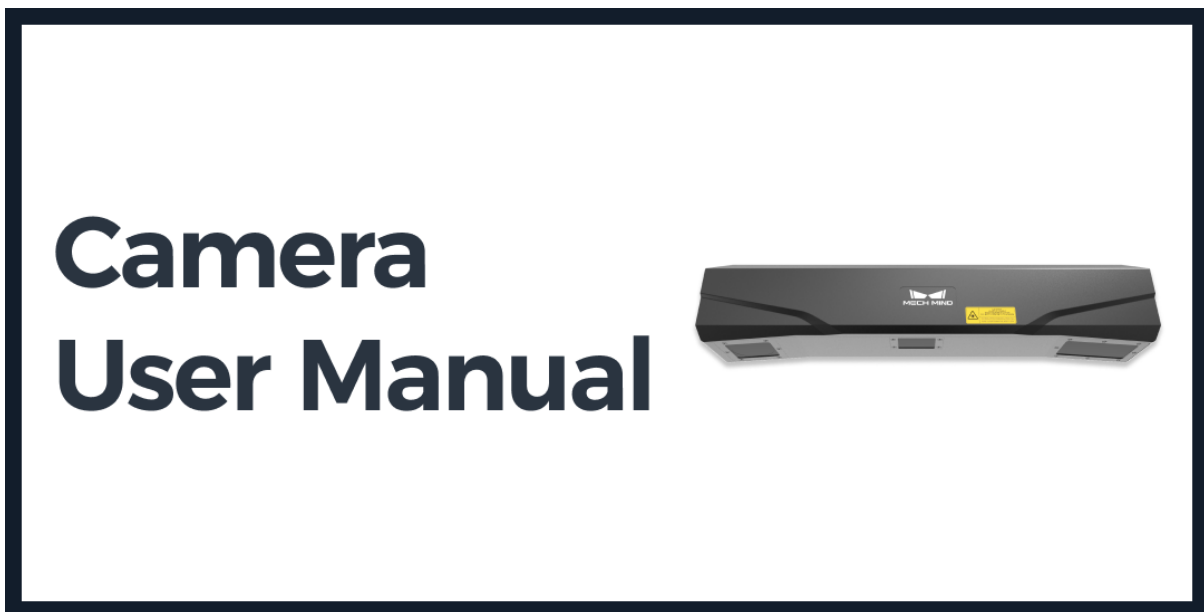
- Getting Started
- Mech-Eye Viewer User Manual
- Mech-Eye API User Manual
- Genicam
- Support

---

[Mech-Eye SDK 2.0 Release Notes](#)

[Mech-Eye SDK Installation Guide](#)

## Resources



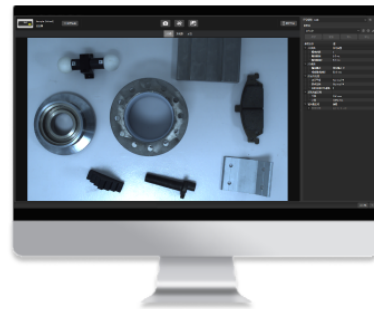
[Mech-Eye Industrial 3D Camera User Manual](#)

# Quick Start



*Mech-Eye Viewer Quick Start Guide*

# Mech-Eye Viewer



*Mech-Eye Viewer User Manual*

# Mech-Eye API



*Mech-Eye API User Manual*

# GenlCam



*GenlCam*

# Support



## *Troubleshooting*

---

Your feedback is important in helping us provide accurate, high-quality information. If you have any comments on the documentation, contact us in one of these ways:

- Use the [Leave a Comment](#) form at the Mech-Mind Docs website.
- Send your comments to the following email address: [docs@mech-mind.net](mailto:docs@mech-mind.net).

## MECH-EYE INDUSTRIAL 3D CAMERA USER MANUAL

Mech-Eye Industrial 3D Camera (hereinafter referred to as “camera”) is the industrial 3D camera developed by Mech-Mind Robotics. This manual provides information about the basics, general operation, and maintenance of the camera. Please read the manual thoroughly before using the camera to reduce risks and hazards.

---

Introduction of **different models of cameras**:

*Camera Models*

---

The **contents of the package**:

*Package Contents*

---

Descriptions of the **indicator lights and ports of the camera**:

*Indicator Lights and Ports*

---

Instructions on **mounting the camera**:

*Mount the Camera*

---

Instructions on **connecting the camera**:

*Connect the Camera*

---

**Technical specifications of the camera**:

*Technical Specifications*

---

**Safety, regulatory and maintenance information of the camera**:

*Safety, Regulations, and Maintenance Instructions*

---

## 1.1 Camera Models

This topic introduces different models of Mech-Eye Industrial 3D Camera, as well as the characteristics and suitable applications of the models.

### 1.1.1 Basic Information

This section provides the basic information of the third (V3) and fourth (V4) generations of Mech-Eye Industrial 3D Camera.

#### V4 Cameras

Model	Projector Type	No. of 2D cameras	Resolution
PRO S	DLP	Monocular	1920 × 1200
PRO M		Monocular	1920 × 1200
NANO		Monocular	1280 × 1024
UHP-140		Dual-monocular*	2048 × 1536
LSR L	Laser	Trinocular	Depth map: 2048 × 1536 RGB: 4000 × 3000, 2000 × 1500
DEEP			Depth map: 2048 × 1536 RGB: 2000 × 1500

\* UHP-140 can capture images with either of the two 2D cameras or with both cameras and merge the captured images, the latter reduces blind spots compared to monocular or binocular cameras.

---

**Hint:** For technical specifications, please refer to [Technical Specifications](#).

---

#### V3 Cameras

Model	Projector Type	No. of 2D cameras	Resolution
Pro S Enhanced	DLP	Monocular	1920 × 1200
Pro M Enhanced		Monocular	1920 × 1200
Log S		Monocular	1280 × 1024
Log M		Monocular	1280 × 1024
Nano		Monocular	1280 × 1024
Pro XS		Binocular	1280 × 1024
Deep		Binocular	2048 × 1536
Pro L Enhanced		Binocular	4096 × 3000
Laser L		Laser	Binocular
Laser L Enhanced	Binocular		4096 × 3000



**Hint:** For technical specifications, please refer to [Technical Specifications](#).

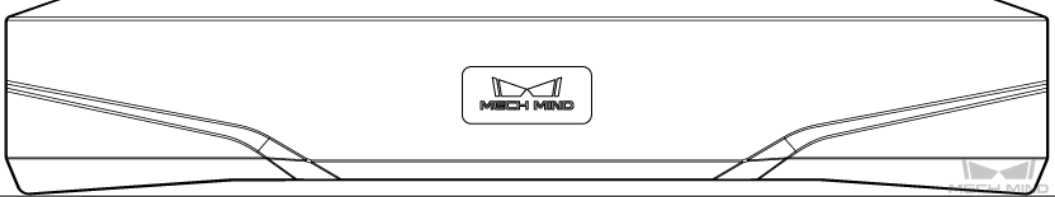

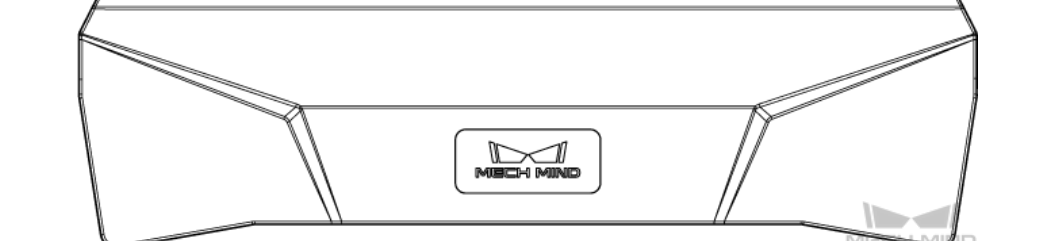
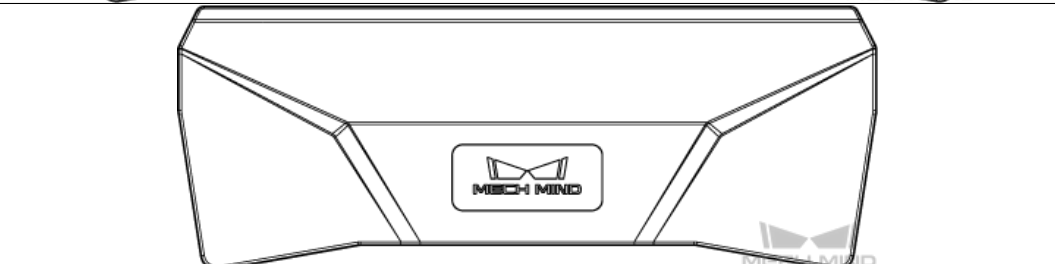
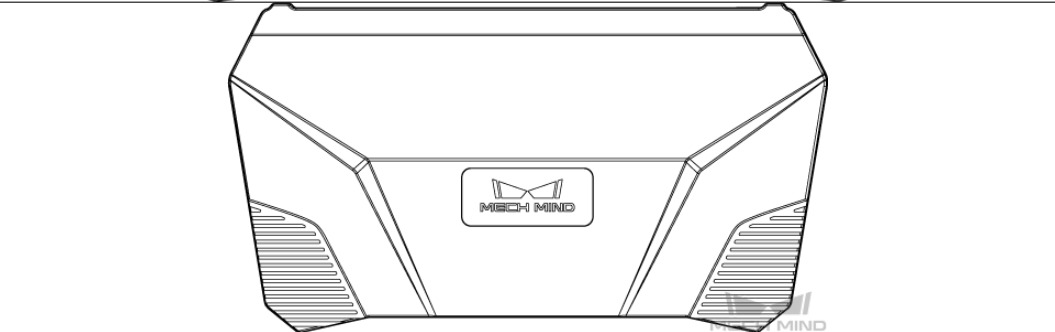
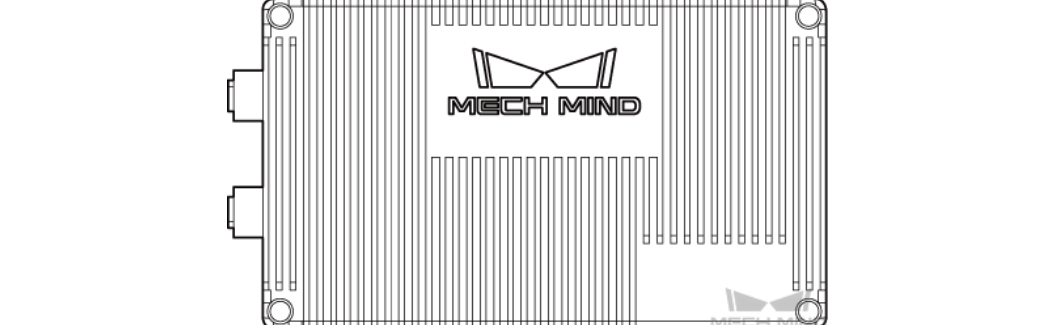
### 1.1.2 Characteristics and Suitable Applications

Different models have different characteristics that are suited for different applications. The following table summarizes the characteristics and suitable applications of the models.

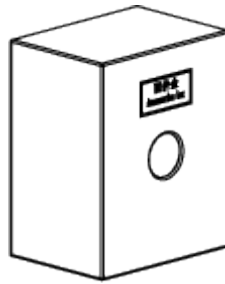
Model	Characteristics	Suitable applications
UHP-140 (V4)	<ul style="list-style-type: none"> <li>• Micron-level accuracy</li> <li>• Independently developed image fusion algorithm</li> <li>• Handles reflective objects well</li> </ul>	<ul style="list-style-type: none"> <li>• Automotive part manufacturing and assembly</li> <li>• Position error, gap, and flush inspection</li> </ul>
LSR L (V4)	<ul style="list-style-type: none"> <li>• High accuracy</li> <li>• Large field of view</li> <li>• High resistance to ambient light</li> </ul>	<ul style="list-style-type: none"> <li>• Manufacturing plants and other environments with ambient light interference</li> </ul>
Laser L (V3)		
Laser L Enhanced (V3)		
NANO (V4)	<ul style="list-style-type: none"> <li>• Small volume</li> <li>• High accuracy</li> <li>• High resistance to ambient light</li> </ul>	<ul style="list-style-type: none"> <li>• On-robot mounting</li> <li>• Positioning and assembly, high-precision picking, and other high-accuracy applications</li> </ul>
Nano (V3)		
Pro XS (V3)		
DEEP (V4)	<ul style="list-style-type: none"> <li>• Large field of view and deep depth of field</li> <li>• Fast data acquisition</li> <li>• Comprehensive, detailed, precise and color-accurate point cloud</li> </ul>	<ul style="list-style-type: none"> <li>• Logistics applications such as palletizing and depalletizing</li> </ul>
Deep (V3)		
Pro L Enhanced (V3)		
PRO S (V4)	<ul style="list-style-type: none"> <li>• High accuracy</li> <li>• Fast data acquisition</li> <li>• High resistance to ambient light</li> <li>• Color 2D camera variant available</li> </ul>	<ul style="list-style-type: none"> <li>• Medium-range applications with high accuracy requirements</li> <li>• Such as random picking, positioning, assembly, academic research</li> </ul>
Pro S Enhanced (V3)		
PRO M (V4)		
Pro M Enhanced (V3)		
Log M (V3)	<ul style="list-style-type: none"> <li>• Fast data acquisition</li> <li>• Suited for medium to long range</li> </ul>	<ul style="list-style-type: none"> <li>• Designed for logistics applications</li> <li>• Suited for product picking and parcel feeding</li> </ul>
Log S (V3)	<ul style="list-style-type: none"> <li>• Fast data acquisition</li> <li>• Suited for medium to long range</li> </ul>	<ul style="list-style-type: none"> <li>• Designed for logistics applications</li> <li>• Suited for product picking and parcel feeding</li> </ul>

## 1.2 Package Contents

Camera

Model	Diagram
LSR L	 <p>A line drawing of the LSR L camera, a long, thin rectangular unit with a central lens area and a MECH MIND logo on the front panel.</p>
DEEP	 <p>A line drawing of the DEEP camera, a wide, low-profile rectangular unit with a central lens area and a MECH MIND logo on the front panel.</p>
PRO M	 <p>A line drawing of the PRO M camera, a rectangular unit with a central lens area and a MECH MIND logo on the front panel.</p>
PRO S	 <p>A line drawing of the PRO S camera, a rectangular unit with a central lens area and a MECH MIND logo on the front panel.</p>
UHP-140	 <p>A line drawing of the UHP-140 camera, a rectangular unit with a central lens area and a MECH MIND logo on the front panel.</p>
NANO	 <p>A line drawing of the NANO camera, a small, square-shaped unit with a central lens area and a MECH MIND logo on the front panel.</p>

Accessory Box



User Manual

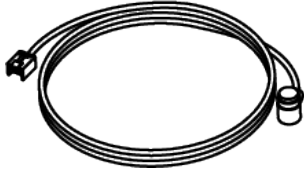

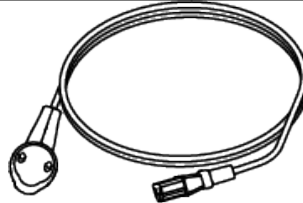
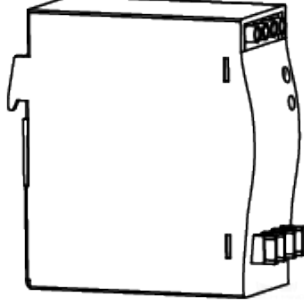

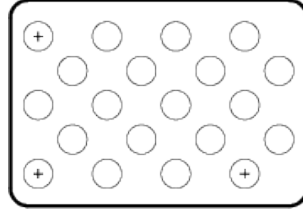


---

Hint:

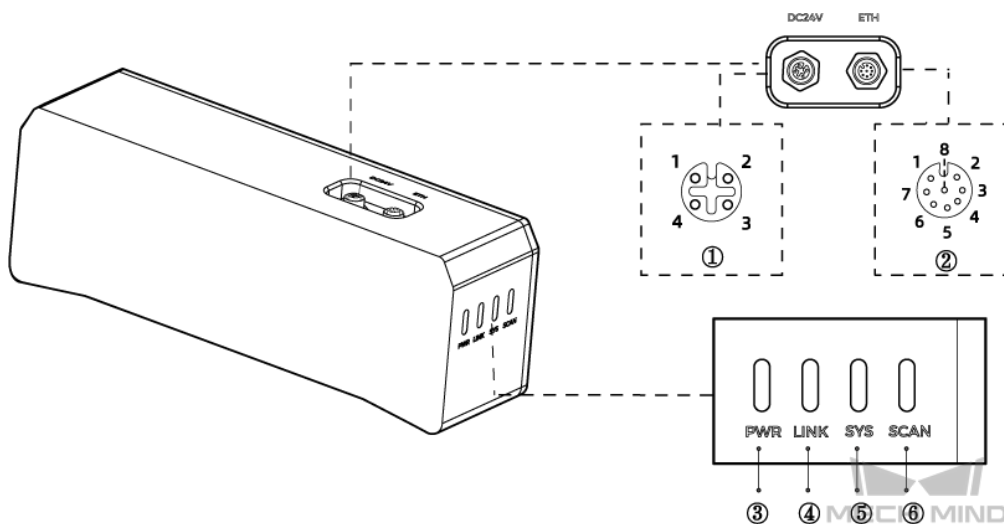
- The camera bracket is attached to the back of the camera when packaged.
  - Please contact us if there are any damages or missing parts.
-

### 1.2.1 Cables and Accessories

<p>Ethernet Cable</p> 	<p>Adapter</p> 	<p>AC Power Cable</p> 
<p>DIN Rail Power Supply</p> 	<p>DC Power Cable</p> 	<p>Calibration Board</p> 

**Hint:** All the above cables and accessories are optional; please select the type and length based on your actual needs. For detailed instructions on using, please see [Connect the Camera](#).

### 1.3 Indicator Lights and Ports



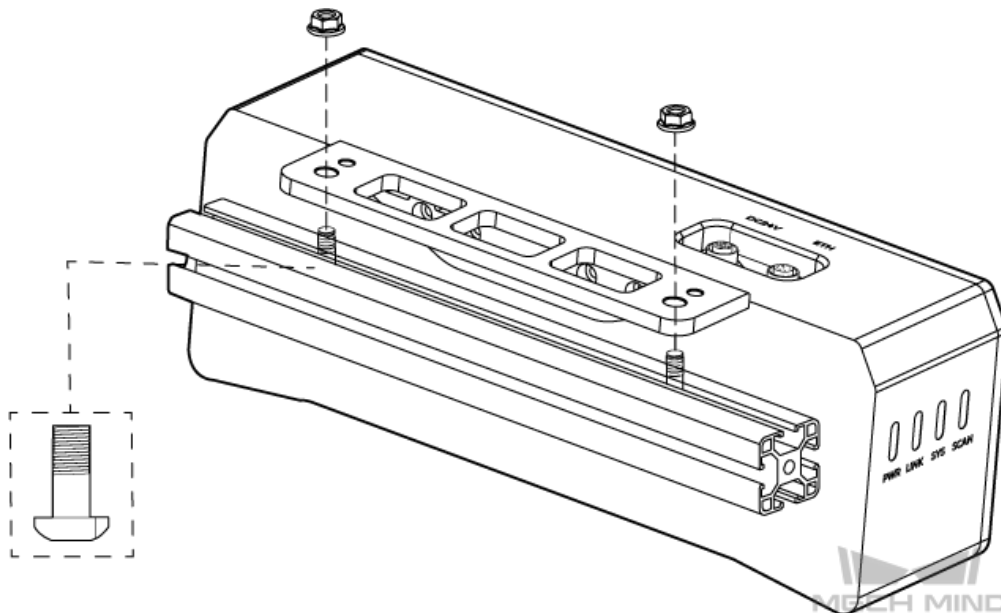
No.	Name	Function
1	DC 24V port	1: GND 2: GND 3: 24 V DC 4: 24 V DC
2	ETH port	1: MD3_P 2: MD2_N 3: MD2_P 4: MD0_P 5: MD1_P 6: MD0_N 7: MD3_N 8: MD1_N
3	PWR indicator light	Off: not connected to power Green: normal voltage Yellow: abnormal voltage, but system is still functioning Red: voltage too high 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

**Hint:** The above diagram is for illustration only; the actual product may differ.

## 1.4 Mount the Camera

### 1.4.1 Mount through the Camera Bracket

Use a wrench to tighten the two nuts, as shown below.

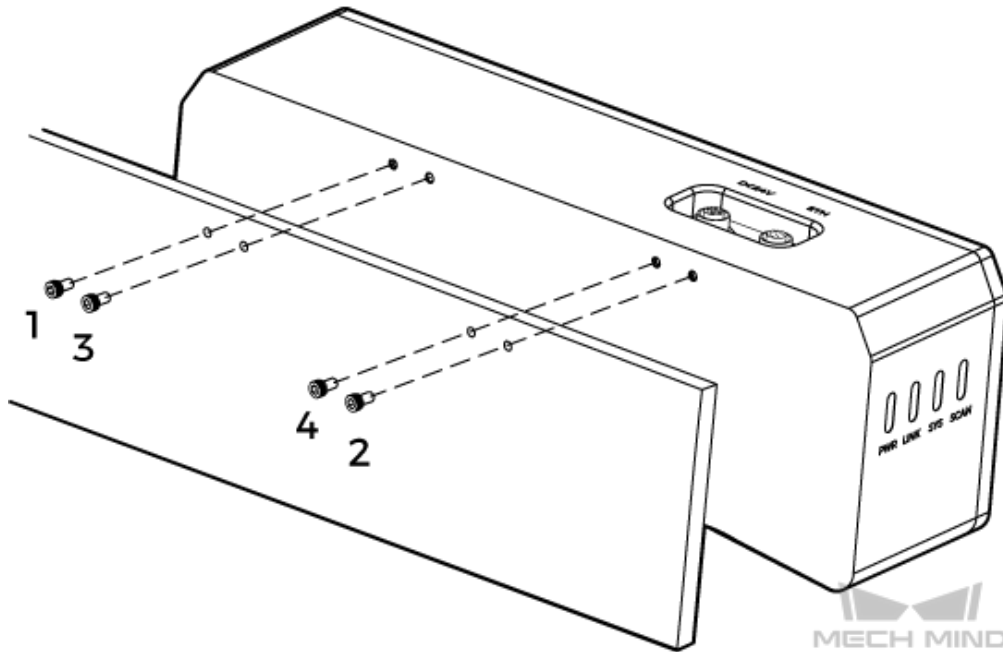


Hint:

- The camera bracket is attached to the back of the camera when packaged.
- The wrench is not included in the package; please prepare one beforehand.

### 1.4.2 Mount through Threaded Holes

Use a wrench to loosely screw on the bolts in the order specified below, and then fully tighten all the bolts.



Hint:

- Please remove the camera bracket with a wrench before mounting.
- The wrench is not included in the package; please prepare one beforehand.

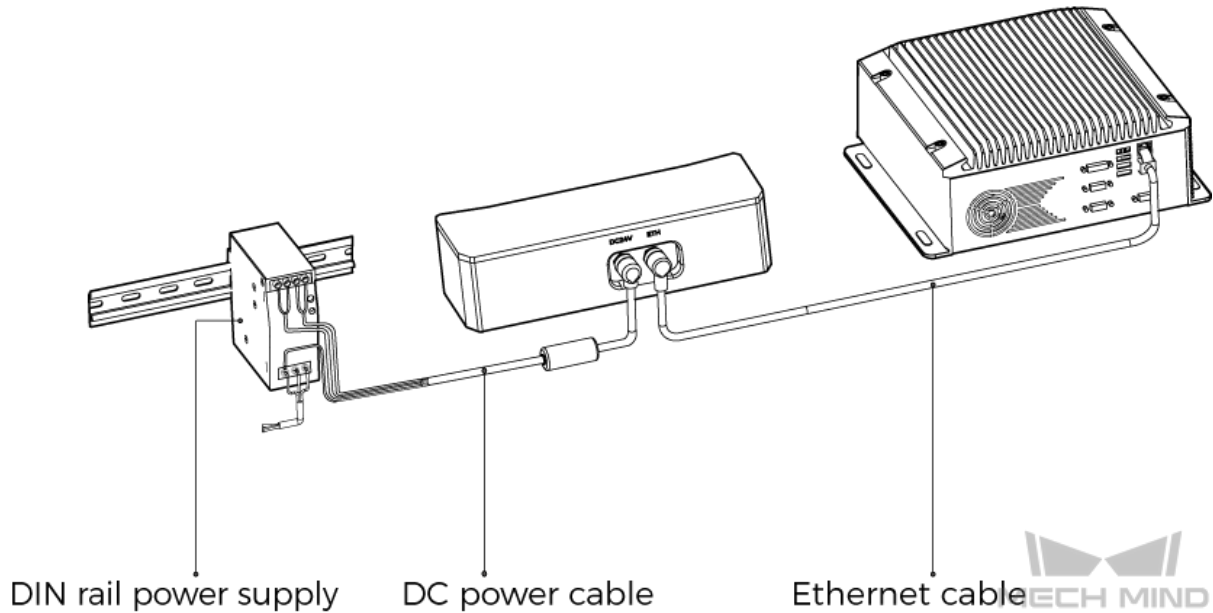
## 1.5 Connect the Camera

**Attention:**

- Please connect the power cable last. The PWR indicator light should turn on after the cables are connected. If the light is not on, please contact Technical Support.
- The recommended tightening torque for the bolts is 16 N • m.

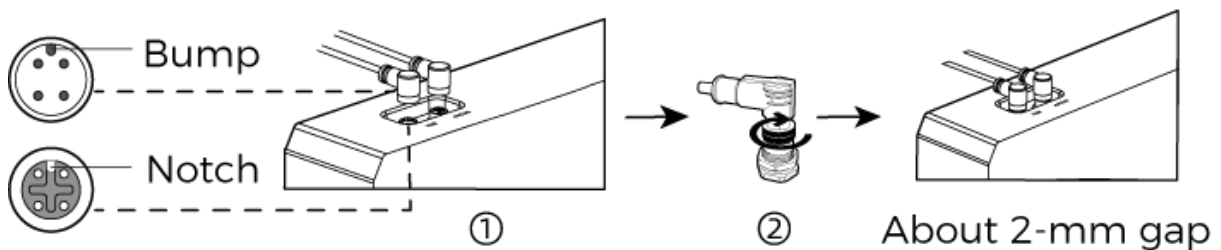
- The DIN rail power supply should be reliably grounded. If using multiple DIN rail power supplies, ensure enough distance in between.
- When mounting the camera onto a robot arm or other mobile equipment, please fasten the DC and Ethernet cables on the camera end properly to avoid damaging the cables or connectors due to strain.

- **Ethernet Cable:** Insert one end into the port on the IPC and the other into the **ETH port** on the camera.
- **Power Adapter:** Insert the DC power cable connector of the adapter/DIN rail power supply into the DC 24V port, as shown below.



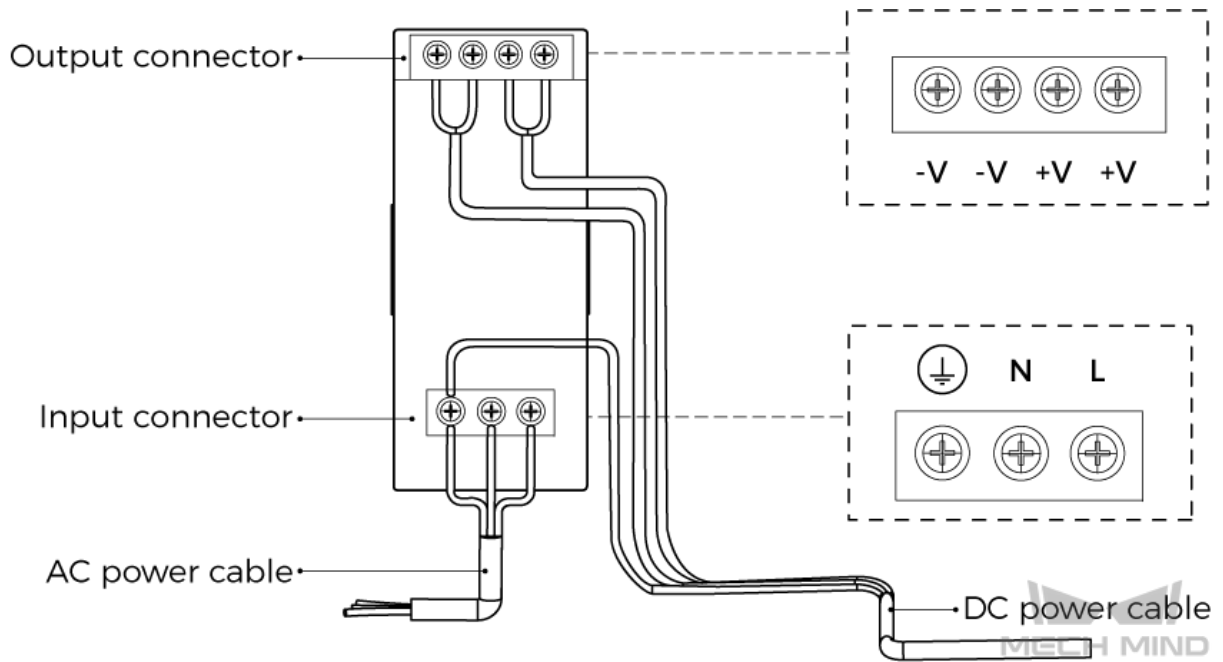
When inserting the Ethernet cable and power cable:

1. Make sure the bump on the connector is aligned with the notch of the port.
2. Tighten the nut after inserting the cable.



- DIN Rail Power Supply





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

- The AC power cable has three wires: L, N, and PE (⌚).
- The DC power cable has three wires: +V, -V and PE (⌚).

**Warning:** The PE wires must be properly grounded! Please install the DIN rail power supply inside a control cabinet.

**Note:** If multiple cameras or IPCs need to be connected, a network switch can be used.

## 1.6 Technical Specifications

Technical specifications of V4 cameras:

### 1.6.1 LSR L

#### Specifications

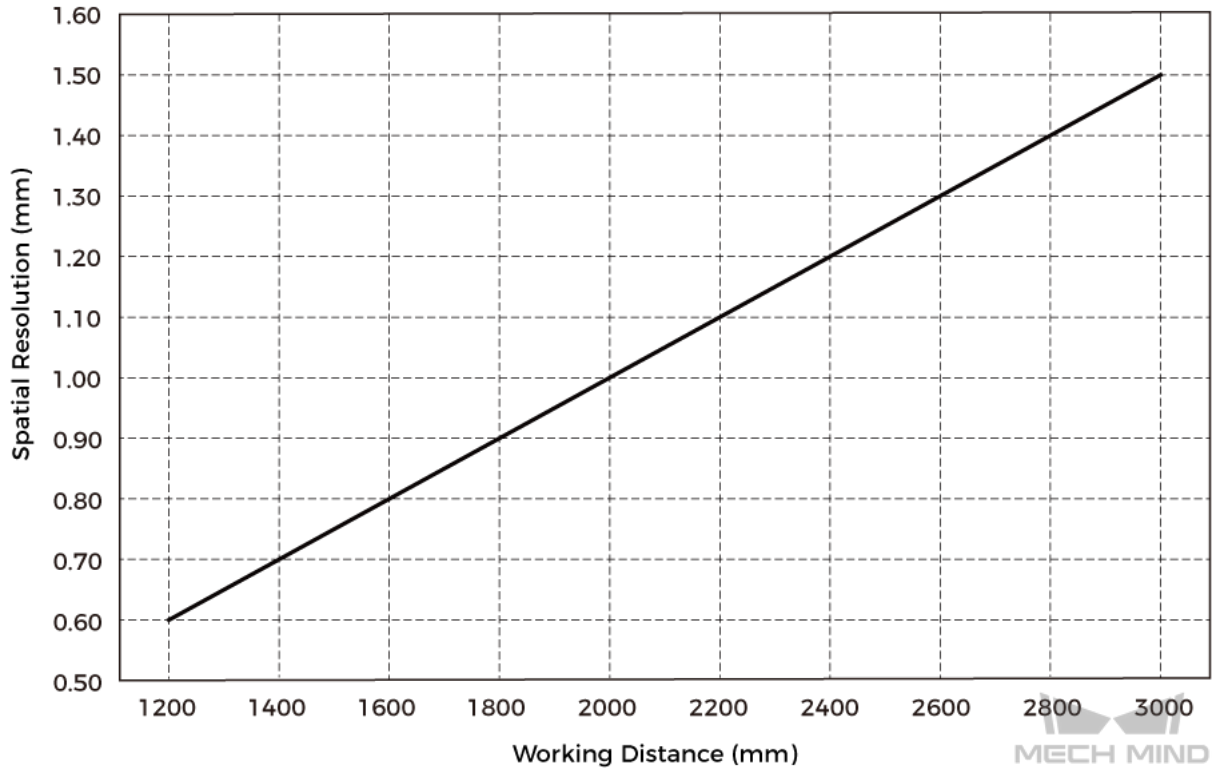
Product Name	Mech-Eye Industrial 3D Camera
Model	LSR L
Recommended Working Distance	1200–3000 mm
FOV	1200 × 1000 mm @ 1.2 m
	3000 × 2400 mm @ 3.0 m
Depth Map Resolution	2048 × 1536
RGB Resolution	4000 × 3000 / 2000 × 1500
Point Z-Value Repeatability ( $\sigma$ ) <sup>[1]</sup>	0.5 mm @ 3 m
Measurement Accuracy (VDI/VDE) <sup>[2]</sup>	1.0 mm @ 3 m
Typical Capture Time	0.5–0.9 s
Weight	About 2.9 kg
Baseline	About 380 mm
Dimensions	About 459 × 77 × 86 mm
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 / UKCA / KC
IP Rating <sup>[3]</sup>	IP65
Cooling	Passive

[1] One standard deviation of 100 Z-value measurements of the same point. The measurement target was a ceramic plate.

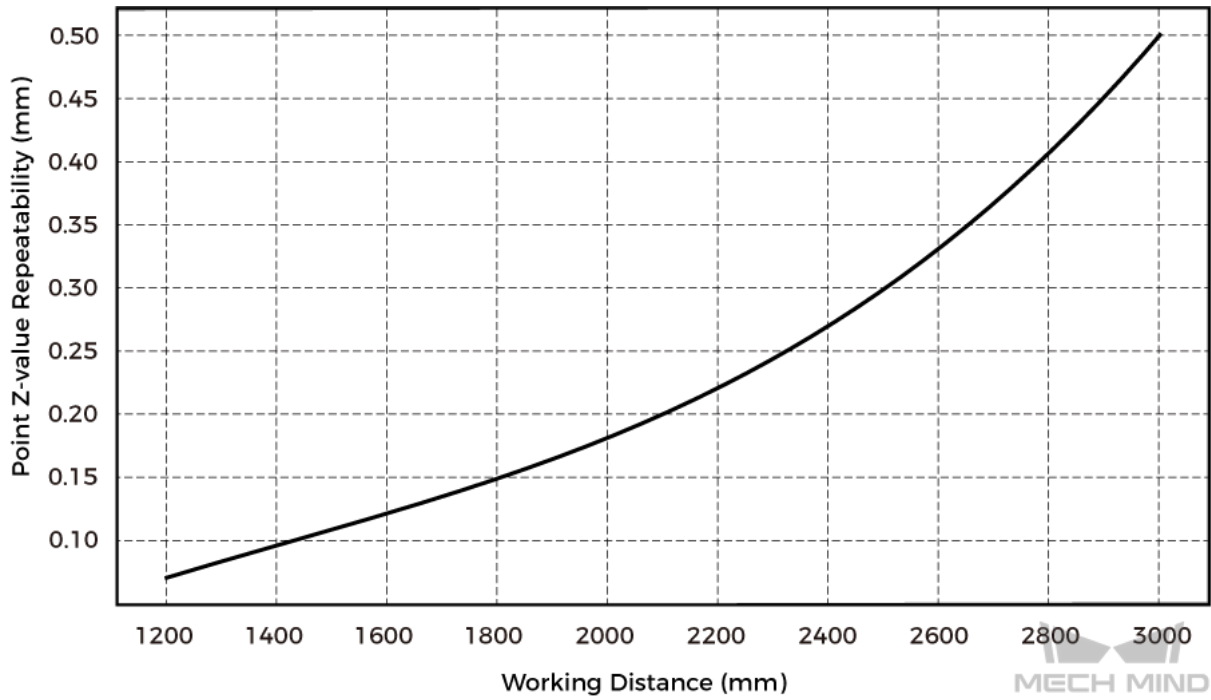
[2] According to VDI/VDE 2634 Part II.

[3] Test implemented based on IEC 60529. 6: dust-tight; 5: protected against water jets.

Spatial Resolution

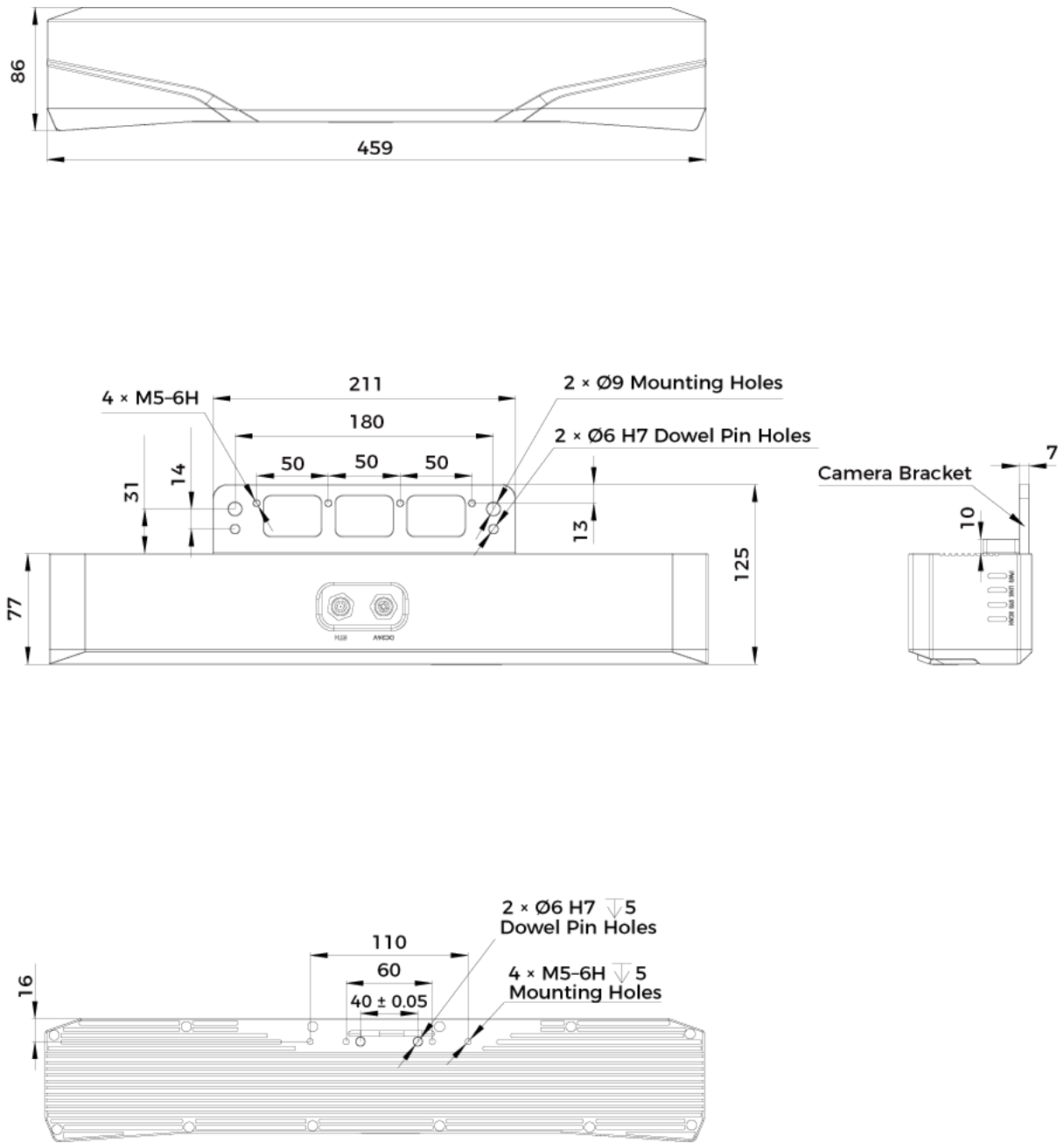


### Point Z-Value Repeatability



### Dimensions

Unit: mm



## 1.6.2 DEEP

### Specifications

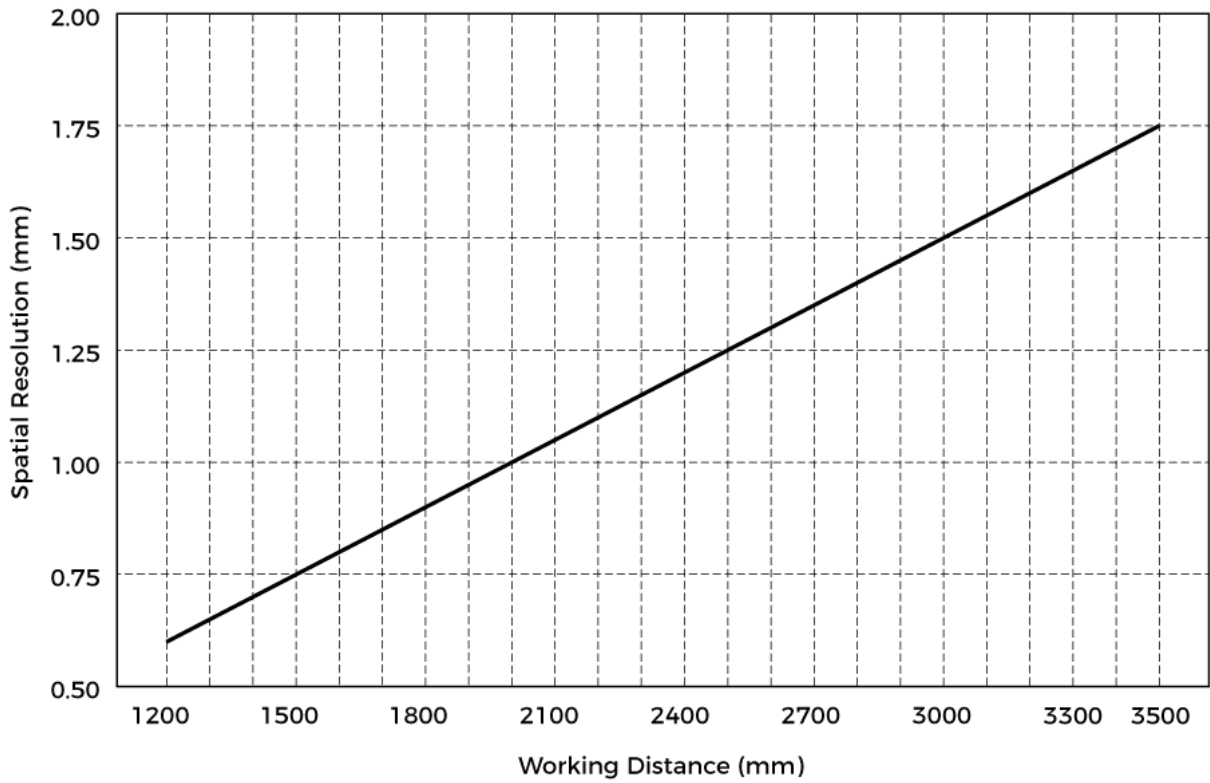
Product Name	Mech-Eye Industrial 3D Camera
Model	DEEP
Recommended Working Distance	1200–3500 mm
FOV	1200 × 1000 mm @ 1.2 m
	3500 × 2800 mm @ 3.5 m
Depth Map Resolution	2048 × 1536
RGB Resolution	2000 × 1500
Point Z-Value Repeatability ( $\sigma$ ) <sup>[1]</sup>	1.0 mm @ 3 m
Measurement Accuracy (VDI/VDE) <sup>[2]</sup>	3.0 mm @ 3 m
Typical Capture Time	0.5–0.9 s
Weight	About 2.4 kg
Baseline	About 300 mm
Dimensions	About 366 × 77 × 92 mm
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 / UKCA / KC
IP Rating <sup>[3]</sup>	IP65
Cooling	Passive

[1] One standard deviation of 100 Z-value measurements of the same point. The measurement target was a ceramic plate.

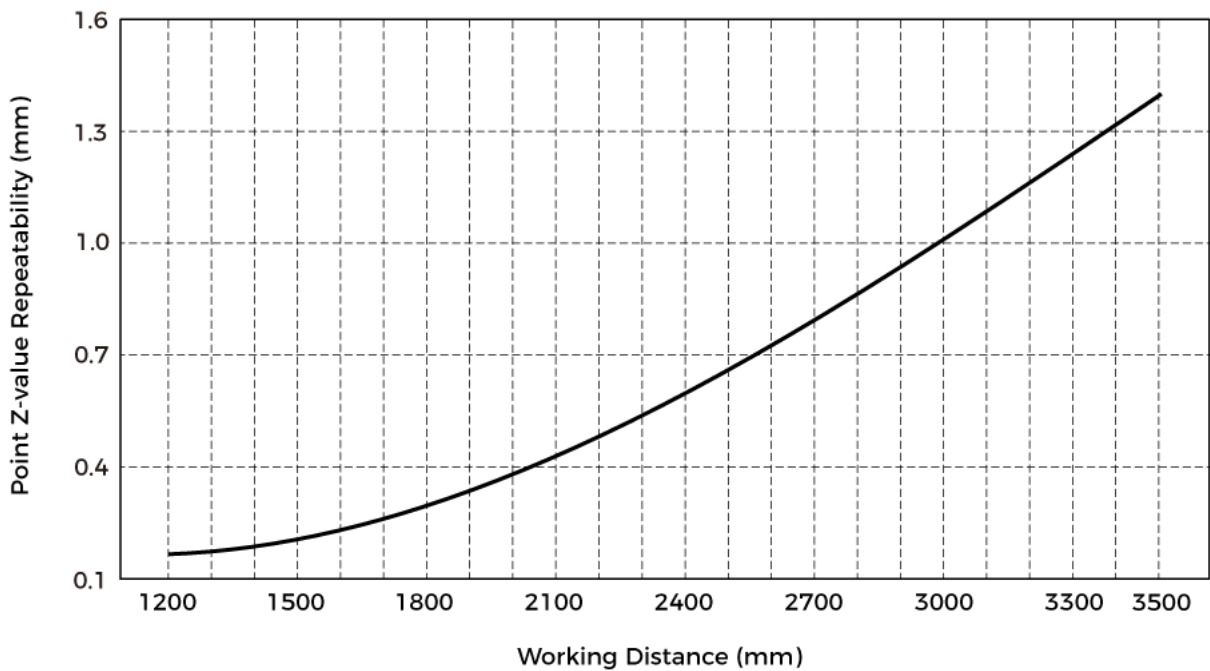
[2] According to VDI/VDE 2634 Part II.

[3] Test implemented based on IEC 60529. 6: dust-tight; 5: protected against water jets.

### Spatial Resolution

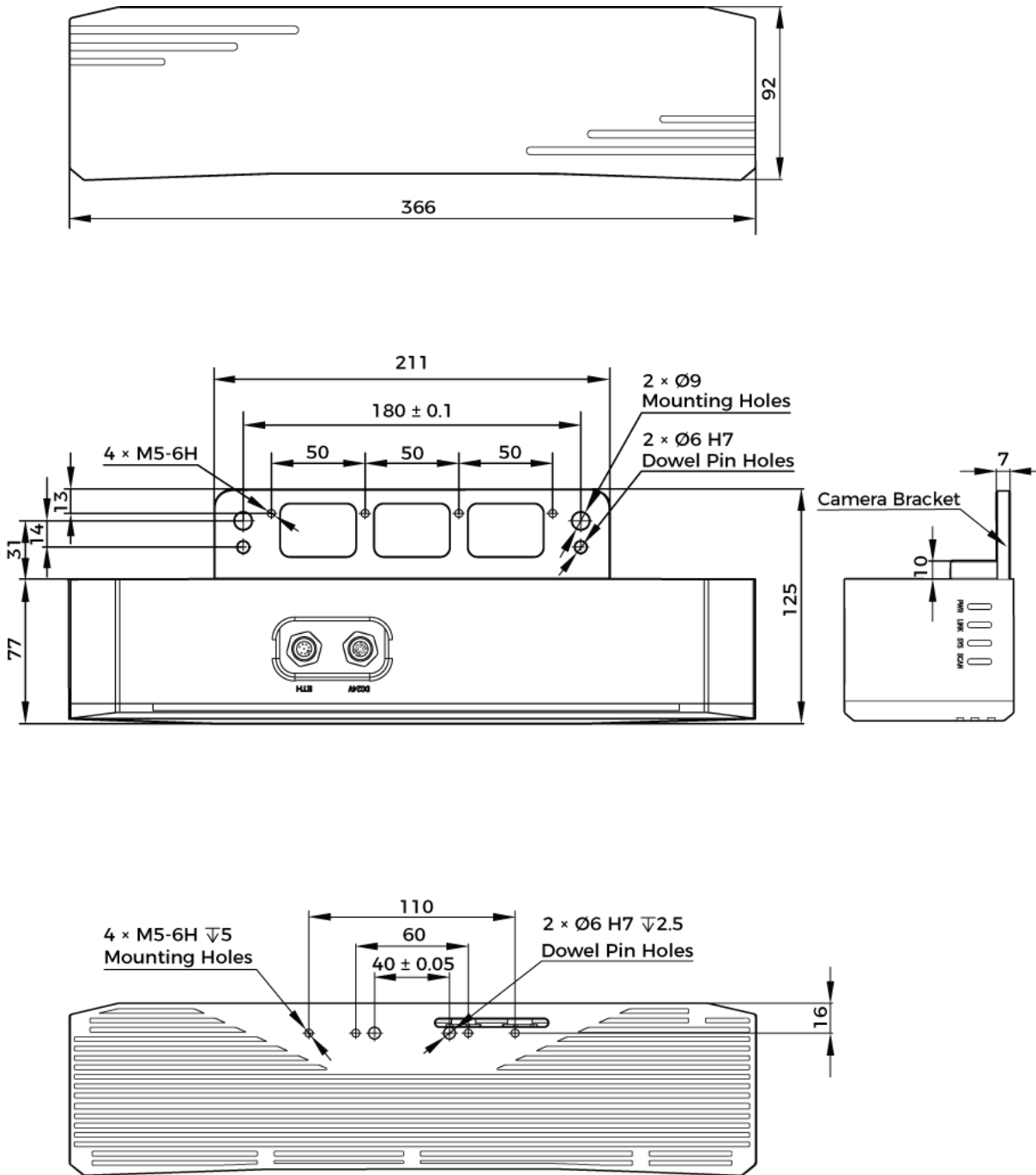


### Point Z-Value Repeatability



Dimensions

Unit: mm





### 1.6.3 PRO S & PRO M

#### Specifications

Product Name	Mech-Eye Industrial 3D Camera	
Model	PRO S	PRO M
Recommended Working Distance	500–1000 mm	1000–2000 mm
FOV	370 × 240 mm @ 0.5 m	800 × 450 mm @ 1 m
	800 × 450 mm @ 1 m	1500 × 890 mm @ 2 m
Resolution	1920 × 1200	
Megapixels	2.3 MP	
Point Z-Value Repeatability ( $\sigma$ ) <sup>[1]</sup>	0.05 mm @ 1 m	0.2 mm @ 2 m
Measurement Accuracy (VDI/VDE) <sup>[2]</sup>	0.1 mm @ 1 m	0.2 mm @ 2 m
Weight	About 1.6 kg	About 1.9 kg
Baseline	About 180 mm	About 270 mm
Dimensions	About 265 × 57 × 100 mm	About 353 × 57 × 100 mm
Typical Capture Time	0.3–0.6 s	
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 / UKCA / KC	
IP Rating <sup>[3]</sup>	IP65	
Cooling	Passive	

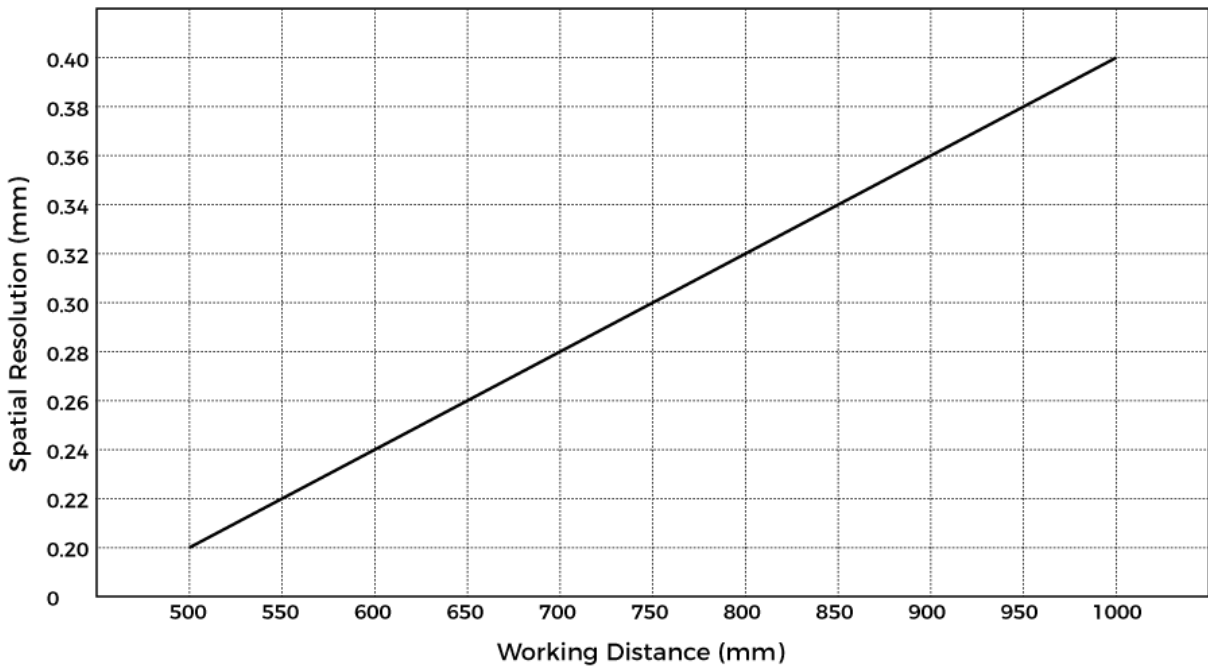
[1] One standard deviation of 100 Z-value measurements of the same point. The measurement target was a ceramic plate.

[2] According to VDI/VDE 2634 Part II.

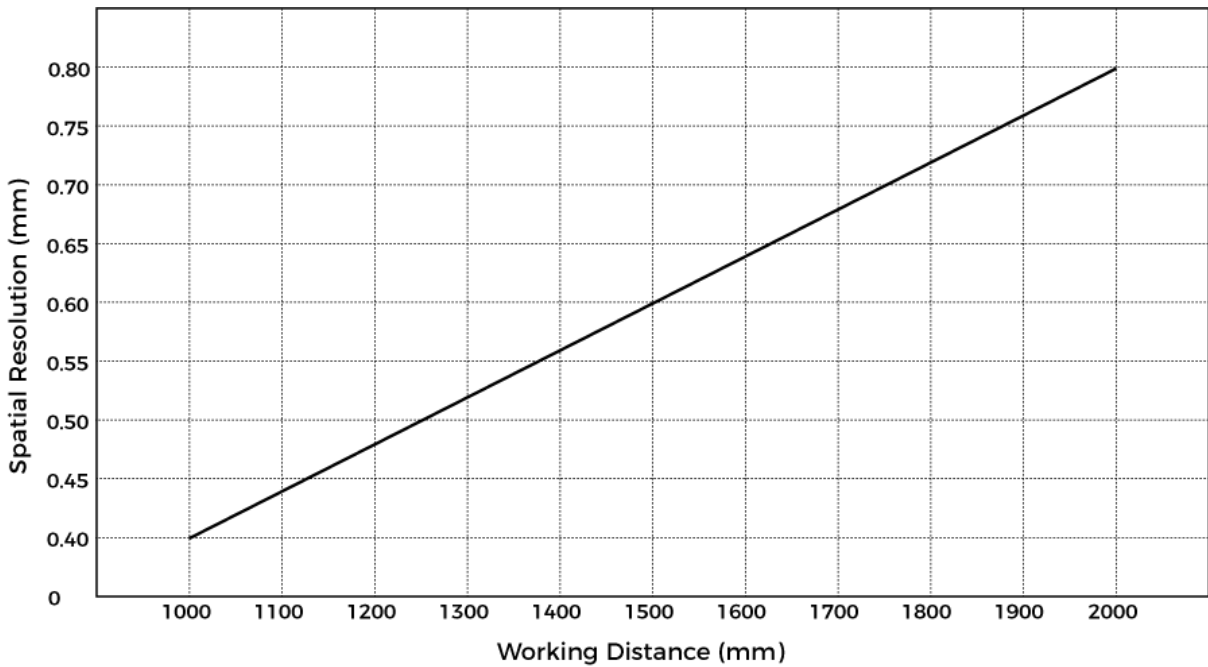
[3] Test implemented based on IEC 60529. 6: dust-tight; 5: protected against water jets.

Spatial Resolution

PRO S

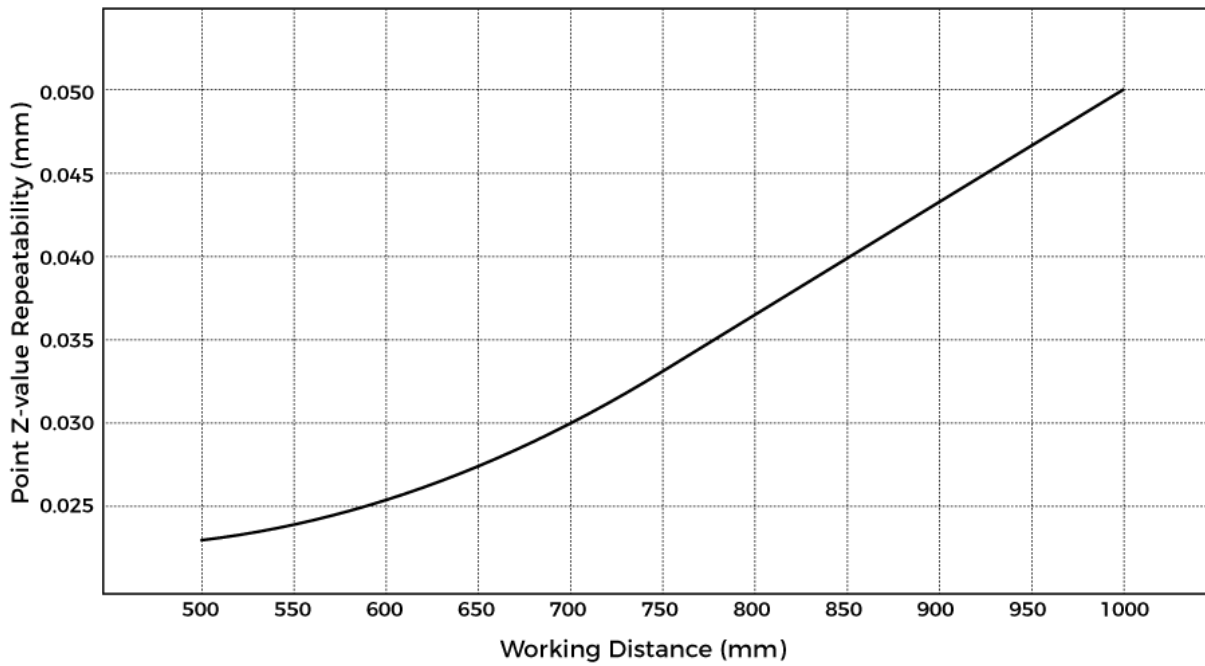


PRO M

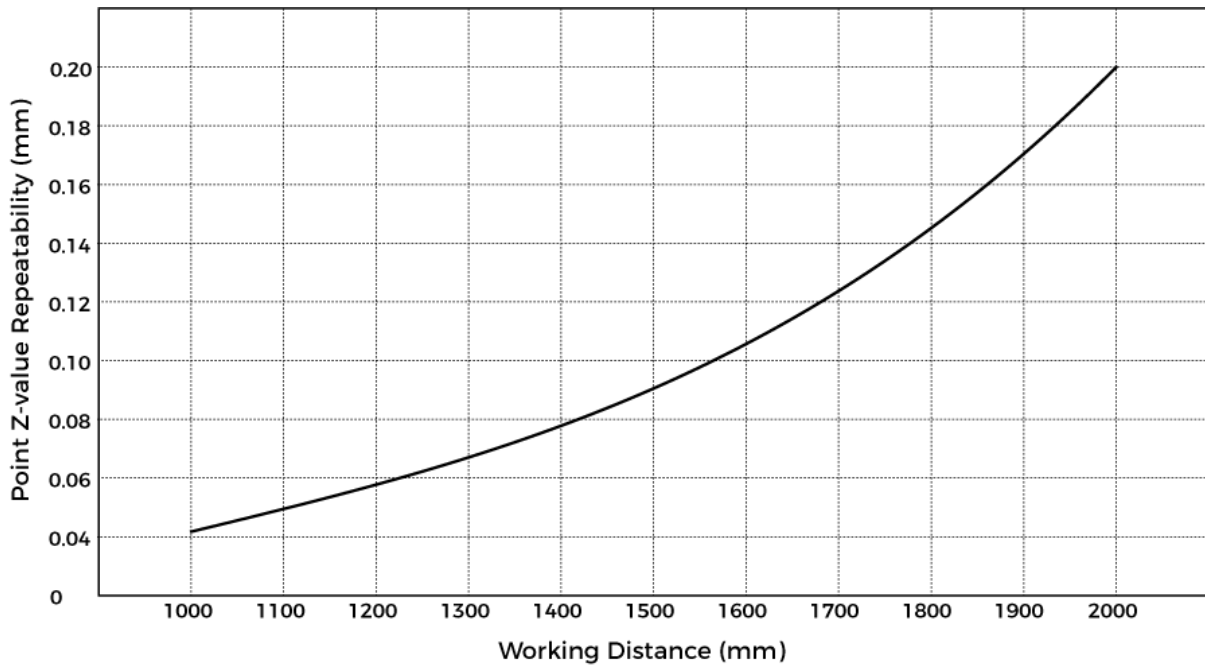


Point Z-Value Repeatability

PRO S



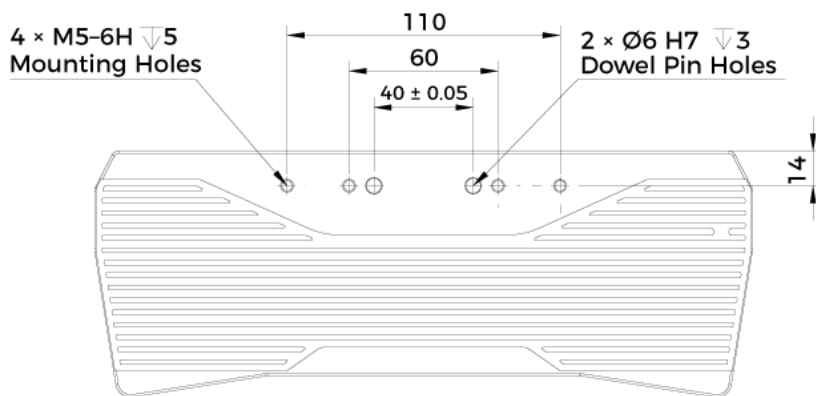
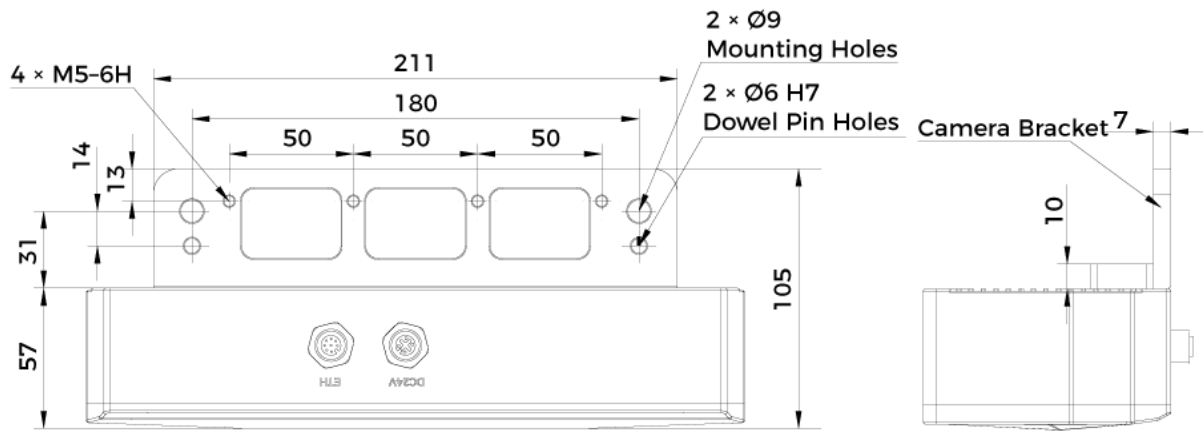
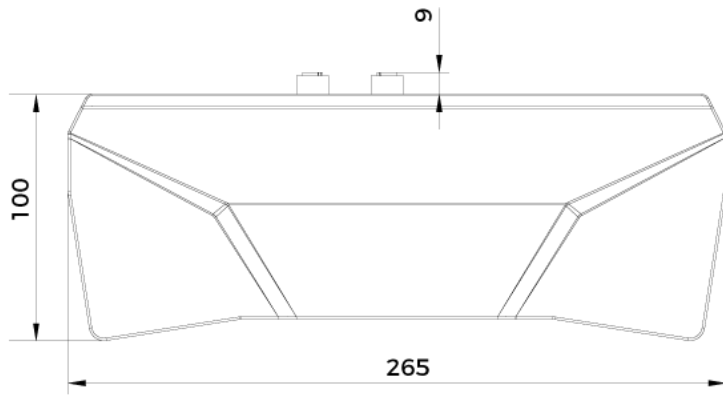
PRO M



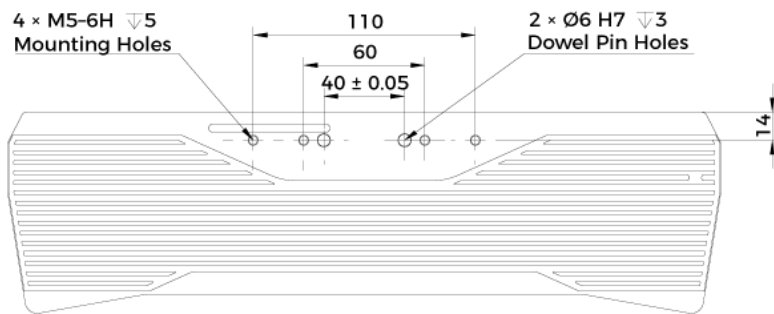
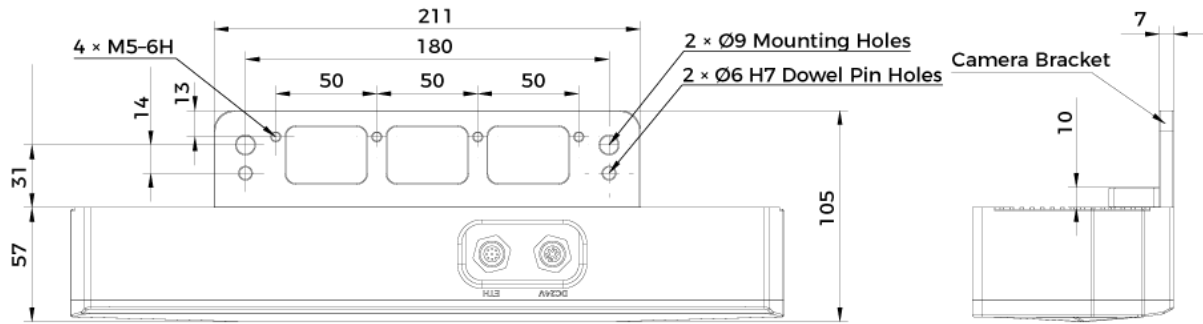
## Dimensions

Unit: mm

**PRO S**



**PRO M**



## 1.6.4 UHP-140

### Specifications

Product Name	Mech-Eye Industrial 3D Camera
Model	UHP-140
Recommended Working Distance	280–320 mm
FOV	135 × 90 mm @ 280 mm 150 × 100 mm @ 320 mm
Resolution	2048 × 1536
Megapixels	3 MP
Point Z-Value Repeatability ( $\sigma$ ) <sup>[1]</sup>	2.6 $\mu\text{m}$ @ 0.3 m
Regional Z-Value Repeatability ( $\sigma$ ) <sup>[2]</sup>	0.09 $\mu\text{m}$ @ 0.3 m
Measurement Accuracy (VDI/VDE) <sup>[3]</sup>	0.03 mm @ 0.3 m
Typical Capture Time	0.6–0.9 s
Weight	About 1.9 kg
Baseline	About 80 mm
Dimensions	About 260 × 65 × 142 mm
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 / UKCA / KC
IP Rating <sup>[4]</sup>	IP65
Cooling	Passive

[1] One standard deviation of 100 Z-value measurements of the same point. The measurement target was a ceramic plate.

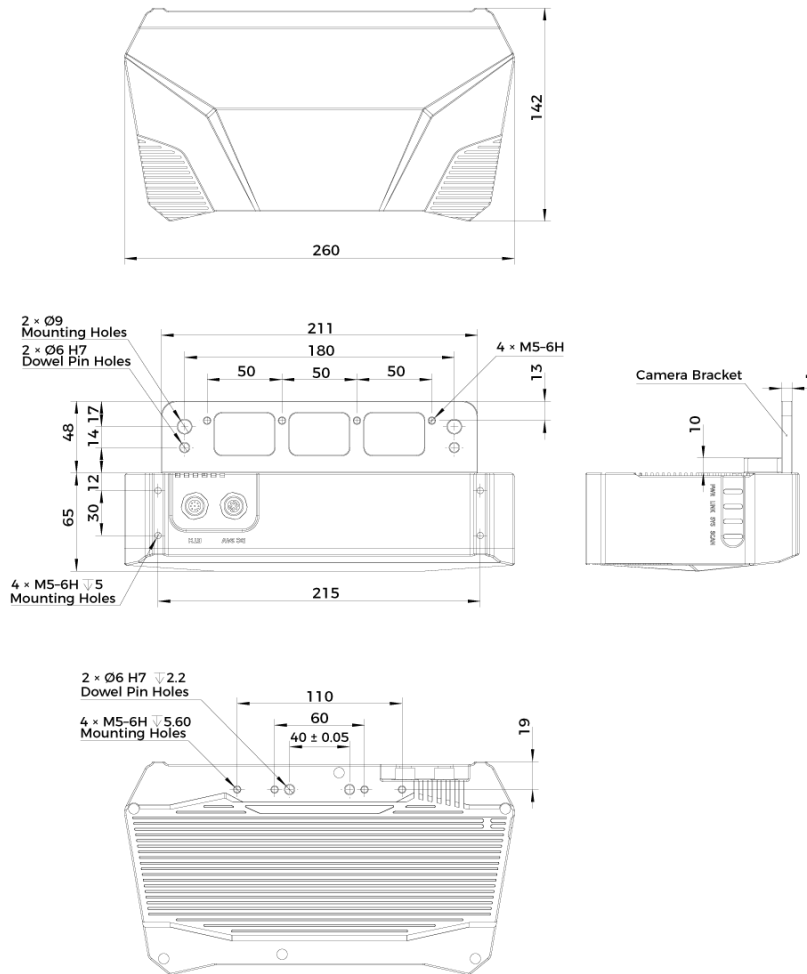
[2] One standard deviation of 100 measurements of the difference between the Z-value means of two same-sized regions. The measurement target was a ceramic plate.

[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



1.6.5 NANO



**Specifications**

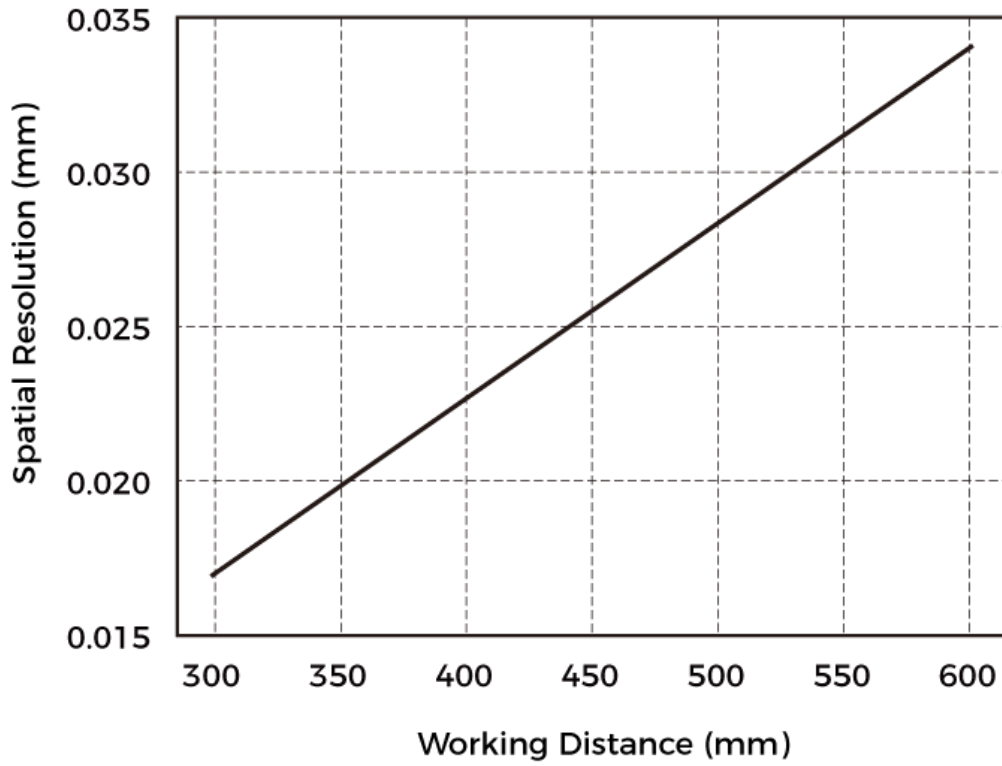
Product Name	Mech-Eye Industrial 3D Camera
Model	NANO
Recommended Working Distance	300–600 mm
FOV	220 × 150 mm @ 0.3 m
	440 × 300 mm @ 0.6 m
Resolution	1280 × 1024
Megapixels	1.3 MP
Point Z-Value Repeatability ( $\sigma$ ) <sup>[1]</sup>	0.1 mm @ 0.5 m
Measurement Accuracy (VDI/VDE) <sup>[2]</sup>	0.1 mm @ 0.5 m
Typical Capture Time	0.6–1.1 s
Weight	About 0.7 kg
Baseline	About 68 mm
Dimensions	About 145 × 51 × 85 mm
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 / UKCA / KC
IP Rating <sup>[3]</sup>	IP65
Cooling	Passive

[1] One standard deviation of 100 Z-value measurements of the same point. The measurement target was a ceramic plate.

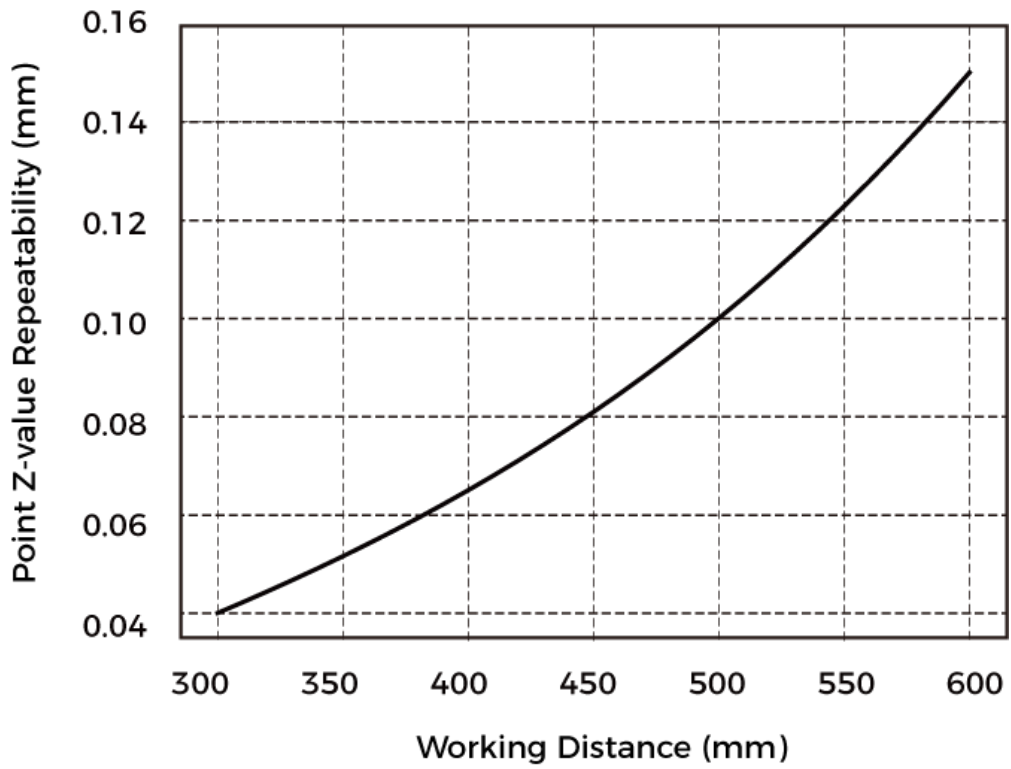
[2] According to VDI/VDE 2634 Part II.

[3] Test implemented based on IEC 60529. 6: dust-tight; 5: protected against water jets.

Dimensions

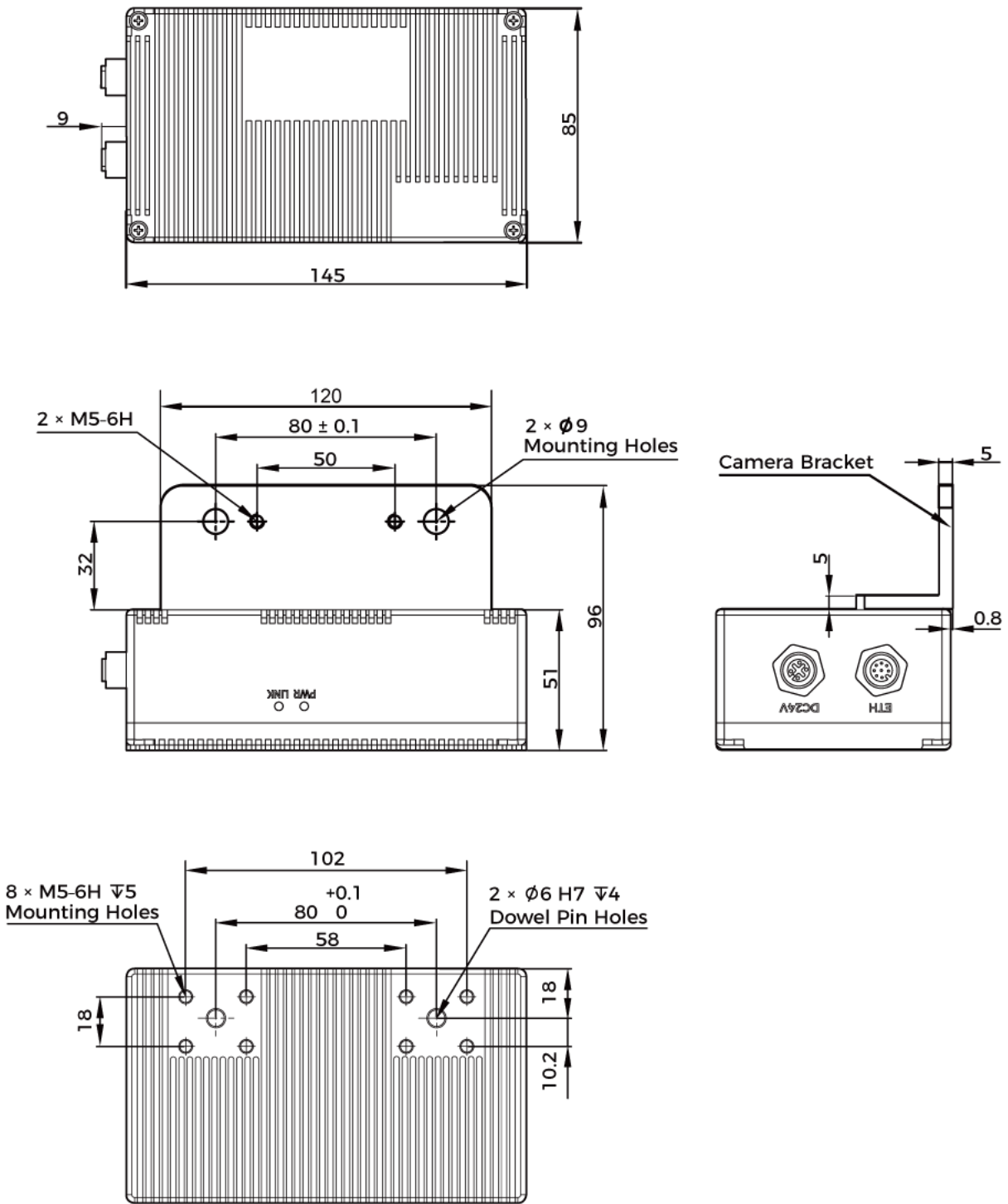


Point Z-Value Repeatability



Dimensions

Unit: mm



Technical specifications of V3 cameras:

## 1.6.6 Specifications (V3)

### Specifications of Non-Laser Structured Light Cameras

Product Name	Mech-Eye Industrial 3D Camera				
Model	Pro S Enhanced	Pro M Enhanced	Log S	Log M	Nano
Recommended Working Distance	500–1000 mm	800–2000 mm	500–1000 mm	800–2000 mm	300–600 mm
FOV	350 × 220 mm @ 0.5 m	500 × 350 mm @ 0.8 m	360 × 250 mm @ 0.5 m	520 × 390 mm @ 0.8 m	220 × 160 mm @ 0.3 m
	690 × 430 mm @ 1.0 m	1360 × 860 mm @ 2.0 m	710 × 490 mm @ 1.0 m	1410 × 960 mm @ 2.0 m	430 × 320 mm @ 0.6 m
Resolution	1920 × 1200	1920 × 1200	1280 × 1024	1280 × 1024	1280 × 1024
Megapixels	2.3 MP	2.3 MP	1.3 MP	1.3 MP	1.3 MP
Point Z-Value Repeatability ( $\sigma$ )	0.05 mm @ 1 m	0.2 mm @ 2 m	0.1 mm @ 1 m	0.3 mm @ 2 m	0.1 mm @ 0.5 m
Measurement Accuracy	0.1 mm @ 1 m	0.2 mm @ 2 m	0.2 mm @ 1 m	0.3 mm @ 2 m	0.1 mm @ 0.5 m
Typical Capture Time	0.5–0.8 s	0.5–0.8 s	0.3–0.5 s	0.3–0.5 s	0.6–1.1 s
Baseline	150 mm	280 mm	150 mm	280 mm	68 mm
Dimensions	About 270 × 72 × 130 mm	About 387 × 72 × 130 mm	About 270 × 72 × 130 mm	About 387 × 72 × 130 mm	About 145 × 51 × 85 mm
Weight	About 2.2 kg	About 2.4 kg	2.2 kg	2.4 kg	About 0.7 kg
Operating Temperature	0–45°C				
Communication Interface	Ethernet				
Operating Voltage	24V DC				
Safety and EMC	CE/FCC/VCCI				
IP Rating	IP65				
Cooling	Passive				

**Specifications of Laser Structured Light Cameras**

Product Name	Mech-Eye Industrial 3D Camera	
Model	Laser L	Laser L Enhanced
Recommended Working Distance	1500–3000 mm (59.1–118.1 inches)	
FOV	1500 × 1200 mm @ 1.5 m (59.1 × 47.2 inch @ 4.9 ft)	
	3000 × 2400 mm @ 3.0 m (118.1 × 94.5 inch @ 9.8 ft)	
Resolution	2048 × 1536	4096 × 3000
Megapixels	3.0 MP	12.0 MP
Point Z-Value Repeatability ( $\sigma$ )	0.5 mm @ 3 m (0.02 inch @ 9.8 ft)	0.5 mm @ 3 m (0.02 inch @ 9.8 ft)
Measurement Accuracy	1.0 mm @ 3 m (0.04 mm @ 9.8 ft)	0.5 mm @ 3 m (0.02 inch @ 9.8 ft)
Typical Capture Time	0.5–0.9 s	1.4–1.7 s
Weight	About 3.7 kg (8.2 lb)	About 3.9 kg (8.6 lb)
Baseline	About 400 mm (15.7 inch)	
Dimensions	About 459 × 89 × 121 mm (18.1 × 3.5 × 47.6 inch)	
Operating Temperature	-10–45°C (14°F–113°F)	
Communication Interface	Ethernet	
Power Supply	24 V DC	
Max. Power	70 W	
Laser Class	Class 2	
Safety and EMC	CE / FCC / VCCI	
IP Rating	IP65	
Cooling	Passive	

The laser emitted by Laser series cameras can be classified as Class 2 or Class 3R depending on the local laser safety standards.

## 1.7 Safety, Regulations, and Maintenance Instructions

This chapter includes safety, maintenance and regulatory information of the camera. Please read through the following contents carefully before using the camera to reduce risks and hazards.

---

Safety instructions of the camera:

*[Safety Instructions](#)*

---

Certifications of the camera:

*[Certifications](#)*

---

Maintenance instructions of the camera:

*[Maintenance](#)*

**Trademark and legal statement:**

*Trademark and Legal Statement*

### 1.7.1 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 can not totally eliminate all risks.
- Every step has been inspected during the drafting of this user manual. Please do not hesitate to contact Mech-Mind if you find any problems or mistakes in the manual.
- This 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 camera to open fire or high temperature.
- Do not throw, drop, or collide the camera with other objects. The camera may be damaged by strong shock or vibration. It is PROHIBITED to modify the camera in any form. 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, wood chips, etc., to enter the camera. Failure to do so may lead to fire, electric shock, malfunction, etc.
- Do not use the camera at extremely high or low temperature. The operating temperature range is -10–45°C for the LSR and DEEP series cameras, and 0–45°C for other series of Mech-Eye Industrial 3D Camera.
- Please use the camera indoors.
- Please use the camera at elevations below 4,000 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 if the power port, 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.
- For power, please use a 24 V isolated adapter with an output power of at least 90 W or a 24 V isolated DIN rail power supply.
- 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.
- Use copper conductors only.
- 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.




### 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. Failure to do 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.
- 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. Please do not dispose of the camera irresponsibly. Improper disposal may pollute the environment.
- The pollution degree of the camera is 3.

**Note:**  This icon indicates that failure to follow the instruction may lead to injury or death.

### 1.7.2 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.

- European Electromagnetic Compatibility Standards.
- U.S. ANSI C63.4 and 47 CFR PART 15B.
- Canada ICES-003.
- Japan VCCI-CISPR 32: 2016.

### Safety of Laser Products

The Laser classification is implemented based on IEC 60825-1 in accordance with the requirements of Laser Notice No. 56 of the FDA (CDRH).

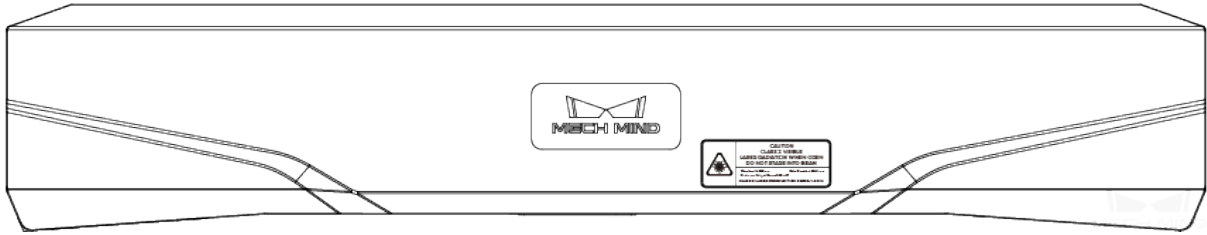
Model	Wavelength	IEC60825-1	
		Maximum output power	Laser class
LSR L	638 nm	2.46 mW	Class 2
DEEP	638 nm	2.46 mW	Class 2

### Warning Label

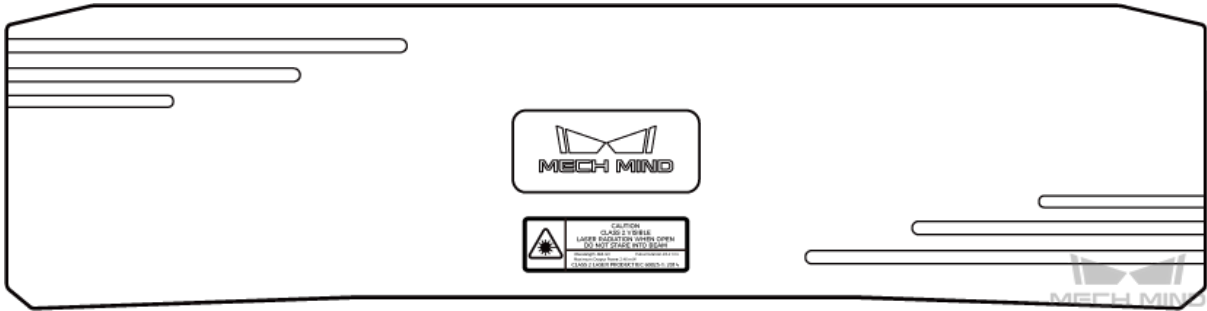


### Warning Label Attachment

LSR L



DEEP



### Conformit  Europ enne (CE) Certifications



Hereby [Mech-Mind Robotics Technologies Ltd.] declares that [LSR L, PRO S, PRO M and UHP-140] is in compliance with the Electromagnetic Compatibility Directive 2014/30/EU.

The full text of the EU Declaration of Conformity is available [here](#).

### European Electromagnetic Compatibility Standards

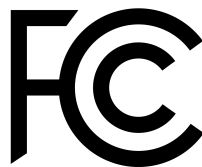
- EN 55032: 2015 + A1: 2020
- EN IEC 61000-3-2: 2019 + A1: 2021
- EN 61000-3-3: 2013 + A1: 2019 + A2: 2021
- EN 55035: 2017 + A11: 2020

### Important WEEE Information



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.

### Federal Communications Commission (FCC) Certifications



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.”

**Voluntary Control Council for Interference by Information Technology Equipment (VCCI) Certifications**

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.

**UK Conformity Assessed (UKCA) Certification****1.7.3 Maintenance****Cleaning the Camera**

When cleaning the surface of the camera, please blow off the dust gently, and then use a clean soft cloth to gently wipe the surface. When cleaning the camera lens, a liquid lens cleaner and clean lint-free cloth can be used to carefully wipe the lens to avoid scratching.

**Warning**

- 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 camera. Water entering the camera may lead to malfunction, fire, or even explosion. Water or other liquid damage to the camera is not covered by the warranty.


**Storing the Camera**

This product is rated as IP65. The enclosure of the camera can prevent dust from entering and affecting the functions of the camera. Soaking the camera in water or placing it in a highly humid environment may lead to malfunction. Rusting of the internal unit will lead to irreversible damage. When not using, please store the camera in an indoor, dry, cool and well ventilated place. To avoid damage caused by rain, snow, and other undesirable conditions, please do not place the camera outdoors for an extended period of time. Storage temperature is -20–60°C.

**Warning**

- Disconnect the camera from the power supply when storing to avoid fire.
- Do not leave the lens pointed at the sun or other intense light source for an extended period of time. Intense light may cause the image sensor to deteriorate and produce a white blur effect in images.

---

**Note:**  This icon indicates that failure to follow the instruction may lead to injury or death.

---

#### 1.7.4 Trademark and Legal Statement

Mech-Mind and Mech-Mind series logos including  are registered trademarks of Mech-Mind Robotics Technologies Ltd. ("Mech-Mind") and other related entities.

Unless authorized in writing in advance by 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 in writing by Mech-Mind, this user manual shall not be reproduced, modified, reprinted 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 in writing by Mech-Mind, the contents of the user manual may not be used for any other purposes.

## MECH-EYE SDK 2.0 RELEASE NOTES

This topic introduces the new features, improvements and resolved issues in Mech-Eye SDK 2.0.

### 2.1 Mech-Eye SDK 2.0.2 Release Notes

This topic introduces the new features, improvements and resolved issues in Mech-Eye SDK 2.0.2.

#### 2.1.1 New Features

##### Mech-Eye Viewer

##### Added Display of Connected Status

If a camera is already connected by a GenICam client, the status **Connected** is displayed in the info card of the camera in **Camera List**.

To connect the camera to Mech-Eye Viewer, please disconnect the camera from the GenICam client first.

##### Mech-Eye API

##### Added Method and Sample Program for 2D Image - Depth Map Mapping

New methods for mapping the 2D image to the depth map are added in Mech-Eye API 2.0.2, and a corresponding sample program is provided.

With these methods, you can input a masked 2D image and the corresponding depth map, and obtain the point cloud of the wanted area within the mask.

The corresponding sample program sets a mask for the 2D image using the ROI function of the camera, and generates point clouds of the wanted area from the last image data captured by the camera. You can also input locally saved 2D image (masked) and depth map files to generate point clouds.

New methods:

- C++:
  - Obtain the untextured point cloud of the wanted area within the mask:

```

ErrorStatus MechEyeDevice::getCloudFromTextureMask(DepthMap &, ColorMap &,
↳DeviceIntri &, PointXYZMap &)

```

- Obtain the textured point cloud of the wanted area within the mask:

```

ErrorStatus MechEyeDevice::getCloudFromTextureMask(DepthMap &, ColorMap &,
↳DeviceIntri &, PointXYZBGRMap &)

```

- C#:

- Obtain the untextured point cloud of the wanted area within the mask:

```

ErrorStatus MechEyeDevice.GetCloudFromTextureMask(mmind.apiSharp.DepthMap, mmind.
↳apiSharp.ColorMap, mmind.apiSharp.DeviceIntri, ref mmind.apiSharp.PointXYZMap)

```

- Obtain the untextured point cloud of the wanted area within the mask:

```

ErrorStatus MechEyeDevice.GetBGRCloudFromTextureMask(mmind.apiSharp.DepthMap, mmind.
↳apiSharp.ColorMap, mmind.apiSharp.DeviceIntri, ref mmind.apiSharp.PointXYZBGRMap)

```

- Python:

- Obtain the untextured point cloud of the wanted area within the mask:

```

Device.get_cloud_from_texture_mask(self, depth, textureMask, intri)

```

- Obtain the untextured point cloud of the wanted area within the mask:

```

Device.get_bgr_cloud_from_texture_mask(self, depth, textureMask, intri)

```

New sample programs:

- C++: CapturePointCloudFromTextureMask
- C#: CapturePointCloudFromTextureMask
- Python: CapturePointCloudFromTextureMask

### Added C++ Sample Program for Saving HALCON-Readable Point Clouds

A new C++ sample program that *saves HALCON-readable point clouds* is added in Mech-Eye API 2.0.2. Through this sample program, you can obtain point clouds faster than through HALCON, and it is possible to obtain textured point clouds directly. Then you can read the point clouds into HALCON and perform further processing.

## 2.1.2 Improvements

### Improved Capture Speed of Camera

After the firmware of laser cameras is upgraded to 2.0.2, the capture speed of the camera is improved.

### Improved Performance of Accurate Coding Mode for Nano (V3) and NANO (V4)

After the camera firmware of Nano (V3) and NANO (V4) is upgraded to 2.0.2, the quality of the point cloud obtained when **Fringe Coding Mode** is set to **Accurate** is improved. Fewer outliers are present in the point cloud under complex conditions such as multiple reflections.

### Mech-Eye Viewer

#### Improved Camera IP Address Configuration

*The IP address configuration function* is improved in Mech-Eye Viewer 2.0.2. The IP address and subnet mask of the IPC network interface card connected with the camera are now automatically displayed. In addition, the **Interface** menu in the **Camera** section is renamed to the more accurate **IP Address Class**.

#### Added Calibration Board Models

Options of the calibration board models in the CGB series are added to the **Check Camera Intrinsic Parameter** tool.

### Mech-Eye API

#### All Cameras Connected to Different Network Interface Cards Can Be Found

When the IPC is equipped with multiple network interface cards, Mech-Eye API 2.0.2 can find all the cameras connected to all the network interface cards.

#### Improved Python Code Structure and Sample Programs

The Python code structure and sample programs are improved in Mech-Eye API 2.0.2. The newest sample programs can be downloaded from [GitHub](#).



## GenICam

### Added the GevSCPD Parameter

After the camera firmware is upgraded to 2.0.2, the **GevSCPD** parameter is available in GenICam clients. You can adjust the inter-packet delay through this parameter. Increasing the value of this parameter increases the inter-packet delay, reduces packet loss but also reduces the data transmission speed.

### 2.1.3 Resolved Issues

The following issues are resolved in Mech-Eye Viewer 2.0.2:

- When adjusting the size of ROI selection box with the handles on the box, the position of the box also changed.
- If the 2D camera was switched after *ViewOn* had been clicked in **View 2D Camera and Set Parameters**, the displayed image was not automatically fit to the window.
- Software occasionally crashed when the parameters were adjusted.
- After selecting **Full Pallet Simulator** in the **More Tools** window of the toolbar, **Camera Firmware Upgrade** and **Camera Controller** were also displayed in the toolbar.
- The IP address configuration window could be opened for a virtual camera started by loading the camera raw data, even though the camera IP address cannot be configured.
- In **Depth Analyzer**, when the line was drawn along the direction of lower-left to upper-right, the plotted curve reflected the depth fluctuation along the direction of upper-left to lower-right.

## 2.2 Mech-Eye SDK 2.0.1 Release Notes

This section introduces the resolved issues in Mech-Eye SDK 2.0.1.

### 2.2.1 Resolved Issues

The following issues are resolved in Mech-Eye SDK 2.0.1.

- In point clouds saved through GenICam clients, the coordinates of the points deviated in all directions.
- PRO series (V4), NANO (V4) and Pro XS (V3): when **Exposure Time** in **3D Parameters** was set to 15 ms, the depth map and point cloud data were occasionally lost.
- LSR series (V4): The white-balance parameter value was reset after the camera was powered off.
- Data containing chunk format could not be correctly transmitted to some GenICam clients.

## 2.3 Mech-Eye SDK 2.0.0 Release Notes

This section introduces the new features, improvements and resolved issues in Mech-Eye SDK 2.0.0.

### 2.3.1 New Features

#### Compliance with GenICam Standard

Starting from Mech-Eye SDK 2.0.0, Mech-Eye Industrial 3D Camera is compliant with the machine vision industrial standard *GenICam*.

Compliance with the GenICam standard provides the following values:

- The camera can be controlled by third-party machine vision software that are compliant with the GenICam/GigE Vision 2, such as HALCON (by MVTec) and VisionPro (by Cognex). Currently, HALCON has been tested.
- User experience of controlling the camera with a third-party software is improved in aspects including network configuration, parameter adjustment, image capturing and data transmission.
- By upgrading the camera firmware to 2.0.0, you can connect to the camera with a third-party software directly.

---

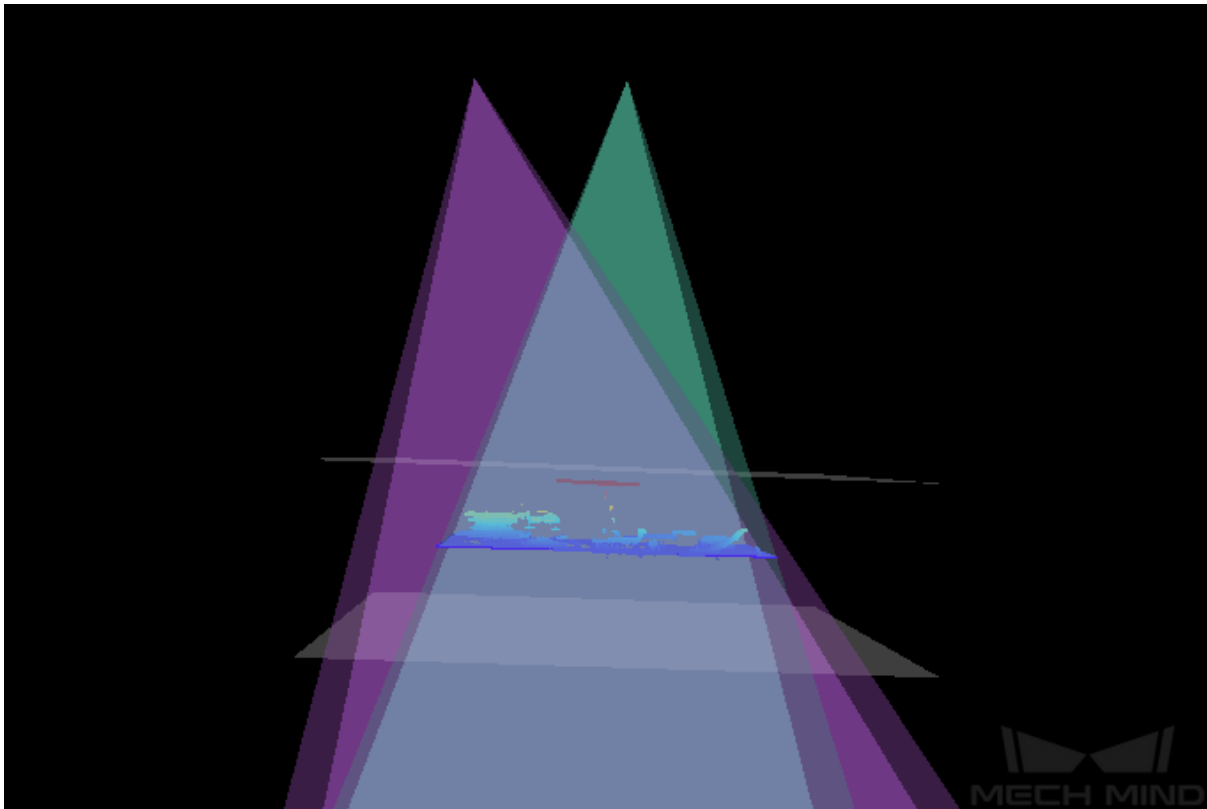
**Hint:** It is recommended to use Mech-Vision 1.6.2 and above with camera firmware 2.0.0. If you use Mech-Vision 1.6.1 or below with camera firmware 2.0.0, connecting to cameras in **External2D EBUSCamera** may cause Mech-Vision to crash.

---

#### Mech-Eye Viewer

- Added Tool for Setting Depth Range

A tool for *setting depth range* is added in Mech-Eye Viewer 2.0.0. With this tool, you can see the upper and lower limits of the depth range relative to the obtained point cloud while adjusting.



- Added Full Pallet Simulator Tool

The *Full Pallet Simulator* tool is added in Mech-Eye Viewer 2.0.0. With this tool, you can generate a cuboid to simulate a full load on a pallet, and check if the top layer is entirely located within the camera's field of view. This tool is intended for checking if the camera mounting height is sufficient before project deployment, and provide an estimate amount of adjustment if insufficient.

- Added Point Cloud Exhibit Feature

The *Point Cloud Exhibit* feature is added in Mech-Eye Viewer 2.0.0. When enabled, this feature can move the point cloud along a predetermined trajectory to display the point cloud from various angles.

To enable: check **Point Cloud Exhibit** in the **View** menu, and then click the *Play* button in the upper right of the point cloud display area.

- Added Image Information Feature

The Image Information feature is added in Mech-Eye Viewer 2.0.0. When moving the cursor onto the 2D image or depth map, the information of the pixel where the cursor is located is displayed in the lower right.

To disable this feature, uncheck **Image Information** in the **View** menu.

- Added Camera Name Customization Feature

A *camera name customization* feature is added in Mech-Eye Viewer 2.0.0. After connecting to a real camera, you can set a custom camera name that is easy to recognize.

- Added 2D Camera Binning Feature (**Administrator** mode)

The 2D camera binning feature is added for the LSR series. When enabled, this feature reduces the resolution of the 2D image obtained by the color 2D camera from 12 MP to 3 MP.

To enable: in **Administrator** mode, go to *Tools* → *View 2D Camera and Set Parameters* and select the color 2D camera. Then, select the **EnableBinning** checkbox in the bottom and click *Apply Config*.

- Added Link to Troubleshooting Documentation

A ? button is added to the lower right of the Camera List interface in Mech-Eye Viewer 2.0.0. Clicking this button directs you to the troubleshooting guidance in the user manual, where you can find solutions to *common camera connection problems*.

## Mech-Eye API

The following parameters are added to Mech-Eye API 2.0.0:

- ProjectorPowerLevel: available for DLP cameras, excluding Deep (V3) and Pro L Enhanced (V3).
- FringeCodingMode: available for PRO S (V4), PRO M (V4), Nano (V3) and Pro XS (V3).
- AntiFlickerMode: available for PRO S (V4), PRO M (V4), NANO (V4) and Nano (V3).

The corresponding parameters and methods for C++, C# and Python are listed below.

- C++

Parameter	Method
ProjectorSettings:: PowerLevel::powerLevel	ErrorStatus MechEyeDevice::getProjectorPowerLevel (ProjectorSettings::PowerLevel &)
	ErrorStatus MechEyeDevice::setProjectorPowerLevel (ProjectorSettings::PowerLevel)
ProjectorSettings:: FringeCodingMode::fringeCodingMode	ErrorStatus MechEyeDevice::getProjectorFringeCodingMode (ProjectorSettings::FringeCodingMode &)
	ErrorStatus MechEyeDevice::setProjectorFringeCodingMode (ProjectorSettings::FringeCodingMode)
ProjectorSettings:: AntiFlickerMode::antiFlickerMode	ErrorStatus MechEyeDevice::getProjectorAntiflickerMode (ProjectorSettings:AntiFlickerMode &)
	ErrorStatus MechEyeDevice::setProjectorAntiflickerMode (ProjectorSettings:AntiFlickerMode)

- C#

Parameter	Method
ProjectorSettings.projectorPowerLevel Type: ProjectorSettings::PowerLevel	ErrorStatus MechEyeDevice.GetProjectorPowerLevel (mmind.apiSharp.PowerLevel ref)
	ErrorStatus MechEyeDevice.SetProjectorPowerLevel (mmind.apiSharp.PowerLevel)
ProjectorSettings.projectorFringeCodingMode Type: ProjectorSettings::FringeCodingMode	ErrorStatus MechEyeDevice.GetProjectorFringeCodingMode (mmind.apiSharp.FringeCodingMode ref)
	ErrorStatus MechEyeDevice.SetProjectorFringeCodingMode (mmind.apiSharp.FringeCodingMode)
ProjectorSettings.projectorAntiFlickerMode Type: ProjectorSettings::AntiFlickerMode	ErrorStatus MechEyeDevice.GetProjectorAntiFlickerMode (mmind.apiSharp.AntiFlickerMode ref)
	ErrorStatus MechEyeDevice.SetProjectorAntiFlickerMode (mmind.apiSharp.AntiFlickerMode)

- Python (new methods only)

Device.get_projector_power_level(self)
Device.set_projector_power_level(self, value)
Device.get_projector_fringe_coding_mode(self)
Device.set_projector_fringe_coding_mode(self, value)
Device.get_projector_anti_flicker_mode(self)
Device.set_projector_anti_flicker_mode(self, value)

## 2.3.2 Feature Improvements

### Mech-Eye Viewer

- Improved the tool for setting an ROI.
  - Handles are added to the ROI selection box, allowing for size adjustment and movement of the box.
  - Scrolling to zoom the 2D image is added.
  - Limits of the max and min zoom size of the 2D image are added.
- Improved the *display depth range adjustment feature* of the depth map/point cloud display.
  - The colorbar and slider are combined for more intuitive range adjustment.
  - Fields for entering the upper and lower limits of the display depth range are added.
  - Unit change feature is added: Units include mm and  $\mu\text{m}$ . For the UHP series,  $\mu\text{m}$  is recommended. For other series, mm is recommended.
- Improved the parameter names and descriptions. For a list of changed parameter names, please refer to [Appendix](#).

- The **Anti-Flicker Mode** parameter is now available for the Nano (V3) and NANO (V4) series.
- DLP Projector temperature can now be read and displayed for the PRO (V4) series (in the **Camera Controller** tool).
- Updated the minimum value of **3D Parameters - Exposure Time** to 8 ms for the LSR series.
- Added DEEP (V4) to *FOV Calculator*.
- Changed the visibility level of **Sharpen Factor** to **Guru**.
- Improved release notes: By selecting *Help* → *What's New*, you can now open the release notes in the user manual, which provides detailed information on the changes introduced in the current version.

### Mech-Eye API

- Improved the error messages.
- Improved the *names of C++ parameters in MechEyeSettings*.
- Improved the *names of C# parameters and methods*.

## 2.3.3 Resolved Issues

### Mech-Eye Viewer

- Resolved the abnormality of the textured point cloud obtained by the 12 MP variants of Pro L Enhanced and Laser L Enhanced (V3 and V3S) when binning had been enabled.
- Resolved the issue where firmware 1.4.0 could not be upgraded to the latest version.
- Resolved the issue where both the new and old camera IP addresses showed up in Mech-Vision after the camera IP address had been changed in Mech-Eye Viewer.
- Resolved the occasional failure of depth data acquisition with Nano and Pro XS (V3).

### Mech-Eye API

Resolved the issue where the Connect method could not be called repetitively.

## 2.3.4 Appendix

The following table lists the names of **Beginner** level parameters and parameter categories that are updated in Mech-Eye SDK 2.0.0.

Before 2.0.0		2.0.0	
3D Scanning		3D Parameters	
	Scan Multiplier		Exposure Multiplier
	Exposure Time		-
	UHP		-
	Capture Mode		-
2D Scanning		2D Parameters	
	Exposure Mode		-
	Exposure Time		-
	Scan2D Expected Gray Value		Gray Value
	2D Auto-Exposure ROI		Auto-Exposure ROI
	Geometry		Values
	Scan2D Tone Mapping Enable		Tone Mapping
	HDR Exposure Sequence		Exposure Sequence
	Exposure Sequence		Values
	Texture Role Camera Exposure Mode		Color Camera Exposure Mode
	Pattern Role Camera Exposure Mode		Monochrome Camera Exposure Mode
	Pattern Role Camera Exposure Time		Monochrome Camera Exposure Time
Point Cloud Processing		-	
	Cloud Smooth Mode		Point Cloud Smoothing
	Cloud Outlier Filter Mode		Outlier Removal
	Fringe Contrast Threshold		Stripe Contrast Threshold
Depth Range		-	
	Depth Upper Limit		-
	Depth Lower Limit		-
3D ROI		ROI	
	Geometry		Values

**Note:**

- “-” indicates that the parameter name remains unchanged. Only the parameter description is updated.
- Some parameters are only available for certain models.

## MECH-EYE SDK INSTALLATION GUIDE

This chapter guides you through the downloading, installation, upgrade, uninstallation, repair, and installation modification of Mech-Eye SDK on Windows.

---

**Note:** Mech-Eye SDK includes Mech-Eye API and Mech-Eye Viewer.

---

### 3.1 Download Mech-Eye SDK Installer

You can download the Mech-Eye SDK installer from [Mech-Mind's website](#) or obtain it from Mech-Mind pre-sales or support engineers.

---

**Hint:**

- When downloading, you will be prompted to log in before you can proceed. If you don't have an account yet, please sign up using an email address first.
- 

### 3.2 Verify the Integrity of Mech-Eye SDK Installer

Since the Mech-Eye SDK installer may be corrupted during transmission or downloading, you need to verify the integrity of the installer. To facilitate integrity verification, a CRC32 value is provided in the **Mech-Eye SDK Installation Guide** PDF file, which you can obtain by:

- Extracting the compressed file downloaded from Mech-Mind's website.
- Contacting Mech-Mind pre-sales or support engineers.

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

1. Copy the obtained Mech-Eye SDK installer to a specified directory on the IPC or PC, such as *D:\*.
2. Right-click the installer and select *CRC SHA* → *CRC32* to calculate the CRC32 value for the installer.
3. Check the calculated CRC32 value against the one in the **Mech-Eye SDK Installation Guide** PDF file. The two values should be identical.



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.

### 3.3 Install Mech-Eye SDK

1. Double-click the installer to open the 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*.

---

**Note:**

- Make sure that the **Add to PATH** option is checked.
- If checked, the **Create desktop shortcut(s)** option adds a shortcut of Mech-Eye Viewer to your desktop.
- It is recommended to check the **Mech-Eye SDK Docs** option, which allows the Mech-Eye Viewer to open the user manual without Internet connection.

- 
5. In the **Set Path** window, change the installation path as required, such as *D:\project*, and then click *Next*.
  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, click *Finish* to exit the Setup Wizard.

---

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

---

### 3.4 Upgrade Mech-Eye SDK

---

**Note:**

- Only Setup Wizard for version 2.0.0 supports this function.

- The Setup Wizard does not support direct upgrade from a version lower than 1.6.1 to 2.0.0 or above. If you want to upgrade Mech-Eye SDK of a lower version to 2.0.0 or above, uninstall the lower version and then *install the new version*.
- 

If Mech-Eye SDK has already been installed on the computer, you can use the Setup Wizard to upgrade it.

1. Double-click the installer to open the Setup Wizard.
2. In the **Upgrade** window, click either *Upgrade and keep history versions* or *Upgrade and delete history versions*.
3. Wait for the upgrade to complete.

## 3.5 Uninstall Mech-Eye SDK

You can uninstall Mech-Eye SDK either with the Setup Wizard or in Windows Control Panel.

### 3.5.1 Uninstall Mech-Eye SDK Using the Setup Wizard

1. Double-click the installer to open the Setup Wizard.
2. In the **Maintain** window, click the *Uninstall* button.
3. In the **Uninstall** window, click either *Keep user configuration files* or *Do not keep user configuration files*.
4. Wait for the uninstallation to complete.

### 3.5.2 Uninstall Mech-Eye SDK Using the Control Panel

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**.
4. Wait for the uninstallation to complete.

## 3.6 Repair Mech-Eye SDK

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

1. Double-click the installer to open the Setup Wizard.
2. In the **Maintain** window, click *Repair*.
3. Wait for the repair to complete.

---

## 3.7 Modify Installed Components

---

**Note:** Only Setup Wizard for version 2.0.0 supports this function.

---

After installation, you can use the Setup Wizard to change the installed components of Mech-Eye SDK.

1. Double-click the installer to open the Setup Wizard.
2. In the **Maintain** window, click *Modify*.
3. In the **Select Product** window, check the components to install, or uncheck the components to uninstall.
4. Complete remaining installation process.

## 3.8 Software License Agreement

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

## 3.9 Troubleshoot Common Installation Problems

### 3.9.1 Failed to Start the Installation Wizard

#### Symptom

After double-clicking 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 installation package.

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

### 3.9.2 Installation Failed

#### Symptom

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

#### Possible Causes

- Corrupted package 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

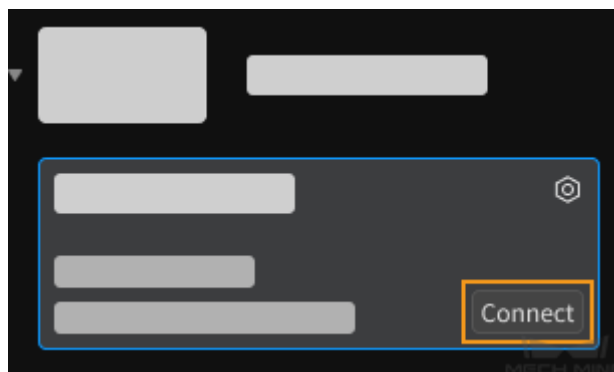
1. Obtain a new installation package and *check its integrity*, and then try the installation again. If the problem still exists, proceed to step 2.
2. Right-click the installation file and select “Run as administrator” to reinstall. If the problem still exists, proceed to step 3.
3. Try the installation again after the other Wizard finishes installation or after the Windows system upgrade completes. If the problem still exists, proceed to step 4.
4. Restart your computer and try the installation again. If the problem still exists, click the **installation log** link to collect installation logs and send them to Mech-Mind Technical Support.

## MECH-EYE VIEWER QUICK START GUIDE

This topic provides a quick start guide to get you familiar with the basic functions in Mech-Eye Viewer.

### 4.1 Connect to the Camera

Open Mech-Eye Viewer, select the camera you'd like to connect to in the **Camera List**, and then click *Connect*.



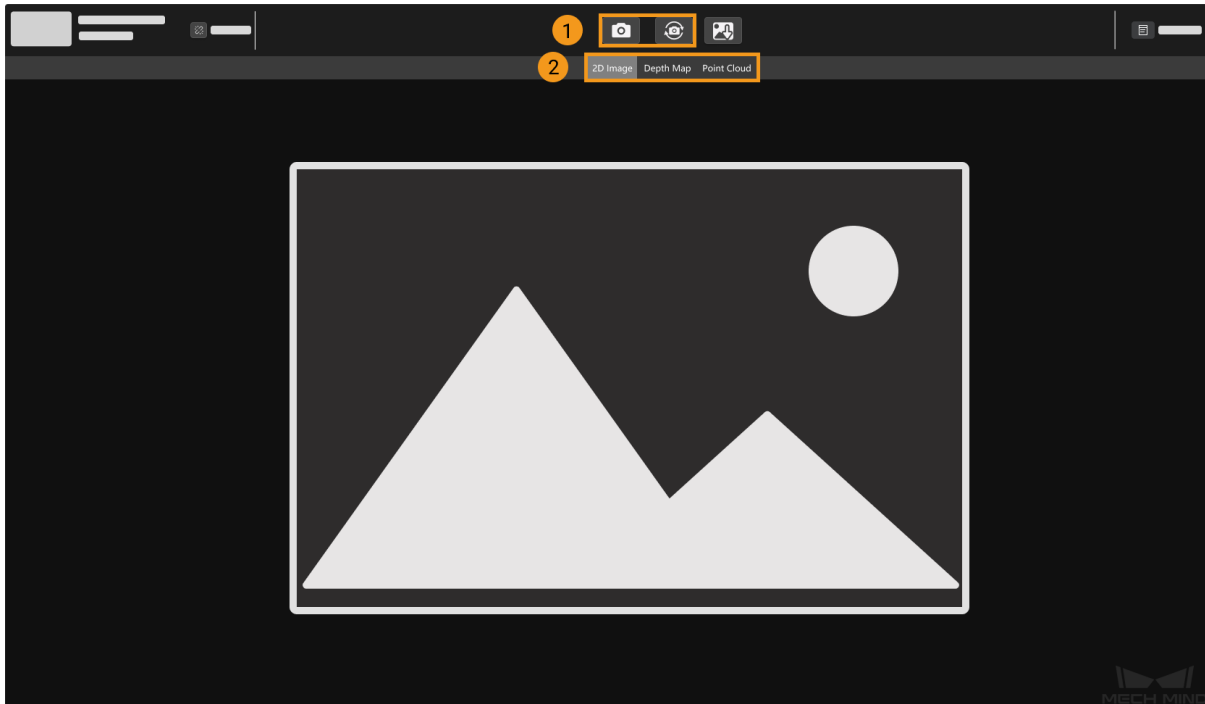
---



**Note:** Refer to [Connect to Camera and Set IP Address](#) for instructions on setting camera IP address.

---

### 4.2 Capture Images

If the camera is successfully connected, Mech-Eye Viewer displays the *following interface*.




1. Click  or  to perform image capturing.
2. Switch the displayed data type by clicking **2D Image**, **Depth Map**, and **Point Cloud**.

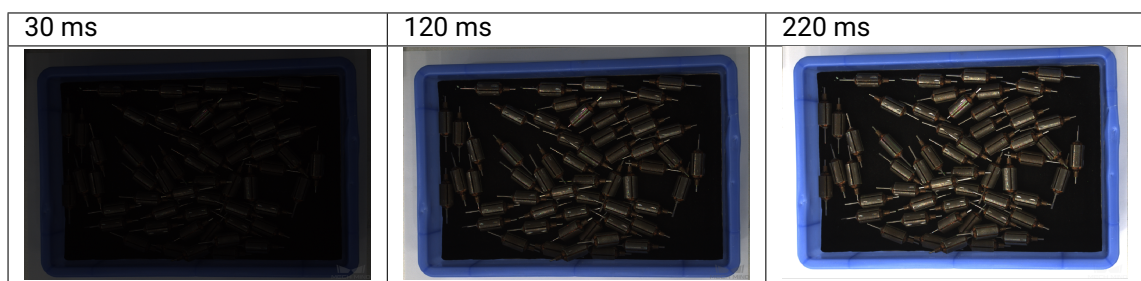
## 4.3 Adjust Parameters

You can adjust parameters in the *parameter panel* to the right to improve data quality.

### 4.3.1 Adjust Parameters for 2D Image

Adjust the **Exposure Mode** and relevant parameters of *2D Parameters* in the parameter panel. You can use  to see the effect in almost real time.

- 2D images captured with different **Exposure Time** values in **Timed** exposure mode:



---

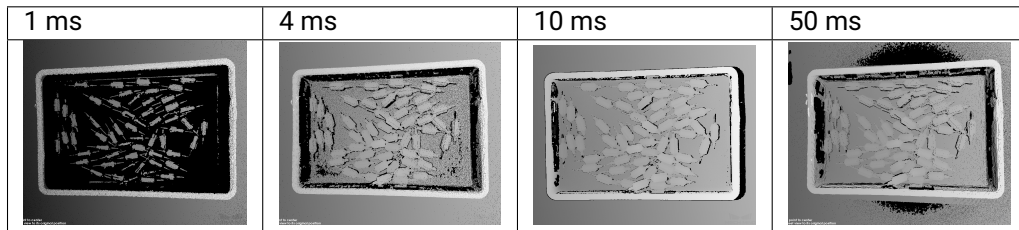
**Hint:** The exposure time affects the brightness of the image. The longer the exposure time, the brighter the image.

---

### 4.3.2 Adjust Parameters for Depth Map and Point Cloud

Adjust the **Exposure Multiplier** and **Exposure Time** of *3D Parameters* in the parameter panel. You can use *Exposure Assistant* to obtain the recommended values for these parameters. Please refer to *Basic Parameters for Depth Map and Point Cloud* for detailed instructions.


- Point clouds generated with different **Exposure Time** values:



### 4.3.3 Adjust Point Cloud Processing Settings

Adjust **Point Cloud Smoothing**, **Outlier Removal**, and **Stripe Contrast Threshold** of **Point Cloud Processing** in the parameter panel. Please refer to *Point Cloud Processing* for detailed explanations.

## 4.4 Save Data

Click  in the camera toolbar to save the last obtained 2D image, depth map, and/or point cloud.

If you'd like to save the raw data in MRAW format, go to **File** in the menu bar and click **Save Raw Data of Camera**. You can use **Start Virtual Camera** in the **File** menu to open MRAW data.

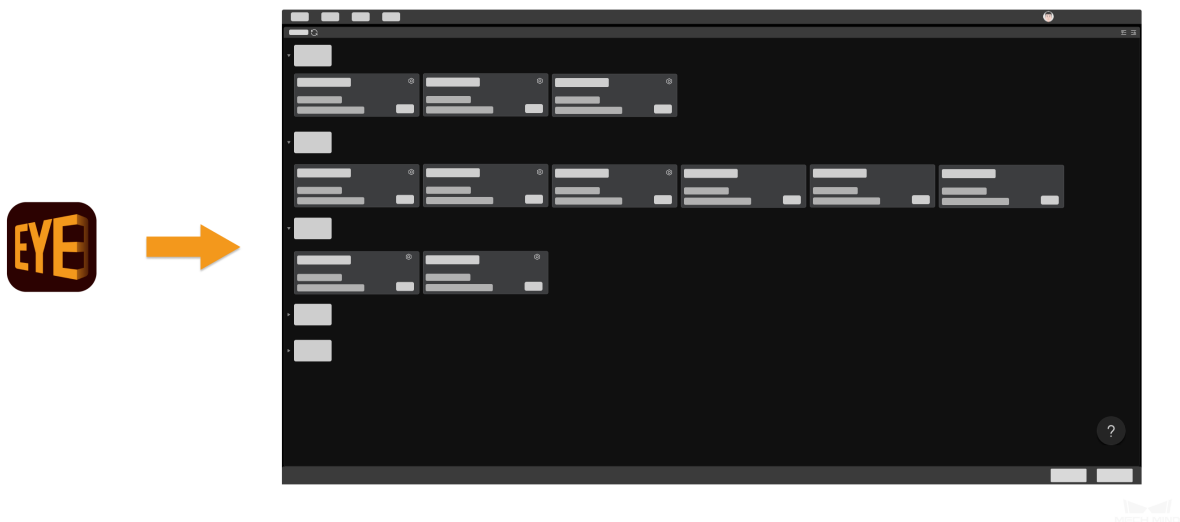
---

For detailed information on Mech-Eye Viewer, please refer to *the Mech-Eye Viewer User Manual*.

## MECH-EYE VIEWER QUICK FACTS

Mech-Eye Viewer is a camera configuration and data visualization software independently developed by Mech-Mind Robotics.

In Mech-Eye Viewer, based on the characteristics of the target objects, you can adjust the parameters of Mech-Eye Industrial 3D Cameras to conveniently obtain high-quality 2D images, depth maps, and point clouds.



---

Instructions on installing Mech-Eye Viewer:

[\*Mech-Eye SDK Installation Guide\*](#)

---

Introduction of the user interface:

[\*User Interface of Mech-Eye Viewer\*](#)

---

Instructions on obtaining, adjusting and saving data with Mech-Eye Viewer:

[\*Using Mech-Eye Viewer\*](#)

---



Instructions on built-in tools in Mech-Eye Viewer:

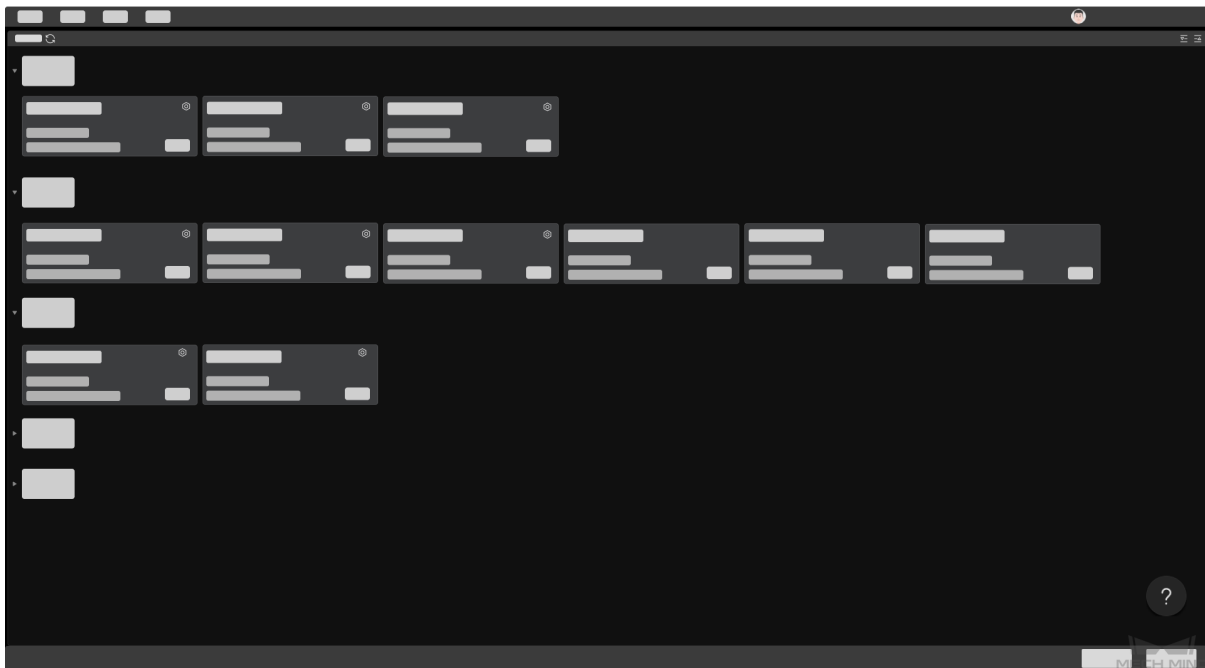
*Built-In Tools*

## USER INTERFACE OF MECH-EYE VIEWER

This section introduces the user interface of Mech-Eye Viewer.

### 6.1 Camera List

When you open Mech-Eye Viewer, the **Camera List** is displayed, as shown below.



In this interface, you can view all the available cameras and their information, *set camera IP address and connect to a camera*.

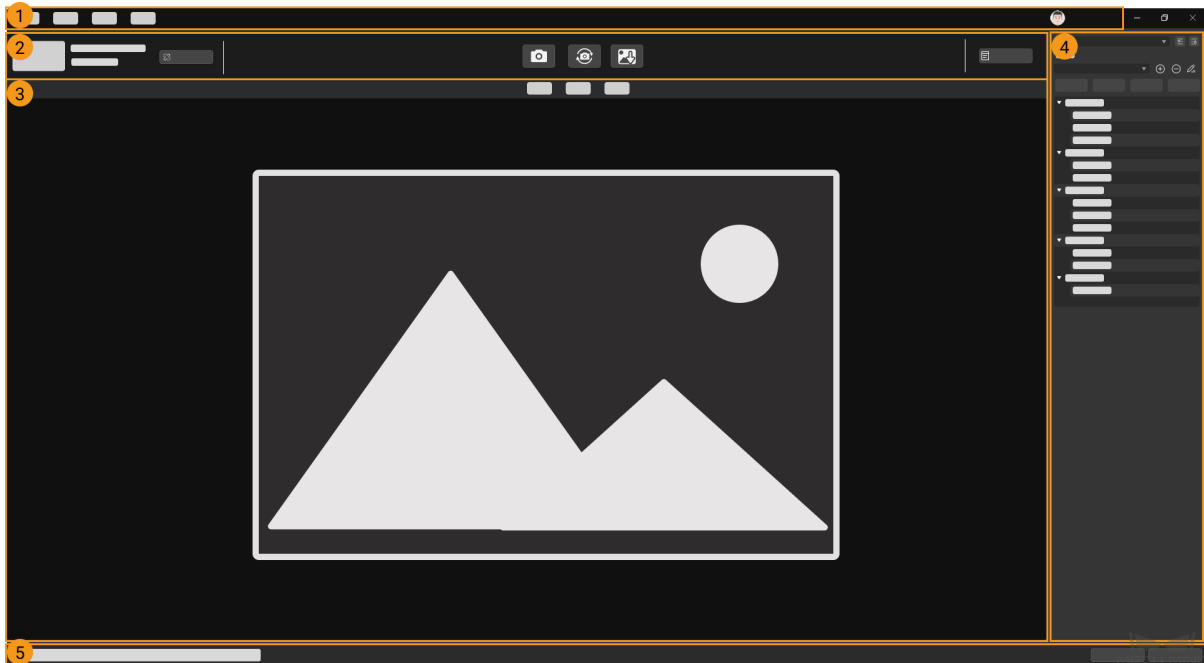
If you encountered any problems connecting to the camera, click the ? button in the lower right to view *troubleshooting guidance*.

**Note:** Once a camera is connected, you can't return to this interface unless you disconnect from the camera. Therefore, if you need to change the camera IP address, please do so before connecting to the

camera.

## 6.2 Camera Viewer

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




This interface consists of the following parts:

1. *Menu bar*: save raw data, start virtual camera, open built-in tools, check software information, etc.
2. *Camera toolbar*: disconnect the camera, obtain and save images, and view the camera log.
3. *Data display area*: display obtained 2D image, depth map, and point cloud.
4. *Parameter panel*: view and adjust parameters, and save parameter groups for different purposes.
5. *Status bar*: display statuses and error messages. Use *Previous* and *Next* to switch between entries.

## 6.3 Menu Bar

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

Click to display a table summarizing the menu options.

Menu	Option	Description
File	Save Raw Data of Camera	Save the original data in MRAW format for debugging and analysis.
	Start Virtual Camera	Load saved data and adjust the depth range and point cloud processing parameters of these data.
Tools	Provide supplementary tools. For detailed information, please refer to <i>Built-In Tools</i> .	
View	Toolbar	Check to display the <i>toolbar</i> . Unchecked by default.
	Image Information	Check to display the image information box on the 2D image and depth map. Checked by default.
	Point Cloud Exhibit	Check to display the <i>Play</i> button on the point cloud. Unchecked by default.
Help	About	View software version.
	What's New	View the changes made to the current version.
	Manual	View the User Manual.
	Options	Switch the software display language; change is applied after restarting the software.
 Standard ▾	Switch user account. The default account is <b>Standard</b> ; to use other user accounts, contact Technical Support for passwords.	

## 6.4 Camera Toolbar

In the camera tool bar, you can:

- *Set a custom camera name*
- *Perform image capturing*
- *Save obtained data to local*
- *View the camera's log*
- Disconnect from the camera

## 6.5 Data Display Area

This area displays the last obtained data. 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*.

## 6.6 Parameter Panel

You can view all the available parameters, adjust the visibility level, use parameter groups and view parameter information here.

For detailed information, please refer to [Parameters](#).

---

**Note:** It is recommended to maximize the interface of Mech-Eye Viewer for better display of components in the parameter panel.

---

The next topic provides instructions on using Mech-Eye Viewer.

## USING MECH-EYE VIEWER

This section includes instructions on using Mech-Eye Viewer, such as connecting to a camera, obtaining data, adjusting parameters, and saving the data.

---

Instructions on **connecting to a camera and setting camera IP address**:

*[Connect to Camera and Set IP Address](#)*

---

Information on **obtaining data and data types**:

*[Capture Images and Switch Data Type](#)*

---

Explanations on **parameters for image capturing and data processing**:

*[Parameters](#)*

---

Instructions on **saving obtained data**:

*[Save Data](#)*

---

Information on the **camera log**:

*[Log](#)*

### 7.1 Connect to Camera and Set IP Address

The camera must be in the same subnet as the IPC for the camera to be connected in Mech-Eye Viewer.

By default, the IP address of a camera is automatically assigned via the Dynamic Host Configuration Protocol (DHCP). If this meets your needs, you can *[connect to the camera](#)* directly.

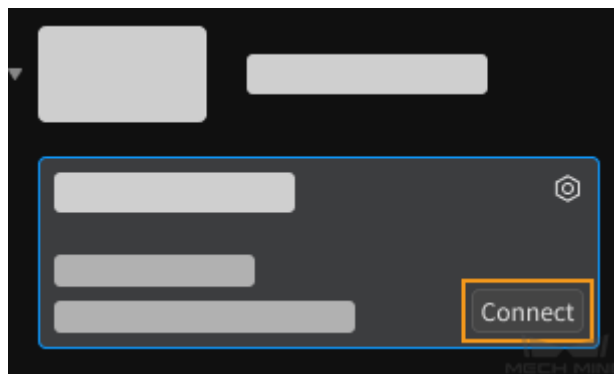
If you need to set a static IP for the camera, please check the instructions in *[Set IP Address of Camera](#)* below.

- *Connect to a Camera*
- *Set a Custom Camera Name*
- *Set IP Address of Camera*
- *Set IP Address on IPC*

### 7.1.1 Connect to a Camera

All the detected cameras are shown in the **Camera List** when you open Mech-Eye Viewer.

Select a camera and click *Connect* to connect to the camera.



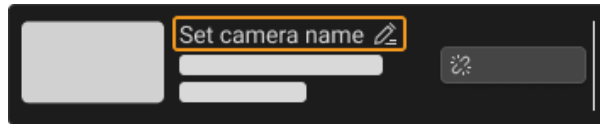
**Attention:**

- For the camera to be connected in Mech-Eye Viewer, the camera firmware's version must be consistent with Mech-Eye Viewer's version. If not consistent, you will be prompted to upgrade the firmware with the *Camera Firmware Upgrade* tool.
- For the very first time that the camera is connected to the IPC, it may take up to a few minutes for the camera to show up in Mech-Eye Viewer.
- If **Connected** is displayed in the camera info card, the camera is already connected to a GenI-Cam client. To connect the camera to Mech-Eye Viewer, please disconnect the camera from the GenI-Cam client first.

### 7.1.2 Set a Custom Camera Name

After the camera is connected to Mech-Eye Viewer, you can set a custom camera name that is easy to recognize.


In the camera toolbar, click  or **Set camera name** to enter a name for the camera.

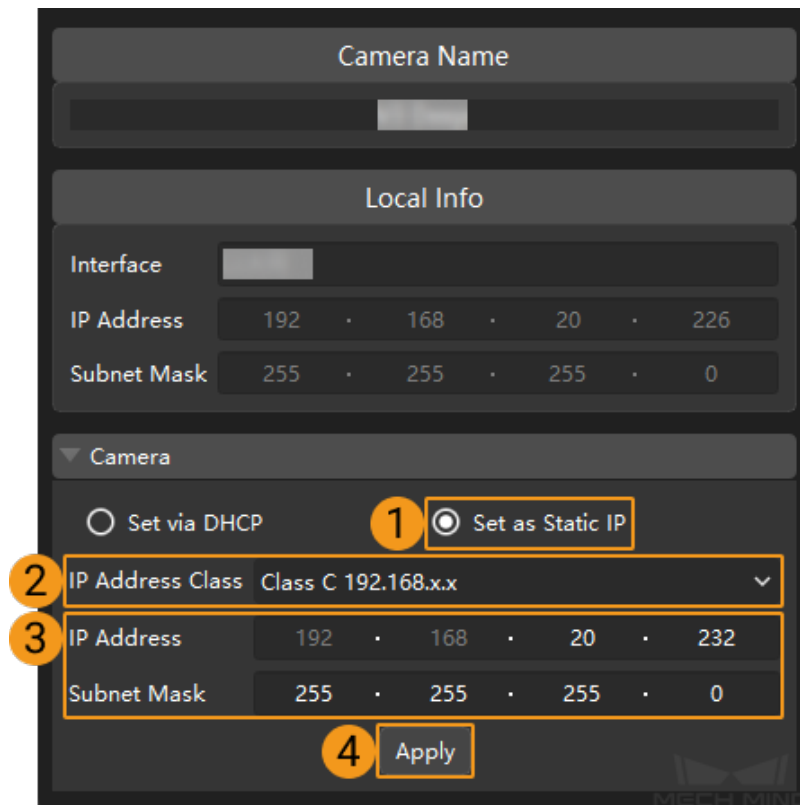


**Note:**

- The following characters cannot be used: \, /, :, \*, ?, ", <, >, |.
- This function is unavailable for a virtual camera.

### 7.1.3 Set IP Address of Camera

1. Open Mech-Eye Viewer and select the camera for which you'd like to set the IP address. Click  to open the IP configuration window.
2. Select **Set as Static IP**, and select the **IP Address Class** according to your actual situation. Then, set the IP address and subnet mask for the camera and click *Apply*.



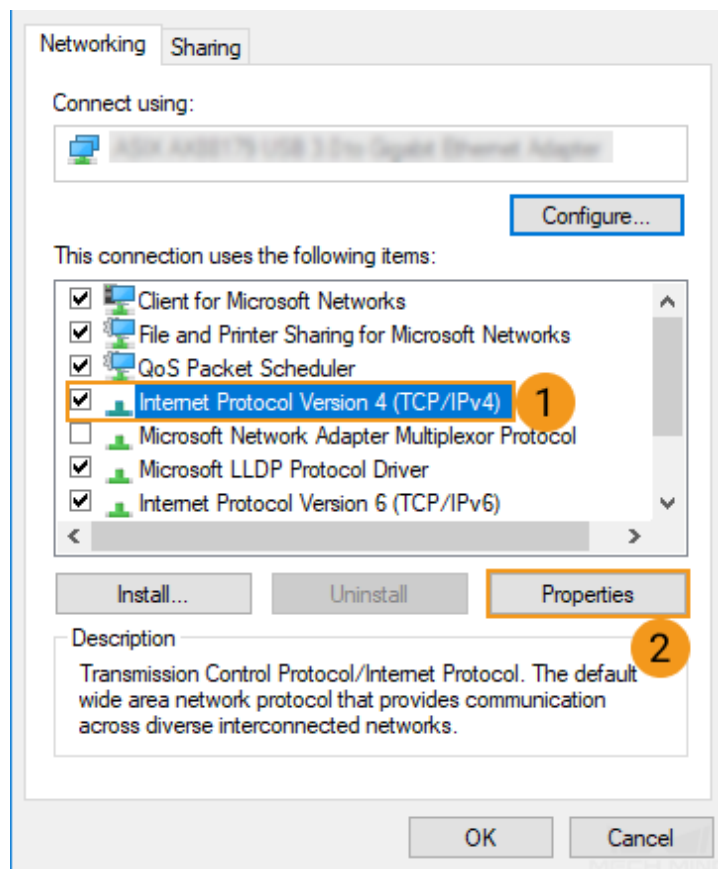
**Hint:**



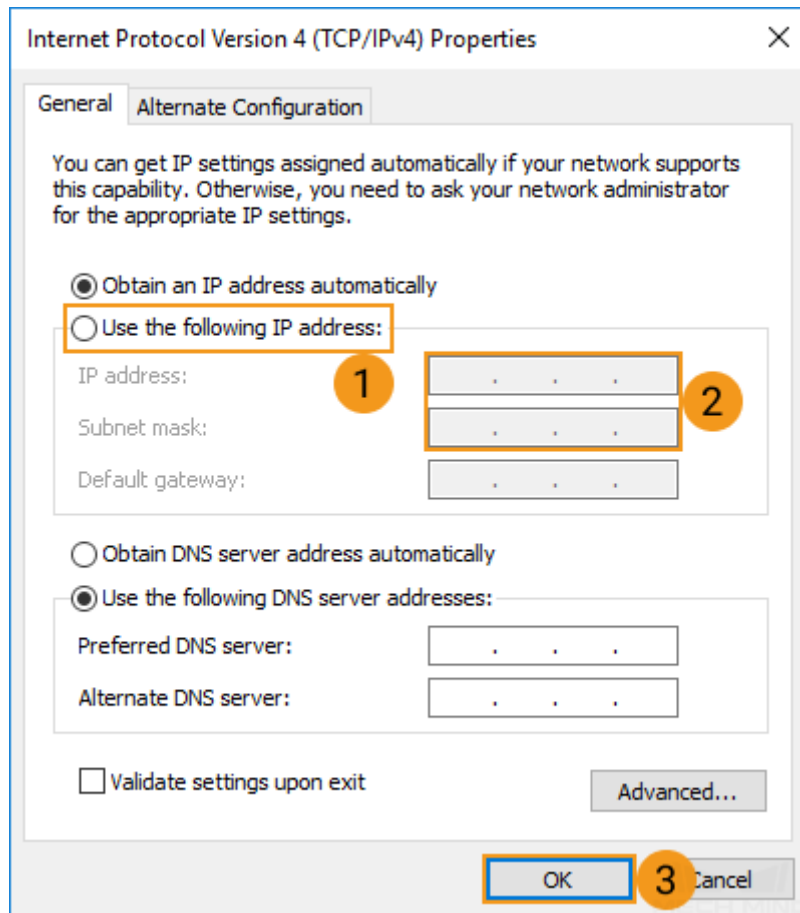
- Record the set camera IP address for future reference.
- The **Local Info** section displays the IP address and subnet mask of the IPC network interface card connected to the camera. You can set the IP address and subnet mask of the camera according to the information displayed here.
- To switch back to automatically assigned IP address, select **Set via DHCP**.

### 7.1.4 Set IP Address on IPC

1. Go to *Control Panel* → *Network and Internet* → *Network and Sharing Center* → *Change adapter settings*.
2. Right-click the network interface card connected to the camera, and select **Properties**.
3. In the pop-up window, click **Internet protocol version 4 (TCP/IPv4)**, and select **Properties**.



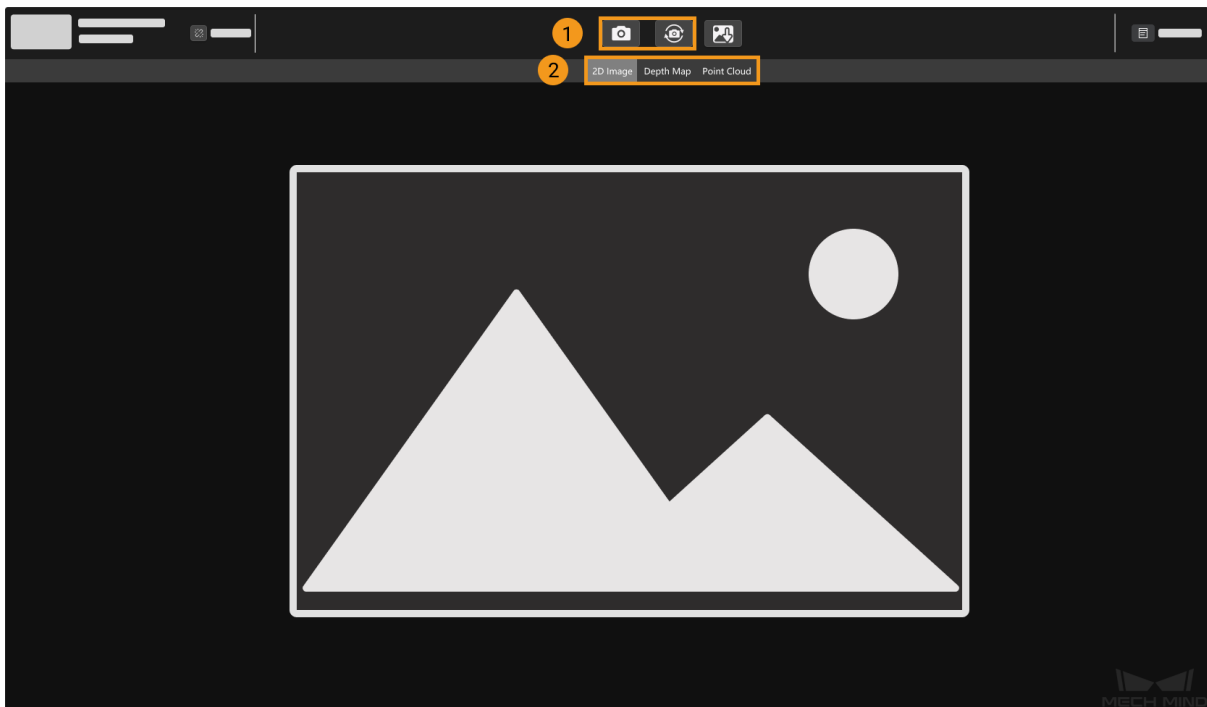
4. Select **Use the following IP address**, and set the IP address and subnet mask according to your actual needs. Then, click **OK** to save the changes.



Once you are connected to a camera in Mech-Eye Viewer, you can start capture images.  
The next topic introduces the actions for capturing images and viewing the obtained data.

## 7.2 Capture Images and Switch Data Type

Once the camera is successfully connected, the following interface is displayed. In this interface, you can perform image capturing and view the obtained 2D image, depth map, or point cloud.





1. Perform image capturing.
2. Switch the displayed data type.

The following sections explain the actions in detail.

### 7.2.1 Capture Images

There are two ways to perform image capturing:

- : the camera performs one round of image capturing and output the data to Mech-Eye Viewer.
- : the camera continuously performs image capturing at predetermined intervals and output the data to Mech-Eye Viewer.

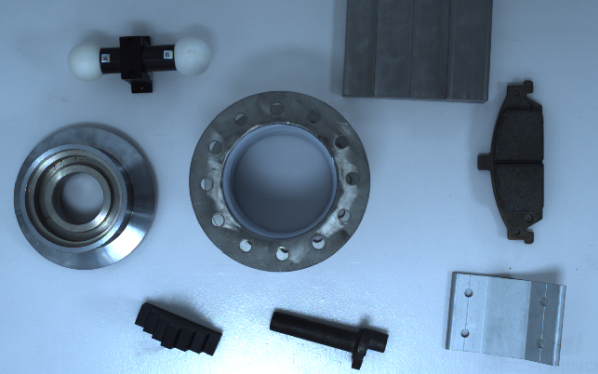
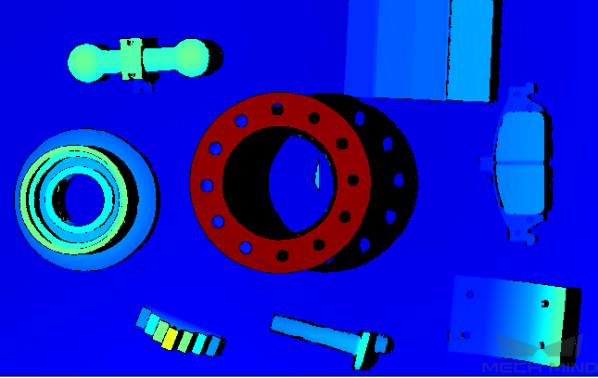
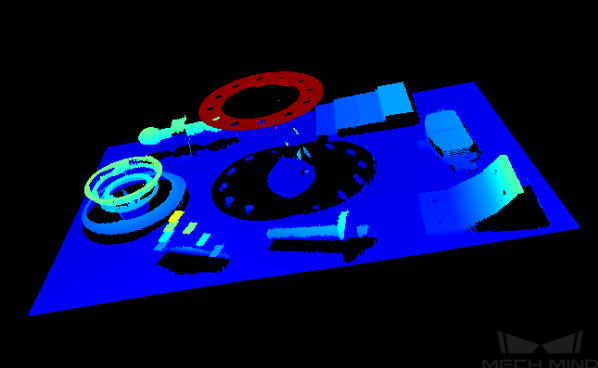
---

#### Hint:

- For an explanation of the image capturing and data output processes, please refer to [Structured-Light 3D Scanning Mechanism](#).
  - For binocular cameras, two 2D images are output with each round of image capturing. Binocular models include Laser L, Laser L Enhanced, Deep and Pro XS.
-

### 7.2.2 Switch Data Type

Three types of data are output by the camera to Mech-Eye Viewer: 2D image, depth map and point cloud. Click the corresponding button to view the data.

Data Type	Description and Example
2D Image	Black-and-white/color image (depending on the camera model). 
Depth Map	An image containing depth information. 
Point Cloud	A collection of data points that represent the 3D shape of the object surface. 

In the data display area, a number of actions are provided to aid you in examining the data. The next topic explains these actions in detail.

## 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 Mech-Eye Viewer.

Check out the full list of actions you can take for each data type:

- *Adjust 2D Image Display*
- *Adjust Depth Map Display*
- *Adjust Point Cloud Display*

## Adjust 2D Image Display

### Zoom

You can zoom the 2D image in or out by scrolling the mouse wheel while placing the cursor on the 2D image.

To restore the 2D image to the default size: right-click the 2D image and select **Reset View**.

### View Image Information

When you move the cursor on the 2D image, the information of the pixel where the cursor is located is displayed in the lower-right corner.

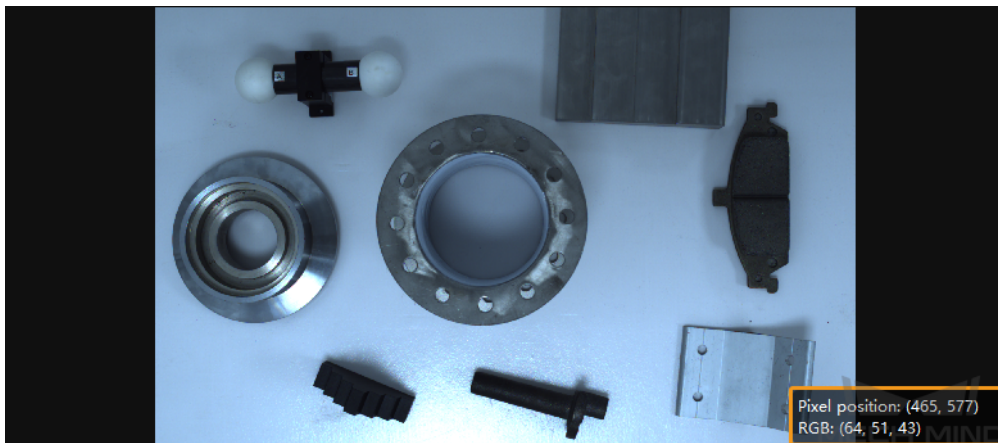


Image information displayed includes:

- Pixel position
- RGB values (for color 2D image) or grayscale value (for black-and-white 2D image)

You can turn off this display box by unchecking **Image Informaiton** in the **View** menu.

## Adjust Depth Map Display

### Zoom

You can zoom the depth map in or out by scrolling the mouse wheel while placing the cursor on the depth map.

To restore the depth map to the default size: right-click the 2D image and select **Reset View**.

### View Image Information

When you move the cursor on the depth map, the information of the pixel where the cursor is located is displayed in the lower-right corner.

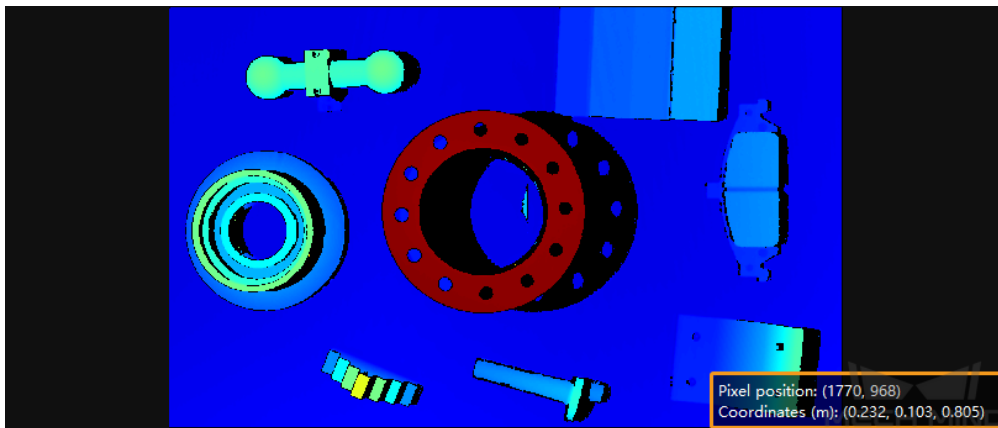


Image information displayed includes:

- Pixel position
- Coordinates in the camera reference frame (unit: m)

You can turn off this display box by unchecking **Image Information** in the **View** menu.

### Adjust Depth Data Display

In the left panel of the depth map display, you can change the reference frame, color rendering scheme and range of the depth data.

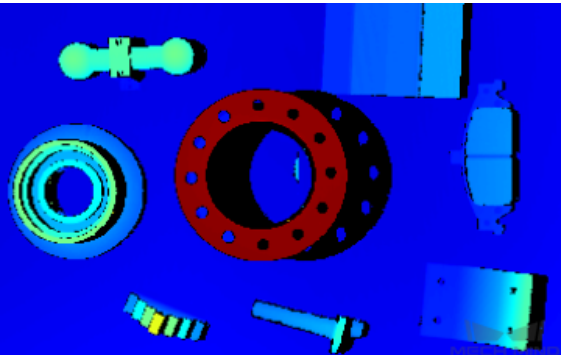
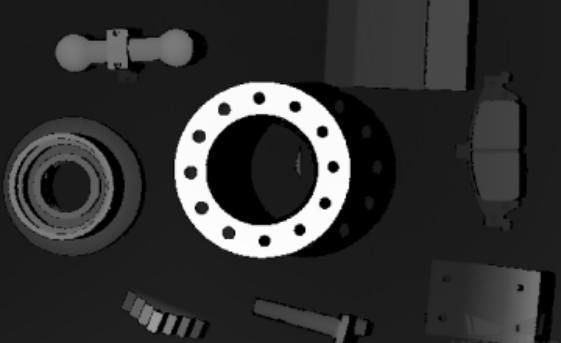
### Change Reference Frame

Select from the drop-down menu of **Coordinate System** to change the reference frame in which the depth data is displayed.

- **Camera:** Depth data is displayed in the camera reference frame whose origin is located inside the camera.
- **Customized:** Depth data is displayed in the custom reference frame defined with the *Custom Coordinate System* tool. This tool can be open in the **Tool** menu.

### Change Color Rendering Scheme

Select from the drop-down menu of **Color** to change how the depth map is colored. Options include **Jet** and **Gray Scale**.

Option	Description and Example
Jet	<p data-bbox="336 770 1390 804">Different depth values correspond to different colors in the Jet colormap.</p> 
Gray Scale	<p data-bbox="336 1151 1390 1184">Different depth values correspond to different grayscale values.</p> 

## Adjust Display Depth Range

You can adjust the display depth range in the **Range** section of the left panel.

---

**Note:** The settings here only affect the display. The depth data in the depth map is not trimmed.

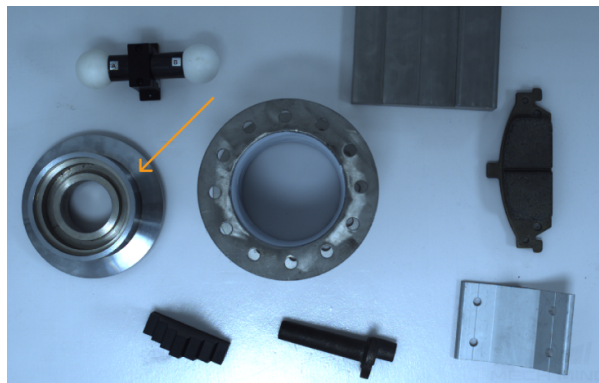
---

Each time that you perform image capturing, the display depth range is automatically adjusted to match the range of the obtained data.

If you want to check the depth fluctuation in a specific interval more closely, you can adjust the display depth range to this interval. The entire color scheme will be re-distributed to this interval, making it easier to check depth fluctuation by color variation.

Click to read an example use case.

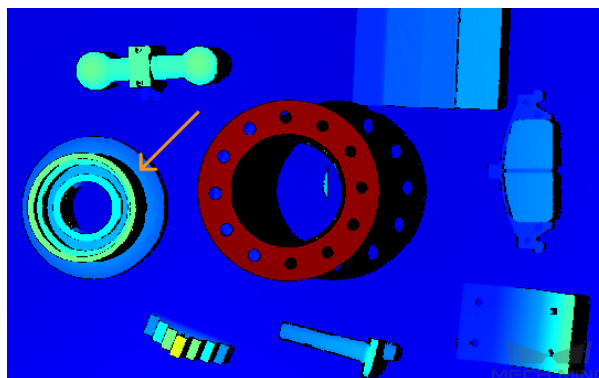
- Goal: check if the surface pointed by the orange arrow is flat enough.



- Actions:

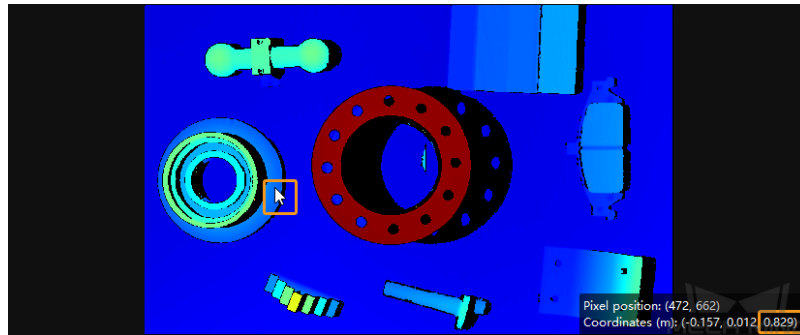
1. Perform image capturing with 

2. Switch to depth map display. You can see the surface appears to be roughly the same color. It is hard to tell if the surface is perfectly flat or not.

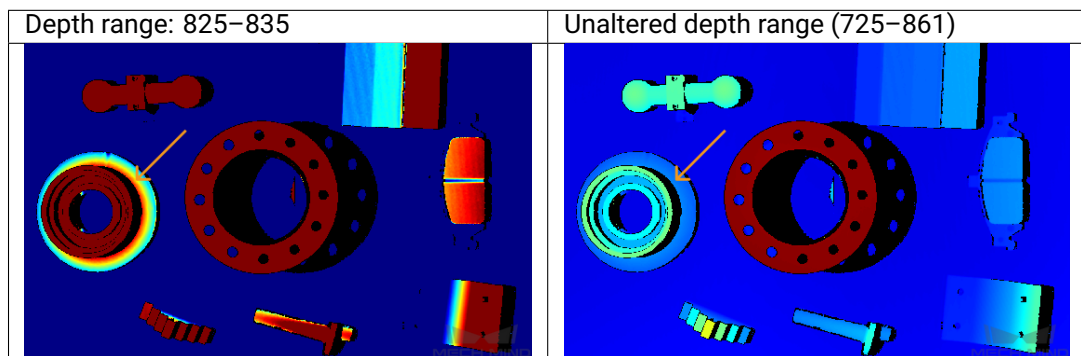


3. Move the cursor onto the surface to check the z coordinate. Here the z coordinate is **0.829** (unit: m).





4. Move the slider handles or change the values in the text boxes to adjust the range to 825–835 mm, a much narrower interval encompassing the above depth value.
5. Now the depth fluctuation of the surface is easy to see with the color variation.



**Note:** Depth values outside the set range are displayed as the extreme colors at the two ends.

- Adjust the display depth range roughly/quickly: Move the slider handles.
- Adjust the display depth range more precisely: Enter the values in the text boxes.

**Note:** If you are using a UHP series camera, you can change the unit of the depth data to  $\mu\text{m}$  from the drop-down menu.

- Lock the display depth range: Check the **Lock** option. When you perform image capturing again, the newly obtained depth map is displayed with the same depth range. This is useful if you want to compare the depth fluctuation in the same interval across different rounds of image capturing.

**Note:** All functions in the **Range** section are disabled when **Lock** is checked.

- Reset the display depth range: Click *Reset*. The range is reset to match the range of the obtained data.

**Hint:** The **Max** and **Min** values at the two ends of the slider correspond to the values set in the *Depth*

*Range* parameter.

---

## Adjust Point Cloud Display

### Zoom and Move

- Zoom the point cloud: scroll the mouse wheel while placing the cursor on the point cloud.
- Rotate the point cloud: hold the left mouse button and drag the point cloud.
- Pan the point cloud: hold the mouse wheel and drag the point cloud.
- Reset the view to default: press the **Backspace** key.
- Center the view: press the **R** key. The zoom is reset but rotation is not.

### Adjust Depth Data Display

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

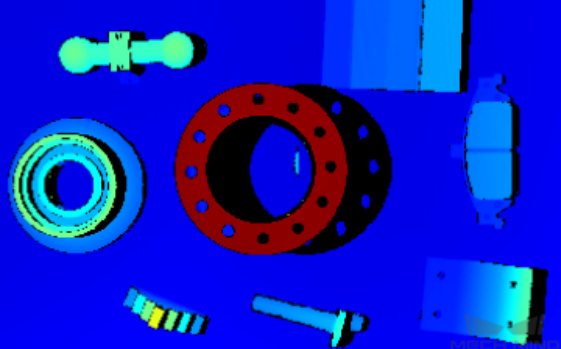
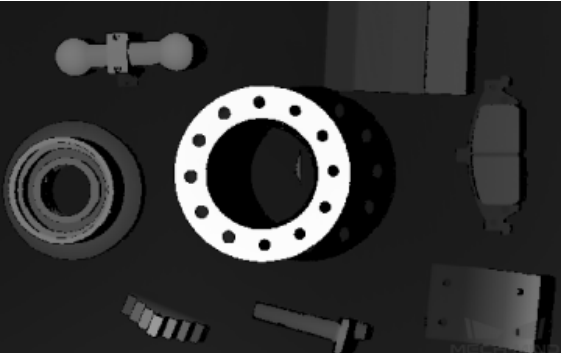
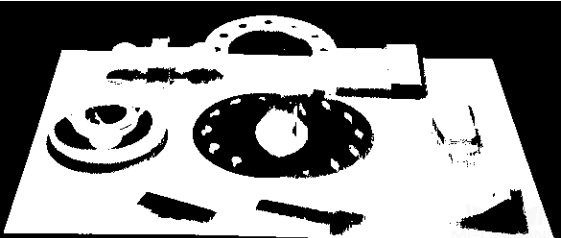
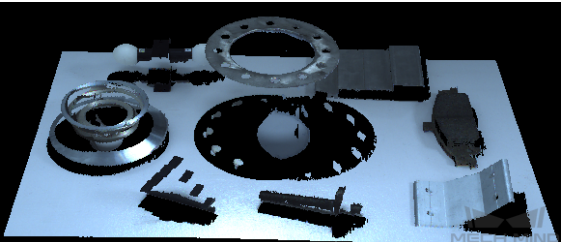
#### Change Reference Frame

Select from the drop-down menu of **Coordinate System** to change the reference frame in which the depth data is displayed.

- **Camera**: Depth data is displayed in the camera reference frame whose origin is located inside the camera.
- **Customized**: Depth data is displayed in the custom reference frame defined with the *Custom Coordinate System* tool. This tool can be open in the **Tool** menu.

#### Change Color Rendering Scheme

Select from the drop-down menu of **Color** to change how the point cloud is colored. Options include **Jet**, **Gray Scale**, **White**, and **Texture**.

Option	Description and Example
Jet	<p data-bbox="336 255 1390 284">Different depth values correspond to different colors in the Jet colormap.</p> 
Gray Scale	<p data-bbox="336 640 1390 669">Different depth values correspond to different grayscale values.</p> 
White	<p data-bbox="336 1025 1390 1055">The point cloud is untextured and displayed as completely white.</p> 
Texture	<p data-bbox="336 1299 1390 1328">The 2D image is used to texture the point cloud.</p> 

**Note:** When **White** or **Texture** is selected, if you switch to the depth map and then back, the **Color** option is automatically changed to the same one selected for the depth map.

## Adjust Display Depth Range

Please refer to *Adjust Display Depth Range* above for the depth map for detailed information.

---

**Note:** When you select **White** or **Texture** in **Color**, adjusting the display depth range will have no effect on the point cloud display. You can see the effect of the adjustment in the depth map display.

---

## Point Cloud Exhibit

This function moves the point cloud along a predetermined trajectory to display it from various angles. It is designed for exhibition and promotion purposes.

To use this function:

1. Enable the function by checking **Point Cloud Exhibit** in the **View** menu.
2. In the upper-right corner of the point cloud display area, click *Play* to start the movement.

---

**Note:** All other functions are disabled while the point cloud is in motion.

---

3. Click the same button (now *Stop*) to end the movement.

---

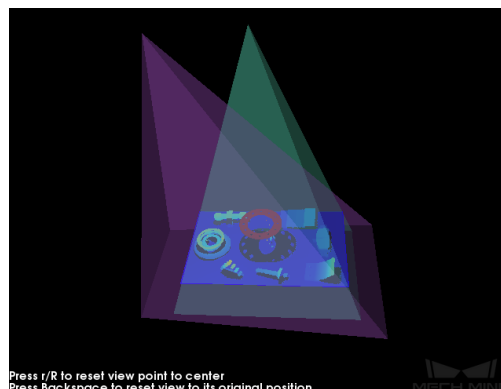
**Hint:** You can hide the *Play* button by unchecking **Point Cloud Exhibit** in the **View** menu.

---

## Other Actions

Right-click the point cloud for the following options:

- **Save Point Cloud:** save the point cloud in PLY format (unit: m) to a specific path.
- **Show Frustrum:** display the view cones of the projector and 2D camera(s).



- **Show Axes:** display the origin and axes of the selected reference frame.

Continue reading for detailed information on the parameter panel and all the parameters.

## 7.3 Parameters

You can adjust the image capturing and processing parameters in the parameter panel to improve the quality of the 2D image, depth map and point cloud.

---

Introduction to the **parameter panel**:

*Parameter Panel*

---

Descriptions of **parameters for the 2D image**:

*Parameters for 2D Image*

---

Descriptions of **the basic parameters for the depth map and point cloud**:

*Basic Parameters for Depth Map and Point Cloud*

---

Descriptions of **the advanced parameters for the depth map and point cloud**:

*Advanced Parameters for Depth Map and Point Cloud*

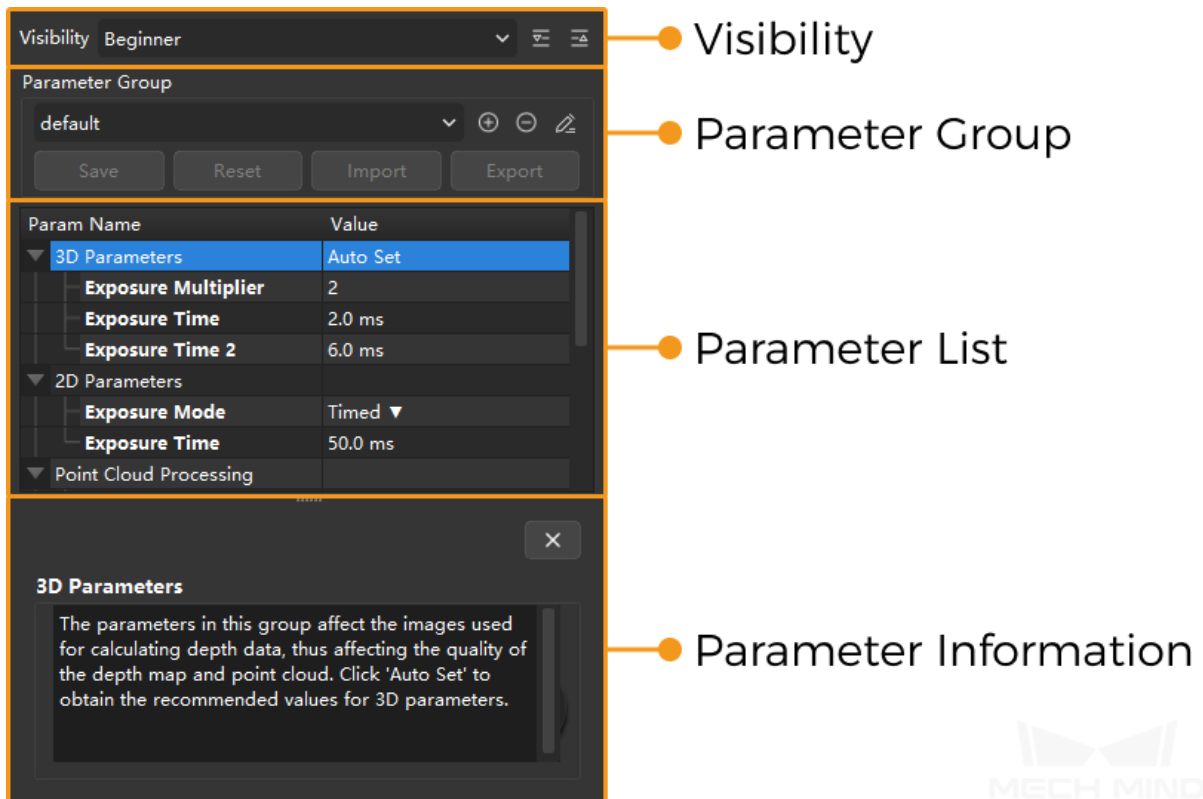
---

### 7.3.1 Parameter Panel

This topic introduces the components and functions of the parameter panel, located on the right side of Mech-Eye Viewer's interface.

The parameter panel consists of the following sections:

- *Visibility*
  - *Parameter Group*
  - *Parameter List*
  - *Parameter Information*
-



### Visibility

Visibility affects what parameters you can see below in the parameter list. There are three levels: **Beginner**, **Expert** and **Guru**.



---

**Note:** The **Guru** level is only available in the **Administrator** account. If you need to use them, please contact Technical Support for the password.

---

- **Beginner** parameters are the basic and frequently used parameters, and are also available under the other **Visibility** levels.
- Additional parameters in **Expert** and **Guru** levels are advanced parameters.


Functions of the buttons to the right:

Button	Description
	Expand all the parameter categories in the parameter list.
	Collapse all the parameter categories in the parameter list.

## Parameter Group

In actual applications, usually different parameter values are used for different target objects, different lighting conditions, etc. You can create parameter groups to save different combinations of parameter values and change the parameter values with a couple of clicks.

### Create and Use a Parameter Group

1. In the **Parameter Group** section, click  to the right of the drop-down menu. A small window will pop up.
2. In the pop-up window, enter a name for the new parameter group, and then click *OK* to save it.

---

**Hint:** The new parameter group is automatically selected after creation.

---

3. Adjust the parameters. Then, click *Save* in the **Parameter Group** section to save the new values to this parameter group.

---

**Note:** An asterisk (\*) after a parameter's name indicates that the current value of this parameter is not saved to the current parameter group. Make sure you save the changes promptly.

---

4. If your project requires different parameter values under different circumstances, create a parameter group for each circumstance.
  - To use a different parameter group:  
In the **Parameter Group** section, click the drop-down button and select from the list.
  - To change the saved values of a parameter group:  
After selecting the parameter group, adjust the parameter values, and then click *Save* in the **Parameter Group** section.




---

**Note:** The **default** and **calib** parameter groups are built-in and cannot be deleted or renamed.

---

### Buttons in the Parameter Group Section

Click to view a table summarizing the functions of the buttons in the **Parameter Group** section.

Button	Description
	Create a new parameter group.
	Delete the currently selected parameter group.
	Rename the currently selected parameter group.
<i>Save</i>	Save the parameter values to the currently selected parameter group. Shortcut: Ctrl + S.
<i>Reset</i>	Reset the parameter values in the currently selected parameter group to default values.
<i>Import</i>	Import parameter groups.
<i>Export</i>	Export parameter groups.

---

**Note:** The *Save*, *Reset*, *Import* and *Export* buttons are not available for virtual cameras.

---

### Parameter List

All the available parameters under the currently selected **Visibility** are listed in this section.

You can adjust parameter values here to improve the quality of the 2D image, depth map and point cloud.

- For descriptions of the parameters for the 2D image, please refer to [Parameters for 2D Image](#).
- For descriptions of the parameters for the depth map and point cloud, please refer to [Basic Parameters for Depth Map and Point Cloud](#) and [Advanced Parameters for Depth Map and Point Cloud](#).

### Parameter Information

Select a parameter in the parameter list section, and the corresponding parameter information is displayed here.

Parameter information in this section includes a brief description of the parameter, **Name**, data **Type**, **Minimum** and **Maximum** values, **Default Value**, and **Visibility**.

---

Continue reading for explanations of parameters. The next topic is on parameters for the 2D image.

## 7.3.2 Parameters for 2D Image

The parameters in the **2D Parameters** category affect the quality of the 2D image.

The general criteria for a good 2D image:

- Not over-exposed.
- Not under-exposed.
- The details that you need are visible and clear enough.

---

**Note:** Depending on your project needs, you may not need to ensure that the entire 2D image has the right brightness. As long as the parts needed for subsequent processes (such as deep learning



inference and template matching) are clear enough for the processing, it is acceptable for irrelevant parts to be over- or under-exposed.

---

**Note:** Before proceeding, make sure you are familiar with the *components and functions of the parameter panel*.

---

The **Beginner** level parameters are basic parameters, and the additional parameters in **Expert** level are advanced parameters.

It is recommended to adjust the basic parameters first. If the quality of the 2D image is still unsatisfactory, adjust the advanced parameters.

- *Basic Parameters for 2D Image*
- *Advanced Parameters for 2D Image*

### **Basic Parameters for 2D Image**










Depending on what camera is connected, you may see different parameters in the **2D Parameters** category, and the options therein also differ.

- If you are connected to a LSR or DEEP series camera: **Color Camera Exposure Mode** and **Monochrome Camera Exposure Mode** are displayed.
- If you are connected to a camera from another series: **Exposure Mode** is displayed.

**Note:** For the LSR and DEEP series:

- **Color Camera Exposure Mode** affects the color 2D camera. The 2D image is obtained by this camera.
  - **Monochrome Camera Exposure Mode** affects the monochrome 2D cameras. The images obtained by these cameras are used for *checking camera intrinsic parameters*, setting the ROI, and calibrating camera extrinsic parameters with Mech-Vision's **Camera Calibration**.
- 

The following table summarizes the options available for each of the above parameters.

	Exposure Mode	Color Camera Exposure Mode	Monochrome Camera Exposure Mode
Timed			
Auto			
HDR			
Flash			

The following table summarizes the suitable scenarios for each exposure mode option.

Exposure mode	Suitable scenario
Timed	Usually used in stable lighting conditions.
Auto	Usually used in varying lighting conditions.
HDR	Usually used for objects with various colors or textures.
Flash	Usually used in dark environments.

The exposure mode option selected determines what other parameters are displayed in the **2D Parameters** category. The following sections explain the parameters corresponding to each option.

- *Timed Exposure Mode*
- *Auto Exposure Mode*
- *HDR Exposure Mode*
- *Flash Exposure Mode*

### Timed Exposure Mode

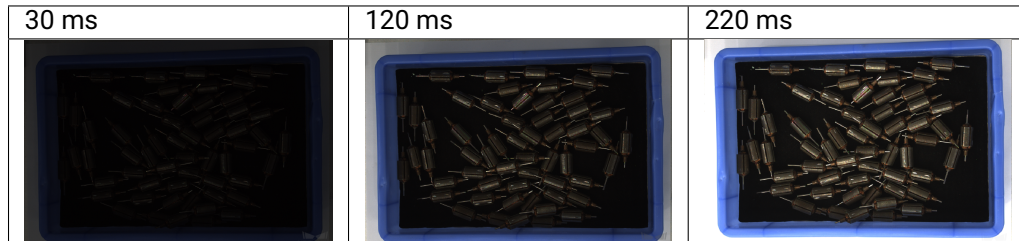
With the **Timed** option, you can set a single exposure time for capturing the image.

- For **Exposure Mode** and **Color Camera Exposure Mode**: adjust the value of **Exposure Time**.
- For **Monochrome Camera Exposure Mode**: adjust the value of **Monochrome Camera Exposure Time**.

### Exposure Time/Monochrome Camera Exposure Time

The value of **Exposure Time/Monochrome Camera Exposure Time** affects the brightness of the image. Usually, long exposure time is used in dark environments, and short exposure time is used in bright environments.

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



- Monochrome images captured with different **Monochrome Camera Exposure Time** values (all other conditions identical):



### Auto Exposure Mode

With the **Auto** option, the exposure time is automatically adjusted based on your settings for **Auto-Exposure ROI** and **Gray Value**.

### Auto-Exposure ROI

If an auto-exposure ROI is set, the exposure time is adjusted based on the lighting, object colors, etc., in the selected region.

---

**Note:** If you do not set an auto-exposure ROI, the entire camera FOV is taken into consideration during exposure time adjustment.

---

To set an auto-exposure ROI:



1. Double-click *Edit* next to **Auto-Exposure ROI**.

2. Drag the cursor on the image to select an auto-exposure ROI. Adjust the size and position of the selection box if needed.







**Hint:** Please select the area where the target objects are located and avoid including irrelevant objects as much as possible.

3. Click *Apply* to finish setting the auto-exposure ROI.

**Hint:** Click *Clear* to delete the set auto-exposure ROI.

4. Click  or  to perform image capturing. The brightness of the 2D image is automatically adjusted based on the auto-exposure ROI.

- 2D images captured with different auto-exposure ROIs (all other conditions identical):

Auto-Exposure ROI	2D Image
	
	
	

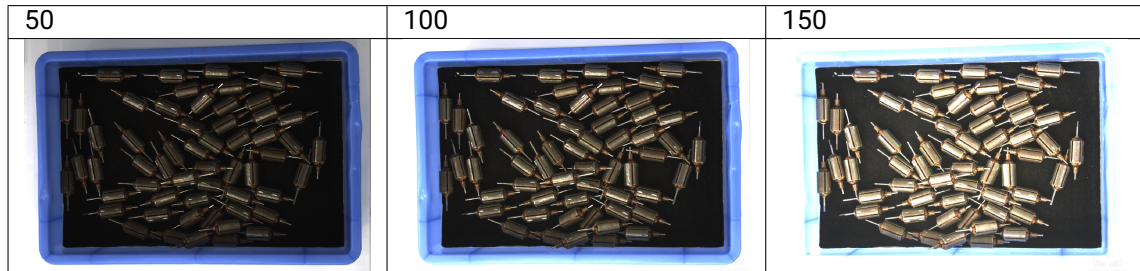
## Gray Value

This parameter affects the brightness of the 2D image:

- Increasing the parameter value makes the 2D image brighter.
- Decreasing the parameter value makes the 2D image darker.

It is recommended to set an auto-exposure ROI first. Adjust this parameter if the brightness of the 2D image is still unsatisfactory.

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




---

### Note:

- For monochrome images, this parameter sets the intensity of the image.
  - For color images, this parameter sets the intensity of each color channel.
- 

## HDR Exposure Mode

With the **HDR** option, you can set multiple exposure times, and the captured images are merged to generate a 2D image that retains more details in the highlights and shadows.

The parameters available with the **HDR** option are **Exposure Sequence** and **Tone Mapping**.

### Exposure Sequence

Set an exposure sequence of multiple exposure times, and the captured images are merged to generate a 2D image that retains more details in the highlights and shadows.

To set an exposure sequence:

1. Double-click *Edit* next to **Exposure Sequence**.
2. Click **+** to add an exposure time. Then, adjust the duration of this exposure time.

---

**Hint:** To delete an exposure time, select it and click **-**.

---

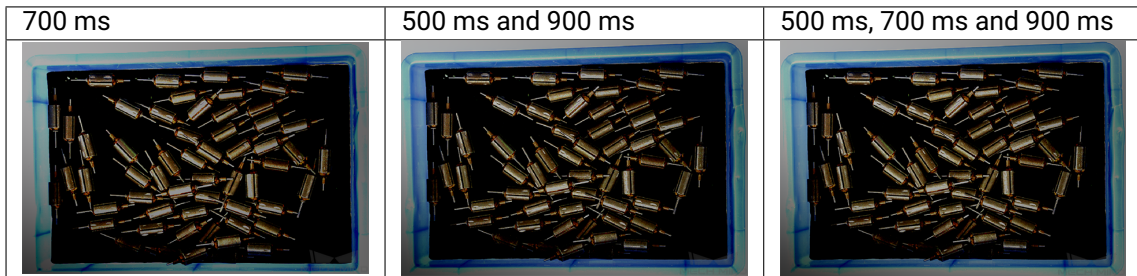
3. Click *Apply* to save the exposure sequence.

**Hint:**

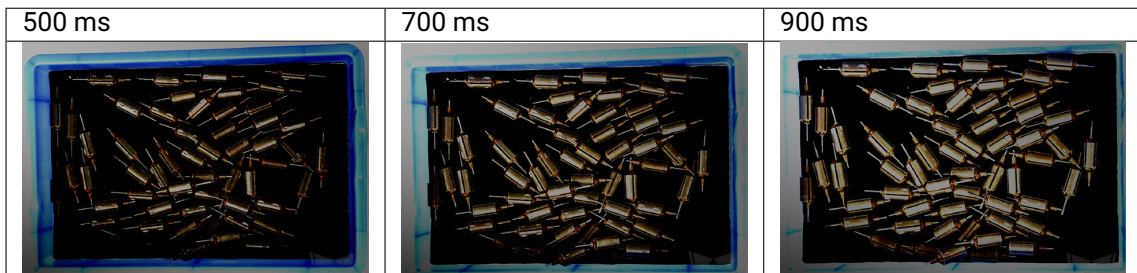
- Click *Reset* to delete all the set exposure times.
- Click *Cancel* to exit the window without saving the changes.

**Note:** If you add only one exposure time to the sequence, it is the same as selecting the **Timed** mode.

- 2D images captured with different exposure sequences (all other conditions identical):



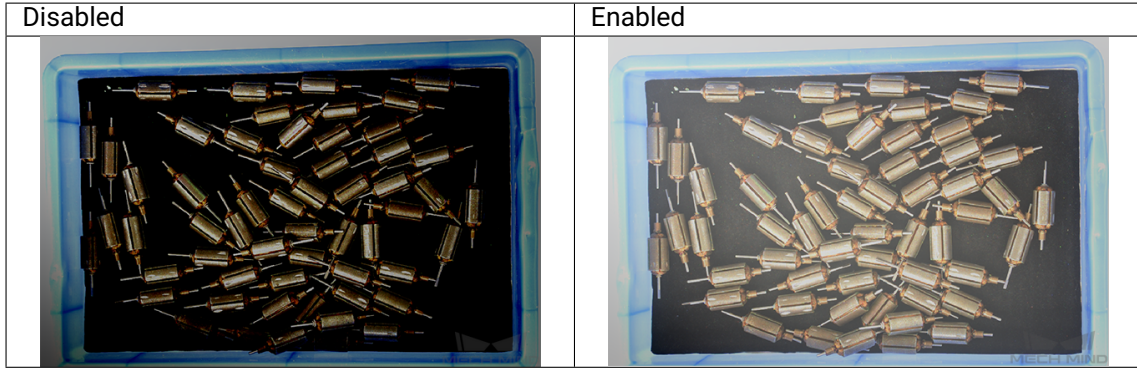
- For comparison: 2D images captured with single exposure times (all other conditions identical):



### Tone Mapping

Enabling **Tone Mapping** can make the colors of the image look more natural. If the 2D image appears very different from the actual objects, please enable this function.

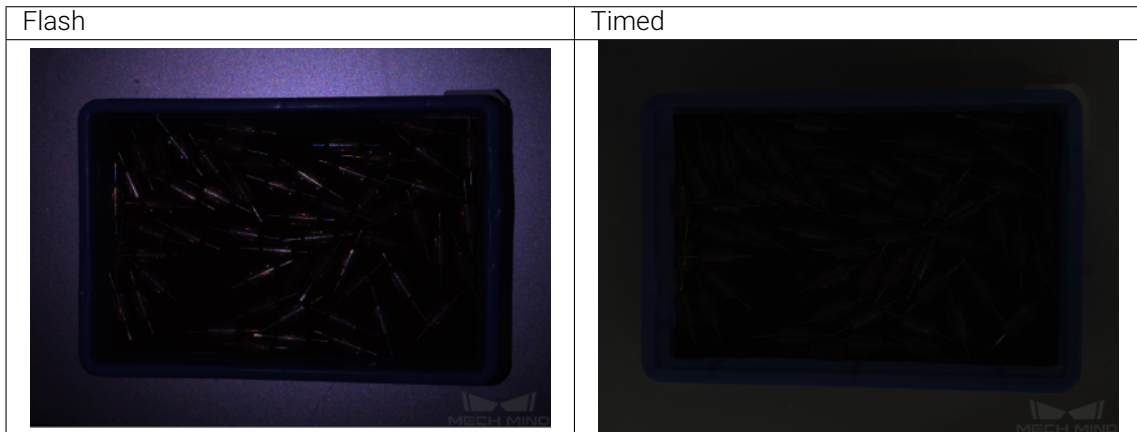
- 2D images captured with **Tone Mapping** disabled and enabled (all other conditions identical):



### Flash Exposure Mode

With the Flash option, *the projector* is used to project supplemental light.

- 2D images obtained with the Flash exposure mode and Timed exposure mode (all other conditions identical):



The brightness of the 2D image obtained with the Flash exposure mode can be adjusted by Exposure Time in the 3D Parameters Category.

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



### Advanced Parameters for 2D Image

The **Gain** parameter in the **3D Parameters** category also affects the 2D image. This parameter is available at **Expert** and **Guru** visibility level.

#### Gain

This parameter belongs to the **3D Parameters** category. It affects the brightness of the captured image. At the same time, more noise is introduced.

The higher the value of **Gain**, the brighter the captured image, and the more noise introduced.

**Note:** If adjusting exposure mode/time does not produce the desired image brightness, you can increase the value of **Gain**. However, this introduces more noise to the 2D image.

- 2D images captured with different **Gain** values (all other conditions identical):










**Hint:** For models with color 2D cameras, the color of the 2D image may differ significantly from that of the actual objects. In such cases, you can adjust the white balance setting of the color 2D camera. For detailed information, please refer to [Auto White Balance](#).

Continue reading for descriptions of the basic parameters for the depth map and point cloud.

### 7.3.3 Basic Parameters for Depth Map and Point Cloud

The following table summarizes which categories of parameters are adjusted for the depth map and point cloud:



	Depth map	Point Cloud
<i>3D Parameters</i>		
<i>Point Cloud Processing</i>		
<i>Depth Range</i>		
<i>ROI</i>		

**Note:** Before proceeding, make sure you are familiar with the *components and functions of the parameter panel*.

The **Beginner** level parameters are basic parameters and are covered in this topic. The additional parameters in **Expert** and **Guru** levels are advanced parameters and are covered in the next topic.

### 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.

The **Beginner** level parameters in this category include **Exposure Multiplier** and **Exposure Time**.

#### Exposure Multiplier

This parameter sets the number of exposure times. If the value of **Exposure Multiplier** is greater than 1, multiple **Exposure Time** parameters must be set. You can set at most three exposure times.

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.

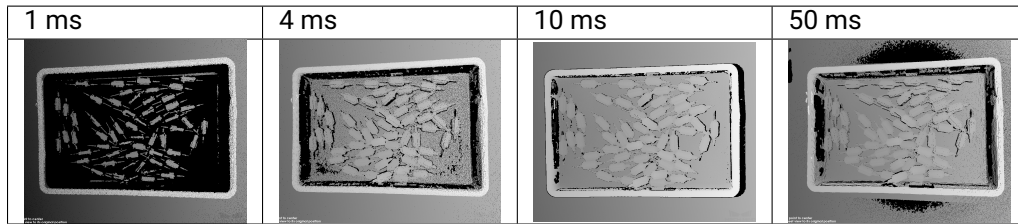
#### Exposure Time

This parameter sets the exposure time for capturing *the images used for calculating depth data*.

Usually, long exposure time is used for dark objects, and short exposure time is used for light objects.

- Exposure time too long or too short both may cause loss of depth information in the depth map and point cloud.

Point clouds generated from images captured with different **Exposure Time** values (all other conditions identical):



**Hint:** The regions in black lack the points corresponding the object surface and thus are showing the background color.

**Note:**

- The exposure time of a laser camera must be a multiple of 4. The entered value is automatically adjusted.
- The minimum value that can be set is 4 ms for the Laser series and 8 ms for other laser cameras.

**Hint:** Long exposure time(s) increases the capture time and thus may negatively affect the project cycle time. You can try using a larger value of *Gain* in combination with relatively short exposure time(s).

You can use *Exposure Assistant* to obtain the recommended values for 3D parameters. To open this tool, double click **Auto Set** next to **3D Parameters** in the parameter panel or click *Tools* → *Exposure Assistant*.

**UHP**

**Hint:** This parameter category is only available for the UHP series.

**Capture Mode**

Set which 2D camera(s) is used for image capturing. A UHP camera contains two 2D cameras, the numbering below corresponds to that displayed in *View 2D Camera and Set Parameters*.

Option	Description
Camera1	Use Camera 1 to capture images.
Camera2	Use Camera 2 to capture images.
Merge	Use both 2D cameras to capture images. The depth map and point cloud are generated by merging the images. 2D images are captured by Camera 1.

**Note:** Only Camera1 mode supports setting an ROI.

### Point Cloud Processing

The parameters in this category are used to process the generated point cloud. Please perform image capturing again after adjusting the parameters to see the result.

The **Beginner** level parameters in this category include **Point Cloud Smoothing**, **Outlier Removal** and **Stripe Contrast Threshold**.

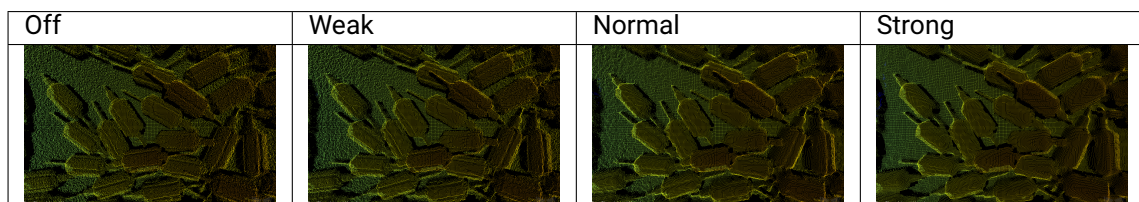
#### Point Cloud Smoothing

This function reduces the depth fluctuation in the point cloud and improves its resemblance to the actual object surface.

The intensity levels of point cloud smoothing include **Weak**, **Normal** and **Strong**. Select **Off** to disable point cloud smoothing.

**Note:** Point cloud smoothing causes loss of object surface details. The more intense the smoothing, the more details are lost.

- Point clouds generated with different smoothing intensities (all other conditions identical):



#### Outlier Removal

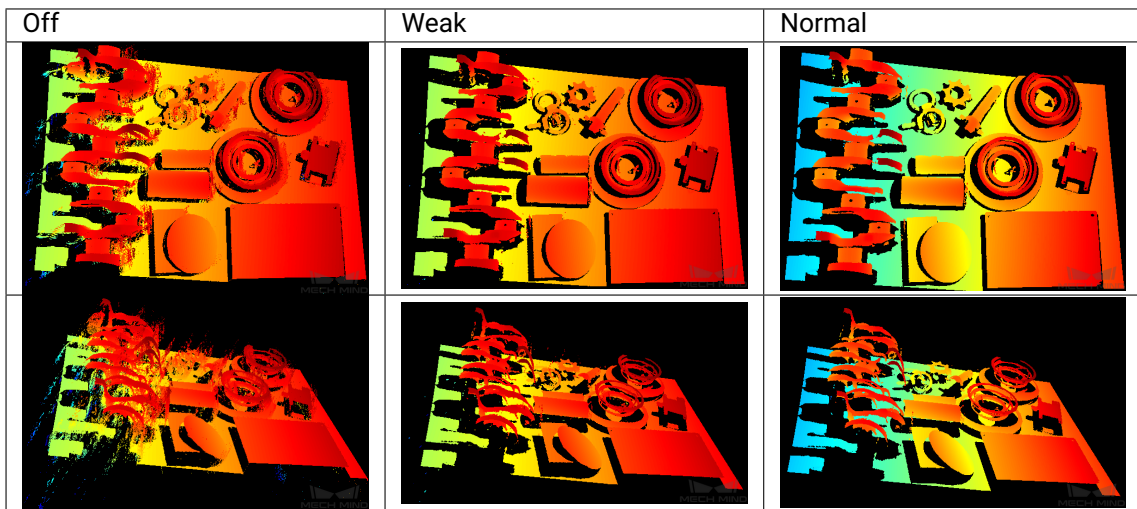
This function reduces the amount of noise in the point cloud, thus reducing the impact on the precision and accuracy of subsequent calculation.

The intensity levels of outlier removal include **Weak** and **Normal**. Select **Off** to disable outlier removal.

**Note:**

- Outlier removal might remove some of the object point cloud. The more intense the outlier removal, the more object point cloud might be removed.
- If this function removes the point cloud needed, please reduce the intensity. However, more noise will be retained.

- Point clouds generated with different outlier removal intensities (all other conditions identical):



### Stripe Contrast Threshold

Increasing the value of this parameter can remove noise in the depth map and point cloud. However, the points of dark objects might be lost.

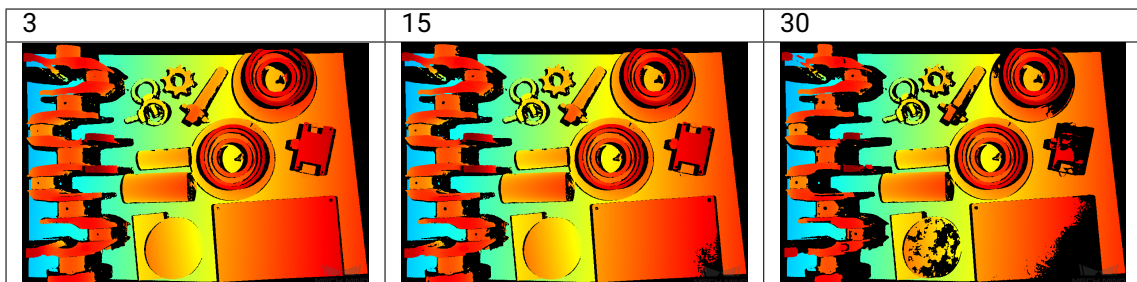
---

**Note:** It is recommended to adjust **Point Cloud Smoothing** and **Outlier Removal** first. If the level of noise is still high, please increase the value of this parameter.

---

The “stripe” in the parameter name refers to stripes in *the structured light pattern captured in the images*.

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



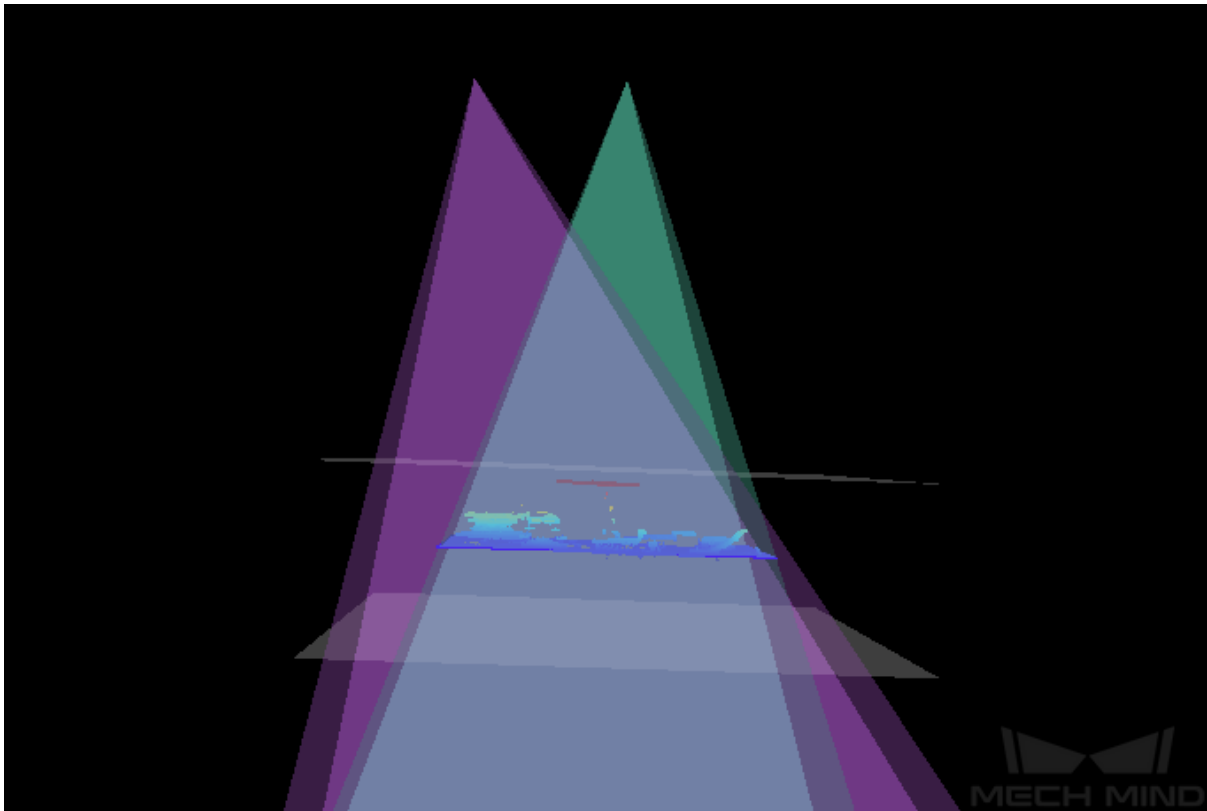
## Depth Range

This parameter sets the depth range in the camera reference frame. Points outside this range are removed from the depth map and point cloud.

**Hint:** Adjusting this parameter simultaneously changes the max and min depth values in the depth map and point cloud display.

To set the depth range:

1. Double-click *Edit* next to **Depth Range**.
2. In the pop-up window, adjust the view of the point cloud and the depth range.
  - Adjust the view of the point cloud: Use the actions listed [here](#) to adjust the view. A good view allows you to see the two gray squares representing the lower and upper limits of the depth range and the obtained point cloud. All points between the two squares will be preserved.



- Adjust the depth range: Drag the handles on the slider or enter the values in the text boxes. An appropriate range would sandwich all the desired points between the two gray squares while excluding unwanted points as much as possible.

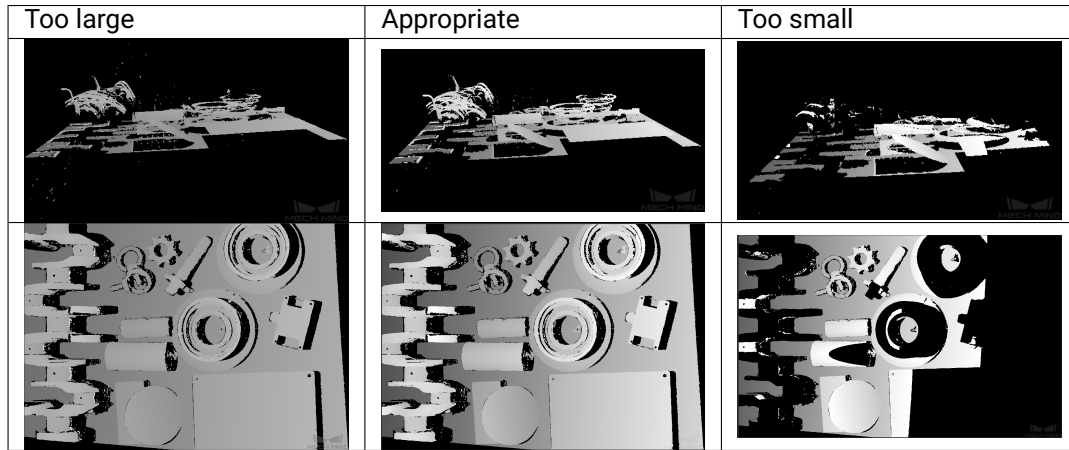
**Note:**

- Click *Recommended Range* to automatically adjust the depth range to the *recommended working distance* of the camera model.

- Click *Reset* to restore the last saved depth range.

3. Click *Save* to save the changes.

- The same point cloud with different **Depth Range** values:



**Note:** If the depth range is too large, more noise are preserved. If the depth range is too small, points that are needed may be cut off.

**Hint:** You can use *Depth Analyzer* to check the depth map quality.

## ROI

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



To set an ROI:

1. Double-click *Edit* next to **ROI**.
2. Drag the cursor on the image to select an ROI. Adjust the size and position of the selection box if needed.

**Note:** For the LSR series, if the image displayed here is too bright/dark, please adjust the *Monochrome Camera Exposure Mode* parameter.

3. Click *Apply* to finish setting the ROI.

**Hint:** Click *Clear* to delete the set ROI.

4. Click  or  to perform image capturing. The obtained depth map and point cloud is now trimmed by the set ROI.

---

**Hint:** If the quality of the depth map and point cloud is still unsatisfactory after adjusting the parameters, you can save the camera raw data in the **File** menu and send it to Technical Support.

---

Continue reading for descriptions of the advanced parameters for the depth map and point cloud.

### 7.3.4 Advanced Parameters for Depth Map and Point Cloud

---

**Note:** Before proceeding, make sure you are familiar with the *components and functions of the parameter panel*.

---

The advanced parameters refer to the additional parameters available in **Expert** and **Guru** visibility levels. This topic currently only cover those available in the **Expert** level.

---

**Note:** It is recommended to adjust the *basic parameters* first before you move on to the advanced parameters.

---

- *Gain*
- *Projector*
  - *Light Brightness*
  - *Fringe Coding Mode*
  - *Anti-Flicker Mode*
- *Laser*
  - *Fringe Coding Mode*
  - *Laser Power*

#### Gain

This parameter belongs to the **3D Parameters** category. It affects the brightness of the captured image. At the same time, more noise is introduced.

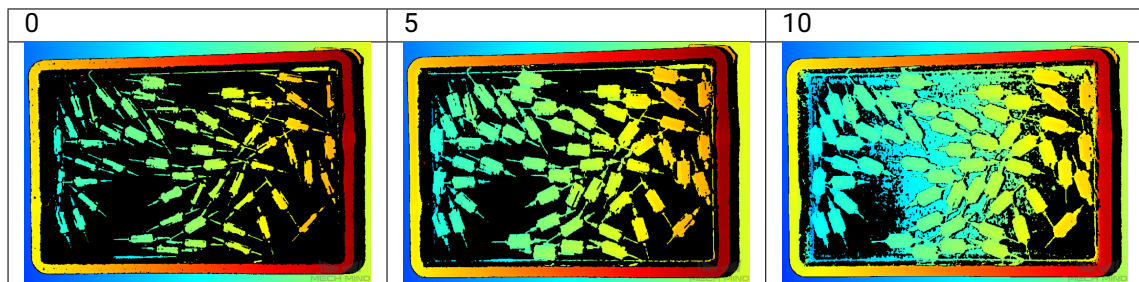
The higher the value of **Gain**, the brighter the captured image, and the more noise introduced.

---

**Note:** This parameter affects the quality of the depth map and point cloud by changing the brightness of *the images used for calculating depth data*.

---

- Point clouds generated with different **Gain** values (all other conditions identical):




---

**Hint:** For dark objects, *long exposure time(s)* may be required for obtaining the depth map and point cloud with relatively complete depth information. Long exposure time(s) increases the capture time and thus may negatively affect the project cycle time. You can try using a larger value of **Gain** in combination with relatively short exposure time(s).

---

### Projector

---

**Hint:** This parameter category is only available for non-laser structured light cameras.

---

This parameter category belongs to the **3D Parameters** category.

### Light Brightness

- Set the brightness of the projected structured light.
- Options:
  - **High:** High brightness. For dark objects.
  - **Normal:** Normal brightness. For objects of regular surfaces.
  - **Low:** Low brightness. For reflective objects.

### Fringe Coding Mode

---

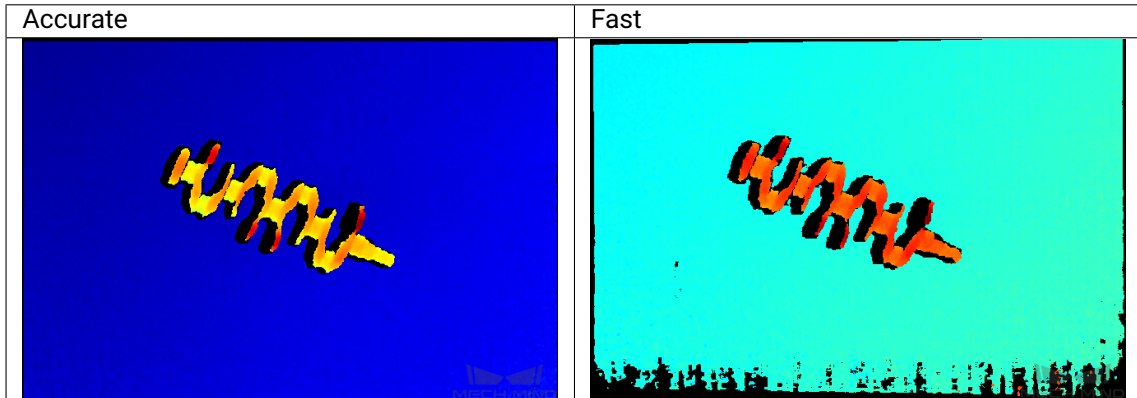
**Hint:** This parameter is only available for the Nano (V3), Pro XS (V3), NANO (V4), PRO S (V4) and PRO M (V4) series.

---

- This parameter changes the structured light patterns used, which affects the image capturing time and depth data quality.
- Options:
  - **Fast:** faster image capturing, but the quality of the depth data is lower.



- **Accurate:** slower image capturing, but the quality of the depth data is higher.
- Point clouds captured with different **Fringe Coding Mode** (all other conditions identical):



### Anti-Flicker Mode

---

**Hint:** This parameter is only available for the Nano (V3), NANO (V4), PRO S (V4) and PRO M (V4) series.

---

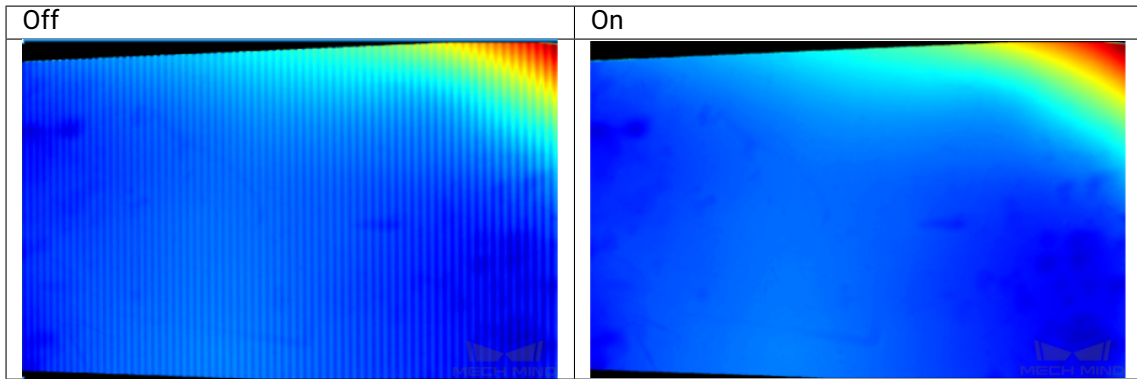
- This function can reduce depth fluctuation caused by flicker. Flicker refers to the rapid and periodical change in the intensity of artificial light. This function adjusts the projection frequency of the structured light to reduce the fluctuation caused by flicker.
- Options:
  - Off
  - AC50Hz
  - AC60Hz

---

**Note:** Please select the option that corresponds to the frequency of the power supply in your region. The AC frequency in most countries and regions is 50 Hz. The AC frequency in the U.S. and some Asian countries is 60 Hz.

---

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



### Laser

---

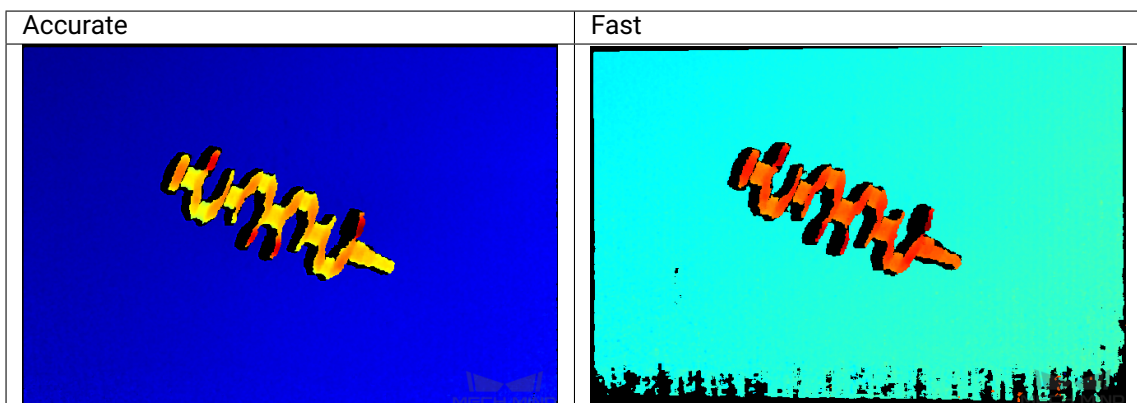
**Hint:** This parameter category is only available for the Laser (V3) and LSR (V4) series.

---

This parameter category belongs to the **3D Parameters** category.

### Fringe Coding Mode

- This parameter changes the structured light patterns used, which affects the image capturing time and depth data quality.
- Options:
  - **Fast:** faster image capturing, but the quality of the depth data is lower.
  - **Accurate:** slower image capturing, but the quality of the depth data is higher.
- Point clouds captured with different **Fringe Coding Mode** (all other conditions identical):



### Laser Power

- This parameter sets the projection intensity of the laser, which affects the brightness of the structured light. Higher intensity is usually used for dark objects, and lower intensity is usually used for highly reflective objects.
- Value range: 20–100% (Default is 100%)
- The default value of 100% works for most cases.


This is the last topic on the parameters in Mech-Eye Viewer.

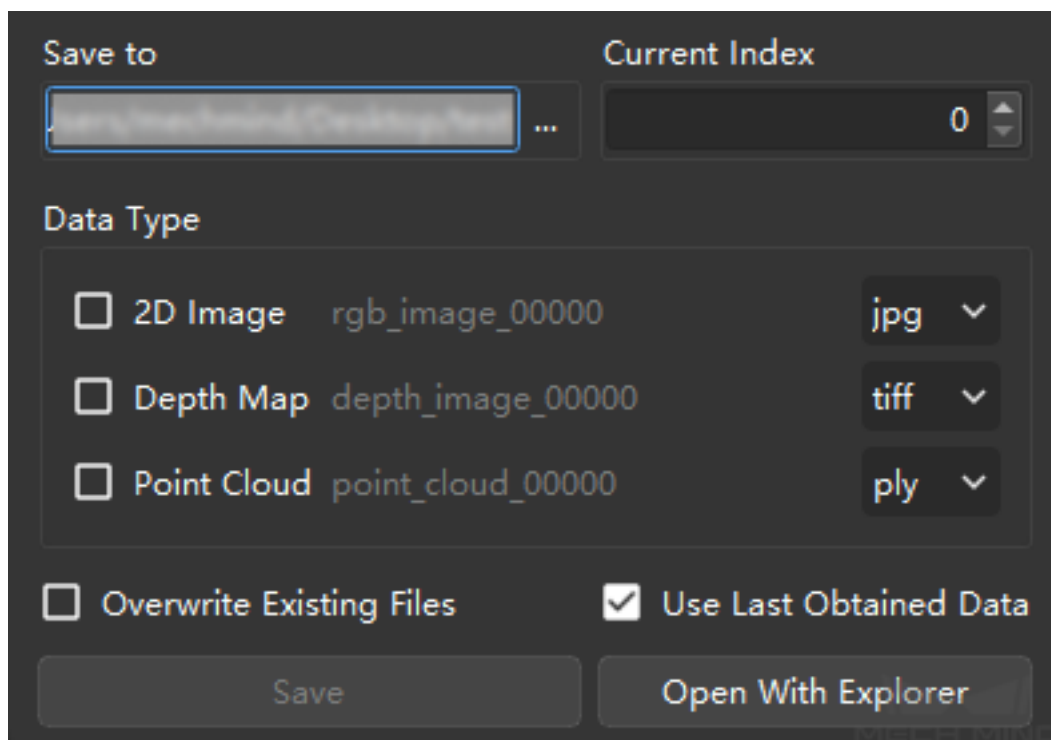
Continue reading for ways you can save data in Mech-Eye Viewer.

## 7.4 Save Data

In Mech-Eye Viewer, you can save the obtained 2D image, depth map and point cloud or the raw data from the camera.

### 7.4.1 Save Images

You can save the last obtained 2D image, depth map, and/or point cloud by clicking  in the camera toolbar.



Options in the **Save Images** window are as follows:

Option	Description
Save to	Select the path to which the data will be saved.
Current Index	Set an index and the software will automatically add the index to the filenames.
Data Type	Select the data type(s) that you would like to save.
Overwrite Existing Files	If checked, existing files will be overwritten by new files with the same names.
Use Last Obtained Data	If checked, the last obtained data will be saved. If unchecked, the camera will perform capturing after you click <b>Save</b> .
Save	Click to save the selected data.
Open With Explorer	Open the selected path in file explorer.

### 7.4.2 Save Raw Data of Camera

You can save the raw data captured by the camera in MRAW format by clicking *File* → *Save Raw Data of Camera*.

The raw data can help Technical Support locate the causes of issues. Therefore, please save the raw data of the problematic capture before contacting Technical Support.

In addition, if you wish to adjust the **Point Cloud Processing**, **Depth Range** and **ROI** parameters later on, you can load the raw data through *File* → *Start Virtual Camera* and adjust these parameters. You can save the 2D image, depth map and point cloud from a virtual camera as well.

## 7.5 Log

Click *Show Log* in the camera toolbar to open the **Server Logs** window. You can view, clear and export logs in this window.

```

0,"cmd":"CaptureImage","image_type":1308})
8/12 16:19:47[!] 0.0018 s 3D Camera Capture Images Time
8/12 16:19:47[!] Command: CaptureImage. Send data: 6912071 bytes.
8/12 16:19:48[!] sendAndRcvMsg {
  "camera_2d_index": 0,
  "cmd": "CaptureImage",
  "image_type": 1308
}

8/12 16:19:48[!] Camera server receive request: QJsonObject({"camera_2d_index":
0,"cmd":"CaptureImage","image_type":1308})
8/12 16:19:48[!] 0.0012 s 3D Camera Capture Images Time
8/12 16:19:48[!] Command: CaptureImage. Send data: 6912071 bytes.
8/12 16:19:52[!] sendAndRcvMsg {
  "command": 1,
  "fileName": "20220812161927_592.log",
  "seekAtStart": false
}
    
```

i W C F
Auto Refresh Clear Clear Server Logs Show Log Entries

- *Auto Refresh*: automatically refresh the log when enabled.
- *Clear*: clear the displayed log entries.
- *Clear Server Logs*: permanently delete all logs.
- *Show Log Entries*: display a list of log entries. you can export logs here (see below).

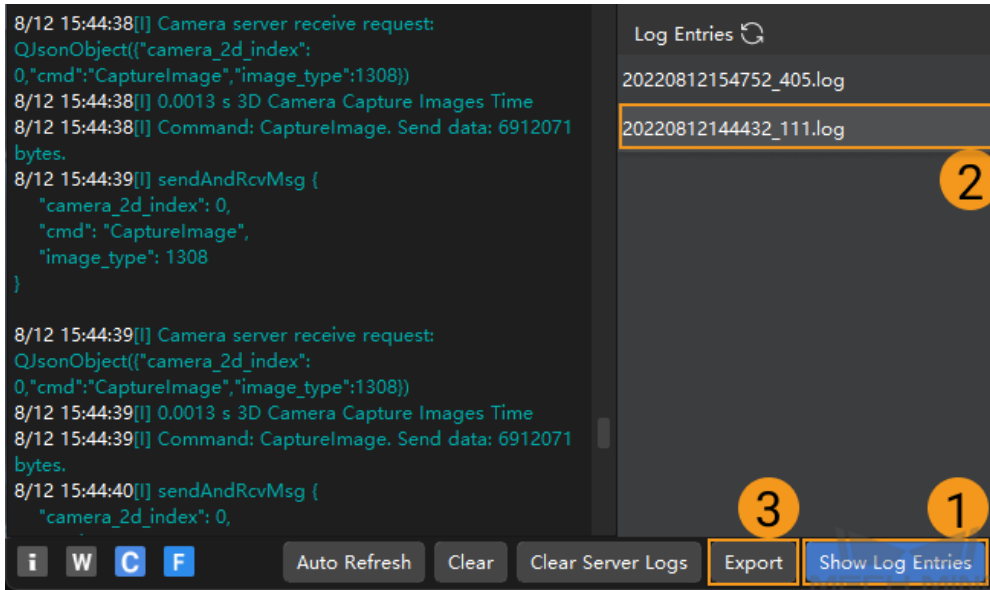
### 7.5.1 Log Level

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 camera from operating.
- *F*: FATAL, errors that severely affects the camera's operation.

### 7.5.2 Export Logs

1. Click *Show Log Entries*, and the *Export* button appears.



2. Select the log you'd like to export, and click *Export*.

---

**Note:** Only a single log can be selected at a time.

---

## BUILT-IN TOOLS

Mech-Eye Viewer comes with built-in tools that can be accessed from the **Tools** menu.

Tool	Description
<i>Check Camera Intrinsic Parameters</i>	Check whether the camera's intrinsic parameters are consistent with the factory settings.
<i>Exposure Assistant</i>	Obtain recommended values for <b>3D Parameters</b> .
<i>Depth Analyzer</i>	Check the imaging quality by checking the depth fluctuation along the selected section.
<i>View 2D Camera and Set Parameters</i>	View and set parameters of the 2D camera(s).
<i>Camera Firmware Upgrade</i>	Update camera firmware.
<i>FOV Calculator</i>	Calculate the camera's FOV based on the selected working distance.
<i>Custom Coordinate System</i>	Set a custom reference frame for displaying the depth data of the depth map and point cloud.
<i>Camera Controller</i>	View camera type, date and time, CPU temperature and projector temperature, etc.
<i>Full Pallet Simulator</i>	Check if the top layer of a full palletized load is within the camera's FOV.

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

You can display/hide the tools in toolbar by clicking *More Tools*. To display a tool in the toolbar, check the box in front of it; to hide, uncheck the box. Click *OK* to save the changes.

## 8.1 Check Camera Intrinsic Parameters

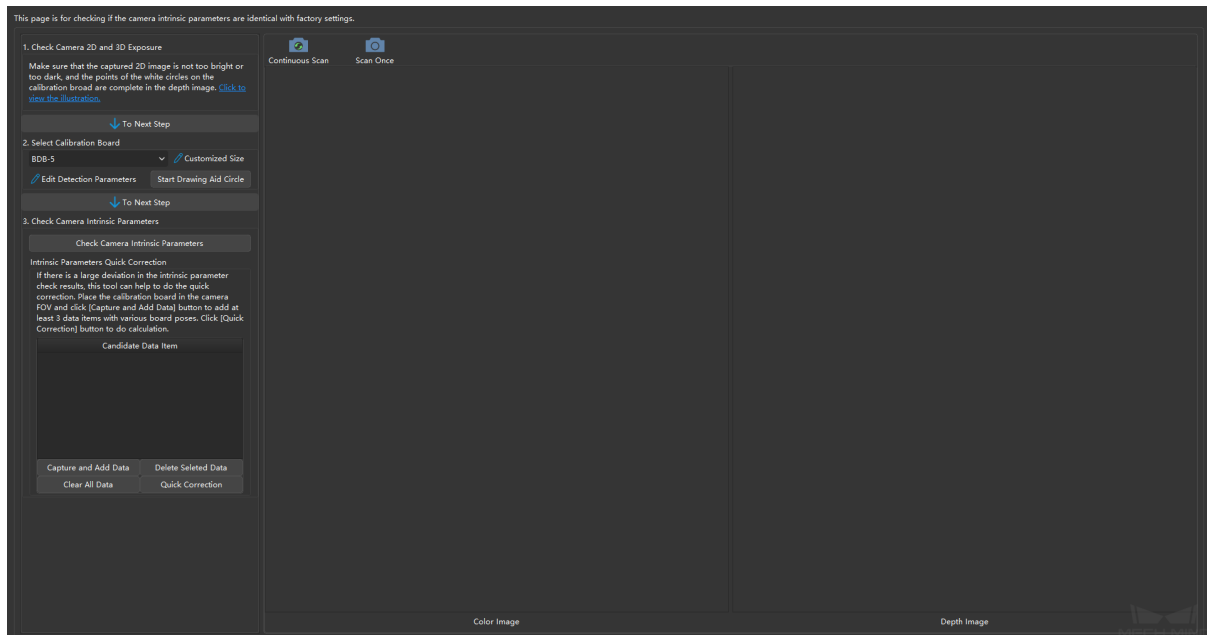
### 8.1.1 Note

- Checking camera intrinsic parameters is to check whether the actual camera intrinsic parameters are **consistent with the factory settings**.
- It is important that the camera's intrinsic parameters are correct. **If not**, the calibration result will be affected, the extrinsic parameters obtained from calibration will be incorrect, and **object poses obtained will thus be incorrect**.
- Camera intrinsic parameter checking needs to be completed with a **calibration board**.

## 8.1.2 Interface

The interface for checking camera intrinsic parameters is as shown in the figure below.

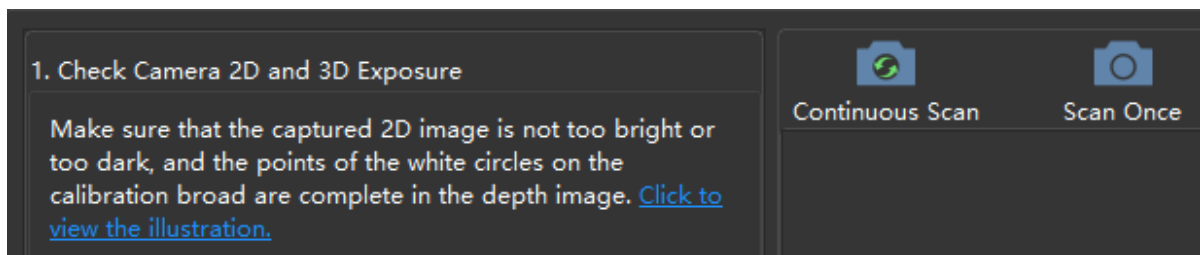
You can open the interface via *Tools* → *Check Camera Intrinsic Parameters*.



## 8.1.3 Steps

### 1. Check camera 2D and 3D exposure

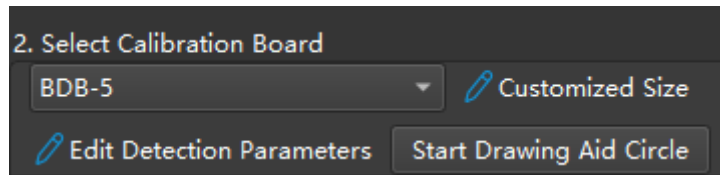
Please click *Continuous Scan* or *Scan Once* to capture images. Adjust *2D* and *3D* parameters to ensure the 2D images are neither too bright nor too dark and the depth maps contain the complete circles on the calibration board.





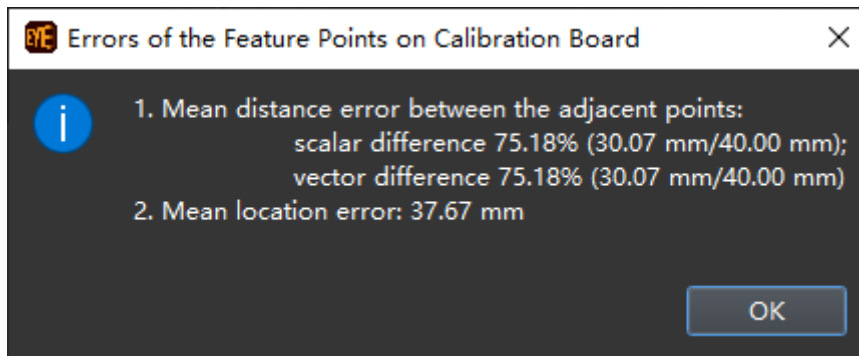
### 1. Select a calibration board

You can select a listed calibration board or use a customized calibration board by setting the size via *Customized Size*. The interface is as shown in the figure below.

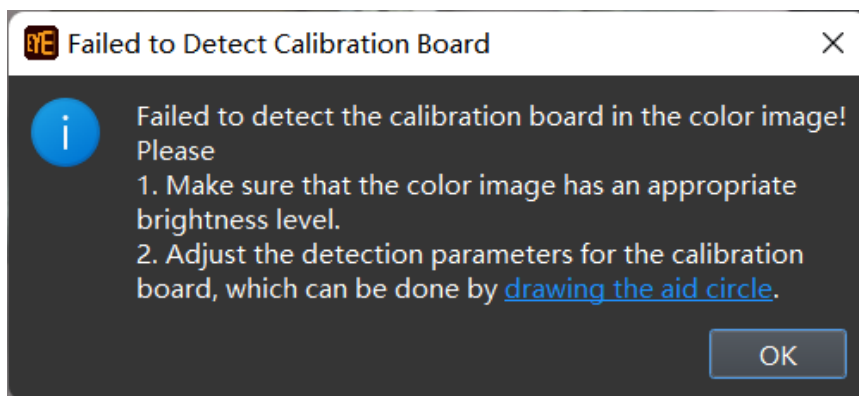


### 3. Check camera intrinsic parameters

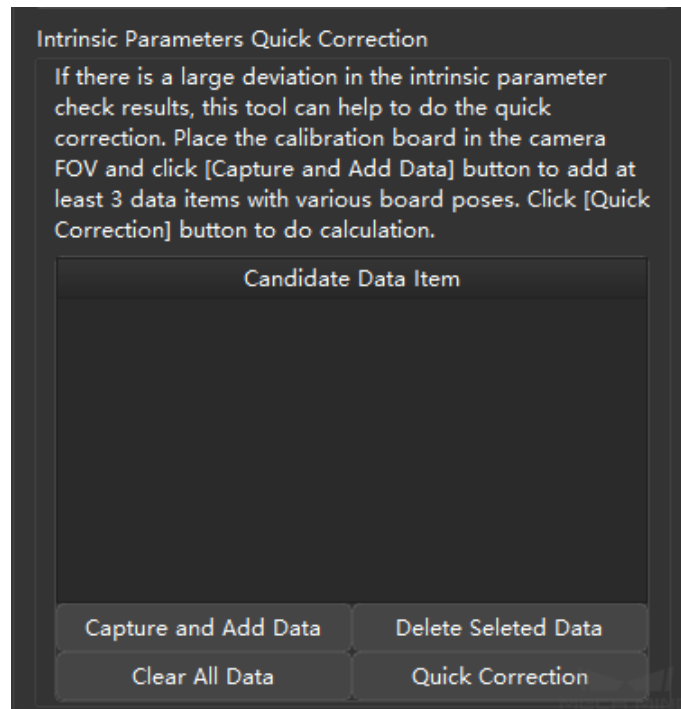
Please click *Check Camera Intrinsic Parameters* to start checking. If the checking is successfully completed, a window indicating errors of the feature points on calibration board as shown below will pop up.



If the checking failed, a window saying **failed to detect calibration board** as shown below will pop up.



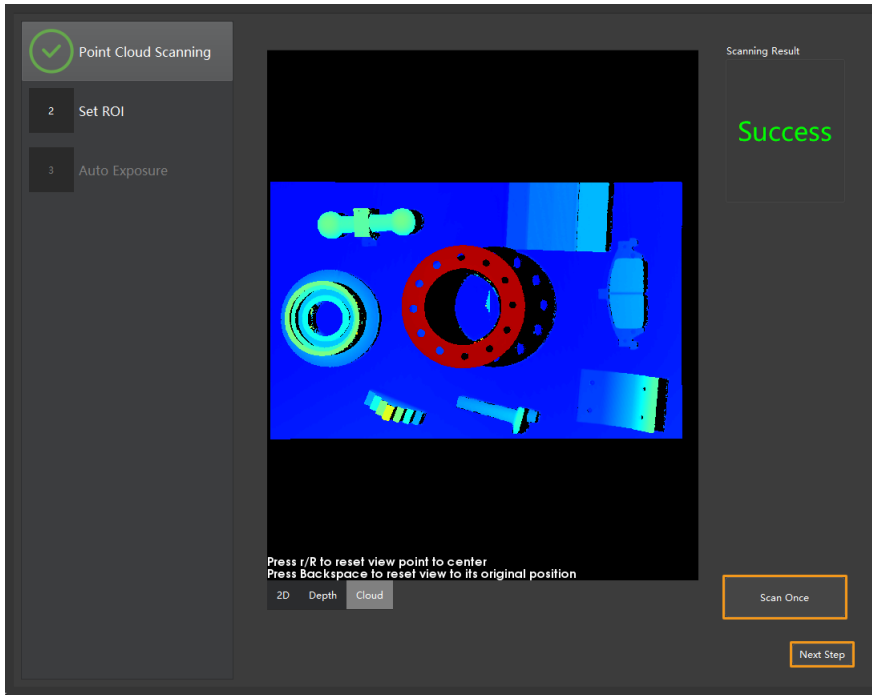
If the intrinsic parameters are inconsistent with the factory settings, please click *Quick Correction* and follow the instructions, as shown below.



## 8.2 Exposure Assistant

You can obtain the recommended values for **3D Parameters** with this tool. Click *Tools* → *Exposure Assistant* in the menu bar.

1. Click *Scan Once* in the lower right, as shown below. Once a point cloud is displayed, click *Next Step*.

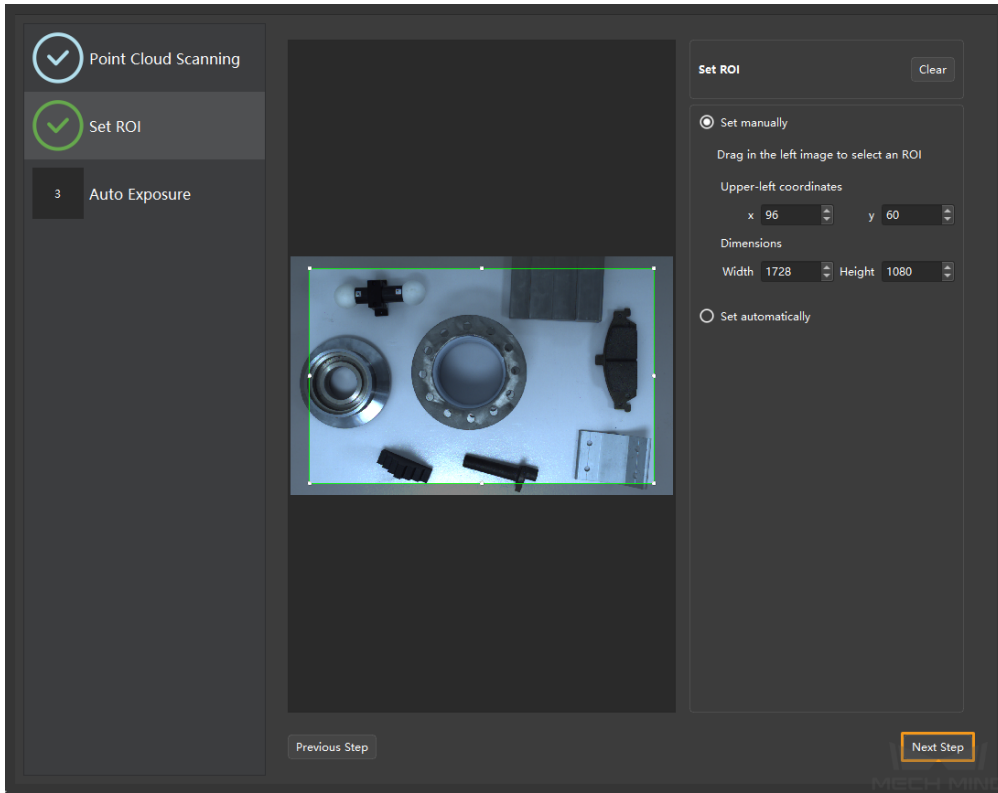


2. Set ROI: click and drag on the 2D image to manually select an ROI, or select **Set automatically** to let the tool automatically select an ROI. Then, click *Next Step*.

---

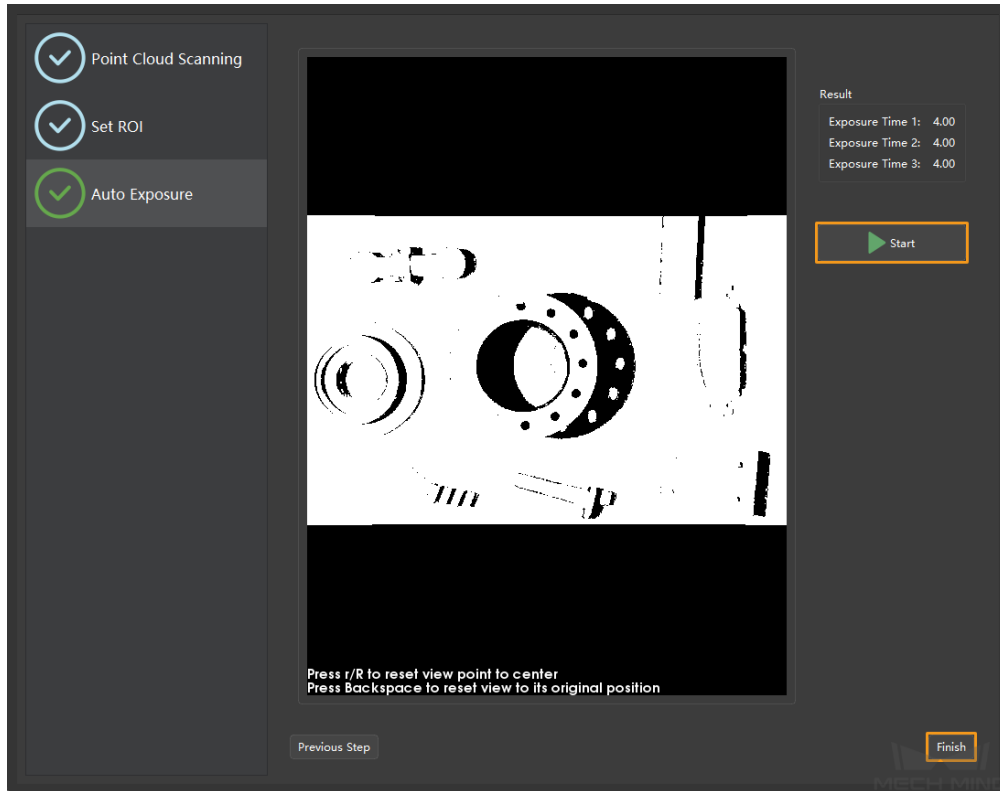
**Hint:** Click *Clear* in the upper-right to clear the selection.

---

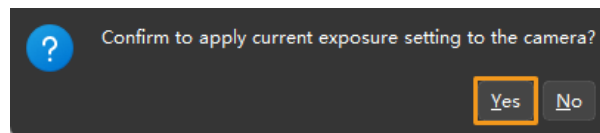


**Note:** The parameters of the ROI selected here are automatically filled to ROI in the parameter panel.

3. Click *Start* to obtain the recommended values for **3D Parameters**. Once a point cloud is displayed, click *Finish* in the lower right.

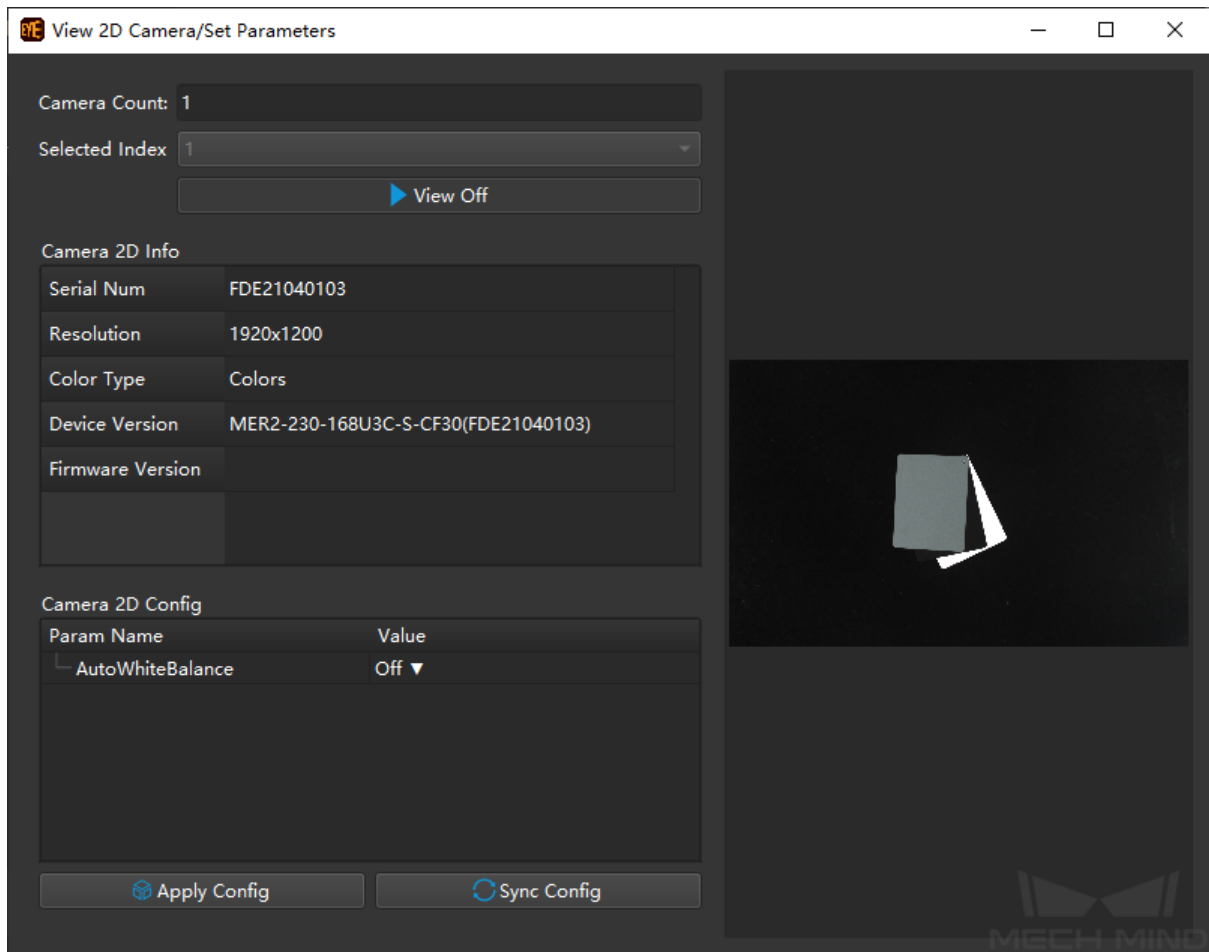


4. A window will pop up asking if you would like to apply the obtained values to the camera. Click Yes to apply. The values are automatically filled to 3D Parameters in the parameter panel.



### 8.3 View 2D Camera and Set Parameters

This tool helps view the camera information and configuration. You can open the tool via *Tools* → *View 2D Camera and Set Parameters*. The interface is as shown in the figure below.



### 8.3.1 Camera Information

Under **Camera 2D Info**, you can view the serial number, resolution, color type, device version, and firmware version.

### 8.3.2 Camera Configuration

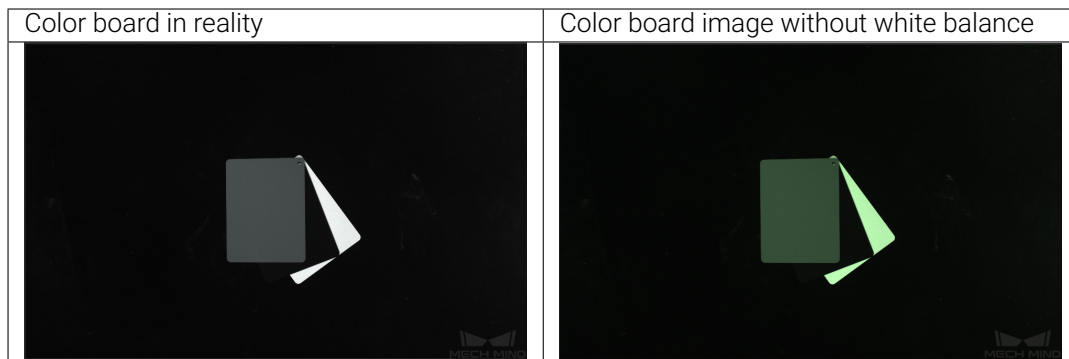
#### Auto White Balance

If the captured images' colors are obviously different from the reality, white balance needs to be applied; otherwise, the colors of 2D images and point clouds may be distorted, thus affecting further processing.

In particular, for deep learning, if images with color deviations are used for model training, the deviations will be extracted and trained as the object features, thus affecting the model's performance on object recognition.

You can enable the auto white balance function by changing the value of **AutoWhiteBalance**.

An example of the contrast between reality and an image without white balance:



### Instructions

1. Select the camera that needs white balance by setting the **Selected Index**, and click *View On*.
2. For scenes with relatively constant ambient light, please set the value of parameter **AutoWhiteBalance** to **Once**; for scenes with changing ambient light, please set the value of parameter **AutoWhiteBalance** to **Continuous**.
3. Once the image's color is not distorted, please set the value of parameter **AutoWhiteBalance** to **Off**, and click *View Off* to finish.
4. Click *Apply Config* to save the configuration to the camera.

---

**Hint:** For white balance, the configuration to save is the white balance parameters for the color channels obtained from auto white balance, so that the camera knows how to adjust white balance in future image capturing.

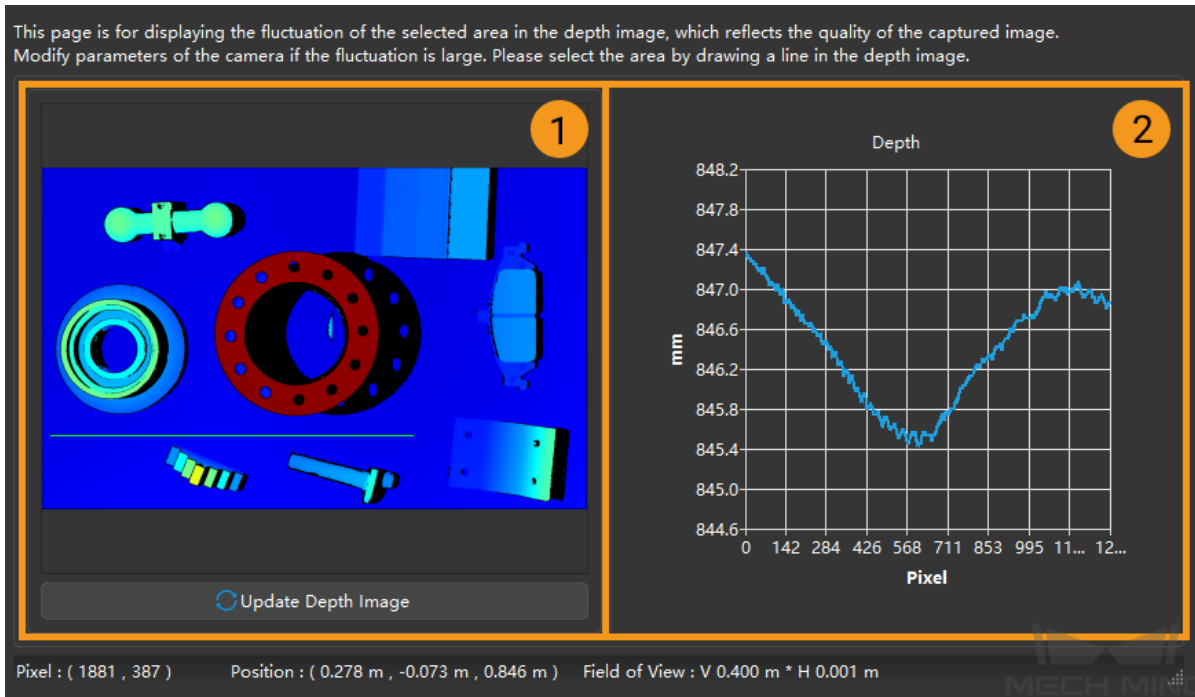
---

## 8.4 Depth Analyzer

This tool helps check the quality of the depth map. You can open this tool via *Tools* → *Depth Analyzer*.

### 8.4.1 Interface

This page is for displaying the fluctuation of the selected area in the depth image, which reflects the quality of the captured image. Modify parameters of the camera if the fluctuation is large. Please select the area by drawing a line in the depth image.



1. Depth map: display the generated depth map where you can draw a line for generating the depth plot.
2. Depth plot: display the depth variation along the line drawn on the depth map.

### 8.4.2 Instructions

1. On the depth map, locate the area whose depth variation you'd like to check. Draw a line across this area by holding the left mouse button and drag.
2. A depth plot along the line will be displayed on the right.
3. Analysis:
  - Imagine a cross-section of the scene along the line you drew.
  - The depth plot reflects the depth (height) of object top surfaces along this cross-section.
  - Pay attention to the Y-axis of the depth plot; the value range is automatically adjusted to the min and max values along the cross-section. Check if the depth fluctuation along the line is within your acceptable range.
  - If not, please *adjust the parameters for depth map* to obtain a better result.

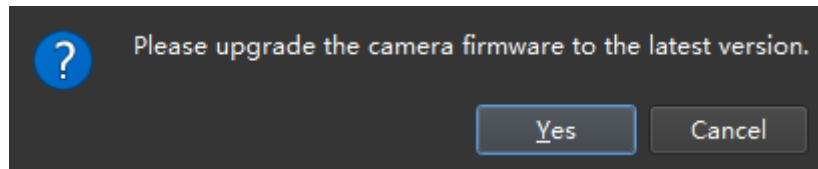


## 8.5 Camera Firmware Upgrade

The camera's firmware version must be consistent with Mech-Eye Viewer's version for the two to be connected. That is, the two versions should have the same version numbers.

### 8.5.1 Automatic Upgrade

If the firmware version is not consistent with Mech-Eye Viewer's version, the following message will pop up in Mech-Eye Viewer. Click Yes to open the **Camera Firmware Upgrade** tool.

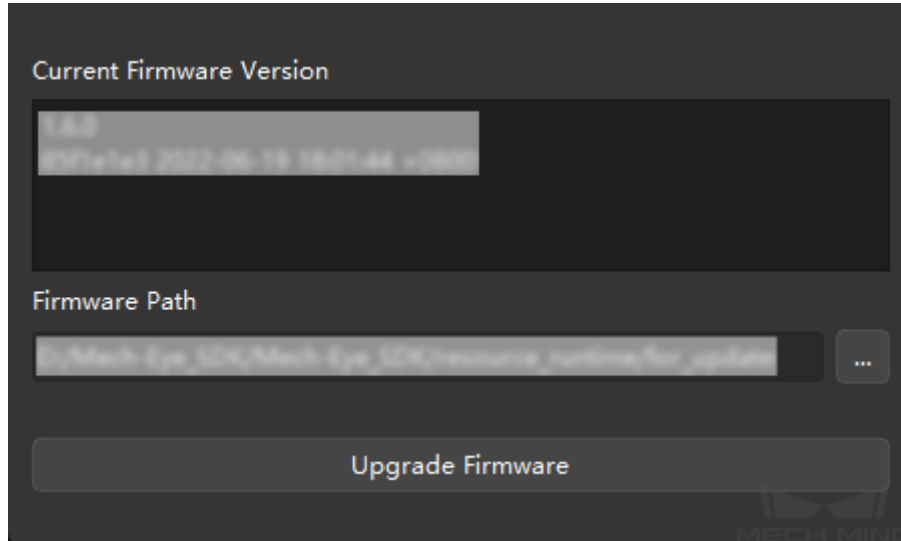


Instructions for actions within the tool are provided in the section below.

### 8.5.2 Manual Upgrade

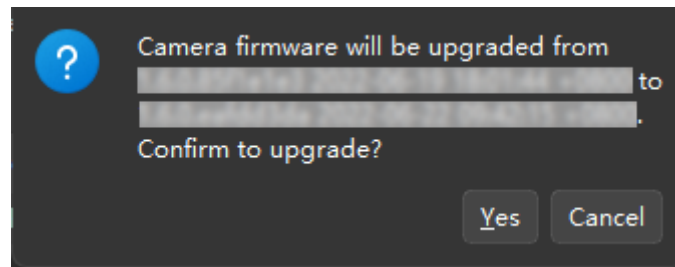
You can also manually upgrade the camera's firmware through the following steps:

1. Click *Tools* → *Camera Firmware Upgrade* in the menu bar.

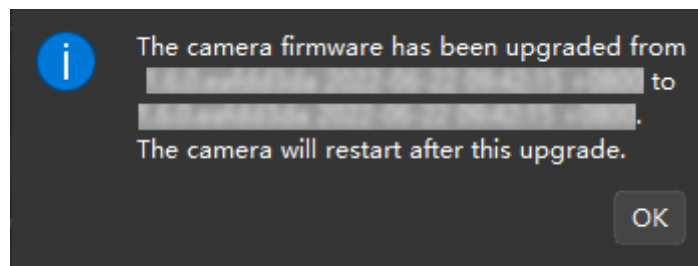


- Current Firmware Version: display the information of the camera's current firmware.
- Firmware Path: the path where the firmware compatible with the Mech-Eye Viewer version in use is stored. You can also click ... to select a different path.

2. Click *Upgrade Firmware*, and the following window will pop up. Click Yes to start the upgrade.




If the upgrade is successful, the following message will pop up. Click *OK* to restart the camera.




---

**Note:** The restart process may take several minutes, during which the camera is still available in the **Camera List**, and the firmware version appears to be not upgraded.

You can click  next to **Camera List** to update the list several times. The camera will disappear when it's restarting, and when it reappears, the firmware upgrade has been completed.

---

If the upgrade fails, please try to upgrade the firmware again. If it still fails, please contact Technical Support.

## 8.6 FOV Calculator

This tool calculates the camera's FOV on the object plane based on selected working distance. You can use this tool to help determine the mounting height of the camera based on your needs for camera FOV.

Click *Tools* → *FOV Calculator* in the menu bar.

The following table lists the default values and ranges of working distances in **FOV Calculator**:

Model	Working Distance (in m)	
	Default	Range
NANO	0.5	0.3–0.6
PRO XS	0.5	0.3–0.6
UHP-140	0.3	0.28–0.32
PRO S	1	0.5–1
PRO M	2	1–2
LSR L	2.5	1.2–3
DEEP	3	1.2–3.5
Log S	1	0.5–1
Log M	2	0.8–2
Pro S Enhanced	1	0.5–1
Pro M Enhanced	2	0.8–2
Laser L	2.5	1.5–3
Laser L Enhanced	2.5	1.5–3
Deep	3	1.2–3.5

### 8.6.1 Adjust the Working Distance

You can adjust the working distance by:

- Clicking the + and - to the right of the text field.
- Clicking the text field and then scrolling to adjust. The value changes in increments of 0.1 m.
- Inputting a value directly in the text field.

---

**Note:** The input value must be:

- Within the value range for the selected camera series.
- an multiple of 0.1.

Otherwise, the FOV values will not be calculated.

---

### 8.6.2 Calculated FOV

- **FOV Width** represents the length of the camera's FOV on the object plane.
- **FOV Height** represents the width of the camera's FOV on the object plane.

These values are automatically updated as you adjust the working distance.

---

**Note:** These values are automatically calculated based on the selected camera series and working distance and cannot be changed manually.

---

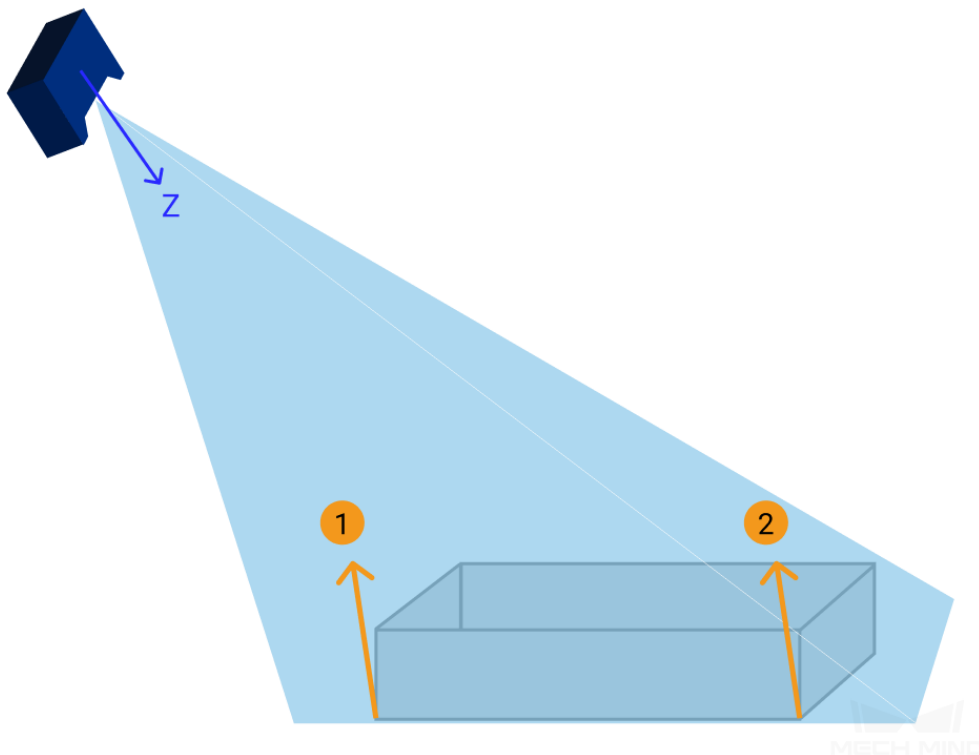
## 8.7 Custom Coordinate System

You can define a custom reference frame in **Custom Coordinate System** for displaying the depth information of depth maps and point clouds.

Click *Tools* → *Custom Coordinate System* in the menu bar to access this tool.

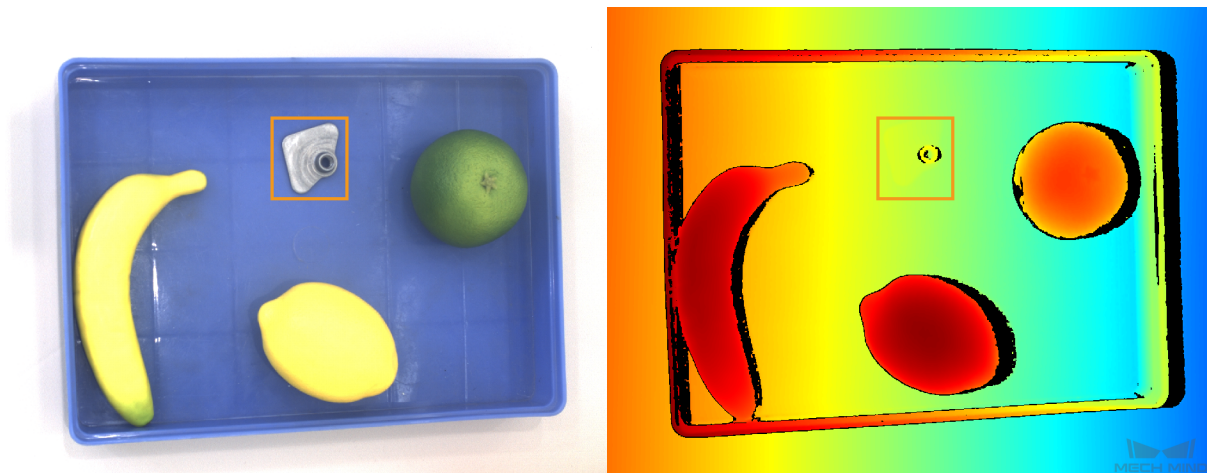
### 8.7.1 Suitable Scenarios

Under certain circumstances, the camera cannot be mounted directly above the target objects, but has to capture images at an angle, as shown below.



In such cases, the Z-axis of the camera reference frame isn't perpendicular to the object plane. Therefore, two points that appear to be at the same height in the eyes of an observer, such as points 1 and 2 in the figure above, will have different depth values in the obtained depth maps and point clouds.

When you perform image capturing with a camera mounted at an angle, the obtained 2D image and depth map will look like the following:



A surface horizontal to the observer's eyes, such as the bottom of the bin, will have varying depth from left to right. A flat, thin object, such as the one in the orange box, seemingly disappears in the depth map, because its own thickness is too small compared to the background depth variation.

To resolve this discrepancy between observed height and obtained depth data, you can define a custom reference frame in **Custom Coordinate System**. Once an appropriate custom reference frame is defined, the depth data will appear to be consistent to object heights that you see.

## 8.7.2 Define a Custom Reference Frame

### Basic Principle

In **Custom Coordinate System**, you can define a reference frame by selecting three points:

- The first selected point is the origin of your custom reference frame.
- The second and first points together define the X-axis.
- All three points together define the XoY plane.
- The third point's relative position to the X-axis determines the positive direction of the Y-axis.

The Z-axis is then determined through the right-hand rule.

---

**Hint:** To ensure that the displayed depth information is consistent with what you actually see, select the three points on a plane that appears to be horizontal to you.

---

**Instructions**

1. Open **Custom Coordinate System**, press down the Shift key while clicking three points on the displayed point cloud.

---

**Hint:** Scroll to zoom in/out, drag to rotate the point cloud, and drag with the scrolling wheel to move your point of view.

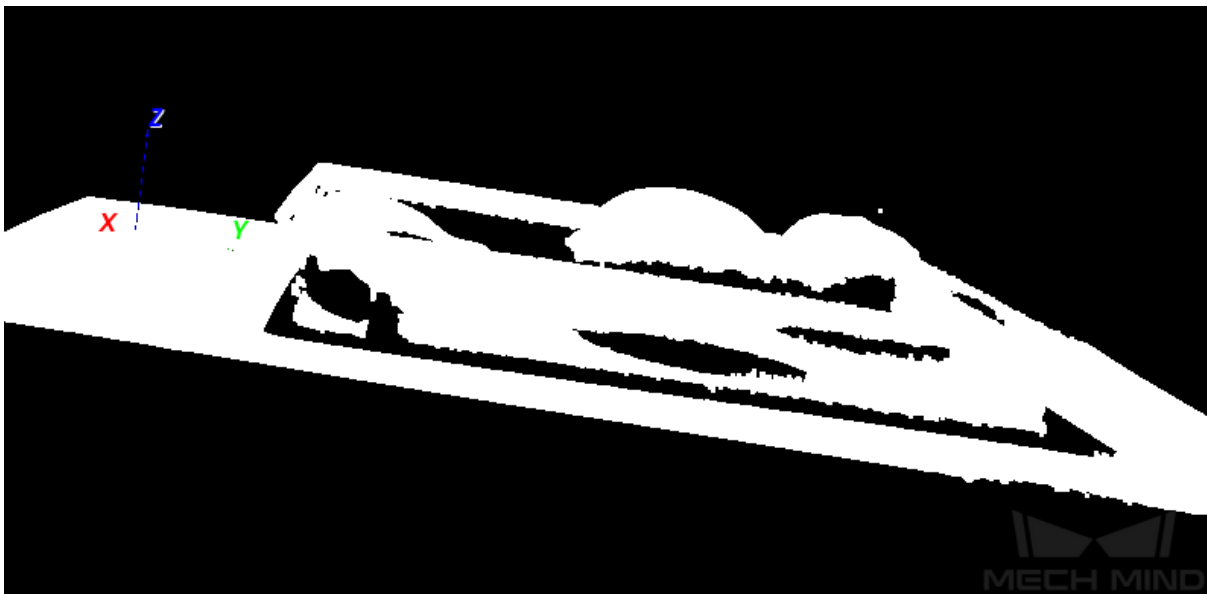
---

2. After selecting the three points, the axes of the custom reference frame will appear on the point cloud. Rotate the point cloud to check:
  - If the Z-axis is perpendicular to the plane that appears horizontal to you.
  - If the Z-axis is pointing in the direction you want. If not, choose the third point on the other side of the X-axis defined by the first two points.



If the custom reference frame doesn't meet your needs, click *Reset* to clear the reference frame and re-select the points.

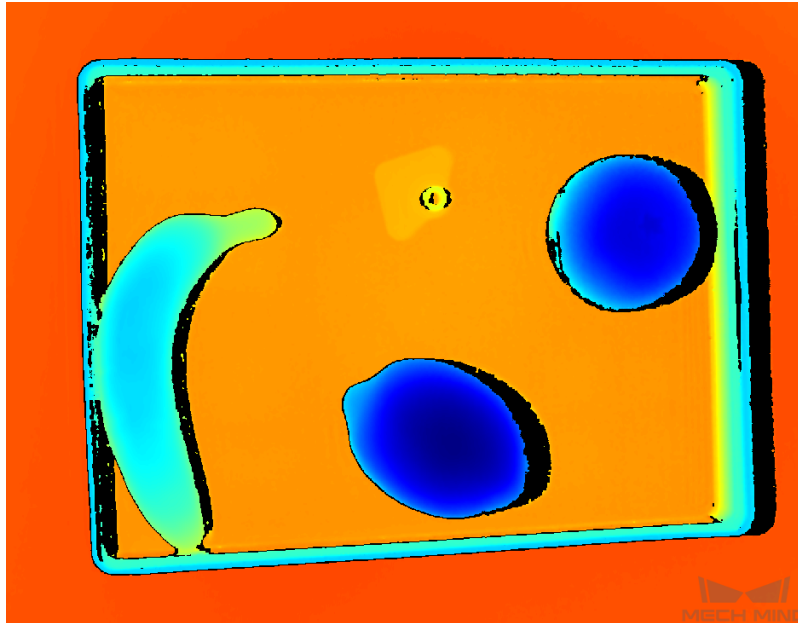
3. Click *OK* to save the custom reference frame.

For the following section, a custom reference frame is defined to have the XoY plane on the surface where the bin is placed:



### 8.7.3 Use the Custom Reference Frame

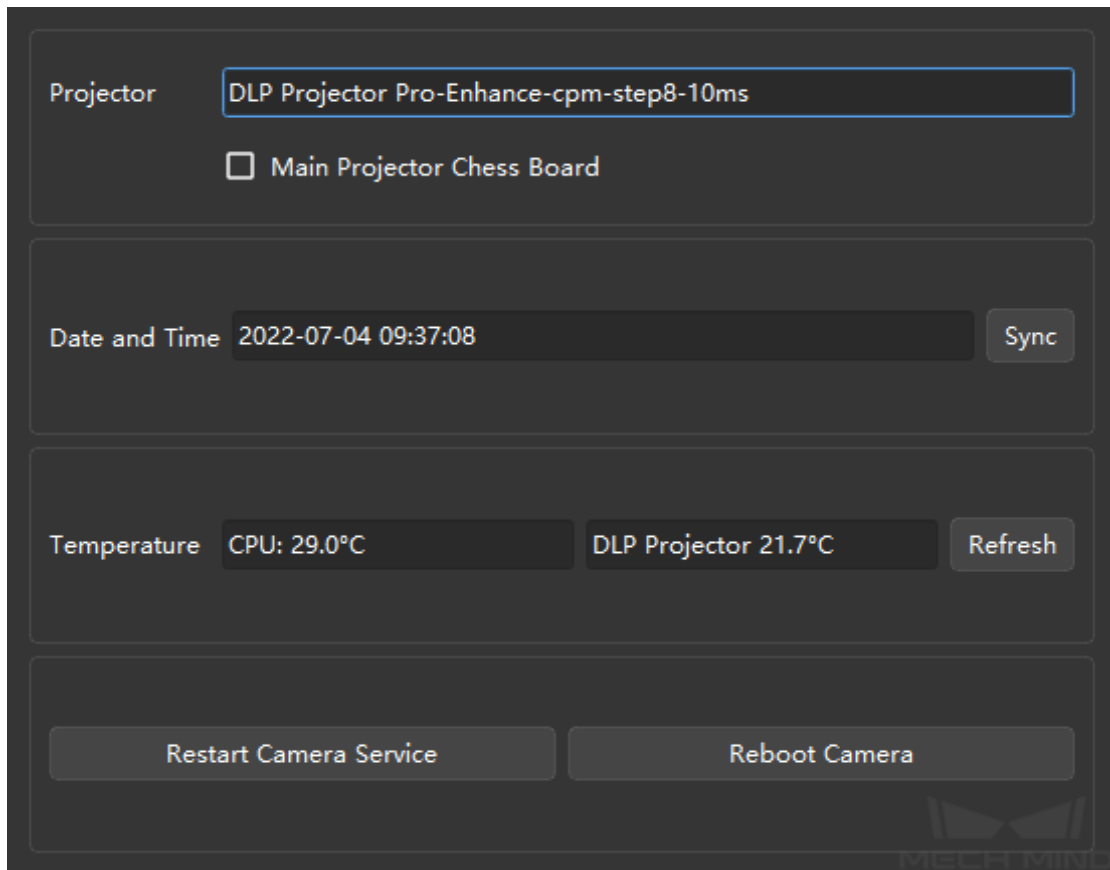
1. Once you have a custom reference frame defined, perform image capturing again with  or .
2. In the Image Display area, select **Depth Map** or **Point Cloud**. In the left panel, select **Customized** from the drop-down menu of **Coordinate System**.
3. Now the depth information is displayed in the custom reference frame: the bin bottom has the same color that represents the same depth.



## 8.8 Camera Controller

Click *Tools* → *Camera Controller* in the menu bar to access this tool.

You can view the following information of the camera in **Camera Controller**:



- **Projector:** display the projector information of the connected camera.
  - Main Projector Chess Board:** if checked, the projector will project a chessboard pattern used to check the focus of the projector.
- **Date and Time:** display the date and time recorded in the camera.
  - Click *Sync* to sync the IPC's date and time to the camera.
- **Temperature:** display the temperatures of the CPU and projector.
  - Click *Refresh* to obtain the latest temperature data.

## 8.9 Full Pallet Simulator

**Note:** This tool can only be used in projects where the camera is mounted perpendicularly above the pallet/load.

This tool is designed for simulating a fully loaded pallet to check if the mounting height of the camera is sufficient.

Before a palletizing/depalletizing project is deployed, it may be difficult to acquire enough material to actually create a fully loaded pallet. In such cases, the camera may not be mounted high enough for the



entire top layer of the load to be captured by the camera.

Using this tool, you can create a cuboid to simulate a full load, facilitating you in adjusting the camera mounting height.


**Attention:** The displayed result is for reference only.

### 8.9.1 Prerequisites

Before you start using **Full Pallet Simulator**, have the following ready at hand:

- Pallet (empty one recommended)
- Dimensions of the full load
- Any limitations on the place where the camera will be mounted (such as the maximum mounting height allowed)

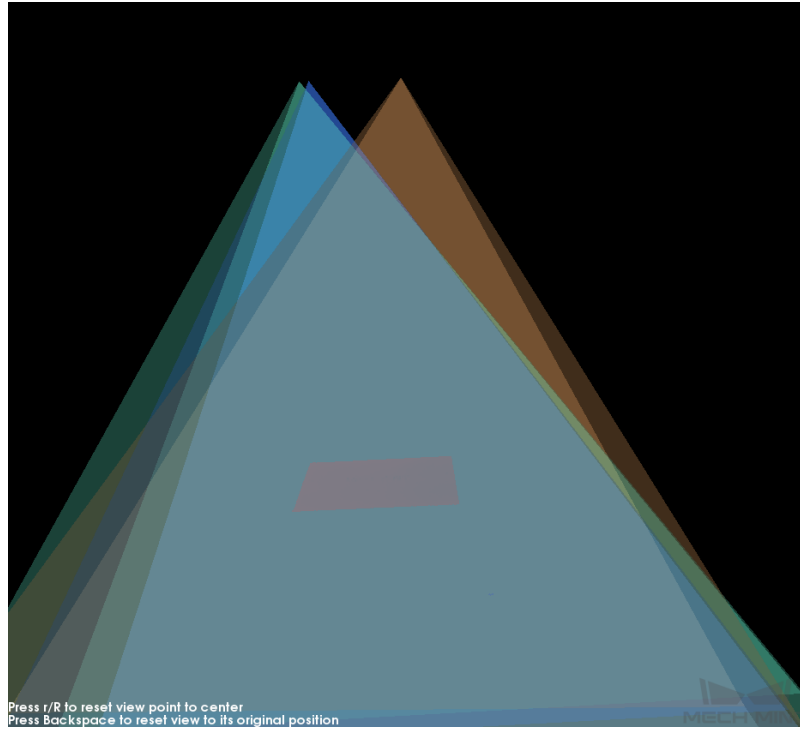
### 8.9.2 Use Full Pallet Simulator

1. Place the pallet at its designated position within the camera's FOV.
2. Perform image capturing with . Check the point cloud and make sure that the entire pallet is represented.
3. *Set an ROI* to retain only the point cloud of the pallet and exclude irrelevant points as much as possible.
4. In the menu bar, click **Tools** and then **Full Pallet Simulator**.
5. In the pop-up window, adjust the view of the point cloud until you can see the entire point cloud from the side.

---

**Hint:** A list of actions for adjusting the view can be found [here](#).

---

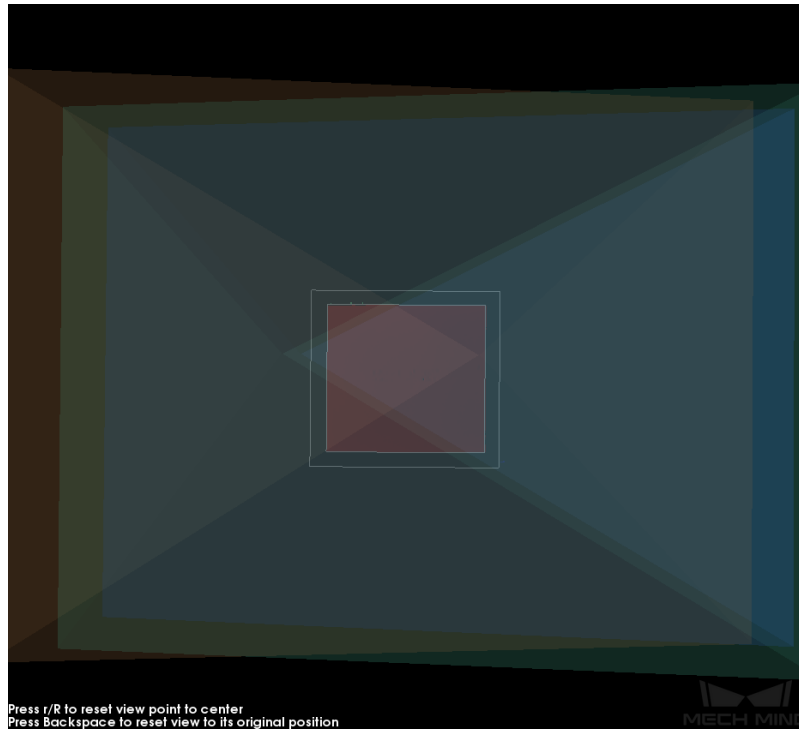


6. Click *Create cuboid* to add a cuboid that represents the full load.

---

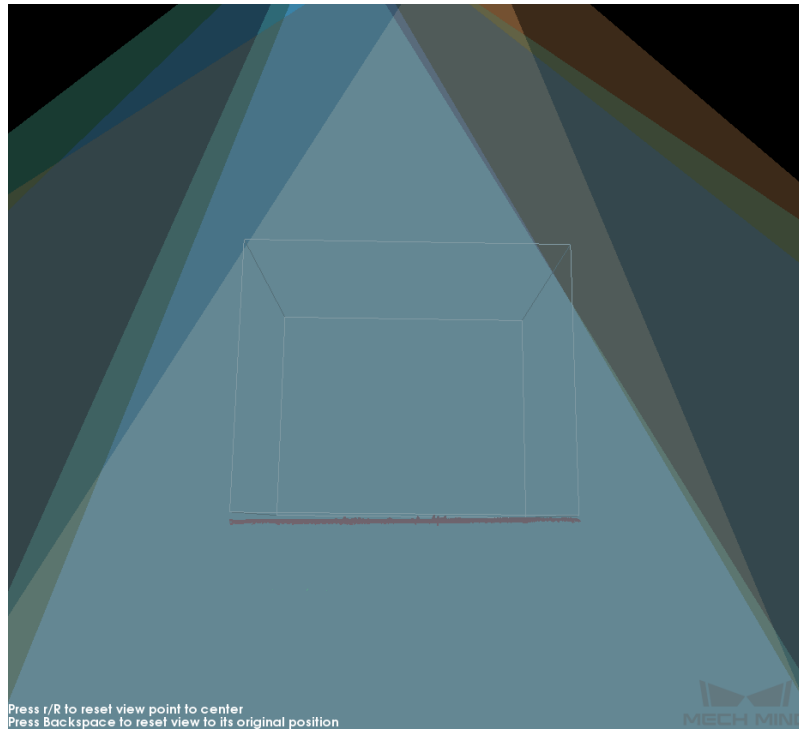
**Note:** The cuboid cannot be rotated. If the cuboid and the pallet point cloud are not aligned, please adjust the position of the pallet and perform image capturing again.

You can rotate the point cloud and view from the top to see if the cuboid is aligned with the point cloud:



7. Enter the actual dimensions of a full load in the **Cuboid Dimensions** section.
8. Adjust the position of the cuboid: As the cuboid represents the load on the pallet, its bottom should be placed on the pallet's point cloud. If it's not, use the arrow buttons in the **Center of Cuboid Bottom** section to move the cuboid.

**Note:** Record the amount of z value adjustment made.

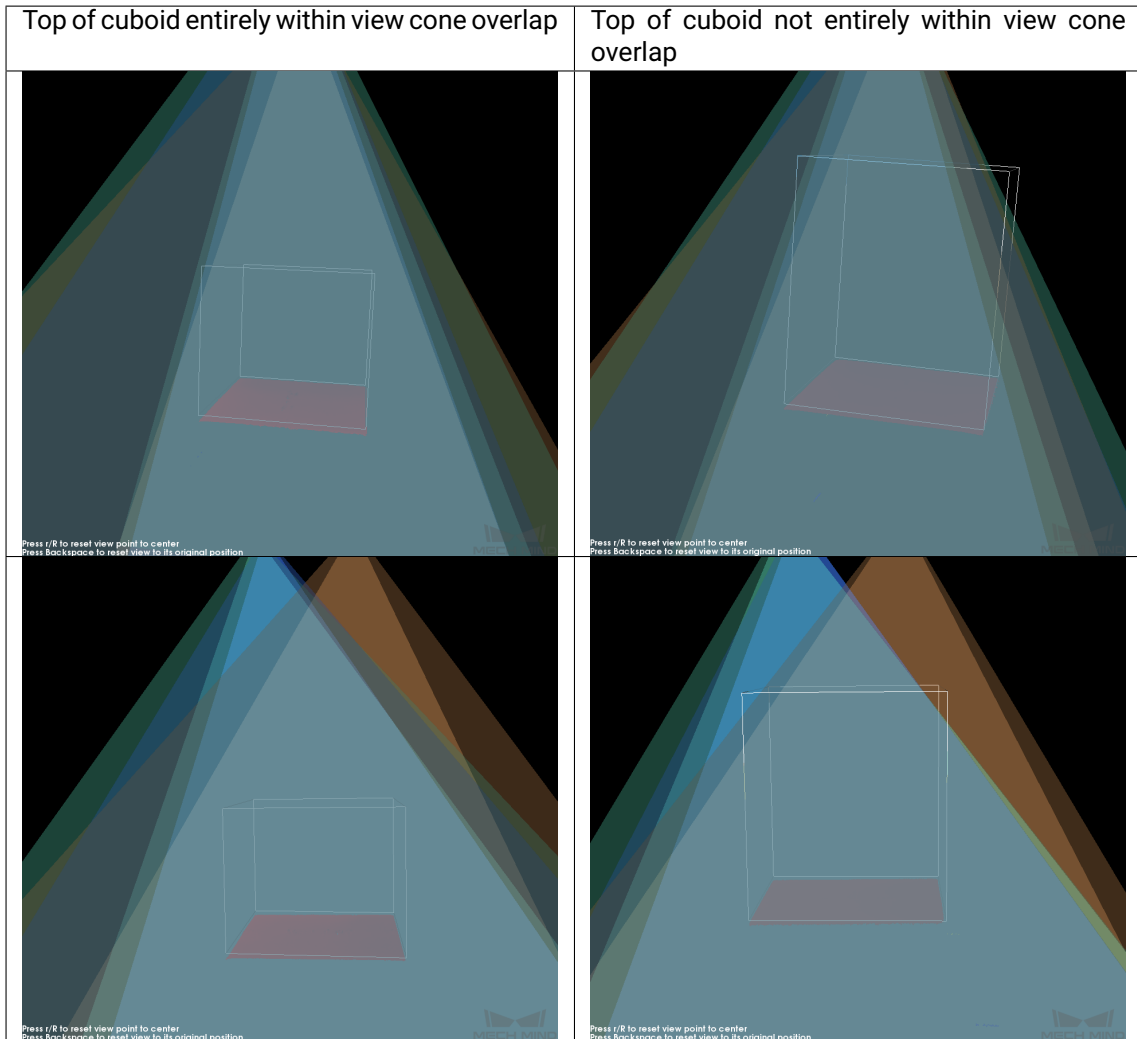


9. Check if the entire top of the cuboid is confined to the space where the view cones overlap. Use the examples below as a reference.

---

**Note:** You may need to rotate the view around to make sure if a small part of the top is not confined.

---



10. Evaluate the camera mounting height:

- If the top of the cuboid is entirely confined to the view cone overlap: The camera mounting height is appropriate, no adjustment needed.
- If the top of the cuboid is not entirely confined to the view cone overlap: The camera is mounted too low.
  1. Increase the **z** value of the **Center of Cuboid Bottom** section to move the cuboid downward, until the top of the cuboid is entirely confined to the view cone overlap.
  2. Record the amount of increase. The camera mounting height should be raised by (this amount - the amount of adjustment in step 6).

---

**Note:** Modifications made are not saved when you close this window. Therefore, make sure you note down the above amount of increase in the **z** value.

---

## MECH-EYE API QUICK FACTS

You can obtain images and point clouds from Mech-Eye Industrial 3D Cameras using Mech-Eye API.

C++ and C# Mech-Eye API are included in Mech-Eye SDK. Python Mech-Eye API can be installed through pip.

ROS API is available on GitHub.

---

How to **install Mech-Eye SDK**:

*[Install Mech-Eye SDK](#)*

---

A tutorial on **how to use Mech-Eye API to capture images and obtain data**:

*[Capture Tutorial](#)*

---

How to **install, configure, build, and run Mech-Eye API** using sample programs:

*[Samples](#)*

---

A reference guide of **Mech-Eye API Reference**:

*[Mech-Eye API Reference](#)*

---

## 9.1 Install Mech-Eye SDK

### 9.1.1 Install Mech-Eye SDK on Windows

Please see *[Mech-Eye SDK Installation Guide](#)* for detailed instructions.

## 9.1.2 Install Mech-Eye SDK on Ubuntu

Download [Mech-Eye API](#).

---

### Hint:

- When downloading, you will be prompted to log in before you can proceed. If you don't have an account yet, please sign up using an email address first.
- Command `arch` prints the system architecture.

---

If the system architecture is AMD64, please execute the following command to install:

```
sudo dpkg -i MechEyeApi_x.x.x_amd64.deb
```

If the system architecture is ARM64, please execute the following command to install:

```
sudo dpkg -i MechEyeApi_x.x.x_arm64.deb
```

---

**Note:** Please replace `x.x.x` with the version number in the file name.

---

If you need to check whether Mech-Eye SDK is installed, please enter the following command:

```
dpkg -l | grep mecheyeapi
```

If you need to uninstall Mech-Eye SDK, please enter the following command:

```
sudo dpkg -P MechEyeApi
```

## **CAPTURE TUTORIAL**

This section describes how to use Mech-Eye API to connect to a camera, set camera parameters, obtain and save the data.

**Attention:** Mech-Eye API 2.0.0 updated the method and parameter names of C++ and C#. If you have installed Mech-Eye SDK 2.0.0 but still use programs written before, please refer to the following topics and modify your program accordingly.

- [C++ Parameter Name Changes \(2.0.0\)](#)
- [C# Method and Parameter Name Changes \(2.0.0\)](#)

### **10.1 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:: FringeContrast-Threshold	PointCloudProcessingSettings:: fringeContrast-Threshold
PointCloudProcessingSet-tings::FringeMinThreshold	PointCloudProcessingSet-tings::fringeMinThreshold
PointCloudProcessingSettings:: CloudOutlierFil-terMode::OutlierFilterMode	PointCloudProcessingSettings:: CloudOutlierFil-terMode::outlierFilterMode
PointCloudProcessingSettings:: CloudSmooth-Mode::SmoothMode	PointCloudProcessingSettings:: CloudSmooth-Mode::smoothMode
LaserSettings::LaserFringeCodingMode:: Fringe-CodingMode	LaserSettings::LaserFringeCodingMode:: fringe-CodingMode
LaserSettings::FrameRangeStart	LaserSettings::frameRangeStart
LaserSettings::FrameRangeEnd	LaserSettings::frameRangeEnd
LaserSettings::FramePartitionCount	LaserSettings::framePartitionCount
LaserSettings::PowerLevel	LaserSettings::powerLevel
UhpSettings::UhpCaptureMode::CaptureMode	UhpSettings::UhpCaptureMode::captureMode
UhpSettings::UhpFringeCodingMode:: FringeCod-ingMode	UhpSettings::UhpFringeCodingMode:: fringeCod-ingMode

## 10.2 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](#)
- [Parameter Name Changes](#)

### 10.2.1 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()
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)

continues on next page

Table 1 – continued from previous page

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)
MechEyeDevice.getUhpFringeCodingMode (mmind.apiSharp.UhpFringeCodingMode)	MechEyeDevice.GetUhpFringeCodingMode (mmind.apiSharp.UhpFringeCodingMode)
MechEyeDevice.getUhpSettings (mmind.apiSharp.UhpSettings)	MechEyeDevice.GetUhpSettings (mmind.apiSharp.UhpSettings)
MechEyeDevice.saveAllSettingsToUserSets()	MechEyeDevice.SaveAllSettingsToUserSets()
MechEyeDevice.setCloudOutlierFilterMode (mmind.apiSharp.CloudOutlierFilterMode)	MechEyeDevice.SetCloudOutlierFilterMode (mmind.apiSharp.CloudOutlierFilterMode)
MechEyeDevice.setCloudSmoothMode (mmind.apiSharp.CloudSmoothMode)	MechEyeDevice.SetCloudSmoothMode (mmind.apiSharp.CloudSmoothMode)
MechEyeDevice.setCurrentUserSet(string)	MechEyeDevice.SetCurrentUserSet(string)
MechEyeDevice.setDepthRange (mmind.apiSharp.DepthRange)	MechEyeDevice.SetDepthRange (mmind.apiSharp.DepthRange)
MechEyeDevice.setFringeContrastThreshold(int)	MechEyeDevice.SetFringeContrastThreshold(int)
MechEyeDevice.setFringeMinThreshold(int)	MechEyeDevice.SetFringeMinThreshold(int)
MechEyeDevice.setLaserSettings (mmind.apiSharp.LaserSettings)	MechEyeDevice.SetLaserSettings (mmind.apiSharp.LaserSettings)
MechEyeDevice.setScan2DExpectedGrayValue (int)	MechEyeDevice.SetScan2DExpectedGrayValue (int)
MechEyeDevice.setScan2DExposureMode (mmind.apiSharp.Scan2DExposureMode)	MechEyeDevice.SetScan2DExposureMode (mmind.apiSharp.Scan2DExposureMode)
MechEyeDevice.setScan2DExposureTime(double)	MechEyeDevice.SetScan2DExposureTime(double)
MechEyeDevice.setScan2DHDRExposureSequence (System.Collections.Generic.List<double>)	MechEyeDevice.SetScan2DHDRExposureSequence (System.Collections.Generic.List<double>)

continues on next page

Table 1 – continued from previous page

MechEyeDevice.setScan2DROI (mmind.apiSharp.ROI)	MechEyeDevice.SetScan2DROI (mmind.apiSharp.ROI)
MechEyeDevice.setScan2DSharpenFactor(double)	MechEyeDevice.SetScan2DSharpenFactor(double)
MechEyeDevice.setScan2DToneMappingEnable (bool)	MechEyeDevice.SetScan2DToneMappingEnable (bool)
MechEyeDevice.setScan3DExposure (Sys- tem.Collections.Generic.List<double>)	MechEyeDevice.SetScan3DExposure (Sys- tem.Collections.Generic.List<double>)
MechEyeDevice.setScan3DGain(double)	MechEyeDevice.SetScan3DGain(double)
MechEyeDevice.setScan3DROI (mmind.apiSharp.ROI)	MechEyeDevice.SetScan3DROI (mmind.apiSharp.ROI)
MechEyeDevice.setUhpCaptureMode (mmind.apiSharp.UhpCaptureMode)	MechEyeDevice.SetUhpCaptureMode (mmind.apiSharp.UhpCaptureMode)
MechEyeDevice.setUhpFringeCodingMode (mmind.apiSharp.UhpFringeCodingMode)	MechEyeDevice.SetUhpFringeCodingMode (mmind.apiSharp.UhpFringeCodingMode)
MechEyeDevice.setUhpSettings (mmind.apiSharp.UhpSettings)	MechEyeDevice.SetUhpSettings (mmind.apiSharp.UhpSettings)
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()

## 10.2.2 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
Scanning3DSettings.DepthRange	Scanning3DSettings.depthRange
Scanning3DSettings.ExposureSequence	Scanning3DSettings.exposureSequence
Scanning3DSettings.Gain	Scanning3DSettings.gain
Scanning3DSettings.Scan3DROI	Scanning3DSettings.scan3DROI
UhpSettings.CaptureMode	UhpSettings.captureMode
UhpSettings.FringeCodingMode	UhpSettings.fringeCodingMode

## 10.3 Requirements

For C++, C# and Python, please *install Mech-Eye SDK* first.

Also for Python, please *complete the relevant configurations*.

## 10.4 Detect Cameras

Once the requirements are satisfied, you can detect the connectable cameras using the following commands.

Detect Cameras (C++)

```
std::vector<mmind::api::MechEyeDeviceInfo> deviceInfoList =
↳ mmind::api::MechEyeDevice::enumerateMechEyeDeviceList();
```

Detect Cameras (C#)

```
List<MechEyeDeviceInfo> deviceInfoList = MechEyeDevice.EnumerateMechEyeDeviceList();
```

Detect Cameras (Python)

```
self.device_list = self.device.get_device_list()
```

## 10.5 Connect to a Camera

After finding the connectable cameras, you can connect to a camera using the following commands (Example codes connecting to the first camera in the list).

Connect to a Camera (C++)

```
mmind::api::MechEyeDevice device;  
device.connect(deviceInfoList[0]);
```

Connect to a Camera (C#)

```
MechEyeDevice device = new MechEyeDevice();  
device.Connect(deviceInfoList[0]);
```

Connect to a Camera (Python)

```
self.device.connect(self.device_list[int(0)])
```

## 10.6 Set the Camera Parameters

After connecting to the camera, you can set the camera parameters using the following commands.

### 10.6.1 Set Parameters for 2D Image

Before capturing 2D images, you need to set the 2D Parameters, including Exposure Mode, Exposure Time, etc.

Set Parameters for 2D Image (C++)

```
device.  
->setScan2DExposureMode(mmind::api::Scanning2DSettings::Scan2DExposureMode::Timed);  
device.setScan2DExposureTime(100);
```

Set Parameters for 2D Image (C#)

```
device.SetScan2DExposureMode(Scan2DExposureMode.Timed);  
device.SetScan2DExposureTime(100);
```

Set Parameters for 2D Image (Python)

```
self.device.set_scan_2d_exposure_mode("Timed")
self.device.set_scan_2d_exposure_time(100.0)
```

## 10.6.2 Set Parameters for Depth Map and Point Cloud

Before capturing images used for calculating depth data, you need to set the parameters that affect the quality of the depth map and point cloud, including Exposure Time, Depth Range, ROI, Point Cloud Processing, etc.

Set Parameters for Depth Map and Point Cloud (C++)

```
device.setScan3DExposure(std::vector<double>{5, 10});
device.setDepthRange(mmind::api::DepthRange(100, 1000));
device.setScan3DROI(mmind::api::ROI(0, 0, 500, 500));
device.setCloudSmoothMode(
    mmind::api::PointCloudProcessingSettings::CloudSmoothMode::Normal);
device.setCloudOutlierFilterMode(
    mmind::api::PointCloudProcessingSettings::CloudOutlierFilterMode::Normal);
```

Set Parameters for Depth Map and Point Cloud (C#)

```
device.SetScan3DExposure(new List<double> {5, 10});
device.SetDepthRange(new DepthRange(100, 1000));
device.SetScan3DROI(new ROI(0, 0, 500, 500));
device.SetCloudSmoothMode(CloudSmoothMode.Normal);
device.SetCloudOutlierFilterMode(CloudOutlierFilterMode.Normal);
```

Set Parameters for Depth Map and Point Cloud (Python)

```
self.device.set_scan_3d_exposure([5.0, 10.0])
self.device.set_depth_range(100, 1000)
self.device.set_scan_3d_roi(0, 0, 500, 500)
self.device.set_cloud_smooth_mode("Normal")
self.device.set_cloud_outlier_filter_mode("Normal")
```

## 10.7 Obtain Data

After setting the parameters, you can trigger the camera to capture images and return the 2D image and point cloud data.

### 10.7.1 Obtain 2D Image Data

Obtain 2D Image Data (C++)

```
mmind::api::ColorMap color;
device.captureColorMap(color);
```

Obtain 2D Image Data (C#)

```
ColorMap color = new ColorMap();
device.CaptureColorMap(ref color);
```

Obtain 2D Image Data (Python)

```
color_map = device.capture_color()
```

### 10.7.2 Obtain Point Cloud Data

Obtain Point Cloud Data (C++)

```
mmind::api::PointXYZMap pointXYZMap;
device.capturePointXYZMap(pointXYZMap);
```

Obtain Point Cloud Data (C#)

```
PointXYZMap pointXYZMap = new PointXYZMap();
device.CapturePointXYZMap(ref pointXYZMap);
```

Obtain Point Cloud Data (Python)

```
point_xyz = device.capture_point_xyz()
```

## 10.8 Save Data

### 10.8.1 Save 2D Image

You can convert the obtained 2D image data to OpenCV format and save the 2D image as a PNG file.

Save 2D Image (C++)



```
const std::string colorFile = "ColorMap.png";
cv::Mat color8UC3 = cv::Mat(colorMap.height(), colorMap.width(), CV_8UC3, colorMap.
↪data());
cv::imwrite(colorFile, color8UC3);
```

Save 2D Image (C#)

```
string colorFile = "colorMap.png";
Mat color8UC3 = new Mat(unchecked((int)colorMap.height()), unchecked((int)colorMap.
↪width()), DepthType.Cv8U, 3, colorMap.data(), unchecked((int)colorMap.width()) * 3,
↪3);
CvInvoke.Imwrite(colorFile, color8UC3);
```

Save 2D Image (Python)

```
color_file = "ColorMap.png"
cv2.imwrite(color_file, color_map.data())
```

## 10.8.2 Save Point Cloud

You can save the point cloud data as a PLY file.

Save Point Cloud (C++)

```
std::string pointCloudPath = "PointCloudXYZ.ply";
savePLY(pointXYZMap, pointCloudPath);
```

Save Point Cloud (C#)

```
Mat depth32FC3 = new Mat(unchecked((int)pointXYZMap.height()),
↪unchecked((int)pointXYZMap.width()), DepthType.Cv32F, 3, pointXYZMap.data(),
↪unchecked((int)pointXYZMap.width()) * 12);
string pointCloudPath = "pointCloudXYZ.ply";
CvInvoke.WriteCloud(pointCloudPath, depth32FC3);
```

Save Point Cloud (Python)

```
point_cloud_xyz = o3d.geometry.PointCloud()
points_xyz = np.zeros(
    (point_xyz.width() * point_xyz.height(), 3), dtype=np.float64)

pos = 0
for dd in np.nditer(point_xyz_data):
    points_xyz[int(pos / 3)][int(pos % 3)] = 0.001 * dd
    pos = pos + 1

point_cloud_xyz.points = o3d.utility.Vector3dVector(points_xyz)
o3d.io.write_point_cloud("PointCloudXYZ.ply", point_cloud_xyz)
```

## SAMPLES

C++ samples are provided in Mech-Eye SDK. Python, and C# samples are available on GitHub.

Dive straight into programming Mech-Eye Industrial 3D Camera with your preferred programming language.

### 11.1 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*
- *Prerequisites*
- *Build and Run Samples*

#### 11.1.1 Sample List

Samples are divided into six categories, **Basic**, **Advanced**, **Util**, **Laser**, **UHP** and **Halcon**.

- **Basic**: camera connection and basic capturing functions.
- **Advanced**: advanced capturing functions.
- **Util**: obtain information from a camera and set camera parameters.
- **Laser**: for Laser, LSR and DEEP series cameras only.
- **UHP**: for UHP series cameras only.
- **Halcon**: obtain HALCON-readable point clouds via Mech-Eye API.

##### Basic

- **ConnectToCamera**: Connect to a Mech-Eye Industrial 3D Camera.
- **ConnectAndCaptureImage**: Connect to a camera and obtain the 2D image, depth map and point cloud data.
- **CaptureColorMap**: Obtain and save the 2D image from a camera.

- **CaptureDepthMap**: Obtain and save the depth map from a camera.
- **CapturePointCloud**: Obtain and save untextured and textured point clouds generated from images captured with a single exposure time.
- **CaptureHDRPointCloud**: Obtain and save untextured and textured point clouds generated from images captured with multiple exposure times.
- **CapturePointCloudROI**: Obtain and save untextured and textured point clouds of the objects in the ROI from a camera.
- **CapturePointCloudFromTextureMask**: Construct and save untextured and textured point clouds generated from a depth map and masked 2D image.

#### Advanced

- **CaptureCloudFromDepth**: Construct and save point clouds from the depth map and 2D image obtained from a camera.
- **CaptureSequentiallyMultiCamera**: Obtain and save 2D images, depth maps and point clouds sequentially from multiple cameras.
- **CaptureSimultaneouslyMultiCamera**: Obtain and save 2D images, depth maps and point clouds simultaneously from multiple cameras.
- **CaptureTimedAndPeriodically**: Obtain and save 2D images, depth maps and point clouds periodically for the specified duration from a camera.

#### Util

- **GetCameraIntri**: Get and print a camera's intrinsic parameters.
- **PrintDeviceInfo**: Get and print a camera's information such as model, serial number and firmware version.
- **SetDepthRange**: Set the range of depth values to be retained by a camera.
- **SetParameters**: Set specified parameters to a camera.
- **SetUserSets**: Perform functions related to parameter groups, such as switching to a different parameter group, adding and deleting parameter groups, etc. The parameter group feature allows user to save and quickly apply a set of parameter values.

#### Laser

- **SetLaserFramePartitionCount**: Divide the projector FOV into partitions and project structured light in one partition at a time. The output of the entire FOV is composed from images of all partitions.
- **SetLaserFrameRange**: Set the projection range of the structured light. The entire projector FOV is from 0 to 100.
- **SetLaserFringeCodingMode**: Set the coding mode of the structured light pattern.
- **SetLaserPowerLevel**: Set the output power of the laser projector in percentage of max power. This affects the intensity of the laser light.

#### UHP

- **SetUHPCaptureMode**: Set the capture mode (capture images with camera 1, with camera 2, or with both 2D cameras and compose the outputs).
- **SetUHPFringeCodingMode**: Set the coding mode of the structured light pattern.

#### Halcon

- [CaptureHalconPointCloud](#): Obtain point cloud data from a camera, and then transform and save the point clouds using HALCON C++ interface.

### 11.1.2 Prerequisites

In order to use the C++ samples of Mech-Eye API, the following prerequisites must be satisfied.

- *Connect the camera and IPC correctly.*
- Install the required software.
- Install the optional software on which samples depend and add relevant environment variables.

For instructions on installing the required and optional software, please refer to the following sections.

- [Install Required Software](#)
- [Install Optional Software](#)
- [Add Environment Variables](#)

#### 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:

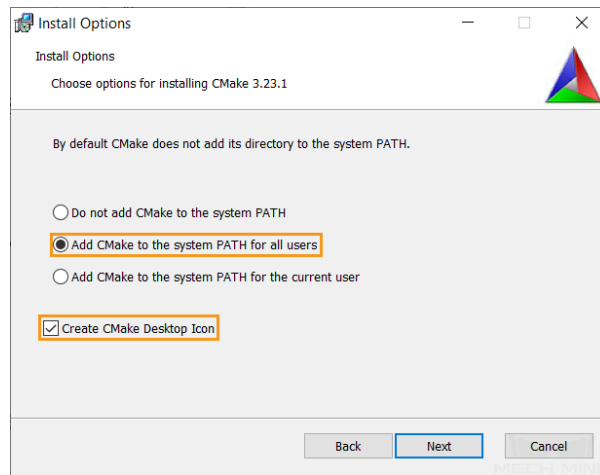
Platform	Files
Unix/Linux Source (has \n line feeds)	<a href="#">cmake-3.23.1.tar.gz</a>
Windows Source (has \r\n line feeds)	<a href="#">cmake-3.23.1.zip</a>

Binary distributions:

Platform	Files
Windows x64 Installer	<a href="#">cmake-3.23.1-windows-x86_64.msi</a>
Windows x64 ZIP	<a href="#">cmake-3.23.1-windows-x86_64.zip</a>
Windows i386 Installer	<a href="#">cmake-3.23.1-windows-i386.msi</a>
Windows i386 ZIP	<a href="#">cmake-3.23.1-windows-i386.zip</a>

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 2015 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

---

**Hint:** Please restart your computer after installing Visual Studio. If you still need to install the optional software, you can restart your computer after *adding environment variables*.

---

### Install Optional Software

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 on which the samples depend.

- OpenCV: version 3.4.5 or above
- PCL: version 1.12.1

---

**Note:** PCL is not supported in Visual Studio 2015.

---

- HALCON: version 20.11 or above

**Attention:** If any of the optional software 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.

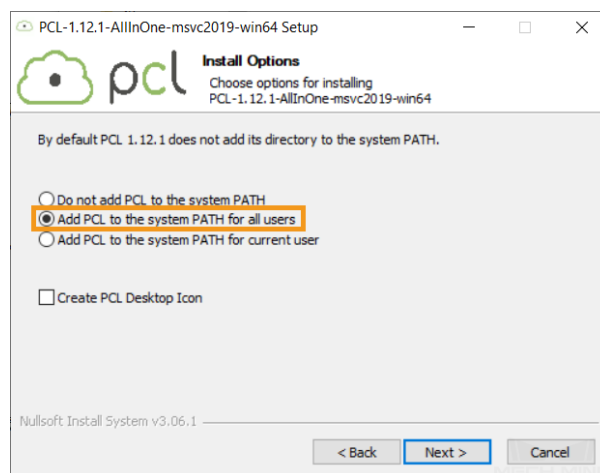
Sample	OpenCV	PCL	HALCON
CaptureColorMap	✓		
CaptureDepthMap	✓		
CapturePointCloud		✓	
CaptureHDRPointCloud		✓	
CapturePointCloudROI		✓	
CapturePointCloudFromTextureMask		✓	
CaptureCloudFromDepth		✓	
CaptureTimedAndPeriodically	✓	✓	
CaptureSimultaneouslyMultiCamera	✓	✓	
CaptureSequentiallyMultiCamera	✓	✓	
CaptureHalconPointCloud			✓

### Install OpenCV

1. [Download and install OpenCV](#). Note down the installation directory of OpenCV.
2. Add OpenCV to environment variables. Refer to [Add Environment Variables](#) for instructions.

### Install PCL

1. Open the [PCL download page](#), click on **Assets** under **PCL 1.12.1**, and download **PCL-1.12.1-AllInOne-msvc2019-win64.exe**.
2. Run **PCL-1.12.1-AllInOne-msvc2019-win64.exe** to install PCL. When installing, select **Add PCL to the system PATH for all users** to add PCL to environment variables.




---

**Hint:** 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.

---

3. 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 click *Environment Variables* in the popped up **System Properties** 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 directories 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*
  - For OpenCV: *xxx\opencv\build\x64\vc14\lib*

If you encountered the error message while installing PCL, please add the following directories as well:

- *xxx\PCL 1.12.1\bin*
- *xxx\PCL 1.12.1\3rdParty\VTK\bin*

---

**Hint:** Please restart your computer after adding the environment variables. If not restarted, the environmental variables added to PATH may not be effective.

---

### 11.1.3 Build and Run Samples

You can build all samples at once or build a single sample individually.

- *Disable Unneeded Samples (Optional)*
- *Configure Samples in CMake*
- *Build Samples in Visual Studio*
- *Run a Sample*

### Disable Unneeded Samples (Optional)

**Attention:** If you did not install some of the optional software 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-x.x.x\API\samples`.
2. Edit lines containing **options**: find the lines starting **options**, change **ON** at the end of the line to **OFF** to disable the samples that depend on the software 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 line starting with **option(USE\_HALCON** 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 directories of source code and binaries.
  - To build all samples at once, enter the following directories.

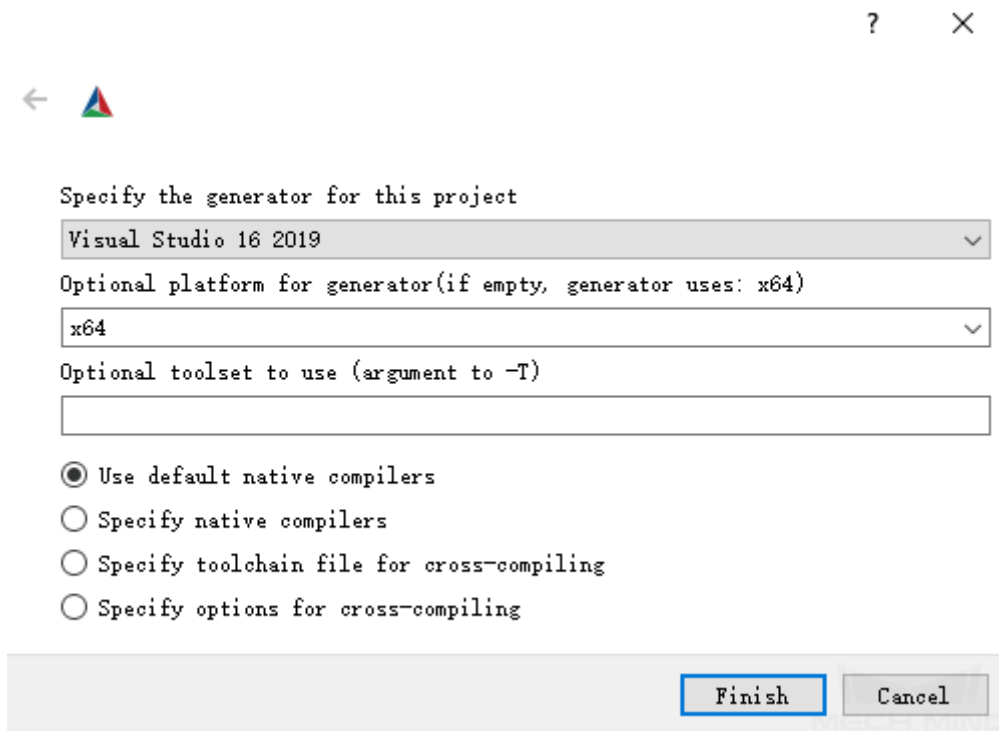
Where is the source code	xxx\Mech-Eye SDK-x.x.x\API\samples
Where to build the binaries	xxx\Mech-Eye SDK-x.x.x\API\samples\build

- To build one sample individually, enter the following directories. Replace **Category** with the category name of the sample and **SampleName** with the name of the sample.

Where is the source code	xxx\Mech-Eye SDK-x.x.x\API\samples\Category\SampleName
Where to build the binaries	xxx\Mech-Eye SDK-x.x.x\API\samples\Category\SampleName\build

3. Click *Configure* in the bottom, and select Visual Studio version, and then click *Finish*. When the configuration completes, the log will display **Configuring done**.





4. 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

1. In Visual Studio toolbar, change the solution configuration from **Debug** to **Release**.
2. In **Solution Explorer**, right-click **Solution 'xxx'** at the top, and select **Build Solution**. An EXE format executable file 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 a Sample

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 **Solution Explorer**, 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 device index of the camera to which you want to connect according to the instruction, and press **Enter** to connect.

3. 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

1. Navigate to the **Where to build the binaries** directory that you entered in CMake, and open the **Release** folder.
2. Run the EXE file with the same name as the sample.  
While the sample is running, input the device index of the camera to which you want to connect according to the instruction, and press **Enter** to connect.
3. If image and/or point cloud files are saved by the sample, you can find the files in the **Release** folder.

## 11.2 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](#)
- [Prerequisites](#)
- [Build and Run Samples](#)

### 11.2.1 Sample List

Samples are divided into six categories, **Basic**, **Advanced**, **Util**, **Laser**, **UHP** and **Halcon**.

- **Basic**: camera connection and basic capturing functions.
- **Advanced**: advanced capturing functions.
- **Util**: obtain information from a camera and set camera parameters.
- **Laser**: for Laser, LSR and DEEP series cameras only.
- **UHP**: for UHP series cameras only.
- **Halcon**: obtain HALCON-readable point clouds via Mech-Eye API.

#### Basic

- **ConnectToCamera**: Connect to a Mech-Eye Industrial 3D Camera.
- **ConnectAndCaptureImage**: Connect to a camera and obtain the 2D image, depth map and point cloud data.
- **CaptureColorMap**: Obtain and save the 2D image from a camera.
- **CaptureDepthMap**: Obtain and save the depth map from a camera.
- **CapturePointCloud**: Obtain and save untextured and textured point clouds generated from images captured with a single exposure time.
- **CaptureHDRPointCloud**: Obtain and save untextured and textured point clouds generated from images captured with multiple exposure times.

- `CapturePointCloudROI`: Obtain and save untextured and textured point clouds of the objects in the ROI from a camera.
- `CapturePointCloudFromTextureMask`: Construct and save untextured and textured point clouds generated from a depth map and masked 2D image.

#### Advanced

- `CaptureCloudFromDepth`: Construct and save point clouds from the depth map and 2D image obtained from a camera.
- `CaptureSequentiallyMultiCamera`: Obtain and save 2D images, depth maps and point clouds sequentially from multiple cameras.
- `CaptureSimultaneouslyMultiCamera`: Obtain and save 2D images, depth maps and point clouds simultaneously from multiple cameras.
- `CaptureTimedAndPeriodically`: Obtain and save 2D images, depth maps and point clouds periodically for the specified duration from a camera.

#### Util

- `GetCameraIntri`: Get and print a camera's intrinsic parameters.
- `PrintDeviceInfo`: Get and print a camera's information such as model, serial number and firmware version.
- `SetDepthRange`: Set the range of depth values to be retained by a camera.
- `SetParameters`: Set specified parameters to a camera.
- `SetUserSets`: Perform functions related to parameter groups, such as switching to a different parameter group, adding and deleting parameter groups, etc. The parameter group feature allows user to save and quickly apply a set of parameter values.

#### Laser

- `SetLaserFramePartitionCount`: Divide the projector FOV into partitions and project structured light in one partition at a time. The output of the entire FOV is composed from images of all partitions.
- `SetLaserFrameRange`: Set the projection range of the structured light. The entire projector FOV is from 0 to 100.
- `SetLaserFringeCodingMode`: Set the coding mode of the structured light pattern.
- `SetLaserPowerLevel`: Set the output power of the laser projector in percentage of max power. This affects the intensity of the laser light.

#### UHP

- `SetUHPCaptureMode`: Set the capture mode (capture images with camera 1, with camera 2, or with both 2D cameras and compose the outputs).
- `SetUHPFringeCodingMode`: Set the coding mode of the structured light pattern.

#### Halcon

- `CaptureHalconPointCloud`: Obtain point cloud data from a camera, and then transform and save the point clouds using HALCON C++ interface.

## 11.2.2 Prerequisites

In order to use the C++ samples of Mech-Eye API, the following prerequisites must be satisfied.

- *Connect the camera and IPC correctly.*
- The version of the Ubuntu operating system is 18 or above.
- Install the required software.
- Install the optional software on which samples depend.

**Attention:** 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.

For instructions on installing the required and optional software, please refer to the following sections.

- *Install Required Software*
- *Install Optional Software*

### 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 according to *Install Mech-Eye SDK on Ubuntu*.

Once Mech-Eye SDK is successfully installed, the Mech-Eye API C++ samples are located in `/opt/mech-mind/mech-eye-sdk/samples/`.

### Install Optional Software

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 on which the samples depend.

- OpenCV: latest version
- PCL: latest version
- HALCON: version 20.11 or above

**Attention:** If any of the optional software 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
CaptureColorMap	√		
CaptureDepthMap	√		
CapturePointCloud		√	
CaptureHDRPointCloud		√	
CapturePointCloudROI		√	
CapturePointCloudFromTextureMask		√	
CaptureCloudFromDepth		√	
CaptureTimedAndPeriodically	√	√	
CaptureSimultaneouslyMultiCamera	√	√	
CaptureSequentiallyMultiCamera	√	√	
CaptureHalconPointCloud		√	√

- Install 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.

```
sudo apt-get install libpcl-dev
```

- Install HALCON and add environmental variables.

1. After downloading HALCON installer, use the following command to install HALCON (taking version 20.11 as an example):

```
tar zxvf halcon-20.11.3.0-linux.tar.gz
sudo sh install-linux.sh #Note down the installation directory of HALCON.
```

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; export HALCONARCH
HALCONROOT="/opt/halcon"; export HALCONROOT
HALCONEXAMPLES=${HALCONROOT}/examples; export HALCONEXAMPLES
HALCONIMAGES=${HALCONROOT}/examples/images; export HALCONIMAGES
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
```

(continues on next page)

(continued from previous page)

```
else
    LD_LIBRARY_PATH=${HALCONROOT}/lib/${HALCONARCH}; export LD_LIBRARY_PATH
fi
```

---

**Note:**

- The changes are applied when you log in again. Or, you can `source /etc/profile/` before you configure and build the samples.
  - For more information, please refer to HALCON's installation guide.
- 

### 11.2.3 Build and Run Samples

You can build all samples at once or build a single sample individually.

- *Disable Unneeded Samples (Optional)*
- *Build Samples*
- *Run Samples*

#### Disable Unneeded Samples (Optional)

**Attention:** If you did not install some of the optional software 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-x.x.x\API\samples` 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 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 line starting with **option(USE\_HALCON** to **OFF**.

3. Save the changes in the editor and close the CMakeLists file.

## Build Samples

1. Enter the directory of the sample(s).

- If building all samples at once, please use the following command.

```
cd /opt/mech-mind/mech-eye-sdk/samples/
```

- If building a single sample individually, please 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/Category/SampleName/
```

2. Create a **build** folder in the directory for saving the built samples and so on.

```
mkdir build && cd build
```

3. Configure and build the sample(s).

```
cmake ..  
make
```

---

**Tip:** 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

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 building all samples at once, please use the following command.

```
cd /opt/mech-mind/mech-eye-sdk/samples/build/
```

- If building a single sample individually, please 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/Category/SampleName/build/
```

2. Run the sample: replace **SampleName** with the name of the sample. While the sample is running, input the device index of the camera to which you want to connect according to the instruction, and press **Enter** to connect.

```
./SampleName
```

3. If image and/or point cloud files are saved by the sample, you can find the files in the **build** folder.

## 11.3 C#

This section will show you how to run a C# sample in Windows.

### 11.3.1 Sample List

Samples are divided into five categories, **Basic**, **Advanced**, **Util**, **Laser** and **UHP**.

- **Basic**: camera connection and basic capturing functions.
- **Advanced**: advanced capturing functions.
- **Util**: obtain information from a camera and set camera parameters.
- **Laser**: for Laser, LSR and DEEP series cameras only.
- **UHP**: for UHP series cameras only.

#### Basic

- **ConnectToCamera**: Connect to a Mech-Eye Industrial 3D Camera.
- **ConnectAndCaptureImage**: Connect to a camera and obtain the 2D image, depth map and point cloud data.
- **CaptureColorMap**: Obtain and save the 2D image from a camera.
- **CaptureDepthMap**: Obtain and save the depth map from a camera.
- **CapturePointCloud**: Obtain and save untextured and textured point clouds generated from images captured with a single exposure time.
- **CaptureHDRPointCloud**: Obtain and save untextured and textured point clouds generated from images captured with multiple exposure times.
- **CapturePointCloudROI**: Obtain and save untextured and textured point clouds of the objects in the ROI from a camera.
- **CapturePointCloudFromTextureMask**: Construct and save untextured and textured point clouds generated from a depth map and masked 2D image.

#### Advanced

- **CaptureCloudFromDepth**: Construct and save point clouds from the depth map and 2D image obtained from a camera.
- **CaptureSequentiallyMultiCamera**: Obtain and save 2D images, depth maps and point clouds sequentially from multiple cameras.
- **CaptureSimultaneouslyMultiCamera**: Obtain and save 2D images, depth maps and point clouds simultaneously from multiple cameras.
- **CaptureTimedAndPeriodically**: Obtain and save 2D images, depth maps and point clouds periodically for the specified duration from a camera.

#### Util

- **GetCameraIntri**: Get and print a camera's intrinsic parameters.
- **PrintDeviceInfo**: Get and print a camera's information such as model, serial number and firmware version.



- **SetDepthRange**: Set the range of depth values to be retained by a camera.
- **SetParameters**: Set specified parameters to a camera.
- **SetUserSets**: Perform functions related to parameter groups, such as switching to a different parameter group, adding and deleting parameter groups, etc. The parameter group feature allows user to save and quickly apply a set of parameter values.

#### Laser

- **SetLaserFramePartitionCount**: Divide the projector FOV into partitions and project structured light in one partition at a time. The output of the entire FOV is composed from images of all partitions.
- **SetLaserFrameRange**: Set the projection range of the structured light. The entire projector FOV is from 0 to 100.
- **SetLaserFringeCodingMode**: Set the coding mode of the structured light pattern.
- **SetLaserPowerLevel**: Set the output power of the laser projector in percentage of max power. This affects the intensity of the laser light.

#### UHP

- **SetUHPCaptureMode**: Set the capture mode (capture images with camera 1, with camera 2, or with both 2D cameras and compose the outputs).
- **SetUHPFringeCodingMode**: Set the coding mode of the structured light pattern.

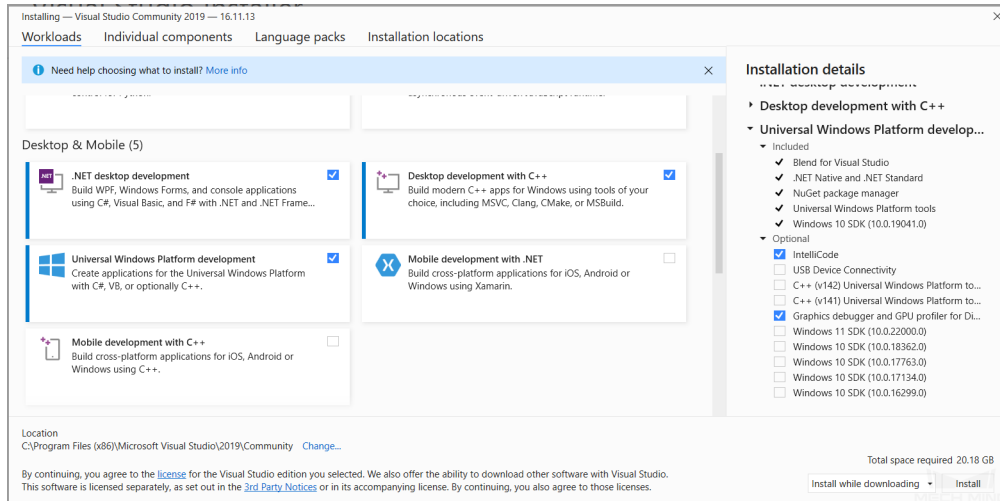
### 11.3.2 Requirements

1. Download Visual Studio 2019

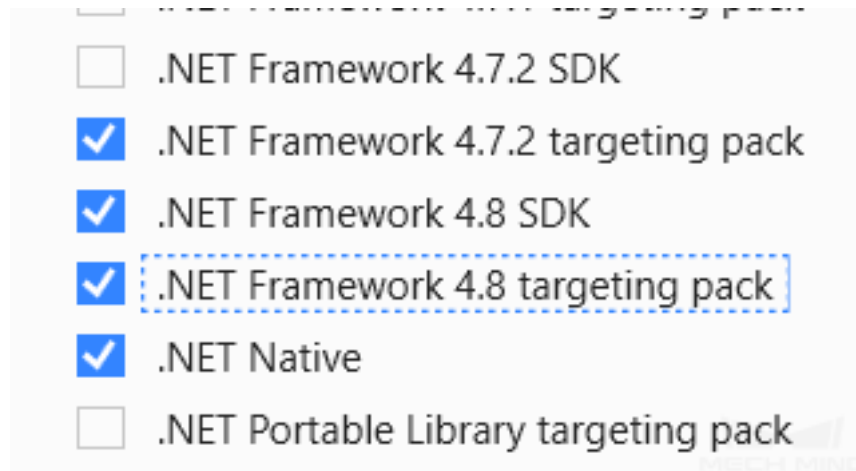
It is recommended to use Visual Studio 2019 or Visual Studio 2015.

2. Choose and install workloads and individual components in Visual Studio

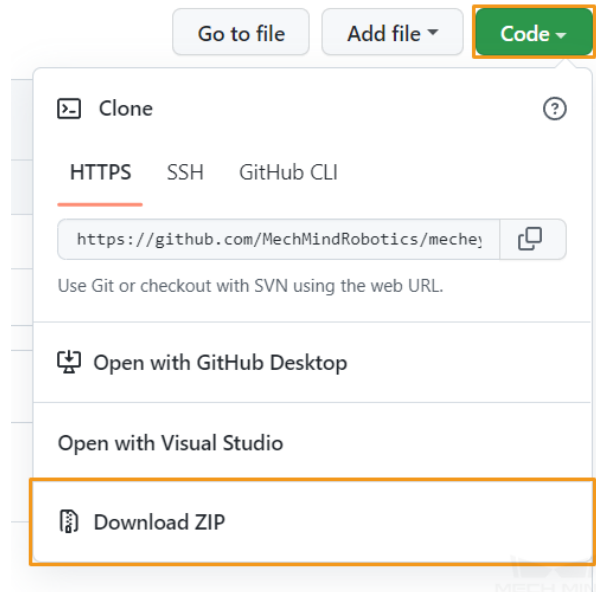
Workloads: .NET desktop development, Desktop development with C++, Universal Windows Platform development



Individual components: .NET Framework 4.8 targeting pack



3. *Download and install Mech-Eye SDK*
4. *Download samples from GitHub: select Code → Download as ZIP.*



5. *Connect a functioning camera*

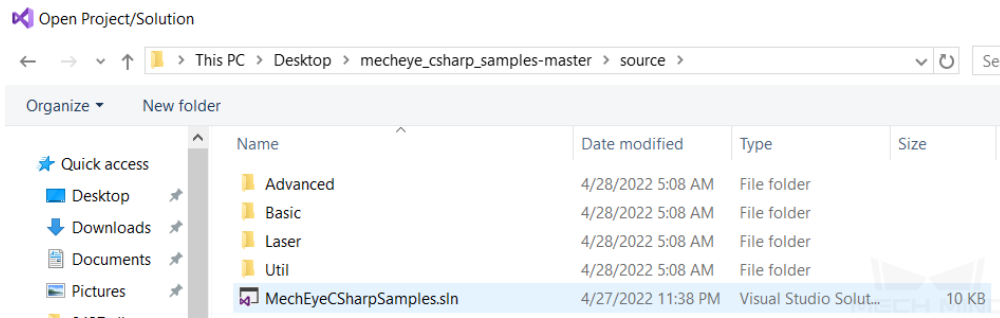
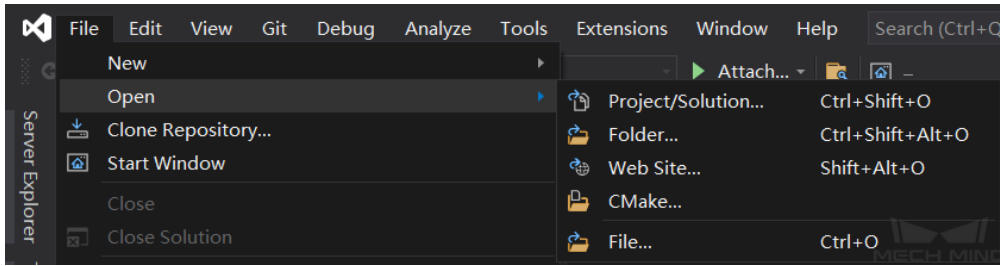
6. To use the following samples, `Emgu.CV.runtime.windows` must be installed in Visual Studio via *Project* → *Manage NuGet Packages*. Please refer to *Install and manage packages in Visual Studio* using the *NuGet Package Manager* for detailed instructions.

- `CaptureColorMap`
- `CaptureDepthMap`
- `CapturePointCloud`
- `CaptureHDRPointCloud`
- `CapturePointCloudROI`
- `CapturePointCloudFromTextureMask`
- `CaptureCloudFromDepth`
- `CaptureSequentiallyMultiCamera`
- `CaptureSimultaneouslyMultiCamera`
- `CaptureTimedAndPeriodically`

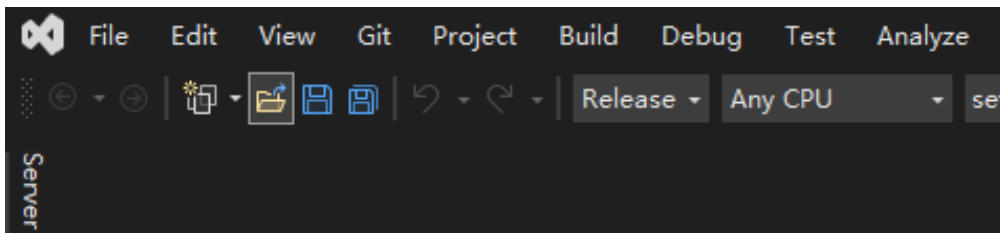
### 11.3.3 Instructions

#### Build Solution in Visual Studio

1. Open `MechEyeCSharpSamples.sln` in Visual Studio.

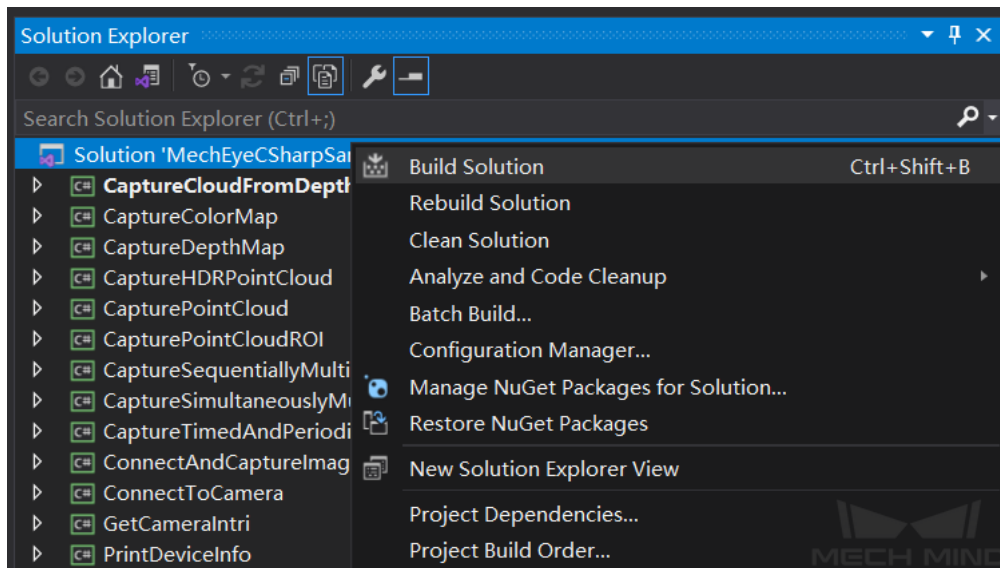


2. Change the Solution Configuration from Debug to Release.



3. Build solution.

Right-click Solution 'MechEyeCSharpSamples' in the Solution Explorer, and click **Build Solution** in the context menu.



### Run Sample Program

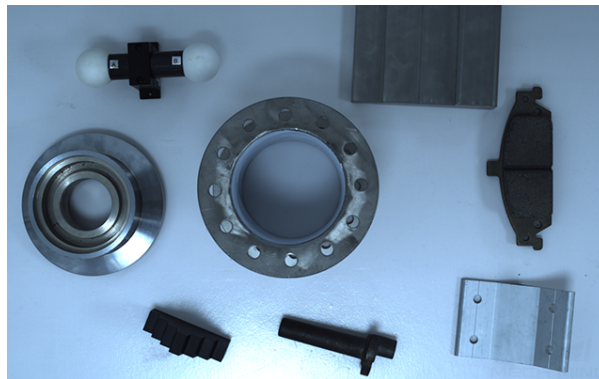
The example below shows how to run CaptureColorMap.exe.

1. Go to *mecheye\_csharp\_samples-master* → *source* → *Build*.

2. Double-click CaptureColorMap.exe.

While the program is running, please follow the instructions to input the camera index, and then wait for the program to finish.

3. After running the program, you will get colorMap in Build directory.



## 11.4 Python

This section will show you how to run a Python sample in Windows.

### 11.4.1 Sample List

Samples are divided into five categories, **Basic**, **Advanced**, **Util**, **Laser** and **UHP**.

- **Basic**: camera connection and basic capturing functions.
- **Advanced**: advanced capturing functions.
- **Util**: obtain information from a camera and set camera parameters.
- **Laser**: for Laser, LSR and DEEP series cameras only.
- **UHP**: for UHP series cameras only.

#### Basic

- **ConnectToCamera**: Connect to a Mech-Eye Industrial 3D Camera.
- **ConnectAndCaptureImage**: Connect to a camera and obtain the 2D image, depth map and point cloud data.
- **CaptureColorMap**: Obtain and save the 2D image from a camera.
- **CaptureDepthMap**: Obtain and save the depth map from a camera.
- **CapturePointCloud**: Obtain and save untextured and textured point clouds generated from images captured with a single exposure time.
- **CaptureHDRPointCloud**: Obtain and save untextured and textured point clouds generated from images captured with multiple exposure times.
- **CapturePointCloudROI**: Obtain and save untextured and textured point clouds of the objects in the ROI from a camera.
- **CapturePointCloudFromTextureMask**: Construct and save untextured and textured point clouds generated from a depth map and masked 2D image.

#### Advanced

- **CaptureCloudFromDepth**: Construct and save point clouds from the depth map and 2D image obtained from a camera.
- **CaptureSequentiallyMultiCamera**: Obtain and save 2D images, depth maps and point clouds sequentially from multiple cameras.
- **CaptureSimultaneouslyMultiCamera**: Obtain and save 2D images, depth maps and point clouds simultaneously from multiple cameras.
- **CaptureTimedAndPeriodically**: Obtain and save 2D images, depth maps and point clouds periodically for the specified duration from a camera.

#### Util

- **GetCameraIntri**: Get and print a camera's intrinsic parameters.
- **PrintDeviceInfo**: Get and print a camera's information such as model, serial number and firmware version.

- **SetDepthRange**: Set the range of depth values to be retained by a camera.
- **SetParameters**: Set specified parameters to a camera.
- **SetUserSets**: Perform functions related to parameter groups, such as switching to a different parameter group, adding and deleting parameter groups, etc. The parameter group feature allows user to save and quickly apply a set of parameter values.

**Laser**

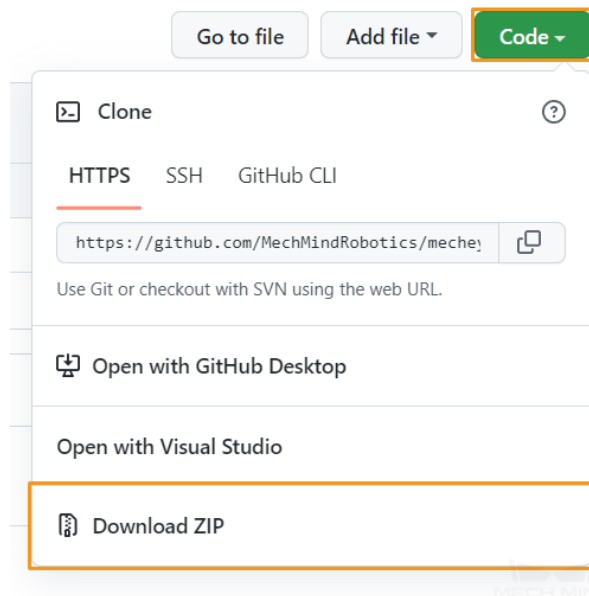
- **SetLaserFramePartitionCount**: Divide the projector FOV into partitions and project structured light in one partition at a time. The output of the entire FOV is composed from images of all partitions.
- **SetLaserFrameRange**: Set the projection range of the structured light. The entire projector FOV is from 0 to 100.
- **SetLaserFringeCodingMode**: Set the coding mode of the structured light pattern.
- **SetLaserPowerLevel**: Set the output power of the laser projector in percentage of max power. This affects the intensity of the laser light.

**UHP**

- **SetUHPCaptureMode**: Set the capture mode (capture images with camera 1, with camera 2, or with both 2D cameras and compose the outputs).
- **SetUHPFringeCodingMode**: Set the coding mode of the structured light pattern.

**11.4.2 Requirements**

1. *Download and install Mech-Eye SDK*
2. Download samples from GitHub: select *Code* → *Download as ZIP*.



3. Make sure that the version of Python installed is between 3.6.5 and 3.10

**Attention:** If **Add Python to PATH** was not selected during installation, please add the following two directories to the **Path** environment variable:

- xxx\AppData\LocalProgramsPython\Python36\
- xxx\AppData\LocalProgramsPython\Python36\Scripts\

4. *Connect a functioning camera*

### 11.4.3 Instructions

#### Setting up Python

Press Win + R on the keyboard to open **Run** box. Type "PowerShell" and then press Ctrl+Shift+Enter to run PowerShell as Administrator.

- Required: MechEyeAPI.

```
pip install MechEyeAPI
```

If an error occurs, please enter the command as below:

```
pip install MechEyeAPI -i http://pypi.douban.com/simple/ --trusted-host pypi.douban.com
```

- Optional: open3d, opencv-python, opencv-contrib-python.

Samples	open3d	opencv-python	opencv-contrib-python
CaptureColorMap		✓	✓
CaptureDepthMap		✓	✓
CapturePointCloud	✓		
CaptureHDRPointCloud	✓		
CapturePointCloudROI	✓		
CapturePointCloudFromTextureMask	✓		
CaptureCloudFromDepth	✓		

```
pip install opencv-python
pip install opencv-contrib-python
```

**Attention:** The version of opencv-python should be the same as that of opencv-contrib-python.



## Run Sample Program

The example below shows how to run CaptureColorMap.py.

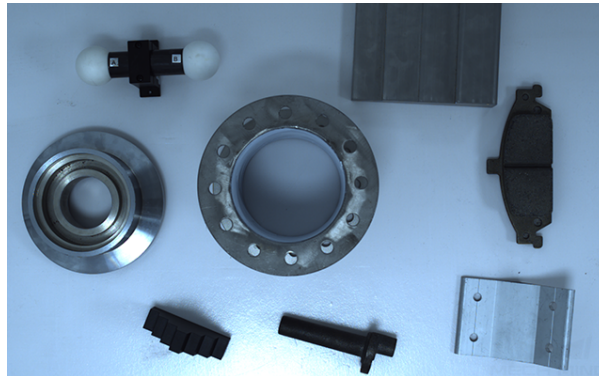
1. Navigate to the directory where CaptureColorMap.py is located.

```
cd xxx\mecheye_python_samples\source\Basic
```

2. Run the sample.

```
python.exe CaptureColorMap.py
```

3. After running the program, you will get colorMap in samples directory.



## 11.5 Python (Ubuntu)

This section describes how to run Python samples on Ubuntu.

### 11.5.1 Sample List

Samples are divided into five categories, **Basic**, **Advanced**, **Util**, **Laser** and **UHP**.

- **Basic**: camera connection and basic capturing functions.
- **Advanced**: advanced capturing functions.
- **Util**: obtain information from a camera and set camera parameters.
- **Laser**: for Laser, LSR and DEEP series cameras only.
- **UHP**: for UHP series cameras only.

#### Basic

- **ConnectToCamera**: Connect to a Mech-Eye Industrial 3D Camera.
- **ConnectAndCaptureImage**: Connect to a camera and obtain the 2D image, depth map and point cloud data.

- **CaptureColorMap**: Obtain and save the 2D image from a camera.
- **CaptureDepthMap**: Obtain and save the depth map from a camera.
- **CapturePointCloud**: Obtain and save untextured and textured point clouds generated from images captured with a single exposure time.
- **CaptureHDRPointCloud**: Obtain and save untextured and textured point clouds generated from images captured with multiple exposure times.
- **CapturePointCloudROI**: Obtain and save untextured and textured point clouds of the objects in the ROI from a camera.
- **CapturePointCloudFromTextureMask**: Construct and save untextured and textured point clouds generated from a depth map and masked 2D image.

#### Advanced

- **CaptureCloudFromDepth**: Construct and save point clouds from the depth map and 2D image obtained from a camera.
- **CaptureSequentiallyMultiCamera**: Obtain and save 2D images, depth maps and point clouds sequentially from multiple cameras.
- **CaptureSimultaneouslyMultiCamera**: Obtain and save 2D images, depth maps and point clouds simultaneously from multiple cameras.
- **CaptureTimedAndPeriodically**: Obtain and save 2D images, depth maps and point clouds periodically for the specified duration from a camera.

#### Util

- **GetCameraIntri**: Get and print a camera's intrinsic parameters.
- **PrintDeviceInfo**: Get and print a camera's information such as model, serial number and firmware version.
- **SetDepthRange**: Set the range of depth values to be retained by a camera.
- **SetParameters**: Set specified parameters to a camera.
- **SetUserSets**: Perform functions related to parameter groups, such as switching to a different parameter group, adding and deleting parameter groups, etc. The parameter group feature allows user to save and quickly apply a set of parameter values.

#### Laser

- **SetLaserFramePartitionCount**: Divide the projector FOV into partitions and project structured light in one partition at a time. The output of the entire FOV is composed from images of all partitions.
- **SetLaserFrameRange**: Set the projection range of the structured light. The entire projector FOV is from 0 to 100.
- **SetLaserFringeCodingMode**: Set the coding mode of the structured light pattern.
- **SetLaserPowerLevel**: Set the output power of the laser projector in percentage of max power. This affects the intensity of the laser light.

#### UHP

- **SetUHPCaptureMode**: Set the capture mode (capture images with camera 1, with camera 2, or with both 2D cameras and compose the outputs).
- **SetUHPFringeCodingMode**: Set the coding mode of the structured light pattern.

The following samples require third-party libraries to be installed (instructions provided below).

Samples	open3d	opencv-python
CaptureColorMap		√
CaptureDepthMap		√
CapturePointCloud	√	
CaptureHDRPointCloud	√	
CapturePointCloudROI	√	
CapturePointCloudFromTextureMask	√	
CaptureCloudFromDepth	√	

## 11.5.2 Requirements

1. *Connect a functioning camera.*
2. Make sure that the version of Python installed is between 3.6.5 and 3.8.
3. *Download and install Mech-Eye SDK.*
4. (Optional) Install OpenCV:

```
sudo apt-get install libopencv-dev  
sudo apt-get install python3-opencv
```

5. (Optional) Install Open3D:

```
sudo pip install open3d
```

## 11.5.3 Instructions

### Install Required Software

1. Install pip:

```
sudo apt install python3-pip
```

If you have already installed pip, please make sure that the version of pip is 20.3 or above. Use the following command to upgrade pip:

```
python3 -m pip install --upgrade pip
```

2. Install MechEyeApi:

```
sudo pip3 install MechEyeApi
```

## Run the Sample

1. Download the samples:

```
cd ~
git clone https://github.com/MechMindRobotics/mecheye_python_samples.git
```

2. Open a sample folder and run the sample (taking ConnectToCamera.py as an example):

```
cd ~/mecheye_python_samples/source/Basic
sudo python3 ConnectToCamera.py
```

For ConnectToCamera, you will be prompted to enter the index of the camera that you want to connect:

```
Please enter the device index you want to connect:
```

Please enter the index and press **Enter** to confirm.

Below is an example of the full output of ConnectToCamera.py (0 is entered for the index):

```
Find Mech-Eye device...
Mech-Eye device index : 0
.....
Camera Model Name: Mech-Eye Pro M Enhanced
Camera ID:         NEC15221A3000001
Camera IP Address: 192.168.xx.xx
Hardware Version:  Vx.x.x
Firmware Version:  Vx.x.x
.....

Please enter the device index you want to connect: 0
Connected to the Mech-Eye device successfully.
Disconnected from the Mech-Eye device successfully.
```

### Attention:

- For samples that involve point cloud capturing, a regular user is unable to write point cloud files, so please use the sudo command to run the sample as the admin.
- For samples that involve image and point cloud capturing, when the sample execution is complete, the image and point cloud data is by default saved to the folder of the sample.

**Note:** When writing your own program, use the import command to import Mech-Eye API first. Please import Mech-Eye API before importing Open3D.

```
python3
import MechEye
```

## 11.6 ROS

This section will show you how to use Mech-Eye SDK on ROS on Ubuntu.

### 11.6.1 Requirements

1. Operating system: [Ubuntu](#).
2. ROS installed according to instructions on [Documentation of ROS](#).

**Note:**

1. In the example below, the Ubuntu system's version is **18.04**, but the installation of ROS will be different for different versions of Ubuntu systems.
2. If using a virtual machine to install the software, please reserve **more than 20G** of disk space; otherwise, the installation may fail.
3. If after following the instructions below, you can run turtlesim normally, you have successfully installed ROS.

```
# Open a terminal in any directory and enter the command:  
roscore  
# Open a second terminal and enter the command:  
roslaunch turtlesim turtlesim_node  
# Open a third terminal and enter the command:  
roslaunch turtlesim turtle_teleop_key
```

After executing the above command, in the third terminal window, you will be able to control the little turtle by pressing the up, down, left, and right keys on the keyboard, as shown in the figure below.



## 11.6.2 Instructions

### Install Mech-Eye SDK for Ubuntu

Please see [:ref:ref-install-sdk-on-ubuntu](#) for instructions on installing.

### Install ROS API

1. Install libzmq5 and libzmq3-dev. ... code-block:: sh  
sudo apt install libzmq5 libzmq3-dev
2. Create directory `~/ros_ws/src` and open it.

```
mkdir -p ~/ros_ws/src && cd ~/ros_ws/src
```

3. Clone repository `mecheye_ros_interface` to directory `~/ros_ws/src`.

```
git clone https://github.com/MechMindRobotics/mecheye_ros_interface
```

4. Open directory `~/ros_ws` and build.

```
cd ~/ros_ws && catkin_make
```

5. Modify the files.

- If you need to store the images, please change `false` to `true` in the line `<arg name="save_file" default="true"/>` in the file `~/ros_ws/src/mecheye_ros_interface/launch/start_camera.launch`.
- The path to store the images can be modified in the file `~/ros_ws/src/mecheye_ros_interface/src/main.cpp`.

## Run the Service

1. Open a new terminal and run the following command to start the service.

```
source ~/ros_ws/devel/setup.bash
roslaunch mecheye_ros_interface start_camera.launch
```

The output is as follows:

```
... logging to /home/mechmind/.ros/log/69ffe3a2-d290-11ec-828a-
000c29351753/roslaunch-ubuntu-4349.log
Checking log directory for disk usage. This may take a while.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://ubuntu:38711/

SUMMARY
=====

PARAMETERS
 * /mechmind_camera_start/camera_ip: 192.168.0.123
 * /mechmind_camera_start/fx: 1727.46410256
 * /mechmind_camera_start/fy: 1727.45869267
 * /mechmind_camera_start/save_file: False
 * /mechmind_camera_start/u: 655.818082573
 * /mechmind_camera_start/use_external_intri: False
 * /mechmind_camera_start/v: 516.630650061
 * /rostdistro: melodic
 * /rosversion: 1.14.13

NODES
 /
  mechmind_camera_start (mecheye_ros_interface/start)

auto-starting new master
process[master]: started with pid [4359]
ROS_MASTER_URI=http://localhost:11311
```

(continues on next page)

(continued from previous page)

```

setting /run_id to 69ffe3a2-d290-11ec-828a-000c29351753
process[rosout-1]: started with pid [4370]
started core service [/rosout]
process[mechmind_camera_start-2]: started with pid [4377]
Find Mech-Eye devices...
Mech-Eye device index : 0
.....
Camera Model Name: Mech-Eye Pro M Enhanced
Camera ID:      Sample
Camera IP Address: 192.168.xx.xx
Hardware Version: Vx.x.x
Firmware Version: Vx.x.x
.....
Please enter the device index you want to connect: 0
Connected to the Mech-Eye device successfully.
.....
Camera Model Name: Mech-Eye Nano
Camera ID:      TAM06218A3020706
Camera IP Address: 192.168.xx.xx
Hardware Version: Vx.x.x
Firmware Version: Vx.x.x
.....

```

2. Open another new terminal and run the following command to capture images.

```

source ~/ros_ws/devel/setup.bash
rosservice call /run_mechmind_camera

```

The output is as follows:

```

success: True
message: ''

```

At this point, the camera has started working. Data such as 2D images and depth maps are saved in the folder /tmp by default.

**Attention:** Every time you modify a file under ~/ros\_ws/src/mecheye\_ros\_interface/src, please do `cd ~/ros_ws && catkin_make` to update the executable.



## **MECH-EYE API REFERENCE**

Please refer to [Mech-Eye API Reference](#) for detailed information.

## GENICAM QUICK FACTS

Mech-Eye Industrial 3D Cameras (hereinafter referred to as cameras) support the GenICam standard, which allows GenICam-compliant third-party machine vision software (hereinafter referred to as GenICam clients) to control them.

After *upgrading the firmware* to 2.0.0 or above, you can directly control the camera with GenICam clients, such as HALCON (MVTec) and VisionPro (Cognex), without any extra procedures.

---

Basic knowledge on **GenICam**:

*GenICam Basics*

---

Instructions on **using HALCON to control the camera**:

*Use HALCON to Control the Camera*

---

**Useful reference information** for using GenICam:

*Camera Parameters Supported for GenICam*

*Use Mech-Eye Viewer to Adjust Camera Parameters for GenICam Client*

*Obtain HALCON-Readable Point Cloud through Mech-Eye API*

## GENICAM BASICS

This topic covers the following contents:

- *Introduction*
- *Working Principles*
- *Supporting GenICam*

### 14.1 Introduction

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 (MVTec), VisionPro (Cognex) can be used to conveniently connect and control cameras without camera-specific configuration.

### 14.2 Working 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.
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: <https://www.emva.org/standards-technology/genicam>.

## 14.3 Supporting GenICam

After *the firmware has been upgraded* to 2.0.0 or above, you can control cameras with third-party machine vision software such as HALCON (MVTec) and VisionPro (Cognex).

For camera parameters provided for GenICam by the camera and their description, refer to *Camera Parameters Supported by GenICam* .

If you want to connect and manage the camera with the HALCON software, refer to *Use HALCON to Control the Camera* .

## USE HALCON TO CONTROL THE CAMERA

This topic provides instructions on controlling Mech-Eye Industrial 3D Camera with HALCON and introduction on the HALCON samples.

- *Prerequisites*
- *Control the Camera with Image Acquisition Assistant*
- *Control the Camera with Sample Program*

### 15.1 Prerequisites

- *The camera and IPC are properly connected.*

---

**Hint:** For Nano and Pro XS, it is recommended to connect the camera directly to the IPC, without using a network switch.

---

- HALCON 20.11 or above has been installed on IPC.
- Mech-Eye SDK 2.0.0 or above has been installed on IPC.

---

**Hint:** 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 IPC are in the same subnet.

---

**Hint:** For instructions on setting the IP addresses, please refer to *Connect to Camera and Set IP Address*.

---

---

In HALCON, you can control the camera with Image Acquisition Assistant or the sample program.

- *Control the Camera with Image Acquisition Assistant*
- *Control the Camera with Sample Program*

## 15.2 Control the Camera with Image Acquisition Assistant

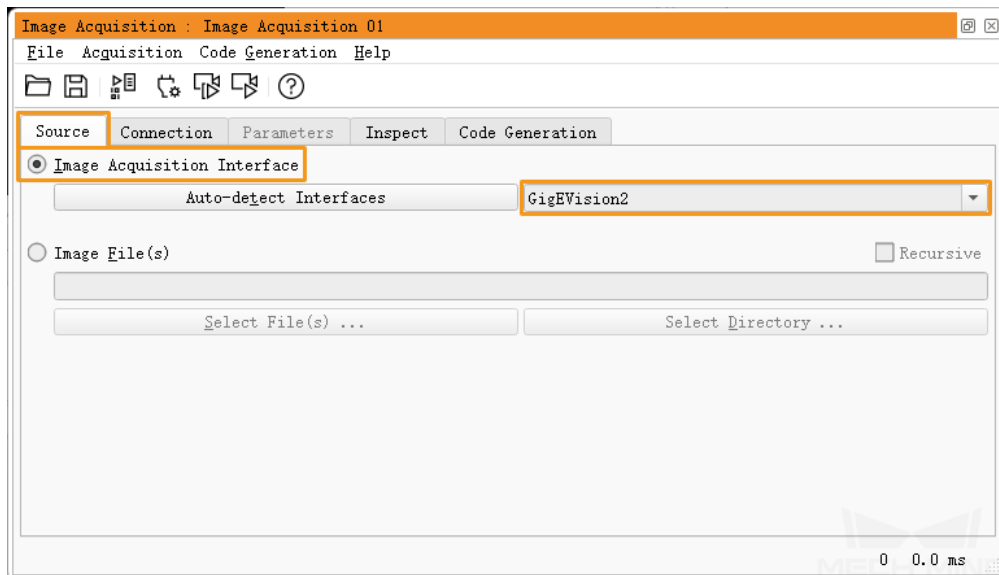
Using the image acquisition assistant provided by HDevelop, you can quickly connect to the camera, perform image capturing and adjust camera parameters.

To open the image acquisition assistant: open HDevelop, and in the menu bar, select *Assistants* → *Open New Image Acquisition*.

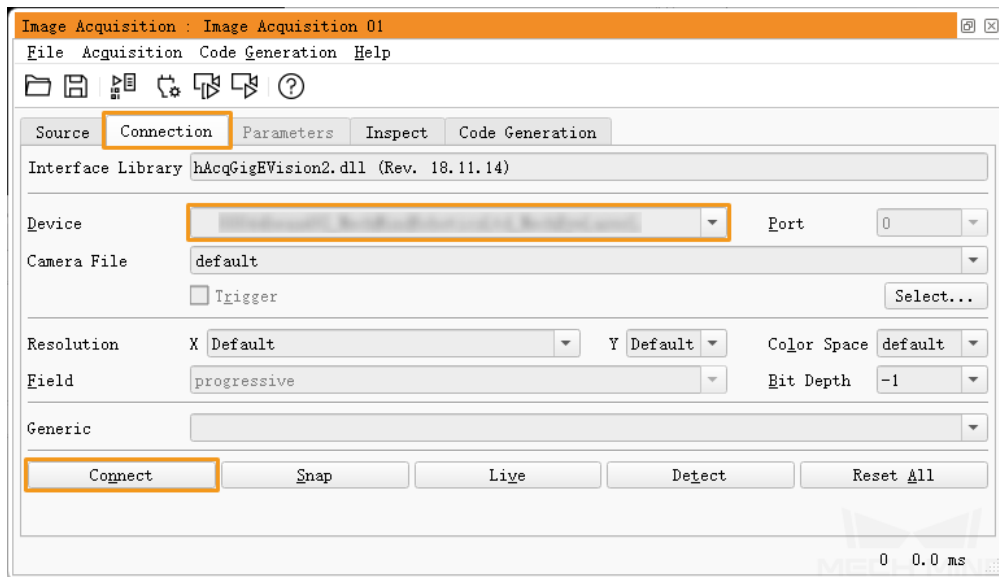
### 15.2.1 Connect to the Camera

1. In the **Image Acquisition** window, under the **Source** tab, select **Image Acquisition Interface**, and select **GigEvision2** from the drop-down menu.

**Hint:** 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.


**Hint:**

- If the camera is successfully connected, the *Connect* button changes to *Disconnect*. If this change does not occur, the camera connection failed.
- If the camera is already connected by another client, it cannot be connected by HDevelop. Please disconnect the camera from the other client first.
- To disconnect from the camera in HDevelop, click *Disconnect*.
- The camera name displayed here *can be customized in Mech-Eye Viewer*. It is recommended to avoid using special characters such as letters with accent marks, as these characters cannot be displayed in HDevelop.

## 15.2.2 Perform Image Capturing

To perform image capturing once: click *Snap* under the **Connection** tab in the **Image Acquisition** window.

**Note:** If the image capturing process takes too long, you can *enable jumbo frames* on your IPC.

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:
  1. Switch to the **Parameters** tab, set the value of the **AcquisitionMode** parameter to **MultiFrame**.
  2. Click *Refresh* in the upper right, and then set the number of times for image capturing in the **AcquisitionFrameCount** parameter.
  3. Switch to the **Connection** tab and click *Live* to perform image capturing.

4. 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:
    1. Switch to the **Parameters** tab, set the value of the **AcquisitionMode** parameter to **Continuous**.
    2. Switch to the **Connection** tab and click *Live* to perform image capturing.
    3. The *Live* button becomes *Stop*. Click *Stop* to stop image capturing.

**Note:**

- 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.

### 15.2.3 Select Data Type

After the camera is connected, 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:

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.

**Hint:** 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.

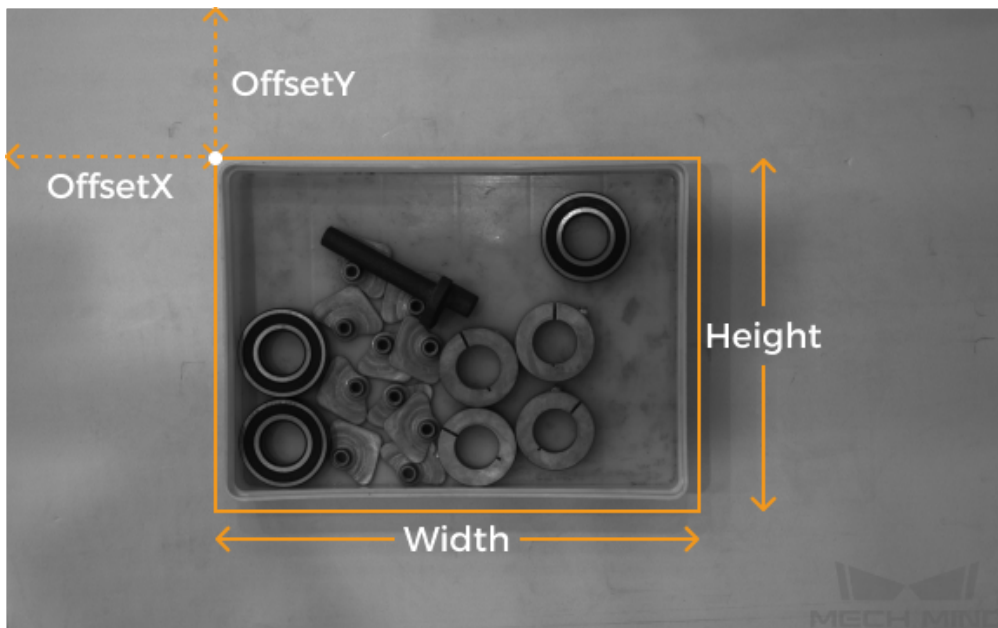
### 15.2.4 Set Capture Region

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 with *Snap* to see 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

**Note:** The above four parameters must satisfy the following requirements:

- (Width + OffsetX) not greater than the width of the original image
- (Height + 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.

**Hint:** If **Update Image** in the upper right is checked, the image in the **Canvas** window is automatically updated as you adjust the parameters.

5. Switch to the **Code Generation** tab, click *Inser Code* to generate the corresponding code.
6. If you need to set a capture region for the other data type:
  1. Disconnect from the camera in the current image acquisition assistant.
  2. Open a new image acquisition assistant, and connect to the camera.
  3. Select the other data type and repeat the above steps.

**Note:**

- The values of the four capture region parameters are not saved to parameter groups and are reset if the camera is powered off. 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 <i>visualized tool in Mech-Eye Viewer</i>

### 15.2.5 Adjust Parameters

If the quality of the obtained data is unsatisfactory, you can adjust the camera parameters under the **Parameters** tab.

#### Hint:

- 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. After configuring the parameter group in Mech-Eye Viewer, click the *Refresh* button in the upper-right corner of the **Parameters** tab in 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.



- Find the **UserSetLoad** parameter, and click *Apply* to the right to read in the configuration.

---

**Note:** If parameter values are not updated after you click *Apply*, please click it again.

---

- Find the parameter that you want to adjust and change its value.
- Find the **UserSetSave** parameter, and click *Apply* to the right to save the configuration.
- Switch to the **Code Generation** tab, click *Inser Code* to generate the corresponding code.

#### References:

- For descriptions and explanations of camera parameters supported for GenICam, download and read [Parameters in GenICam Client](#).
- Camera parameters supported for GenICam correspond with those provided by Mech-Eye Viewer. For their correspondence, 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](#).

---

**Hint:** 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.

---


## 15.3 Control the Camera with Sample Program

A HALCON sample program is provided on GitHub. With this sample program, you can connect to the camera, capture images, adjust parameters and save the data. You can run the program as is or modify it to better suit your needs.

**Note:** The HALCON sample program is written in C++ and can be executed in HDevelop of HALCON directly.

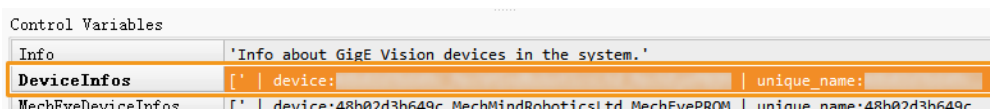
### 15.3.1 Run the HALCON Sample Program

1. Obtain the HALCON sample program.
2. Open the sample program in any of the following ways:
  - Open HDevelop, and drag the sample program into HDevelop.
  - Right-click the sample program and select *Open with* → *HDevelop*.
3. Obtain all the available cameras in any of the following ways:

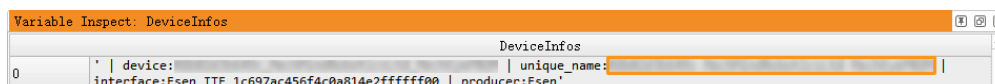
- Select the line containing the `info_framegrabber` operator in **Program Windows** and click  in the toolbar.
- Select the line containing the `info_framegrabber` operator in **Program Windows** and press the **F6** key to execute the first operator.

**Hint:** All the obtained cameras can be viewed in the **DeviceInfos** variable in the **Control Variables** area of the **Variable Window**.

4. Select a camera to connect in any of the following ways:
  - Select a camera by its **unique\_name**:
    - (1) In the **Control Variables** area, double-click **DeviceInfos** to display a list of all the available cameras.



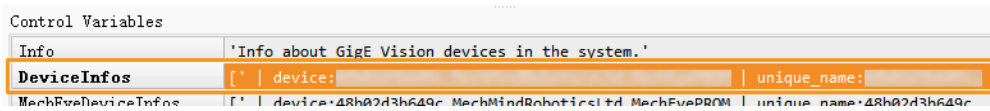
- (2) In the list, double-click the camera that you want to connect, and copy the camera name after **unique\_name**:



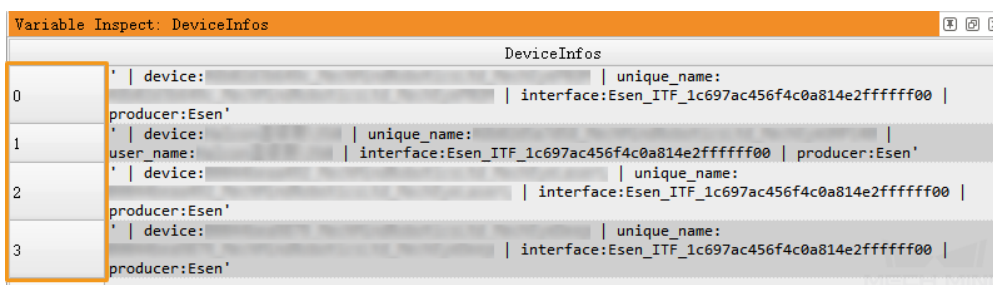
(3) In the **Program Window**, locate the line containing the `tuple_regexp_select` operator, and replace **MechEye** in single quotation marks with the copied `unique_name`.

- Select a camera by its index number:


(1) In the **Control Variables** area, double-click **DeviceInfos** to display a list of all the available cameras.




(2) Check the index number of the camera that you want to connect.



(3) In the **Program Window**, locate the line containing the `open_framegrabber` operator, and change the number in **MechEyeCameraInfos[X]** accordingly.

5. Run the sample program by clicking  or pressing the **F5** key.
6. If the program is successfully executed, you can find the saved 2D image (**image2d.bmp** by default) and point cloud (**PointCloud.ply** by default) in the folder where the sample program is located.

**Note:**

- After executing the entire sample program, 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 *enable jumbo frames* on your IPC.
- 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)
```

### 15.3.2 Explanation of the Sample Program

- Connect to the camera:

```
info_framegrabber ('GigEVision2', 'device', Info, DeviceInfos)
tuple_regexp_select(DeviceInfos, 'MechEye', MechEyeCameraInfos)
open_framegrabber ('GigEVision2', 1, 1, 0, 0, 0, 0, 'default', -1, 'default', -1, 'false',
↪ 'default', MechEyeCameraInfos[0], 0, -1, AcqHandle)
```

- Perform image capturing:

```
grab_image (Image2d, AcqHandle)
grab_data(Image3d, Region, Contours, AcqHandle, ObjectModel3D)
```

- Obtain available parameters:

```
get_framegrabber_param (AcqHandle, 'available_param_names', ParameterValues)
```

- Modify parameters:

Parameters included in the sample program are 2D parameters, 3D parameters, ROI, depth range and point cloud settings.

- 2D parameters:

```
* Switch to the 2D scanning mode.
set_framegrabber_param (AcqHandle, 'DeviceScanType', 'Areascan')
set_framegrabber_param (AcqHandle, 'AcquisitionMode', 'SingleFrame')

get_framegrabber_param (AcqHandle, 'Width', Width)
get_framegrabber_param (AcqHandle, 'Height', Height)
get_framegrabber_param (AcqHandle, 'PixelFormat', PixelFormat)

* Set the 2D scanning parameters.
set_framegrabber_param (AcqHandle, 'Scan2DExposureMode', 'Timed')
set_framegrabber_param (AcqHandle, 'Scan2DExposureTime', 100)
```

- 3D parameters:

```
* Switch to the 3D scanning mode:
set_framegrabber_param (AcqHandle, 'DeviceScanType', 'Areascan3D')

* Open the 3D object model generator.
set_framegrabber_param (AcqHandle, 'create_objectmodel3d', 'enable')
set_framegrabber_param (AcqHandle, 'add_objectmodel3d_overlay_attrib', 'enable')

get_framegrabber_param (AcqHandle, 'Width', Width)
get_framegrabber_param (AcqHandle, 'Height', Height)
get_framegrabber_param (AcqHandle, 'PixelFormat', PixelFormat)
```

(continues on next page)

(continued from previous page)

```
* Set the 3D scanning exposure time parameter.
set_framegrabber_param (AcqHandle, 'Scan3DExposureCount', 1)
set_framegrabber_param (AcqHandle, 'Scan3DExposureTime', 8)
```

- Depth range:

```
* Set the 3D scanning depth range parameter(unit: mm).
set_framegrabber_param (AcqHandle, 'DepthLowerLimit', 1)
set_framegrabber_param (AcqHandle, 'DepthUpperLimit', 3000)
```

- Point cloud settings:

```
* Set the 3D scanning point cloud filter parameter.
set_framegrabber_param (AcqHandle, 'CloudOutlierFilterMode', 'Normal')
set_framegrabber_param (AcqHandle, 'CloudSmoothMode', 'Normal')
```

- Save the obtained data:

- Save 2D image:

```
* Acquire the 2D image from the camera.
grab_image (Image2d, AcqHandle)

* Save the 2D image result in "Image2d.bmp"
write_image( Image2d , 'bmp' , 0 , 'Image2d' )
```

- Save point cloud:

```
* Acquire the 3D data from the camera.
grab_data(Image3d, Region, Contours, AcqHandle, ObjectModel3D)

get_object_model_3d_params(ObjectModel3D, 'num_points', NumOfPoints)

if(NumOfPoints != 0)
    * Save the point cloud result in "PointCloud.ply"
    write_object_model_3d (ObjectModel3D, 'ply', 'PointCloud.ply', [], [])
endif
```

- Disconnect from the camera:

```
clear_object_model_3d (ObjectModel3D)
close_framegrabber (AcqHandle)
```

### 15.3.3 Adjust Parameters with HALCON Program

If the quality of the obtained data is unsatisfactory, you can adjust the camera parameters.

---

#### Hint:

- 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:

1. Use the following operator to obtain a list of available parameter groups.

```
get_framegrabber_param (AcqHandle, 'UserSetSelector_values', ParameterValues)
```

2. View the list of parameter groups in the **ParameterValues** variable in the **Control Variables** area.
3. 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. .. code-block:

```
set_framegrabber_param (AcqHandle, 'UserSetSelector', 'ParameterGroupName')
set_framegrabber_param (AcqHandle, 'UserSetLoad', 'ParameterGroupName')
```

4. 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.

```
get_framegrabber_param (AcqHandle, 'ParameterName', ParameterValues)
```

5. View the parameter value in the **ParameterValues** variable in the **Control Variables** area.
6. 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')
```

7. 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:

- For descriptions and explanations of camera parameters supported for GenICam, download and read [Parameters in GenICam Client](#).
- Camera parameters supported for GenICam correspond with those provided by Mech-Eye Viewer. For their correspondence, 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 by 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](#).

---

**Hint:** 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.

---



## GENICAM REFERENCE

This topic provides reference information required for you to use a GenICam client to control the camera.

---

Camera parameters supported for GenICam and their descriptions:

*Camera Parameters Supported for GenICam*

---

Instructions on using Mech-Eye Viewer to adjust camera parameters for 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*

---

### 16.1 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.

- *Supported Camera Parameters*
- *Correspondence Between Camera Parameters Supported for GenICam and By Mech-Eye Viewer*



#### 16.1.1 Supported Camera Parameters

For descriptions and explanations of camera parameters supported for GenICam, download and read [Parameters in GenICam Client](#).

### 16.1.2 Correspondence Between Camera Parameters Supported for GenICam and By Mech-Eye Viewer

Camera parameters supported for GenICam generally correspond with those provided by Mech-Eye Viewer. Mech-Eye Viewer provides more detailed information such as parameter descriptions and parameter values. You can use these information to assist you to adjust parameters in GenICam clients.

The following table provides the correspondence between camera parameters supported in GenICam clients and Mech-Eye Viewer:

Parameter in GenICam Client	Parameter in Mech-Eye Viewer
Scan2DROIHeight	2D Parameters - <i>Auto-Exposure ROI</i>
Scan2DROILeft	
Scan2DROITop	
Scan2DROIWidth	
Scan2DExposureMode	2D Parameters - <i>Exposure Mode - Color Camera Exposure Mode</i>
Scan2DExpectedGrayValue	2D Parameters - <i>Gray Value</i>
Scan2DHDRExposureSequence	2D Parameters - <i>Exposure Sequence</i>
Scan2DToneMappingEnable	2D Parameters - <i>Tone Mapping</i>
Scan2DExposureTime	2D Parameters - <i>Exposure Time</i>
Scan2DSharpenFactor	2D Parameters - <i>Sharpen Factor</i>
Scan2DPatternRoleExposureMode	2D Parameters - <i>Monochrome Camera Exposure Mode</i>
Scan2DPatternRoleExposureTime	2D Parameters - <i>Monochrome Camera Exposure Time</i>
Scan3DROIHeight	<i>ROI</i>
Scan3DROILeft	
Scan3DROITop	
Scan3DROIWidth	
ProjectorLightColor	3D Parameters - Projector — <i>Light Color</i>
ProjectorPowerLevel	3D Parameters - Projector — <i>Light Brightness</i>
ProjectorFringeCodingMode	3D Parameters - Projector — <i>Fringe Coding Mode</i>
AntiFlickerMode	3D Parameters - Projector — <i>Anti-Flicker Mode</i>
ProjectorSelectionMode	3D Parameters - Projector — <i>Projector Selection Mode</i>
ProjectorSelector	3D Parameters - Projector — <i>Projector Selector</i>
LaserFringeCodingMode	3D Parameters - Laser — <i>Fringe Coding Mode</i>
LaserPowerLevel	3D Parameters - Laser — <i>Laser Power</i>
LaserFrameAmplitude	3D Parameters - Laser Projection Frame Control — <i>Laser Frame Amplitude</i>
LaserFrameOffset	3D Parameters - Laser Projection Frame Control — <i>Laser Frame Offset</i>
LaserFramePartitionCount	3D Parameters - Laser Projection Frame Control — <i>Laser Frame Partition Count</i>
Scan3DBinningEnable	3D Parameters - <i>3D Binning</i>
Scan3DExposureCount	3D Parameters - <i>Exposure Multiplier</i>
Scan3DExposureTime	3D Parameters - <i>Exposure Time</i>
Scan3DExposureTime2	3D Parameters - <i>Exposure Time 2</i>
Scan3DExposureTime3	3D Parameters - <i>Exposure Time 3</i>
Scan3DGain	3D Parameters - <i>Gain</i>
FringeCodingMode	3D Parameters - UHP — <i>FringeCodingMode</i>
UhpCaptureMode	3D Parameters - UHP — <i>Capture Mode</i>
AcquisitionMode	 and 
DepthLowerLimit	<i>Depth Range</i>

DepthUpperLimit	
DeviceScanType	-
CloudOutlierFilterMode	Point Cloud Processing - <i>Outlier Removal</i>
CloudSmoothMode	Point Cloud Processing - <i>Point Cloud Smoothing</i>
FringeContrastThreshold	Point Cloud Processing - <i>Stripe Contrast Threshold</i>
FringeMinThreshold	Point Cloud Processing - Minimum Fringe Intensity Threshold
UserSetDefault	-
UserSetSelector	Parameter Group drop-down menu

## 16.2 Use Mech-Eye Viewer to Adjust Camera Parameters for GenICam 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 GenICam-compliant third-party machine vision software (GenICam clients) does not provide visualized tools, you can set these parameters by using the visualized tools provided by Mech-Eye Viewer.

After setting these parameters using Mech-Eye Viewer, you can read the latest camera configuration in a GenICam client directly.

- *Preparation*
- *Adjust Parameters with Visualized Tools*
- *Read Altered Values into GenICam Client*

### 16.2.1 Preparation

Before you adjust parameters in Mech-Eye Viewer, complete the following steps:

1. Save the changes that you made in the GenICam client.
2. Disconnect from the camera in the GenICam client.
3. *Connect to the camera in Mech-Eye Viewer.*
4. In Mech-Eye Viewer's parameter panel on the right, select the parameter group to which you want to save the parameter values.

---

**Note:** A GenICam client can read parameter groups, but not add or delete parameter groups. For instructions on adding/deleting a parameter group, please refer to *Parameter Group*.

---

## 16.2.2 Adjust Parameters with Visualized Tools

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 Mech-Eye Viewer's parameter panel, locate the **2D Parameters** category.
2. From the drop-down menu of **Exposure Mode**, select **Auto**.
3. Double-click *Edit* next to **Auto-Exposure ROI**.
4. You can set an auto-exposure ROI in the pop-up window. For detailed instructions, please refer to [Auto-Exposure ROI](#).

The correspondence between parameters related to **Auto-Exposure ROI** of a GenICam client and Mech-Eye Viewer is as follows:

Parameter in GenICam 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 Mech-Eye Viewer's parameter panel, locate the **Depth Range** category.
2. Double-click *Edit* next to **Depth Range**.
3. You can set the depth range in the pop-up window. For detailed instructions, please refer to [Depth Range](#).

The correspondence between parameters related to **Depth Range** of a GenICam client and 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 Mech-Eye Viewer's parameter panel, locate the **ROI** category.
2. Double-click *Edit* next to **ROI**.
3. You can set an ROI in the pop-up window. For detailed instructions, please refer to [ROI](#).

The correspondence between parameters related to **ROI** of a GenICam client and 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

### 16.2.3 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 Mech-Eye Viewer, click *Save* in the **Parameter Group** category or press **Ctrl+S** on the keyboard to save the configuration changes 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 read the configuration of the parameter group.

## 16.3 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 program [CaptureHalconPointCloud](#). By running this sample program, you can obtain untextured and textured point clouds through Mech-Eye API and convert these data to HALCON-readable format. Then you can read the point clouds into HALCON and perform further processing.

This sample program can be used on Windows and Ubuntu operating system.

---

**Note:** This sample program depends on the HALCON/C++ interface. Please confirm that your HALCON license is still valid before using.

---

### 16.3.1 Use Sample Program on Windows

In order to use this sample program on Windows, the following software must be installed.

- Mech-Eye SDK
- Cmake
- Visual Studio
- HALCON

For instructions on installing these 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.

### 16.3.2 Use Sample Program on Ubuntu

In order to use this sample program on Ubuntu, the following software must be installed.

- Mech-Eye SDK
- Cmake
- PCL
- HALCON

For instructions on installing these 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.

## **TROUBLESHOOTING**

This topic provides troubleshooting guidance for commonly encountered problems.

### **17.1 Camera Not Found in Mech-Eye SDK**

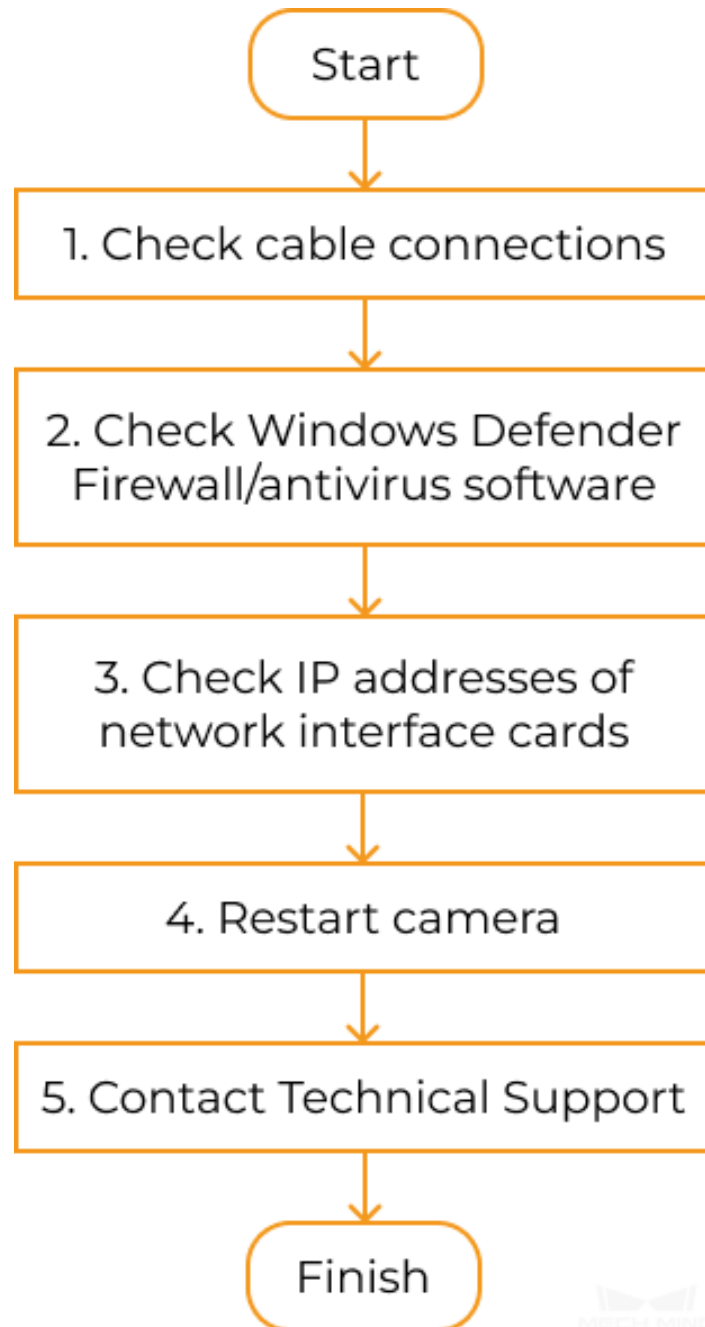
Problem description: In Mech-Eye SDK, the camera that you want to connect does not show up in the list of available cameras.

This topic provides the possible causes, troubleshooting guidelines and detailed troubleshooting procedures for this problem.

#### **17.1.1 Possible Causes**

- Some of the cables are not properly connected.
- Windows Defender Firewall or antivirus software prevents the communication of Mech-Eye SDK.
- Multiple network interface cards of the IPC have IP addresses in the same subnet.
- Other causes.

### 17.1.2 Troubleshooting Guidelines



When the camera that you want to connect does not show up, follow these guidelines to troubleshoot:

1. Make sure that *the cables are properly connected*. If the camera still does not show up in Mech-Eye SDK, proceed to the next step.
2. Make sure that *Windows Defender Firewall and/or antivirus software is not preventing the communication*. 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 network interface cards are not in the same subnet*. If the camera still does not show up in Mech-Eye SDK, proceed to the next step.
4. *Restart the camera*. If the camera still does not show up in Mech-Eye SDK, please contact Technical Support.

### 17.1.3 Troubleshooting Procedures

#### Check Cable Connections

Check if the cables listed below are properly connected on both ends. If a cable is not properly connected, unplug it on both ends and plug it in again. Then, open Mech-Eye SDK and see if the camera shows up.

For instructions on connecting the power cable and Ethernet cable to the camera, please refer to [Connect the Camera](#).

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 <i>blinking or solid green</i> .
IPC	Ethernet cable	The indicator light of the Ethernet port is lit.
Router/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.

---

#### Hint:

- If the camera and IPC are connected directly, you do not need to check for the cable connection of the router/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.
- 

#### Check Windows Defender Firewall and Antivirus Software

Windows Defender Firewall and antivirus software may prevent the communication between Mech-Eye Viewer and the camera.

- Check antivirus software: If an antivirus software is running, please turn it off.
- Configure Windows Defender Firewall: You can turn off Windows Defender Firewall completely, or unblock Mech-Eye Viewer in Windows Defender Firewall.
  - Turn off Windows Defender Firewall:
    1. Click the magnifying glass icon in the taskbar, and search for **Windows Defender Firewall**.

2. Click **Windows Defender Firewall** in the search results to open it, and click **Turn Windows Defender Firewall on or off** on the left.
  3. Select **Turn off Windows Defender Firewall** for both settings, and then click *OK* at the bottom.
- Unblock Mech-Eye Viewer in Windows Defender Firewall:
1. Click the magnifying glass icon in the taskbar, and search for **Windows Defender Firewall**.
  2. Click **Windows Defender Firewall** in the search results to open it, and click **Allow an app or feature through Windows Defender Firewall** on the left.
  3. Click *Change settings* at the top and then click *Allow another app...* at the bottom.
  4. In the pop-up window, click *Browse*, navigate to the installation path of Mech-Eye Viewer, and select **Mech-Eye Viewer.exe**.
  5. Click *Add* to unblock Mech-Eye Viewer.

After performing the above, open Mech-Eye SDK and see if the camera shows up.

### IP Addresses of Multiple Network Interface Cards in the Same Subnet

1. Go to *Control Panel* → *Network and Internet* → *Network and Sharing Center* → *Change adapter settings*.
2. Disable network interface cards not connected to the camera: Right-click a network device and select **Disable**. Repeat until all network interface cards not connected to the camera are disabled.
3. Set the IP address of the network interface card connected to the camera: Make sure it is in the same subnet as the IP address of the camera. Please refer to [Connect to Camera and Set IP Address](#) for instructions.
4. Open Mech-Eye SDK and see if the camera shows up:
  - If the camera shows up in the list of available cameras: one or more of the disabled network interface cards have IP addresses in the same subnet as the card connected to the camera.
    - If you do not need to use the other network interface cards, you can keep them disabled.
    - Otherwise, enable the other cards one at a time and search for the camera in Mech-Eye SDK each time. The camera will not show up when the network interface card with the problematic IP address is enabled. Please change the IP address of this network interface card to a different subnet.
  - If the camera still does not show up in the list of available cameras, proceed to the next step.
5. Disable the network interface card connected to the camera and enable it again. Then, open Mech-Eye SDK and see if the camera shows up.
  - If the camera still does not show up in the list of available cameras, proceed to the next step.
6. Connect the camera to a different Ethernet port. Then, open Mech-Eye SDK and see if the camera shows up.

### Restart the Camera

1. Unplug the power cable from the camera.
2. Wait for about 20 seconds, and replug the power cable.
3. Open Mech-Eye SDK and see if the camera shows up. It may take about 2 minutes after restarting for the camera to show up in the list of available cameras.

### Contact Technical Support

If the camera still does not show up in Mech-Eye SDK after all the above procedures have been conducted, please contact Technical Support.

## 17.2 Camera Cannot Be Connected in Mech-Eye SDK (Unreachable)

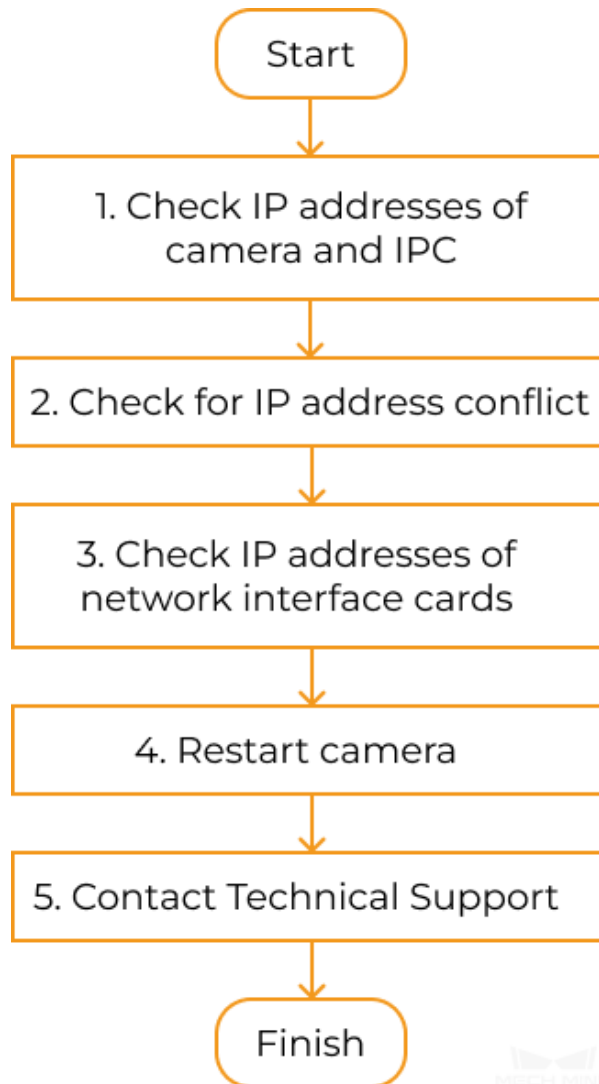
Problem description: The camera shows up in Mech-Eye SDK but cannot be connected. The status of the IP address displayed in the camera info card in Mech-Eye Viewer is **Unreachable**.

This topic provides the possible causes, troubleshooting guidelines and detailed troubleshooting procedures for this problem.

### 17.2.1 Possible Causes

- The IP addresses of the camera and IPC are not in the same subnet.
- Multiple network interface cards of the IPC have identical IP addresses.
- Multiple network interface cards of the IPC have IP addresses in the same subnet.
- Other causes.

## 17.2.2 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 IPC are in the same subnet*. If the camera still cannot be connected, proceed to the next step.
2. Make sure that *no IP address conflict presents*. If the camera still cannot be connected, proceed to the next step.
3. Make sure that *the IP addresses of multiple network interface cards are not in the same subnet*. If the camera still cannot be connected, proceed to the next step.
4. *Restart the camera*. If the camera still cannot be connected, please contact Technical Support.

## 17.2.3 Troubleshooting Procedures

### Check IP Addresses of Camera and IPC

1. Check the IP addresses of the camera and IPC. For instructions, please refer to [Connect to Camera and Set IP Address](#).
2. If the two IP address are in different subnets, please follow the instructions in the link and modify the addresses to the same subnet.
3. Connect to the camera in Mech-Eye SDK.

**Attention:** Please make sure that the two IP addresses are not identical. If the camera and IPC have identical IP addresses, the camera's status is displayed as **Local** and cannot be connected.

### Check for IP Address Conflict

1. Disconnect all devices other than the camera from the IPC, router and/or network switch.
2. Open Mech-Eye SDK and connect to the camera.
  - 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 Mech-Eye SDK 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.
  - If the camera still cannot be connected, proceed to the next step.
3. Disable the network interface card connected to the camera and enable it again. Then, open Mech-Eye SDK and connect to the camera.
  - If the camera still cannot be connected, proceed to the next step.
4. Connect the camera to a different Ethernet port. Then, open Mech-Eye SDK and connect to the camera.

### IP Addresses of Multiple Network Interface Cards in the Same Subnet

1. Go to *Control Panel* → *Network and Internet* → *Network and Sharing Center* → *Change adapter settings*.
2. Disable network interface cards not connected to the camera: Right-click a network device and select **Disable**. Repeat until all network interface cards not connected to the camera are disabled.
3. Set the IP address of the network interface card connected to the camera: Make sure it is in the same subnet as the IP address of the camera. Please refer to [Connect to Camera and Set IP Address](#) for instructions.
4. Open Mech-Eye SDK and connect to the camera:

- If the camera is successfully connected: one or more of the disabled network interface card have IP addresses in the same subnet as the card connected to the camera.
    - If you do not need to use the other network interface cards, you can keep them disabled.
    - Otherwise, enable the other cards one at a time and connect to the camera in Mech-Eye SDK each time. The camera connection will fail when the network interface card with the problematic IP address is enabled. Please change the IP address of this network interface card to a different subnet.
  - If the camera still cannot be connected, proceed to the next step.
5. Disable the network interface card connected to the camera and enable it again. Then, open Mech-Eye SDK and connect to the camera.
    - If the camera still cannot be connected, proceed to the next step.
  6. Connect the camera to a different Ethernet port. Then, open Mech-Eye SDK and connect to the camera.

### Restart the Camera

1. Unplug the power cable from the camera.
2. Wait for about 20 seconds, and replug the power cable.
3. Open Mech-Eye SDK and connect to the camera. It may take about 2 minutes after restarting for the camera to show up in the list of available cameras.

### Contact Technical Support

If the camera still cannot be connected in Mech-Eye SDK after all the above procedures have been conducted, please contact Technical Support.

## 17.3 Camera Cannot Be Connected in Mech-Eye SDK (Windows Server)

Problem description: 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.

### 17.3.1 Possible Cause

This is a bug in Mech-Eye SDK 1.6.1 and below.

### 17.3.2 Troubleshooting Procedure

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.

## 17.4 Enable Jumbo Frames (GenICam)

When a GenICam client is used to perform image capturing, the data transmission may be slow depending on your network conditions. Enabling jumbo frames on the IPC can solve the problem.

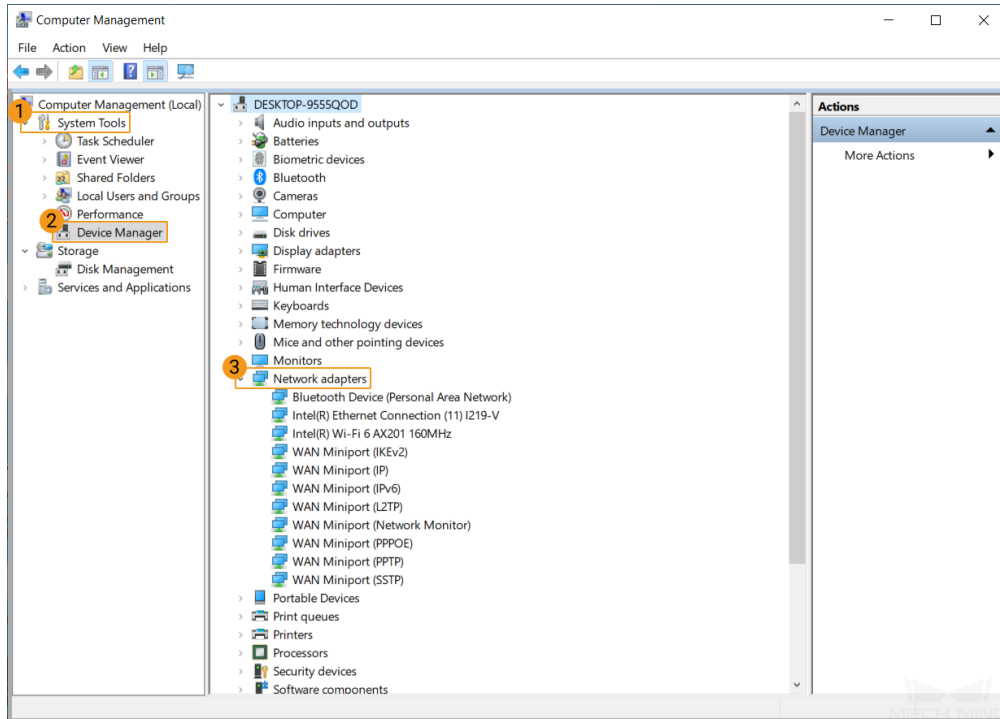
---

**Note:** Jumbo frames are Ethernet frames that can carry up to 9000 bytes of data. Standard Ethernet frames carry up to 1500 bytes of data. Using jumbo frames can reduce CPU processing load and speed up data transmission.

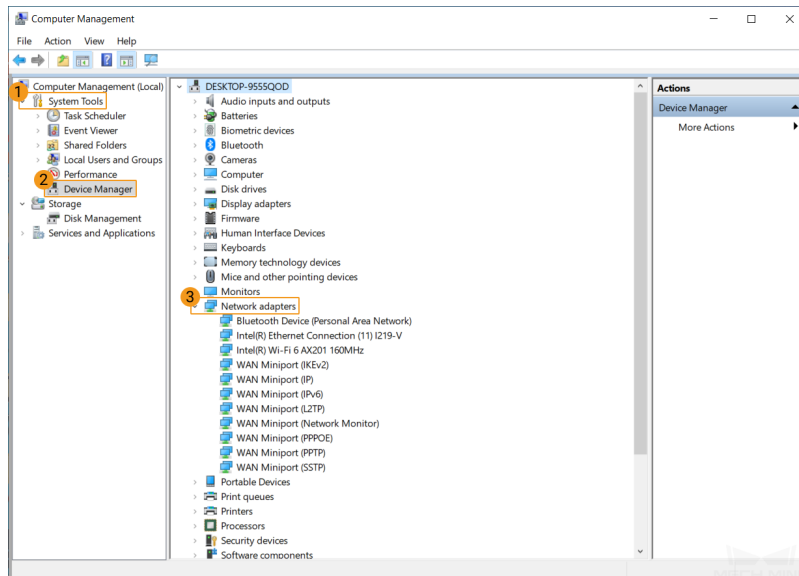
---

### 17.4.1 Instructions

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**. Click **OK** to save the change.





## EXTENDED READING

Here you can find helpful and informative topics that might aid you with using our product.

---

A brief introduction of the **imaging mechanism of structured light cameras**:

*Structured-Light 3D Scanning Mechanism*

### 18.1 Structured-Light 3D Scanning Mechanism

As a structured-light 3D scanner, Mech-Eye Industrial 3D Camera (“3D camera” for short) 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 objects.

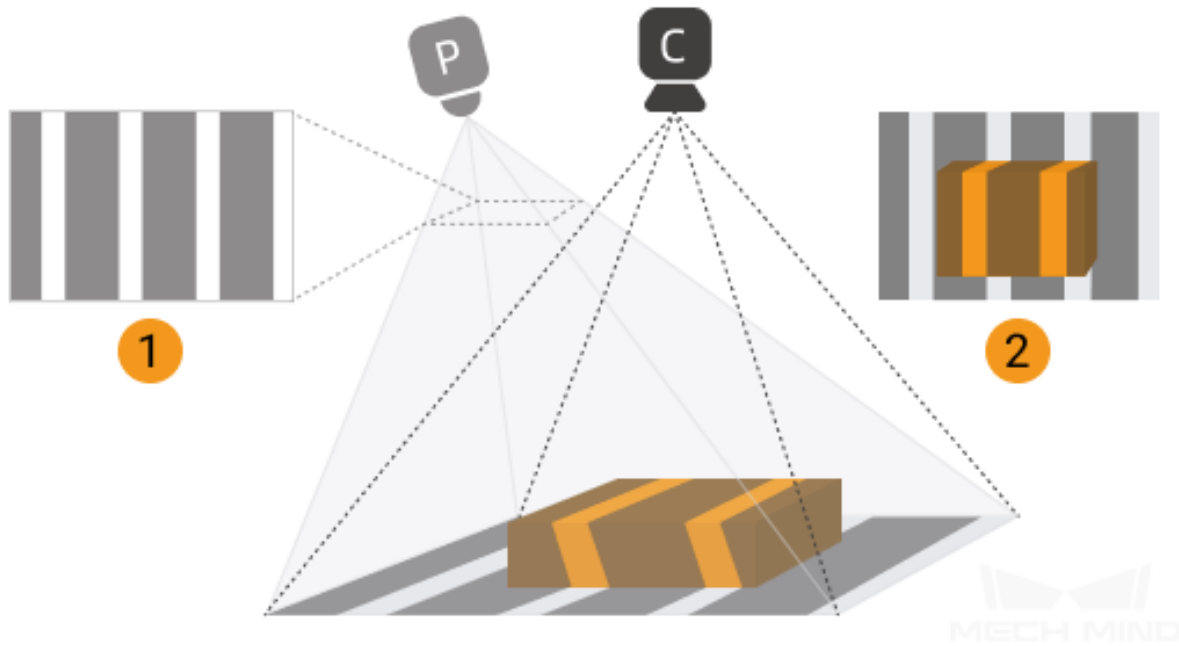
The entire scanning process can be divided into three phases:

- *Image Capturing*
- *Calculation and Processing*
- *Output*

#### 18.1.1 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.



### 18.1.2 Calculation and Processing

The 3D camera calculates depth data based on the deformation of the stripe patterns. Using the depth data, the 3D camera then generates the depth map and point cloud.

### 18.1.3 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!

---

**Note:**

- 2D image: the image captured without structured light projection.
  - Depth map: an image containing depth information.
  - Point cloud: a collection of data points that represent the 3D shape of the object surface.
-