Mech-Eye Industrial 3D Cameras

Mech-Mind

Jun 28, 2023

GETTING STARTED

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 GenICam 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

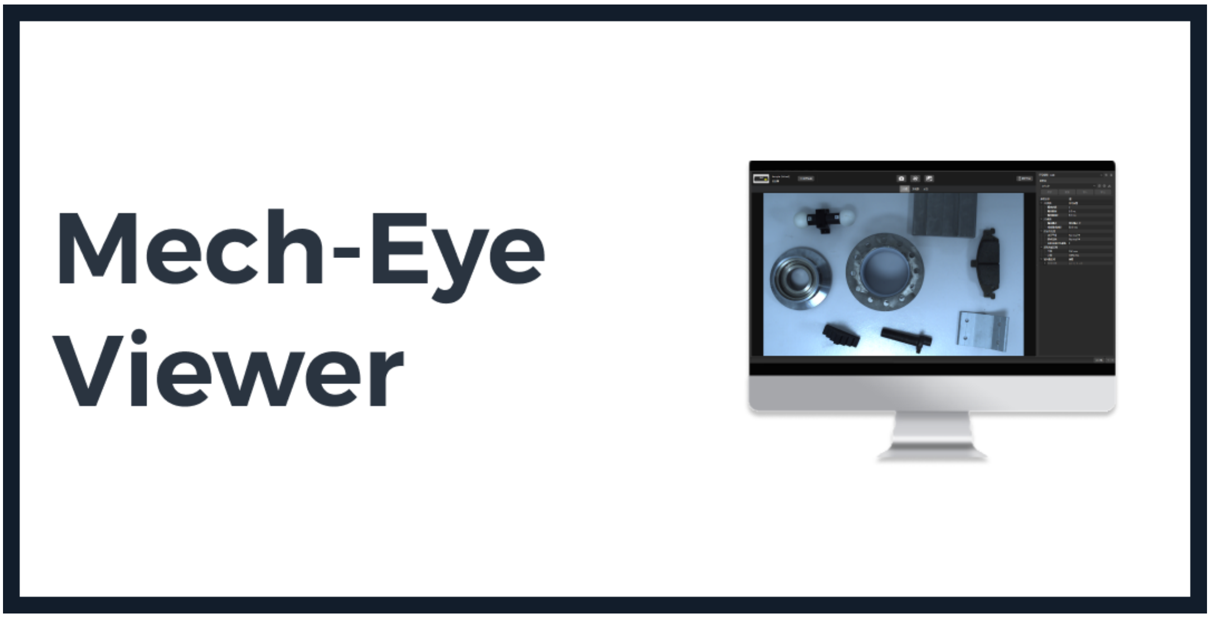
Mech-Eye SDK 2.1 Release Notes

Mech-Eye SDK Installation Guide

Resources

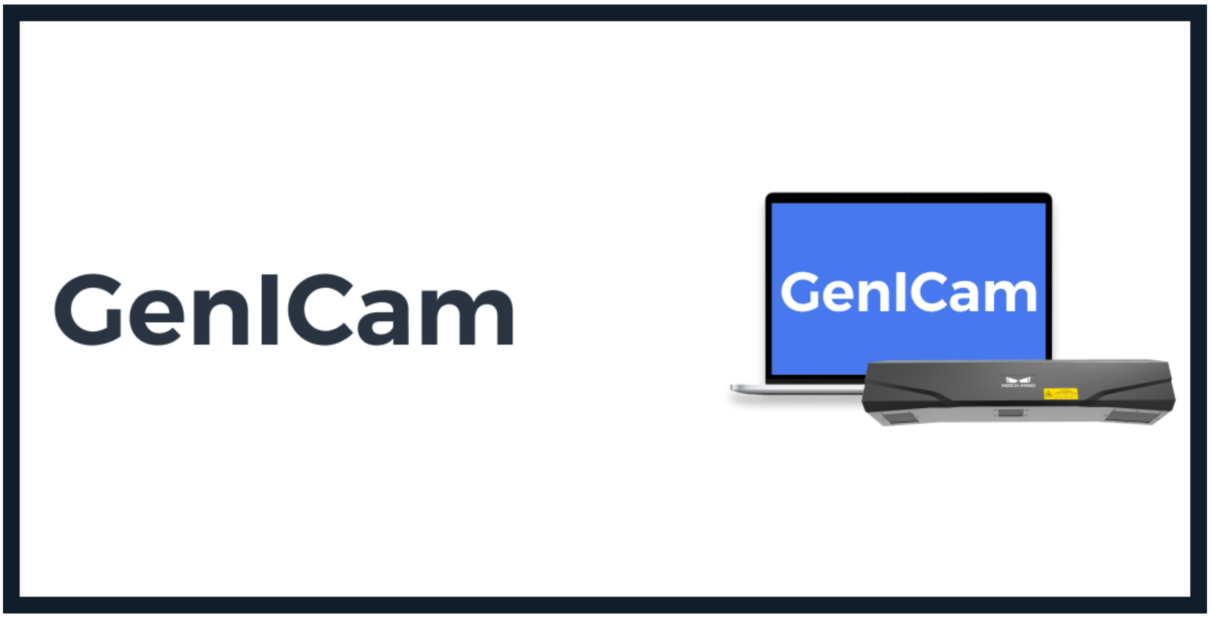
Mech-Eye Industrial 3D Camera User Manual

Mech-Eye Viewer Quick Start Guide



Mech-Eye Viewer User Manual

Mech-Eye API User Manual



GenICam

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.

CHAPTER

ONE

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 / 1024 × 768 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	2048 × 1536
Laser L Enhanced		Binocular	4096 × 3000

Hint: For technical specifications, please refer to *Technical Specifications*.

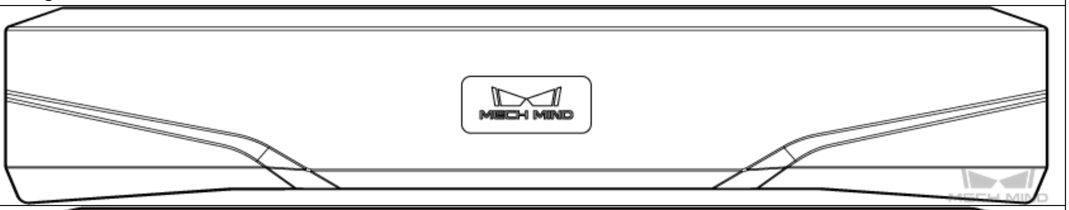
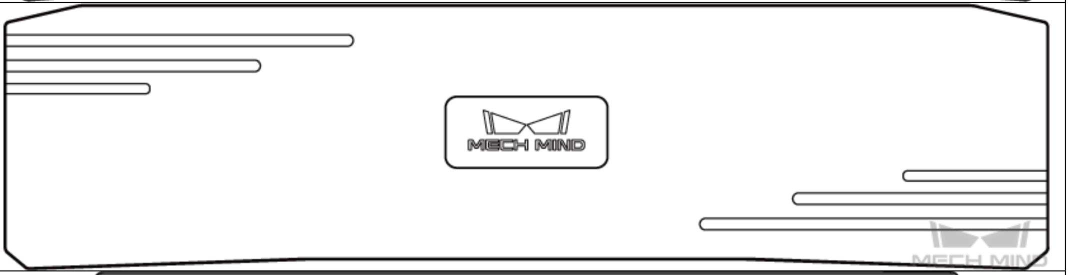
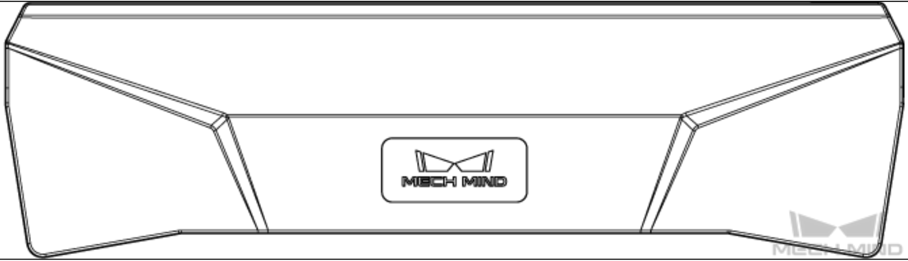
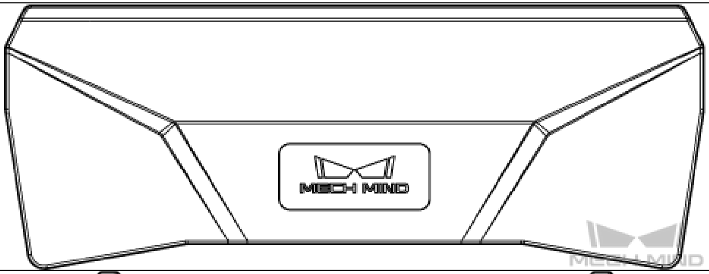
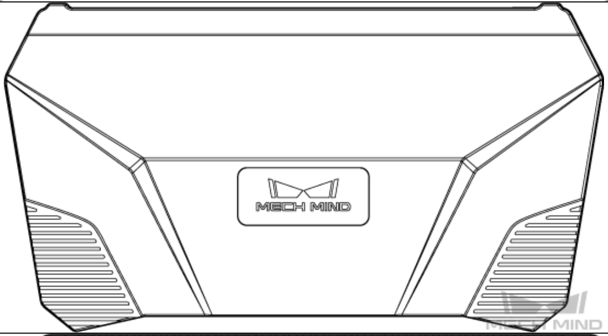
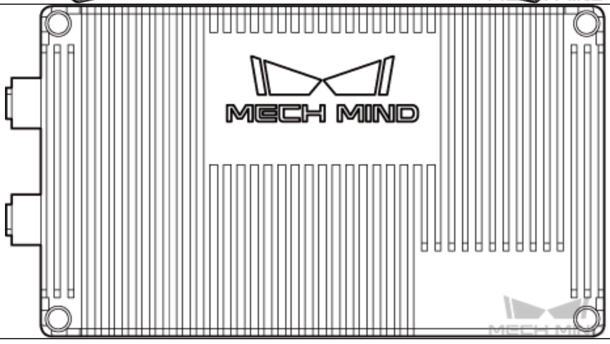
1.1.2 Characteristics and Suitable Applications

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

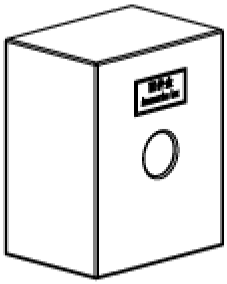
Model	Characteristics	Suitable applications
UHP-140 (V4)	• Micron-level accuracy • Independently developed image fusion algorithm • Handles reflective objects well	• Automotive part manufacturing and assembly • Position error, gap, and flush inspection
LSR L (V4)	• High accuracy • Large field of view • High resistance to ambient light	• Manufacturing plants and other environments with ambient light interference
Laser L (V3)		
Laser L Enhanced (V3)		
NANO (V4)	• Small volume • High accuracy • High resistance to ambient light	• On-robot mounting • Positioning and assembly, high-precision picking, and other high-accuracy applications
Nano (V3)		
Pro XS (V3)		
DEEP (V4)	• Large field of view and deep depth of field • Fast data acquisition • Comprehensive, detailed, precise and color-accurate point cloud	• Logisitics applications such as palletizing and depalletizing
Deep (V3)		
Pro L Enhanced (V3)		
PRO S (V4)	• High accuracy • Fast data acquisition • High resistance to ambient light • Color 2D camera variant available	• Medium-range applications with high accuracy requirements • Such as random picking, positioning, assembly, academic research
Pro S Enhanced (V3)		
PRO M (V4)		
Pro M Enhanced (V3)		
Log M (V3)	• Fast data acquisition • Suited for medium to long range	• Designed for logisitics applications • Suited for product picking and parcel feeding
Log S (V3)	• Fast data acquisition • Suited for short to medium range	• Designed for logisitics applications • Suited for product picking and parcel feeding

1.2 Package Contents

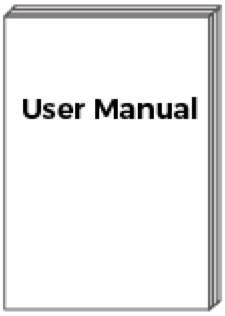
Camera

Model	Diagram
LSR L	
DEEP	
PRO M	
PRO S	
UHP- 140	
NANO	

Accessory Box



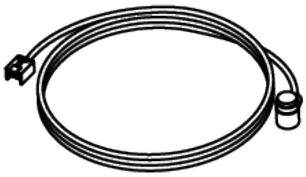
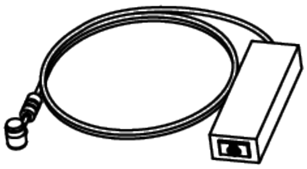
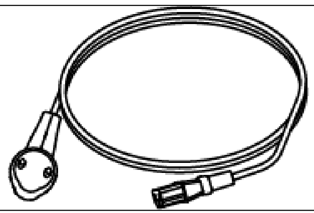
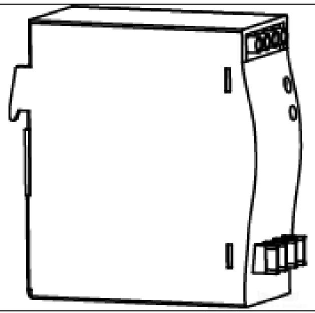
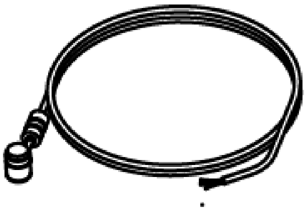
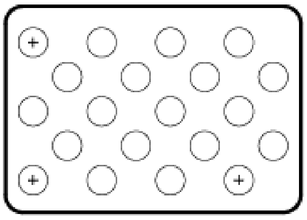
User Manual



Hint:

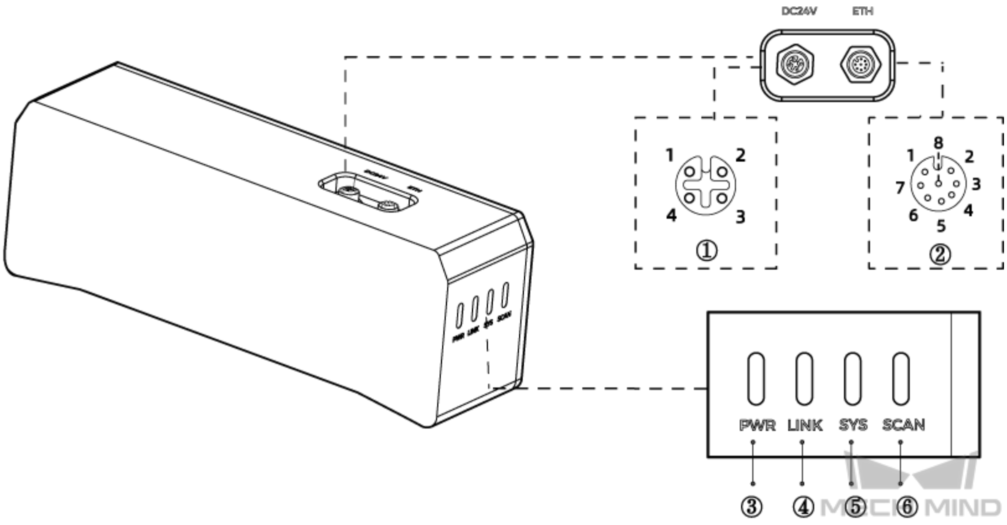
- The camera bracket is attached to the back of the camera when packaged.
- Please contact us if there are any damages or missing parts.

1.2.1 Cables and Accessories

Ethernet Cable	Adapter	AC Power Cable
		
DIN Rail Power Supply	DC Power Cable	Calibration Board
		

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



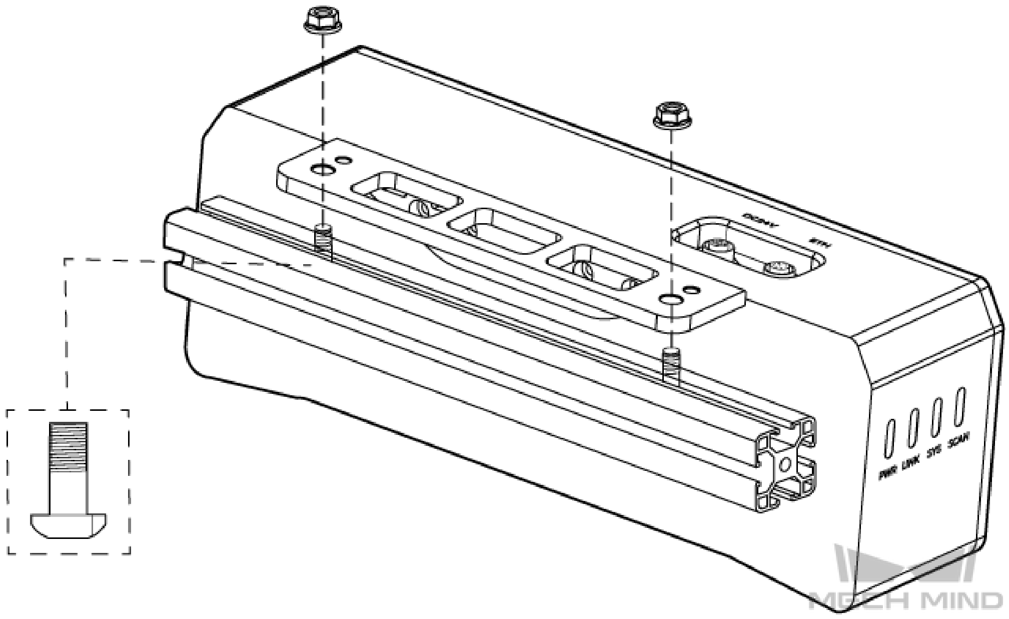
No.	Name	Function
1	DC 24V port	1: GND 2: GND 3: 24 V DC 4: 24 V DC
2	ETH port	1: MD3_P 2: MD2_N 3: MD2_P 4: MD0_P 5: MD1_P 6: MD0_N 7: MD3_N 8: MD1_N
3	PWR indicator light	Off: not connected to power Green: normal voltage Yellow: abnormal voltage, but system is still functioning Red: voltage too high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

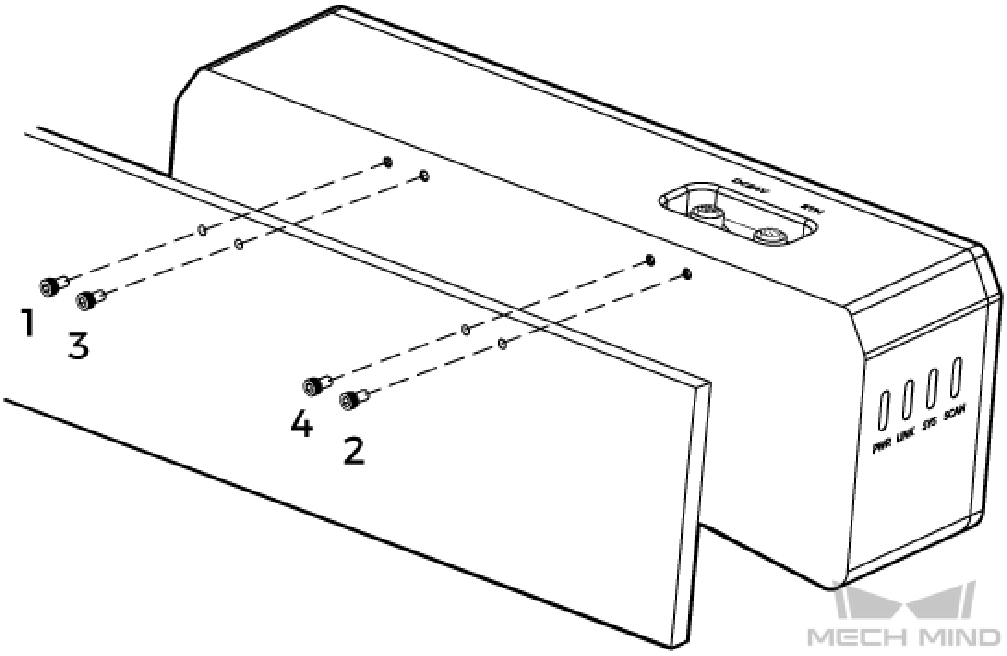


Hint:

- 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.

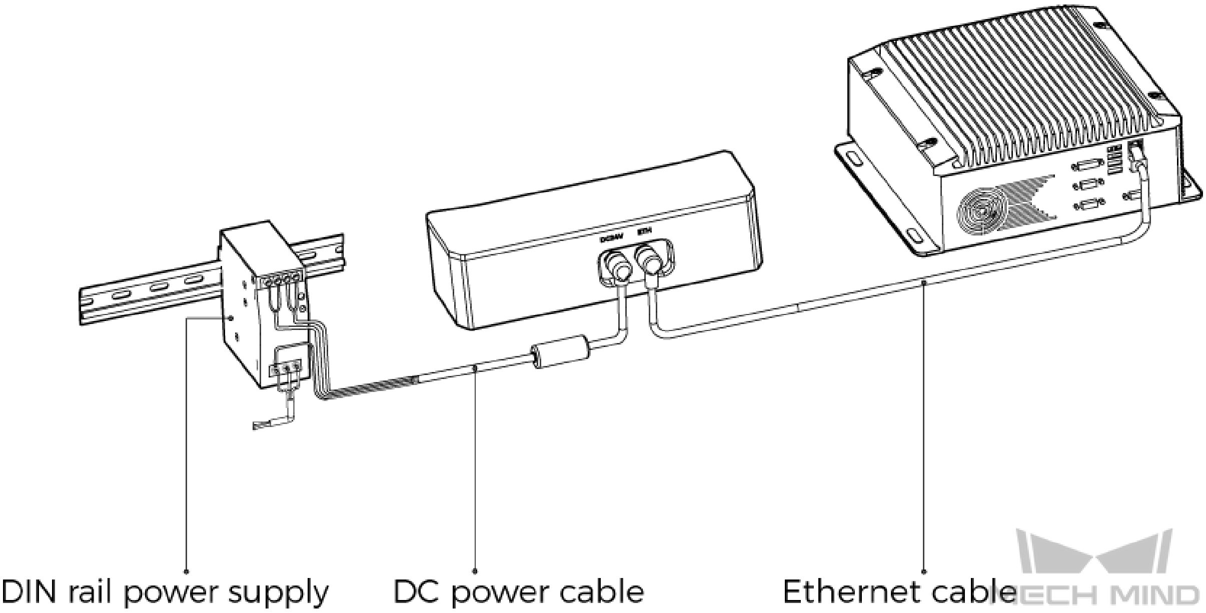
1.5 Connect the Camera

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.

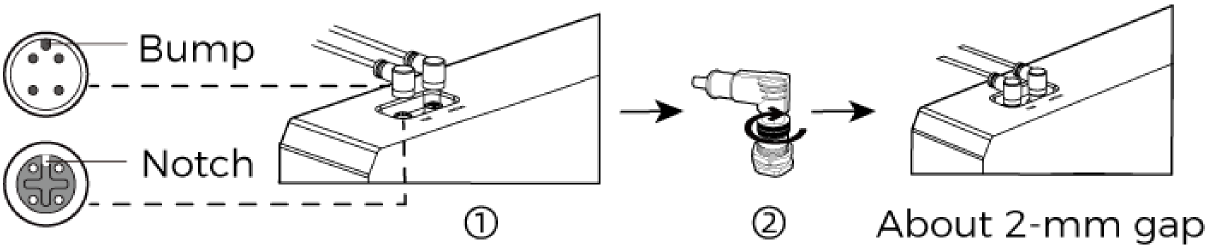
- The DIN rail power supply should be reliably grounded. If using multiple DIN rail power supplies, ensure enough distance in between.
- When mounting the camera onto a robot arm or other mobile equipment, please fasten the DC and Ethernet cables on the camera end properly to avoid damaging the cables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

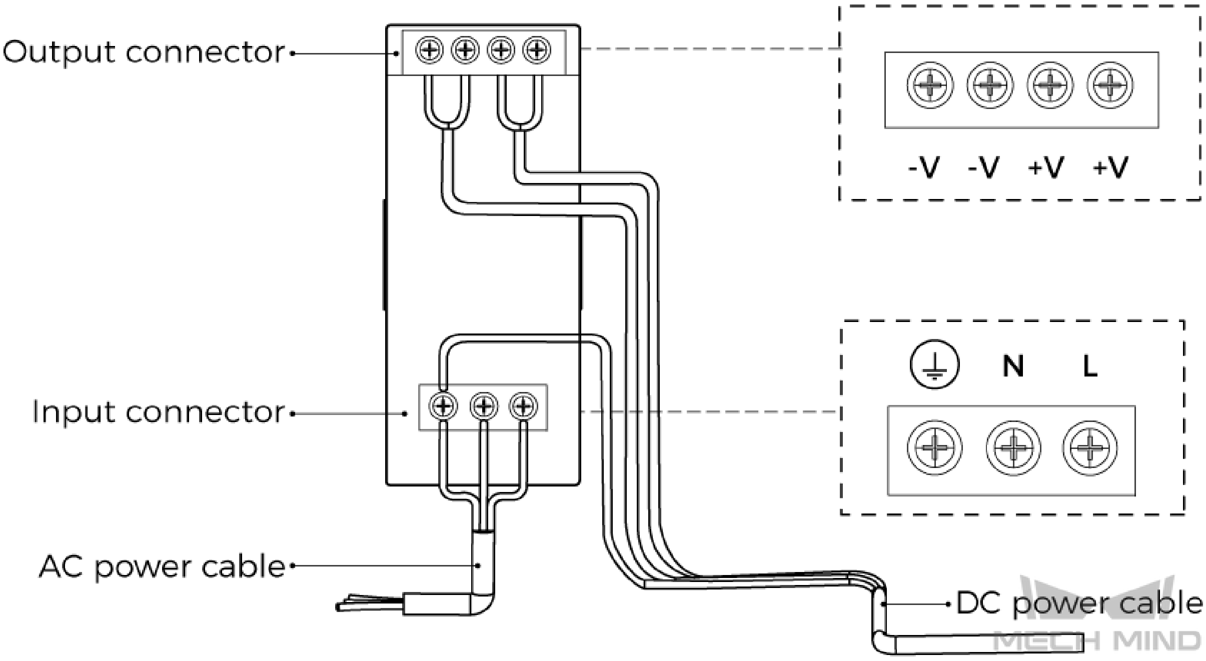


When inserting the Ethernet cable and power cable:

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

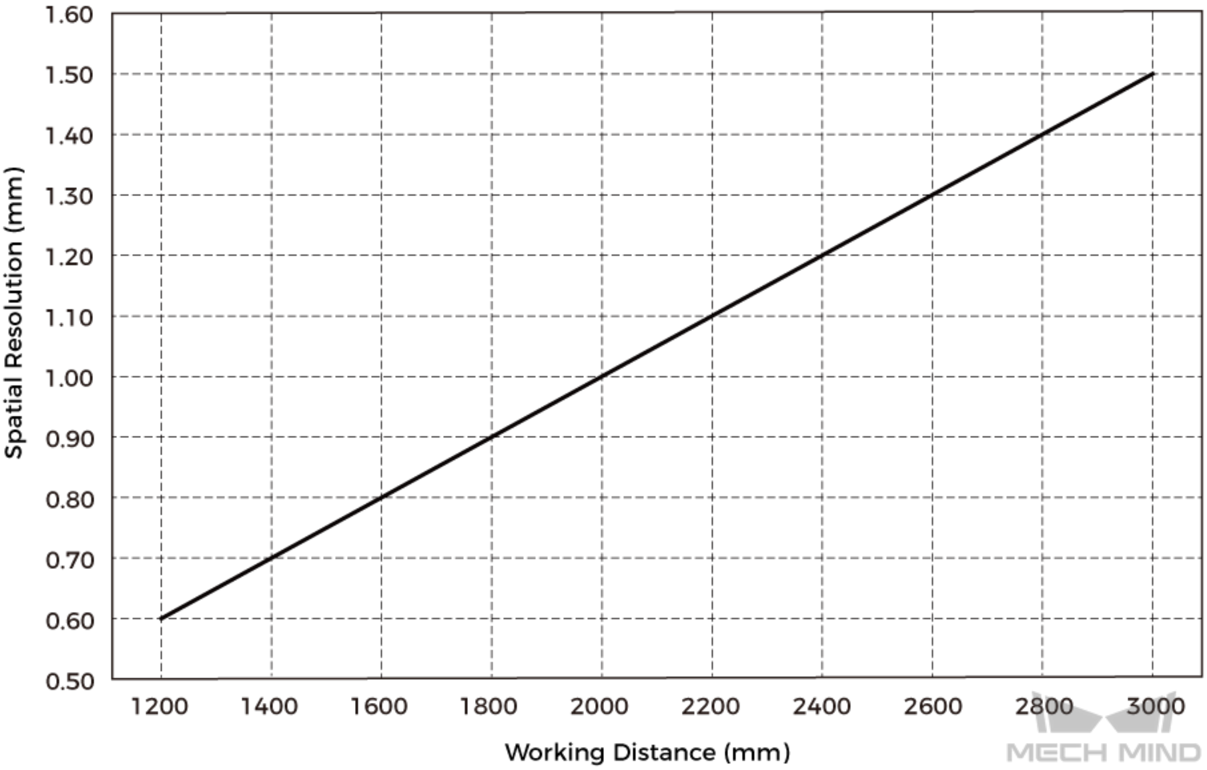
Product Name	Mech-Eye Industrial 3D Camera
Model	LSR L
Recommended Working Distance	1200–3000 mm
FOV	1200 × 1000 mm @ 1.2 m
	3000 × 2400 mm @ 3.0 m
Depth Map Resolution	2048 × 1536
RGB Resolution	4000 × 3000 / 2000 × 1500
Point Z-Value Repeatability (σ) [1]	0.5 mm @ 3 m
Measurement Accuracy (VDI/VDE) [2]	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 [3]	IP65
Cooling	Passive

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

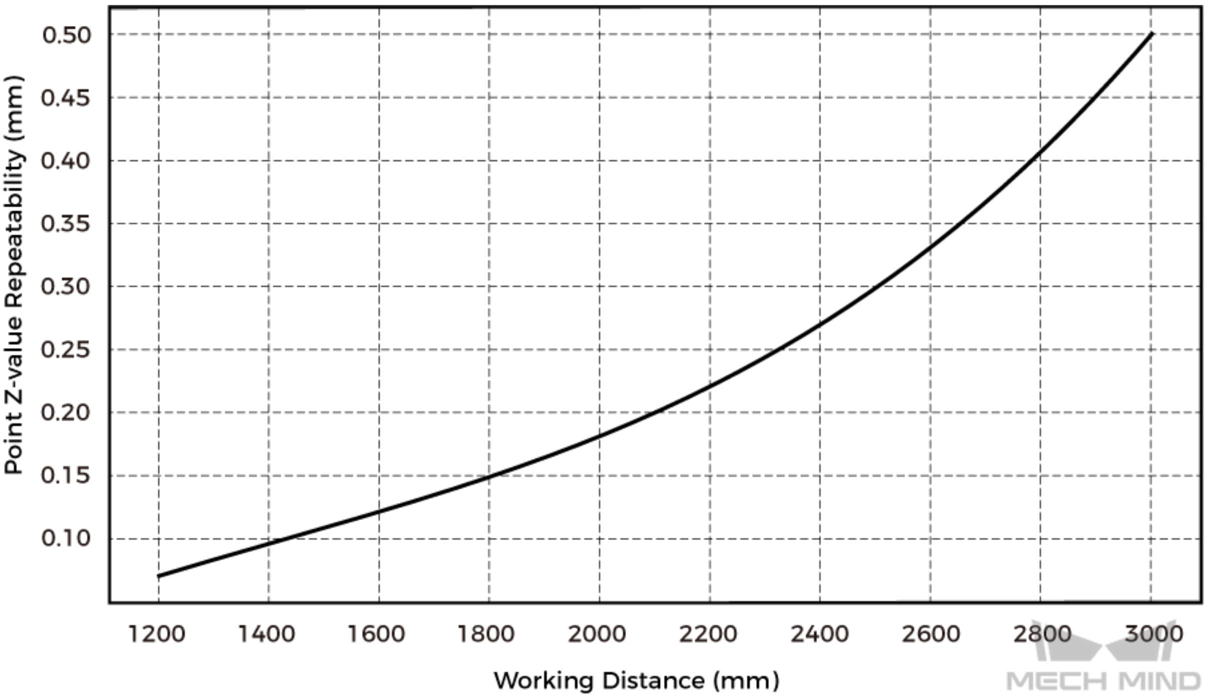
[2] According to VDI/VDE 2634 Part II.

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

Spatial Resolution

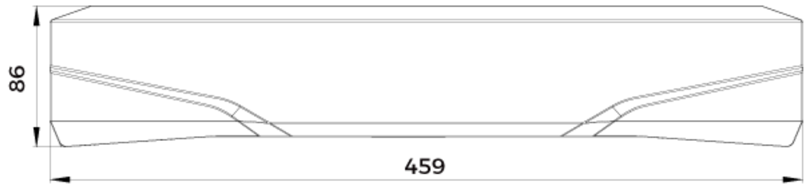


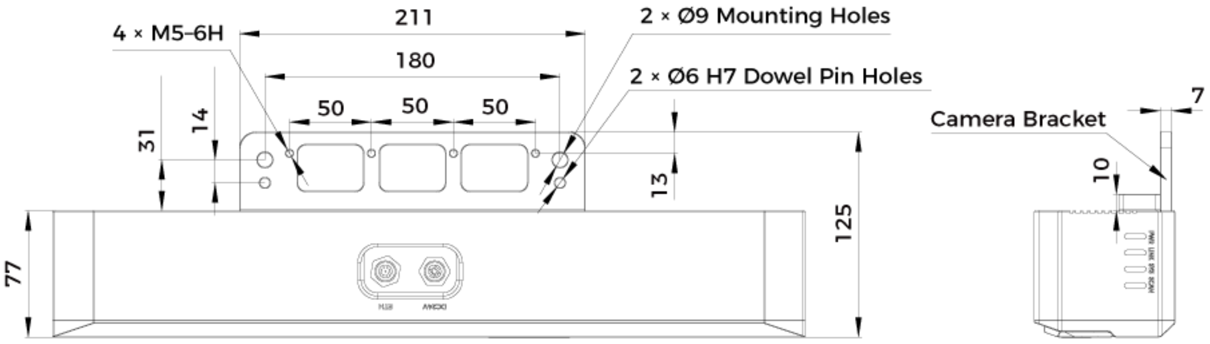
Point Z-Value Repeatability

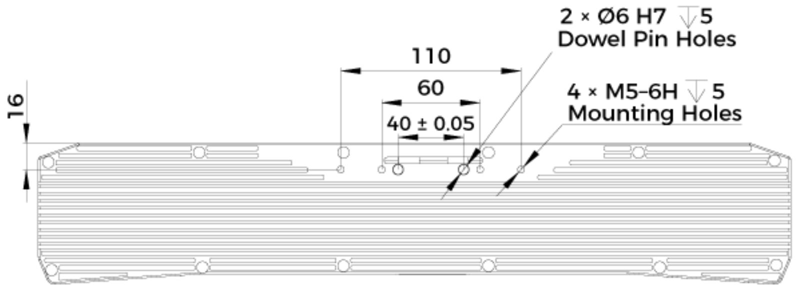


Dimensions

Unit: mm


86
459


4 × M5-6H
211
180
50
50
50
2 × Ø9 Mounting Holes
2 × Ø6 H7 Dowel Pin Holes
31
14
13
125
77
Camera Bracket
7
10


2 × Ø6 H7 ↓5
Dowel Pin Holes
110
60
40 ± 0.05
4 × M5-6H ↓5
Mounting Holes
16

1.6.2 DEEP

Specifications

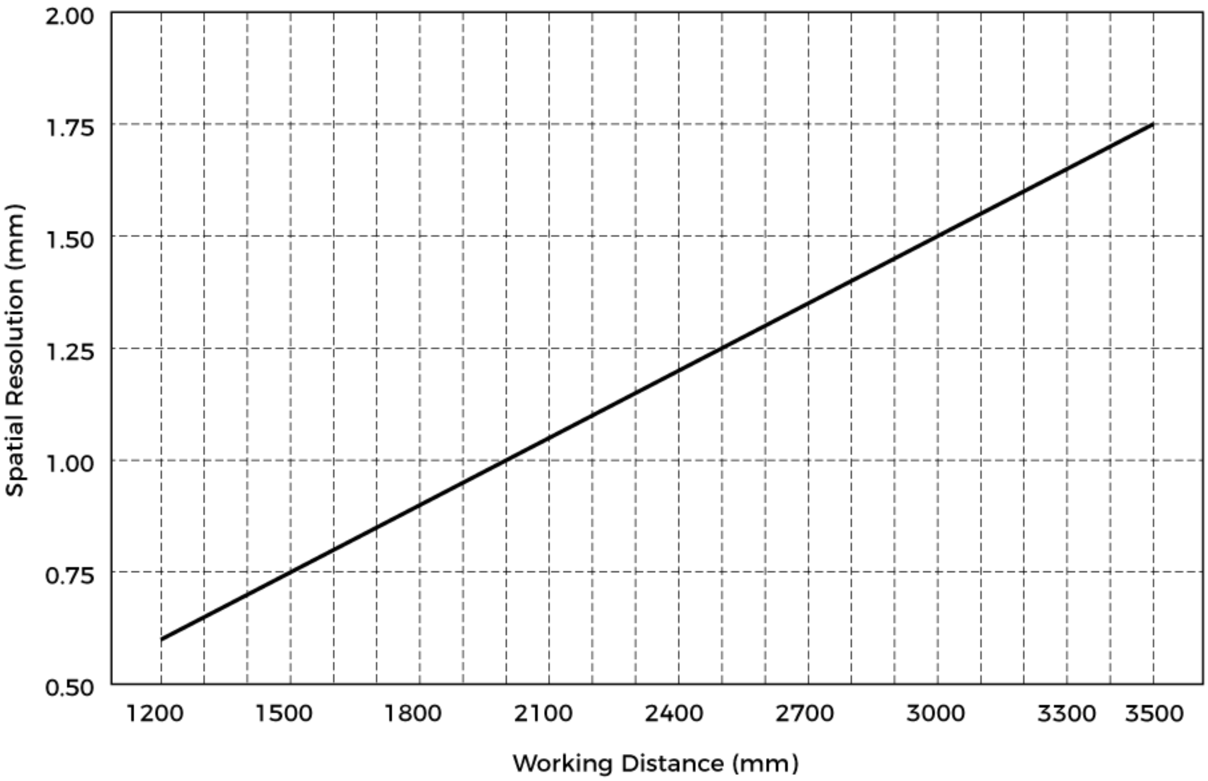
Product Name	Mech-Eye Industrial 3D Camera
Model	DEEP
Recommended Working Distance	1200–3500 mm
FOV	1200 × 1000 mm @ 1.2 m
	3500 × 2800 mm @ 3.5 m
Depth Map Resolution	2048 × 1536 / 1024 × 768
RGB Resolution	2000 × 1500
Point Z-Value Repeatability (σ) [1]	1.0 mm @ 3 m
Measurement Accuracy (VDI/VDE) [2]	3.0 mm @ 3 m
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 [3]	IP65
Cooling	Passive

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

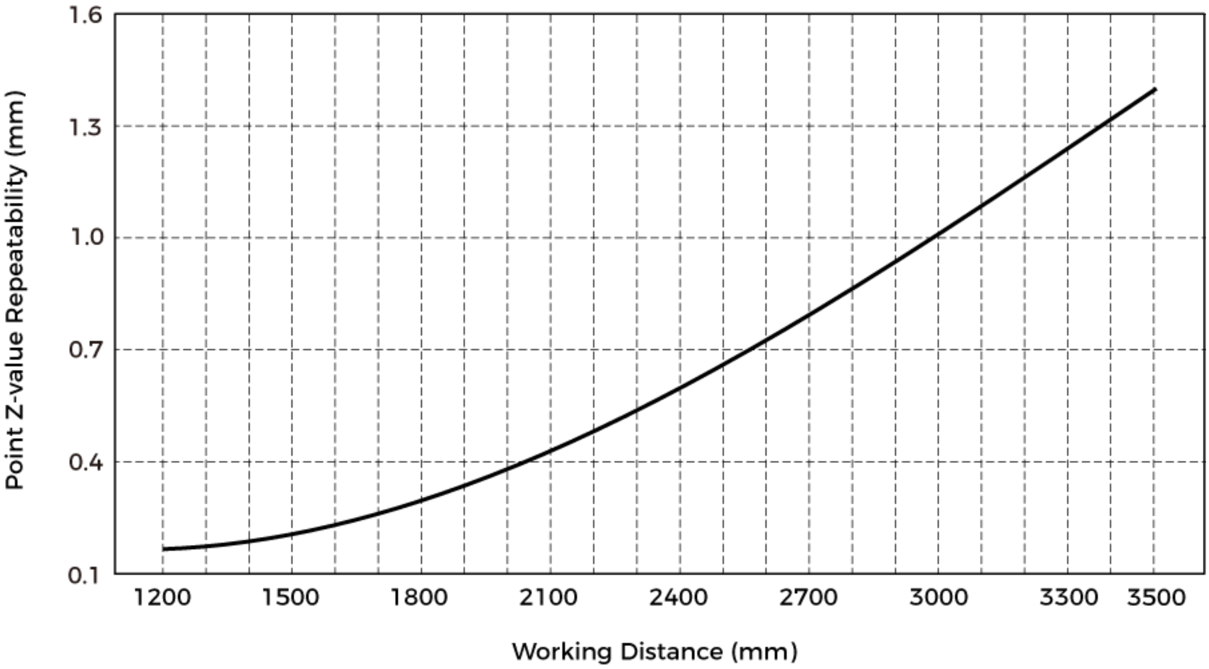
[2] According to VDI/VDE 2634 Part II.

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

Spatial Resolution

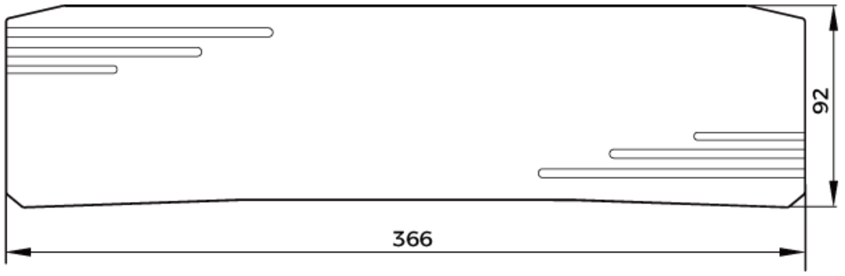


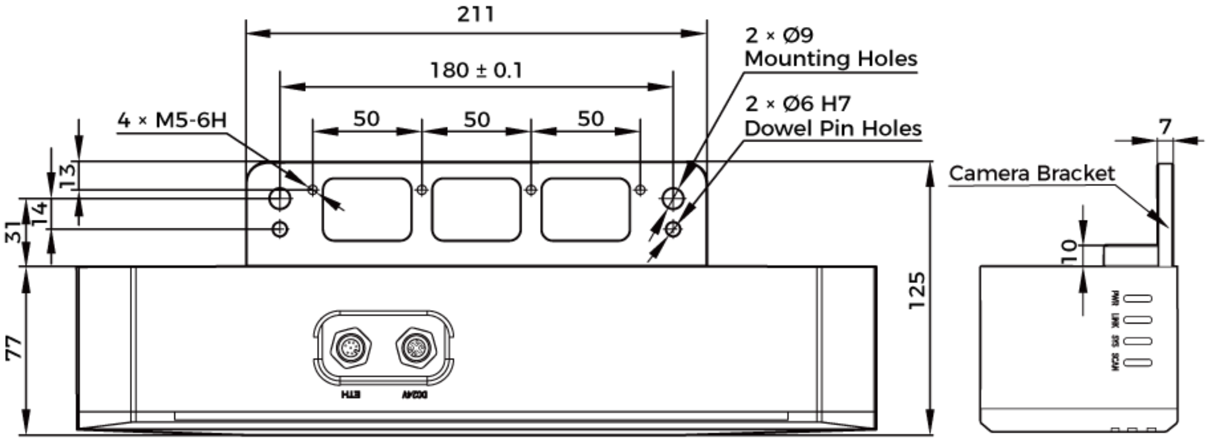
Point Z-Value Repeatability

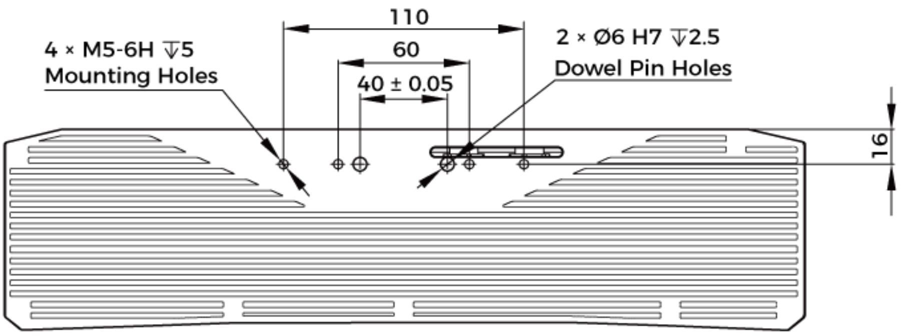


Dimensions

Unit: mm







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 (σ) [1]	0.05 mm @ 1 m	0.2 mm @ 2 m
Measurement Accuracy (VDI/VDE) [2]	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 [3]	IP65	
Cooling	Passive	

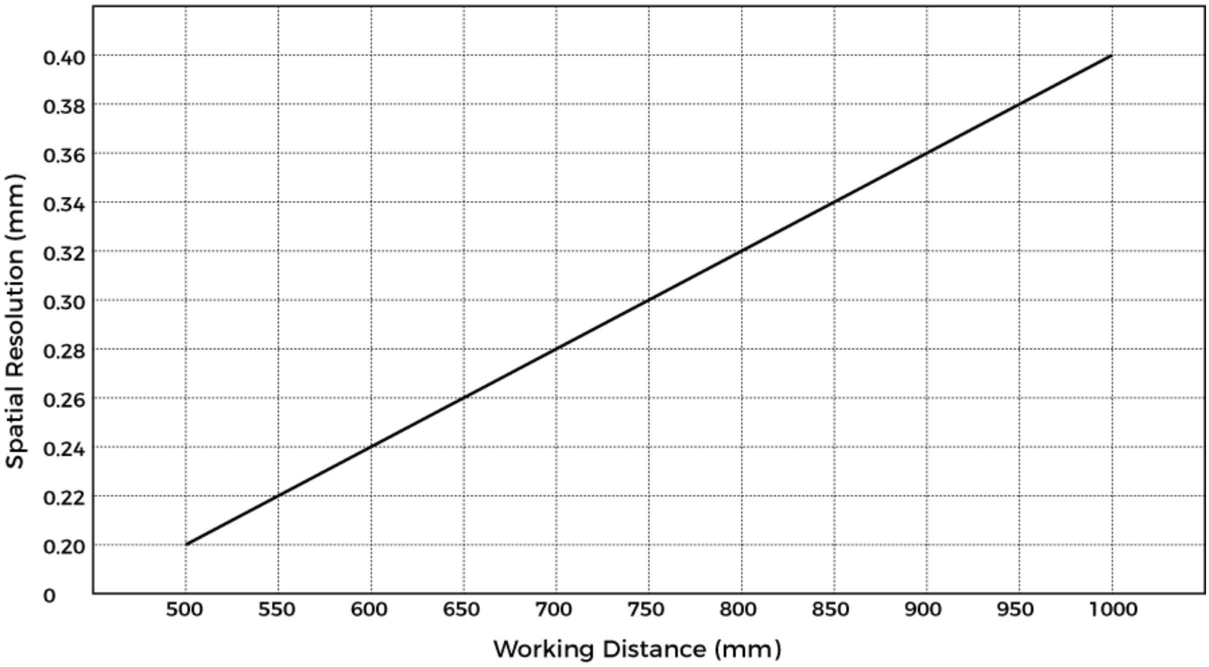
[1] One standard deviation of 100 Z-value measurements of the same point. The measurement target was a ceramic plate.

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

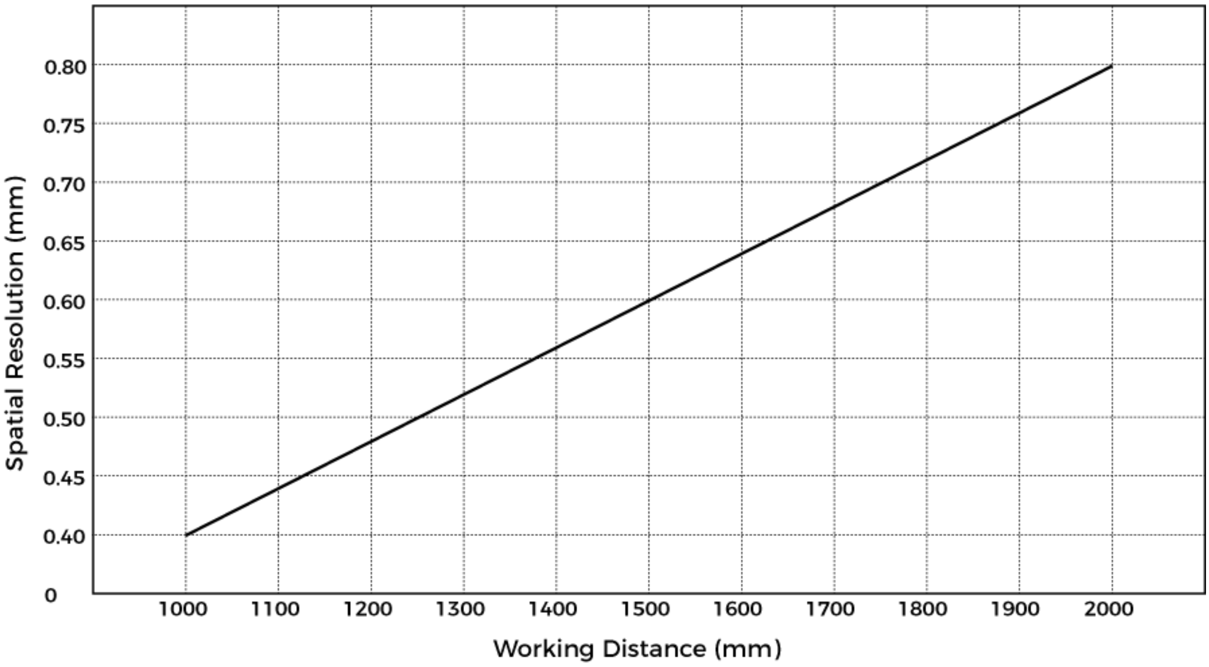
[3] Test implemented based on IEC 60529. 6: dust-tight; 5: protected against water jets.

Spatial Resolution

PRO S

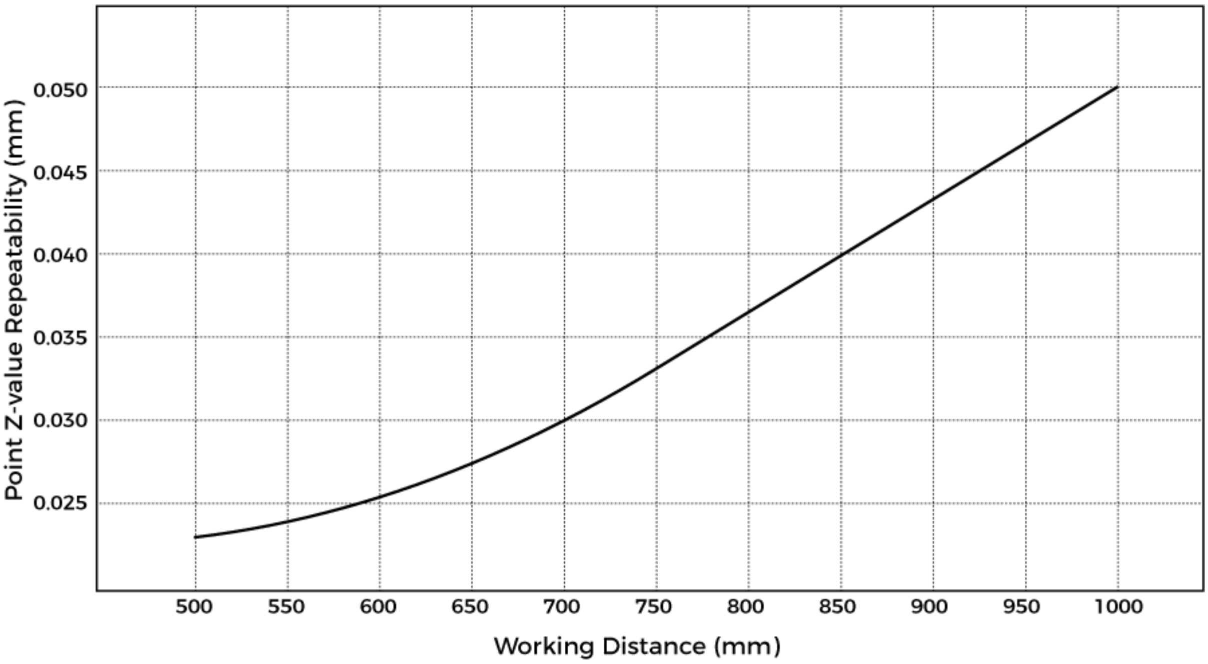


PRO M

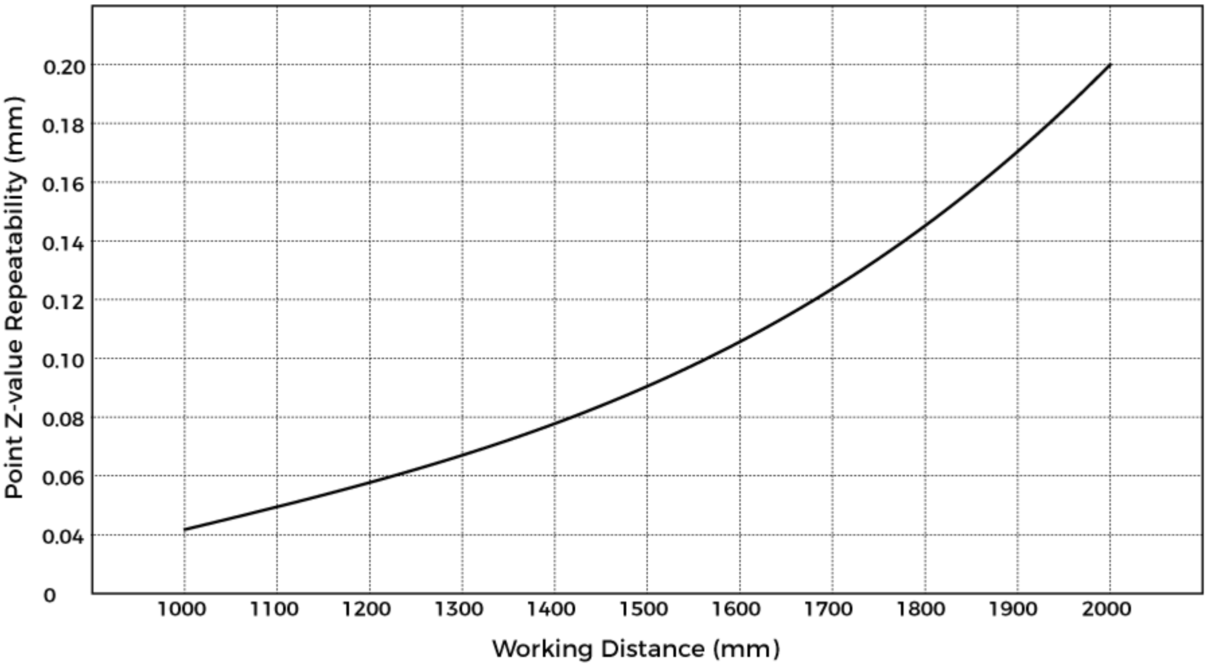


Point Z-Value Repeatability

PRO S



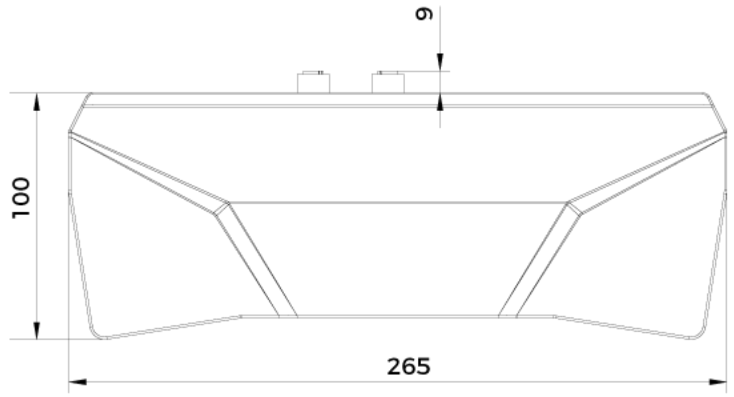
PRO M

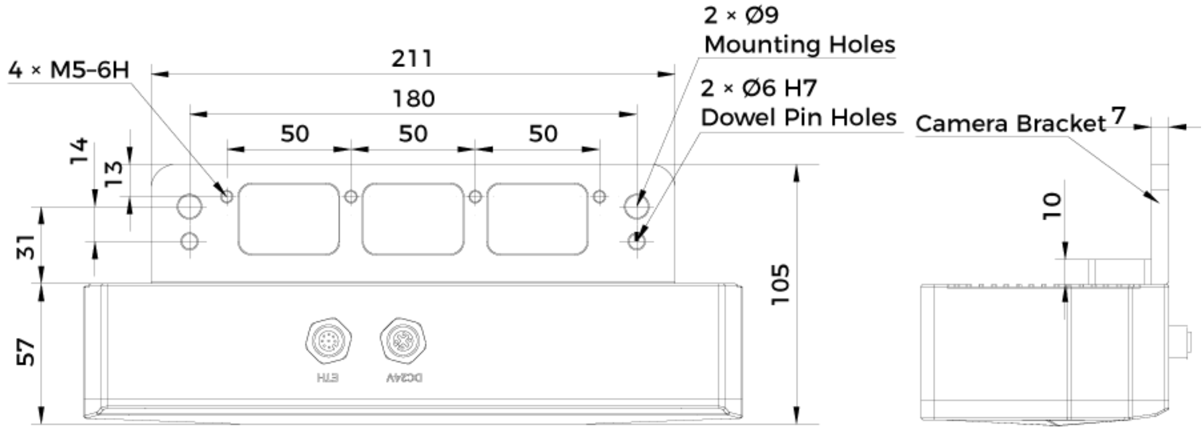


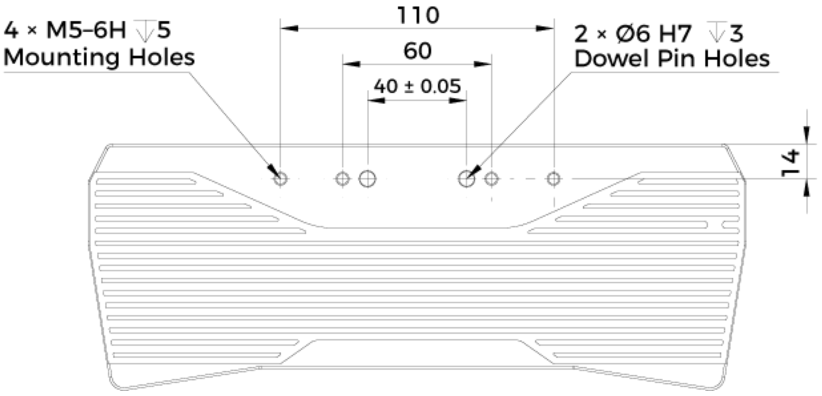
Dimensions

Unit: mm

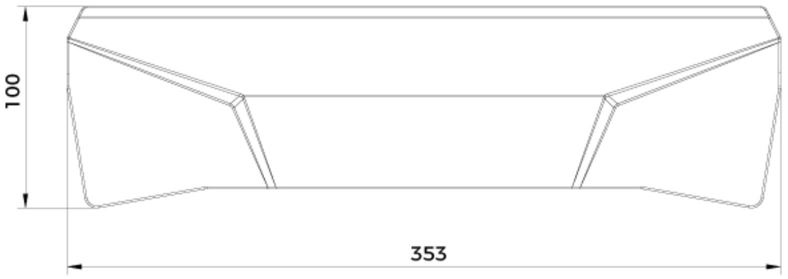
PRO S

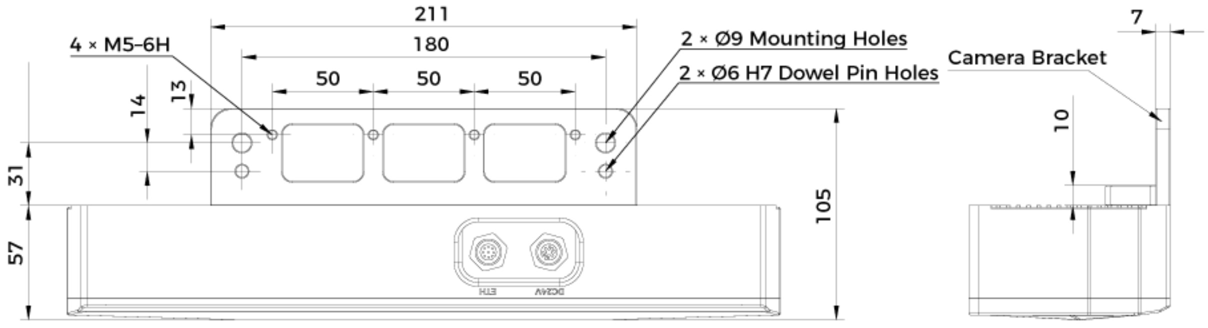

9
100
265

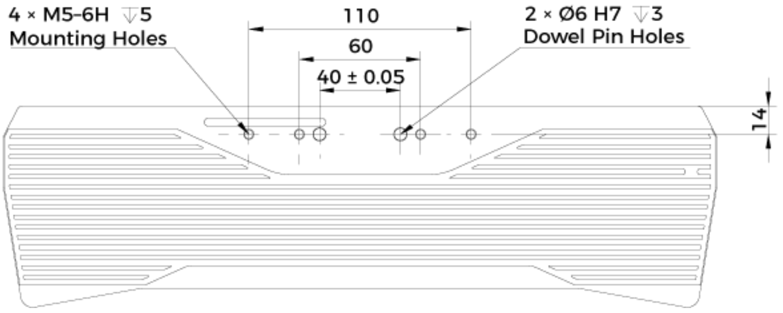

2 × Ø9
Mounting Holes
4 × M5-6H
211
180
50
50
50
2 × Ø6 H7
Dowel Pin Holes
Camera Bracket
7
14
13
31
57
105
10
ETH
DC24V


4 × M5-6H ↧5
Mounting Holes
110
60
40 ± 0.05
2 × Ø6 H7 ↧3
Dowel Pin Holes
14

PRO M


100
353


211
180
50
50
50
4 × M5-6H
2 × Ø9 Mounting Holes
2 × Ø6 H7 Dowel Pin Holes
7
Camera Bracket
10
13
14
31
57
105
ETH
DC24V


4 × M5-6H ↧5
Mounting Holes
110
60
40 ± 0.05
2 × Ø6 H7 ↧3
Dowel Pin Holes
14

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 (σ) [1]	2.6 μm @ 0.3 m
Regional Z-Value Repeatability (σ) [2]	0.09 μm @ 0.3 m
Measurement Accuracy (VDI/VDE) [3]	0.03 mm @ 0.3 m
Typical Capture Time	0.6–0.9 s
Weight	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 [4]	IP65
Cooling	Passive

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

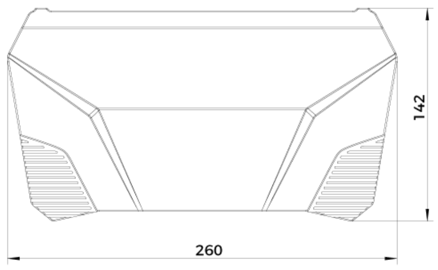
[2] One standard deviation of 100 measurements of the difference between the Z-value means of two same-sized regions. The measurement target was a ceramic plate.

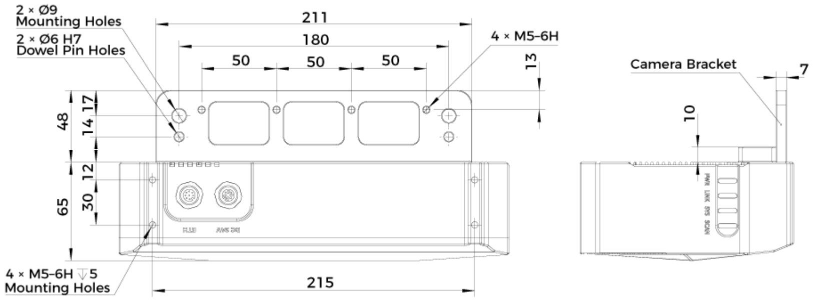
[3] According to VDI/VDE 2634 Part II.

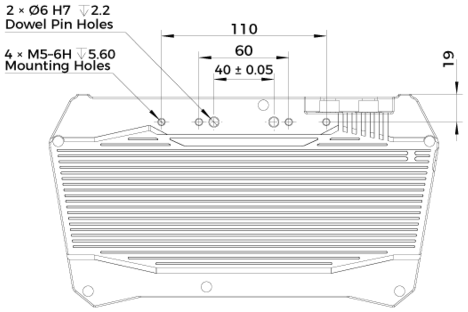
[4] Test implemented based on IEC 60529. 6: dust-tight; 5: protected against water jets.

Dimensions

Unit: mm







1.6.5 NANO

Specifications

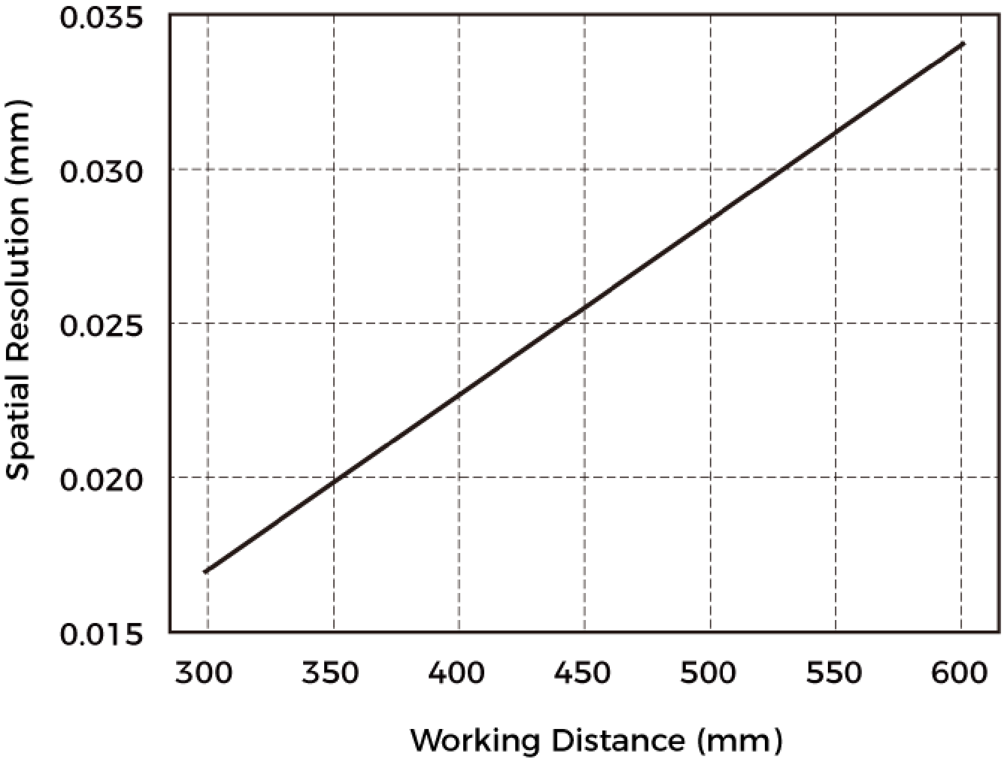
Product Name	Mech-Eye Industrial 3D Camera
Model	NANO
Recommended Working Distance	300–600 mm
FOV	220 × 150 mm @ 0.3 m
	440 × 300 mm @ 0.6 m
Resolution	1280 × 1024
Megapixels	1.3 MP
Point Z-Value Repeatability (σ) [1]	0.1 mm @ 0.5 m
Measurement Accuracy (VDI/VDE) [2]	0.1 mm @ 0.5 m
Typical Capture Time	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 [3]	IP65
Cooling	Passive

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

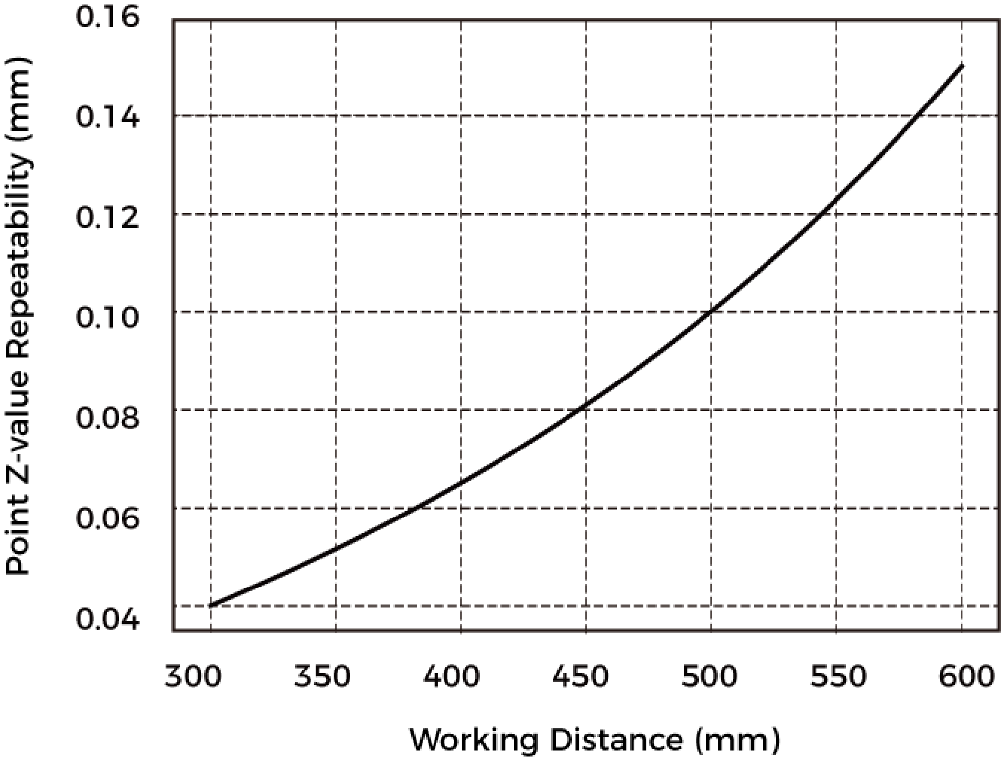
[2] According to VDI/VDE 2634 Part II.

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

Dimensions

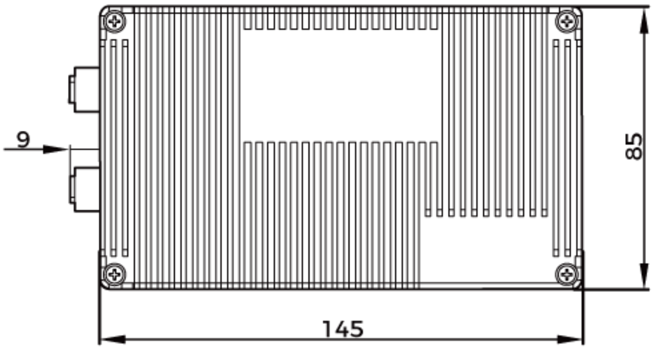

0.035
0.030
0.025
0.020
0.015
Spatial Resolution (mm)
300
350
400
450
500
550
600
Working Distance (mm)

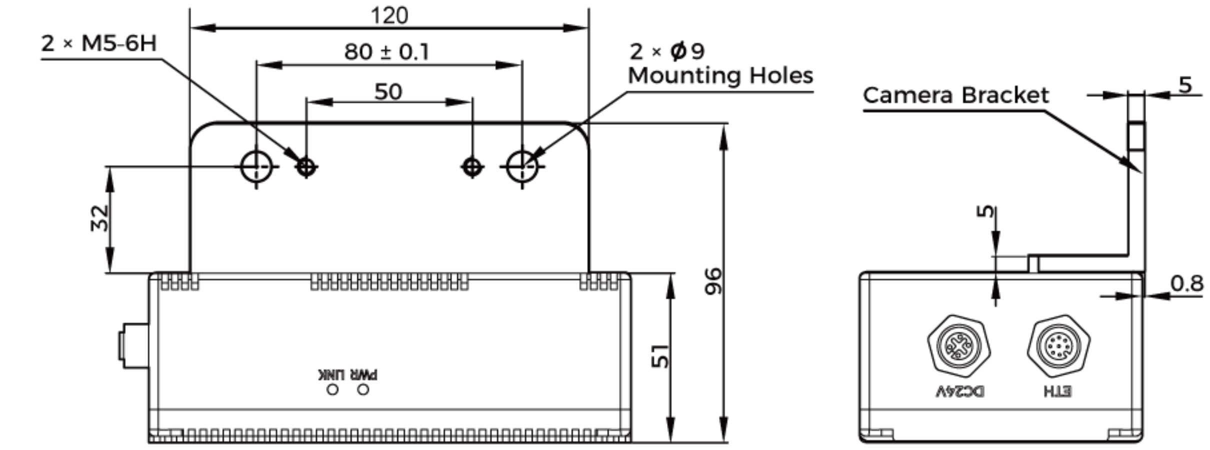
Point Z-Value Repeatability

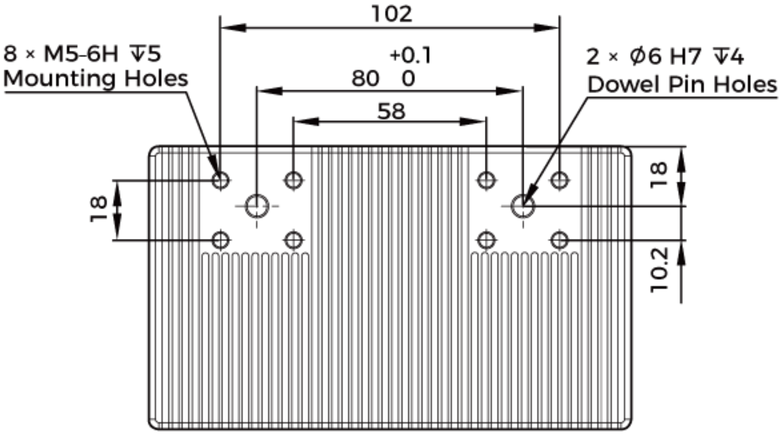


Dimensions

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 (σ)	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 (σ)	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 maintaind 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.

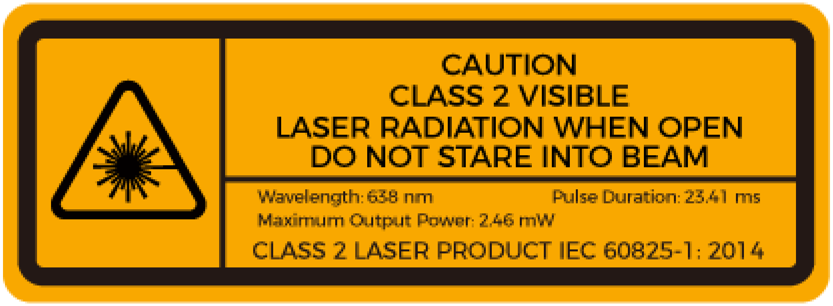
- European Electromagnetic Compatibility Standards.
- U.S. ANSI C63.4 and 47 CFR PART 15B.
- Canada ICES-003.
- Japan VCCI-CISPR 32: 2016.

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

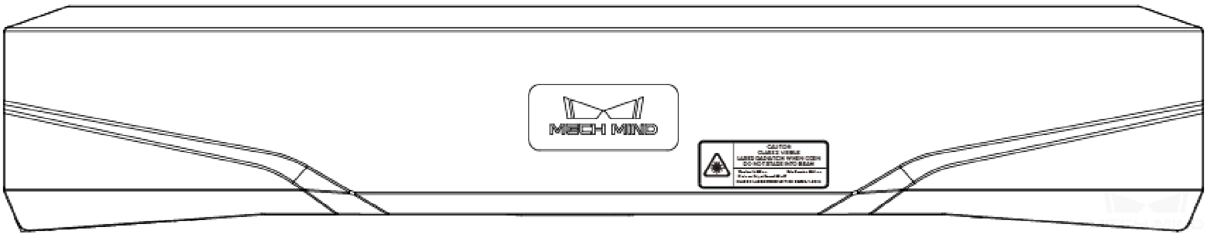


CAUTION
CLASS 2 VISIBLE
LASER RADIATION WHEN OPEN
DO NOT STARE INTO BEAM

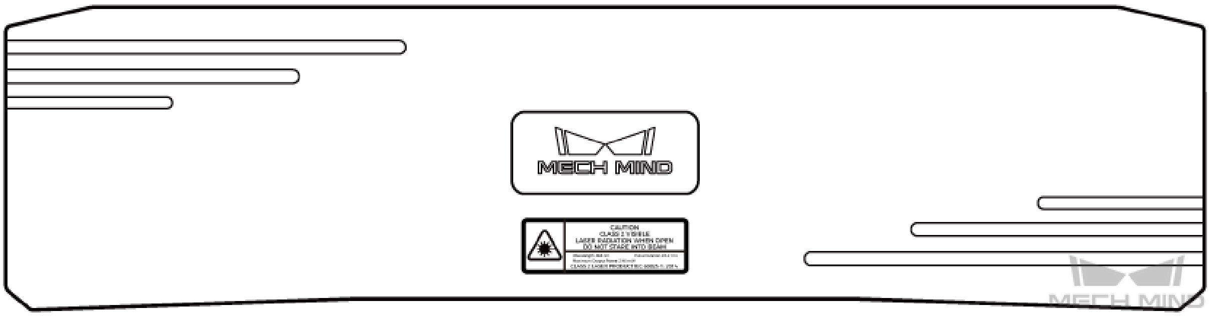
Wavelength: 638 nm Pulse Duration: 23.41 ms
Maximum Output Power: 2.46 mW
CLASS 2 LASER PRODUCT IEC 60825-1: 2014

Warning Label Attachment

LSR L



DEEP



Conformitè Europëenne (CE) Certifications

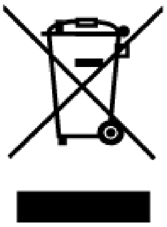
Hereby [Mech-Mind Robotics Technologies Ltd.] declares that [LSR L, PRO S, PRO M and UHP-140] is in compliance with the Electromagnetic Compatibility Directive 2014/30/EU.

The full text of the EU Declaration of Conformity is available here.

European Electromagnetic Compatibility Standards

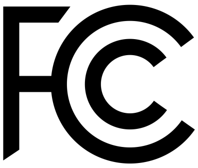
- EN 55032: 2015 + A1: 2020
- EN IEC 61000-3-2: 2019 + A1: 2021
- EN 61000-3-3: 2013 + A1: 2019 + A2: 2021
- EN 55035: 2017 + A11: 2020

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 tempurature 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 MECH MIND 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.

CHAPTER

TWO

MECH-EYE SDK 2.1 RELEASE NOTES

This topic introduces the new features, improvements and resolved issues in Mech-Eye SDK 2.1.

2.1 Mech-Eye SDK 2.1.0 Release Notes

This document introduces the new features, improvements, and resolved issues of Mech-Eye SDK 2.1.0. The highlights of this version include the following.

- Improved point cloud quality: the **Point Cloud Processing** parameters have been redesigned and can effectively remove outliers and noises while presevering the object edge information in the point cloud, improving point cloud quality.
- Enhanced the functionality of interfaces:
  - Parameters used for hand-eye calibration are added to the GenICam interface and Mech-Eye API, and corresponding HALCON sample programs are provided. You can now perform hand-eye calibration directly with the interfaces.
  - Parameters for obtaining multi-part payload type data are added to the GenICam interface, and corresponding sample programs are provided. You can now easily obtain textured point clouds and depth maps that contain only the Z values.
  - A firmware upgrade tool is added. You can now upgrade the camera firmware without Mech-Eye Viewer.

2.1.1 New Features

GenICam

- Added Parameters and Sample Programs for Hand-Eye Calibration

  Parameters used for hand-eye calibration are added to the GenICam interface, and *corresponding HALCON sample programs* are provided. Through these sample programs, you can perform hand-eye calibration, obtain extrinsic parameters, and switch the reference frame in which the point cloud is output from the camera.

- Added Parameters and Sample Programs for Multi-Part Payload Type
  - Parameters for obtaining multi-part payload type data in the **AreaScan3D** mode are added to the GenICam interface. The Range and Intensity channels are supported.

  - Two HALCON sample programs demonstrating the applications of the above parameters are added.

    - obtain_textured_point_cloud: uses data from the Range and Intensity channels to *generate the textured point cloud*.
    - obtain_depth_map: *obtains the depth map that contains only the Z values of points* through the Range channel, speeding up the data transmission and improving efficiency.

- Added Parameters and Sample Program for Camera IP Settings

  Parameters for obtaining the current IP settings of the camera are added to the GenICam interface, and *a HALCON sample program for obtaining and modifying the camera IP settings* is provided. You can now obtain and modify the camera IP settings with a GenICam client. Through the HALCON sample program, you can obtain the current IP address, subnet mask and gateway of the camera, as well as modifying these settings.

- Added Read-Only Parameters for Camera Intrinsic Parameters and Temperatures

  - Parameters for obtaining the camera intrinsic parameters are added. You can now obtain the camera intrinsic parameters with a GenICam client.

    - Scan3dFocalLengthX
    - Scan3dFocalLengthY
    - Scan3dPrincipalPointU
    - Scan3dPrincipalPointV

  - Parameters for obtaining the temperatures are added. You can now obtain the temperatures of the camera CPU and projector with a GenICam client.

    - DeviceTemperatureSelector: for selecting the camera component whose temperature you want to obtain. Options include MainBoard (CPU) and Sensor (projector).
    - DeviceTemperature: displays the temperature of the selected camera component.

Mech-Eye API

- Added Methods and Sample Programs for Hand-Eye Calibration (C++ Only)

  C++ methods used for hand-eye calibration are added, and a corresponding sample program is provided. Through this sample program, you can perform hand-eye calibration, and obtain extrinsic parameters.

- Added Firmware Upgrade Tool

  A *firmware upgrade tool* is added. With this tool, Ubuntu users can now upgrade the camera firmware as well.

- Added Methods for Obtaining Device Temperature

  - C++:

```
ErrorStatus MechEyeDevice::getDeviceTemperature(DeviceTemperature &)
```

  - C#:

```
ErrorStatus MechEyeDevice.GetDeviceTemperature(ref mmind.apiSharp.DeviceTemperature␣
↪temperature)
```

- Python:

```
Device.get_device_temperature(self)
```

Mech-Eye Viewer

- Two 2D Image Types Displayed for LSR (V4) and DEEP (V4) Series

  For the LSR (V4) or DEEP (V4) series, you can now select whether to display **2D Image (Texture)** or **2D Image (Depth Source)** in Mech-Eye Viewer. The 2D image (texture) is used for texturing the point cloud, and the 2D images (depth source) are used in hand-eye calibration, checking intrinsic parameters and setting ROI. Switching the 2D image display helps you adjust the exposure of different 2D images.

  Meanwhile, the names of the relevant **2D Parameters** have been updated accordingly.

Old name	New name
Color Camera Exposure Mode	2D Image (Texture) Exposure Mode
Monochrome Camera Exposure Mode	2D Image (Depth Source) Exposure Mode
Monochrome Camera Exposure Time	2D Image (Depth Source) Exposure Time

- Added MTU Value Setting

  The setting for adjusting the MTU value is added to the IP address configuration window. This setting is available in the **Administrator** account. Meanwhile, after the camera firmware is upgraded to 2.1.0, the default MTU value of the camera will change to 1500.

  Note: The MTU value of the camera should be consistent with the *jumbo frame value* set on the IPC.

- Adjustable Depth Map Resolution for DEEP (V4) Series

  The resolution of the depth map obtained by the DEEP (V4) series can now be adjusted. Options include 2048 × 1536 and 1024 × 768. Using the lower resolution can speed up image capturing effectively (about 35%) and thus better cope with applications with short cycle time.

  In the **Administrator** account, select *Tools* → *Camera Controller*, and adjust the resolution in the **Depth Map** menu.

- Added Reset to Factory Intrinsic Parameters Function

  The reset to factory intrinsic parameters function is added to *Intrinsic Parameter Tool*. This function is available in the **Administrator** account.

2.1.2 Improvements

Improved Point Cloud Processing Parameters

The **Point Cloud Processing** parameters have been redesigned, upgrading from 2 parameters to 4, and the performance of smoothing and unwanted-point removal has been greatly enhanced.

- The old **Point Cloud Smoothing** parameter is upgraded to *Surface Smoothing*. The smoothing algorithm is enhanced, and the same intensity now provides better smoothing result.
- The old **Noise Removal** parameter is upgraded to *Outlier Removal* and *Noise Removal*. Unwanted points are divided into outliers and noise, with separate, more targeted removal parameters. The removal algorithm is also enhanced, and the same intensity now provides better removal result.
- A new *Edge Preservation* parameter is added. You can now select the type of edge preservation to perform based on the object type, better preserving the object edge features.

If you want to achieve the processing result similar to the old parameters, please adjust the new parameters in the following way.

- If the old **Point Cloud Smoothing** parameter was set to **Normal**, set the new **Surface Smoothing** parameter to **Weak**.
- If the old **Noise Removal** parameter was set to **Normal**, set the new **Outlier Removal** parameter to **Weak**.

Improved Capture Speed of Camera

After the camera firmware is upgraded to 2.1.0, the capture speed of the camera is improved.

- When obtaining the depth map through Mech-Eye API C++, the capture speed is improved by at least 2% and at most 35% or more.
- When obtaining the depth map through Mech-Vision, the capture speed is improved by at least 5% and at most 20% or more.

Note: The 2D image distortion correction function has been added to V4 laser cameras, which resulted in a reduction of the capture speed of the color image.

Mech-Eye Viewer

- Changed the Default Format of Depth Map When Saving

  The default format of depth map when saving is changed from TIFF to PNG, improving usability.

- Improved Intrinsic Parameter Tool
  - The user interface of *Intrinsic Parameter Tool* has been improved, with more tips and hints added, improving the usability of the tool.
  - A high-accuracy mode is added, in which stricter standards are used to check the intrinsic parameters. This mode is suitable for projects with high accuracy requirements. This mode is available in the **Administrator** account.

- Improved Display of Parameter Description Section

  After the close button of the **Parameter Description** section is clicked, this section is no longer displayed, so that the parameters are not overlapped, improving the usability of the parameter panel.

  To display the **Parameter Description** section again, check the **Parameter Description** option in the **View** menu.

- Improved FOV Calculator Tool
  - The user interface of the *FOV Calculator* tool has been improved, improving the usability of the tool.
  - The drop-down list of camera models has been improved and is more convenient for finding the more frequently used models.
  - The LSR S (V4) model is added to the list.
- The color camera binning function of LSR L (V4) has been moved to *Camera Controller* and redesigned as the color image resolution setting function.
- The default value of **Camera Gain** for the DEEP (V4) series is updated from **0** to **5**.
- The value range of **Laser Power** for laser cameras is updated from **20–100%** to **50–100%**.

Mech-Eye API

The **Point Cloud Processing** parameters have been redesigned, upgrading from 2 parameters to 4. The corresponding Mech-Eye API methods are upgraded as follows.

- C++:
  - Surface Smoothing:

    ```
    ErrorStatus␣
    ↪MechEyeDevice::setCloudSurfaceSmoothingMode(PointCloudProcessingSettings::PointCloudSurfaceSmoothin
    ErrorStatus␣
    ↪MechEyeDevice::getCloudSurfaceSmoothingMode(PointCloudProcessingSettings::PointCloudSurfaceSmoothin
    ↪&)
    ```

  - Noise Removal:

    ```
    ErrorStatus␣
    ↪MechEyeDevice::setCloudNoiseRemovalMode(PointCloudProcessingSettings::PointCloudNoiseRemoval)
    ErrorStatus␣
    ↪MechEyeDevice::getCloudNoiseRemovalMode(PointCloudProcessingSettings::PointCloudNoiseRemoval␣
    ↪&)
    ```

  - Outlier Removal:

    ```
    ErrorStatus␣
    ↪MechEyeDevice::setCloudOutlierRemovalMode(PointCloudProcessingSettings::PointCloudOutlierRemoval)
    ErrorStatus␣
    ↪MechEyeDevice::getCloudOutlierRemovalMode(PointCloudProcessingSettings::PointCloudOutlierRemoval␣
    ↪&)
    ```

  - Edge Preservation:

```
ErrorStatus␣
↪MechEyeDevice::setCloudEdgePreservationMode(PointCloudProcessingSettings::PointCloudEdgePreservatio
ErrorStatus␣
↪MechEyeDevice::getCloudEdgePreservationMode(PointCloudProcessingSettings::PointCloudEdgePreservatio
↪&)
```

- C#:
  - Surface Smoothing:

```
ErrorStatus MechEyeDevice.SetCloudSurfaceSmoothingMode(mmind.apiSharp.
↪PointCloudSurfaceSmoothing mode)
ErrorStatus MechEyeDevice.GetCloudSurfaceSmoothingMode(ref mmind.apiSharp.
↪PointCloudSurfaceSmoothing mode)
```

  - Noise Removal:

```
ErrorStatus MechEyeDevice.SetCloudNoiseRemovalMode(mmind.apiSharp.
↪PointCloudNoiseRemoval mode)
ErrorStatus MechEyeDevice.GetCloudNoiseRemovalMode(ref mmind.apiSharp.
↪PointCloudNoiseRemoval mode)
```

  - Outlier Removal:

```
ErrorStatus MechEyeDevice.SetCloudOutlierRemovalMode(mmind.apiSharp.
↪PointCloudOutlierRemoval mode)
ErrorStatus MechEyeDevice.GetCloudOutlierRemovalMode(ref mmind.apiSharp.
↪PointCloudOutlierRemoval mode)
```

  - Edge Preservation:

```
ErrorStatus MechEyeDevice.SetCloudEdgePreservationMode(mmind.apiSharp.
↪PointCloudEdgePreservation mode)
ErrorStatus MechEyeDevice.GetCloudEdgePreservationMode(ref mmind.apiSharp.
↪PointCloudEdgePreservation mode)
```

- Python:
  - Surface Smoothing:

```
Device.set_cloud_surface_smoothing_mode(self, info)
Device.get_cloud_surface_smoothing_mode(self)
```

  - Noise Removal:

```
Device.set_cloud_noise_removal_mode(self, info)
Device.get_cloud_noise_removal_mode(self)
```

  - Outlier Removal:

```
Device.set_cloud_outlier_removal_mode(self, info)
Device.get_cloud_outlier_removal_mode(self)
```

  - Edge Preservation:

```
Device.set_cloud_edge_preservation_mode(self, info)
Device.get_cloud_edge_preservation_mode(self)
```

2.1.3 Resolved Issues

The following issues are resolved in Mech-Eye SDK 2.1.0:

- When a laser camera used 3 exposure times, and **Laser Frame Partition Count** was set to 3 or 4, the camera would disconnect occasionally due to insufficient memory.
- For Laser L (V3) and Laser L Enhanced (V3), if **Laser Frame Partition Count** was set to 2 or higher, and exposure time(s) was set to 20 ms or less, the depth data would be abnormal.
- **ROI** setting occasionally became ineffective when the parameter group had been switched.
- For the LSR (V4) and DEEP (V4) series, if **Exposure Mode** was set to **Auto** or **HDR**, the **Flash** option of **Monochrome Camera Exposure Mode** would not take effect.
- When the camera was connected to a router, a large MTU value of the camera would result in occasional malfunction of the camera.
- Data frames were occasionally lost due to insufficient camera memory.
- For the PRO (V4), LSR (V4) and DEEP (V4) series, the default low power consumption mode would result in occasional data frame loss.
- When multiple cameras had the same IP address, changing the IP address of one camera would change the IP addresses of the other cameras at the same time.
- The 2D image distortion correction function of monocular cameras resulted in slower capture speed.

CHAPTER

THREE

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 Online Community or obtain it from Mech-Mind pre-sales or support engineers.

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 recomended 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.

CHAPTER

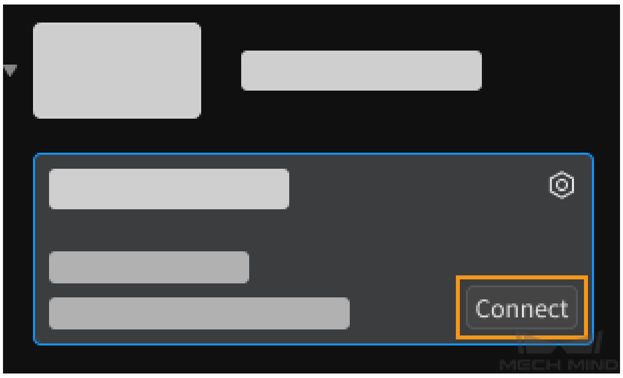
FOUR

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.

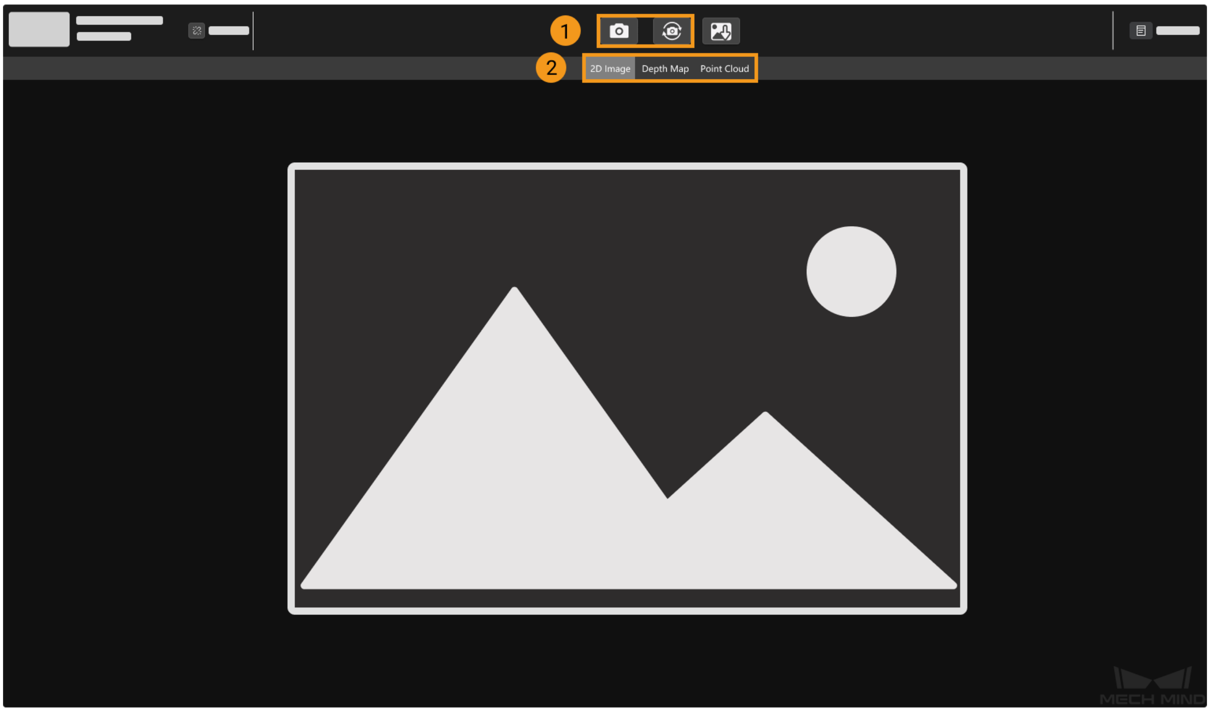
4.1 Connect to the Camera

Open Mech-Eye Viewer, select the camera in the **Camera List**, and click ⚙ to *set the camera IP address*. Then, click *Connect* to connect to the camera.



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**.

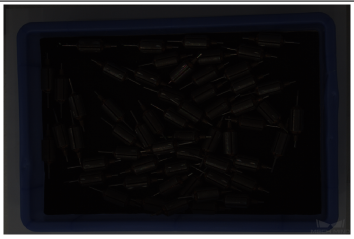
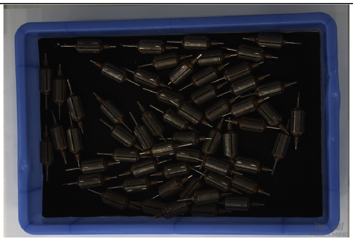
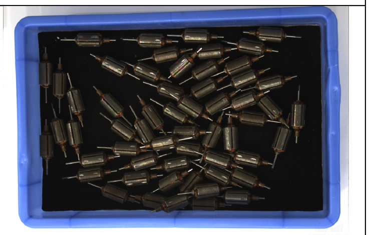
4.3 Adjust Parameters

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:

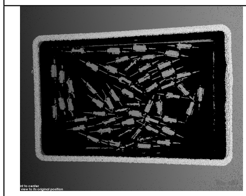
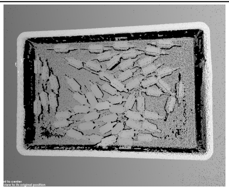
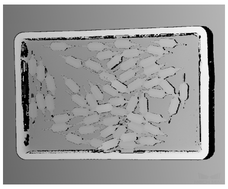
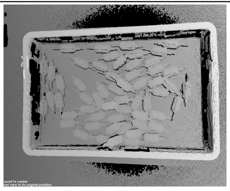
30 ms	120 ms	220 ms
		

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:

1 ms	4 ms	10 ms	50 ms
			

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*.

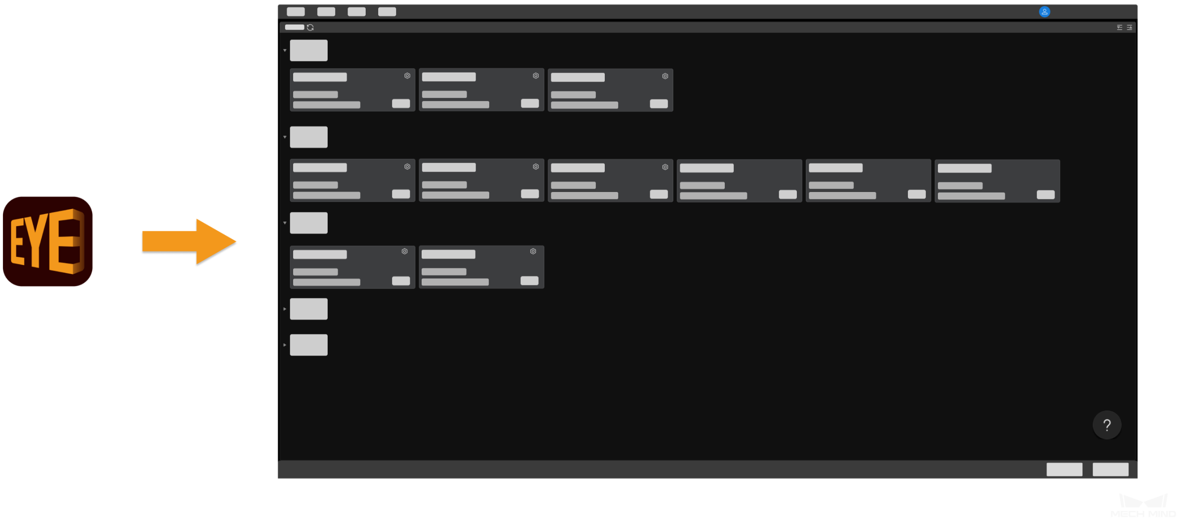
CHAPTER

FIVE

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

CHAPTER

SIX

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.

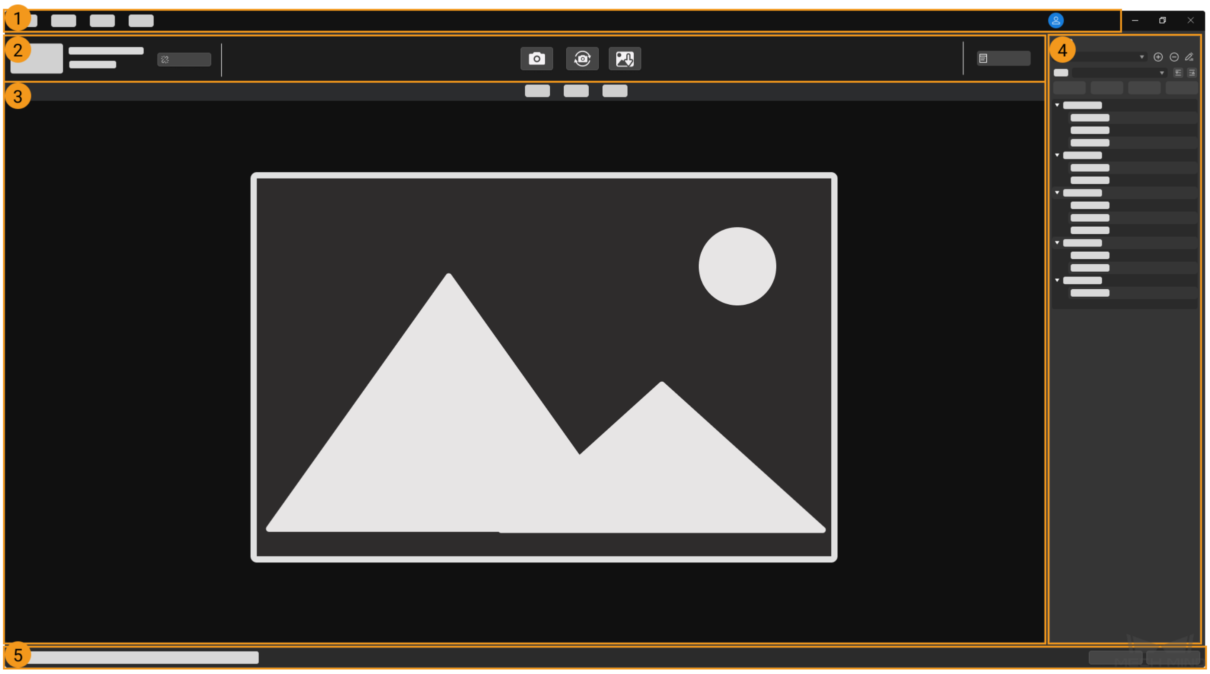
In this interface, you can view all the available cameras and their information, *set camera IP address and connect to a camera*.

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 *Built-In Tools*.	
View	Toolbar	Check to display the *toolbar*. Unchecked by default.
	Image Information Box	Check to display the image information box on the 2D image and depth map. Checked by default.
	Point Cloud Exhibit Button	Check to display the *Play* button on the point cloud. Unchecked by default.
	Parameter Description	Xheck to display the **Parameter Description** section in the parameter panel. Checked by default.
Help	About	View software version.
	What's New	View the changes made to the current version.
	User Manual	View the User Manual.
	Options	Switch the software display language; change is applied after restarting the software.
👤	Switch user account. The default account is **Standard**; 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 descriptions 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.

CHAPTER

SEVEN

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**:

Set Camera IP Address and Connect to Camera

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 Set Camera IP Address and Connect to Camera

A camera can only be connected in Mech-Eye Viewer when the following IP addresses are in the same subnet.

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

Please make sure that the above two IP addresses are in the same subnet and are both unique, and then connect to the camera.

After connecting to the camera, you can also set a custom camera name for easy recognition of different cameras.

7.1.1 Set Camera IP Address

You can set a static IP address for the camera, or set it to dynamic IP address, which is automatically assigned.

Set Static IP Address

Hint: Record the set camera IP address for future reference.

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

Hint:

- The IP address must be unique.
- The **Camera IP Configuration** section displays the IP address and subnet mask of the IPC Ethernet port connected to the camera. You can set the IP address and subnet mask of the camera according to the information displayed here.

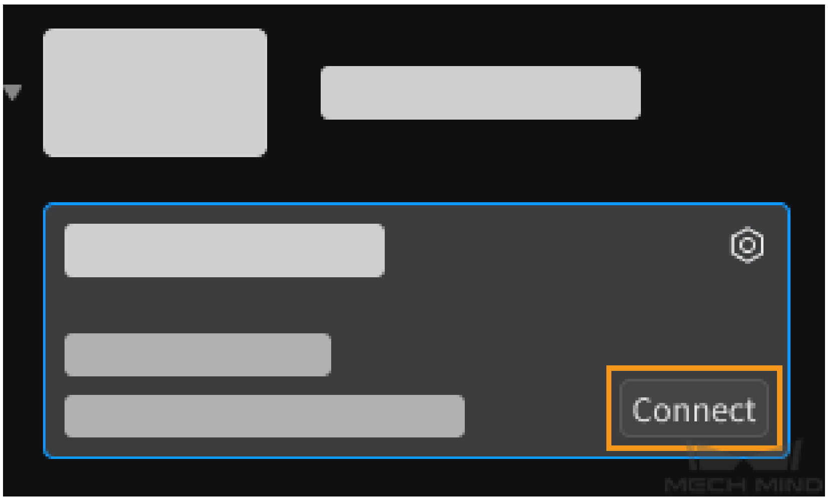
Set Dynamic IP Address

1. Open Mech-Eye Viewer and select the camera for which you'd like to set the IP address. Click ⚙ to open the IP address configuration window.
2. Select **Set via DHCP** and click *Apply*.

Hint: If you want to set the IP address of the IPC Ethernet port connected to the camera, please refer to *Set IP Address on IPC*.

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

7.1.3 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. The camera name will be displayed in the **Camera List**.

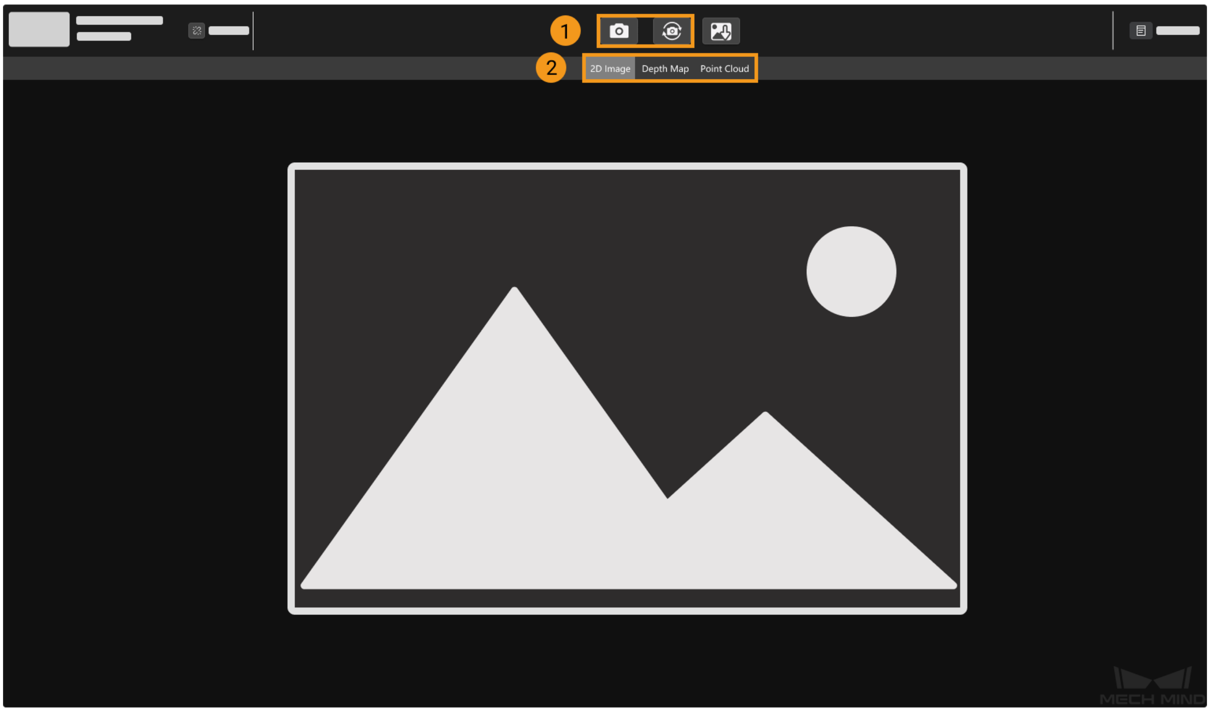
Note:

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

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 caturing at predetermined intervals and output the data to Mech-Eye Viewer.

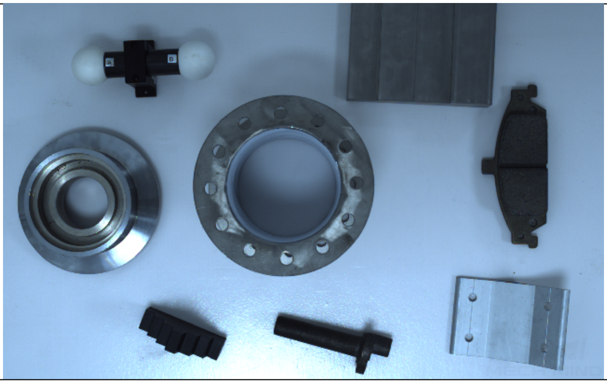
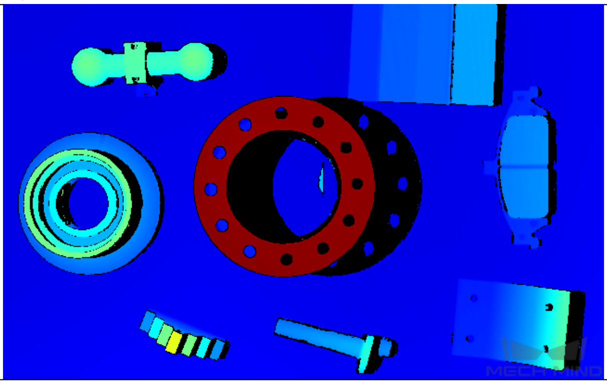
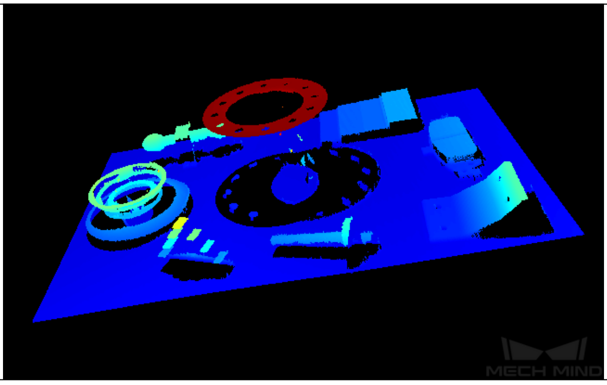
Hint:

- 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.

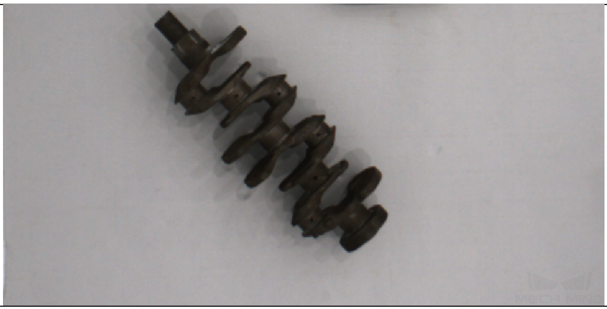
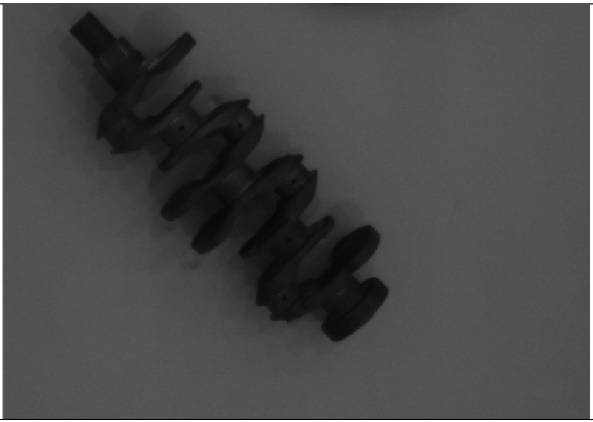
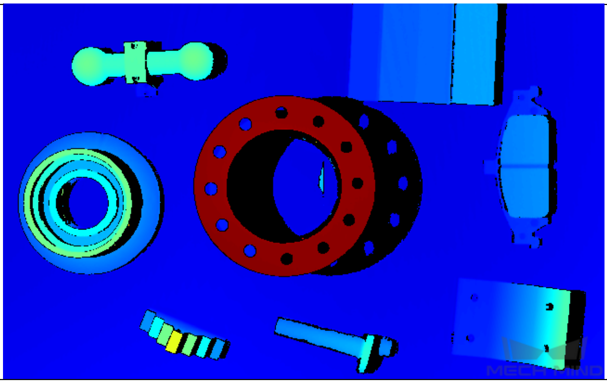
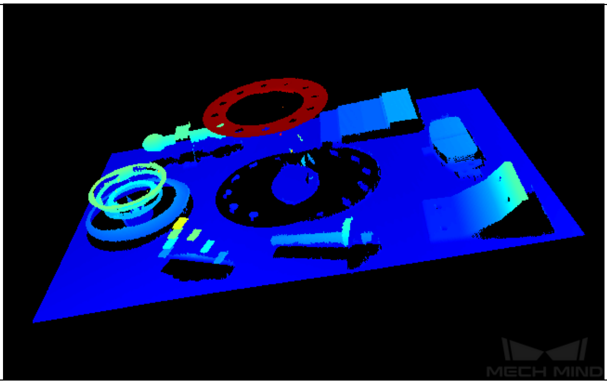
7.2.2 Switch Data Type

Click the buttons below the capture buttons to switch the displayed data type.

For cameras from series other than DEEP (V4) and LSR (V4):

Data Type	Description and Example
2D Image	Monochrome/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. 

For cameras from the DEEP (V4) and LSR (V4) series:

Data Type	Description and Example
2D Image (Texture)	2D image used for texturing the point cloud. 
2D Image (Depth Source)	2D images captured by the 2D cameras that acquire the depth data. 
Depth Map	An image containing depth information. 
Point Cloud	A collection of data points that represent the 3D shape of the object surface. 

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

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.

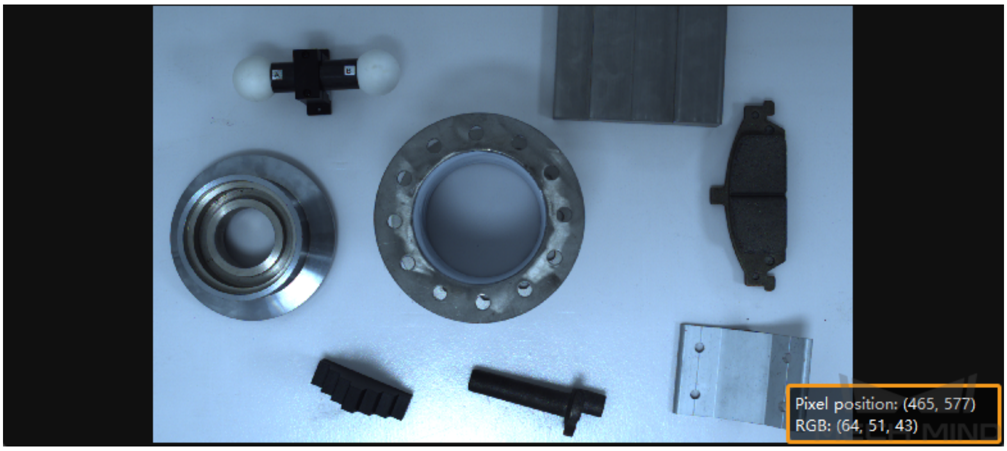


Image information displayed includes:

- Pixel position
- RGB values (for color 2D image) or grayscale value (for monochrome 2D image)

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

Adjust Depth Map Display

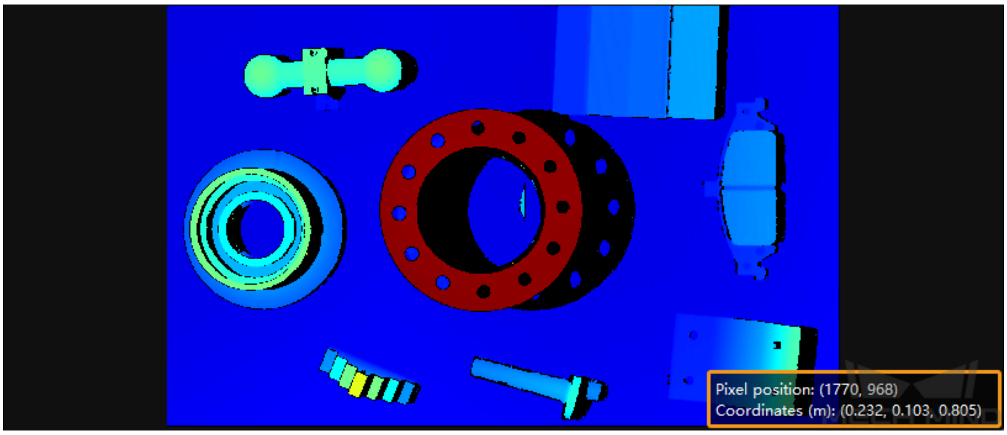
Zoom

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.



Pixel position: (1770, 968)
Coordinates (m): (0.232, 0.103, 0.805)

Image information displayed includes:

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

You can turn off this display box by unchecking **Image Informaiton** 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.

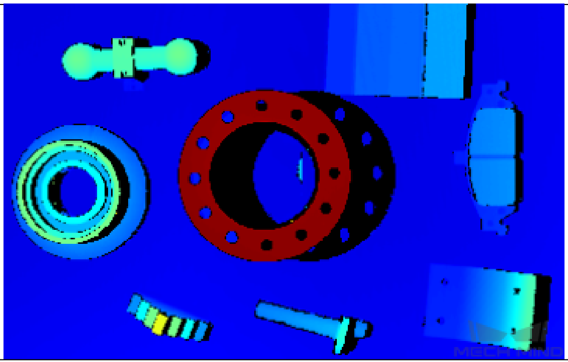
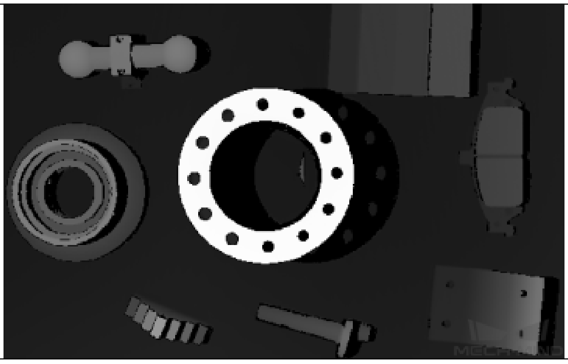
Change Reference Frame

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	Different depth values correspond to different colors in the Jet colormap. 
Gray Scale	Different depth values correspond to different grayscale values. 

Adjust Display Depth Range

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

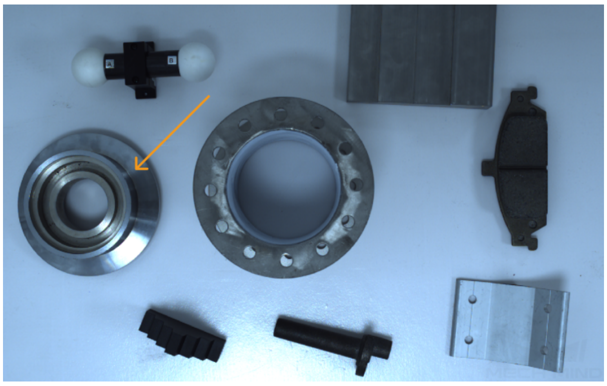
Note: The settings here only affect the display. The depth data in the depth map is not trimmed.

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.

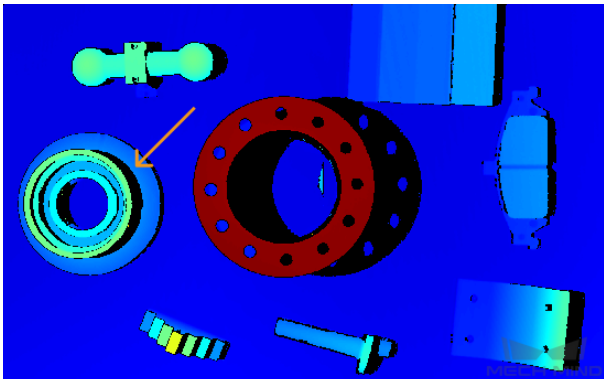
Click to read an example use case.

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

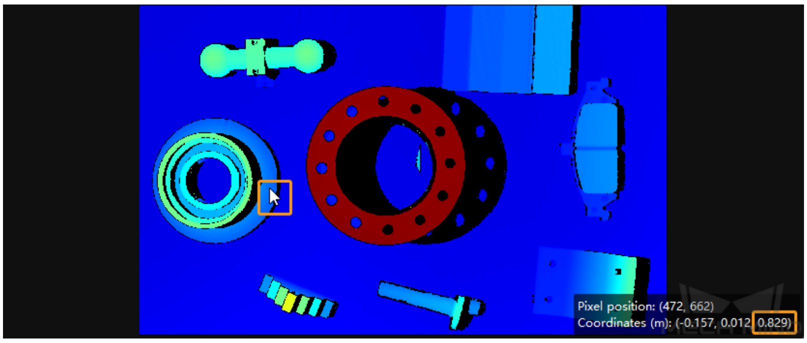


- Actions:

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.

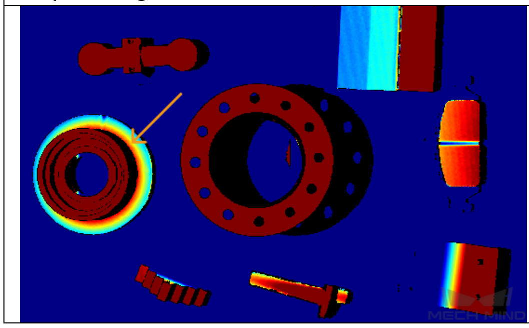
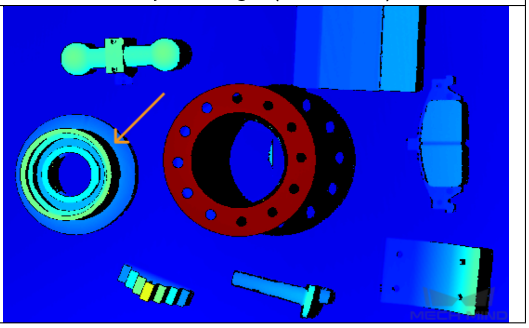


3. Move the cursor onto the surface to check the z coordinate. Here the z coordinate is 0.829 (unit: m).



Pixel position: (472, 662)
Coordinates (m): (-0.157, 0.012, 0.829)

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.

Depth range: 825–835	Unaltered depth range (725–861)
	

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 μm 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.

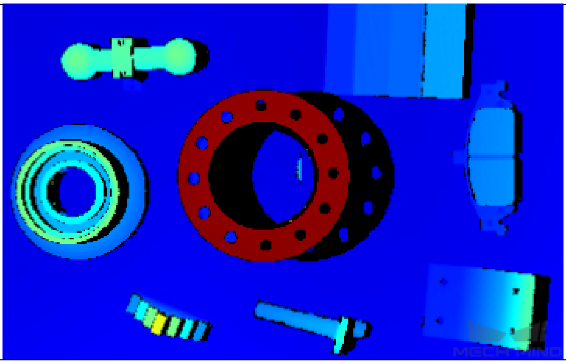
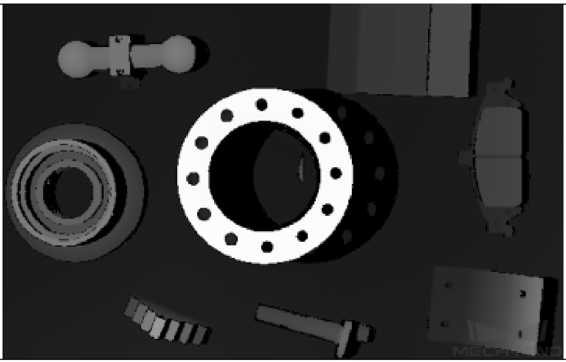
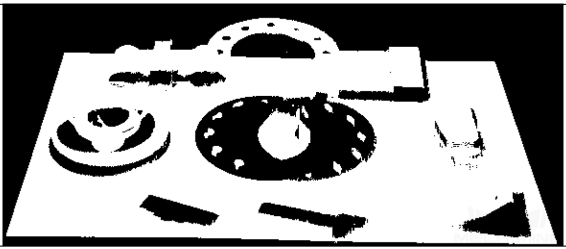
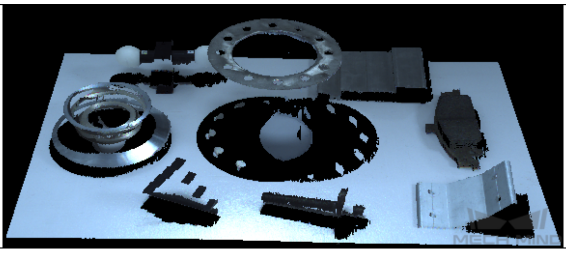
Change Reference Frame

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	Different depth values correspond to different colors in the Jet colormap. 
Gray Scale	Different depth values correspond to different grayscale values. 
White	The point cloud is untextured and displayed as completely white. 
Texture	The 2D image is used to texture the point cloud. 

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.

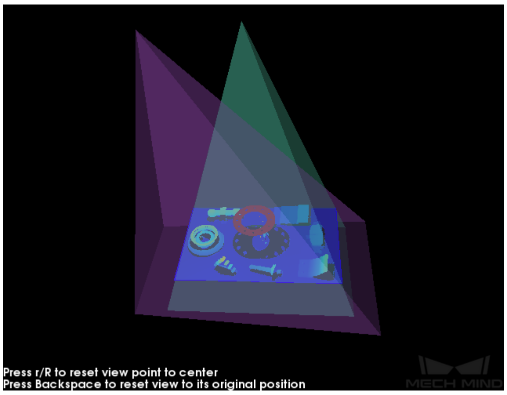
3. Click the same button (now *Stop*) to end the movement.

> **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).



Press r/R to reset view point to center
Press Backspace to reset view to its original position

- **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**:

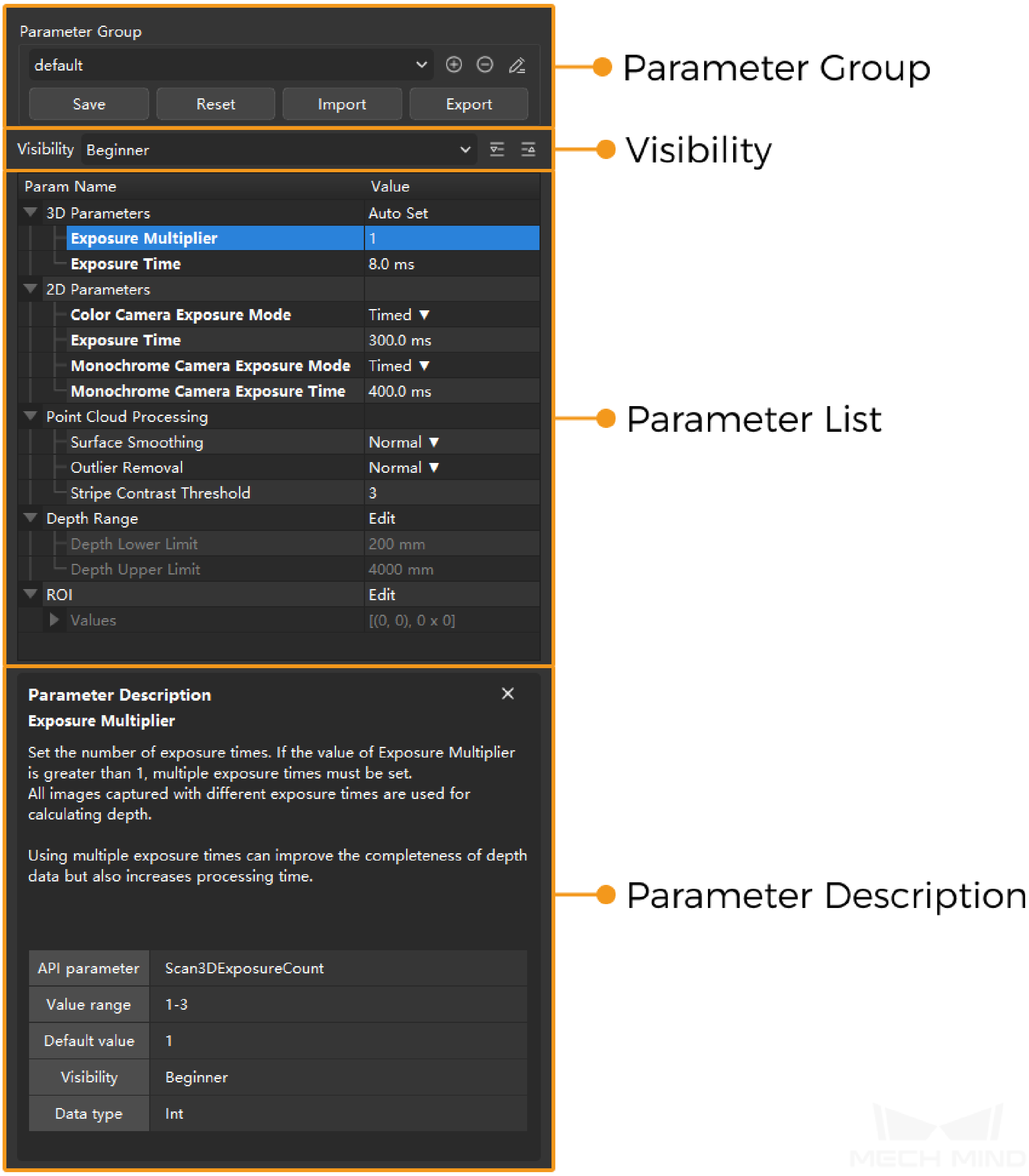
Advanced Parameters for Depth Map and Point Cloud

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:

- *Parameter Group*
- *Visibility*
- *Parameter List*
- *Parameter Description*



Parameter Group: default

Visibility: Beginner

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

Parameter Description

Exposure Multiplier

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

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

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

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.
Save	Save the parameter values to the currently selected parameter group. Shortcut: Ctrl + S.
Reset	Reset the parameter values in the currently selected parameter group to default values.
Import	Import parameter groups.
Export	Export parameter groups.

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

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 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 Description

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

Parameter description in this section includes a brief explanation of the parameter, the name of the corresponding **API Parameter**, **Value range**, **Default value**, **Visibility** and **Data type**.

If you installed Mech-Eye Viewer for the first time, the **Parameter Description** section is displayed by default. If you close this section, it will not be displayed again when you select a parameter. To display this section again, go to the **View** menu in the menu bar and check the **Parameter Description** option.

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 (V4) or DEEP (V4) series camera: **2D Image (Texture) Exposure Mode** and **2D Image (Depth Source) Exposure Mode** are displayed.
- If you are connected to a camera from another series: **Exposure Mode** is displayed.

Note: For the LSR (V4) and DEEP (V4) series:

- **2D Image (Texture) Exposure Mode** affects the 2D image used for texturing the point cloud.
- **2D Image (Depth Source) Exposure Mode** affects the 2D images captured by the 2D cameras that acquire the depth data. These images are used in hand-eye calibration, *checking intrinsic parameters*, and setting ROI.
- The "Color Image" port in Mech-Vision obtains the 2D image (depth source). If you need to adjust the "color images" in Mech-Vision, please adjust **2D Image (Depth Source) Exposure Mode**.

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

	Exposure Mode	2D Image (Texture) Exposure Mode	2D Image (Depth Source) 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 cameras from series other than DEEP (V4) and LSR (V4):

  Adjust **Exposure Time**.

- For DEEP (V4) and LSR (V4) series cameras:

  DEEP (V4) and LSR (V4) series cameras provide two types of 2D images. Please adjust **Exposure Time** and/or **2D Image (Depth Source) Exposre Time** according to your actual needs.

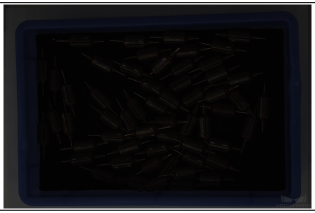
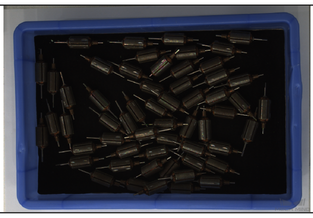
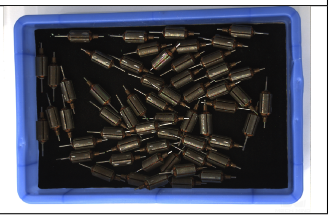
Parameter	Adjusting···	Image usage
Exposure Time	2D Image (Texture)	2D image used for texturing the point cloud
2D Image (Depth Source) Exposure Time	2D Image (Depth Source)	Check intrinsic parameters
		Set ROI
		Hand-Eye calibration

Exposure Time/2D Image (Depth Source) Exposure Time

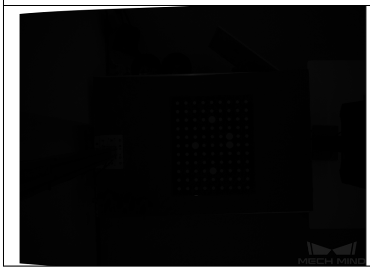
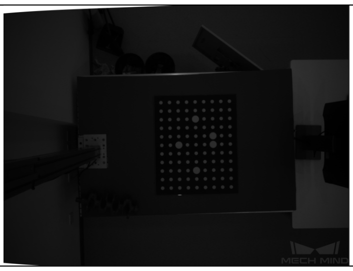
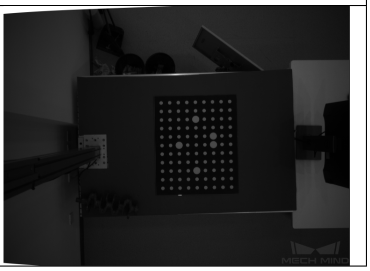
The value of **Exposure Time/2D Image (Depth Source) Exposure Time** affects the brightness of the image.

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):

30 ms	120 ms	220 ms
		

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

10 ms	40 ms	80 ms
		

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.

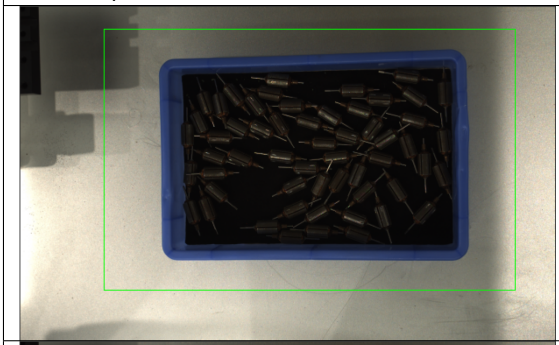
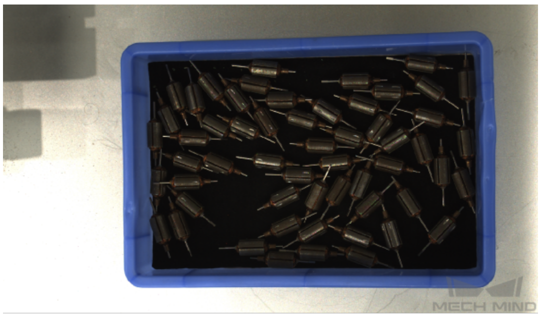
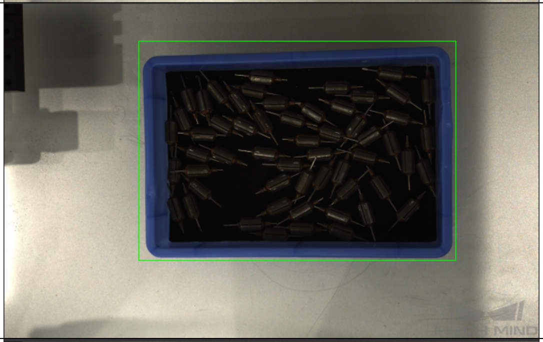
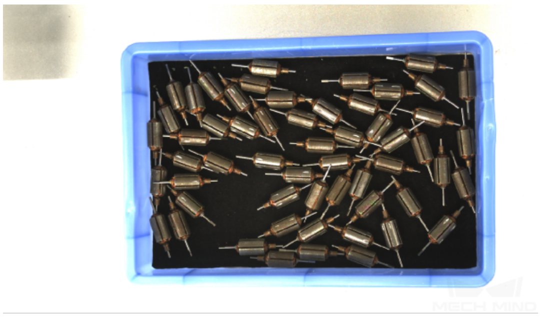
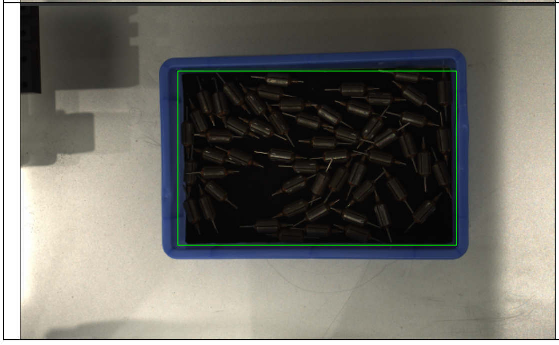
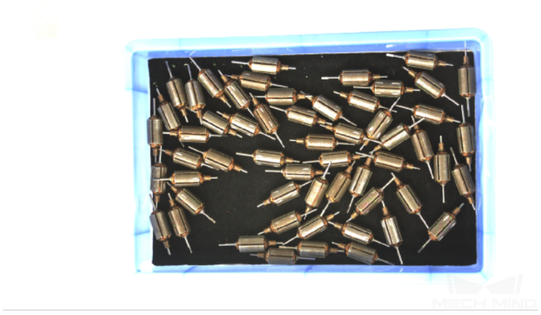
   Hint: Please select the area where the target objects are located and avoid including irrelevant objects as much as possible.

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
	
	
	

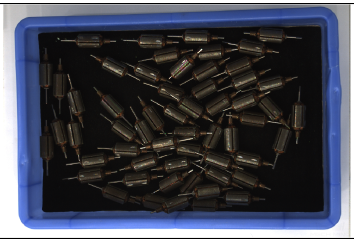
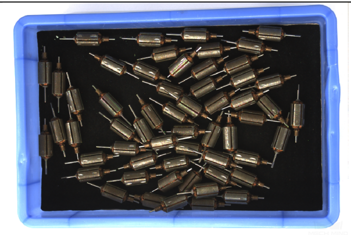
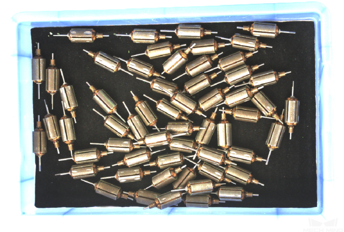
Gray Value

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):

50	100	150
		

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 a 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 -.

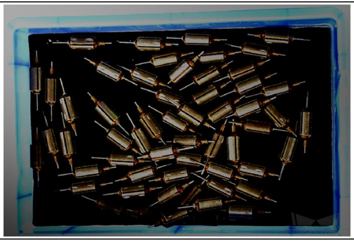
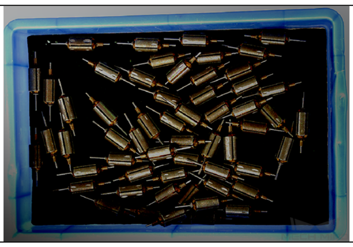
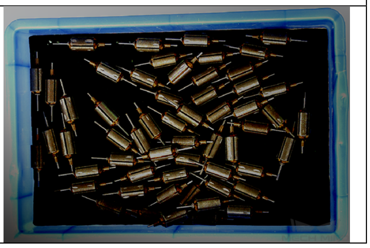
3. Click *Apply* to save the exposure sequence.

   Hint:

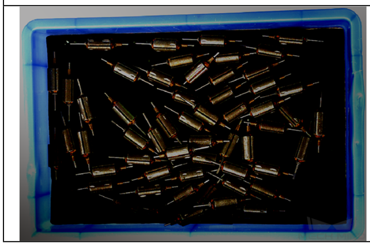
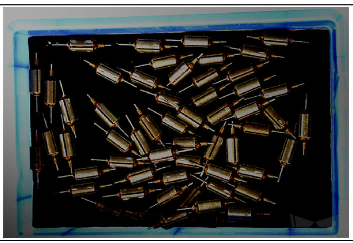
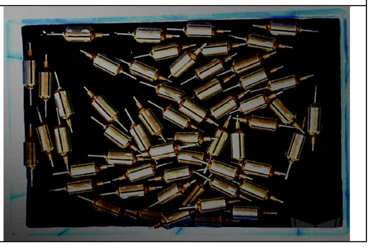
   - Click *Reset* to delete all the set exposure times.
   - Click *Cancel* to exit the window without saving the changes.

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):

700 ms	500 ms and 900 ms	500 ms, 700 ms and 900 ms
		

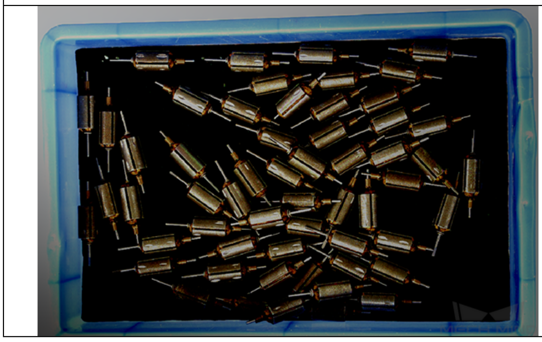
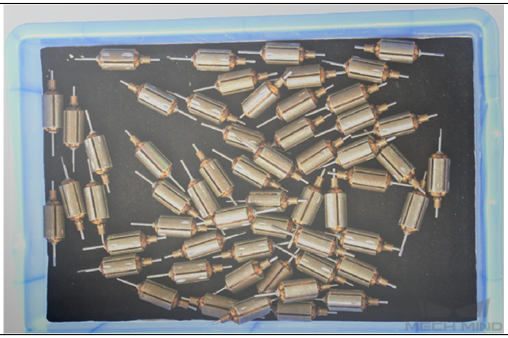
- For comparison: 2D images captured with single exposure times (all other conditions identical):

500 ms	700 ms	900 ms
		

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.

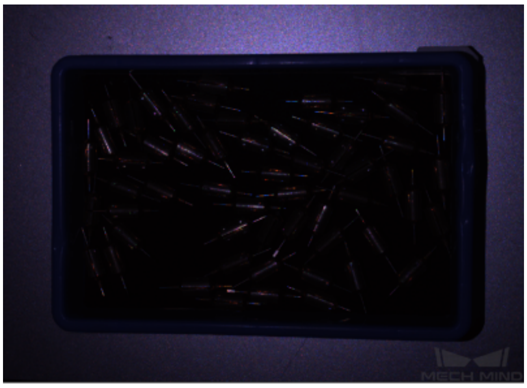
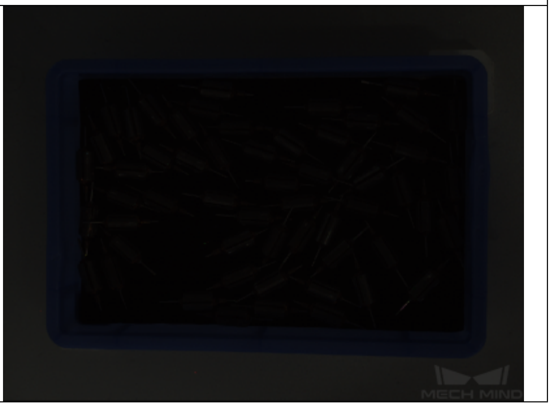
- 2D images captured with **Tone Mapping** disabled and enabled (all other conditions identical):

Disabled	Enabled
	

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):

Flash	Timed
	

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):

4 ms	10 ms	15 ms

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.

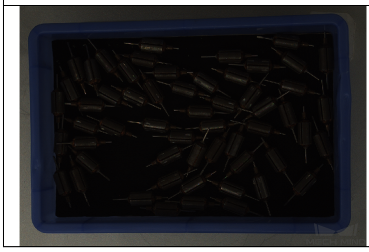
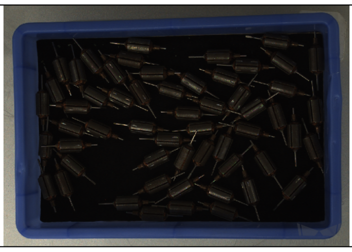
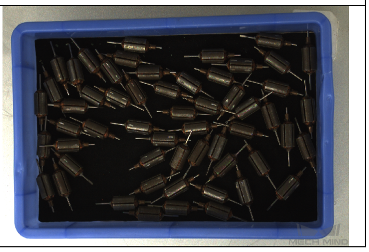
Gain

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):

0	5	10
		

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 *Adjust 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
3D Parameters	✓	✓
Point Cloud Processing		✓
Depth Range	✓	✓
ROI	✓	✓

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.

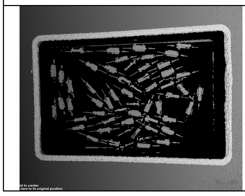
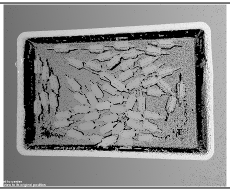
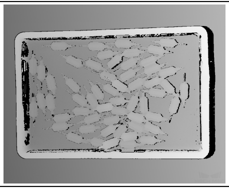
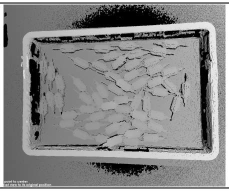
Exposure Time

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):

1 ms	4 ms	10 ms	50 ms
			

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 *Check and Configure 2D Camera*.

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.

Principles for Adjusting Point Cloud Processing Parameters

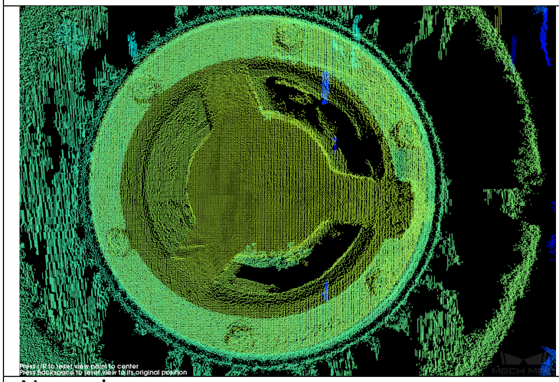
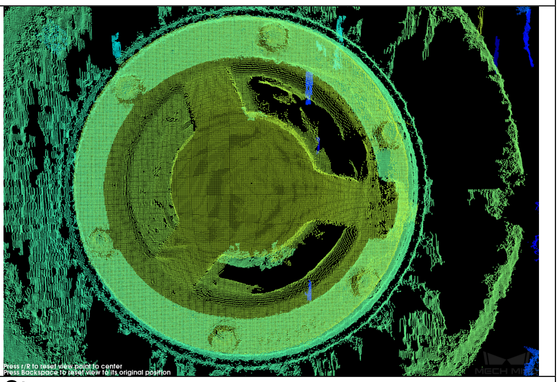
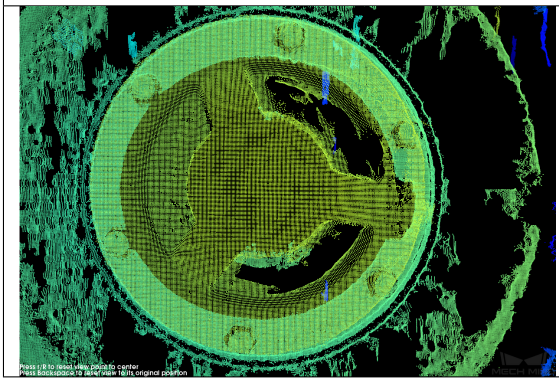
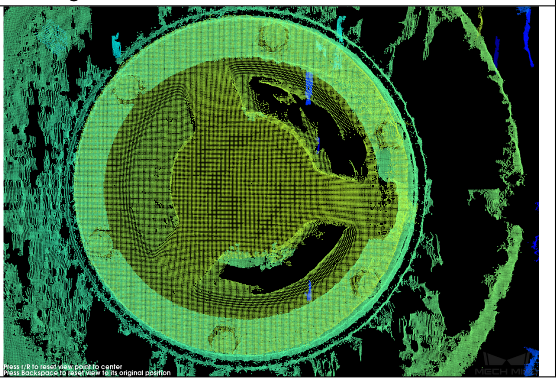
When adjusting the **Point Cloud Processing** parameters, the capture time and cycle time can be reduced if you follow these principles:

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

Surface Smoothing

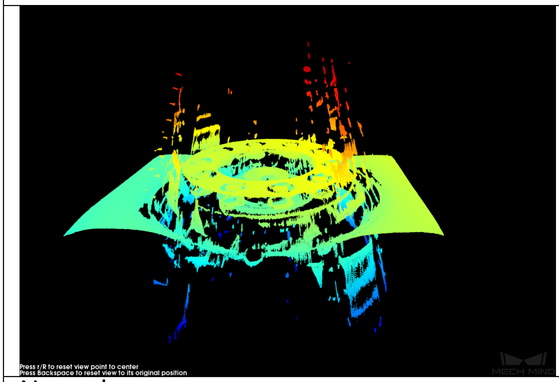
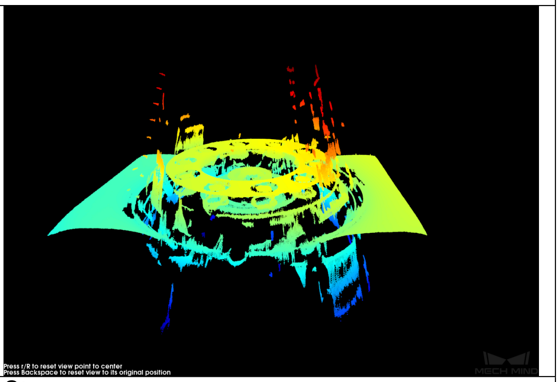
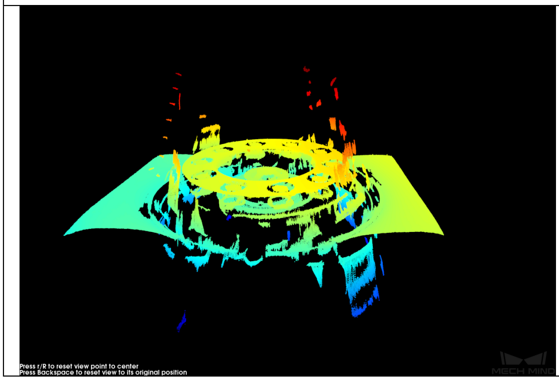
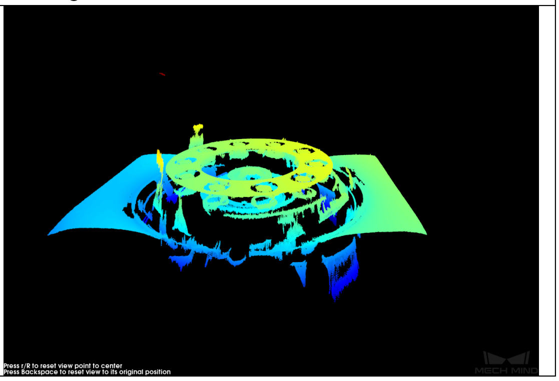
- Surface smoothing reduces the depth fluctuation in the point cloud and improves its resemblance to the actual object surface. At the same time, surface smoothing causes loss of object surface details.
- Values:
  - Off
  - Weak
  - Normal
  - Strong
- Default value: Off
- Instructions:
  - The more intense the surface smoothing, the more details are lost.
  - The more intense the surface smoothing, the longer it takes.

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

Off	Weak
	
Normal	Strong
	

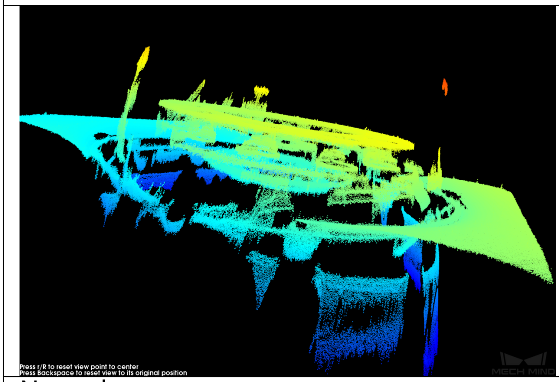
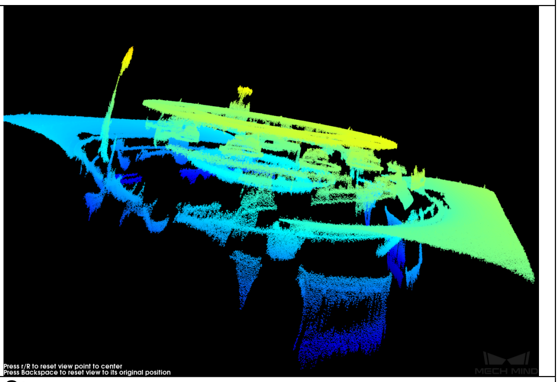
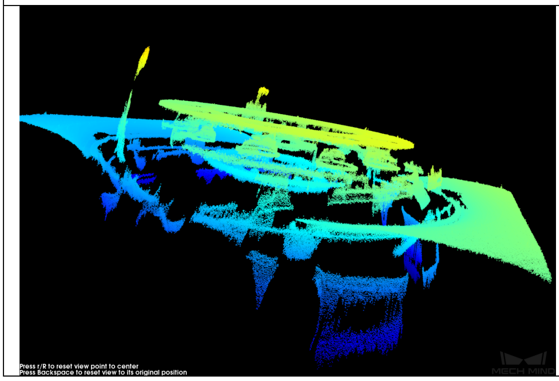
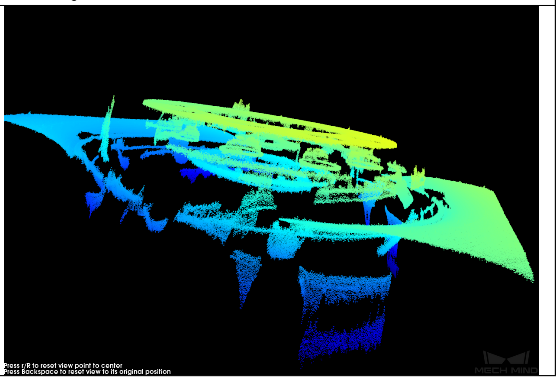
Outlier Removal

- Outlier removal removes the outliers in the point cloud. Outliers are clustered points away from the object point cloud.
- Values:
  - Off
  - Weak
  - Normal
  - Strong
- Default value: Weak
- Instructions:
  - The more intense the outlier removal, the more outliers are removed.
  - If the object point cloud contains multiple parts, high intensities of outlier removal might remove some parts. For example, if the object is a cup, outlier removal might remove the point cloud of the handle part.
- Point clouds generated with different outlier removal intensities (all other conditions identical):

Off	Weak
 Press r/R to reset view point to center Press Backspace to reset view to its original position	 Press r/R to reset view point to center Press Backspace to reset view to its original position
Normal	Strong
 Press r/R to reset view point to center Press Backspace to reset view to its original position	 Press r/R to reset view point to center Press Backspace to reset view to its original position

Noise Removal

- Noise removal removes the noise in the point cloud. Noise is the scattered points close to the object surface.
- Values:
  - Off
  - Weak
  - Normal
  - Strong
- Default value: Weak
- Instructions:
  - The more intense the noise removal, the more noise is removed, but some object feature might also be removed.
  - The more intense the noise removal, the longer it takes.
- Point clouds generated with different noise removal intensities (all other conditions identical):

Off	Weak
 Press r/R to reset view point to center Press Backspace to reset view to its original position	 Press r/R to reset view point to center Press Backspace to reset view to its original position
Normal	Strong
 Press r/R to reset view point to center Press Backspace to reset view to its original position	 Press r/R to reset view point to center Press Backspace to reset view to its original position

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

Edge Preservation

Note: This parameter is only available at the **Guru** visibility level.

- This parameter preserves the sharpness of object edges during surface smoothing.
- Values:
  - Sharp
  - Normal
  - Smooth
- Default value: Normal
- Instructions:
  - Sharp: Preserves the sharpness of object edges as much as possible. However, the effect of surface smoothing will be reduced.

  - Normal: Balances between edge preservation and surface smoothing.
  - Smooth: Does not preserve the edges. The object surface will be well smoothed, but the object edges will be distorted.

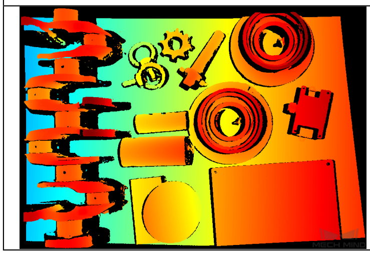
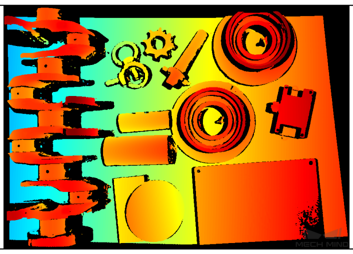
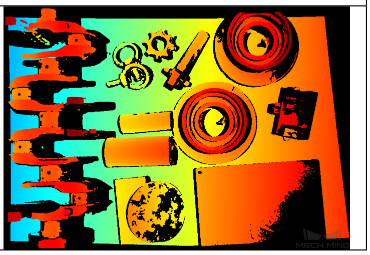
Stripe Contrast Threshold

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):

3	15	30
		

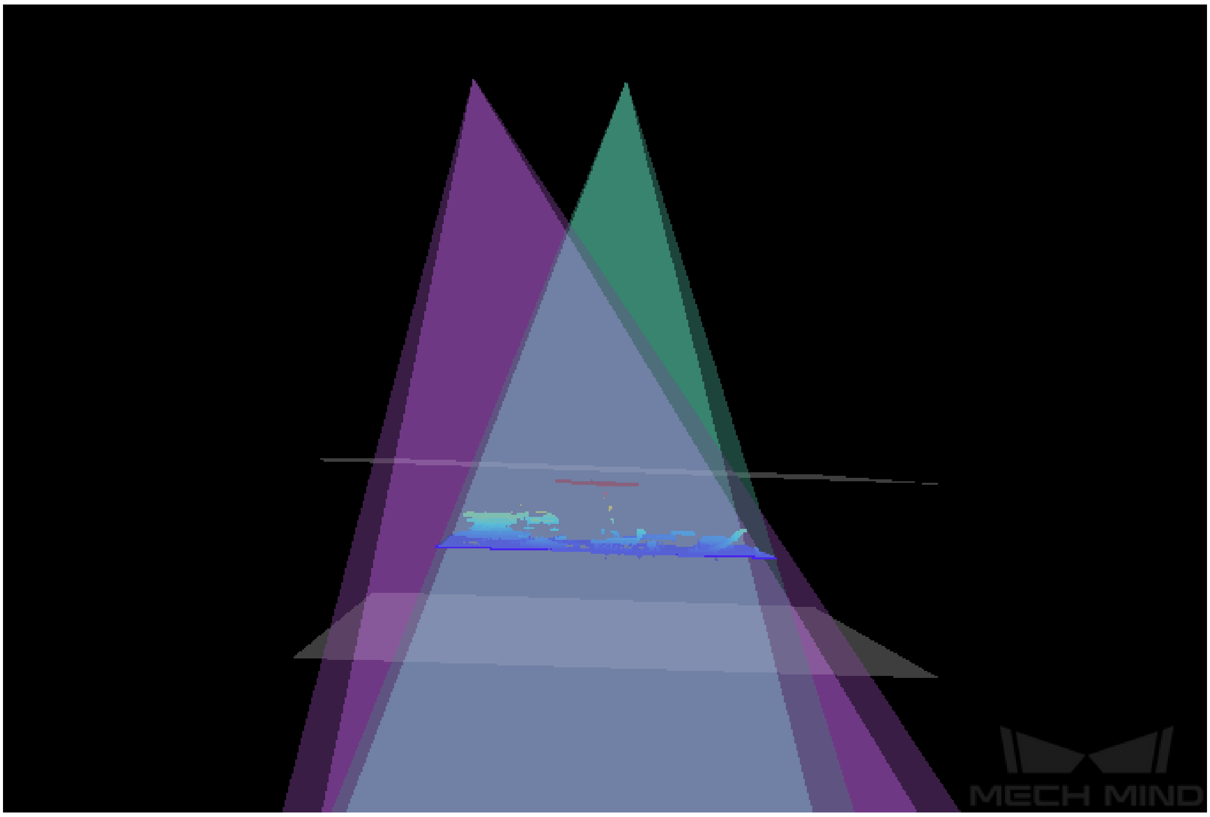
Depth Range

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.



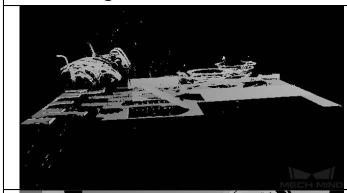
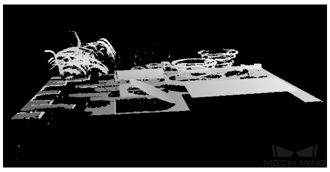
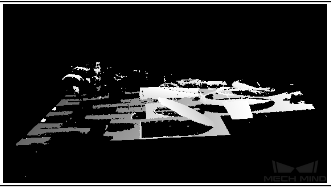
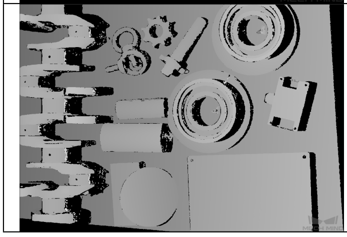
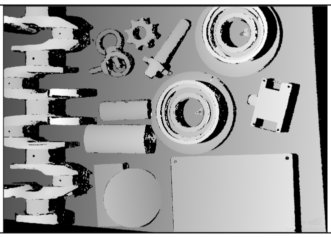
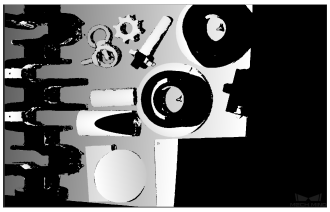
- Adjust the depth range: Drag the handles on the slider or enter the values in the text boxes. An appropriate range would sandwich all the desired points between the two gray squares while excluding unwanted points as much as possible.

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:

Too large	Appropriate	Too small
		
		

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 *2D Image (Depth Source) 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.

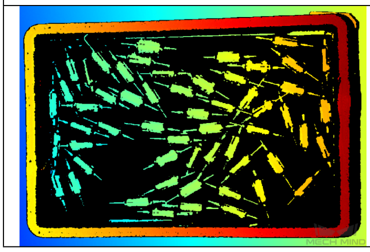
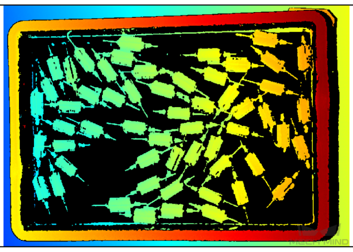
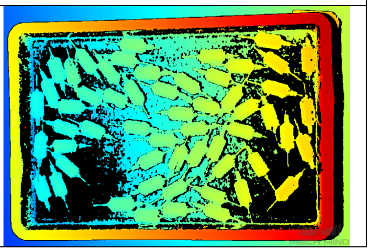
Gain

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):

0	5	10
		

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

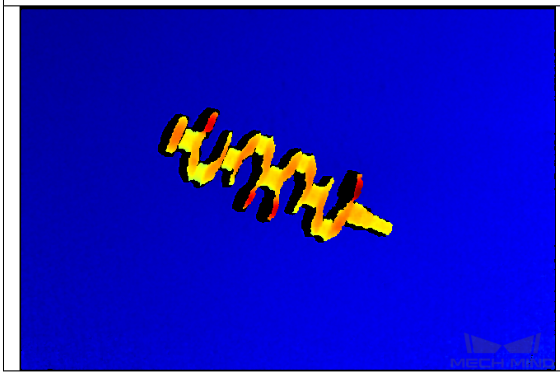
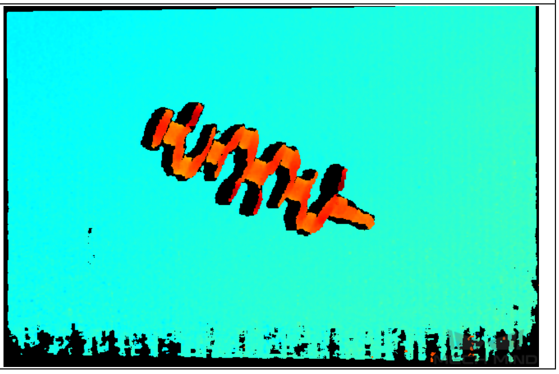
- Set the brightness of the projected structured light.
- Options:
  - **High**: High brightness. For dark objects.
  - **Normal**: Normal brightness. For objects of regular surfaces.
  - **Low**: Low brightness. For reflective objects.

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):

Accurate	Fast
	

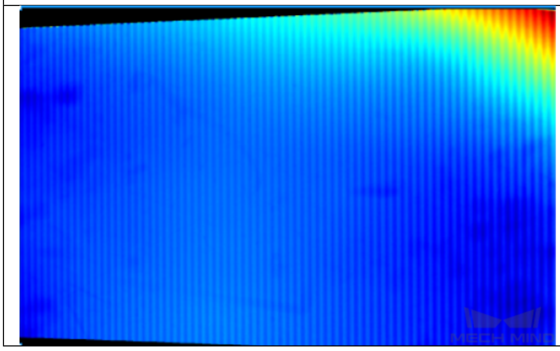
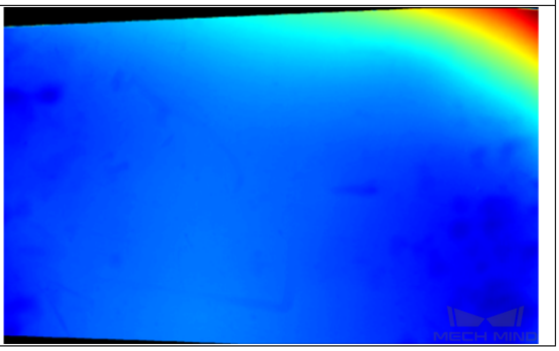
Anti-Flicker Mode

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):

Off	On
	

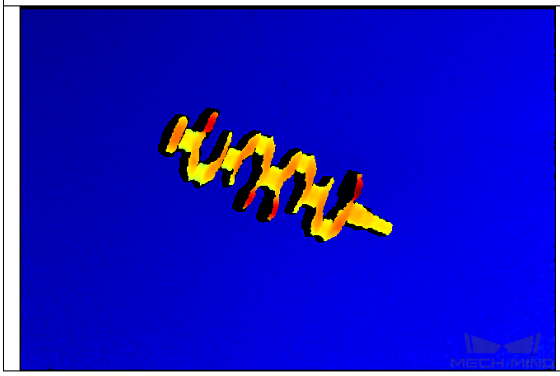
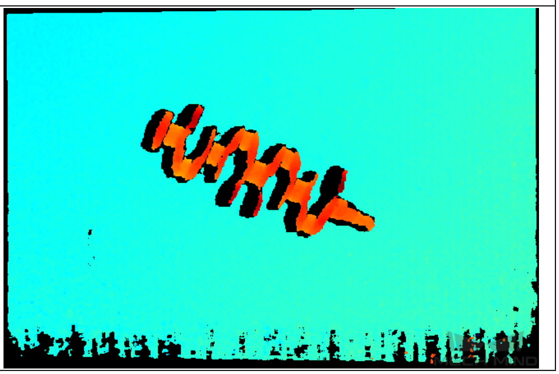
Laser

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):

Accurate	Fast
	

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: 50–100% (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.

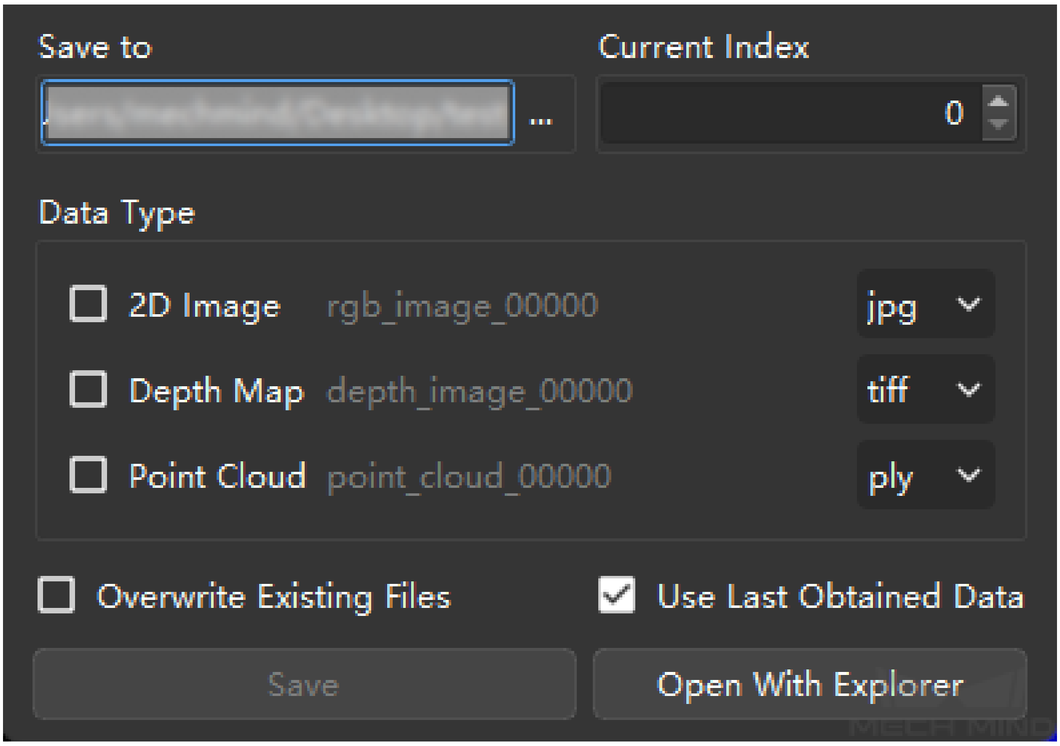
7.4 Save Data

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.

Note: For DEEP (V4) and LSR (V4) series, only the 2D image (texture) can be saved. The 2D images (depth source) cannot be saved.



Save to: [redacted] ...

Current Index: 0

Data Type

☐ 2D Image rgb_image_00000 jpg

☐ Depth Map depth_image_00000 tiff

☐ Point Cloud point_cloud_00000 ply

☐ Overwrite Existing Files ☑ Use Last Obtained Data

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 *Save*.
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

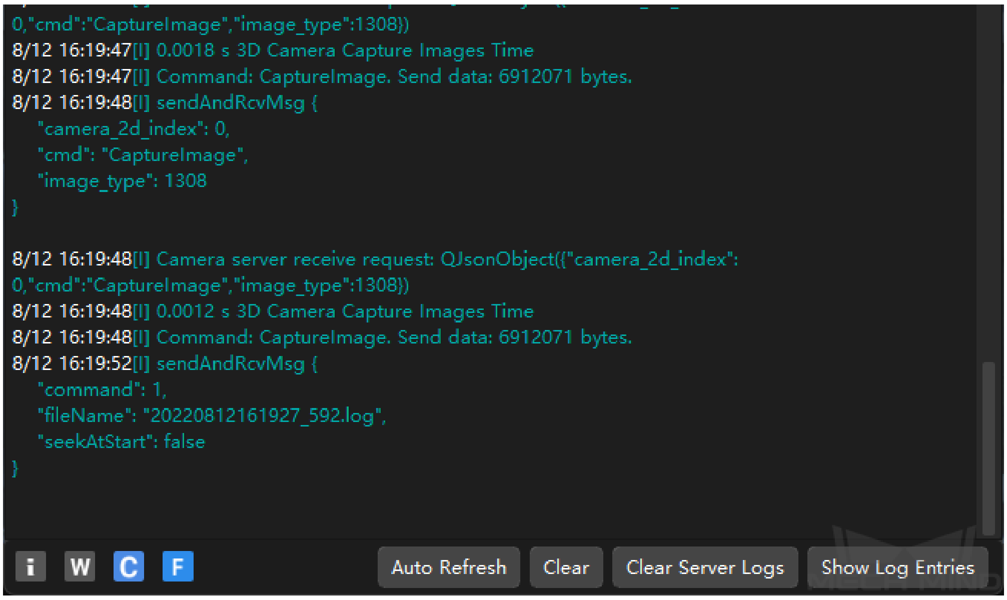
You can save the raw data captured by the camera in MRAW format by clicking *File → Save Raw Data of Camera*.

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

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

- *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).

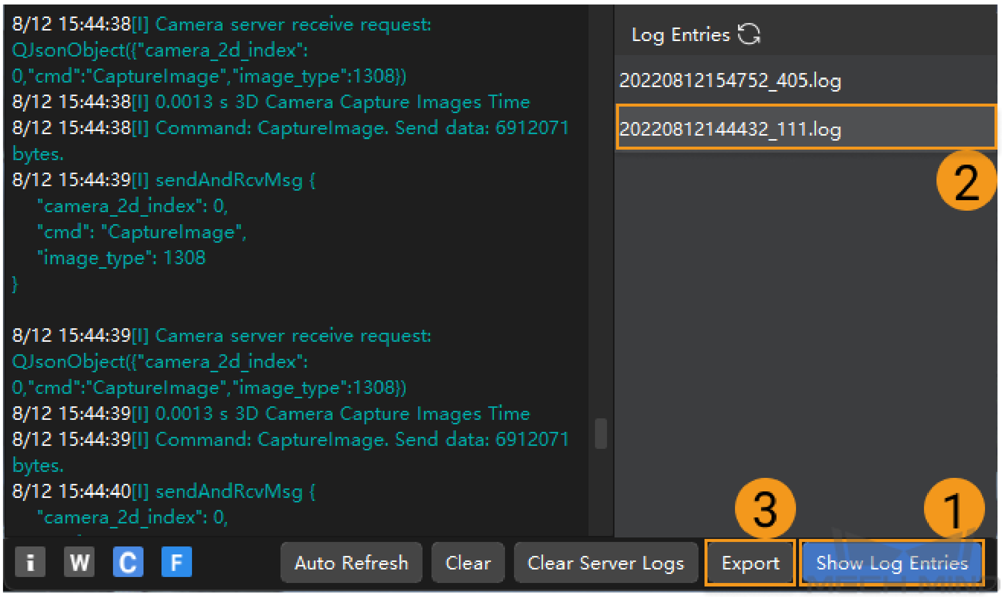
7.5.1 Log Level

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.



```
8/12 15:44:38[I] Camera server receive request:
QJsonObject({"camera_2d_index":
0,"cmd":"CaptureImage","image_type":1308})
8/12 15:44:38[I] 0.0013 s 3D Camera Capture Images Time
8/12 15:44:38[I] Command: CaptureImage. Send data: 6912071
bytes.
8/12 15:44:39[I] sendAndRcvMsg {
    "camera_2d_index": 0,
    "cmd": "CaptureImage",
    "image_type": 1308
}

8/12 15:44:39[I] Camera server receive request:
QJsonObject({"camera_2d_index":
0,"cmd":"CaptureImage","image_type":1308})
8/12 15:44:39[I] 0.0013 s 3D Camera Capture Images Time
8/12 15:44:39[I] Command: CaptureImage. Send data: 6912071
bytes.
8/12 15:44:40[I] sendAndRcvMsg {
    "camera_2d_index": 0,
```

Log Entries

20220812154752_405.log

20220812144432_111.log

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

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

CHAPTER

EIGHT

BUILT-IN TOOLS

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

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

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

Note: For projects with high accuracy requirements, you can enable the high-accuracy mode, in which stricter standards are used to check the intrinsic parameters. High-accuracy mode is available in the **Administrator** account. If needed, please contact Technical Support.

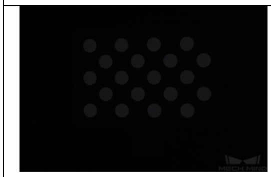
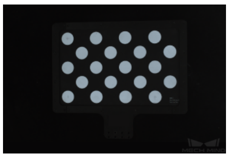
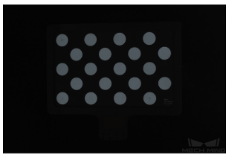
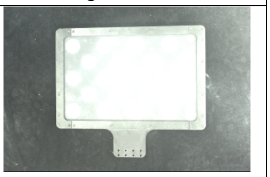
8.1.1 Preparation

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

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

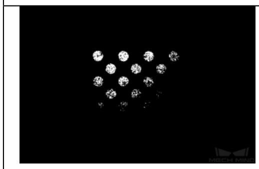
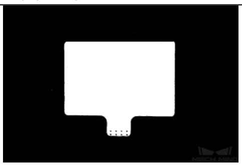
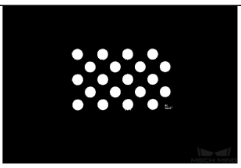
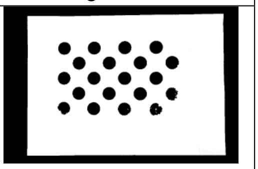
- The region containing the circles on the calibration board is completely captured.
- The 2D image is not too bright or too dark, and the circles on the calibration board are clear and complete.
- The circles on the calibration board in the depth map are complete.

Examples of 2D images of the calibration board:

Too dark	Appropriate		Too bright
			

If the obtained 2D image does not satisfy the requirements, please adjust the *parameters for the 2D image*.

Examples of depth maps of the calibration board:

Too dark	Appropriate		Too bright
			

If the obtained depth map does not satisfy the requirements, please adjust the *parameters for the depth map*.

8.1.2 Check Intrinsic Parameters

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

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

1. In **1. Place calibration board and check image quality**, click *Acquire images* to obtain the 2D image and depth map of the calibration board. Check if the images satisfy the requirements.
2. In **2. Select model of placed calibration board**, set the specifications of the calibration board.
   - If using a standard calibration board, select the model from the drop-down menu.

- If using a customized calibration board, select **Customized** first, and then set the model, rows, columns and spacing according to the information provided by the engineer.

3. In **3. Check intrinsic parameters**, click *Check intrinsic parameters* to start checking the intrinsic parameters. When the check is completed, a window of the check results will pop up.
    - If the check results are displayed in green, then the check is passed, and the camera functions properly.
    - If the check results are displayed in red, or an error message is displayed, then the errors in the intrinsic parameters are large, or the check has failed. Please troubleshoot according to the instructions in the window.

Hint: For projects with high accuracy requirements: switch to the **Administrator** account, check **High-accuracy mode** in **3. Check intrinsic parameters**, and place the calibration board according to the displayed distance. If the calibration board is not placed at the designated distance, the intrinsic parameter check may fail.

8.1.3 Troubleshoot Common Problems

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

Errors in the Intrinsic Parameters Are Large

Symptom:

The intrinsic parameter check results are displayed in red, indicating that the errors in the intrinsic parameters are large.

Solutions:

Please try the following solutions:

1. Check if the calibration board specifications are correct.
    - If incorrect, enter the correct specifications. Then, check the intrinsic parameters again.
    - If correct, proceed to the next step.
2. Check if the 2D image and depth map meet the image requirements. Please refer to the *image requirements*.
    - If the 2D image does not satisfy the requirements, please adjust the *parameters for the 2D image*. If the depth map does not satisfy the requirements, please adjust the *parameters for the depth map*. Then, check the intrinsic parameters again.
    - If the images satisfy the requirements, proceed to the next step.

3. Use the **Correct Intrinsic Parameters** function to correct the intrinsic parameters. Please follow these steps:

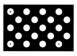
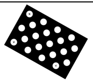
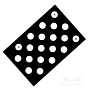
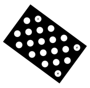
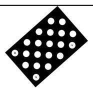
   (1) Place the calibration board within the camera FOV.

   (2) Check the **Correct Intrinsic Parameters** option, and then click *Acquire data*.

   (3) Move the calibration board, and click *Acquire data* again.

   Hint:

   - Please add at least 3 sets of data.
   - When moving the calibration board, make sure the orientation of the board is changed along with the position. The recommended positions are the center and four corners of the FOV, as shown below.

Position 1	Position 2	Position 3	Position 4	Position 5
				

   (4) After adding the data, click *Correct intrinsic parameters*. When the correction is completed, a window of the correction results will pop up.

   - If the correction succeeded, click *Check intrinsic parameters* in this window to check the intrinsic parameters again.
   - If the correction failed, click *Confirm* in this window, and retry the correction or contact Technical Support.

Circles on the Calibration Board not Detected

Symptom:

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

Solutions:

Please try the following solutions:

1. Check if the calibration board specifications are correct.
   - If incorrect, enter the correct specifications. Then, check the intrinsic parameters again.
   - If correct, proceed to the next step.
2. Check if the 2D image and depth map meet the image requirements. Please refer to the *image requirements*.
   - If the 2D image does not satisfy the requirements, please adjust the *parameters for the 2D image*. If the depth map does not satisfy the requirements, please adjust the *parameters for the depth map*. Then, check the intrinsic parameters again.
   - If the images satisfy the requirements, proceed to the next step.

3. Use the **Draw aid circle** to facilitate the detection. Please follow these steps:

   (1) In **3. Check intrinsic parameters**, click *Draw aid circle*.

   (2) Find a complete circle on the calibration board in the 2D image. Move the cursor to the center of this circle. Click while holding the `Ctrl` key and then move diagonally up or down to adjust the size of the aid circle. Click again to finish drawing the aid circle.

   > Note:
   >
   > - Only one aid circle can be drawn. The aid circle should be as close possible to the boundary of the circle on the calibration board.
   > - If you need to re-draw the aid circle, click *Draw aid circle* again.
   > - To zoom the 2D image, move the cursor onto the 2D image and scroll the mouse wheel.

   (3) In **3. Check intrinsic parameters**, click *Check intrinsic parameters* to check the intrinsic parameters again.

   - If the check succeeded, the troubleshooting has been completed.
   - If the intrinsic parameter check result window still displays **Circles on the calibration board not detected**, proceed to the next step.

4. Use the **Edit Feature Detection Parameters** to facilitate the detection. Please follow these steps:

   (1) In **3. Check intrinsic parameters**, click **Edit Feature Detection Parameters**.

   (2) Adjust the displayed feature detection parameters according to the actual situation.

   (3) If not all the circles on the calibration board can be captured, check **Allow absence of some circles**, and adjust the newly displayed feature detection parameters.

   (4) In **3. Check intrinsic parameters**, click *Check intrinsic parameters* to check the intrinsic parameters again.

   - If the intrinsic parameter check result window still displays **Circles on the calibration board not detected**, please contact Technical Support.

Valid Feature Points Not Found in Depth Map

Symptom:

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

Solutions:

Check if the depth map satisfies the requirements. Please refer to the *image requirements*.

- If the depth map does not satisfy the requirements, please adjust the *parameters for the depth map*. Then, check the intrinsic parameters again.
- If the depth map satisfies the requirements, please contact Technical Support.

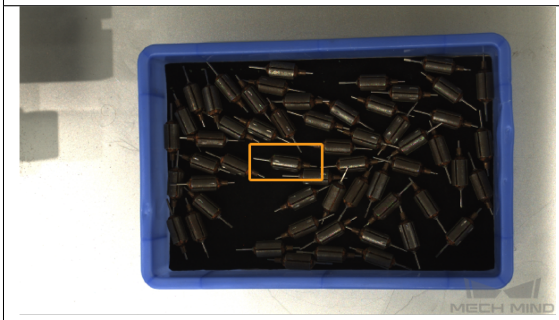
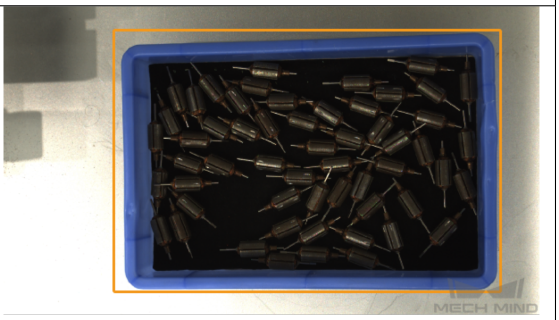
8.2 Exposure Assistant

You can obtain the recommended values for **3D Parameters** with the **Exposure Assistant** tool.

To use this tool to obtain the recommended values for **3D Parameters**, please follow these steps.

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: To obtain better parameter values, the ROI should include only one target object.

Good ROI	Bad ROI
	

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 Check and Configure 2D Camera

This tool is used for checking and configuring the 2D cameras in a Mech-Eye Industrial 3D Camera. You can open the tool via *Tools* → *View 2D Camera and Set Parameters*.

8.3.1 2D Camera Information

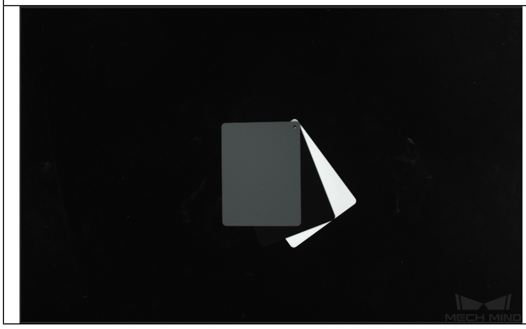
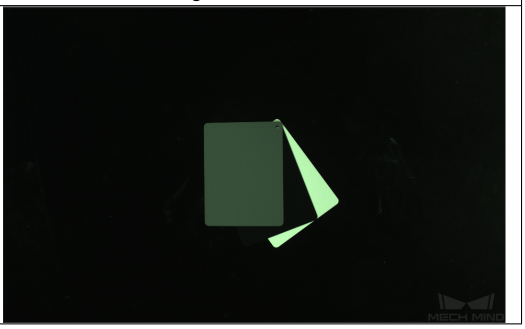
Under **2D Camera Information**, you can view the serial number, resolution, color, device version, and firmware version.

8.3.2 Adjust White Balance

Note: Only applicable to color 2D camera. Select a 2D camera and then check the Color Type in 2D Camera Information to see if it is a color 2D camera.

If the colors in the obtained image are obviously different from the actual object, white balance must be adjusted to correct the colors. Otherwise, the colors of the 2D image and point cloud will be distorted and thus affect further processing. For deep learning, if images with distorted colors are used for model training, the distortions will be extracted and trained as the object features, thus affecting the model performance on object recognition.

Comparison Example

Actual color board	Color board image without white balance
	

Preparation

A gray color board is required for adjusting the white balance. Place the color board close to the camera and perform image capturing to ensure as much as possible that only the color board is captured.

Instructions

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

1. Select the 2D camera whose white balance needs to be adjusted, and click *ViewOn*. The camera will perform image capturing continuously.
2. In 2D Camera Configuration, set AutoWhiteBalance to Once or Continuous, and the camera will start to adjust the white balance.
   - If the lighting conditions remain relatively unchanged, it is recommended to set AutoWhiteBalance to Once.
   - If the lighting conditions vary, it is recommended to set AutoWhiteBalance to Continuous.
3. Check the color of the gray color board in the image.

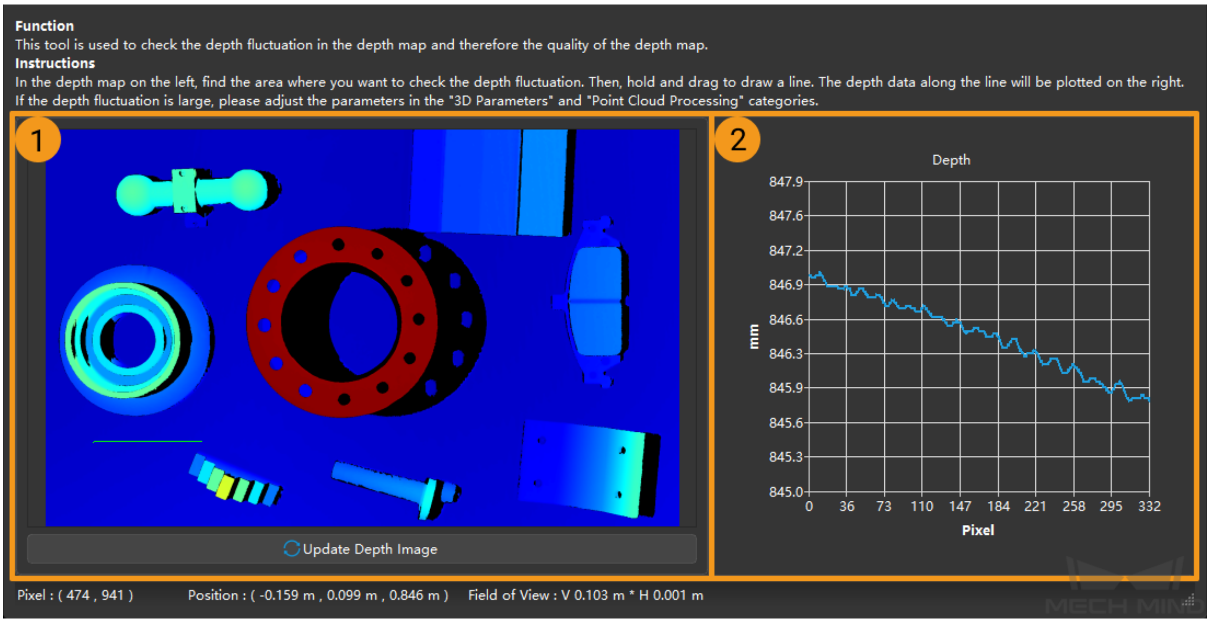
- If the color still appears distorted, move the board until the color appears consistent to the actual color board.
- If the color appears consistent to the acutal color board, set **AutoWhiteBalance** to **Off**, and click *ViewOff* to finish adjusting the white balance.

4. Click *Apply Config* to save the configuration to the camera.

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



Function

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

Instructions

In the depth map on the left, find the area where you want to check the depth fluctuation. Then, hold and drag to draw a line. The depth data along the line will be plotted on the right.
If the depth fluctuation is large, please adjust the parameters in the "3D Parameters" and "Point Cloud Processing" categories.

Pixel : (474 , 941)   Position : (-0.159 m , 0.099 m , 0.846 m)   Field of View : V 0.103 m * H 0.001 m

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

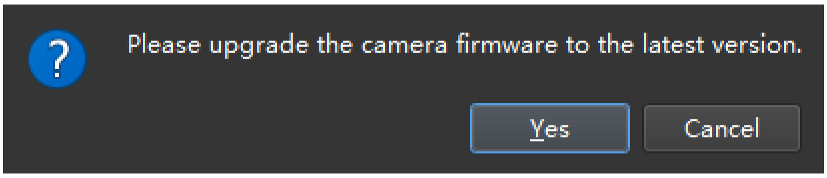
1. On the depth map, locate the area whose depth variation you'd like to check. Draw a line across this area by holding the left mouse button and drag.
2. A depth plot along the line will be displayed on the right.
3. Analysis:
   - Imagine a cross-section of the scene along the line you drew.
   - The 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.

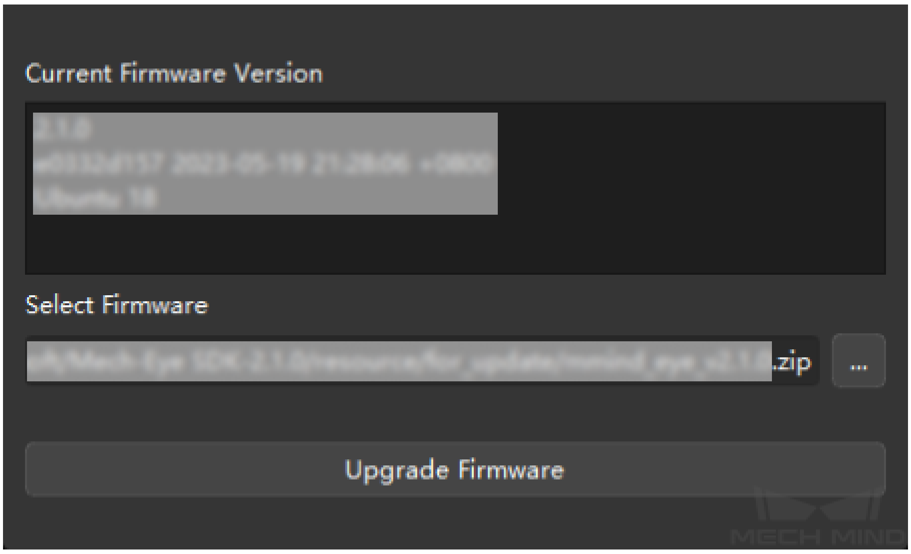

Please upgrade the camera firmware to the latest version.

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.



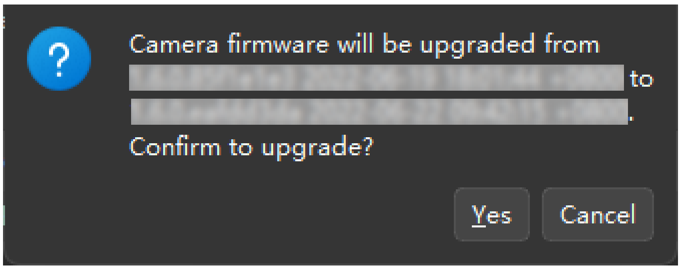
Current Firmware Version

[redacted]

Select Firmware

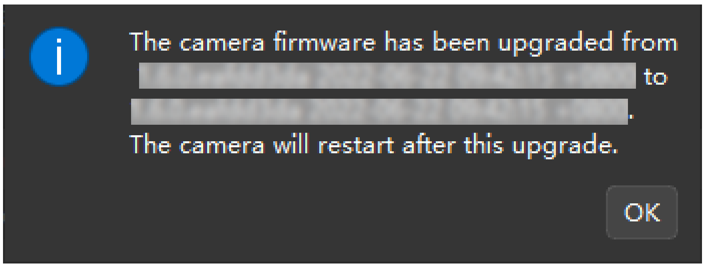
[redacted].zip

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



Camera firmware will be upgraded from [redacted] to [redacted]. Confirm to upgrade?

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



The camera firmware has been upgraded from [redacted] to [redacted]. The camera will restart after this upgrade.

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

The **FOV Calculator** tool is used to help determine the mounting height of the camera based on your needs for camera FOV. Based on the entered working distance, this tool calculates the FOV height and width at this distance.

8.6.1 Calculate Camera FOV

To calculate the camera FOV with the **FOV Calculator** tool, please follow these steps.

1. Select the camera model in the **Select camera model** drop-down menu.

   Hint: The frequently used models are displayed on the first page. Click **Other models** to view the rest of the models.

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

   Note:

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

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

8.6.2 Camera Working Distance

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

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

8.7 Custom Coordinate System

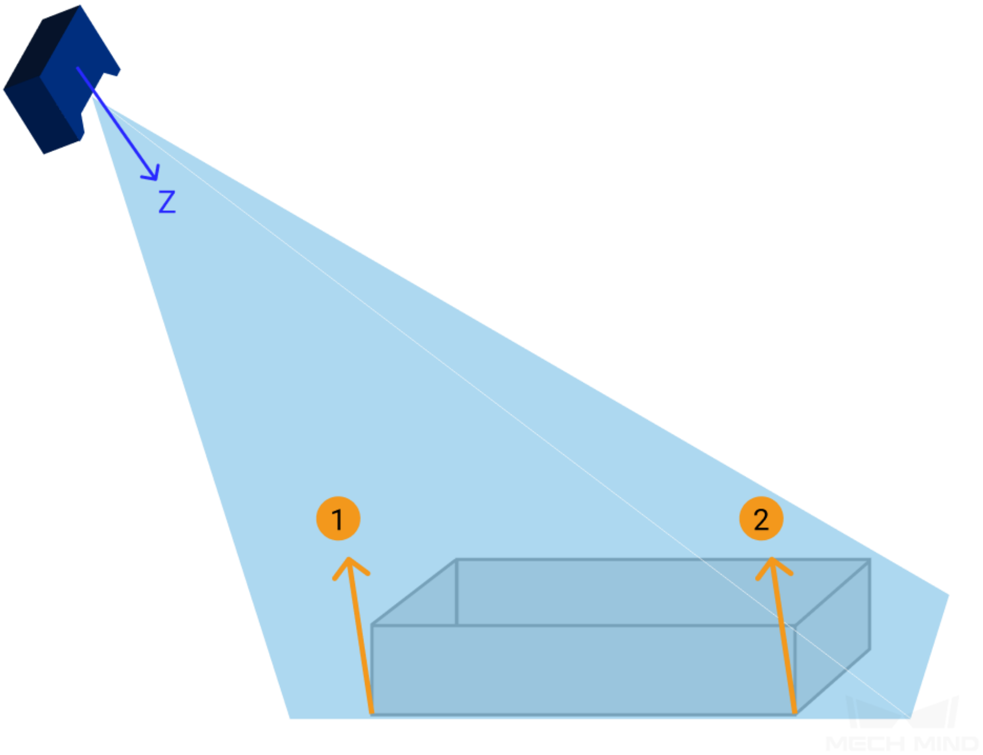
You can define a custom reference frame in Custom Coordinate System for displaying the depth information of depth maps and point clouds.

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

Note: The custom reference frame is for display in Mech-Eye Viewer only. The depth values stored in the saved depth map and point cloud are unchanged.

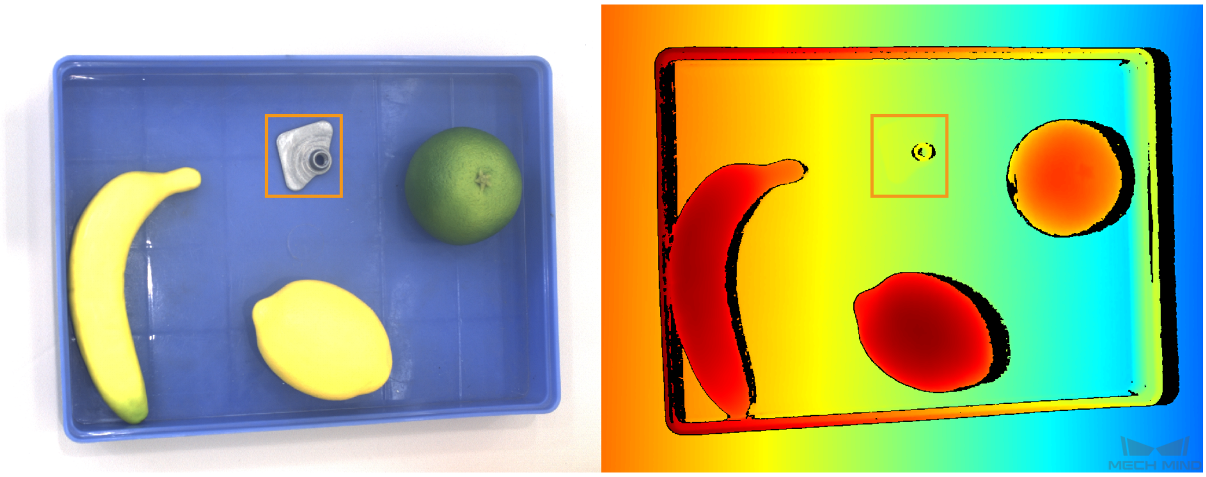
8.7.1 Suitable Scenarios

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 discrepency 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.

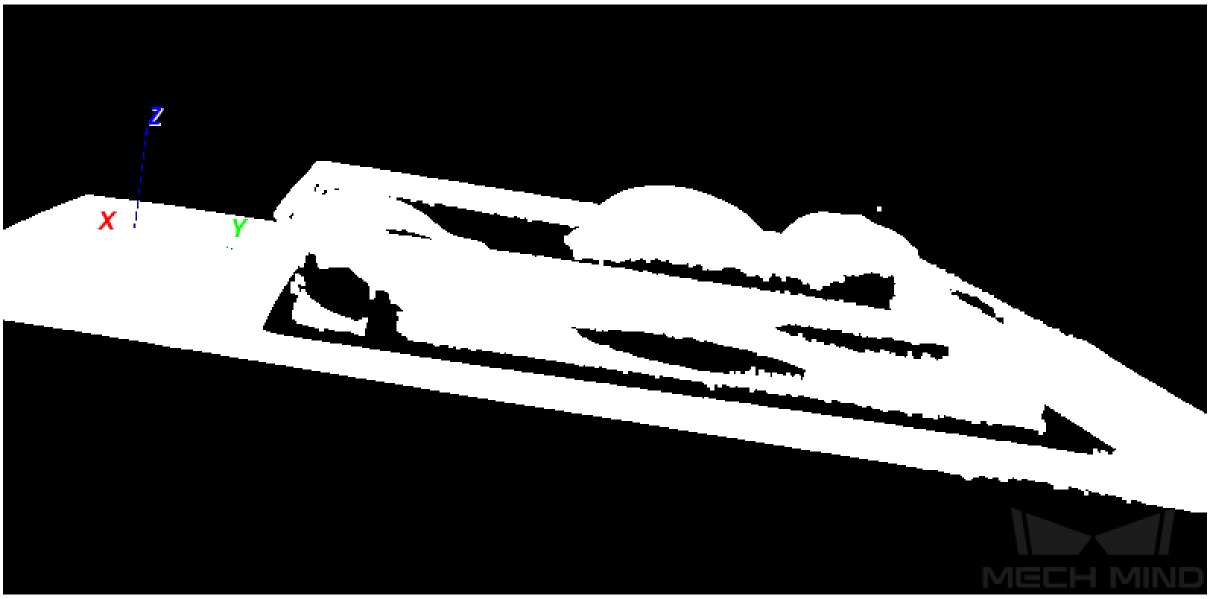
2. After selecting the three points, the axes of the custom reference frame will appear on the point cloud. Rotate the point cloud to check:

   - 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.

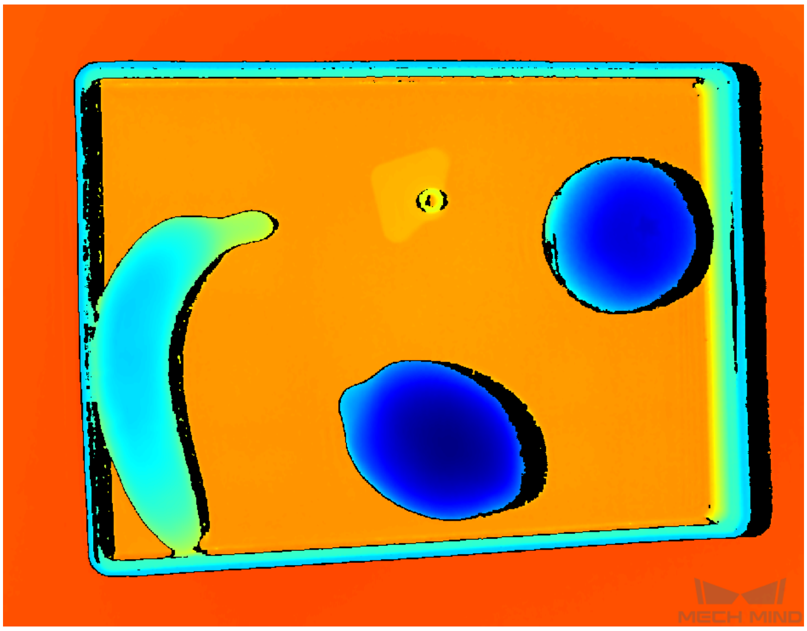
3. Click *OK* to save the custom reference frame.

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

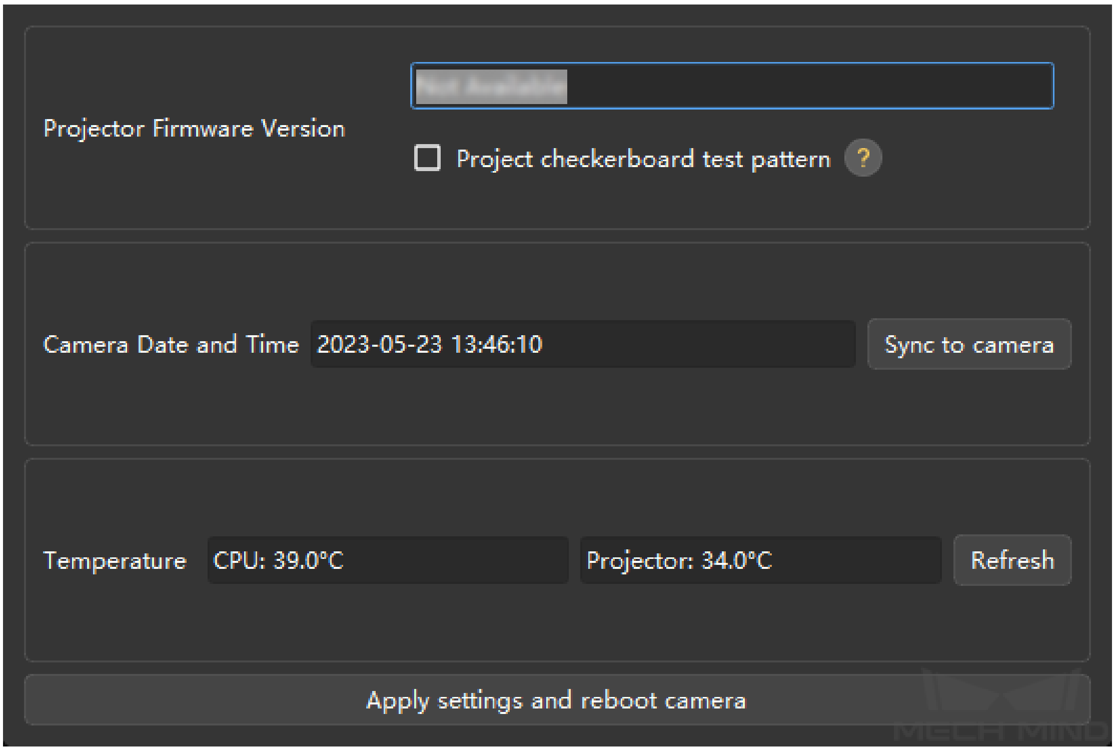
1. Once you have a custom reference frame defined, perform image capturing again with 📷 or 🔄.
2. In the Image Display area, select **Depth Map** or **Point Cloud**. In the left panel, select **Customized** from the drop-down menu of **Coordinate System**.
3. Now the depth information is displayed in 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 Firmware Version [redacted]

☐ Project checkerboard test pattern ?

Camera Date and Time 2023-05-23 13:46:10

Temperature CPU: 39.0°C Projector: 34.0°C

- **Projector Firmware Version**: display the projector firmware version of the connected camera.
- **Project chessboard test pattern**: Check this option to project a chessboard pattern used for checking the connection and focus of the projector.
  - If the pattern is successfully projected, the project is connected.
  - If the pattern is clear and not blurry, the projector is in focus.
  - If the pattern is not projected or is blurry, please contact Technical Support.

  Note: This option is only available for DLP cameras.

- **Camera Date and Time**: display the date and time recorded in the camera.

  Click *Sync to camera* 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.
- **Resolution**: Change the resolution of the 2D image and depth map obtained by the camera. Click *Apply settings and reboot camera* after making changes. The changes will take effect after the camera restarts.

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

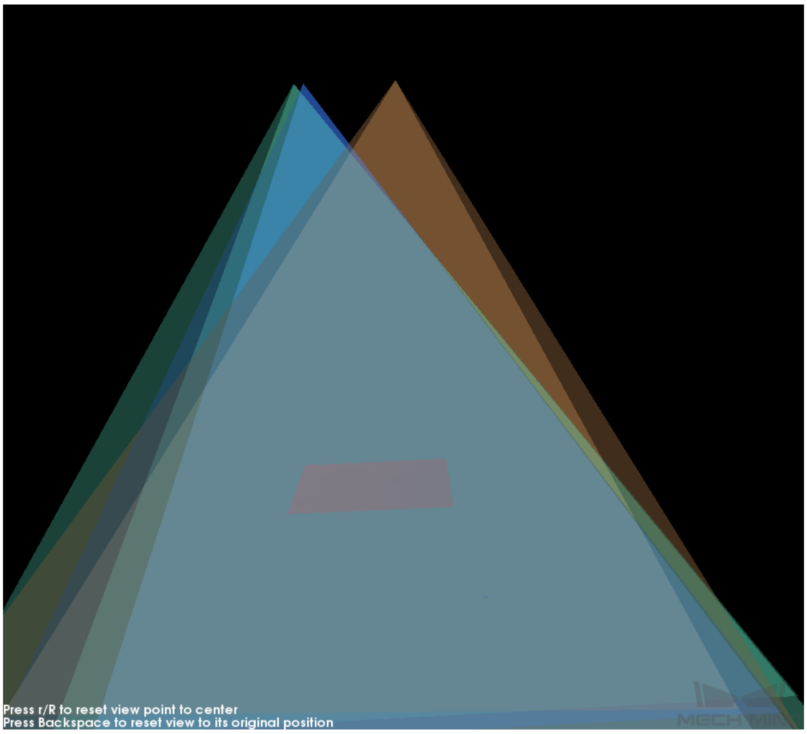
Before you start using **Full Pallet Simulator**, have the following ready at hand:

- 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*.

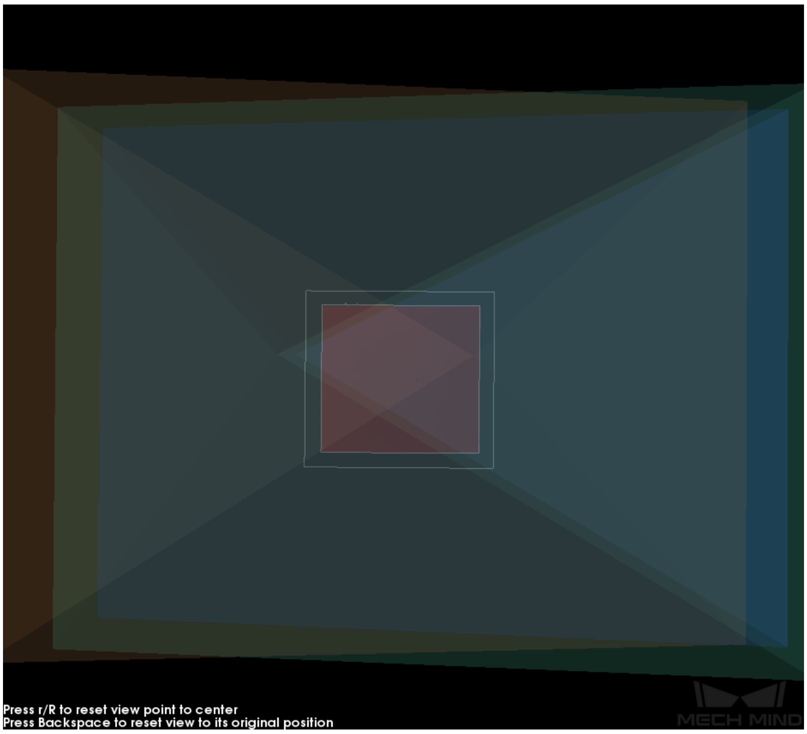


Press r/R to reset view point to center
Press Backspace to reset view to its original position

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.

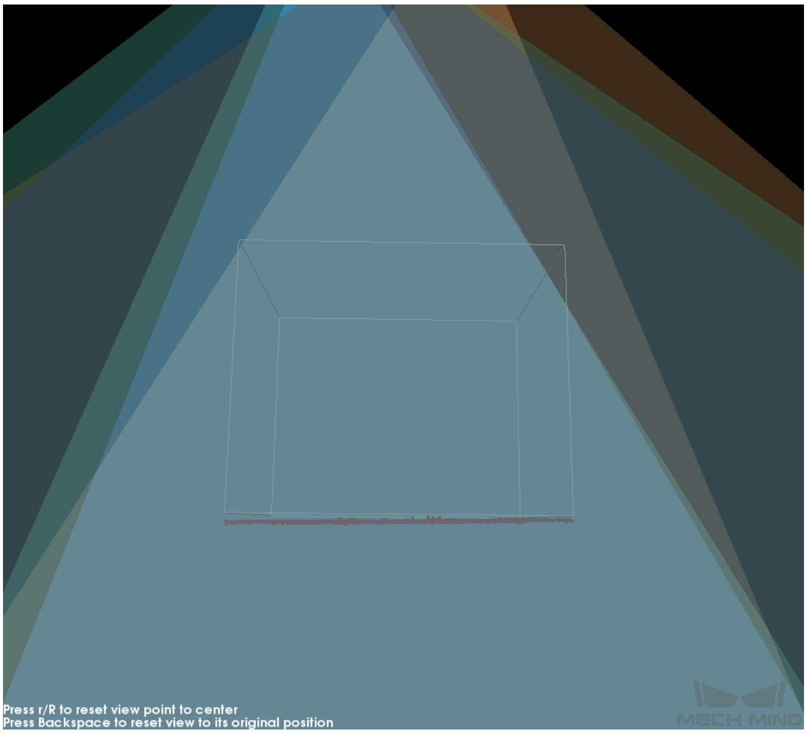
You can rotate the point cloud and view from the top to see if the cuboid is aligned with the point cloud:



Press r/R to reset view point to center
Press Backspace to reset view to its original position

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.

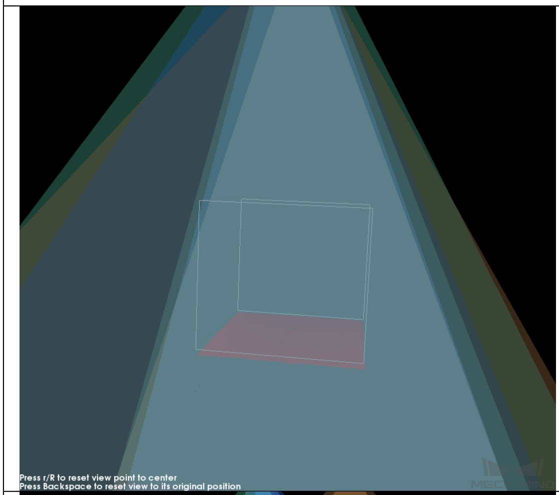
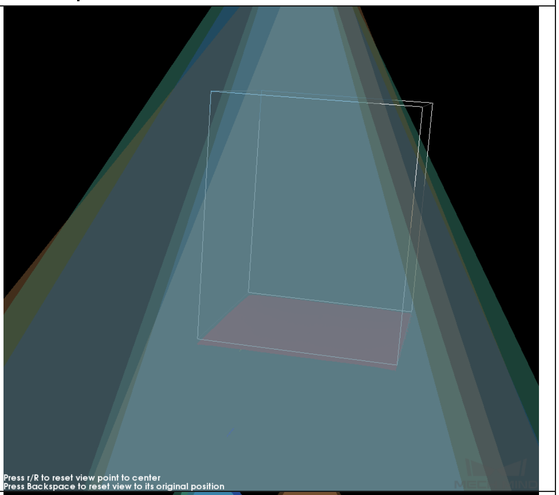
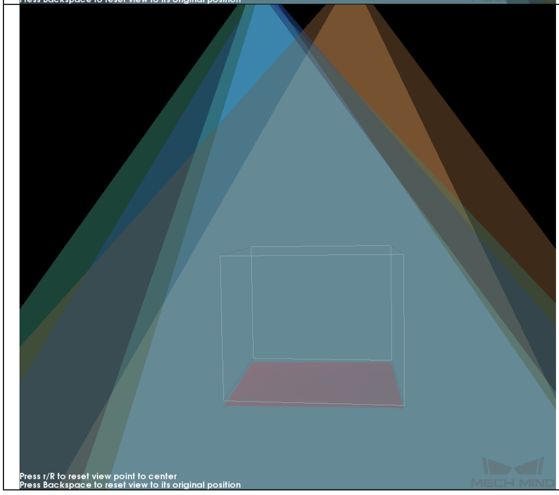
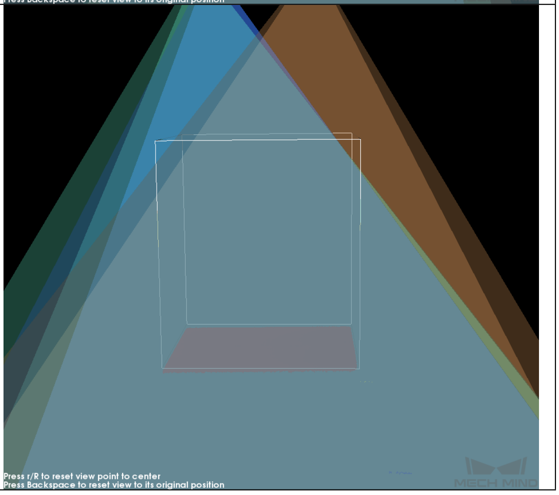
Note: Record the amount of z value adjustment made.



Press r/R to reset view point to center
Press Backspace to reset view to its original position

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.

Top of cuboid entirely within view cone overlap	Top of cuboid not entirely within view cone overlap
 Press r/R to reset view point to center Press Backspace to reset view to its original position	 Press r/R to reset view point to center Press Backspace to reset view to its original position
 Press r/R to reset view point to center Press Backspace to reset view to its original position	 Press r/R to reset view point to center Press Backspace to reset view to its original position

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.

CHAPTER

NINE

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

Instructions on using **Mech-Eye API Tools**:

Mech-Eye API: Camera Firmware Upgrade Tool

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: 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
```

CHAPTER

TEN

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:: FringeContrastThreshold	PointCloudProcessingSettings:: fringeContrastThreshold
PointCloudProcessingSettings::FringeMinThreshold	PointCloudProcessingSettings::fringeMinThreshold
PointCloudProcessingSettings:: CloudOutlierFilterMode::OutlierFilterMode	PointCloudProcessingSettings:: CloudOutlierFilterMode::outlierFilterMode
PointCloudProcessingSettings:: CloudSmoothMode::SmoothMode	PointCloudProcessingSettings:: CloudSmoothMode::smoothMode
LaserSettings::LaserFringeCodingMode:: FringeCodingMode	LaserSettings::LaserFringeCodingMode:: fringeCodingMode
LaserSettings::FrameRangeStart	LaserSettings::frameRangeStart
LaserSettings::FrameRangeEnd	LaserSettings::frameRangeEnd
LaserSettings::FramePartitionCount	LaserSettings::framePartitionCount
LaserSettings::PowerLevel	LaserSettings::powerLevel
UhpSettings::UhpCaptureMode::CaptureMode	UhpSettings::UhpCaptureMode::captureMode
UhpSettings::UhpFringeCodingMode:: FringeCodingMode	UhpSettings::UhpFringeCodingMode:: fringeCodingMode

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.

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 (System.Collections.Generic.List<double>)	MechEyeDevice.SetScan3DExposure (System.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);
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)
```

CHAPTER

ELEVEN

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.

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.
- **Calibration**: perform hand-eye calibration through Mech-Eye API.
- **Halcon**: obtain HALCON-readable point clouds through 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, firmware version and temperatures.
- 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.

Calibration

- HandEyeCalibration: Perform hand-eye calibration.

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*

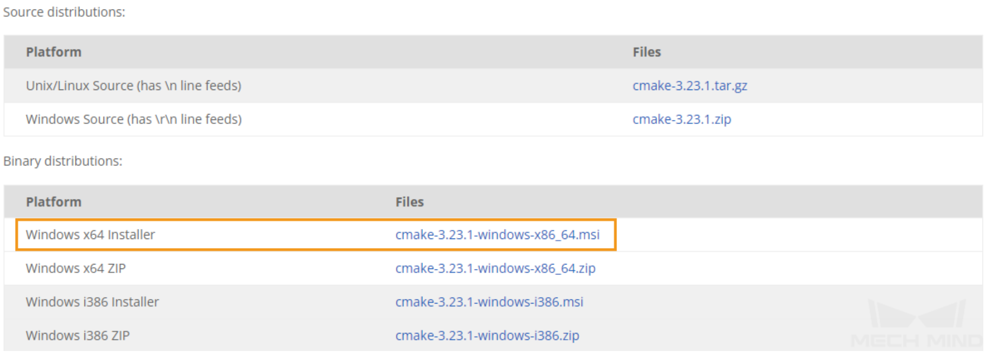
Install Required Software

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:

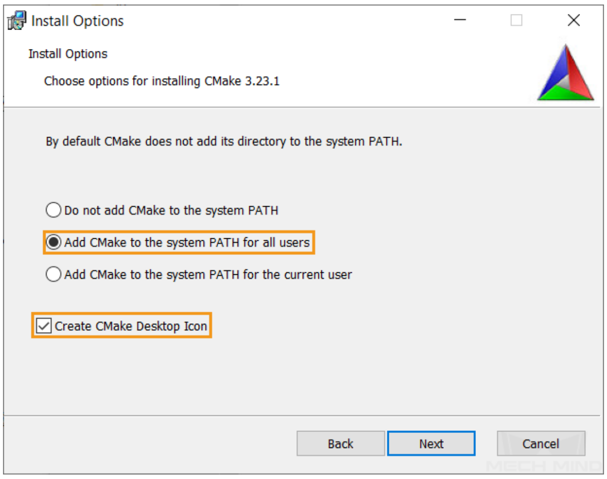
Platform	Files
Unix/Linux Source (has \n line feeds)	cmake-3.23.1.tar.gz
Windows Source (has \r\n line feeds)	cmake-3.23.1.zip

Binary distributions:

Platform	Files
Windows x64 Installer	cmake-3.23.1-windows-x86_64.msi
Windows x64 ZIP	cmake-3.23.1-windows-x86_64.zip
Windows i386 Installer	cmake-3.23.1-windows-i386.msi
Windows i386 ZIP	cmake-3.23.1-windows-i386.zip

2. When installing, select the following two options to add CMake to environment variables and create a desktop shortcut for CMake.

- Add CMake to the system PATH for all users
- Create Cmake Desktop Icon



Install Options

Install Options

Choose options for installing CMake 3.23.1

By default CMake does not add its directory to the system PATH.

○ Do not add CMake to the system PATH

● Add CMake to the system PATH for all users

○ Add CMake to the system PATH for the current user

☑ Create CMake Desktop Icon

Install Visual Studio (version 2017 or above)

1. Download the Visual Studio installer.
2. When installing, select the following two workloads in the **Desktop & Mobile** category, and then click *Install*.
   - **Desktop development with C++**
   - **Universal Windows Platform development**

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 1.8.1 or above
- HALCON: version 20.11 or above

Note: HALCON versions below 20.11 are not fully tested.

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

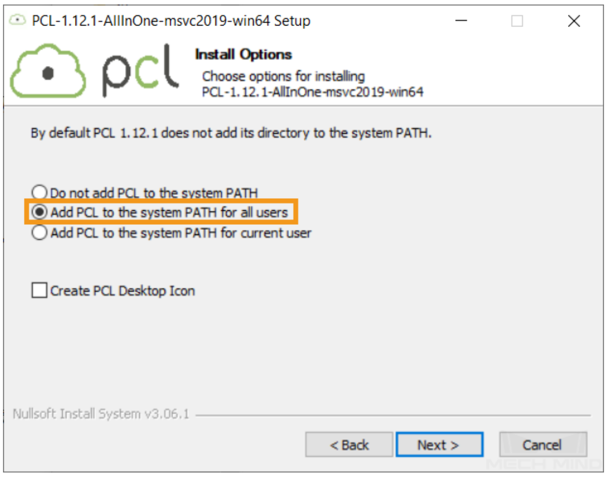
1. Download and install OpenCV. Note down the installation directory of OpenCV.
2. Add OpenCV to environment variables. Refer to *Add Environment Variables* for instructions.

Install PCL

1. Refer to the following table and determine the version of PCL to install based on the version of Visual Studio.

Visual Studio version	Supported PCL versions
2017	1.8.1−1.9.1
2019	1.8.1−1.12.1
2022	1.8.1 and above

2. Open the PCL download page, click on **Assets** under the corresponding version, and download the EXE installer.
3. Run the installer to install PCL. When installing, select **Add PCL to the system PATH for all users** to add PCL to environment variables.



PCL-1.12.1-AllInOne-msvc2019-win64 Setup

Install Options

Choose options for installing PCL-1.12.1-AllInOne-msvc2019-win64

By default PCL 1.12.1 does not add its directory to the system PATH.

○ Do not add PCL to the system PATH
● Add PCL to the system PATH for all users
○ Add PCL to the system PATH for current user

□ Create PCL Desktop Icon

Nullsoft Install System v3.06.1

> 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.

4. Add OpenNI2 on which PCL depends to environment variables. Refer to *Add Environment Variables* for instructions.

Install HALCON Install HALCON 20.11 or above. Please confirm that your HALCON license is still valid.

Add Environment Variables

If you need to use samples that depend on OpenCV or PCL, please add the relevant environment variables according to the following steps.

1. Right-click **This PC** on the desktop and select **Properties**.
2. Click **Advanced system settings**, and 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)

> **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

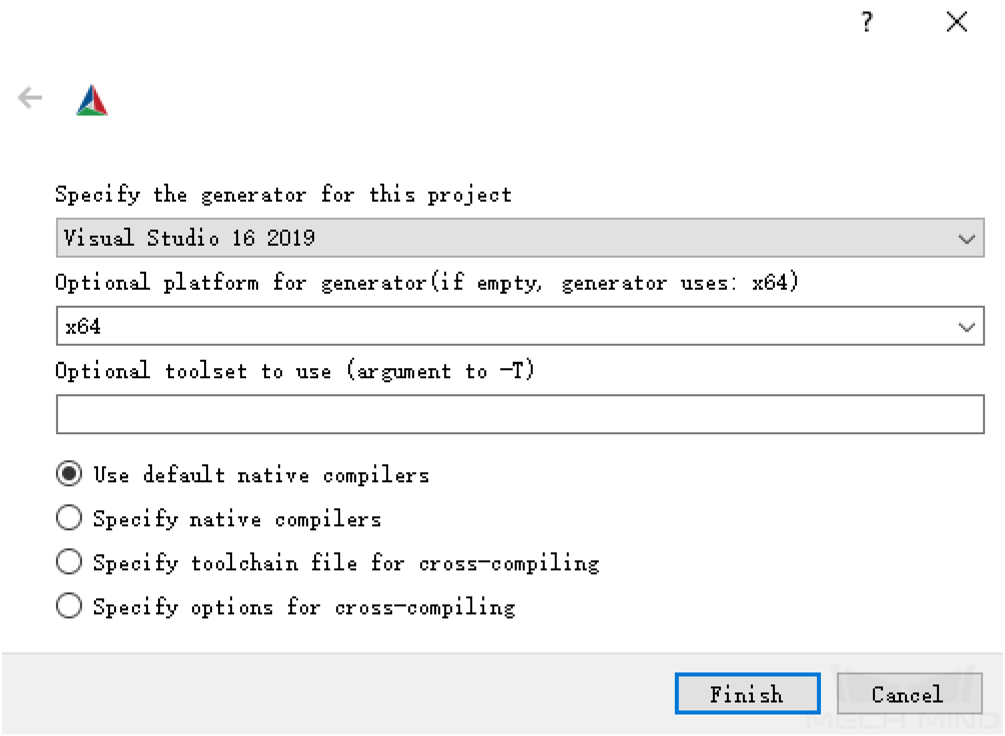
1. Right-click CMake and select **Run as administrator**.
2. Enter the directories of source code and binaries.
   - To build all samples at once, enter the following directories.

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 to open the configuration page.
4. Select Visual Studio version, and set the platform to **x64**. Then, click *Finish*. When the configuration completes, the log will display **Configuring done**.



Specify the generator for this project

Visual Studio 16 2019

Optional platform for generator(if empty, generator uses: x64)

x64

Optional toolset to use (argument to -T)

● Use default native compilers

○ Specify native compilers

○ Specify toolchain file for cross-compiling

○ Specify options for cross-compiling

5. 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.

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.
- **Calibration**: perform hand-eye calibration through Mech-Eye API.
- **Halcon**: obtain HALCON-readable point clouds through 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, firmware version and temperatures.
- 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.

Calibration

- HandEyeCalibration: Perform hand-eye calibration.

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

> Note: HALCON versions below 20.11 are not fully tested.

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
```

Note: On different versions of Ubuntu, this command installs different versions of PCL. On Ubuntu 18.04, PCL 1.8.1 is installed; on Ubunt 20.04, PCL 1.10.0 is installed.

- 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 actuall installation directory of HALCON.

```
HALCONARCH=x64-linux; export HALCONARCH
HALCONROOT="/opt/halcon"; export HALCONROOT
HALCONEXAMPLES=${HALCONROOT}/examples; export HALCONEXAMPLES
```

(continues on next page)

(continued from previous page)

```
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
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, firmware version and temperatures.

- 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.

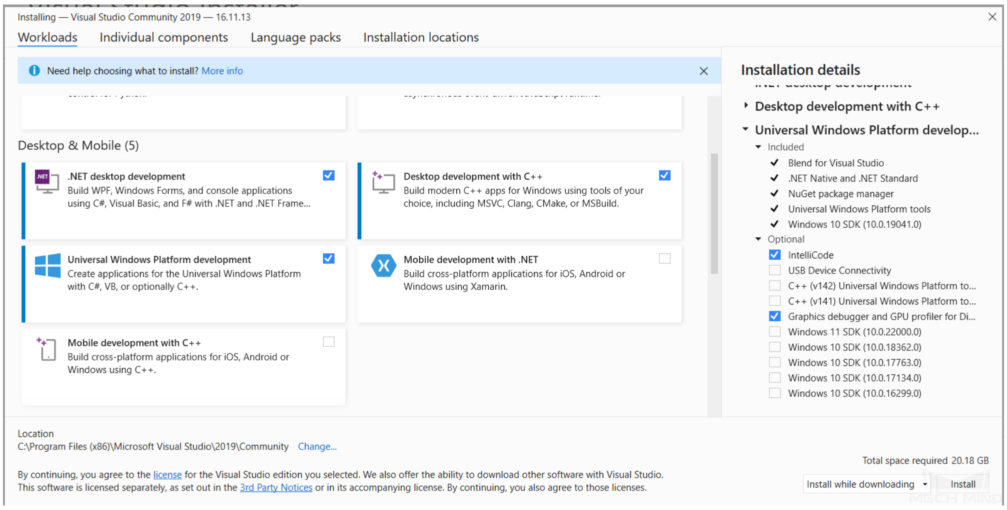
11.3.2 Requirements

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



Installing — Visual Studio Community 2019 — 16.11.13

Need help choosing what to install? More info

Desktop & Mobile (5)

☑ .NET desktop development
Build WPF, Windows Forms, and console applications using C#, Visual Basic, and F# with .NET and .NET Frame...

☑ Desktop development with C++
Build modern C++ apps for Windows using tools of your choice, including MSVC, Clang, CMake, or MSBuild.

☑ Universal Windows Platform development
Create applications for the Universal Windows Platform with C#, VB, or optionally C++.

□ Mobile development with .NET
Build cross-platform applications for iOS, Android or Windows using Xamarin.

□ Mobile development with C++
Build cross-platform applications for iOS, Android or Windows using C++.

Installation details

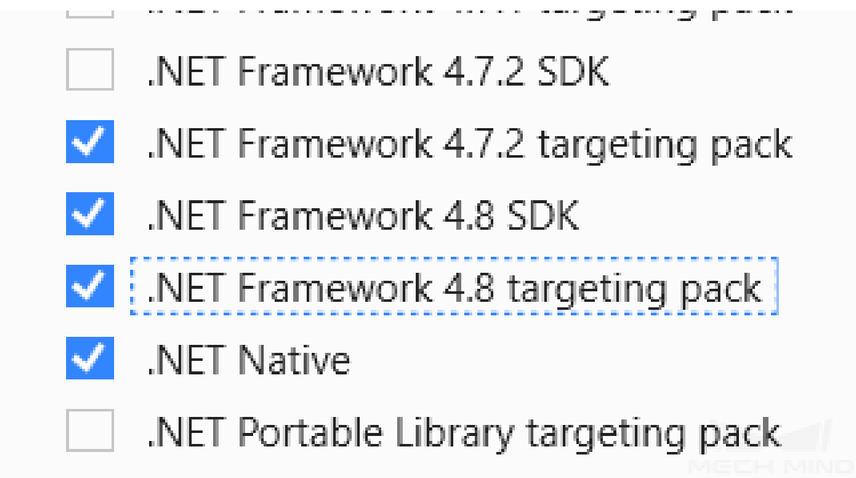
- Desktop development with C++
- Universal Windows Platform develop...
  - Included
    - ✓ Blend for Visual Studio
    - ✓ .NET Native and .NET Standard
    - ✓ NuGet package manager
    - ✓ Universal Windows Platform tools
    - ✓ Windows 10 SDK (10.0.19041.0)
  - Optional
    - ☑ IntelliCode
    - □ USB Device Connectivity
    - □ C++ (v142) Universal Windows Platform to...
    - □ C++ (v141) Universal Windows Platform to...
    - ☑ Graphics debugger and GPU profiler for Di...
    - □ Windows 11 SDK (10.0.22000.0)
    - □ Windows 10 SDK (10.0.18362.0)
    - □ Windows 10 SDK (10.0.17763.0)
    - □ Windows 10 SDK (10.0.17134.0)
    - □ Windows 10 SDK (10.0.16299.0)

Location
C:\Program Files (x86)\Microsoft Visual Studio\2019\Community Change...

By continuing, you agree to the license for the Visual Studio edition you selected. We also offer the ability to download other software with Visual Studio. This software is licensed separately, as set out in the 3rd Party Notices or in its accompanying license. By continuing, you also agree to those licenses.

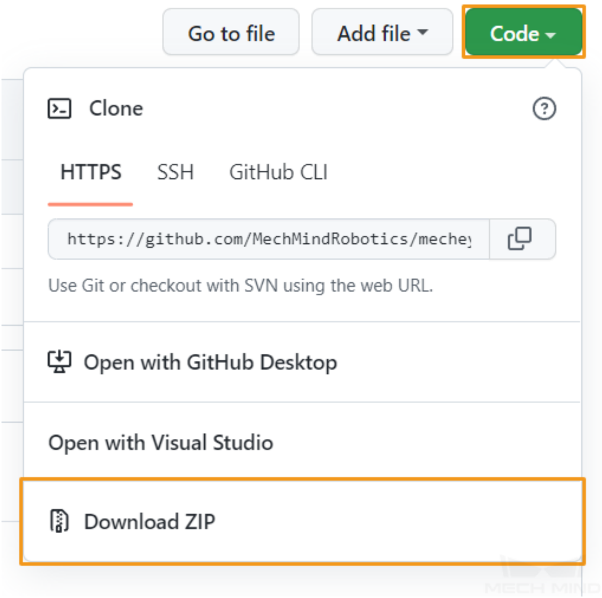
Total space required 20.18 GB

Individual components: .NET Framework 4.8 targeting pack



- □ .NET Framework 4.7.2 SDK
- ☑ .NET Framework 4.7.2 targeting pack
- ☑ .NET Framework 4.8 SDK
- ☑ .NET Framework 4.8 targeting pack
- ☑ .NET Native
- □ .NET Portable Library targeting pack

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

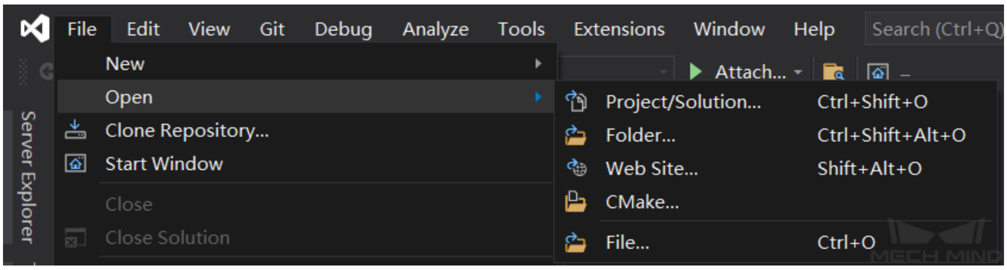


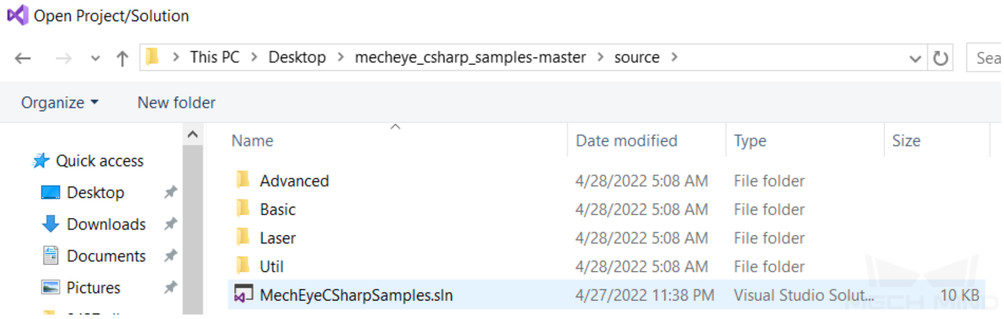
5. *Connect a functioning camera*
6. To use the following samples, Emgu.CV.runtime.windows must be installed in Visual Studio via *Project* → *Manage NuGet Packages*. Please refer to Install and manage packages in Visual Studio using the NuGet Package Manager for detailed instructions.
    - CaptureColorMap
    - CaptureDepthMap
    - CapturePointCloud
    - CaptureHDRPointCloud
    - CapturePointCloudROI
    - CapturePointCloudFromTextureMask
    - CaptureCloudFromDepth
    - CaptureSequentiallyMultiCamera
    - CaptureSimultaneouslyMultiCamera
    - CaptureTimedAndPeriodically

11.3.3 Instructions

Build Solution in Visual Studio

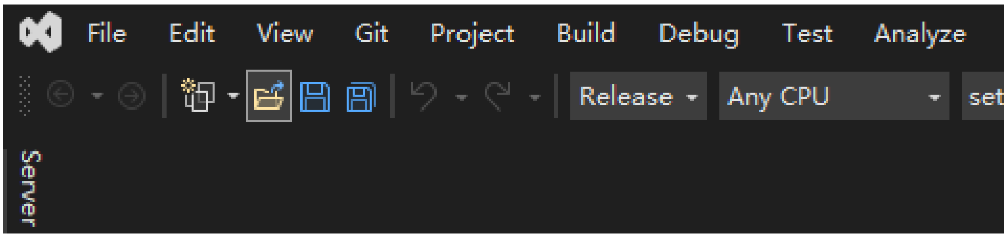
1. Open MechEyeCSharpSamples.sln in Visual Studio.





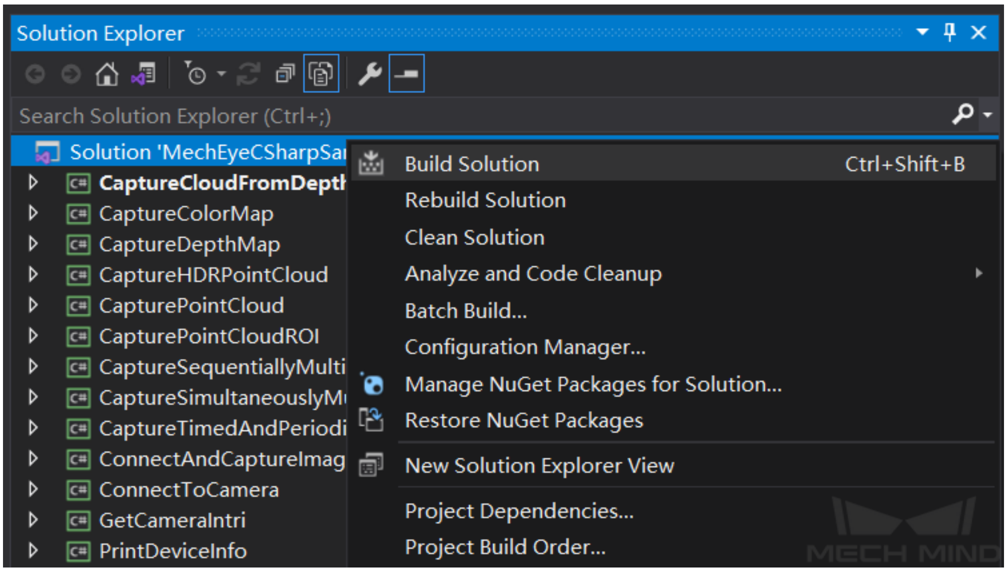
Name	Date modified	Type	Size
Advanced	4/28/2022 5:08 AM	File folder	
Basic	4/28/2022 5:08 AM	File folder	
Laser	4/28/2022 5:08 AM	File folder	
Util	4/28/2022 5:08 AM	File folder	
MechEyeCSharpSamples.sln	4/27/2022 11:38 PM	Visual Studio Solut...	10 KB

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

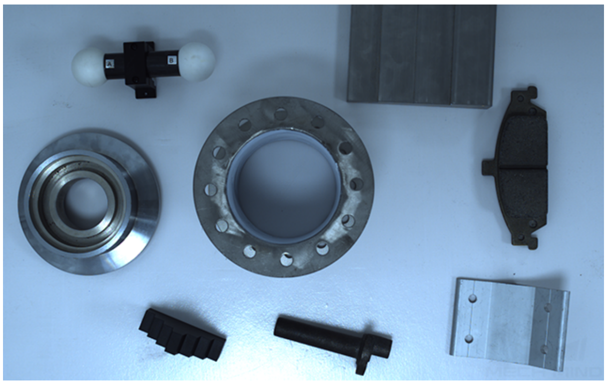
The example below shows how to run CaptureColorMap.exe.

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, firmware version and temperatures.

- 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

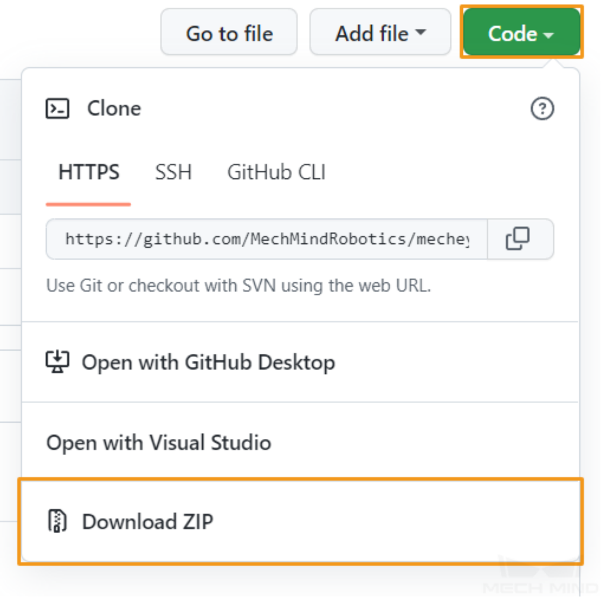
- SetLaserFramePartitionCount: Divide the projector FOV into partitions and project structured light in one partition at a time. The output of the entire FOV is composed from images of all partitions.
- SetLaserFrameRange: Set the projection range of the structured light. The entire projector FOV is 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 and opencv-python.

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

```
pip install opencv-python
```

Run Sample Program

The example below shows how to run CaptureColorMap.py.

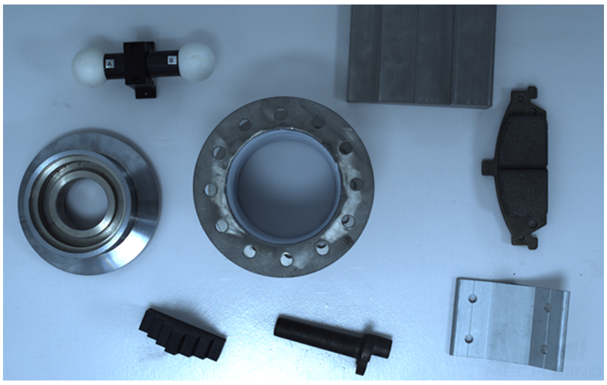
1. Navigate to the directory where CaptureColorMap.py is located.

```
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, firmware version and temperatures.
- 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.

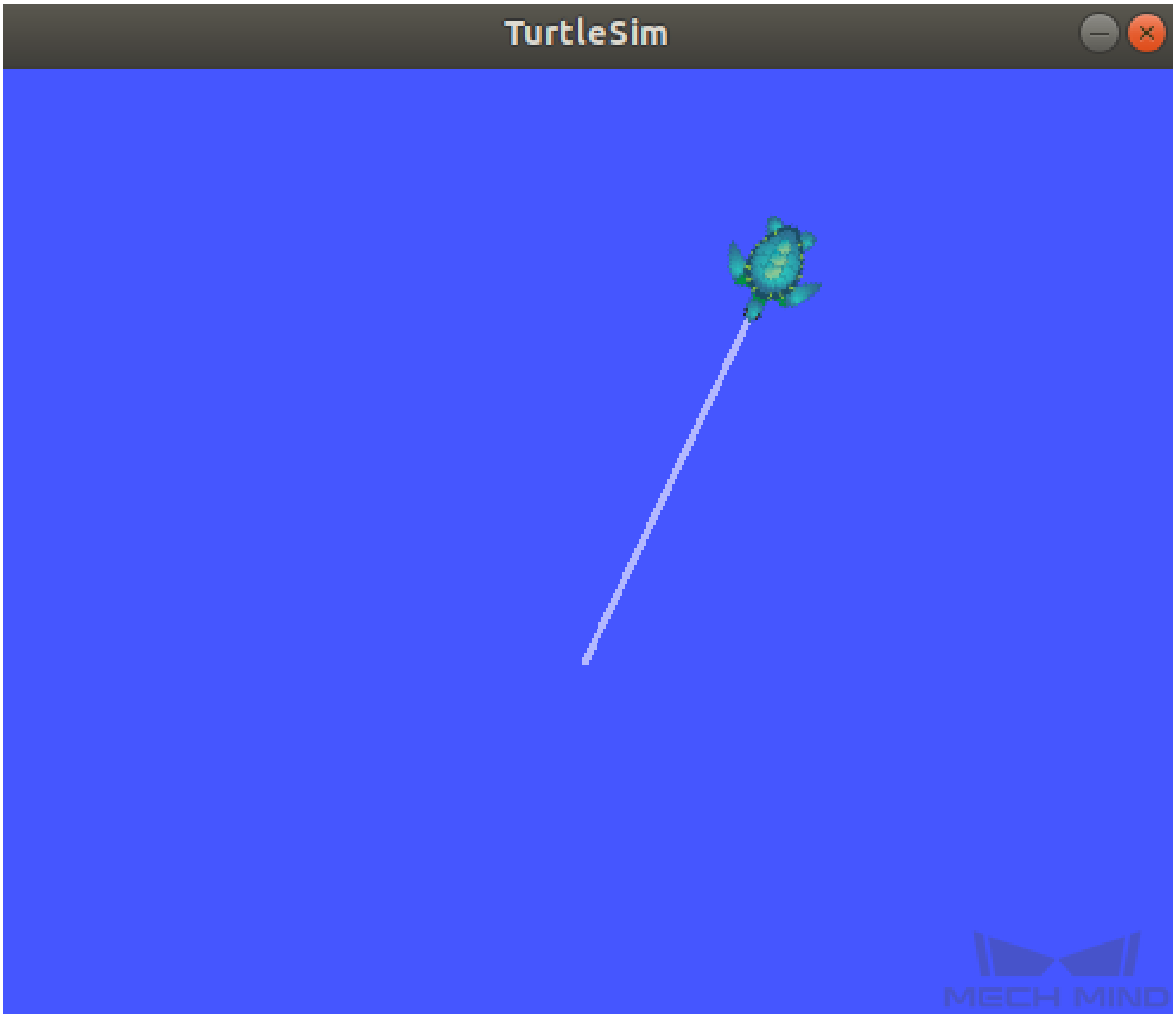
Note:

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:
rosrun turtlesim turtlesim_node
# Open a third terminal and enter the command:
rosrun 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
========

```

(continues on next page)

(continued from previous page)

```
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
    * /rosdistro: 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

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.

CHAPTER

TWELVE

MECH-EYE API: CAMERA FIRMWARE UPGRADE TOOL

With this tool, you can upgrade the camera firmware on Ubuntu directly, simplifying the processing of camera firmware upgrade.

The camera firmware upgrade tool consists of two parts: the executable file **firmware_update_tool** and the firmware upgrade package **mmind_eye_vx.x.x.zip**.

Please follow these steps to upgrade the camera firmware:

1. *Download and install the latest version of Mech-Eye SDK*.
2. Execute the following command to give execute permission to **firmware_update_tool**.

```
chmod 777 /opt/mech-mind/mech-eye-sdk/bin
```

3. Execute the following command to upgrade the firmware. Make sure to replace **x.x.x** with the actual installed version of Mech-Eye SDK.

```
sudo /opt/mech-mind/mech-eye-sdk/bin/firmware_update_tool /opt/mech-mind/mech-eye-sdk/
↪resource/mmind_eye_vx.x.x.zip
```

Hint:

- The two paths must be separated by a space. If not separated, the paths cannot be recognized, and the camera firmware upgrade tool cannot be executed.
- Write the path of **firmware_update_tool** first and the path of **mmind_eye_vx.x.x.zip** second. If the order is incorrect, the camera firmware upgrade tool cannot be executed.

4. Input the device index of the camera whose firmware needs to be upgraded according to the instruction, and press `Enter` to connect. If the following content is displayed, the firmware has been successfully upgraded.

```
Firmware version: VX.X.X install completed.
Upgrade Completed
The camera will restart after this upgrade.
Sending file update_reboot.sh ...
Execute shell: sh /home/ubuntu/Downloads/update_reboot.sh.
```

Note: If the firmware upgrade failed, please retry or contact Technical Support.

CHAPTER

THIRTEEN

MECH-EYE API REFERENCE

Please refer to Mech-Eye API Reference for detailed information.

CHAPTER

FOURTEEN

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**:

- *HALCON: Connect to Camera, Perform Image Capturing, and Adjust Parameters*
- *HALCON: Hand-Eye Calibration*
- *HALCON: Set IP Address*
- *HALCON: Obtain Depth Values Only*
- *HALCON: Obtain Textured Point Cloud*
- *Obtain HALCON-Readable Point Cloud through Mech-Eye API*

Useful reference information for using GenICam:

Camera Parameters Supported for GenICam

Use Mech-Eye Viewer to Adjust Camera Parameters for GenICam Client

CHAPTER

FIFTEEN

GENICAM BASICS

This topic covers the following contents:

15.1 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.

15.2 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 .

15.3 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* .

CHAPTER

SIXTEEN

HALCON: CONNECT TO CAMERA, PERFORM IMAGE CAPTURING, AND ADJUST PARAMETERS

This topic provides instructions on connecting to a Mech-Eye Industrial 3D Camera, perform image capturing, and adjust camera parameters of the camera with HALCON and introduction on the HALCON samples.

16.1 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.

  Hint: HALCON versions below 20.11 are not fully tested.

- 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 *Set Camera IP Address and Connect to Camera*.

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*

16.2 Control the Camera with Image Acquisition Assistant

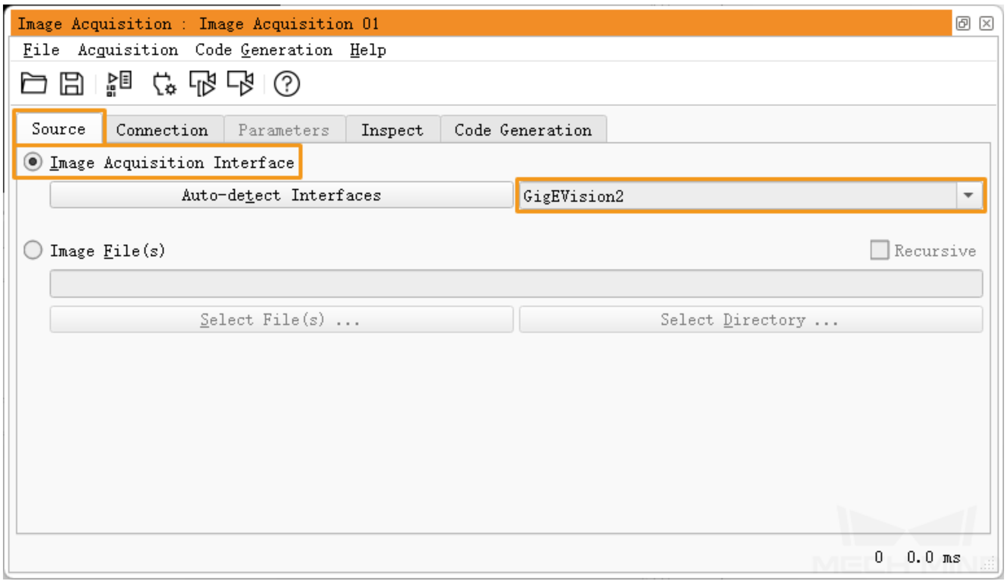
Using the image acquisition assistant provided by HDevelop, you can quickly connect to the camera, perform image capturing and adjust camera parameters.

To open the image acquisition assitant: open HDevelop, and in the menu bar, select *Assistants* → *Open New Image Acquisition*.

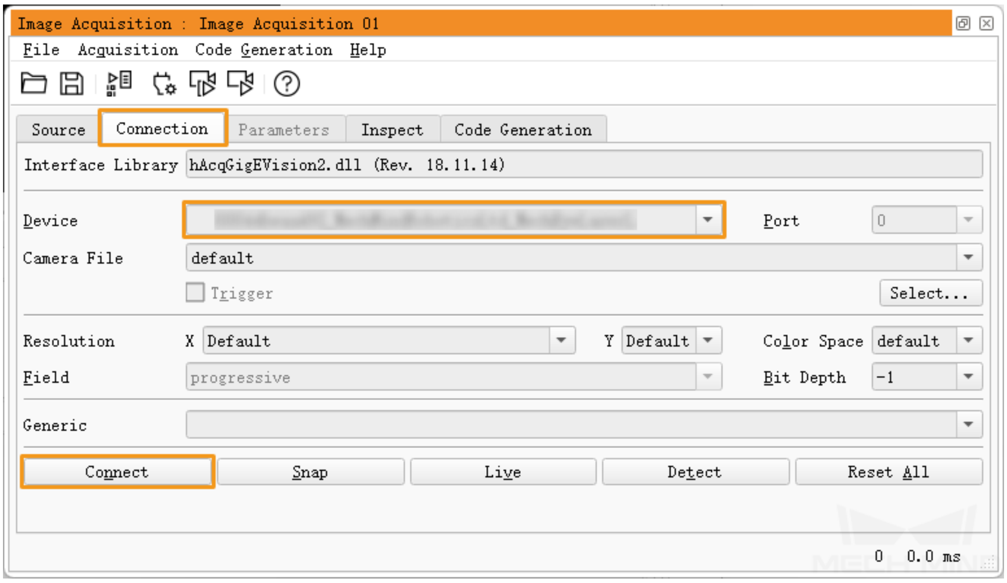
16.2.1 Connect to the Camera

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.



Interface Library hAcqGigEVision2.dll (Rev. 18.11.14)

Device [redacted]

Port 0

Camera File default

□ Trigger

Resolution X Default Y Default

Color Space default

Field progressive

Bit Depth -1

Generic

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.

16.2.2 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.

16.2.3 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.

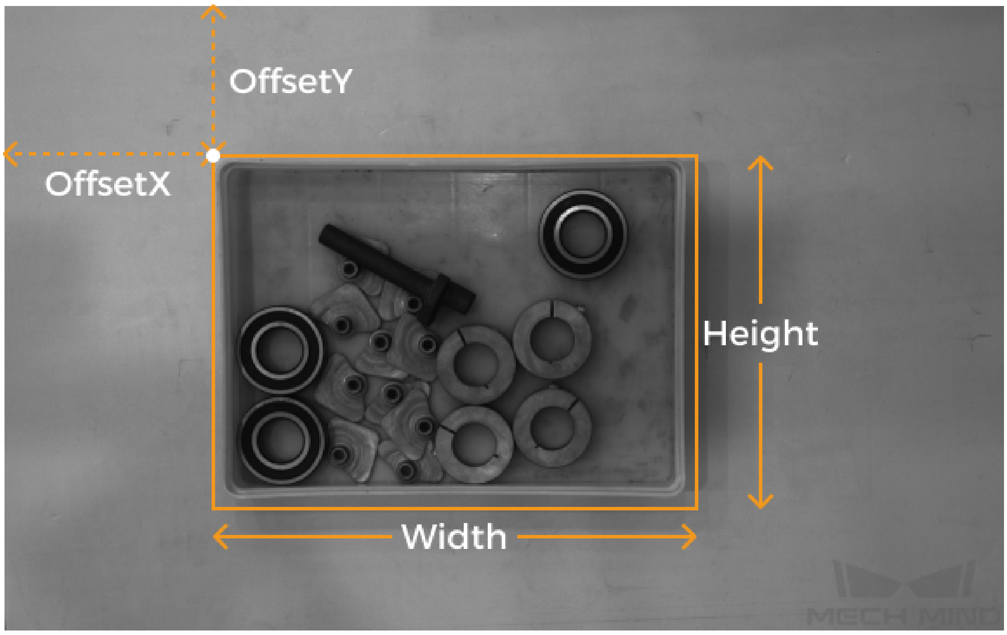
Hint: If **Update Image** in the upper right of the **Parameters** tab is checked, the image in the **Canvas** winodw is automatically updated as you adjust the parameters.

16.2.4 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** winodw 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 *visualized tool in Mech-Eye Viewer*

16.2.5 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.

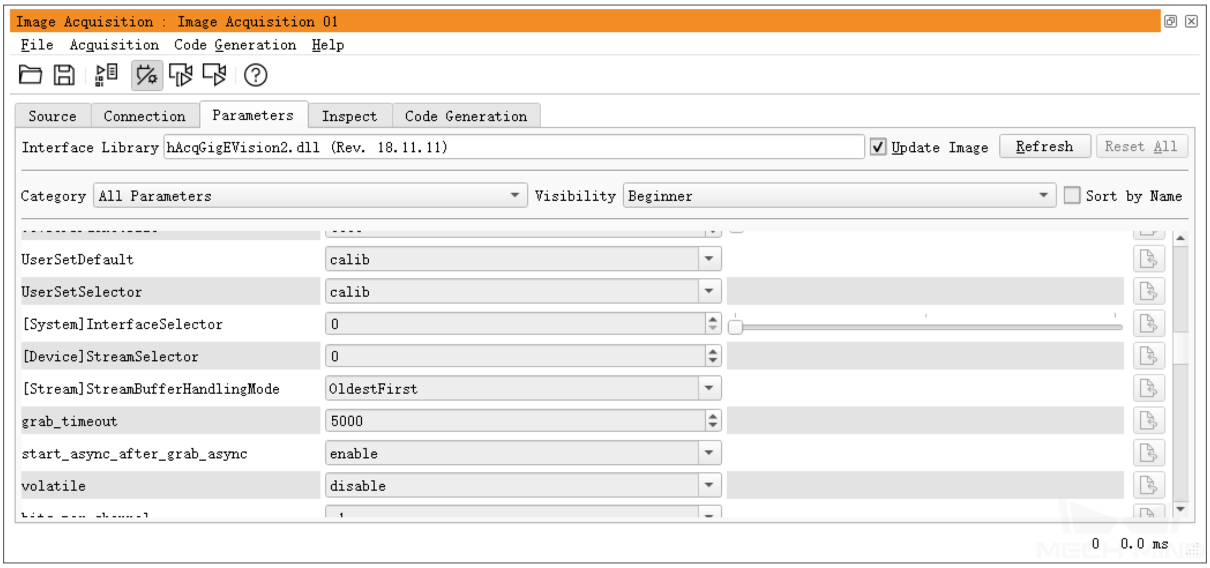


Image Acquisition : Image Acquisition 01

File Acquisition Code Generation Help

Interface Library hAcqGigEVision2.dll (Rev. 18.11.11) ☑ Update Image Refresh Reset All

Category All Parameters Visibility Beginner □ Sort by Name

UserSetDefault	calib
UserSetSelector	calib
[System]InterfaceSelector	0
[Device]StreamSelector	0
[Stream]StreamBufferHandlingMode	OldestFirst
grab_timeout	5000
start_async_after_grab_async	enable
volatile	disable

0 0.0 ms

2. 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.

3. Find the parameter that you want to adjust and change its value.
4. Find the **UserSetSave** parameter, and click *Apply* to the right to save the configuration.
5. Switch to the **Code Generation** tab, click *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.

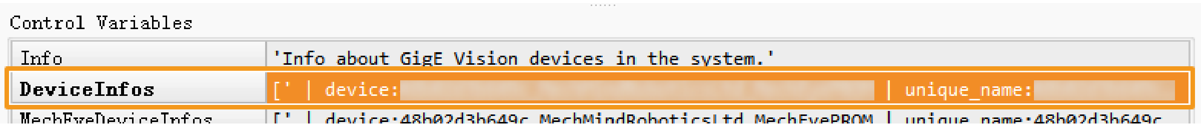
16.3 Control the Camera with Sample Program

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.

16.3.1 Run the HALCON Sample Program

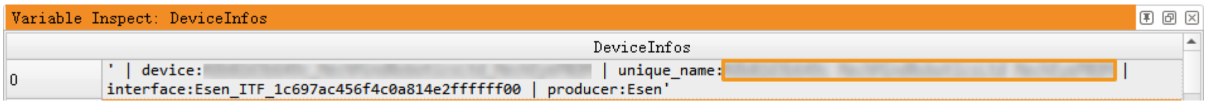
1. Obtain the HALCON sample program.
2. Open the sample program in HDevelop: Open HDevelop, and drag the sample program into HDevelop.
3. Select the line containing the **info_framegrabber** operator in **Program Windows**, and click ▷ in the toolbar or press the `F6` key to run this line.
4. In the **Control Variables** area, double-click **DeviceInfos** to display a list of all the available cameras.



Control Variables

Info	'Info about GigE Vision devices in the system.'
DeviceInfos	[' | device:[redacted] | unique_name:[redacted]
MechEyeDeviceInfos	[' | device:48b02d3b649c_MechMindRoboticsLtd_MechEyePROM | unique_name:48b02d3b649c

5. In the list, double-click the camera that you want to connect, and copy the camera name after **unique_name:** or **user_name:**.



Variable Inspect: DeviceInfos

	DeviceInfos
0	' | device:[redacted] | unique_name:[redacted] | interface:Esen_ITF_1c697ac456f4c0a814e2ffffff00 | producer:Esen'

6. Locate the following line, and replace **MechEye** with the copied **unique_name** or **user_name**.

```
DeviceInfo := 'MechEye'
```

7. Run the sample program by clicking ▷ or pressing the `F5` key.
8. 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)
```

16.3.2 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')
```

16.3.3 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.

CHAPTER

SEVENTEEN

HALCON: HAND-EYE CALIBRATION

This topic introduces how to perform hand-eye calibration in Windows using the HALCON sample programs provided by Mech-Mind. Mech-Mind provide the following two sample programs:

- **determine_calibration_poses**: used to determine the CALIBRATION POSES (During the hand-eye calibration, the camera needs to obtain images of the calibration board at a series of robot poses. These robot poses are the calibration poses.) used during the hand-eye calibration.
- **perform_hand_eye_calibration**: used to perform the hand-eye calibration.

Note:

- Currently, the sample programs only support hand-eye calibration of 6-axis robots.
- For UHP series, to perform the hand-eye calibration, the **Capture Mode** parameter must be set to **Camera1**.

17.1 Prerequisites

Before performing the hand-eye calibration, please make sure that the following requirements are satisfied.

1. The accuracy of the robot is good enough, and the robot can operate properly.
2. Prepare the calibration board that came with the camera, and mount the camera and calibration board.
3. Download and install *Mech-Eye SDK*.

4. Make sure that the quality of the 2D image and depth map meet *the requirements*.
5. Make sure that the *camera intrinsic parameters* are accurate.
6. Download the HALCON sample programs: select *Code* → *Download as ZIP*.

17.2 Edit the Sample Programs

After determining the calibration poses with the **determine_calibration_poses** sample program, you will use the **perform_hand_eye_calibration** sample program to perform the hand-eye calibration.

The following information must be identical in the two sample programs:

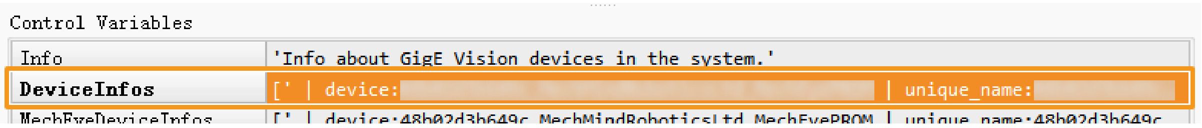
- The camera to be connected
- Calibration board model

Besides, the Euler angle convention in the **robot_pose** JSON file should be set before determining the calibration poses.

17.2.1 Select the Camera

Before running the sample programs, you need to set the camera to be connected in the sample program to make sure the two programs connect to the same camera. Please follow these steps.

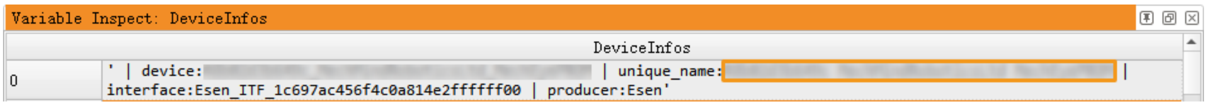
1. Open the sample program in HDevelop: Open HDevelop, and drag the sample program into HDevelop.
2. Select the line containing the **info_framegrabber** operator in **Program Windows**, and click [▷ in the toolbar or press the `F6` key to run this line.
3. In the **Control Variables** area, double-click **DeviceInfos** to display a list of all the available cameras.



Control Variables

Info	'Info about GigE Vision devices in the system.'
DeviceInfos	[' | device:[redacted] | unique_name:[redacted]
MechEyeDeviceInfos	[' | device:48b02d3b649c MechMindRoboticsLtd MechEyePROM | unique_name:48b02d3b649c

4. In the list, double-click the camera that you want to connect, and copy the camera name after **unique_name:** or **user_name:**.



Variable Inspect: DeviceInfos

	DeviceInfos
0	' | device:[redacted] | unique_name:[redacted] | interface:Esen_ITF_1c697ac456f4c0a814e2ffffff00 | producer:Esen'

5. Locate the following line, and replace **MechEye** with the copied **unique_name** or **user_name**.

```
DeviceInfo := 'MechEye'
```

6. Run the sample program by clicking ▷ or pressing the `F5` key.

17.2.2 Set Calibration Board Model

Please follow these steps to set the calibration board model.

1. Open the sample program in HDevelop: Open HDevelop, and drag the sample program into HDevelop.
2. Set the calibration board model: The default model is BDB-5. If you are using another model, locate the following operator, and change **BDB-5** to the *calibration board model in use*.

```
set_framegrabber_param (AcqHandle, 'BoardType', 'BDB-5')
```

17.2.3 Set Euler Angle Convention and Unit

The calibration poses determined using the **determine_calibration_poses** sample program need to be entered into the **robot_pose** JSON file. The default Euler angle convention in this file is **sxyz**, and the default Euler angle unit is degree. Please follow these steps to set the Euler angle convention and unit.

1. Open the **robot_pose** JSON file.
2. Set the Euler angle convention: locate the following line, and replace **sxyz** with the Euler angle convention of the robot in use. For the Euler angle conventions already supported in the sample program, please refer to *Supported Euler Angle Conventions*.

```
"EulerType":"sxyz"
```

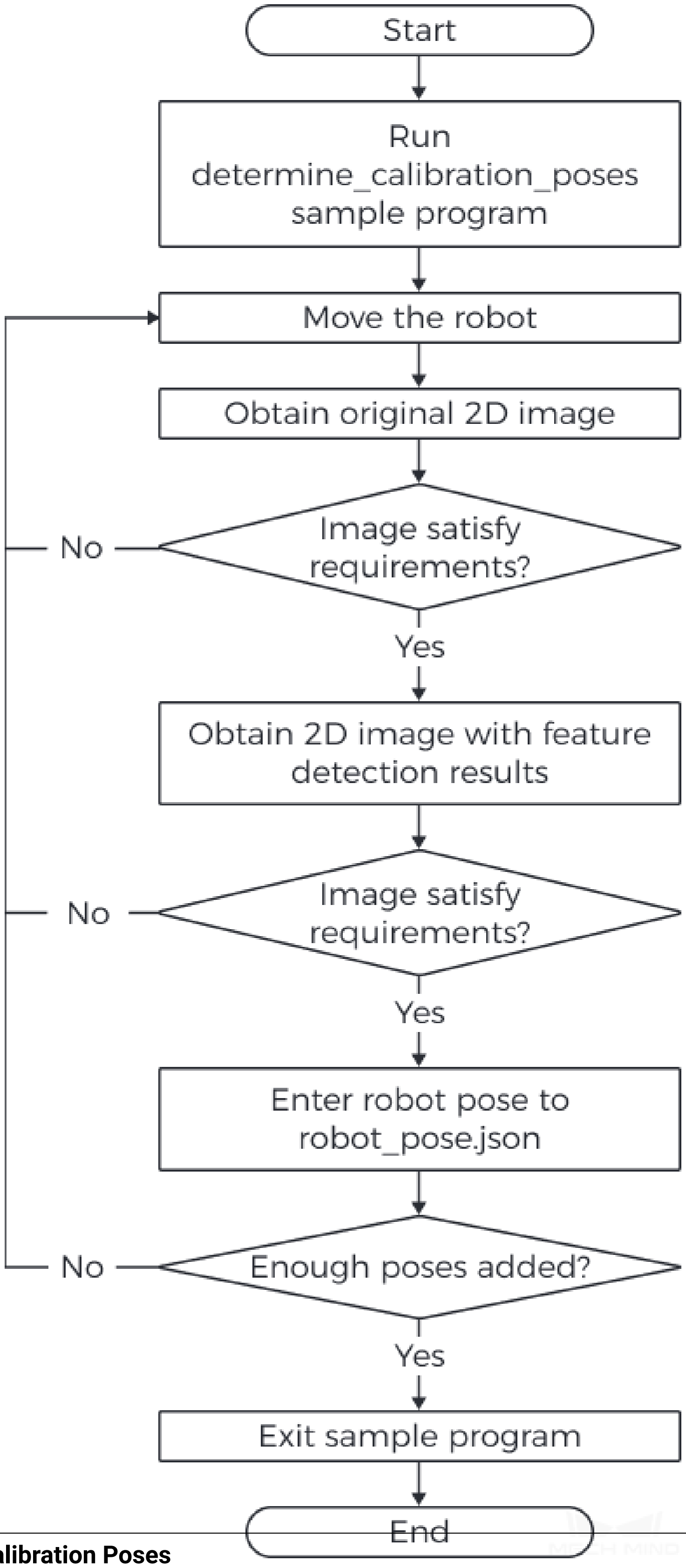
3. Set the Euler angle unit: if you need to enter Euler angles in radiants, locate the following line, and replace **true** with **false**.

```
"FromDegree":true
```

4. Save the **robot_pose** JSON file.

17.3 Determine Calibration Poses

17.3.1 Workflow


Start
Run determine_calibration_poses sample program
Move the robot
Obtain original 2D image
Image satisfy requirements?
No
Yes
Obtain 2D image with feature detection results
Image satisfy requirements?
No
Yes
Enter robot pose to robot_pose.json
Enough poses added?
No
Yes
Exit sample program
End

17.3.2 Instructions

Before performing the hand-eye calibration, you need to determine at least 15 calibration poses. Please follow these steps to determine the calibration poses.

1. Run the sample program by clicking ▷ or pressing the `F5` key. When the program runs to the line **read_char (WindowHandle, Char, ReCode)**, the program will pause and wait for command input in the following steps.

   Hint: If the camera cannot be connected, please check if it is already connected by Mech-Eye Viewer or another GenICam client.

2. Move the robot to a candidate calibration pose with the teach pendant.

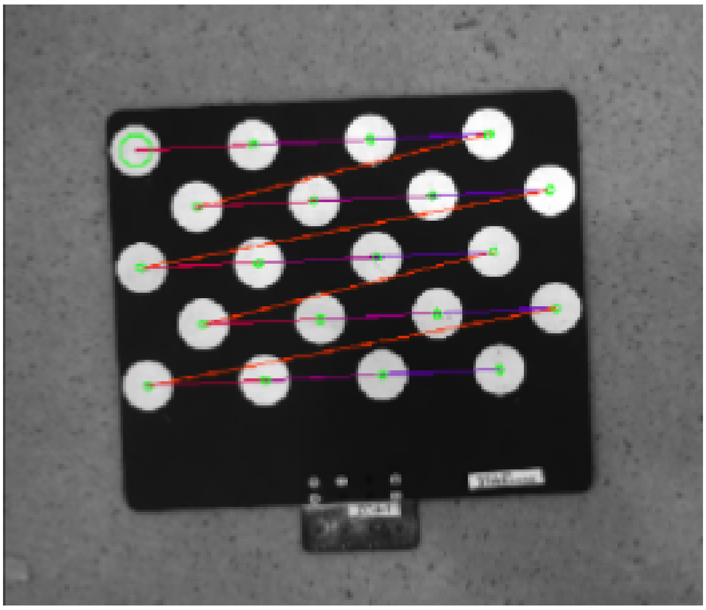
   Hint: For requirements on the calibration poses, you can refer to the **Calibration** chapter in **Help** > **HALCON Reference**.

3. Press `P` to obtain the 2D image from the camera.

   - If the calibration board is not entirely captured, please move the robot and obtain the 2D image again.
   - If the entire calibration board is captured, proceed to the next step.

4. Press `T` to obtain the 2D image containing the feature detection results.

   - If the camera did not detect the circles on the calibration board, no image will be displayed in HDevelop. In this case, please move the robot and obtain the original 2D image and the 2D image containing the feature detection results again.
   - If the camera can detect the circles on the calibration board, a 2D image containing the feature detection results (as shown below) will be displayed in HDevelop. Please proceed to the next step.



5. Check the robot teach pendant, and enter the current robot pose to the **robot_pose** JSON file.

> **Hint:** It is recommended to save the pose on the teach pendant. When performing the hand-eye calibration, you can use the saved poses to move the robot.

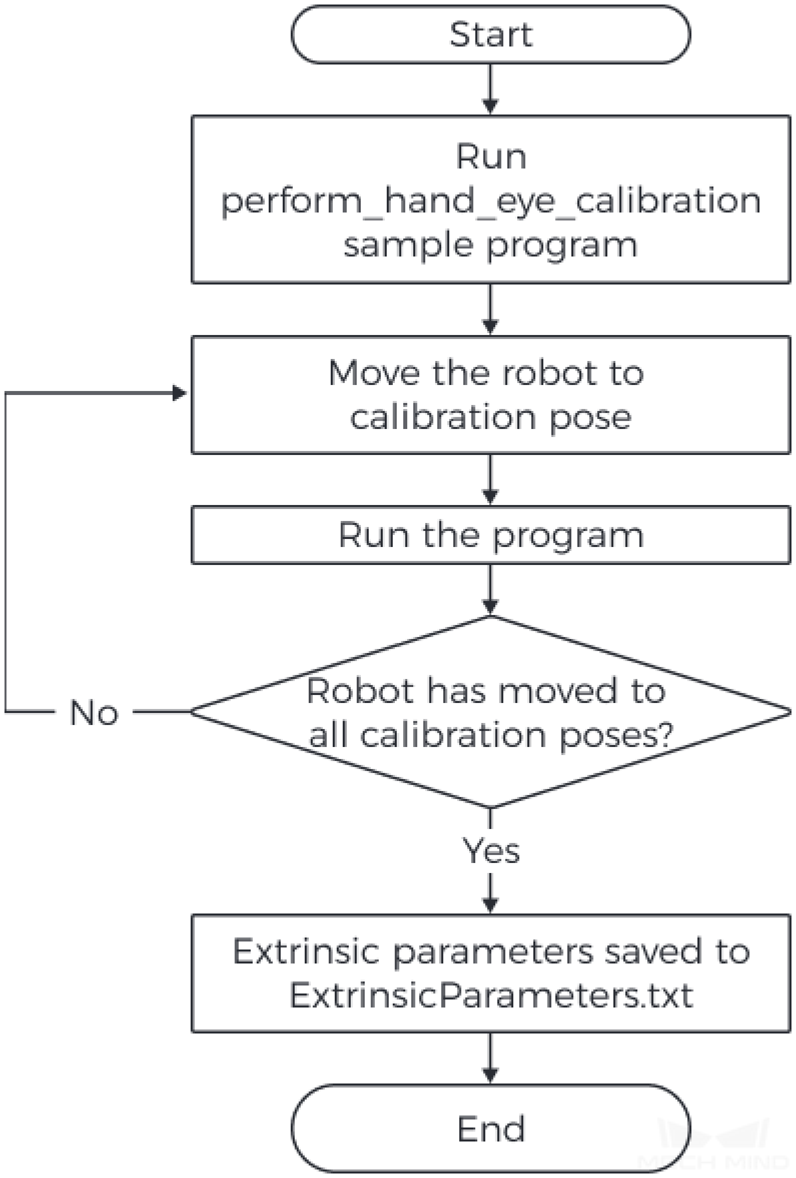
6. Repeat steps 2 to 5 to determine more calibration poses.
7. After determining at least 15 calibration poses, press `Q` to exit the program.
8. Set the number of calibration poses in the **robot_pose** JSON file: Open the **robot_pose** JSON file, and locate the following line. Replace **15** with the actual number of calibration poses determined.

```
"pose_count":15
```

17.4 Perform Hand-Eye Calibration

After determining the robot calibration poses, you can perform the hand-eye calibration by running the **perform_hand_eye_calibration** sample program.

17.4.1 Workflow


Start
Run perform_hand_eye_calibration sample program
Move the robot to calibration pose
Run the program
Robot has moved to all calibration poses?
No
Yes
Extrinsic parameters saved to ExtrinsicParameters.txt
End

17.4.2 Set Camera Mounting Method

Before performing the hand-eye calibration, please set the camera mounting method in the sample program.

The default mounting method is Eye in Hand. If your camera is mounted in the Eye to Hand mode, please locate the following operator, and replace EyeInHand with EyeToHand.

```
set_framegrabber_param (AcqHandle, 'CalibrationType', 'EyeInHand')
```

17.4.3 Switch Reference Frame

The sample program contains an operator that can switch the reference frame in which the point cloud is output from the camera.

The default setting is not to switch the reference frame. If you would like to switch to the robot reference frame, please locate the following operator in the **captureTranformedPointCloud** procedure, and replace **false** with **true**.

```
set_framegrabber_param (AcqHandle, 'Scan3dCoordinateTransformEnable',false)
```

17.4.4 Instructions

Please follow these steps to perform the hand-eye calibration.

1. Run the sample program by clicking ▷ or pressing the `F5` key. The sample program will stop when it runs to the **stop** operator.
2. Move the robot to the calibration pose in the **robot_pose** JSON file.

   Hint: Please move the robot to the calibration poses in the exact order as they are written in the JSON file. Otherwise, the subsequent calculation of the extrinsic parameters will fail.

3. Run the sample program by clicking ▷ or pressing the `F5` key. The camera will perform image capturing.
4. In the **Control Variables** area, check the value of **CollectResult**.
   - If **SUCCESS** is displayed, please proceed to the next step.
   - If an error is displayed, please troubleshoot according to the *displayed error code*, and then determine the calibration poses again.
5. A message saying "Move the robot to the next calibration pose" will be displayed on the image. Please repeat steps 2 to 3.

   Hint: After the robot has moved to all the calibration poses in the **robot_pose** JSON file, when you run the sample program again, the extrinsic parameters will be calculated.

6. In the **Control Variables** area, check the value of **CalibResult**.
   - If **SUCCESS** is displayed, then the hand-eye calibration has succeeded. In the folder where the sample program is stored, you can find the **ExtrinsicParameters** TXT file that contains the extrinsic parameters and the a point cloud file whose coordinates have been transformed.
   - If an error is displayed, please troubleshoot according to the *displayed error code*, and then determine the calibration poses again.

17.5 Supported Euler Angle Conventions

The conversion from the following Euler angle conventions to quaternions is already supported in the sample program.

Angle input order	Euler angle convention	Robot brand
Z-Y'-Z"	rzyz	Kawasaki
Z-Y'-X"	rzyx	ABB
		KUKA
X-Y-Z	sxyz	FANUC
		YASKAWA
		ROKAE
		UR
X-Y'-Z"	rxyz	/
Z-X'-Z"	rzxz	/

Note:

- Even using the same Euler angle convention, different robot brands might display the Euler angles in different orders. Please input the Euler angles in the order specified in the above table.
- If the Euler angle convention used by your robot is not listed in the above table, you need to add the conversion from this Euler angle convention to quaternions. Please refer to the code of the **euler_to_quad** procedure in the **perform_hand_eye_calibration** sample program and add your own conversion.

17.6 Calibration Parameters

This section introduces the parameters used during hand-eye calibration.

- *BoardType*
- *ExtrinErrCode*

17.6.1 BoardType

This parameter is used to set the model of the calibration board in use.

Value list and descriptions:

Value	Description
BDB-5	The recommended distance from the calibration board to the camera is < 0.6 m
BDB-6	The recommended distance from the calibration board to the camera is 0.6–1.5 m
BDB-7	The recommended distance from the calibration board to the camera is > 1.5 m
OCB-005	Only used for Eye-to-Hand projects with high accuracy requirements
OCB-010	
OCB-015	
OCB-020	
CGB-020	The recommended distance from the calibration board to the camera is < 0.6 m
CGB-035	The recommended distance from the calibration board to the camera is 0.6–1.5 m
CGB-050	The recommended distance from the calibration board to the camera is > 1.5 m

17.6.2 ExtrinErrCode

This read-only parameter is used to check the status codes returned during the hand-eye calibration process.

Status code	Description
SUCCESS	Successfully executed.
POSE_INVALID	The pose format is incorrect. Please input quaternions.
IMAGE2D_EMPTY	The 2D image is invalid.
FIND_CORNERS_FAIL	Feature detection of the 2D image failed. Please adjust the *parameters for the 2D image*.
DEPTH_EMPTY	The depth map is invalid.
CORNERS_3D_INVALID	Feature detection of the depth map failed. Please adjust the *parameters for the depth map*.
POSES_INSUFFICIENT	Not enough poses are input. Please input at least 15 calibration poses.

CHAPTER

EIGHTEEN

HALCON: OBTAIN TEXTURED POINT CLOUD

This topic introduces how to obtain the textured point cloud in Windows using the HALCON sample programs provided by Mech-Mind.

18.1 Data Types

This sample program obtains the data in the Range and Intensity channels. The data in these channels are described below.

- Range: a 2D image containing the X, Y and Z values of points. This channel cannot be disabled.
- Intensity: The color or monochrome 2D image used to texture the point cloud.

The pixels of the images in the two channels are perfectly aligned and can be used to generate textured point cloud directly.

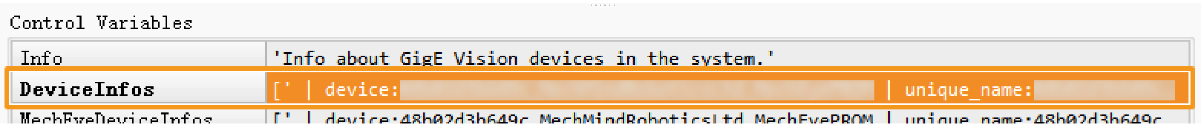
Attention: When you set capture regions for the images in the two channels (the **Width**, **Height**, **OffsetX** and **OffsetY** parameters), if the parameter values are not identical, the pixels of the two images will not be aligned, and the images cannot be used to generate textured point cloud directly.

18.2 Obtain Textured Point Cloud

Please follow these steps to obtain the textured point cloud.

1. Download the obtain_textured_point_cloud sample program <https://github.com/MechMindRobotics/mecheye_halcon_samples>`_: select *Code* → *Download as ZIP*.
2. Open the sample program in HDevelop: Open HDevelop, and drag the sample program into HDevelop.

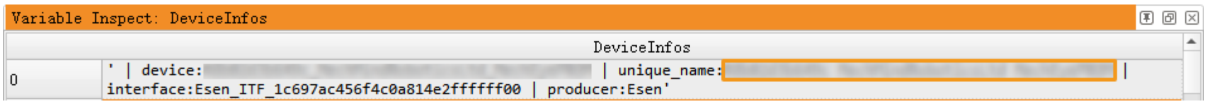
3. Select the line containing the **info_framegrabber** operator in **Program Windows**, and click in the toolbar or press the `F6` key to run this line.
4. In the **Control Variables** area, double-click **DeviceInfos** to display a list of all the available cameras.



Control Variables

Info	'Info about GigE Vision devices in the system.'
DeviceInfos	[' | device:[redacted] | unique_name:[redacted]
MechEyeDeviceInfos	[' | device:48b02d3b649c MechMindRoboticsLtd MechEyePROM | unique_name:48b02d3b649c

5. In the list, double-click the camera that you want to connect, and copy the camera name after **unique_name:** or **user_name:**.



Variable Inspect: DeviceInfos

	DeviceInfos
0	' | device:[redacted] | unique_name:[redacted] | interface:Esen_ITF_1c697ac456f4c0a814e2ffffff00 | producer:Esen'

6. Locate the following line, and replace **MechEye** with the copied **unique_name** or **user_name**.

```
DeviceInfo := 'MechEye'
```

7. Run the sample program by clicking or pressing the `F5` key. The obtained textured point cloud will be displayed in HDevelop and saved to the **ObjectModel3D** variable.

Note: For LSR (V4) and DEEP (V4) series, the color 2D image is obtained for texturing the point cloud by default. If you want to use the monochrome 2D image instead, please locate the following line and activate (uncomment) it.

```
set_framegrabber_param (AcqHandle, 'SourceSelector', 'Monochrome')
```

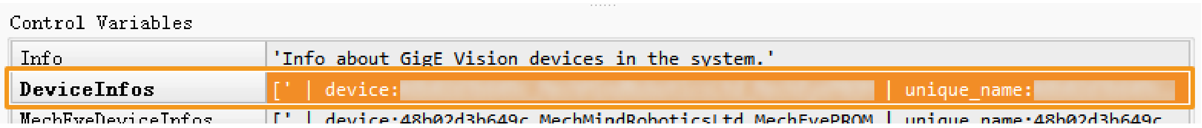
HALCON: OBTAIN DEPTH VALUES ONLY

This topic introduces how to obtain the depth map that contains only the object depth information in Windows using the HALCON sample programs provided by Mech-Mind. Obtaining only the depth information can speed up the data transmission and improve efficiency.

19.1 Obtain Depth Map

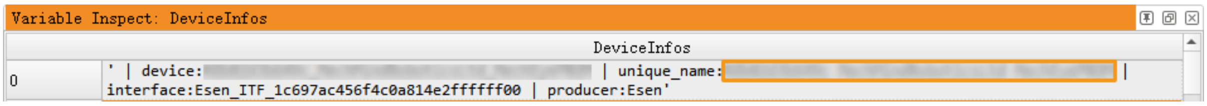
Please follow these steps to obtain the depth map.

1. Download the obtain_depth_map sample program: select *Code* → *Download as ZIP*.
2. Open the sample program in HDevelop: Open HDevelop, and drag the sample program into HDevelop.
3. Select the line containing the **info_framegrabber** operator in **Program Windows**, and click in the toolbar or press the `F6` key to run this line.
4. In the **Control Variables** area, double-click **DeviceInfos** to display a list of all the available cameras.



```
Control Variables
Info                'Info about GigE Vision devices in the system.'
DeviceInfos         [' | device:[redacted] | unique_name:[redacted]
MechEyeDeviceInfos  [' | device:48b02d3b649c MechMindRoboticsLtd MechEyePROM | unique_name:48b02d3b649c
```

5. In the list, double-click the camera that you want to connect, and copy the camera name after **unique_name:** or **user_name:**.



```
Variable Inspect: DeviceInfos
                         DeviceInfos
0   ' | device:[redacted] | unique_name:[redacted] |
    interface:Esen_ITF_1c697ac456f4c0a814e2ffffff00 | producer:Esen'
```

6. Locate the following line, and replace **MechEye** with the copied **unique_name** or **user_name**.

```
DeviceInfo := 'MechEye'
```

7. Run the sample program by clicking ▷ or pressing the `F5` key. The obtained depth map is stored in the **DepthInM** variable.

CHAPTER

TWENTY

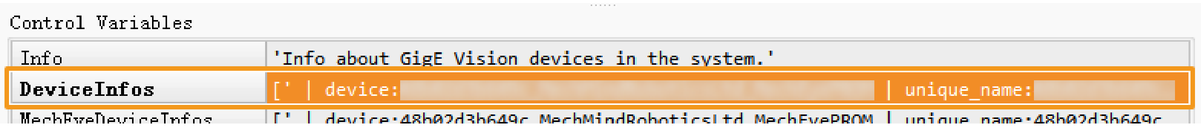
HALCON: SET IP ADDRESS

This topic introduces how to obtain and modify the camera's current IP address, subnet mask, and gateway in Windows using the HALCON sample programs provided by Mech-Mind.

20.1 Obtain Sample Program and Select Camera

Before obtaining or modifying the IP address, you need to download the sample program and set the camera to be connected in the program.

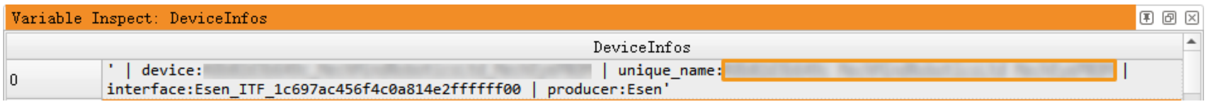
1. Download the configure_camera_ip_address sample program: select *Code* → *Download as ZIP*.
2. Open the sample program in HDevelop: Open HDevelop, and drag the sample program into HDevelop.
3. Select the line containing the **info_framegrabber** operator in **Program Windows**, and click ▷ in the toolbar or press the `F6` key to run this line.
4. In the **Control Variables** area, double-click **DeviceInfos** to display a list of all the available cameras.



Control Variables

Info	'Info about GigE Vision devices in the system.'
DeviceInfos	[' | device:[redacted] | unique_name:[redacted]
MechEyeDeviceInfos	[' | device:48b02d3b649c MechMindRoboticsLtd MechEyePROM | unique_name:48b02d3b649c

5. In the list, double-click the camera that you want to connect, and copy the camera name after **unique_name:** or **user_name:**.



```
Variable Inspect: DeviceInfos
DeviceInfos
0    ' | device:[redacted] | unique_name:[redacted] |
     interface:Esen_ITF_1c697ac456f4c0a814e2ffffff00 | producer:Esen'
```

6. Locate the following line, and replace **MechEye** with the copied **unique_name** or **user_name**.

```
DeviceInfo := 'MechEye'
```

20.2 Obtain Camera IP Address, Subnet Mask and Gateway

Please follow these steps to obtain the current IP address, subnet mask and gateway of the camera:

Run the sample program by clicking ▷ or pressing the `F5` key. In the **Control Variables** area, check the following variables for the current IP address, subnet mask and gateway of the camera.

- CurrentIPAddress: IP address
- CurrentSubnetMaskString: subnet mask
- CurrentDefaultGatewayString: gateway

Note: If the camera firmware version is 2.0.2 or below, the above information cannot be obtained, and the error message will be recorded in the **Exception** variable.

20.3 Set Static IP address

This sample program sets a static IP address by default. Please follow these steps to modify the IP address, subnet mask and gateway of the camera.

1. Locate the following line, and change the IP address, subnet mask and gateway to the values to be set.

```
IPAddressString := '192.168.1.100'
SubnetMaskString := '255.255.255.0'
DefaultGatewayString := '192.168.1.1'
```

2. Run the sample program by clicking ▷ or pressing the `F5` key.
3. Restart the camera to make the new settings take effect.

20.4 Set Dynamic IP Address

Please follow these steps to dynamically assign an IP address to the camera.

1. Locate the following line, and replace **true** with **false**.

   ```
   set_framegrabber_param (AcqHandle, 'GevCurrentIPConfigurationPersistentIP', true)
   ```

2. Run the sample program by clicking ▷ or pressing the `F5` key.
3. Restart the camera to make the new settings take effect.

Note: When the camera is connected to the IPC directly, the IP address of the camera cannot be dynamically assigned.

CHAPTER

TWENTYONE

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

21.1 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.

21.1.1 Supported Camera Parameters

For descriptions and explanations of camera parameters supported for GenICam, download and read Parameters in GenICam Client.

21.1.2 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 - *Auto-Exposure ROI*
Scan2DROILeft	
Scan2DROITop	
Scan2DROIWidth	
Scan2DExposureMode	2D Parameters - *Exposure Mode* - *2D Image (Texture) Exposure Mode*
Scan2DExpectedGrayValue	2D Parameters - *Gray Value*
Scan2DHDRExposureSequence	2D Parameters - *Exposure Sequence*
Scan2DToneMappingEnable	2D Parameters - *Tone Mapping*
Scan2DExposureTime	2D Parameters - *Exposure Time*
Scan2DSharpenFactor	2D Parameters - Sharpen Factor
Scan2DPatternRoleExposureMode	2D Parameters - *2D Image (Depth Source) Exposure Mode*
Scan2DPatternRoleExposureTime	2D Parameters - *2D Image (Depth Source) Exposure Time*
Scan3DROIHeight	*ROI*
Scan3DROILeft	
Scan3DROITop	
Scan3DROIWidth	
ProjectorLightColor	3D Parameters - Projector —Light Color
ProjectorPowerLevel	3D Parameters - Projector —*Light Brightness*
ProjectorFringeCodingMode	3D Parameters - Projector —*Fringe Coding Mode*
AntiFlickerMode	3D Parameters - Projector —*Anti-Flicker Mode*
ProjectorSelectionMode	3D Parameters - Projector —Projector Selection Mode
ProjectorSelector	3D Parameters - Projector —Projector Selector
LaserFringeCodingMode	3D Parameters - Laser —*Fringe Coding Mode*
LaserPowerLevel	3D Parameters - Laser —*Laser Power*
LaserFrameAmplitude	3D Parameters - Laser Projection Frame Control —Laser Frame Amplitude
LaserFrameOffset	3D Parameters - Laser Projection Frame Control —Laser Frame Offset
LaserFramePartitionCount	3D Parameters - Laser Projection Frame Control —Laser Frame Partition Count
Scan3DBinningEnable	3D Parameters - 3D Binning
Scan3DExposureCount	3D Parameters - *Exposure Multiplier*
Scan3DExposureTime	3D Parameters - *Exposure Time*
Scan3DExposureTime2	3D Parameters - *Exposure Time 2*
Scan3DExposureTime3	3D Parameters - *Exposure Time 3*
Scan3DGain	3D Parameters - *Gain*
FringeCodingMode	3D Parameters - UHP —FringeCodingMode
UhpCaptureMode	3D Parameters - UHP —*Capture Mode*
AcquisitionMode	📷 and 🔄
DepthLowerLimit	*Depth Range*

Table 1 – continued from previous page

DepthUpperLimit	
DeviceScanType	-
PointCloudSurfaceSmoothing	Point Cloud Processing - *Surface Smoothing*
PointCloudOutlierRemoval	Point Cloud Processing - *Outlier Removal*
PointCloudNoiseRemoval	Point Cloud Processing - *Noise Removal*
PointCloudEdgePreservation	Point Cloud Processing - *Edge Preservation*
FringeContrastThreshold	Point Cloud Processing - *Stripe Contrast Threshold*
FringeMinThreshold	Point Cloud Processing - Minimum Fringe Intensity Threshold
UserSetDefault	-
UserSetSelector	Parameter Group drop-down menu

21.2 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.

21.2.1 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*.

21.2.2 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

21.2.3 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 **Prameter 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.

21.3 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.

21.3.1 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 runing 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.

21.3.2 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 runing 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.

CHAPTER

TWENTYTWO

TROUBLESHOOTING

This topic provides troubleshooting guidance for commonly encountered problems.

22.1 Camera Not Found in Mech-Eye SDK

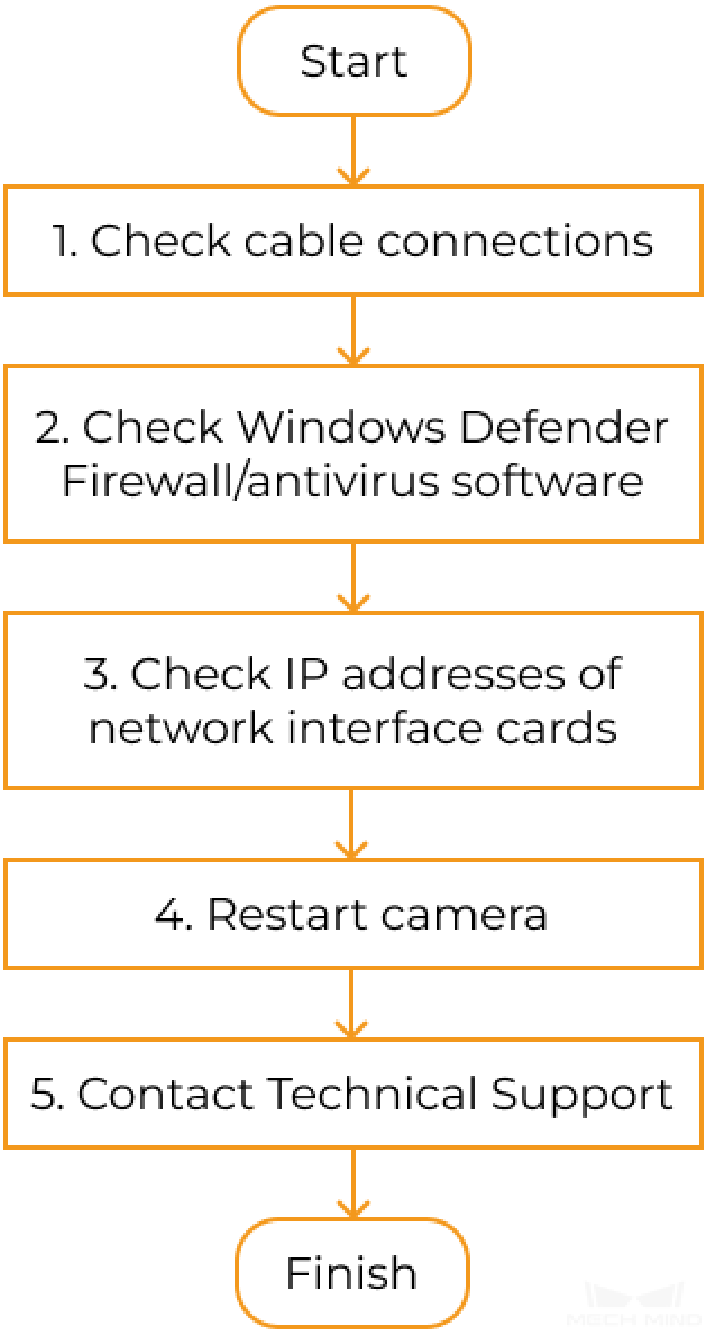
Problem description: In Mech-Eye SDK, the camera that you want to connect does not show up in the list of available cameras.

This topic provides the possible causes, troubleshooting guidelines and detailed troubleshooting procedures for this problem.

22.1.1 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.

22.1.2 Troubleshooting Guidelines


Start
1. Check cable connections
2. Check Windows Defender Firewall/antivirus software
3. Check IP addresses of network interface cards
4. Restart camera
5. Contact Technical Support
Finish

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 antivirous 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.

22.1.3 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 *blinking or solid green*.
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 antivirous 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 untill 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 *Set Camera IP Address and Connect to Camera* 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 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 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.

22.2 Camera Cannot Be Connected in Mech-Eye SDK (Unreachable)

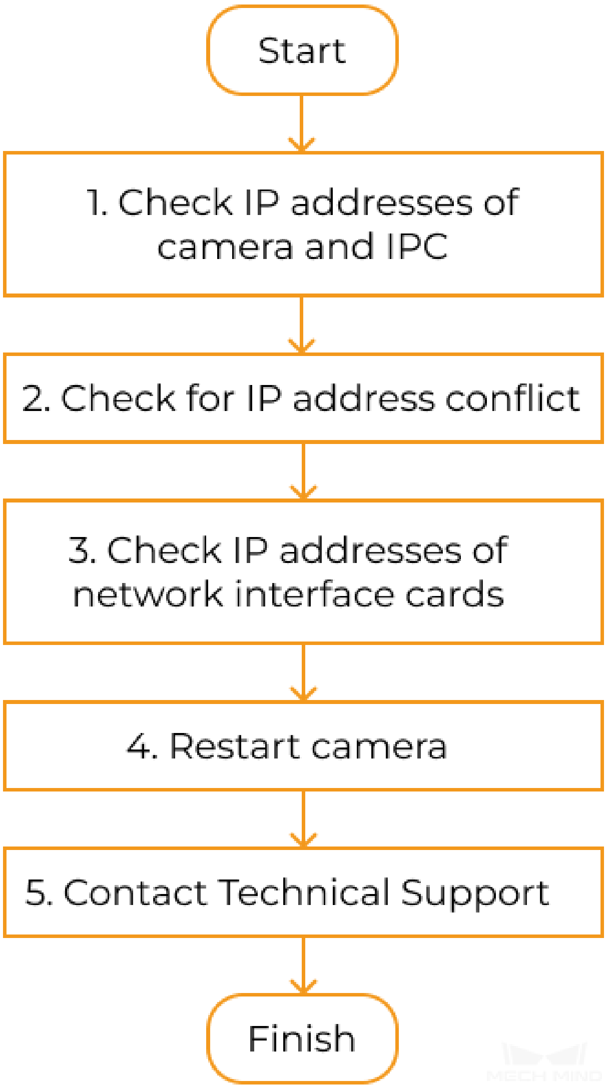
Problem description: The camera shows up in Mech-Eye SDK but cannot be connected. The status of the IP address displayed in the camera info card in Mech-Eye Viewer is **Unreachable**.

This topic provides the possible causes, troubleshooting guidelines and detailed troubleshooting procedures for this problem.

22.2.1 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.

22.2.2 Troubleshooting Guidelines


Start
1. Check IP addresses of camera and IPC
2. Check for IP address conflict
3. Check IP addresses of network interface cards
4. Restart camera
5. Contact Technical Support
Finish

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.

22.2.3 Troubleshooting Procedures

Check IP Addresses of Camera and IPC

1. Check the IP addresses of the camera and IPC. For instructions, please refer to *Set Camera IP Address and Connect to Camera*.
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 untill 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 *Set Camera IP Address and Connect to Camera* 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.

22.3 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.

22.3.1 Possible Cause

This is a bug in Mech-Eye SDK 1.6.1 and below.

22.3.2 Troubleshooting Procedure

If this problem occurs, it is recommended to upgrade Mech-Eye SDK to the latest version. For instrucitons, please refer to *Upgrade Mech-Eye SDK*.

If you cannot upgrade Mech-Eye SDK, please contact Technical Support.

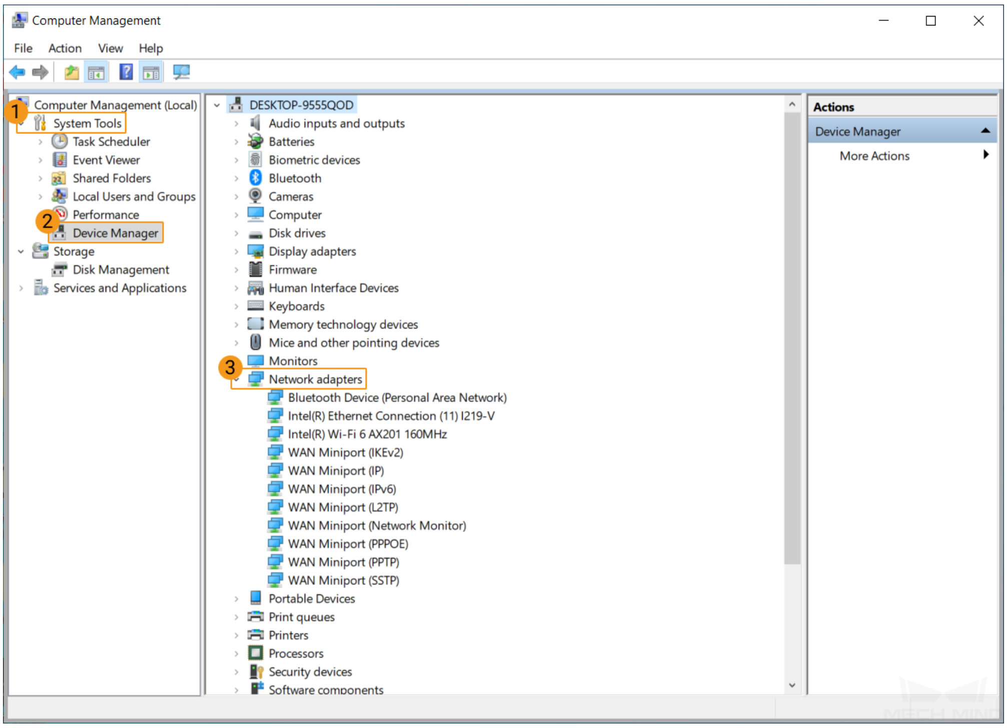
22.4 Enable Jumbo Frames (GenICam)

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.

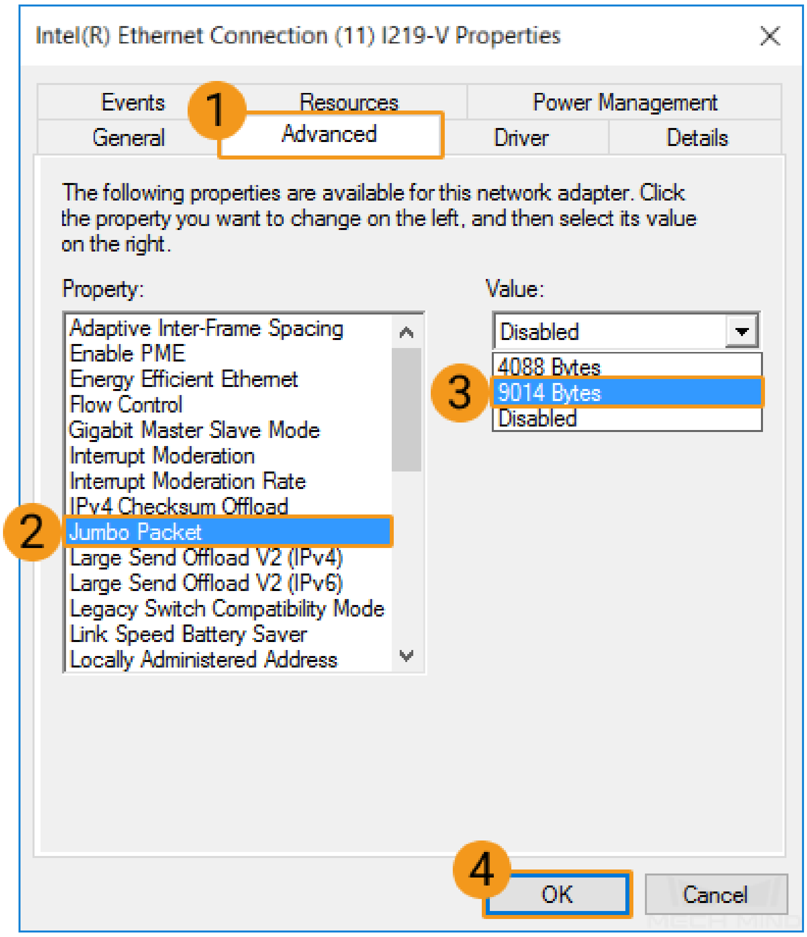
Note: Jumbo frames are Ethernet frames that can carry up to 9000 bytes of data. Standard Ethernet frames carry up to 1500 bytes of data. Using jumbo frames can reduce CPU processing load and speed up data transmission.

22.4.1 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.



The following properties are available for this network adapter. Click the property you want to change on the left, and then select its value on the right.

Property:

- Adaptive Inter-Frame Spacing
- Enable PME
- Energy Efficient Ethernet
- Flow Control
- Gigabit Master Slave Mode
- Interrupt Moderation
- Interrupt Moderation Rate
- IPv4 Checksum Offload
- Jumbo Packet
- Large Send Offload V2 (IPv4)
- Large Send Offload V2 (IPv6)
- Legacy Switch Compatibility Mode
- Link Speed Battery Saver
- Locally Administered Address

Value:

Disabled

- 4088 Bytes
- 9014 Bytes
- Disabled

CHAPTER

TWENTYTHREE

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

Instructions on **setting the IP address of the IPC Ethernet port connected to the camera**:

Set IP Address on IPC

23.1 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:

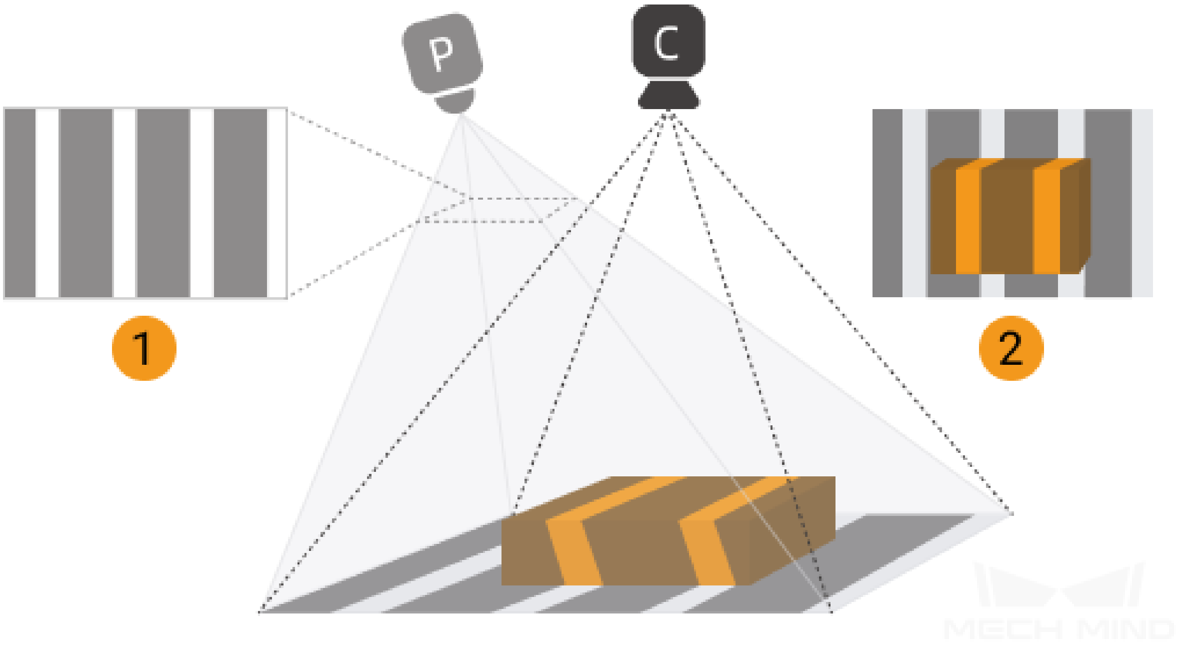
- *Image Capturing*
- *Calculation and Processing*
- *Output*

23.1.1 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.



23.1.2 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.

23.1.3 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!

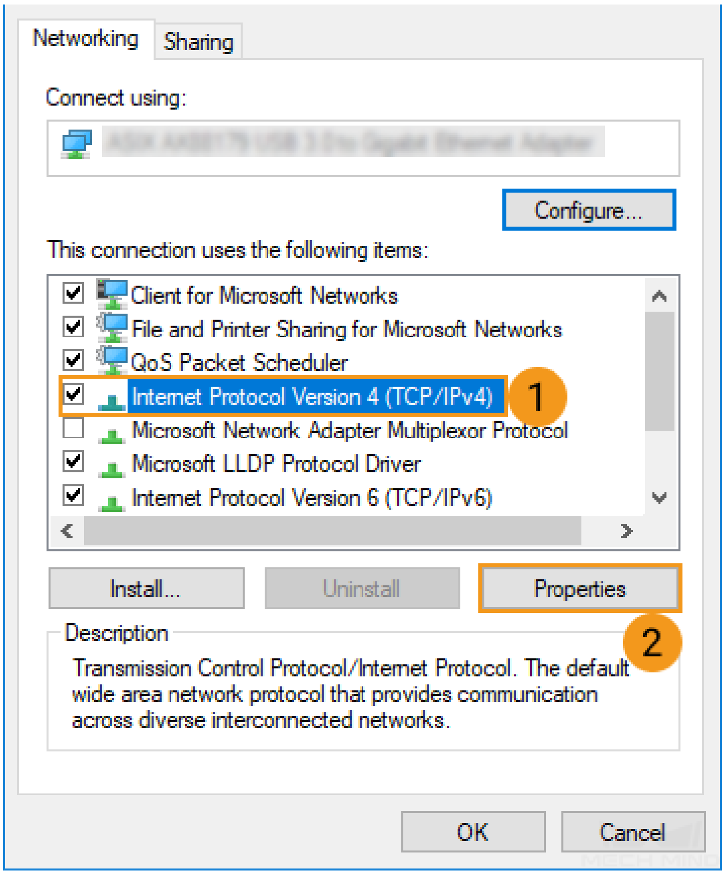
Note:

- 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.

23.2 Set IP Address on IPC

This topic provides instructions on setting the IP address of the IPC Ethernet port connected to the camera.

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**.



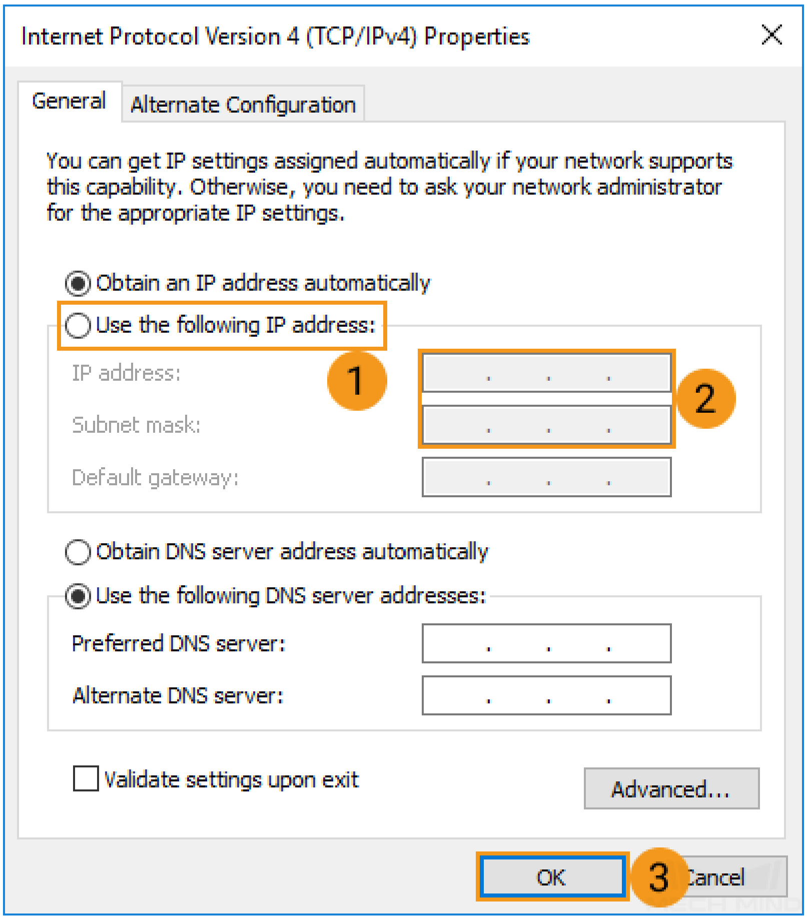
This connection uses the following items:

- ☑ Client for Microsoft Networks
- ☑ File and Printer Sharing for Microsoft Networks
- ☑ QoS Packet Scheduler
- ☑ Internet Protocol Version 4 (TCP/IPv4)
- ☐ Microsoft Network Adapter Multiplexor Protocol
- ☑ Microsoft LLDP Protocol Driver
- ☑ Internet Protocol Version 6 (TCP/IPv6)

Description

Transmission Control Protocol/Internet Protocol. The default wide area network protocol that provides communication across diverse interconnected networks.

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.



Internet Protocol Version 4 (TCP/IPv4) Properties

General | Alternate Configuration

You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

● Obtain an IP address automatically

○ Use the following IP address:

IP address: . . .

Subnet mask: . . .

Default gateway: . . .

○ Obtain DNS server address automatically

● Use the following DNS server addresses:

Preferred DNS server: . . .

Alternate DNS server: . . .

□ Validate settings upon exit