Mech-Eye Industrial 3D Cameras

Mech-Mind

May 17, 2022

CONTENTS

Thank you for using Mech-Eye Industrial 3D Camera!

In conjunction with Mech-Eye SDK (including Mech-Eye Viewer and Mech-Eye API), Mech-Eye Industrial 3D Camera can be used to capture and edit 2D images, depth maps, and point clouds. This manual includes the following contents:

- Quick Start Guide
- Mech-Eye Industrial 3D Camera User Manual
- Mech-Eye Viewer User Manual
- Mech-Eye SDK User Manual

Resources

Quick Start Guide

Quick Start Guide

Mech-Eye Industrial 3D Camera User Manual

Mech-Eye Industrial 3D Camera User Manual

Mech-Eye Viewer User Manual

Mech-Eye Viewer User Manual

Mech-Eye SDK User Manual

Mech-Eye SDK User Manual

If you cannot find what you are looking for in our manual, please contact us directly at support@mech-mind.net.

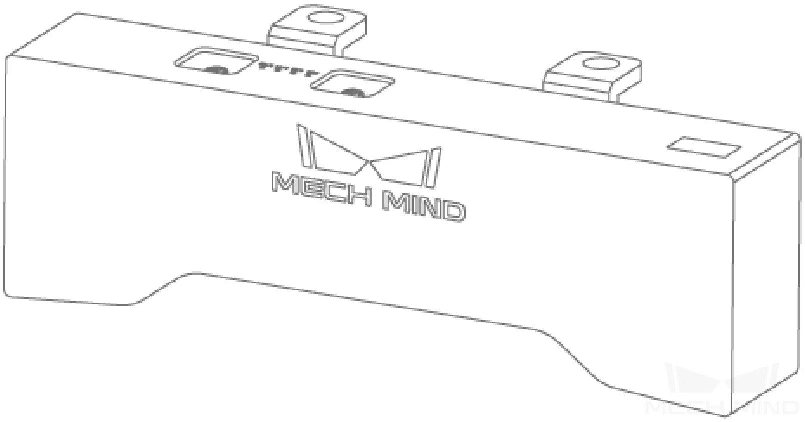
CHAPTER

ONE

QUICK START GUIDE

Package Contents

- **Camera**



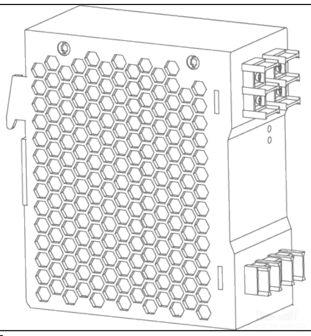
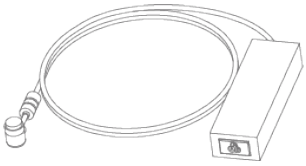
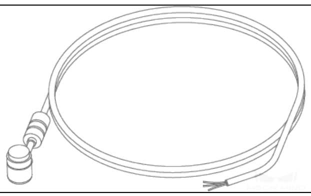
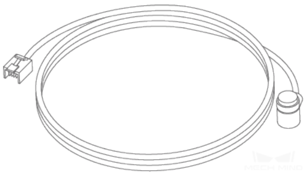
- **User Manual**

Hint:

- When the camera leaves the factory, one or more L shaped brackets have been attached to the back of the camera with 8 screws.

- Before using, please make sure that the package and the camera are intact, and there is no missing accessories.

Cables and Accessories

DIN rail power supply	Adapter
	
Power Cable	Ethernet Cable
	

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 Camera and Industrial PC*.

Mounting the Camera

- **Mounting by threaded holes** Use a wrench to pre-tighten the screws in a specified sequence, and then tighten all the screws, as shown in *Figure 1* .

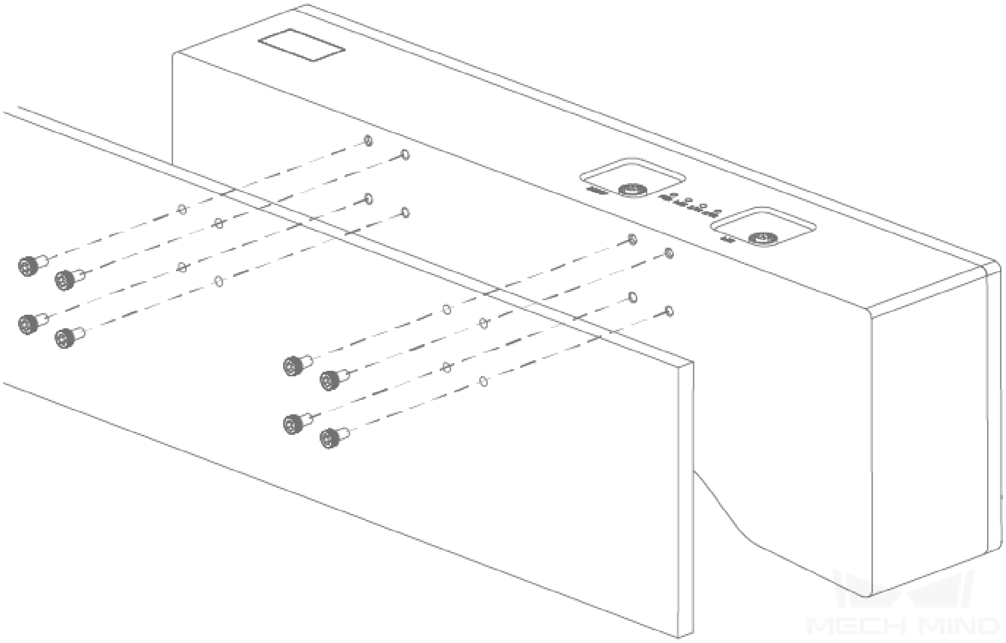


Figure 1 Mounting by threaded holes

- **Mounting with L shaped bracket(s)** Use a wrench to tighten the two screws, as show in *Figure 2*.

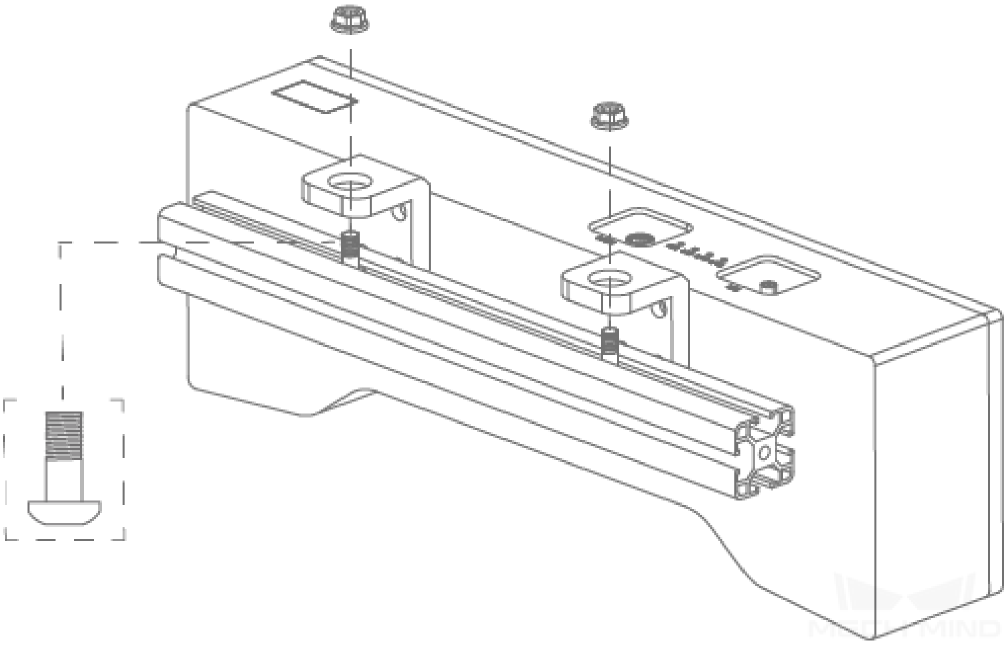
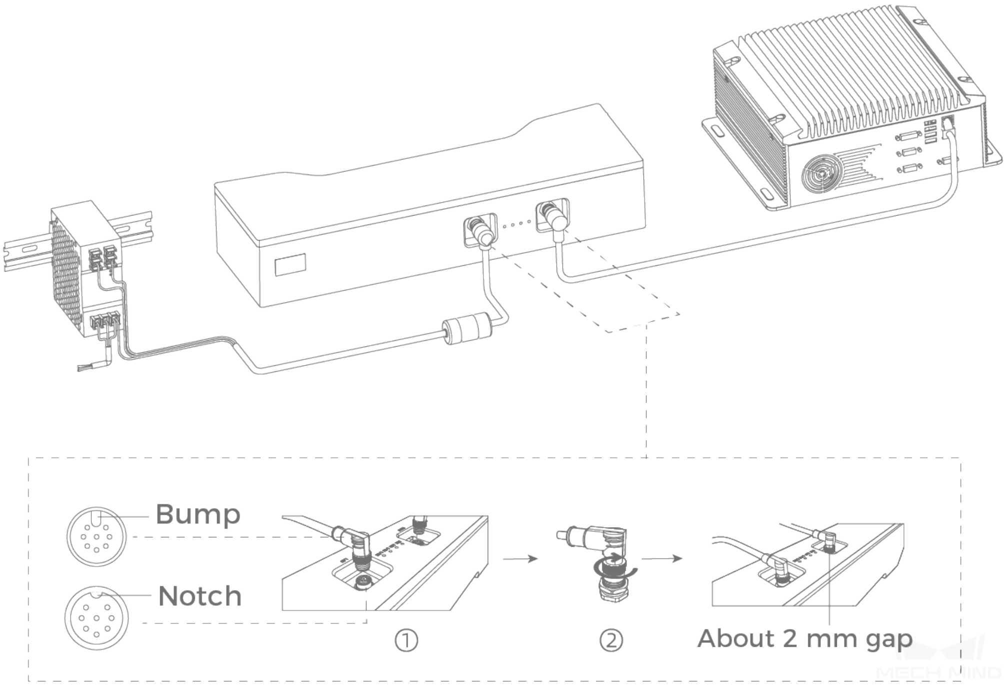


Figure 2 Mount with a L shaped bracket

Hint: The L shaped bracket(s) has(have) been attached to the back of the camera after manufacturing.

Connect the Camera

- Use the **Network Cable** to connect the Industrial PC and the **LAN port** of the camera.
- Use the **Power Adapter** to connect the power supply and **DC 24V connector**, as shown below.



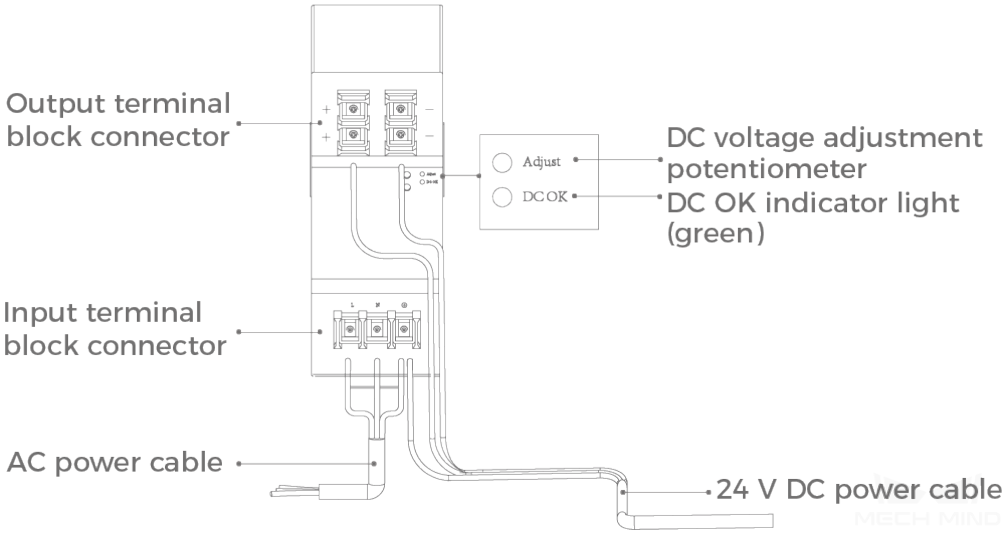
When inserting the ethernet cable and power cable:

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

Attention:

- If multiple cameras or IPCs need to be connected, a network switch can be used.
- 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 or the DIN rail power supply should be reliably grounded. If using multiple DIN rail power supplies, ensure enough distance in between when mounting.

- **DIN Rail Power Supply**


Output terminal block connector
DC voltage adjustment potentiometer
Adjust
DC OK
DC OK indicator light (green)
Input terminal block connector
AC power cable
24 V DC power cable

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

The AC power cable has three wires: L, N, and PE (⏚).
The 24 V DC power cable has three wires: V+, V- and PE (⏚).

Hint: Please attach the plastic cover once you finish wiring to ensure the wires are isolated from each other.

Mech-Eye Viewer Quick Start Guide

After connecting the camera to Mech-Eye Viewer, you can capture images, adjust parameters according to the characteristics of the target object, and obtain 2D images, depth maps, and point clouds of high quality. Please refer to *Mech-Eye Viewer Quick Start Guide* for detailed information.

1.1 Mech-Eye Viewer Quick Start Guide

1.1.1 1. Connect to the Camera

Open Mech-Eye Viewer and go to *Device Finder*, as shown below. If the camera is connected properly , it will be displayed in the **Detected Camera List**. Select the camera in the list and click on *Connect Camera*; you can also double click on the camera you would like to connect in the list to connect it.

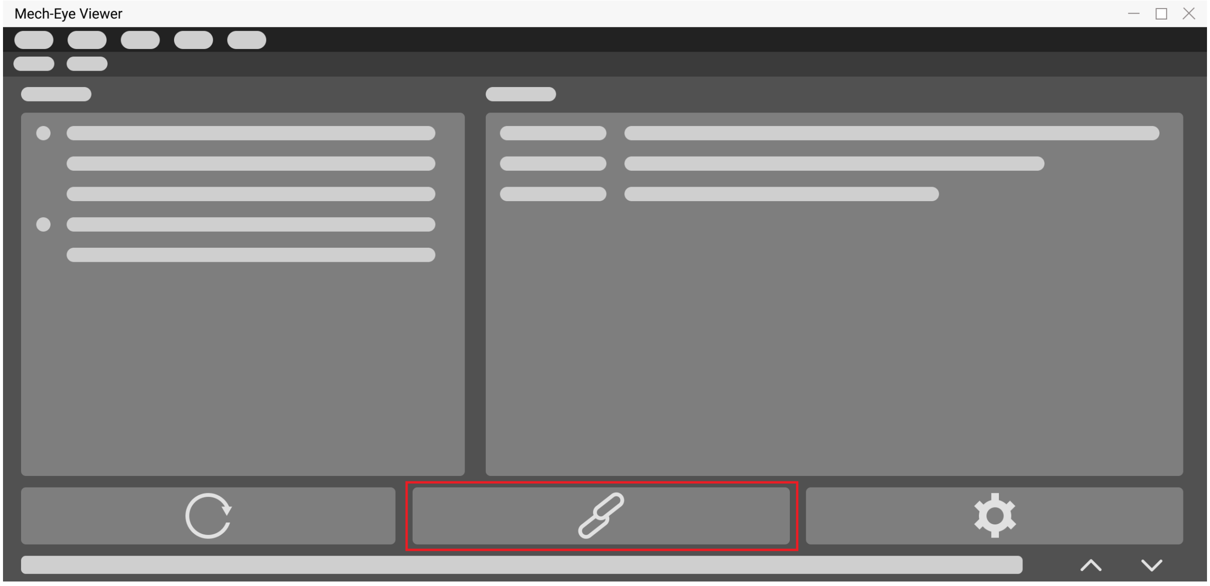


Figure 1 Device finder

1.1.2 2. Capture Images

You will open the *Camera Viewer* window if the camera is connected to the software successfully , as shown in *Figure 2*. Click on *Continuous Scan* or *Scan Once* to capture images, and you can switch among 2D image, depth map or point cloud in the lower left tabs.

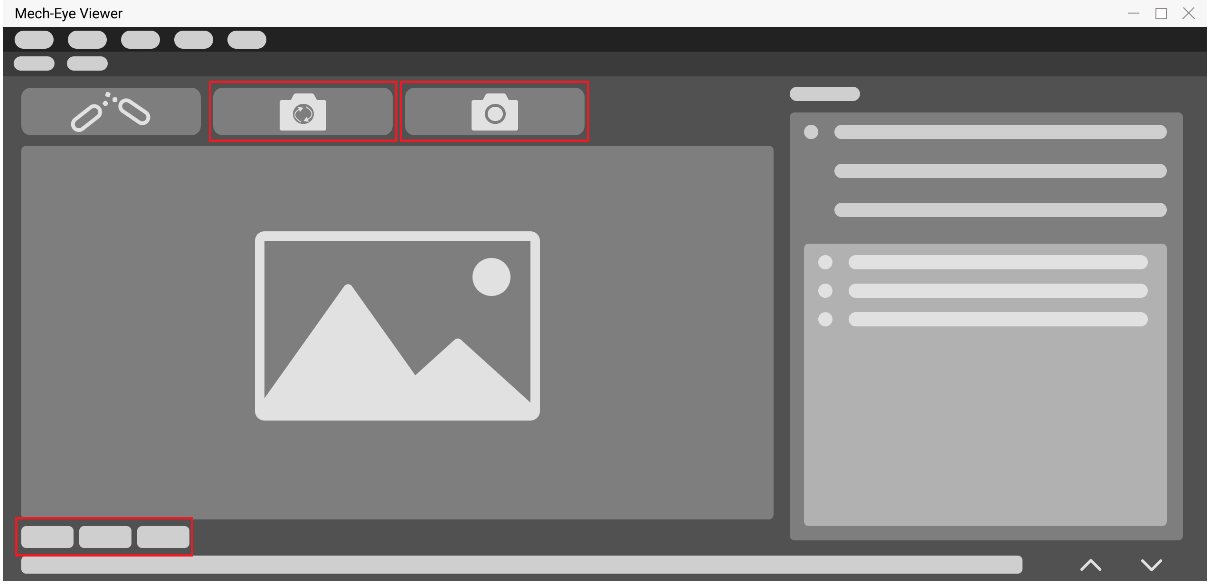


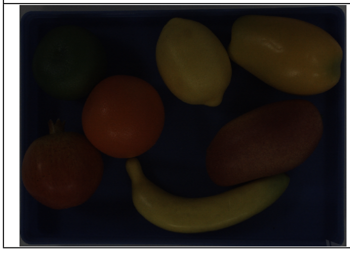
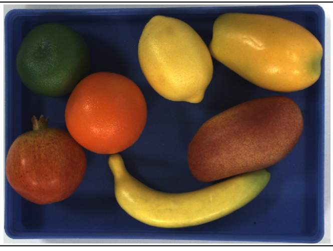
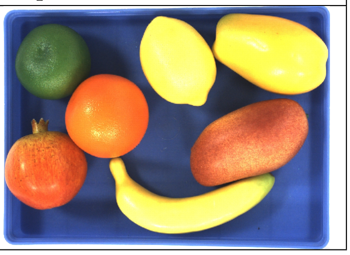
Figure 2 Camera viewer

1.1.3 3. Adjust Parameters

Adjust parameters to optimize the images.

- **2D Image Optimization** Adjust the **Exposure Mode** and relevant parameters of **2D Scanning** in *Parameters*. Use continuous scan can visualize the result when adjusting exposure parameters. Please refer to *Parameter Adjustment for 2D Image* for more detailed instructions.

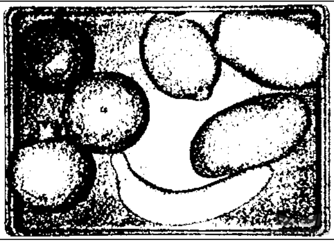
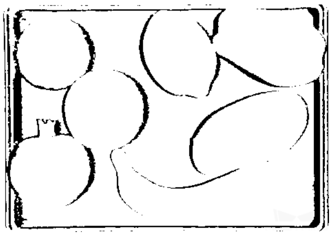
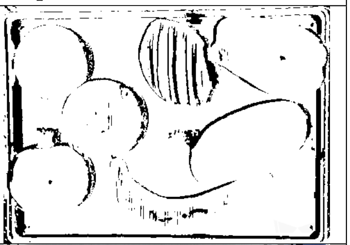
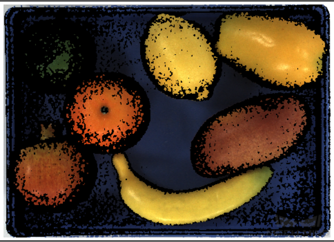
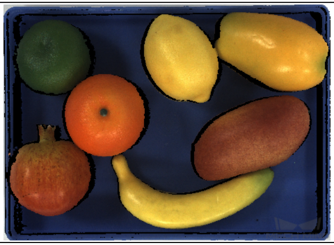
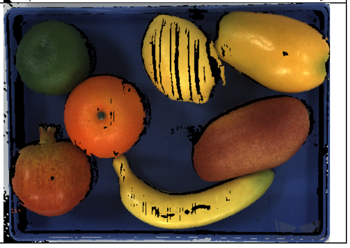
  The results of different exposure time in **Timed Exposure** mode are illustrated in the table below:

Exposure Time 5ms	Exposure Time 20ms	Exposure Time 40ms
		

  Hint: The exposure time will affect the brightness of the image. The longer the exposure time, the brighter the image will be; the shorter the exposure time, the darker the image will be.

- **Depth Map and Point Cloud Optimization** Adjust the **Exposure Time** and **Scan Multiplier** of **3D Scanning** in *Parameters*. The point cloud of the object can only be displayed while using Scan Once. You can use *Exposure Assistant* to obtain the optimal parameter sets. Please refer to *Parameter Adjustment for Depth Map* for more detailed instructions.

  The results of adjusting **3D Parameters** are illustrated in the table below:

Exposure Time 0.25ms	Exposure Time 20ms	Exposure Time 9ms
		
		

- **Point Cloud Optimization** Adjust **Cloud Smooth Mode**, **Cloud Outlier Filter Mode**, and **Fringe Contrast Threshold** of Point Cloud Processing in *Parameters*. Please refer to *Parameter Adjustment for Point Cloud* for more detailed instructions.

1.1.4 1. Save Data

Go to **File** and click on **Save Raw Data of Camera** and **Save Images**. You can click either **Save Raw Data of Camera**, or **Save Images** which enables to select the image type you would like to save.

CHAPTER

TWO

MECH-EYE INDUSTRIAL 3D CAMERA USER MANUAL

Mech-Eye Industrial 3D Camera (hereinafter camera) is developed by Mech-Mind Robotics. This manual provides information about the basics, general operation, and maintenance of the camera. Please read the manual before using the camera to avoid incidents.

2.1 Camera Models

Mech-Mind Robotics developed cameras of different models for different scenarios. Please choose the camera model according to actual needs. The characteristics and applicable scenarios of different camera models are shown in the table below.

Models	Characteristics	Applicable Scenarios
Nano	Small size, high precision, able to generate complete and precise point cloud of objects with complex structures	Assembling, screwing, high-precision picking, detection and other scenarios that require a high precision
Pro M Enhanced	High speed, high precision, large field of view, able to yield good imaging results of different objects (including dark objects and objects with varying reflectivity to a certain extent)	Typical application scenarios that require a high precision, such as industrial inspection, measurement, academic research, etc.
Pro S Enhanced		
Laser L	Use structured light from lasers, work against unwanted ambient light while maintaining a high precision and high speed	Machine tending, positioning, assembling, etc.
Log M	High precision, work against reflection of light and other complex situations, capable of recognizing objects with complex structures or dark objects	Typical logistics scenarios, such as parcel and product sorting
Log S		
Deep	Large field of view, deep depth of field, high speed, applicable to various common pallet patterns	Depalletizing of boxes, sacks, etc.

2.2 Safety Instructions and Regulations

This chapter includes safety instructions and regulations. Please read through the following contents carefully before using the camera to avoid accidents.

2.2.1 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, tested, operated, and maintained properly.
- Laser is hazardous; please be acquired of 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 or fabricate the camera in any form. Do not repair or disassemble the camera by yourself. Damage caused by modifications will void the warranty.
- 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.
- This product should only be mounted, used, cleaned, or maintained by adults. This product should be mounted by trained personnel. This product may include detachable components, which should be kept away from children to avoid harm caused by swallowing and misuse.
- Do not use the camera in extremely high or low temperature. The operating temperature range of the laser cameras is -10–45 °C.
- 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.

⚠ Check the Camera

- Before using, please check the camera carefully for damage, water damage, 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 occurs, please disconnect the power and stop using immediately.

⚠ 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 power cable and ethernet cable should be properly grounded. It is recommended to use the isolated adapter or DIN rail power supply provided by our company.
- Use copper conductors only.

⚠ Precautions for Using Laser Cameras

- Operator should wear suitable laser safety eyewear to avoid damage caused by direct eye exposure (applicable for class 3R).
- 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 metal object in the path of the laser beam.

⚠ Notice for Disposal

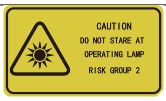
- Please comply with local laws and regulations when disposing the camera to avoid polluting the environment. Please do not dispose the camera irresponsibly. Improper disposal may pollute the environment.
- The pollution degree of the camera is 3.

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

2.2.2 Regulations

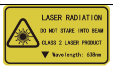
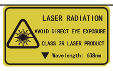
Warning Signs

LED Light Radiation Warning Sign

Camera Model	Risk Group	Warning Sign	Precautions
Nano	Risk Group 2	 CAUTION DO NOT STARE AT OPERATING LAMP RISK GROUP 2	Do NOT stare into the beam.
Pro S Enhanced			
Pro M Enhanced			
Log S			
Log M			
Deep			

Laser Radiation Warning Sign

Laser L emits a red laser with the center wavelength of 638nm, which is classified as Class 2 or Class 3R.

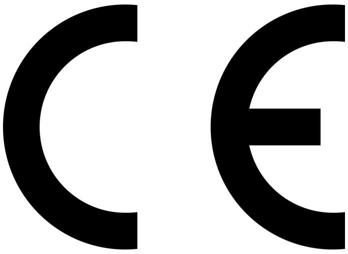
Class	Warning Sign	Precautions
Class 2	 LASER RADIATION DO NOT STARE INTO BEAM CLASS 2 LASER PRODUCT Wavelength: 638nm	a. DO NOT look into the laser beam or the reflected laser beam directly or with optical instruments. b. A clear laser warning sign must be displayed in each area the laser output reaches. c. The laser beam must be lower or higher than and never at eye level.
Class 3R	 LASER RADIATION AVOID DIRECT EYE EXPOSURE CLASS 3R LASER PRODUCT Wavelength: 638nm	a. Operator should wear suitable laser safety eyewear to avoid direct eye exposure. b. A clear laser warning sign must be displayed in each area the laser output reaches. c. The laser beam must be lower or higher than and never at eye level.

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

Certificates

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.

CE



Mech-Eye Industrial 3D camera is compliant with the requirements and standards of all applicable EU directives.

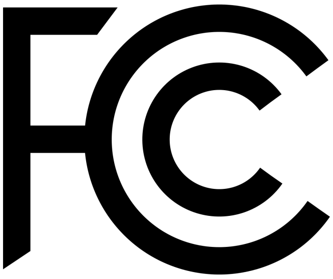
Safety of Laser Products:

- IEC 60825-1: 2014

EMC Standards:

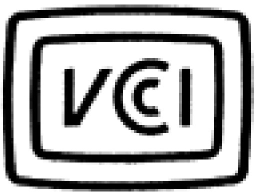
- EN55032: 2015+A11: 2020
- EN IEC 61000-3-2: 2019+A1: 2021
- EN 61000-3-3: 2013+A1: 2019
- EN 55035: 2017+A11: 2020

FCC



Mech-Eye Industrial 3D Camera is compliant with US ANSI C63.4 and 47 CFR PART 15B electromagnetic compatibility assessment requirements.

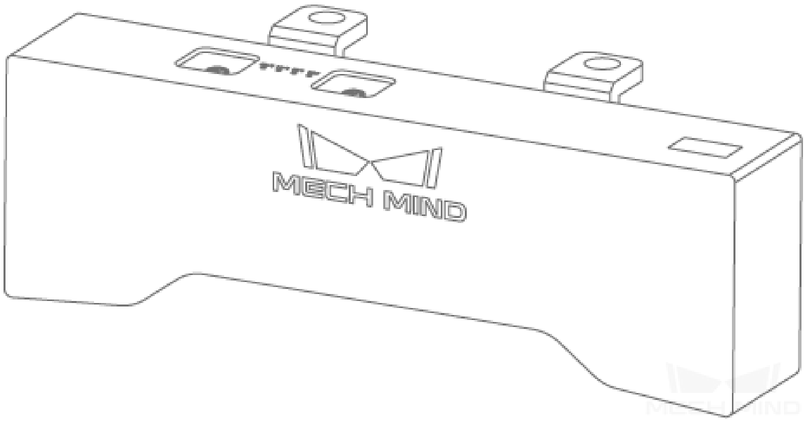
VCCI



Mech-Eye Industrial 3D Camera is compliant with Japanese VCCI-CISPR 32: 2016 standards.

2.3 Package Contents

Camera

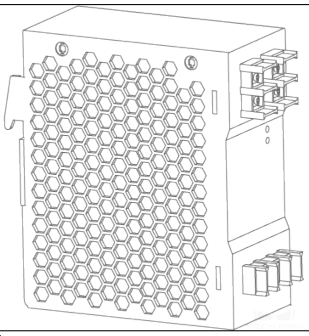
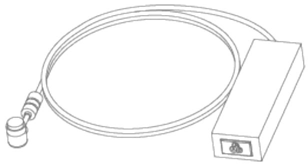
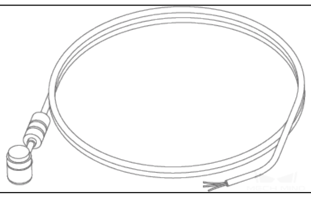
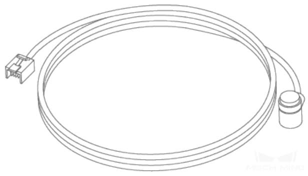


User Manual

Hint:

- The L shaped bracket(s) is attached to the back of the camera with 8 bolts when packaged.
- Before using, please make sure that the package and the camera are intact, and that no accessories are missing.

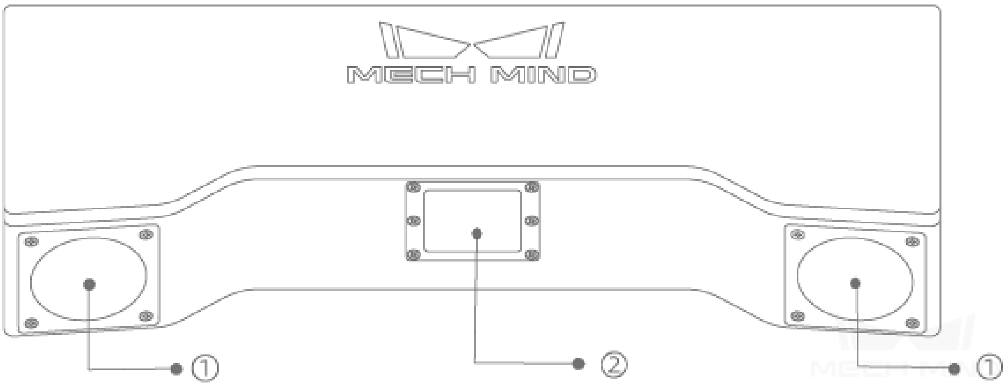
2.4 Cables and Accessories

DIN rail power supply	Adapter
	
Power Cable	Ethernet Cable
	

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 Camera and Industrial PC*.

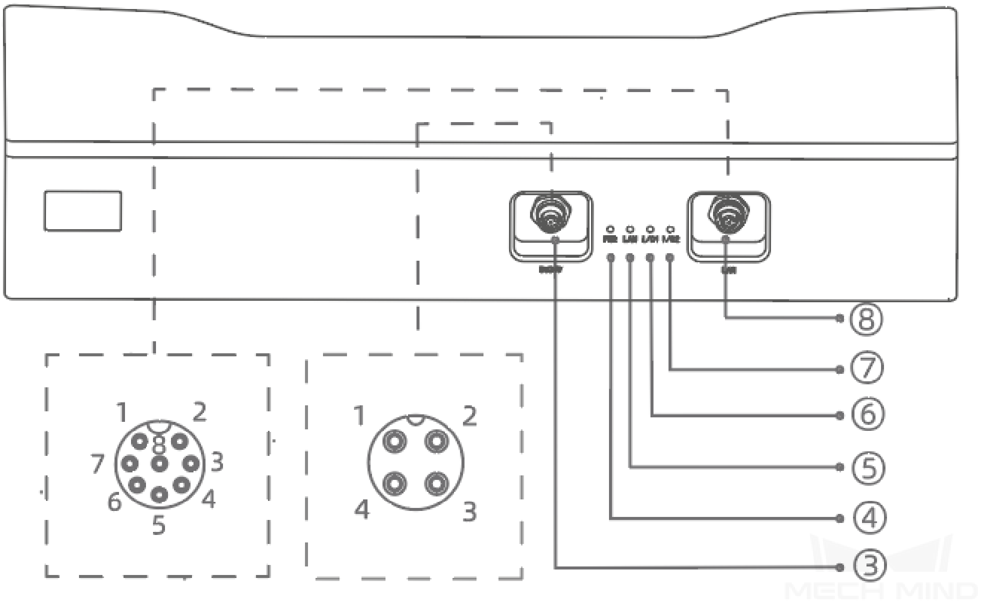
2.5 Camera Diagram

Camera Front



No.	Name	Function
	Camera lens	Capture images
	Emitter	Emit structural light

Camera Back



No.	Name	Function
	DC 24V port	1: GND 2: GND 3: 24 V DC 4: 24 V DC
	PWR indicator light	The indicator light turns on when power is supplied to the camera, and off otherwise.
	LAN indicator light	The indicator light turns on when the camera is connected to the Ethernet, and off otherwise. * Gigabit network: green * Megabit network: yellow
	I/01	N/A
	I/02	N/A
	LAN 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

Hint: The above diagrams are for illustration only; the actual product may differ.

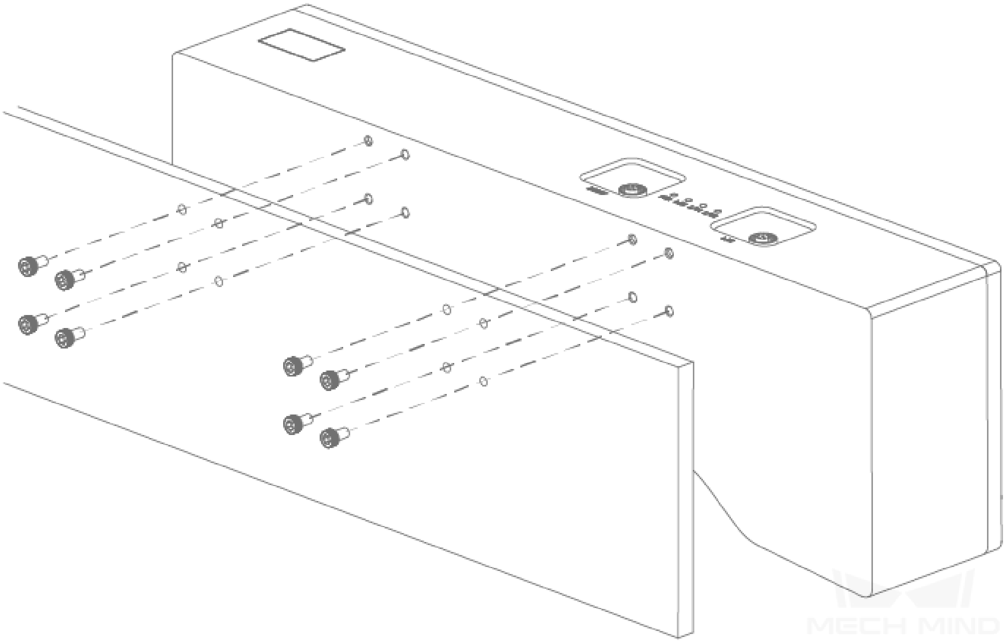
2.6 Mounting the Camera

Hint:

- Preparation beforehand: wrench, and bolts if the ones included are not suitable.
- The following instructions are illustrated with Log S.

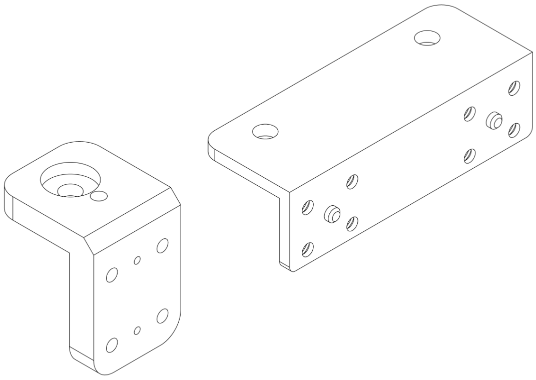
2.6.1 Mount through Threaded Holes

Use the wrench to pre-tighten the bolts in the sequence specified in the figure below, and then fully tighten all the bolts.

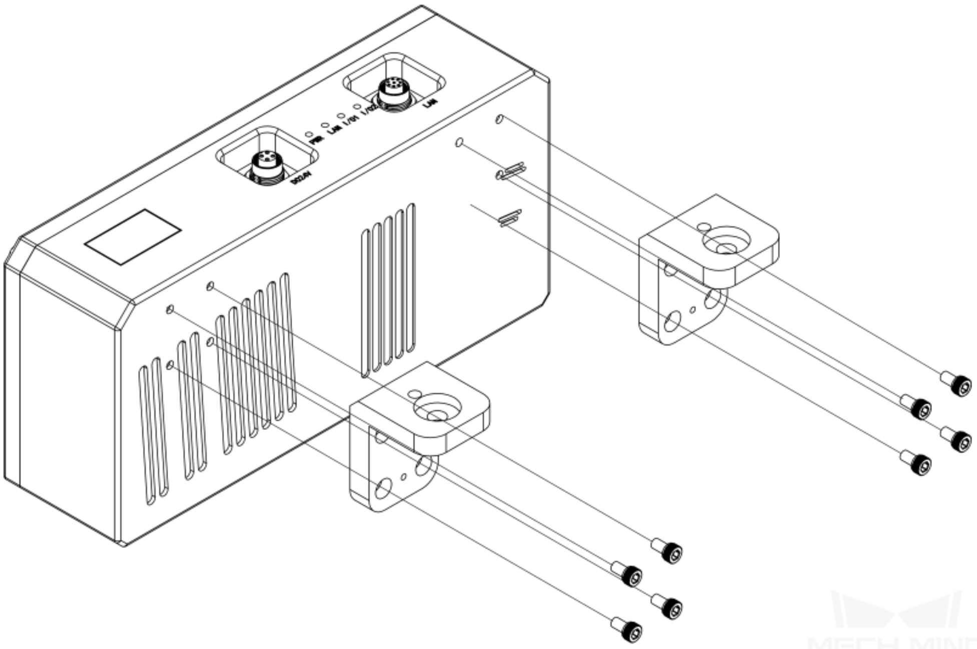


Hint: Please disassemble the L shaped bracket(s) before mounting.

- **Types of L shaped bracket:** As shown below.

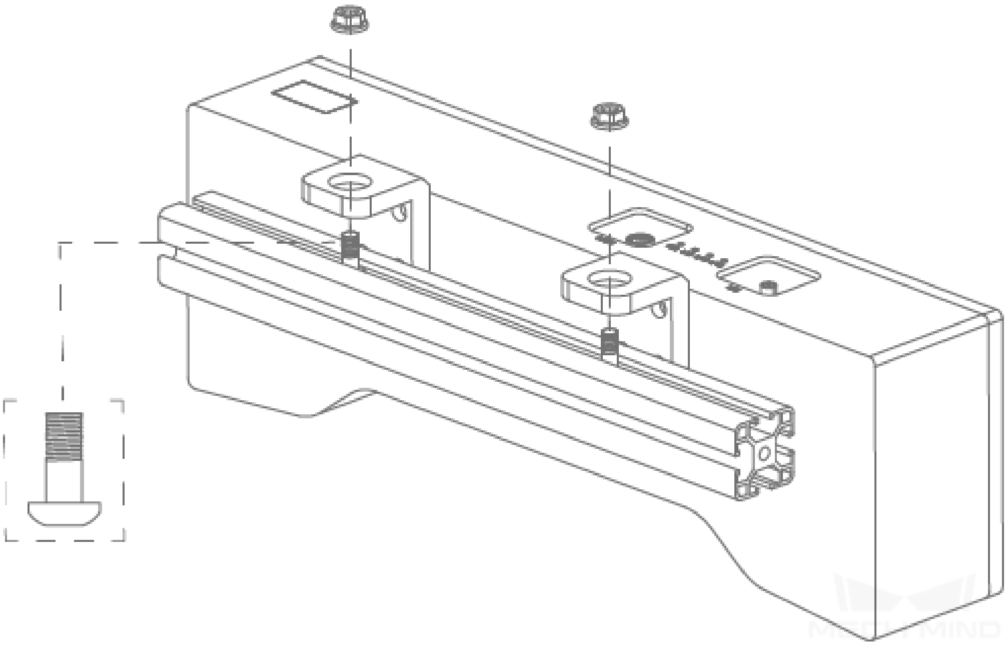


- **Disassemble the L shaped bracket(s):** Use the wrench to disassemble the L shaped bracket(s), as shown below.



2.6.2 Mount through the L Shaped Bracket(s)

Use the wrench to tighten the two bolts, as shown below.

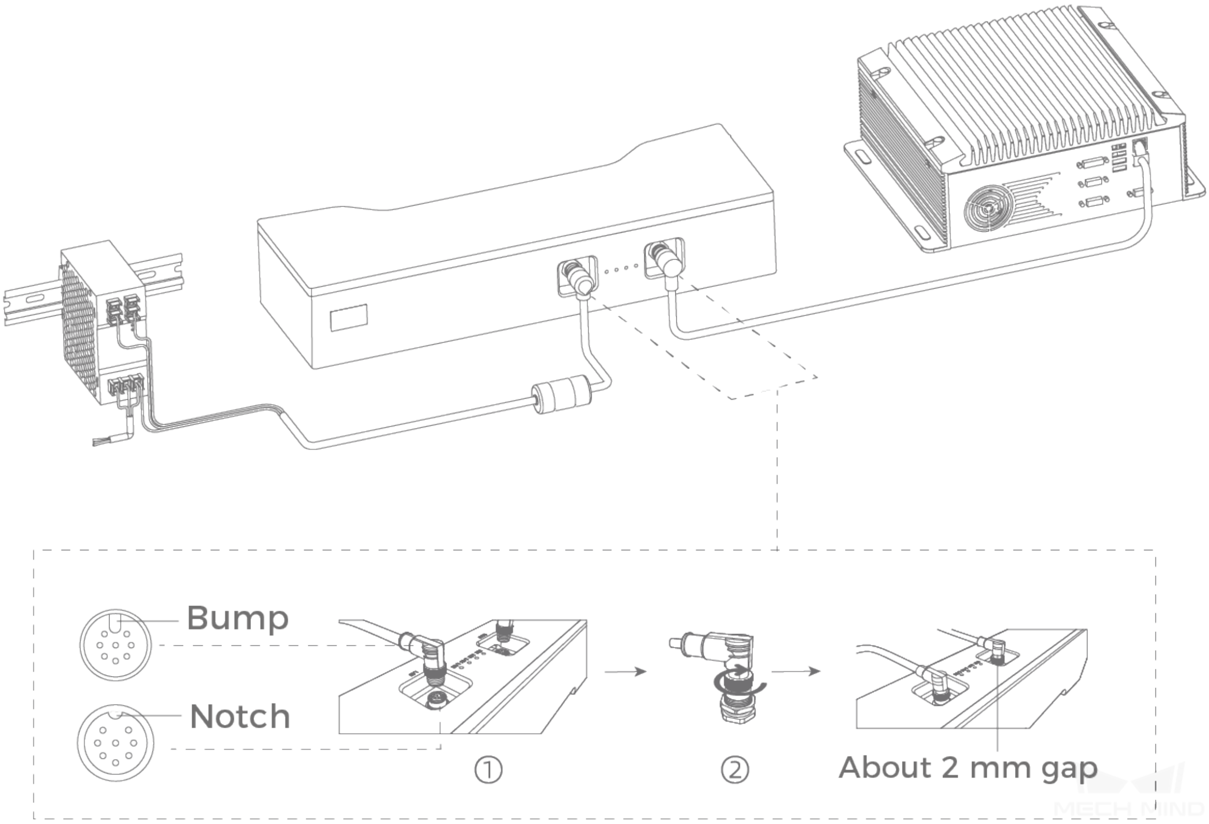


Hint: The L shaped bracket(s) is attached to the back of the camera when packaged.

2.7 Connect Camera and Industrial PC

2.7.1 Connect Directly

- **Ethernet Cable**: insert one end into the port on the IPC and the other into the **LAN port** on the camera.
- **Power Adapter**: insert the connector into the **DC 24V port** and the plug into a power supply, as shown below.



When inserting the ethernet cable and power cable:

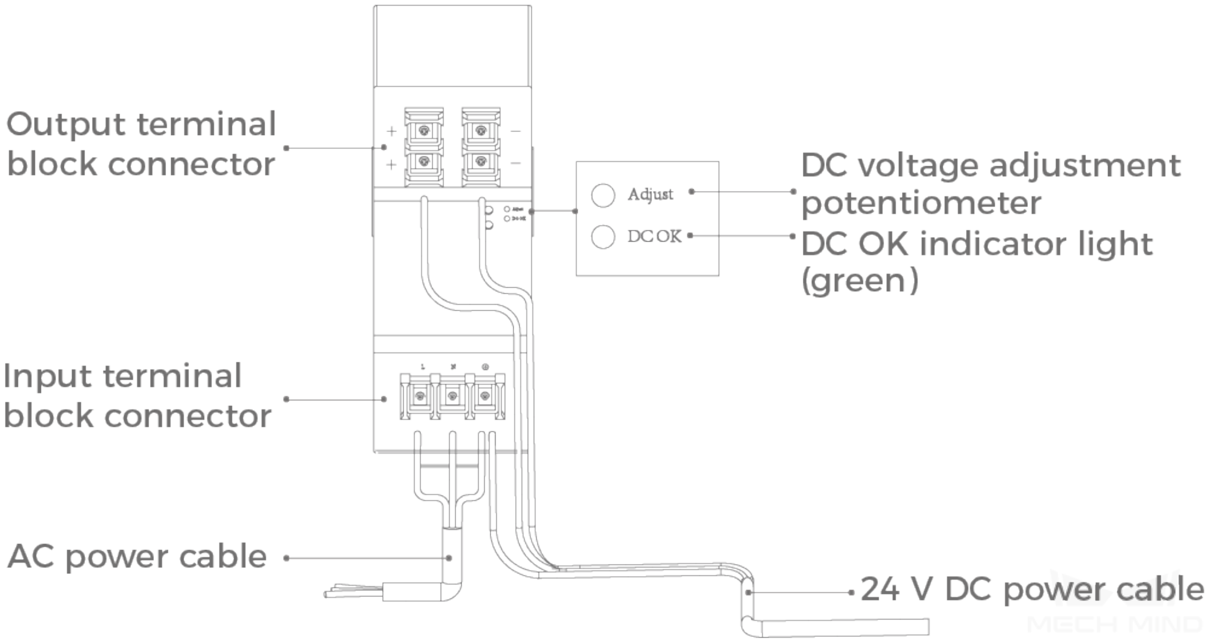
Make sure the bump on the connector is aligned with the notch of the port.
Tighten the nut after inserting the cable.

Attention:

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

- 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 or the DIN rail power supply should be reliably grounded. If using multiple DIN rail power supplies, ensure enough distance in between when mounting.

- **DIN Rail Power Supply**


Output terminal block connector
DC voltage adjustment potentiometer
Adjust
DC OK
DC OK indicator light (green)
Input terminal block connector
AC power cable
24 V DC power cable

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

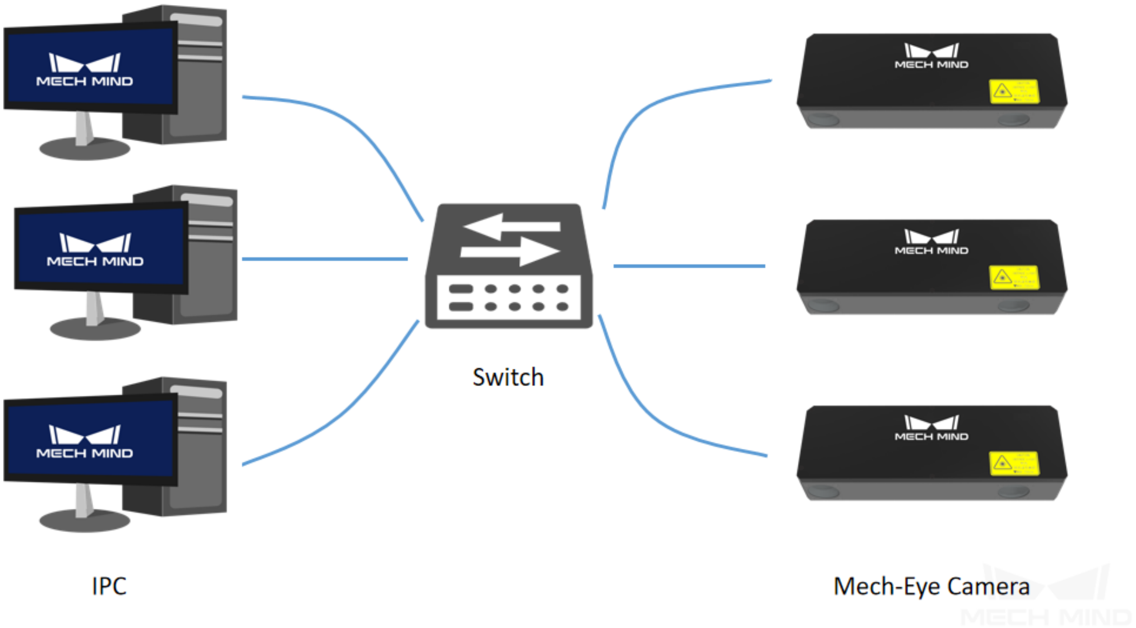
The AC power cable has three wires: L, N, and PE (⏚).

The 24 V DC power cable has three wires: V+, V- and PE (⏚).

Hint: Please attach the plastic cover once you finish wiring to ensure the wires are isolated from each other.

2.7.2 Connect through a Network Switch

- If multiple cameras or IPCs need to be connected, a network switch can be used, as shown below.


MECH MIND
MECH MIND
MECH MIND
Switch
MECH MIND
MECH MIND
MECH MIND
IPC
Mech-Eye Camera

2.8 Specifications

2.8.1 Specifications of Non-Laser Cameras

Product Name	Mech-Eye Industrial 3D Camera				
Model	Pro S Enhanced	Pro M Enhanced	Log S	Log M	Nano
Recommended working distance	500–1000mm	800–2000mm	500–1000mm	800–2000mm	300–600mm
FOV	350 × 220mm @ 0.5m	500 × 350mm @ 0.8m	360 × 250mm @ 0.5m	520 × 390mm @ 0.8m	220 × 160mm @ 0.3m
	690 × 430mm @ 1.0m	1360 × 860mm @ 2.0m	710 × 490mm @ 1.0m	1410 × 960mm @ 2.0m	430 × 320mm @ 0.6m
Resolution	1920 × 1200	1920 × 1200	1280 × 1024	1280 × 1024	1280 × 1024
Pixels	2.3 MP	2.3 MP	1.3 MP	1.3 MP	1.3 MP
Z repeatability ( )	0.05mm @ 1m	0.2mm @ 2m	0.1mm @ 1m	0.3mm @ 2m	0.1mm @ 0.5m
Calibration accuracy	0.1mm @ 1m	0.2mm @ 2m	0.2mm @ 1m	0.3mm @ 2m	0.1mm @ 0.5m
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	150mm	280mm	150mm	280mm	68mm
Dimensions	About 270 × 72 × 130mm	About 387 × 72 × 130mm	About 270 × 72 × 130mm	About 387 × 72 × 130mm	About 145 × 51 × 85mm
Weight	About 2.2kg	About 2.4kg	2.2kg	2.4kg	About 0.7kg
Operating temperature	0–45 °C				
Communication Interface	Ethernet				
Operating voltage	24V DC				
Safety and EMC	CE/FCC/VCCI				
Ingress Protection Rating	IP65				
Cooling	Passive				

2.8.2 Specifications of Laser Camera

Product Name	Mech-Eye Industrial 3D Camera	
Model	Laser L	Laser L Enhanced -12MP-M V3S
Optimal Scanning Range	1500–3000mm (59.1–118.1 inch)	
Near FOV	1500 × 1200 mm @ 1.5 m (59.1 × 47.2 inch @ 4.9 ft)	
Far FOV	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
Z Repeatability ( )	0.5 mm @ 3 m (0.02 inch @ 9.8 ft)	0.5 mm @ 3 m (0.02 inch @ 9.8 ft)
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 –113)	
Communication Interface	Ethernet	
Power Supply	24 VDC	
Max. Power	70 W	
Laser Class	Class 2 / Class 3R[1]	
Safety and EMC	CE / FCC / VCCI	
IP Rating	IP65	
Cooling	Passive	

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

2.9 Maintenance

2.9.1 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, kerosene, etc. 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.

2.9.2 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.

⚠ **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.

2.10 Trademark and Legal Statement

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

Unless authorized in writing in advance by Mech-Mind Robotics Technologies Ltd., 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 the Company’ s trademark rights will be punished in accordance with the law.

Mech-Mind Robotics reserves all rights regarding this user manual. According to copyright laws, unless authorized in writing by Mech-Mind Robotics, this user manual shall 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 Robotics, the contents of the user manual may not be used for any other purposes.

CHAPTER

THREE

MECH-EYE SDK USER MANUAL

Mech-Eye SDK includes Mech-Eye API, user documentation, and Mech-Eye Viewer, a camera configuration and data visualization software developed using Mech-Eye API. You can obtain images and point clouds from Mech-Eye Industrial 3D cameras using Mech-Eye SDK.

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.

Check the following section for information on how to **install Mech-Eye SDK**

3.1 Install Mech-Eye SDK

3.1.1 Install Mech-Eye SDK on Windows

Download and install Mech-Eye SDK.

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

When installing, make sure to check the **Add to Path** option.

3.1.2 Install Mech-Eye SDK on Ubuntu

Download Mech-Eye API.

Hint:

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

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

```
sudo dpkg -i MechEyeApi_1.5.2_amd64.deb
```

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

```
sudo dpkg -i MechEyeApi_1.5.2_arm64.deb
```

If you need to check if MechEyeSDK is installed, please enter the following command:

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

If you need to uninstall MechEyeSDK, please enter the following command:

```
sudo dpkg -P MechEyeApi
```

Check the following section for a tutorial on **how to use Mech-Eye API**

3.2 Capture Tutorial

This section describes how to use Mech-Eye API to capture 2D images, depth maps, and 3D point clouds, and save the data.

3.2.1 Requirements

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

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

3.2.2 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()
```

3.2.3 Connect to a Camera

After finding the connectable cameras, you can connect to a camera using the following commands.

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)])
```

3.2.4 Set the Camera Parameters

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

2D Scanning Parameters

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

Set 2D Scanning Parameters (C++)

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

Set 2D Scanning Parameters (C#)

```
device.setScan2DExposureMode(Scan2DExposureMode.Timed);
device.setScan2DExposureTime(100);
```

Set 2D Scanning Parameters (Python)

```
self.device.set2D_exposure_mode("Timed")
self.device.set2D_exposure_time(100.0)
```

3D Scanning Parameters

Before capturing 3D data, you need to set the 3D scanning parameters, including **Scan Multiplier**, **Exposure Time**(s), **Scan 3D ROI**, **Cloud Smooth Mode**, etc.

Set 3D Scanning Parameters (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 3D Scanning Parameters (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 3D Scanning Parameters (Python)

```
self.device.set3D_exposure([5.0, 10.0])
self.device.set_depth_range(100, 1000)
self.device.set_3D_roi(0, 0, 500, 500)
self.device.set_cloud_smooth_mode("Normal")
self.device.set_cloud_outlier_filter_mode("Normal")
```

3.2.5 Capture Images

After setting the parameters, you can start capturing images by sending the commands to control the camera.

Capture 2D Images

Capture 2D Images (C++)

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

Capture 2D Images (C#)

```
ColorMap colorMap = new ColorMap();
device.captureColorMap(ref color);
```

Capture 2D Images (Python)

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

Capture 3D Images

Capture 3D Images (C++)

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

Capture 3D Images (C#)

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

Capture 3D Images (Python)

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

3.2.6 Save Data

Save 2D Images

You can convert captured 2D image data into OpenCV image data and save them as .png files.

Save 2D Images (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 Images (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 Images (Python)

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

Save Point Clouds

You can generate point clouds from the captured images and save the point clouds to .ply files.

Save Point Clouds (C++)

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

Save Point Clouds (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 Clouds (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)
```

The following section illustrates how to **install, configure, build, and run Mech-Eye API** using sample programs

3.3 Samples

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

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

3.3.1 C++ (Windows)

This section demonstrates how to configure the C++ samples with CMake and then build and run the samples with Visual Studio in Windows.

Sample List

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

Basic samples: camera connection and basic capturing functions.
Advanced samples: advanced capturing functions.
Util samples: obtain information and set parameters.
Laser samples: for laser cameras only.

Basic

- ConnectToCamera: connect to a camera.
- ConnectAndCaptureImage: connect to a camera and capture color images and depth maps.
- CaptureColorMap (OpenCV): capture color images from a camera.
- CaptureDepthMap (OpenCV): capture depth maps from a camera.
- CapturePointCloud (PCL): capture point clouds from a camera.
- CaptureHDRPointCloud (PCL): capture point clouds in HDR mode from a camera.
- CapturePointCloudROI (PCL): set an ROI and capture point clouds from a camera.

Advanced

- CaptureCloudFromDepth (PCL): generate point clouds from depth maps and color images.
- CaptureTimedAndPeriodically (OpenCV & PCL): capture images periodically for a specified length of time.
- CaptureSimultaneouslyMultiCamera (OpenCV & PCL): capture images from multiple cameras simultaneously.
- CaptureSequentiallyMultiCamera (OpenCV & PCL): capture images from multiple cameras sequentially.

Util

- GetCameraIntri: obtain and print the intrinsic parameters of the camera.
- PrintDeviceInfo: print version information of Mech-Eye SDK and the camera firmware.
- SetDepthRange: set the depth range of a camera.
- SetUserSets: provide functions related to parameter groups, such as switching to a different parameter group, adding and deleting parameter groups, etc.
- SetParameters: provide functions related to camera parameters, including all the functions of SetUserSets.

Laser

- SetLaserFramePartitionCount: set the laser scan partition number for a laser camera.
- SetLaserFrameRange: set the field of view for a laser camera.
- SetLaserFringeCodingMode: set the fringe coding mode for a laser camera.
- SetLaserPowerLevel: set the power level for a laser camera.

Requirements

- *Connect to a real camera that functions properly*
- Required software:
  - Mech-Eye SDK
  - Visual Studio (version 2015 and above recommended)
  - CMake (version 3.2 or above recommended)
- Optional software: please install the software required by the samples you would like to use.

	OpenCV (version 3.4.5 recommended)	PCL (version 1.12.1 recommended)
CaptureColorMap	√	
CaptureDepthMap	√	
CapturePointCloud		√
CaptureHDRPointCloud		√
CapturePointCloudROI		√
CaptureCloudFromDepth		√
CaptureTimedAndPeriodically	√	√
CaptureSimultaneouslyMultiCamera	√	√
CaptureSequentiallyMultiCamera	√	√

Note: Instructions on installing the above software are provided below.

Instructions

Below are the step-by-step instructions from installing the required software to running a sample program.

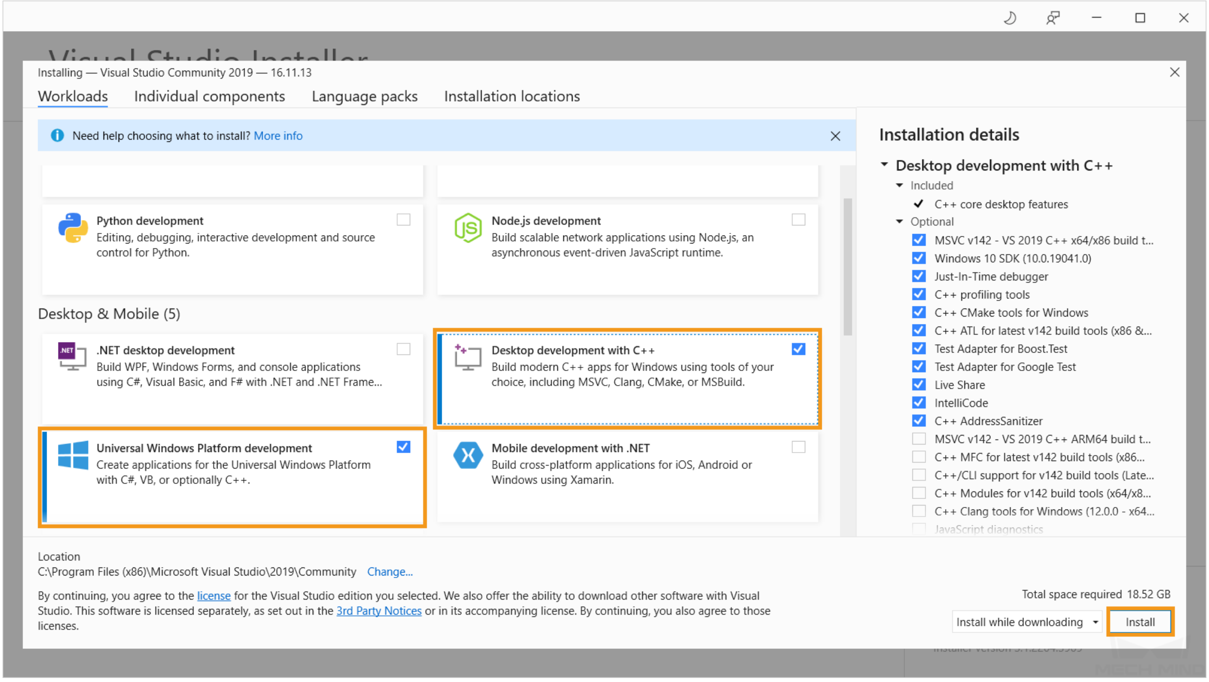
Install Software

Install Mech-Eye SDK

Please refer to *Install Mech-Eye SDK* for instructions on downloading and installation.

Install Visual Studio

1. Download Visual Studio 2019 installer and double-click on the file to start the installation.
2. When the following window is displayed, scroll down to **Desktop & Mobile**, and check **Desktop development with C++** and **Universal Windows Platform development**.



Installing — Visual Studio Community 2019 — 16.11.13

Need help choosing what to install? More info

□ Python development — Editing, debugging, interactive development and source control for Python.

□ Node.js development — Build scalable network applications using Node.js, an asynchronous event-driven JavaScript runtime.

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.

Installation details

- Desktop development with C++
  - Included
    - ✓ C++ core desktop features
  - Optional
    - ☑ MSVC v142 - VS 2019 C++ x64/x86 build t...
    - ☑ Windows 10 SDK (10.0.19041.0)
    - ☑ Just-In-Time debugger
    - ☑ C++ profiling tools
    - ☑ C++ CMake tools for Windows
    - ☑ C++ ATL for latest v142 build tools (x86 &...
    - ☑ Test Adapter for Boost.Test
    - ☑ Test Adapter for Google Test
    - ☑ Live Share
    - ☑ IntelliCode
    - ☑ C++ AddressSanitizer
    - □ MSVC v142 - VS 2019 C++ ARM64 build t...
    - □ C++ MFC for latest v142 build tools (x86...
    - □ C++/CLI support for v142 build tools (Late...
    - □ C++ Modules for v142 build tools (x64/x8...
    - □ C++ Clang tools for Windows (12.0.0 - x64...
    - □ JavaScript diagnostics

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

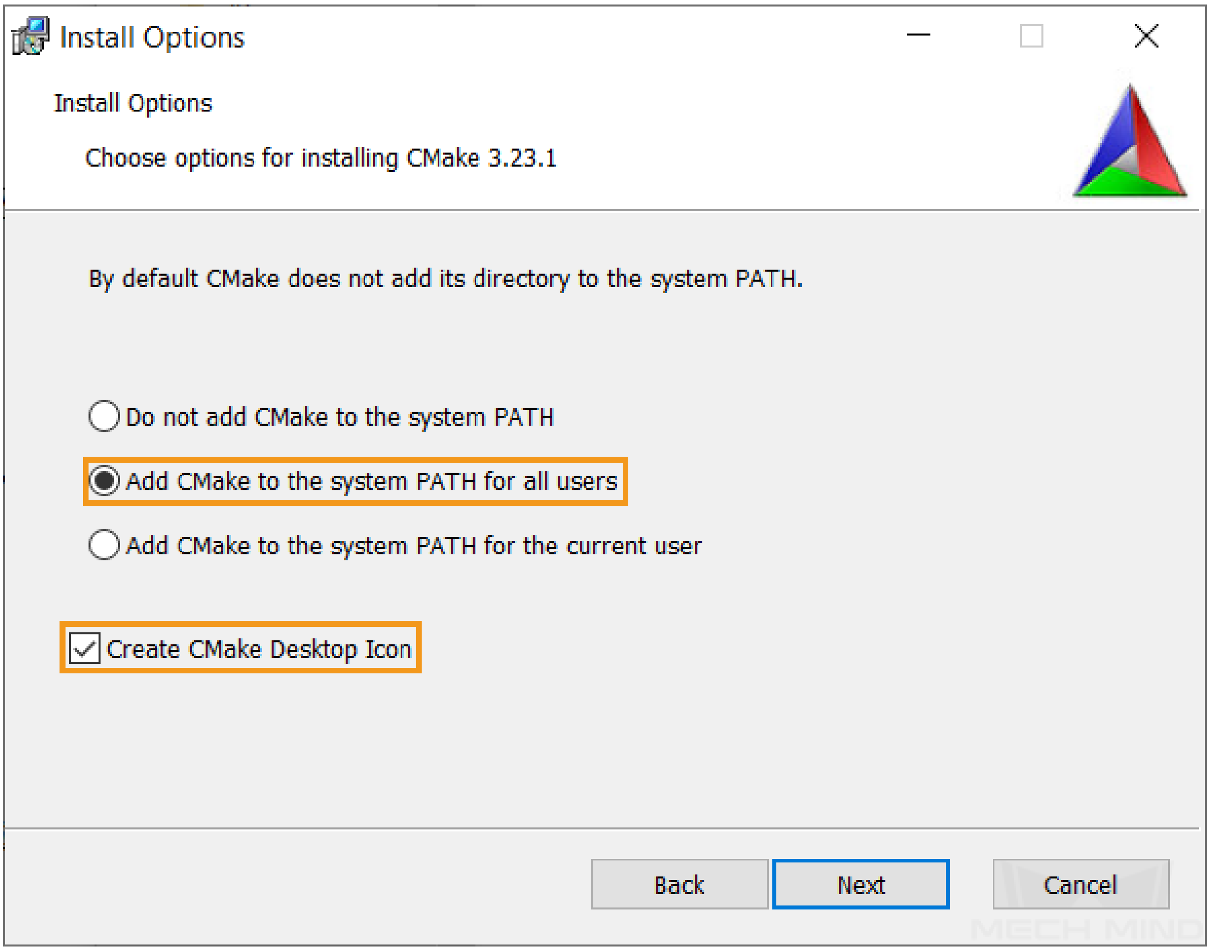
By continuing, you agree to the license for the Visual Studio edition you selected. We also offer the ability to download other software with Visual Studio. This software is licensed separately, as set out in the 3rd Party Notices or in its accompanying license. By continuing, you also agree to those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

Total space required 18.52 GB

3. Click on *Install* in the lower right to proceed. If the installer prompts you to restart your computer, please do so.

Install CMake

1. Download the CMake installer (the MSI file) and double-click on the file to start the installation.
2. Select **Add CMake to the system PATH for all users** and check **Create CMake Desktop Icon**, and then click on *Next*.



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

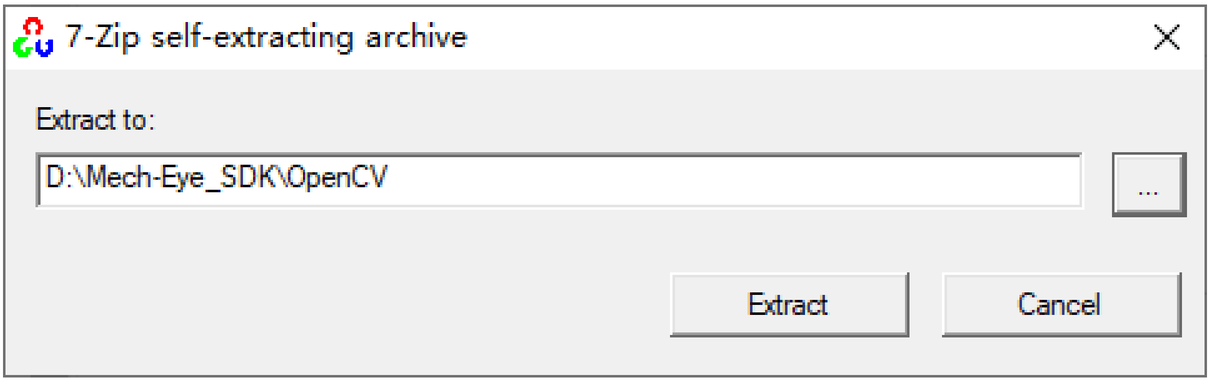
● Add CMake to the system PATH for all users

○ Add CMake to the system PATH for the current user

☑ Create CMake Desktop Icon

Install OpenCV

1. Download the OpenCV installer and double-click on the file to start the installation. Click on *Extract* to proceed.



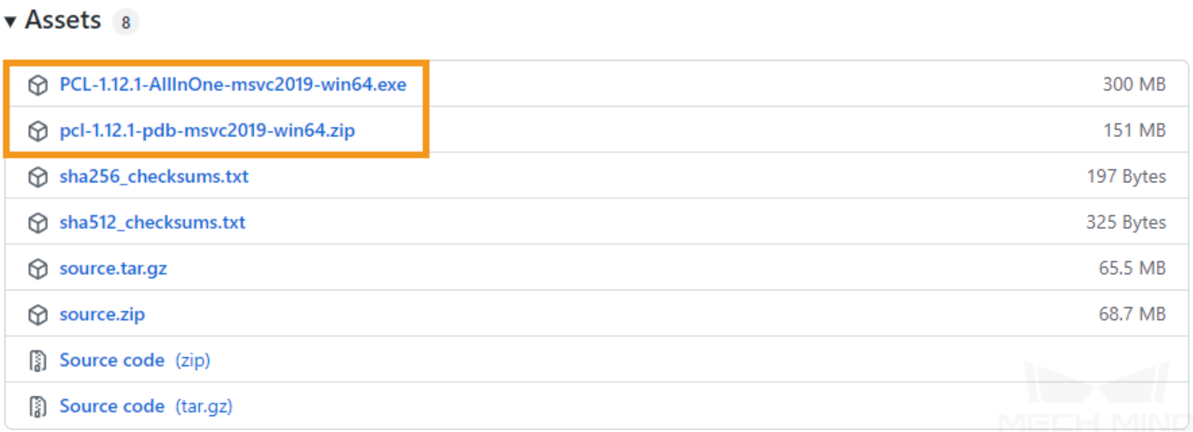
7-Zip self-extracting archive

Extract to:

D:\Mech-Eye_SDK\OpenCV

Install PCL

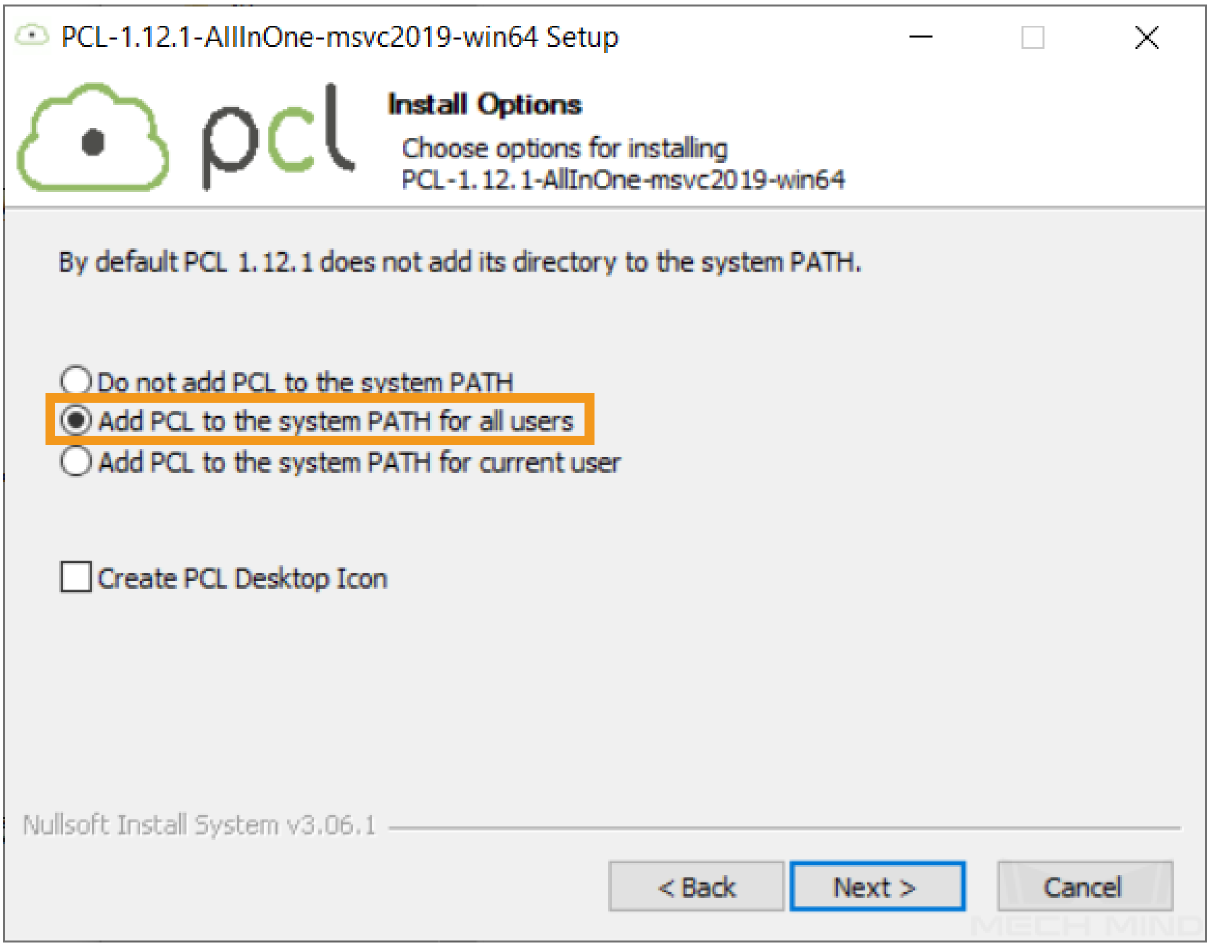
1. Download **PCL-1.12.1-AllInOne-msvc2019-win64.exe** and **pcl-1.12.1-pdb-msvc2019-win64.zip** from the GitHub repository.



▾ Assets 8

PCL-1.12.1-AllInOne-msvc2019-win64.exe	300 MB
pcl-1.12.1-pdb-msvc2019-win64.zip	151 MB
sha256_checksums.txt	197 Bytes
sha512_checksums.txt	325 Bytes
source.tar.gz	65.5 MB
source.zip	68.7 MB
Source code (zip)	
Source code (tar.gz)	

2. Extract **pcl-1.12.1-pdb-msvc2019-win64.zip**.
3. Double-click on **PCL-1.12.1-AllInOne-msvc2019-win64.exe** to start the installation.
4. On the **Install Options** page, select **Add PCL to the system PATH for all users**, and then click on *Next*.



PCL-1.12.1-AllInOne-msvc2019-win64 Setup

Install Options

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

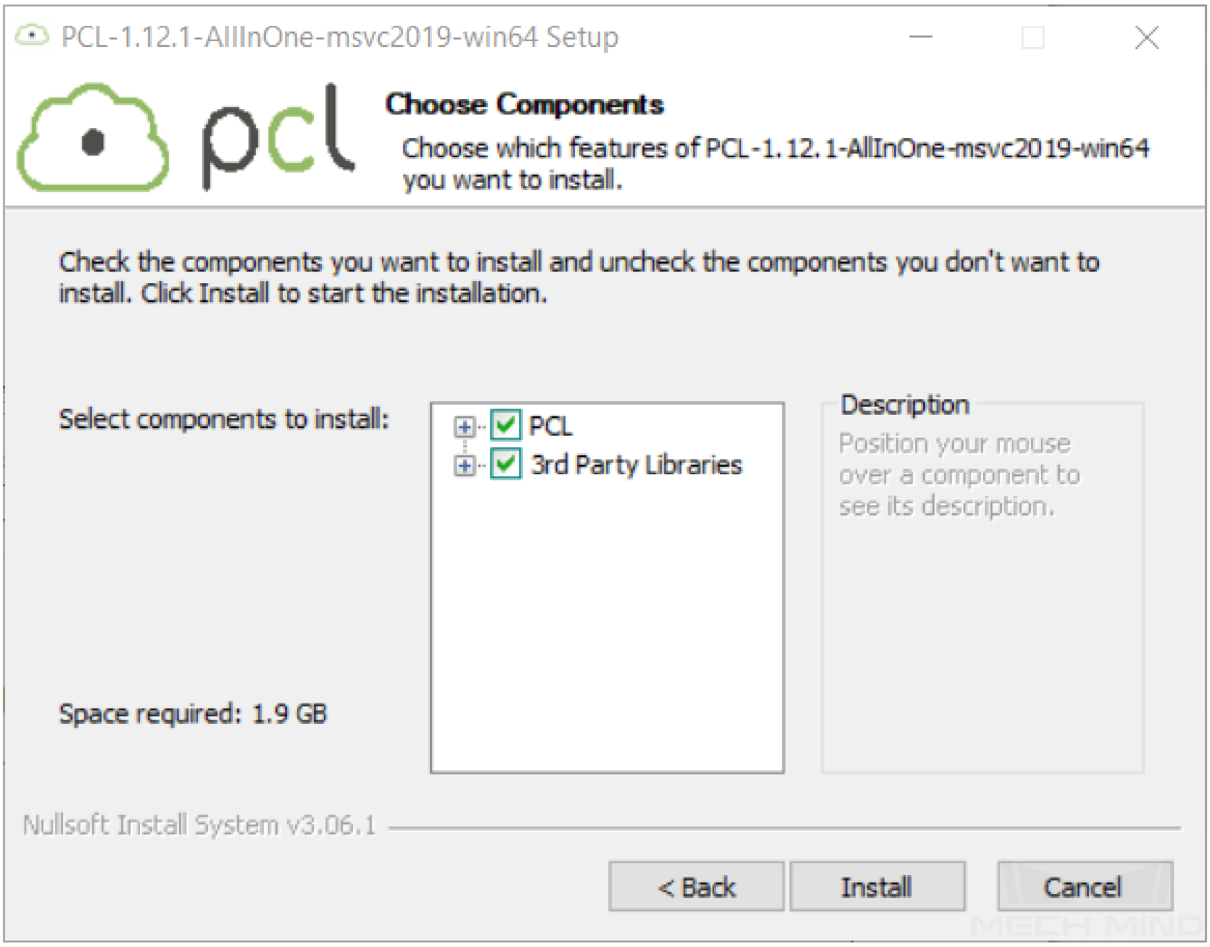
By default PCL 1.12.1 does not add its directory to the system PATH.

○ 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

5. On the **Choose Components** page, make sure that **3rd Party Libraries** is checked (it should be checked by default), and then click on *Install* to proceed.



PCL-1.12.1-AllInOne-msvc2019-win64 Setup

Choose Components

Choose which features of PCL-1.12.1-AllInOne-msvc2019-win64 you want to install.

Check the components you want to install and uncheck the components you don't want to install. Click Install to start the installation.

Select components to install:

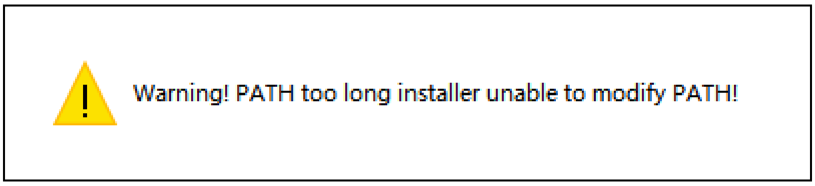
☑ PCL
☑ 3rd Party Libraries

Description
Position your mouse over a component to see its description.

Space required: 1.9 GB

Nullsoft Install System v3.06.1

Note: If the following error message pops up:



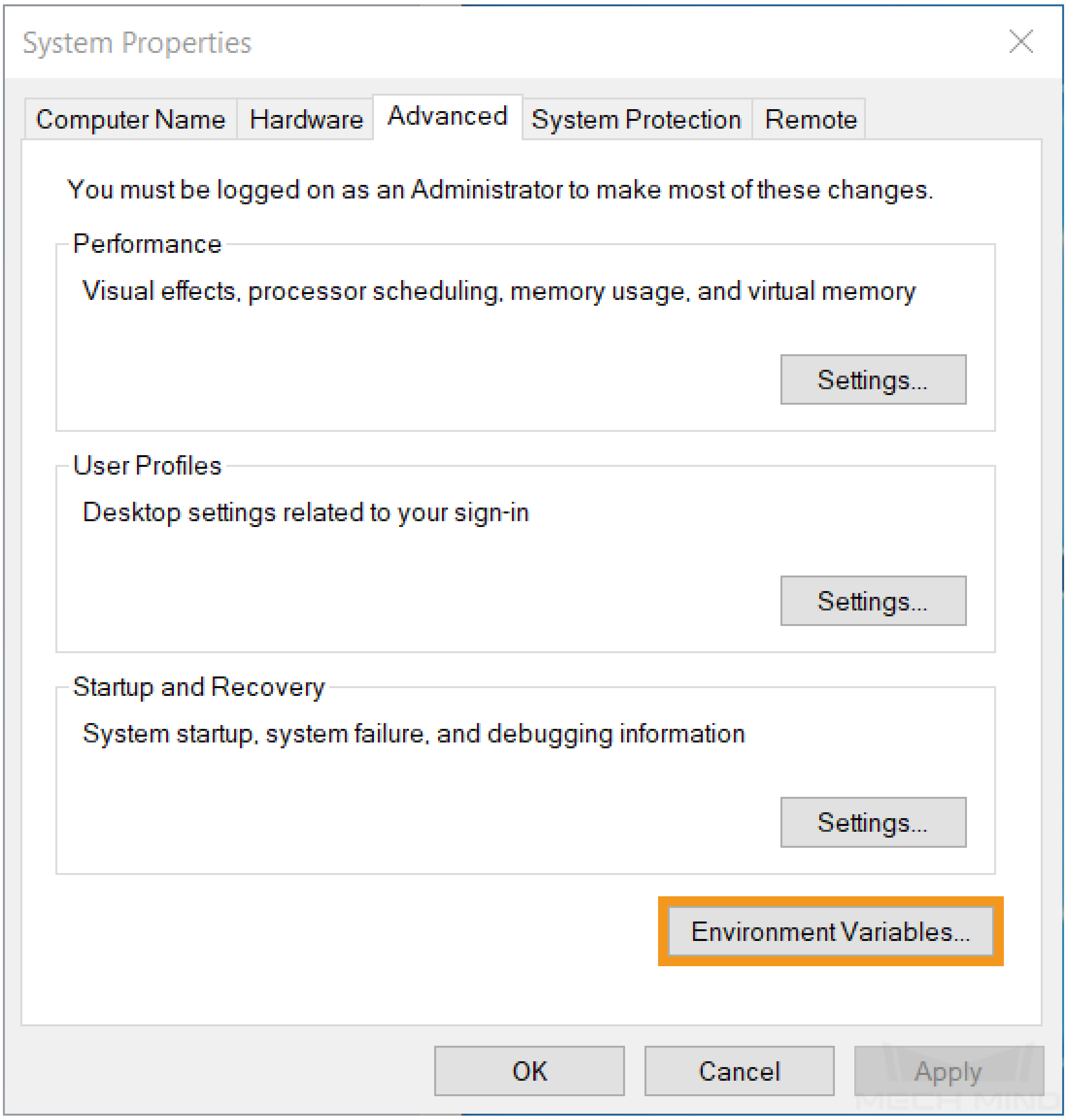
Warning! PATH too long installer unable to modify PATH!

Please add the following directories to **Path** manually. For instructions, please see Section 1.3.2.

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

Set Environment Variables

1. Right-click on **This PC** on the desktop, select **Properties**, and click on **Advanced system settings** in the pop-up window. In the popped up **System Properties** window, click on *Environment Variables*.



You must be logged on as an Administrator to make most of these changes.

Performance

Visual effects, processor scheduling, memory usage, and virtual memory

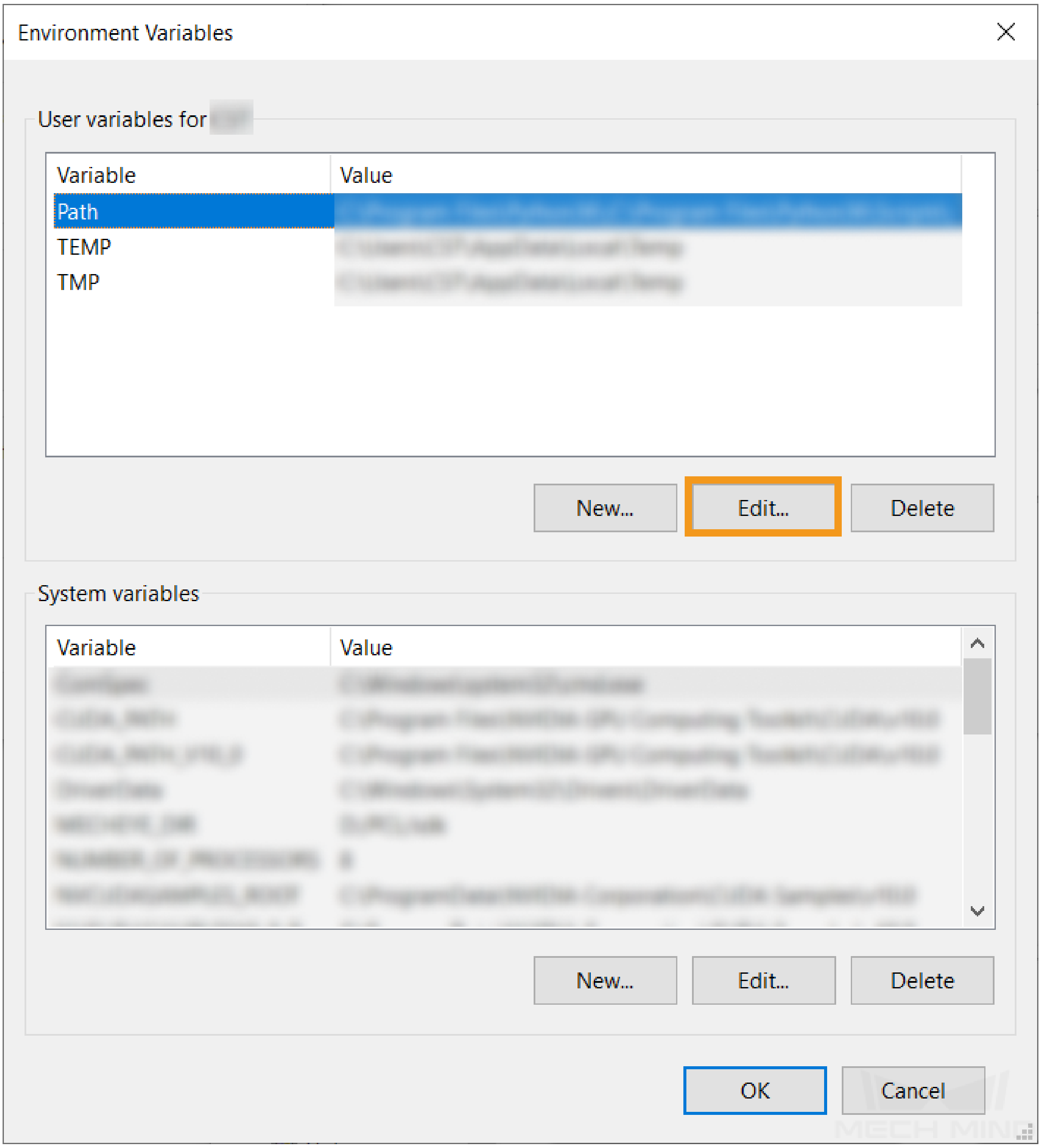
User Profiles

Desktop settings related to your sign-in

Startup and Recovery

System startup, system failure, and debugging information

2. Select **Path** in the user variables, and click on *Edit⋯* to enter the **Edit environment variable** page.



Environment Variables

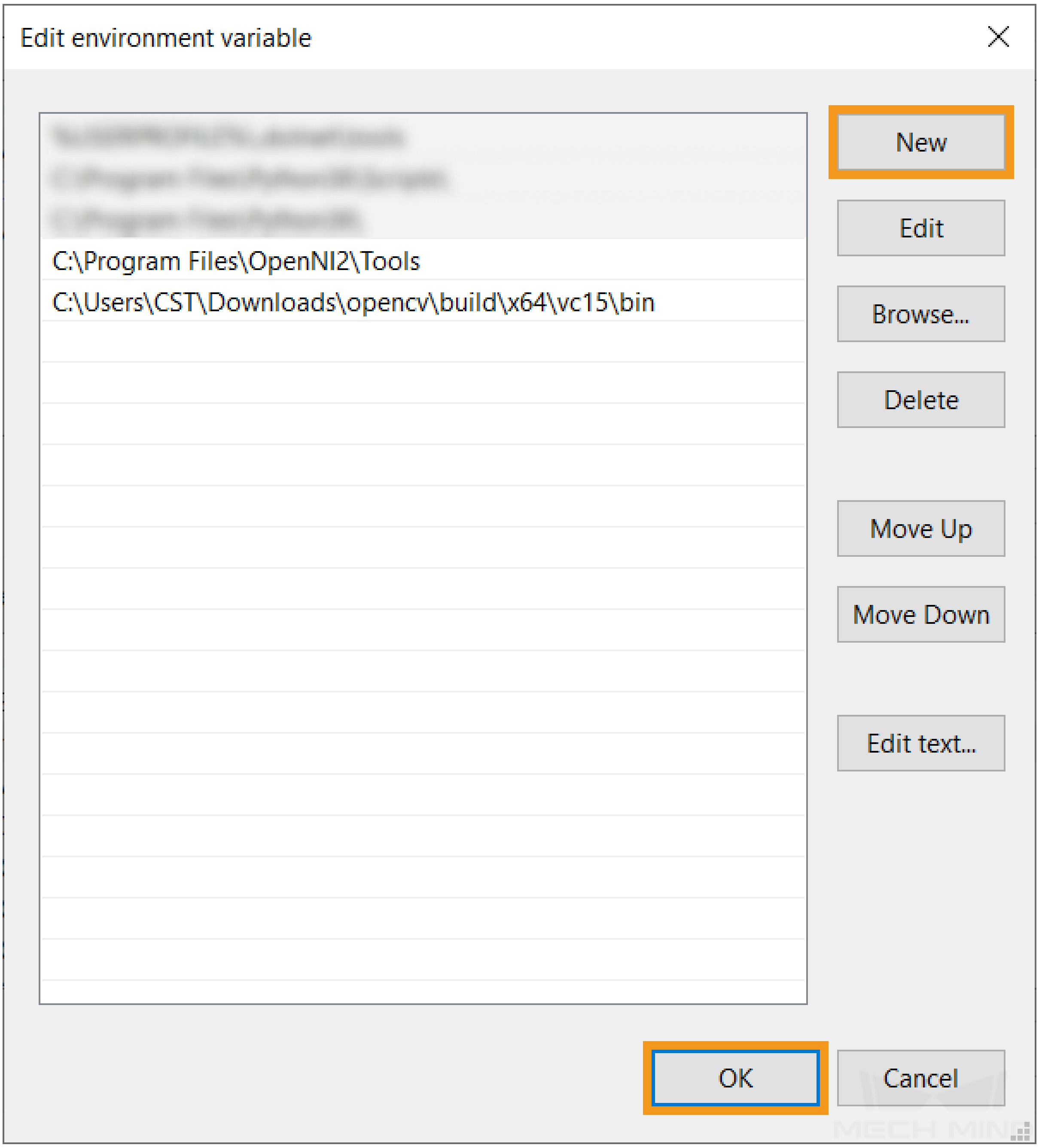
User variables for [redacted]

Variable	Value
Path	[redacted]
TEMP	[redacted]
TMP	[redacted]

System variables

Variable	Value
[redacted]	[redacted]
[redacted]	[redacted]
[redacted]	[redacted]
[redacted]	[redacted]
[redacted]	[redacted]
[redacted]	[redacted]
[redacted]	[redacted]

3. Click on *New* to add a new directory, and press the Enter key to confirm adding. Repeat until you finish adding all the directories below, and click on *OK* to exit the window.



Edit environment variable

[redacted]

C:\Program Files\OpenNI2\Tools

C:\Users\CST\Downloads\opencv\build\x64\vc15\bin

The following directories need to be added to **Path**:

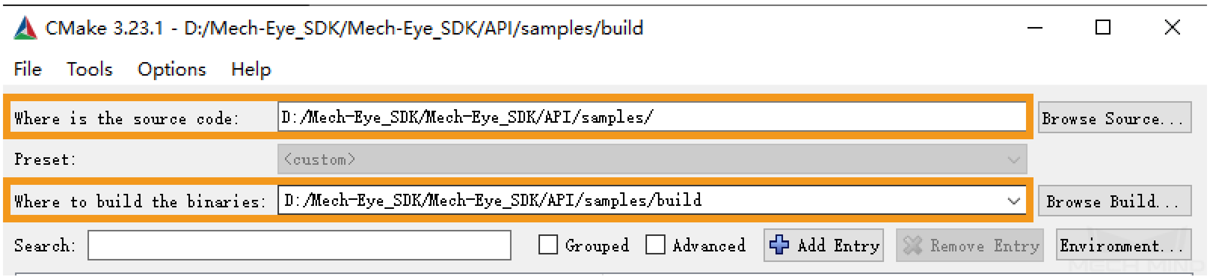
- *C:/Program Files/OpenNI/Tools*
- *xxx/opencv/build/x64/vc15/bin*

Configure Solution in CMake

1. Run **CMake (cmake-gui)**.

   Note: It is recommended to run CMake as Administrator.

2. Set the paths as follows:


CMake 3.23.1 - D:/Mech-Eye_SDK/Mech-Eye_SDK/API/samples/build

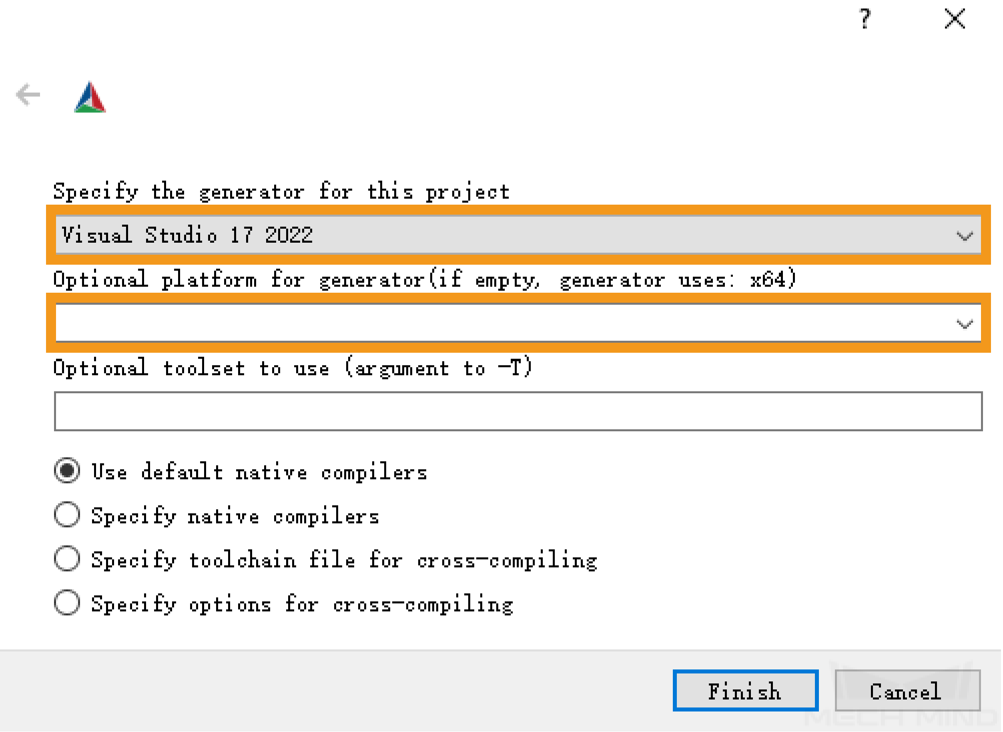
Where is the source code: D:/Mech-Eye_SDK/Mech-Eye_SDK/API/samples/

Preset: <custom>

Where to build the binaries: D:/Mech-Eye_SDK/Mech-Eye_SDK/API/samples/build

Where is the source code	xxx/Mech-Eye SDK/API/samples
Where to build the binaries	xxx/Mech-Eye SDK/API/samples/build

3. Click on *Configure* in the bottom, and set the following two fields according to the actual situation, and then click on *Finish*.


Specify the generator for this project: Visual Studio 17 2022

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

Optional toolset to use (argument to -T)

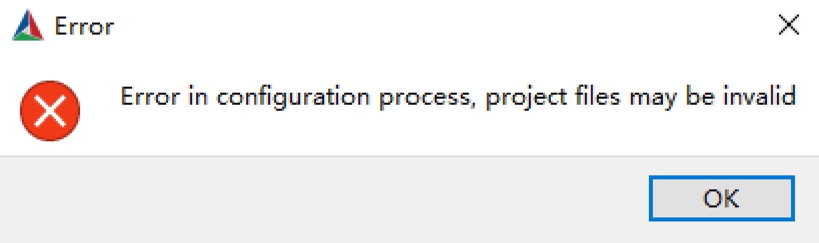
● Use default native compilers

○ Specify native compilers

○ Specify toolchain file for cross-compiling

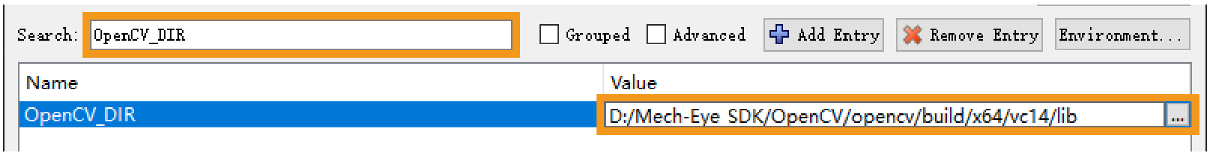
○ Specify options for cross-compiling

4. In the case where the following error message pops up:



Error in configuration process, project files may be invalid

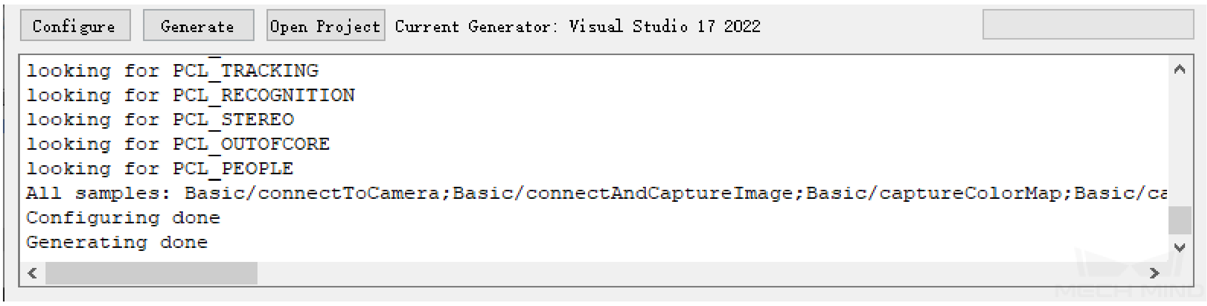
Please type *OpenCV_DIR* in the **Search** field, double-click on the directory and change it to *xxx/opencv/build/x64/vc14/lib*. Then, click on *Configure* again.



Search: OpenCV_DIR

Name	Value
OpenCV_DIR	D:/Mech-Eye SDK/OpenCV/opencv/build/x64/vc14/lib

5. When the log displays **Configuring done**, click on *Generate*. When the log displays **Generating done**, click on *Open Project*, and select Visual Studio in the pop-up window.

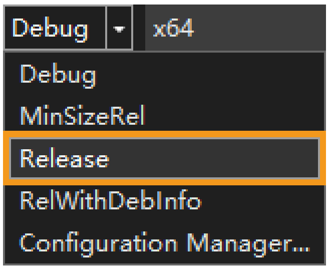


Current Generator: Visual Studio 17 2022

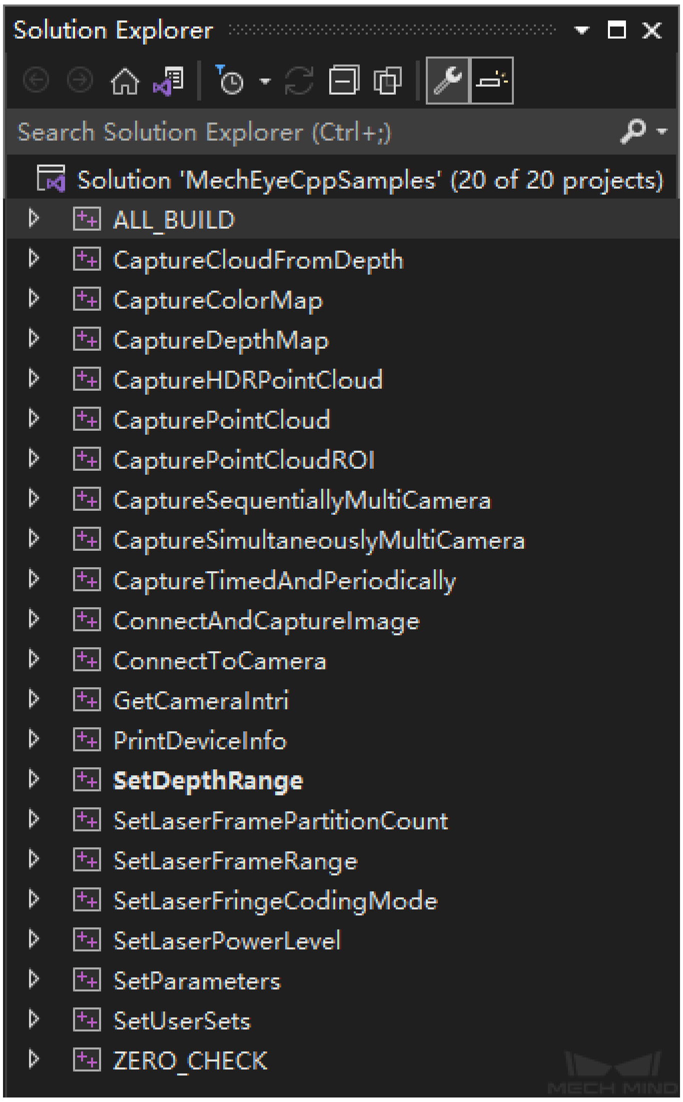
```
looking for PCL_TRACKING
looking for PCL_RECOGNITION
looking for PCL_STEREO
looking for PCL_OUTOFCORE
looking for PCL_PEOPLE
All samples: Basic/connectToCamera;Basic/connectAndCaptureImage;Basic/captureColorMap;Basic/ca
Configuring done
Generating done
```

Build Solution in Visual Studio

1. In the toolbar, click on **Debug** and select **Release**.

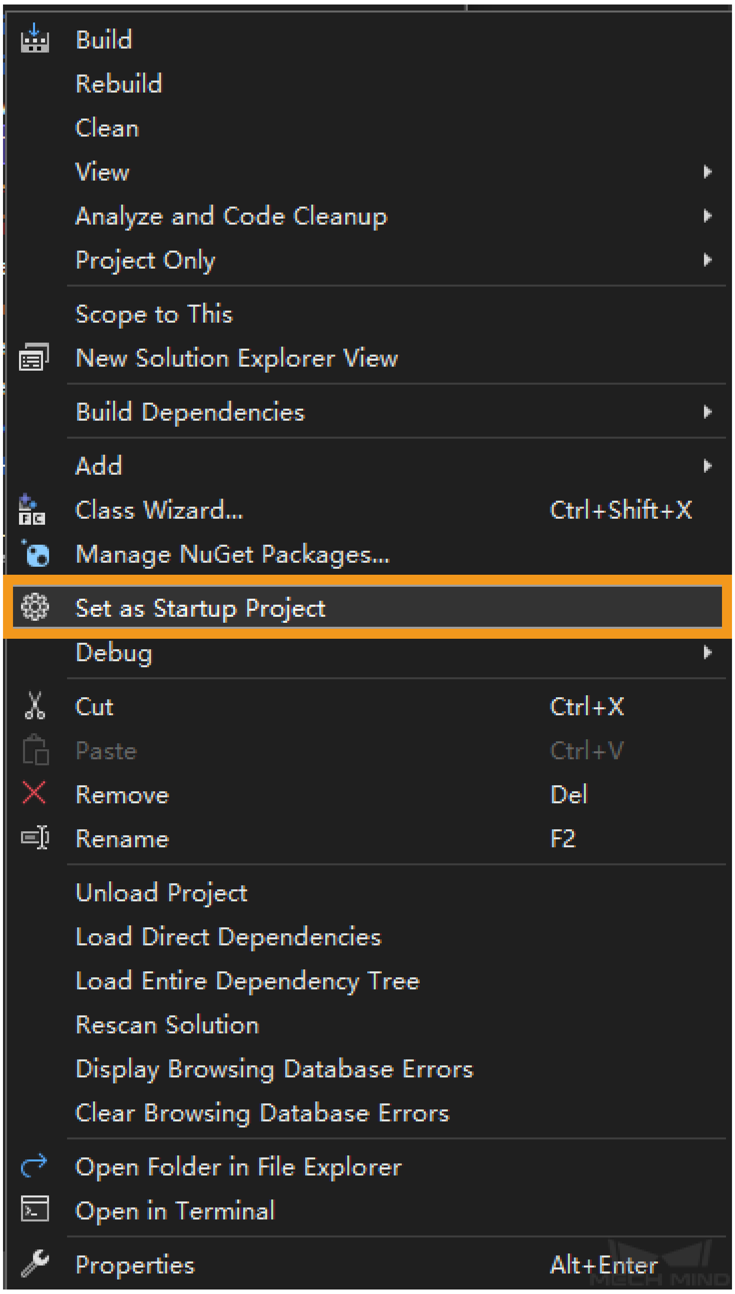


2. You can see all the available samples in **Solution Explorer**.



Solution 'MechEyeCppSamples' (20 of 20 projects)
- ALL_BUILD
- CaptureCloudFromDepth
- CaptureColorMap
- CaptureDepthMap
- CaptureHDRPointCloud
- CapturePointCloud
- CapturePointCloudROI
- CaptureSequentiallyMultiCamera
- CaptureSimultaneouslyMultiCamera
- CaptureTimedAndPeriodically
- ConnectAndCaptureImage
- ConnectToCamera
- GetCameraIntri
- PrintDeviceInfo
- **SetDepthRange**
- SetLaserFramePartitionCount
- SetLaserFrameRange
- SetLaserFringeCodingMode
- SetLaserPowerLevel
- SetParameters
- SetUserSets
- ZERO_CHECK

3. Right-click on a sample, and select **Set as Startup Project**.

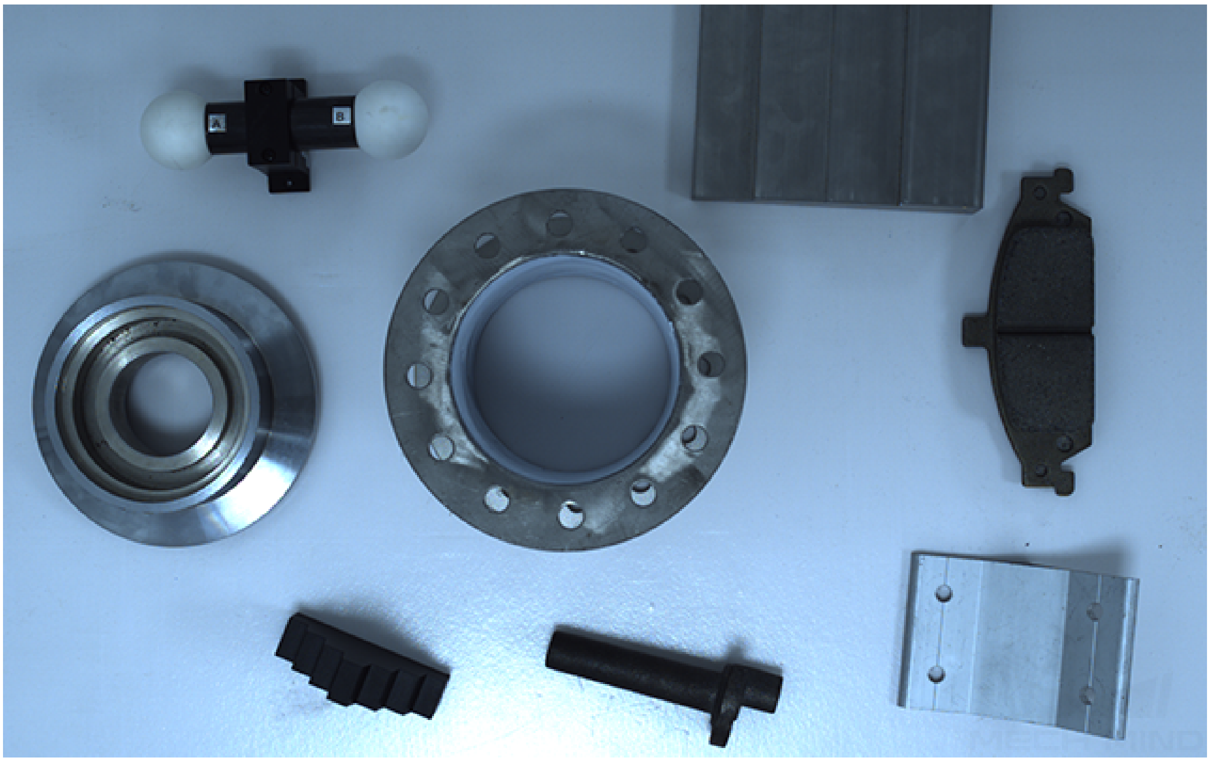


4. Click on ▶ Local Windows Debugger ▾ in the toolbar to build the solution.

Run Sample Program

Take **CaptureColorMap.exe** as an example.

1. Navigate to *xxx/Mech-Eye SDK/API/samples/build/Release*, and run **CaptureColorMap.exe**.
2. Input the index of the camera you want to connect according to the instructions.
3. When the program exits, you can find a **colorMap.png** (color image) under *xxx/Mech-Eye SDK/API/samples/build*.



3.3.2 C++ (Ubuntu)

This section will show you how to configure the Mech-Eye API C++ samples on Ubuntu.

Sample List

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

Basic samples: camera connection and basic capturing functions.
Advanced samples: advanced capturing functions.
Util samples: obtain information and set parameters.
Laser samples: for laser cameras only.

- Basic
  - ConnectToCamera : connect to a camera.
  - ConnectAndCaptureImage : connect to a camera can capture 2D images.
  - CaptureColorMap : capture color images from a camera.
  - CaptureDepthMap : capture depth maps from a camera.
  - CapturePointCloud : capture point clouds from a camera.
  - CaptureHDRPointCloud : capture point clouds in HDR mode from a camera.
  - CapturePointCloudROI : set an ROI and capture point clouds from a camera.
- Advanced
  - CaptureCloudFromDepth : generate point clouds from depth maps.
  - CaptureTimedAndPeriodically : capture images periodically for a specified length of time to ready the camera.
  - CaptureSimultaneouslyMultiCamera : capture images from multiple cameras simultaneously.
  - CaptureSequentiallyMultiCamera : capture images from multiple cameras sequentially.
- Util
  - GetCameraIntri : obtain and print the intrinsic parameters of the camera.
  - PrintDeviceInfo : print version information of Mech-Eye SDK and the camera firmware.
  - SetDepthRange : set the depth range of a camera.
  - SetUserSets : provide functions related to parameter groups, such as switching to a different parameter group, adding and deleting parameter groups, etc.
  - SetParameters : provide functions related to camera parameters, including all the functions of SetUserSets.
- Laser
  - SetLaserFramePartitionCount : set the laser scan partition number for a laser camera.
  - SetLaserFrameRange : set the field of view for a laser camera.
  - SetLaserFringeCodingMode : set the fringe coding mode for a laser camera.
  - SetLaserPowerLevel : set the power level for a laser camera.

Attention: The samples above have been included in the installation package of Mech-Eye SDK.

Requirements

1. Update the software source list and install the dependent libraries.

   ```
   sudo apt-get update
   sudo apt-get install -y build-essential pkg-config cmake
   ```

2. Install Mech-Eye SDK.

   Please refer to *Install Mech-Eye SDK on Ubuntu* for the installation method.

   After installation, the path to the samples folder is `/opt/mech-mind/mech-eye-sdk/samples/` .

3. Install third-party libraries.

   Samples marked with (OpenCV) or marked with (PCL) require third-party libraries to be installed.

   > **Attention:** If using a virtual machine to install the software, please reserve **more than 20G** of disk space, otherwise the installation may fail.

   ```
   # Install PCL
   sudo apt-get install libpcl-dev
   ```

   ```
   wget https://github.com/opencv/opencv/archive/3.4.5.tar.gz
   tar -zxvf 3.4.5.tar.gz
   cd opencv-3.4.5
   mkdir build
   cd build
   cmake -D CMAKE_BUILD_TYPE=RELEASE -D CMAKE_INSTALL_PREFIX=`pwd`/install ..
   make && make install
   pkg-config --modversion opencv
   ```

Instructions

1. Compile samples.

   You can compile all samples at a time, or compile a selected sample.

   > **Attention:** Every time you modify a sample’ s source file, please recompile the sample to update the executable

   - Compile all samples.

     a. Go to the folder `samples`.

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

b. Under the path `/opt/mech-mind/mech-eye-sdk/samples/`, open the file `CMakeLists.txt`, go to line 88 `set(OpenCV_DIR "/home/ubuntu/3rdParties/opencv-3.4.5/build")`, and replace the path in double quotes with your local OpenCV build path.

```
sudo vi CMakeLists.txt
```

c. Execute cmake and make in the current path.

```
sudo cmake .
sudo make
```

After successful compilation, the samples are located in `/opt/mech-mind/mech-eye-sdk/samples/`.

- Compile one sample.

a. Take CaptureSequentiallyMultiCamera as an example. Go to the folder of the sample to compile.

```
cd /opt/mech-mind/mech-eye-sdk/samples/Advanced/
↪CaptureSequentiallyMultiCamera/
```

b. This step is only required for samples using OpenCV. Under the path `/opt/mech-mind/mech-eye-sdk/samples/Advanced/CaptureSequentiallyMultiCamera/`, open the file `CMakeLists.txt`, go to line `set(OpenCV_DIR "/home/ubuntu/3rdParties/opencv-3.4.5/build")`. Replace the path in double quotes with your local OpenCV build path.

```
sudo vi CMakeLists.txt
```

c. Execute cmake and make in the current path.

```
sudo cmake .
sudo make
```

After successful compilation, the sample is located in `/opt/mech-mind/mech-eye-sdk/samples/Advanced/CaptureSequentiallyMultiCamera/` 。

2. Run a sample.

Taking ConnectToCamera as an example. Under the path `/opt/mech-mind/mech-eye-sdk/samples/`.

```
sudo ./ConnectToCamera
```

The output is as follows:

```
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:  V3.0.0
Firmware Version:  V1.5.2
```

(continues on next page)

(continued from previous page)

```
............................

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

During the running of the program, please follow the prompts to select the ID of the camera to be connected, and wait for the program to finish running.

Attention:

- For samples involving point cloud capturing, regular users do not have permissions to write point cloud files, so sudo command should be used to execute the sample programs as the administrator.
- For samples involving image or point cloud capturing, after the running is completed, the image or point cloud files are by default stored at `/opt/mech-mind/mech-eye-sdk/samples/`.

Tip: In the folder of each sample, the file `ReadMe.txt` provides a brief description of the sample.

3.3.3 C#

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

Sample List

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

Basic samples: camera connection and basic capturing functions.
Advanced samples: advanced capturing functions.
Util samples: obtain information and set parameters.
Laser samples: for laser cameras only.

Basic

- ConnectToCamera: connect to a camera.
- ConnectAndCaptureImage: connect to a camera and capture color images and depth maps.
- CaptureColorMap (EmguCV): capture color images from a camera.
- CaptureDepthMap (EmguCV): capture depth maps from a camera.
- CapturePointCloud (EmguCV): capture point clouds from a camera.

- CaptureHDRPointCloud (EmguCV): capture point clouds in HDR mode from a camera.
- CapturePointCloudROI (EmguCV): set an ROI and capture point clouds from a camera.

Advanced

- CaptureCloudFromDepth (EmguCV): generate point clouds from depth maps and color images.
- CaptureTimedAndPeriodically (EmguCV): capture images periodically for a specified length of time.
- CaptureSimultaneouslyMultiCamera (EmguCV): capture images from multiple cameras simultaneously.
- CaptureSequentiallyMultiCamera (EmguCV): capture images from multiple cameras sequentially.

Util

- GetCameraIntri: obtain and print the intrinsic parameters of the camera.
- PrintDeviceInfo: print version information of Mech-Eye SDK and the camera firmware.
- SetDepthRange: set the depth range of a camera.
- SetUserSets: provide functions related to parameter groups, such as switching to a different parameter group, adding and deleting parameter groups, etc.
- SetParameters: provide functions related to camera parameters, including all the functions of SetUserSets.

Laser

- SetLaserFramePartitionCount: set the laser scan partition number for a laser camera.
- SetLaserFrameRange: set the field of view for a laser camera.
- SetLaserFringeCodingMode: set the fringe coding mode for a laser camera.
- SetLaserPowerLevel: set the power level for a laser camera.

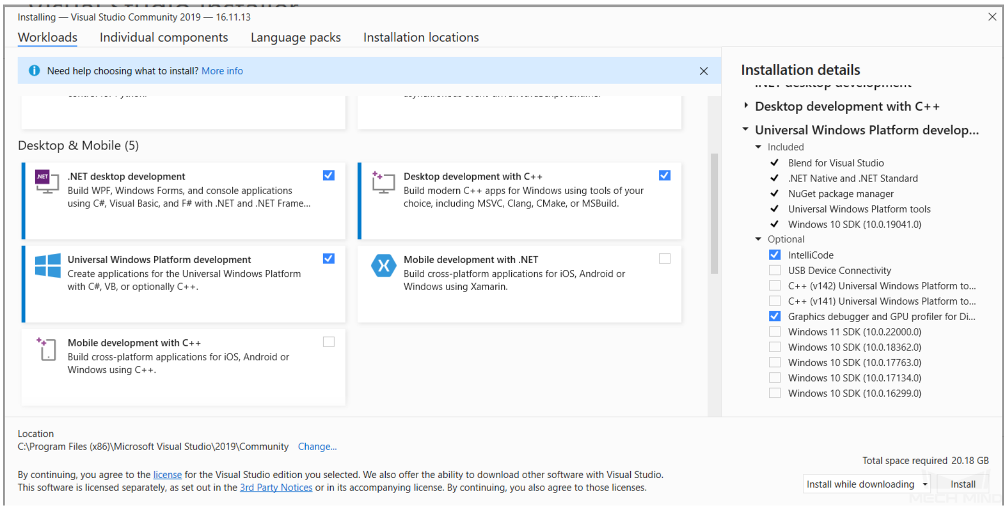
Requirements

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

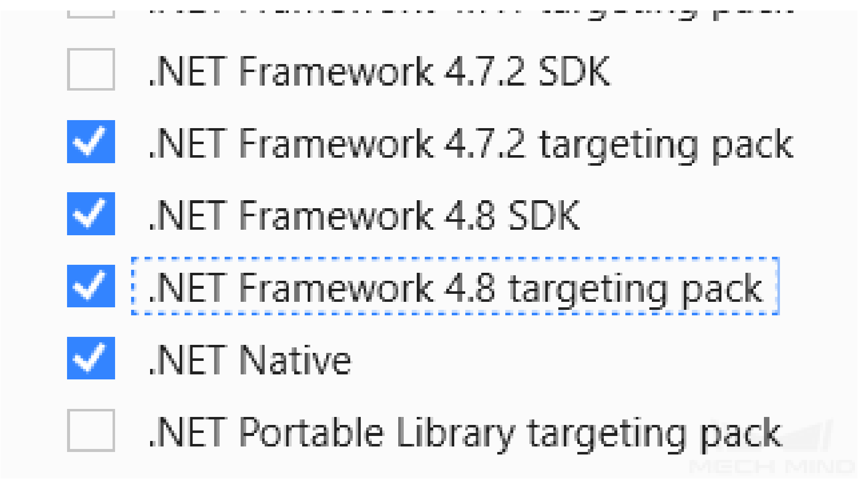
- Desktop development with C++
- Universal Windows Platform develop...
  - Included
    - ✓ Blend for Visual Studio
    - ✓ .NET Native and .NET Standard
    - ✓ NuGet package manager
    - ✓ Universal Windows Platform tools
    - ✓ Windows 10 SDK (10.0.19041.0)
  - Optional
    - ☑ IntelliCode
    - □ 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



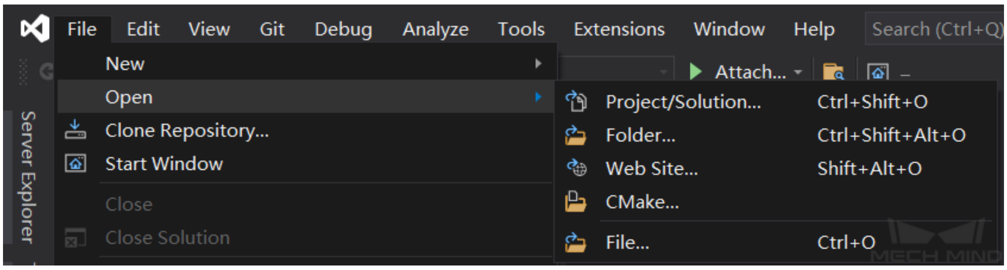
□ .NET Framework 4.7.2 SDK
☑ .NET Framework 4.7.2 targeting pack
☑ .NET Framework 4.8 SDK
☑ .NET Framework 4.8 targeting pack
☑ .NET Native
□ .NET Portable Library targeting pack

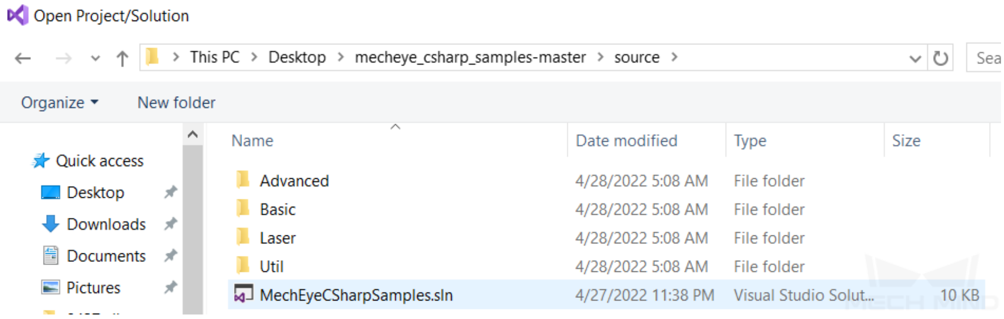
3. *Download and install Mech-Eye SDK*
4. Download or clone C# samples
5. *Connect to a real camera that functions properly*
6. The samples marked with (EmguCV) must be installed via NuGet Package Manager Emgu.CV.runtime.windows. Please refer to Install and manage packages in Visual Studio using the NuGet Package Manager for detailed instructions.

Instructions

Build Solution in Visual Studio

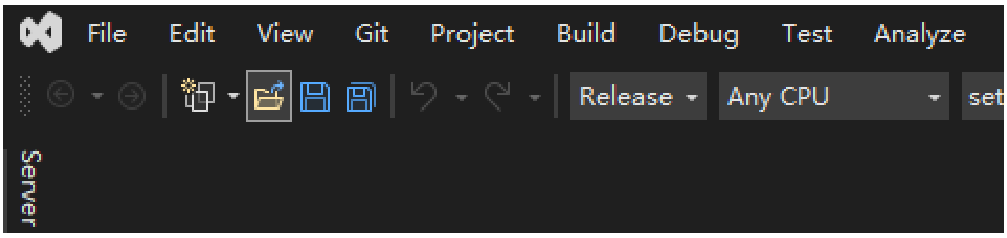
1. Open MechEyeCSharpSamples.sln in Visual Studio.





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

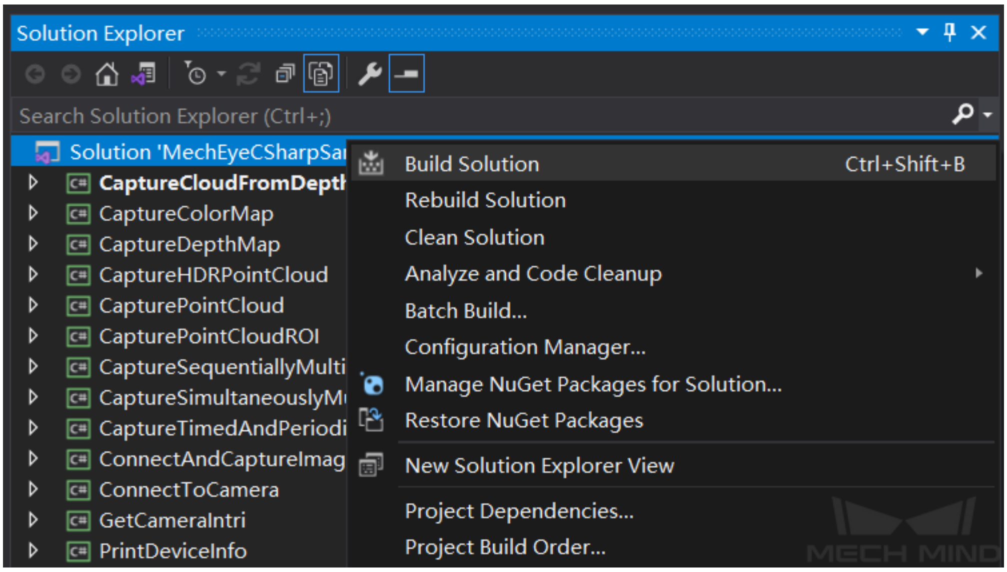
2. Change the Solution Configuration from **Debug** to **Release**.



Release
Any CPU

3. Build solution.

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



Solution Explorer

Search Solution Explorer (Ctrl+;)

Solution 'MechEyeCSharpSa
- CaptureCloudFromDepth
- CaptureColorMap
- CaptureDepthMap
- CaptureHDRPointCloud
- CapturePointCloud
- CapturePointCloudROI
- CaptureSequentiallyMulti
- CaptureSimultaneouslyMu
- CaptureTimedAndPeriodi
- ConnectAndCaptureImag
- ConnectToCamera
- GetCameraIntri
- PrintDeviceInfo

Run Sample Program

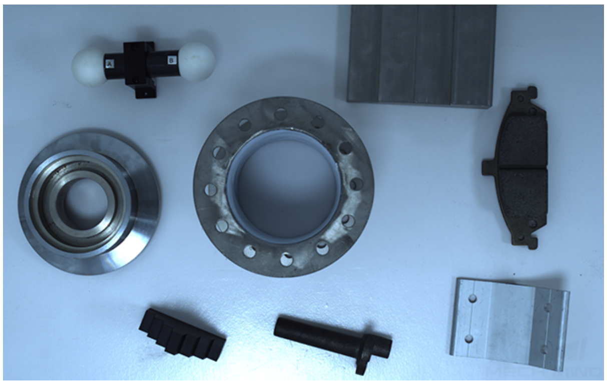
The example below shows how to run CaptureColorMap.exe.

1. Go to *mecheye_csharp_samples-master* → *source* → *Build*.

2. Double-click on 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.



3.3.4 Python

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

Sample List

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

Basic samples: camera connection and basic capturing functions.
Advanced samples: advanced capturing functions.
Util samples: obtain information and set parameters.
Laser samples: for laser cameras only.

Basic

- ConnectToCamera: connect to a camera.
- ConnectAndCaptureImage: connect to a camera and capture color images and depth maps.
- CaptureColorMap: capture color images from a camera.
- CaptureDepthMap: capture depth maps from a camera.
- CapturePointCloud: capture point clouds from a camera.
- CaptureHDRPointCloud: capture point clouds in HDR mode from a camera.
- CapturePointCloudROI: set an ROI and capture point clouds from a camera.

Advanced

- CaptureCloudFromDepth: generate point clouds from depth maps and color images.
- CaptureTimedAndPeriodically: capture images periodically for a specified length of time.
- CaptureSimultaneouslyMultiCamera: capture images from multiple cameras simultaneously.
- CaptureSequentiallyMultiCamera: capture images from multiple cameras sequentially.

Util

- GetCameraIntri: obtain and print the intrinsic parameters of the camera.
- PrintDeviceInfo: print version information of Mech-Eye SDK and the camera firmware.
- SetDepthRange: set the depth range of a camera.
- SetUserSets: provide functions related to parameter groups, such as switching to a different parameter group, adding and deleting parameter groups, etc.
- SetParameters: provide functions related to camera parameters, including all the functions of SetUserSets.

Laser

- SetLaserFramePartitionCount: set the laser scan partition number for a laser camera.

- SetLaserFrameRange: set the field of view for a laser camera.
- SetLaserFringeCodingMode: set the fringe coding mode for a laser camera.
- SetLaserPowerLevel: set the power level for a laser camera.

Requirements

1. *Download and install Mech-Eye SDK*
2. Download or clone Python samples
3. Make sure that the Python you installed is version 3.6.8 or above
4. *Connect to a real camera that functions properly*

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.

- Compulsory: MechEyeAPI 。

  ```
  pip install MechEyeAPI
  ```

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

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

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

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

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

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

Run Sample Program

The example below shows how to run CaptureColorMap.py.

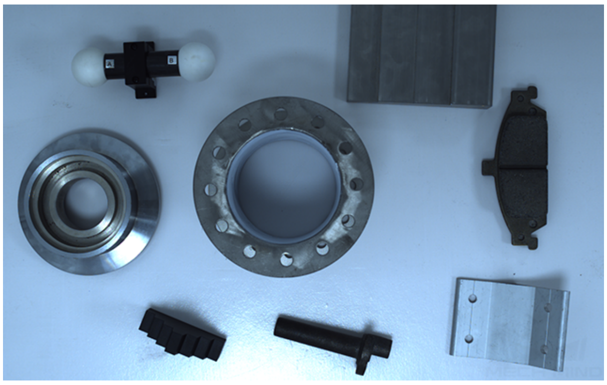
1. Navigate to the directory where CaptureColorMap.py is located.

```
cd  XXX\mecheye_python_samples/source/Basic
```

2. Run the sample.

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

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



3.3.5 ROS

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

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.

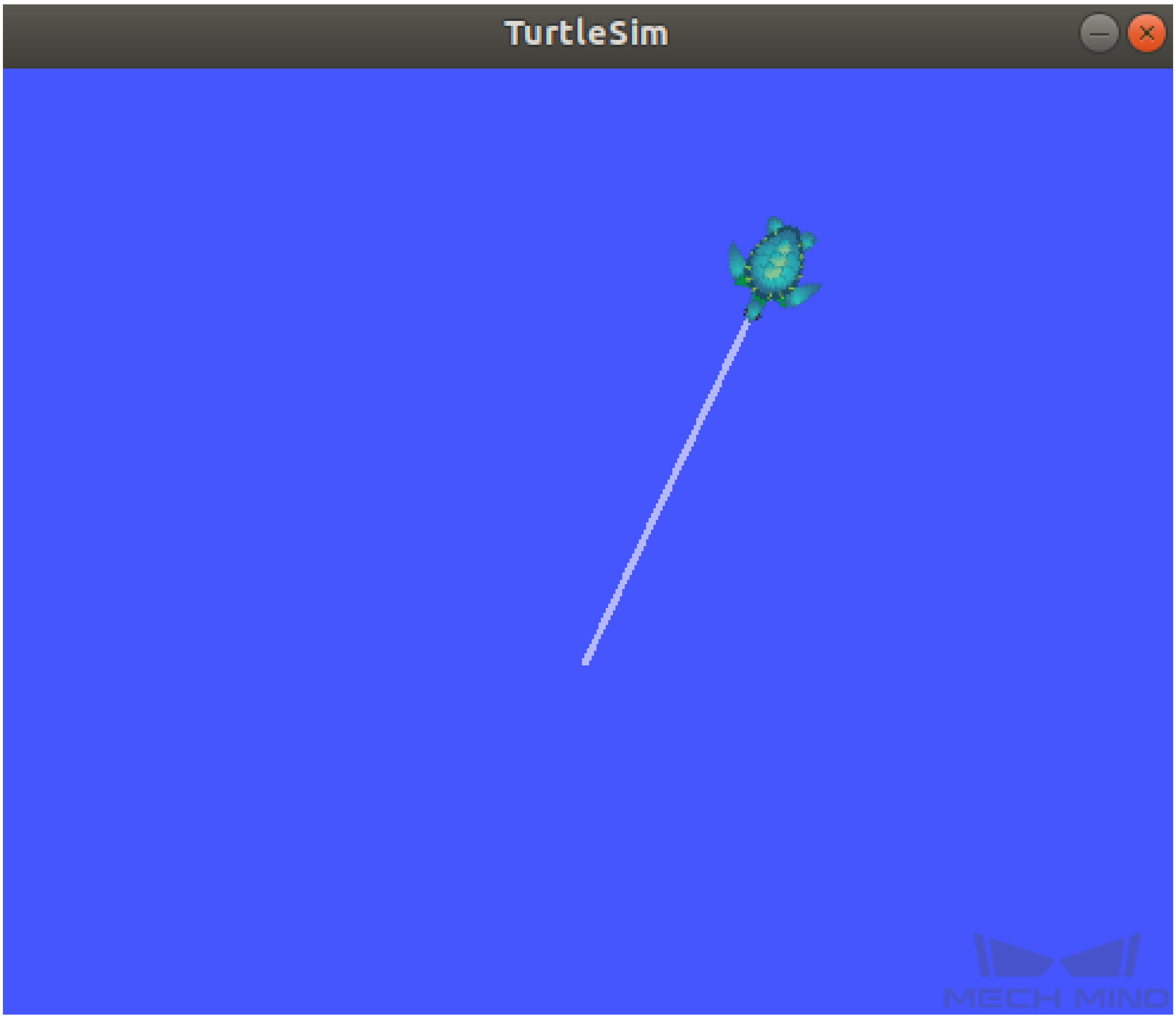
```
# Open a terminal in any directory and enter the command:
roscore
# Open a second terminal and enter the command:
```

(continues on next page)

(continued from previous page)

```
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.



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:  V4.0.0
Firmware Version:  V1.5.2
............................
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:  V3.0.0
Firmware Version:  V1.5.2
............................
```

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

```
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.

Check the following section to learn about **Mech-Eye Viewer**

3.4 Mech-Eye Viewer User Manual

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

On 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.

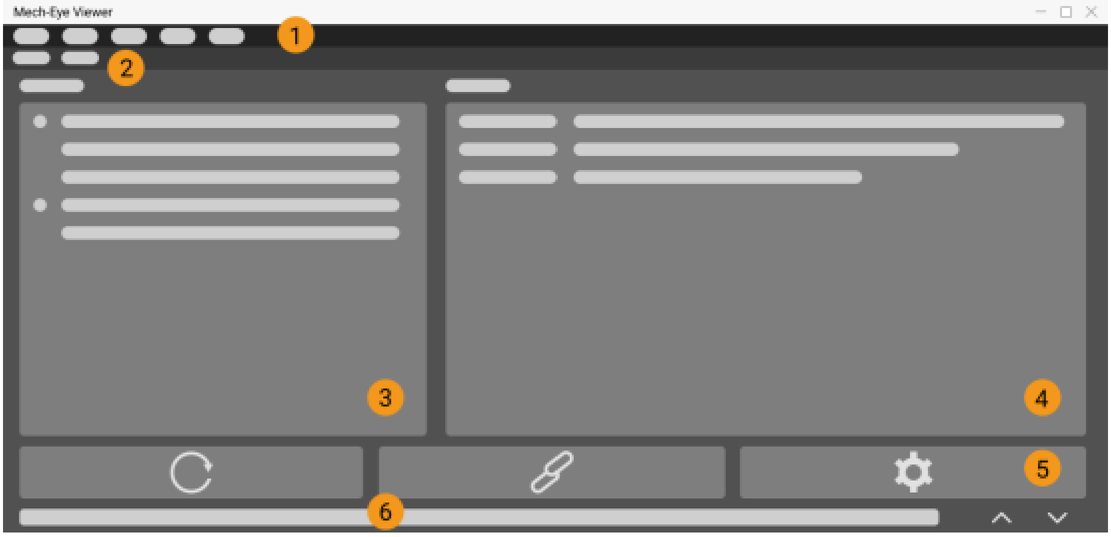
3.4.1 Interface

This section introduces the interface of the Mech-Eye Viewer and includes the instructions on using the Device Finder and the Camera Viewer.

Device Finder

After opening Mech-Eye Viewer, the software shows the device finder interface, as in the figure below.

This interface is for connecting the camera to Mech-Eye Viewer. In this interface, you can view the list of connectable cameras and their information, connect a camera, and set the camera IP.



Device Finder’ s Interface

Device Finder’ s Interface

No.	Component	Use
1	Menubar	Switch software languages, switch users, view version information, and start virtual cameras.
2	Switching tabs	Switch between device finder and camera viewer.
3	Detected cameras list	View all available cameras and their model info in the current network.
4	Camera info	View the model, IP address, port, serial number, firmware version of the selected camera.
5	Toolbar	Refresh camera list, connect camera and set camera IP.
6	Log	View the status of the client program over time by scrolling up and down.

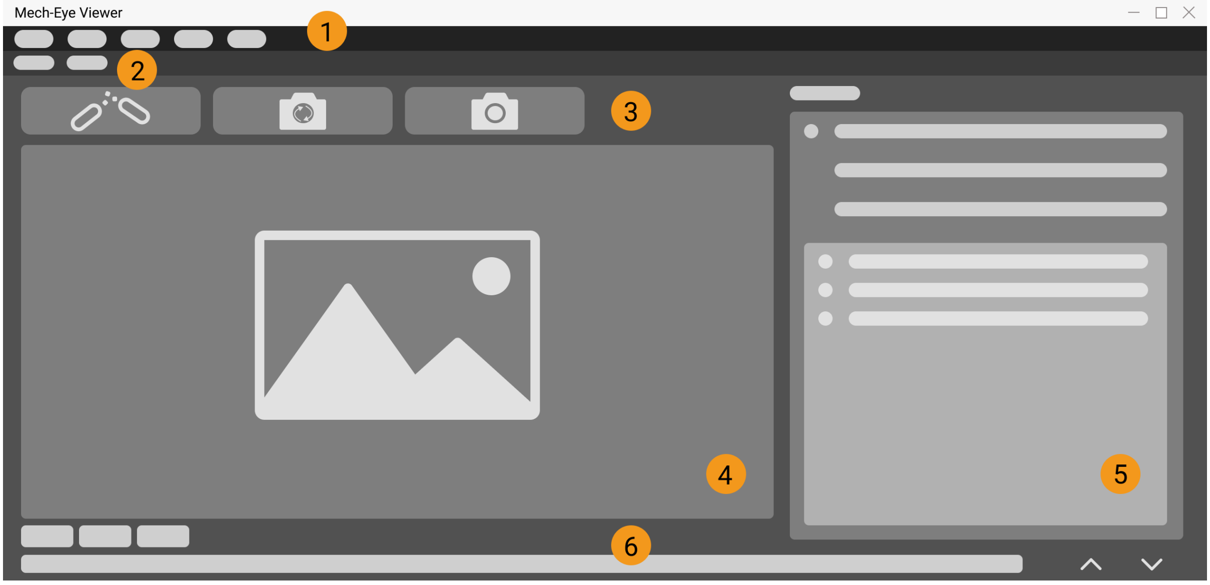
Device Finder’ s Toolbar

Feature	Note
Refresh camera list	If the camera is powered on and connected to the current network, but does not appear in the camera list, please refresh the camera list. After refreshing, the camera will appear in the list of detected cameras.
Connect a camera	Select a camera, and click to connect the camera.
Set camera IP	For checking the machine’ s network connection and setting the camera’ s IP address.

Camera Viewer

After the camera is successfully connected, the software will display the camera viewer interface, as shown in the figure below.

In this interface, you can capture images, switch the displayed data type, and adjust image and data information by adjusting parameters.



Camera Viewer’ s Interface

The components and their uses are listed below.

No.	Component	Use
1	*Camera Viewer’ s Menubar*	Save image, switch software language, switch user, view version information, start virtual camera, open image analysis tools.
2	Switching tabs	Switch between device finder and camera viewer.
3	*Camera Viewer’ s Toolbar*	Capture images or disconnect the camera.
4	*Camera Viewer’ s Image Display*	View images captured by the camera, and switch between displayed data types: 2D image, depth map, point cloud.
5	*Parameters*	View and adjust image parameters, and save parameter groups for different purposes.
6	Log	View the status of the client program over time by scrolling up and down.

Camera Viewer’ s Menubar

Option		Description
File	Save Raw Data of Camera	Save the original data and generate a .mraw file for debugging and analysis later.
	Save Images	Save different types of images, including 2D images, depth maps, and point clouds.
	Start Virtual Camera	Adjust the saved data after loading the data from a virtual camera.
Tools	Check Camera Intrinsic Parameters	Check whether the camera’ s intrinsic parameters are consistent with the factory settings.
	Exposure Assistant	Obtain the optimal exposure parameter values of the camera by trying different exposure time combinations.
	Depth Analyzer	Check the imaging quality of the selected region by checking the fluctuation of the depth map in the selected area.
	View 2D Camera and Set Parameters	Set and view 2D camera related parameters.
Advanced	User	Set to Standard by default. Other options are not available.
	Camera Controller	View camera type, date and time, CPU temperature and DLP temperature, etc.
View	Log Operation	Off by default. Open/close the log management panel.
	Property (Parameter)	On by default. Open/close the property (parameter) panel.
	2D View	On by default. Open/close the 2D image display.
	Depth View	On by default. Open/close the depth map display.
	Cloud View	On by default. Open/close the point cloud display.
Help	Language	Switch the software language. Restart the software to apply.
	About	View software version.

Camera Viewer’ s Toolbar

Icon	Name	Use
🔗	Disconnect	Disconnect the currently connected camera.
📷	Continuous Scan	Let the camera continuously capture images and view the images, depth maps, and point clouds.
📷	Scan Once	Let the camera takes a single image and view the image, depth map, and point cloud.

Camera Viewer′ s Image Display

Through **View** in the menubar, you can select the type of data to be displayed, and you can switch the display data type in the lower-left corner.

Option	**Description**
2D image	View 2D images captured by the camera.
Depth map	View depth maps captured by the camera.
Point cloud	View point clouds captured by the camera.

Parameters

Visibility

Visibility options include **Beginner** and **Expert**. More parameters are available for adjustment in **Expert** mode. It is recommended to choose the **Beginner** mode first; if the quality of the 2D image, depth map, or point cloud is not satisfactory, please choose the **Expert** mode.

Config group

By default, parameters of the software are all expanded and all parameters that can be adjusted are displayed.

Icon	Use
default ▾	Options: default, calib (calibration).
	Click the small triangle on the right side of the icon to view all parameter groups, please select a parameter group based on your needs.
☰	Expand all parameters.
☰	Collapse all parameters.
+	Add a new parameter group.
−	Delete the current parameter group.
✎	Rename the parameter group.
Save	Save the parameter set. Shortcut: Ctrl S.
Reset	Reset the parameter group (restore the factory settings).
Import	Import parameter groups.
Export	Export parameter groups.

Note:

- *default* is a default parameter group and cannot be deleted.
- It is recommended to create corresponding parameter groups for different projects.

Property (Parameter) Panel

The panel consists of two parts: **parameter name** and **value**. It is mainly used to view parameter definitions, set regions of interest, and adjust parameters for 2D images, depth maps and point clouds. For details, please see *Parameter Adjustment*.

- **Parameter name**:

  Click on the **parameter name** to view its definition. Double-click on a parameter category (2D parameters, 3D parameters, point cloud processing, etc.) to expand/collapse all parameters under a category.

- **Value**:

  For setting the region of interest or adjusting the camera settings for obtaining the optimal 2D images, depth maps, or point clouds.

3.4.2 Capture Images with Mech-Eye Viewer

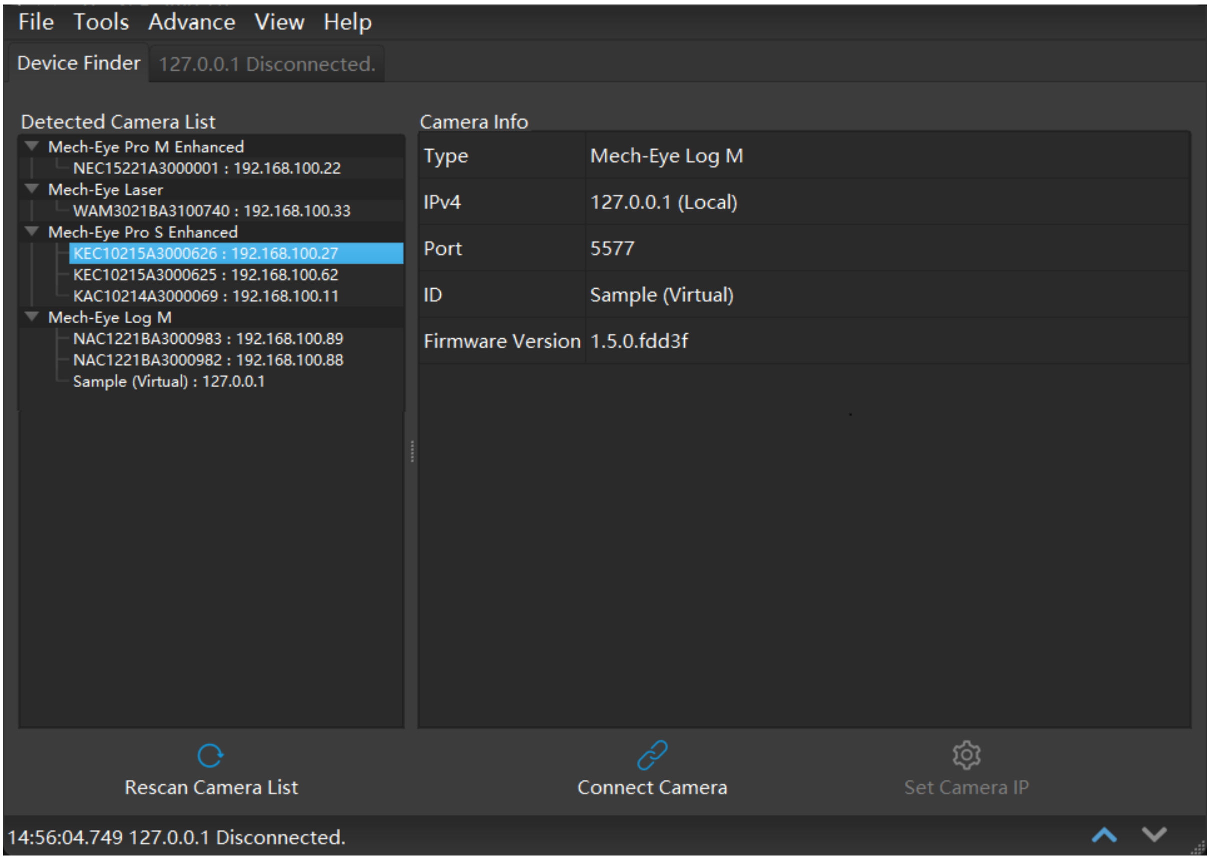
This section includes instructions on using Mech-Eye Viewer to capture 2D images, depth maps, and point clouds, including instructions on connecting the camera, collecting data, adjusting parameters, and saving the data.

Connect a Camera

After opening Mech-Eye Viewer, in the Device Finder interface, the available cameras are shown in the **Detected Camera List**.

You can connect a camera by:

- Double-clicking on the camera to connect, or
- Clicking on *Connect Camera* after selecting the camera to connect.



Detected Camera List

- Mech-Eye Pro M Enhanced
  - NEC15221A3000001 : 192.168.100.22
- Mech-Eye Laser
  - WAM3021BA3100740 : 192.168.100.33
- Mech-Eye Pro S Enhanced
  - KEC10215A3000626 : 192.168.100.27
  - KEC10215A3000625 : 192.168.100.62
  - KAC10214A3000069 : 192.168.100.11
- Mech-Eye Log M
  - NAC1221BA3000983 : 192.168.100.89
  - NAC1221BA3000982 : 192.168.100.88
  - Sample (Virtual) : 127.0.0.1

Camera Info

Type	Mech-Eye Log M
IPv4	127.0.0.1 (Local)
Port	5577
ID	Sample (Virtual)
Firmware Version	1.5.0.fdd3f

14:56:04.749 127.0.0.1 Disconnected.

Device Finder’ s Interface

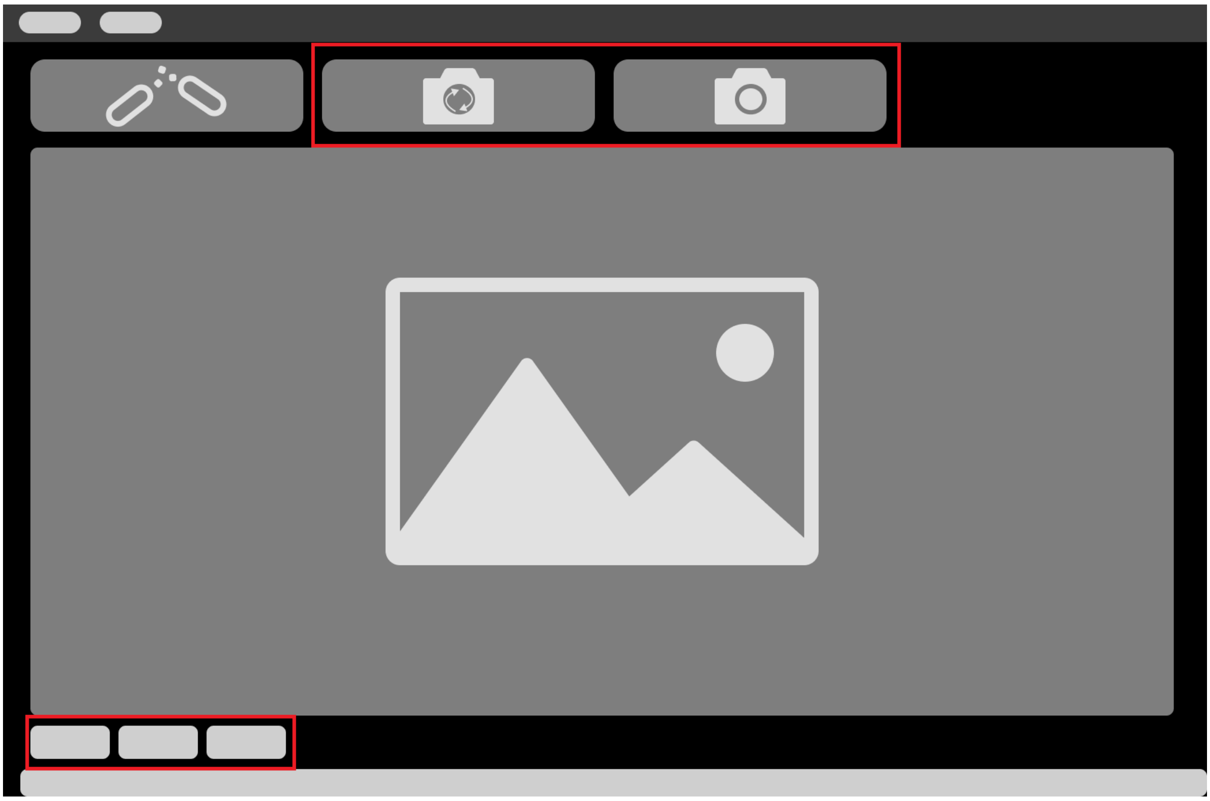
Attention:

- The camera and the IPC must be in the same network segment before they can be used together.
  - If you use an automatically assigned IP, you do not need to manually set the camera’ s IP and the IPC’ s IP.
  - If you use a fixed IP, please consult the network administrator before filling in the IP. Please see *IP Settings of Camera and IPC* for details.
- The camera’ s firmware version must be consistent with Mech-Eye Viewer’ s version.
- If you already have image data and do not need to capture images from a camera, you can open the image data by starting a virtual camera. However, with a virtual camera, you cannot adjust exposure parameters.

Image Capturing and Data Types

Image Capturing

After the camera is successfully connected, you can start capturing images by clicking on one of the two scan buttons in the camera viewer interface, as shown in the figure below.



Scan buttons

The two buttons in the interface correspond to two modes of scanning.

Mode	Description
Continuous Scan	Capture images at regular time intervals. The 2D image, depth map, point cloud reflect the status of the scene almost in real-time.
Scan Once	Capture image once.

Data Types

After the camera started capturing images, you can view the captured object information in the following data types. You can switch between the data types by clicking on the corresponding tabs at the bottom of the interface.

Data type	Description
2D Image	2D monochrome/color image (depending on whether the camera is a color camera)
Depth map	2D map containing information about the distances from the camera to points on the object surfaces. The value of each pixel is the actual distance the sensor measured.
Point cloud	The collection of points in the 3D space that reflect object surface features

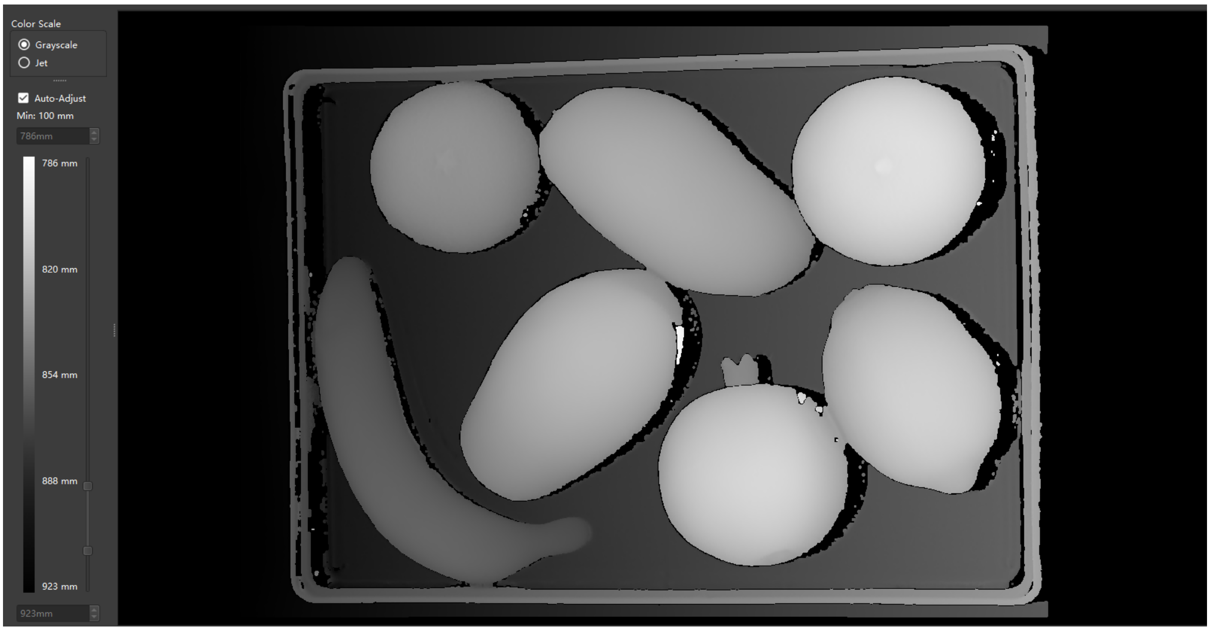
2D Image

Below is an example of a 2D image. You can scroll to zoom in/out.

2D image example

Depth Map

Below is an example of a depth map. You can set the display color under **Color Scale** and scroll to zoom in/out.



Color Scale

● Grayscale

○ Jet

☑ Auto-Adjust

Min: 100 mm

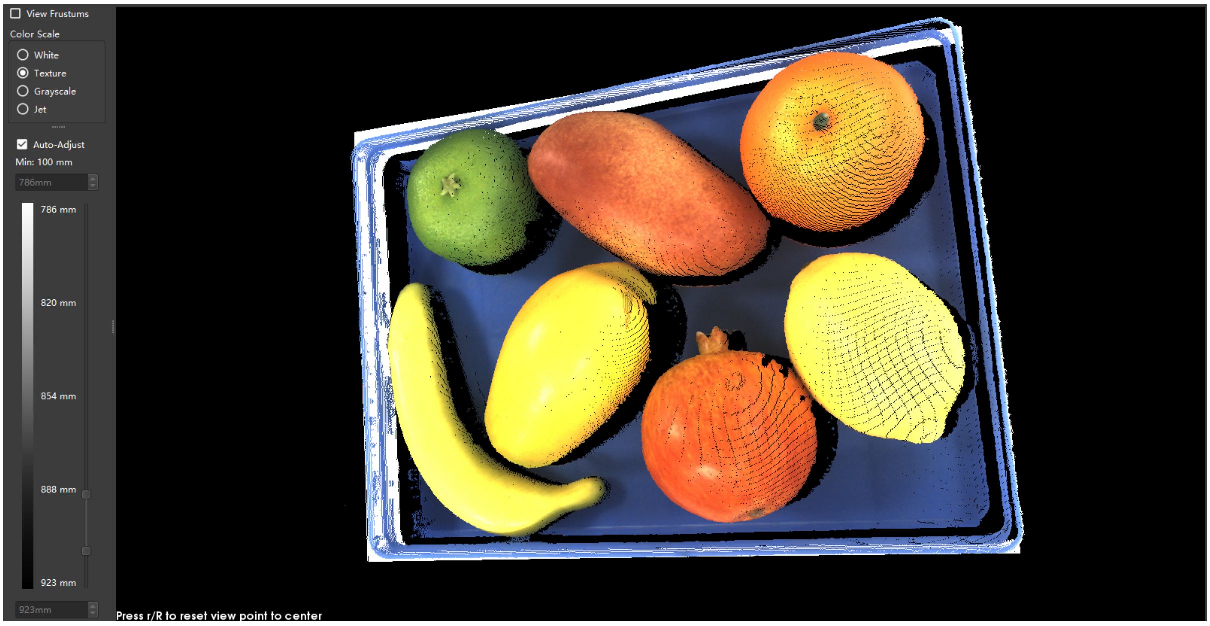
786mm

923mm

Depth map example

Point Cloud

Below is an example of a point cloud. You can set the display color under **Color Scale**, scroll to zoom in/out, and hold and drag to rotate the point cloud.



□ View Frustums

Color Scale

○ White

● Texture

○ Grayscale

○ Jet

☑ Auto-Adjust

Min: 100 mm

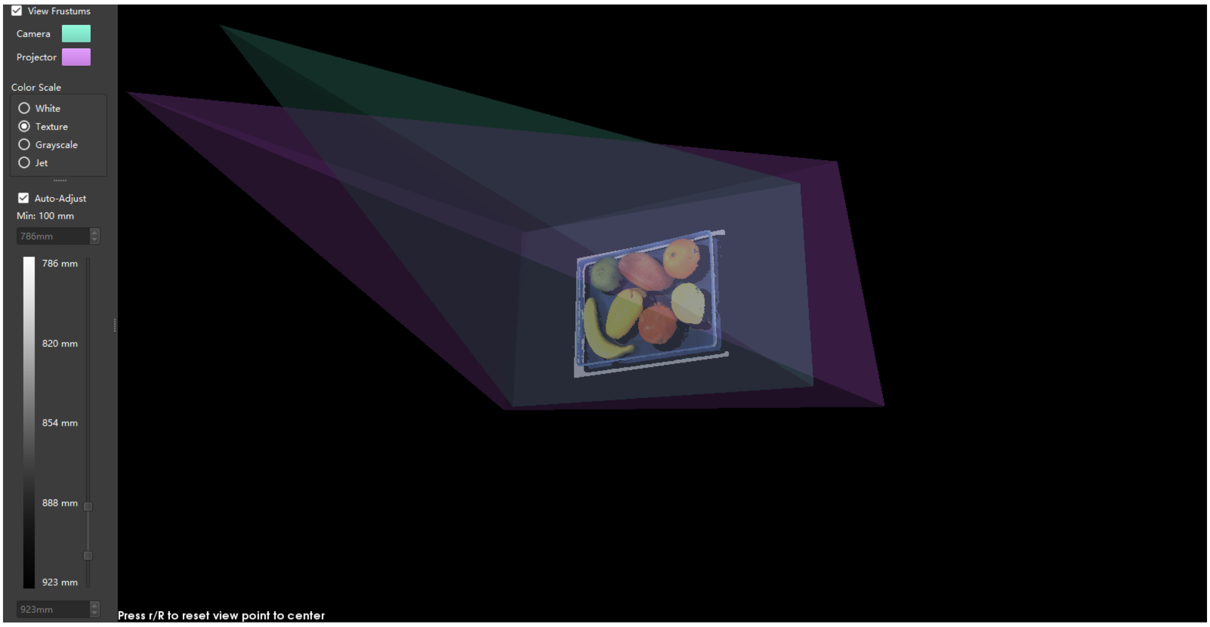
786mm

923mm

Press r/R to reset view point to center

Point cloud example

You can view the **view frustums** by checking the option on the upper left corner when viewing the point cloud.



☑ View Frustums

Camera 🟩

Projector 🟪

Color Scale

○ White
● Texture
○ Grayscale
○ Jet

☑ Auto-Adjust

Min: 100 mm

786mm

923mm

Press r/R to reset view point to center

View Frustums

Parameter Adjustment

If the captured image or point cloud is not as expected, please adjust the **parameters** to obtain the 2D images, depth maps, and point clouds that meet your needs.

Parameter Adjustment for 2D Image

2D Scanning parameters are for the 2D image.

Usually, the parameters available under the **Beginner** mode will suffice.

If after tuning the parameters, the effect is still not satisfactory, please switch to the **Expert** mode.

Exposure Mode

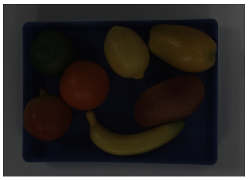
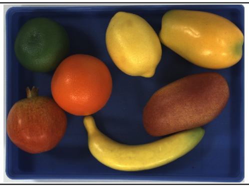
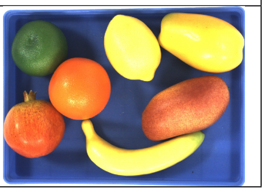
Exposure mode	Applicable scenarios
Timed	Stable light conditions.
Auto	Changing light conditions.
HDR	Objects with varying surfaces, i.e., dark or bright colors, uneven light distribution, etc. that require superimposing images of different exposures to expand the dynamic range and enhance details.
Flash	Objects that need additional light.

Timed

- **Exposure Time:**

  This parameter affects the image brightness. The longer the exposure time, the brighter the image is.

  2D images given different lengths of exposure time:

5 ms	20 ms	40 ms
		

Auto

- **Auto Function ROI**:

  This parameter is an entry for setting the ROI where the image processing algorithm of the expected gray value applies.

  Please click on *Edit* to open the ROI setting window.

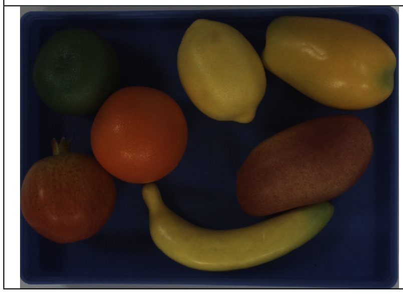
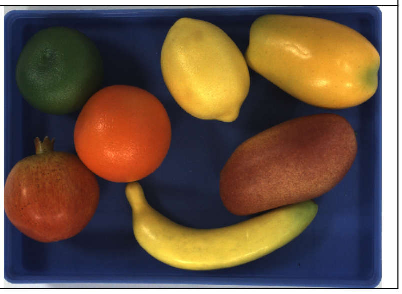
- **Expected Gray Value**:

  For monochrome cameras, the expected gray value affects brightness. For color cameras, the expected gray value affects the brightness of each color channel.

  The higher the expected gray value, the brighter the image looks.

  Default: 100
  Range: 0 t0 255

  2D images under different expected gray values:

100	180
	

220	240

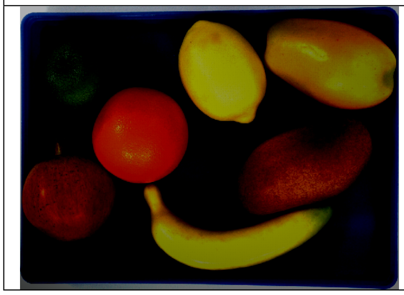
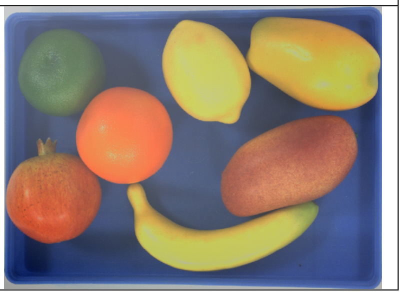
HDR

- **Tone Mapping**

  Tone mapping is for better displaying the 2D image' s details by enhancing the colors. Please set this parameter to *True* to enable this function.

  Default: *False*

  2D image without tone mapping (left) and 2D image with tone mapping (right):

False	True
	

- **HDR Exposure Sequence**

  This parameter is an entry for setting the HDR exposure time sequence. Please click on *Edit* to open the setting window.

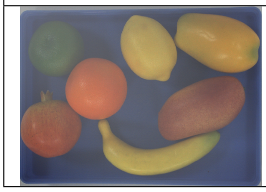
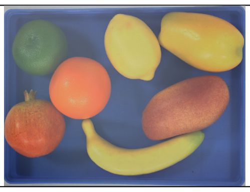
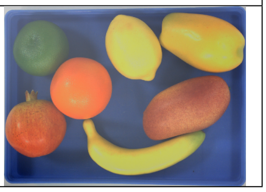
  In the window, you can add the HDR exposure time sequence and set the duration for each exposure.

  Please click on *Apply* to finish the setting.

  2D images of different exposure durations given **one exposure**:

10 ms	30 ms	50 ms

2D images of different exposure durations given **one exposure** and **multiple exposures**:

10 ms	30 ms + 10 ms	30 ms
		

- **Auto Function ROI**:

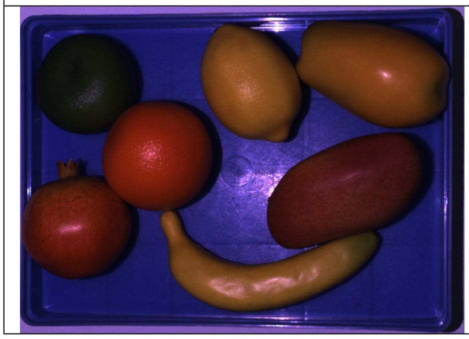
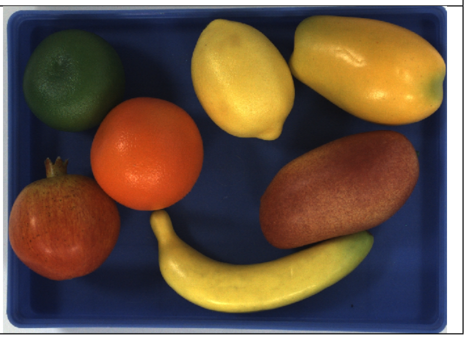
  This parameter is an entry for setting the ROI where the HRD algorithm applies.

  Please click on *Edit* to open the ROI setting window.

Flash

You can use this option to add additional light on the objects before capturing the image.

The contrast between using flash and not using flash:

Flash	No Flash
	

Sharpen Factor

Sharpening is for obtaining clearer object edge details, but it may cause more image noise.

This parameter sets the degree of sharpening on the image.

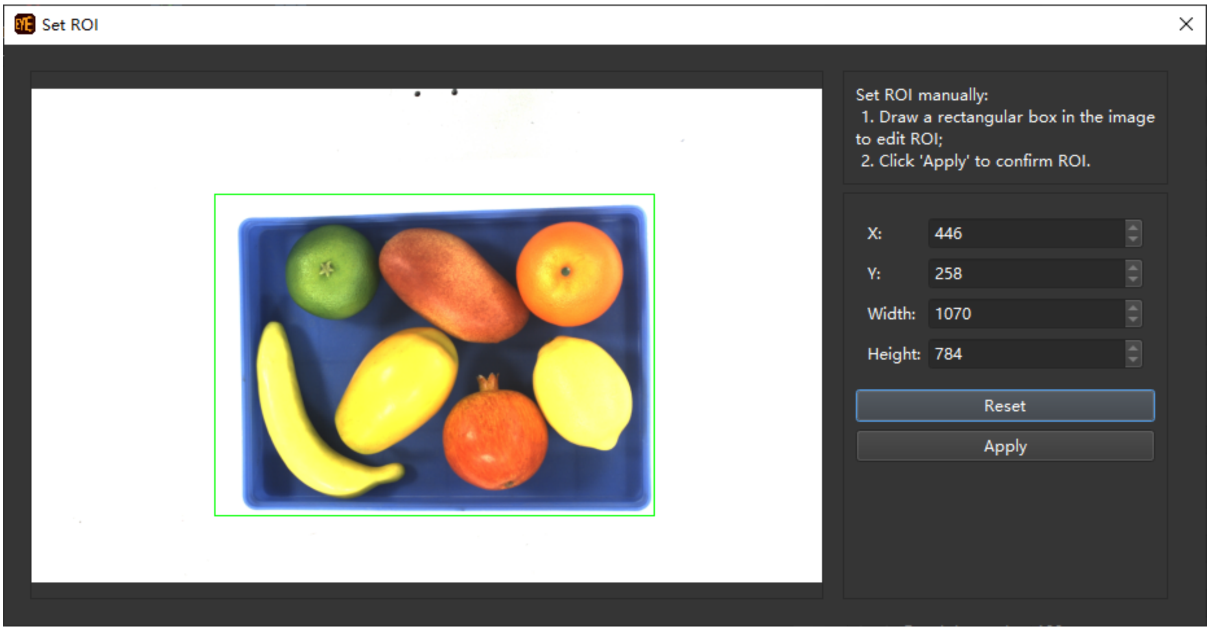
The higher the value, the intenser the sharpening, the heavier the noise.

ROI

This parameter is an entry for setting the 2D ROI of the image and thus ignoring the unwanted parts.

Click on *Edit* to open the ROI setting window.

Click on *Apply* to finish the setting.



Set ROI

Set ROI manually:
1. Draw a rectangular box in the image to edit ROI;
2. Click 'Apply' to confirm ROI.

X: 446
Y: 258
Width: 1070
Height: 784

White Balance

The object color in the image may not be the same as that of the real object because of the ambient light.

Please adjust the white balance to make the 2D image reflect the real object colors.

See *View 2D Camera and Set Parameters* for details.

Hint: This parameter is only applicable on color cameras.

Parameter Adjustment for Depth Map

3D Scanning

Scan Multiplier

Scan multiplier sets the number of exposure times.

Multiple exposures are for superimposing images of different exposure effects and thus obtaining the optimal image.

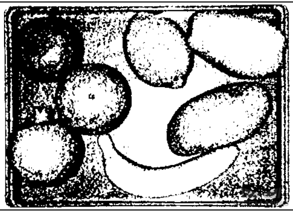
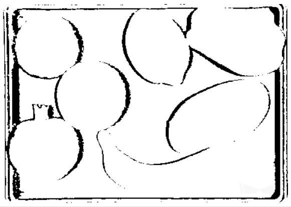
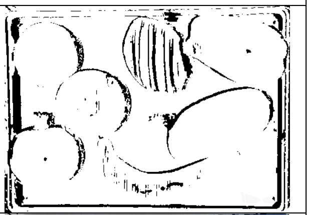
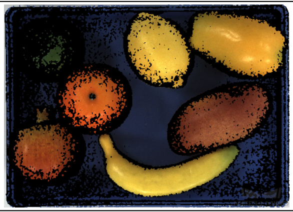
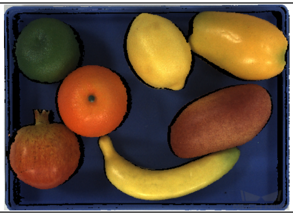
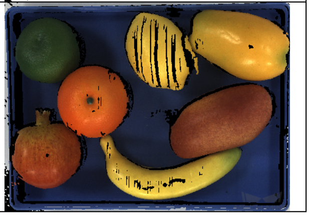
Exposure Time

Exposure time sets the duration of each exposure. If you set the scam multiplier to be greater than 1, please set the exposure times individually.

If the image is too dark, please increase the exposure time.

If the image is too bright, please decrease the exposure time.

Images obtained with different **exposure time** settings:

0.25 ms	1 ms	9 ms
		
		

Please see *Exposure Assistant* to obtain the optimal exposure parameter settings.

Gain

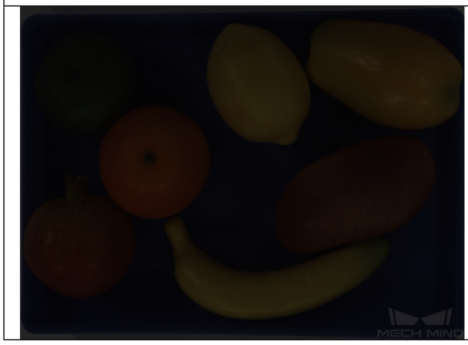
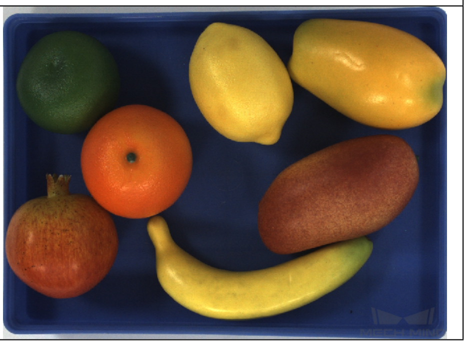
You can increase the gain to brighten the image.

When increasing the exposure time can lead to an ideal degree of brightness, it is recommended to increase the gain. However, Increasing the gain may introduce image noise.

Default: 0 dB

Range: 0 to 16 dB

Images obtained with different gain values:

0 dB	16 dB
	

Hint: If the exposure time is too long, the time cost for image capturing will be higher, so it is recommended to increase the exposure time and the gain together to increase the brightness so that the exposure time will not be too long.

Laser

Hint: Laser parameters are only for laser cameras.

Coding Quality:

- **Fast**: Fast caputuring. Low image quality.
- **High**: Slow capturing. High image quality.

Laser Gray Power:

This parameter sets the projection intensity of the laser. The larger the value, the stronger the intensity; the smaller the value, the weaker the intensity.

The default value of 100% works for most regular cases.

For objects with special surface conditions, it is recommended to apply high intensity on dark objects and low intensity on reflective objects.

Default: 100%
Range: 0 to 100%

Experimental Settings

Hint: Experimental settings are only for DLP cameras.

Projector

Projector Light Brightness:

- **High**: High brightness. For dark objects.
- **Normal**: Normal brightness. For objects of regular surfaces.
- **Low**: Low brightness. For reflective objects.

Depth Range

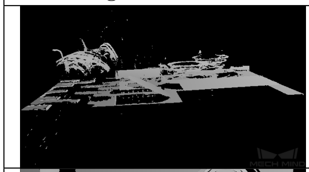
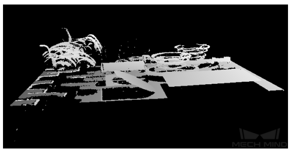
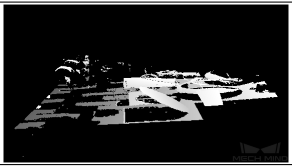
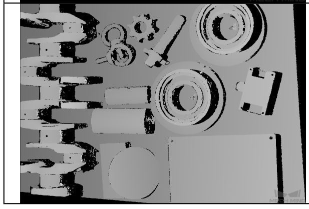
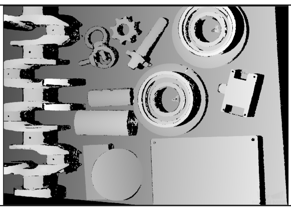
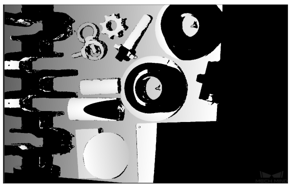
This parameter sets the region of interest on the vertical direction.

Range: 1 to 9,999 mm

Please set the depth range based on dimensions and the positions of the objects to ensure there are no missing parts.

The range should not be too large, otherwise, it may introduce interference; nor should it be too small, otherwise key parts may be missing.

Below are the examples of different depth ranges:

Too large	Moderate	Too small
		
		

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

Parameter Adjustment for Point Cloud

3D Scanning

Tuning parameters under **3D Scanning** can optimize the point cloud quality by optimizing the exposure. Please see *3D Scanning* for details.

Point Cloud Processing

Tuning parameters under **Point Cloud Processing** can optimize the point cloud quality by optimizing the processing of the obtained data.

Cloud Smooth Mode

This parameter sets whether and how strongly the point cloud surface smoothing algorithm is applied. The options and descriptions are listed below.

Default: **Normal**

Hint: Intensity of smoothing: **Off** < **Weak** < **Normal** < **Strong**.

Option	Description
Off	Apply no smoothing algorithm to retain object details as much as possible.
Normal	Apply moderate smoothing. Balance between smoothness and detail preservation.
Weak	Apply weak smoothing. Keep more details and apply less smoothing.
Strong	Apply strong smoothing. Keep fewer details and apply more smoothing.

Cloud Outlier Filter Mode

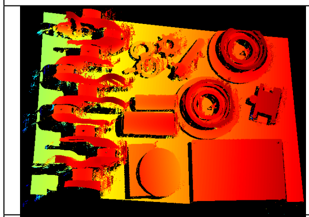
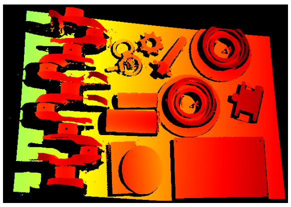
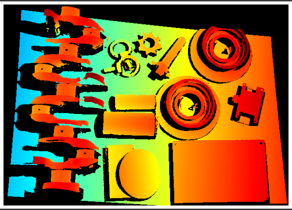
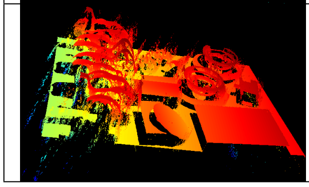
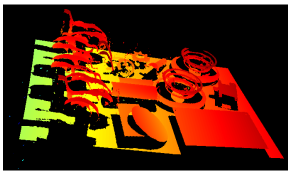
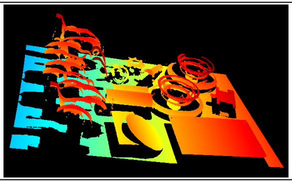
This parameter sets whether and how strongly the point cloud outlier removal algorithm is applied. The options and descriptions are listed below.

Default: **Normal**

Hint: Intensity of outlier removal: **Off** < **Weak** < **Normal**.

Option	Description
Off	Apply no outlier removal. Keep all noise.
Weak	Apply weak outlier removal. Remove a small amount of noise. Produce more accurate and complete edges.
Normal	Apply outlier removal. Remove all noise detected. Mild edge erosion may occur.

Point Clouds under different intensities of outlier removal:

Off	Weak	Normal
		
		

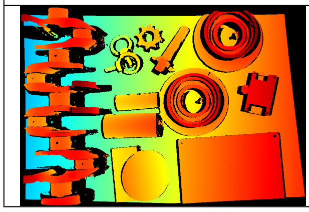
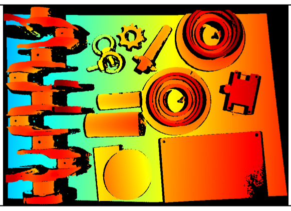
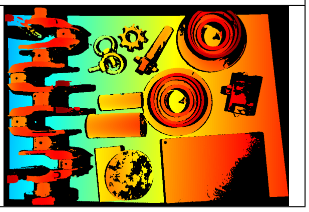
Fringe Contrast Threshold

This parameter sets the signal contrast threshold for selecting valid pixels. Pixels with contrast less than this threshold will be ignored. A higher value means filtering more image noise but may remove point clouds of dark objects.

Default: 3

Usually, the default value 3 works well but you can increase the value for images with heavy noise.

Point clouds obtained with different values of **Fringe Contrast Threshold**:

3	15	30
		

Hint:

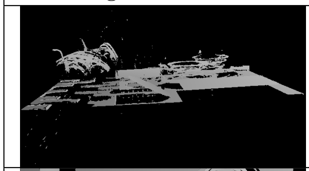
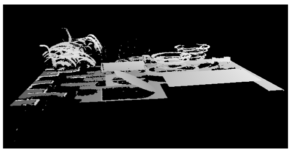
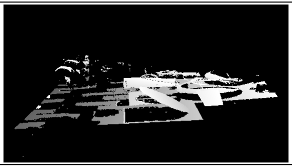
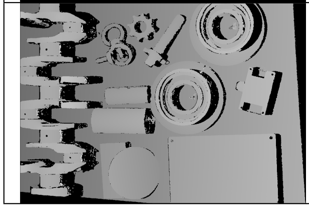
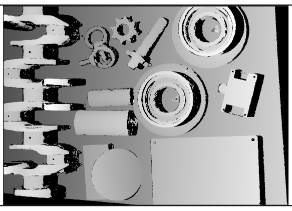
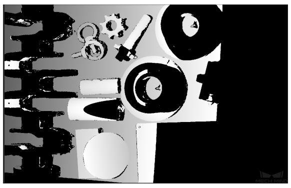
- Usually, setting **Cloud Smooth Mode** and **Cloud Outlier Filter Mode** to `Normal` and setting **Fringe Contrast Threshold** to `3` will help obtain satisfactory point clouds.
- If after tuning the parameters, the point cloud is still not satisfactory, please save the raw data of the camera and save the images via *File* → *Save Raw Data of Camera* and *File* → *Save Images*, and contact the support team.

Depth Range

The upper limit and the lower limit of depth range are to set the interested range of depth on the depth map and ignore the parts of the depth map with depth values beyond the depth range.

Max range: 1 to 9,999 mm

Depth maps under different depth range settings:

Too large	Moderate	Too small
		
		

Hint: Please set an appropriate range to select the parts that are relevant to your projects. The range should not be too large or interference may be introduced. Nor should it be too small or the point cloud the key parts may be missing.

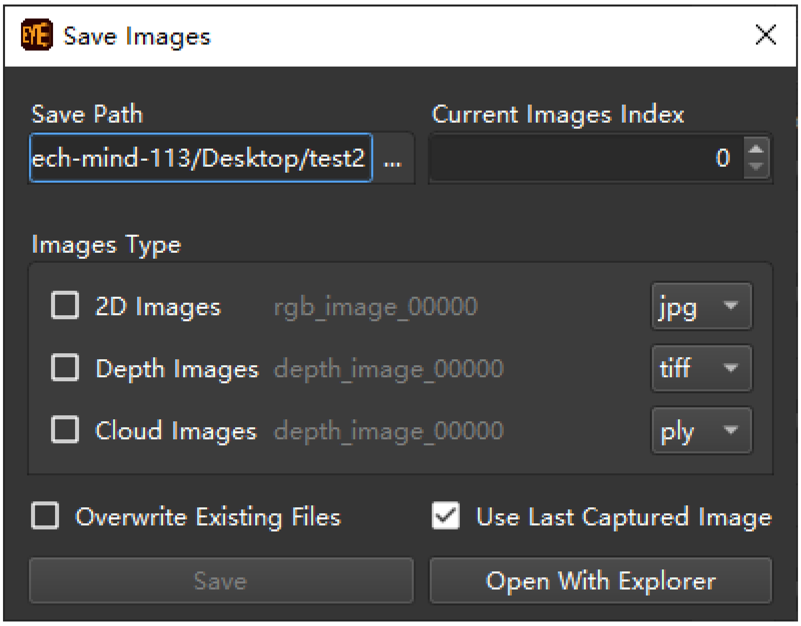
Save Data

Save Raw Data of Camera

You can save the raw image data captured by the camera in the .mraw format by clicking on *File –> Save Raw Data of Camera.*

Save Images

You can save images by data type by clicking on *File –> Save Images* and selecting the image type(s) in the pop-up window **Save Images**.



Save Images

Save Path: ech-mind-113/Desktop/test2

Current Images Index: 0

Images Type

- ☐ 2D Images — rgb_image_00000 — jpg
- ☐ Depth Images — depth_image_00000 — tiff
- ☐ Cloud Images — depth_image_00000 — ply

☐ Overwrite Existing Files ☑ Use Last Captured Image

The pop-up window of **Save Images**

Component	**Instructions**
Saving path	Please select a path to save the data
Current image index	Set an index and the software will automatically add the index into the filename strings to ensure there are no repeating filenames.
Data types	Please select the data type(s) that need to be saved.
Overwrite existing files	If checked, existing files with the same filenames as those to save will be overwritten.
Use last captured image	If checked, the software will save data from the image captured most recently. If unchecked, you will be redirected to the image capturing interface to capture a new image for data saving.
Save	Click to save the data.
Open in explorer	Open the folder where the software is installed in the explorer.

3.4.3 Built-In Tools

The software comes with built-in tools that can be opened at *Tools*.

Check Camera Intrinsic Parameters

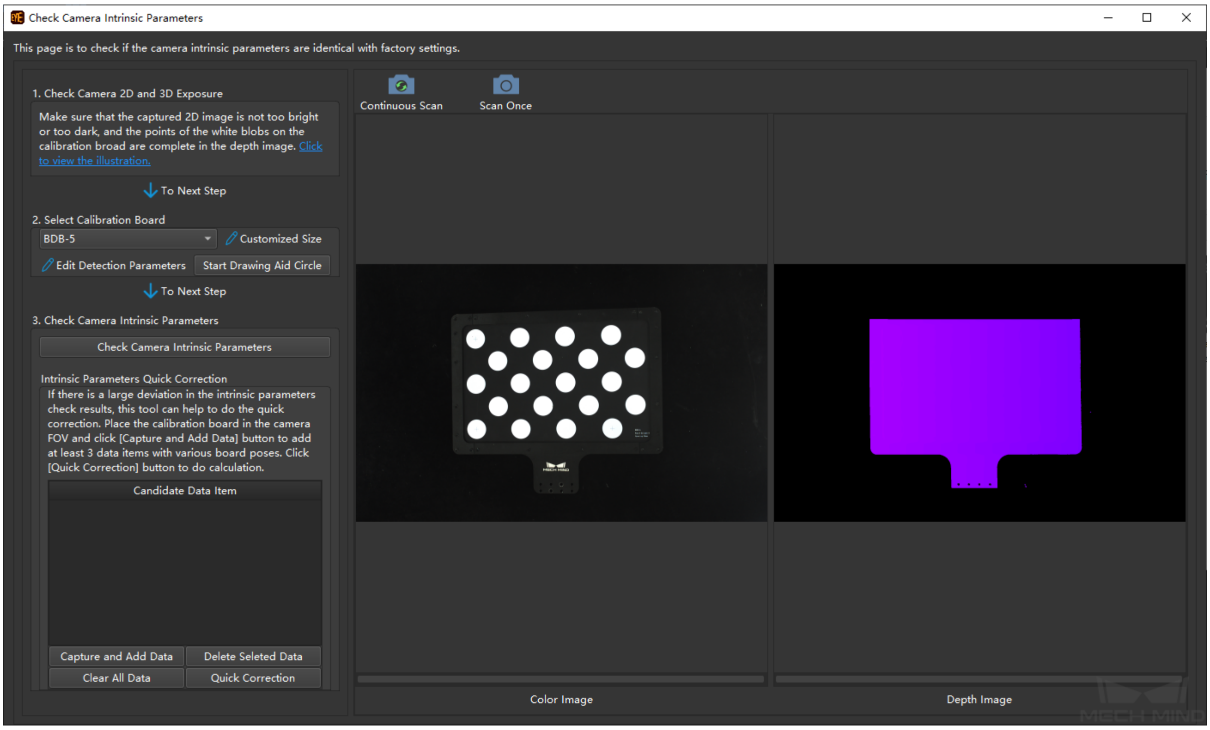
Note

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

Interface

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

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



Check Camera Intrinsic Parameters

This page is to check if the camera intrinsic parameters are identical with factory settings.

1. Check Camera 2D and 3D Exposure

Make sure that the captured 2D image is not too bright or too dark, and the points of the white blobs on the calibration broad are complete in the depth image. Click to view the illustration.

2. Select Calibration Board

BDB-5

3. Check Camera Intrinsic Parameters

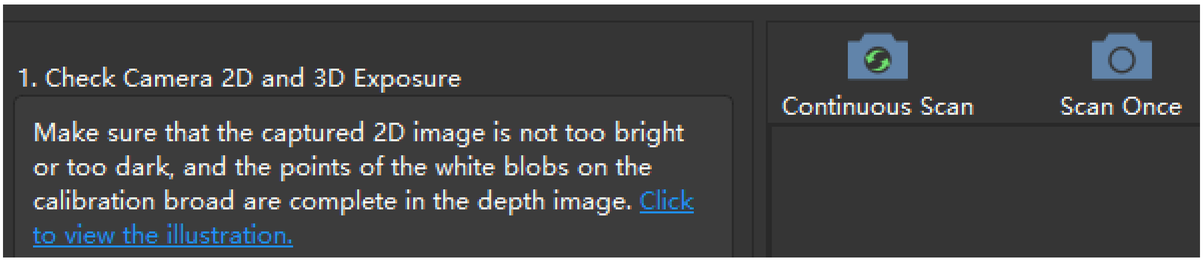
Intrinsic Parameters Quick Correction

If there is a large deviation in the intrinsic parameters check results, this tool can help to do the quick correction. Place the calibration board in the camera FOV and click [Capture and Add Data] button to add at least 3 data items with various board poses. Click [Quick Correction] button to do calculation.

Steps

1. Check camera 2D and 3D exposure

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

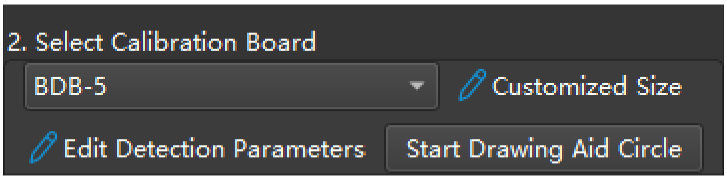


1. Check Camera 2D and 3D Exposure

Make sure that the captured 2D image is not too bright or too dark, and the points of the white blobs on the calibration broad are complete in the depth image. Click to view the illustration.

2. Select a calibration board

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

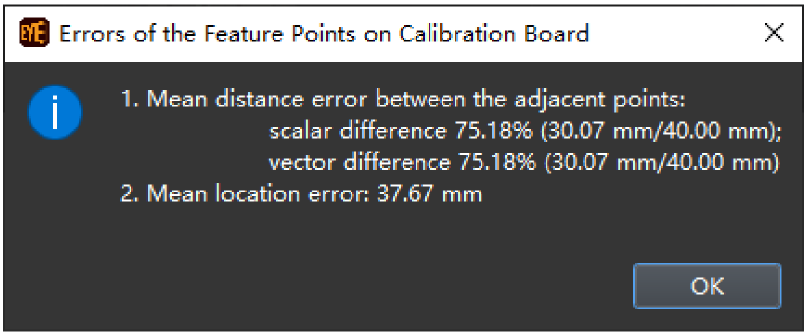


2. Select Calibration Board

BDB-5

3. Check camera intrinsic parameters

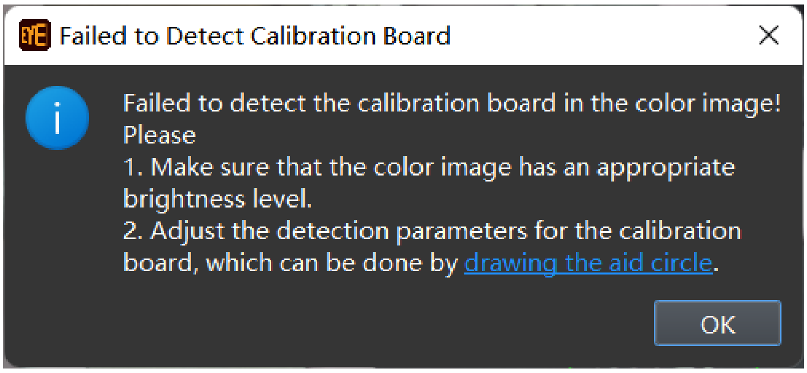
Please click on *Check Camera Intrinsic Parameters* to start checking. If the checking is successfully completed, a window indicating **errors of the feature points on calibration board** as shown below will pop up.



Errors of the Feature Points on Calibration Board

1. Mean distance error between the adjacent points:
   scalar difference 75.18% (30.07 mm/40.00 mm);
   vector difference 75.18% (30.07 mm/40.00 mm)
2. Mean location error: 37.67 mm

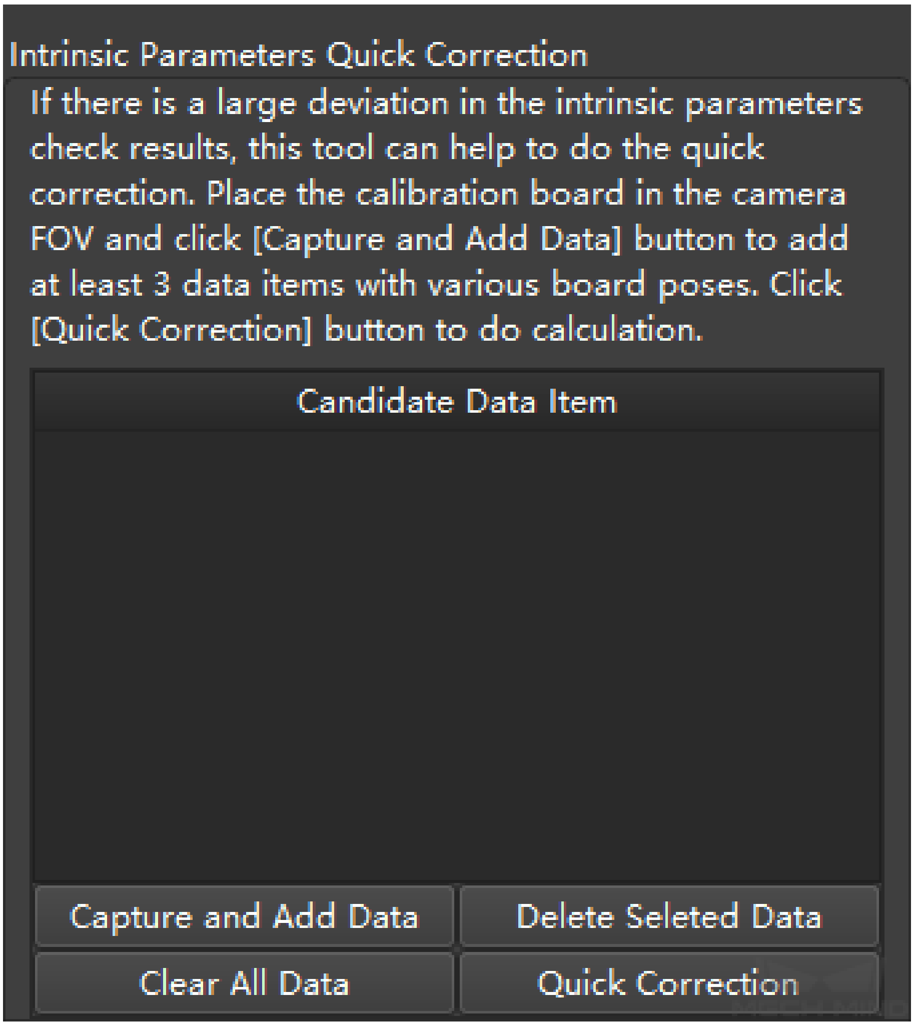
If the checking failed, a window saying **failed to detect calibration board** as shown below will pop up.



Failed to Detect Calibration Board

Failed to detect the calibration board in the color image! Please
1. Make sure that the color image has an appropriate brightness level.
2. Adjust the detection parameters for the calibration board, which can be done by drawing the aid circle.

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



Intrinsic Parameters Quick Correction

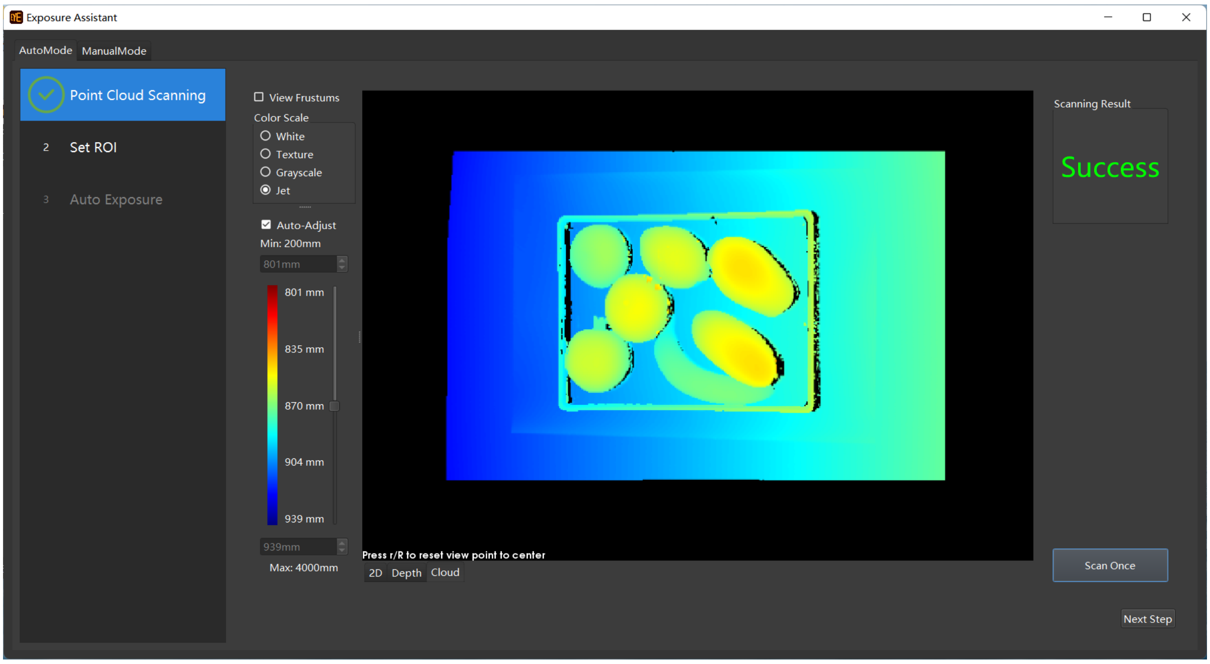
If there is a large deviation in the intrinsic parameters check results, this tool can help to do the quick correction. Place the calibration board in the camera FOV and click [Capture and Add Data] button to add at least 3 data items with various board poses. Click [Quick Correction] button to do calculation.

Exposure Assistant

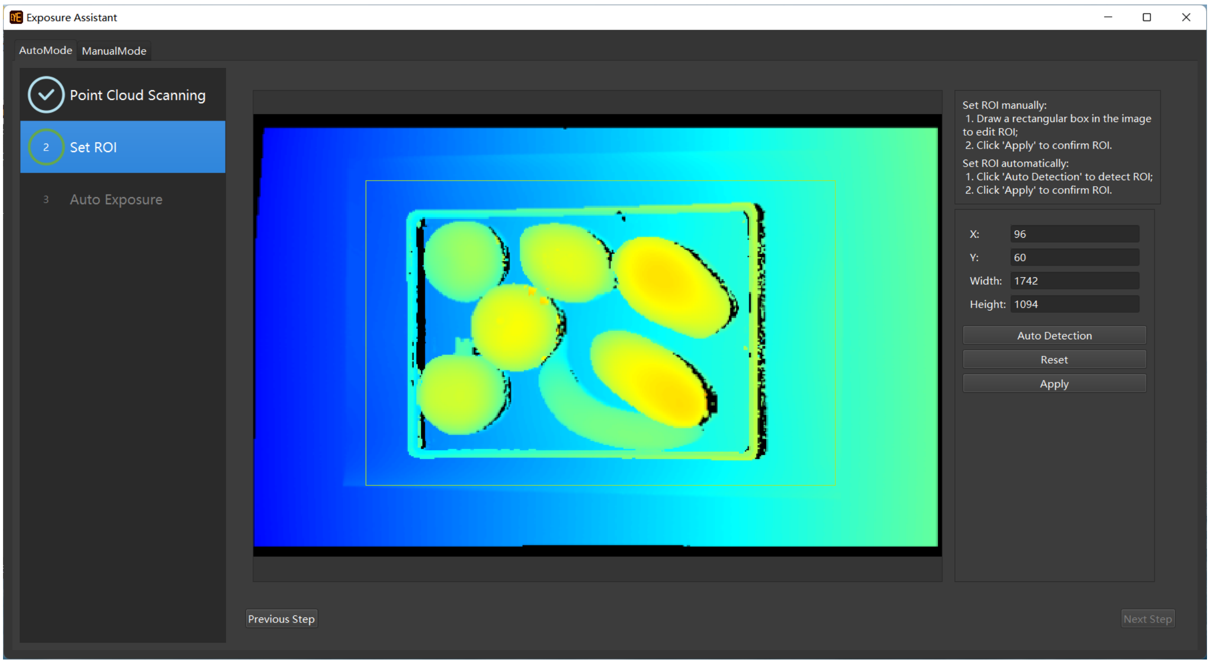
This tool is to help you obtain the optimal 3D exposure parameter settings. It works in two moes: **AutoMode**, **ManualMode**. You can open this tool via *Tools* → *Exposure Assistant*.

AutoMode

1. Click on *Scan Once* in the lower right corner of the Exposure Assistant window, as shown below.



2. In AutoMode, the software will automatically set an ROI. If you need to customize the ROI, please manually set the ROI and click on *Apply*. Click on *Next Step*.

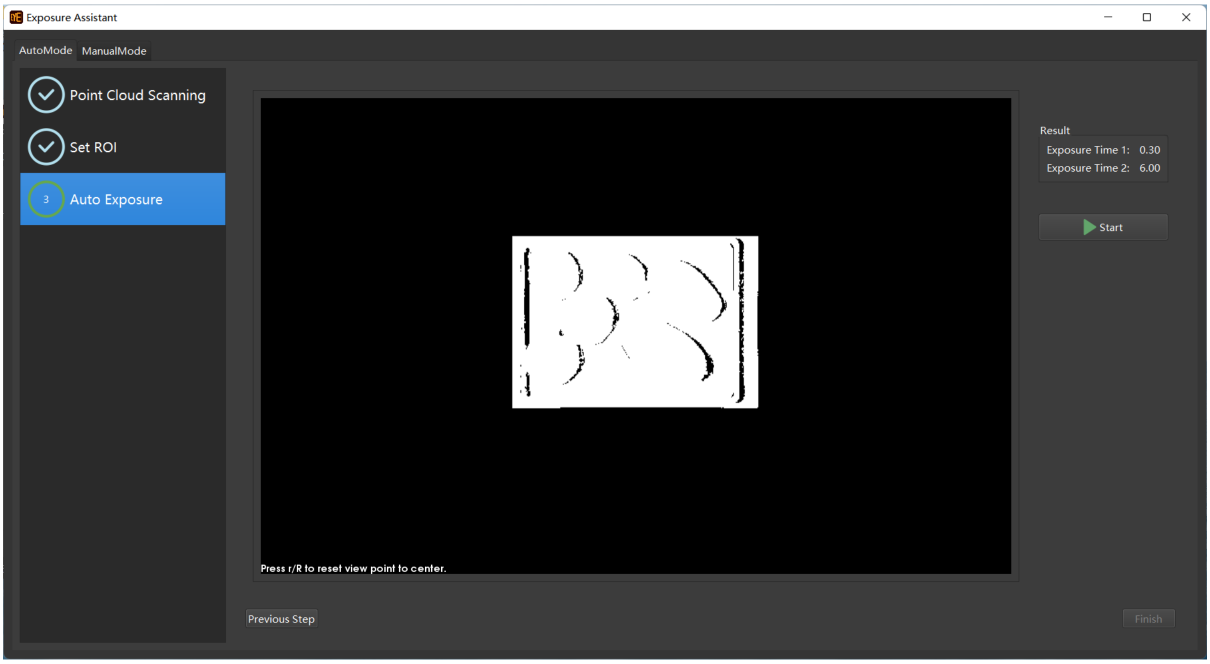


Set ROI manually:
1. Draw a rectangular box in the image to edit ROI;
2. Click 'Apply' to confirm ROI.

Set ROI automatically:
1. Click 'Auto Detection' to detect ROI;
2. Click 'Apply' to confirm ROI.

X: 96
Y: 60
Width: 1742
Height: 1094

3. Click on *Start* to obtain the optimal 3D exposure parameter settings, as shown in the figure below.



Result

Exposure Time 1: 0.30

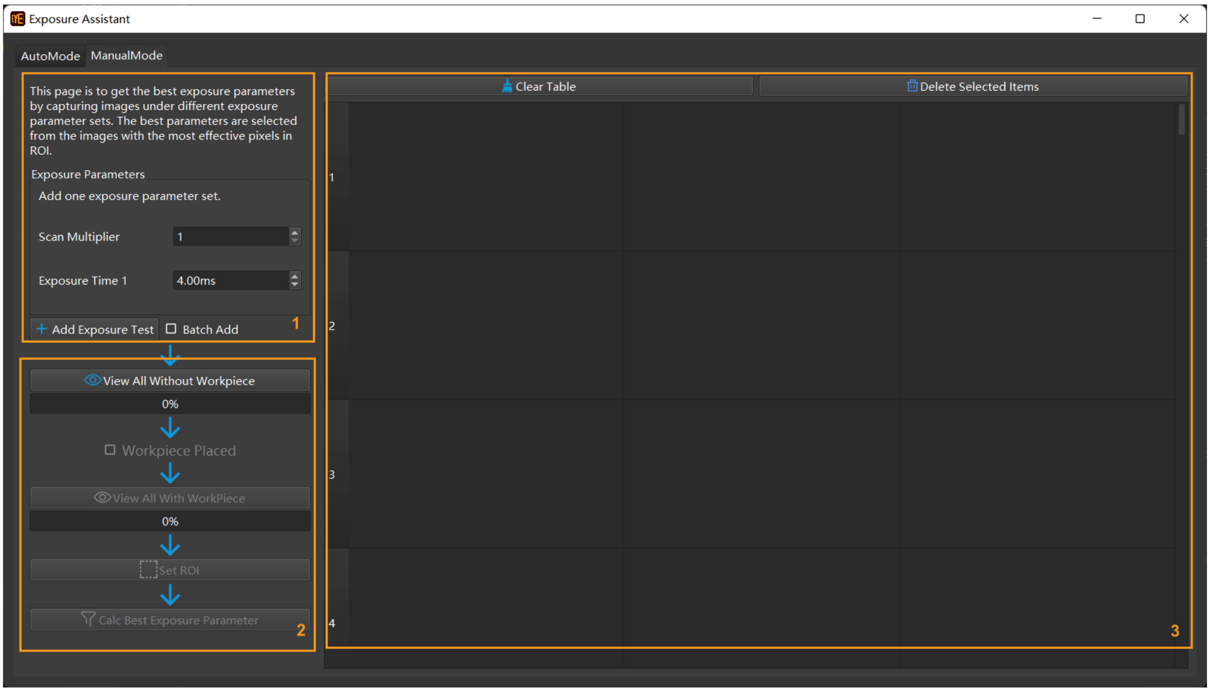
Exposure Time 2: 6.00

Please manually copy the optimal 3D exposure parameter settings into the camera’ s 3D exposure parameters, i.e., scan multiplier, exposure time(s).

ManualMode

Interface

The interface of Exposure Assistant in the manual mode is as shown below:



This page is to get the best exposure parameters by capturing images under different exposure parameter sets. The best parameters are selected from the images with the most effective pixels in ROI.

Exposure Parameters

Add one exposure parameter set.

Scan Multiplier 1

Exposure Time 1 4.00ms

No.	Section	Use
1	Exposure parameters	You can add test samples by adding different settings on scan multiplier and exposure time for testing in this section.
2	Workflow	After adding all the test samples required, you can start the workflow to obtain the optimal parameter settings.
3	Test samples	All the test samples added are shown here. You can delete test samples or modify sample parameter settings in this section.

Steps

1. Add test samples

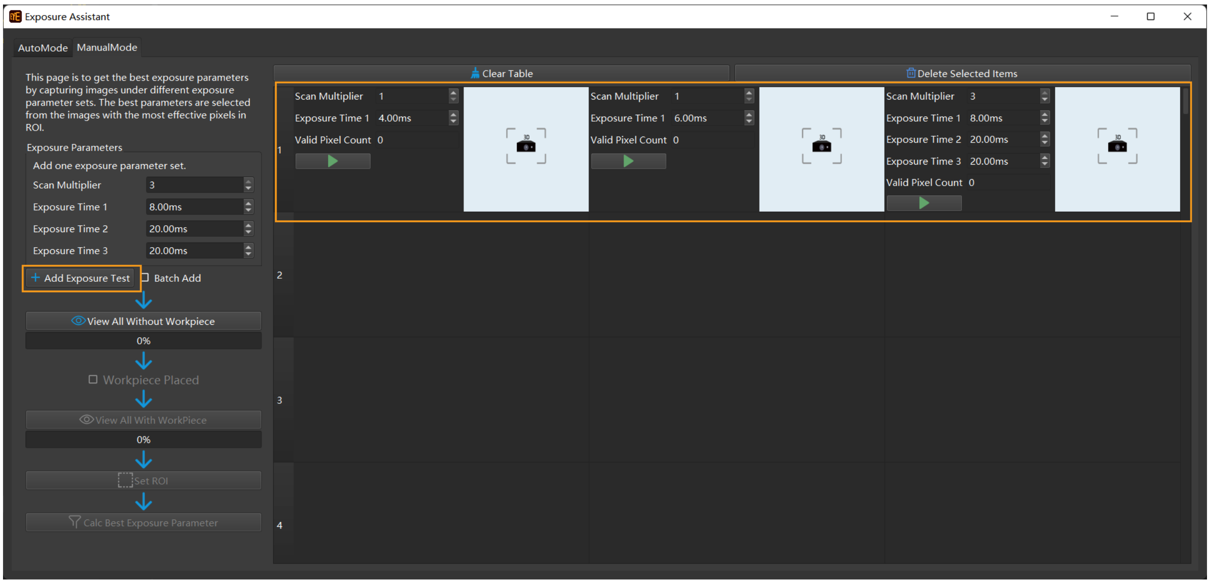
There are two ways to add test samples: **add single exposure test** and **batch add**.

Attention: For dark objects please set longer exposure times. For reflective objects, please set shorter exposure times. For laser cameras, the scan times (in ms) must be multiples of 4.

Add single exposure test

You can set the scan multiplier and exposure time(s) for a new test sample and click on *Add Exposure Test* to add the test sample.

The new sample added will appear on the right.



Exposure Assistant

AutoMode ManualMode

This page is to get the best exposure parameters by capturing images under different exposure parameter sets. The best parameters are selected from the images with the most effective pixels in ROI.

Exposure Parameters

Add one exposure parameter set.

Scan Multiplier	3
Exposure Time 1	8.00ms
Exposure Time 2	20.00ms
Exposure Time 3	20.00ms

+ Add Exposure Test ☐ Batch Add

View All Without Workpiece 0%

☐ Workpiece Placed

View All With WorkPiece 0%

Set ROI

Calc Best Exposure Parameter

Clear Table Delete Selected Items

1

Scan Multiplier	1
Exposure Time 1	4.00ms
Valid Pixel Count	0

Scan Multiplier	1
Exposure Time 1	6.00ms
Valid Pixel Count	0

Scan Multiplier	3
Exposure Time 1	8.00ms
Exposure Time 2	20.00ms
Exposure Time 3	20.00ms
Valid Pixel Count	0

2

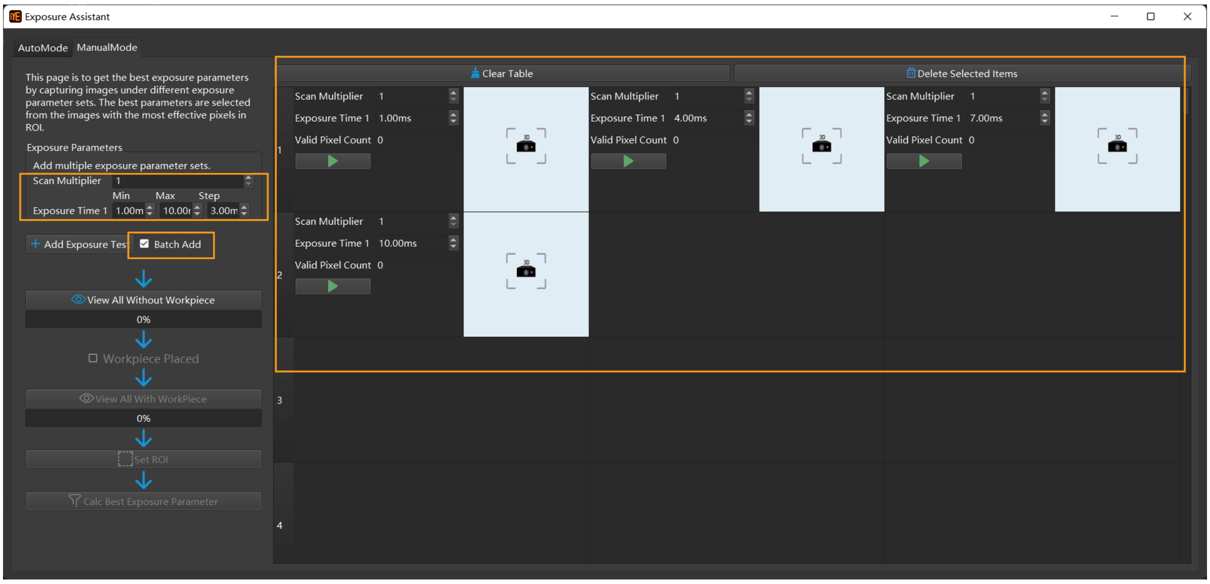
3

4

Batch add

After checking the box *Batch add*, please set the scan multiplier for the test samples in the batch to add, and set the min, max and step for generating the exposure times for the test samples.

When done setting, please click on *Add Exposure Test* to add the batch. The new samples added will appear on the right.



Exposure Assistant

AutoMode ManualMode

This page is to get the best exposure parameters by capturing images under different exposure parameter sets. The best parameters are selected from the images with the most effective pixels in ROI.

Exposure Parameters

Add multiple exposure parameter sets.

	Min	Max	Step
Scan Multiplier	1		
Exposure Time 1	1.00m	10.00	3.00m

+ Add Exposure Tes ☑ Batch Add

View All Without Workpiece 0%

☐ Workpiece Placed

View All With WorkPiece 0%

Set ROI

Calc Best Exposure Parameter

Clear Table Delete Selected Items

1

Scan Multiplier	1
Exposure Time 1	1.00ms
Valid Pixel Count	0

Scan Multiplier	1
Exposure Time 1	4.00ms
Valid Pixel Count	0

Scan Multiplier	1
Exposure Time 1	7.00ms
Valid Pixel Count	0

2

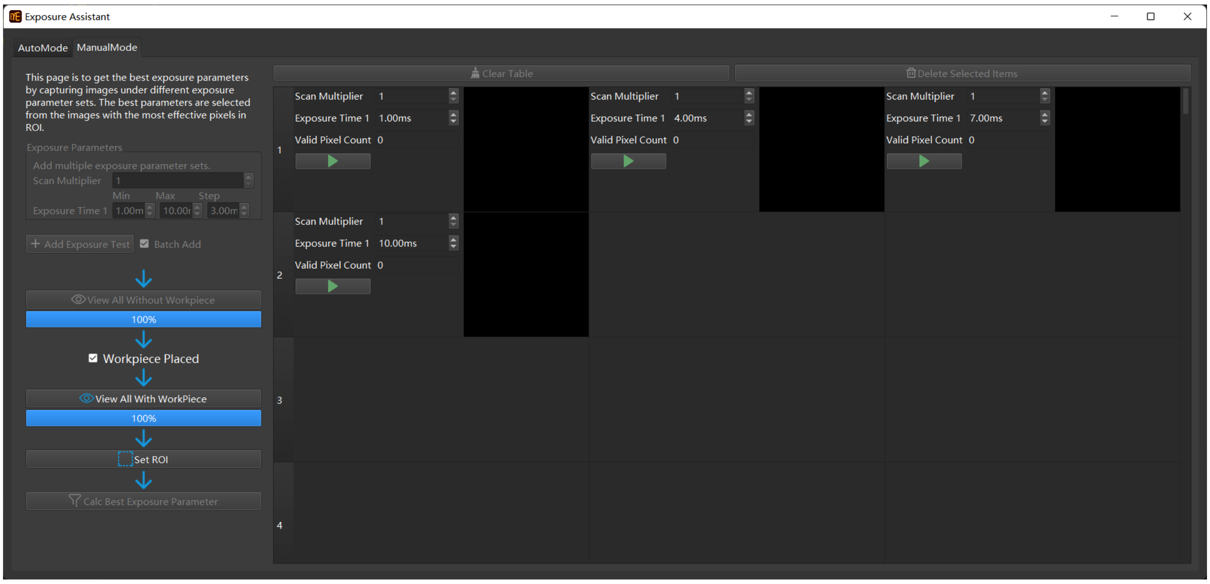
Scan Multiplier	1
Exposure Time 1	10.00ms
Valid Pixel Count	0

3

4

2. View all without workpieces (objects)

After clicking on *View All without Workpiece*, the software will start the exposure tests on the scene without objects sample by sample and will show the progress by displaying captured images in completed samples.



Exposure Assistant

AutoMode ManualMode

This page is to get the best exposure parameters by capturing images under different exposure parameter sets. The best parameters are selected from the images with the most effective pixels in ROI.

Exposure Parameters

Add multiple exposure parameter sets.

Scan Multiplier 1

	Min	Max	Step
Exposure Time 1	1.00m	10.00r	3.00m

+ Add Exposure Test ☑ Batch Add

View All Without Workpiece

100%

☑ Workpiece Placed

View All With WorkPiece

100%

Set ROI

Calc Best Exposure Parameter

Clear Table | Delete Selected Items

1: Scan Multiplier 1; Exposure Time 1 1.00ms; Valid Pixel Count 0 — Scan Multiplier 1; Exposure Time 1 4.00ms; Valid Pixel Count 0 — Scan Multiplier 1; Exposure Time 1 7.00ms; Valid Pixel Count 0

2: Scan Multiplier 1; Exposure Time 1 10.00ms; Valid Pixel Count 0

3

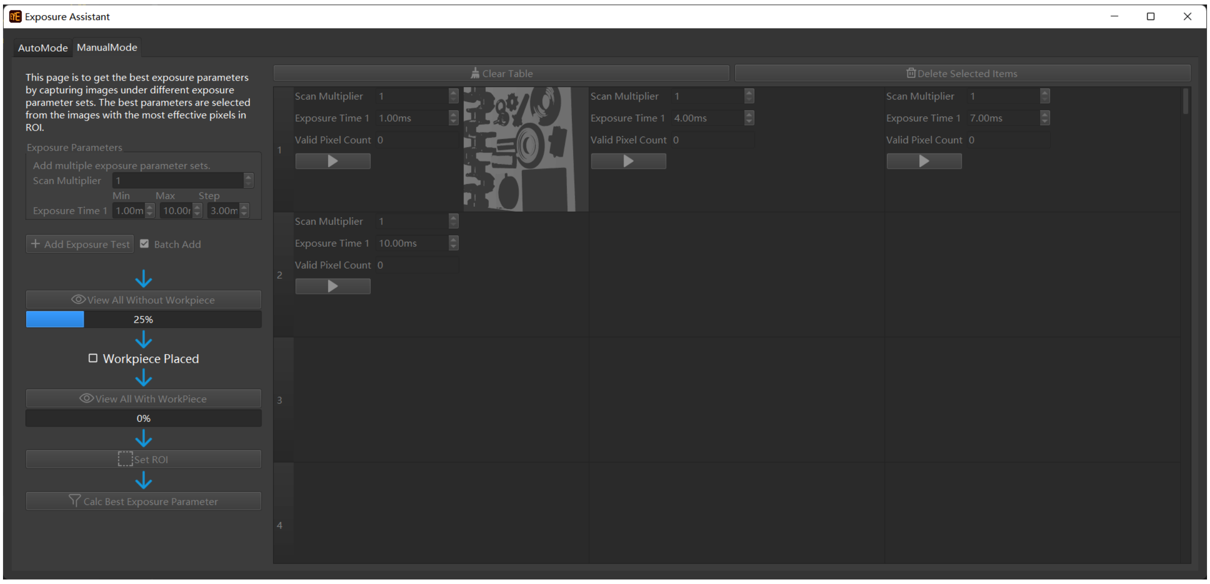
4

1. Place workpieces (objects)

After the software is done capturing images of the scene without objects for all test samples, please place the objects into the scene, and check the box *Workpiece Placed*.

4. View all with workpieces (objects)

After you click on *View All without Workpiece*, the software will start the exposure tests on the scene with objects sample by sample and will show the progress by displaying captured images in completed samples.



This page is to get the best exposure parameters by capturing images under different exposure parameter sets. The best parameters are selected from the images with the most effective pixels in ROI.

Exposure Parameters

Add multiple exposure parameter sets.

		Min	Max	Step
Scan Multiplier	1			
Exposure Time 1		1.00m	10.00r	3.00m

☑ Batch Add

25%

□ Workpiece Placed

0%

Scan Multiplier	Exposure Time 1	Valid Pixel Count
1	1.00ms	0
1	4.00ms	0
1	7.00ms	0
1	10.00ms	0

5. Set ROI

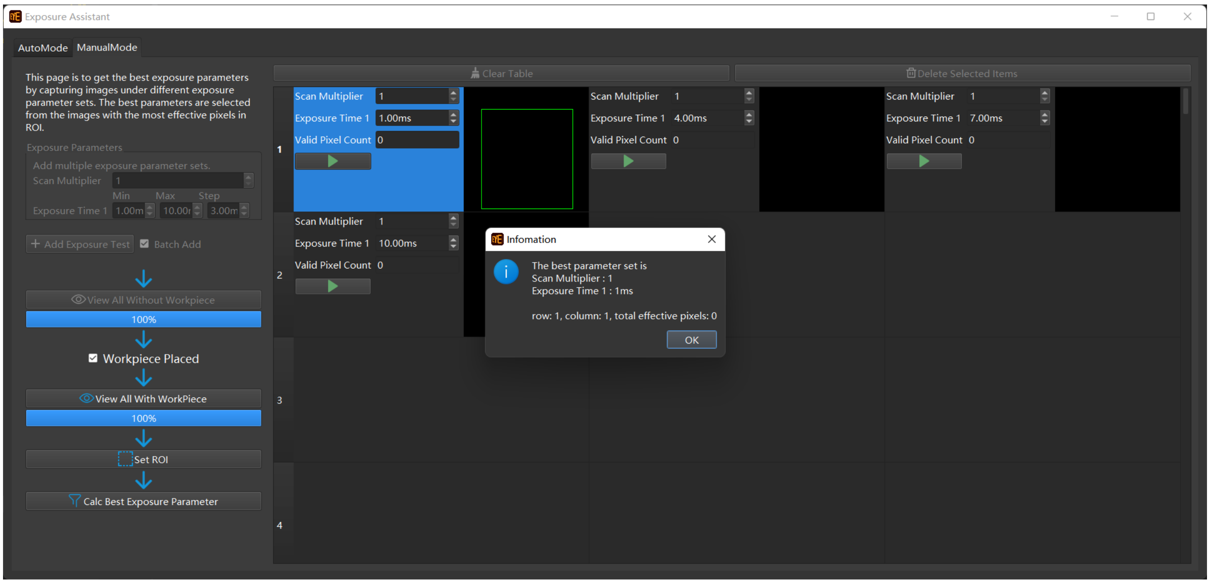
After clicking on *Set ROI*, you can set the ROI on the image in the first test sample on the right.

6. Calculate optimal exposure parameter settings

After clicking on *Calc Best Exposure Parameter*, a window indicating the optimal exposure parameter settings will pop up. The optimal parameter settings are obtained from the test set with the largest quantity of effective pixels.

7. Apply the optimal exposure parameter settings to the camera

After getting the optimal exposure parameter settings, you can close the Exposure Assistant and a pop-up window will ask you if you want to apply the optimal settings to the camera. Please click on *OK* to apply the settings.



This page is to get the best exposure parameters by capturing images under different exposure parameter sets. The best parameters are selected from the images with the most effective pixels in ROI.

Add multiple exposure parameter sets.

Scan Multiplier 1

	Min	Max	Step
Exposure Time 1	1.00m	10.00r	3.00m

☑ Batch Add

☑ Workpiece Placed

Scan Multiplier 1
Exposure Time 1 1.00ms
Valid Pixel Count 0

Scan Multiplier 1
Exposure Time 1 4.00ms
Valid Pixel Count 0

Scan Multiplier 1
Exposure Time 1 7.00ms
Valid Pixel Count 0

Scan Multiplier 1
Exposure Time 1 10.00ms
Valid Pixel Count 0

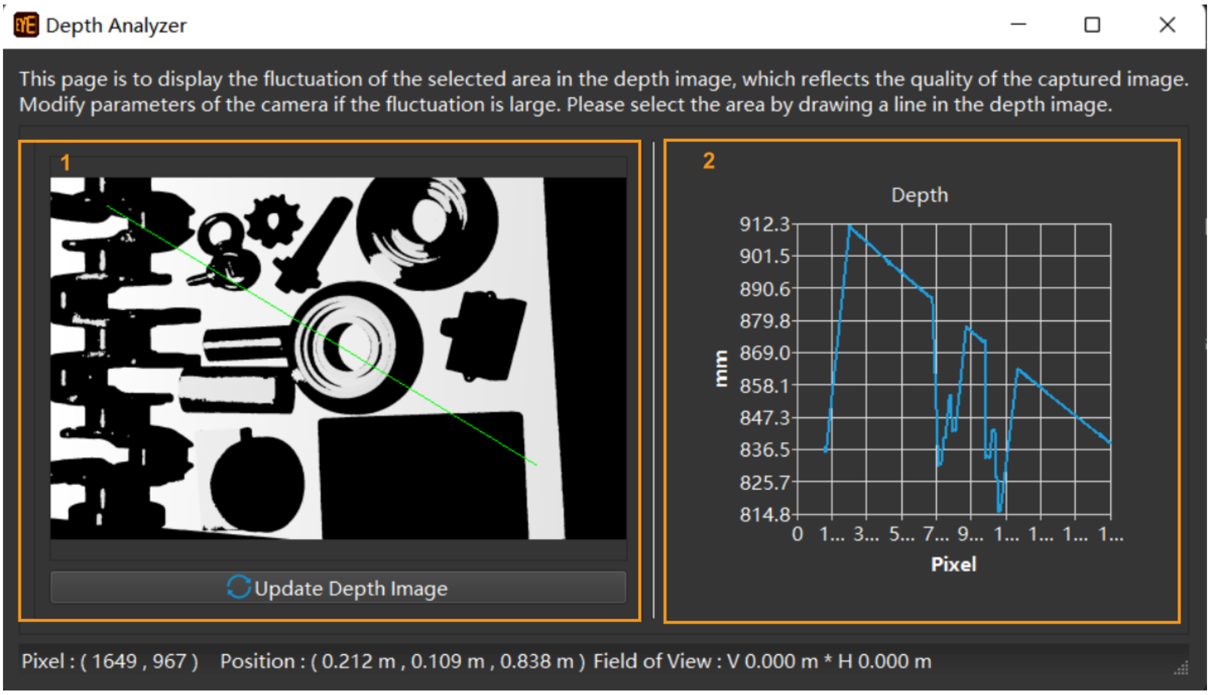
The best parameter set is
Scan Multiplier : 1
Exposure Time 1 : 1ms

row: 1, column: 1, total effective pixels: 0

Depth Analyzer

This tool is to help check the imaging quality of the depth map. You can open this tool via *Tools → Depth Analyzer*。

Interface



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

Pixel : (1649 , 967) Position : (0.212 m , 0.109 m , 0.838 m) Field of View : V 0.000 m * H 0.000 m

No.	Section	Function
1	Depth map	Display the depth map captured by the camera to draw the horizontal line for the depth plot.
2	Depth plot	Display the depth plot along the line drawn on the depth map.

Instructions

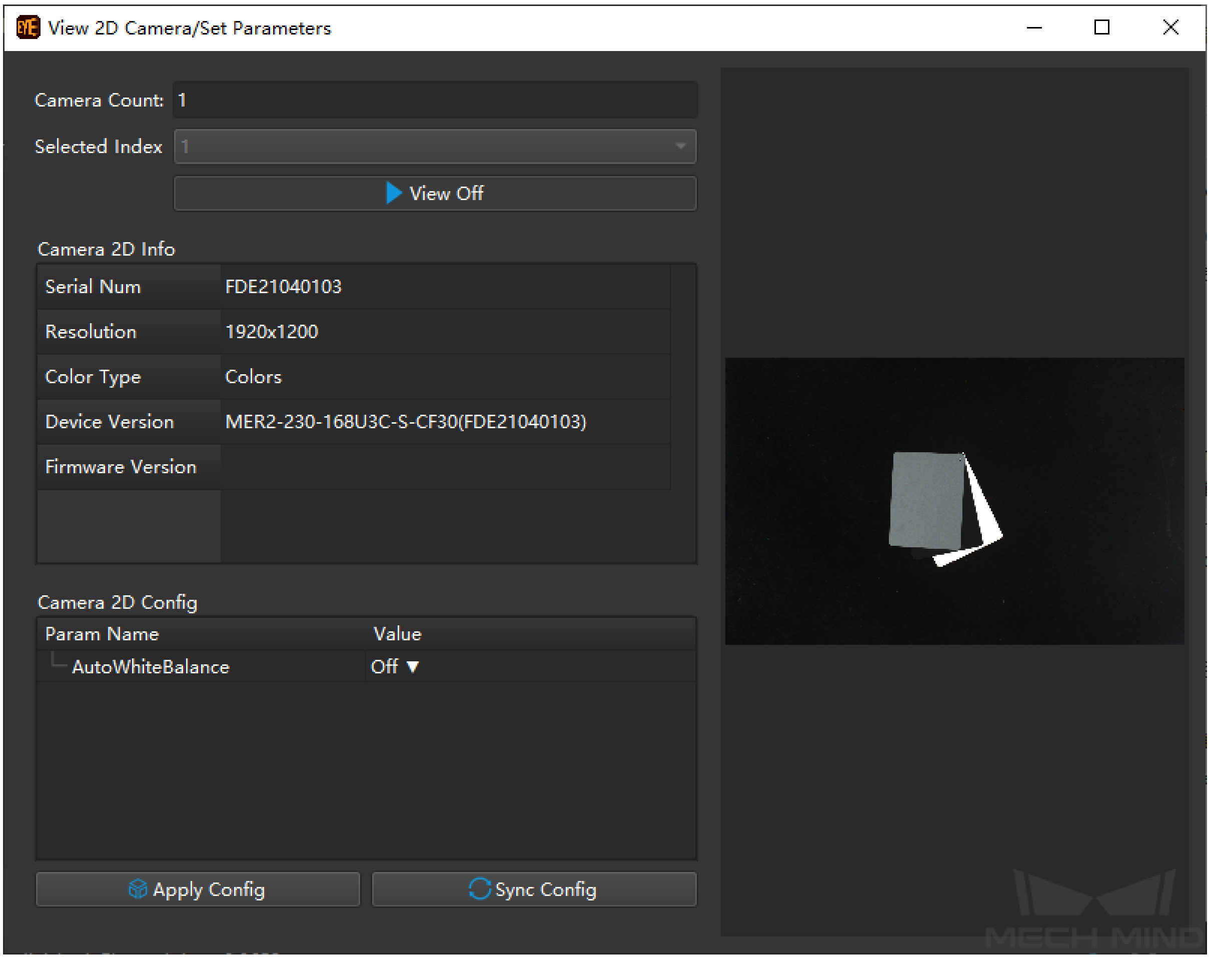
Please hold and drag on the depth map on the left to draw a straight line (the green line on the depth map in the figure above). A depth plot along the line will be displayed on the right.

Please check whether the depth plot reflects the actual shapes of the objects along the line.

If yes, the depth map’ s quality is satisfactory. If no, please adjust the parameters for the depth map to obtain a better depth map.

View 2D Camera and Set Parameters

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



Camera Count: 1

Selected Index 1

Camera 2D Info

Serial Num	FDE21040103
Resolution	1920x1200
Color Type	Colors
Device Version	MER2-230-168U3C-S-CF30(FDE21040103)
Firmware Version	

Camera 2D Config

Param Name	Value
AutoWhiteBalance	Off ▼

Camera Information

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

Camera Configuration

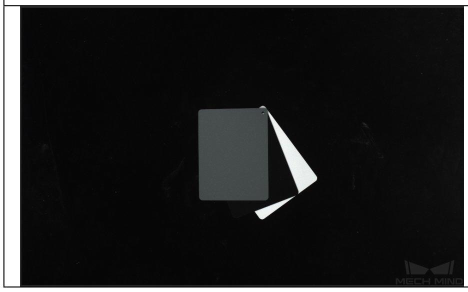
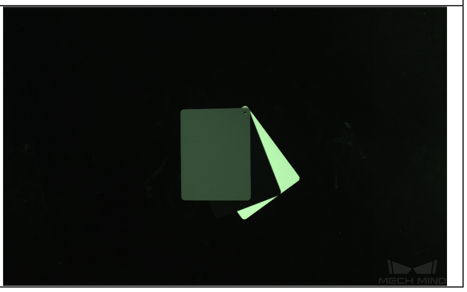
Auto White Balance

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

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

You can enable the auto white balance function at *Tools → View 2D Camera and Set Parameters → Camera 2D Config → AutoWhiteBalance*.

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

Color board in reality	Color board image without white balance
	

Instructions

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

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

3.4.4 Other Settings

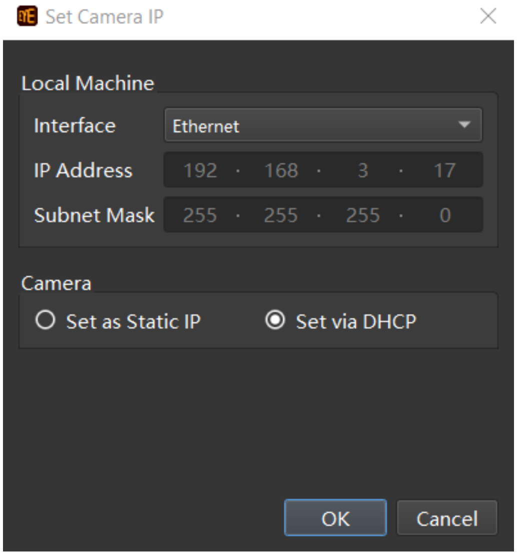
IP Settings of Camera and IPC

Attention:

- The camera must be in the same network segment as the IPC before the two can be used together. To facilitate future use, please store the IP address after it is set.
- To make sure the IP address is correct, please consult your network administrator before setting a fixed IP.

Set camera IP

Open the software and enter the device finder interface. Click on **Set Camera IP** to enter the IP setting interface, as shown in the figure below.



Set Camera IP

Local Machine

Interface: Ethernet

IP Address: 192 . 168 . 3 . 17

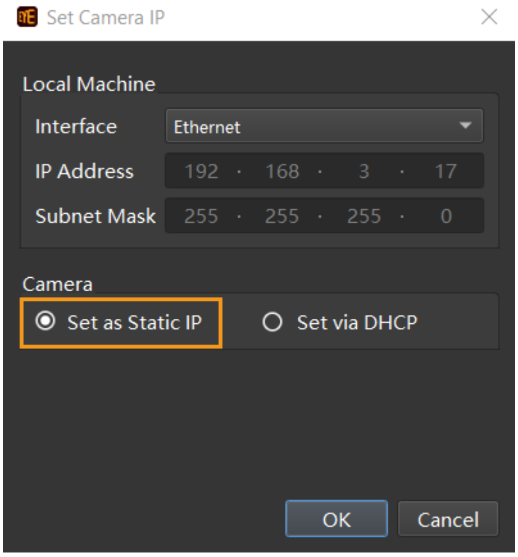
Subnet Mask: 255 . 255 . 255 . 0

Camera

○ Set as Static IP ● Set via DHCP

Static IP

Check *Set as Static IP* and manually set the IP address class, IP address, and subnet mask.



Set Camera IP

Local Machine

Interface: Ethernet

IP Address: 192 · 168 · 3 · 17

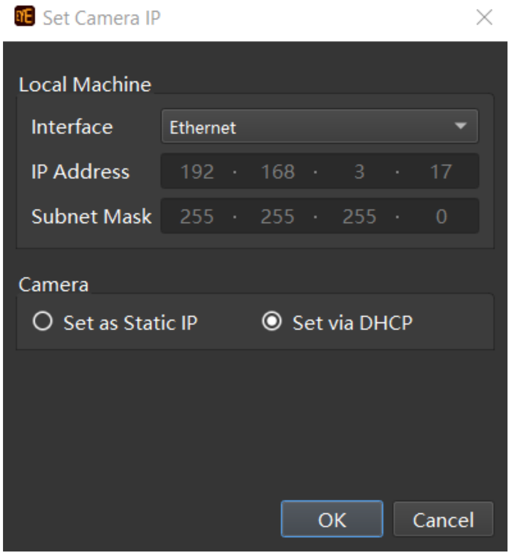
Subnet Mask: 255 · 255 · 255 · 0

Camera

● Set as Static IP ○ Set via DHCP

Dynamic IP

Check *Set via DHCP* and click on *OK*. The IP address will be set via the Dynamic Host Configuration Protocol (DHCP).



Set Camera IP

Local Machine

Interface: Ethernet

IP Address: 192 · 168 · 3 · 17

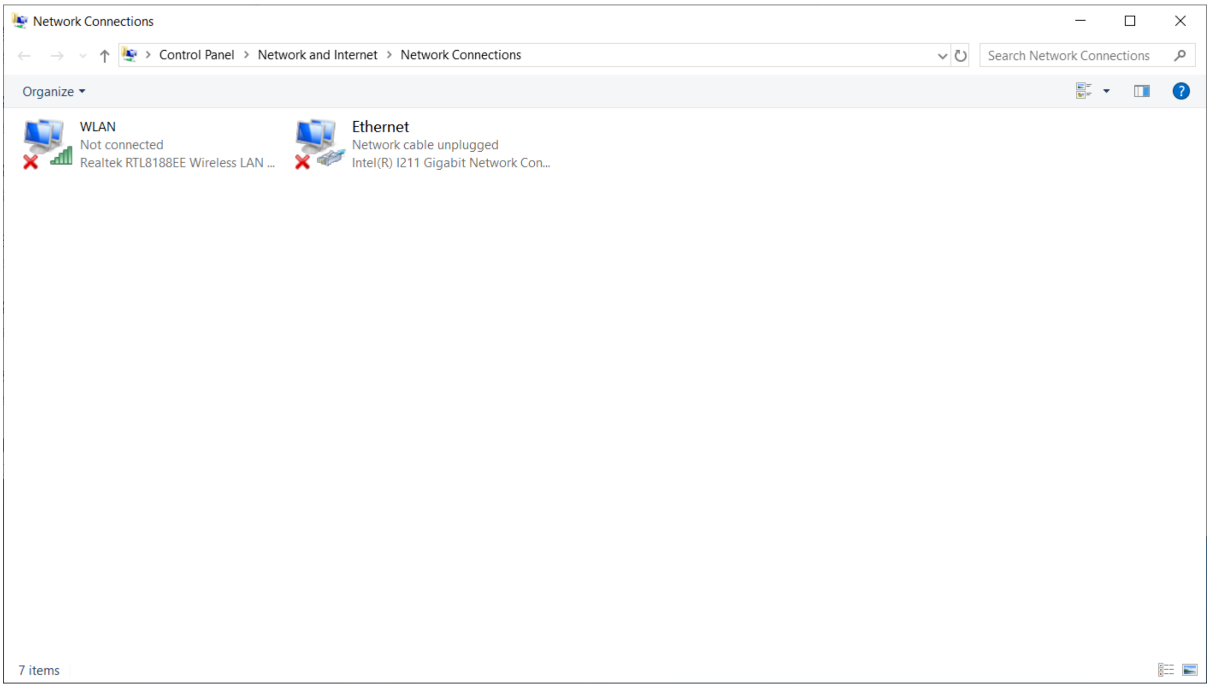
Subnet Mask: 255 · 255 · 255 · 0

Camera

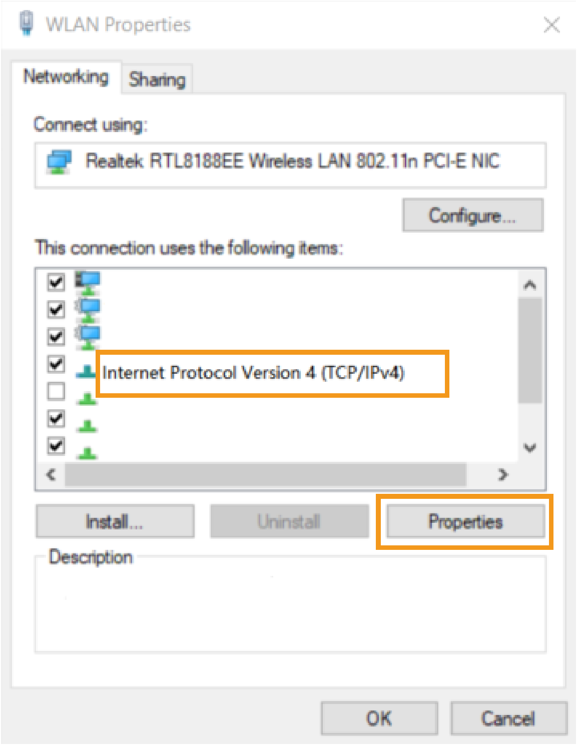
○ Set as Static IP ● Set via DHCP

Set IPC IP

1. Go to *Control Panel* → *Network and Internet* → *Network and Sharing Center -> Change adapter settings.*

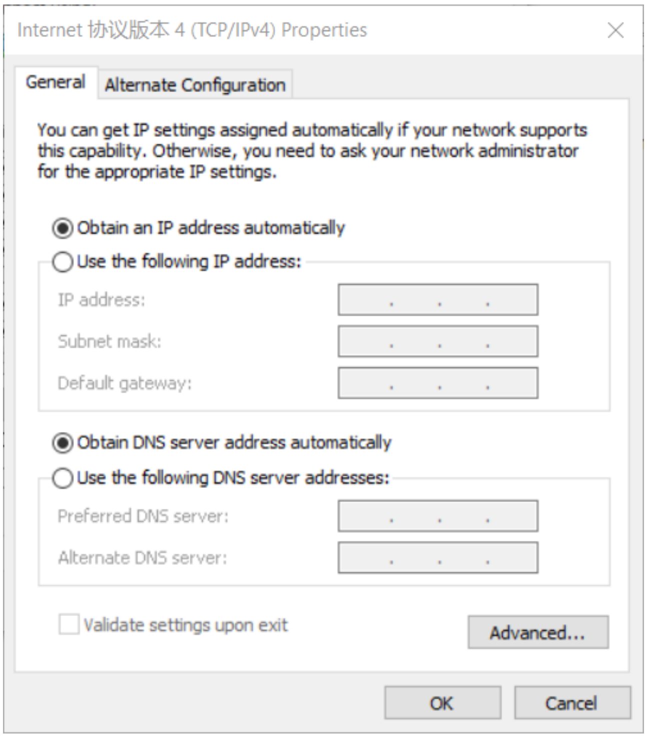

Network Connections
Control Panel > Network and Internet > Network Connections
Search Network Connections
Organize ▾
WLAN
Not connected
Realtek RTL8188EE Wireless LAN ...
Ethernet
Network cable unplugged
Intel(R) I211 Gigabit Network Con...
7 items

1. Right-click on the camera, and open **Property**.



WLAN Properties
Networking
Sharing
Connect using:
Realtek RTL8188EE Wireless LAN 802.11n PCI-E NIC
Configure...
This connection uses the following items:
☑
☑
☑
☑ Internet Protocol Version 4 (TCP/IPv4)
☐
☑
☑
Install...
Uninstall
Properties
Description
OK
Cancel

1. Right-click on *Internet protocol version 4 (TCP/IPv4)*, open **Property**, and then set the IPC IP.



Internet 协议版本 4 (TCP/IPv4) Properties

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: . . .

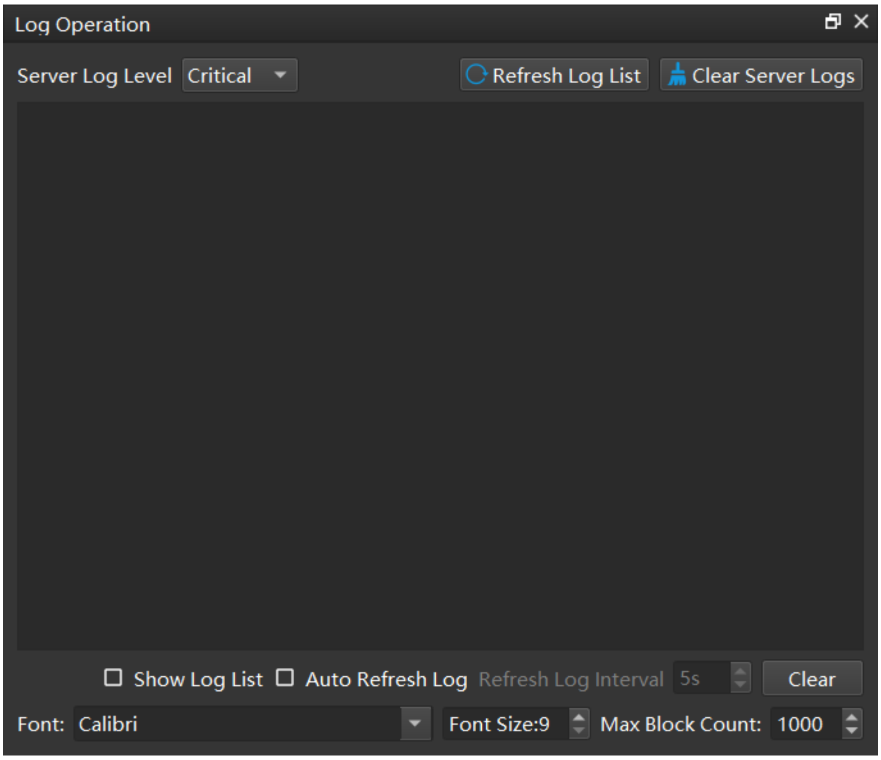
Alternate DNS server: . . .

□ Validate settings upon exit

Log Operation

You can open the **Log Operation** at *View* → *Log Operation* to view, clear, or export the log.

The interface of log operation is as follows:



User

The default option is `Standard` and is suitable for most uses.

It is not recommended to select other options.

If you need to switch to other options, please contact the support team.

Language

You can change the software language at *Help* → *Language*.

Available languages include Chinese, English, Japanese, Korean.

Please restart the software to apply the language settings.

Check the following section to further learn about **Mech-Eye API**

3.5 API Reference

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