

---

# Mech-Eye Industrial 3D Cameras

**Mech-Mind**

2023 年 06 月 28 日

1	相机使用指南	4
2	Mech-Eye SDK 2.1 更新说明	46
3	Mech-Eye SDK 安装指南	53
4	Mech-Eye Viewer 快速使用	58
5	快速了解 Mech-Eye Viewer	61
6	界面介绍	63
7	操作指南	66
8	内置工具	108
9	快速了解 Mech-Eye API	126
10	快速入门	128
11	例程实操指南	137
12	API——固件升级工具	167
13	Mech-Eye API 参考手册	169
14	快速了解 GenICam	170
15	GenICam 基础信息	171
16	HALCON——连接相机、调节参数、采集图像	173
17	HALCON——手眼标定	184
18	HALCON——设置 IP 地址	194

<b>19 HALCON——仅获取 Z 向值</b>	<b>197</b>
<b>20 HALCON——获取纹理点云</b>	<b>199</b>
<b>21 参考手册</b>	<b>201</b>
<b>22 故障排除</b>	<b>208</b>
<b>23 扩展阅读</b>	<b>220</b>

欢迎使用 Mech-Eye 工业级 3D 相机！Mech-Eye 工业级 3D 相机，需与 Mech-Eye SDK 配合使用，获取并调整 2D 图、深度图及点云。

Mech-Eye SDK 包含 Mech-Eye API 和 Mech-Eye Viewer，其中 Mech-Eye Viewer 是一款基于 Mech-Eye API 开发的软件，可获取图像及点云信息。

通过 GenICam 标准，相机可适配更多第三方机器视觉软件，例如 HALCON（MVTec）和 VisionPro（Cognex），也可配合使用获取 2D 图、深度图及点云。

本部分内容将分为以下几个部分介绍：

- 开始使用
- Mech-Eye Viewer 使用说明
- Mech-Eye API 使用说明
- GenICam
- 更多支持

---

## 知识库

### 相机

[相机选型](#)

[相机使用指南](#)

[技术参数](#)



### 快速使用

[Mech-Eye SDK 2.1 更新说明](#)

[Mech-Eye SDK 安装指南](#)

[Mech-Eye Viewer 快速使用](#)

[Mech-Eye API 快速入门](#)

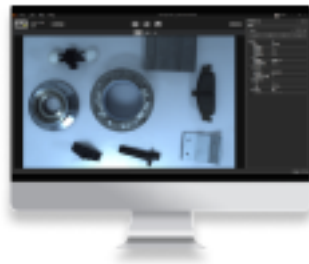


### Mech-Eye Viewer

界面介绍

使用指南

内置工具



### Mech-Eye API

快速入门

例程实操指南

API——固件升级工具

Mech-Eye API 参考手册



### GenICam

GenICam 基础信息

HALCON——连接相机、调节参数、采集图像

HALCON——手眼标定  
参考手册



更多支持  
搜索不到相机  
搜索到但无法连接相机  
结构光相机工作原理



您的反馈对帮助我们提供准确、高质量的信息至关重要。如果您对文档有任何意见，请通过以下方式联系我们：

- 使用 Mech-Mind Docs 站点上方的 [留言板](#) 功能。
- 请将您的意见发送到邮箱：[docs@mech-mind.net](mailto:docs@mech-mind.net)。

---

## 相机使用指南

---

Mech-Eye 工业级 3D 相机（以下简称相机）是梅卡曼德机器人研制的工业级 3D 相机。通过本使用说明书可了解相机的基本信息、简单操作与维护保养。使用前请仔细阅读使用说明书，减少事故发生。

---

查看以下内容，了解 [相机型号及适用场景](#)。

[相机选型](#)

---

查看以下内容，了解 [相机包装清单](#)。

[包装清单](#)

---

查看以下内容，了解 [相机指示灯及接口说明](#)。

[功能示意图](#)

---

查看以下内容，了解 [如何安装相机](#)。

[相机安装](#)

---

查看以下内容，了解 [如何连接相机与工控机](#)。

[相机和工控机连接](#)

---

查看以下内容，了解 [技术参数](#)。

---

查看以下内容，了解相机的安全使用须知与法规要求。

[安全须知与法规要求](#)

## 1.1 相机选型

本章主要介绍本公司的相机版本、型号及各型号的特点与适用场景，请根据需要选择相机型号。

### 1.1.1 相机介绍

本公司相机在不断升级优化中，目前主要介绍 V3、V4 代相机。

#### V4 相机基础信息

型号	相机类型	2D 相机个数	分辨率
PRO S	DLP 相机	单目	1920 × 1200
PRO M		单目	1920 × 1200
NANO		单目	1280 × 1024
UHP-140		双单目 *	2048 × 1536
LSR L	激光相机	三目	深度：2048 × 1536 RGB：4000 × 3000、2000 × 1500
DEEP			深度：2048 × 1536、1024 × 768 RGB：2000 × 1500

\* UHP-140 相机既可以使用两台 2D 相机获取图像，也可以使用一台 2D 相机获取图像，因此称为双单目相机。

提示：如需查看相机参数，请查看[技术参数](#)。

#### V3 相机基础信息

型号	相机类型	2D 相机个数	分辨率
Pro S Enhanced	DLP 相机	单目	1920 × 1200
Pro M Enhanced		单目	1920 × 1200
Log S		单目	1280 × 1024
Log M		单目	1280 × 1024
Nano		单目	1280 × 1024
Pro XS		双目	1280 × 1024
Deep		双目	2048 × 1536
Pro L Enhanced		双目	4096 × 3000
Laser L		激光相机	双目
Laser L Enhanced	双目		4096 × 3000



提示：如需查看相机参数，请查看基本参数（V3）。

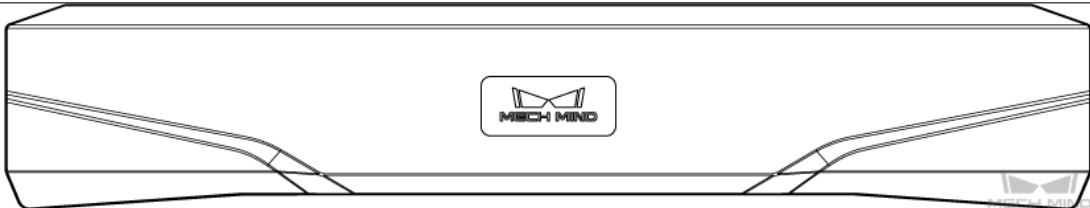
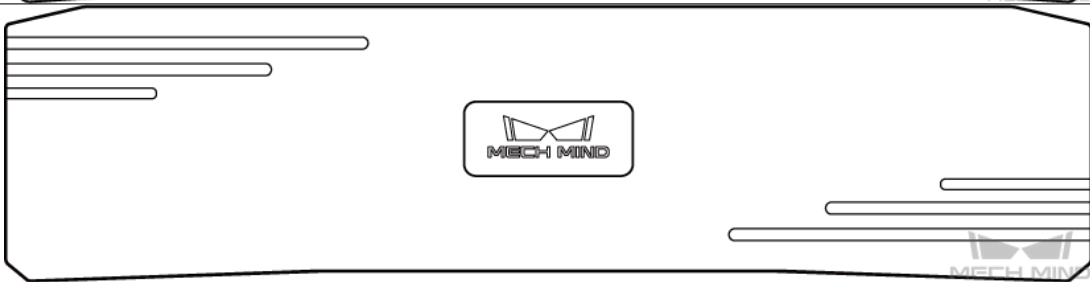
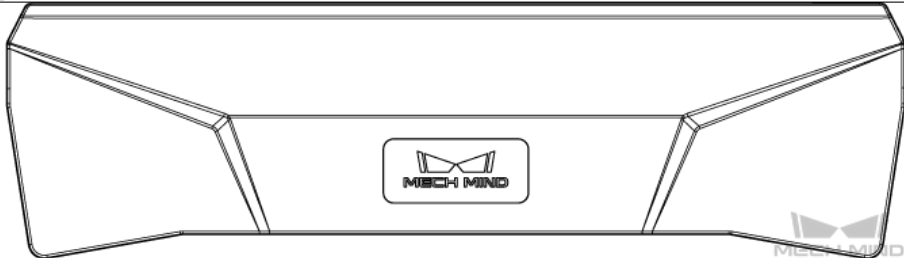
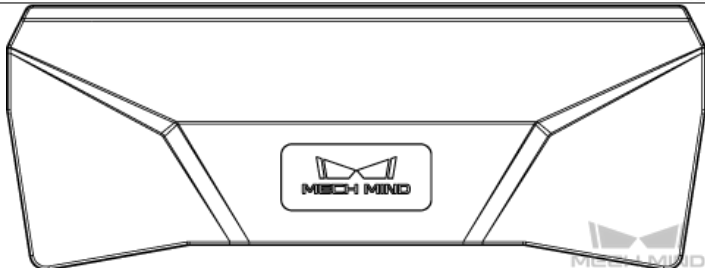
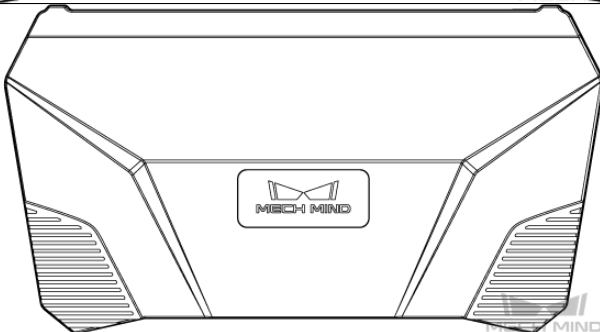
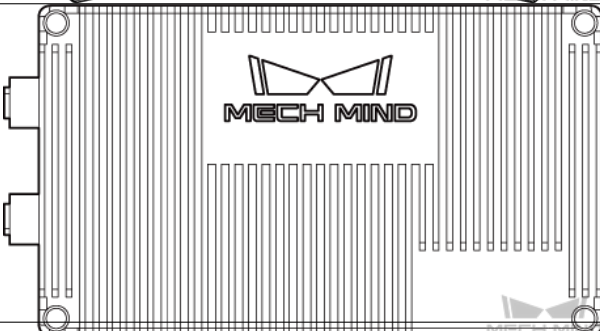
### 1.1.2 特点与适用场景

针对不同场景，本公司研制了不同型号的相机，不同相机型号的特点与适用场景详见下表。

相机型号	特点	适用场景
UHP-140 (V4)	微米级精度，使用自研融合成像算法。处理高亮工件时，成像效果优异。	适用于汽车零部件生产或组装等工艺中，位置度、间隙、面差等检测/量测类应用场景。
LSR L (V4)	高精度，大视野，抗环境光性能优异。	适用于各类制造车间等易被环境光干扰的场景。
Laser L (V3)		
Laser L Enhanced (V3)		
NANO (V4)	超小体积，超高精度，抗环境光性能优异。	适合安装在机械臂上使用，适用于定位装配、高精度抓取等精细化作业场景。
Nano (V3)		
Pro XS (V3)		
DEEP (V4)	大视野，大景深，速度快，可对纸箱、麻袋、周转箱等物体生成完整、细致、精确、颜色准确的点云数据。	适用于拆码垛等典型物流场景。
Deep (V3)		
Pro L Enhanced (V3)		
PRO S (V4)	精度高，速度快，抗环境光性能优异，运行更稳定，可选彩色版本。	适用于无序抓取、定位、装配、学术研究等对精度要求较高的中距离作业场景。
Pro S Enhanced (V3)		
PRO M (V4)		
Pro M Enhanced (V3)		
Log M (V3)	速度快，适合中远距离。	专为物流场景设计，适合货品拣选、快递供包等物流场景。
Log S (V3)	速度快，适合中近距离。	专为物流场景设计，适合货品拣选、快递供包等物流场景。

## 1.2 包装清单

相机

相机型号	示意图
LSR L	
DEEP	
PRO M	
PRO S	
UHP-140	
NANO	

配件盒



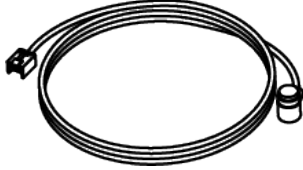
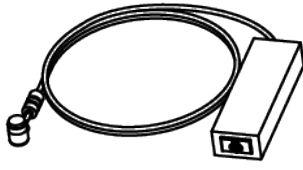
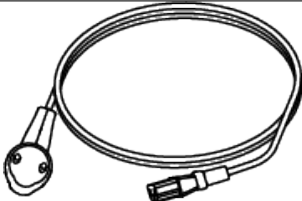
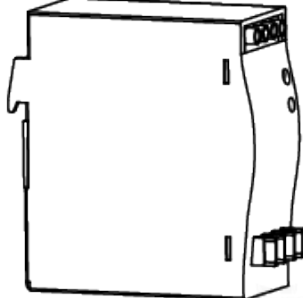
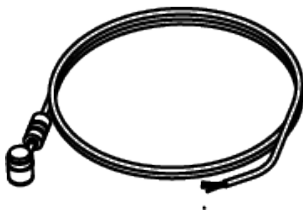
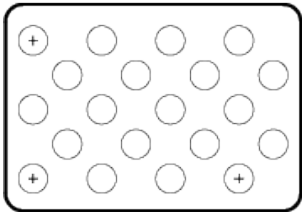
使用说明书



提示:

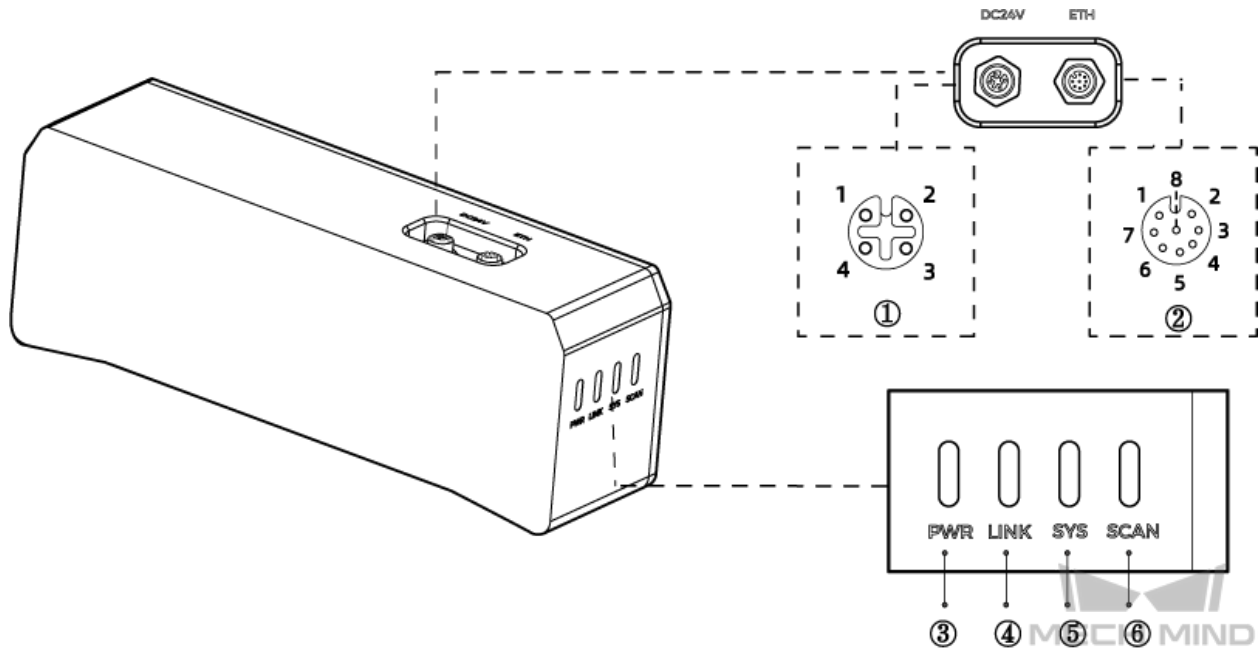
- 相机发货时，L 型转接件已安装。
- 使用前，请确认包装完好，相机功能正常，配件无缺失。

### 1.2.1 线缆与配件

<p>网线</p> 	<p>适配器</p> 	<p>AC 电源线</p> 
<p>导轨电源</p> 	<p>DC 电源线</p> 	<p>标定板</p> 

提示：以上线缆配件均为可选配件，请根据实际情况选择类型和长度。具体使用方法见[相机和工控机连接](#)。

### 1.3 功能示意图



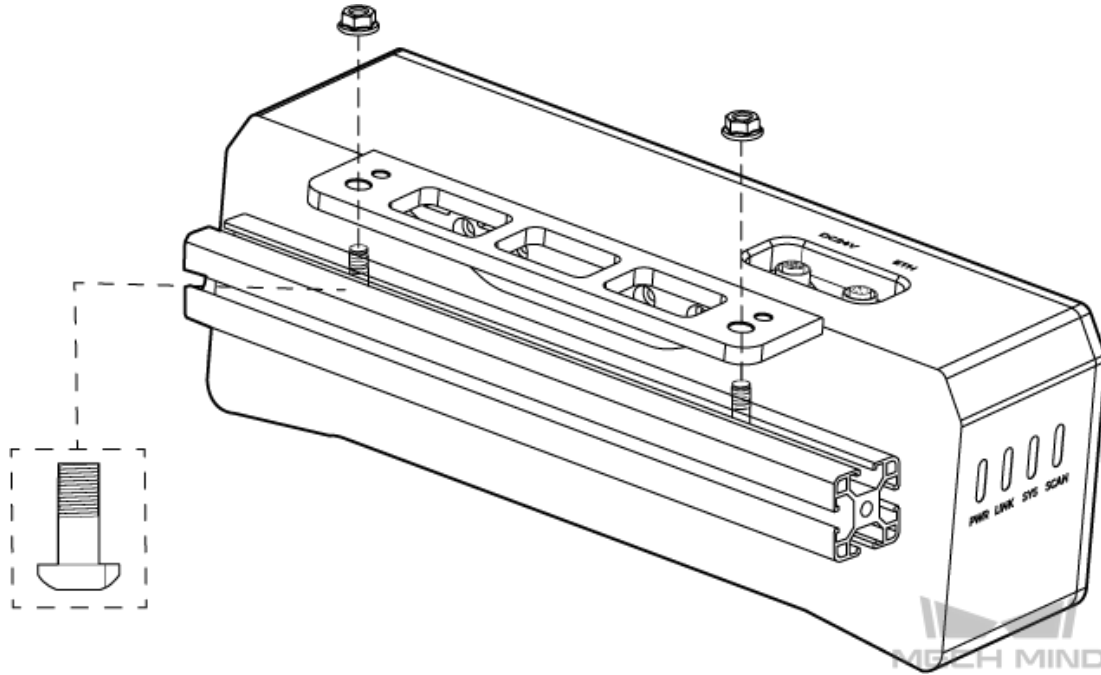
No.	名称	功能
1	DC 24V 电源接口	1: GND 2: GND 3: 24 V DC 4: 24 V DC
2	ETH 网口	1: MD3_P 2: MD2_N 3: MD2_P 4: MD0_P 5: MD1_P 6: MD0_N 7: MD3_N 8: MD1_N
3	PWR 指示灯	熄灭: 电源未连接 绿灯亮: 电压正常 黄灯亮: 电压报警, 但相机可使用 红灯亮: 电压异常, 无法使用
4	LINK 指示灯	熄灭: 网络未连接 绿灯闪烁: 数据传输中 绿灯常亮: 无数据传输
5	SYS 指示灯	熄灭: 系统未运行 绿灯常亮: 系统启动中 绿灯闪烁: 系统正常运行 黄灯闪烁: 系统报警, 但相机可使用 红灯闪烁: 系统错误, 无法使用
6	SCAN 指示灯	常亮: 采集和计算中 熄灭: 未采集

提示: 以上图示仅供参考, 请以实物为准。

## 1.4 相机安装

### 1.4.1 使用 L 型转接件安装

使用扳手拧紧两颗螺母，固定相机，如下图所示。

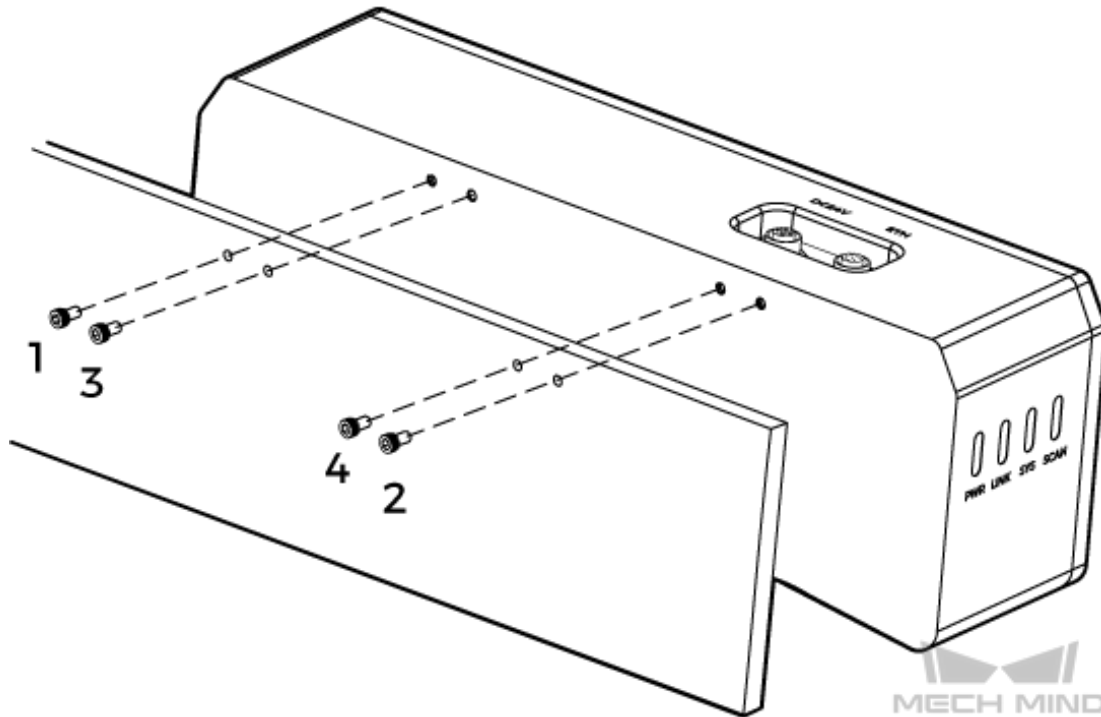


提示:

- 发货时，L 型转接件已安装至相机背部。
- 请自行准备扳手。

### 1.4.2 使用相机螺纹孔安装

安装相机，使用扳手按顺序先预紧，后拧紧螺钉，如下图所示。



提示:

- 安装前需使用扳手拆卸 L 型转接件。
- 请自行准备扳手。

## 1.5 相机和工控机连接

注意:

- 连接时，请最后接通电源。接通电源后，PWR 指示灯绿色常亮；如指示灯不正常，请及时联系工作人员。
- 拧紧螺母时，推荐扭矩：16N·m。
- 导轨或连接导轨的配电箱应可靠接地。如有多台，安装时应保持一定间距。
- 当相机安装在机械臂或其他移动装置上时，连接相机端的 DC 电源线与网线需妥善固定，防止拉扯损坏线缆或插头。

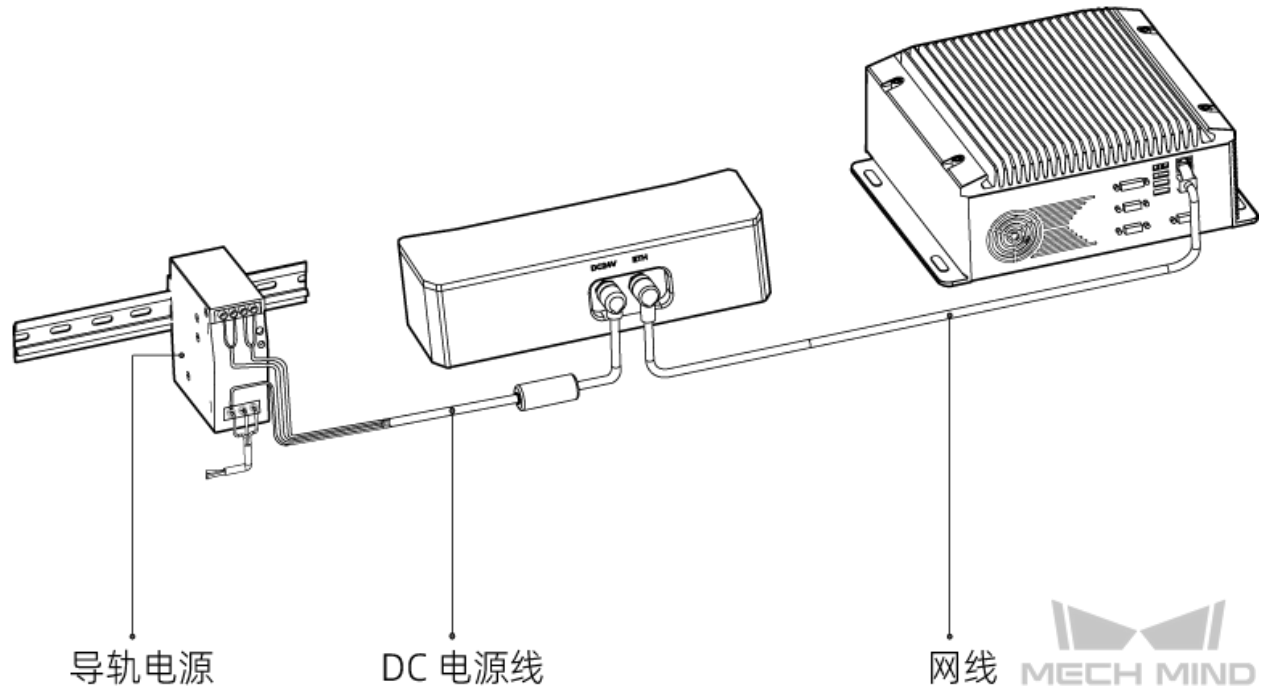
## 1.5.1 直接连接

### 网线

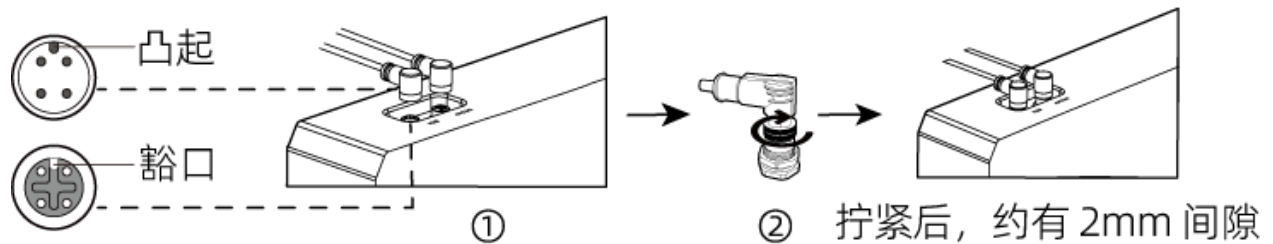
一端连接工控机，另一端连接相机 ETH 网口。

### DC 电源线

电源适配器/导轨电源的 DC 电源线插入 DC 24V 电源接口中，如下图所示。



### 电源线接口

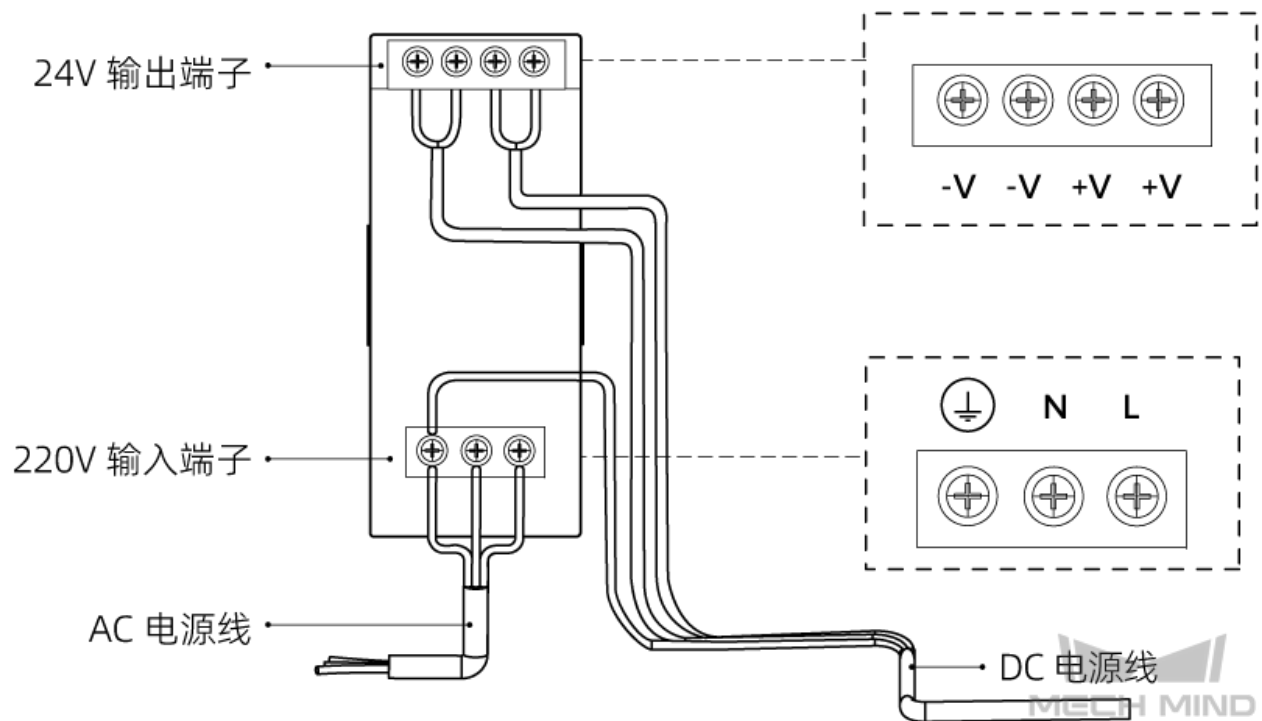


1. 将电源线插头的凸起对准电源接口的豁口插入。
2. 拧紧紧固螺母。

提示：网线接口与电源线接口连接方法相同。



## 导轨电源



导轨电源接线连接时，插头需接入对应的输入/输出电压端子，如上图所示。

- AC 电源线共有三股接线插头，分别为：L、N、PE (⊕)。
- DC 电源线（24V）共有三股接线插头，分别为：+V、-V、PE (⊕)。

**警告：** 导轨电源的接地端子必须接地！导轨电源需放在配电箱中使用。

**注解：** 如需连接多台相机或多台工控机，可使用交换机连接。

## 1.6 技术参数

V4 相机技术参数，请查阅：

## 1.6.1 LSR L

### 基本参数

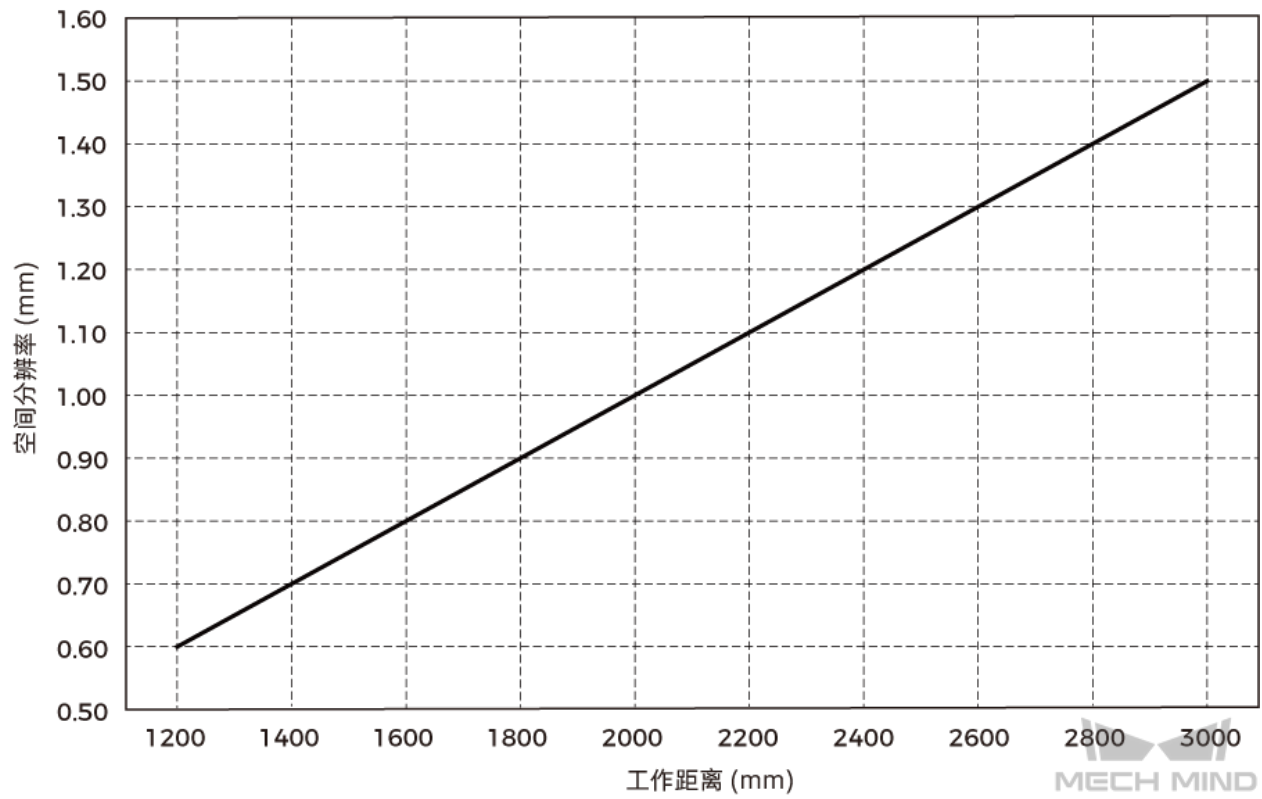
产品名称	Mech-Eye 工业级 3D 相机
型号	LSR L
推荐工作距离范围	1200~3000mm
近端视场	1200 × 1000mm @ 1.2m
远端视场	3000 × 2400mm @ 3.0m
深度图分辨率	2048 × 1536
RGB 分辨率	4000 × 3000 / 2000 × 1500
Z 向单点重复精度 ( $\sigma$ ) <sup>[1]</sup>	0.5mm @ 3m
VDI/VDE 测量精度 <sup>[2]</sup>	1.0mm @ 3m
典型采集时间	0.5~0.9s
重量	约 2.9kg
基线长度	约 380mm
尺寸	约 459 × 77 × 86mm
光源	红色激光 (638nm, 2 类)
工作温度范围	-10~45°C
通讯接口	千兆以太网
输入	24V DC, 3.75A
安全与电磁兼容	CE/FCC/VCCI/UKCA/KC
防护等级 <sup>[3]</sup>	IP65
散热	被动

[1] 单点 Z 值 100 次测量的一倍标准差，测量目标为陶瓷板。

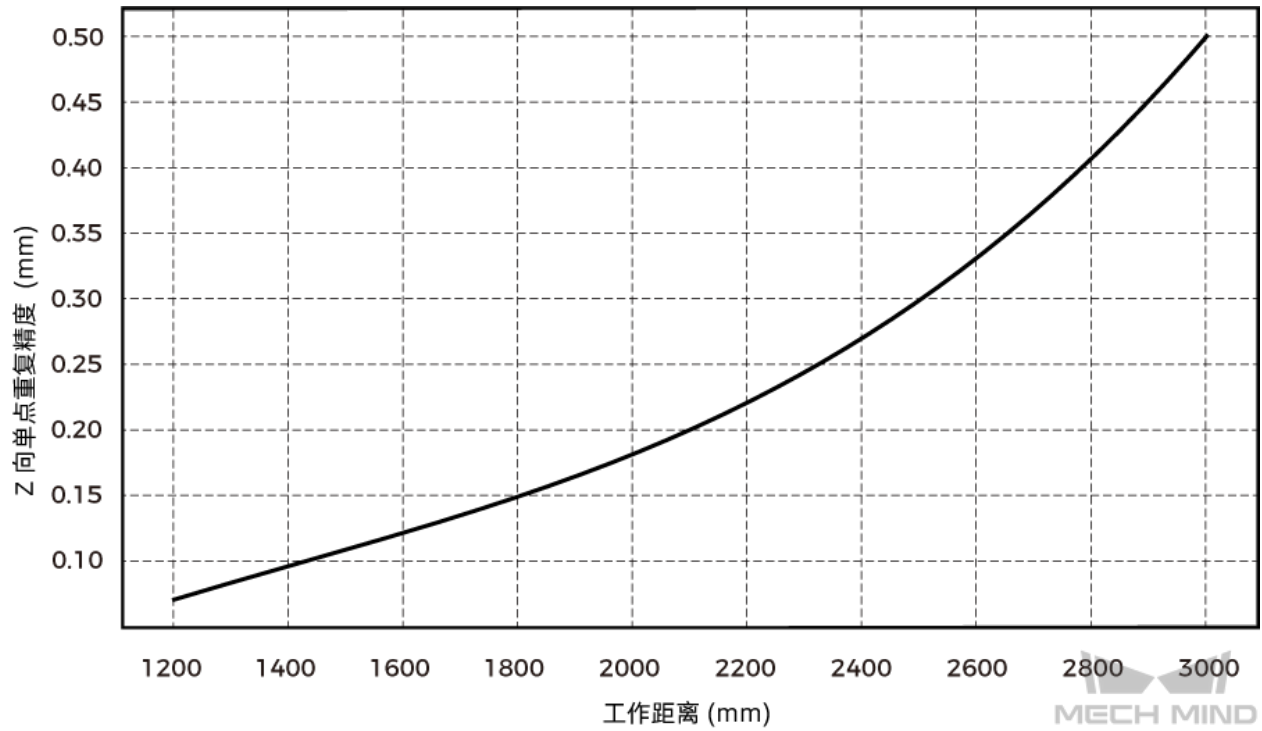
[2] 基于 VDI/VDE 2634 Part II 标准。

[3] 根据 IEC 60529 标准测试。其中，6 代表防尘等级，5 代表防水等级。

### 空间分辨率

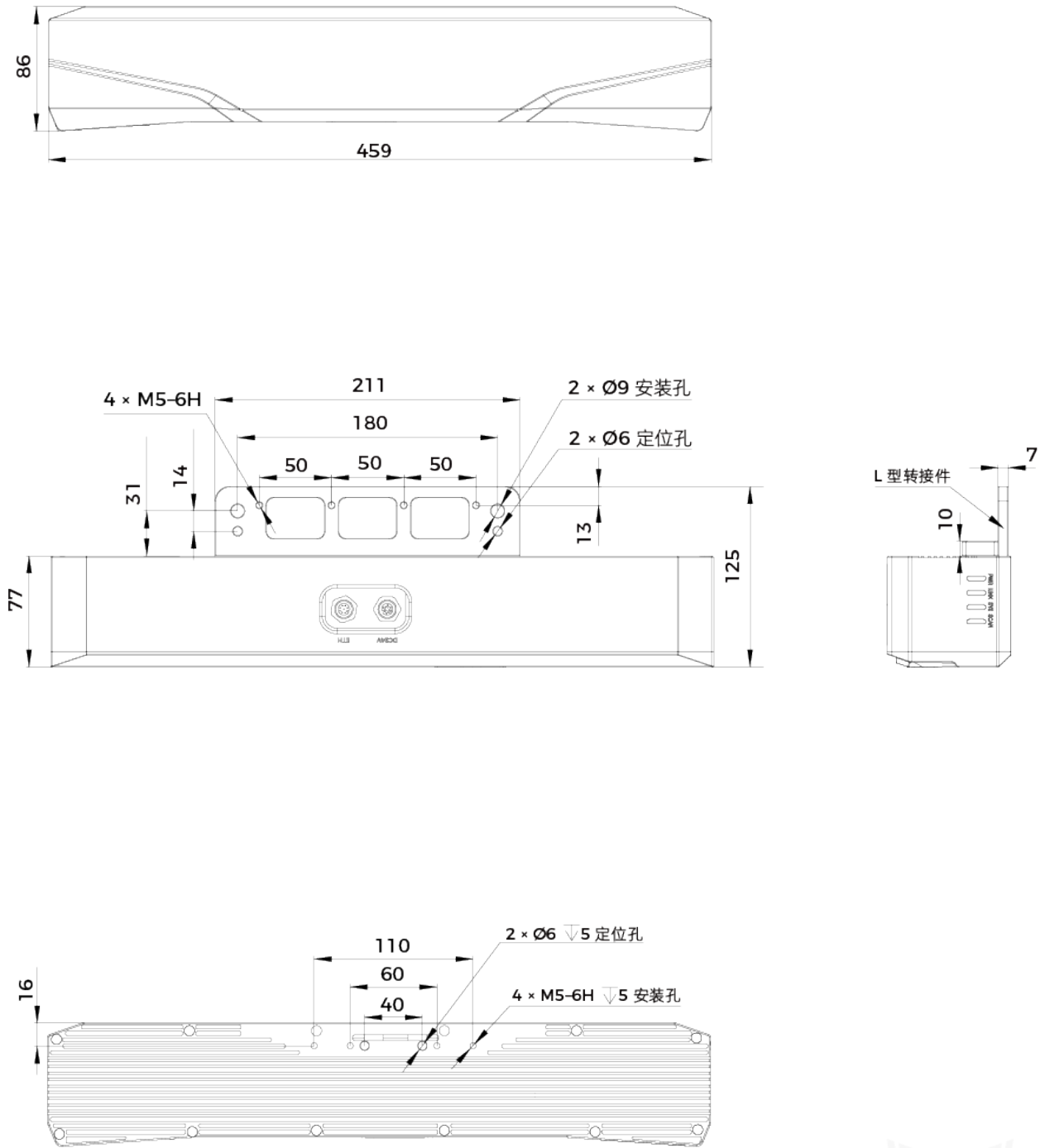


### Z 向单点重复精度



### 相机尺寸

单位: mm



相机背面图（无 L 型转接件）

## 1.6.2 DEEP

### 基本参数

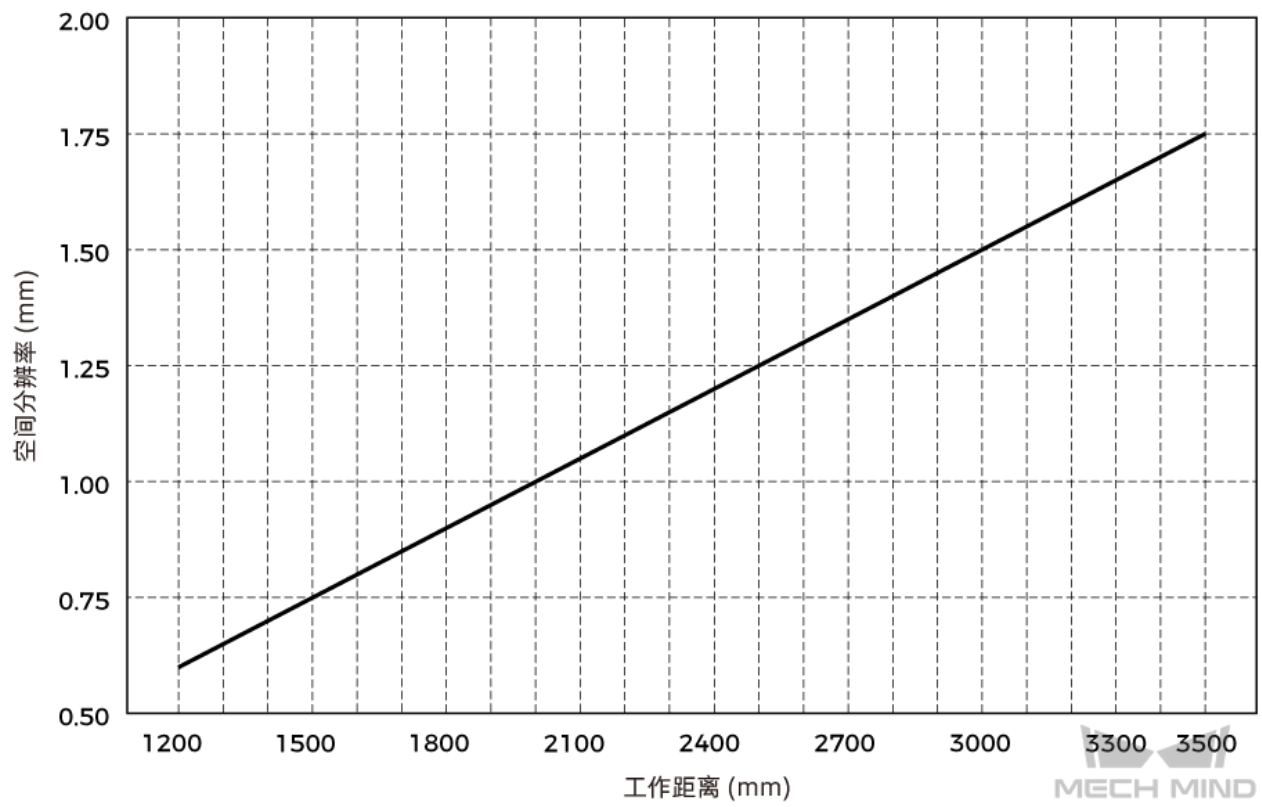
产品名称	Mech-Eye 工业级 3D 相机
型号	DEEP
推荐工作距离范围	1200~3500mm
近端视场	1200 × 1000mm @ 1.2m
远端视场	3500 × 2800mm @ 3.5m
深度图分辨率	2048 × 1536 / 1024 × 768
RGB 分辨率	2000 × 1500
Z 向单点重复精度 ( $\sigma$ ) <sup>[1]</sup>	1.0mm @ 3m
VDI/VDE 测量精度 <sup>[2]</sup>	3.0mm @ 3m
典型采集时间	0.5~0.9s
重量	约 2.4kg
基线长度	约 300mm
尺寸	约 366 × 77 × 92mm
光源	红色激光 (638nm, 2 类)
工作温度范围	-10~45°C
通讯接口	千兆以太网
输入	24V DC, 3.75A
安全与电磁兼容	CE/FCC/VCCI/UKCA/KC
防护等级 <sup>[3]</sup>	IP65
散热	被动

[1] 单点 Z 值 100 次测量的一倍标准差，测量目标为陶瓷板。

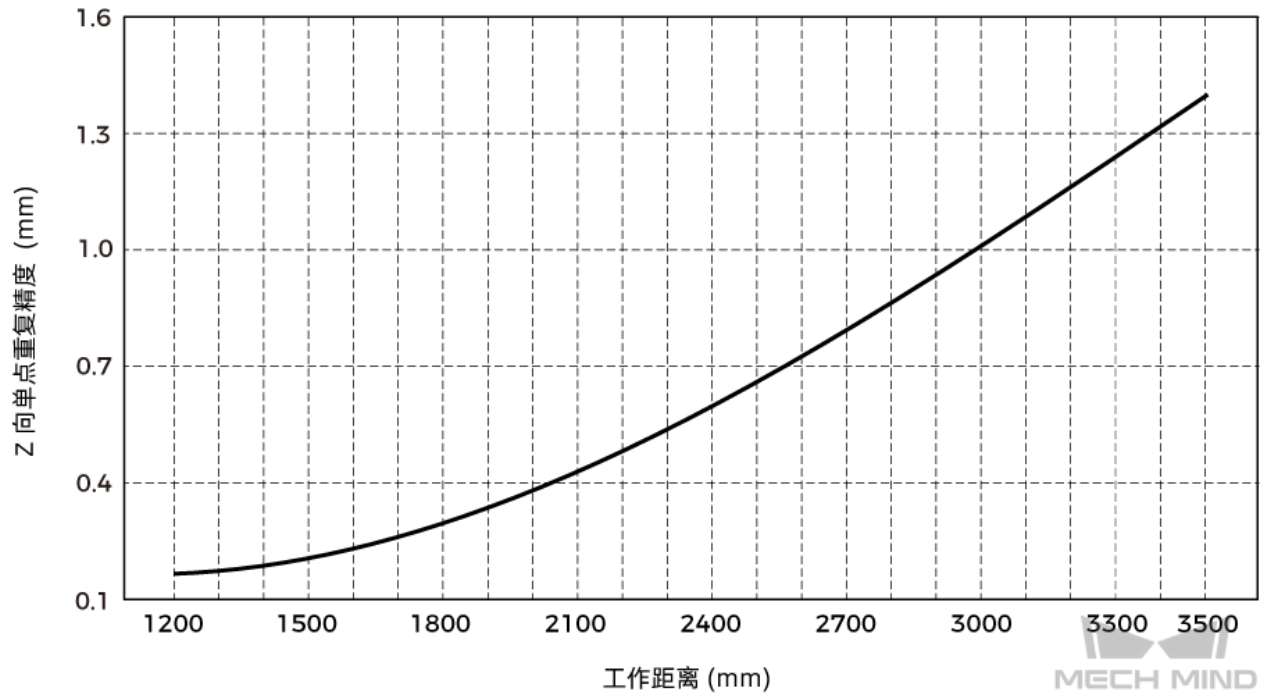
[2] 基于 VDI/VDE 2634 Part II 标准。

[3] 根据 IEC 60529 标准测试。其中，6 代表防尘等级，5 代表防水等级。

### 空间分辨率



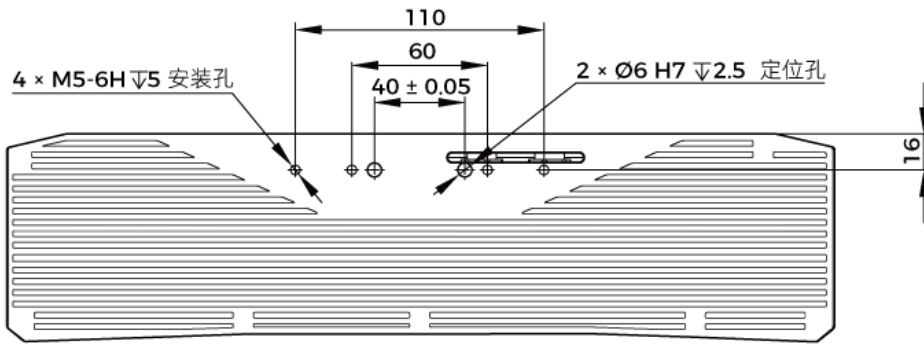
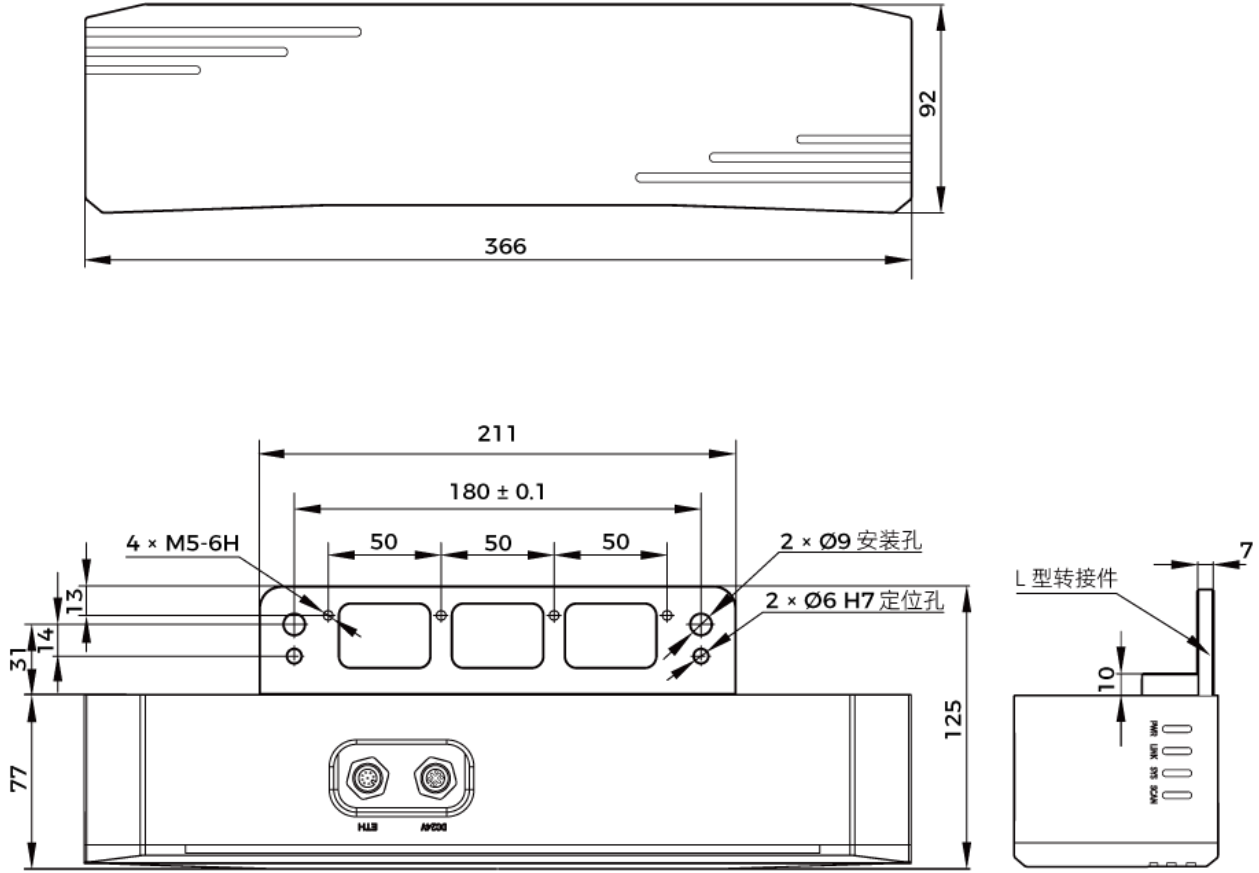
Z 向单点重复精度



相机尺寸

单位: mm





相机背面图 (无 L 型转接件)

### 1.6.3 PRO S & PRO M

#### 基本参数

产品名称	Mech-Eye 工业级 3D 相机	
型号	PRO S	PRO M
推荐工作距离范围	500~1000mm	1000~2000mm
近端视场	370 × 240mm @ 0.5m	800 × 450mm @ 1m
远端视场	800 × 450mm @ 1m	1500 × 890mm @ 2m
分辨率	1920 × 1200	
像素数	2.3MP	
Z 向单点重复精度 ( $\sigma$ ) <sup>[1]</sup>	0.05mm @ 1m	0.2mm @ 2m
VDI/VDE 测量精度 <sup>[2]</sup>	0.1mm @ 1m	0.2mm @ 2m
重量	约 1.6kg	约 1.9kg
基线长度	约 180mm	约 270mm
尺寸	约 265 × 57 × 100mm	约 353 × 57 × 100mm
典型采集时间	0.3~0.6s	
光源	蓝光 LED (459nm, RG2)	
工作温度范围	0~45°C	
通讯接口	千兆以太网	
输入	24V DC, 3.75A	
安全与电磁兼容	CE/FCC/VCCI/UKCA/KC	
防护等级 <sup>[3]</sup>	IP65	
散热	被动	

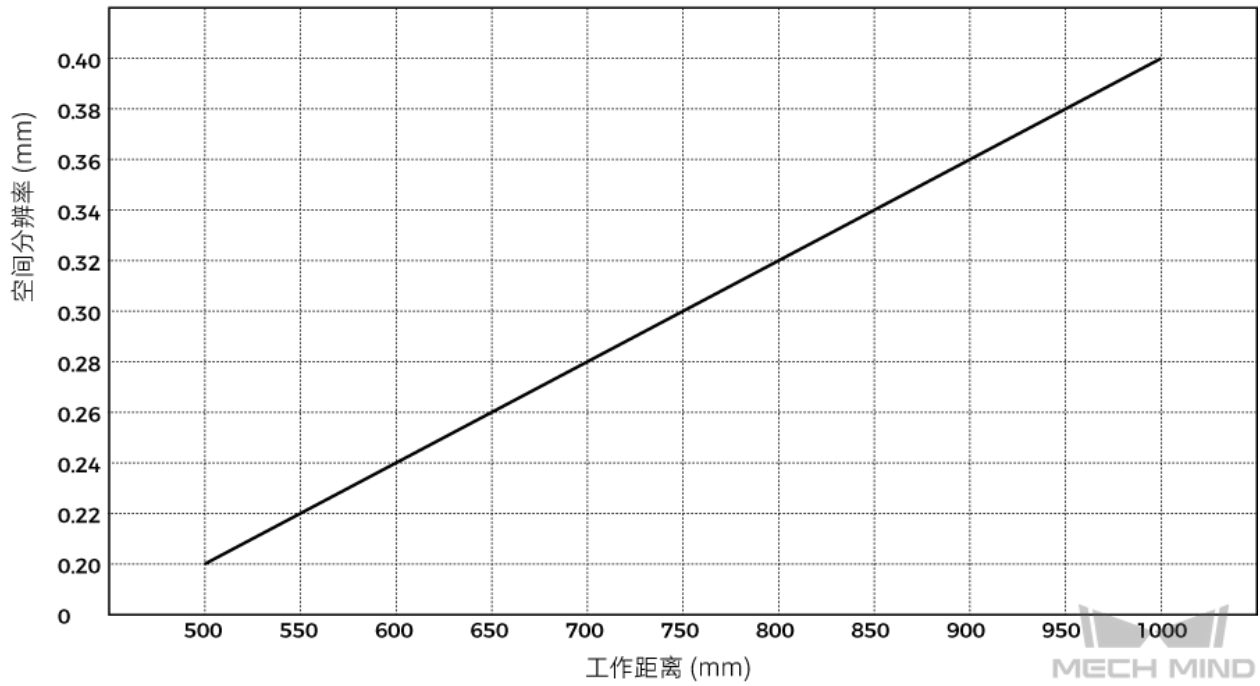
[1] 单点 Z 值 100 次测量的一倍标准差，测量目标为陶瓷板。

[2] 基于 VDI/VDE 2634 Part II 标准。

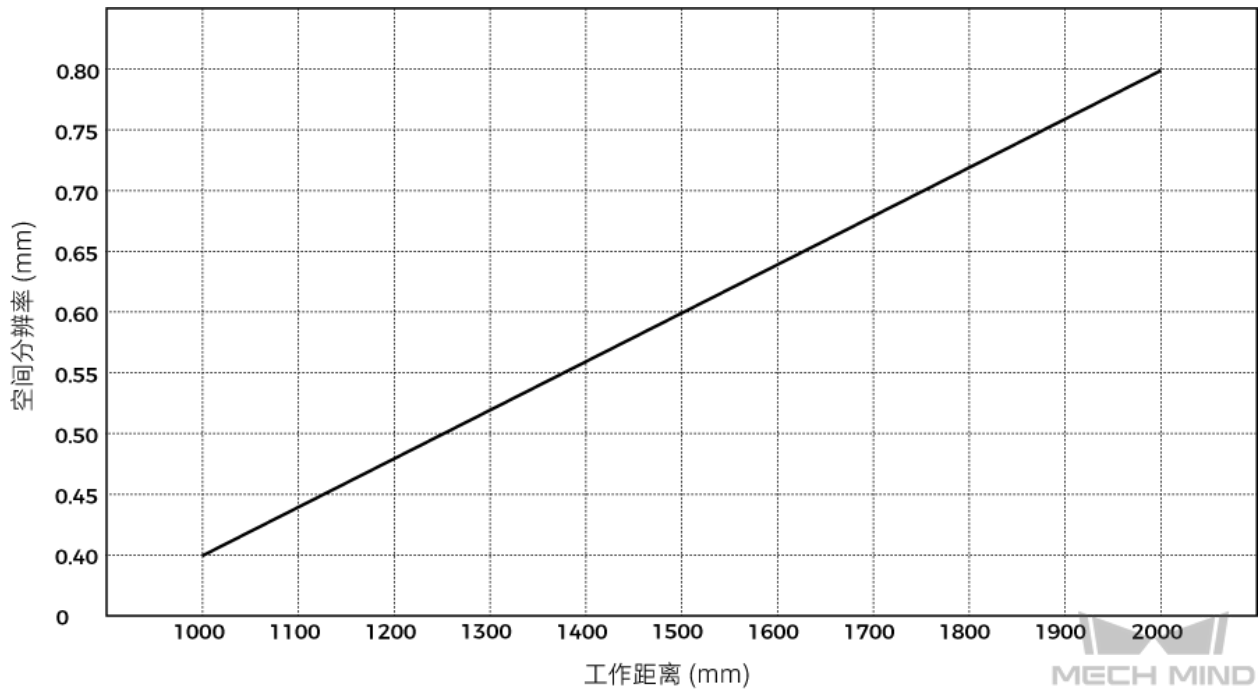
[3] 根据 IEC 60529 标准测试。其中，6 代表防尘等级，5 代表防水等级。

#### 空间分辨率

PRO S

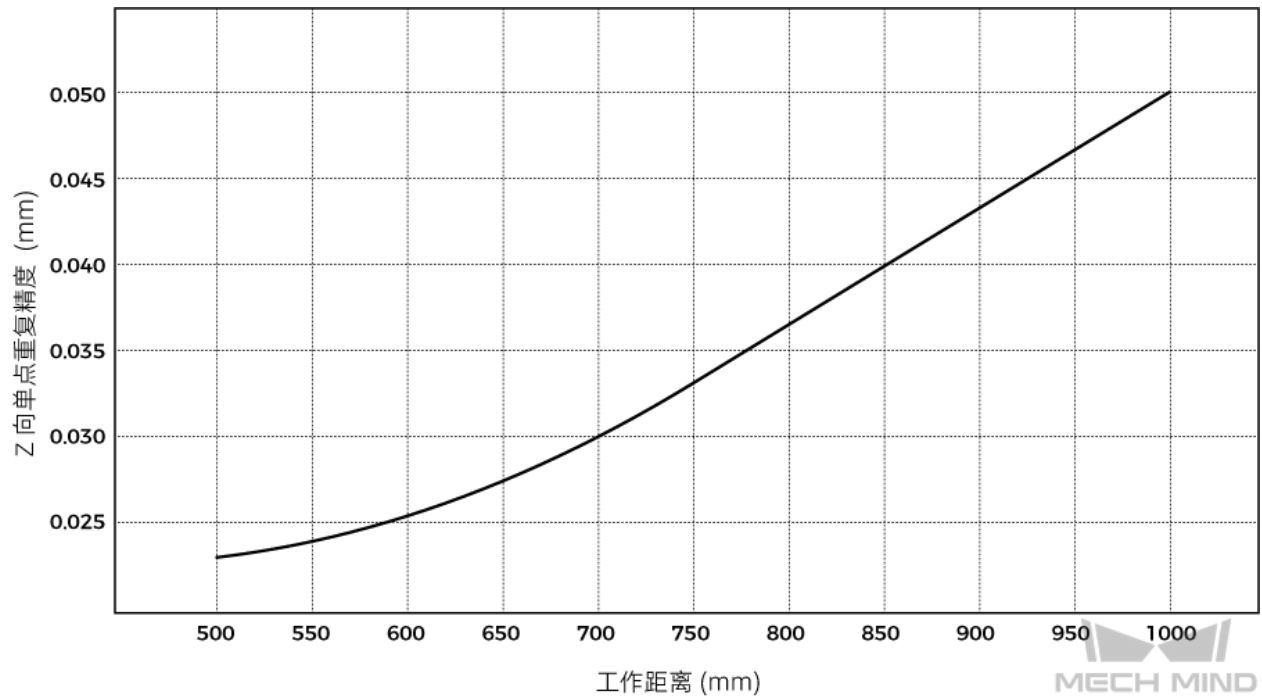


PRO M

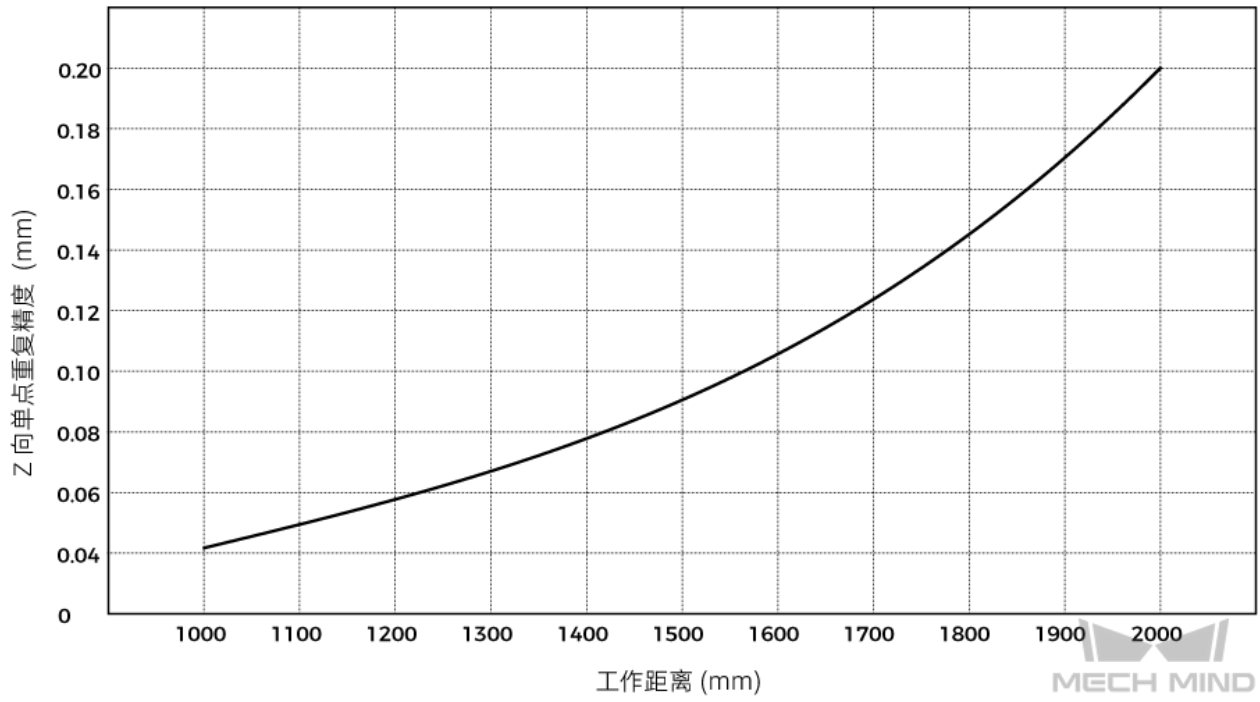


### Z 向单点重复精度

#### PRO S



PRO M

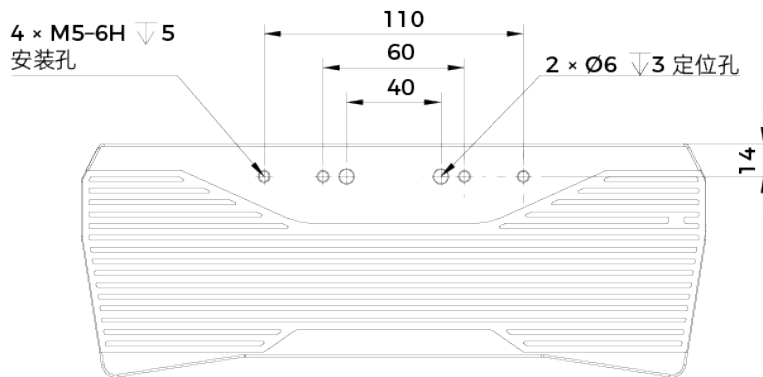
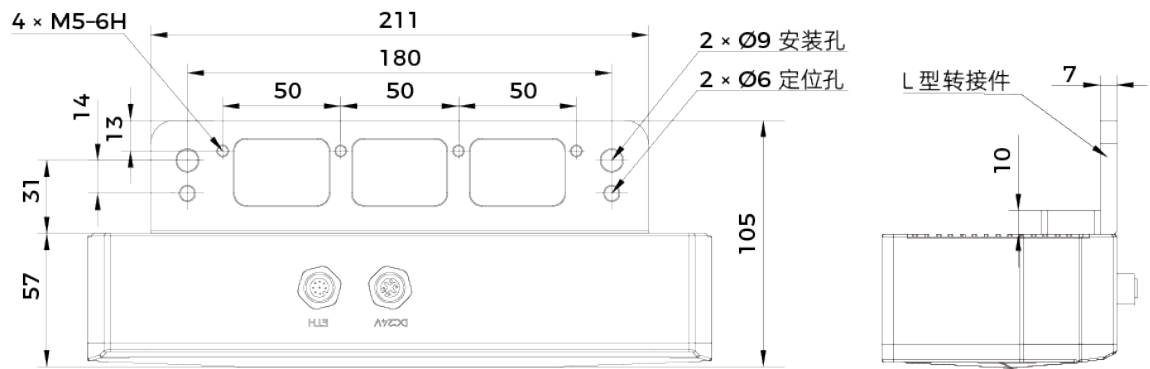
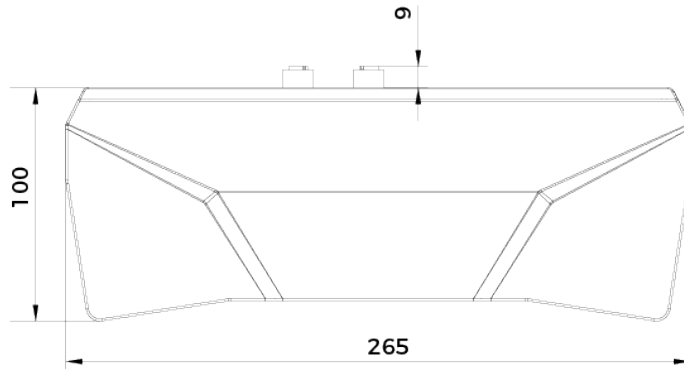


相机尺寸

单位: mm

PRO S

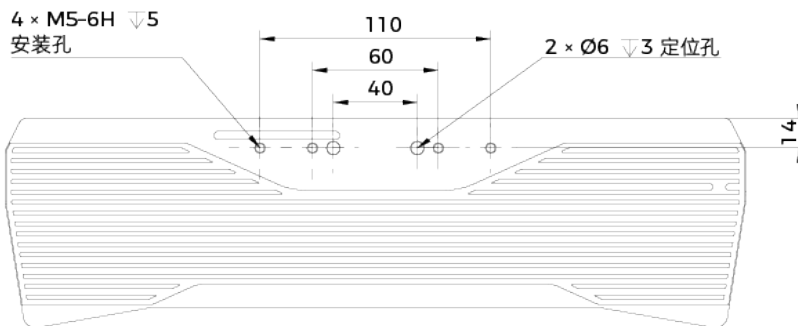
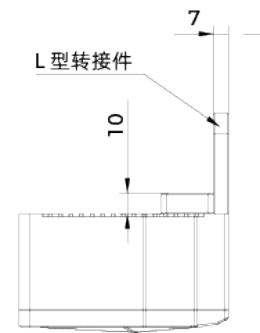
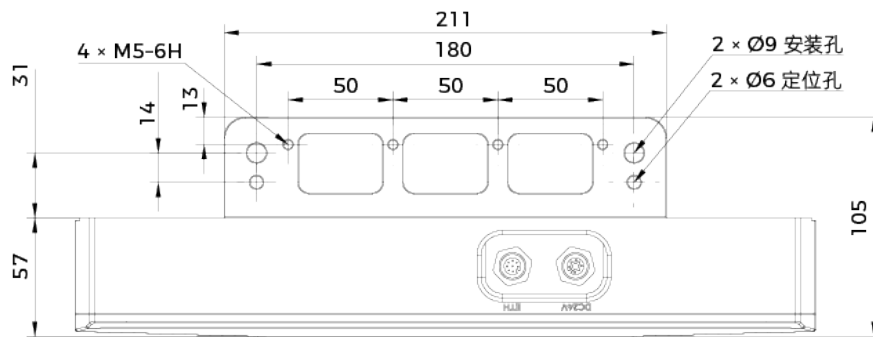
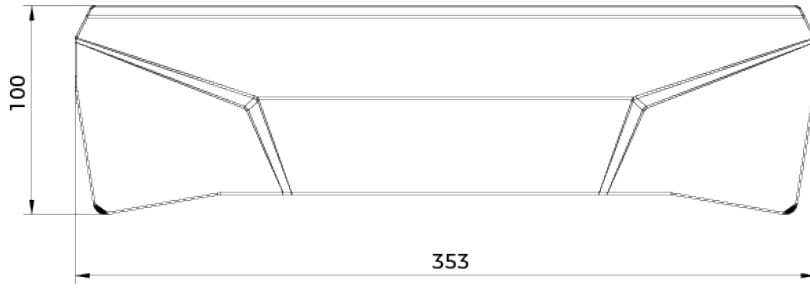
PRO S



相机背面图 (无 L 型转接件)



PRO M



相机背面图（无 L 型转接件）



## 1.6.4 UHP-140

### 基本参数

产品名称	Mech-Eye 工业级 3D 相机
型号	UHP-140
推荐工作距离范围	300 ± 20mm
近端视场	135 × 90mm @ 280mm
远端视场	150 × 100mm @ 320mm
分辨率	2048 × 1536
像素数	3MP
Z 向单点重复精度 ( $\sigma$ ) <sup>[1]</sup>	2.6 $\mu\text{m}$ @ 0.3m
Z 向区域重复精度 ( $\sigma$ ) <sup>[2]</sup>	0.09 $\mu\text{m}$ @ 0.3m
VDI/VDE 测量精度 <sup>[3]</sup>	0.03mm @ 0.3m
典型采集时间	0.6~0.9s
重量	约 1.9kg
基线长度	约 80mm
尺寸	约 260 × 65 × 142mm
光源	蓝光 LED (459nm, RG2)
工作温度范围	0~45°C
通讯接口	千兆以太网
输入	24V DC, 3.75A
安全与电磁兼容	CE/FCC/VCCI/UKCA/KC
防护等级 <sup>[4]</sup>	IP65
散热	被动

[1] 单点 Z 值测量 100 次的一倍标准差，测量目标为陶瓷板。

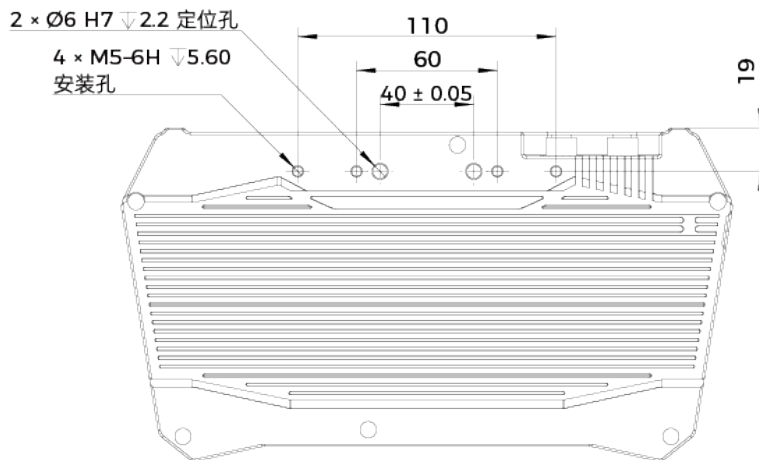
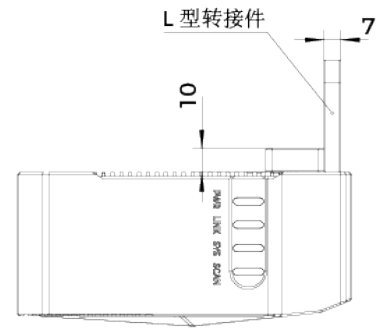
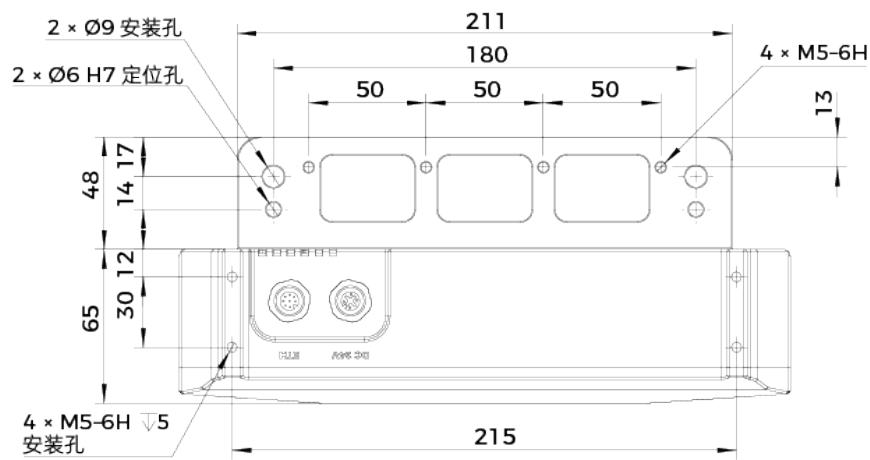
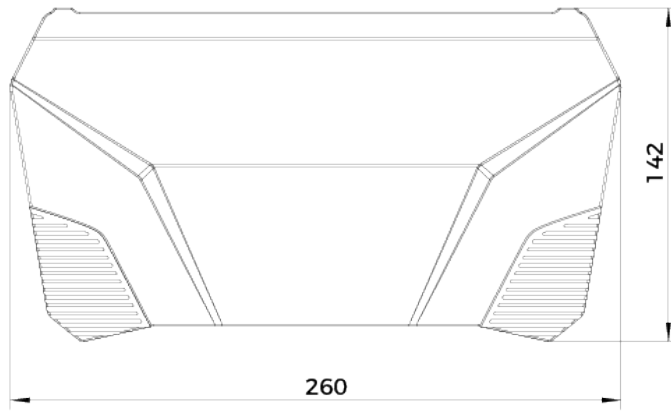
[2] 两个区域的 Z 均值差测量 100 次的一倍标准差。测量目标为陶瓷板。

[3] 基于 VDI/VDE 2634 Part II 标准。

[4] 根据 IEC 60529 标准测试。其中，6 代表防尘等级，5 代表防水等级。

## 相机尺寸

单位: mm



相机背面图（无 L 型转接件）

## 1.6.5 NANO

### 基本参数

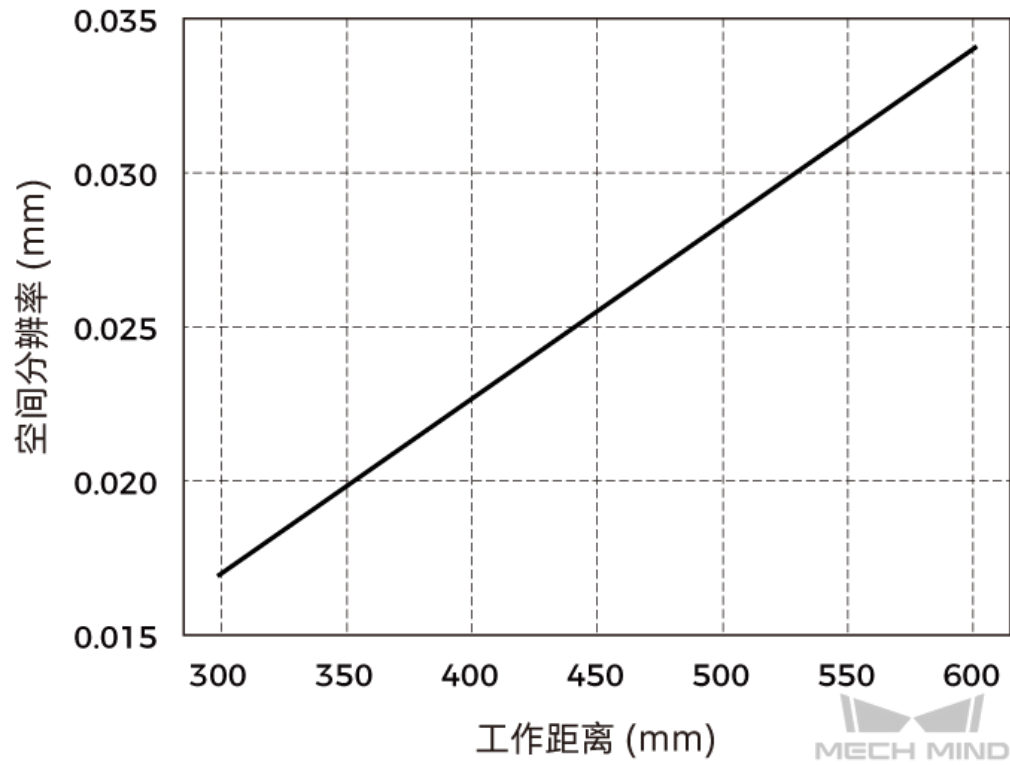
产品名称	Mech-Eye 工业级 3D 相机
型号	NANO
推荐工作距离范围	300~600mm
近端视场	220 × 150mm @ 0.3m
远端视场	440 × 300mm @ 0.6m
分辨率	1280 × 1024
像素数	1.3MP
Z 向单点重复精度 ( $\sigma$ ) <sup>[1]</sup>	0.1mm @ 0.5m
VDI/VDE 测量精度 <sup>[2]</sup>	0.1mm @ 0.5m
典型采集时间	0.6~1.1s
重量	约 0.7kg
基线长度	约 68mm
尺寸	约 145 × 51 × 85mm
光源	蓝光 LED (459nm, RG2)
工作温度范围	0~45°C
通讯接口	千兆以太网
输入	24V DC, 1.5A
安全与电磁兼容	CE/FCC/VCCI/UKCA/KC
防护等级 <sup>[3]</sup>	IP65
散热	被动

[1] 单点 Z 值 100 次测量的一倍标准差，测量目标为陶瓷板。

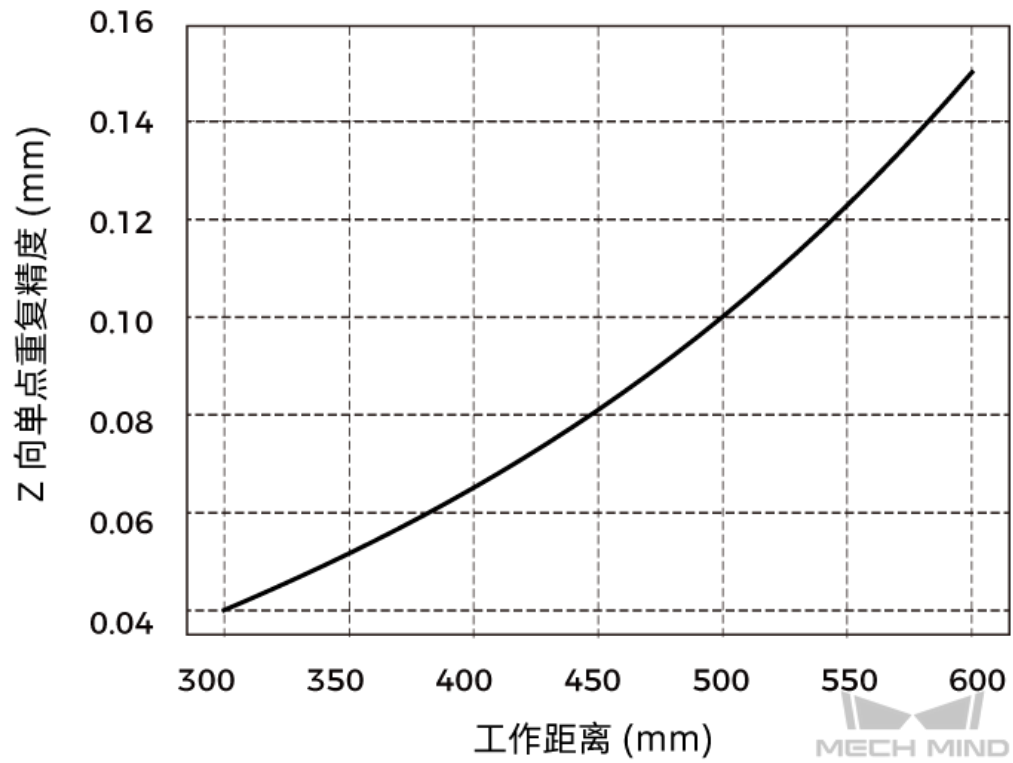
[2] 基于 VDI/VDE 2634 Part II 标准。

[3] 根据 IEC 60529 标准测试。其中，6 代表防尘等级，5 代表防水等级。

空间分辨率

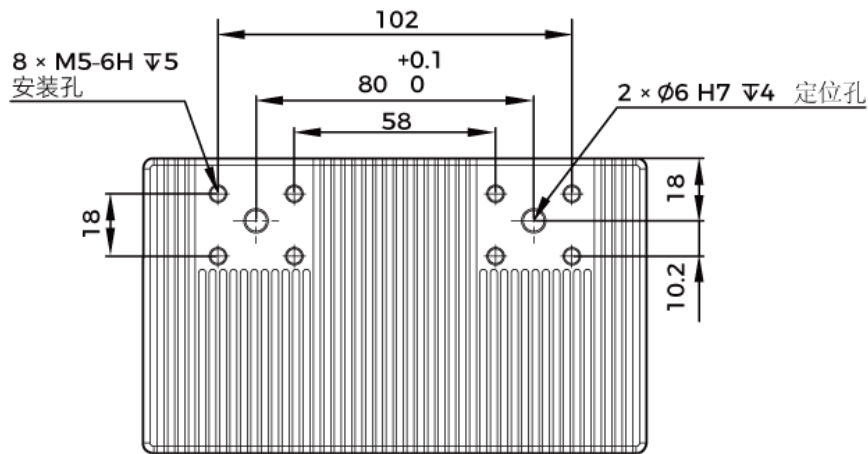
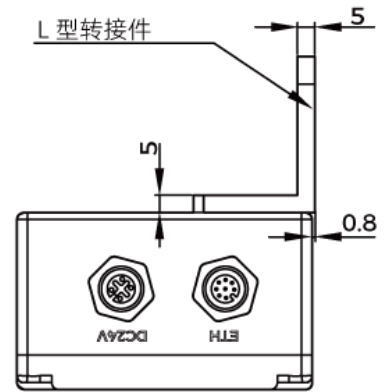
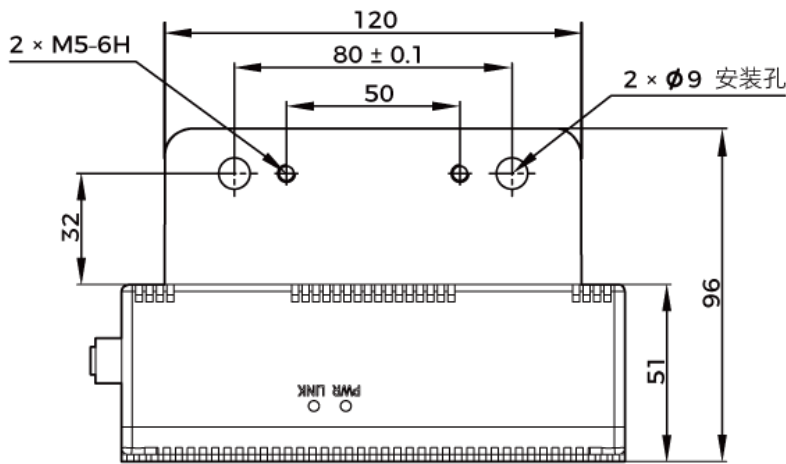
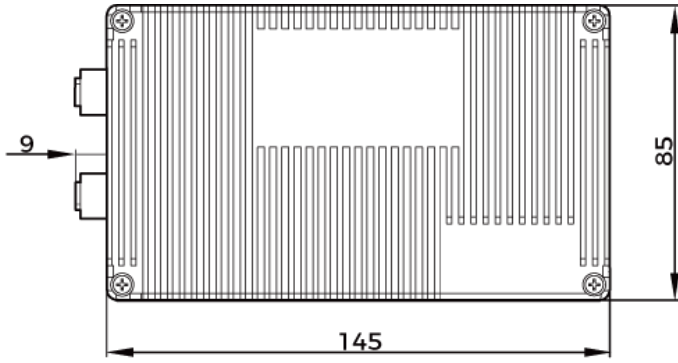


Z 向单点重复精度



相机尺寸

单位: mm



相机背面图 (无 L 型转接件)

V3 相机基本参数，请查阅：

## 1.6.6 基本参数 (V3)

### DLP 相机参数

产品名称	Mech-Eye 工业级 3D 相机				
型号	Pro S Enhanced	Pro M Enhanced	Log S	Log M	Nano
推荐工作距离范围	500~1000mm	800~2000mm	500~1000mm	800~2000mm	300~600mm
近端视场	350 × 220mm @ 0.5m	500 × 350mm @ 0.8m	360 × 250mm @ 0.5m	520 × 390mm @ 0.8m	220 × 160mm @ 0.3m
远端视场	690 × 430mm @ 1.0m	1360 × 860mm @ 2.0m	710 × 490mm @ 1.0m	1410 × 960mm @ 2.0m	430 × 320mm @ 0.6m
分辨率	1920 × 1200	1920 × 1200	1280 × 1024	1280 × 1024	1280 × 1024
像素数	2.3 MP	2.3 MP	1.3 MP	1.3 MP	1.3 MP
Z 向重复精度 (σ)	0.05mm @ 1m	0.2mm @ 2m	0.1mm @ 1m	0.3mm @ 2m	0.1mm @ 0.5m
标定精度	0.1mm @ 1m	0.2mm @ 2m	0.2mm @ 1m	0.3mm @ 2m	0.1mm @ 0.5m
典型采集时间	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
基线长度	150mm	280mm	150mm	280mm	68mm
外形尺寸	约 270 × 72 × 130mm	约 387 × 72 × 130mm	约 270 × 72 × 130mm	约 387 × 72 × 130mm	约 145 × 51 × 85mm
重量	约 2.2kg	约 2.4kg	2.2kg	2.4kg	约 0.7kg
工作温度范围	0~45°C				
通讯接口	以太网				
工作电压	24V DC				
安全和电磁兼容	CE/FCC/VCCI				
防护等级	IP65				
散热	被动				



## 激光相机参数

产品名称	Mech-Eye 工业级 3D 相机	
型号	Laser L	Laser L Enhanced
推荐工作距离范围	1500~3000mm	
近端视场	1500 × 1200mm @ 1.5m	
远端视场	3000 × 2400mm @ 3.0m	
分辨率	2048 × 1536	4096 × 3000
像素数	3.0MP	12.0MP
Z 向重复精度 ( $\sigma$ )	0.5mm @ 3m	0.5mm @ 3m
标定精度	1.0mm @ 3m	0.5mm @ 3m
典型采集时间	0.5~0.9s	1.4~1.7s
重量	约 3.7kg	约 3.9kg
基线长度	约 400mm	
外形尺寸	约 459 × 89 × 145mm	
工作温度范围	-10~45°C	
通讯接口	以太网	
工作电压	24V DC	
最大功率	70W	
激光安全等级	Class 2	
安全和电磁兼容	CE/FCC/VCCI	
防护等级	IP65	
散热	被动	

## 1.7 安全须知与法规要求

使用前请仔细阅读，以防发生意外事故。

查看以下内容，了解 **相机安全须知**。

[安全须知](#)

查看以下内容，了解 **认证信息**。

[法规要求](#)

查看以下内容，了解如何 **维护与保养相机**。

[维护保养](#)

查看以下内容，了解相机 **有害物质声明**。

[有害物质声明](#)

查看以下内容，了解 商标与法律声明。

商标与法律声明

### 1.7.1 安全须知

- 为确保安全使用，在阅读本使用说明书并知悉如何正确使用本产品前，请勿使用。如不按本使用说明书使用和保养，可能导致相机损坏或其他伤害。由于您操作不当导致的人员伤亡或第三方遭受的任何损失，与梅卡曼德无关，梅卡曼德不承担任何责任。
- 遵循本使用说明书中的警告，可有效降低风险，但无法消除所有风险。
- 本使用说明书的内容在编写过程中，每个步骤均经过检查。如您发现任何疑问或错误等，请随时与本公司联系。
- 本产品需由已成年的技术人员安装、连接、使用与维护。请正确运输、储存、安装、连接、使用和维护，以确保产品的安全运行。
- 激光有危害，请了解激光相机使用须知后，再使用本产品。



#### 相机使用环境须知

- 严禁在相机附近放置易燃易爆等危险品；请勿将相机置于明火处或高温下。
- 请勿碰撞、扔掷或摔跌相机。如相机受到强烈碰撞或震动，可能导致损坏或运行故障。禁止对相机进行任何形式的改装，改装、自行维修或拆卸等导致的损坏或损失，本公司概不负责。
- 相机内部请勿混入金属片、粉尘、纸张、木屑等异物，否则可能导致火灾、触电、功能故障等现象。
- 请勿在过高或过低温度环境中使用相机。LSR 系列与 DEEP 相机工作温度范围：-10~45°C；DLP 相机工作温度范围：0~45°C。
- 请室内使用相机。
- 需在海拔 4000 米以下的环境中使用相机。
- 相机需安装在通风且开阔的地方。



#### 相机检查

- 每次使用前，请您仔细检查相机，确保相机处于可正常工作的状态，且无损坏、进水、散发异味、冒烟或螺钉损坏、脱落等现象。如有上述现象，请立即切断电源，停止使用。
- 高温会导致电源线老化，请定期检查电源线，确保电源线正常，无老化现象。如电源线老化，请联系梅卡曼德更换电源线。

 **适配器使用须知**


- 请不要在插口、适配器 / 导轨电源或电源插座潮湿的情况下使用。
- 切勿将适配器 / 导轨电源或电源线投入火中或加热。
- 供电请使用不小于 90W 的 24V 适配器/导轨电源。
- 请使用正确的电源电压，否则可能导致火灾或触电等故障。电源线和适配器/导轨电源需要可靠接地。推荐使用梅卡曼德提供的适配器/导轨电源。如需更换，请使用满足相应安全标准要求的适配器/导轨电源，或使用获得 CCC 认证的适配器/导轨电源。
- 插座接地良好，适配器/导轨电源请勿安放在难以断电的地方。
- 导轨电源需要与配电柜一起使用。

 **激光相机使用须知**

- 请勿直视激光束及其反射激光束；请勿使用光学仪器直视激光束，否则可能对眼睛造成伤害。激光不得对着人发射。
- 激光束不可与眼睛处于同一水平线，必须高于或低于眼睛的水平线。
- 请勿在激光束的路径上放置反光物体，并应充分考虑激光光路。激光被镜面反射/漫反射，可能导致暴露在反射光的危险中，请用遮挡物遮挡反光物体。
- 请不要在激光束的路径上放置金属物体。

 **废弃须知**

- 废弃本相机时请遵守当地法规，共同保护自然环境。请勿随意丢弃废旧相机，不当处置可能污染环境。

注解:  警告：表示如不遵守，可能会导致人员伤亡。

## 1.7.2 法规要求


### 警示标识

#### LED 光警示标识

V3

型号	等级	警告标识	预防措施
Nano	Risk Group 2		不要凝视光束。
Pro S Enhanced			
Pro M Enhanced			
Log S			
Log M			
Deep			

#### V4

型号	等级	警告标识	预防措施
PRO M	Risk Group 2		不要凝视光束。
PRO S			
UHP-140			
NANO			

#### 激光产品安全

依据 GB 7247.1 标准施行激光等级分级。

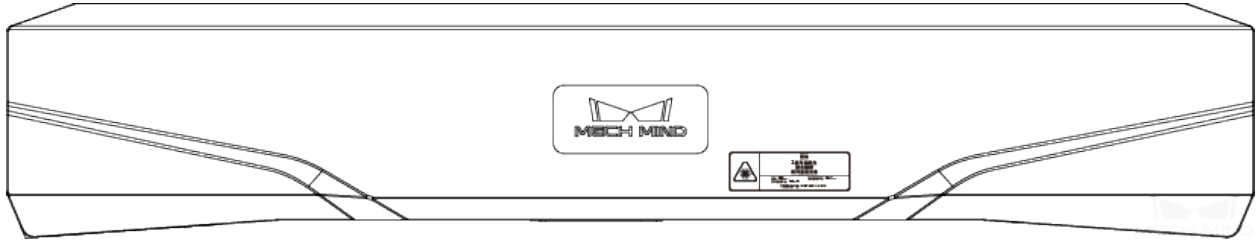
型号	波长	GB 7247.1	
		最大输出功率	激光等级
LSR L	638nm	2.46mW	Class 2
DEEP	638nm	2.46mW	Class 2

激光警示标签：

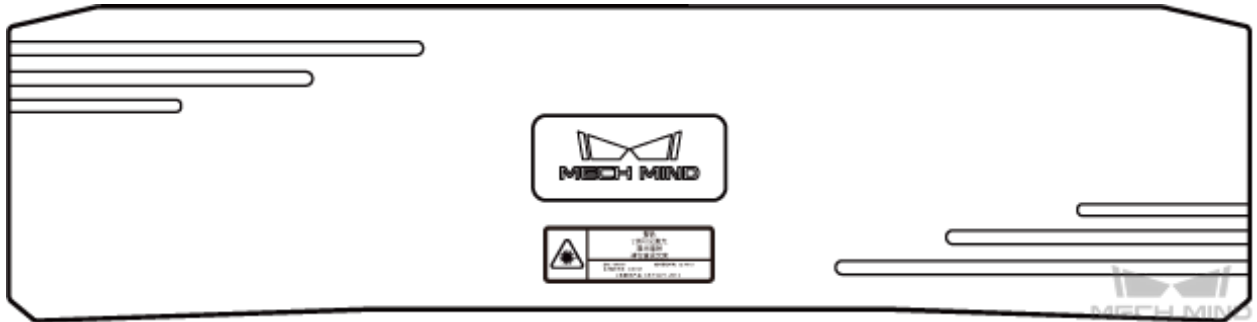


标签粘贴位置:

LSR L



DEEP



## 认证

Mech-Eye 工业级 3D 相机符合以下标准和测试规范。请注意，认证状态可能会有更新。如果您需要了解更多信息，请咨询当地销售代表。

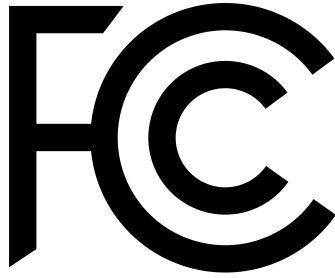
## CE



符合欧盟电磁兼容标准要求。相关标准:

- 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



- 符合美国 ANSI C63.4 和 47 CFR PART 15B 电磁兼容标准要求。
- 符合加拿大 ICES-003 电磁兼容标准要求。

## VCCI



符合日本 VCCI-CISPR 32: 2016 标准要求。

本设备是已取得符合性认证的工业设备，如在家用环境中使用，可能会引起电波干扰。

### 1.7.3 维护保养

#### 清洁

清洁相机机身时，请先轻轻吹去灰尘和浮屑，再用一块干净软布轻轻擦拭。若要去除镜头上污渍，可使用滴有镜头清洁剂（玻璃清洁剂）的干净柔软无绒布小心擦拭，以免造成相机镜头划伤。



#### 警告

- 请勿使用酒精、汽油、煤油或其它有腐蚀性、挥发性的溶剂清洗相机。这些物质可能会损坏相机的外观和内部结构。
- 请勿使用压力水枪或水管喷淋冲刷。相机进水可能导致功能损坏、火灾甚至爆炸的风险。由于进水而导致的损害和损失不在保修范围内。

## 存储

本产品防护等级 IP65，可避免粉尘进入相机内部，影响相机功能。如果将其浸入水中或置于高湿度的环境中，可能会发生故障。内部装置生锈将导致无法挽回的损坏。不使用时，请将相机存于室内阴凉干燥、通风良好的地方。请不要长时间放在户外，避免雨雪等恶劣环境下进水而导致的损坏。



### 警告

- 存放前请断开与电源适配器的连接，以免火灾发生。
- 请勿将镜头正对太阳；请勿长时间将镜头对准太阳或其他强光源。强光可能会损坏影像传感器，导致照片出现白色模糊现象。



注解：警告：表示如不遵守，可能会导致人员伤亡。

## 1.7.4 有害物质声明

部件名称	有害物质					
	铅 (Pb)	汞 (Hg)	镉 (Cd)	六价铬 (Cr(VI))	多溴联苯 (PBB)	多溴二苯醚 (PBDE)
相机主体	×	○	○	○	○	○
电源	×	○	○	○	○	○
线缆	×	○	○	○	○	○
配件	×	○	○	○	○	○

本表格依据SJ/T 11364 的规定编制。


○：表示该有害物质在该部件所有均质材料中的含量均在GB/T 26572 规定的限量要求以下。

×：表示该有害物质至少在该部件的某一均质材料中的含量超出GB/T 26572 规定的限量要求。



## 1.7.5 商标与法律声明

## 商标声明

Mech-Mind、等 Mech-Mind 系列商标、标识等是梅卡曼德（北京）机器人科技有限公司或具有关联关系主体之注册商标或商标，受法律保护，侵权必究。

未经梅卡曼德（北京）机器人科技有限公司事先书面许可，任何单位及个人不得以任何方式或理由对上述商标的全部或任何部分以使用、复制、修改、传播、抄录等任何方式侵权，亦不得与其它产品捆绑使用销售。

凡侵犯本公司商标权的，本公司必依法追究其法律责任。

梅卡曼德公司对于本使用说明书拥有一切权利。根据著作权相关法律规定，未经梅卡曼德公司授权，任何个人或组织不得对本使用说明书的部分内容或全部内容进行复制、修改或发行。对于购买并使用本相机的用户，您可以下载并打印、或复制相应的使用说明书作为个人或组织内部使用，未经梅卡曼德公司的授权，禁止将本使用说明书的内容用于任何其他用途。未经梅卡曼德公司书面同意，任何单位或个人不得转载本使用说明书中的部分或全部内容。



---

## Mech-Eye SDK 2.1 更新说明

---

本文介绍 Mech-Eye SDK 2.1 版本的新增功能、功能优化以及问题修复。

### 2.1 Mech-Eye SDK 2.1.0 更新说明

本文介绍 Mech-Eye SDK 2.1.0 版本的新增功能、功能优化以及问题修复。本次版本重大变更如下：

- 提升点云质量：点云后处理参数全面优化，在保留点云中物体边缘信息的同时，可有效去除离群点与噪点，提升点云质量。
- 丰富接口功能：
  - GenlCam 接口与 Mech-Eye API 中新增手眼标定参数，并提供了相应例程，可直接进行手眼标定。
  - GenlCam 接口新增获取 multi-part payload type 相关参数并提供相应例程，轻松获取纹理点云与仅含 Z 值的深度图。
  - 新增固件升级工具，无需 Mech-Eye Viewer 即可升级相机固件。
- 提升采集速度：优化内存占用，且新增分辨率选择功能以提升相机采集速度，更好地适应对节拍要求比较高的场景。

## 2.1.1 新增功能

### GenICam

- 新增手眼标定参数及例程
 

GenICam 接口新增手眼标定所需的参数，并提供了相应 *HALCON* 例程。通过该例程可进行手眼标定，获取外参，并切换相机输出点云的坐标系。
- 新增 multi-part payload type 相关参数及例程
  - 新增 GenICam 参数，支持在 **AreaScan3D** 模式下获取 multi-part payload type 数据。已支持 Range 与 Intensity 两个通道。
  - 新增应用以上参数的 *HALCON* 例程：
    - \* `obtain_textured_point_cloud` 例程：通过 Range 与 Intensity 两个通道的数据生成纹理点云。
    - \* `obtain_depth_map` 例程：通过 Range 通道获取仅包含物体 Z 值的深度图，加快 *HALCON* 的传输速度，提升效率。
- 新增相机 IP 设置相关参数及例程
 

GenICam 接口新增获取相机当前 IP 设置的参数，并提供了获取及修改相机 IP 设置的 *HALCON* 例程，可直接通过 GenICam 客户端获取和修改 IP 设置。通过该 *HALCON* 例程可读取当前相机的 IP 地址、子网掩码与网关，也可通过该例程修改这些设置。
- 新增相机内参和温度的只读参数
  - 新增内参参数，使用 GenICam 客户端即可读取相机内参。
    - \* `Scan3dFocalLengthX`
    - \* `Scan3dFocalLengthY`
    - \* `Scan3dPrincipalPointU`
    - \* `Scan3dPrincipalPointV`
  - 新增温度参数，使用 GenICam 客户端即可读取相机 CPU 和投影仪的温度。
    - \* `DeviceTemperatureSelector`: 选择需要读取温度的相机组件，包含 **MainBoard**（CPU）和 **Sensor**（投影仪）两个选项。
    - \* `DeviceTemperature`: 该参数显示所选相机组件的温度。

### Mech-Eye API

- 新增手眼标定方法及例程（仅 C++）
 

新增用于手眼标定的 C++ 方法，并提供了相应例程。通过该例程可进行手眼标定，获取外参。
- 新增固件升级工具
 

新增固件升级工具，Ubuntu 系统下也可升级相机固件。
- 新增获取温度的方法
  - C++:

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

- C#:

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

- Python:

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

## Mech-Eye Viewer

- LSR (V4) 与 DEEP (V4) 系列相机显示两种 2D 图

LSR (V4) 与 DEEP (V4) 系列相机连接 Mech-Eye Viewer 后, 可在相机查看器界面选择显示 **2D 图 (纹理)** 或 **2D 图 (深度源)**。其中, 2D 图 (纹理) 用于为点云添加纹理, 2D 图 (深度源) 用于手眼标定、内参检查及设置感兴趣区域。切换 2D 图类型便于调节不同图像的曝光。

同时, 相关的 **2D 参数** 名称更改如下:

原有名称	新名称
彩色相机曝光模式	2D 图 (纹理) 曝光模式
黑白相机曝光模式	2D 图 (深度源) 曝光模式
黑白相机曝光时间	2D 图 (深度源) 曝光时间

- 新增 MTU 功能

配置相机网络服务窗口新增 MTU 设置功能, 该功能在 **管理员** 用户下可见。同时, 将相机固件版本升级至 2.1.0 后, 相机的默认 MTU 值将变为 1500。

---

**注解:** Mech-Eye Viewer 中的 MTU 值与工控机中 **巨型帧** 的值应保持一致。

---

- DEEP (V4) 新增选择深度图分辨率功能

DEEP (V4) 相机的深度图可选择分辨率, 2048 × 1536 或 1024 × 768。使用低分辨率时, 可有效提高相机的采集速度 (约 35%), 更好的应对高节拍场景。

在 **管理员** 用户下, 选择 **工具** ▶ **相机管理器**, 在 **深度图** 下切换分辨率。

- 新增恢复出厂内参功能

**内参工具** 新增 **恢复出厂内参** 功能, 该功能在 **管理员** 用户下可见。

## 2.1.2 功能优化

### 优化点云后处理参数

点云后处理参数全面优化，由两个参数更新为四个参数，点云平滑和去除无用点的效果均得到提升。

- 原 **点云平滑** 更新为 **表面平滑**。优化算法，使用同等强度时，可达到更好的平滑效果。
- 原 **噪点去除** 更新为 **离群点去除** 与 **噪点去除**。将无用点云拆分为离群点与噪点，去除时更有针对性；且优化算法，使用同等强度时，可达到更好的去除效果。
- 新增 **边缘保持**。可根据物体类型选择边缘保持类型，更好地保留真实的物体边缘。

如需达到与原参数类似的后处理效果，请按以下方式调节新参数：

- 如原 **点云平滑** 设为 **Normal**，请将新 **表面平滑** 设为 **Weak**；
- 如原 **噪点去除** 设为 **Normal**，请将新 **离群点去除** 设为 **Weak**。

### 优化相机图像采集速度

将相机的固件版本升级至 2.1.0 后，相机图像采集速度有一定提升。

- 使用 Mech-Eye API C++ 采集深度图时，采集速度最小提升约 2%，采集速度最大提升超过 20%。
- 使用 Mech-Vision 采集深度图时，采集速度最小提升约 5%，采集速度最大提升超过 20%。

---

**注解：** V4 激光相机增加了彩色图去畸变功能，导致彩色图采集速度有所降低。

---

### Mech-Eye Viewer

- 更改深度图保存的默认格式  
保存深度图时，默认格式由 **.tiff** 更新为 **.png**，提升易用性。
- 优化内参工具
  - 优化 **内参工具** 的界面，增加提示信息，提升易用性。
  - 新增高精度模式，提升相机内参精度，适用于对精度要求高的项目。该模式在 **管理员** 用户下可见。
- 优化参数说明区的显示  
点击参数说明区的关闭按钮后，将不再显示参数说明区，避免参数被遮挡，提升易用性。  
如需再次显示参数说明区，请勾选视图菜单中的 **参数说明** 选项。
- 优化视野计算器工具
  - 优化 **视野计算器** 工具的界面，提升易用性。
  - 优化相机型号下拉菜单，便于查找相机。
  - 增加相机型号 **LSR S (V4)**。
- 将 **LSR L (V4)** 的彩色相机降采样功能移动至 **相机管理器**，并更改为彩色图分辨率设置功能。
- **DEEP (V4)** 系列相机的 **相机增益** 默认值由 **0** 更新为 **5**。

- 激光相机的 激光强度值范围由 20~100% 更新为 50~100% 。

## Mech-Eye API

点云后处理参数全面优化，由两个参数更新为四个参数。对应的 Mech-Eye API 方法更新为：

- C++:

- 表面平滑:

```

ErrorStatus_
↪ MechEyeDevice::setCloudSurfaceSmoothingMode (PointCloudProcessingSettings::PointCloudSurfaceSmoothingMode mode)
ErrorStatus_
↪ MechEyeDevice::getCloudSurfaceSmoothingMode (PointCloudProcessingSettings::PointCloudSurfaceSmoothingMode mode)
↪ &
    
```

- 噪点去除:

```

ErrorStatus_
↪ MechEyeDevice::setCloudNoiseRemovalMode (PointCloudProcessingSettings::PointCloudNoiseRemovalMode mode)
ErrorStatus_
↪ MechEyeDevice::getCloudNoiseRemovalMode (PointCloudProcessingSettings::PointCloudNoiseRemovalMode mode)
↪ &
    
```

- 离群点去除:

```

ErrorStatus_
↪ MechEyeDevice::setCloudOutlierRemovalMode (PointCloudProcessingSettings::PointCloudOutlierRemovalMode mode)
ErrorStatus_
↪ MechEyeDevice::getCloudOutlierRemovalMode (PointCloudProcessingSettings::PointCloudOutlierRemovalMode mode)
↪ &
    
```

- 边缘保持:

```

ErrorStatus_
↪ MechEyeDevice::setCloudEdgePreservationMode (PointCloudProcessingSettings::PointCloudEdgePreservationMode mode)
ErrorStatus_
↪ MechEyeDevice::getCloudEdgePreservationMode (PointCloudProcessingSettings::PointCloudEdgePreservationMode mode)
↪ &
    
```

- C#:

- 表面平滑:

```

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

- 噪点去除:

```

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

- 离群点去除:

```

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

- 边缘保持:

```

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

• Python:

- 表面平滑:

```

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

- 噪点去除:

```

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

- 离群点去除:

```

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

- 边缘保持:

```

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

### 2.1.3 问题修复

Mech-Eye SDK 2.1.0 修复了以下问题:

- 激光相机使用 3 次曝光且 激光投影分块数为 3 或 4 时, 相机内存不足导致相机偶发断连。
- Laser L (V3) 与 Laser L Enhanced (V3) 相机设置 激光投影分块数, 曝光时间小于等于 20ms 时深度数据异常。
- 切换参数组时, 偶发 感兴趣区域 (ROI) 设置失效。
- LSR (V4) 与 DEEP (V4) 系列相机的 曝光模式为 自动曝光或 HDR 时, 黑白相机曝光模式的 Flash 不生效。
- 使用路由器连接相机时, 相机 MTU 值过大导致相机有时无法正常工作。
- 相机内存不足导致偶发的丢帧。
- PRO (V4)、LSR (V4) 与 DEEP (V4) 系列相机由于默认开启低功耗模式, 导致偶发的丢帧。

- 多个相机 IP 地址相同时，修改其中一个相机的 IP 地址时，剩余相机的 IP 地址也同时更改。
- 单目相机 2D 图去畸变功能导致采集速度变慢。

---

## Mech-Eye SDK 安装指南

---

本章节介绍如何在 Windows 操作系统中下载、校验软件安装包的完整性、安装、升级、卸载和修复 Mech-Eye SDK（包含 Mech-Eye Viewer 和 Mech-Eye API），以及如何修改安装组件。

### 3.1 下载 Mech-Eye SDK 安装包

请进入 [梅卡曼德在线社区](#) 下载 Mech-Eye SDK 安装包，或联系梅卡曼德售前工程师或销售人员获取。

### 3.2 校验 Mech-Eye SDK 安装包的完整性

由于 Mech-Eye SDK 安装包可能在传输或下载过程中被损坏，因此安装软件前，应先校验软件安装包的完整性。为了方便校验软件安装包的完整性，本公司提供 CRC32 校验码，获取途径为：

- 解压从官网下载的 Mech-Eye SDK 安装包，在 **Mech-Eye SDK 安装指南.pdf** 文件中查看 CRC32 校验码。
- 联系售前或技术支持获取 CRC32 校验码。

要校验软件安装包的完整性，执行如下步骤：

1. 将获取的软件安装包拷贝到工控机或 PC 的指定目录，例如 “D:\”。
2. 右键单击软件安装包，在显示的快捷菜单中，选择 *CRC SHA* ▶ *CRC-32* 计算软件安装包的 CRC32 校验码。
3. 确认计算的 CRC32 校验码与获取的 CRC32 校验码相同。

---

如果您第一次使用 Mech-Eye SDK，请参照安装 *Mech-Eye SDK* 安装软件。

如果您已安装 Mech-Eye SDK：



- Mech-Eye SDK 为 1.6.1 版本时，请参照升级 *Mech-Eye SDK* 将软件升级到最新版本。
- Mech-Eye SDK 为 1.6.1 之前版本时，请参照安装 *Mech-Eye SDK* 安装软件。

---

### 3.3 安装 Mech-Eye SDK

1. 双击 **Mech-Eye SDK Installer x.x.x.exe**，运行 Mech-Eye SDK 安装向导。
2. 在 **欢迎**窗口，浏览软件说明信息，并单击 **下一步**。
3. 在 **许可协议**窗口，仔细阅读许可协议，勾选 **我接受“许可协议”中的全部条款和条件**复选框，然后单击 **下一步**。
4. 在 **选择产品**窗口，选择需要安装的组件，根据需要勾选 **桌面快捷方式**复选框，然后单击 **下一步**。

---

#### 注解：

- 请确认 **添加至环境变量**复选框已勾选。
- 推荐勾选 **Mech-Eye SDK Docs** 选项，可直接从软件中打开用户手册。

- 
5. 在 **设置安装路径**窗口，选择安装路径，然后单击 **下一步**。

---

**注解：** 软件默认安装路径：C:\Mech-Mind\Mech-Eye SDK-x.x.x。

- 
6. 在 **安装前确认**窗口，确认安装路径无误后，单击 **安装**。
  7. 在 **执行安装**窗口，等待软件安装完成。
  8. 软件安装完成后，在 **结束**窗口，单击 **完成**，退出安装向导。

---

**提示：** 安装完成后，请重启工控机或 PC，否则可能导致添加的环境变量不生效。

---

### 3.4 升级 Mech-Eye SDK

---

**注解：** 仅 2.0.0 版本的 Mech-Eye SDK 安装向导支持该功能，且仅支持 1.6.1 版本的 Mech-Eye SDK 升级到 2.0.0 版本。

---

如果您的工控机或 PC 已经安装 Mech-Eye SDK 1.6.1 版本，可使用 Mech-Eye SDK 安装向导升级软件。

1. 双击 **Mech-Eye SDK Installer x.x.x.exe**，运行 Mech-Eye SDK 安装向导。
2. 在 **升级**窗口，单击 **升级**，保留历史版本或 **升级**，删除历史版本。
3. 根据提示安装软件，直至软件完成升级。

## 3.5 卸载 Mech-Eye SDK

卸载 Mech-Eye SDK 有两种方法：Mech-Eye SDK 安装向导卸载或控制面板卸载。

### 3.5.1 使用安装向导卸载

1. 双击 **Mech-Eye SDK Installer x.x.x.exe**，运行 Mech-Eye SDK 安装向导。
2. 在 **维护**窗口，点击 **卸载**。
3. 在 **卸载**窗口，选择 **保留用户配置文件或不保留用户配置文件**。
4. 等待软件完成卸载。

### 3.5.2 使用控制面板卸载

1. 在工控机或 PC 上，打开 **控制面板**。
2. 选择 **程序** ▶ **程序和功能**。
3. 右键单击 Mech-Eye SDK，选择 **卸载**。
4. 等待软件完成卸载。

## 3.6 修复 Mech-Eye SDK

如果 Mech-Eye SDK 出现异常导致无法正常使用，请使用 Mech-Eye SDK 安装向导对其进行修复。

1. 双击 **Mech-Eye SDK Installer x.x.x.exe**，打开 Mech-Eye SDK 安装向导。
2. 在 **维护**窗口，点击 **修复**。
3. 等待软件完成修复。

## 3.7 修改安装的组件

---

**注解：**仅 2.0.0 版本的安装包支持该功能。

---

安装完成后，如果要修改安装的组件，执行如下步骤：

1. 双击 **Mech-Eye SDK Installer x.x.x.exe**，运行 **Mech-Eye SDK 安装向导**。
2. 在 **维护**窗口，点击 **修改**。
3. 在 **选择产品**窗口，选择并勾选需要安装的组件。
4. 根据提示安装软件，直至完成。

## 3.8 软件许可协议

关于 Mech-Eye SDK 的许可协议，请参考 [终端用户许可协议](#)。

## 3.9 安装常见问题

### 3.9.1 安装包未能正常启动

问题现象：

运行安装包后，安装包未能正常启动或启动界面一闪而过。

可能原因：

系统盘的磁盘空间不足。

处理步骤：

确认系统盘可用空间是否大于安装包的大小。

- 如果可用空间小于安装包大小，删除不需要文件为软件安装腾出足够空间，然后尝试重新安装。如果问题仍未解决，联系梅卡曼德技术支持。
- 如果可用空间大于安装包大小，联系梅卡曼德技术支持。

### 3.9.2 安装失败

问题现象：

软件安装过程中，出现“安装失败”错误提示。

可能原因：


- 安装包损坏或文件缺失。
- 当前计算机用户不具备管理员权限。
- 另一个程序正在安装或 Windows 系统正在升级。
- 其他原因。

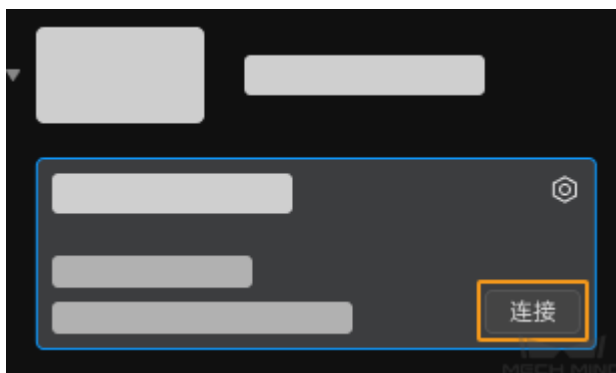
处理步骤：

1. 重新获取安装包并校验软件安装包的完整性，然后重新尝试安装。
  - 如果解决，故障处理结束；
  - 否则，执行步骤 2。
2. 右键单击安装包，选择“以管理员身份运行”安装软件。
  - 如果解决，故障处理结束；
  - 否则，执行步骤 3。
3. 等待其他程序安装完成后或 Windows 系统升级完成后重新尝试安装。
  - 如果解决，故障处理结束；

- 否则，执行步骤 4。
4. 重启工控机或 PC，然后尝试重新安装软件。
    - 如果解决，故障处理结束。
    - 如果未解决，请点击 **结束**窗口的“安装日志”链接收集安装日志，并反馈给技术支持。

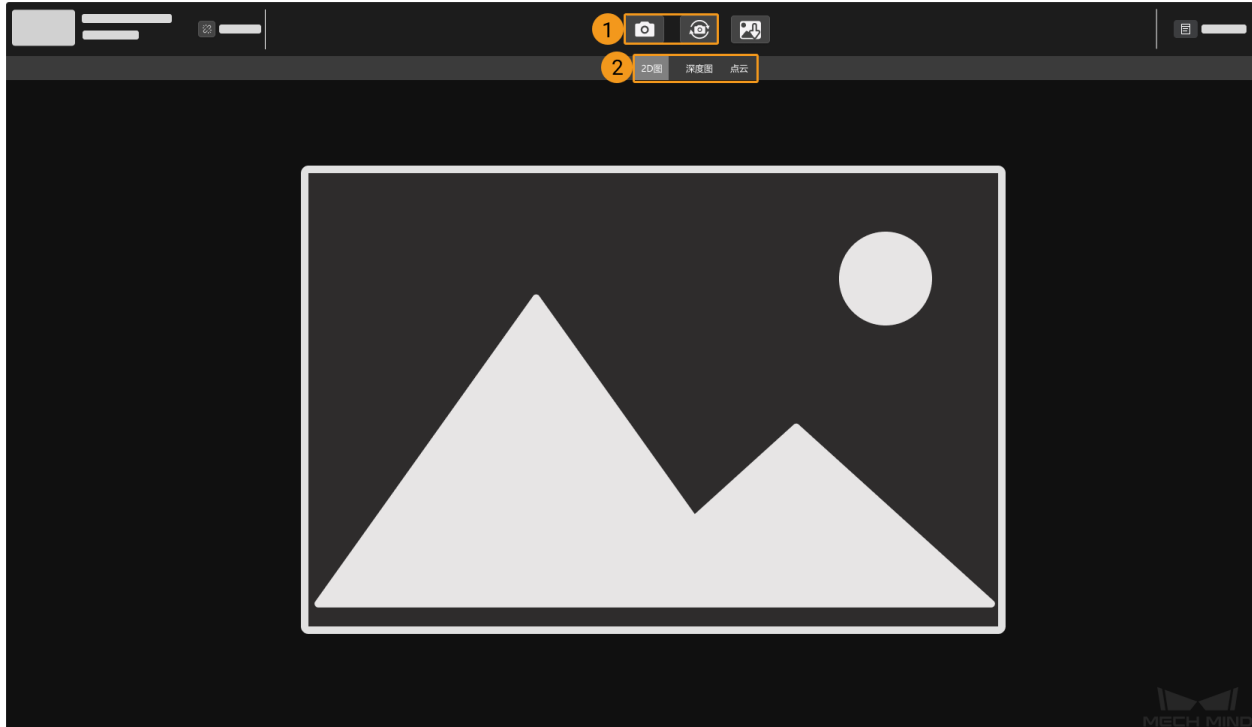
### 4.1 连接相机

打开软件进入相机待连接界面。选中相机，单击  设置相机 IP 地址。然后，单击 连接，完成连接。



## 4.2 获取图像

相机连接成功后进入相机查看器，如下图所示。点击  或  采集图像，可切换 2D 图、深度图或点云。






## 4.3 调整参数

调整参数，改善图像效果。

- **2D 图优化** 调整 **2D** 参数的 **曝光模式** 及相关参数。调整曝光参数时，使用连续采集可看出明显效果。更多 2D 图优化请参见 **2D 图** 参数调节。

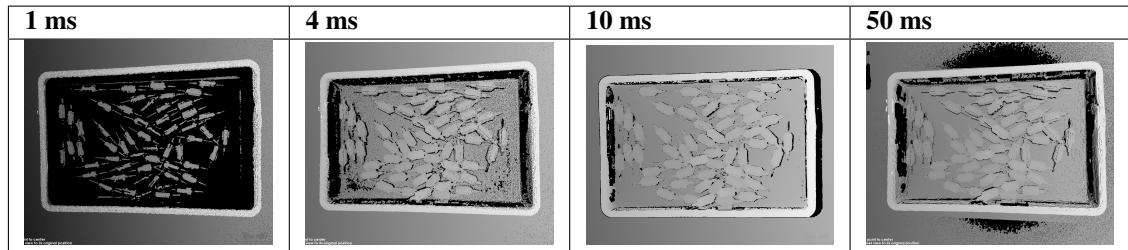
固定曝光模式下不同曝光时间的效果图见下表：

30 ms	120 ms	220 ms
		

提示：曝光时间影响亮度。曝光时间越大，亮度越高；曝光时间越小，亮度越低。


- 深度图及点云优化 调整 3D 参数的 曝光次数及 曝光时间，使用曝光助手 可得到最佳曝光参数组合。更多深度图优化请参见深度图与点云常用参数调节。

3D 参数调整效果见下表：



- 点云效果优化 调整 点云后处理的 点云平滑、噪点去除及 条纹对比度阈值。具体优化请参见点云后处理。

## 4.4 保存数据

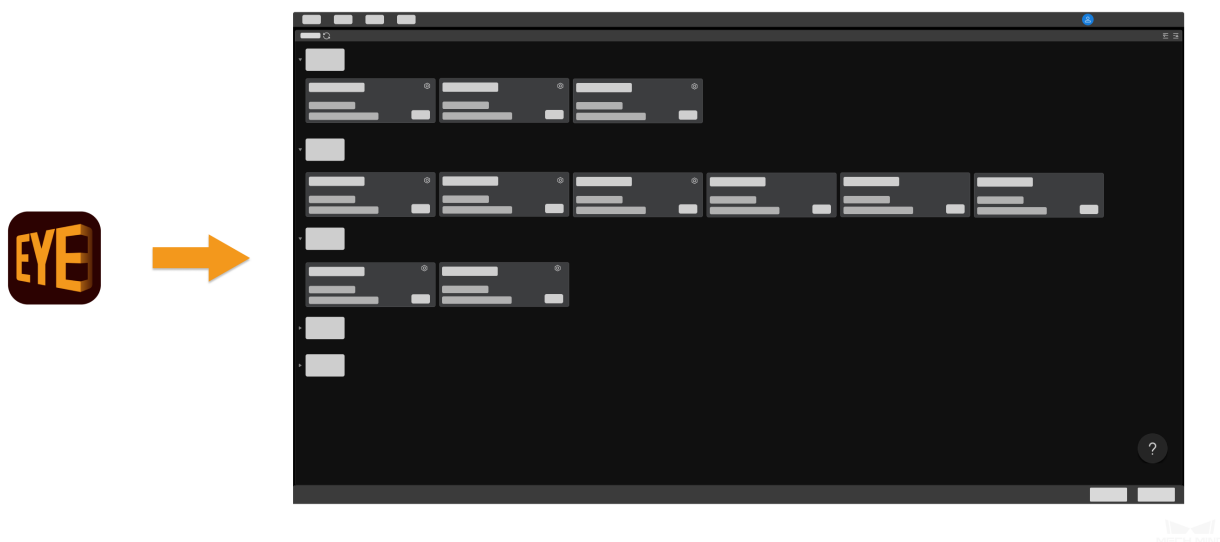
单击 文件 ▶ 保存相机原始数据保存相机原始数据。或点击相机工具栏中的 ，自行选择图像存储类型。

---

## 快速了解 Mech-Eye Viewer

---

Mech-Eye Viewer 是梅卡曼德机器人自主研发的相机配置及可视化软件，可根据目标物体特性，调节 Mech-Eye 工业级 3D 相机的各种参数，快速获得高质量的 2D 图、深度图和点云。



---

查看以下内容，了解如何 **安装 Mech-Eye Viewer**。

*Mech-Eye SDK 安装指南*

---

查看以下内容，了解 **软件界面介绍**。

---



界面介绍

---

查看以下内容，了解如何 采集、调节并保存数据。

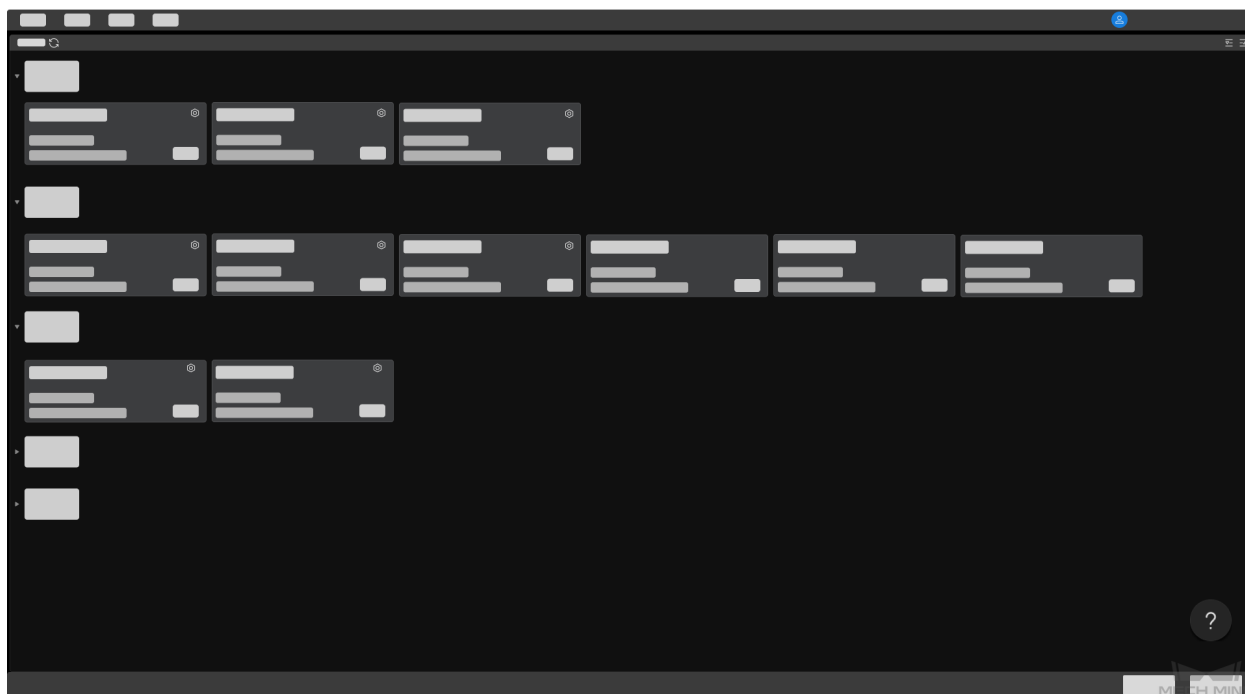
操作指南

---


查看以下内容，了解如何 使用内置工具。

内置工具

启动 Mech-Eye Viewer，进入如下界面。

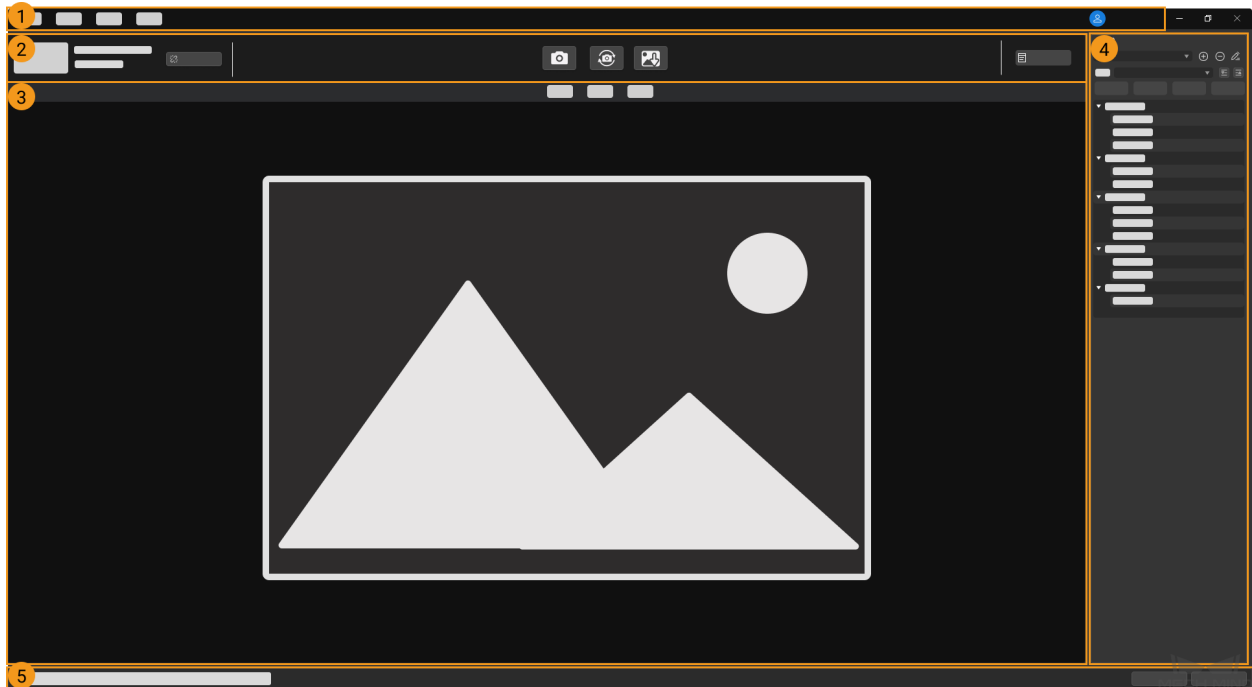


该界面用于查看当前可用相机及其信息，[连接相机并设置相机 IP 地址](#)。

单击  可查看如何解决[相机连接异常](#)的问题。

## 6.1 相机查看器界面

相机连接成功后进入相机查看器界面，如下图所示。




相机查看器界面主要包含菜单栏、相机工具栏、数据查看区、参数区和信息栏。

1. **菜单栏**：用于保存原始数据、启动虚拟相机、打开相机内置工具，并查看软件相关信息。
2. **相机工具栏**：用于断开相机连接、采集并保存数据、查看日志信息等。
3. **数据查看区**：显示从相机获取的 2D 图、深度图和点云。
4. **参数区**：查看并调整参数，并保存不同类型的参数组。
5. **信息栏**：单击 上一条 或 下一条 切换显示信息。

### 6.1.1 菜单栏

菜单栏包含 文件、工具、视图、帮助 以及用户切换菜单。

菜单	选项	描述
文件	保存相机原始数据	保存 MRAW 格式的原始数据，可用于后期查看、调试和分析。
	启动虚拟相机	加载已保存的原始数据，查看拍摄时设置的参数等。
工具	包含一系列辅助工具，更多信息请参阅 <a href="#">内置工具</a> 。	
视图	工具栏	勾选以显示工具栏。默认不勾选。
	图像信息框	显示图像信息框，用于查看图像的位置与颜色信息。默认勾选。
	点云展示按钮	勾选后点云查看界面显示播放按钮。默认不勾选。
	参数说明	查看参数说明。默认勾选。
帮助	关于	软件版本信息。
	更新说明	此版软件相对于上一版的更新说明。
	使用手册	在浏览器中打开软件使用手册。
	设置	切换软件语言，重启软件后生效。
	单击切换用户类型。默认为 <b>标准</b> ，其他类型需要密码。如需使用，请联系梅卡曼德技术支持。	

## 6.1.2 相机工具栏

在相机工具栏中，您可进行以下操作：

- 设置相机备注
- 采集图像
- 保存数据
- 查看日志
- 断开与当前相机的连接

## 6.1.3 数据查看区

显示相机输出的数据，可在2D图、深度图、点云之间切换。

各数据显示界面的更多详细功能，请参阅查看数据时的操作。

## 6.1.4 参数区

包含所有可调节的相机参数。您也可更改参数的可见级别、保存参数组、查看参数说明等。更多信息请参阅参数调节。

本章介绍了 Mech-Eye Viewer 的界面，下一章将介绍如何使用 Mech-Eye Viewer。

本章主要介绍如何使用 Mech-Eye Viewer 获取图像并调节参数。

正式调节参数前，请检查相机内参。只有相机内参准确，才能保证计算得出的深度图各点坐标准确。

---

查看以下内容，了解如何 **连接相机并设置 IP**。

[设置相机 IP 地址并连接相机](#)

---

查看以下内容，了解如何 **获取数据**，并了解 **数据类型**。

[采集与查看数据](#)

---

查看以下内容，了解如何 **调节数据**。

[参数调节](#)

---

查看以下内容，了解如何 **保存数据**。

[数据保存](#)

---

当软件出现问题时，查看以下内容，了解 **日志信息**，并根据日志信息解决问题。

[日志管理](#)

---

## 7.1 设置相机 IP 地址并连接相机

为使 Mech-Eye Viewer 成功连接相机，建议连接前先设置 IP 地址。请确保以下两个 IP 地址在同一网段，且 IP 地址唯一。

- 相机 IP 地址
- 与相机相连的工控机网口的 IP 地址

IP 地址设置完成后，即可连接相机；相机连接后，您可设置相机备注，以便区分相机。

---

### 提示:

- 如 Mech-Eye Viewer 无法搜索到相机，请参考 *Mech-Eye SDK* 搜索不到相机。
  - 如 Mech-Eye Viewer 无法连接相机，请参考 *Mech-Eye SDK* 无法连接相机 (*Unreachable*)。
- 

### 7.1.1 设置相机 IP 地址

您可将相机设置为静态 IP 地址，或者设置为动态分配。


#### 设置静态 IP 地址

---

**提示:** 请保存好相机 IP 地址，以备后用。

---

使用 Mech-Eye Viewer 设置静态 IP 地址的步骤如下:

1. 打开 Mech-Eye Viewer，将鼠标移动至所选相机的信息栏。单击 ，打开 IP 地址设置窗口。
2. 选择 **设置为静态 IP**。
3. 根据 **电脑 IP 配置**中的 IP 地址及子网掩码，选择 **IP 地址类型**，并输入相机 IP 地址和子网掩码。完成后，单击 **应用**。


---

### 提示:

- IP 地址必须唯一。
  - **电脑 IP 配置**中显示的是与相机连接的工控机网口的 IP 地址及子网掩码。
-

## 设置动态 IP 地址

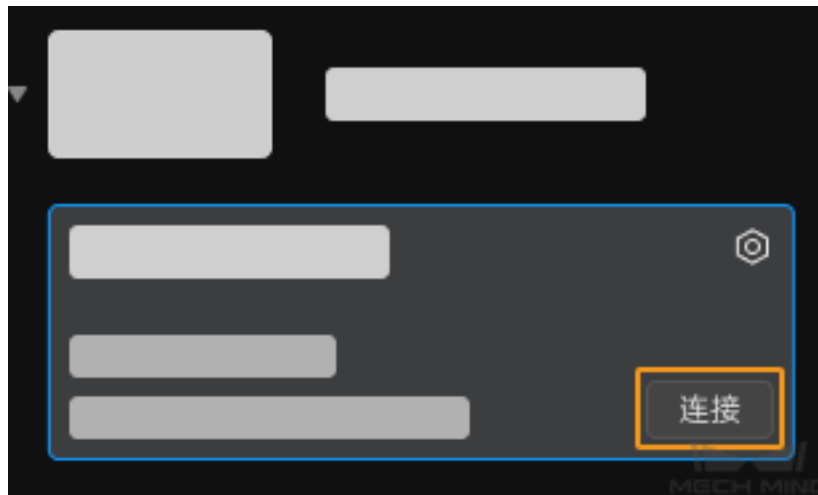
使用 Mech-Eye Viewer 设置动态 IP 地址的步骤如下：

1. 打开 Mech-Eye Viewer，将鼠标移动至所选相机的信息栏。单击 ，打开 IP 地址设置窗口。
2. 选择 设置为动态分配，并单击 应用。

**提示：**如需设置与相机相连的工控机网口的 IP 地址，请参考[设置工控机 IP 地址](#)。

### 7.1.2 连接相机

打开 Mech-Eye Viewer 后，**相机列表**中显示所有可连接的相机。选中相机，并单击 连接。



#### 注意：

- 相机固件版本需与软件版本一致，才可连接。如版本不一致，请升级相机固件。
- 首次连接相机时，需等待几分钟，相机列表中才出现该相机。
- 如相机显示 已连接，则该相机已被 GenICam 客户端连接。如需在 Mech-Eye Viewer 中连接该相机，需先在 GenICam 客户端中断开连接。

### 7.1.3 设置相机备注

相机备注用于区分相机，请根据需要自行设置。设置完成后，在相机列表界面中可见。

连接相机后，单击工具栏中的  或者 请设置相机备注，可设置相机备注。

注解:

- 不可输入的特殊字符包含：\、/、:、\*、?、"、<、>、|。
- 虚拟相机不可设置相机备注。

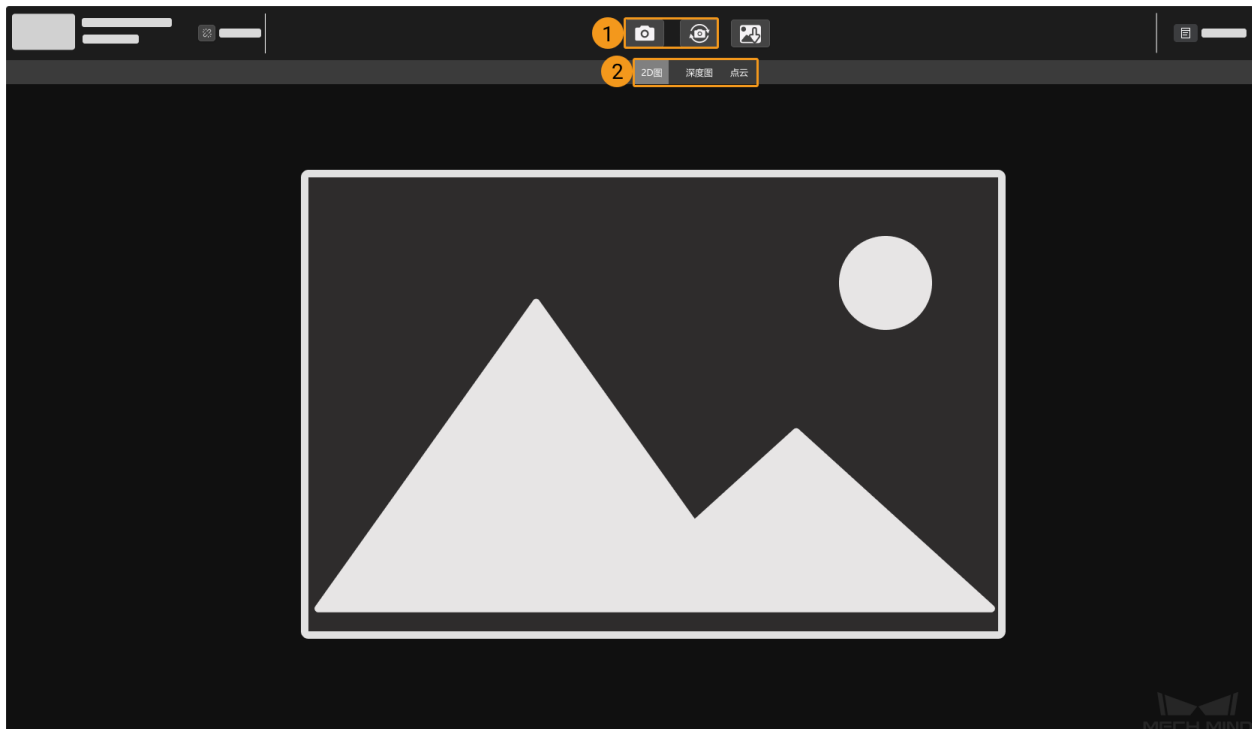
本章介绍了如何设置相机 IP 地址及连接相机，连接相机之后即可开始采集图像。下一章将介绍如何进行图像采集，并切换不同数据类型。

## 7.2 采集与查看数据

本章主要介绍如何采集图像，切换数据类型，以及查看数据时的常用操作。

### 7.2.1 采集图像与切换数据类型

相机连接成功后进入相机查看器界面，在此界面可采集并查看数据。







- 1: 采集图像
- 2: 切换数据类型

### 采集图像

采集图像有两种方式：

-  单次采集：单击可获取一次图像采集输出的数据。
-  连续采集：以固定时间间隔连续采集图像。再次点击停止采集。

---

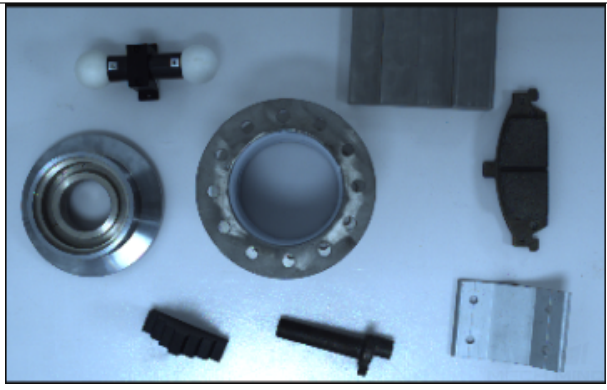
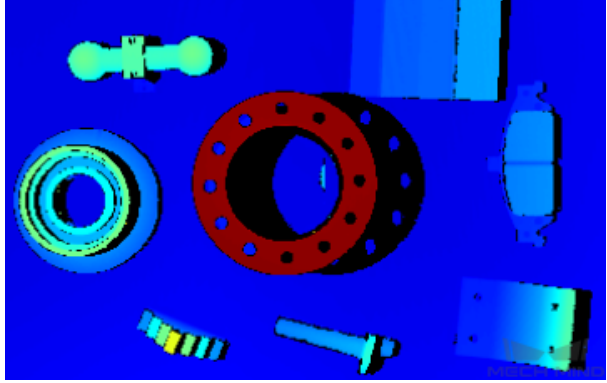
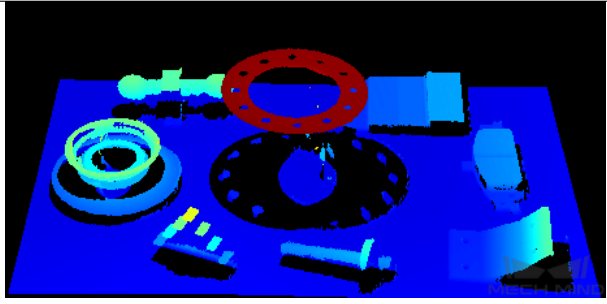
**提示：** 双目相机采集图像时，采集一次有左图和右图两张 2D 图，其中左图为主视图。双目相机具体型号可在[相机选型](#) 中查看。

---


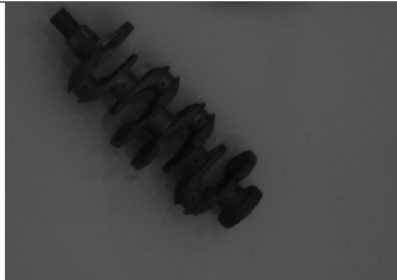

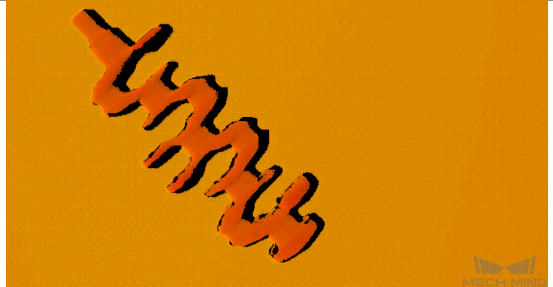
### 数据类型

相机输出至 Mech-Eye Viewer 的数据，单击对应的按钮即可切换所查看的数据类型。

非 DEEP (V4) 与 LSR (V4) 系列数据类型为：

类型	说明	示例
2D 图	黑白或彩色 2D 图（取决于相机型号）	
深度图	包含深度数据的 2D 图像	
点云	3D 空间中反映物体表面特征的点的集合	

DEEP 与 LSR 系列数据类型为：

类型	说明	示例
2D 图（纹理）	用于为点云添加纹理的 2D 图像	
2D 图（深度源）	由采集深度数据的 2D 相机拍摄的 2D 图像	
深度图	包含深度数据的 2D 图像	
点云	3D 空间中反映物体表面特征的点的集合	

提示：2D 图（深度源）用于手眼标定、内参检查及设置感兴趣区域。

Mech-Eye Viewer 提供一系列的操作辅助查看这些数据。具体的操作说明请参阅下一章内容。

## 7.2.2 查看数据时的操作

在数据查看区，您可通过鼠标、键盘、界面中的按钮及菜单，调整数据的显示方式。  
请根据数据类型查看相应的操作说明：

- 2D 图/彩色图/黑白图
- 深度图
- 点云

### 2D 图/彩色图/黑白图

#### 缩放

在 2D 图显示界面中，滚动鼠标滚轮可对图像进行缩放。  
单击鼠标右键，选择 重置视图，可将图像恢复为未缩放时的尺寸。

#### 图像信息

光标移动至 2D 图，右下角显示光标所在像素的像素位置及 RGB/灰度。  
该功能可通过 视图菜单关闭。

### 深度图

#### 缩放

在深度图显示界面中，滚动鼠标滚轮可对图像进行缩放。  
单击鼠标右键，选择 重置视图，可将图像恢复为未缩放时的尺寸。

#### 深度显示调节

在左边栏中，可对深度值的坐标系、颜色和范围等进行调整。

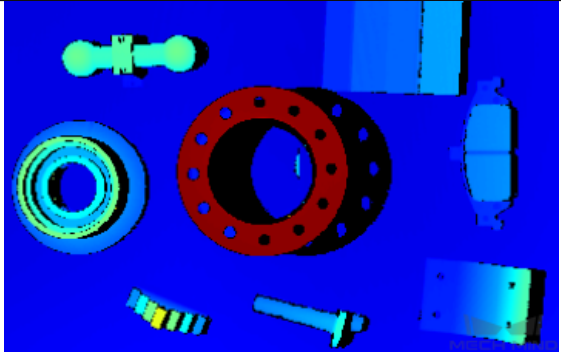
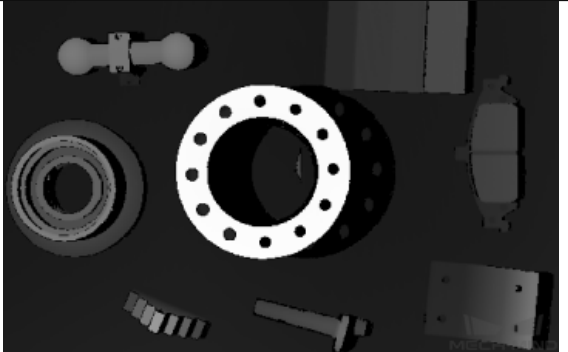
## 坐标系

用于显示深度值的坐标系，分为 相机与 自定义两种。

- 相机：默认选项，深度值以相机坐标系显示。
- 自定义：深度值以用户设置的自定义坐标系显示。依次单击 工具 ▶ 自定义坐标系即可进行设置。

## 颜色样式

深度值的显示颜色。

选项	Jet	灰度
说明	以 Jet 配色方案显示深度	以灰度显示深度
示例		

## 颜色条

颜色条上下两端的 最小值 和 最大值 对应参数中的 深度范围。

- 两个滑块分别对应当前深度图中的最大和最小深度值，右侧数值为滑块对应深度值，所选颜色样式均匀分布在两滑块之间。
- 深度图颜色显示有两种方案：
  - 滑块位置自动调整：默认方式。每次采集图像后，根据计算得到的实际深度范围，自动调整滑块位置。
  - 滑块位置不变：勾选 锁定，滑块将不可移动。重新采集图像后，深度图的颜色按照被锁定的深度范围显示。

---

**提示：** 深度图颜色锁定后，如需恢复自动计算深度范围，先取消勾选 锁定，再单击 复位即可。

---

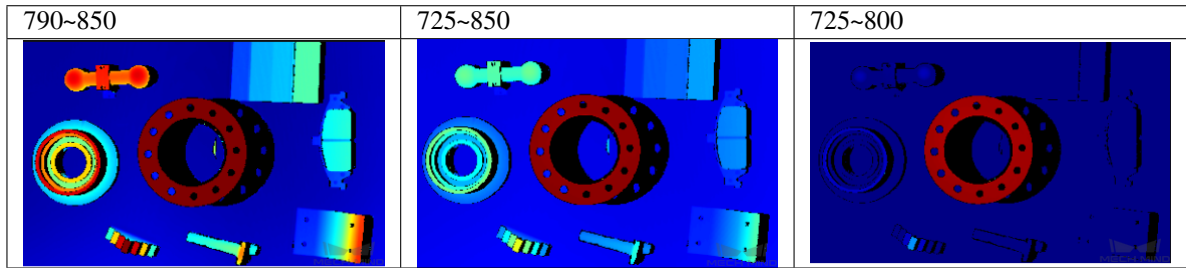
- 单位：mm（默认值）或  $\mu\text{m}$ 。

---

**提示：** 使用 UHP 系列相机时，推荐使用  $\mu\text{m}$ ；其他系列使用 mm。

---

- 查看选中范围内深度值的变化：可通过移动滑块或输入数值来调整显示范围。调整后，所选颜色样式将被重新分布至设定的深度范围内，范围外的深度值均以极端值颜色显示。



提示：查看深度值变化时，可先移动滑块确认大致范围，再输入数值进行微调。

## 图像信息

光标移动至深度图，显示光标所在像素的像素位置及坐标。

该功能可通过 **视图** 菜单关闭。

## 点云

### 缩放和移动

在点云显示界面中可进行以下操作：

- 滚动鼠标滚轮：缩放
- 鼠标左键拖拽：旋转点云
- 鼠标滚轮拖拽：平移点云
- 单击键盘 Backspace 键：将点云复位至调整前的状态
- 单击键盘 R 键：将视角复位至中心

### 深度显示调节

在左边栏中，可对深度值的坐标系、颜色和范围等进行调整。

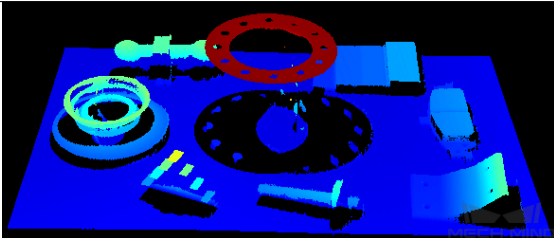
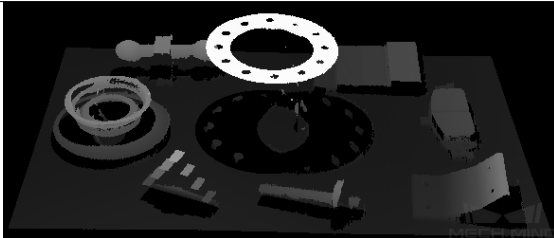
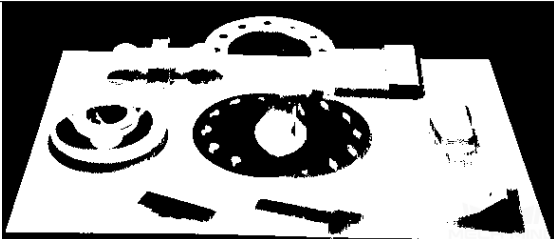
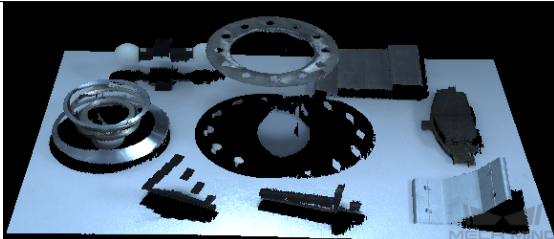
## 坐标系

用于显示深度值的坐标系，分为 相机与 自定义两种。

- 相机：默认选项，深度值以相机坐标系显示。
- 自定义：深度值以用户设置的自定义坐标系显示。依次单击 工具 ▶ 自定义坐标系即可进行设置。

## 颜色样式

深度值的显示颜色。

选项	Jet	灰度
说明	以 Jet 配色方案显示深度	以灰度显示深度
示例		
选项	白色	纹理
说明	显示纯白无纹理的点云	使用 2D 图为点云上色
示例		

**注解：**选择 白色或 纹理时，调整颜色条无效。切换至深度图显示界面，再切换回点云界面时，颜色样式的设置将被自动更改为与深度图显示界面一致的选项。

## 颜色条

点云颜色条与深度图颜色条相同，请查阅上文 获取详细信息。

## 点云展示

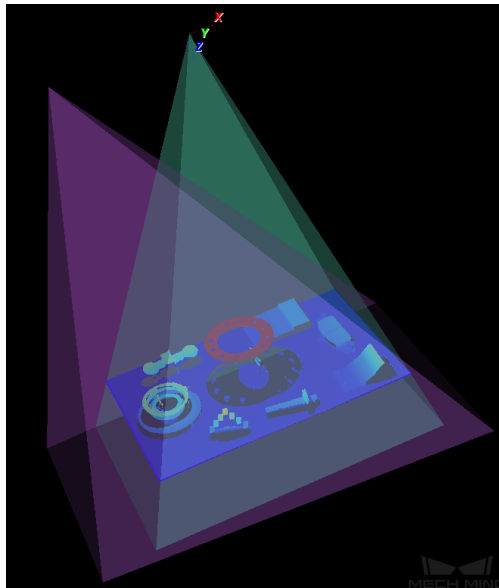
依次单击 视图 ▶ 点云展示，点云视图右上角中出现 播放按钮。

单击 播放，点云开始按照固定轨迹移动；单击 停止，点云停止移动，并恢复到初始状态。

## 其他操作

在点云显示界面中，右键菜单包含以下选项：

- **保存点云：** 将当前点云保存到本地（PLY 格式，单位为米）
- **显示视锥：** 以视锥形式显示 2D 相机和投影仪的视野



- **显示坐标轴：** 显示所选坐标系的原点及坐标轴

---

数据类型的介绍至此全部完成，下一章将介绍如何调节参数获得理想数据。



## 7.3 参数调节

本章主要了解参数区的构成，如何通过参数调节图像与点云，并了解典型场景下的点云滤波推荐值。

查看以下内容，了解 [参数区](#) 的相关内容。

[参数区](#)

查看以下内容，了解 [如何通过 2D 参数调节 2D 图](#)。

[2D 图参数调节](#)

初级下的参数为常用参数。查看以下内容，了解 [深度图与点云的常用参数调节](#)。

[深度图与点云常用参数调节](#)

专家与 大师级别下的参数为高级参数。查看以下内容，了解 [深度图与点云的高级参数调节](#)。

[深度图与点云高级参数调节](#)

**提示：** 优先调节常用参数。如效果不理想，再调节高级参数。

### 7.3.1 参数区

本章介绍参数区的界面组成和基本操作，主要包含以下几个区域：

- [参数组区](#)
- [可见级别区](#)
- [参数调节区](#)
- [参数说明区](#)

1

2

3

4

参数组

default ▼ ⊕ ⊖ ✎

保存 重置 导入 导出

可见级别 初级 ▼ ☰ ☱

参数名称	值
3D参数	自动设置
曝光次数	1
曝光时间	8.0 ms
2D参数	
2D图 (纹理) 曝光模式	固定曝光 ▼
曝光时间	200.0 ms
2D图 (深度源) 曝光模式	固定曝光 ▼
2D图 (深度源) 曝光时间	300.0 ms
点云后处理	
表面平滑	Normal ▼
离群点去除	Normal ▼
条纹对比度阈值	3
深度范围	编辑
下限	200 mm
上限	4000 mm
感兴趣区域	编辑
设置值	[(0, 0), 0 x 0]

**参数说明** ✕

**离群点去除**


可去除点云中的离群点。离群点为游离于物体点云之外的成团的点。

说明：如物体点云包含成团且与其他部分有深度差的点，高强度的离群点去除可能会去除该部分物体点云。

API参数名	PointCloudOutlierRemoval
值范围	-
默认值	1
可见级别	初级
数据类型	Enum

## 参数组区


通常不同场景需要使用不同的参数值。建议为需要的场景建立不同的参数组，并根据不同的场景切换对应的参数组。

单击参数组右侧的 ，显示所有参数组，请根据需要切换参数组。

软件内置两个参数组：**default**（默认）和 **calib**（标定）。

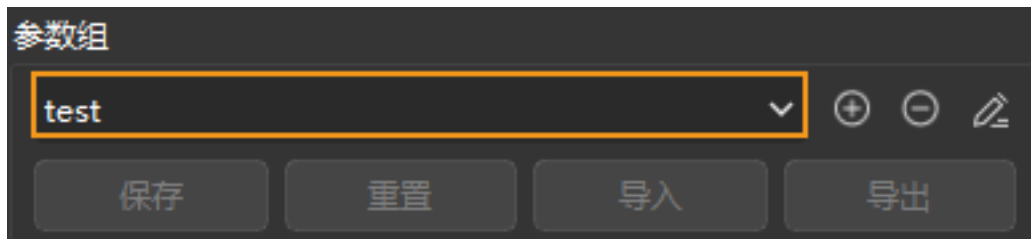
## 如何创建参数组


不同工程、不同场景需要的参数值不同。针对不同的场景设定对应的参数组，可快速使用对应的参数组。

1. 单击  新增参数组，在提示页面中输入参数组名称，并单击 确定。



2. 新增完成后，参数组显示新增参数组名称。



3. 调节参数，参数名称后显示星号 (\*)，表示参数未保存。参数调节完成后，单击 ，保存参数。参数组创建完成。

## 按钮说明

按钮	功能
	新增参数组。
	删除当前参数组。
	重命名当前使用参数组。
	保存参数组。快捷键为 Ctrl S。
	重置参数组。当前参数组恢复默认值。
	导入参数组。
	导出参数组。

提示：连接虚拟相机时，以上按钮均不可用。



## 可见级别区

可见级别分为 初级、专家与 大师，不同 可见级别下开放的参数不同。

- 初级的参数为常用参数
- 专家与大师的参数为高级参数

注解：大师级别下的参数仅 管理员模式可见，如需使用，请咨询技术支持。

按钮说明：

按钮	功能
	展开当前 可见级别下的所有参数
	折叠当前 可见级别下的所有参数

## 参数调节区

调节参数的 **值**，可改善 2D 图、深度图与点云的质量。调节参数时，可参考参数描述。

- 2D 图调节请查看 [2D 图参数调节](#)。
- 深度图和点云调节请查看 [深度图与点云常用参数调节](#) 与 [深度图与点云高级参数调节](#)。

## 参数说明区

选中 **参数名称**或其对应的 **值**，可在此区域查看该参数对应的参数说明。

首次安装 Mech-Eye Viewer 时，参数说明默认开启。参数说明包含参数解释、**API 参数名**、**值范围**、**默认值**、**可见级别**与 **数据类型**。调节参数时，可参考参数说明与取值范围。

关闭 / 开启参数说明：

- 关闭：单击参数说明窗口右上角的 ×。关闭参数说明区后，再选中参数或参数值时，将不再显示。
- 开启：依次单击 视图 ▶ 参数说明。

本章介绍了参数区的基本内容，接下来将介绍如何通过调节参数获得需要的 2D 图、深度图或点云。

下一页将介绍如何使用 2D 参数调节 2D 图。

### 7.3.2 2D 图参数调节

**2D 参数调节 2D 图**。2D 图调节时，需保证图像亮度合适，且所需细节清晰。

初级级别下的参数为常用参数，优先使用；**专家级别**下的参数为高级参数。参数调节包含：

- 常用参数调节
- 高级参数调节

#### 常用参数调节

**2D 参数曝光模式**参数与相机系列、包含选项的对应关系，如下表所示：

	曝光模式	2D 图（纹理）曝光模式	2D 图（深度源）曝光模式
	适用于非 DEEP 与 LSR 系列相机	适用于 DEEP 与 LSR 系列相机	
固定曝光	✔	✔	✔
自动曝光	✔	✔	
HDR	✔	✔	
Flash	✔		✔

**提示：** Mech-Vision 连接 DEEP (V4) 与 LSR (V4) 系列相机时，Mech-Vision 中的“彩色图”端口获取的是 2D 图（深度源）。如需调节 Mech-Vision 中的“彩色图”，请调节 **2D 图（深度源）曝光模式**。

不同曝光模式的适用场景如下所示：

曝光模式	功能与适用场景
固定曝光	设置固定曝光时间，常用于光照条件稳定的场景。
自动曝光	自动调整曝光时间，常用于光照条件变化的场景。
HDR	设置多个曝光时间并融合所有图像，常用于颜色或物体多样的场景。
Flash	使用投影机补光，常用于较暗的环境。

## 固定曝光

- DLP 相机：

仅需要调节 **曝光时间**。

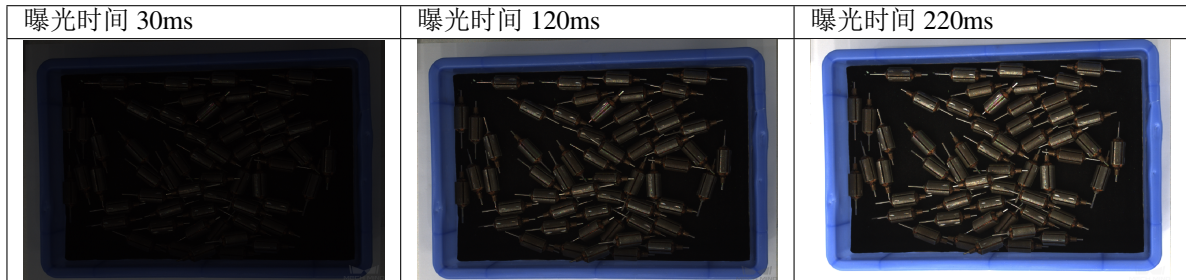
- DEEP 与 LSR 系列相机：

DEEP 与 LSR 系列相机可提供两种 2D 图。请根据使用场景调节 **曝光时间**或 **2D 图（深度源）曝光时间**。

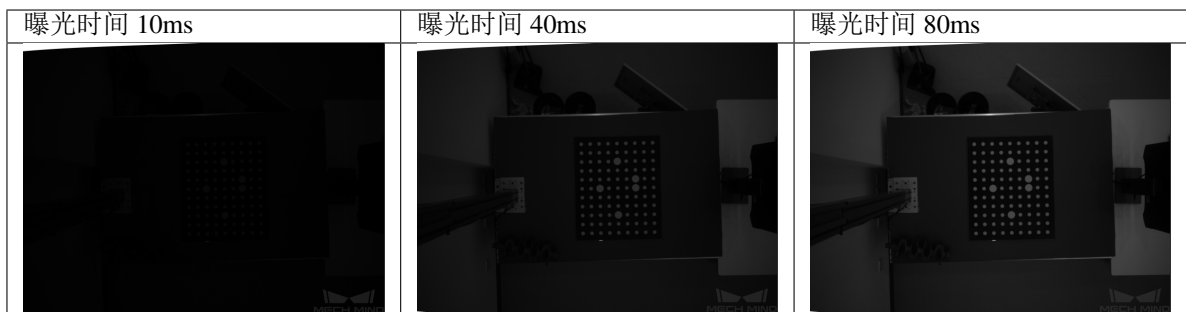
参数	对象	使用场景
曝光时间	2D 图（纹理）	用于为点云添加纹理的 2D 图像
2D 图（深度源）曝光时间	2D 图（深度源）	内参检查
		设置感兴趣区域
		进行手眼标定

### 曝光时间 / 2D 图 (深度源) 曝光时间

- 曝光时间 / 2D 图 (深度源) 曝光时间影响亮度：
  - 曝光时间越长，图像亮度越高，通常在较暗的环境下使用。
  - 曝光时间越短，图像亮度越低，通常在较亮的环境下使用。
- 其他条件相同，仅 曝光时间 不同的 2D 图对比如下：



- 其他条件相同，仅 2D 图 (深度源) 曝光时间 不同的 2D 图对比如下：

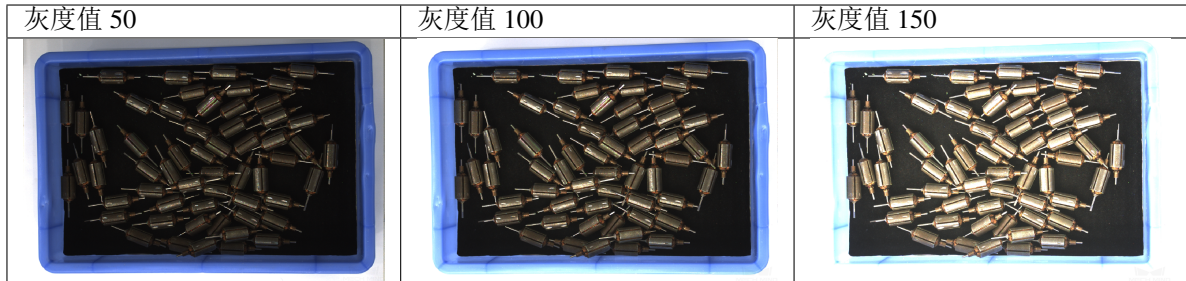


### 自动曝光

自动曝光采集的 2D 图，由 灰度值 与 自动曝光感兴趣区域 共同调节。

### 灰度值

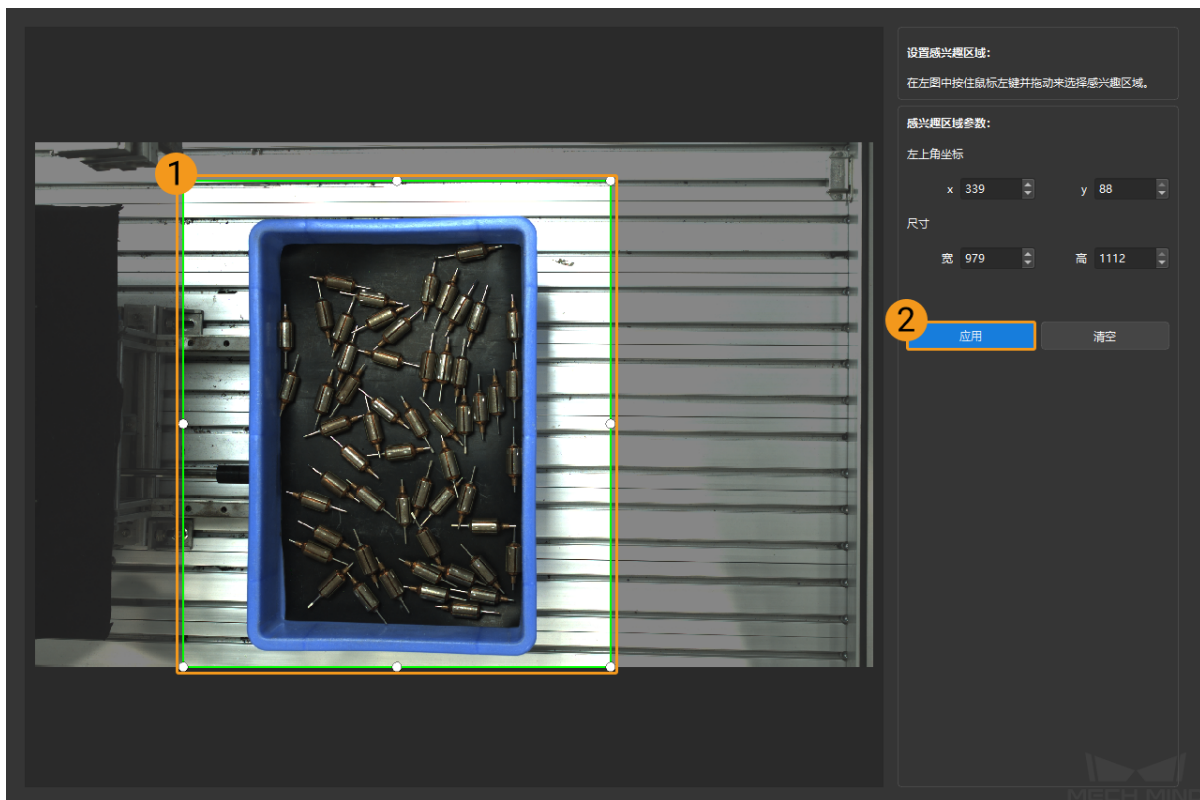
- 影响亮度。减小灰度值降低图片亮度，增加灰度值提高图片亮度。
- 其他条件相同，仅 灰度值 不同的 2D 图对比如下：



**注解：**黑白图像的灰度值相当于图像亮度；彩色图像的灰度值相当于每个颜色通道的亮度。

### 自动曝光感兴趣区域

- 设置自动曝光时的感兴趣区域，根据感兴趣区域自动调节 2D 图的亮度。未选择感兴趣区域时，相机根据整个视野自动计算曝光时间。
- 设置步骤：
  1. 双击 编辑 进入设置感兴趣区域页面。
  2. 在左侧选择感兴趣区域。点击 应用 完成，自动退出当前页面。



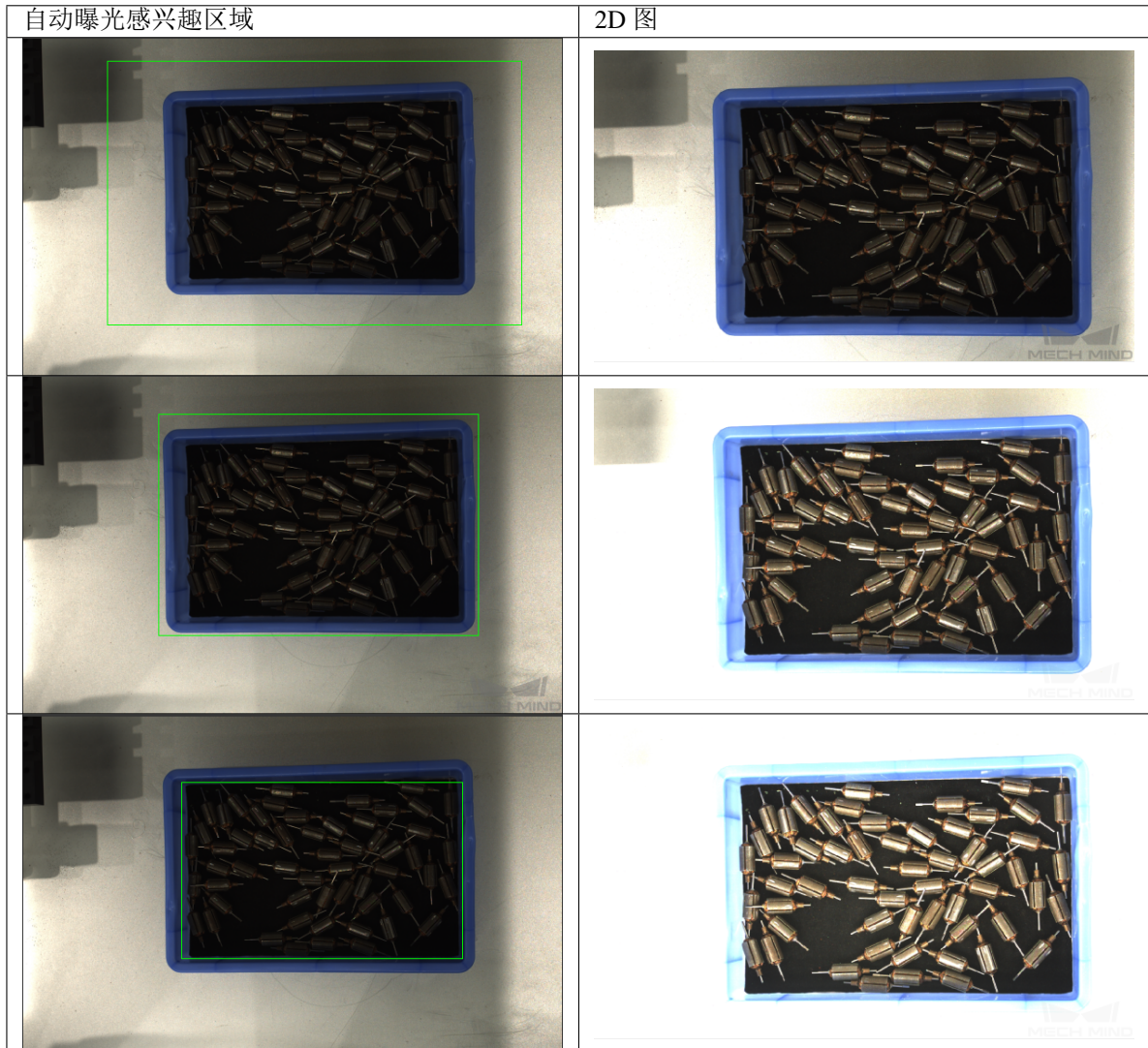


注解：设置的感兴趣区域可手动拖拽，调整大小。

3. 单击  或  重新获取图像，2D 图亮度更新。

提示：点击 清空，清除当前设置的感兴趣区域。

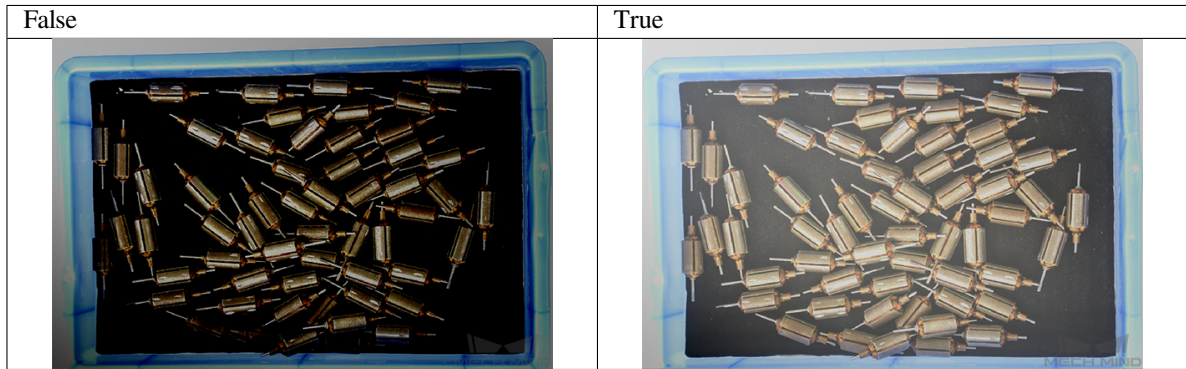
- 同一张图，设置不同的自动曝光感兴趣区域后，对比图如下：



## HDR

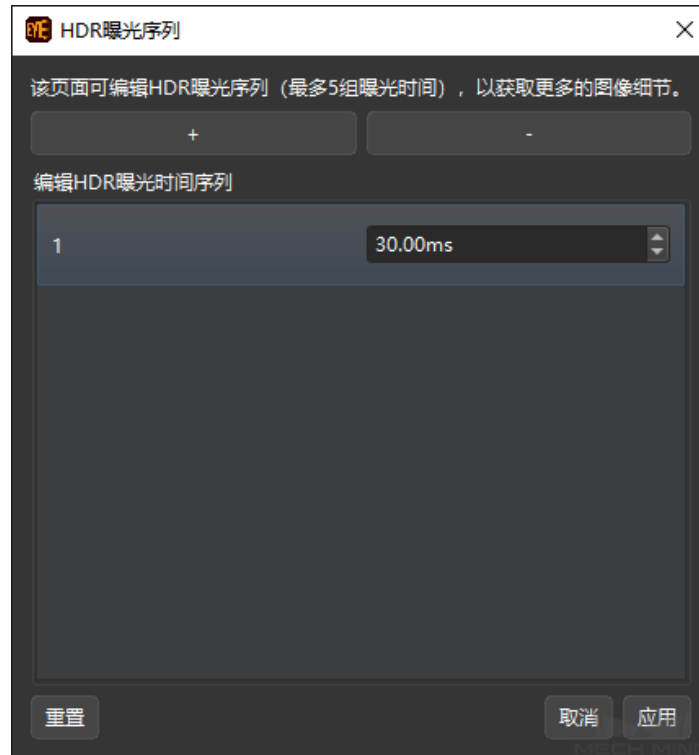
### 色调映射

- 色调映射用于增强图像色彩，以显示更多图像细节。勾选后开启色调映射。
- 色调映射开启前后对比图：



### 曝光时间序列

- 设置多个曝光时间，并融合各曝光时间下拍摄的图像，2D 图可保留更多明暗细节，以获得期望的图像效果。
- 步骤：
  1. 双击 编辑进入 HDR 曝光序列界面。



2. 单击 + ， 新增曝光序列并设置曝光时间。
3. 如需删除曝光序列，选中并单击 - 。
4. 曝光序列确认后，单击 应用， 设置生效。

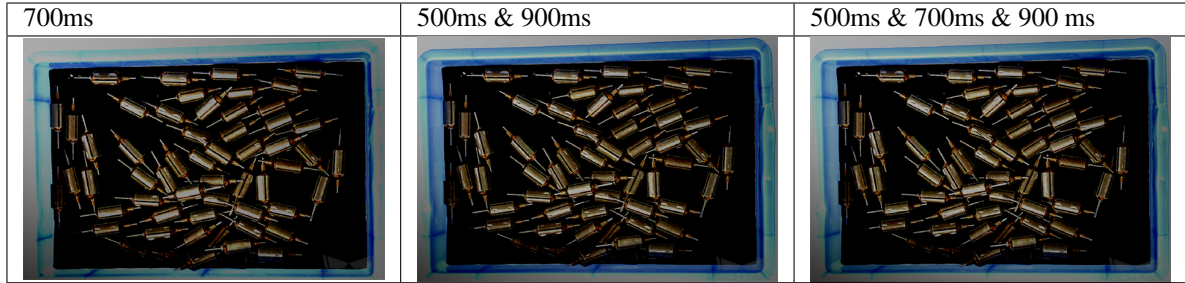
**注解:**

- 取消: 不保存当前设置，直接退出该页面。
- 重置: 清空当前所有的曝光时间序列。

- 一个曝光时间对比图:

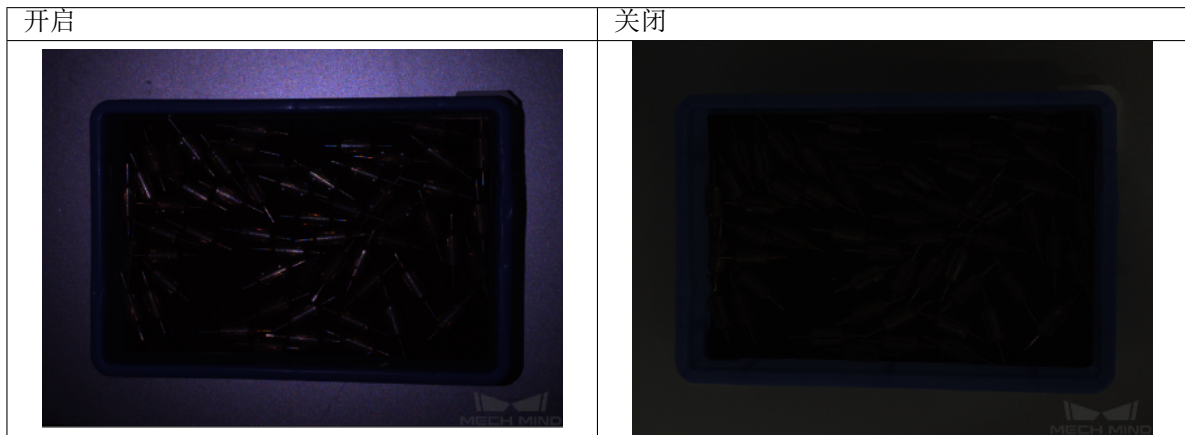


- 多个曝光时间对比图:

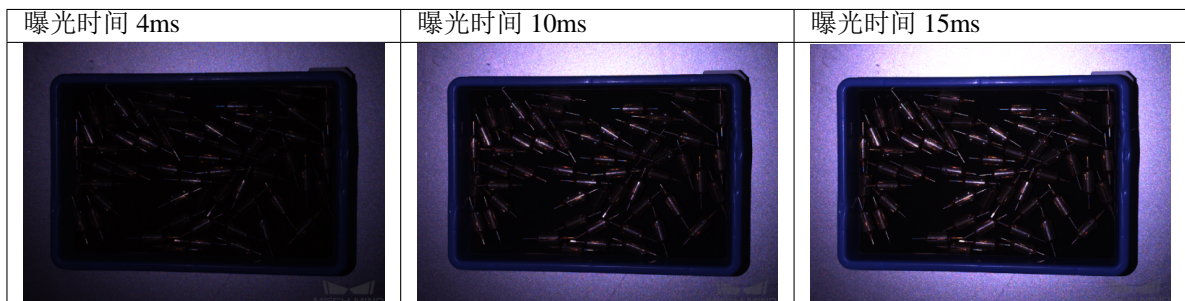


### Flash

- 此模式添加投影补光，适用于较暗的场景。
- **Flash** 与其他模式下采集的 2D 图对比：



- Flash 模式下的 2D 图，可通过调节 3D 参数的曝光时间调节明暗程度。不同曝光时间对比图如下：



提示：彩色相机采集图像时，由于现场光照条件限制，图像颜色与实际差别较大，请调节白平衡。详细操作请参考查看并配置 2D 相机。

## 高级参数调节

### 相机增益








- 属于 **3D 参数**，用于增加图像亮度。当设置曝光时间无法达到期望亮度时，建议增加 **相机增益**，但可能会引入图像噪声。范围：0~16dB。
- 不同 **相机增益**效果对比：



本章介绍了如何调节 2D 图，下一章将介绍如何通过常用参数调节深度图与点云。

### 7.3.3 深度图与点云常用参数调节

调节深度图与点云时所需要的参数如下所示：

	深度图	点云
3D 参数		
点云后处理		
深度范围		
感兴趣区域设置		

### 3D 参数

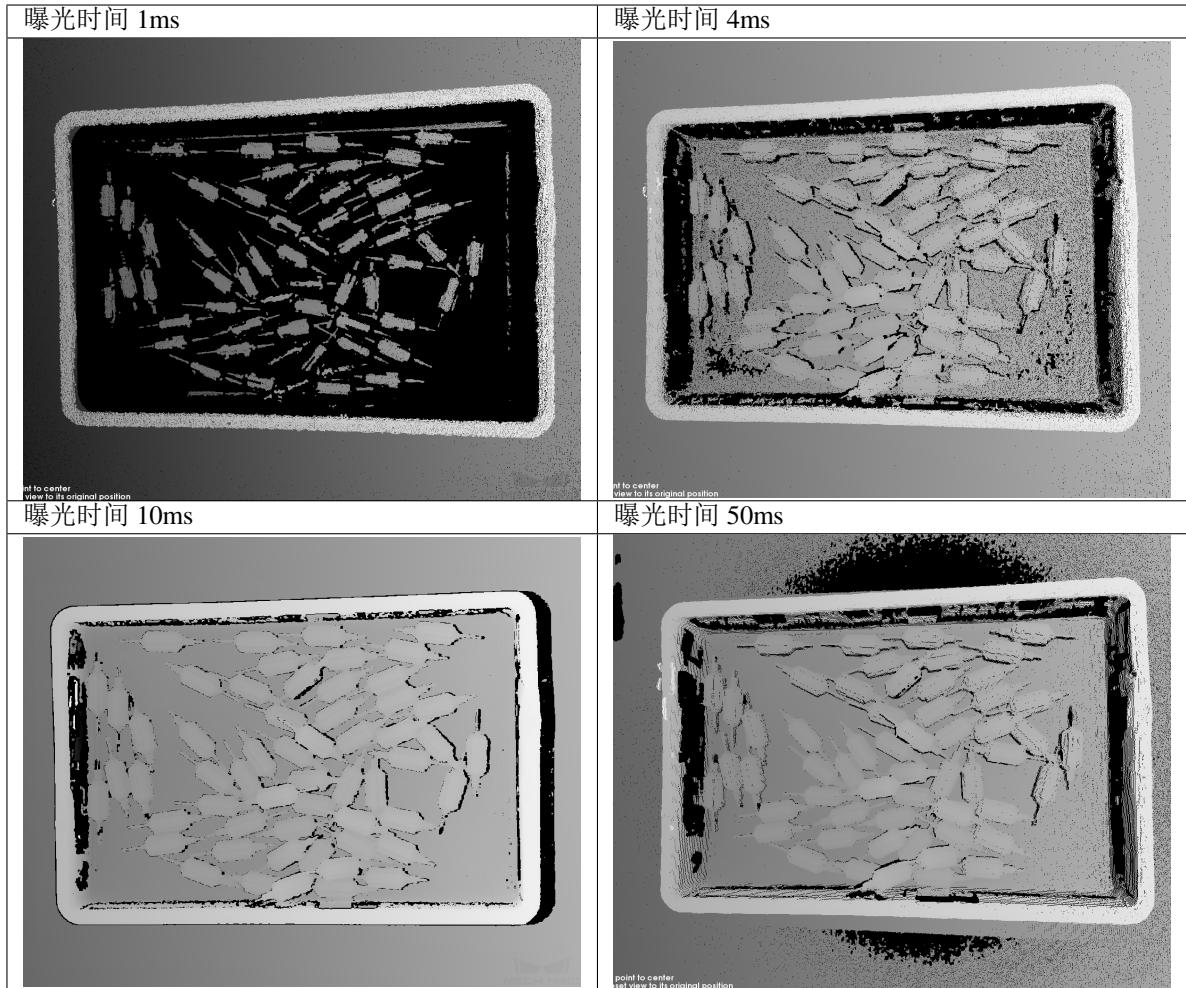
包含 **曝光次数**与 **曝光时间**，影响用于计算深度数据的图像，从而影响深度图及点云质量。

#### 曝光次数

- 设置曝光时间的个数。曝光次数大于 1 时，需设置多个曝光时间。
- 在不同曝光时间下分别拍摄，通过融合所有图像来计算深度。增加曝光次数可提升深度数据的完整性，但处理时间变长。
- 值：1~3。

#### 曝光时间

- 设置采集深度信息时的曝光时间，需根据曝光次数分别设置曝光时间。
- 深色物体通常使用较长曝光时间，浅色物体通常使用较短曝光时间。
- 曝光时间过长或过短都会导致信息缺失。不同 **曝光时间**对比图如下：


**注解:**

- 黑色部分缺少对应的物体点云。
- 曝光次数越多，获取深度图与点云所需要的时间越长。请在保证图像质量的情况下，尽量减少曝光次数。
- 激光相机的曝光时间须为 4 的倍数，输入值将被自动调整。Laser 系列可设置的最小值为 4ms，其他激光相机可设置的最小值为 8ms。

## 拍摄模式

提示：仅适用于 UHP 系列相机。

- 设置 UHP 系列相机的拍摄模式。推荐使用 **Camera 1** 或 **Merge** 模式。
- 值列表：

值列表	说明
Camera1	默认值。使用 Camera1 拍摄。 只有 Camera1 模式支持设置感兴趣区域。
Camera2	使用 Camera2 拍摄。
Merge	同时使用两个 2D 相机拍摄，并通过融合图像生成深度图和点云。 2D 图通过 Camera1 拍摄。

## 点云后处理

调节点云后处理分组下的参数，可优化点云质量。

- 点云调参原则
- 表面平滑
- 离群点去除
- 噪点去除
- 边缘保持
- 条纹对比度阈值

## 点云调参原则

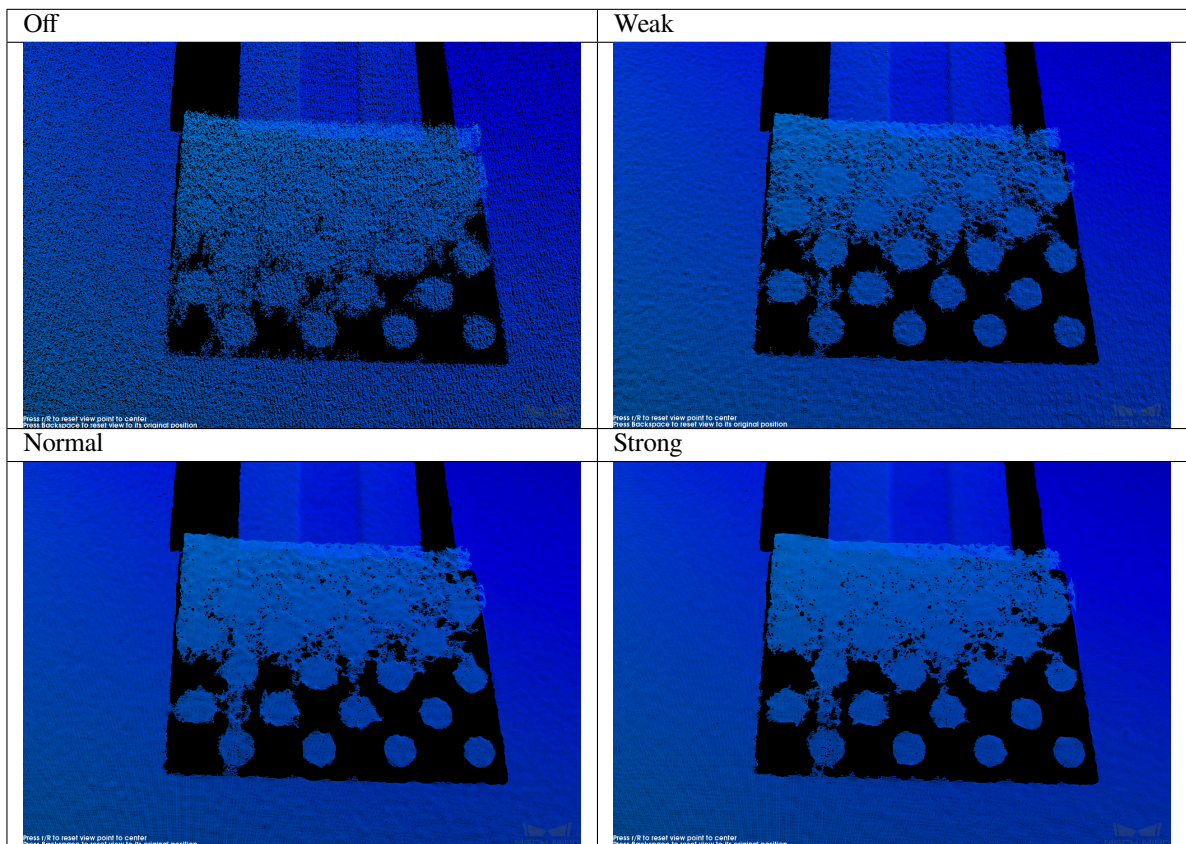
遵循以下调参原则可减少相机采集时间，优化节拍。

1. 优先调节 **离群点去除**。该参数各强度计算时间基本相同，即使使用较高强度也不会增加太多计算时间。
2. 建议使用低强度 **表面平滑** 与 **噪点去除**。这两个参数强度越高，计算时间越长。



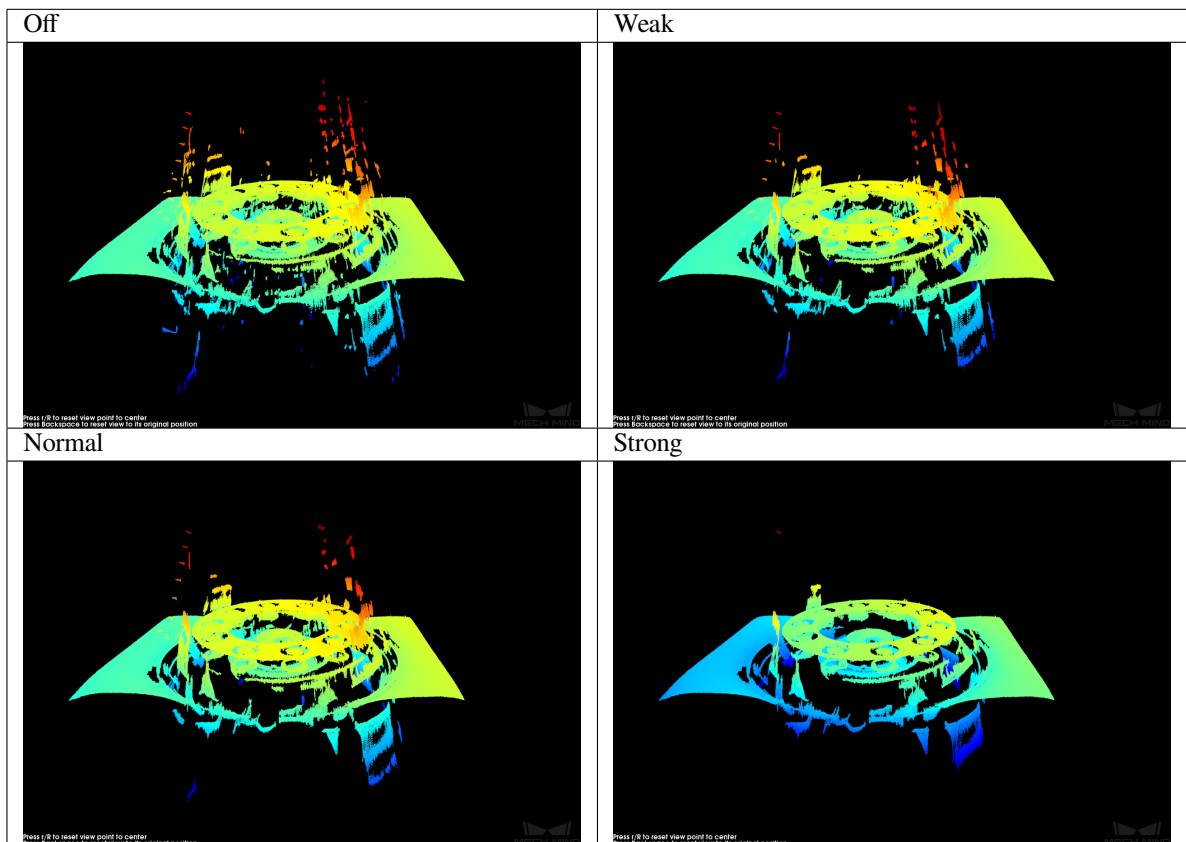
## 表面平滑

- 表面平滑可减少点云中的深度波动，使点云更接近真实的物体表面，但使用表面平滑会损失部分物体表面细节。
- 值列表：
  - Off
  - Weak
  - Normal
  - Strong
- 默认值：Off
- 调节说明：
  - 表面平滑强度越高，损失的物体表面细节越多；表面平滑强度越低，损失的物体表面细节越少。
  - 表面平滑强度越高，计算时间越长；强度越低，计算时间越短。
- 不同强度效果图对比：



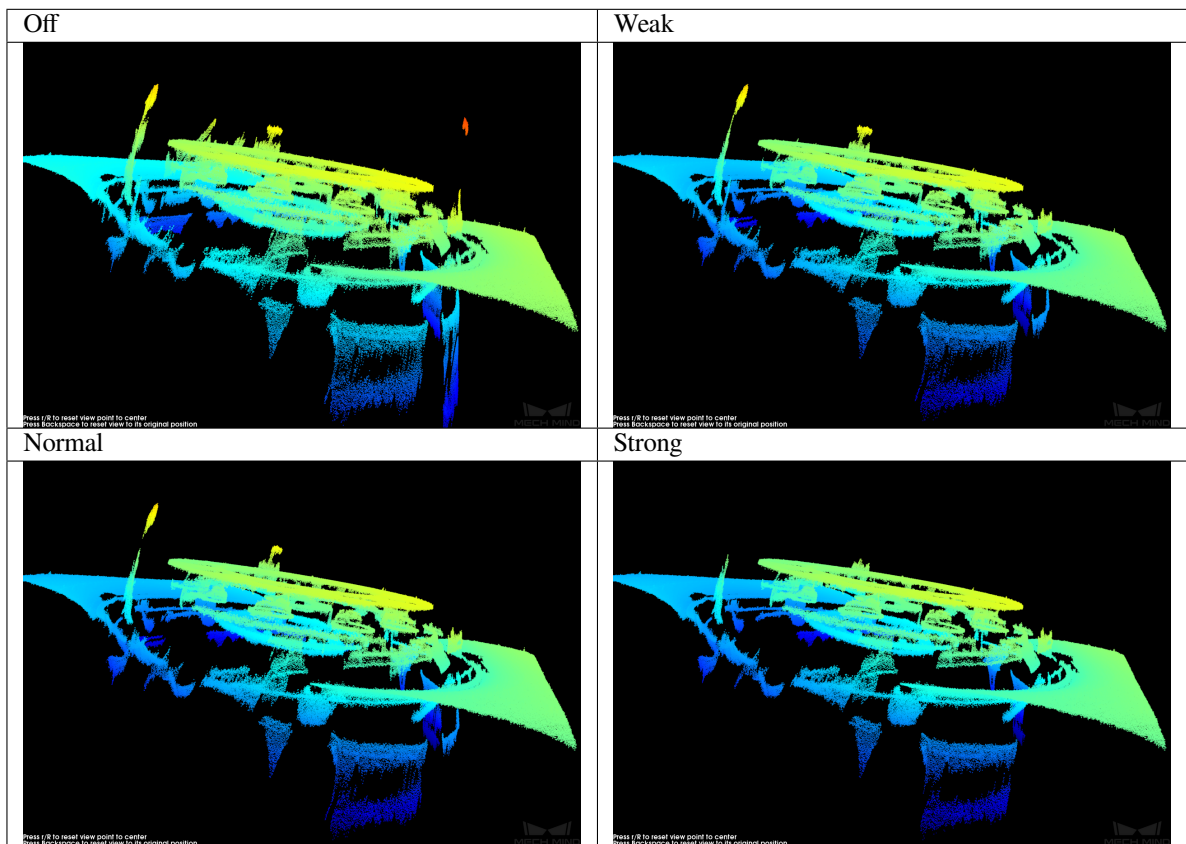
## 离群点去除

- 该参数用于去除该参数用于去除点云中的离群点。离群点为游离于物体点云之外的成团的点。
- 值列表：
  - Off
  - Weak
  - Normal
  - Strong
- 默认值：Weak
- 调节说明：
  - 离群点去除强度越高，去除的离群点越多；离群点去除强度越低，去除的离群点越少。
  - 物体包含多个组成部分时，高强度的离群点去除可能会去除部分物体点云。比如物体为水杯或茶壶时，使用离群点去除可能去除掉把手部分的点云。
- 不同强度效果图对比：



## 噪点去除

- 该参数可去除物体表面附近的噪点。噪点为位于物体表面附近的离散点。
- 值列表：
  - Off
  - Weak
  - Normal
  - Strong
- 默认值：Weak
- 调节说明：
  - 噪点去除强度越高，去除的噪点越多，但可能腐蚀物体表面特征；噪点去除强度越低，去除的噪点越少，物体表面特征保存的越完整。
  - 噪点去除强度越高，计算时间越长；噪点去除强度越低，计算时间越短。
- 不同强度效果图对比：



提示：如该功能移除了所需点云，可将噪点去除强度调低，但将保留更多噪点。

## 边缘保持

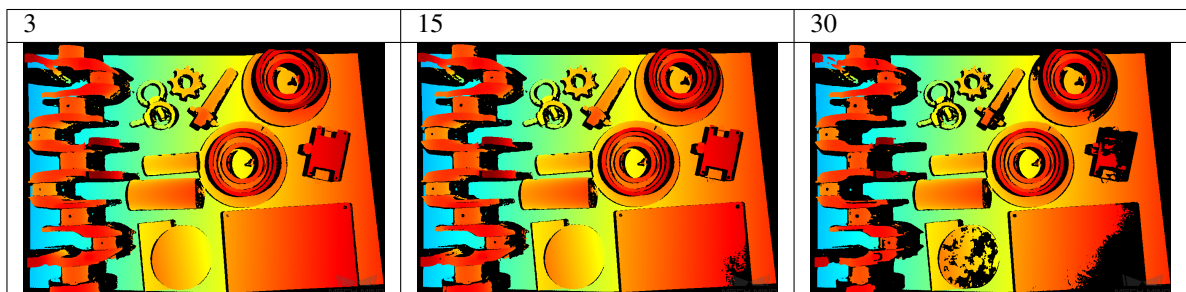
注解：该参数为 大师级别下的参数。

- 该参数在进行表面平滑时将保持物体边缘的锐利度。
- 值列表：
  - Sharp
  - Normal
  - Smooth
- 默认值：Normal
- 调节说明：
  - Sharp：最大程度保持物体边缘的锐利度，但表面平滑的效果较差。
  - Normal：在保持边缘的同时达到较好的表面平滑效果。
  - Smooth：不进行边缘保持。表面平滑效果最好，但物体边缘会失真。

## 条纹对比度阈值

提示：调节 表面平滑与 噪点去除后仍无法获得需要的点云时，再调节 条纹对比度阈值。

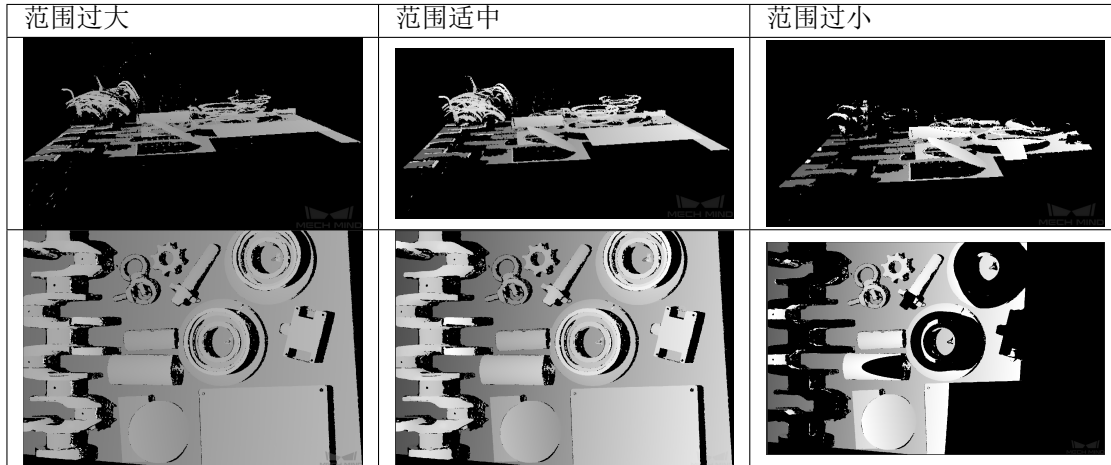
- 调大可过滤图像中的噪点，但可能造成较暗的物体也被滤除。
- 推荐值为 3。
- 不同强度效果图对比：



## 深度范围

设置 Z 向感兴趣区域，可在相机的工作距离范围内设置 **深度范围**，滤除 **深度范围**外的数据。

不同 **深度范围**效果对比：



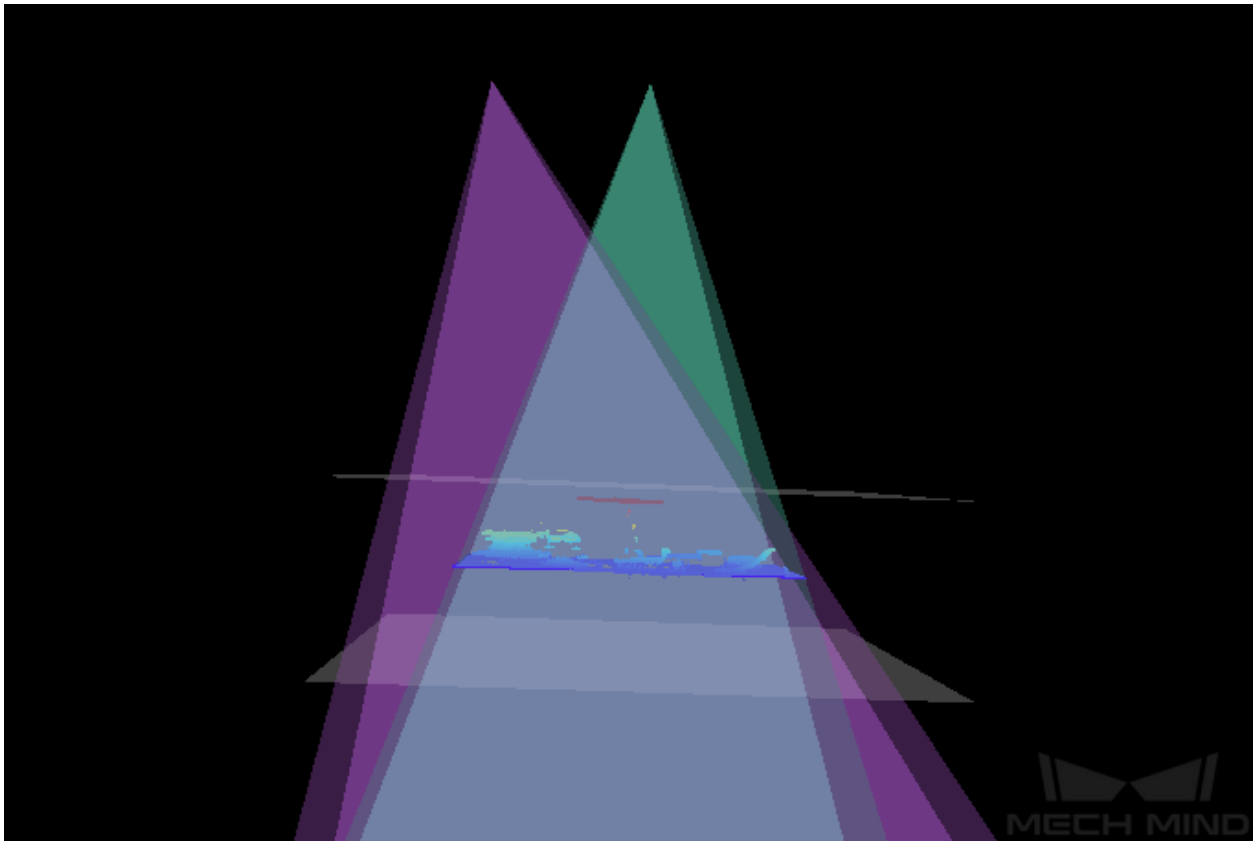
提示：

- **深度范围**需调节至合适范围，保证深度图与点云的完整；范围不可过大，否则可能造成干扰；也不可过小，否则可能造成关键部分缺失。
- 深度图调整完成后，可使用**深度图分析器**查看深度图质量。

## 设置深度范围

双击编辑，进入设置深度范围工具，此工具可进行可视化深度范围调节。调节步骤如下：

1. 调节点云位置：按住鼠标左键，向上移动鼠标旋转点云，向下滚动鼠标滚轮缩小点云，调整到可看到划分深度上下限的两个灰色长方形即可。



---

**提示：** 点云的操作详见[缩放和移动](#)。

---

2. 调节深度范围：调整数轴上的滑块，调整大致范围；再输入数值，调节准确范围。
3. 设置完成后，单击 **保存**。

---

**提示：**

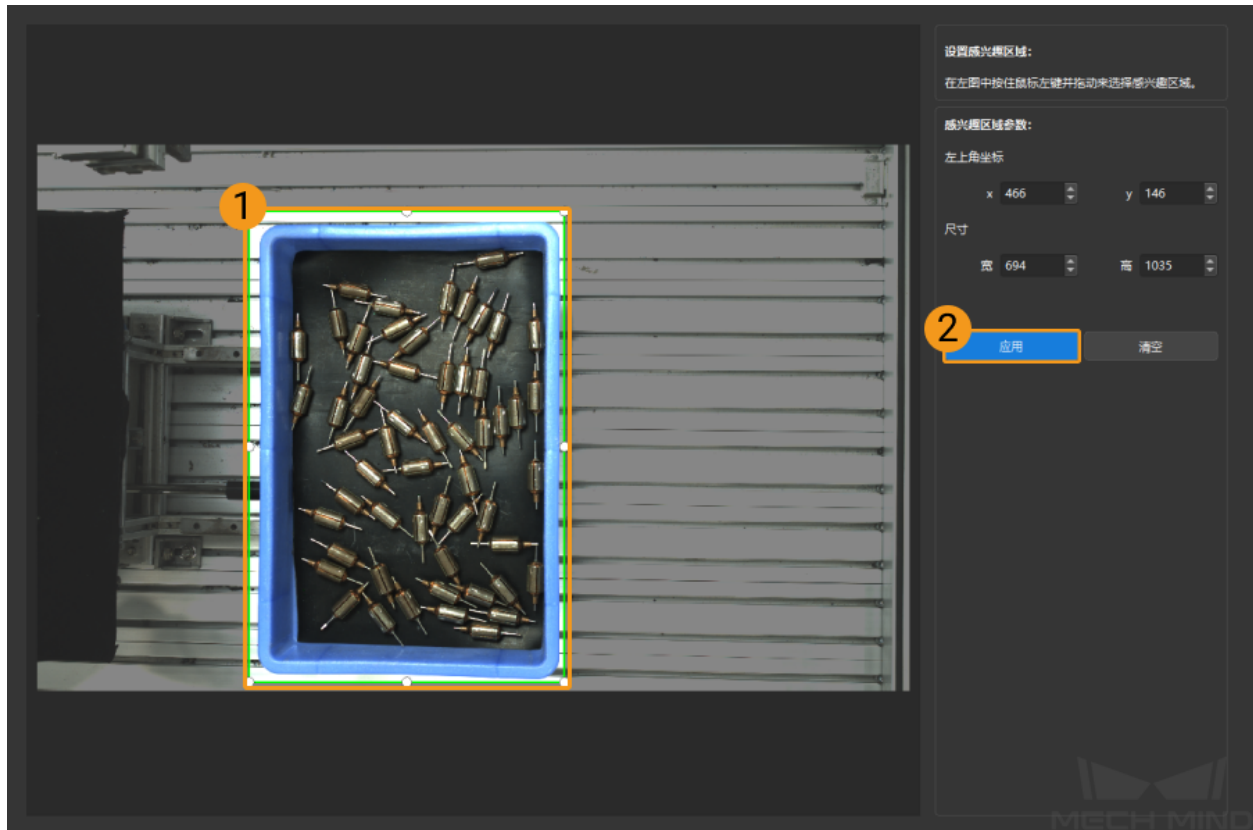
- 单击 **推荐值**，深度范围调整为当前相机的推荐工作距离。
  - 如深度范围不理想，单击 **重置**，深度范围恢复为上次保存的数值。
- 
-

## 感兴趣区域设置

设置深度图和点云 XOY 方向的感兴趣区域，滤除感兴趣区域外的点。



步骤：

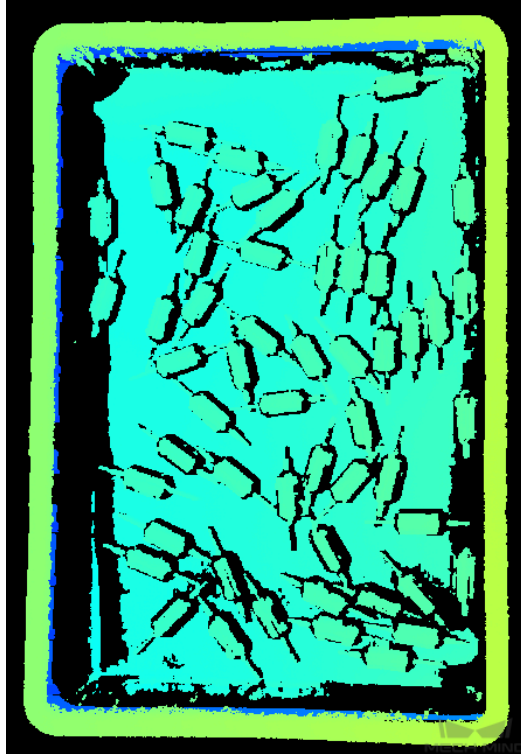
1. 双击 编辑 进入 设置感兴趣区域 页面。
2. 在左侧框选感兴趣区域，单击 应用 设置生效，并自动退出当前页面。



提示：

- 设置的感兴趣区域可手动拖拽，调整大小。
- 使用 DEEP 和 LSR 系列相机时，如果 2D 图像过暗或过亮，请查看 2D 图（深度源）曝光调节。

3. 单击  或  重新获取图像，深度图与点云更新，如下图所示。



---

**提示：**如现场采集效果不佳，可保存相机原始数据，并反馈给技术支持。

---

本节介绍了使用常用参数调节深度图与点云，下一页将介绍如何使用高级参数调节深度图与点云。

### 7.3.4 深度图与点云高级参数调节

高级参数仅 **专家**与 **大师**级别时可见。部分相机有定制参数，请根据相机设置相关参数。

---

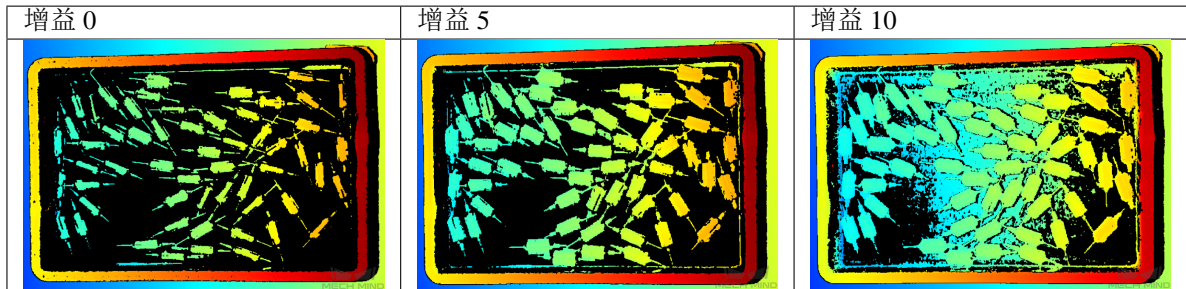
**提示：**以下参数均属于 **3D** 参数。

---



## 相机增益

- 用于增加图像亮度。当设置曝光时间无法达到期望亮度时，建议增加 **相机增益**，但可能会引入图像噪声。
- 值范围：0~16dB。
- 其他条件相同，**相机增益**不同时的效果对比：



## DLP 相机专属参数

### 投影

#### 投影光亮度

- 投影仪投射的结构光亮度。
- 值列表：

值列表	说明
High	高亮度，适用于采集深色物体图像。
Normal	正常亮度，适用于采集普通物体图像。
Low	低亮度，适用于采集反光物体图像。

## 专属参数

### 编码模式

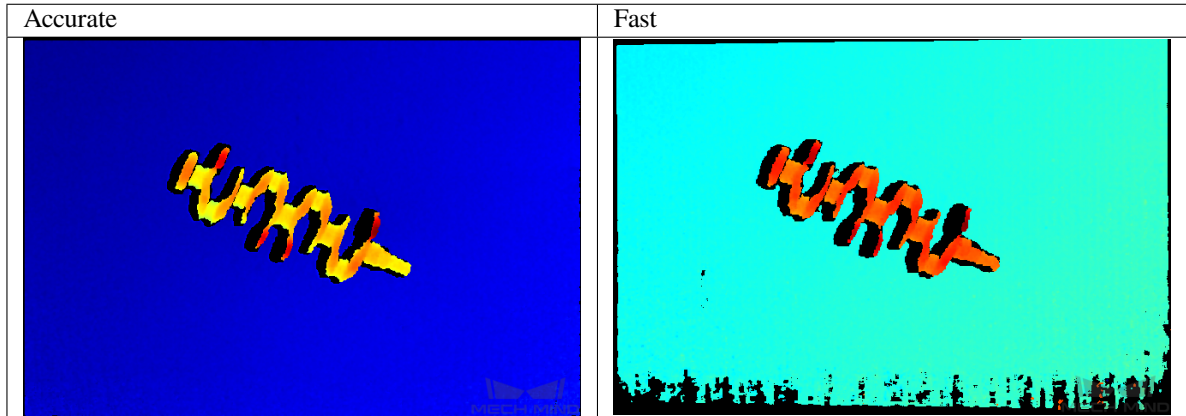
提示：适用相机型号：Nano (V3)、Pro XS (V3)、NANO (V4)、PRO S (V4) 与 PRO M (V4)。

- 影响采集速度与深度数据质量。

- 值列表:

值列表	说明
Fast	采图快, 但深度图和点云质量较差。
Accurate	采图慢, 但深度图和点云质量较好。

- 同一场景在不同 编码模式下的对比:



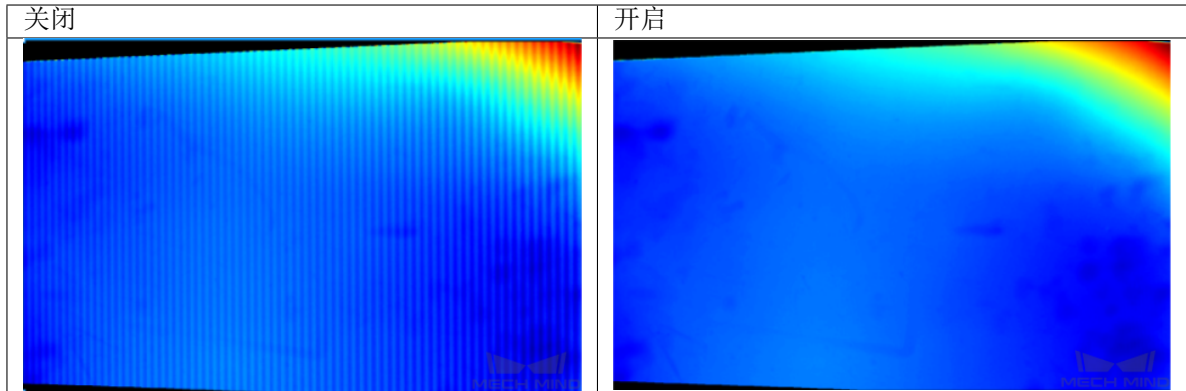
### 抗频闪模式

提示: 适用相机型号: Nano (V3)、NANO (V4)、PRO S (V4) 与 PRO M (V4)。

- 频闪是指环境灯光的快速周期性明暗变化。该现象会导致深度数据发生波动。通过调整结构光的投影频率, 可减少此类波动。
- 值列表:
  - Off
  - AC50Hz
  - AC60Hz

注解: 请根据所在国家的交流电频率选择。大部分国家或地区的交流电频率为 50Hz, 美国和部分亚洲国家的交流电频率为 60Hz。

- 关闭与开启抗频闪模式时深度图对比:



## 激光相机专属参数

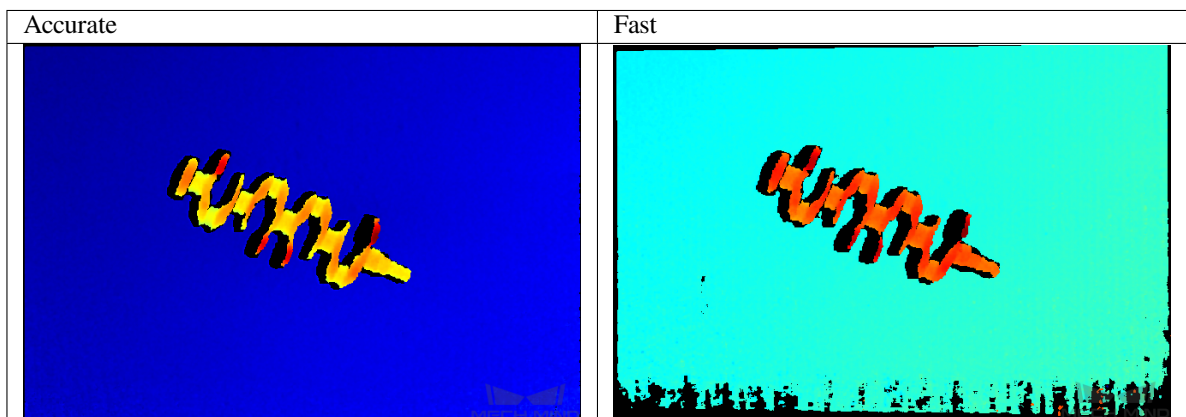
### 激光

#### 编码模式

- 影响采集速度与深度数据质量。
- 值列表：

值列表	说明
Fast	采图快，但深度数据质量较差。
Accurate	采图慢，但深度数据质量较好。

- 同一场景在不同 编码模式下的对比：



## 激光强度

- 设置激光器的投影强度，影响结构光的亮度。建议对深色物体使用高强度，对反光物体使用低强度。
- 数值越大，强度越高；数值越小，强度越低。
- 值范围：50~100%。一般情况下设定为 100% 即可。

本章介绍了深度图与点云的高级参数调节。至此，如何调节参数已全部介绍完成。下一章将介绍如何保存数据。

## 7.4 数据保存

可保存 2D 图、深度图和点云，也可以保存相机的原始数据。

- 保存图像
- 保存相机原始数据

### 7.4.1 保存图像

保存 2D 图、深度图和点云。

**注解：**DEEP (V4) 与 LSR (V4) 系列相机，仅 2D 图（纹理）可以保存，2D 图（深度源）不可保存。

单击相机工具栏的 ，进入保存图像界面，如下图所示。

名称	说明
保存路径	图像存储位置，需自行选择
当前图片序号	当图像命名相同时，自动编号，为确保图像名不重复
图片类型	可根据实际情况，自行选择存储类型：2D 图、深度图或点云
覆盖同名文件	选中后，文件重名时直接覆盖
使用上次拍摄图片	选中可保存上次拍摄图片；取消后需重新拍摄图片再保存
保存	保存图片
在文件夹中显示	跳转至存储图像所在的文件夹

**提示：**选中图片类型后，才可单击 保存。

## 7.4.2 保存相机原始数据

依次单击 文件 ▶ 保存相机原始数据，将以 **.mraw** 格式保存相机拍摄的原始数据。

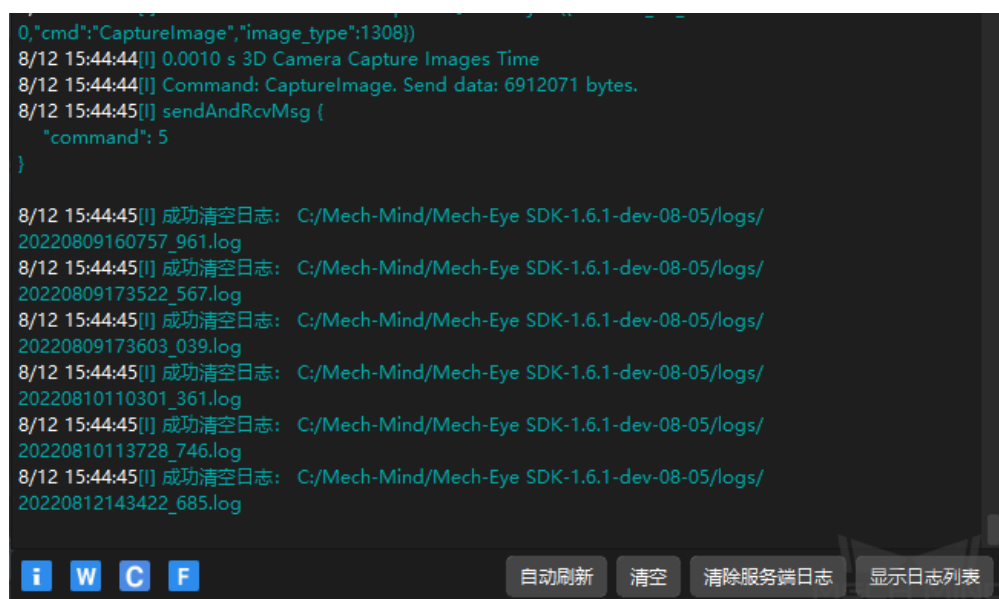
相机原始数据可以帮助技术支持定位问题的原因。联系技术支持前，请保存出现问题的相机原始数据。

另外，如果您想之后再调整 点云后处理、深度范围和感兴趣区域参数，通过 文件 ▶ 启动虚拟相机，即可加载保存的原始数据，并调整以上参数。使用虚拟相机时也可保存 2D 图、深度图和点云。

## 7.5 日志管理

日志管理用于管理软件的所有日志信息。当软件出现故障时，可通过日志管理排查问题。

可在相机工具栏中单击 显示日志 打开。日志管理可查看、清除或导出日志信息。日志管理界面如下图所示。



```
0,"cmd":"CaptureImage","image_type":1308))
8/12 15:44:44[!] 0.0010 s 3D Camera Capture Images Time
8/12 15:44:44[!] Command: CaptureImage. Send data: 6912071 bytes.
8/12 15:44:45[!] sendAndRcvMsg {
  "command": 5
}

8/12 15:44:45[!] 成功清空日志: C:/Mech-Mind/Mech-Eye SDK-1.6.1-dev-08-05/logs/
20220809160757_961.log
8/12 15:44:45[!] 成功清空日志: C:/Mech-Mind/Mech-Eye SDK-1.6.1-dev-08-05/logs/
20220809173522_567.log
8/12 15:44:45[!] 成功清空日志: C:/Mech-Mind/Mech-Eye SDK-1.6.1-dev-08-05/logs/
20220809173603_039.log
8/12 15:44:45[!] 成功清空日志: C:/Mech-Mind/Mech-Eye SDK-1.6.1-dev-08-05/logs/
20220810110301_361.log
8/12 15:44:45[!] 成功清空日志: C:/Mech-Mind/Mech-Eye SDK-1.6.1-dev-08-05/logs/
20220810113728_746.log
8/12 15:44:45[!] 成功清空日志: C:/Mech-Mind/Mech-Eye SDK-1.6.1-dev-08-05/logs/
20220812143422_685.log
```

Log management interface controls: i, W, C, F, 自动刷新, 清空, 清除服务端日志, 显示日志列表

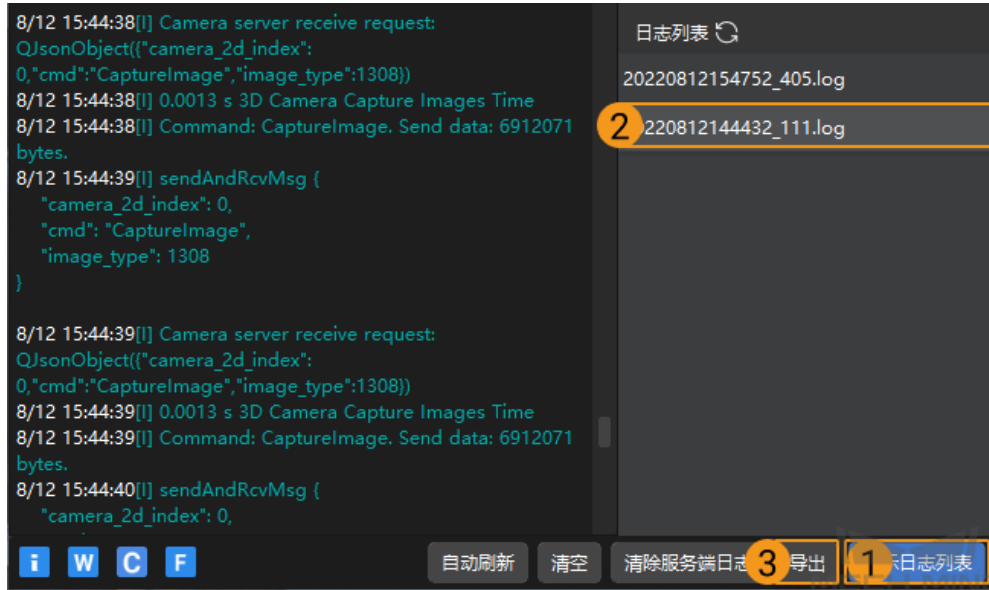
### 7.5.1 日志等级

日志等级分为 i、W、C 及 F 四个等级。

- **i**: INFO，描述信息，描述应用运行过程。
- **W**: WARNING，警告信息，指潜在的危险状态。
- **C**: CRITICAL，严重错误，指出错误事件，但可能还能继续运行。
- **F**: FATAL，致命信息，非常严重，且可能导致应用终止运行。

## 7.5.2 常用操作

- 自动刷新：自动更新日志。
- 清空：删除软件界面的日志。
- 清除服务端日志：同时删除相机及软件界面的日志。
- 导出日志：单击 显示日志列表进入日志列表界面。选中日志，单击 导出，导出日志。



## 内置工具

软件内置工具可通过菜单栏中的 **工具** 菜单打开：

工具名称	功能描述
内参工具	检查相机内参是否与出厂设置一致。
曝光助手	获取推荐的 <b>3D 参数</b> 。
深度图分析器	检验深度图成像质量。
查看并配置 2D 相机	查看和设置 2D 相机的相关参数。
相机固件升级	升级相机固件。
视野计算器	根据工作距离计算相机的视野。
自定义坐标系	设置自定义坐标系，用于查看深度图和点云。
相机管理器	查看相机类型，日期时间，CPU 温度及投影仪温度等。
满垛模拟器	用于确认，满垛时垛顶层是否全部位于相机视野内。

勾选 **视图** 菜单中的 **工具栏**，可显示包含常用工具的工具栏，如下图所示。



单击 **更多工具**，可更改工具的显示状态。勾选显示，取消勾选隐藏。设置完成，单击 **OK** 保存修改。

## 8.1 内参工具

该工具用于检查相机的内参是否准确、矫正内参及恢复出厂内参。只有相机内参准确，才能保证计算得出的各点坐标准确。

**注解：**对于精度要求很高的项目，可开启高精度模式，根据更严格的标准检查内参。高精度模式在 **管理员模式** 下可见，如需使用，请咨询技术支持。

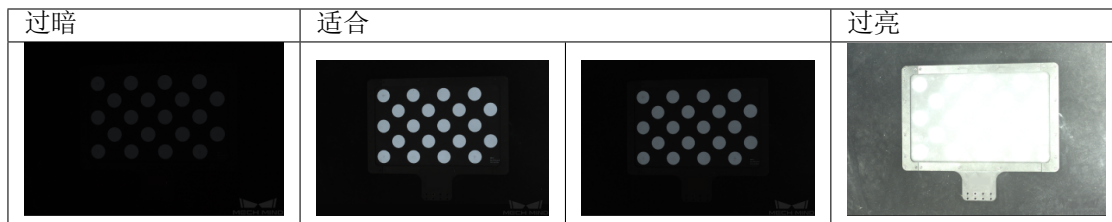
### 8.1.1 准备工作

检查相机内参前，请准备好相机随附的标定板。

将标定板置于相机视野内，获取标定板的 2D 图和深度图。请确保获取的 2D 图和深度图符合如下标准：

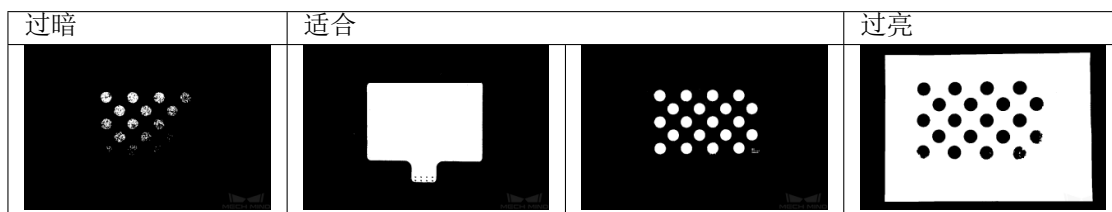
- 标定板上圆所在的区域拍摄完整。
- 2D 图不过亮或过暗，标定板上的圆清晰完整。
- 深度图中标定板上的圆完整。

标定板 2D 图示例如下：



如获取的 2D 图不符合标准，请调节 2D 参数。

标定板深度图示例如下：



如获取的深度图不符合标准，请调节深度图相关参数。



## 8.1.2 检查内参

单击菜单栏中的 **工具** 菜单，选择 **内参工具**，进入 **内参工具** 窗口。

使用 **内参工具** 检查相机内参时，请执行如下步骤。

1. 单击 **1. 放置标定板并检查图像质量** 中的 **采集图像**，获取标定板的 2D 图和深度图。确认图像是否符合标准。
2. 在 **2. 选择放置的标定板型号** 中设置标定板的规格：
  - 如使用标准型号的标定板，单击右侧小三角选择标定板型号。
  - 如使用定制标定板，请先选择 **自定义**，再根据工程师提供的信息设置 **型号**、**行**、**列** 及 **间距**。
3. 在 **3. 检查内参** 中单击 **检查内参**，开始检查相机内参。检查完成后自动弹出检查结果的弹窗。
  - 如弹窗中内参检查结果数据显示绿色，则内参检查合格，相机可正常使用。
  - 如弹窗中内参检查结果数据显示红色或出现其他错误提示，则内参误差较大或检查失败，请根据弹窗内的提示解决问题。

**提示：**对于精度要求很高的项目，请切换至管理员模式。在 **3. 检查内参** 中勾选 **高精度模式**，并根据 **高精度检测距离** (相机到标定板的距离) 放置标定板。如不按该值放置标定板，可能导致相机内参检查失败。

## 8.1.3 解决检查内参可能出现的问题

检查内参时，可能遇到如下问题：

- 内参误差较大
- 未检测到标定板上的圆
- 深度图中没有有效的特征点

### 内参误差较大

**问题描述：**

内参检查结果弹窗数据显示红色，表示相机内参误差较大。

**解决方案：**

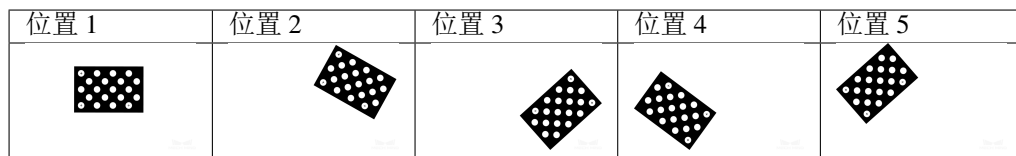
请尝试如下解决方案：

1. 检查标定板型号或规格是否正确。
  - 如错误，请输入正确的标定板型号或规格。然后，请重新检查相机内参。
  - 如无误，请查看下一步。
2. 检查 2D 图或深度图是否符合标准，请参考 **2D 图与深度图检查标准**。

- 如 2D 图不符合标准，请调节 2D 参数；如深度图不符合标准，请调节深度图相关参数。然后，请重新检查相机内参。
  - 如图像符合标准，请查看下一步。
3. 使用 矫正内参 功能矫正内参。使用步骤如下：
- (1) 将标定板置于相机视野内。
  - (2) 勾选 矫正内参 后，单击 采集数据。
  - (3) 移动标定板，再次单击 采集数据。

**提示：**

- 请添加至少 3 组数据。
- 移动标定板时，建议同时平移并旋转标定板，推荐位置为视野范围的中心与四个角共 5 组，如下图所示。



- (4) 添加数据完成后，单击 矫正内参。完成后将自动弹出矫正结果的弹窗。
  - 如内参矫正成功，单击该窗口中的 检查内参，重新进行内参检查。
  - 如内参矫正失败，单击该窗口中的 确定，重新尝试或联系技术支持。

## 未检测到标定板上的圆

### 问题描述：

内参检查结果弹窗提示“未检测到标定板上的圆”。

### 解决方案：

请尝试如下解决方案：

1. 检查标定板型号或规格是否正确。
  - 如错误，请输入正确的标定板型号或规格。然后，请重新检查相机内参。
  - 如无误，请查看下一步。
2. 检查 2D 图或深度图是否符合标准，请参考 2D 图与深度图检查标准。
  - 如 2D 图不符合标准，请调节 2D 参数；如深度图不符合标准，请调节深度图相关参数。然后，请重新检查相机内参。
  - 如图像符合标准，请查看下一步。
3. 使用 画辅助圆 功能辅助检测。使用步骤如下：
  - (1) 单击 3. 检查内参 中的 画辅助圆。

(2) 在 2D 图上找到一个完整的标定板上的圆。移动光标至该圆圆心位置，按住 `Ctrl` 并单击鼠标左键，向斜上或斜下方向移动光标调整辅助圆的大小。再次单击完成画圆。

---

**注解:**

- 仅需要画一个辅助圆，辅助圆应与标定板上圆的边缘尽量贴合。
  - 如需重新画辅助圆，请重新单击 **画辅助圆**。
  - 将光标移动至 2D 图上，滚动鼠标滚轮，可以放大或缩小 2D 图。
- 

(3) 在 **3. 检查内参** 中单击 **检查内参**，重新检查内参。

- 如内参检查成功，流程结束。
- 如内参检查结果弹窗仍提示“未检测到标定板上的圆”，请查看下一步。

4. 使用 **编辑特征检测参数** 功能辅助检测。使用步骤如下：

(1) 单击 **3. 检查内参** 中 **编辑特征检测参数**。

(2) 请根据实际情况调整显示的特征检测参数。

(3) 如无法拍摄到标定板上的所有圆，请勾选 **允许部分圆缺失**，并编辑新增的特征检测参数。

(4) 在 **3. 检查内参** 中单击 **检查内参**，重新检查内参。

- 如内参检查成功，流程结束。
- 如内参检查结果弹窗仍提示“未检测到标定板上的圆”，请联系技术支持。

## 深度图中没有有效的特征点

### 问题描述:

内参检查结果弹窗提示“深度图中没有有效的特征点”。

### 解决方案:

检查深度图是否符合标准，请参考 [2D 图与深度图检查标准](#)。

- 如深度图不符合标准，请调节 [深度图相关参数](#)。然后，请重新检查相机内参。
- 如深度图符合标准，请联系技术支持。

## 8.2 曝光助手

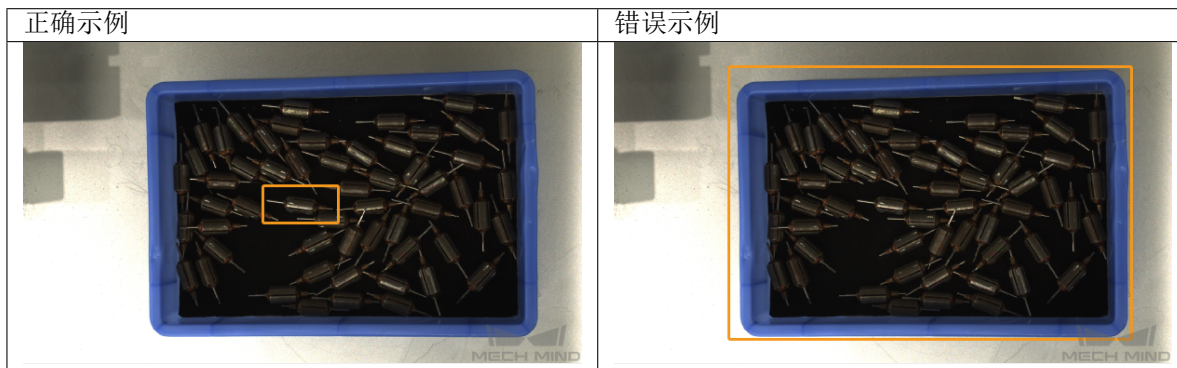
双击 3D 参数中的 **自动设置** 或单击工具栏中的 **曝光助手** 进入曝光助手页面。此工具用于获得最佳的曝光参数，即 **3D 参数**，包含 **曝光次数** 与 **曝光时间**。

## 8.2.1 设置步骤

选择工具 > 曝光助手进入曝光助手窗口，使用曝光助手获取曝光参数的步骤如下：

1. 在 采集点云页面中单击 单次采集获取图像，采集成功后单击 下一步。
2. 在 设置感兴趣区域页面中，软件将自动框选感兴趣区域。拖动锚点或在 手动设置中调节参数（如下图所示），均可调节感兴趣区域，完成后单击 下一步。

提示：框选感兴趣区域时，选中其中的一个物体即可。



3. 在 自动曝光页面中单击 开始，曝光助手开始计算。计算完成后，在 结果中可查看曝光次数与曝光时间。
4. 点击 完成，页面出现 确认弹窗。如需将结果添加至相机参数，单击弹窗中的 是。

## 8.3 查看并配置 2D 相机

此功能用于查看 2D 相机参数信息及配置。

### 8.3.1 查看 2D 相机信息

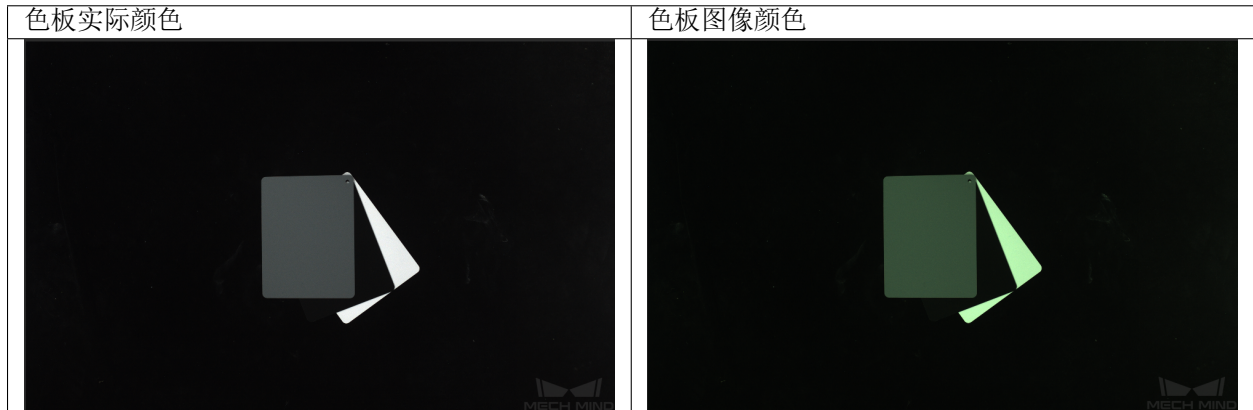
2D 相机信息一栏中可查看相机的序列号、分辨率、颜色类型、设备版本和固件版本信息。

### 8.3.2 白平衡调节

注解：仅用于彩色 2D 相机。选择 2D 相机后，查看 2D 相机信息中的 颜色类型可确认是否为彩色 2D 相机。

采集图像时，如果图像颜色与实际差别较大，需调节白平衡。否则 2D 图像及点云的颜色会失真，影响后续处理。在深度学习中，如果颜色失真的图像用于训练深度学习模型，其中的颜色偏误会被当做物体特征用于训练，从而影响后续模型表现。

## 对比示例



## 准备工作

调节白平衡时，需要使用灰色色板。请将色板放置在距离相机较近的位置采集图像，使得采集到的图像尽量仅包含色板。

## 操作步骤

请执行以下步骤调节彩色 2D 相机的白平衡：

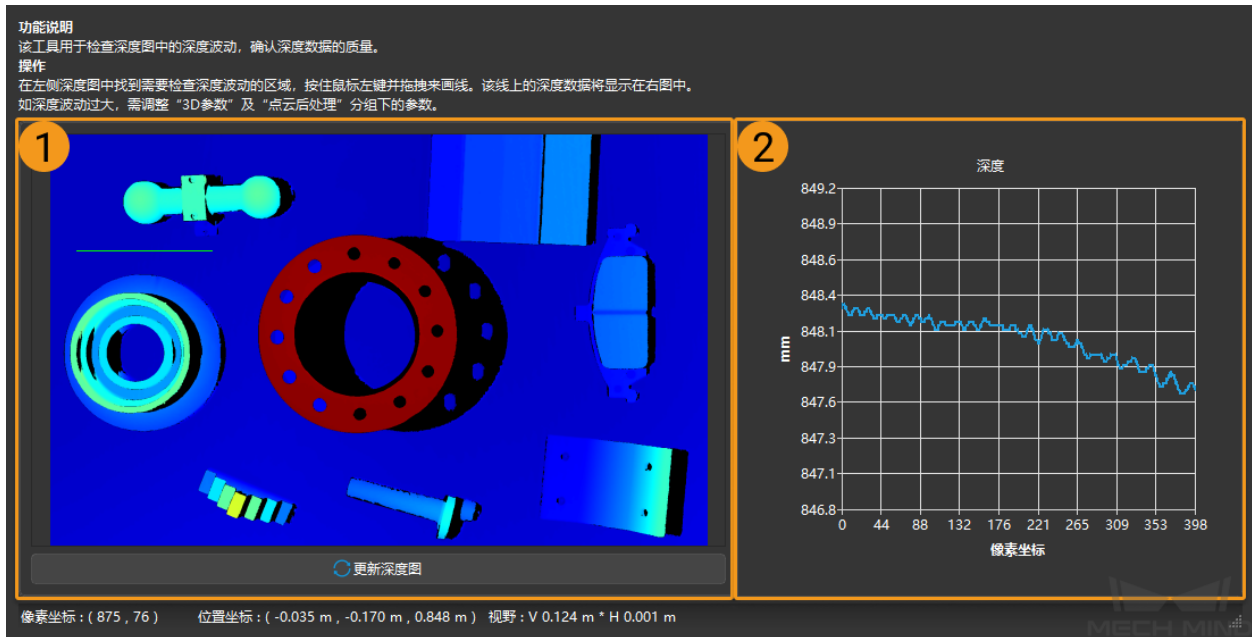
1. 选择要调节白平衡的 2D 相机，并点击 图像开启，相机将自动获取图像。
2. 在 **2D 相机配置**中，将 **自动白平衡**设置为 **Once** 或 **Continuous**，相机开始调节白平衡。
  - 对于环境光线相对恒定的场景，建议使用 **Once**。
  - 对于环境光线变化较大的场景，建议使用 **Continuous**。
3. 查看图像中灰色色板的颜色。
  - 如色板颜色存在明显偏差，请移动灰色色板，直至色板颜色无明显偏差。
  - 如色板颜色无明显偏差，请将 **自动白平衡**设置为 **Off**，并点击 图像关闭，完成白平衡调节。
4. 点击 **应用参数**，将配置保存到相机。

## 8.4 深度图分析器

此工具用于帮助检验深度图的成像质量。

## 8.4.1 界面介绍

深度图分析器界面如下图所示。



No.	名称	功能
1	深度图区域	显示相机采集到的深度图。
2	深度图波动显示区域	显示所选区域内的深度波动情况。

## 8.4.2 使用

1. 在深度图区域画线：点击鼠标左键画线（如上图中的绿线）。
2. 在深度图波动显示区域查看深度波动情况：显示绿线的深度波动情况。
3. 分析：请检查深度波动是否如实反应绿线的实际场景。如果是，则深度图质量良好；如果不是，请重新调节深度图相关参数。

## 8.5 相机固件升级

当软件与固件版本不一致时，请升级相机固件。

### 8.5.1 自动升级

当软件与固件版本不一致时，相机会自动检查并有固件升级提示。此时根据软件提示更新即可。

### 8.5.2 手动升级


如无提示，需手动升级相机固件。

1. 单击 **工具** ▶ **相机固件升级** 或者单击工具栏中的 **相机固件升级**，进入固件升级页面。



2. 单击 **升级固件**，出现固件升级提示，根据提示升级固件。

提示：

- 固件升级完成，需重启相机。
- 重启相机需花费几分钟，请耐心等待。这个过程中，相机固件版本仍显示为未升级固件版本。单击  刷新相机列表，相机显示为最新固件版本时，固件升级全部完成。

如升级失败，请重新升级；如仍不成功，请联系技术支持。

## 8.6 视野计算器

视野计算器用于辅助确定合适的相机安装高度。根据输入的工作距离，即可计算出相机在该工作距离下的视野高度与视野宽度。

### 8.6.1 计算相机视野

使用 视野计算器 工具获取相机视野时，请执行如下步骤。

1. 在 **选择相机型号** 中选择相机型号。

---

**注解：** 默认显示常用型号。单击 **其他型号**，可查看所有相机型号。

---

2. 在 **输入工作距离** 中输入工作距离，软件将根据当前工作距离计算 **视野高度** 与 **视野宽度**。如当前视野范围不符合实际需求，请执行下一步。

---

**注解：**

- 相机的工作距离范围为该相机的推荐工作距离，详见 [相机工作距离](#)。
  - 工作距离需在所选相机的推荐工作距离范围内，否则输入不生效。
- 

3. 单击右侧箭头按钮或滚动鼠标滚轮，调整工作距离，直至视野范围符合需求。根据该工作距离可确定相机的安装高度。

---

**提示：** 计算结果仅供参考，请以实际情况为准。

---



## 8.6.2 相机工作距离

各相机型号的工作距离范围和默认值如下（单位：m）。

相机型号	工作距离	
	默认值	范围
DEEP	3	1.2~3.5
LSR L	2.5	1.2~3
LSR S	1	0.5~1.5
Log M	2	0.8~2
Log S	1	0.5~1
NANO	0.5	0.3~0.6
PRO M	2	1~2
PRO S	1	0.5~1
PRO XS	0.5	0.3~0.6
UHP-140	0.3	0.28~0.32
Deep	3	1.2~3.5
Laser L	2.5	1.5~3
Laser L Enhanced	2.5	1.5~3
Pro M Enhanced	2	0.8~2
Pro S Enhanced	1	0.5~1

## 8.7 自定义坐标系

选取一个平面，设置自定义坐标系，软件会显示该坐标系下的深度图和点云。

设置完成后，深度图和点云中的自定义平面高度一致，颜色（深度）相同。

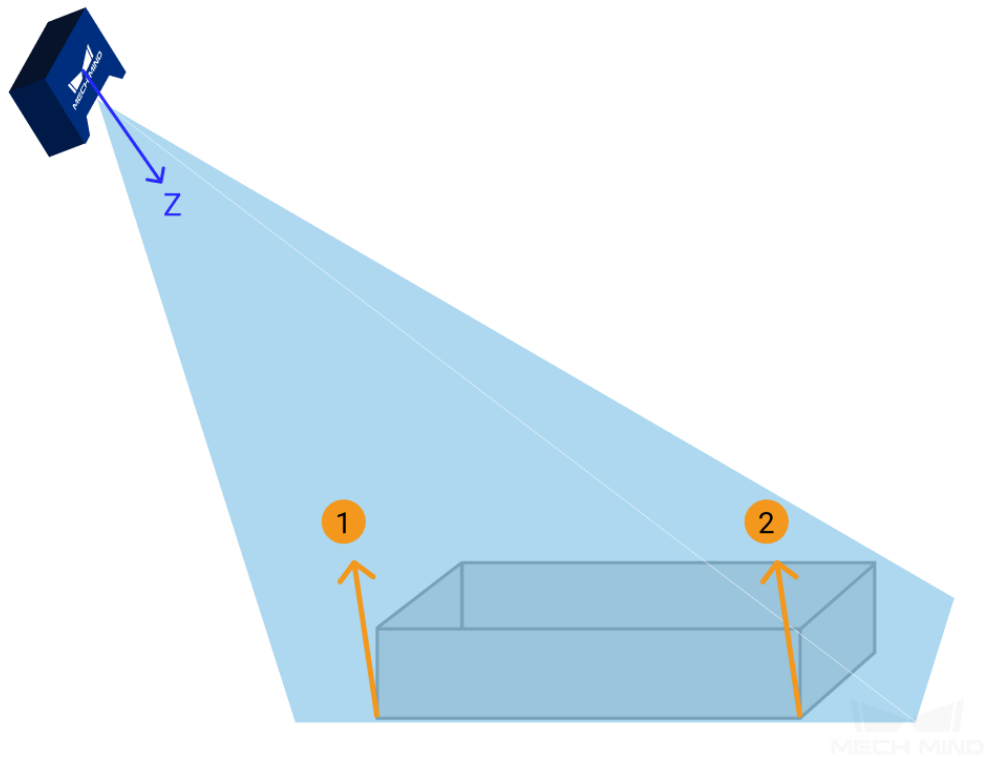
---

**注解：** 自定义坐标系仅影响 Mech-Eye Viewer 中的深度值显示。保存的深度图和点云中存储的深度值不受影响。

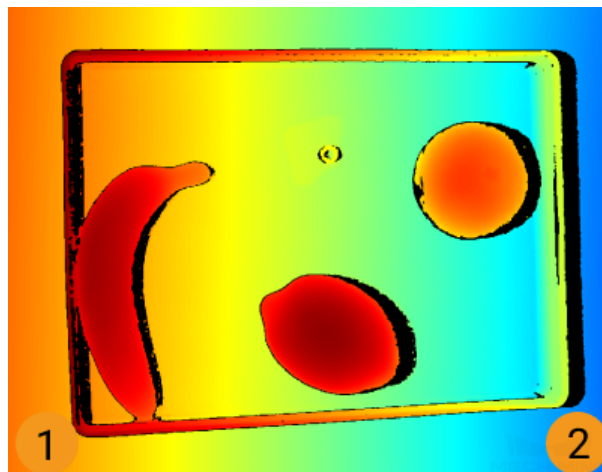
---

### 8.7.1 原理

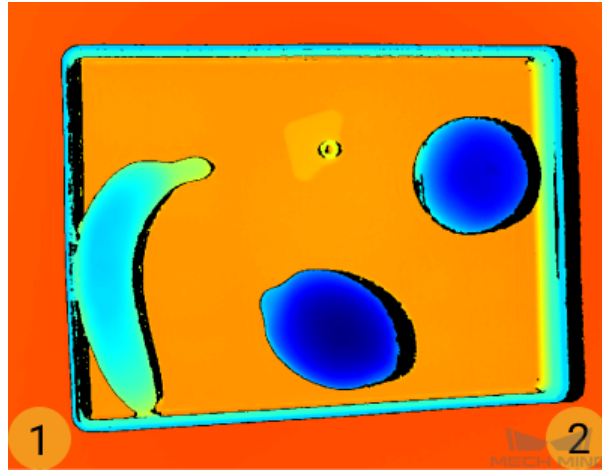
通常相机不会垂直对准目标物体安装。如下图所示，料筐为水平放置，1 与 2 在同一水平面。



采集图像，深度图中同一水平面的高度不相同，深度图显示为颜色不相同。



通过自定义坐标系，可调整深度图和点云的显示，使 1 和 2 高度相同，如下图所示。



## 8.7.2 选择平面

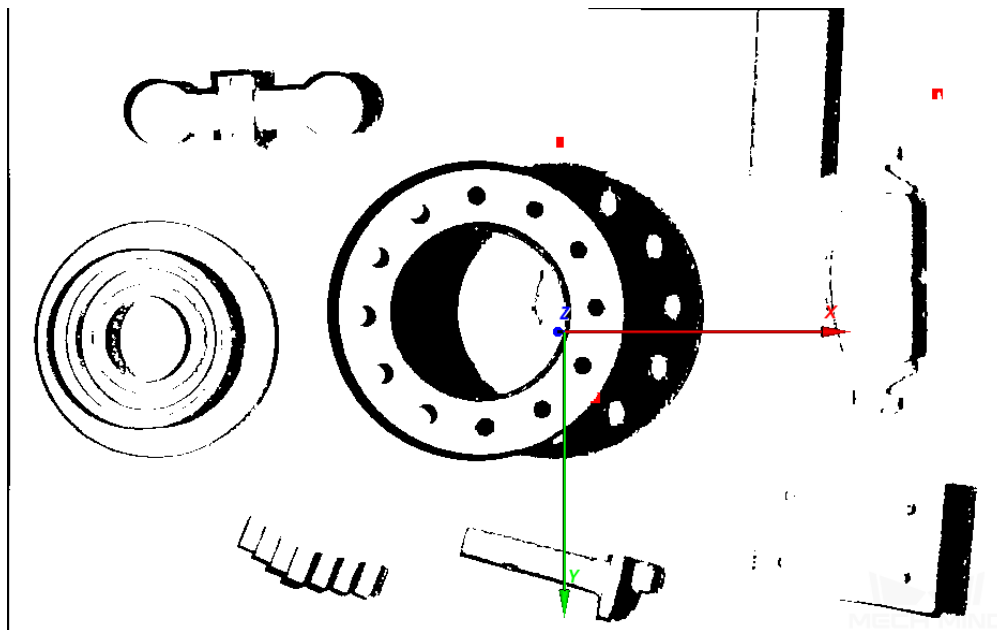
选择需检查物体，确认后选择合适的平面。

由于三点确定一个面，自定义坐标系时需选择三个点，选择的点必须在同一平面上，且不能在同一直线上。

## 8.7.3 步骤

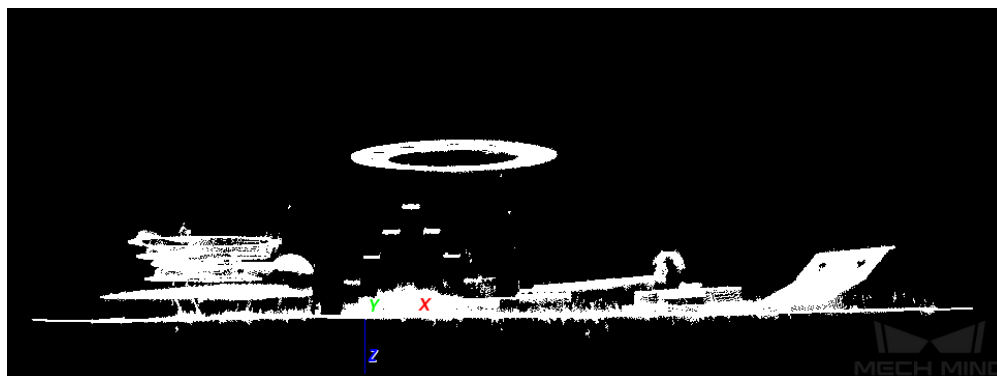
在所选平面选择三个点，即可完成坐标自定义。

1. 按住 **Shift** 键，在点云上单击鼠标左键选择点，需设置点的数量为 3。如不理想，点击 **重置** 重新选择点。已选择的点可在右侧查看 **Z** 轴坐标系的值。

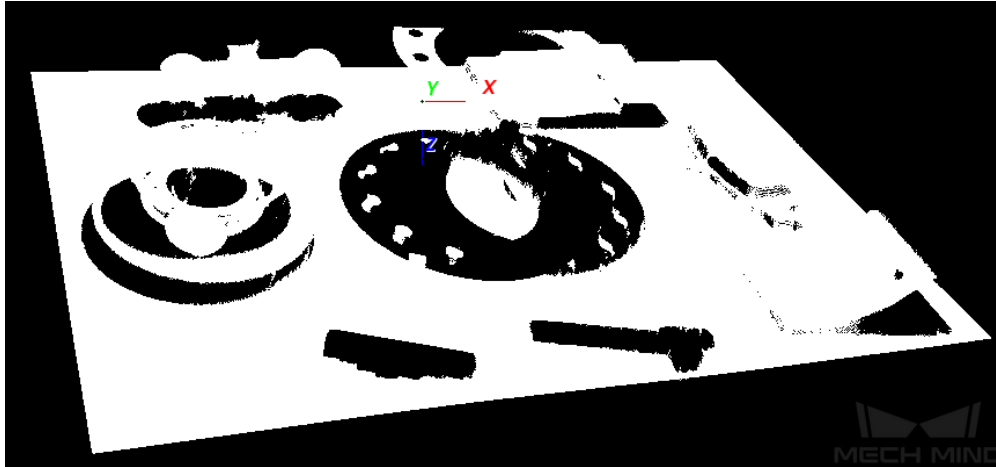


2. 三个点设置完成后，即可预览效果，此时点云上出现坐标系。旋转点云，查看坐标系状态。

- 如用户坐标系设置无误，Z轴垂直于物体平面，X、Y轴平行于物体平面。



- 其他则为设置失败，需重新调整用户坐标系。



3. 用户坐标系设置完成后，单击 确定，设置生效，自动退出当前页面。

## 8.8 相机管理器

相机管理器主要查看相机当前状态。



- 投光单元固件版本：显示当前连接相机的投光单元固件版本。
- 投射棋盘格测试图案：勾选此选项来投射棋盘格图案，用于确认投光单元的连接及对焦。
  - 如图案投射成功，则投光单元连接正常。
  - 如图案清晰不模糊，则投光单元对焦正常。
  - 如图案投射失败或图案模糊，请联系技术支持。

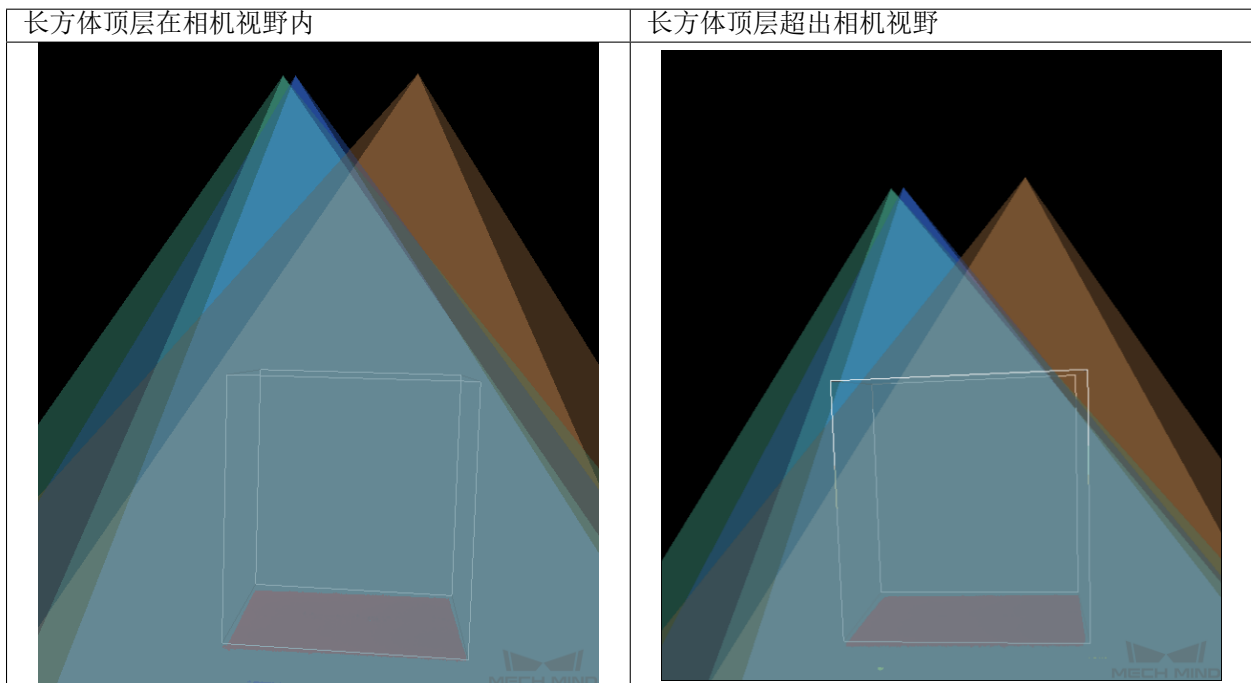
**注解：**该选项仅在连接 DLP 相机时可见。

- **相机时间：**显示相机中储存的日期和时间。  
单击 **同步至相机**，可将相机系统时间更新为工控机系统时间。
- **温度：**  
单击 **刷新参数**可查看当前相机 CPU 温度及投影设备温度。
- **分辨率：**更改相机采集的 2D 图和深度图的分辨率。完成更改后，点击 **应用设置**并重启相机。更改将在相机重启后生效。

## 8.9 满垛模拟器

满垛模拟器用于确认，满垛时垛顶层是否全部位于相机视野内，请按照满垛尺寸设置长方体尺寸。

- 如长方体顶层超出相机视野，相机位置不合理，需重新调整相机位置。
- 如长方体顶层在相机视野内，相机位置基本合理。



**提示：**显示结果仅供参考。

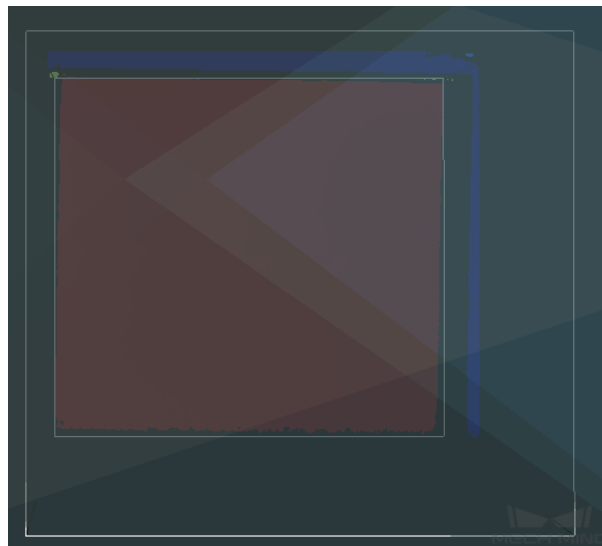
### 8.9.1 准备工作

1. 确认托盘位置合理。托盘的长边平行于相机的长边；托盘的短边平行于相机的短边。
2. 确认感兴趣区域内只保留托盘点云。

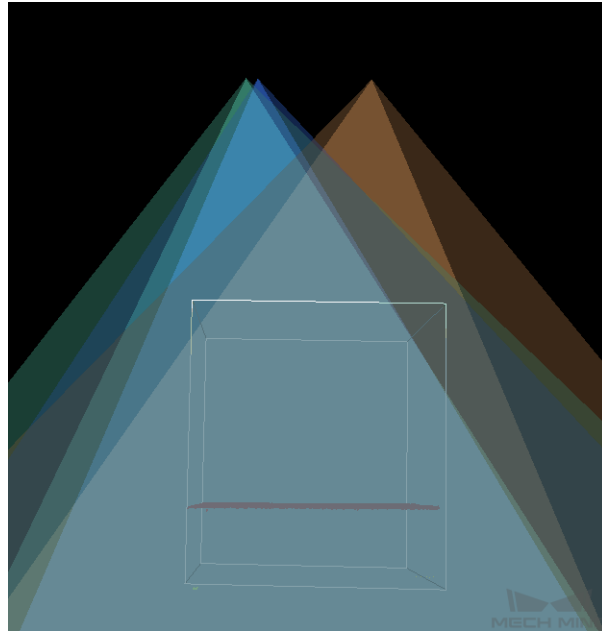
### 8.9.2 检查步骤

1. 单击 创建长方体，点云界面会自动创建长方体。在 长方体尺寸 中输入满垛尺寸，生成对应尺寸的长方体。

**提示：**创建的长方体无法旋转。如长方体（白框）长/短边与托盘（点云）长/短边不平行，请调整托盘的位置后重新采集图像；待平行后（如下图所示），再进行下一步。



2. 旋转并缩小左侧点云，调整视野，查看长方体顶层是否在相机视野内。
3. 如长方体顶层在相机视野内，则垛型合理。如长方体超出相机视野，请查看下一步。
4. 修改 长方体位置 中的位置信息，移动长方体位置。
5. 调整后，需保证长方体顶层位于相机视野内，如下图所示。此时可根据 长方体位置 调整相机位置。



---

**提示:**

- 调整位置时，建议先调整 x 与 y 方向的位置；如无法满足需求，再调整 z 方向。
  - 调整相机位置时，需注意不超出相机工作距离范围。相机工作距离可通过[视野计算器](#)查看。
-



---

## 快速了解 Mech-Eye API

---

您可使用 Mech-Eye API 连接相机，以获取 2D 图、深度图与点云。

C++ 与 C# Mech-Eye API 包含在 Mech-Eye SDK 安装包中，Python Mech-Eye API 需单独下载安装，ROS Mech-Eye API 可通过 GitHub 获取。

---

查看以下内容，了解如何 **安装 Mech-Eye SDK** 。

[安装 Mech-Eye SDK](#)

---

查看以下内容 **快速了解 Mech-Eye API** 。

[快速入门](#)

---

查看以下内容，了解如何 **安装、配置、编译、运行 Mech-Eye API** 。

[例程实操指南](#)

---

查看以下内容，了解 **Mech-Eye API 的工具**。

[API——固件升级工具](#)

---

以下内容为 Mech-Eye API 参考手册。

[Mech-Eye API 参考手册](#)

---

---

查看以下内容，了解 2.0.0 更新的方法名称与参数名称。

C++ 参数名称更新

C# 方法名称与参数名称更新

---

提示：不适用于 2.0.0 之前的 Mech-Eye API。

## 9.1 安装 Mech-Eye SDK

### 9.1.1 在 Windows 上安装 Mech-Eye SDK

如何在 Windows 系统上安装 Mech-Eye SDK，请参考软件安装指南 ([Windows 系统](#))。

### 9.1.2 在 Ubuntu 上安装 Mech-Eye SDK

Mech-Eye SDK 提供 AMD64 和 ARM64 两种类型架构的软件包，其中 AMD64 架构的软件包为 **MechEyeApi\_x.x.x\_amd64.deb**，ARM64 架构的软件包为 **MechEyeApi\_x.x.x\_arm64.deb**。请根据需要从梅卡曼德在线社区下载相应的 [Mech-Eye Api](#) 软件包。

---

提示：arch 命令可输出系统架构。

---

安装软件包：

注解：请先进入软件包所在路径，再安装软件包。

- 架构为 AMD64 时，执行如下命令：

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

- 架构为 ARM64 时，执行如下命令：

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

---

查询是否已安装 Mech-Eye SDK，执行如下命令：

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

卸载已安装的 Mech-Eye SDK，执行如下命令：

```
sudo dpkg -P MechEyeApi
```

---

本章主要介绍如何使用 Mech-Eye API 来连接相机、设置相机参数、获取并保存数据。

**注意：** Mech-Eye API 2.0.0 更新了 C++ 与 C# 的方法与参数名称。如果您已经安装 Mech-Eye SDK 2.0.0，但是仍使用的是 2.0.0 以前的接口，请更新方法与参数名称。

- C++ 参数名称更新
- C# 方法名称与参数名称更新

## 10.1 C++ 参数名称更新

Mech-Eye API 2.0.0 的 C++ 语言更新了参数名称。

2.0.0 之前	2.0.0
Scanning3DSettings::ExposureSequence	Scanning3DSettings::exposureSequence
Scanning3DSettings::Gain	Scanning3DSettings::gain
Scanning3DSettings::Scan3DROI	Scanning3DSettings::scan3DROI
Scanning3DSettings::DepthRange	Scanning3DSettings::depthRange
Scanning2DSettings::Scan2DExposureMode:: ExposureMode	Scanning2DSettings::Scan2DExposureMode:: exposureMode
Scanning2DSettings::ExposureTime	Scanning2DSettings::exposureTime
Scanning2DSettings::SharpenFactor	Scanning2DSettings::sharpenFactor
Scanning2DSettings::ExpectedGray Value	Scanning2DSettings::expectedGray Value
Scanning2DSettings::Scan2DROI	Scanning2DSettings::scan2DROI
Scanning2DSettings::ToneMappingEnable	Scanning2DSettings::toneMappingEnable
Scanning2DSettings::HDRExposureSequence	Scanning2DSettings::hdrExposureSequence
PointCloudProcessingSettings:: FringeContrastThreshold	PointCloudProcessingSettings:: fringeContrastThreshold
PointCloudProcessingSettings::FringeMinThreshold	PointCloudProcessingSettings::fringeMinThreshold
PointCloudProcessingSettings:: CloudOutlierFilterMode::OutlierFilterMode	PointCloudProcessingSettings:: CloudOutlierFilterMode::outlierFilterMode
PointCloudProcessingSettings:: CloudSmoothMode::SmoothMode	PointCloudProcessingSettings:: CloudSmoothMode::smoothMode
LaserSettings::LaserFringeCodingMode:: FringeCodingMode	LaserSettings::LaserFringeCodingMode:: fringeCodingMode
LaserSettings::FrameRangeStart	LaserSettings::frameRangeStart

## 10.2 C# 方法名称与参数名称更新

Mech-Eye API 2.0.0 的 C# 语言更新了方法名称与参数名称。

- 方法名称
- 参数名称

### 10.2.1 方法名称

2.0.0 之前	2.0.0
ColorMap.at(uint, uint)	ColorMap.At(uint, uint)
ColorMap.data()	ColorMap.Data()
ColorMap.empty()	ColorMap.Empty()
ColorMap.height()	ColorMap.Height()
ColorMap.release()	ColorMap.Release()
ColorMap.resize(uint, uint)	ColorMap.Resize(uint, uint)
ColorMap.width()	ColorMap.Width()
DepthMap.at(uint, uint)	DepthMap.At(uint, uint)
DepthMap.data()	DepthMap.Data()
DepthMap.empty()	DepthMap.Empty()
DepthMap.height()	DepthMap.Height()
DepthMap.release()	DepthMap.Release()
DepthMap.resize(uint, uint)	DepthMap.Resize(uint, uint)
DepthMap.width()	DepthMap.Width()
MechEyeDevice.addUserSet(string)	MechEyeDevice.AddUserSet(string)
MechEyeDevice.captureColorMap (mmind.apiSharp.ColorMap)	MechEyeDevice.CaptureColorMap (mmind.apiSharp.ColorMap)
MechEyeDevice.captureDepthMap (mmind.apiSharp.DepthMap)	MechEyeDevice.CaptureDepthMap (mmind.apiSharp.DepthMap)
MechEyeDevice.capturePointXYZBGRMap (mmind.apiSharp.PointXYZBGRMap)	MechEyeDevice.CapturePointXYZBGRMap (mmind.apiSharp.PointXYZBGRMap)
MechEyeDevice.capturePointXYZMap (mmind.apiSharp.PointXYZMap)	MechEyeDevice.CapturePointXYZMap (mmind.apiSharp.PointXYZMap)
MechEyeDevice.connect (mmind.apiSharp.MechEyeDeviceInfo)	MechEyeDevice.Connect (mmind.apiSharp.MechEyeDeviceInfo)
MechEyeDevice.connect(string)	MechEyeDevice.Connect(string)
MechEyeDevice.connect(string, ushort)	MechEyeDevice.Connect(string, ushort)
MechEyeDevice.deleteUserSet(string)	MechEyeDevice.DeleteUserSet(string)
MechEyeDevice.disconnect()	MechEyeDevice.Disconnect()
MechEyeDevice.enumerateMechEyeDeviceList()	MechEyeDevice.EnumerateMechEyeDeviceList()
MechEyeDevice.getAllUserSets (System.Collections.Generic.List<string>)	MechEyeDevice.GetAllUserSets (System.Collections.Generic.List<string>)
MechEyeDevice.getApiInformation()	MechEyeDevice.GetApiInformation()
MechEyeDevice.getCloudOutlierFilterMode (mmind.apiSharp.CloudOutlierFilterMode)	MechEyeDevice.GetCloudOutlierFilterMode (mmind.apiSharp.CloudOutlierFilterMode)

MechEyeDevice.getCloudSmoothMode (mmind.apiSharp.CloudSmoothMode)	MechEyeDevice.GetCloudSmoothMode (mmind.apiSharp.CloudSmoothMode)
MechEyeDevice.getCurrentUserSet(string)	MechEyeDevice.GetCurrentUserSet(string)
MechEyeDevice.getDepthRange (mmind.apiSharp.DepthRange)	MechEyeDevice.GetDepthRange (mmind.apiSharp.DepthRange)
MechEyeDevice.getDeviceInfo (mmind.apiSharp.MechEyeDeviceInfo)	MechEyeDevice.GetDeviceInfo (mmind.apiSharp.MechEyeDeviceInfo)
MechEyeDevice.getDeviceIntri (mmind.apiSharp.DeviceIntri)	MechEyeDevice.GetDeviceIntri (mmind.apiSharp.DeviceIntri)
MechEyeDevice.getDeviceResolution (mmind.apiSharp.DeviceResolution)	MechEyeDevice.GetDeviceResolution (mmind.apiSharp.DeviceResolution)
MechEyeDevice.getFringeContrastThreshold(int)	MechEyeDevice.GetFringeContrastThreshold(int)
MechEyeDevice.getFringeMinThreshold(int)	MechEyeDevice.GetFringeMinThreshold(int)
MechEyeDevice.getLaserSettings (mmind.apiSharp.LaserSettings)	MechEyeDevice.GetLaserSettings (mmind.apiSharp.LaserSettings)
MechEyeDevice.getScan2DExpectedGrayValue(int)	MechEyeDevice.GetScan2DExpectedGrayValue(int)
MechEyeDevice.getScan2DExposureMode (mmind.apiSharp.Scan2DExposureMode)	MechEyeDevice.GetScan2DExposureMode (mmind.apiSharp.Scan2DExposureMode)
MechEyeDevice.getScan2DExposureTime(double)	MechEyeDevice.GetScan2DExposureTime(double)
MechEyeDevice.getScan2DHDRExposureSequence (System.Collections.Generic.List<double>)	MechEyeDevice.GetScan2DHDRExposureSequence (System.Collections.Generic.List<double>)
MechEyeDevice.getScan2DROI (mmind.apiSharp.ROI)	MechEyeDevice.GetScan2DROI (mmind.apiSharp.ROI)
MechEyeDevice.getScan2DSharpenFactor(double)	MechEyeDevice.GetScan2DSharpenFactor(double)
MechEyeDevice.getScan2DToneMappingEnable (bool)	MechEyeDevice.GetScan2DToneMappingEnable (bool)
MechEyeDevice.getScan3DExposure (System.Collections.Generic.List<double>)	MechEyeDevice.GetScan3DExposure (System.Collections.Generic.List<double>)
MechEyeDevice.getScan3DGain(double)	MechEyeDevice.GetScan3DGain(double)
MechEyeDevice.getScan3DROI (mmind.apiSharp.ROI)	MechEyeDevice.GetScan3DROI (mmind.apiSharp.ROI)
MechEyeDevice.getUhpCaptureMode (mmind.apiSharp.UhpCaptureMode)	MechEyeDevice.GetUhpCaptureMode (mmind.apiSharp.UhpCaptureMode)
MechEyeDevice.getUhpFringeCodingMode (mmind.apiSharp.UhpFringeCodingMode)	MechEyeDevice.GetUhpFringeCodingMode (mmind.apiSharp.UhpFringeCodingMode)
MechEyeDevice.getUhpSettings (mmind.apiSharp.UhpSettings)	MechEyeDevice.GetUhpSettings (mmind.apiSharp.UhpSettings)
MechEyeDevice.saveAllSettingsToUserSets()	MechEyeDevice.SaveAllSettingsToUserSets()
MechEyeDevice.setCloudOutlierFilterMode (mmind.apiSharp.CloudOutlierFilterMode)	MechEyeDevice.SetCloudOutlierFilterMode (mmind.apiSharp.CloudOutlierFilterMode)
MechEyeDevice.setCloudSmoothMode (mmind.apiSharp.CloudSmoothMode)	MechEyeDevice.SetCloudSmoothMode (mmind.apiSharp.CloudSmoothMode)
MechEyeDevice.setCurrentUserSet(string)	MechEyeDevice.SetCurrentUserSet(string)
MechEyeDevice.setDepthRange (mmind.apiSharp.DepthRange)	MechEyeDevice.SetDepthRange (mmind.apiSharp.DepthRange)
MechEyeDevice.setFringeContrastThreshold(int)	MechEyeDevice.SetFringeContrastThreshold(int)
MechEyeDevice.setFringeMinThreshold(int)	MechEyeDevice.SetFringeMinThreshold(int)

MechEyeDevice.setLaserSettings (mmind.apiSharp.LaserSettings)	MechEyeDevice.SetLaserSettings (mmind.apiSharp.LaserSettings)
MechEyeDevice.setScan2DExpectedGrayValue (int)	MechEyeDevice.SetScan2DExpectedGrayValue (int)
MechEyeDevice.setScan2DExposureMode (mmind.apiSharp.Scan2DExposureMode)	MechEyeDevice.SetScan2DExposureMode (mmind.apiSharp.Scan2DExposureMode)
MechEyeDevice.setScan2DExposureTime(double)	MechEyeDevice.SetScan2DExposureTime(double)
MechEyeDevice.setScan2DHDRExposureSequence (System.Collections.Generic.List<double>)	MechEyeDevice.SetScan2DHDRExposureSequence (System.Collections.Generic.List<double>)
MechEyeDevice.setScan2DROI (mmind.apiSharp.ROI)	MechEyeDevice.SetScan2DROI (mmind.apiSharp.ROI)
MechEyeDevice.setScan2DSharpenFactor(double)	MechEyeDevice.SetScan2DSharpenFactor(double)
MechEyeDevice.setScan2DToneMappingEnable (bool)	MechEyeDevice.SetScan2DToneMappingEnable (bool)
MechEyeDevice.setScan3DExposure (System.Collections.Generic.List<double>)	MechEyeDevice.SetScan3DExposure (System.Collections.Generic.List<double>)
MechEyeDevice.setScan3DGain(double)	MechEyeDevice.SetScan3DGain(double)
MechEyeDevice.setScan3DROI (mmind.apiSharp.ROI)	MechEyeDevice.SetScan3DROI (mmind.apiSharp.ROI)
MechEyeDevice.setUhpCaptureMode (mmind.apiSharp.UhpCaptureMode)	MechEyeDevice.SetUhpCaptureMode (mmind.apiSharp.UhpCaptureMode)
MechEyeDevice.setUhpFringeCodingMode (mmind.apiSharp.UhpFringeCodingMode)	MechEyeDevice.SetUhpFringeCodingMode (mmind.apiSharp.UhpFringeCodingMode)
MechEyeDevice.setUhpSettings (mmind.apiSharp.UhpSettings)	MechEyeDevice.SetUhpSettings (mmind.apiSharp.UhpSettings)
PointXYZBGRMap.at(uint, uint)	PointXYZBGRMap.At(uint, uint)
PointXYZBGRMap.data()	PointXYZBGRMap.Data()
PointXYZBGRMap.empty()	PointXYZBGRMap.Empty()
PointXYZBGRMap.height()	PointXYZBGRMap.Height()
PointXYZBGRMap.release()	PointXYZBGRMap.Release()
PointXYZBGRMap.resize(uint, uint)	PointXYZBGRMap.Resize(uint, uint)
PointXYZBGRMap.width()	PointXYZBGRMap.Width()
PointXYZMap.at(uint, uint)	PointXYZMap.At(uint, uint)
PointXYZMap.data()	PointXYZMap.Data()
PointXYZMap.empty()	PointXYZMap.Empty()
PointXYZMap.height()	PointXYZMap.Height()
PointXYZMap.release()	PointXYZMap.Release()
PointXYZMap.resize(uint, uint)	PointXYZMap.Resize(uint, uint)
PointXYZMap.width()	PointXYZMap.Width()

### 10.2.2 参数名称

2.0.0 之前	2.0.0
LaserSettings.FramePartitionCount	LaserSettings.framePartitionCount
LaserSettings.FrameRangeEnd	LaserSettings.frameRangeEnd
LaserSettings.FrameRangeStart	LaserSettings.frameRangeStart
LaserSettings.FringeCodingMode	LaserSettings.fringeCodingMode
LaserSettings.PowerLevel	LaserSettings.powerLevel
PointCloudProcessingSettings.CloudSmoothMode	PointCloudProcessingSettings.cloudSmoothMode
PointCloudProcessingSettings.FringeContrastThreshold	PointCloudProcessingSettings.fringeContrastThreshold
PointCloudProcessingSettings.FringeMinThreshold	PointCloudProcessingSettings.fringeMinThreshold
PointCloudProcessingSettings.OutlierFilterMode	PointCloudProcessingSettings.outlierFilterMode
Scanning2DSettings.ExpectedGrayValue	Scanning2DSettings.expectedGrayValue
Scanning2DSettings.ExposureMode	Scanning2DSettings.exposureMode
Scanning2DSettings.ExposureTime	Scanning2DSettings.exposureTime
Scanning2DSettings.Scan2DROI	Scanning2DSettings.scan2DROI
Scanning2DSettings.SharpenFactor	Scanning2DSettings.sharpenFactor
Scanning2DSettings.ToneMappingEnable	Scanning2DSettings.toneMappingEnable
Scanning3DSettings.DepthRange	Scanning3DSettings.depthRange
Scanning3DSettings.ExposureSequence	Scanning3DSettings.exposureSequence
Scanning3DSettings.Gain	Scanning3DSettings.gain
Scanning3DSettings.Scan3DROI	Scanning3DSettings.scan3DROI
UhpSettings.CaptureMode	UhpSettings.captureMode
UhpSettings.FringeCodingMode	UhpSettings.fringeCodingMode

## 10.3 准备工作

对于 C++、C#、Python 语言，需进行 *Mech-Eye SDK* 安装。

对于 Python 语言，还需进行 *Python* 相关配置。

## 10.4 搜索相机

准备工作完成后，可通过以下指令来搜索相机（使用枚举方法获得可连接的相机列表）。

搜索相机（C++）

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

搜索相机（C#）

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

搜索相机（Python）

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

## 10.5 连接相机

搜索相机后，可通过以下指令连接相机，以连接相机列表中的第一个相机为例。

连接相机 (C++)

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

连接相机 (C#)

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

连接相机 (Python)

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

## 10.6 设置相机参数

连接相机后，可通过以下指令设置相机参数。

### 10.6.1 设置 2D 图参数

在使用相机采集 2D 图之前，需要设置相机的 **2D** 参数，包括 曝光模式、曝光时间等。

设置 2D 图参数 (C++)

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

设置 2D 图参数 (C#)

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

设置 2D 图参数 (Python)

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



## 10.6.2 设置深度图与点云参数

在使用相机采集用于计算深度数据的图像之前，需要设置影响深度图和点云质量的参数，包括曝光时间、深度范围、感兴趣区域、点云后处理等。

设置深度图与点云参数 (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);
```

设置深度图与点云参数 (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);
```

设置深度图与点云参数 (Python)

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

## 10.7 获取数据

完成相机参数设置后，可以触发相机采集图像并返回 2D 图及点云数据。

### 10.7.1 获取 2D 图

获取 2D 图 (C++)

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

获取 2D 图 (C#)

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

获取 2D 图 (Python)

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

## 10.7.2 获取点云数据

获取点云数据 (C++)

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

获取点云数据 (C#)

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

获取点云数据 (Python)

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

## 10.8 保存数据

### 10.8.1 保存 2D 图

将获取的 2D 图数据转化为 OpenCV 数据类型，并保存为 png 格式的图片文件。

保存 2D 图 (C++)

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

保存 2D 图 (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);
```

保存 2D 图 (Python)

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

## 10.8.2 保存点云

将获取的点云数据保存为 ply 格式的点云。

保存点云 (C++)

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

保存点云 (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);
```

保存点云 (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)
```

选择对应的编程语言来开始对 Mech-Eye 相机进行编程。

### 11.1 C++

本章介绍如何在 Windows 系统中使用 CMake 配置 Mech-Eye API 的 C++ 例程，并使用 Visual Studio 构建例程。

- 例程简介
- 使用前提
- 构建及运行例程

#### 11.1.1 例程简介

例程分为 7 类：**Basic**、**Advanced**、**Util**、**Laser**、**UHP**、**Calibration** 和 **Halcon**。

**Basic** 例程：与连接和基础采集相关。

**Advanced** 例程：与高级采集技巧相关。

**Util** 例程：与获取相机信息及设置参数相关。

**Laser** 例程：仅适用于 Laser, LSR 和 DEEP 系列相机。

**UHP** 例程：仅适用于 UHP 系列相机。

**Calibration** 例程：通过 Mech-Eye API 进行手眼标定。

**Halcon** 例程：通过 Mech-Eye API 获取 HALCON 可读取的点云。

- **Basic**
  - **ConnectToCamera**: 连接相机。
  - **ConnectAndCaptureImage**: 连接相机并获取 2D 图、深度图及点云数据。
  - **CaptureColorMap**: 从相机获取并保存 2D 图。
  - **CaptureDepthMap**: 从相机获取并保存深度图。
  - **CapturePointCloud**: 使用单个曝光时间采集图像，生成并保存白色点云和彩色点云。
  - **CaptureHDRPointCloud**: 使用多个曝光时间采集图像，生成并保存白色点云和彩色点云。
  - **CapturePointCloudROI**: 获取并保存感兴趣区域内的白色点云和彩色点云。
  - **CapturePointCloudFromTextureMask**: 从覆盖掩膜的 2D 图和深度图生成并保存白色点云和彩色点云。
- **Advanced**
  - **CaptureCloudFromDepth**: 从 2D 图与深度图生成并保存点云。
  - **CaptureSequentiallyMultiCamera**: 使用多台相机按序获取并保存 2D 图、深度图及点云。
  - **CaptureSimultaneouslyMultiCamera**: 使用多台相机同时获取并保存 2D 图、深度图及点云。
  - **CaptureTimedAndPeriodically**: 在设定时间内，定时获取并保存 2D 图、深度图和点云。
- **Util**
  - **GetCameraIntri**: 获取并打印相机内参。
  - **PrintDeviceInfo**: 获取并打印相机型号、序列号、固件版本、温度等信息。
  - **SetDepthRange**: 设置相机深度范围。
  - **SetParameters**: 设置相机参数。
  - **SetUserSets**: 执行参数组相关功能，如获取参数组名称，修改参数组名称，保存参数组参数信息等。通过参数组可以保存一组参数值并快速应用。
- **Laser**
  - **SetLaserFramePartitionCount**: 将投影仪视野分为多个分区，结构光一次投射到一个分区。整个视野的输出由所有分区的图像融合而成。
  - **SetLaserFrameRange**: 设置结构光投射范围，整个视野范围从 0 到 100。
  - **SetLaserFringeCodingMode**: 设置结构光样式的编码模式。
  - **SetLaserPowerLevel**: 设置激光器的输出功率（最大输出功率的百分比），影响激光强度。
- **UHP**
  - **SetUHPCaptureMode**: 设置采集模式（分为：2D 相机 1，2D 相机 2 及同时使用两个 2D 相机并融合输出）。
  - **SetUHPFringeCodingMode**: 设置结构光样式的编码模式。
- **Calibration**
  - **HandEyeCalibration**: 用于进行手眼标定。
- **Halcon**

- CaptureHalconPointCloud：从相机获取点云数据，通过 HALCON C++ 接口转换并保存点云。

### 11.1.2 使用前提

使用 Mech-Eye API 的 C++ 例程，需先满足以下使用前提：

- 正确连接相机和工控机。
- 安装必需软件。
- 安装例程依赖的可选软件，并添加相关的环境变量。

必需软件和可选软件及其安装注意事项，详见以下小节。

- 安装必需软件
- 安装可选软件
- 添加环境变量

### 安装必需软件

使用 Mech-Eye API 的 C++ 例程，必须安装 Mech-Eye SDK、CMake 及 Visual Studio。

安装最新版本的 **Mech-Eye SDK** 请根据 [Mech-Eye SDK 安装指南](#) 安装或升级 Mech-Eye SDK。

安装 **CMake (3.2 或以上版本)**

1. 下载 **CMake**：下载 **Windows x64 Installer** 右侧的安装包。

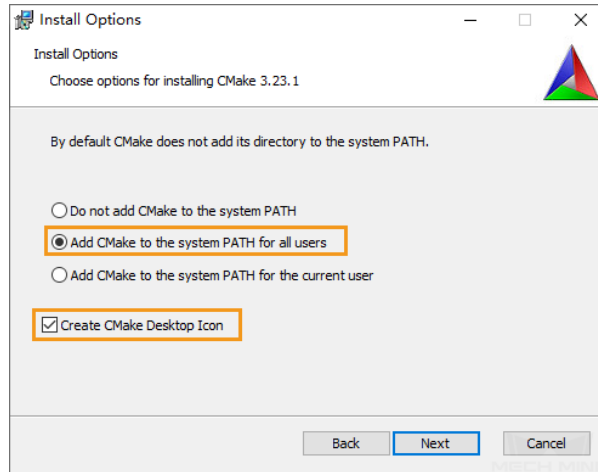
Source distributions:

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

Binary distributions:

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

2. 安装时，请选择以下两个选项，以将 CMake 添加至环境变量，并创建 CMake 的桌面快捷方式。
  - **Add CMake to the system PATH for all users**
  - **Create Cmake Desktop Icon**



### 安装 Visual Studio（2017 或以上版本）

1. 下载 Visual Studio 安装包。
2. 安装时，勾选 桌面应用和移动应用分类中的以下两个工作负荷，再点击 安装。
  - 使用 C++ 的桌面开发
  - 通用 Windows 平台开发

---

**提示：** 安装完成后，请重启工控机或 PC。如仍需安装可选软件，可在添加环境变量后再重启。

---

### 安装可选软件

部分例程中的功能依赖以下第三方软件库。如需使用下表中列出的例程，则必须安装其依赖的软件。

- OpenCV: 3.4.5 或以上版本
- PCL: 1.8.1 或以上版本
- HALCON: 20.11 或以上版本

---

**注解：** 低于 20.11 版本的 HAICON 未经完全验证。

---

**注意：** 如未安装可选软件，则配置例程前必须执行禁用不需要的例程（可选）中的操作。

依赖第三方软件库的例程详见下表。

例程	OpenCV	PCL	HALCON
CaptureColorMap	√		
CaptureDepthMap	√		
CapturePointCloud		√	
CaptureHDRPointCloud		√	
CapturePointCloudROI		√	
CapturePointCloudFromTextureMask		√	
CaptureCloudFromDepth		√	
CaptureTimedAndPeriodically	√	√	
CaptureSimultaneouslyMultiCamera	√	√	
CaptureSequentiallyMultiCamera	√	√	
CaptureHalconPointCloud			√

### 安装 OpenCV

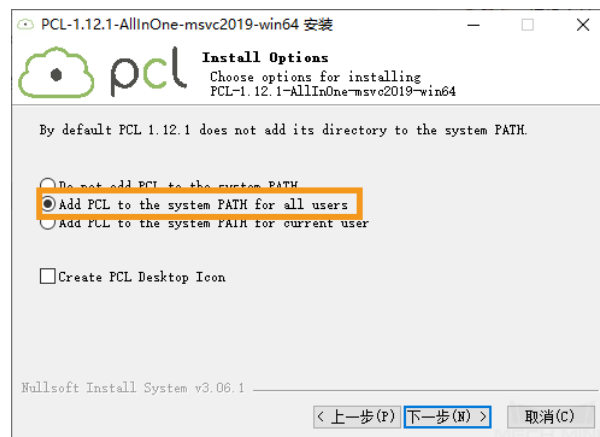
1. 下载并安装 OpenCV。请记录 OpenCV 的安装路径。
2. 将 OpenCV 添加至环境变量。详见下一节添加环境变量。

### 安装 PCL

1. 参照下表，根据安装的 Visual Studio 版本确定需安装的 PCL 版本。

Visual Studio 版本	支持的 PCL 版本
2017	1.8.1~1.9.1
2019	1.8.1~1.12.1
2022	1.8.1 及以上

2. 打开 PCL 下载页面，单击对应版本下的 **Assets**，并下载 exe 格式的安装包。
3. 运行安装包以安装 PCL。安装时，请选择 **Add PCL to the system PATH for all users**，以将 PCL 添加至环境变量。



提示：安装过程中，可能弹出如下报错：

**Warning! PATH too long installer unable to modify PATH!**



此情况下，需安装完成后手动将 PCL 添加至系统变量。详见下一节添加环境变量。

4. 将 PCL 依赖的 OpenNI2 添加至环境变量。详见下一节添加环境变量。

安装 HALCON 安装 HALCON 20.11 或以上版本。需确保 HALCON 的证书在有效期内。

### 添加环境变量

如需使用依赖 OpenCV 或 PCL 的例程，请根据以下步骤添加相关的环境变量。

1. 右键单击桌面上的 **此电脑**，选择 **属性**。
2. 选择 **高级系统设置**，再在弹出的 **系统属性**窗口单击 **环境变量**，进入 **环境变量**界面。
3. 双击 **Path** 变量进入 **编辑环境变量**页面。单击右上角 **新建**，依次添加以下路径。添加完成后，单击右下角 **确定**。
  - PCL 相关: *C:\Program Files\OpenNI2\Tools*
  - OpenCV 相关: *xxx\OpenCV\build\x64\vc14\bin*
  - OpenCV 相关: *xxx\OpenCV\build\x64\vc14\lib*

如安装 PCL 时弹出报错，则需添加以下路径：

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

**提示：**添加环境变量后，请重启工控机或 PC，否则可能导致添加的环境变量不生效。

### 11.1.3 构建及运行例程

您可以一次性构建全部例程，也可以仅构建单独一个例程。

- 禁用不需要的例程（可选）
- 使用 *CMake* 配置例程
- 使用 *Visual Studio* 构建例程
- 运行例程

## 禁用不需要的例程（可选）

**注意：**未安装可选软件的情况下一次性构建全部例程时，必须进行该节的操作。否则使用 CMake 配置例程时将报错。

如不需要使用依赖 OpenCV、PCL 或 HALCON 的例程，在一次性构建全部例程前，需对 CMakeLists 文件进行编辑，禁用这些例程。

1. 打开 `xxx\Mech-Eye SDK-x.x.x\API\samples` 路径下的 **CMakeLists.txt**。
2. 编辑 **option** 语句：找到以 **option** 开头的行，将行末的 **ON** 改为 **OFF**，可禁用依赖该行提及软件的例程。  
 示例：不需要构建依赖 HALCON 的例程，则将以 **option(USE\_HALCON** 开头的行末尾的 **ON** 改为 **OFF**。
3. 保存修改后，关闭 **CMakeLists.txt**。

## 使用 CMake 配置例程

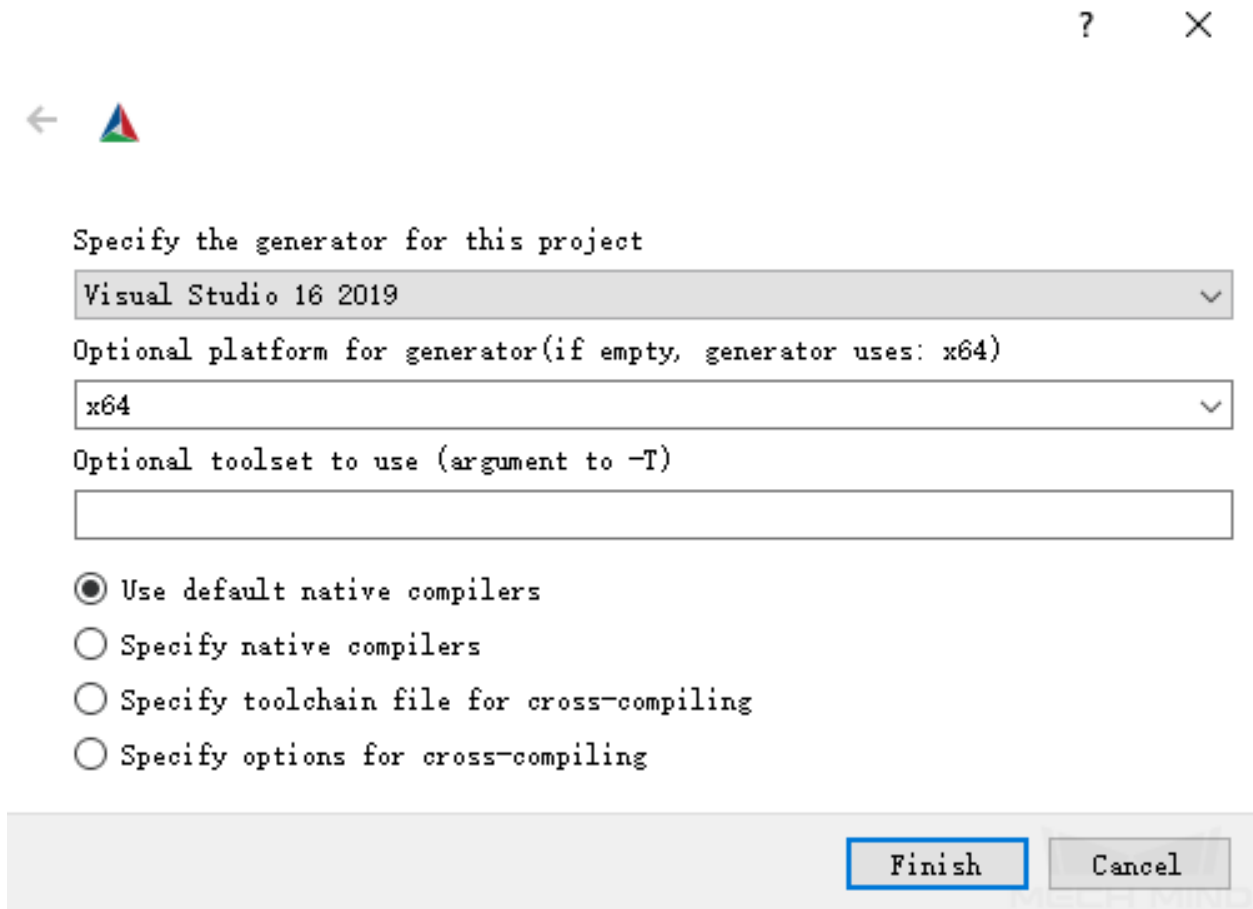
1. 右键单击 CMake 并选择 **以管理员身份运行**。
2. 输入源码路径与构建目录路径。
  - 如需一次性构建所有例程，请输入如下路径。

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

- 如需单独构建某一例程，请输入如下路径。将路径中的 **Category** 替换为例程所在类别的名称，将 **SampleName** 替换为例程名称。

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

3. 单击 *Configure*，进入配置页面。
4. 选择 Visual Studio 的版本，并选择平台为 **x64**。然后，单击 *Finish*。配置成功后，日志最末行将显示 **Configuring done**。



5. 单击 *Generate* 生成解决方案。生成成功后，日志最末行将显示 **Generating done**。然后，单击 *Open Project*，使用 Visual Studio 打开解决方案。

### 使用 Visual Studio 构建例程

1. 在 Visual Studio 工具栏中，将解决方案配置从 **Debug** 改为 **Release** 模式。
2. 在 **解决方案资源管理器**窗口中右键单击顶端的 **解决方案 “xxx”**，并选择 **生成解决方案**。每个例程会生成对应的 exe 格式可执行文件，保存在 **Release** 文件夹中，位于在 CMake 中输入的 **Where to build the binaries** 目录下。

## 运行例程

您可以在 Visual Studio 中直接运行例程，也可以双击运行例程的可执行文件。

### 在 Visual Studio 中运行例程

1. 在 **解决方案资源管理器** 窗口中右键单击想要运行的例程，并选择 **设为启动项目**。
2. 单击工具栏中的 **本地 Windows 调试器** 即可在 Visual Studio 中运行例程。

例程运行过程中，请根据提示输入待连接相机的编号 (device index)，并按回车键连接相机。

3. 如例程保存了图像或点云，将在 **build** 文件夹中得到保存的文件（即在 CMake 中输入的 **Where to build the binaries** 目录）。

### 运行例程可执行文件

1. 进入在 CMake 中输入的 **Where to build the binaries** 目录，打开其中的 **Release** 文件夹。
2. 运行与例程同名的 exe 文件。

例程运行过程中，请根据提示输入待连接相机的编号 (device index)，并按回车键连接相机。

3. 如例程保存了图像或点云，将在 **Release** 文件夹中得到保存的文件。

## 11.2 C++ (Ubuntu)

本章介绍如何在 Ubuntu 系统中使用 CMake 配置 Mech-Eye API 的 C++ 例程，并使用 make 指令构建例程。

### 11.2.1 例程简介

例程分为 7 类：**Basic**、**Advanced**、**Util**、**Laser**、**UHP**、**Calibration** 和 **Halcon**。

**Basic** 例程：与连接和基础采集相关。

**Advanced** 例程：与高级采集技巧相关。

**Util** 例程：与获取相机信息及设置参数相关。

**Laser** 例程：仅适用于 Laser, LSR 和 DEEP 系列相机。

**UHP** 例程：仅适用于 UHP 系列相机。

**Calibration** 例程：通过 Mech-Eye API 进行手眼标定。

**Halcon** 例程：通过 Mech-Eye API 获取 HALCON 可读取的点云。

#### • Basic

- **ConnectToCamera**: 连接相机。
- **ConnectAndCaptureImage**: 连接相机并获取 2D 图、深度图及点云数据。
- **CaptureColorMap**: 从相机获取并保存 2D 图。
- **CaptureDepthMap**: 从相机获取并保存深度图。
- **CapturePointCloud**: 使用单个曝光时间采集图像，生成并保存白色点云和彩色点云。

- **CaptureHDRPointCloud**: 使用多个曝光时间采集图像，生成并保存白色点云和彩色点云。
- **CapturePointCloudROI**: 获取并保存感兴趣区域内的白色点云和彩色点云。
- **CapturePointCloudFromTextureMask**: 从覆盖掩膜的 2D 图和深度图生成并保存白色点云和彩色点云。

- **Advanced**

- **CaptureCloudFromDepth**: 从 2D 图与深度图生成并保存点云。
- **CaptureSequentiallyMultiCamera**: 使用多台相机按序获取并保存 2D 图、深度图及点云。
- **CaptureSimultaneouslyMultiCamera**: 使用多台相机同时获取并保存 2D 图、深度图及点云。
- **CaptureTimedAndPeriodically**: 在设定时间内，定时获取并保存 2D 图、深度图和点云。

- **Util**

- **GetCameraIntri**: 获取并打印相机内参。
- **PrintDeviceInfo**: 获取并打印相机型号、序列号、固件版本、温度等信息。
- **SetDepthRange**: 设置相机深度范围。
- **SetParameters**: 设置相机参数。
- **SetUserSets**: 执行参数组相关功能，如获取参数组名称，修改参数组名称，保存参数组参数信息等。通过参数组可以保存一组参数值并快速应用。

- **Laser**

- **SetLaserFramePartitionCount**: 将投影仪视野分为多个分区，结构光一次投射到一个分区。整个视野的输出由所有分区的图像融合而成。
- **SetLaserFrameRange**: 设置结构光投射范围，整个视野范围从 0 到 100。
- **SetLaserFringeCodingMode**: 设置结构光样式的编码模式。
- **SetLaserPowerLevel**: 设置激光器的输出功率（最大输出功率的百分比），影响激光强度。

- **UHP**

- **SetUHPCaptureMode**: 设置采集模式（分为：2D 相机 1，2D 相机 2 及同时使用两个 2D 相机并融合输出）。
- **SetUHPFringeCodingMode**: 设置结构光样式的编码模式。

- **Calibration**

- **HandEyeCalibration**: 用于进行手眼标定。

- **Halcon**

- **CaptureHalconPointCloud**: 从相机获取点云数据，通过 HALCON C++ 接口转换并保存点云。

## 11.2.2 使用前提

使用 Mech-Eye API 的 C++ 例程，需先满足以下使用前提：

- 正确连接相机和工控机。
- Ubuntu 系统版本为 18 或以上。
- 安装必需软件。
- 安装例程依赖的可选软件。

**注意：**若使用 Ubuntu 系统的虚拟机，请保证磁盘可用空间大于 20G，否则软件安装可能失败。

必需软件和可选软件及其安装注意事项，详见以下小节。

### 安装必需软件

使用 Mech-Eye API 的 C++ 例程，必须更新软件源列表，并安装依赖库及 Mech-Eye SDK。

1. 更新软件源列表。

```
sudo apt-get update
```

2. 安装依赖库。

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

3. 安装 Mech-Eye SDK，详见在 [Ubuntu](#) 上安装 [Mech-Eye SDK](#)。

安装完成后，例程所在文件夹的路径为 `/opt/mech-mind/mech-eye-sdk/samples/`。

### 安装可选软件

部分例程中的功能依赖以下第三方软件库。如需使用下表中列出的例程，则必须安装其依赖的软件。

- OpenCV：最新版本
- PCL：最新版本
- HALCON：20.11 或以上版本

**注解：**低于 20.11 版本的 HAICON 未经完全验证。

**注意：**如未安装可选软件，则配置例程前必须执行禁用不需要的例程（可选）中的操作。

依赖第三方软件库的例程详见下表。

例程	OpenCV	PCL	HALCON
CaptureColorMap	√		
CaptureDepthMap	√		
CapturePointCloud		√	
CaptureHDRPointCloud		√	
CapturePointCloudROI		√	
CapturePointCloudFromTextureMask		√	
CaptureCloudFromDepth		√	
CaptureTimedAndPeriodically	√	√	
CaptureSimultaneouslyMultiCamera	√	√	
CaptureSequentiallyMultiCamera	√	√	
CaptureHalconPointCloud		√	√

- 安装 OpenCV:

```
sudo apt update && sudo apt install -y unzip
wget -O opencv.zip https://github.com/opencv/opencv/archive/4.x.zip
unzip opencv.zip
mkdir build && cd build
cmake ../opencv-4.x
cmake --build .
sudo make install
```

- 安装 PCL:

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

**注解:** 不同版本的 Ubuntu, 运行该指令安装的 PCL 版本不同。Ubuntu 18.04 将安装 PCL 1.8.1, Ubuntu 20.04 将安装 PCL 1.10.0.

- 安装 HALCON 并添加环境变量:

1. 下载 HALCON 安装包后使用以下指令安装 HALCON (以 20.11 版本为例):

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

2. 添加 HALCON 的环境变量: 在编辑器 (如 vi) 中打开 `/etc/profile`, 并将以下内容复制到该文件末尾。将 `/opt/halcon` 替换为 HALCON 的实际安装路径。

```
HALCONARCH=x64-linux; export HALCONARCH
HALCONROOT="/opt/halcon"; export HALCONROOT
HALCONEXAMPLES=${HALCONROOT}/examples; export HALCONEXAMPLES
HALCONIMAGES=${HALCONROOT}/examples/images; export HALCONIMAGES
PATH=${HALCONROOT}/bin/${HALCONARCH}:${PATH}; export PATH
if [ ${LD_LIBRARY_PATH} ] ; then
    LD_LIBRARY_PATH=${HALCONROOT}/lib/${HALCONARCH}:${LD_LIBRARY_PATH}; export LD_LIBRARY_PATH
else
    LD_LIBRARY_PATH=${HALCONROOT}/lib/${HALCONARCH}; export LD_LIBRARY_PATH
fi
```

**注解:**

- 环境变量在重新登录后生效。或者请在构建例程前先执行 `source /etc/profile/` 指令。
- 更多关于 HALCON 安装的内容，请参阅 HALCON 的安装指南。

### 11.2.3 构建及运行例程

您可以一次性构建全部例程，也可以仅构建单独一个例程。

- 禁用不需要的例程（可选）
- 构建例程
- 运行例程

#### 禁用不需要的例程（可选）

**注意:** 未安装可选软件的情况下一次性构建全部例程时，必须进行该节的操作。否则使用 CMake 配置例程时将报错。

如不需要使用依赖 OpenCV、PCL 或 HALCON 的例程，在一次性构建全部例程前，需对 CMakeLists 文件进行编辑，禁用这些例程。

1. 在编辑器（如 vi）中打开 `/opt/mech-mind/mech-eye-sdk/samples/CMakeLists.txt`。
2. 编辑 **option** 语句：找到以 **option** 开头的行，将行末的 **ON** 改为 **OFF**，可禁用依赖该行提及软件的例程。  
示例：不需要构建依赖 HALCON 的例程，则将以 **option(USE\_HALCON** 开头的行末尾的 **ON** 改为 **OFF**。
3. 在编辑器中保存对 CMakeLists 文件的修改。

#### 构建例程

1. 进入例程所在路径：
  - 如一次性构建全部例程，请使用以下指令。

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

- 如单独构建某个例程，请使用以下指令。将路径中的 **Category** 替换为例程所在类别的名称，将 **SampleName** 替换为例程名称。

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

2. 在例程所在路径下新建 **build** 文件夹，用于保存编译完成的例程等。



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

### 3. 配置并构建例程:

```
cmake ..  
make
```

**小技巧:** 如修改了例程文件夹中的源码文件, 需删除 **build** 文件夹后, 重新执行第 2 和 3 步, 重新构建例程。

## 运行例程

### 1. 进入例程路径下的 **build** 文件夹 (如构建例程后直接运行例程, 可跳过此步):

- 如一次性构建了全部例程, 请使用以下指令。

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

- 如单独构建了某个例程, 请使用以下指令。将路径中的 **Category** 替换为例程所在类别的名称, 将 **SampleName** 替换为例程名称。

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

### 2. 运行例程: 将 **SampleName** 替换为例程名称。例程运行过程中, 请根据提示输入待连接相机的编号 (device index), 并按回车键连接相机。

```
./SampleName
```

### 3. 如例程保存了图像或点云, 将在 **build** 文件夹中得到保存的文件。

## 11.3 C#

本章将介绍如何在 Windows 上运行 C# 例程。

### 11.3.1 例程简介

例程分为 5 类: **Basic**、**Advanced**、**Util**、**Laser** 和 **UHP**。

**Basic** 例程: 与连接和基础采集相关。

**Advanced** 例程: 与高级采集技巧相关。

**Util** 例程: 与获取相机信息及设置参数相关。

**Laser** 例程: 仅适用于 Laser, LSR 和 DEEP 系列相机。

**UHP** 例程: 仅适用于 UHP 系列相机。

- **Basic**

- **ConnectToCamera**: 连接相机。
- **ConnectAndCaptureImage**: 连接相机并获取 2D 图、深度图及点云数据。
- **CaptureColorMap**: 从相机获取并保存 2D 图。
- **CaptureDepthMap**: 从相机获取并保存深度图。
- **CapturePointCloud**: 使用单个曝光时间采集图像，生成并保存白色点云和彩色点云。
- **CaptureHDRPointCloud**: 使用多个曝光时间采集图像，生成并保存白色点云和彩色点云。
- **CapturePointCloudROI**: 获取并保存感兴趣区域内的白色点云和彩色点云。
- **CapturePointCloudFromTextureMask**: 从覆盖掩膜的 2D 图和深度图生成并保存白色点云和彩色点云。

• **Advanced**

- **CaptureCloudFromDepth**: 从 2D 图与深度图生成并保存点云。
- **CaptureSequentiallyMultiCamera**: 使用多台相机按序获取并保存 2D 图、深度图及点云。
- **CaptureSimultaneouslyMultiCamera**: 使用多台相机同时获取并保存 2D 图、深度图及点云。
- **CaptureTimedAndPeriodically**: 在设定时间内，定时获取并保存 2D 图、深度图和点云。

• **Util**

- **GetCameraIntri**: 获取并打印相机内参。
- **PrintDeviceInfo**: 获取并打印相机型号、序列号、固件版本、温度等信息。
- **SetDepthRange**: 设置相机深度范围。
- **SetParameters**: 设置相机参数。
- **SetUserSets**: 执行参数组相关功能，如获取参数组名称，修改参数组名称，保存参数组参数信息等。通过参数组可以保存一组参数值并快速应用。

• **Laser**

- **SetLaserFramePartitionCount**: 将投影仪视野分为多个分区，结构光一次投射到一个分区。整个视野的输出由所有分区的图像融合而成。
- **SetLaserFrameRange**: 设置结构光投射范围，整个视野范围从 0 到 100。
- **SetLaserFringeCodingMode**: 设置结构光样式的编码模式。
- **SetLaserPowerLevel**: 设置激光器的输出功率（最大输出功率的百分比），影响激光强度。

• **UHP**

- **SetUHPCaptureMode**: 设置采集模式（分为：2D 相机 1，2D 相机 2 及同时使用两个 2D 相机并融合输出）。
- **SetUHPFringeCodingMode**: 设置结构光样式的编码模式。

### 11.3.2 准备工作

1. 下载 Visual Studio 2019 。

建议使用 Visual Studio 2019 或 Visual Studio 2015。

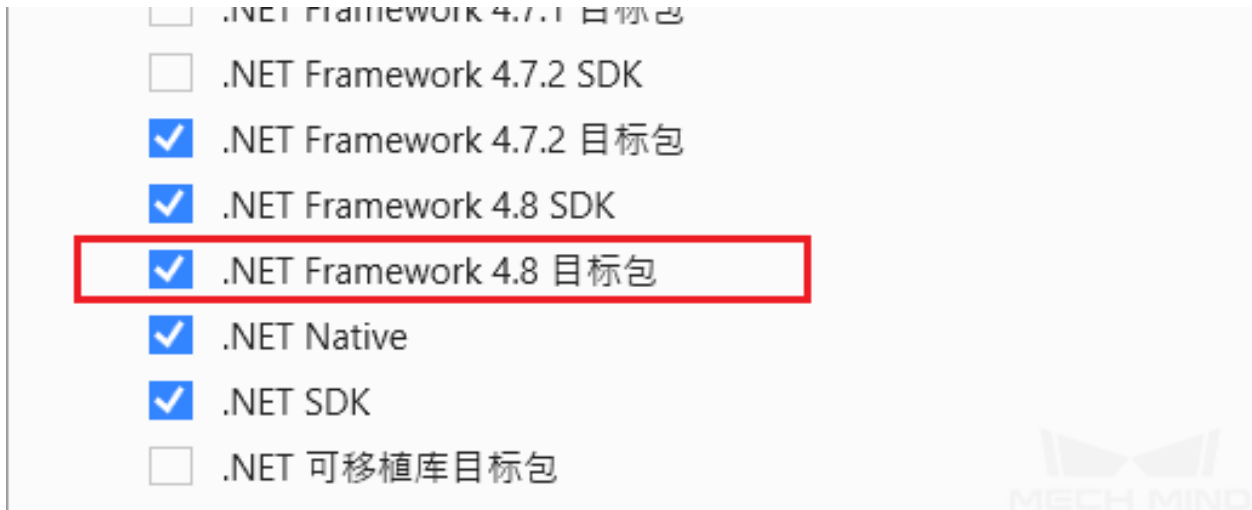
2. 安装 Visual Studio 相关组件：

工作负荷：.NET 桌面开发、使用 C++ 的桌面开发、通用 Windows 平台开发。

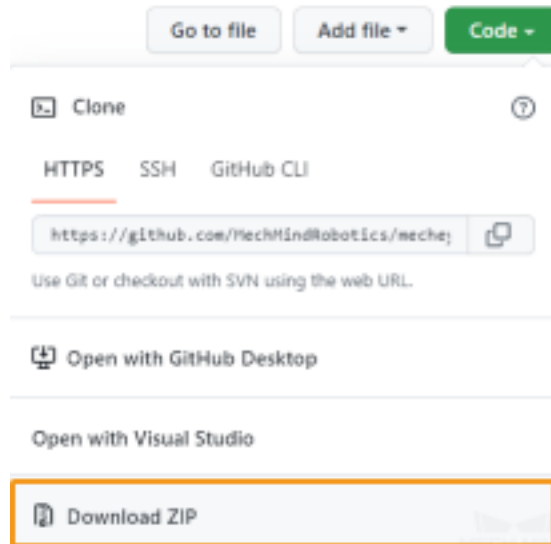
桌面应用和移动应用 (5)



单个组件：.NET Framework 4.8 目标包。



3. 下载并安装 *Mech-Eye SDK* 。
4. 下载 C# 例程。依次单击 *Code* ▶ *Download ZIP* 即可下载。

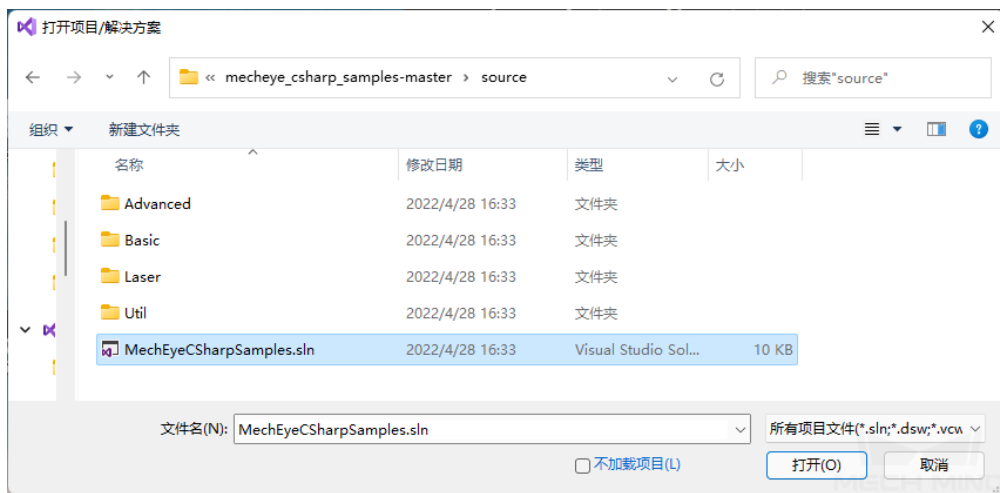


5. 连接真实可用的相机。
6. 如需使用以下例程，需通过 Visual Studio 的 NuGet 包管理器安装 *Emgu.CV.runtime.windows*。使用方法参见：[使用 NuGet 包管理器在 Visual Studio 中安装和管理包](#)。
  - *CaptureColorMap*
  - *CaptureDepthMap*
  - *CapturePointCloud*
  - *CaptureHDRPointCloud*
  - *CapturePointCloudROI*
  - *CapturePointCloudFromTextureMask*
  - *CaptureCloudFromDepth*
  - *CaptureSequentiallyMultiCamera*
  - *CaptureSimultaneouslyMultiCamera*
  - *CaptureTimedAndPeriodically*

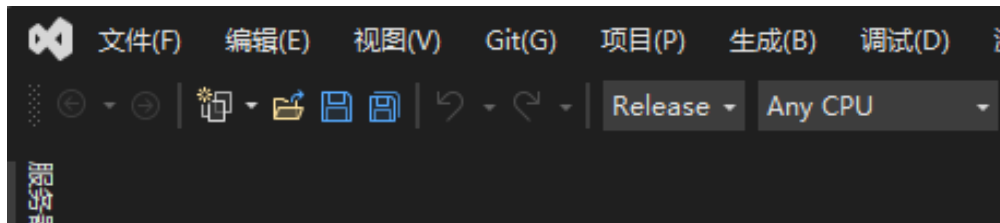
### 11.3.3 使用指南

#### 使用 Visual Studio 生成解决方案

1. 使用 Visual Studio 打开解决方案文件 MechEyeCSharpSamples.sln 。

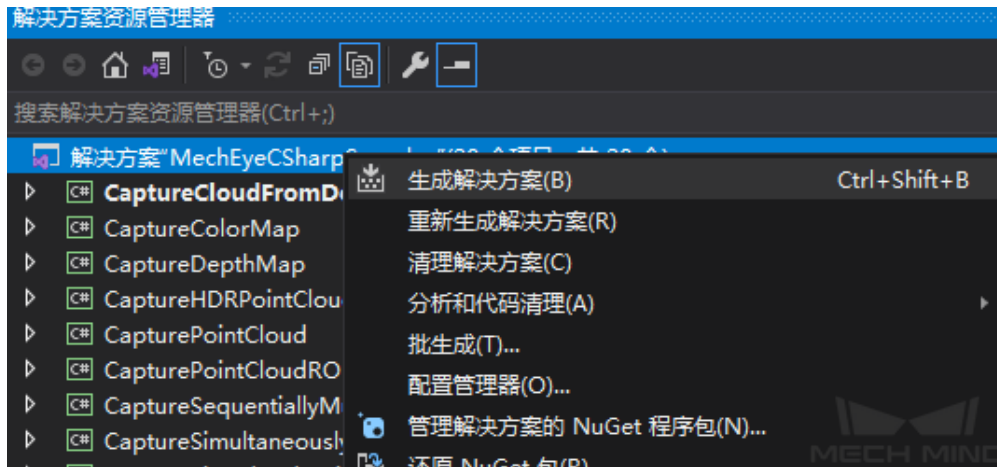


2. 将活动解决方案配置由 Debug 更改为 Release 。



3. 生成解决方案。

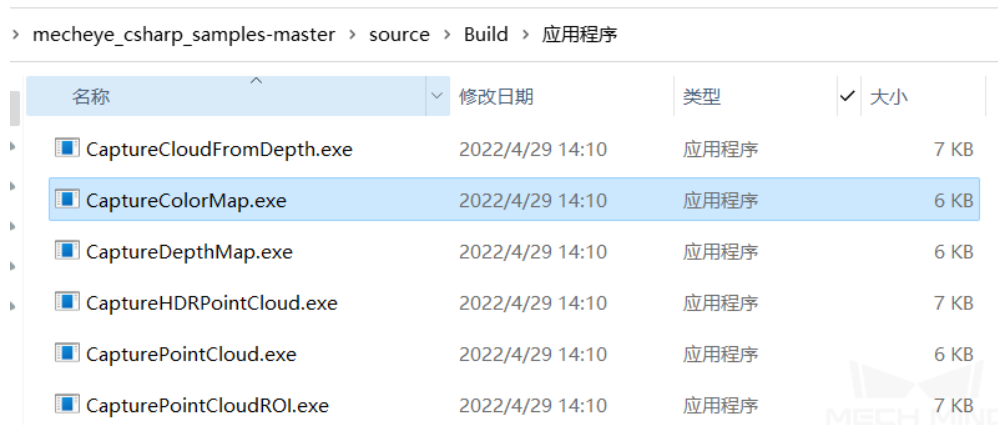
在 解决方案资源管理器窗口中，右键单击 解决方案 “MechEyeCSharpSamples”，在弹出的菜单中，单击 生成解决方案。



## 运行例程

以 CaptureColorMap.exe 为例：

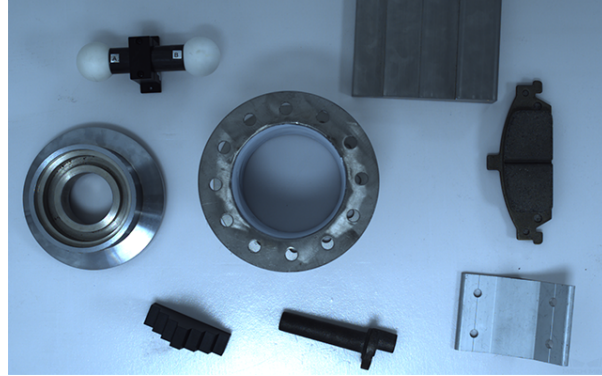
1. 进入到 `mecheye_csharp_samples-master` ▶ `source` ▶ `Build` 目录下。



2. 双击 CaptureColorMap.exe。

程序运行过程中，请根据提示选择待连接相机编号，并等待程序运行完毕。

3. 例程运行完成后，将在 Build 目录下得到：ColorMap（彩色图）。



## 11.4 Python

本章将介绍如何在 Windows 上运行 Python 例程。

### 11.4.1 例程简介

例程分为 5 类：**Basic**、**Advanced**、**Util**、**Laser** 和 **UHP**。

**Basic** 例程：与连接和基础采集相关。

**Advanced** 例程：与高级采集技巧相关。

**Util** 例程：与获取相机信息及设置参数相关。

**Laser** 例程：仅适用于 Laser, LSR 和 DEEP 系列相机。

**UHP** 例程：仅适用于 UHP 系列相机。

- **Basic**

- **ConnectToCamera**: 连接相机。
- **ConnectAndCaptureImage**: 连接相机并获取 2D 图、深度图及点云数据。
- **CaptureColorMap**: 从相机获取并保存 2D 图。
- **CaptureDepthMap**: 从相机获取并保存深度图。
- **CapturePointCloud**: 使用单个曝光时间采集图像，生成并保存白色点云和彩色点云。
- **CaptureHDRPointCloud**: 使用多个曝光时间采集图像，生成并保存白色点云和彩色点云。
- **CapturePointCloudROI**: 获取并保存感兴趣区域内的白色点云和彩色点云。
- **CapturePointCloudFromTextureMask**: 从覆盖掩膜的 2D 图和深度图生成并保存白色点云和彩色点云。

- **Advanced**

- `CaptureCloudFromDepth`: 从 2D 图与深度图生成并保存点云。
- `CaptureSequentiallyMultiCamera`: 使用多台相机按序获取并保存 2D 图、深度图及点云。
- `CaptureSimultaneouslyMultiCamera`: 使用多台相机同时获取并保存 2D 图、深度图及点云。
- `CaptureTimedAndPeriodically`: 在设定时间内，定时获取并保存 2D 图、深度图和点云。

- **Util**

- `GetCameraIntri`: 获取并打印相机内参。
- `PrintDeviceInfo`: 获取并打印相机型号、序列号、固件版本、温度等信息。
- `SetDepthRange`: 设置相机深度范围。
- `SetParameters`: 设置相机参数。
- `SetUserSets`: 执行参数组相关功能，如获取参数组名称，修改参数组名称，保存参数组参数信息等。通过参数组可以保存一组参数值并快速应用。

- **Laser**

- `SetLaserFramePartitionCount`: 将投影仪视野分为多个分区，结构光一次投射到一个分区。整个视野的输出由所有分区的图像融合而成。
- `SetLaserFrameRange`: 设置结构光投射范围，整个视野范围从 0 到 100。
- `SetLaserFringeCodingMode`: 设置结构光样式的编码模式。
- `SetLaserPowerLevel`: 设置激光器的输出功率（最大输出功率的百分比），影响激光强度。

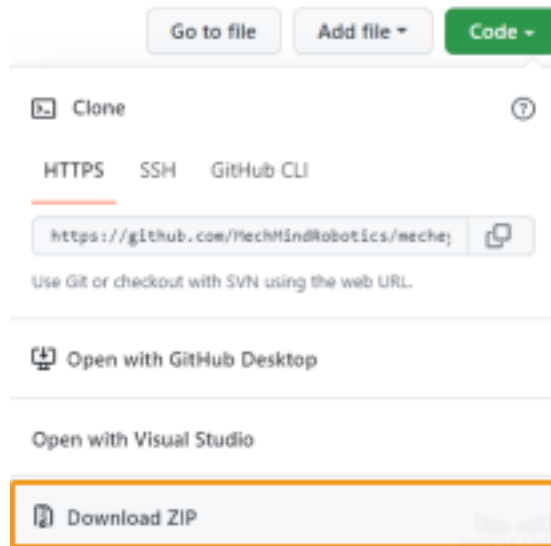
- **UHP**

- `SetUHPCaptureMode`: 设置采集模式（分为：2D 相机 1，2D 相机 2 及同时使用两个 2D 相机并融合输出）。
- `SetUHPFringeCodingMode`: 设置结构光样式的编码模式。

## 11.4.2 准备工作

1. 下载并安装 *Mech-Eye SDK*。
2. 下载 Python 例程。依次单击 *Code* ▶ *Download ZIP* 即可下载。





3. 确保 Python 版本在 3.6.5 至 3.10 之间。

**注意：**如安装 Python 时未勾选 **Add Python to PATH**，请将以下两个路径添加至 **Path** 环境变量中：

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

4. 连接真实可用的相机。

### 11.4.3 使用指南

#### Python 相关配置

以管理员身份运行 Windows PowerShell。

- 必装：MechEyeAPI。

```
pip install MechEyeAPI
```

若安装报错，可使用以下命令：

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

- 可选：open3d 和 opencv-python。

例程	open3d	opencv-python
CaptureColorMap		√
CaptureDepthMap		√
CapturePointCloud	√	
CaptureHDRPointCloud	√	
CapturePointCloudROI	√	
CaptureCloudFromDepth	√	

安装命令:

```
pip install opencv-python
```

## 运行例程

以 CaptureColorMap.exe 为例:

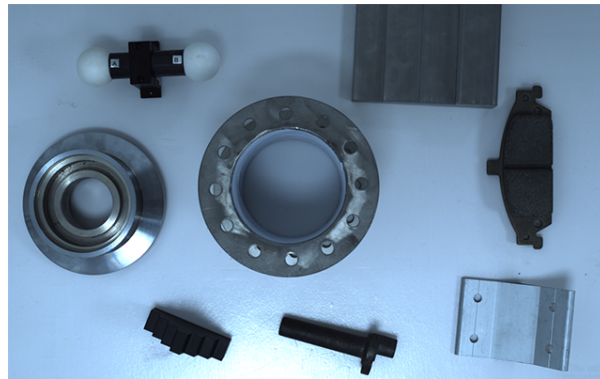
1. 使用 cd 命令切换到 CaptureColorMap.py 所在的目录。

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

2. 运行例程。

```
python CaptureColorMap.py
```

3. 例程运行完成后, 将在 Basic 目录下得到: ColorMap (彩色图)。



## 11.5 Python (Ubuntu)

本章将介绍如何在 Ubuntu 上运行 Python 例程。

### 11.5.1 例程简介

例程分为 5 类：**Basic**、**Advanced**、**Util**、**Laser** 和 **UHP**。

**Basic** 例程：与连接和基础采集相关。

**Advanced** 例程：与高级采集技巧相关。

**Util** 例程：与获取相机信息及设置参数相关。

**Laser** 例程：仅适用于 Laser，LSR 和 DEEP 系列相机。

**UHP** 例程：仅适用于 UHP 系列相机。

- **Basic**

- **ConnectToCamera**: 连接相机。
- **ConnectAndCaptureImage**: 连接相机并获取 2D 图、深度图及点云数据。
- **CaptureColorMap**: 从相机获取并保存 2D 图。
- **CaptureDepthMap**: 从相机获取并保存深度图。
- **CapturePointCloud**: 使用单个曝光时间采集图像，生成并保存白色点云和彩色点云。
- **CaptureHDRPointCloud**: 使用多个曝光时间采集图像，生成并保存白色点云和彩色点云。
- **CapturePointCloudROI**: 获取并保存感兴趣区域内的白色点云和彩色点云。
- **CapturePointCloudFromTextureMask**: 从覆盖掩膜的 2D 图和深度图生成并保存白色点云和彩色点云。

- **Advanced**

- **CaptureCloudFromDepth**: 从 2D 图与深度图生成并保存点云。
- **CaptureSequentiallyMultiCamera**: 使用多台相机按序获取并保存 2D 图、深度图及点云。
- **CaptureSimultaneouslyMultiCamera**: 使用多台相机同时获取并保存 2D 图、深度图及点云。
- **CaptureTimedAndPeriodically**: 在设定时间内，定时获取并保存 2D 图、深度图和点云。

- **Util**

- **GetCameraIntri**: 获取并打印相机内参。
- **PrintDeviceInfo**: 获取并打印相机型号、序列号、固件版本、温度等信息。
- **SetDepthRange**: 设置相机深度范围。
- **SetParameters**: 设置相机参数。
- **SetUserSets**: 执行参数组相关功能，如获取参数组名称，修改参数组名称，保存参数组参数信息等。通过参数组可以保存一组参数值并快速应用。

- **Laser**

- **SetLaserFramePartitionCount**: 将投影仪视野分为多个分区，结构光一次投射到一个分区。整个视野的输出由所有分区的图像融合而成。
- **SetLaserFrameRange**: 设置结构光投射范围，整个视野范围从 0 到 100。
- **SetLaserFringeCodingMode**: 设置结构光样式的编码模式。
- **SetLaserPowerLevel**: 设置激光器的输出功率（最大输出功率的百分比），影响激光强度。

• **UHP**

- **SetUHPCaptureMode**: 设置采集模式（分为：2D 相机 1，2D 相机 2 及同时使用两个 2D 相机并融合输出）。
- **SetUHPFringeCodingMode**: 设置结构光样式的编码模式。

部分例程需安装第三方库后，才可使用，详见下表：

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

## 11.5.2 准备工作

1. 连接真实可用的相机。
2. 确保 Python 版本在 3.6.5 至 3.10 之间。
3. 下载并安装 *Mech-Eye SDK*。
4. (可选) 安装 OpenCV。

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

5. (可选) 安装 Open3D。

```
sudo pip install open3d
```

## 11.5.3 使用指南

### 安装软件

#### 在线安装

1. 安装 pip。

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

如已安装 pip, 需保证 pip 版本在 20.3 或以上。使用以下指令升级 pip:

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

## 2. 安装 MechEyeApi。

```
sudo pip3 install MechEyeApi
```

### 离线安装

1. 确认本地已有 .whl 安装包。
2. 安装 .whl 安装包。

```
pip3 install xxx.whl
```

3. 查看软件包是否已安装。

```
pip3 list
```

### 运行例程

1. 下载例程。

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

2. 进入各个例程文件夹内, 运行程序 (此处以 ConnectToCamera.py 为例)。

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

输出结果如下所示:

```
Find Mech-Eye device...  
Mech-Eye device index : 0  
Camera Model Name: Mech-Eye Pro M Enhanced  
Camera ID:          NEC15221A3000001  
Camera IP Address: 192.168.xx.xx  
Hardware Version:  Vx.x.x  
Firmware Version:  Vx.x.x  
.....  
Please enter the device index you want to connect: 0
```

请根据提示选择待连接相机编号, 程序运行完毕, 输出结果如下所示:

```
Connected to the Mech-Eye device successfully.  
Disconnected from the Mech-Eye device successfully.
```

**注意:**

- 涉及采集点云的例程，普通用户没有权限对点云文件进行写入操作，所以使用 `sudo` 命令以系统管理者的身份执行例程程序。
- 涉及采集图像或点云的例程，运行结束后，默认将图片或点云数据保存至该例程文件所在的目录下。
- 调用 Mech-Eye API 前，需调用接口命令，命令为：

```
python3
import MechEye
```

- 请在 `import open3d` 前 `import MechEye`。

## 11.6 ROS

本章介绍在 Ubuntu 系统下的 ROS 平台中使用 Mech-Eye SDK。

### 11.6.1 准备工作

1. 操作系统为 Ubuntu 系统。
2. 根据 ROS 官方安装文档 安装 ROS。

**注意事项:**

1. 示例中使用的 Ubuntu 系统版本号为 **18.04**，不同版本的 Ubuntu 系统安装 ROS 会有所不同。
2. 若使用虚拟机安装软件，请预留大于 **20G** 的磁盘空间，否则可能会导致安装失败。
3. 若以下操作可正常运行小乌龟程序，即可视为 ROS 安装成功：
  - a. 任意路径下，打开一个终端，输入命令：

```
roscore
```

- b. 打开第二个终端，输入命令：

```
roslaunch turtlesim turtlesim_node
```

- c. 打开第三个终端，输入命令：

```
roslaunch turtlesim turtle_teleop_key
```

执行完上述命令后，在第三个终端窗口中，通过操作键盘上的上下左右键即可操控小乌龟，如下图所示。



## 11.6.2 使用指南

### 安装 Ubuntu 版本 Mech-Eye SDK

安装方法参考在 *Ubuntu* 上安装 *Mech-Eye SDK* 。

## 安装 ROS API

1. 安装 libzmq5 和 libzmq3-dev。

```
sudo apt install libzmq5 libzmq3-dev
```

2. 创建 ~/ros\_ws/src 目录并进入该目录。

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

3. 将 mecheye\_ros\_interface 仓库克隆至 ~/ros\_ws/src 目录。

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

4. 进入 ~/ros\_ws 目录并构建代码。

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

5. 修改文件。

- 如需存储图片，请将 ~/ros\_ws/src/mecheye\_ros\_interface/launch/start\_camera.launch 文件中此行 `<arg name="save_file" default="true"/>` 的 false 更改为 true。
- 存储图片的路径可以在 ~/ros\_ws/src/mecheye\_ros\_interface/src/main.cpp 中进行修改。

## 运行服务

1. 打开一个新终端，运行以下命令以开启服务。

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

输出结果如下所示：

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

started roslaunch server http://ubuntu:38711/

SUMMARY
=====

PARAMETERS
* /mechmind_camera_start/camera_ip: 192.168.0.123
* /mechmind_camera_start/fx: 1727.46410256
* /mechmind_camera_start/fy: 1727.45869267
* /mechmind_camera_start/save_file: False
* /mechmind_camera_start/u: 655.818082573
```

(下页继续)



(续上页)

```

* /mechmind_camera_start/use_external_intrintri: False
* /mechmind_camera_start/v: 516.630650061
* /rostdistro: melodic
* /rosversion: 1.14.13

NODES
/
  mechmind_camera_start (mecheye_ros_interface/start)

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

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

2. 打开另一个新的终端，运行以下命令采集图片。

```

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

输出结果如下所示：

```

success: True
message: ''
    
```

此时相机已经启用！2D图及深度图等数据默认保存在 /tmp 文件夹中。

**注意：** 每次修改 ~/ros\_ws/src/mecheye\_ros\_interface/src 下的文件，都需要执行 cd ~/ros\_ws && catkin\_make 更新可执行文件。

---

## API——固件升级工具

---

使用该工具可直接在 Ubuntu 系统中升级相机固件，简化相机固件升级的流程。

固件升级工具包含两部分：可执行文件 **firmware\_update\_tool** 和固件升级文件包 **mmind\_eye\_vx.x.x.zip**。

请执行以下步骤升级相机固件：

1. 下载并安装 *Mech-Eye SDK*。
2. 请执行以下命令，为 **firmware\_update\_tool** 增加可执行权限。

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

3. 请执行以下命令，进行固件升级。需将 **x.x.x** 替换为所安装 Mech-Eye SDK 的实际版本号。

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

---

### 提示：

- 两个路径需要用空格区分。路径之间无空格时，路径无法被识别，将导致固件升级工具无法运行。
- **firmware\_update\_tool** 的路径在先，**mmind\_eye\_vx.x.x.zip** 的路径在后。路径顺序错误将导致固件升级工具无法运行。

4. 根据提示输入需升级固件的相机编号 (device index)，并按回车键连接相机。如显示以下内容，固件升级成功。

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

**注解：** 如固件升级失败，请重新尝试或联系技术支持。

---

## CHAPTER 13

---

### Mech-Eye API 参考手册

---

欢迎使用 Mech-Eye API! 更多信息请查阅: [Mech-Eye API 参考手册](#)。

---

## 快速了解 GenICam

---

Mech-Eye 工业级 3D 相机（简称相机）支持 GenICam 标准，允许被兼容 GenICam 标准的第三方机器视觉软件控制（简称“GenICam 客户端”）。

相机固件升级至 2.0.0 或以上版本后，不需要更多操作，就可以直接使用 GenICam 客户端，例如 HALCON (MVTec)、VisionPro (Cognex)，来控制相机。

---

查看以下内容，了解 **GenICam** 。

*GenICam 基础信息*

---

查看以下内容，了解 **如何使用 HALCON 控制相机**。

- *HALCON*——连接相机、调节参数、采集图像
  - *HALCON*——手眼标定
  - *HALCON*——设置 *IP* 地址
  - *HALCON*——仅获取 *Z* 向值
  - *HALCON*——获取纹理点云
  - 通过 *Mech-Eye API* 获取 *HALCON* 可读取的点云
- 

查看以下内容，了解更多参考信息。

*GenICam 支持的相机参数*

借助 *Mech-Eye Viewer* 为 *GenICam* 客户端设置相机参数

---

本文包含如下内容：

- 介绍
- 工作原理
- 支持 *GenICam*

### 15.1 介绍

GenICam 是由欧洲机器视觉协会（EMVA）制定的机器视觉行业通用标准，允许使用通用的编程接口控制机器视觉相机。

GenICam 提供了一种普遍适用的编程接口，它为包括 GigE Vision、USB 3.0 Vision、Camera Link 和 IEEE 1394 等在内的接口提供了一个端到端配置接口。

对于符合 GenICam 标准的相机，可以使用兼容 GenICam 的机器视觉软件，例如 HALCON (MVTec)、VisionPro (Cognex)，便捷地连接和控制相机，而无需对相机进行特定的配置。

## 15.2 工作原理

1. 符合 GenICam 标准的相机，提供一个 XML 描述符文件。XML 文件用于描述相机参数，并指明相机提供的特性和功能。
2. 兼容 GenICam 的机器视觉软件，将 XML 文件转换为 GenAPI 应用编程接口或图形用户界面的元素。
3. 用户通过使用机器视觉软件上的应用编程接口或图形用户界面，轻松访问和使用相机提供的特性和功能（例如曝光时间等）。

关于 GenICam 的更多信息，请访问：<https://www.emva.org/standards-technology/genicam>。

## 15.3 支持 GenICam

相机固件升级至 2.0.0 或以上版本后，相机支持 GenICam/GigE Vision 标准，允许被 HALCON (MVTec)、VisionPro (Cognex) 等第三方机器视觉软件控制。

关于相机为 GenICam 提供的相机参数及其说明，请参见 *GenICam* 支持的相机参数。

如果需要使用 HALCON 连接和管理相机，请参见使用 *HALCON* 控制相机。

---

## HALCON——连接相机、调节参数、采集图像

---

本章介绍使用 HALCON 例程连接 Mech-Eye 工业级 3D 相机、调节参数并采集图像。HALCON 软件支持通过 GigE Vision2 协议与相机进行连接。本章主要内容包含：

- 使用前提
- 使用 *HALCON* 助手
  - 连接相机
  - 采集图像
  - 选择数据类型
  - 设置采集区域
  - 调节参数
- 使用 *HALCON* 例程
  - 运行例程
  - 调节参数
  - 参考信息



## 16.1 使用前提

- 相机和工控机已正确连接。

---

**提示：** Nano 和 Pro XS 相机推荐与工控机直接连接，不使用交换机。

---

- 工控机上已安装 HALCON 20.11 或以上。

---

**提示：** 低于 20.11 版本的 HALCON 未经完全验证。

---

- 工控机上已安装 Mech-Eye SDK 2.0.0 或以上。

---

**提示：** 相机固件版本与软件版本一致时，才可使用 Mech-Eye SDK。如需升级固件版本，请参考升级相机固件。

---

- 相机与工控机 IP 地址在同一网段。

---

**提示：** 建议将相机与工控机设置为静态 IP 地址，IP 地址设置请参考使用 *Mech-Eye Viewer* 设置 IP 地址。

---

安装 HALCON 软件后，您可以通过以下方式连接相机。

- 使用 HALCON 助手
- 使用 HALCON 例程

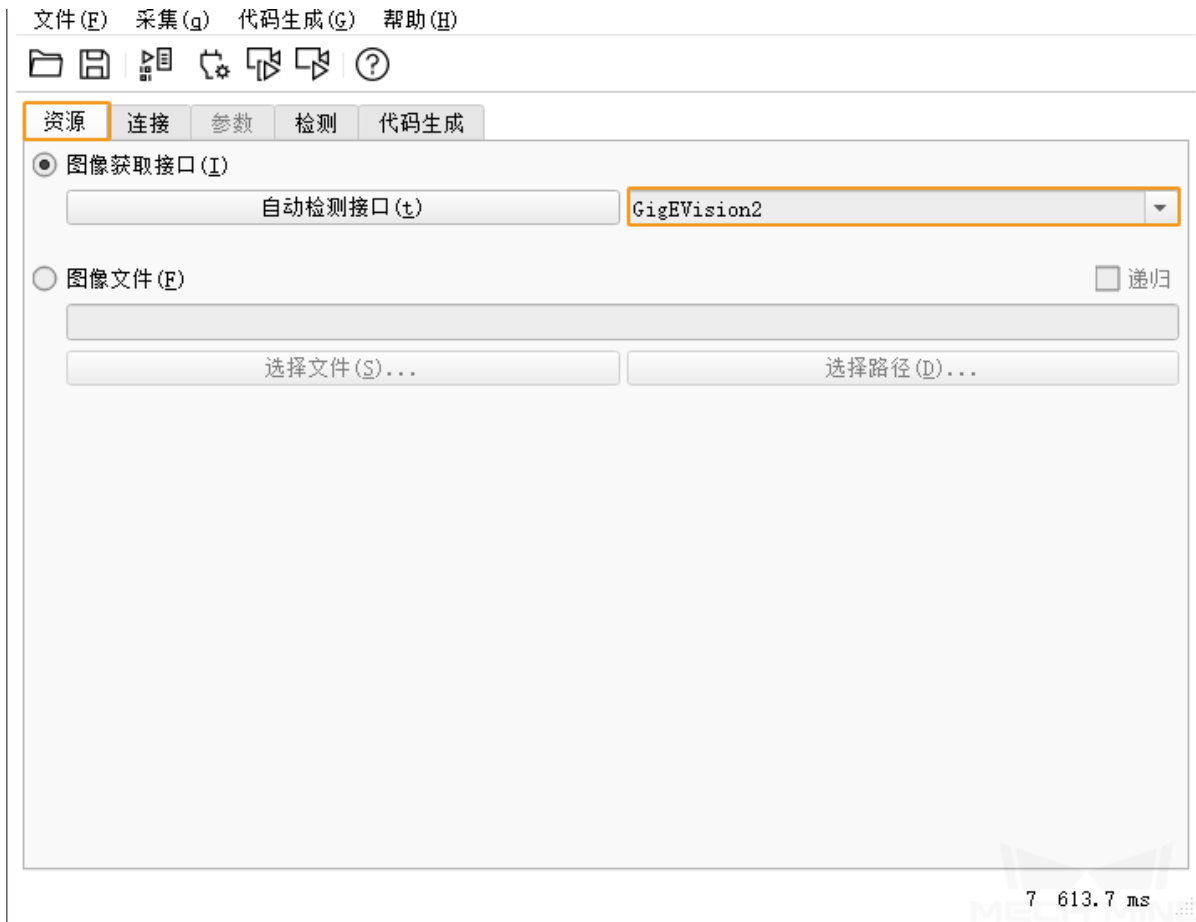
## 16.2 使用 HALCON 助手

使用 HALCON 中的图像采集助手可快速连接相机、采集图像和调节参数。

在工控机上运行 HALCON 软件，选择 助手 ▶ 打开新的 *Image Acquisition*，即可打开图像采集助手。

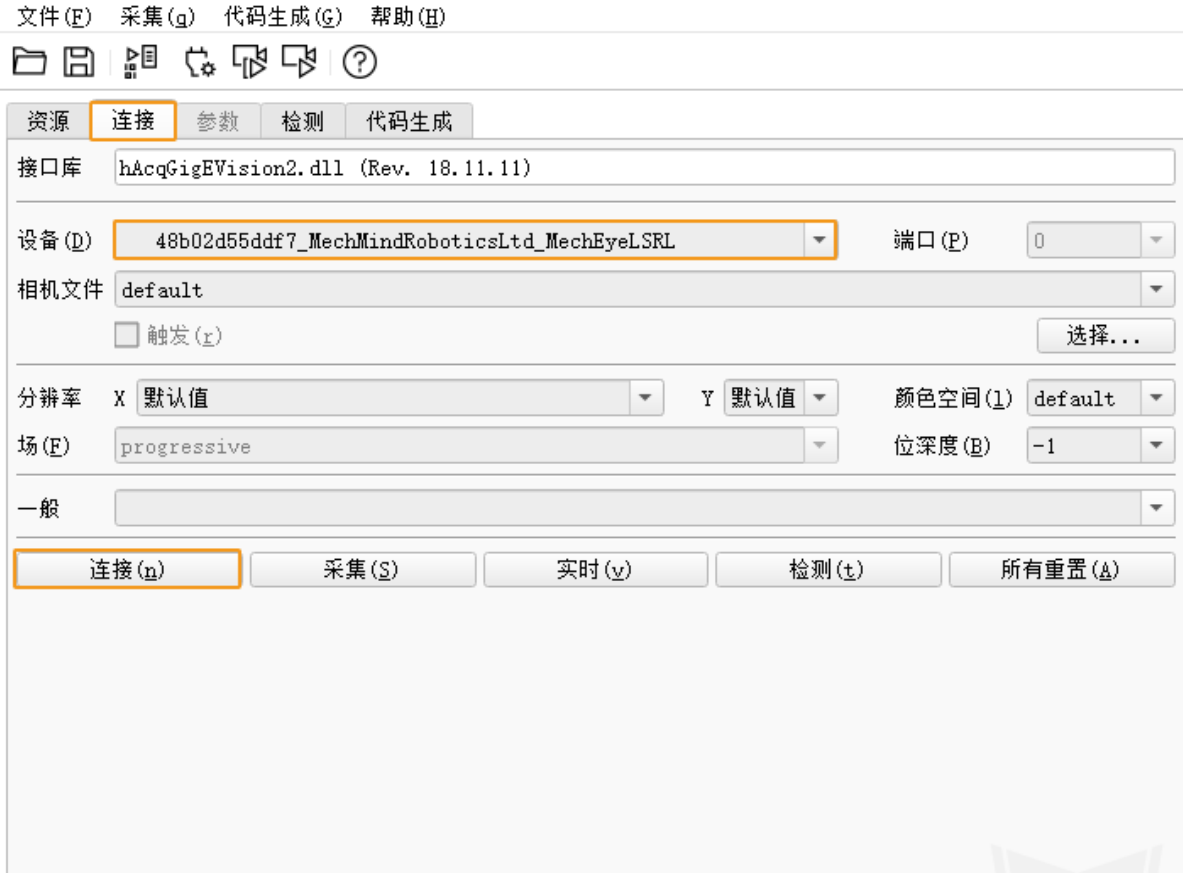
### 16.2.1 连接相机

1. 在助手窗口的 资源选项卡中，选择 图像获取接口，并从下拉菜单中选择 **GigEVision2**。



**提示：**如下拉菜单中无 **GigEVision2** 选项，表明 GigEVision2 图像获取接口未安装。请参考 HALCON 的 [安装指南](#)，通过 MVTec 软件管理器（SOM）安装接口。

2. 在 **连接** 选项卡中，选择 **设备**，单击 **连接** 可连接相机。



7 613.7 ms

**提示:**

- 相机连接成功，连接按钮变为 断开；如无变化，则为连接失败。如需断开与相机的连接，在 连接选项卡中单击 断开按钮。
- 如相机已被连接，则连接一定不成功。请检查相机是否被其他客户端连接，相机断开连接后，再使用 HALCON 连接。
- 设备中的相机显示名称较长，可使用 Mech-Eye Viewer 连接相机后，自定义相机显示名称，便于查找需要连接的相机。推荐设置为英语名称。

## 16.2.2 采集图像

相机连接后，在 **连接** 选项卡中单击 **采集** 即可进行单次图像采集。

---

**注解：** 如相机采图时间较长，工控机可以开启 **巨型帧** 功能。

---

您也可以进行多次或连续图像采集，为此需先调节 **AcquisitionMode** 参数的值。

- 进行多次图像采集：
  1. 切换至 **参数** 选项卡，将 **AcquisitionMode** 参数设置为 **MultiFrame**。
  2. 单击右上角的 **刷新** 后，在 **AcquisitionFrameCount** 中设置需进行的图像采集次数。
  3. 切换至 **连接** 选项卡，单击 **实时** 进行图像采集。
  4. 设定次数的图像采集完成后，**实时** 按钮变为 **停止**。单击 **停止** 停止图像采集。
- 进行连续图像采集：
  1. 切换至 **参数** 选项卡，将 **AcquisitionMode** 参数设置为 **Continuous**。
  2. 切换至 **连接** 选项卡，单击 **实时** 进行图像采集。
  3. **实时** 按钮变为 **停止**。单击 **停止** 停止图像采集。

---

**注解：**

- **AcquisitionMode** 参数设置为 **MultiFrame** 或 **Continuous** 时，仍可单击 **采集** 进行单次图像采集。
  - **AcquisitionMode** 参数设置为 **SingleFrame** 时，只可进行单次图像采集。
- 

## 16.2.3 选择数据类型

相机连接后，默认采集的数据类型是 2D 图。您可通过调节 **DeviceScanType** 参数选择要获取的数据类型（2D 图或深度图）。

1. 单击 **参数** 选项卡，将 **DeviceScanType** 参数设置为需要的数据类型。数据类型说明如下：

值	数据类型
Areascan	2D 图
Areascan3D	深度图（包含深度信息的 2D 图）

2. 进行图像采集，获取所选的数据类型。

---

**提示：** 勾选 **参数** 选项卡右上角的 **更新图像**，调节参数后 **Canvas** 窗口中的图像将自动更新。

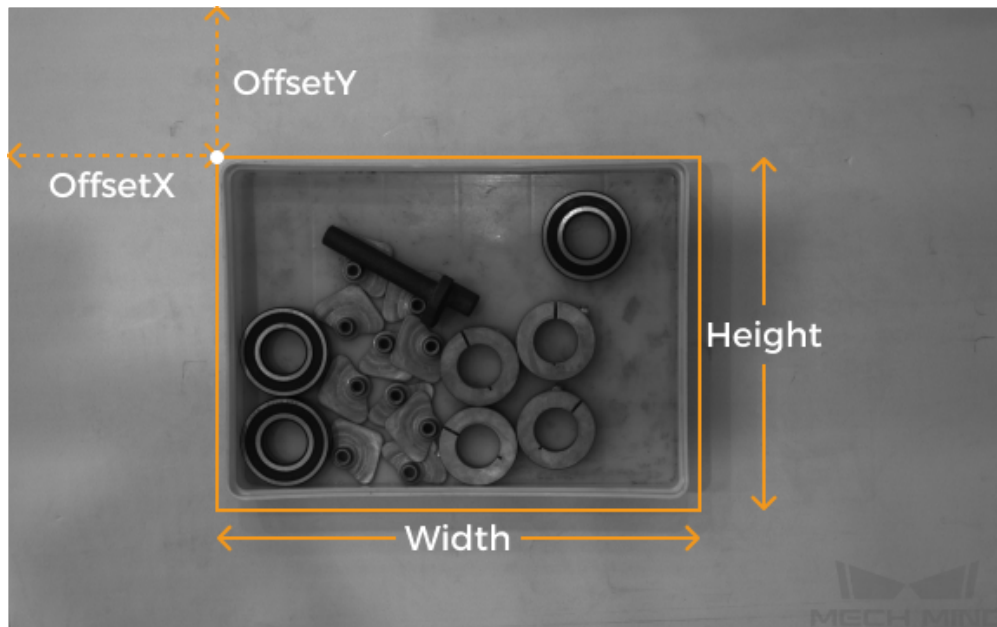
---

## 16.2.4 设置采集区域

相机连接后，如需对获取的图像进行裁剪，可通过调整 **Height**、**Width**、**OffsetX** 和 **OffsetY** 四个参数设置采集区域。

请按照如下步骤设置采集区域：

1. 选择需要设置采集区域的数据类型。
2. 单击 采集进行单次图像采集，确认当前图像。
3. 切换至 参数选项卡，调节 **Height**、**Width**、**OffsetX** 和 **OffsetY** 参数。下图展示这四个参数及其所定义的采集区域（橙色方框）与原始图像的关系。



- **Width** : 采集区域宽度
- **Height** : 采集区域高度
- **OffsetX** : 采集区域左上角的 x 坐标（原始图像左上角坐标为 (0, 0)）
- **OffsetY** : 采集区域左上角的 y 坐标

---

注解：以上四个参数的设置需满足如下要求：

- $(\text{Width} + \text{OffsetX})$  不超过原始图像的宽度
- $(\text{Height} + \text{OffsetY})$  不超过原始图像的高度

原始图像的宽度和高度可在 只读参数的 **WidthMax** 和 **HeightMax** 中查看（需将 可视度 设为 专家或以上）。

---

4. 再次进行图像采集，确认裁剪结果。

提示：勾选 参数选项卡右上角的 **更新图像**，调节参数后 **Canvas** 窗口中的图像将自动更新。

5. 切换至 **代码生成**选项卡，单击 **插入代码**生成相应的代码。
6. 如需为另一数据类型设置采集区域：
  1. 在当前助手窗口中断开相机连接。
  2. 打开一个新的助手窗口并连接相机。
  3. 选择另一数据类型并重复上述步骤。

#### 注解：

- 上述四个采集区域参数不会被保存至参数组中。如相机断电，其参数值将被重置。如需记录其参数值，请生成并保存代码。
- 更改 **DeviceScanType** 和 **Scan3DBinningEnable** 参数也将重置上述采集区域参数的值。

### 采集区域与 Scan3DROI 的区别

Mech-Eye 工业级 3D 相机还提供了另外一组用于设置感兴趣区域的参数：**Scan3DROILeft**、**Scan3DROITop**、**Scan3DROIHeight** 和 **Scan3DROIWidth**（统称为“Scan3DROI”）。

采集区域参数和 Scan3DROI 的区别如下表。请根据需求选择要使用的参数。

采集区域	Scan3DROI
不可保存至参数组，相机断电将重置参数值	可保存至参数组
2D 图和深度图均可设置	2D 图不可设置
对图像进行裁剪	不对图像进行裁剪
只可在 HALCON 中设置	可用 <i>Mech-Eye Viewer</i> 中的可视化工具设置

### 16.2.5 调节参数

如获取的数据不满足需求，可调节相应参数。

#### 提示：

- 相机支持参数组功能，通过切换参数组可以快速切换相机的参数配置。相机参数需要在指定的参数组下设置。
- 如果需要增加或删除参数组，请通过 *Mech-Eye Viewer* 配置。使用 *Mech-Eye Viewer* 配置完参数组后，在 HALCON 参数选项卡下，单击右上角的 **刷新按钮**读取相机最新配置。

执行如下步骤调节参数：

1. 相机连接后，单击 **参数**选项卡，将 **UserSetSelector** 参数设置为要修改的参数组。



2. 找到 **UserSetLoad** 参数，单击右侧的 应用读取配置。

**注解：**如单击 应用后，各参数的值未更新，请再次单击。

3. 找到需要修改的参数，并修改参数值。
4. 找到 **UserSetSave** 参数，单击右侧的 应用保存配置。
5. 切换至 代码生成选项卡，单击 插入代码生成相应的代码。

参考信息：

- 关于 GenICam 支持的相机参数的描述和解释，请下载和查阅 [GenICam 参数说明](#)。
- GenICam 支持的相机参数与 Mech-Eye Viewer 提供的参数基本上一一对应。关于参数的对应关系，请参见 [GenICam 支持的相机参数](#)。
- 设置 自动曝光感兴趣区域、深度设置及 感兴趣区域等相机参数时，通常需要借助可视化工具进行不断调整从而获取理想的值。由于 HALCON 并未提供辅助参数设置的可视化工具，您可以借助 Mech-Eye Viewer 的可视化设置工具来设置这些参数。详细信息请查看借助 [Mech-Eye Viewer 为 GenICam 客户端设置相机参数](#)。

**提示：**使用 Mech-Eye Viewer 连接相机前，需断开相机与 HALCON 的连接。如连接不成功，请关闭 HALCON 软件后，再重新连接。


## 16.3 使用 HALCON 例程

为方便使用，本公司在 GitHub 上提供了 HALCON 例程。该例程可用于连接相机、采集数据、修改参数并保存数据。用户可以直接使用或者根据实际需要进行修改以满足项目需求。

**注解：**HALCON 例程使用 C++ 语言编写，可以在 HALCON 软件的 HDevelop 集成开发环境中直接运行。

### 16.3.1 运行例程

请执行以下步骤运行例程：

1. 获取 `connect_to_camera_and_capture_images` 例程，依次单击 *Code* ▶ *Download ZIP* 即可下载。
2. 在 HALCON 中打开例程：运行 HALCON，将例程拖拽至 HALCON 的 **程序窗口**中。
3. 选中 **程序窗口**中 `info_framegrabber` 算子所在的行，单击工具栏中的  按钮或单击键盘上的 F6，运行该算子。
4. 双击 **控制变量区域**的 `DeviceInfos`，可查看所有可连接的相机。

控制变量


Info	'Info about GigE Vision devices in the system.'
DeviceInfos	[ '   device:Halcon测试报\351   unique_name:48b02d6244eb_MechMindRoboticsLtd_Me...

5. 选中相机并双击，复制 `unique_name:` 或 `user_name:` 的相机名。


变量监视: DeviceInfos	
	DeviceInfos
0	'   device:P022206030802H   unique_name:dca5012fa8cf_MindVision_MGE134GM2   user_name:P022206030802H   interface:Esen_ITF_b07b2528849ac0a81425fffff00   producer:Esen'

6. 定位至如下命令行，将 `MechEye` 替换为 `unique_name` 或 `user_name`。

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

7. 单击工具栏中的  按钮或单击键盘上的 F5，运行例程。
8. 运行结束后，在例程文件夹内查看本次运行获取的 2D 图和点云数据结果，默认保存文件名为 `image2d.bmp` 和 `PointCloud.ply`。

**提示：**

- 例程运行结束后需单击工具栏中的  重置程序执行，否则将导致 Mech-Eye Viewer 无法连接到相机。



- 图形窗口显示点云后，需单击该窗口中的 *Continue* 继续执行。否则，程序会卡在 **visualize\_object\_model\_3d** 算子中。
- 如相机采图时间较长，可为工控机开启巨型帧功能。
- 网络较差导致数据丢失时，可以使用如下指令增大最大允许丢失包数。**ParameterValues** 替换为新的最大允许丢失包数。

```
set_framegrabber_param (AcqHandle, '[Stream]GevStreamMaxPacketGaps', ↵
↵ParameterValues)
```

### 16.3.2 调节参数

如获取的数据不满足需求，可调节相应参数。

提示：

- 相机支持参数组功能，通过切换参数组可以快速切换相机的参数配置。相机参数需要在指定的参数组下设置。
- 如果需要增加或删除参数组，请通过 Mech-Eye Viewer 配置。

执行如下步骤调节参数：

1. 使用以下算子获取相机的参数组列表。

```
get_framegrabber_param (AcqHandle, 'UserSetSelector_values', ParameterValues)
```

2. 在 **控制变量** 区域的 **ParameterValues** 中，查看所有的参数组名称。
3. 使用以下算子选择要修改的参数组。**UserSetSelector** 和 **UserSetLoad** 为相机参数，分别用于选择和加载参数组。**ParameterGroupName** 替换为实际参数组名称。

```
set_framegrabber_param (AcqHandle, 'UserSetSelector', 'ParameterGroupName')
set_framegrabber_param (AcqHandle, 'UserSetLoad', 'ParameterGroupName')
```

4. 使用以下算子获取指定参数的值。**ParameterName** 替换为实际相机参数名称。**ParameterValues** 是保存获取参数值的变量，可以根据需要调整。注意变量不需要置于引号内。

```
get_framegrabber_param (AcqHandle, 'ParameterName', ParameterValues)
```

5. 在 **控制变量** 区域的 **ParameterValues** 中，查看该参数的值。
6. 使用以下算子修改参数设置。**ParameterName** 替换为实际相机参数名称，**NewParameterValue** 替换为新的参数值。

```
set_framegrabber_param (AcqHandle, 'ParameterName', 'NewParameterValue')
```

7. 使用以下算子将参数设置修改保存到参数组中。**UserSetSave** 为相机参数，用于保存参数设置到参数组。**ParameterGroupName** 替换为实际参数组名称。

```
set_framegrabber_param (AcqHandle, 'UserSetSave', 'ParameterGroupName')
```

### 16.3.3 参考信息

- 关于 GenICam 支持的相机参数的描述和解释，请下载和查阅 [GenICam 参数说明](#)。
- GenICam 支持的相机参数与 Mech-Eye Viewer 提供的参数基本上一一对应。关于参数的对应关系，请参见 [GenICam 支持的相机参数](#)。
- 设置 [自动曝光感兴趣区域](#)、[深度设置](#) 及 [感兴趣区域](#) 等相机参数时，通常需要借助可视化工具进行不断调整从而获取理想的值。由于 HALCON 并未提供辅助参数设置的可视化工具，您可以借助 Mech-Eye Viewer 的可视化设置工具来设置这些参数。详细信息请查看借助 [Mech-Eye Viewer 为 GenICam 客户端设置相机参数](#)。

---

**提示：** 使用 Mech-Eye Viewer 连接相机前，需断开相机与 HALCON 的连接。如连接不成功，请关闭 HALCON 软件后，再重新连接。

---

本章介绍如何在 Windows 系统下通过梅卡曼德提供的例程进行手眼标定。梅卡曼德提供如下两个例程：

- `determine_calibration_poses.hdev`: 用于获取进行手眼标定时 的 标定位姿 (进行手眼标定时, 机器人在指定位置使用相机采集标定板图像。这一系列指定位姿即为标定位姿)。
- `perform_hand_eye_calibration.hdev`: 用于进行手眼标定。

注解:

- 目前仅适配六轴机器人。
- 使用 UHP 系列相机进行手眼标定时, 拍摄模式必须为 **Camera1**。

- 使用前提
- 修改例程
- 获取标定位姿
- 进行手眼标定
- 机器人欧拉角类型
- 标定参数说明

## 17.1 使用前提

使用 HALCON 进行手眼标定前，准备工作如下：

1. 确认机器人精度无误，且机器人可正常使用。
2. 准备相机随附的标定板，安装相机与标定板。
3. 下载并安装 *Mech-Eye SDK*。
4. 确认 2D 图与深度图的质量符合要求。
5. 确认相机内参符合要求。
6. 下载 HALCON 例程，依次单击 *Code* ▶ *Download ZIP* 即可下载。

## 17.2 修改例程

使用 `determine_calibration_poses.hdev` 获取标定姿态后，需使用 `perform_hand_eye_calibration.hdev` 进行手眼标定。


两个例程中的以下信息需保持一致：

- 连接的相机
- 标定板型号

另外，获取姿态前，需设置 `robot_pose.json` 文件中的欧拉角类型。

### 17.2.1 选择同一台相机

运行例程前，需修改例程中设置的相机，保证两个例程运行时连接同一台相机。设置步骤如下：

1. 在 HALCON 中打开例程：运行 HALCON，将例程拖拽至 HALCON 的程序窗口中。
2. 选中程序窗口中 `info_framegrabber` 算子所在的行，单击工具栏中的  按钮或单击键盘上的 F6，运行该算子。
3. 双击控制变量区域的 `DeviceInfos`，可查看所有可连接的相机。

控制变量


Info	'Info about GigE Vision devices in the system.'
DeviceInfos	[ '   device:Halcon测试报\351   unique_name:48b02d6244eb_MechMindRoboticsLtd_Me...

4. 选中相机并双击，复制 `unique_name:` 或 `user_name:` 的相机名。

变量监视: DeviceInfos	
	DeviceInfos
0	'   device:P022206030802H   unique_name:dca5012fa8cf_MindVision_MGE134GM2   user_name:P022206030802H   interface:Esen_ITF_b07b2528849ac0a81425ffffff00   producer:Esen'

5. 定位至如下命令行，将 **MechEye** 替换为 **unique\_name** 或 **user\_name**。

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

6. 单击工具栏中的  按钮或单击键盘上的 F5，运行例程。

## 17.2.2 设置标定板型号

设置标定板型号的步骤如下：

1. 在 HALCON 中打开例程：运行 HALCON，将例程拖拽至 HALCON 的程序窗口中。
2. 设置标定板型号。默认标定板型号为 BDB-5；如使用其他标定板，定位至如下算子，将 **BDB-5** 替换为对应的标定板型号。

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

## 17.2.3 设置欧拉角类型和单位

使用 **determine\_calibration\_poses.hdev** 获取的标定位姿需输入至 **robot\_pose.json** 文件中，**robot\_pose.json** 文件中默认的欧拉角类型为 **sxyz**，默认的欧拉角单位为角度。请执行以下步骤设置欧拉角类型与格式：

1. 打开 **robot\_pose.json** 文件。
2. 设置欧拉角类型。定位至如下命令行，将 **sxyz** 替换为所用机器人的欧拉角类型。例程已支持的欧拉角类型请参考 [机器人欧拉角类型](#)。

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

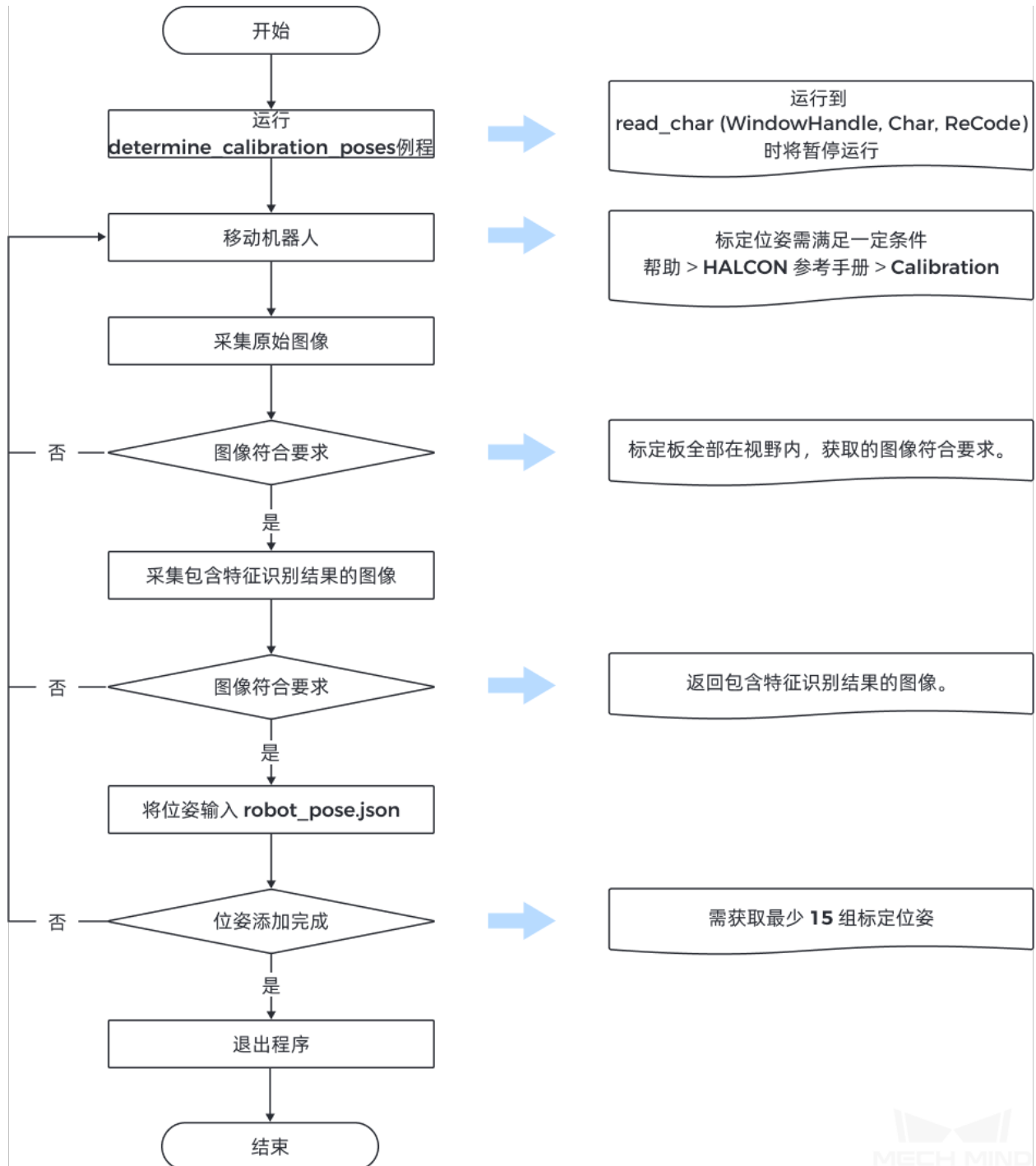
3. 设置欧拉角单位。如需输入以弧度为单位的欧拉角，定位至如下命令行，将 **true** 替换为 **false**。

```
"FromDegree": true
```

4. 保存 **robot\_pose.json** 文件。


## 17.3 获取标定姿态

### 17.3.1 流程图



### 17.3.2 步骤

进行手眼标定前，需获取至少 15 组的标定位姿。获取标定位姿的步骤如下：

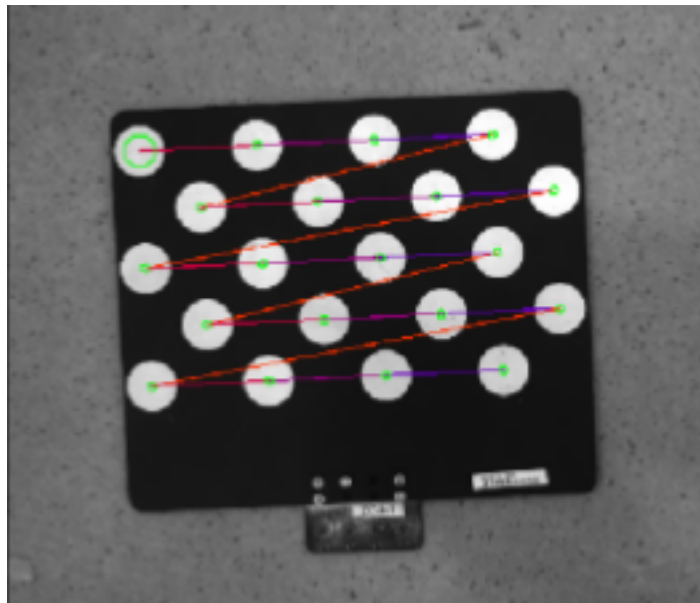
1. 单击工具栏中的  按钮或单击键盘上的 F5，运行例程。例程运行到 `read_char (WindowHandle, Char, ReCode)` 时将暂停运行，需手动输入指令继续运行例程。

---

提示：如相机连接不成功，需检查相机是否已被 Mech-Eye Viewer 或其他 GenICam 客户端连接。

---

2. 使用示教器移动机器人至合适位置。
3. 输入 P，相机采集图像。
  - 如标定板只有部分在视野之内或不在视野之内，请移动机器人，并重新采集图像。
  - 如标定板全部在视野之内，请执行下一步。
4. 输入 T，相机采集包含特征识别结果的图像。
  - 如相机无法识别标定板上的圆，HALCON 将不返回图像。此时请移动机器人，重新采集原始图像与包含特征识别结果的图像。
  - 如相机可以识别标定板上的圆，HALCON 将显示包含特征识别结果的图像（如下图所示），请执行下一步。



5. 查看机器人示教器，将机器人此时的位姿输入 `robot_pose.json` 文件中。

---

提示：

- 请将该位姿存入示教器中。进行手眼标定时，可直接使用示教器中已保存的位姿移动机器人。
-

6. 重复步骤 2~5，继续获取位姿。
7. 确认获取至少 15 组位姿后，采集数据结束，输入 Q，退出程序。
8. 设置 `robot_pose.json` 文件中标定位姿个数。打开 `robot_pose.json` 文件，定位至如下命令行，将 15 替换为实际获取的标定位姿个数。

```
"pose_count":15
```

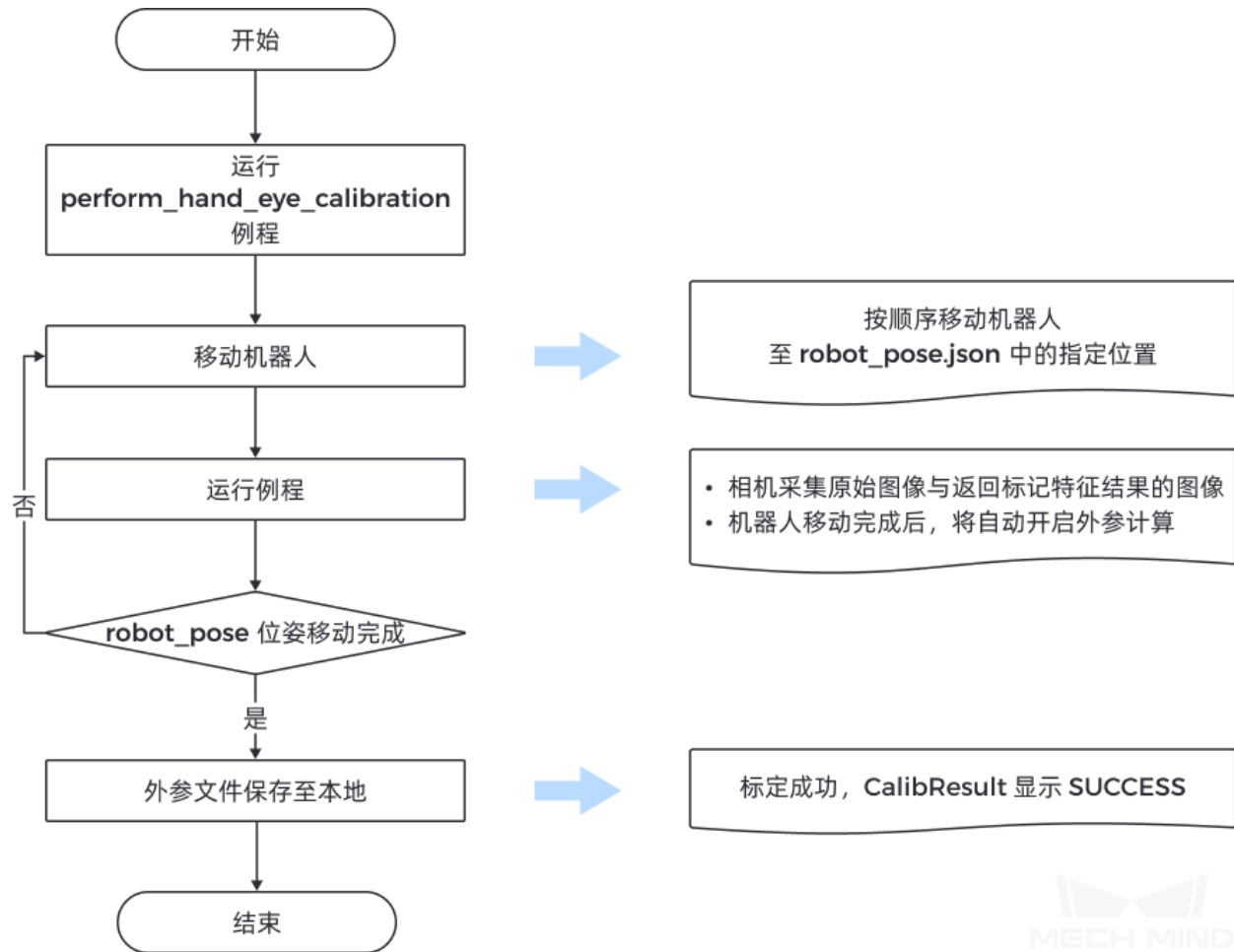
## 17.4 进行手眼标定

获取机器人标定位姿后，运行 `perform_hand_eye_calibration.hdev` 例程，开始手眼标定。

- 流程图
- 检查相机安装方式
- 切换坐标系
- 步骤



### 17.4.1 流程图



### 17.4.2 检查相机安装方式

进行手眼标定前需设置相机安装方式。

默认相机安装方式为 Eye in Hand; 如相机安装方式为 Eye to Hand, 定位至如下算子, 将 **EyeInHand** 替换为 **EyeToHand**。

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

### 17.4.3 切换坐标系


例程中包含切换相机输出点云的坐标系的算子，通过该算子可直接输出机器人坐标系下的点云。

默认设置为不切换坐标系；如需切换至机器人坐标系，请在 `captureTranformedPointCloud` 函数中，定位至如下算子，并将 `false` 替换为 `true`。

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

### 17.4.4 步骤


手眼标定需执行如下步骤：

1. 单击工具栏中的  按钮或单击键盘上的 F5，运行例程。例程运行至 `stop` 行后，将暂停运行。
2. 将机器人移动至 `robot_pose.json` 文件中的标定位姿。

---

**提示：**请按 `robot_pose.json` 文件中的位姿顺序移动机器人，否则将导致后续外参计算失败。

---

3. 单击工具栏中的  按钮或单击键盘上的 F5，运行例程，相机采集图像。
4. 采集图像结束后，查看 **控制变量** 中的 `CollectResult` 值。
  - 如显示 `SUCCESS`，请进行下一步；
  - 如发生报错，请根据 [错误码](#) 排查问题，再重新获取标定位姿。
5. 界面提示“Move the robot to the next calibration pose”，此时请重复执行步骤 2~3。

---

**提示：**机器人到达 `robot_pose.json` 文件中的所有位置后，再运行例程时，将自动计算外参。

---

6. 查看 **控制变量** 中的 `CalibResult` 值。
  - 如显示 `SUCCESS`，则手眼标定成功。请在例程所在文件夹内查看外参文件 `Extrinsics.txt` 与转换坐标系后的点云；
  - 如发生报错，请根据 [错误码](#) 排查问题，再重新获取标定位姿。

## 17.5 机器人欧拉角类型

标定例程中已支持以下欧拉角类型与四元数的转换。

输入顺序	欧拉角类型	机器人品牌
Z-Y' -Z''	rzyz	Kawasaki (川崎)
Z-Y' -X''	rzyx	ABB KUKA (库卡)
X-Y-Z	sxyz	FANUC (发那科) YASKAWA (安川) Rokae (珞石) UR (优傲)
X-Y' -Z''	rxyz	/
Z-X' -Z''	rzxz	/

**注解:**

- 欧拉角类型相同时，机器人欧拉角显示的顺序可能不同。请按照上表的输入顺序输入欧拉角。
- 如所用机器人的欧拉角类型不在上表内，需要您自行添加该欧拉角类型与四元数的转换。请在 `perform_hand_eye_calibration.hdev` 例程的 `euler_to_quad` 函数中，参照已有代码添加转换。

## 17.6 标定参数说明

本节介绍进行手眼标定时使用的参数。

- *BoardType*
- *ExtrinErrCode*

### 17.6.1 BoardType

该参数用于设置使用的标定板型号。

值列表及说明:

值	说明
BDB-5	标定板到相机的推荐距离为 < 0.6m
BDB-6	标定板到相机的推荐距离为 0.6~1.5m
BDB-7	标定板到相机的推荐距离为 > 1.5m
OCB-005	仅用于对精度要求高，且为 Eye to Hand 安装方式的项目
OCB-010	
OCB-015	
OCB-020	
CGB-020	标定板到相机的推荐距离为 < 0.6m
CGB-035	标定板到相机的推荐距离为 0.6~1.5m
CGB-050	标定板到相机的推荐距离为 > 1.5m

## 17.6.2 ExtrinErrCode

该只读参数用于确认手眼标定过程中的状态码和错误码。


状态码	说明
SUCCESS	执行成功。
POSE_INVALID	位姿格式不正确，请输入四元数。
IMAGE2D_EMPTY	2D 图无效。
FIND_CORNERS_FAIL	2D 图特征识别失败，请调节 <a href="#">2D 图参数调节</a> 以获取符合要求的 2D 图。
DEPTH_EMPTY	深度图无效。
CORNERS_3D_INVALID	深度图特征识别失败，请调节 <a href="#">3D 参数</a> 以获取符合要求的深度图。
POSES_INSUFFICIENT	位姿数量不足，请输入至少 15 组位姿。

本章介绍如何在 Windows 系统下通过梅卡曼德提供的例程获取与修改当前相机 IP 地址、子网掩码与网关。

- 获取例程并选择相机
- 获取相机的 IP 地址、子网掩码与网关
- 设置静态 IP 地址
- 设置动态分配 IP 地址

## 18.1 获取例程并选择相机

在获取或修改 IP 地址前，需先下载例程，并选择相机。

1. 下载 `configure_camera_ip_address` 例程，依次单击 *Code* ▶ *Download ZIP* 即可下载。
2. 在 HALCON 中打开例程：运行 HALCON，将例程拖拽至 HALCON 的 **程序窗口** 中。
3. 选中 **程序窗口** 中 `info_framegrabber` 算子所在的行，单击工具栏中的  按钮或单击键盘上的 F6，运行该算子。
4. 双击 **控制变量** 区域的 **DeviceInfos**，可查看所有可连接的相机。

控制变量

Info	'Info about GigE Vision devices in the system.'
DeviceInfos	[ '   device:Halcon测试报\351   unique_name:48b02d6244eb_MechMindRoboticsLtd_Me...

5. 选中相机并双击，复制 **unique\_name:** 或 **user\_name:** 的相机名。


```
变量监视: DeviceInfos
DeviceInfos
0 | ' | device:P022206030802H | unique_name:dca5012fa8cf MindVision_MGE134GM2 | user_name:P022206030802H |
  | interface:Esen_ITF_b07b2528849ac0a81425ffffff00 | producer:Esen'
```

6. 定位至如下命令行，将 **MechEye** 替换为 **unique\_name** 或 **user\_name**。

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

## 18.2 获取相机的 IP 地址、子网掩码与网关

请执行以下步骤，获取相机当前的 IP 地址、子网掩码与网关：

单击工具栏中的  按钮或单击键盘上的 F5，运行例程。然后，可在 **控制变量** 中查看相机当前的 IP 地址、子网掩码与网关。

- CurrentIPAddressString: IP 地址
- CurrentSubnetMaskString: 子网掩码
- CurrentDefaultGatewayString: 网关


**注解：**如相机固件版本为 2.0.2 或以下，将无法获取以上信息，同时报错信息将记录在 Exception 变量中。

## 18.3 设置静态 IP 地址

本例程默认使用静态 IP 地址，执行以下步骤即可修改相机的 IP 地址、子网掩码与网关：

1. 定位至如下命令行，将 IP 地址、子网掩码与网关替换为需设置的值：

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


2. 单击工具栏中的  按钮或单击键盘上的 F5，运行例程。
3. 重启相机，IP 地址修改成功。

## 18.4 设置动态分配 IP 地址

使用动态分配 IP 地址，需执行以下步骤：

1. 定位至如下命令行，将 **true** 替换为 **false**。

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

2. 单击工具栏中的  按钮或单击键盘上的 F5，运行例程。
3. 重启相机，IP 地址修改成功。

---

**注解：**相机与工控机直接连接时，不可设置为动态分配 IP 地址。


---

本章介绍如何在 Windows 系统下通过梅卡曼德提供的例程获取仅包含物体深度信息的深度图，可加快 HALCON 的传输速度，提升效率。

- 获取深度图

## 19.1 获取深度图

请执行以下步骤获取深度图：

1. 下载 `obtain_depth_map` 例程，依次单击 *Code* ▶ *Download ZIP* 即可下载。
2. 在 HALCON 中打开例程：运行 HALCON，将例程拖拽至 HALCON 的 程序窗口中。
3. 选中 程序窗口中 `info_framegrabber` 算子所在的行，单击工具栏中的  按钮或单击键盘上的 F6，运行该算子。
4. 双击 控制变量区域的 `DeviceInfos`，可查看所有可连接的相机。

控制变量

Info	'Info about GigE Vision devices in the system.'
DeviceInfos	['   device:Halcon测试报\351   unique_name:48b02d6244eb_MechMindRoboticsLtd_Me...


5. 选中相机并双击，复制 `unique_name:` 或 `user_name:` 的相机名。



```
变量监视: DeviceInfos
DeviceInfos
0 | ' | device:P022206030802H | unique_name:dca5012fa8cf_MindVision_MGE134GM2 | user_name:P022206030802H |
  | interface:Esen_ITF_b07b2528849ac0a81425ffffff00 | producer:Esen'
```

6. 定位至如下命令行，将 **MechEye** 替换为 **unique\_name** 或 **user\_name**。

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

7. 单击工具栏中的  按钮或单击键盘上的 F5，运行例程。获取的深度图保存在 DepthInM 变量中。

本章介绍如何在 Windows 系统下通过梅卡曼德提供的例程获取纹理点云。

- 数据类型说明
- 获取纹理点云

### 20.1 数据类型说明

通过该例程可获取 Range 与 Intensity 两个通道中的数据，具体介绍如下。


- Range: 包含 X、Y、Z 信息的 2D 图，不可禁用。
- Intensity: 用于为点云添加纹理的彩色 2D 图或黑白 2D 图。

Range 和 Intensity 通道中的图像，像素一一对齐，可直接用于生成纹理点云。

**注意：**设置 Range 或 Intensity 两通道图像的采集区域（Width、Height、OffsetX 和 OffsetY 参数）时，如参数不一致，将导致两通道中的图像像素不对齐，无法生成纹理点云。

## 20.2 获取纹理点云

请执行以下步骤获取纹理点云：

1. 下载 `obtain_textured_point_cloud` 例程，依次单击 *Code* ▶ *Download ZIP* 即可下载。
2. 在 HALCON 中打开例程：运行 HALCON，将例程拖拽至 HALCON 的 **程序窗口** 中。
3. 选中 **程序窗口** 中 `info_framegrabber` 算子所在的行，单击工具栏中的  按钮或单击键盘上的 F6，运行该算子。
4. 双击 **控制变量** 区域的 `DeviceInfos`，可查看所有可连接的相机。

控制变量


Info	'Info about GigE Vision devices in the system.'
DeviceInfos	[ '   device:Halcon测试报\351   unique_name:48b02d6244eb_MechMindRoboticsLtd_Me...

5. 选中相机并双击，复制 `unique_name:` 或 `user_name:` 的相机名。

变量监视: DeviceInfos	
	DeviceInfos
0	'   device:P022206030802H   unique_name:dca5012fa8cf_MindVision_MGE134GM2   user_name:P022206030802H   interface:Esen_ITF_b07b2528849ac0a81425ffffff00   producer:Esen'

6. 定位至如下命令行，将 `MechEye` 替换为 `unique_name` 或 `user_name`。

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

7. 单击工具栏中的  按钮或单击键盘上的 F5，运行例程。HALCON 中将显示获取的纹理点云，纹理点云储存在 `ObjectModel3D` 变量中。

**注解：**LSR (V4) 与 DEEP (V4) 系列默认使用彩色 2D 图生成纹理点云。如需使用黑白 2D 图，请定位至如下命令行，并取消该行注释。

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

---

本手册提供使用 GenICam 客户端控制相机所需要参考的信息。

---

查看以下内容，了解 **GenICam** 支持的相机参数及其解释。

*GenICam 支持的相机参数*

---

查看以下内容，了解 借助 **Mech-Eye Viewer** 为 **GenICam** 客户端设置相机参数。

*借助 Mech-Eye Viewer 为 GenICam 客户端设置相机参数*

---

查看以下内容，了解 如何通过 **Mech-Eye API** 获取 **HALCON** 可读取的点云。

*通过 Mech-Eye API 获取 HALCON 可读取的点云*

---

## 21.1 GenICam 支持的相机参数

本文介绍相机为 GenICam 提供的相机参数，指明了相机提供的特性和功能。

- 支持的相机参数
- *GenICam* 与 *Mech-Eye Viewer* 支持的相机参数的对应关系

### 21.1.1 支持的相机参数

关于 GenICam 支持的相机参数的描述和解释，请下载和查阅 [GenICam 参数说明](#)。

### 21.1.2 GenICam 与 Mech-Eye Viewer 支持的相机参数的对应关系



GenICam 支持的相机参数与 Mech-Eye Viewer 提供的参数基本上一一对应。Mech-Eye Viewer 中提供参数描述、参数值等更多信息，可以使用该信息协助您在 GenICam 客户端中调节参数。

GenICam 与 Mech-Eye Viewer 支持的相机参数的对应关系见下表：

GenICam 支持的参数	Mech-Eye Viewer 参数	可见级别	备注
Scan2DROIHeight	2D 参数 - 自动曝光感兴趣区域	初级	
Scan2DROILeft			
Scan2DROITop			
Scan2DROIWidth			
Scan2DExposureMode	2D 参数 - 曝光模式 - 2D 图 (纹理) 曝光模式	初级	
Scan2DExposureTime	2D 参数 - 曝光时间	初级	
Scan2DExpectedGrayValue	2D 参数 - 灰度值	初级	
Scan2DHDRExposureSequence	2D 参数 - 曝光时间序列	初级	
Scan2DToneMappingEnable	2D 参数 - 色调映射	初级	
Scan2DSharpenFactor	2D 参数 - 锐化因子	专家	
Scan2DPatternRoleExposureMode	2D 参数 - 2D 图 (深度源) 曝光模式	初级	LSR L (V4) DEEP (V4)
Scan2DPatternRoleExposureTime	2D 参数 - 2D 图 (深度源) 曝光时间	初级	
Scan3DROIHeight	感兴趣区域设置	初级	
Scan3DROILeft			
Scan3DROITop			
Scan3DROIWidth			
ProjectorLightColor	3D 参数 - 投影—投影光颜色	大师	Pro M Enhanced (V3) Pro S Enhanced (V3) Log M (V3) Log S (V3)
ProjectorPowerLevel	3D 参数 - 投影—投影光亮度	专家	DLP 相机
ProjectorFringeCodingMode	3D 参数 - 投影—编码模式	专家	Nano (V3) Pro XS (V3) PRO M and PRO S (V4)
AntiFlickerMode	3D 参数 - 投影—抗频闪模式	专家	PRO M and PRO S (V4) NANO (V4)
ProjectorSelectionMode	3D 参数 - 投影—投影机模式	大师	Deep (V3) Pro L Enhanced (V3)
ProjectorSelector	3D 参数 - 投影—选择投影机	大师	

下页继续

表 1 - 续上页

LaserFringeCodingMode	3D 参数 - 激光—编码模式	专家	Laser L (V3) Laser L Enhanced (V3) LSR L (V4) DEEP (V4)
LaserPowerLevel	3D 参数 - 激光—激光强度	专家	
LaserFrameAmplitude	3D 参数 - 激光投影画幅控制—激光投影范围幅值	大师	
LaserFrameOffset	3D 参数 - 激光投影画幅控制—激光投影范围偏移值	大师	
LaserFramePartitionCount	3D 参数 - 激光投影画幅控制—激光投影分块数	大师	
Scan3DBinningEnable	3D 参数 - 3D 降采样	大师	Pro L Enhanced (V3) Laser L Enhanced (V3)
Scan3DExposureCount	3D 参数 - 曝光次数	初级	
Scan3DExposureTime	3D 参数 - 曝光时间	初级	
Scan3DExposureTime2	3D 参数 - 曝光时间 2	初级	
Scan3DExposureTime3	3D 参数 - 曝光时间 3	初级	
Scan3DGain	3D 参数 - 相机增益	专家	
FringeCodingMode	3D 参数 - UHP —编码模式	大师	UHP-140
UhpCaptureMode	3D 参数 - UHP —拍摄模式	初级	UHP-140
AcquisitionMode	 and 	-	
DepthLowerLimit	深度范围	初级	
DepthUpperLimit			
DeviceScanType	-	-	
PointCloudSurfaceSmoothing	点云后处理 - 表面平滑	初级	
PointCloudOutlierRemoval	点云后处理 - 离群点去除	初级	
PointCloudNoiseRemoval	点云后处理 - 噪点去除	专家	
PointCloudEdgePreservation	点云后处理 - 边缘保持	大师	
FringeContrastThreshold	点云后处理 - 条纹对比度阈值	初级	
FringeMinThreshold	点云后处理 - 投影亮度最小阈值	大师	
UserSetDefault	-	-	
UserSetSelector	参数组下拉选项	-	

## 21.2 借助 Mech-Eye Viewer 为 GenICam 客户端设置相机参数

设置自动曝光感兴趣区域、深度设置及感兴趣区域等相机参数时，通常需要借助可视化工具进行不断调整从而获取理想的值。由于兼容 GenICam 标准的第三方机器视觉软件（GenICam 客户端）并未提供辅助参数设置的可视化工具，您可以借助 Mech-Eye Viewer 的可视化设置工具为 GenICam 客户端设置这些参数。

通过 Mech-Eye Viewer 设置这些参数后，您可以直接在 GenICam 客户端中读取最新的相机配置。

- 准备工作
- 调节参数
- 同步配置修改

### 21.2.1 准备工作

在使用 Mech-Eye Viewer 进行可视化调节前，需完成以下步骤：

1. 保存 GenICam 客户端修改的参数。
2. 断开相机与 GenICam 客户端的连接。
3. 使用 *Mech-Eye Viewer* 连接相机。
4. 在 Mech-Eye Viewer 的参数区选择参数组。

**注解：** GenICam 客户端只能读取参数组，不能增加/删除。如需使用增加/删除参数组，请在 *Mech-Eye Viewer* 中配置参数组。

### 21.2.2 调节参数

查看以下内容，分别设置自动曝光感兴趣区域、深度设置及感兴趣区域。

#### 设置自动曝光感兴趣区域

- 设置步骤：
  1. 在 Mech-Eye Viewer 的参数栏中，选择 **2D** 参数。
  2. 在 **曝光模式** 的下拉菜单中，选择 **自动曝光**。
  3. 双击 **自动曝光感兴趣区域** 栏中的 **编辑按钮**，进入 **设置感兴趣区域** 页面。
  4. 设置感兴趣区域，详细信息请参考 [自动曝光感兴趣区域](#)。

- GenICam 与 Mech-Eye Viewer 的自动曝光感兴趣区域参数对应关系如下:

GenICam 参数	Mech-Eye Viewer 参数
Scan2DROILeft	左上角坐标 - x
Scan2DROITop	左上角坐标 - y
Scan2DROIHeight	尺寸 - 高
Scan2DROIWidth	尺寸 - 宽

### 设置深度范围

- 设置步骤:
  1. 在 Mech-Eye Viewer 的参数栏中, 选择 深度范围。
  2. 双击 深度范围栏中的 编辑按钮, 进入 设置深度范围页面。
  3. 设置深度范围, 详细信息请参考[深度范围](#)。
- GenICam 与 Mech-Eye Viewer 中的 深度范围参数对应关系如下:

GenICam 参数	Mech-Eye Viewer 参数
DepthLowerLimit	深度范围 - 下限
DepthUpperLimit	深度范围 - 上限

### 设置感兴趣区域

- 设置步骤:
  1. 在 Mech-Eye Viewer 的参数栏中, 选择 感兴趣区域。
  2. 双击 感兴趣区域栏中的 编辑按钮, 设置感兴趣区域页面。
  3. 设置感兴趣区域, 详细信息请参考[感兴趣区域设置](#)。
- GenICam 与 Mech-Eye Viewer 中的 感兴趣区域参数对应关系如下:

GenICam 参数	Mech-Eye Viewer 参数
Scan3DROILeft	左上角坐标 - x
Scan3DROITop	左上角坐标 - y
Scan3DROIHeight	尺寸 - 高
Scan3DROIWidth	尺寸 - 宽



### 21.2.3 同步配置修改

在 Mech-Eye Viewer 中设置完 自动曝光感兴趣区域、深度设置及 感兴趣区域后，执行如下步骤使 GenICam 客户端读取参数设置。

1. 在 Mech-Eye Viewer 中，单击参数组中的 保存或者使用快捷键 **Ctrl+S**，保存配置修改至参数组。
2. 断开相机与 Mech-Eye Viewer 的连接。
3. 连接相机至 GenICam 客户端。
4. 在 GenICam 客户端中，选择并读取参数组配置。

## 21.3 通过 Mech-Eye API 获取 HALCON 可读取的点云

通过 Mech-Eye API 从相机获取点云，获取速度比 HALCON 更快，且能够直接获取彩色点云。

Mech-Eye API 提供了相关的 C++ 例程 `CaptureHalconPointCloud`。运行该例程，即可通过 Mech-Eye API 获取白色点云和彩色点云，并转换成 HALCON 可读取的格式。只需在 HALCON 中读取这些点云，即可对其进行进一步处理。

该例程可以在 Windows 及 Ubuntu 系统中使用。

---

**注解：**该例程依赖 HALCON 的 C++ 接口。使用前请确认 HALCON 的证书是否在有效期内。

---

### 21.3.1 在 Windows 系统中使用例程

在 Windows 系统中使用该例程，需先安装以下软件：

- Mech-Eye SDK
- Cmake
- Visual Studio
- HALCON

软件安装的注意事项，以及单独构建并运行该例程的操作指南，请参阅 *Mech-Eye API C++ 例程使用指南 (Windows)*。

运行例程后，使用 HALCON 的 `read_object_model_3d` 算子即可读取获取的点云。

### 21.3.2 在 Ubuntu 系统中使用例程

在 Ubuntu 系统中使用该例程，需先安装以下软件：

- Mech-Eye SDK
- Cmake
- PCL
- HALCON

软件安装的注意事项，以及单独构建并运行该例程的操作指南，请参阅 *Mech-Eye API C++ 例程使用指南 (Ubuntu)*。

运行例程后，使用 HALCON 的 `read_object_model_3d` 算子即可读取获取的点云。

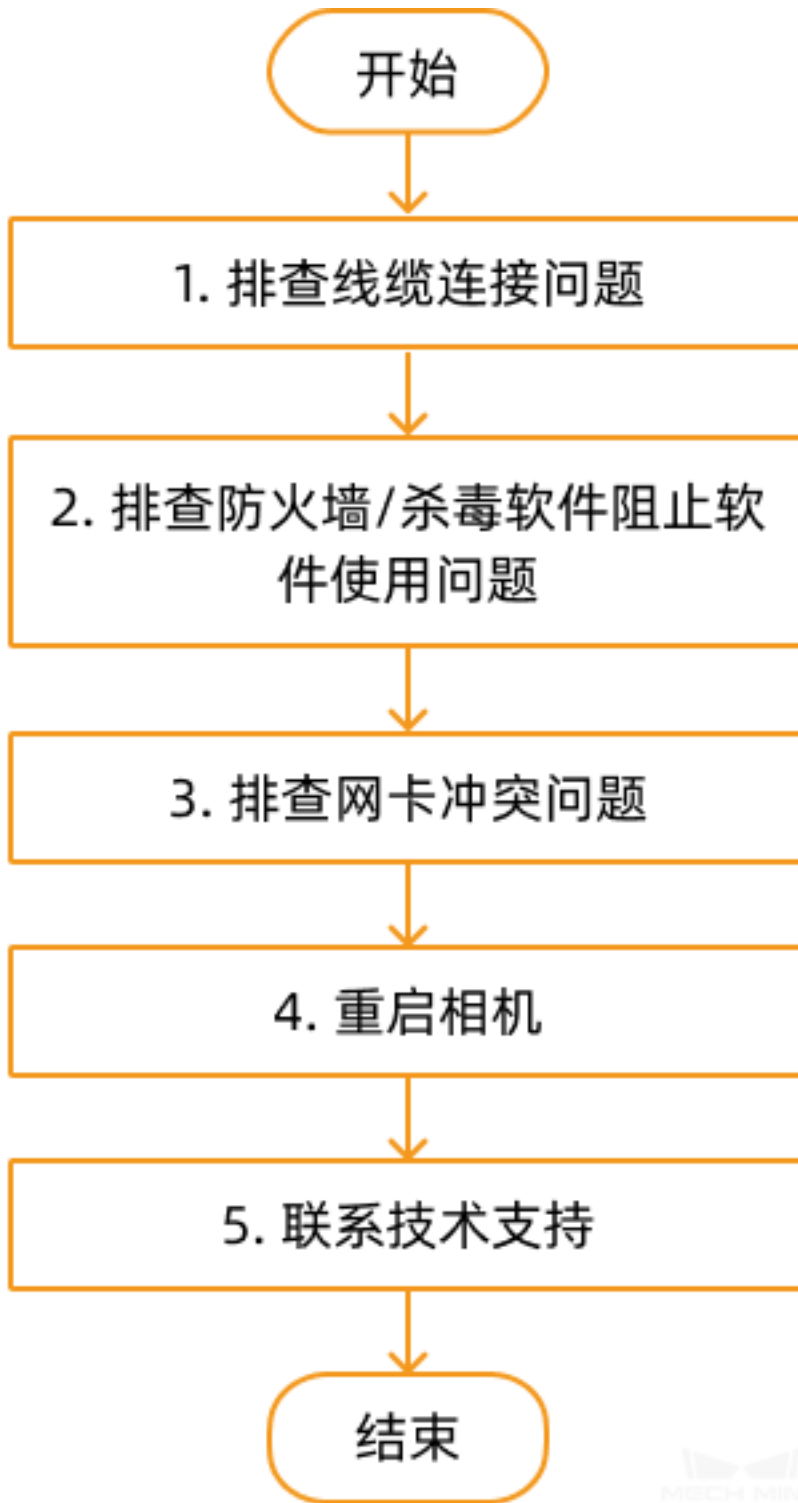
### 22.1 Mech-Eye SDK 搜索不到相机

使用 Mech-Eye SDK 连接相机时，有时会无法搜索到相机。本文将介绍问题的可能原因，排查问题的思路及排查方法。

#### 22.1.1 问题原因

- 相机、工控机、路由器/交换机线缆连接异常
- 防火墙/杀毒软件阻止软件使用
- 网卡冲突
- 其他

### 22.1.2 排查思路



Mech-Eye SDK 搜索不到相机时的排查思路为：

1. 首先排查线缆连接问题。确认线缆连接无误后，请使用 Mech-Eye SDK 搜索相机，如无法搜索到相机，请查看下一步。
2. 排查防火墙/杀毒软件阻止软件使用问题。解决阻止软件使用问题后，如仍无法搜索到相机，请查看下一步。
3. 排查网卡冲突问题。
4. 如解决网卡冲突后仍无法搜索到相机，请重启相机。
5. 如重启相机后仍无法搜索到相机，请联系技术支持。

### 22.1.3 排查方法

#### 排查线缆连接问题

排查步骤	线缆	连接正常说明
检查相机的线缆连接	电源线	PWR 指示灯绿色常亮
	网线	V3 相机：千兆网时网口指示灯为绿色；百兆网时网口指示灯为黄色。
		V4 相机：LINK 指示灯说明详见功能示意图。
检查工控机的线缆连接	网线	网口指示灯常亮
检查路由器/交换机端的线缆连接	网线	连接网口为 LAN 网口，且 LAN 网口指示灯常亮

提示：

- 相机与工控机直接连接时，不需要检查路由器/交换机的连接。
- 如相机网口指示灯先闪烁后熄灭，且该现象反复发生，请检查网线与相机、工控机的连接。通常，该现象是网线连接不良导致的。

#### 排查防火墙/杀毒软件阻止软件使用问题

排查方法：

1. 关闭工控机防火墙或将 Mech-Eye Viewer 加入工控机防火墙白名单。
  - 关闭工控机防火墙：
    - 选择 控制面板 ▶ 系统和安全 ▶ Windows Defender 防火墙 ▶ 启用或关闭 Windows Defender 防火墙 ▶ 关闭 Windows Defender 防火墙 ▶ 确定，关闭防火墙。
  - 将 Mech-Eye Viewer 加入工控机防火墙白名单：
    - (1) 选择 控制面板 ▶ 系统和安全 ▶ Windows Defender 防火墙 ▶ 允许应用或功能通过 Windows Defender 防火墙 ▶ 更改设置 (允许其他应用)，进入 允许应用通过 Windows Defender 防火墙进行通信 页面。

- (2) 选择 更改设置 ▶ 允许其他成员应用，进入 添加应用页面。
- (3) 单击 浏览，定位至安装路径，选择 Mech-Eye Viewer.exe，单击 应用完成。

2. 关闭工控机杀毒软件以及 12345、360 等安全防护软件。

## 排查网卡冲突问题

排查步骤：

1. 在工控机上，选择 控制面板 ▶ 网络和 Internet ▶ 查看网络状态和任务 ▶ 更改适配器设置进入 网络连接页面。
2. 禁用连接相机网卡外的其它网卡。  
选中网卡，右键单击然后选择 禁用即可禁用该网卡。
3. 将工控机与相机 IP 地址设置在同一网段，请参考设置 IP 地址。
4. 使用 Mech-Eye SDK 搜索相机。
5. 成功搜索到相机，则为网卡冲突。此时可启用其他网卡，并将其设置到不同网段。

---

**提示：** 如无法连接相机，请尝试重启相机网卡；如重启网卡后仍无法连接，请更换连接相机的网口。

---

## 重启相机

如以上步骤全部尝试后，仍无法搜索到相机，请尝试重新启动相机。

相机重启步骤：

1. 拔出电源线插头，相机断电。
2. 等待 20 秒左右，重新插入电源线插头，相机通电。

## 联系技术支持

如以上步骤全部尝试后，仍无法搜索到相机，请联系技术支持。

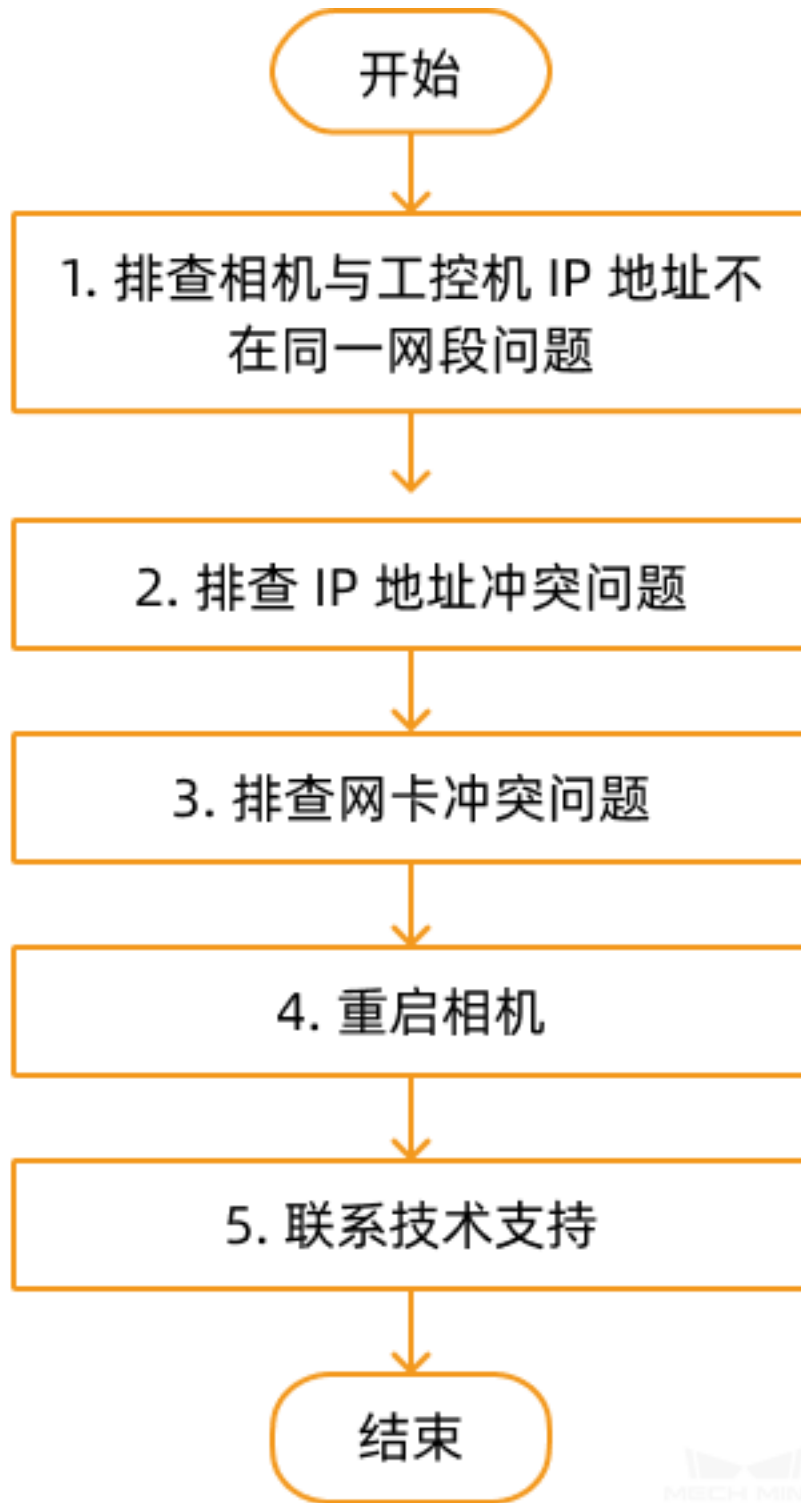
## 22.2 Mech-Eye SDK 无法连接相机 (Unreachable)

使用 Mech-Eye SDK 连接相机时，无法连接搜索到的相机，并且 Mech-Eye Viewer 的相机信息栏中相机的状态显示为 Unreachable。本章将介绍问题的可能原因，排查问题的思路及排查方法。

### 22.2.1 问题原因

- 相机与工控机 IP 地址不在同一网段
- IP 地址冲突
- 网卡冲突
- 其他

## 22.2.2 排查思路





Mech-Eye SDK 搜索到但无法连接相机时的排查思路为：

1. 首先排查相机与工控机 IP 地址不在同一网段问题。确认 IP 地址在同一网段后，请使用 Mech-Eye SDK 连接相机，如无法连接到相机，请查看下一步。
2. 排查 IP 地址冲突问题。如 IP 地址冲突解决后，如仍无法连接到相机，请查看下一步。
3. 排查网卡冲突问题。
4. 如解决网卡冲突后仍无法连接到相机，请重启相机。
5. 如重启相机后仍无法连接到相机，请联系技术支持。

## 22.2.3 排查方法

### 排查相机与工控机 IP 地址不在同一网段问题

排查步骤：

1. 查看工控机 IP 地址，请参考设置 IP 地址。
2. 查看相机 IP 地址，请参考设置 IP 地址。
3. 确保相机与工控机 IP 地址在同一网段。如果不在同一网段，请设置相机 IP 地址，使相机 IP 地址与工控机在同一网段。
4. 运行 Mech-Eye SDK，重新扫描并连接相机。

### 排查 IP 地址冲突问题

工控机与路由器连接的设备中，存在 IP 地址冲突。IP 地址冲突即两个或多个设备拥有相同的 IP 地址。

排查步骤：

1. 在工控机上，仅保留与相机连接的网线，拔掉工控机和路由器/交换机的其他网线。
2. 运行 Mech-Eye SDK，重新扫描并连接相机。
3. 如成功连接到相机，则为 IP 地址冲突。需逐一排查其他设备的 IP 地址，并更改导致冲突设备的 IP 地址；或重新设置相机 IP 地址，使其唯一。

### 排查网卡冲突问题

排查步骤：

1. 禁用工控机中连接相机网卡外的其它网卡。  
选择 控制面板 ▶ 网络和 Internet ▶ 查看网络状态和任务 ▶ 更改适配器设置进入 网络连接 页面。  
选中网卡，右键单击，选择 禁用以禁用该网卡。
2. 运行 Mech-Eye SDK，重新连接相机。
3. 如成功连接到相机，则为网卡冲突。启用其他网卡，并将其设置到不同网段。

---

提示：如无法连接相机，请尝试重启相机网卡；如重启网卡后仍无法连接，请更换连接相机的网口。

---

## 重启相机

如以上步骤全部尝试后，仍无法连接相机，请尝试重新启动相机。

相机重启步骤：

1. 拔出电源线插头，相机断电。
2. 等待 20 秒左右，重新插入电源线插头，相机通电。

## 联系技术支持

如以上步骤全部尝试后，仍无法连接相机，请联系技术支持。

## 22.3 Mech-Eye SDK 无法连接相机 (Windows Server)

使用 Mech-Eye SDK 连接相机时，无法连接搜索到的相机，并且 Mech-Eye Viewer 的相机信息栏中相机的状态显示为 Windows Server。本章将介绍问题的可能原因，排查问题的思路及排查方法。

### 22.3.1 问题原因

软件缺陷。该问题存在于 Mech-Eye SDK 1.6.1 及之前版本的软件。

### 22.3.2 解决方案

出现该问题时，建议将 Mech-Eye SDK 软件升级到最新版本。请参考升级 *Mech-Eye SDK*。

如果 Mech-Eye SDK 软件不能升级到最新版本，请联系技术支持。

## 22.4 启用巨型帧 (GenICam) 功能

以太网帧默认为 64~1518 bytes，大于 1518 bytes 的数据帧称为巨型帧。使用巨型帧可以降低 CPU 计算量、加快数据传输。

使用 GenICam 的第三方机器视觉软件与相机连接采集图像时，如采图时间相对较长，工控机启用巨型帧功能可解决该问题。

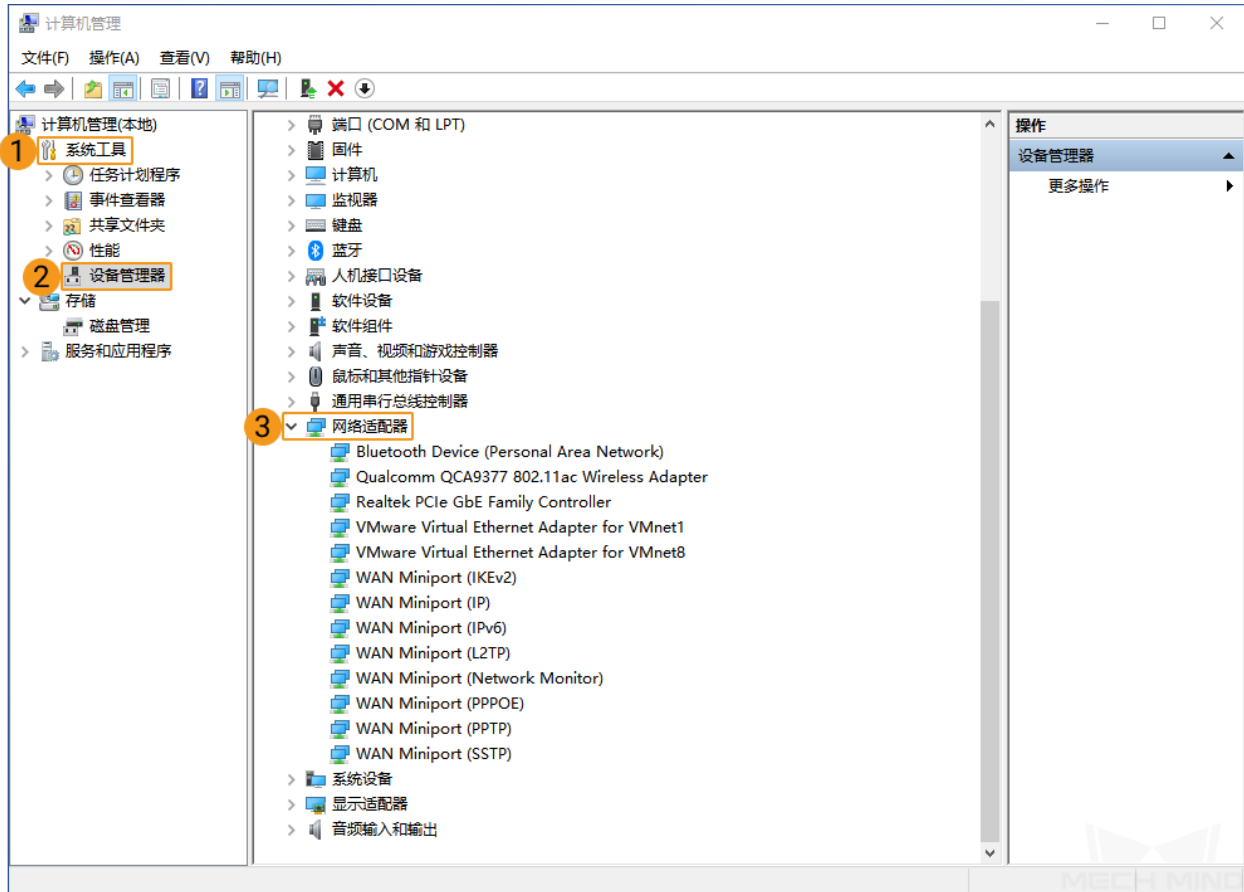
---

**注解：**工控机巨型帧的值需与相机 MTU 值保持一致。如二者不一致，可能影响数据传输。当相机与工控机使用交换机连接时，不受该规则限制。相机 MTU 在 **管理员**模式下可见，如需使用，请咨询技术支持。

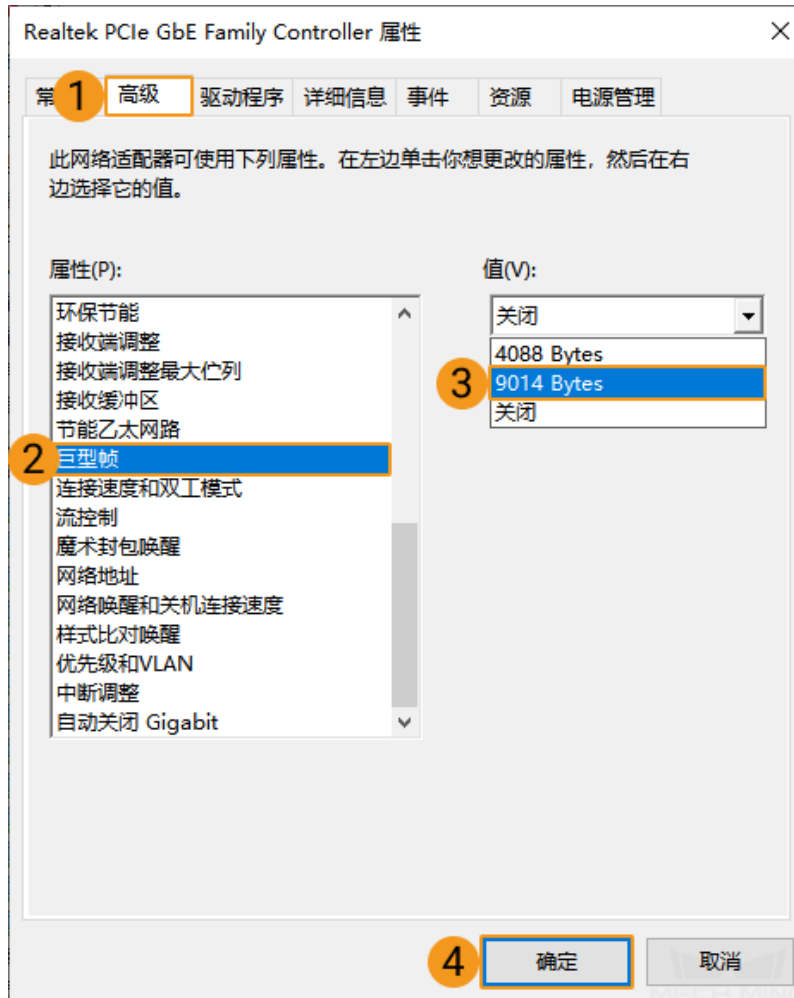
---

## 22.4.1 设置步骤

1. 右键单击工控机中的 **此电脑**，选择 **管理**，进入计算机管理页面。
2. 选择 **系统工具** ▶ **设备管理器** ▶ **网络适配器**。



3. 右键单击与相机连接的网卡，选择 **属性**。
4. 切换至 **高级选项卡**，在 **属性** 栏中下拉滚动条，选择 **巨型帧**，并将值设置为 **9014 Bytes**。单击 **确定修改** 完成。



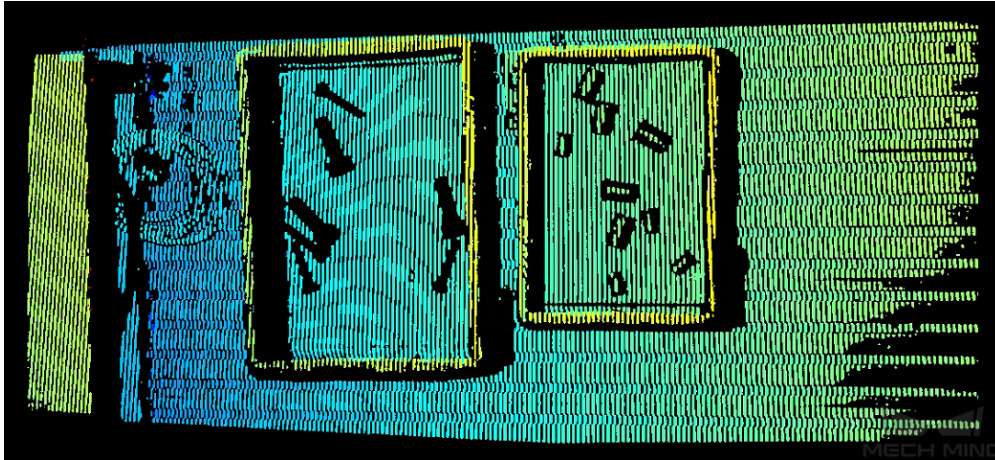
## 22.5 特殊相机型号更新

适用于 V2 版本的 Pro S Enhanced 与 Pro M Enhanced。

V2 版本的 Pro S Enhanced 与 Pro M Enhanced 连接 Mech-Eye Viewer 时，显示型号为 Log S 与 Log M。升级固件后，如未更新相机型号，则会导致相机采图失败。

## 22.5.1 问题描述

V2 版本的 Pro S Enhanced 与 Pro M Enhanced，Mech-Eye Viewer 显示型号为 Log S 与 Log M。如直接升级固件，Mech-Eye Viewer 仍显示为 Log S 与 Log M。此时，相机型号不一致导致点云缺失，如下图所示。



## 22.5.2 处理步骤

由于相机型号不一致会导致点云缺失，因此需先确认相机型号，再升级固件。

- 相机型号为 Log S 或 Log M 时，直接升级固件即可。
- 相机型号为 Pro S Enhanced 或 Pro M Enhanced 时，需修改相机型号后再升级固件。

**提示：** 相机固件升级时，需确保相机与工控机在同一网段。否则会卡在升级界面，不能完成相机固件升级。

### 确认相机型号

确认相机型号有两种方法：通过分辨率确认或查看相机铭牌。

通过分辨率：

如何查看分辨率：

使用 Mech-Eye Viewer 连接相机。选择 工具 ▶ 查看 2D 相机并设置参数进入 查看 2D 相机并设置参数页面，可查看分辨率。

Mech-Eye Viewer 中显示的 Log 系列包含 Log 系列与 Pro Enhanced (V2) 系列，通过分辨率可识别型号。型号对应关系如下：

Mech-Eye Viewer 显示型号，分辨率	实际型号
Log S, 1280 × 1024	Log S
Log S, 1920 × 1200	Pro S Enhanced (V2)
Log M, 1280 × 1024	Log M
Log M, 1920 × 1200	Pro M Enhanced (V2)

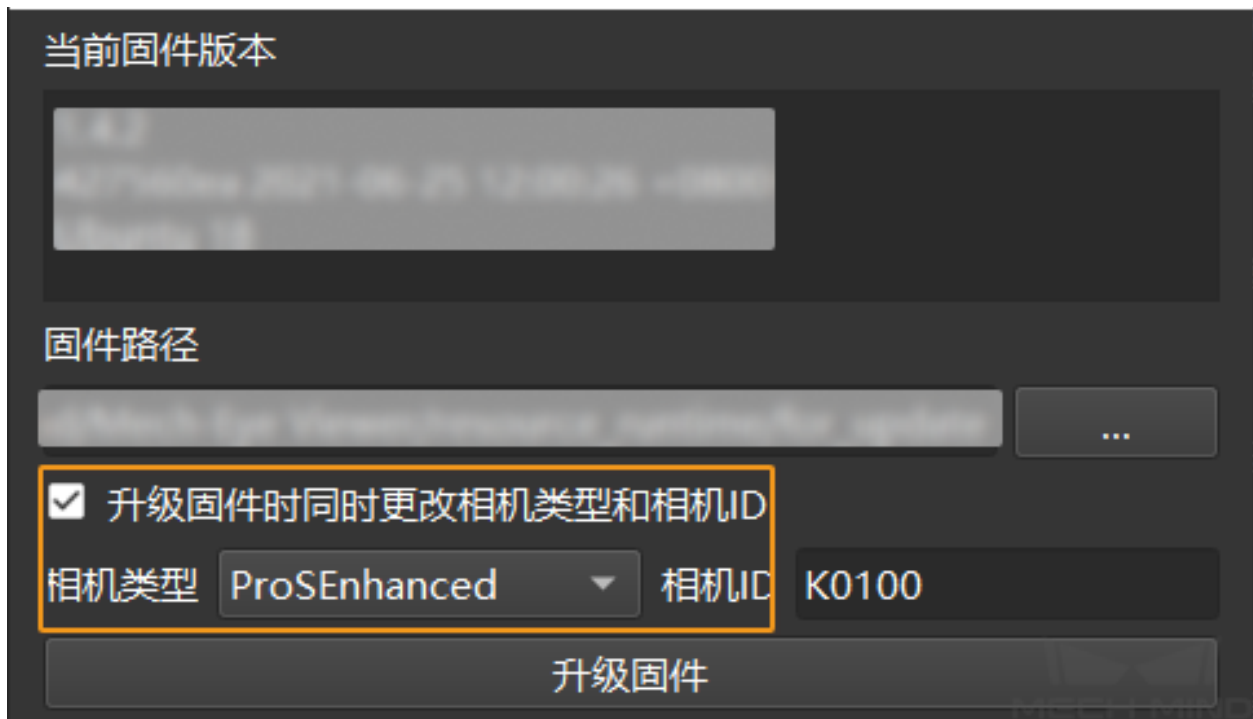
通过相机铭牌:

相机铭牌位于相机正面，MECH MIND 下方，查看铭牌中的 **Model** 即可。

### 修改相机型号并升级固件

当相机型号为 Pro S Enhanced 或 Pro M Enhanced 时，需要修改相机型号再升级固件。需执行如下步骤：

1. 使用 Mech-Eye Viewer 连接相机。单击  标准 切换到管理员模式，输入密码。请联系技术支持获取密码。
2. 依次单击 工具 ▶ 相机固件升级进入 相机固件升级页面。
3. 勾选 升级固件时同时更改相机类型和相机 ID 复选框，并在 相机类型中选择正确的相机类型。



4. 完成后单击 升级固件，等待固件升级即可。

查看本章可了解更多相机知识。

查看以下内容，了解结构光相机的工作原理

[结构光相机工作原理](#)

查看以下内容，了解如何设置与相机相连的工控机网口的 IP 地址

[设置工控机 IP 地址](#)

## 23.1 结构光相机工作原理

Mech-Eye 工业级 3D 相机（简称 3D 相机），由 2D 相机与投影仪组成。投影仪投射特定样式的结构光到物体表面，物体形状导致结构光样式变形。3D 相机根据变形后的结构光样式，计算出物体的深度数据。

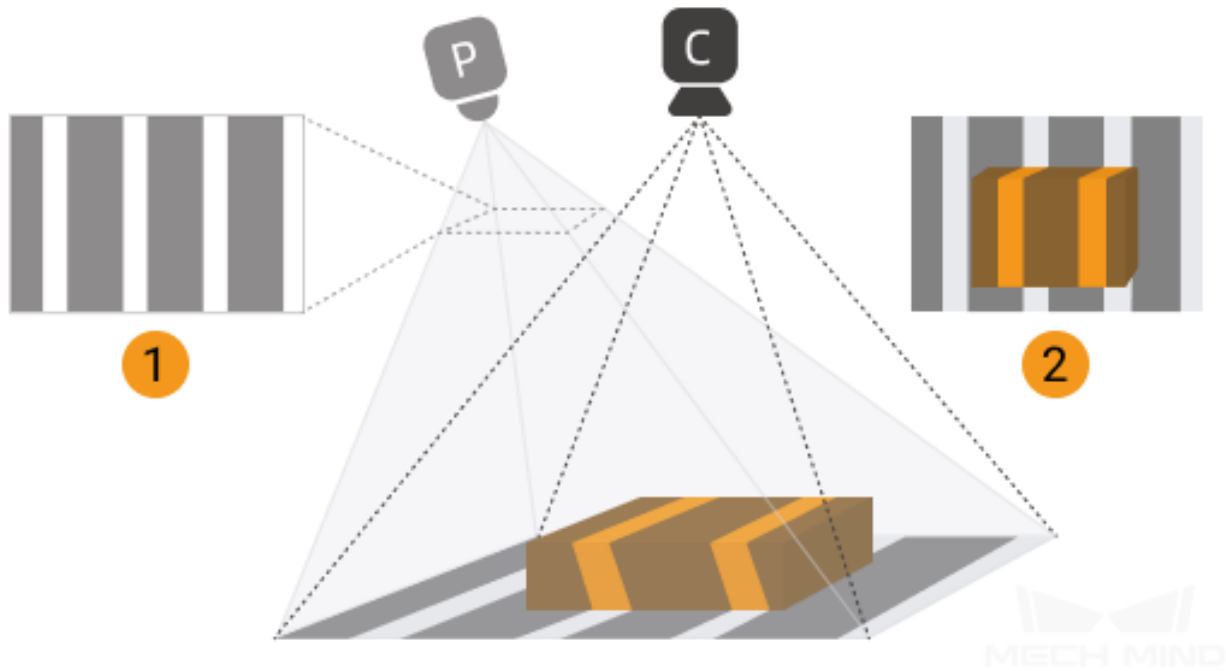
获取并查看相机数据的过程，分为三个阶段：

- 拍摄
- 计算与处理
- 输出

### 23.1.1 拍摄

使用 Mech-Eye SDK 触发 3D 相机采集图像，采集过程为：

1. 投影仪（下图中 P）投射条纹样式的光至被拍摄物体表面。由于物体本身的结构和形状，条纹样式变形。
2. 2D 相机（下图中 C）拍摄，生成包含变形条纹样式的图像。



### 23.1.2 计算与处理

3D 相机根据条纹样式的变形，计算出深度数据；再经过处理生成深度图与点云。

### 23.1.3 输出

3D 相机将生成的数据输出至 Mech-Eye SDK，用户即可看到 2D 图、深度图和点云。

---

注解：

- 2D 图：投影仪不投光时拍摄的图像。
  - 深度图：包含深度信息的图像。
  - 点云：包含三维坐标（XYZ）信息的物体表面点的集合。
-



## 23.2 设置工控机 IP 地址

本章介绍如何设置与相机相连的工控机网口的 IP 地址。

1. 依次单击 控制面板 ▶ 网络和 Internet ▶ 网络和共享中心 ▶ 更改适配器设置进入网络连接页面。
2. 右键单击与相机相连的网口，选择 属性，进入属性页面。
3. 依次单击 Internet 协议版本 4 (TCP/IPv4) ▶ 属性进入 IP 地址设置界面，如下图所示。



4. 单击 使用下面的 IP 地址并输入 IP 地址和 子网掩码，单击 确定，设置完成。

Internet 协议版本 4 (TCP/IPv4) 属性

常规

如果网络支持此功能，则可以获取自动指派的 IP 设置。否则，你需要从网络系统管理员处获得适当的 IP 设置。

自动获得 IP 地址(O)

**1**  使用下面的 IP 地址(S):

**2** IP 地址(I):

子网掩码(U):

默认网关(D):

自动获得 DNS 服务器地址(B)

使用下面的 DNS 服务器地址(E):

首选 DNS 服务器(P):

备用 DNS 服务器(A):

退出时验证设置(L)

高级(V)...

**3** 确定 取消

提示：如需自动获取 IP 地址，在 IP 设置界面切换为 自动获得 IP 地址。

---