



# Mech-Eye工业级3D相机用户 手册

v2.2.2

# 目录

---

1. 欢迎	1
2. 开始使用相机	4
3. Mech-Eye SDK更新说明	15
4. Mech-Eye SDK安装指南	16
5. Mech-Eye Viewer	21
5.1. 界面介绍	21
5.2. 操作指南	25
5.2.1. 设置相机IP地址并连接相机	26
5.2.2. 采集与查看数据	28
5.2.3. 调节参数	35
5.2.4. 保存数据	39
5.2.5. 管理日志	39
5.3. 参数参考指南	40
5.3.1. NANO参数	41
5.3.2. PRO S和PRO M参数	56
5.3.3. PRO XS参数	74
5.3.4. UHP-140参数	89
5.3.5. DEEP / LSR系列参数	105
5.3.6. V3相机参数	121
5.4. 工具使用指南	138
5.4.1. 内参工具	139
5.4.2. 3D曝光助手	142
5.4.3. 深度图分析器	143
5.4.4. 查看并配置2D相机	144
5.4.5. 视野计算器	145
5.4.6. 自定义坐标系	146
5.4.7. 相机管理器	149
5.4.8. 满垛模拟器	150
6. Mech-Eye API	153
6.1. Mech-Eye SDK安装指南 (Ubuntu)	154
6.2. 类的功能层级	156
6.3. 使用流程	158
6.4. 迁移指南 (从2.1.0版本到2.2.0版本)	164
6.5. Mech-Eye API参考手册	190
6.6. 例程使用指南	190
6.6.1. C++ (Windows)	190
6.6.2. C++ (Ubuntu)	198
6.6.3. C#	203
6.6.4. Python (Windows)	207
6.6.5. Python (Ubuntu)	210
6.6.6. VisionPro	214
6.7. 工具	217
6.7.1. 固件升级工具	217

6.7.2. IP配置工具	217
6.8. 向下兼容性相关	218
6.8.1. 2.2.0版本的向下兼容性	218
6.8.2. C++参数名称更新	219
6.8.3. C#方法名称与参数名称更新	220
7. GenICam接口	225
7.1. 什么是GenICam?	225
7.2. HALCON	225
7.2.1. HALCON——连接相机、调节参数、采集图像	226
7.2.2. HALCON——手眼标定	233
7.2.3. HALCON——获取纹理点云	240
7.2.4. HALCON——仅获取Z值	241
7.2.5. HALCON——设置IP地址	242
7.3. 参考手册	244
7.3.1. GenICam支持的相机参数	244
7.3.2. 借助Mech-Eye Viewer为GenICam客户端设置相机参数	248
7.3.3. 通过Mech-Eye API获取HALCON可读取的点云	250
8. 相机硬件	252
8.1. 相机型号对比	252
8.2. 硬件用户手册	253
8.3. 技术参数	264
8.3.1. NANO	264
8.3.2. PRO S / PRO M	267
8.3.3. UHP-140	272
8.3.4. DEEP	274
8.3.5. LSR S	277
8.3.6. LSR L	280
8.3.7. V3相机技术参数	283
9. 扩展阅读	285
9.1. 结构光相机工作原理	285
9.2. 预置参数组	286
9.3. 设置电脑IP地址	287
9.4. 启用巨型帧	289
10. 售后支持	291
10.1. 故障排查	291
10.1.1. 接通电源后，相机指示灯指示异常情况	291
10.1.2. Mech-Eye SDK搜索不到相机	294
10.1.3. Mech-Eye SDK无法连接相机	297
10.1.4. 相机不投光，且未生成深度图	300
10.1.5. Mech-Eye SDK无法连接相机（Windows Server）	302
10.1.6. 图像采集时间过长	302
10.1.7. V2相机升级固件后采集的点云有缺失	304
10.2. FAQ	305
10.2.1. 如何解读相机序列号?	305



# 1. 欢迎

## 产品和用户手册说明

Mech-Eye工业级3D相机（简称“相机”）是梅卡曼德研制的结构光3D相机。通过Mech-Eye SDK或第三方机器视觉软件，你可从相机获取2D图、深度图及点云。

本手册提供Mech-Eye SDK的软件用户手册、相机硬件用户手册、相机技术参数及售后支持等内容。



- 如需下载PDF格式的硬件用户手册和技术参数，或需获取相机3D模型、安装图纸、配件信息、产品宣传册等相关资料，请访问[梅卡曼德下载中心](#)。
- 如使用相机时遇到问题，请访问[梅卡曼德在线社区](#)获取帮助（需注册并登录账户）。

## 新手必读

以下章节提供相机的入门教程、工作原理和型号对比等基本信息。

### 入门教程

带你完成从开箱检查到完成第一次数据采集的一系列操作：

[开始使用相机](#)

### 相机工作原理

了解相机如何生成物体的3D数据：

[结构光相机工作原理](#)

### 型号对比

了解相机不同型号的特点和适用场景：

[相机型号对比](#)

## 软件用户手册

你可通过Mech-Eye SDK或第三方机器视觉软件连接相机、采集数据和调节参数。同时，相机还支持GenICam标准，提供GenICam通用程序接口。

### Mech-Eye Viewer

Mech-Eye Viewer提供图形化的用户界面，操作简单方便，并可在调节参数后即刻查看效果。

以下章节提供Mech-Eye Viewer的使用指导：

[Mech-Eye Viewer使用指南](#)

## Mech-Eye API

Mech-Eye API为相机的应用程序接口，有C++、C#和Python三种语言版本，并可在Windows和Ubuntu系统中使用。

你可基于Mech-Eye API开发自己的相机控制程序：

[Mech-Eye API使用指南](#)

你也可以通过Mech-Eye API使用第三方机器视觉软件（如VisionPro）控制相机并获取数据：

- [VisionPro例程使用指南](#)
- [LabVIEW例程使用指南](#)（敬请期待）

## GenICam接口

兼容GenICam标准的第三方机器视觉软件（如HALCON）可通过GenICam接口控制相机，降低与已有系统的集成难度。

以下章节提供Mech-Eye Viewer的使用指导：

[通过HALCON控制相机](#)

# 参考信息

## 相机资料

以下章节提供相机的硬件用户手册和技术参数：

- [硬件用户手册](#)
- [技术参数](#)

如需下载PDF格式的硬件用户手册和技术参数，或需获取相机3D模型、安装图纸、配件信息、产品宣传册等相关资料，请访问梅卡曼德下载中心：

[下载相机资料](#)

## 故障排查及FAQ

以下章节提供常见问题的故障排查指导和常见疑问的回答：

- [故障排查](#)

- [FAQ](#)

如遇到以上章节未涉及的问题，请访问梅卡曼德在线社区获取帮助（需注册并登录账户）：

- [在社区上提问](#)
- [相机使用经验](#)

## 2. 开始使用相机

本章将引导你完成从开箱检查到使用Mech-Eye Viewer采集图像的一系列操作。

### 开箱检查

1. 收到包装后，请确认包装完好无破损。
2. 找到包装内的《包装清单》，并参照该清单检查物品和配件无缺失或损坏。



以下列表仅供参考，请以实际包装内的《包装清单》为准。

相机	
配件袋	
使用说明书	
标定板 (仅UHP-140)	
DC电源线 (20m) CBL-PWR-20M-LU	
网线 (20m) CBL-ETH-20M-LU	

- 可选配件：

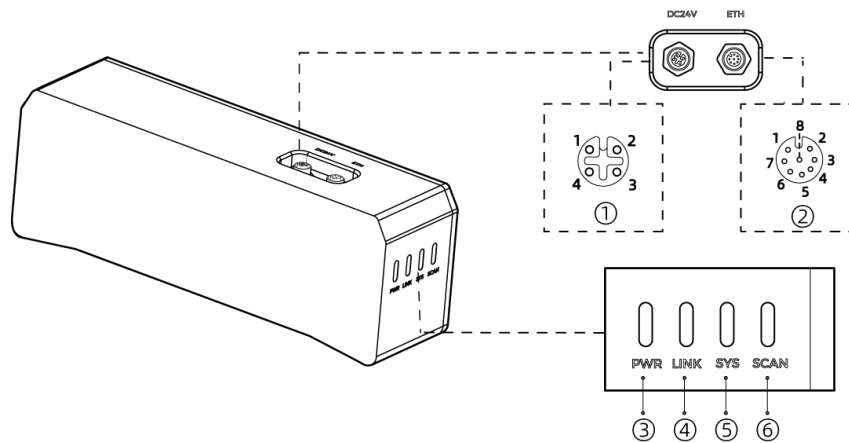


导轨电源	
标定板  (除UHP-140外)	

## 确认接口及指示灯

请对照以下图片及表格，确认相机上各接口及指示灯的功能。

### PRO S/PRO M、DEEP、UHP-140、LSR S和LSR L

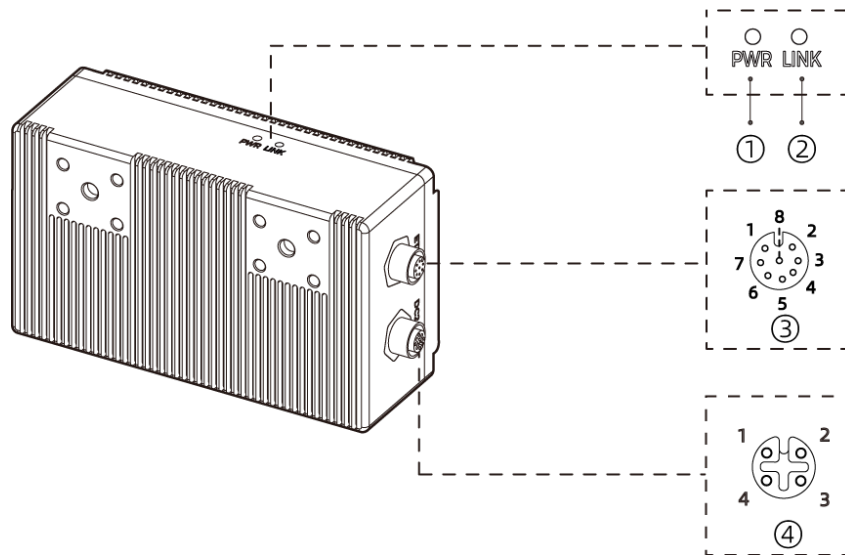


以上图示仅供参考，请以实物为准。

No.	名称	功能	
①	DC 24V电源接口	1: GND 3: 24V DC	2: GND 4: 24V DC
②	ETH网口	1: MD3_P 3: MD2_P 5: MD1_P 7: MD3_N	2: MD2_N 4: MD0_P 6: MD0_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
⑥	SCAN指示灯	常亮：采集和计算中
		熄灭：未采集

## NANO和PRO XS




上图以NANO为例。


No.	名称	功能
①	PWR指示灯	熄灭：电源未连接 绿灯亮：电压正常
②	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
③	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
④	DC 24V电源接口	1: GND                         2: GND 3: 24V DC                      4: 24V DC

## 安装相机

相机有多种安装方式。请根据实际情况选择合适的安装方式。

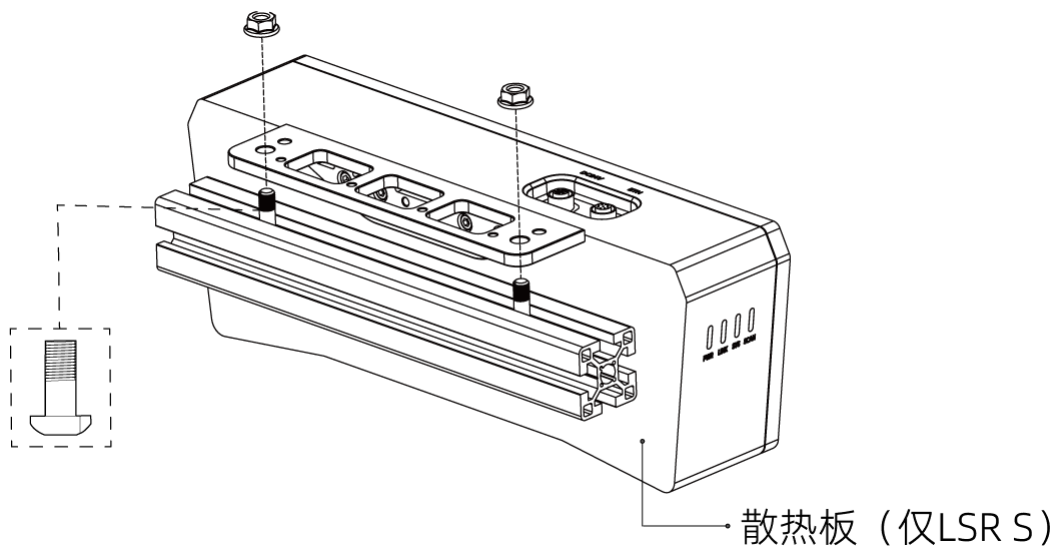
-  请自行准备扳手。

### 使用L型转接件安装

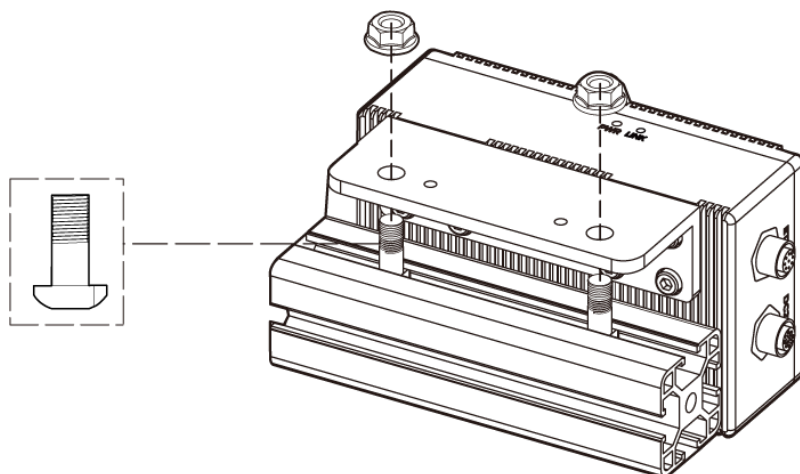
-  发货时，L型转接件与散热板（仅LSR S）已安装至相机背部。

如下图所示，使用扳手拧紧两颗螺母，固定相机。


- PRO S/PRO M、DEEP、UHP-140、LSR S和LSR L:



- NANO和PRO XS:



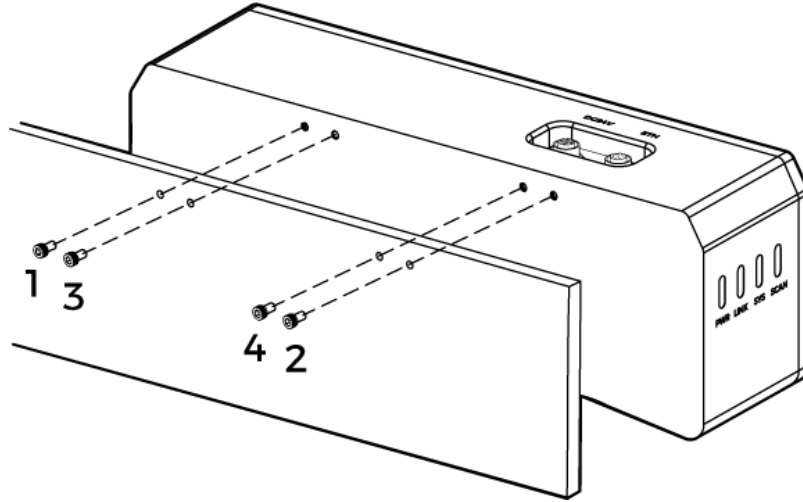
### 使用相机背面螺纹孔安装

- 
  - 安装前请拆卸L型转接件与散热板（仅LSR S）。
  - LSR S使用此安装方式时，需将相机背面贴紧金属散热面安装，防止相机散热不良，影响

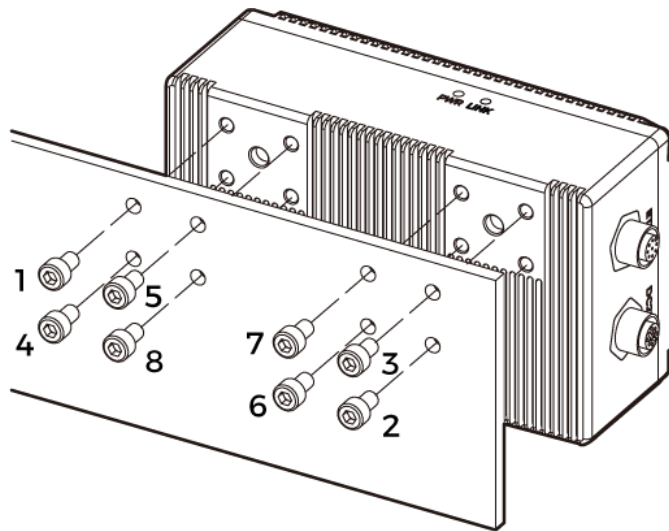
- 正常工作。

如下图所示，使用扳手按顺序先预紧，再按顺序拧紧螺钉。


- PRO S/PRO M、DEEP、UHP-140、LSR S和LSR L:



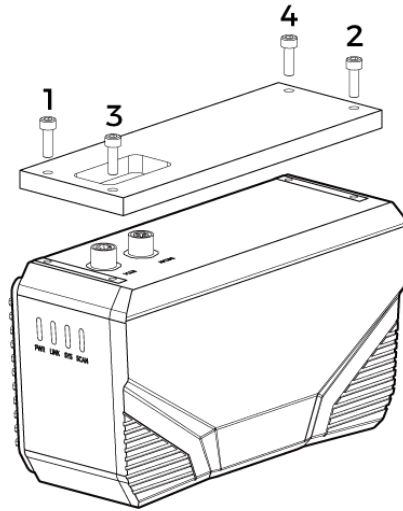
- NANO和PRO XS:



### 使用相机顶部螺纹孔安装（LSR S和UHP-140）

-  安装前请拆卸L型转接件。

如下图所示，使用扳手按顺序先预紧，再按顺序拧紧螺钉。



## 连接线缆

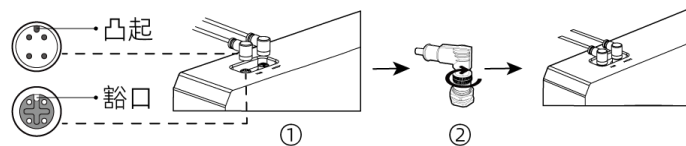
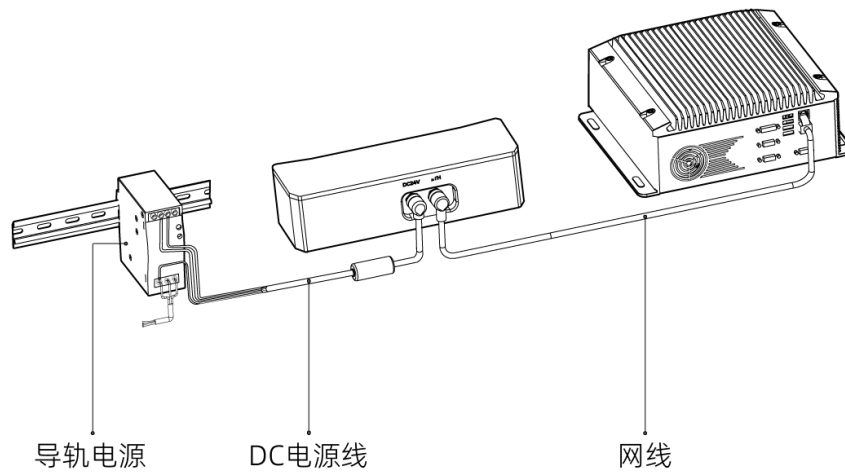
请按照以下步骤，连接相机的网线和电源线。

### 网线

如下图所示，将网线的航插头插入相机的ETH网口，RJ45接头插入工控机的网口。

### DC电源线

如下图所示，将DC电源线的航插头插入相机的DC 24V电源接口。



连接网线与DC电源线时：

1. 将航插的凸起对准对应接口的豁口插入。

2. 拧紧紧固螺母，推荐扭矩为 $0.7N \cdot m$ 。拧紧后，约有2mm间隙。

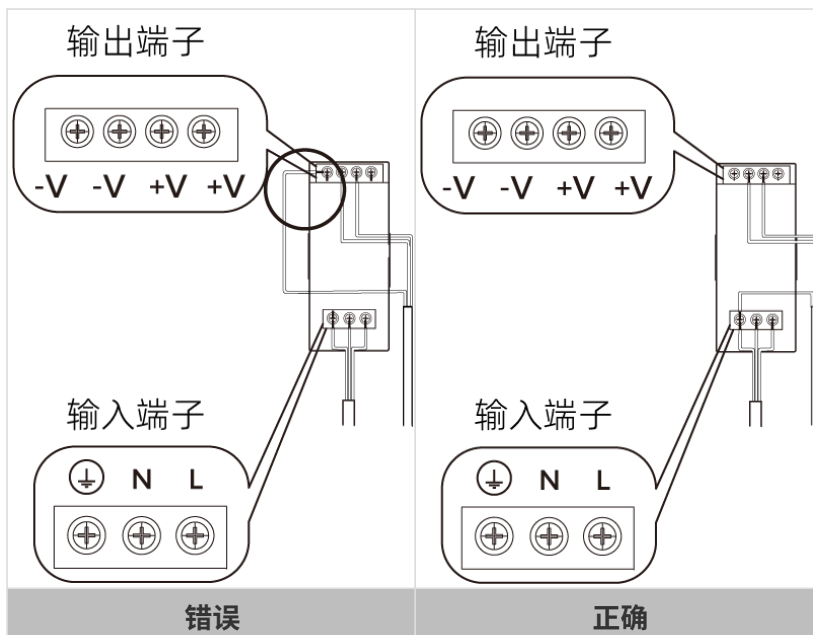


- 连接相机与工控机时，可使用交换机连接。
- 连接时，请最后接通电源。接通电源后，PWR指示灯应绿色常亮。否则，请联系梅卡曼德。
- 当相机安装在机械臂或其他移动装置上时，连接相机端的DC电源线与网线需妥善固定，防止拉扯损坏线缆或插头。

## 导轨电源



- 请自行准备AC电源线。
- 本节的导轨电源为梅卡曼德提供的导轨电源。如您自行准备导轨电源，请查看对应的说明书进行连接。



AC电源线共有三股接线插头，分别为：L、N、PE (⊕)。

DC电源线共有三股接线插头，分别为：+V、-V、PE (⊕)。

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。

### ⚠ 警告

- 导轨电源须放在配电箱中使用。
- 导轨电源或连接导轨电源的导轨应可靠接地。如多台导轨电源安装在同一导轨时，各导轨电源应保持一定间距。
- AC电源插座应采用有保护接地线（PE线）的单相三线电源插座。

相机硬件的安装和连接已完成。接下来将介绍如何使用Mech-Eye Viewer连接相机，并控制相机采

集图像。

## 下载并安装Mech-Eye SDK

请联系梅卡曼德售前工程师或销售人员获取Mech-Eye SDK安装包。


解压安装包后，双击安装文件即可安装Mech-Eye SDK。更多详细信息可参考[Mech-Eye SDK安装指南](#)。

## 设置IP地址

连接相机前，需确保以下两个IP地址唯一，且在同一网段。

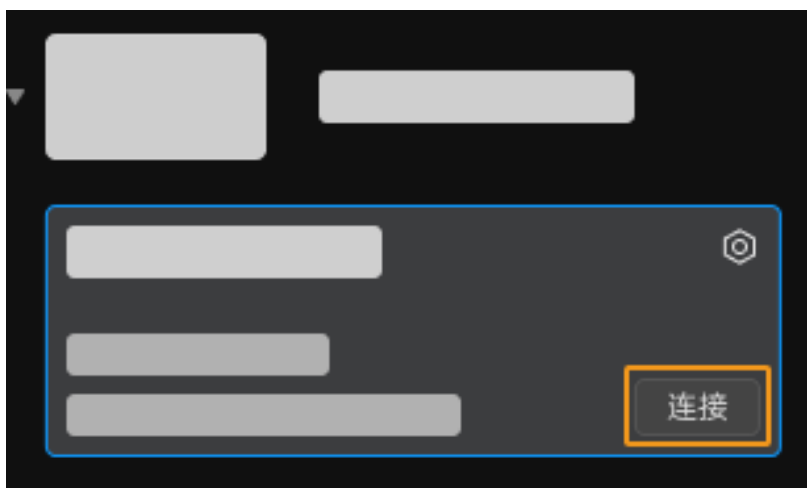
- 相机IP地址
- 与相机相连的电脑网口的IP地址

请按照以下步骤设置相机的IP地址：

1. 双击打开Mech-Eye Viewer。
2. 选择待连接的相机，单击。
3. [设置相机的IP地址](#)。

## 连接相机

1. 在Mech-Eye Viewer中找到待连接的相机，单击[ **连接** ]。




软件或固件需要升级时，将显示[ **升级** ]按钮。请先单击该按钮升级，再连接相机。

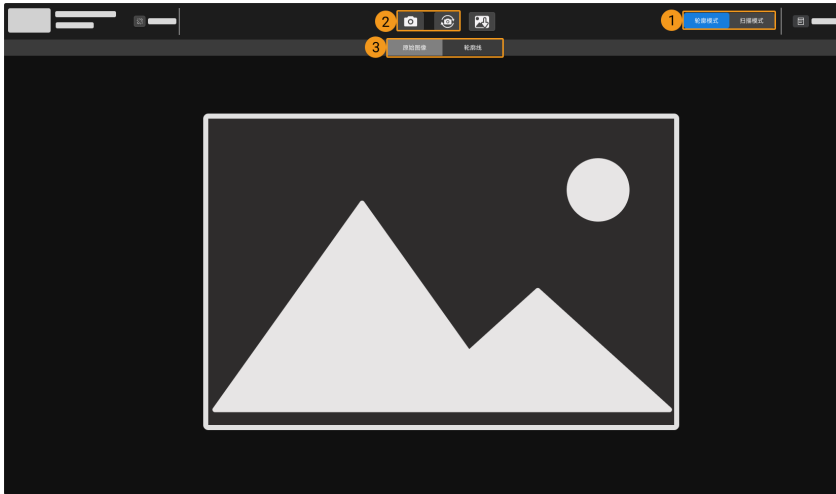
## 采集图像

单击上方的即可进行一次图像采集。



单击 ，可以固定时间间隔连续进行图像采集，再次单击停止图像采集。

单击图像采集按钮下方的数据类型，即可查看从相机获取的2D图、深度图和点云。



更多关于采集数据和数据类型的信息，详见[采集与查看数据](#)。

## 调节参数

如获取的2D图、深度图和点云质量不理想，可在软件最右侧的**参数**标签页中调节相关参数，提升数据质量。

### 判断数据质量

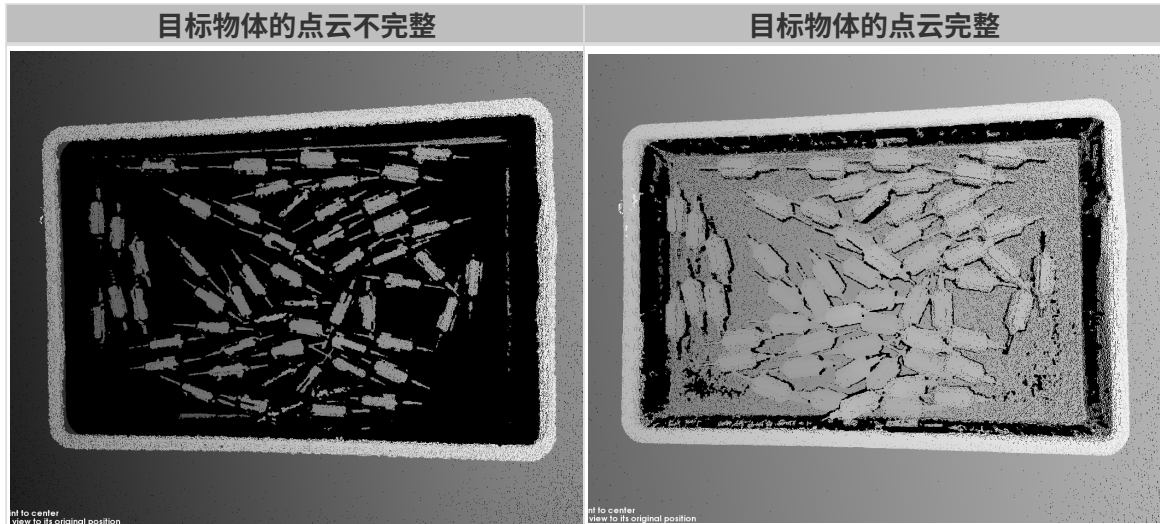
请根据以下标准，初步判断数据质量。

- 2D图：2D图应不过亮或过暗，可看清目标物体的表面特征。



- 深度图和点云：深度图和点云中，目标物体对应的数据应完整。以下示例中，目标物体为转子。





## 提升2D图质量

1. 将2D参数分组中的**曝光模式**参数设置为**固定曝光**，然后调节**曝光时间**参数。
  - 如2D图过暗，请将**曝光时间**调大。
  - 如2D图过亮，请将**曝光时间**调小。



DEEP与LSR系列相机提供两种2D图，分别对应不同的参数。详见[DEEP与LSR系列参数](#)。

2. 重新采集图像，确认2D图的质量。

## 提升深度图及点云质量


1. 调节3D参数分组中的**曝光时间**参数。
  - 如物体颜色较深或不易反光，请将**曝光时间**调大。
  - 如物体颜色较浅或反光，请将**曝光时间**调小。
2. 重新采集图像，确认深度图和点云的质量。



更多参数的说明，详见[参数参考指南](#)。

## 使用数据

通过Mech-Eye Viewer获取的2D图、深度图和点云，可直接保存至本地，也可使用Mech-Vision或第三方机器视觉软件进行后续的处理和计算。

- 保存数据：单击数据采集区的 ，设置保存路径，勾选需要保存的数据类型，再单击[保存]。
- 在Mech-Vision中使用数据：请参考[视觉系统教程](#)，学习搭建包含Mech-Vision在内的整套视觉系统。

- 在第三方机器视觉软件中使用数据：通过[Mech-Eye API](#)或[GenICam接口](#)，可将相机获取的数据传输至第三方软件。

## 3. Mech-Eye SDK更新说明

---

本文介绍Mech-Eye SDK 2.2.2版本的新增功能、功能优化以及问题修复。

### Mech-Eye Viewer

#### 优化内参工具

Mech-Eye Viewer 2.2.2调整了内参检查标准，更符合实际使用场景。

### 历史版本更新说明

- [Mech-Eye SDK 2.2.1更新说明](#)
- [Mech-Eye SDK 2.2.0更新说明](#)
- [Mech-Eye SDK 2.1.0更新说明](#)

## 4. Mech-Eye SDK安装指南

本章节介绍如何在Windows操作系统中下载、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。

使用Mech-Eye Viewer时，所用显示器的分辨率和缩放率需遵循如下对应关系。如果使用非下表所示分辨率和缩放率的显示器，可能存在显示问题。

如果使用双显示器，需确保两个显示器的分辨率和缩放率一致。



分辨率	缩放率
1280×800 (16:10)	100%
1920×1080 (16:9)	100%、125%
2560×1440 (16:9)	125%、150%
3840×2160 (16:9)	150%、175%

### 下载Mech-Eye SDK安装包

请进入[梅卡曼德下载中心](#)下载Mech-Eye SDK安装包。

### 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。可通过CRC-32校验码校验安装包的完整性，该校验码在下载页面提供。



请安装并使用7-Zip软件计算CRC-32校验码。

要校验软件安装包的完整性，执行如下步骤：

1. 将软件安装包拷贝到电脑的指定目录，例如D:/。
2. 解压软件安装包。解压后将得到软件安装文件（Mech-Eye SDK Installer 2.2.2.exe）。
3. 打开7-Zip，并通过地址栏导航至软件安装包所在的目录。
4. 选中软件安装包后，在菜单栏中选择文件 > CRC > CRC-32，计算CRC-32校验码。
5. 确认计算的CRC32校验码与下载页面提供的CRC32校验码相同。
6. 对解压得到的软件安装文件重复步骤3至5。



如果校验码不同，请重新下载软件安装包。

如果你第一次使用Mech-Eye SDK，请参照[安装Mech-Eye SDK](#)安装软件。

如果你已安装Mech-Eye SDK，请参照[升级Mech-Eye SDK](#)升级软件。

## 安装Mech-Eye SDK

要安装Mech-Eye SDK，执行如下步骤：

1. 双击下载的安装包，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 ]。



- 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，可在不联网时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 ]。



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。



安装完成后，请重启电脑，否则可能导致添加的环境变量不生效。

## 升级Mech-Eye SDK



- 如你已安装1.6.1以上版本Mech-Eye SDK，可直接使用更新版本的安装向导升级。
- 如你已安装1.6.1及以下版本Mech-Eye SDK，升级时需要先卸载低版本软件，然后再[安装新版本软件](#)。

要升级Mech-Eye SDK，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**升级**窗口，单击[ 升级，保留历史版本 ]或[ 升级，删除历史版本 ]。
3. 根据安装向导的界面提示升级软件。
4. 软件升级完成后，在**结束**窗口，单击[ 完成 ]，退出安装向导。

## 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。

## 使用安装向导卸载

要使用安装向导卸载Mech-Eye SDK，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[ **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。软件卸载完成后，在**结束**窗口，单击[ **完成** ]，退出安装向导。

## 使用控制面板卸载

要使用控制面板卸载Mech-Eye SDK，执行如下步骤：

1. 在电脑上，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**。将打开Mech-Eye SDK安装向导。
4. 在**维护**窗口，点击[ **卸载** ]。
5. 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** ]。
6. 等待软件完成卸载。软件卸载完成后，在**结束**窗口，单击[ **完成** ]，退出安装向导。

## 修复Mech-Eye SDK



仅2.0.0及以上版本的Mech-Eye SDK安装向导支持该功能。

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。

要修复Mech-Eye SDK，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[ **修复** ]。
3. 等待软件完成修复。软件修复完成后，在**结束**窗口，单击[ **完成** ]，退出安装向导。

## 修改安装的组件



仅2.0.0及以上版本的Mech-Eye SDK安装向导支持该功能。

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. 根据安装向导界面提示安装软件。
5. 软件安装完成后，在**结束**窗口，单击[ **完成** ]，退出安装向导。

## 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](#)。

## 安装常见问题

### 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。

### 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**来安装软件。
  - 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. 重启电脑，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。

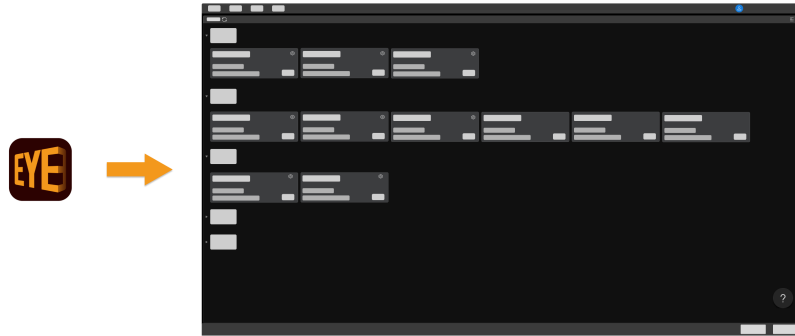


## 5. Mech-Eye Viewer

Mech-Eye Viewer是基于Mech-Eye API开发的图形化界面软件。你可根据目标物体特性，使用Mech-Eye Viewer调节相机的各种参数，快速获得高质量的2D图、深度图和点云。



目前仅支持在Windows系统中使用Mech-Eye Viewer。Ubuntu系统用户可通过[Mech-Eye API](#)控制相机。



查看以下内容，了解**软件界面**。

[界面介绍](#)

查看以下内容，了解如何**连接相机、采集数据、调节参数**。

[操作指南](#)

查看以下内容，了解**参数详细信息**。

[参数参考指南](#)

查看以下内容，了解如何**使用Mech-Eye Viewer中的工具**。

[工具使用指南](#)

### 5.1. 界面介绍

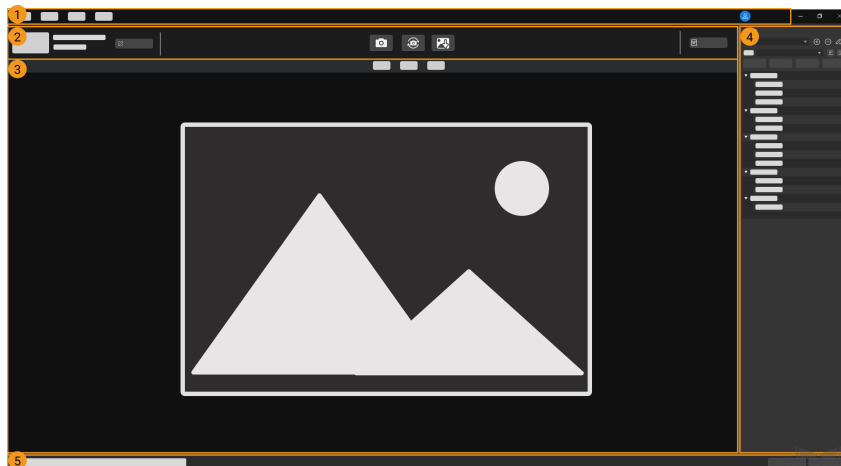
启动Mech-Eye Viewer，进入如下界面。



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

将光标悬浮在  上，并单击[工业级3D相机](#)，可查看如何解决[相机连接异常](#)的问题。

相机连接成功后，进入如下界面。




该界面包含以下部分：

1. [菜单栏](#)：提供[文件](#)、[工具](#)、[视图](#)与[帮助](#)菜单。
2. [数据采集区](#)：用于设置备注、采集数据、保存数据、断开连接和查看日志信息。
3. [数据查看区](#)：显示从相机获取的数据。
4. [参数和采集信息区](#)：用于调节参数、管理参数组和查看采集相关的信息。
5. [状态栏](#)：显示相机运行状态。单击[ [上一条](#) ]或[ [下一条](#) ]切换显示信息。

## 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	保存相机原始数据 (.mraw) , 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, 查看保存原始数据时设置的参数等。
工具	包含一系列辅助工具, 更多信息请参阅 <a href="#">工具使用指南</a> 。	
视图	工具栏	勾选后显示工具栏。默认不勾选。
	图像信息框	勾选后显示图像信息框, 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, 用于 <a href="#">展示点云</a> 。默认不勾选。
	参数说明	勾选后在 <a href="#">参数</a> 标签页中显示 <a href="#">参数说明</a> 区。默认勾选。
	深度图网格	选择是否在深度图上显示网格和网格的样式。
帮助	关于	查看软件版本信息。
	更新说明	在浏览器中打开Mech-Eye SDK的更新说明。
	使用手册	在浏览器中打开软件使用手册。
	在线社区	在浏览器中打开梅卡曼德在线社区。
	设置	切换软件语言, 重启软件后生效。
	切换用户类型。默认用户为 <b>标准</b> , 如需使用 <b>管理员</b> 用户, 请联系梅卡曼德技术支持。	

## 数据采集区

在数据采集区中, 你可进行以下操作:

- [设置备注](#)
- [采集图像](#)
- [保存数据](#)
- [查看日志](#)
- 断开与当前相机的连接

## 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。

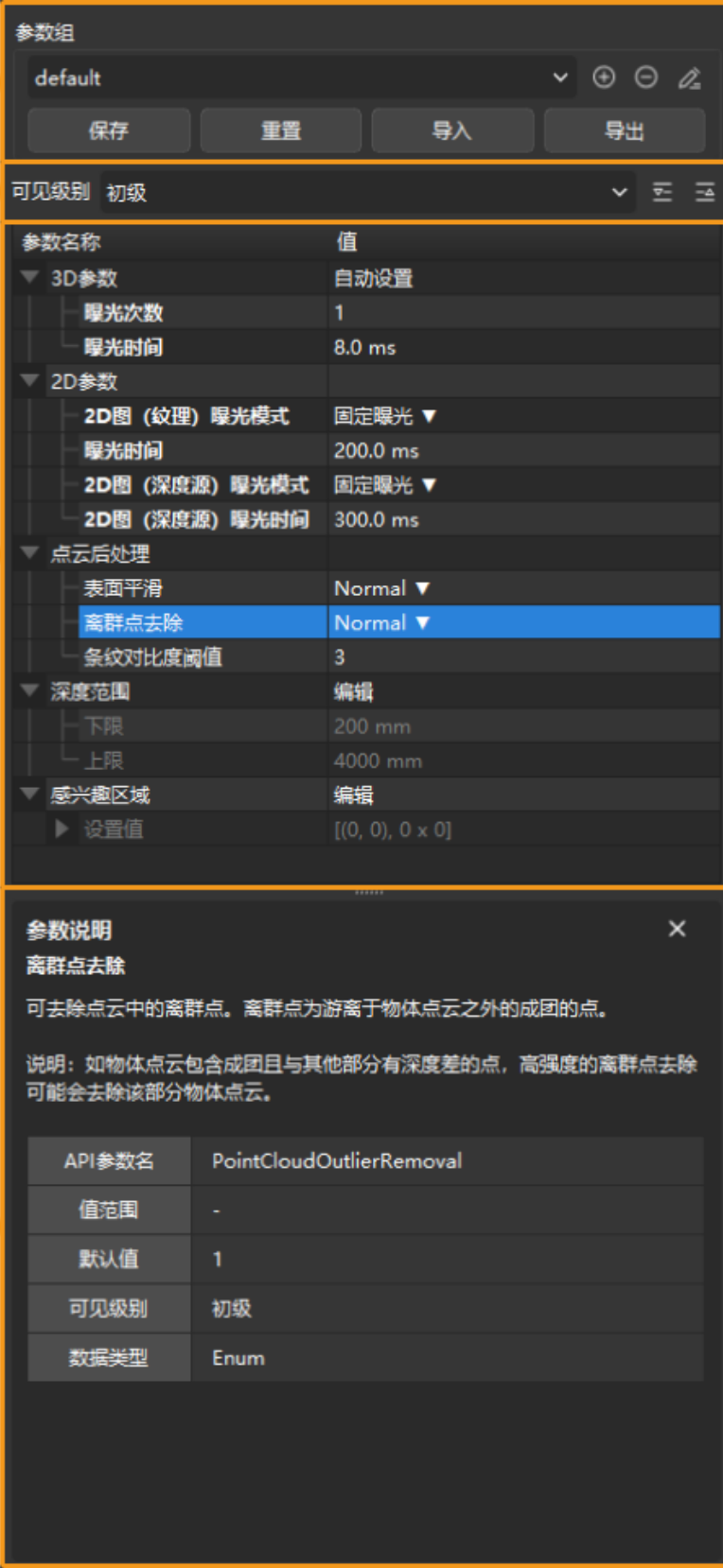
各数据显示界面的更多详细功能, 请参阅[查看数据时的操作](#)。

## 参数和采集信息区

该区域包含两个标签页, [参数](#)和[采集信息](#)。

### 参数标签页

参数标签页分为以下四个部分:



The screenshot shows the 'Parameter Group' (参数组) interface. At the top, there is a dropdown menu set to 'default' with buttons for 'Save' (保存), 'Reset' (重置), 'Import' (导入), and 'Export' (导出). Below this is a 'Visible Level' (可见级别) dropdown set to 'Primary' (初级). The main area is a table of parameters:

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 ▼
<b>离群点去除</b>	<b>Normal ▼</b>
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]

Below the table is a 'Parameter Description' (参数说明) window for 'Outlier Removal' (离群点去除):

**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

1. 参数组：创建新的参数组、将参数值保存至参数组、删除参数组、重命名参数组以及导入和导出参数组。
2. 可见级别：更改参数的可见级别。

3. 参数列表：[调节参数的基本流程](#)。
4. 参数说明：[查看选中参数的更多信息](#)。

## 采集信息标签页

采集信息标签页中显示以下与当次数据采集相关的信息：

- 相机备注和采集的结束时间
- 总采集时间：采集2D图、深度图和点云所用时间
  - 2D图/2D图（纹理）：采集2D图/2D图（纹理）所用时间
  - 深度数据：采集深度图和点云所用时间
- 分辨率：根据型号显示

型号	显示条目	说明
LSR L、LSR S和DEEP以外的型号	2D图	2D图的分辨率
	深度图	深度图的分辨率
LSR L、LSR S和DEEP	2D图（纹理）	2D图（纹理）的分辨率
	2D图（深度源）	2D图（深度源）的分辨率
	深度图	深度图的分辨率



LSR L、LSR S和DEEP的2D图（纹理）或深度图的分辨率可在**管理员**用户下使用**相机管理器**更改。

- 温度
  - CPU：相机CPU的温度
  - 投光单元：相机投光单元的温度

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

## 5.2. 操作指南

本章主要介绍如何使用Mech-Eye Viewer连接相机、采集数据、查看数据、调节参数和保存数据。

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[设置相机IP地址](#)、[连接相机](#)及[修改相机备注](#)。

[设置相机IP地址并连接相机](#)

查看以下内容，了解如何**获取和查看数据**。

[采集与查看数据](#)

---

查看以下内容，了解如何**调节参数**，以获取符合要求的数据。

[调节参数](#)

---

查看以下内容，了解如何**保存数据**。

[保存数据](#)

---

当软件出现问题时，查看以下内容，了解**日志相关的操作**。

[管理日志](#)

---

### 5.2.1. 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](#)
- [与相机相连的电脑网口的IP地址](#)

IP地址设置完成后，即可[连接相机](#)。连接后，你可为相机**设置备注**，以便区分不同设备。



如Mech-Eye Viewer不显示需连接的相机，请参考[Mech-Eye SDK搜索不到相机](#)。

#### 设置相机IP地址


你可为相机设置一个静态IP地址，或者动态分配IP地址。

#### 设置静态IP地址



请保存好相机IP地址，以备后用。

请执行以下步骤为相机设置静态IP地址：

1. 打开Mech-Eye Viewer，选中待连接的相机的信息卡片，并单击卡片右上角的，打开**IP配置**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]。



- IP地址必须唯一。
- **电脑IP配置**中显示的是与相机相连的电脑网口的IP地址及子网掩码。

### 动态分配IP地址

请执行以下步骤为相机动态分配IP地址：

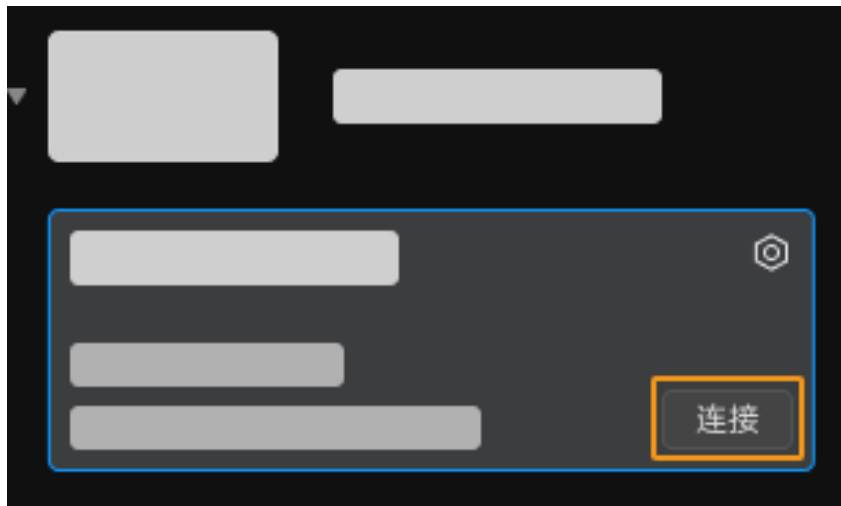
1. 打开Mech-Eye Viewer，选中待连接的相机的信息卡片，并单击卡片右上角的，打开**IP配置**窗口。
2. 选择**设置为动态分配**，并单击[应用]。



如需设置与相机相连的电脑网口的IP地址，请参考[设置电脑IP地址](#)。


### 连接相机

相机的信息卡片中显示[连接]按钮时，单击该按钮即可连接相机。



首次连接相机时，需等待几分钟，直至Mech-Eye Viewer中出现该相机。

如显示其他内容，请根据下表完成相应的操作后，再连接相机。

显示内容	说明
[升级]按钮	需先单击该按钮升级软件或相机固件。
已连接	相机已被GenICam客户端连接。需先在GenICam客户端中断开与相机的连接。
	将光标悬浮在此处查看原因和解决方案，并根据提示内容解决问题。

### 设置备注

相机备注可用于区分不同设备、注明设备用途和位置等，请根据需要自行设置。设置完成后，备注将显示在**设备列表**中。

连接相机后，单击数据采集区左侧的  或者**请设置备注**字样，设置备注。

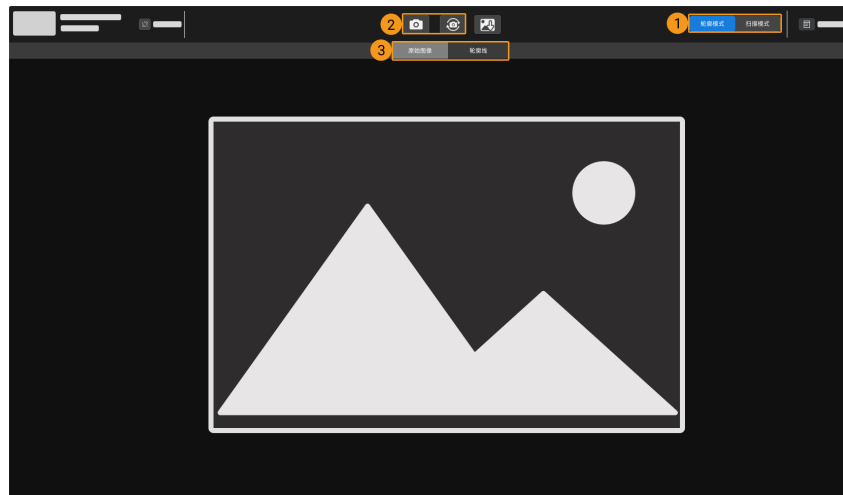


- 以下特殊字符不能使用：\/:\*?"<>|。
- 再次单击备注可更改备注。
- 虚拟相机不可设置备注。

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。

### 5.2.2. 采集与查看数据



相机连接成功后进入以下界面，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：

- ：进行一次图像采集。
- ：以固定时间间隔连续进行图像采集，再次点击停止图像采集。



结构光相机的工作原理，可查看[结构光相机工作原理](#)。

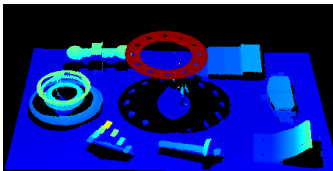
#### 切换数据类型

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

不同型号的相机输出的数据类型不同。



- 非 DEEP (V4) 与 LSR (V4) 系列数据类型为:

数据类型	说明	示例
2D图	用于为点云添加纹理的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	

- DEEP (V4) 与 LSR (V4) 系列数据类型为:

数据类型	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	



2D图 (深度源) 用于手眼标定、内参检查及设置感兴趣区域。

## 调节数据的显示

在数据查看区，你可通过鼠标、键盘、界面中的按钮及菜单，调节数据的显示方式。

请根据数据类型查看相应的操作说明。

### 调节2D图的显示

在2D图显示界面中可进行以下操作：

- **缩放：**光标在2D图上时，滚动鼠标滚轮即可缩放2D图。
- **移动：**2D图放大时，按住鼠标左键并拖拽可移动2D图。
- **重置尺寸：**单击鼠标右键并选择**重置视图**，可将2D图恢复为适应窗口大小的尺寸。
- **查看像素位置及RGB/灰度值：**在2D图右下角的图像信息框中，可查看光标所在像素的像素位置及RGB/灰度值。



- 图像左上角的像素位置为 (0, 0)。
- 要隐藏图像信息框，请取消勾选**视图**菜单中的**图像信息框**。

### 调节深度图的显示

在深度图显示界面中可进行以下操作：

- **缩放：**光标在深度图上时，滚动鼠标滚轮即可缩放深度图。
- **移动：**深度图放大时，按住鼠标左键并拖拽可移动深度图。
- **重置尺寸：**单击鼠标右键并选择**重置视图**，可将深度图恢复为适应窗口大小的尺寸。
- **查看像素位置及坐标：**在深度图右下角的图像信息框中，可查看光标所在像素的像素位置及其在相机坐标系下的坐标（单位为m）。



- 图像左上角的像素位置为 (0, 0)。
- 要隐藏图像信息框，请取消勾选**视图**菜单中的**图像信息框**。

- **调节深度数据的显示：**通过左边栏可调节深度数据的显示，详见[调节深度数据的显示](#)。



可在深度图上显示网格，辅助确定视野中心位置。通过**视图**菜单中的**深度图网格**选项，可选择显示2 × 2或3 × 3的网格。

### 调节点云的显示

在点云显示界面中可进行以下操作：

- **缩放：**光标在点云上时，滚动鼠标滚轮即可缩放点云。
- **旋转：**
  - 在三维空间内旋转：按住鼠标左键并拖拽可旋转点云。

- 在二维平面内旋转：按住 **Ctrl** 的同时，按住鼠标左键并拖拽可旋转点云。
- **平移**：按住鼠标滚轮并拖拽可平移点云。
- **重置视角**：
  - 单击 **R** 键，可重置对视角的平移，同时将缩放过的点云调整为默认大小。
  - 单击 **Backspace** 键，可重置对视角的平移和旋转，同时将缩放过的点云调整为默认大小。
- **显示视锥**：单击鼠标右键并选择**显示视锥**，可以视锥形式显示各个2D相机和投光单元的视野。
- **显示相机坐标系**：单击鼠标右键并选择**显示坐标轴**，可显示相机坐标系的各个坐标轴。
- **调节深度数据的显示**：通过左边栏可调节深度数据的显示，详见[调节深度数据的显示](#)。
- **点云展示**：可使点云以固定轨迹移动，多角度展示点云。详见[点云展示](#)。

### 调节深度数据的显示

在深度图和点云显示界面的左边栏中，可对深度数据的显示坐标系、样式和范围等进行调整。

### 更改坐标系

在左边栏最上方的**坐标系**菜单中，可选择用于显示深度数据的坐标系。

- **设备（默认）**：在相机坐标系下显示深度数据。
- **自定义**：在[用户设置的自定义坐标系](#)下显示深度数据。

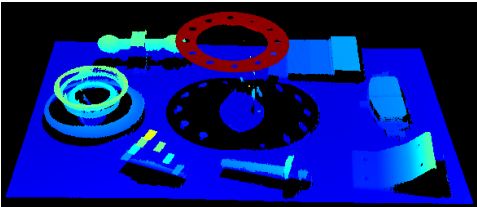
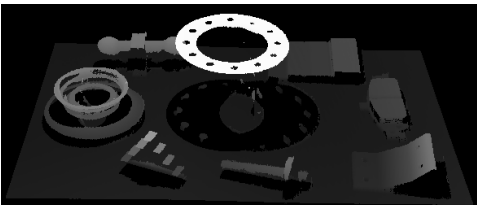


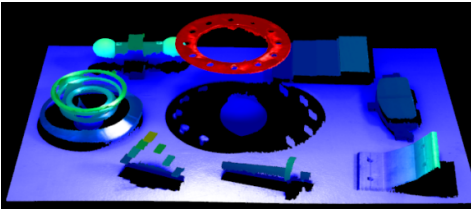
坐标系的设置对深度图和点云同时生效。

### 更改显示样式

在左边栏的**颜色**菜单中，可选择深度数据的显示样式。

**Jet**、**灰度**和**Jet+强度**选项同时适用于深度图和点云。选择这三个选项时，可[调节颜色分布的深度范围](#)。

选项	说明	示例
Jet	不同深度值显示为jet配色方案中的不同颜色。	
灰度	不同深度值显示为不同深浅的灰色。	

选项	说明	示例
Jet+强度	在jet配色方案的基础上叠加物体表面的强度信息，便于同时查看物体表面特征和深度变化。	

无纹理和纹理选项仅适用于点云。

选项	说明	示例
无纹理	显示无纹理的纯白点云。	
纹理	使用2D图为点云添加纹理。	



选择**无纹理**或**纹理**时，如果切换至深度图显示界面，再切换回点云显示界面，**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。

### 调节颜色分布的深度范围

以**Jet**、**灰度**或**Jet+强度**样式显示深度图和点云时，每次采集图像，Mech-Eye Viewer将根据获取的深度数据范围自动调整颜色分布。

通过左边栏**颜色范围**区中的功能，可以调节颜色分布的深度范围，便于观察某一范围内深度数据的变化。



颜色分布的深度范围对深度图和点云同时生效。

#### 适用场景示例：

- 观察工件表面某区域的深度变化，检查该区域是否平整。
- 使某一工件的显示颜色更鲜艳、突出。

请执行以下步骤调节颜色分布的深度范围：

1. 调节**参数**标签页中的**深度范围**参数：左边栏滑动条的**最小值**和**最大值**对应**深度范围**参数。可先将该参数调节至一个合适的范围，排除不需要的深度数据。
2. 确认大致深度：在深度图上，将光标移动至需要观察的区域，查看右下角的图片信息框中**坐标**

的第三个值，即深度值。

3. 调节深度范围：移动滑块或在滑动条右侧输入深度值，根据需求调节颜色分布的深度范围。



- 使用UHP系列相机时，可在滑动条右侧将单位更改为 $\mu\text{m}$ 。
- 调节范围后，范围外的深度值将以最大/最小深度值对应的颜色显示。

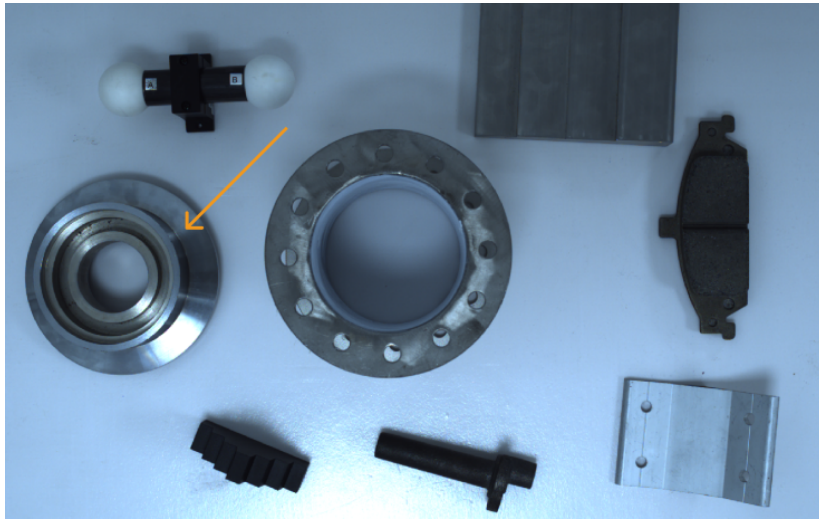
4. 锁定深度范围：如需比较多次采集结果的同一范围内的深度变化，可勾选滑动条上方的**锁定**。再次采集图像时，将继续使用手动调节过的深度范围。



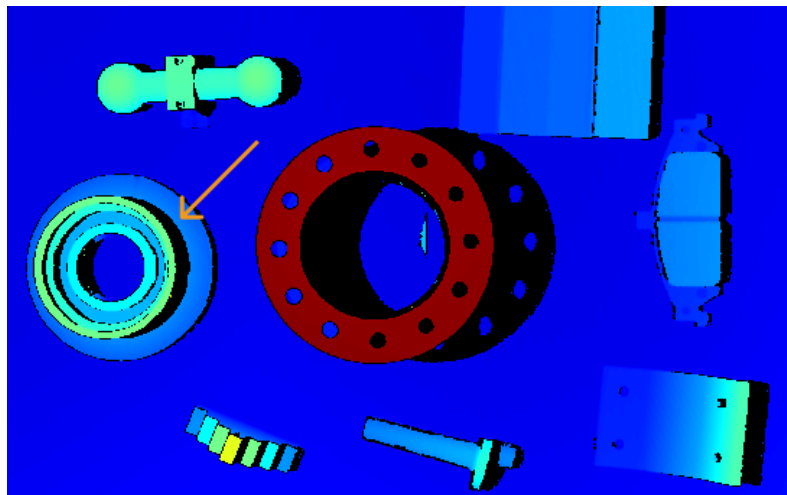
如需恢复至自动调整的深度范围，取消勾选**锁定**，并单击其右上方的[复位]。

#### ▼适用场景1的调节示例

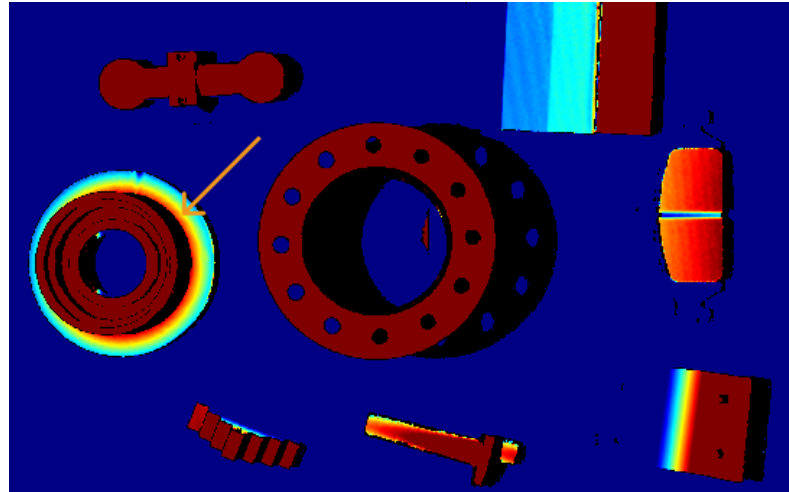
目标：检查箭头所指的表面是否平整。



1. 查看深度图，该平面几乎为同一颜色，不易判断是否完全平整。

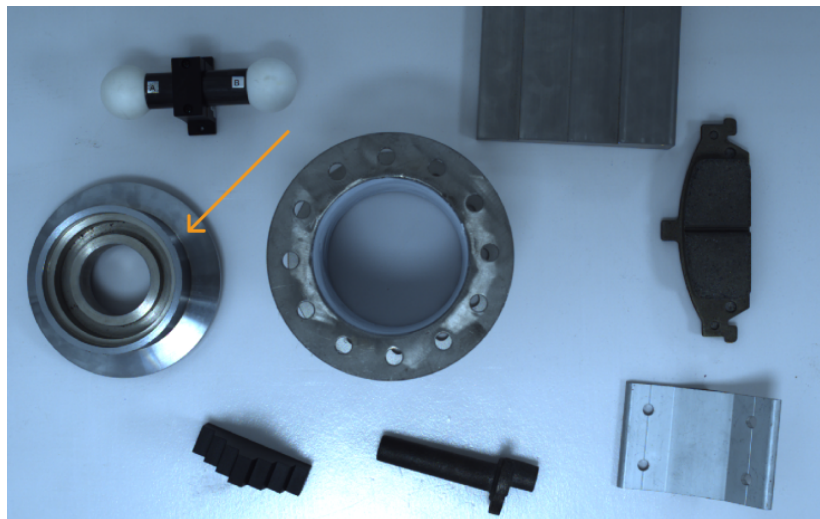


2. 将光标移动至该平面上，查看右下角图像信息框中的深度值（约0.829m）。
3. 调整左边栏滑动条的滑块或输入深度值，将颜色分布的深度范围调整至825~835mm。
4. 再次查看深度图，可通过颜色轻易观察到该平面的深度变化。

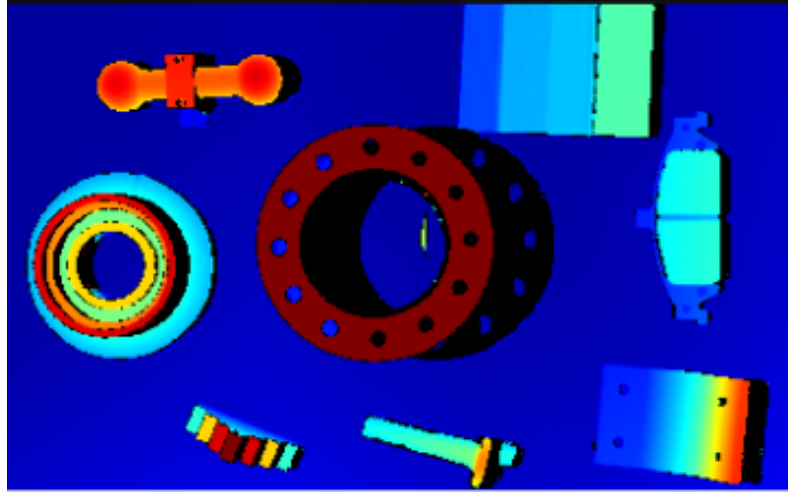


### ▼适用场景2的调节示例

目标：以更丰富的颜色突出展示箭头所指的工件。



1. 在深度图上，将光标移动至该工件最高和最低的位置上，查看右下角图像信息框中的深度值（约0.795m和0.829m）。
2. 调整左边栏滑动条的滑块或输入深度值，将颜色分布的深度范围调整至790~850mm。
3. 再次查看深度图，此时该工件的颜色较丰富，达到突出展示的目的。



### 点云展示

该功能可使点云以固定轨迹移动，多角度展示点云，适用于展会等场景。

请执行以下步骤使用**点云展示**功能：

1. 勾选**视图**菜单中的**点云展示**，点云显示界面右上角出现[ **播放** ]按钮。
2. 单击[ **播放** ]，点云开始按照固定轨迹移动。
3. 播放过程中，[ **播放** ]按钮变为[ **停止** ]。单击[ **停止** ]，点云停止移动。

图像采集和数据类型的介绍至此全部完成，下一章将介绍参数相关的操作。

### 5.2.3. 调节参数

本章介绍参数相关的操作，包括调节参数的基本流程、管理参数组、更改参数可见级别和查看参数说明。

#### 调节参数的基本流程

调节参数时，往往需要重新采集图像，确认参数调节后产生的效果。调节参数的基本流程如下：

1. 在右侧的**参数**标签页最上方**创建参数组**，或选择用于保存参数值的参数组。



可根据实际场景选择合适的预置参数组，简化调参流程。详细说明请查看[预置参数组](#)。

2. 单击  采集图像，确认数据质量。

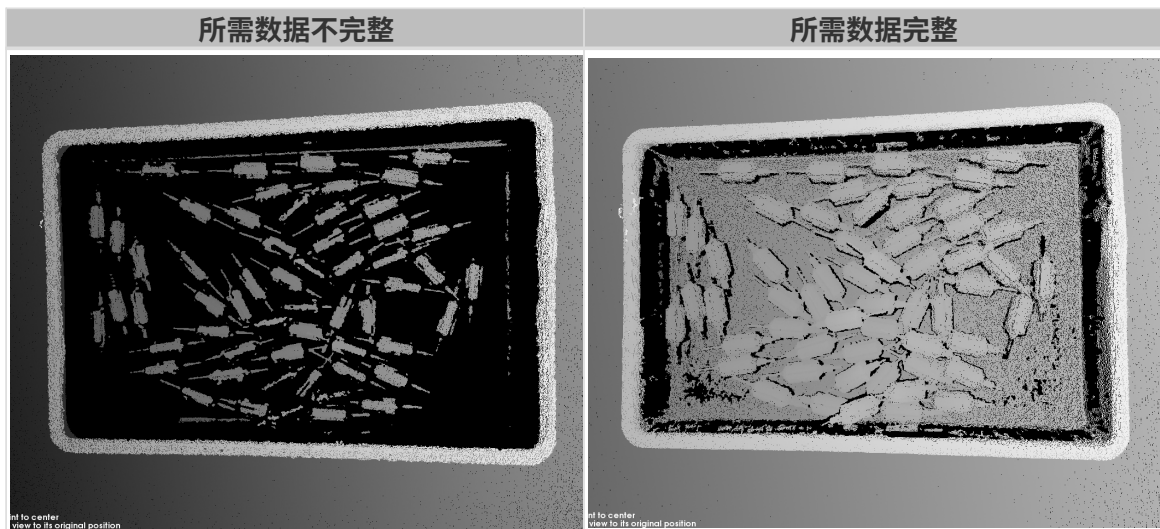
- 2D图：应不过亮或过暗，可看清目标物体的表面特征。



- 深度图和点云：所需的数据应完整。以下示例中，所需数据为料筐上沿及筐内的转子。



根据实际需求判断所需数据的范围。例如，需通过碗沿抓取碗口向上摆放的金属碗时，一般只需确保碗沿部分的数据完整。



### 3. 如数据质量不佳，在参数标签页中调节对应的参数。

- 如2D图质量不佳，请调节**2D参数**分组下的参数。
- 如深度图质量不佳，请调节**3D参数**、**深度范围**和**感兴趣区域**分组下的参数。
- 如点云质量不佳，请调节**3D参数**、**点云后处理**、**深度范围**和**感兴趣区域**分组下的参数。



- 调节参数时，可参考参数标签页下方的[参数说明](#)。
- 各参数的详细说明，请查看[参数参考指南](#)。

### 4. 重复步骤2和3，直至数据质量满足要求后，单击参数标签页最上方的[保存]。



- **点云后处理**、**深度范围**和**感兴趣区域**分组下的参数，调节后立即生效，无需重新采集图像。
- 如数据质量仍不佳，可[更改可见级别](#)，然后调节新增的参数。

## 管理参数组


参数组可保存用于不同场景、工程、工件等的参数值。场景、工程、工件等发生变化时，切换所用的参数组即可快速应用对应的参数值。




在Mech-Eye Viewer中，你可以创建新的参数组、将参数值保存至参数组、删除参数组、重命名参数组以及导入和导出参数组。

### 创建参数组

请执行以下步骤，以当前的参数设置创建参数组：

1. 在参数组区右侧，单击.
2. 在弹出的窗口中输入参数组名，并单击[确定]。


 参数组名长度不可超过32个字符。

3. 创建完成后，参数组区的下拉菜单中显示新增的参数组。

### 保存参数至参数组


请执行以下步骤，将参数保存至参数组：


1. 单击参数组区的下拉菜单，选择用于保存参数值的参数组。
2. [调节参数](#)。
3. 在参数组区，单击[保存]或使用快捷键`Ctrl+S`，将修改过的参数值保存至当前选中的参数组中。

 参数后面带星号 (\*) 时，表示该参数的值未保存至参数组。相机断电后，未保存的更改将被丢弃。

### 删除参数组


如需删除不再使用的参数组，请执行以下步骤：


1. 单击参数组区的下拉菜单，选择要删除的参数组。
2. 单击，并单击弹窗中的[确定]，删除参数组。

 预置的参数组不可删除。

### 重命名参数组

请执行以下步骤重命名参数组：

1. 单击参数组区的下拉菜单，选择要重命名的参数组。
2. 单击，在弹出的窗口中输入新名称，并单击[确定]。

 参数组名长度不可超过32个字符。

3. 修改完成后，参数组区的下拉菜单中显示修改后的名称。

 预置的参数组不可重命名。

## 导出参数组

可将当前相机中保存的所有参数组导出至json文件中。请执行以下步骤导出全部参数组：

1. 在参数组区单击[ **导出** ]，选择保存路径，单击[ **保存** ]。
2. 如参数组导出成功，将出现**参数组导出成功**弹窗。单击弹窗中的[ **确定** ]关闭弹窗。

## 导入参数组

可从json文件导入并替换当前相机中保存的所有参数组。请执行以下步骤导入参数组：

1. 在参数组区单击[ **导入** ]，找到并选中存有参数组信息的json文件，单击[ **打开** ]。
2. 如参数组导入成功，将出现**参数组导入成功**弹窗。单击弹窗中的[ **确定** ]关闭弹窗。

## 重置参数组

如需将参数组中保存的参数值重置为默认值，请执行以下步骤：

1. 单击参数组区的下拉菜单，选择要重置的参数组。
2. 单击[ **重置** ]，并在弹出的窗口中单击[ **是** ]，将参数值重置为与**default**参数组一致的值。

## 更改可见级别

Mech-Eye Viewer中的参数分为三个可见级别：**初级**、**专家**和**大师**。从**可见级别**的下拉菜单中选择，即可更改可见级别。

- **初级**级别包含最常用的基础参数。
- **专家**级别包含**初级**级别的全部参数，并提供更多参数。
- **大师**级别包含**专家**级别的全部参数，并提供更多参数。

通常，调节**初级**参数即可满足需求。如数据质量仍不理想，可调节**专家**级别的参数。在个别情况下，需要调节**大师**级别的参数以满足特殊需求。



**大师**级别在**管理员**用户下可见。如需使用，请联系技术支持。

## 查看参数说明

在参数列表中选中参数后，**参数**标签页最下方将显示该参数的更多信息，包括参数解释、API参数名、值范围、默认值、可见级别与数据类型。调节参数时，可以参考此处的信息。


- 如参数说明区遮挡了参数列表，可点击该区域右上角的[ **×** ]，关闭参数说明区。
- 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。

## 5.2.4. 保存数据

通过该功能，可将采集的2D图、深度图和点云保存至本地。

### 操作步骤

请执行以下步骤保存数据：

1. 单击数据采集区中的 ，打开**保存数据**窗口。
2. 勾选**数据类型**下需要保存的数据类型。详细说明请参考[数据类型及格式](#)。
3. （可选）修改**文件序号**：如所选择的**保存路径**下已有通过Mech-Eye Viewer保存的数据，可修改文件序号，避免同名文件被替换。



如需替换同名文件，可勾选下方的**替换同名文件**选项。

4. 单击**保存路径**右侧的[... ]，选择用于保存数据的文件夹。
5. 单击[ **保存** ]，将弹出**数据保存成功**的弹窗。单击该弹窗中的[ **在文件夹中显示** ]，可打开保存数据所在的文件夹，查看已保存的数据。

### 数据类型及格式

保存数据时，可选择要保存的文件格式、点云的数据结构和颜色。

- 2D图：可以保存为**PNG** (.png) 或**JPEG** (.jpg) 格式。
- 深度图：可以保存为**PNG** (.png) 或**TIFF** (.tiff) 格式。
- 点云：可选择保存的格式、数据结构和颜色。

格式	<ul style="list-style-type: none"> <li>• <b>PLY</b> (.ply)</li> <li>• <b>PCD</b> (.pcd)</li> <li>• <b>CSV</b> (.csv)</li> </ul>
数据结构	<ul style="list-style-type: none"> <li>• 有序：以二维矩阵的形式储存点，数据结构类似点与点之间的实际空间关系。</li> <li>• 无序：以一维列表的形式储存点。</li> </ul>
颜色	<ul style="list-style-type: none"> <li>• Jet</li> <li>• 灰度</li> <li>• Jet+强度</li> <li>• 无纹理</li> <li>• 纹理</li> </ul>

## 5.2.5. 管理日志

使用相机遇到问题时，可查看日志，或将日志导出并发送给技术支持，辅助排查问题。

## 查看日志

请执行以下步骤查看日志：

1. 单击数据采集区中的[ **显示日志** ]，打开相机中保存的日志。
2. 单击右下角的[ **显示日志列表** ]，显示完整日志列表。
3. 选中**日志列表**中的一个日志条目，即可在左侧查看日志内容。



日志条目的名称代表该条目生成的时间。例如，名为**20221117171503\_887**的日志条目于2022年11月17日17时15分03秒生成。

## 导出日志

请执行以下步骤导出日志：

1. 单击数据采集区中的[ **显示日志** ]，打开相机中保存的日志。
2. 单击右下角的[ **显示日志列表** ]，显示完整日志列表。
3. 选中需要导出的日志条目，单击[ **导出** ]，在弹窗中选择保存日志的文件夹，并单击[ **选择文件夹** ]。
4. 如需导出多个日志条目，重复步骤3。

## 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。

## 其他操作

日志界面还可进行以下操作：

- 自动刷新：自动更新日志。
- 清空：清空软件中显示的日志内容。
- 清除服务端日志：删除相机中保存的日志。将同时清空软件中显示的日志内容。

## 5.3. 参数参考指南

本章提供各参数的详细说明。

查看以下内容，了解各型号参数的详细说明。

- [NANO参数](#)
- [PRO S和PRO M参数](#)
- [PRO XS参数](#)
- [UHP-140参数](#)
- [DEEP / LSR系列参数](#)
- [V3相机参数](#)

### 5.3.1. NANO参数

本章介绍NANO相机的参数。参数按照影响的数据类型拆分为2D图参数及深度图与点云参数。

#### 2D图参数

2D图应不过亮或过暗，可看清目标物体的表面特征。

**2D参数**分组下的参数及**3D参数**分组下的**相机增益**影响2D图质量。



彩色相机采集图像时，如因现场光照条件导致图像颜色与实际差别较大，请调节**白平衡**。详细操作请参考[调节白平衡](#)。

#### 2D参数

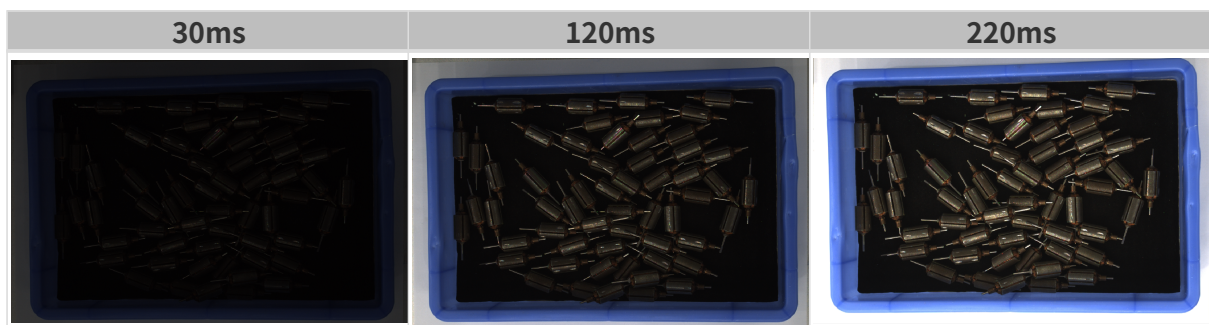
##### 曝光模式

<b>参数说明</b>	设置拍摄2D图时的曝光模式。
<b>可见级别</b>	初级、专家、大师
<b>参数取值</b>	<ul style="list-style-type: none"> <li>• 固定曝光：设置固定曝光时间，常用于稳定的光照条件。</li> <li>• 自动曝光：自动调整曝光时间，常用于多变的光照条件。</li> <li>• HDR：设置多个曝光时间并融合所有图像，常用于颜色或材质多样的物体。</li> <li>• Flash：使用投影机补光，常用于较暗的环境。</li> </ul>
<b>调节说明</b>	选择不同选项后， <b>2D参数</b> 分组中将显示不同的参数供调节： <ul style="list-style-type: none"> <li>• 固定曝光：显示<b>曝光时间</b>。</li> <li>• 自动曝光：显示<b>灰度值</b>与<b>自动曝光感兴趣区域</b>。</li> <li>• HDR：显示<b>色调映射</b>与<b>曝光时间序列</b>。</li> <li>• Flash：无需调节其他参数。采集图像时将自动补光。</li> </ul>

### 固定曝光：曝光时间

	影响图像亮度。
参数说明	<ul style="list-style-type: none"> <li>曝光时间越长，图像亮度越高。</li> <li>曝光时间越短，图像亮度越低。</li> </ul>
可见级别	初级、专家、大师
参数取值	0.1~999ms
调节说明	根据2D图的质量调节。2D图应不过亮或过暗，可看清目标物体的表面特征。 <ul style="list-style-type: none"> <li>较暗的环境下通常使用较长曝光时间。</li> <li>较亮的环境下通常使用较短曝光时间。</li> </ul>

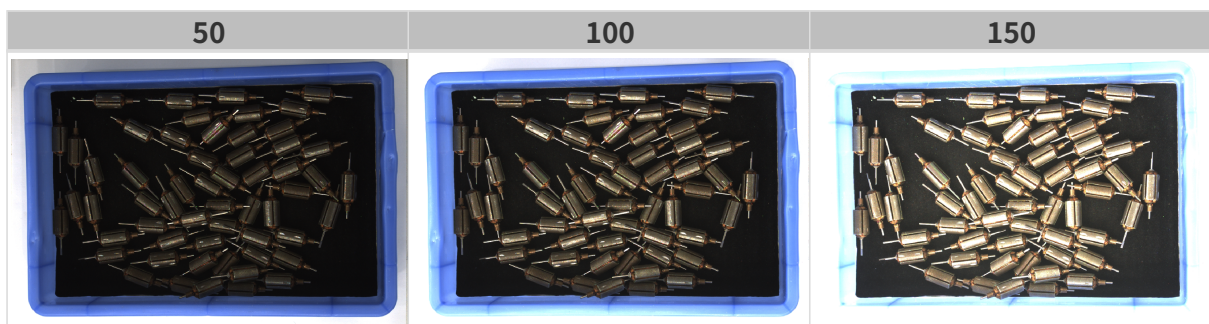
其他条件相同，仅**曝光时间**不同的2D图对比如下：



### 自动曝光：灰度值

参数说明	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。
可见级别	初级、专家、大师
参数取值	0~255
调节说明	无

其他条件相同，仅**灰度值**不同的2D图对比如下：

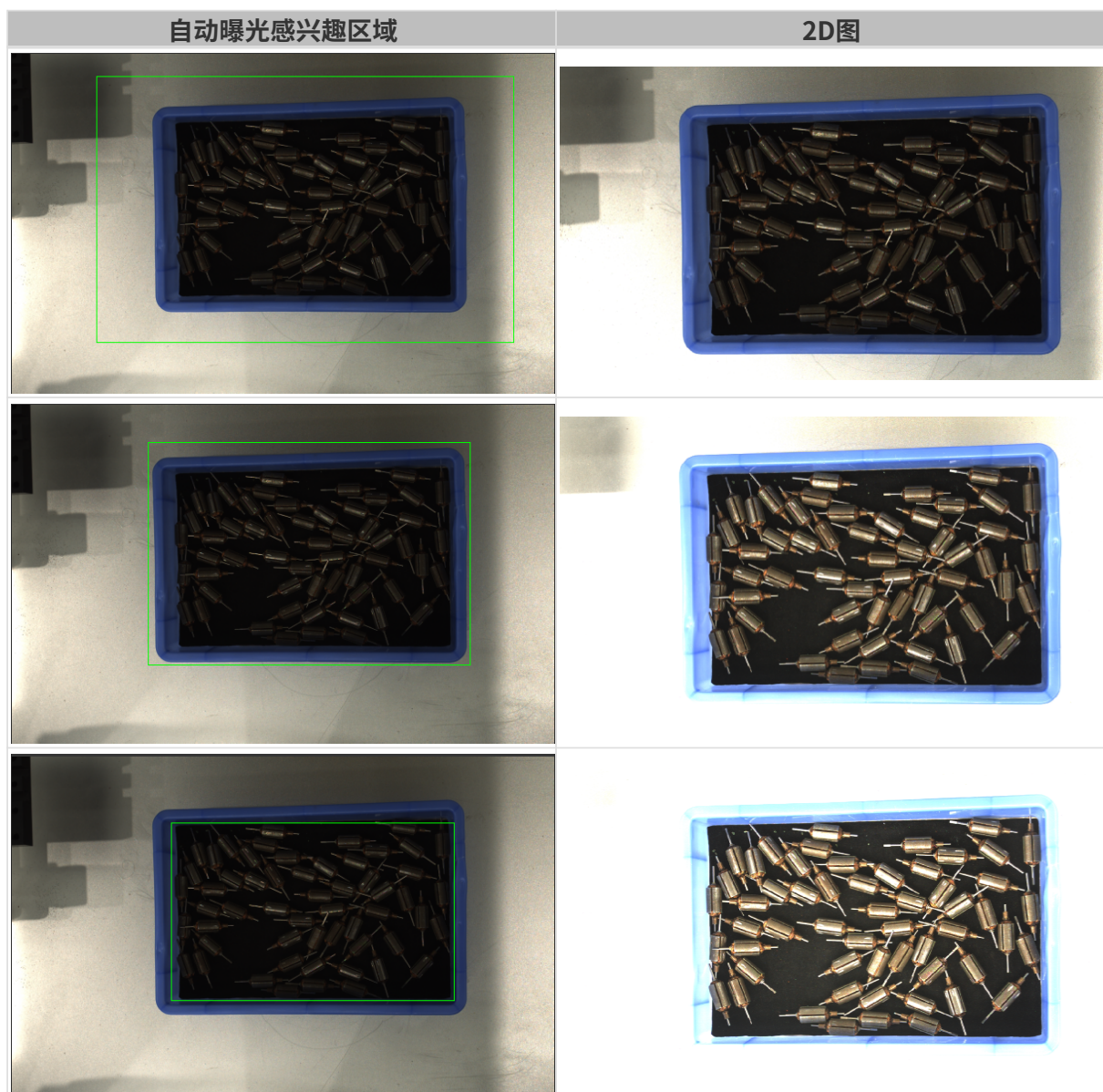


黑白图像的**灰度值**相当于图像亮度；彩色图像的**灰度值**相当于每个颜色通道的亮度。

### 自动曝光：自动曝光感兴趣区域

参数说明	<ul style="list-style-type: none"> <li>相机参照该区域内的光照、物体颜色等自动调整曝光时间。</li> <li>未设置感兴趣区域时，相机根据整个视野自动调整曝光时间。</li> </ul>
可见级别	初级、专家、大师
参数取值	无
调节说明	详见下方 <a href="#">设置自动曝光感兴趣区域</a> 。

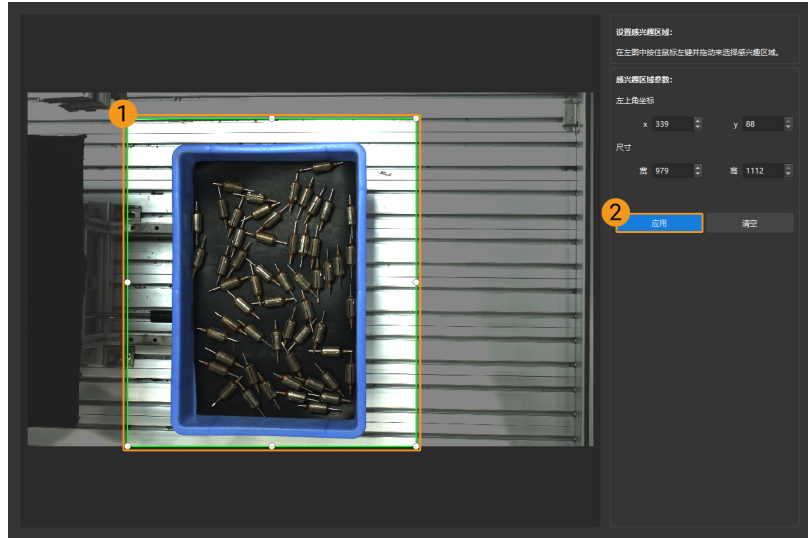
其他条件相同，仅自动曝光感兴趣区域不同的2D图对比如下：



### 设置自动曝光感兴趣区域

1. 双击自动曝光感兴趣区域右侧的[编辑]，进入设置感兴趣区域页面。
2. 在左侧选择并调节感兴趣区域。拖拽选框可调节位置，拖拽白色锚点可调节大小。

3. 单击[ 应用 ], 应用设置的自动曝光感兴趣区域。



**i** 单击[ 清空 ], 可清除当前设置的自动曝光感兴趣区域。

4. 重新采集图像并查看2D图, 确认自动曝光的效果。

### HDR: 色调映射

参数说明	可使图像看起来更自然。如2D图和真实物体相差较大, 可勾选该参数。
可见级别	初级、专家、大师
参数取值	<ul style="list-style-type: none"> <li>False</li> <li>True</li> </ul>
调节说明	勾选即可开启色调映射。


色调映射开启前后的2D图对比如下:



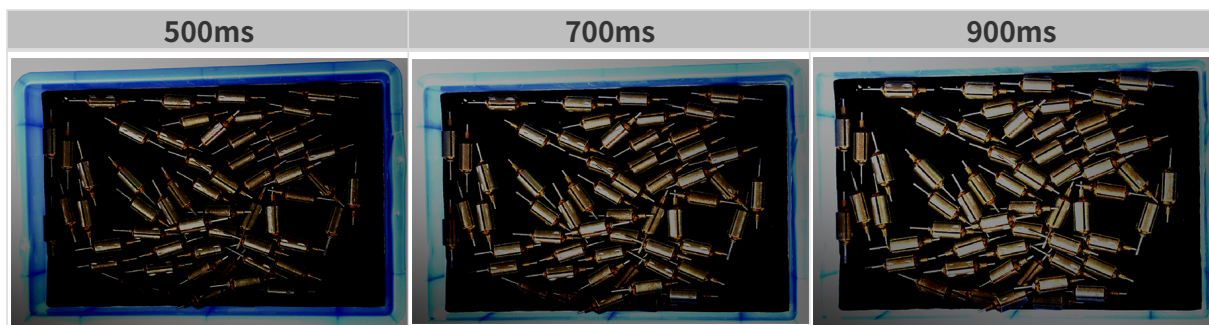
### HDR: 曝光时间序列

参数说明	设置多个曝光时间, 并将所拍摄的图像融合为一张保留更多明暗细节的2D图。
------	--------------------------------------

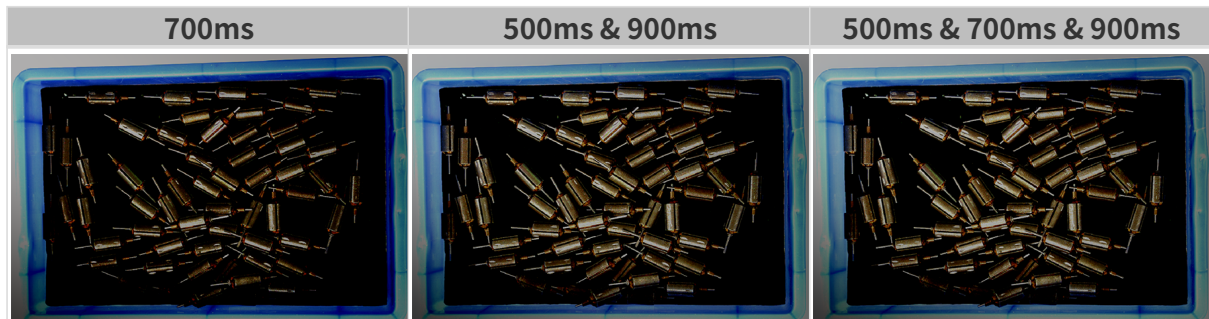


可见级别	初级、专家、大师
参数取值	无
调节说明	<ol style="list-style-type: none"> <li>1. 双击<b>曝光时间序列</b>右侧的[编辑]，进入曝光时间序列的编辑界面。</li> <li>2. 单击[+]，新增并设置曝光时间。</li> <li>3. 如需删除某个曝光时间，选中该曝光时间并单击[-]。</li> <li>4. 编辑完成后，单击[应用]使曝光时间序列生效。</li> </ol> <div style="display: flex; align-items: center; margin-top: 10px;"> <div style="margin-right: 10px;">  </div> <ul style="list-style-type: none"> <li>◦ [取消]: 不保存当前设置，直接退出该页面。</li> <li>◦ [重置]: 清空所有的曝光时间。</li> </ul> </div>

设置单个曝光时间时，不同曝光时间的2D图对比如下：




使用以上曝光时间组合成不同的曝光时间序列时，不同序列的2D图对比如下：

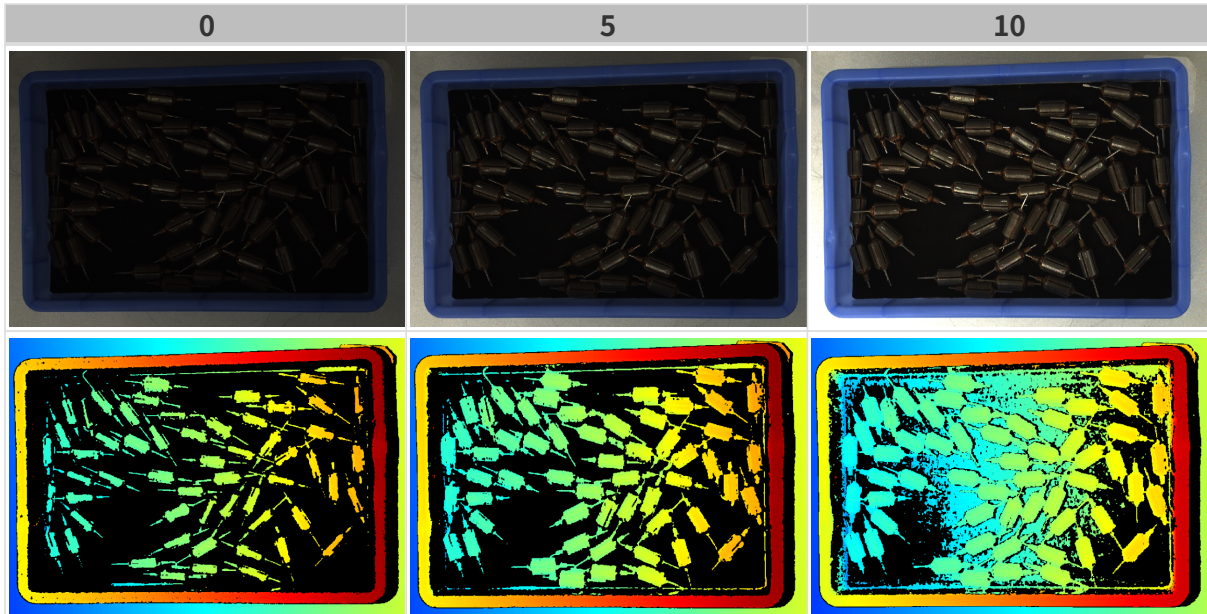


### 3D参数

#### 相机增益

参数说明	用于增加图像亮度，但可能会引入噪点。 <div style="display: flex; align-items: center; margin-top: 10px;"> <div style="margin-right: 10px;">  </div> <p>该参数通过改变用于计算深度数据的图像的亮度影响深度图和点云质量。</p> </div>
可见级别	专家、大师
参数取值	0~16dB
调节说明	当设置 <b>曝光时间</b> 无法达到期望亮度时，可以使用该参数。

其他条件相同，仅**相机增益**不同的2D图和深度图对比如下：



### 深度图与点云参数

深度图与点云中，所需的数据应完整。



根据实际需求判断所需数据的范围。例如，需通过碗沿抓取碗口向上摆放的金属碗时，一般只需确保碗沿部分的数据完整。

以下分组中的参数影响深度图与点云质量。

参数分组	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

该分组下的参数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。

使用**曝光助手**可获得推荐的曝光参数组合。双击**3D参数**右侧的[自动设置]，即可打开**曝光助手**。

### 曝光次数

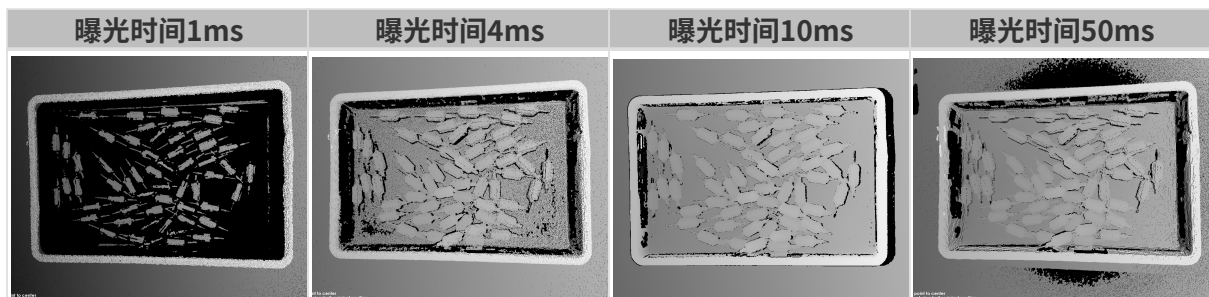
参数说明	设置 <b>曝光时间</b> 的个数。
可见级别	初级、专家、大师
参数取值	1~3

调节说明	<ul style="list-style-type: none"> <li>曝光次数大于1时，需设置多个<b>曝光时间</b>。</li> <li>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，但也将延长处理时间。</li> <li>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。</li> </ul>
------	---

## 曝光时间

参数说明	设置采集深度信息时的曝光时间，需设置的 <b>曝光时间</b> 个数取决于 <b>曝光次数</b> 的值。
可见级别	初级、专家、大师
参数取值	<ul style="list-style-type: none"> <li>取值范围：0.1~99ms</li> </ul>
调节说明	<ul style="list-style-type: none"> <li>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。</li> <li>曝光时间过长或过短都会导致信息缺失。</li> <li>激光相机的曝光时间须为4的倍数，输入值不为4的倍数时将被自动调整。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。</li> </ul>

不同**曝光时间**对比图如下：

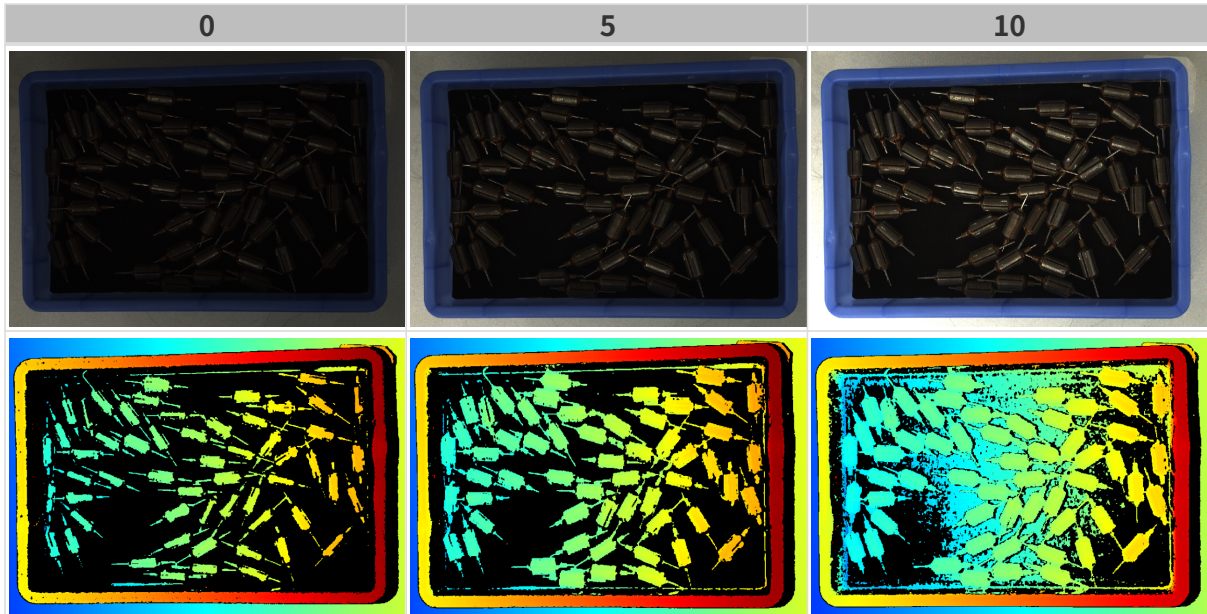


黑色部分缺少对应的物体点云。

## 相机增益

参数说明	用于增加图像亮度，但可能会引入噪点。
可见级别	专家、大师
参数取值	0~16dB
调节说明	当设置 <b>曝光时间</b> 无法达到期望亮度时，可以使用该参数。

其他条件相同，仅**相机增益**不同的2D图和深度图对比如下：



## 投影

### 投影光亮度

参数说明	投影仪投射的结构光亮度。
可见级别	专家、大师
参数取值	<ul style="list-style-type: none"> <li>• High: 高亮度, 适用于深色物体。</li> <li>• Normal: 正常亮度, 适用于普通物体。</li> <li>• Low: 低亮度, 适用于反光物体。</li> </ul>
调节说明	根据物体类型调节。

### 编码模式

参数说明	选择投射的结构光的样式。
可见级别	专家、大师
参数取值	<ul style="list-style-type: none"> <li>• Fast: 适用于不透明的物体, 采集速度快, 但深度数据质量稍差。</li> <li>• Accurate: 适用于不透明的物体, 深度数据质量高, 但采集速度较慢。</li> </ul>
调节说明	根据目标物体类型以及对数据质量和采集速度的实际需求调节。

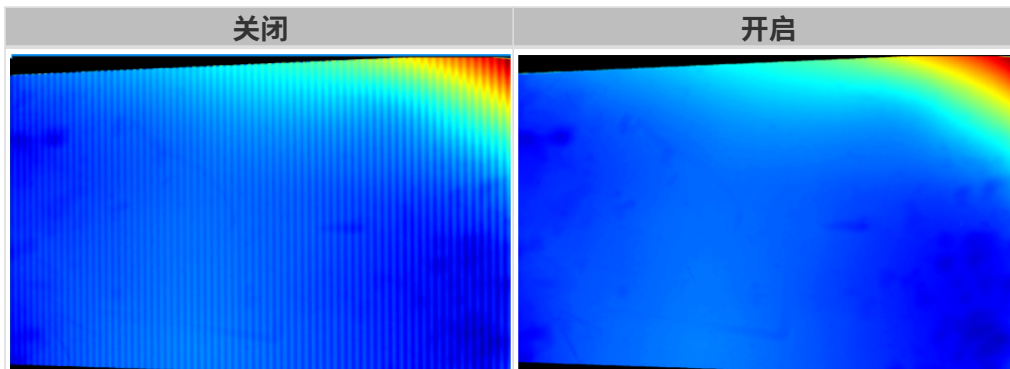
目标物体为非透明物体时, 其他条件相同, **编码模式**分别选择**Accurate**和**Fast**时获取的点云如下:



### 抗频闪模式

参数说明	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。
可见级别	专家、大师
参数取值	<ul style="list-style-type: none"> <li>• Off</li> <li>• AC50Hz</li> <li>• AC60Hz</li> </ul>
调节说明	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。

抗频闪模式开启前后的深度图对比如下：



### 点云后处理

调节点云后处理分组下的参数，可提升点云质量。

#### 调参原则

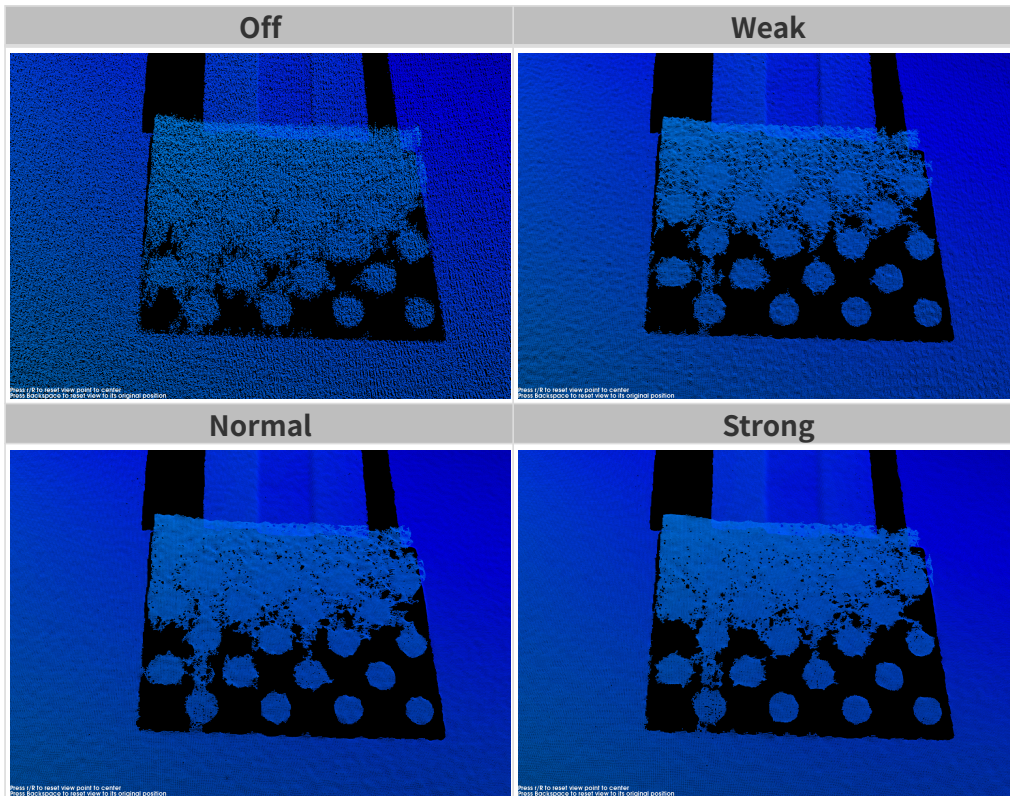
调节点云后处理参数时，遵循以下调参原则可减少相机采集时间，优化节拍。

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

<b>参数说明</b>	可减少点云的深度波动，使点云更接近真实的物体表面，但会损失部分物体表面细节。
<b>可见级别</b>	初级、专家、大师
<b>参数取值</b>	<ul style="list-style-type: none"> <li>• Off</li> <li>• Weak</li> <li>• Normal</li> <li>• Strong</li> </ul>
<b>调节说明</b>	<ul style="list-style-type: none"> <li>• 表面平滑强度越高，损失的物体表面细节越多；强度越低，损失的物体表面细节越少。</li> <li>• 表面平滑强度越高，计算时间越长；强度越低，计算时间越短。</li> </ul>

其他条件相同，仅表面平滑强度不同的点云对比如下：

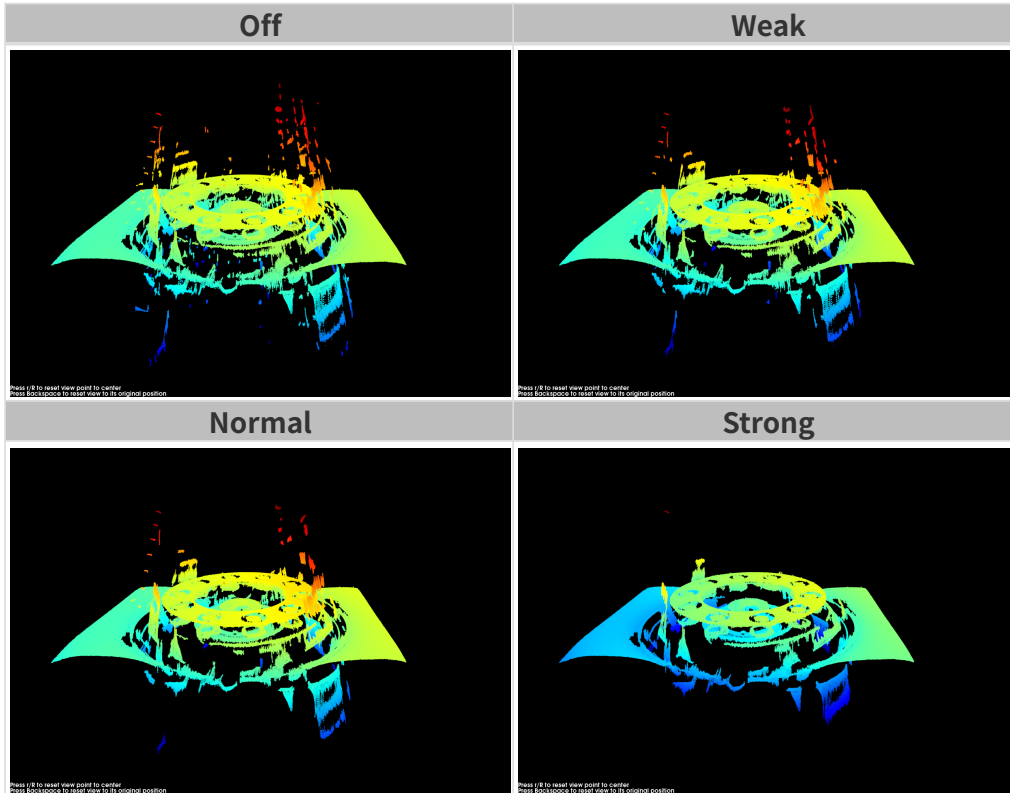


## 离群点去除

<b>参数说明</b>	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。
<b>可见级别</b>	初级、专家、大师
<b>参数取值</b>	<ul style="list-style-type: none"> <li>• Off</li> <li>• Weak</li> <li>• Normal</li> <li>• Strong</li> </ul>

<b>调节说明</b>	<ul style="list-style-type: none"> <li>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。</li> <li>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。</li> </ul>
-------------	---

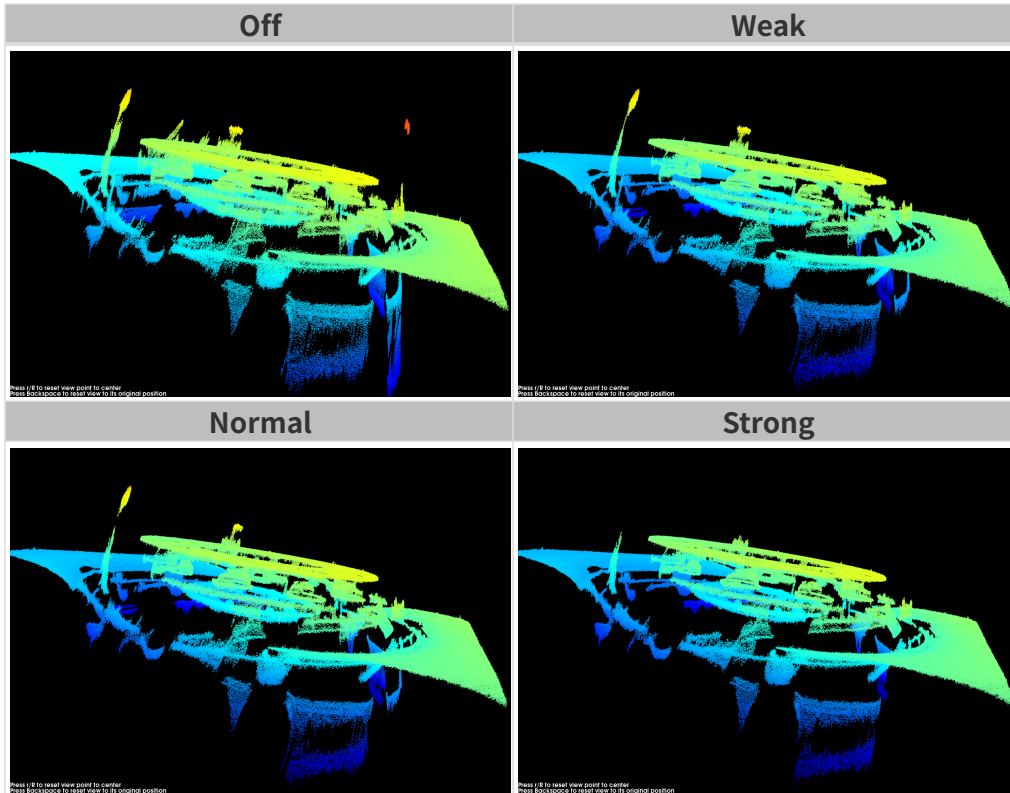
其他条件相同，仅离群点去除强度不同的点云对比如下：



### 噪点去除

<b>参数说明</b>	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。
<b>可见级别</b>	初级、专家、大师
<b>参数取值</b>	<ul style="list-style-type: none"> <li>Off</li> <li>Weak</li> <li>Normal</li> <li>Strong</li> </ul>
<b>调节说明</b>	<ul style="list-style-type: none"> <li>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。</li> <li>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。</li> </ul>

其他条件相同，仅噪点去除强度不同的点云对比如下：



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

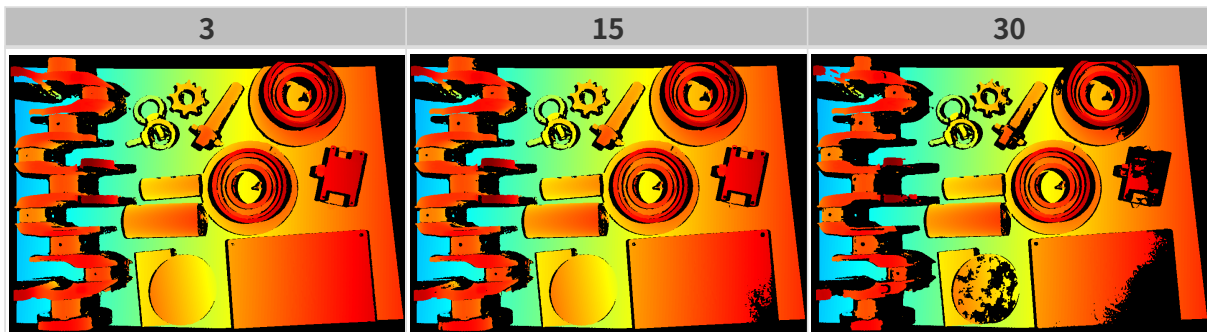
参数说明	在进行表面平滑时保持物体边缘的锐利度。
可见级别	大师
参数取值	<ul style="list-style-type: none"> <li>Sharp: 最大程度保持物体边缘的锐利度，但表面平滑的效果较差。</li> <li>Normal: 在保持边缘的同时达到较好的表面平滑效果。</li> <li>Smooth: 不进行边缘保持。表面平滑效果最好，但物体边缘会失真。</li> </ul>
调节说明	根据对工件边缘特征的需求调节。

## 条纹对比度阈值

参数说明	用于去除点云中的噪点。如调节 <b>离群点去除</b> 与 <b>噪点去除</b> 后，仍无法获得需要的点云，再调节该参数。
可见级别	初级、专家、大师
参数取值	1~100
调节说明	<ul style="list-style-type: none"> <li>该参数值越大，去除的点越多；值越小，去除的点越少。</li> <li>调大该参数可以去除点云中的噪点，但也可能造成较暗物体对应的点丢失。</li> </ul>

其他条件相同，仅**条纹对比度阈值**不同的点云对比如下：

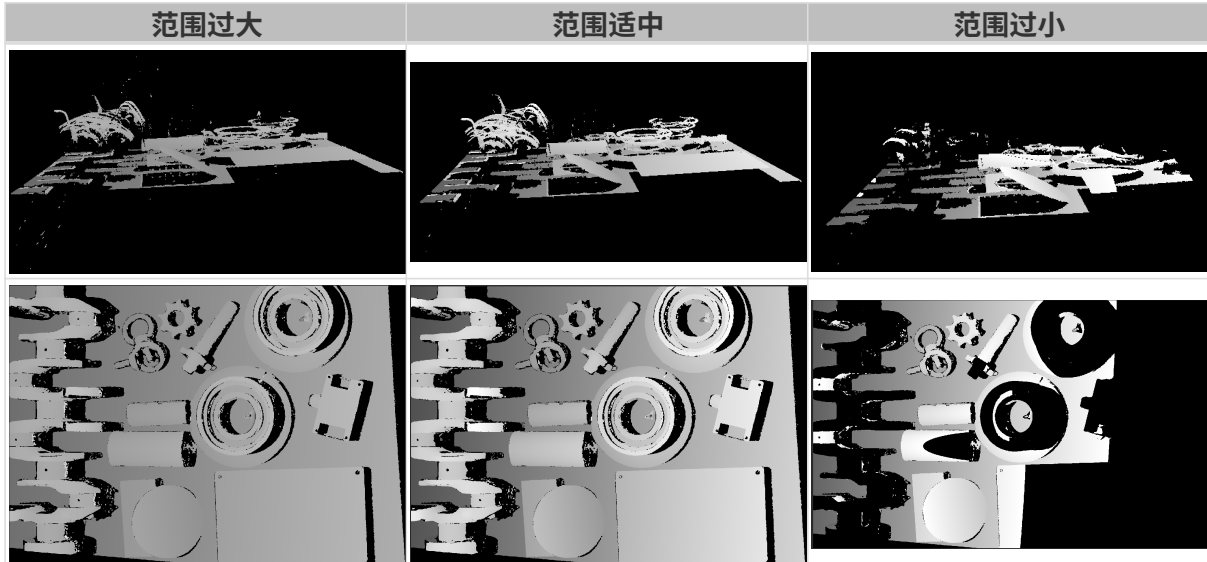




### 深度范围

参数说明	设置Z向感兴趣区域。在相机的工作距离范围内设置 <b>深度范围</b> ，可滤除 <b>深度范围</b> 外的数据。
可见级别	初级、专家、大师
参数取值	<ul style="list-style-type: none"> <li>• 下限：1~4000mm</li> <li>• 上限：1~5000mm</li> </ul>
调节说明	<ul style="list-style-type: none"> <li>• <b>深度范围</b>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。</li> <li>• 如何设置详见下方的<a href="#">设置深度范围</a>。</li> </ul>

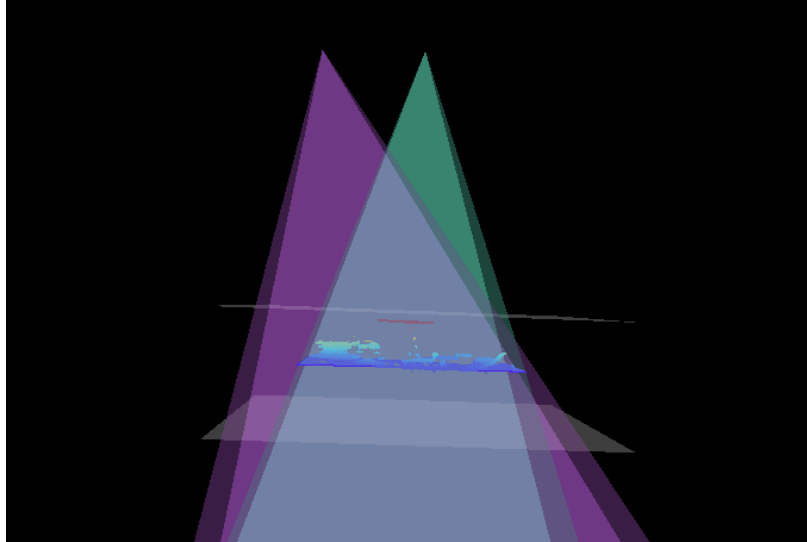
不同**深度范围**效果对比：



### 设置深度范围

执行以下步骤调节**深度范围**：

1. 双击**深度范围**右侧的[编辑]，打开设定**深度范围**界面。
2. 点击右边栏最上方的[更新点云]，获取最新的点云。
3. **调整点云的位置**：调整到可看到代表**深度范围**上下限的两个灰色长方形即可。



4. 调节深度范围：拖动右侧滑动条上的滑块，大致调节**深度范围**。再输入数值，精准调节**深度范围**。



判断深度范围是否合适：所有必需的物体特征均位于两个灰色长方形之间，大多数噪点和离群点位于该区域之外。

5. 设置完成后，单击右下角的[ **保存** ]。



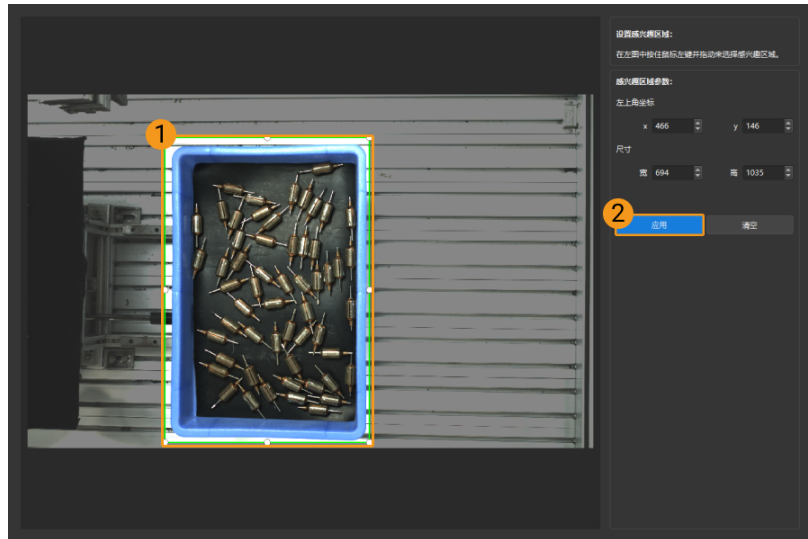
- 单击[ **推荐值** ]，可将**深度范围**调整为当前相机的推荐工作距离。
- 单击[ **重置** ]，可将**深度范围**恢复为上次保存的数值。

## 感兴趣区域

参数说明	设置深度图和点云在XOY平面上的感兴趣区域，所选区域外的点将被移除。
可见级别	初级、专家、大师
参数取值	无
调节说明	详见下方 <a href="#">设置感兴趣区域</a> 。

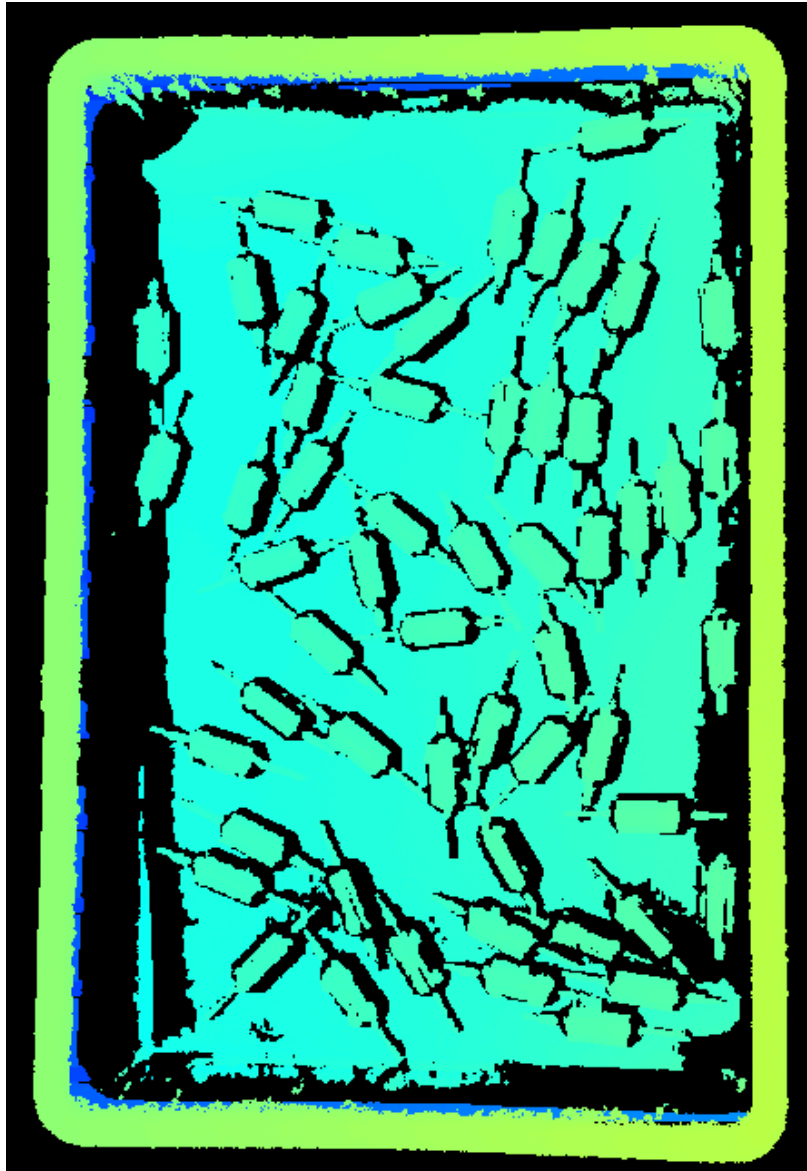
## 设置感兴趣区域

1. 双击**感兴趣区域**右侧的[ **编辑** ]，进入**设置感兴趣区域**页面。
2. 在左侧选择并调节感兴趣区域。拖拽选框可调节位置，拖拽白色锚点可调节大小。
3. 单击[ **应用** ]，应用设置的感兴趣区域。



- 单击[ 清空 ], 可清除当前设置的感兴趣区域。
- DEEP (V4) 与LSR (V4) 系列, 在此界面显示的是2D图 (深度源)。如果图像过暗或过亮, 请调节**2D图 (深度源) 曝光模式**。

4. 重新采集图像, 查看深度图或点云, 确认所设感兴趣区域的效果。



### 5.3.2. PRO S和PRO M参数

本章介绍PRO S和PRO M相机的参数。参数按照影响的数据类型拆分为2D图参数及深度图与点云参数。

#### 2D图参数

2D图应不过亮或过暗，可看清目标物体的表面特征。

2D参数分组下的参数及3D参数分组下的相机增益影响2D图质量。



彩色相机采集图像时，如因现场光照条件导致图像颜色与实际差别较大，请调节白平衡。详细操作请参考调节白平衡。

#### 2D参数

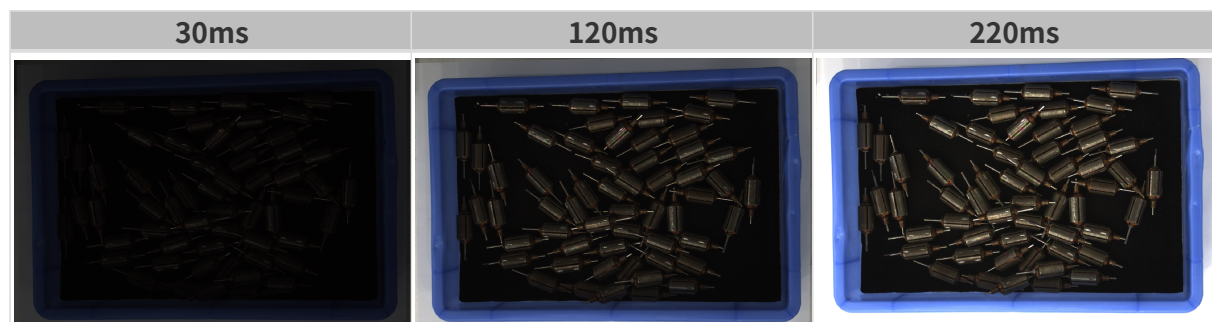
## 曝光模式

<b>参数说明</b>	设置拍摄2D图时的曝光模式。
<b>可见级别</b>	初级、专家、大师
<b>参数取值</b>	<ul style="list-style-type: none"> <li>固定曝光：设置固定曝光时间，常用于稳定的光照条件。</li> <li>自动曝光：自动调整曝光时间，常用于多变的光照条件。</li> <li>HDR：设置多个曝光时间并融合所有图像，常用于颜色或材质多样的物体。</li> <li>Flash：使用投影机补光，常用于较暗的环境。</li> </ul>
<b>调节说明</b>	选择不同选项后， <b>2D参数</b> 分组中将显示不同的参数供调节： <ul style="list-style-type: none"> <li>固定曝光：显示<b>曝光时间</b>。</li> <li>自动曝光：显示<b>灰度值</b>与<b>自动曝光感兴趣区域</b>。</li> <li>HDR：显示<b>色调映射</b>与<b>曝光时间序列</b>。</li> <li>Flash：无需调节其他参数。采集图像时将自动补光。</li> </ul>

### 固定曝光：曝光时间

<b>参数说明</b>	影响图像亮度。 <ul style="list-style-type: none"> <li>曝光时间越长，图像亮度越高。</li> <li>曝光时间越短，图像亮度越低。</li> </ul>
<b>可见级别</b>	初级、专家、大师
<b>参数取值</b>	0.1~999ms
<b>调节说明</b>	根据2D图的质量调节。2D图应不过亮或过暗，可看清目标物体的表面特征。 <ul style="list-style-type: none"> <li>较暗的环境下通常使用较长曝光时间。</li> <li>较亮的环境下通常使用较短曝光时间。</li> </ul>

其他条件相同，仅**曝光时间**不同的2D图对比如下：

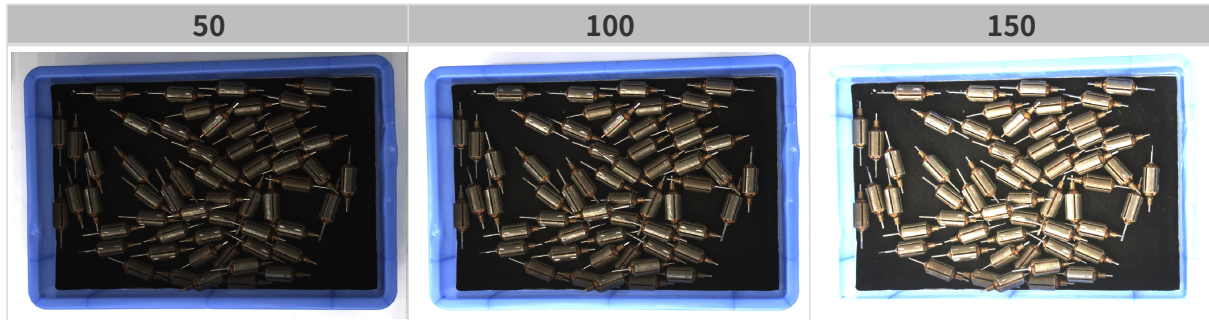


### 自动曝光：灰度值

<b>参数说明</b>	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。
-------------	-------------------------------

可见级别	初级、专家、大师
参数取值	0~255
调节说明	无

其他条件相同，仅灰度值不同的2D图对比如下：

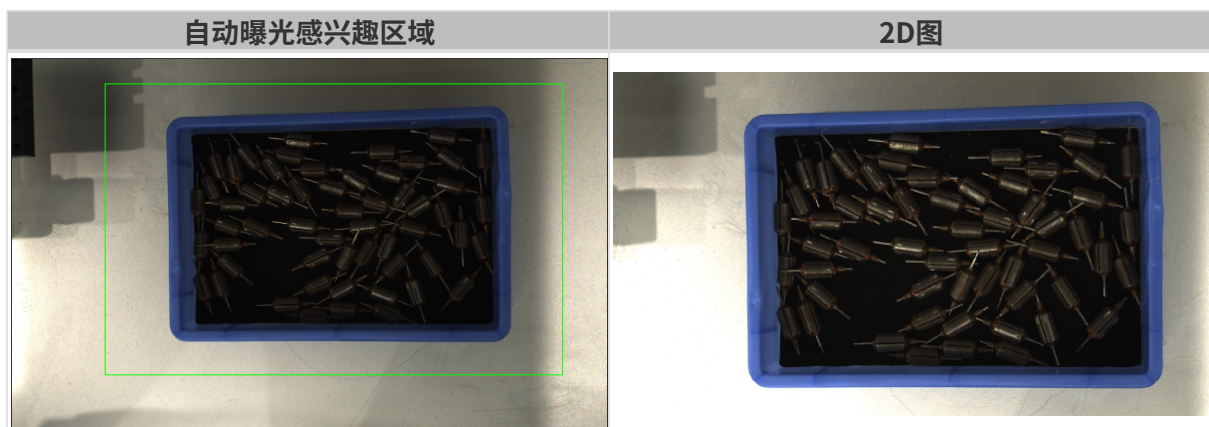


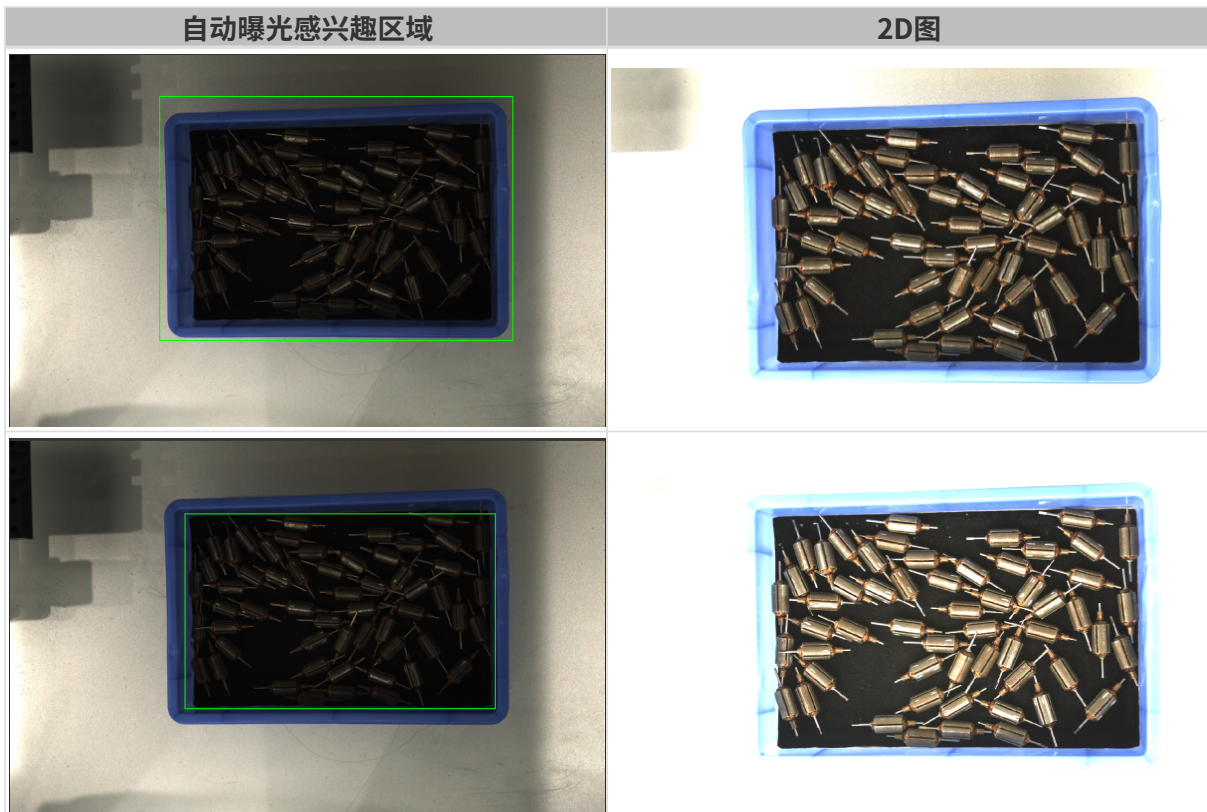
黑白图像的灰度值相当于图像亮度；彩色图像的灰度值相当于每个颜色通道的亮度。

#### 自动曝光：自动曝光感兴趣区域

参数说明	<ul style="list-style-type: none"> <li>相机参照该区域内的光照、物体颜色等自动调整曝光时间。</li> <li>未设置感兴趣区域时，相机根据整个视野自动调整曝光时间。</li> </ul>
可见级别	初级、专家、大师
参数取值	无
调节说明	详见下方 <a href="#">设置自动曝光感兴趣区域</a> 。

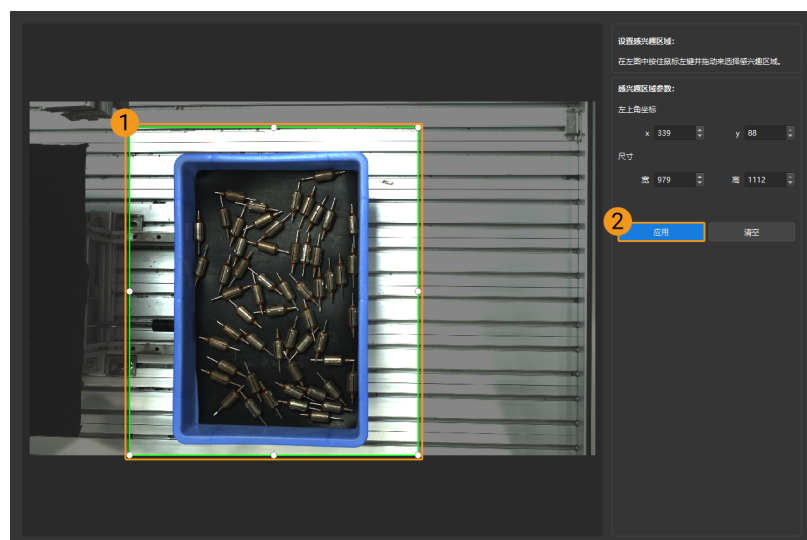
其他条件相同，仅自动曝光感兴趣区域不同的2D图对比如下：





### 设置自动曝光感兴趣区域

1. 双击自动曝光感兴趣区域右侧的[编辑]，进入设置感兴趣区域页面。
2. 在左侧选择并调节感兴趣区域。拖拽选框可调节位置，拖拽白色锚点可调节大小。
3. 单击[应用]，应用设置的自动曝光感兴趣区域。



单击[清空]，可清除当前设置的自动曝光感兴趣区域。

4. 重新采集图像并查看2D图，确认自动曝光的效果。


## HDR：色调映射

参数说明	可使图像看起来更自然。如2D图和真实物体相差较大，可勾选该参数。
可见级别	初级、专家、大师
参数取值	<ul style="list-style-type: none"> <li>• False</li> <li>• True</li> </ul>
调节说明	勾选即可开启色调映射。

色调映射开启前后的2D图对比如下：

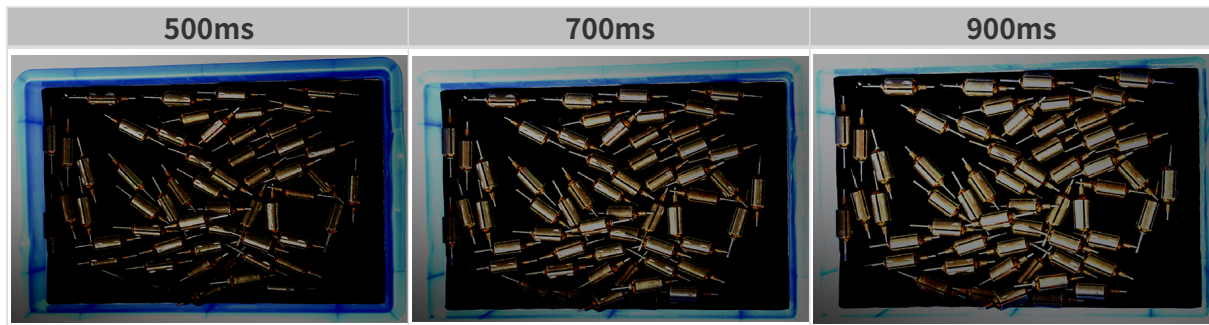


## HDR：曝光时间序列

参数说明	设置多个曝光时间，并将所拍摄的图像融合为一张保留更多明暗细节的2D图。
可见级别	初级、专家、大师
参数取值	无
调节说明	<ol style="list-style-type: none"> <li>1. 双击<b>曝光时间序列</b>右侧的[编辑]，进入曝光时间序列的编辑界面。</li> <li>2. 单击[+]，新增并设置曝光时间。</li> <li>3. 如需删除某个曝光时间，选中该曝光时间并单击[-]。</li> <li>4. 编辑完成后，单击[应用]使曝光时间序列生效。</li> </ol> <div style="display: flex; align-items: center; margin-top: 10px;"> <div style="margin-right: 10px;">  </div> <ul style="list-style-type: none"> <li>◦ [取消]：不保存当前设置，直接退出该页面。</li> <li>◦ [重置]：清空所有的曝光时间。</li> </ul> </div>

设置单个曝光时间时，不同曝光时间的2D图对比如下：





使用以上曝光时间组合成不同的曝光时间序列时，不同序列的2D图对比如下：



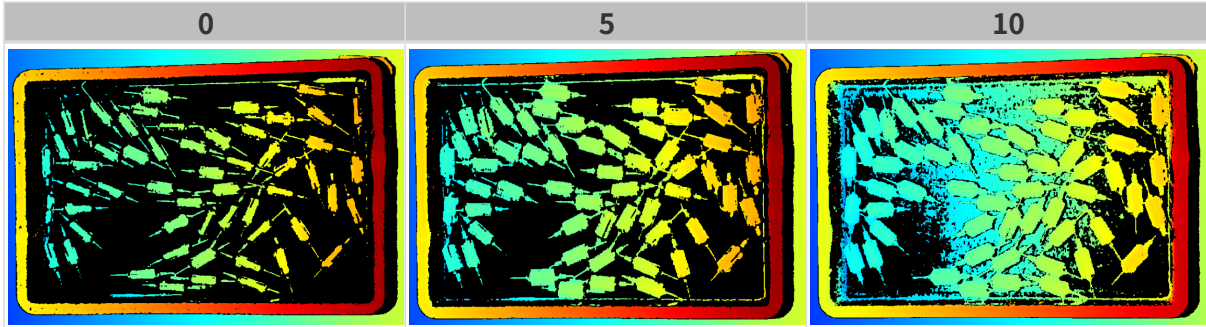
### 3D参数

#### 相机增益

参数说明	用于增加图像亮度，但可能会引入噪点。 <div style="display: flex; align-items: center; margin-top: 5px;"> <span style="font-size: 1.2em; margin-right: 5px;">i</span> <span>该参数通过改变用于计算深度数据的图像的亮度影响深度图和点云质量。</span> </div>
可见级别	专家、大师
参数取值	0~16dB
调节说明	当设置曝光时间无法达到期望亮度时，可以使用该参数。

其他条件相同，仅相机增益不同的2D图和深度图对比如下：





## 深度图与点云参数

深度图与点云中，所需的数据应完整。



根据实际需求判断所需数据的范围。例如，需通过碗沿抓取碗口向上摆放的金属碗时，一般只需确保碗沿部分的数据完整。

以下分组中的参数影响深度图与点云质量。

参数分组	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

该分组下的参数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。

使用**曝光助手**可获得推荐的曝光参数组合。双击**3D参数**右侧的[自动设置]，即可打开**曝光助手**。

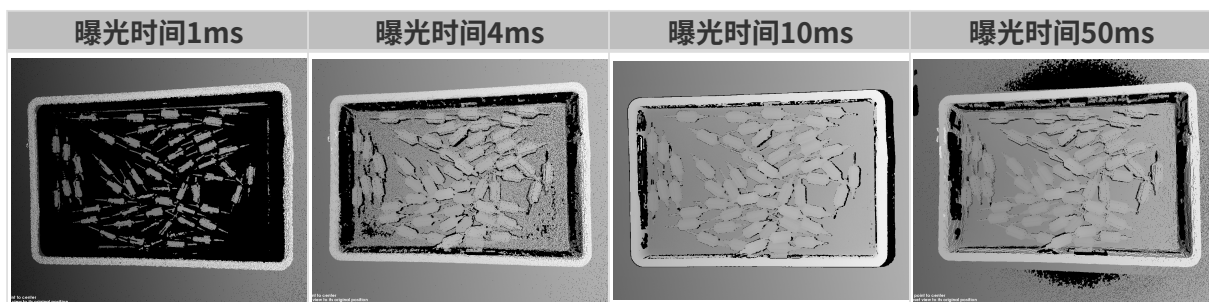
## 曝光次数

<b>参数说明</b>	设置 <b>曝光时间</b> 的个数。
<b>可见级别</b>	初级、专家、大师
<b>参数取值</b>	1~3
<b>调节说明</b>	<ul style="list-style-type: none"> <li>曝光次数大于1时，需设置多个<b>曝光时间</b>。</li> <li>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，但也将延长处理时间。</li> <li>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。</li> </ul>

## 曝光时间


参数说明	设置采集深度信息时的曝光时间，需设置的 <b>曝光时间</b> 个数取决于 <b>曝光次数</b> 的值。
可见级别	初级、专家、大师
参数取值	<ul style="list-style-type: none"> <li>取值范围：0.1~99ms</li> </ul>
调节说明	<ul style="list-style-type: none"> <li>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。</li> <li>曝光时间过长或过短都会导致信息缺失。</li> <li>激光相机的曝光时间须为4的倍数，输入值不为4的倍数时将被自动调整。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。</li> </ul>

不同**曝光时间**对比图如下：

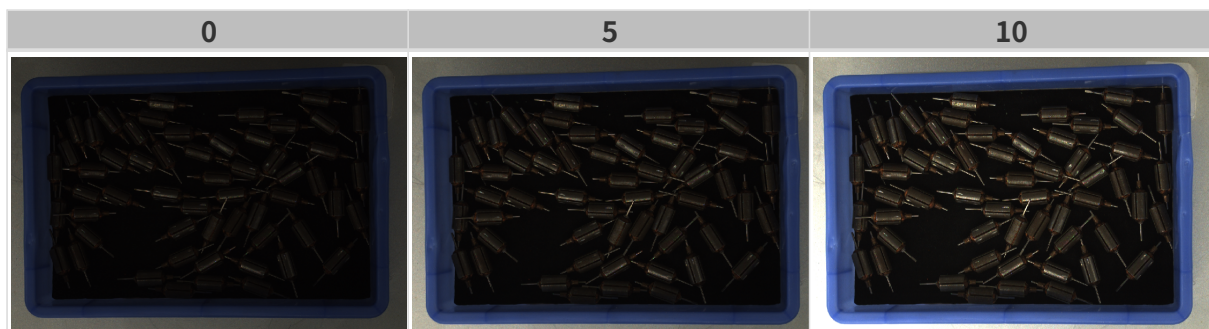


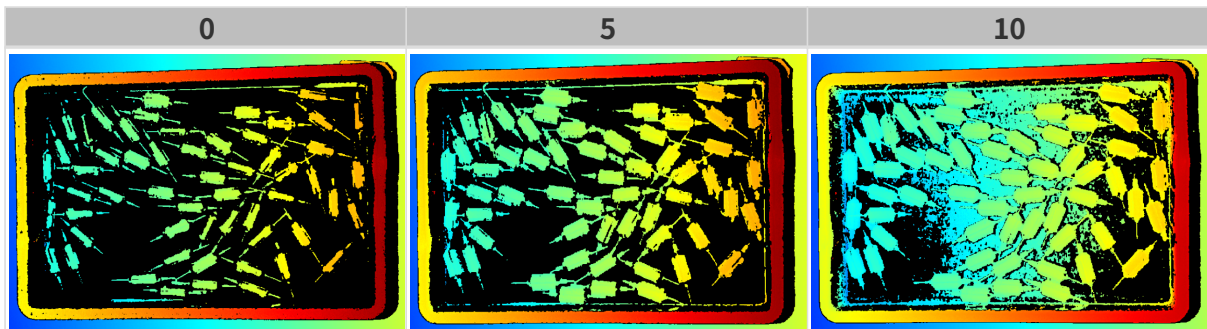
黑色部分缺少对应的物体点云。

## 相机增益

参数说明	用于增加图像亮度，但可能会引入噪点。
	 该参数通过改变 <b>用于计算深度数据的图像</b> 的亮度影响深度图和点云质量。
可见级别	专家、大师
参数取值	0~16dB
调节说明	当设置 <b>曝光时间</b> 无法达到期望亮度时，可以使用该参数。

其他条件相同，仅**相机增益**不同的2D图和深度图对比如下：





## 投影

### 投影光亮度

参数说明	投影仪投射的结构光亮度。
可见级别	专家、大师
参数取值	<ul style="list-style-type: none"> <li>High: 高亮度, 适用于深色物体。</li> <li>Normal: 正常亮度, 适用于普通物体。</li> <li>Low: 低亮度, 适用于反光物体。</li> </ul>
调节说明	根据物体类型调节。

### 编码模式

参数说明	选择投射的结构光的样式。
可见级别	专家、大师
参数取值	<ul style="list-style-type: none"> <li>Fast: 适用于不透明的物体, 采集速度快, 但深度数据质量稍差。</li> <li>Accurate: 适用于不透明的物体, 深度数据质量高, 但采集速度较慢。</li> <li>Translucent: 适用于透明物体, 深度数据质量高, 但采集速度较慢。</li> </ul>
调节说明	根据目标物体类型以及对数据质量和采集速度的实际需求调节。



- 选择**Translucent**时, 以下工具和参数不可用:
  - 3D曝光助手工具
  - 3D参数分组下的**抗频闪模式**
  - 点云后处理分组下的**边缘保持**和**失真校正**
- 选择**Translucent**时, 可通过调节点云后处理分组下的**空缺填充**, 填充点云中缺失的点。

#### *Translucent*选项适用场景

**Translucent**选项可显著提升透明物体的深度数据质量, 以下为部分典型的适用物体:

- 医用盐水袋
- 磨砂塑料试管

- 商品的塑料包装（袋、瓶、罐、盒等）

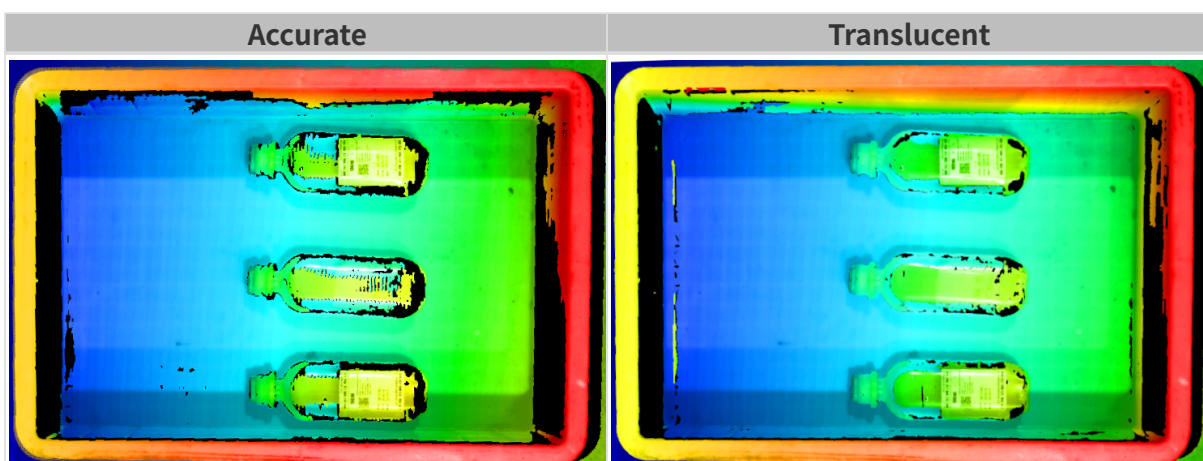
同时，**Translucent**选项的提升效果存在以下限制：

- 对高度透明的物体（如非磨砂表面的玻璃试管）提升效果不明显。
- 对物体上弧度大的部分提升效果不明显。
- 透明物体互相重叠时，提升效果不明显。
- 背景颜色较明亮、或背景易反光时，提升效果不明显。
- 环境光较强或频繁变化时，提升效果不明显。
- 使用2D图颜色为黑白的相机时，因相机投射的结构光为蓝色，如透明物体内包含蓝色物体，提升效果不明显。

目标物体为非透明物体时，其他条件相同，**编码模式**分别选择**Accurate**和**Fast**时获取的点云如下：



目标物体为透明物体时，其他条件相同，**编码模式**分别选择**Accurate**和**Translucent**时获取的点云如下：



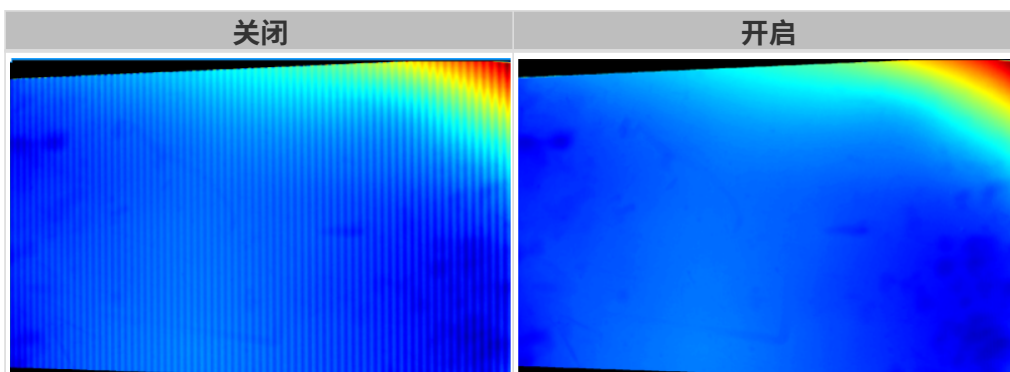
## 抗频闪模式

参数说明	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。
可见级别	专家、大师
参数取值	<ul style="list-style-type: none"> <li>• Off</li> <li>• AC50Hz</li> <li>• AC60Hz</li> </ul>
调节说明	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。



3D参数分组下的**编码模式**设置为Translucent时，**抗频闪模式**不可用。

抗频闪模式开启前后的深度图对比如下：



## 点云后处理

调节点云后处理分组下的参数，可提升点云质量。

### 调参原则

调节点云后处理参数时，遵循以下调参原则可减少相机采集时间，优化节拍。

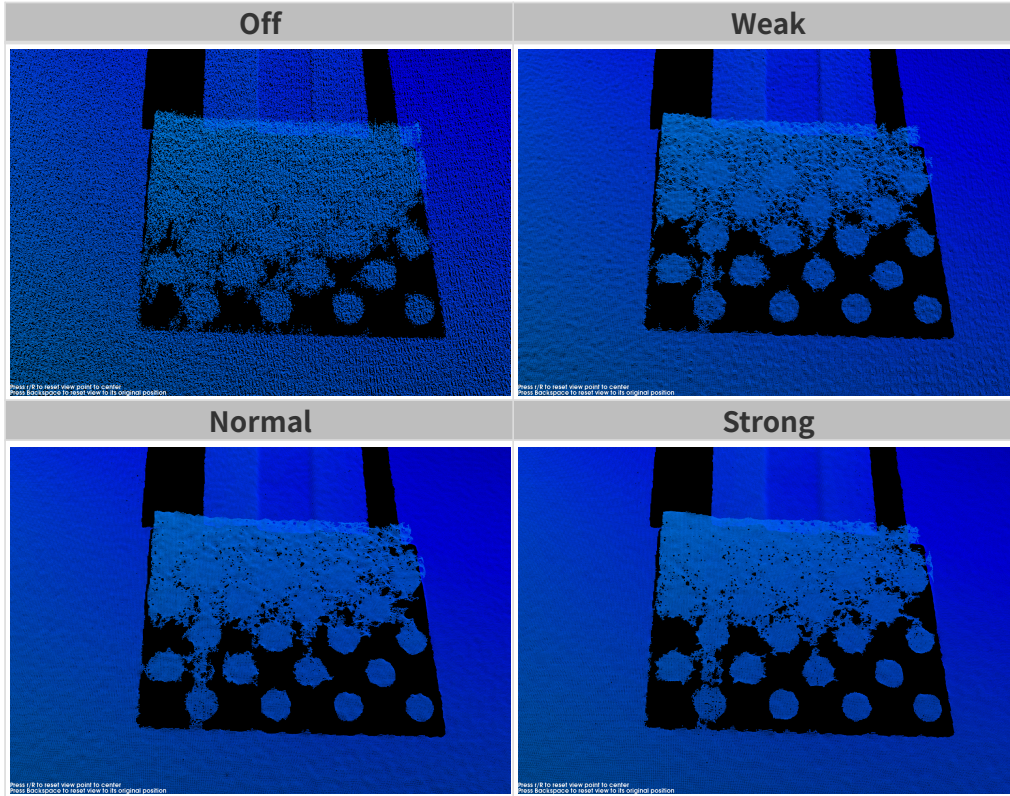
1. 优先调节**离群点去除**。该参数各强度计算时间基本相同，即使使用较高强度也不会增加太多计算时间。
2. 建议使用低强度**表面平滑**与**噪点去除**。这两个参数强度越高，计算时间越长。

## 表面平滑

参数说明	可减少点云的深度波动，使点云更接近真实的物体表面，但会损失部分物体表面细节。
可见级别	初级、专家、大师
参数取值	<ul style="list-style-type: none"> <li>• Off</li> <li>• Weak</li> <li>• Normal</li> <li>• Strong</li> </ul>

<b>调节说明</b>	<ul style="list-style-type: none"> <li>• <b>表面平滑</b>强度越高，损失的物体表面细节越多；强度越低，损失的物体表面细节越少。</li> <li>• <b>表面平滑</b>强度越高，计算时间越长；强度越低，计算时间越短。</li> </ul>
-------------	--

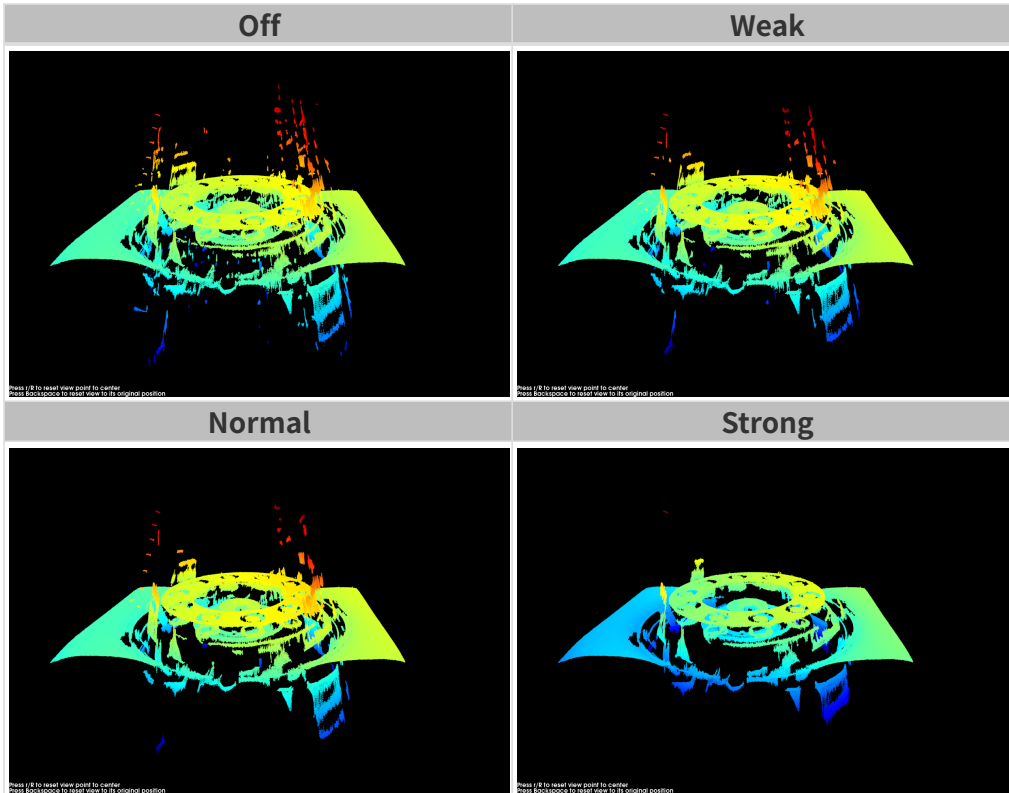
其他条件相同，仅**表面平滑**强度不同的点云对比如下：



### 离群点去除

<b>参数说明</b>	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。
<b>可见级别</b>	初级、专家、大师
<b>参数取值</b>	<ul style="list-style-type: none"> <li>• Off</li> <li>• Weak</li> <li>• Normal</li> <li>• Strong</li> </ul>
<b>调节说明</b>	<ul style="list-style-type: none"> <li>• 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。</li> <li>• 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。</li> </ul>

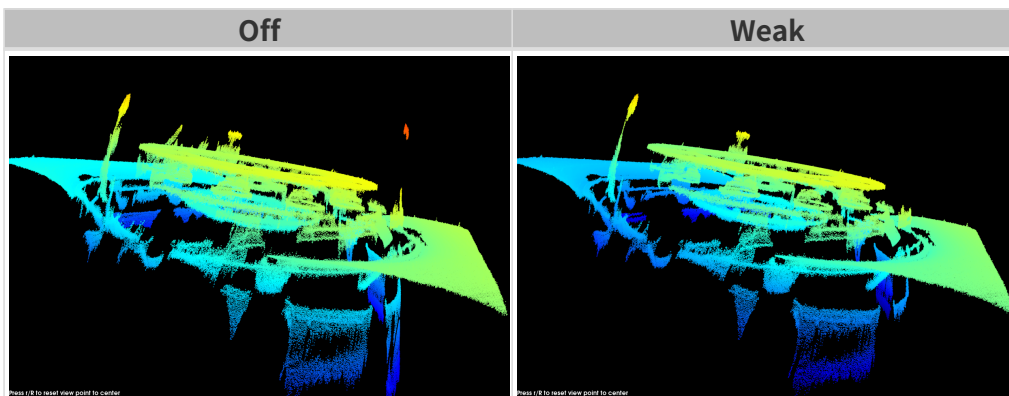
其他条件相同，仅**离群点去除**强度不同的点云对比如下：



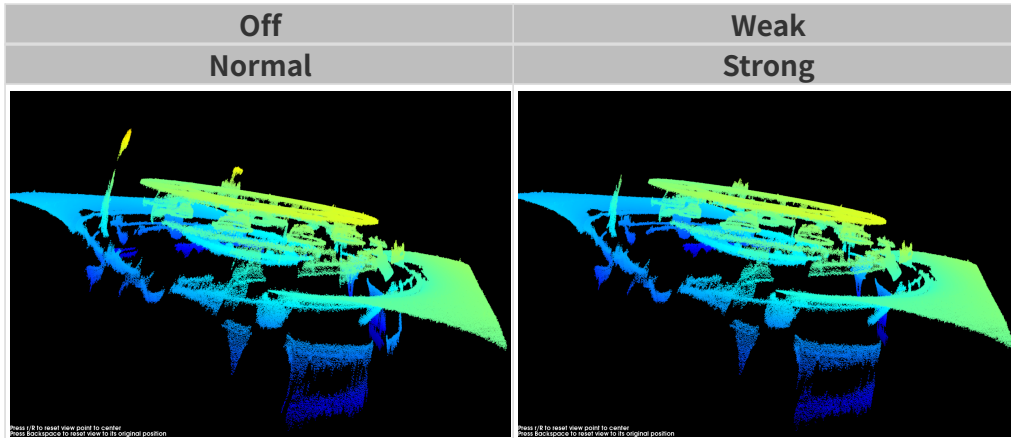
### 噪点去除

参数说明	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。
可见级别	初级、专家、大师
参数取值	<ul style="list-style-type: none"> <li>• Off</li> <li>• Weak</li> <li>• Normal</li> <li>• Strong</li> </ul>
调节说明	<ul style="list-style-type: none"> <li>• 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。</li> <li>• 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。</li> </ul>

其他条件相同，仅噪点去除强度不同的点云对比如下：







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。

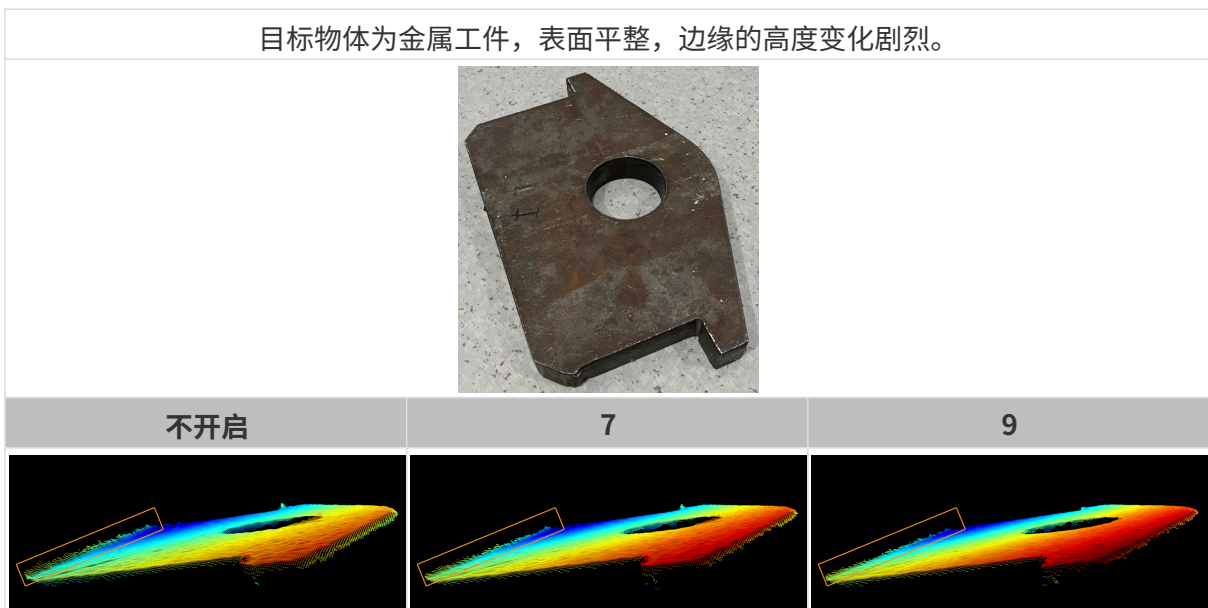
## 失真校正

参数说明	校正深度数据中由物体颜色或高度的剧烈变化导致的失真。 请勾选 <b>启用失真校正</b> ，然后调节 <b>当前值</b> 。
可见级别	专家、大师
参数取值	1~10
调节说明	根据深度图或点云中的失真程度调节 <b>当前值</b> ，重新采集图像确认效果。

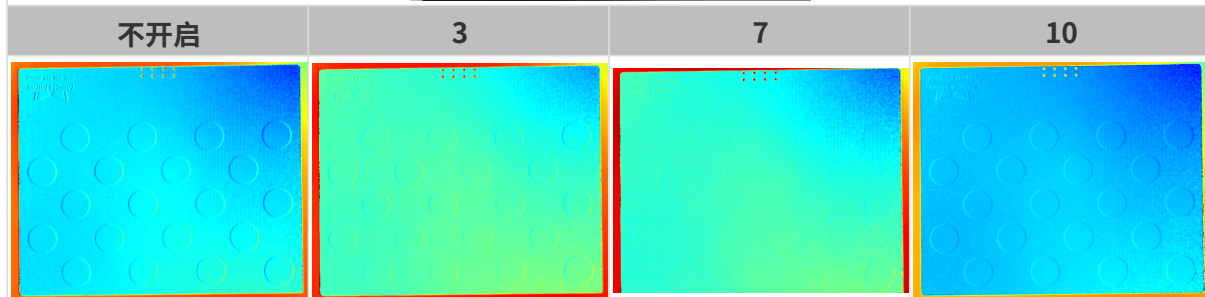
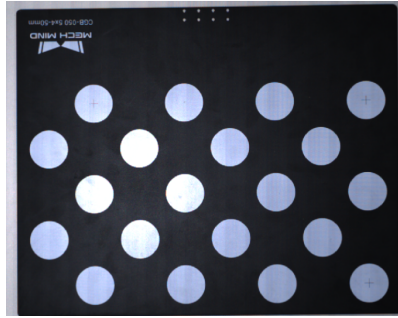


- 启用失真校正将降低采集速度。
- **当前值**过大可能导致反效果。请在调节后重新采集数据，确认该参数对深度图和点云的影响。
- **3D参数**分组下的**编码模式**参数设置为**Translucent**时，失真校正不可用。

其他条件相同，仅**失真校正**的**当前值**不同的深度图对比如下：



目标物体为标定板，表面平整，有从黑色到白色的颜色剧烈变化。



### 空缺填充

参数说明	填充点云中的空缺，使物体表面特征更完整。
可见级别	专家、大师
参数取值	<ul style="list-style-type: none"> <li>Off</li> <li>Weak</li> <li>Normal</li> <li>Strong</li> </ul>
调节说明	根据点云中点的缺失量调节。



- 该参数仅在以下条件下可用：**3D参数**分组下的**编码模式**设置为**Translucent**。
- 空缺填充强度越高，被填充的缺失点越多，但也可能导致物体边缘失真。

### 边缘保持

参数说明	在进行表面平滑时保持物体边缘的锐利度。
可见级别	大师
参数取值	<ul style="list-style-type: none"> <li>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。</li> <li>Normal：在保持边缘的同时达到较好的表面平滑效果。</li> <li>Smooth：不进行边缘保持。表面平滑效果最好，但物体边缘会失真。</li> </ul>
调节说明	根据对工件边缘特征的需求调节。

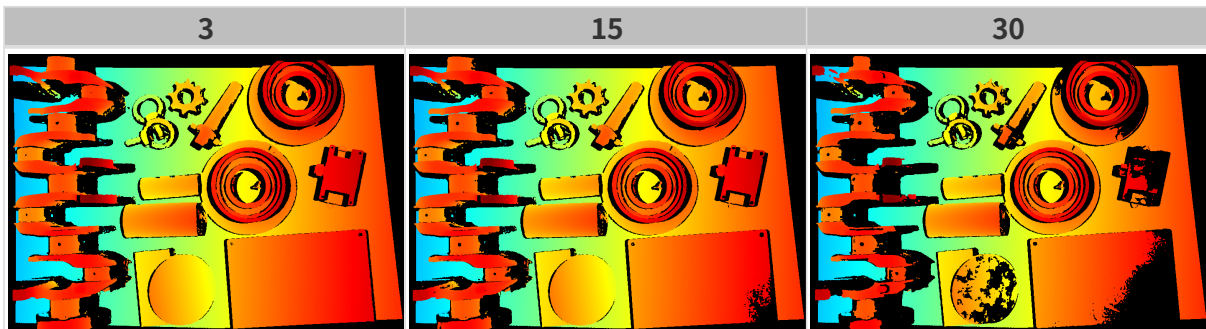


- 3D参数**分组下的**编码模式**设置为**Translucent**时，**边缘保持**不可用。

### 条纹对比度阈值

参数说明	用于去除点云中的噪点。如调节 <b>离群点去除</b> 与 <b>噪点去除</b> 后，仍无法获得需要的点云，再调节该参数。
可见级别	初级、专家、大师
参数取值	1~100
调节说明	<ul style="list-style-type: none"> <li>该参数值越大，去除的点越多；值越小，去除的点越少。</li> <li>调大该参数可以去除点云中的噪点，但也可能造成较暗物体对应的点丢失。</li> </ul>

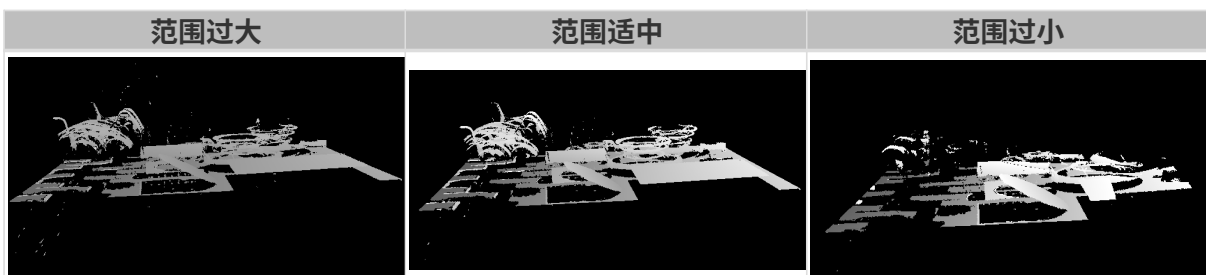
其他条件相同，仅**条纹对比度阈值**不同的点云对比如下：

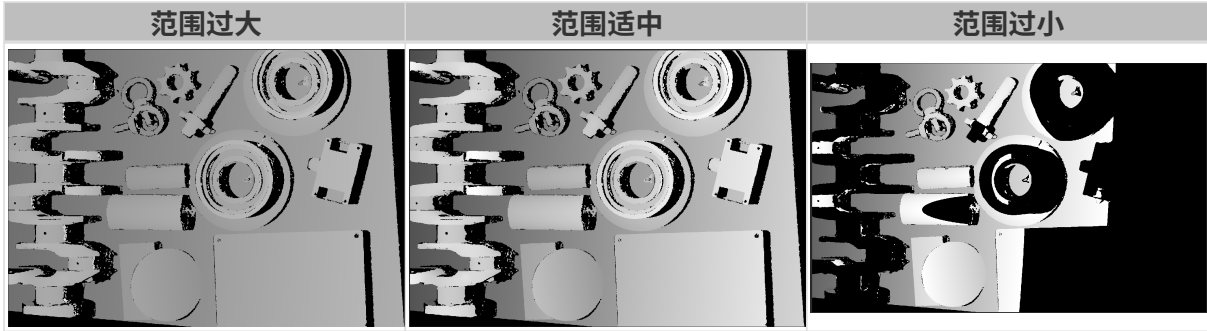


### 深度范围

参数说明	设置Z向感兴趣区域。在相机的工作距离范围内设置 <b>深度范围</b> ，可滤除 <b>深度范围</b> 外的数据。
可见级别	初级、专家、大师
参数取值	<ul style="list-style-type: none"> <li>下限：1~4000mm</li> <li>上限：1~5000mm</li> </ul>
调节说明	<ul style="list-style-type: none"> <li><b>深度范围</b>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。</li> <li>如何设置详见下方的<a href="#">设置深度范围</a>。</li> </ul>

不同**深度范围**效果对比：

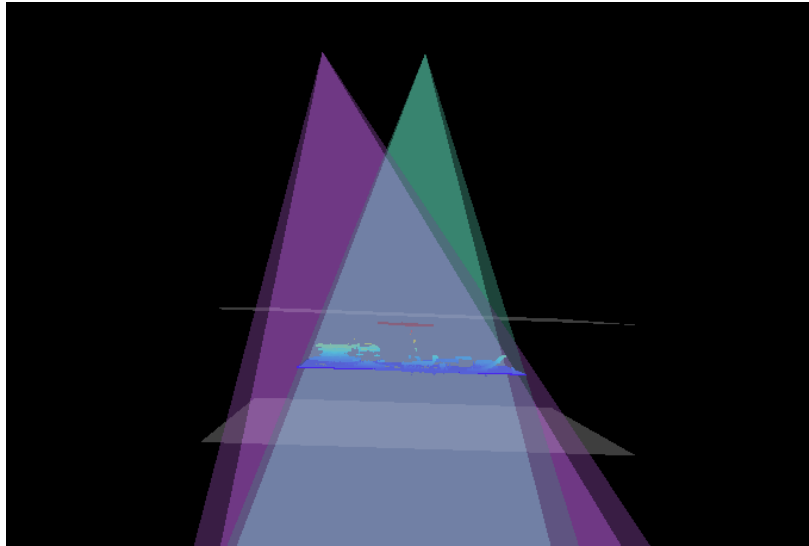




### 设置深度范围

执行以下步骤调节深度范围：

1. 双击深度范围右侧的[编辑]，打开设置深度范围界面。
2. 点击右边栏最上方的[更新点云]，获取最新的点云。
3. 调整点云的位置：调整到可看到代表深度范围上下限的两个灰色长方形即可。



4. 调节深度范围：拖动右侧滑动条上的滑块，大致调节深度范围。再输入数值，精准调节深度范围。



判断深度范围是否合适：所有必需的物体特征均位于两个灰色长方形之间，大多数噪点和离群点位于该区域之外。

5. 设置完成后，单击右下角的[保存]。



- 单击[推荐值]，可将深度范围调整为当前相机的推荐工作距离。
- 单击[重置]，可将深度范围恢复为上次保存的数值。

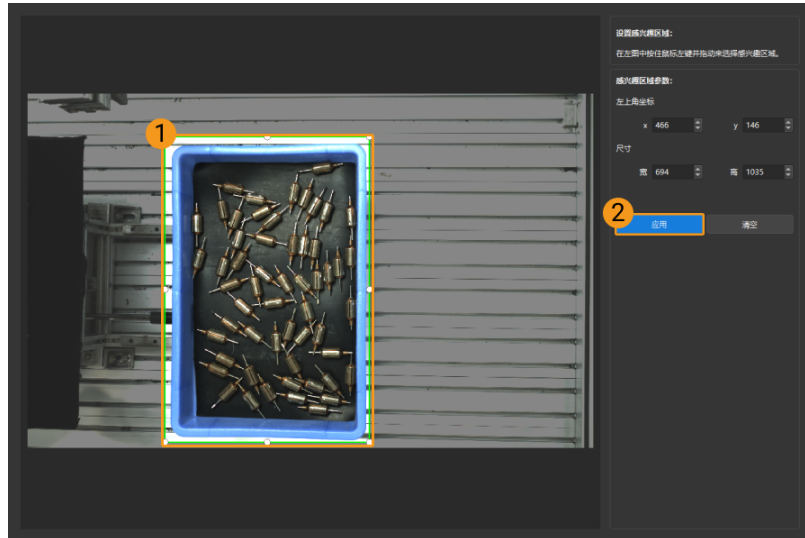
### 感兴趣区域

参数说明	设置深度图和点云在XOY平面上的感兴趣区域，所选区域外的点将被移除。
可见级别	初级、专家、大师

参数取值	无
调节说明	详见下方 <a href="#">设置感兴趣区域</a> 。

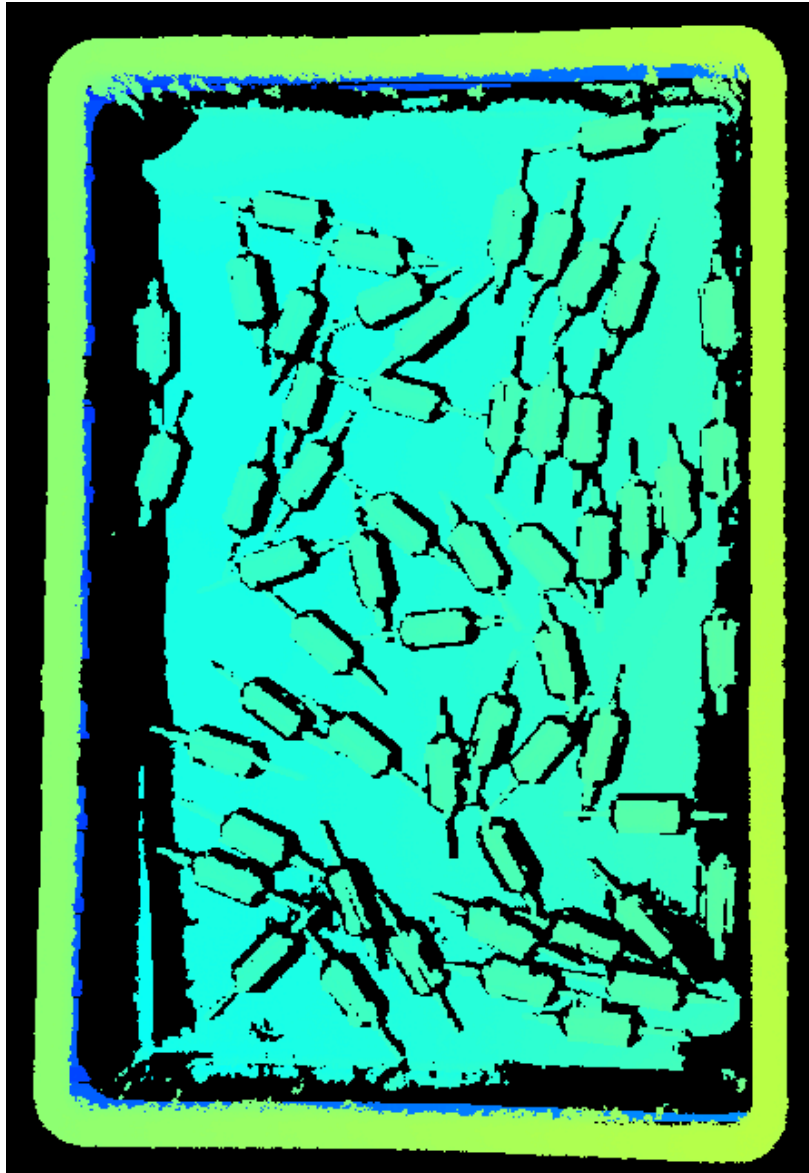
### 设置感兴趣区域

1. 双击**感兴趣区域**右侧的[编辑]，进入**设置感兴趣区域**页面。
2. 在左侧选择并调节感兴趣区域。拖拽选框可调节位置，拖拽白色锚点可调节大小。
3. 单击[应用]，应用设置的感兴趣区域。



- 单击[清空]，可清除当前设置的感兴趣区域。
- DEEP (V4) 与LSR (V4) 系列，在此界面显示的是2D图（深度源）。如果图像过暗或过亮，请调节**2D图（深度源）曝光模式**。

4. 重新采集图像，查看深度图或点云，确认所设感兴趣区域的效果。



### 5.3.3. PRO XS参数

本章介绍PRO XS相机的参数。参数按照影响的数据类型拆分为2D图参数及深度图与点云参数。

#### 2D图参数

2D图应不过亮或过暗，可看清目标物体的表面特征。

2D参数分组下的参数及3D参数分组下的**相机增益**影响2D图质量。



彩色相机采集图像时，如因现场光照条件导致图像颜色与实际差别较大，请调节**白平衡**。详细操作请参考[调节白平衡](#)。

#### 2D参数

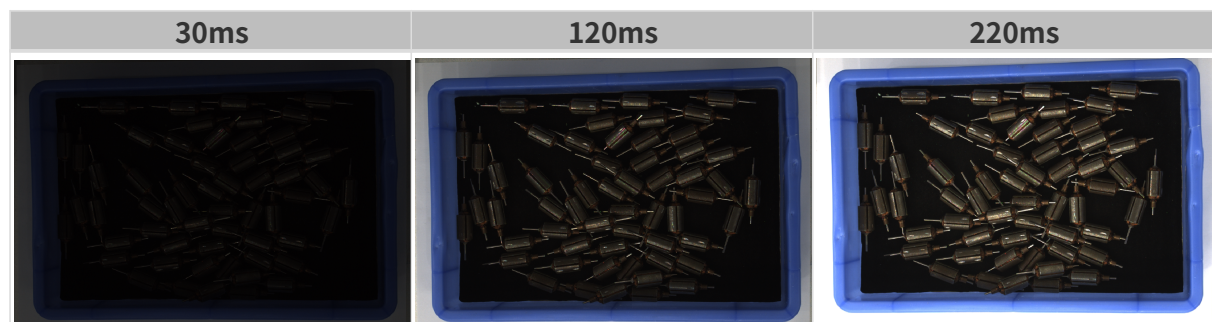
## 曝光模式

<b>参数说明</b>	设置拍摄2D图时的曝光模式。
<b>可见级别</b>	初级、专家、大师
<b>参数取值</b>	<ul style="list-style-type: none"> <li>固定曝光：设置固定曝光时间，常用于稳定的光照条件。</li> <li>自动曝光：自动调整曝光时间，常用于多变的光照条件。</li> <li>HDR：设置多个曝光时间并融合所有图像，常用于颜色或材质多样的物体。</li> <li>Flash：使用投影机补光，常用于较暗的环境。</li> </ul>
<b>调节说明</b>	选择不同选项后， <b>2D参数</b> 分组中将显示不同的参数供调节： <ul style="list-style-type: none"> <li>固定曝光：显示<b>曝光时间</b>。</li> <li>自动曝光：显示<b>灰度值</b>与<b>自动曝光感兴趣区域</b>。</li> <li>HDR：显示<b>色调映射</b>与<b>曝光时间序列</b>。</li> <li>Flash：无需调节其他参数。采集图像时将自动补光。</li> </ul>

### 固定曝光：曝光时间

<b>参数说明</b>	影响图像亮度。 <ul style="list-style-type: none"> <li>曝光时间越长，图像亮度越高。</li> <li>曝光时间越短，图像亮度越低。</li> </ul>
<b>可见级别</b>	初级、专家、大师
<b>参数取值</b>	0.1~999ms
<b>调节说明</b>	根据2D图的质量调节。2D图应不过亮或过暗，可看清目标物体的表面特征。 <ul style="list-style-type: none"> <li>较暗的环境下通常使用较长曝光时间。</li> <li>较亮的环境下通常使用较短曝光时间。</li> </ul>

其他条件相同，仅**曝光时间**不同的2D图对比如下：

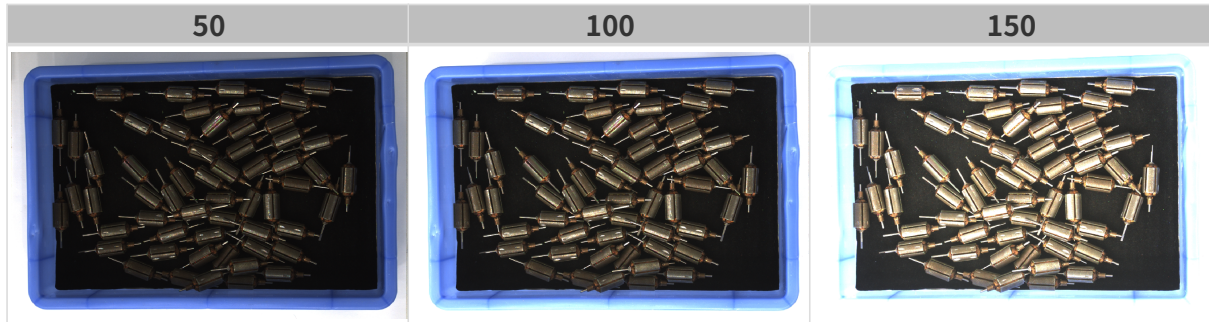


### 自动曝光：灰度值

<b>参数说明</b>	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。
-------------	-------------------------------

可见级别	初级、专家、大师
参数取值	0~255
调节说明	无

其他条件相同，仅灰度值不同的2D图对比如下：

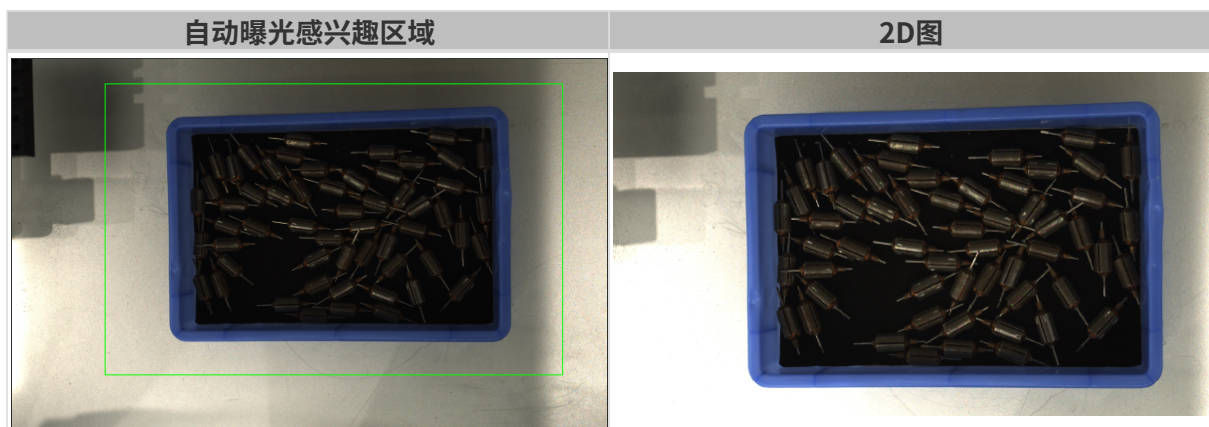


黑白图像的灰度值相当于图像亮度；彩色图像的灰度值相当于每个颜色通道的亮度。

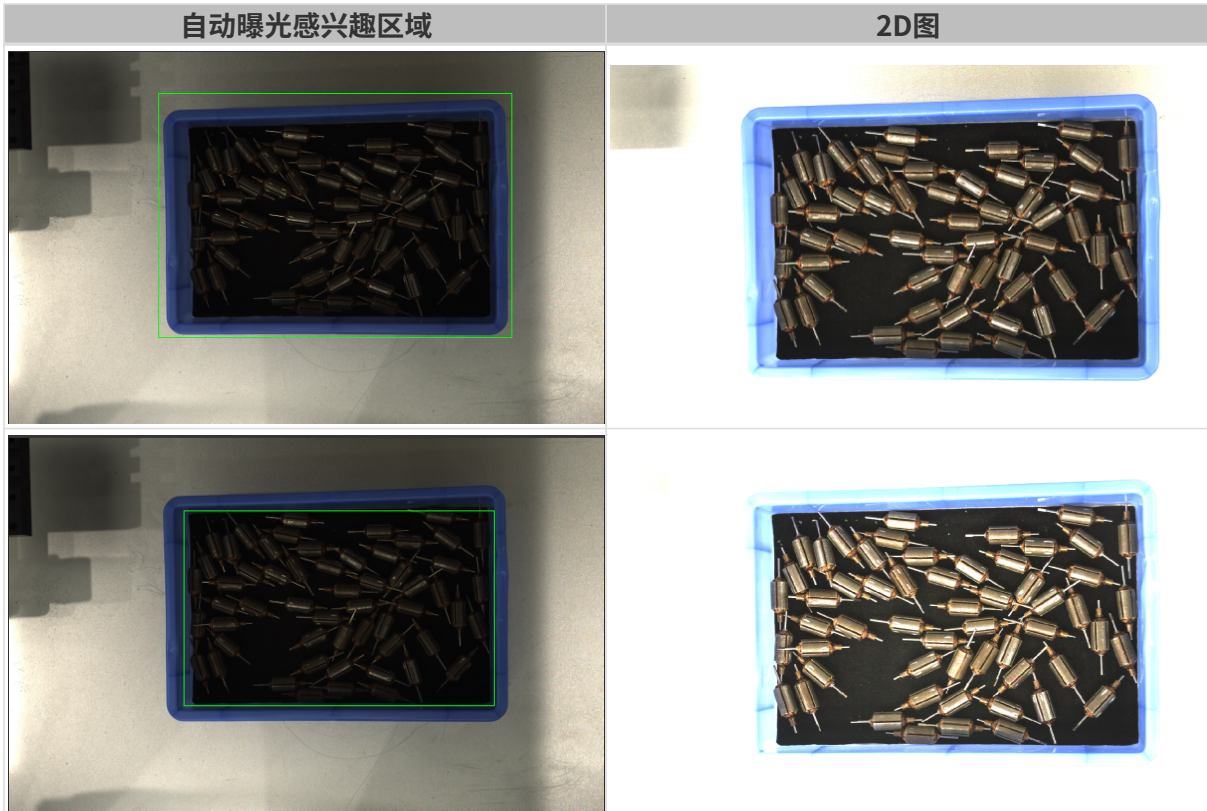
### 自动曝光：自动曝光感兴趣区域

参数说明	<ul style="list-style-type: none"> <li>相机参照该区域内的光照、物体颜色等自动调整曝光时间。</li> <li>未设置感兴趣区域时，相机根据整个视野自动调整曝光时间。</li> </ul>
可见级别	初级、专家、大师
参数取值	无
调节说明	详见下方 <a href="#">设置自动曝光感兴趣区域</a> 。

其他条件相同，仅自动曝光感兴趣区域不同的2D图对比如下：

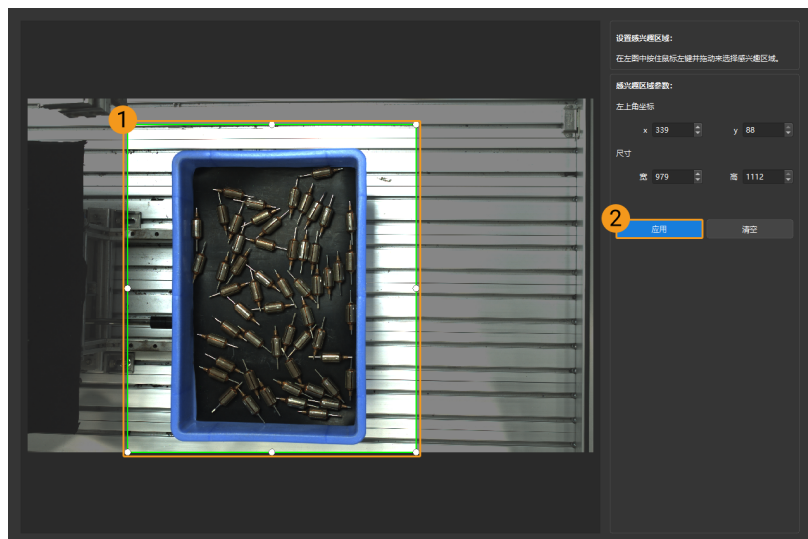






### 设置自动曝光感兴趣区域

1. 双击自动曝光感兴趣区域右侧的[编辑]，进入设置感兴趣区域页面。
2. 在左侧选择并调节感兴趣区域。拖拽选框可调节位置，拖拽白色锚点可调节大小。
3. 单击[应用]，应用设置的自动曝光感兴趣区域。



单击[清空]，可清除当前设置的自动曝光感兴趣区域。

4. 重新采集图像并查看2D图，确认自动曝光的效果。


## HDR：色调映射

参数说明	可使图像看起来更自然。如2D图和真实物体相差较大，可勾选该参数。
可见级别	初级、专家、大师
参数取值	<ul style="list-style-type: none"> <li>• False</li> <li>• True</li> </ul>
调节说明	勾选即可开启色调映射。

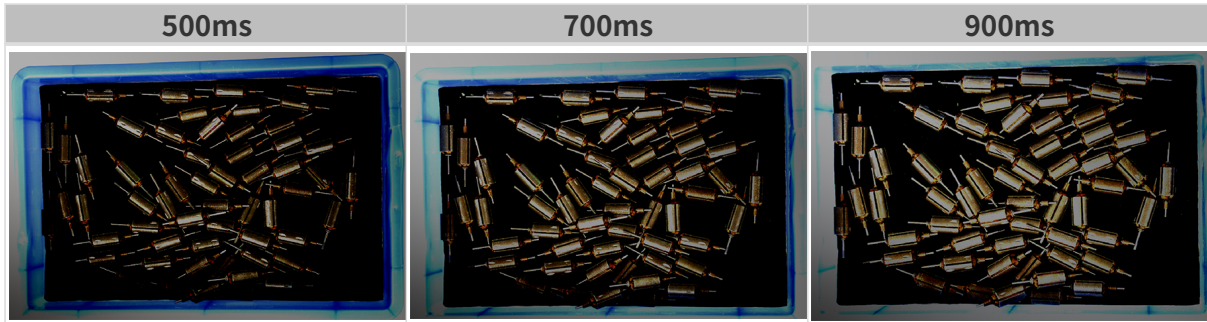
色调映射开启前后的2D图对比如下：



## HDR：曝光时间序列

参数说明	设置多个曝光时间，并将所拍摄的图像融合为一张保留更多明暗细节的2D图。
可见级别	初级、专家、大师
参数取值	无
调节说明	<ol style="list-style-type: none"> <li>1. 双击<b>曝光时间序列</b>右侧的[编辑]，进入曝光时间序列的编辑界面。</li> <li>2. 单击[+]，新增并设置曝光时间。</li> <li>3. 如需删除某个曝光时间，选中该曝光时间并单击[-]。</li> <li>4. 编辑完成后，单击[应用]使曝光时间序列生效。</li> </ol> <div style="display: flex; align-items: center; margin-top: 10px;"> <div style="margin-right: 10px;">  </div> <ul style="list-style-type: none"> <li>◦ [取消]：不保存当前设置，直接退出该页面。</li> <li>◦ [重置]：清空所有的曝光时间。</li> </ul> </div>

设置单个曝光时间时，不同曝光时间的2D图对比如下：



使用以上曝光时间组合成不同的曝光时间序列时，不同序列的2D图对比如下：



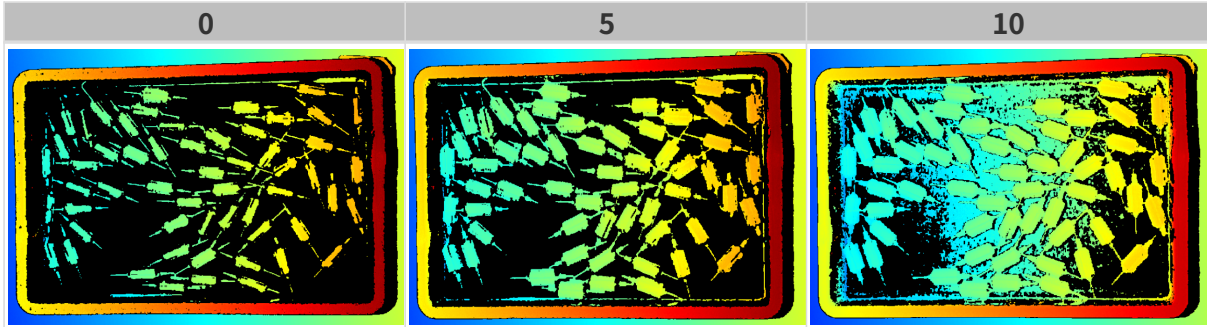
### 3D参数

#### 相机增益

参数说明	用于增加图像亮度，但可能会引入噪点。 <div style="display: flex; align-items: center; margin-top: 5px;"> <span style="font-size: 1.2em; margin-right: 5px;">i</span> <span>该参数通过改变用于计算深度数据的图像的亮度影响深度图和点云质量。</span> </div>
可见级别	专家、大师
参数取值	0~16dB
调节说明	当设置曝光时间无法达到期望亮度时，可以使用该参数。

其他条件相同，仅相机增益不同的2D图和深度图对比如下：





## 深度图与点云参数

深度图与点云中，所需的数据应完整。



根据实际需求判断所需数据的范围。例如，需通过碗沿抓取碗口向上摆放的金属碗时，一般只需确保碗沿部分的数据完整。

以下分组中的参数影响深度图与点云质量。

参数分组	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

该分组下的参数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。

使用**曝光助手**可获得推荐的曝光参数组合。双击**3D参数**右侧的[自动设置]，即可打开**曝光助手**。

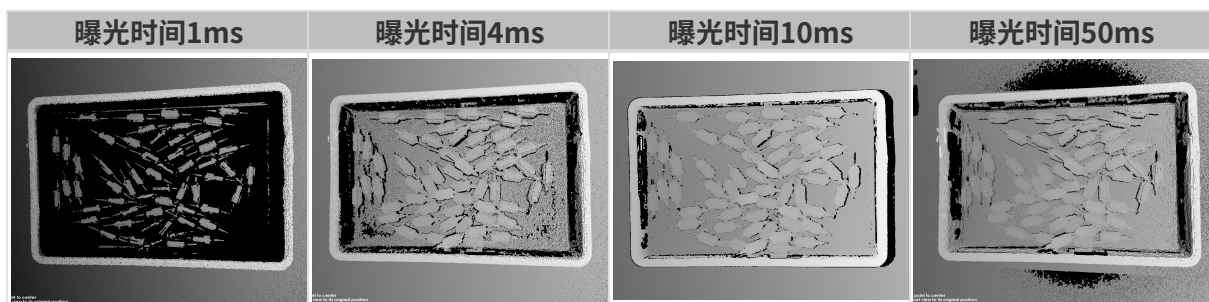
## 曝光次数

<b>参数说明</b>	设置 <b>曝光时间</b> 的个数。
<b>可见级别</b>	初级、专家、大师
<b>参数取值</b>	1~3
<b>调节说明</b>	<ul style="list-style-type: none"> <li>曝光次数大于1时，需设置多个<b>曝光时间</b>。</li> <li>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，但也将延长处理时间。</li> <li>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。</li> </ul>

## 曝光时间

参数说明	设置采集深度信息时的曝光时间，需设置的 <b>曝光时间</b> 个数取决于 <b>曝光次数</b> 的值。
可见级别	初级、专家、大师
参数取值	<ul style="list-style-type: none"> <li>取值范围：0.1~99ms</li> </ul>
调节说明	<ul style="list-style-type: none"> <li>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。</li> <li>曝光时间过长或过短都会导致信息缺失。</li> <li>激光相机的曝光时间须为4的倍数，输入值不为4的倍数时将被自动调整。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。</li> </ul>

不同**曝光时间**对比图如下：

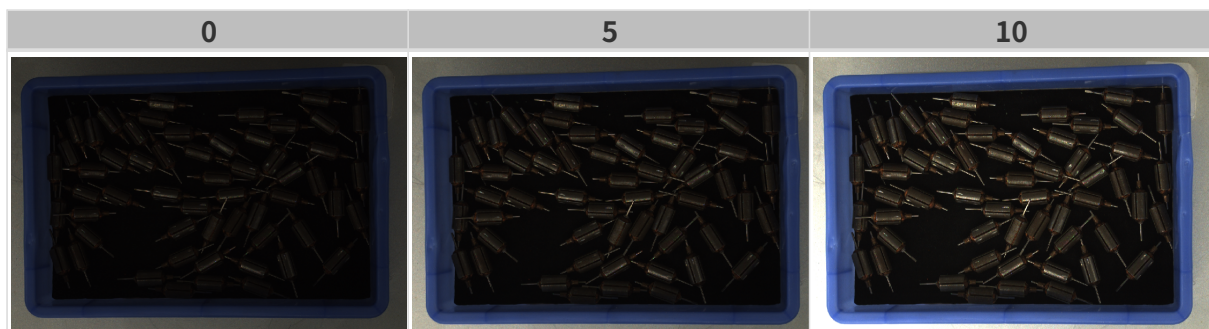


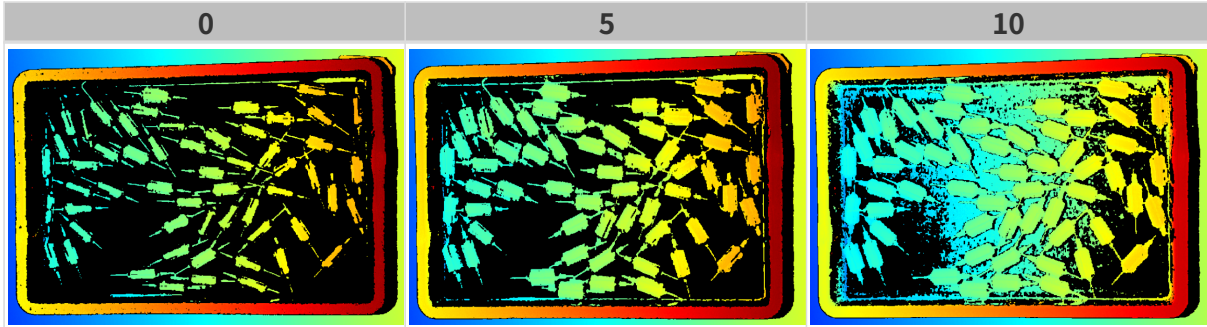
黑色部分缺少对应的物体点云。

## 相机增益

参数说明	用于增加图像亮度，但可能会引入噪点。
	 该参数通过改变 <b>用于计算深度数据的图像</b> 的亮度影响深度图和点云质量。
可见级别	专家、大师
参数取值	0~16dB
调节说明	当设置 <b>曝光时间</b> 无法达到期望亮度时，可以使用该参数。

其他条件相同，仅**相机增益**不同的2D图和深度图对比如下：





## 投影

### 投影光亮度

参数说明	投影仪投射的结构光亮度。
可见级别	专家、大师
参数取值	<ul style="list-style-type: none"> <li>High: 高亮度, 适用于深色物体。</li> <li>Normal: 正常亮度, 适用于普通物体。</li> <li>Low: 低亮度, 适用于反光物体。</li> </ul>
调节说明	根据物体类型调节。

### 编码模式

参数说明	选择投射的结构光的样式。
可见级别	专家、大师
参数取值	<ul style="list-style-type: none"> <li>Fast: 适用于不透明的物体, 采集速度快, 但深度数据质量稍差。</li> <li>Accurate: 适用于不透明的物体, 深度数据质量高, 但采集速度较慢。</li> </ul>
调节说明	根据目标物体类型以及对数据质量和采集速度的实际需求调节。

目标物体为非透明物体时, 其他条件相同, 编码模式分别选择Accurate和Fast时获取的点云如下:



## 点云后处理

调节点云后处理分组下的参数，可提升点云质量。

### 调参原则

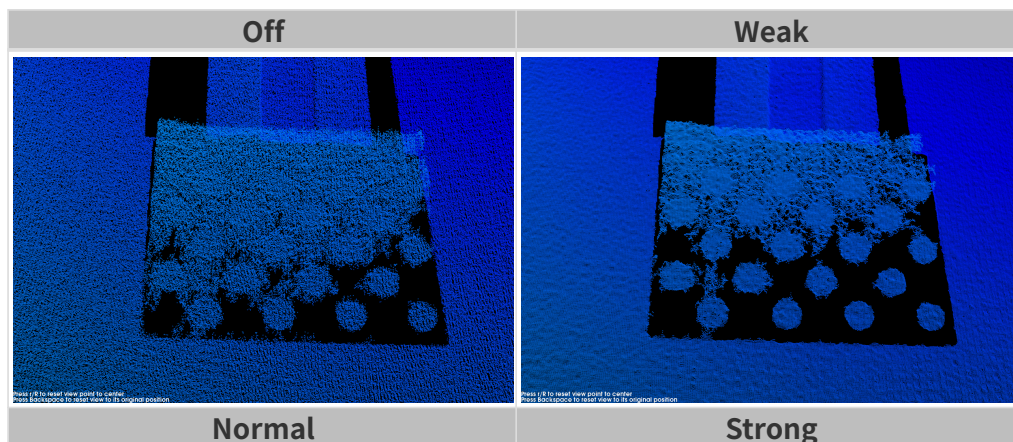
调节点云后处理参数时，遵循以下调参原则可减少相机采集时间，优化节拍。

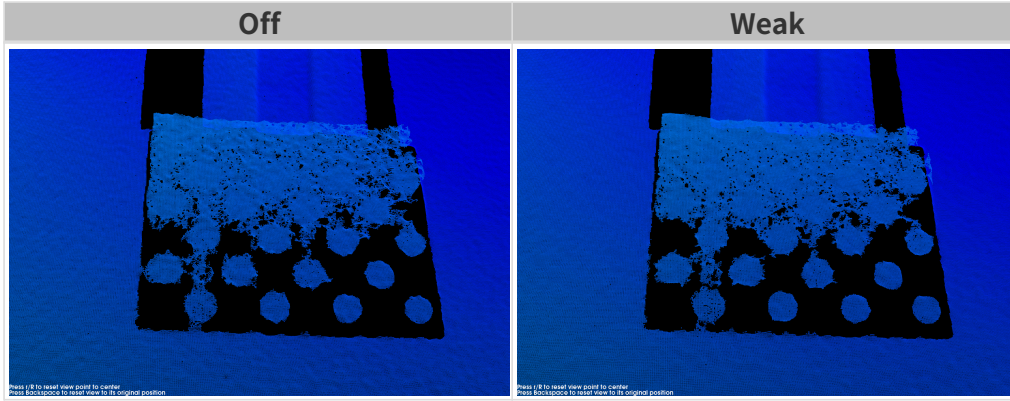
1. 优先调节**离群点去除**。该参数各强度计算时间基本相同，即使使用较高强度也不会增加太多计算时间。
2. 建议使用低强度**表面平滑**与**噪点去除**。这两个参数强度越高，计算时间越长。

### 表面平滑

<b>参数说明</b>	可减少点云的深度波动，使点云更接近真实的物体表面，但会损失部分物体表面细节。
<b>可见级别</b>	初级、专家、大师
<b>参数取值</b>	<ul style="list-style-type: none"> <li>• Off</li> <li>• Weak</li> <li>• Normal</li> <li>• Strong</li> </ul>
<b>调节说明</b>	<ul style="list-style-type: none"> <li>• <b>表面平滑</b>强度越高，损失的物体表面细节越多；强度越低，损失的物体表面细节越少。</li> <li>• <b>表面平滑</b>强度越高，计算时间越长；强度越低，计算时间越短。</li> </ul>

其他条件相同，仅**表面平滑**强度不同的点云对比如下：

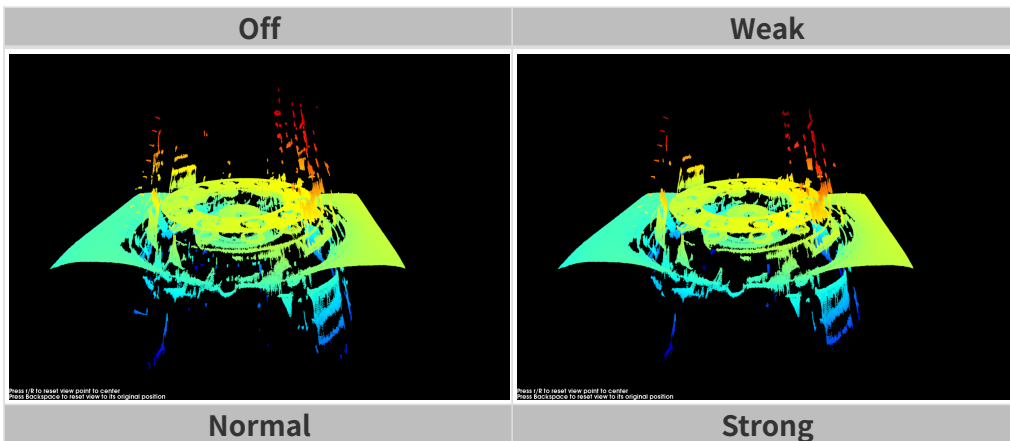




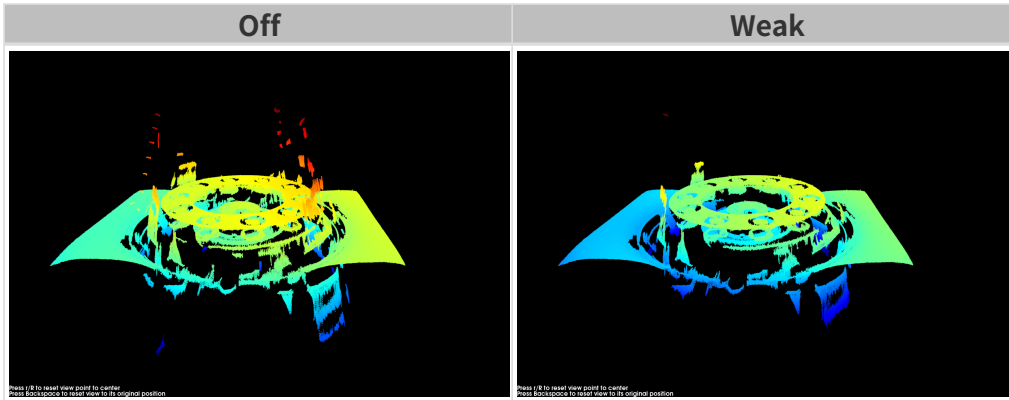
### 离群点去除

<b>参数说明</b>	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。
<b>可见级别</b>	初级、专家、大师
<b>参数取值</b>	<ul style="list-style-type: none"> <li>• Off</li> <li>• Weak</li> <li>• Normal</li> <li>• Strong</li> </ul>
<b>调节说明</b>	<ul style="list-style-type: none"> <li>• 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。</li> <li>• 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。</li> </ul>

其他条件相同，仅离群点去除强度不同的点云对比如下：



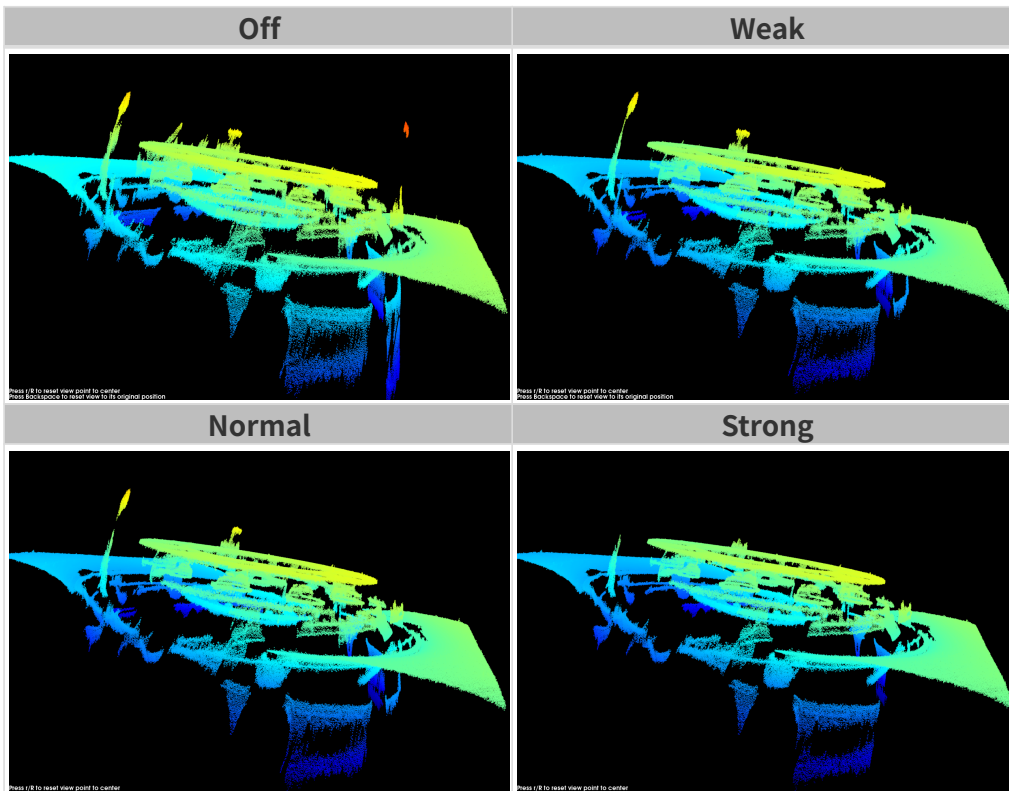




### 噪点去除

参数说明	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。
可见级别	初级、专家、大师
参数取值	<ul style="list-style-type: none"> <li>• Off</li> <li>• Weak</li> <li>• Normal</li> <li>• Strong</li> </ul>
调节说明	<ul style="list-style-type: none"> <li>• 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。</li> <li>• 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。</li> </ul>

其他条件相同，仅噪点去除强度不同的点云对比如下：





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。

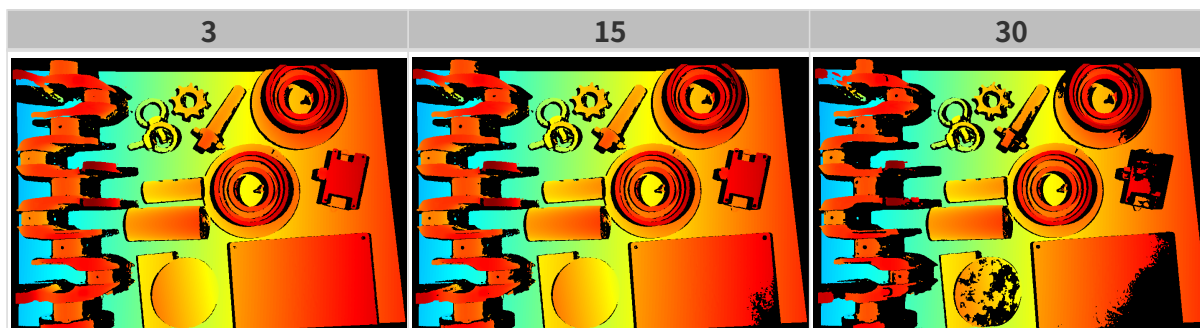
## 边缘保持

参数说明	在进行表面平滑时保持物体边缘的锐利度。
可见级别	大师
参数取值	<ul style="list-style-type: none"> <li>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。</li> <li>Normal：在保持边缘的同时达到较好的表面平滑效果。</li> <li>Smooth：不进行边缘保持。表面平滑效果最好，但物体边缘会失真。</li> </ul>
调节说明	根据对工件边缘特征的需求调节。

## 条纹对比度阈值

参数说明	用于去除点云中的噪点。如调节 <b>离群点去除</b> 与 <b>噪点去除</b> 后，仍无法获得需要的点云，再调节该参数。
可见级别	初级、专家、大师
参数取值	1~100
调节说明	<ul style="list-style-type: none"> <li>该参数值越大，去除的点越多；值越小，去除的点越少。</li> <li>调大该参数可以去除点云中的噪点，但也可能造成较暗物体对应的点丢失。</li> </ul>

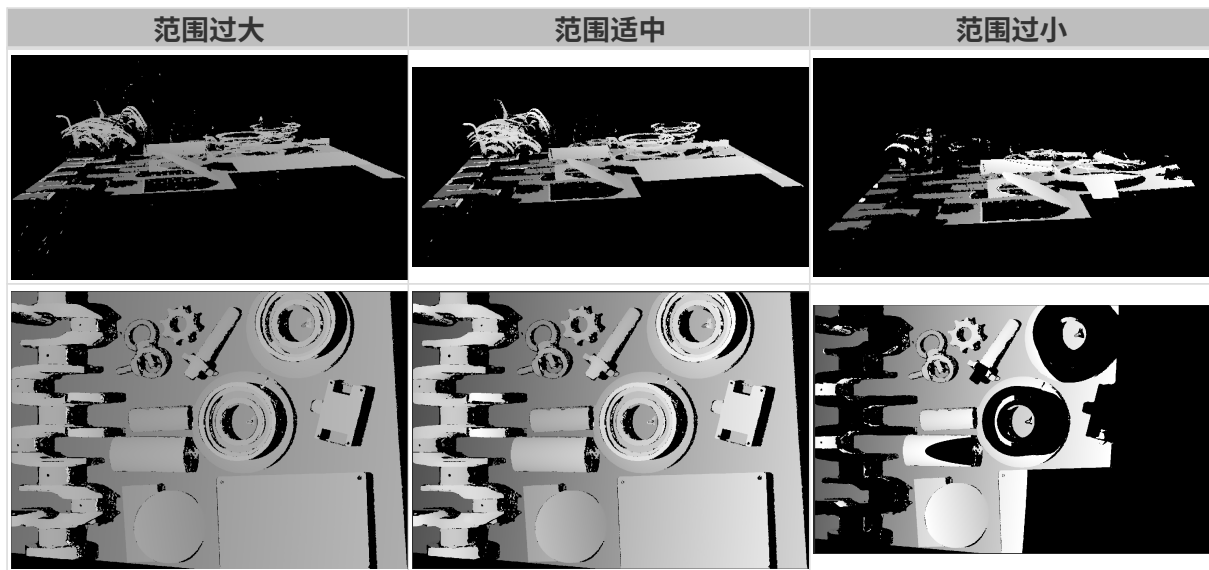
其他条件相同，仅**条纹对比度阈值**不同的点云对比如下：



## 深度范围

参数说明	设置Z向感兴趣区域。在相机的工作距离范围内设置 <b>深度范围</b> ，可滤除 <b>深度范围</b> 外的数据。
可见级别	初级、专家、大师
参数取值	<ul style="list-style-type: none"> <li>下限：1~4000mm</li> <li>上限：1~5000mm</li> </ul>
调节说明	<ul style="list-style-type: none"> <li><b>深度范围</b>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。</li> <li>如何设置详见下方的<a href="#">设置深度范围</a>。</li> </ul>

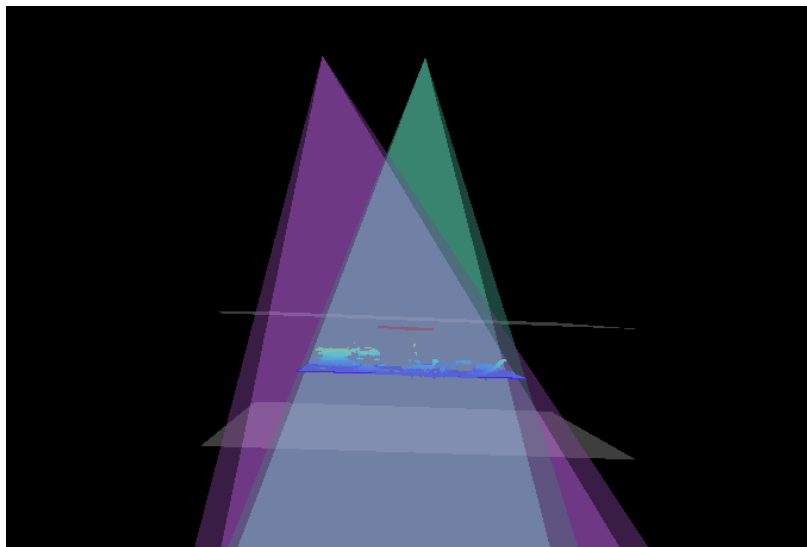
不同深度范围效果对比：



### 设置深度范围

执行以下步骤调节深度范围：

1. 双击深度范围右侧的[ 编辑 ]，打开设置深度范围界面。
2. 点击右边栏最上方的[ 更新点云 ]，获取最新的点云。
3. **调整点云的位置**：调整到可看到代表深度范围上下限的两个灰色长方形即可。



4. 调节深度范围：拖动右侧滑动条上的滑块，大致调节深度范围。再输入数值，精准调节深度范围。



判断深度范围是否合适：所有必需的物体特征均位于两个灰色长方形之间，大多数噪点和离群点位于该区域之外。

5. 设置完成后，单击右下角的[ 保存 ]。



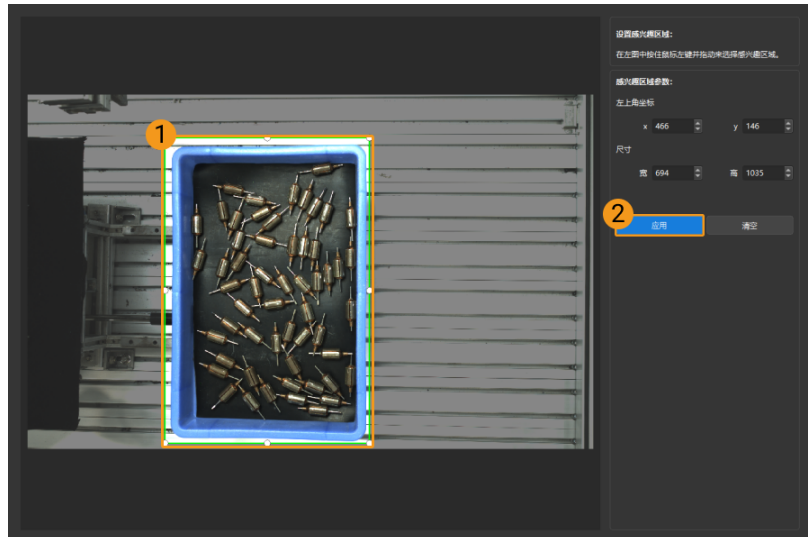
- 单击[ **推荐值** ], 可将**深度范围**调整为当前相机的推荐工作距离。
- 单击[ **重置** ], 可将**深度范围**恢复为上次保存的数值。

## 感兴趣区域

参数说明	设置深度图和点云在XOY平面上的感兴趣区域, 所选区域外的点将被移除。
可见级别	初级、专家、大师
参数取值	无
调节说明	详见下方 <a href="#">设置感兴趣区域</a> 。

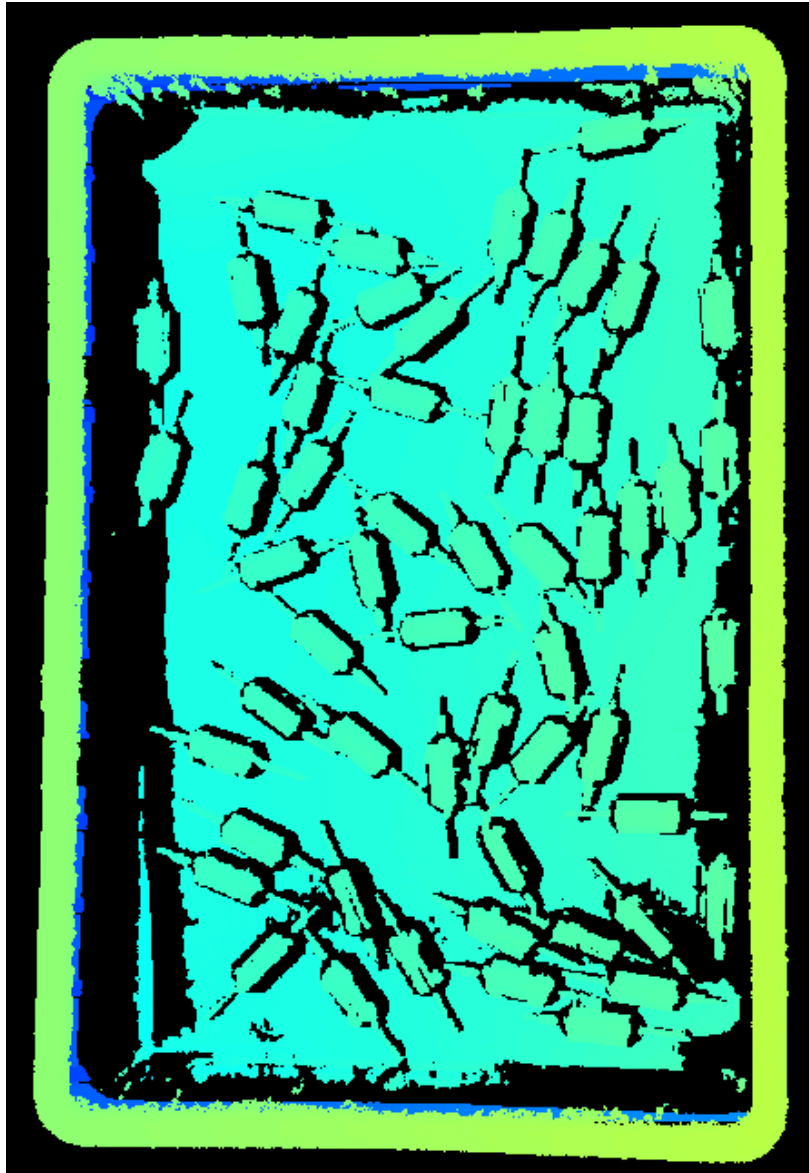
## 设置感兴趣区域

1. 双击**感兴趣区域**右侧的[ **编辑** ], 进入**设置感兴趣区域**页面。
2. 在左侧选择并调节感兴趣区域。拖拽选框可调节位置, 拖拽白色锚点可调节大小。
3. 单击[ **应用** ], 应用设置的感兴趣区域。



- 单击[ **清空** ], 可清除当前设置的感兴趣区域。
- DEEP (V4) 与LSR (V4) 系列, 在此界面显示的是2D图 (深度源)。如果图像过暗或过亮, 请调节**2D图 (深度源) 曝光模式**。

4. 重新采集图像, 查看深度图或点云, 确认所设感兴趣区域的效果。



### 5.3.4. UHP-140参数

本章介绍UHP-140相机的参数。参数按照影响的数据类型拆分为2D图参数及深度图与点云参数。

#### 2D图参数

2D图应不过亮或过暗，可看清目标物体的表面特征。

2D参数分组下的参数及3D参数分组下的**相机增益**影响2D图质量。



彩色相机采集图像时，如因现场光照条件导致图像颜色与实际差别较大，请调节**白平衡**。详细操作请参考[调节白平衡](#)。

#### 2D参数

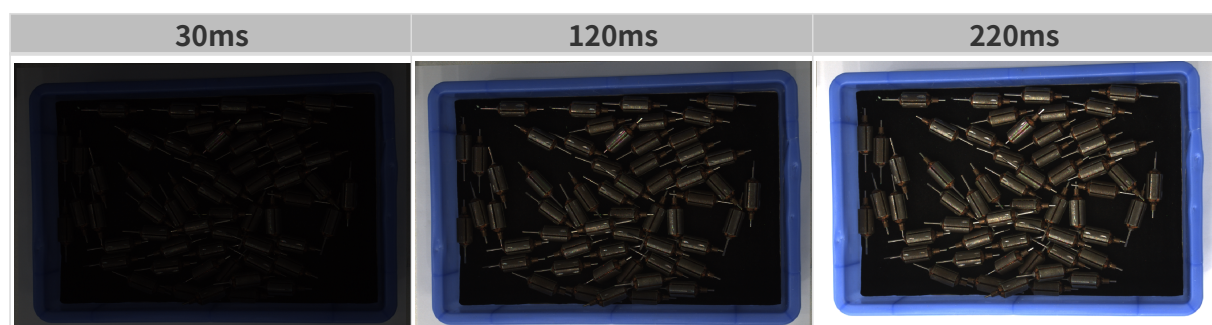
## 曝光模式

<b>参数说明</b>	设置拍摄2D图时的曝光模式。
<b>可见级别</b>	初级、专家、大师
<b>参数取值</b>	<ul style="list-style-type: none"> <li>固定曝光：设置固定曝光时间，常用于稳定的光照条件。</li> <li>自动曝光：自动调整曝光时间，常用于多变的光照条件。</li> <li>HDR：设置多个曝光时间并融合所有图像，常用于颜色或材质多样的物体。</li> <li>Flash：使用投影机补光，常用于较暗的环境。</li> </ul>
<b>调节说明</b>	选择不同选项后， <b>2D参数</b> 分组中将显示不同的参数供调节： <ul style="list-style-type: none"> <li>固定曝光：显示<b>曝光时间</b>。</li> <li>自动曝光：显示<b>灰度值</b>与<b>自动曝光感兴趣区域</b>。</li> <li>HDR：显示<b>色调映射</b>与<b>曝光时间序列</b>。</li> <li>Flash：无需调节其他参数。采集图像时将自动补光。</li> </ul>

### 固定曝光：曝光时间

<b>参数说明</b>	影响图像亮度。 <ul style="list-style-type: none"> <li>曝光时间越长，图像亮度越高。</li> <li>曝光时间越短，图像亮度越低。</li> </ul>
<b>可见级别</b>	初级、专家、大师
<b>参数取值</b>	0.1~999ms
<b>调节说明</b>	根据2D图的质量调节。2D图应不过亮或过暗，可看清目标物体的表面特征。 <ul style="list-style-type: none"> <li>较暗的环境下通常使用较长曝光时间。</li> <li>较亮的环境下通常使用较短曝光时间。</li> </ul>

其他条件相同，仅**曝光时间**不同的2D图对比如下：

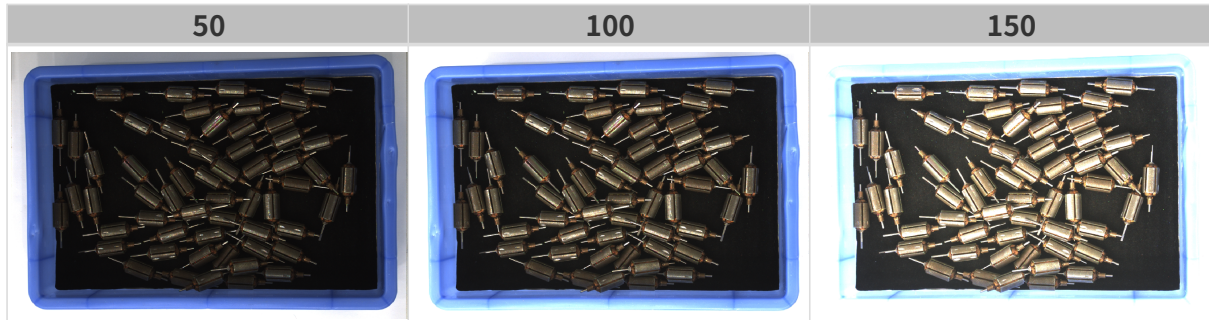


### 自动曝光：灰度值

<b>参数说明</b>	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。
-------------	-------------------------------

可见级别	初级、专家、大师
参数取值	0~255
调节说明	无

其他条件相同，仅灰度值不同的2D图对比如下：

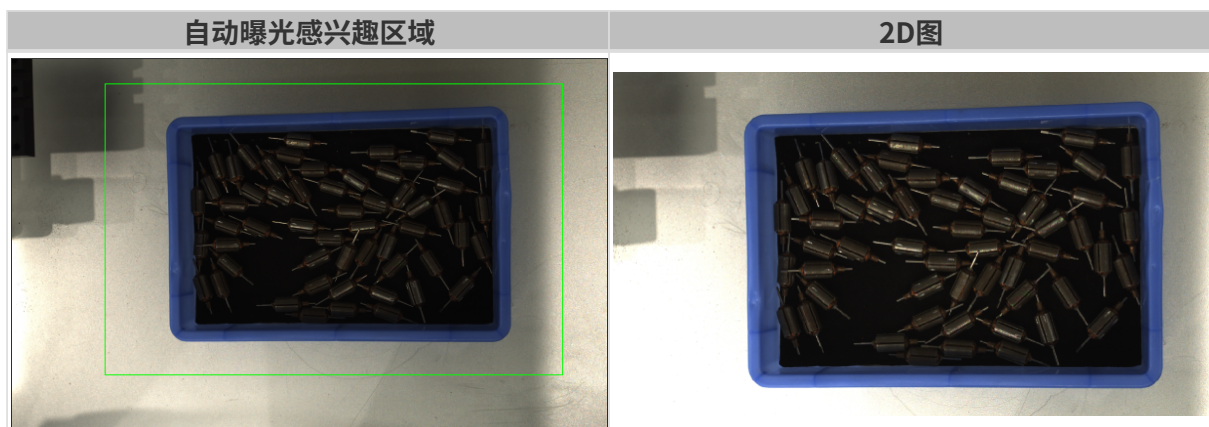


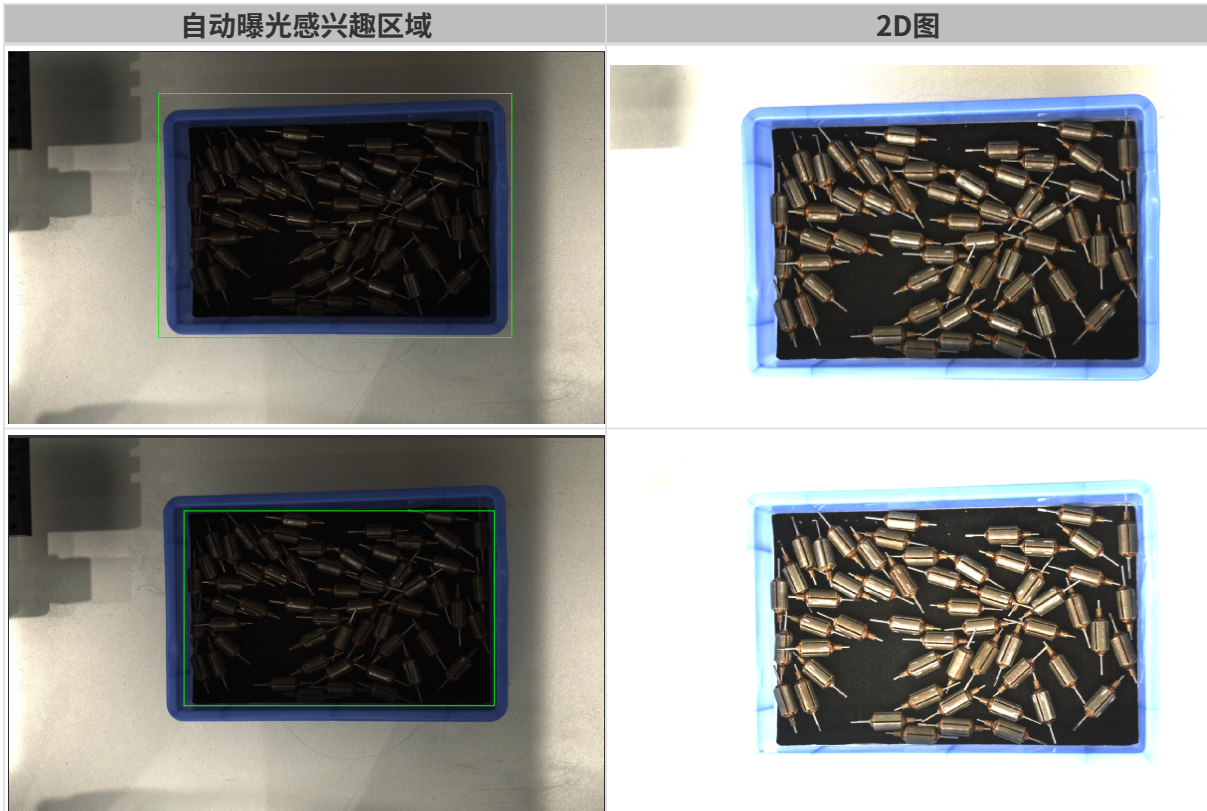
黑白图像的灰度值相当于图像亮度；彩色图像的灰度值相当于每个颜色通道的亮度。

#### 自动曝光：自动曝光感兴趣区域

参数说明	<ul style="list-style-type: none"> <li>相机参照该区域内的光照、物体颜色等自动调整曝光时间。</li> <li>未设置感兴趣区域时，相机根据整个视野自动调整曝光时间。</li> </ul>
可见级别	初级、专家、大师
参数取值	无
调节说明	详见下方 <a href="#">设置自动曝光感兴趣区域</a> 。

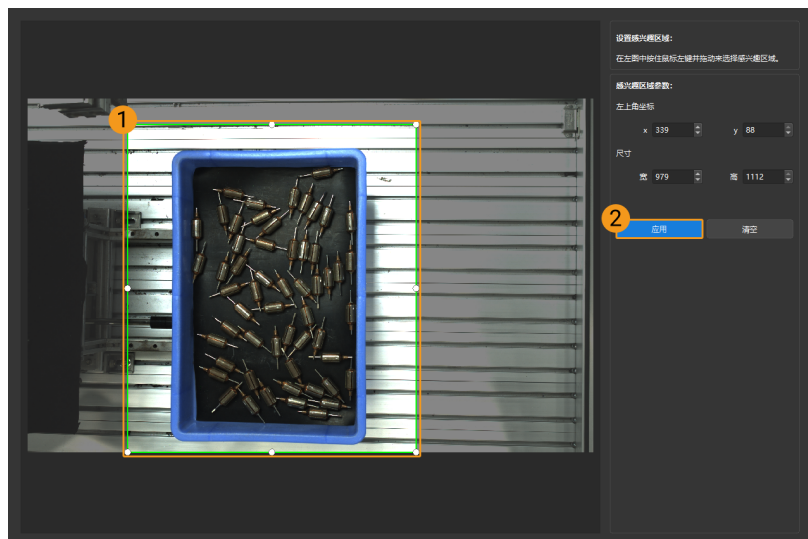
其他条件相同，仅自动曝光感兴趣区域不同的2D图对比如下：





### 设置自动曝光感兴趣区域

1. 双击自动曝光感兴趣区域右侧的[编辑]，进入设置感兴趣区域页面。
2. 在左侧选择并调节感兴趣区域。拖拽选框可调节位置，拖拽白色锚点可调节大小。
3. 单击[应用]，应用设置的自动曝光感兴趣区域。



单击[清空]，可清除当前设置的自动曝光感兴趣区域。

4. 重新采集图像并查看2D图，确认自动曝光的效果。




## HDR：色调映射

参数说明	可使图像看起来更自然。如2D图和真实物体相差较大，可勾选该参数。
可见级别	初级、专家、大师
参数取值	<ul style="list-style-type: none"> <li>• False</li> <li>• True</li> </ul>
调节说明	勾选即可开启色调映射。

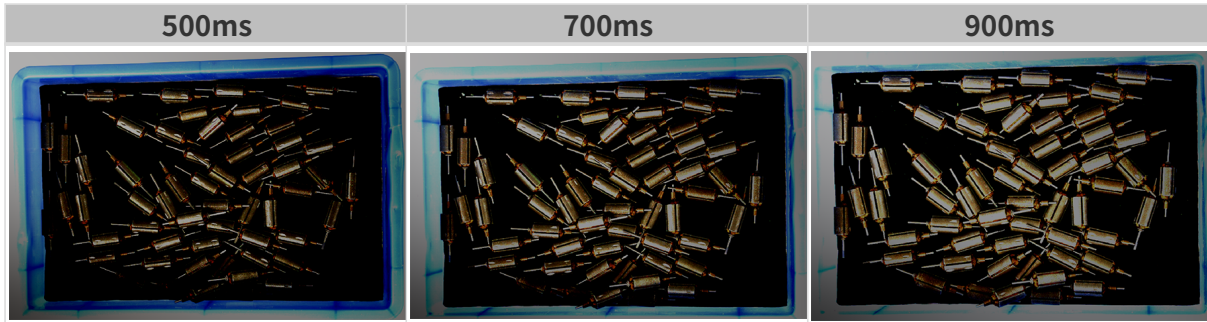
色调映射开启前后的2D图对比如下：



## HDR：曝光时间序列

参数说明	设置多个曝光时间，并将所拍摄的图像融合为一张保留更多明暗细节的2D图。
可见级别	初级、专家、大师
参数取值	无
调节说明	<ol style="list-style-type: none"> <li>1. 双击<b>曝光时间序列</b>右侧的[编辑]，进入曝光时间序列的编辑界面。</li> <li>2. 单击[+]，新增并设置曝光时间。</li> <li>3. 如需删除某个曝光时间，选中该曝光时间并单击[-]。</li> <li>4. 编辑完成后，单击[应用]使曝光时间序列生效。</li> </ol> <div style="display: flex; align-items: center; margin-top: 10px;"> <div style="margin-right: 10px;">  </div> <ul style="list-style-type: none"> <li>◦ [取消]：不保存当前设置，直接退出该页面。</li> <li>◦ [重置]：清空所有的曝光时间。</li> </ul> </div>

设置单个曝光时间时，不同曝光时间的2D图对比如下：



使用以上曝光时间组合成不同的曝光时间序列时，不同序列的2D图对比如下：



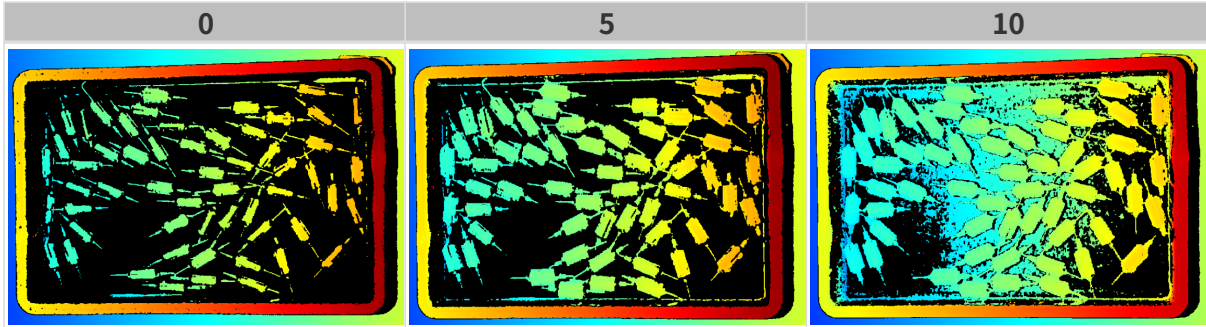
### 3D参数

#### 相机增益

参数说明	用于增加图像亮度，但可能会引入噪点。 <div style="display: flex; align-items: center; margin-top: 5px;"> <span style="font-size: 1.2em; margin-right: 5px;">i</span> <span>该参数通过改变用于计算深度数据的图像的亮度影响深度图和点云质量。</span> </div>
可见级别	专家、大师
参数取值	0~16dB
调节说明	当设置曝光时间无法达到期望亮度时，可以使用该参数。

其他条件相同，仅相机增益不同的2D图和深度图对比如下：





## 深度图与点云参数

深度图与点云中，所需的数据应完整。



根据实际需求判断所需数据的范围。例如，需通过碗沿抓取碗口向上摆放的金属碗时，一般只需确保碗沿部分的数据完整。

以下分组中的参数影响深度图与点云质量。

参数分组	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

该分组下的参数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。

使用**曝光助手**可获得推荐的曝光参数组合。双击**3D参数**右侧的[自动设置]，即可打开**曝光助手**。

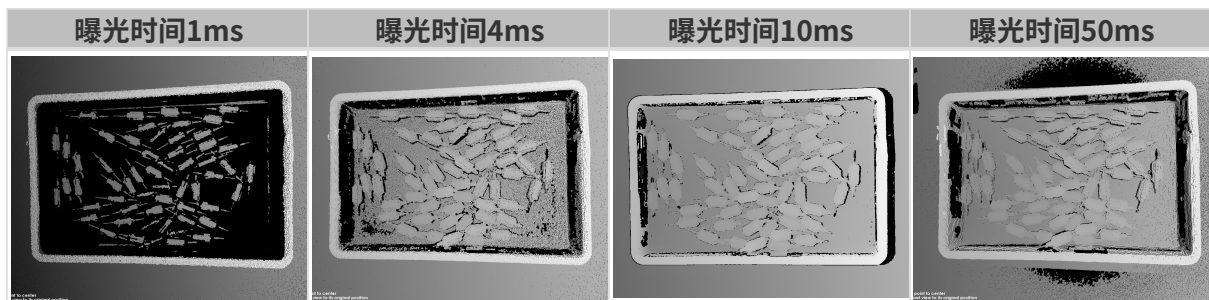
## 曝光次数

<b>参数说明</b>	设置 <b>曝光时间</b> 的个数。
<b>可见级别</b>	初级、专家、大师
<b>参数取值</b>	1~3
<b>调节说明</b>	<ul style="list-style-type: none"> <li>曝光次数大于1时，需设置多个<b>曝光时间</b>。</li> <li>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，但也将延长处理时间。</li> <li>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。</li> </ul>

## 曝光时间

参数说明	设置采集深度信息时的曝光时间，需设置的 <b>曝光时间</b> 个数取决于 <b>曝光次数</b> 的值。
可见级别	初级、专家、大师
参数取值	<ul style="list-style-type: none"> <li>取值范围：0.1~99ms</li> </ul>
调节说明	<ul style="list-style-type: none"> <li>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。</li> <li>曝光时间过长或过短都会导致信息缺失。</li> <li>激光相机的曝光时间须为4的倍数，输入值不为4的倍数时将被自动调整。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。</li> </ul>

不同**曝光时间**对比图如下：



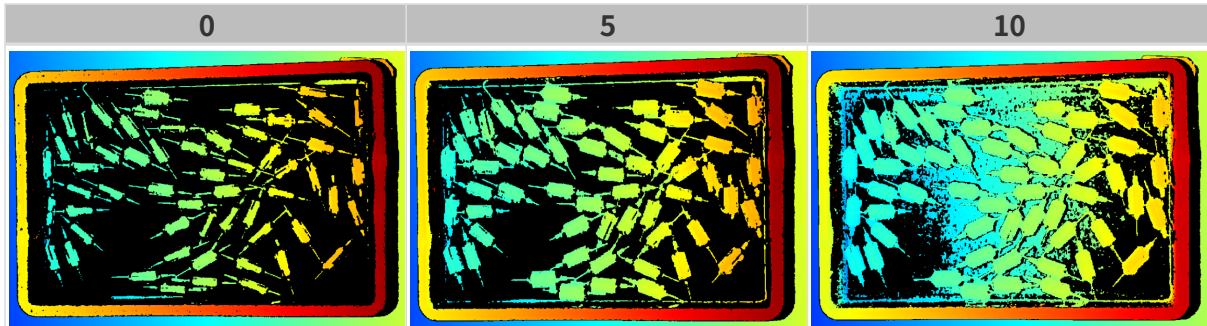
黑色部分缺少对应的物体点云。

## 相机增益

参数说明	用于增加图像亮度，但可能会引入噪点。
	 该参数通过改变 <b>用于计算深度数据的图像</b> 的亮度影响深度图和点云质量。
可见级别	专家、大师
参数取值	0~16dB
调节说明	当设置 <b>曝光时间</b> 无法达到期望亮度时，可以使用该参数。

其他条件相同，仅**相机增益**不同的2D图和深度图对比如下：





## 投影

### 投影光亮度

参数说明	投影仪投射的结构光亮度。
可见级别	专家、大师
参数取值	<ul style="list-style-type: none"> <li>High: 高亮度, 适用于深色物体。</li> <li>Normal: 正常亮度, 适用于普通物体。</li> <li>Low: 低亮度, 适用于反光物体。</li> </ul>
调节说明	根据物体类型调节。

### 编码模式

参数说明	选择投射的结构光的样式。
可见级别	专家、大师
参数取值	<ul style="list-style-type: none"> <li>Fast: 适用于不透明的物体, 采集速度快, 但深度数据质量稍差。</li> <li>Accurate: 适用于不透明的物体, 深度数据质量高, 但采集速度较慢。</li> </ul>
调节说明	根据目标物体类型以及对数据质量和采集速度的实际需求调节。

目标物体为非透明物体时, 其他条件相同, 编码模式分别选择Accurate和Fast时获取的点云如下:



## 点云后处理

调节点云后处理分组下的参数，可提升点云质量。

### 调参原则

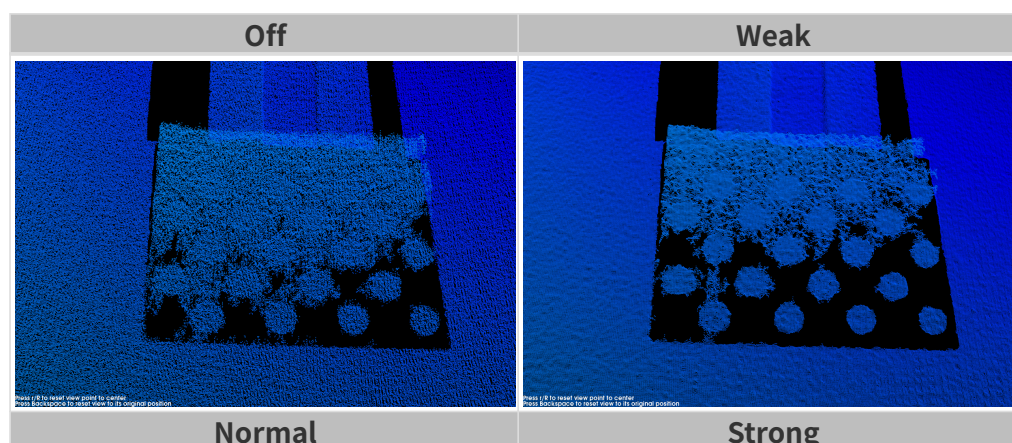
调节点云后处理参数时，遵循以下调参原则可减少相机采集时间，优化节拍。

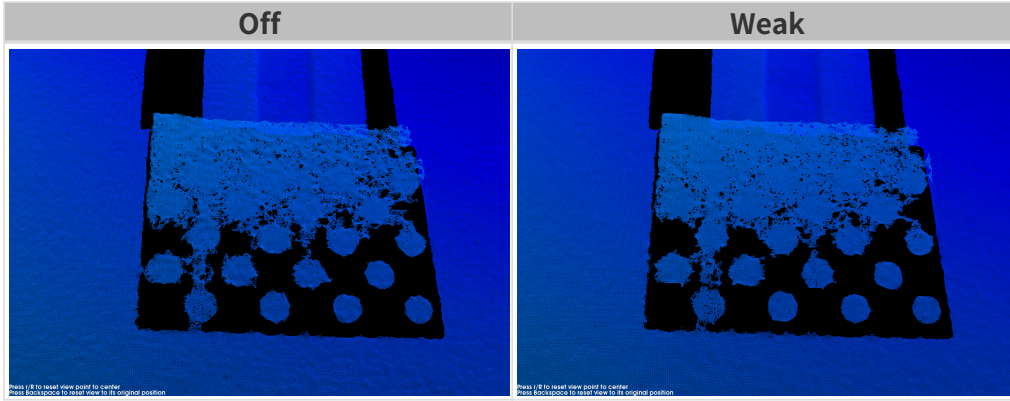
1. 优先调节**离群点去除**。该参数各强度计算时间基本相同，即使使用较高强度也不会增加太多计算时间。
2. 建议使用低强度**表面平滑与噪点去除**。这两个参数强度越高，计算时间越长。

### 表面平滑

<b>参数说明</b>	可减少点云的深度波动，使点云更接近真实的物体表面，但会损失部分物体表面细节。
<b>可见级别</b>	初级、专家、大师
<b>参数取值</b>	<ul style="list-style-type: none"> <li>• Off</li> <li>• Weak</li> <li>• Normal</li> <li>• Strong</li> </ul>
<b>调节说明</b>	<ul style="list-style-type: none"> <li>• <b>表面平滑</b>强度越高，损失的物体表面细节越多；强度越低，损失的物体表面细节越少。</li> <li>• <b>表面平滑</b>强度越高，计算时间越长；强度越低，计算时间越短。</li> </ul>

其他条件相同，仅**表面平滑**强度不同的点云对比如下：

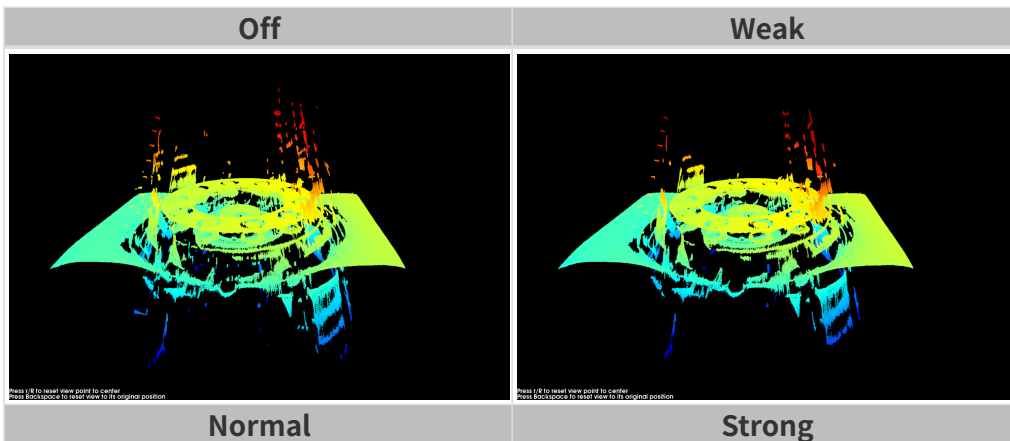


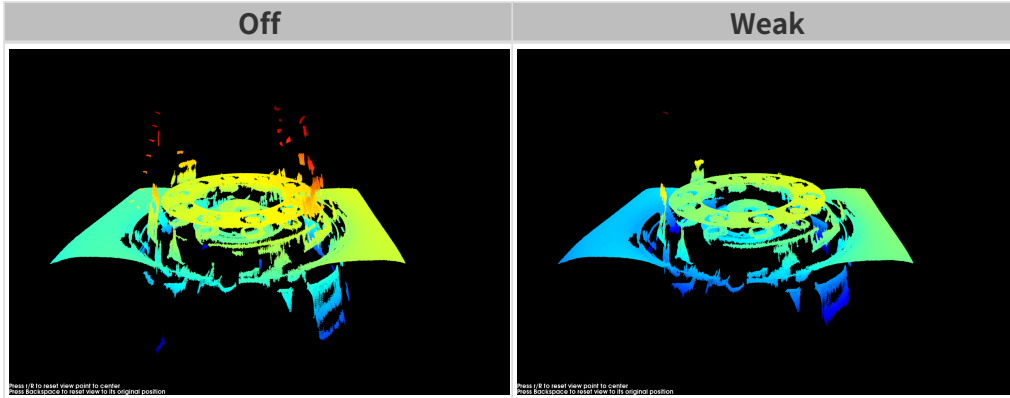


## 离群点去除

<b>参数说明</b>	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。
<b>可见级别</b>	初级、专家、大师
<b>参数取值</b>	<ul style="list-style-type: none"> <li>• Off</li> <li>• Weak</li> <li>• Normal</li> <li>• Strong</li> </ul>
<b>调节说明</b>	<ul style="list-style-type: none"> <li>• 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。</li> <li>• 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。</li> </ul>

其他条件相同，仅离群点去除强度不同的点云对比如下：

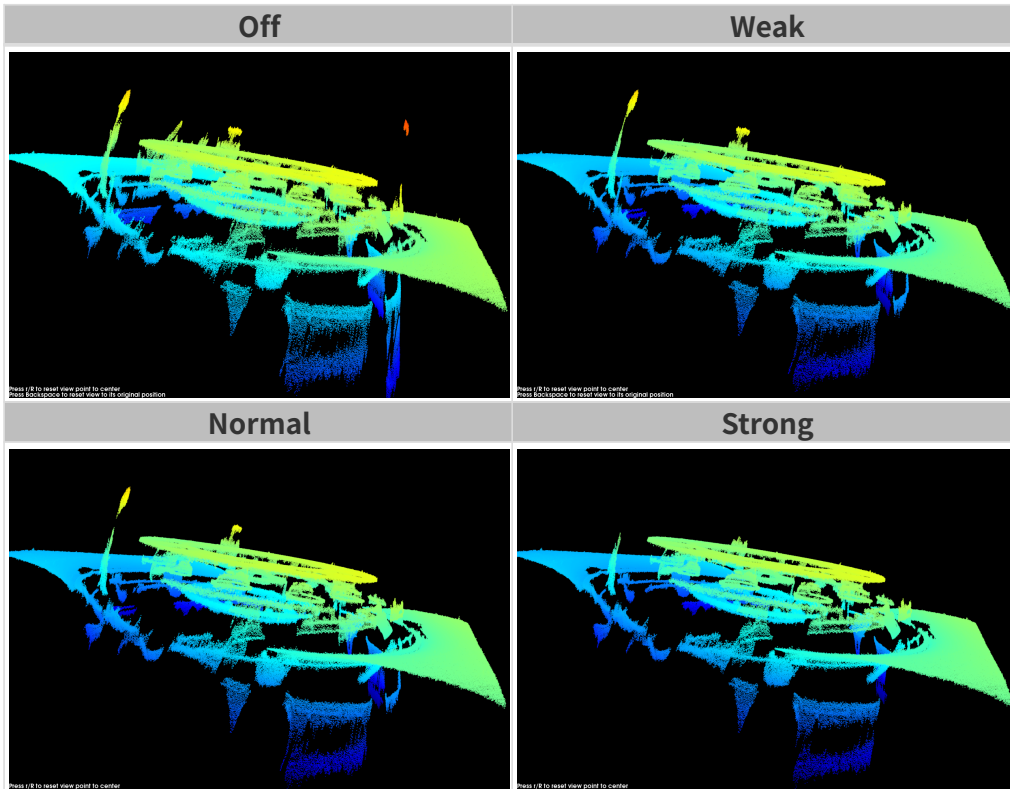




### 噪点去除

参数说明	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。
可见级别	初级、专家、大师
参数取值	<ul style="list-style-type: none"> <li>• Off</li> <li>• Weak</li> <li>• Normal</li> <li>• Strong</li> </ul>
调节说明	<ul style="list-style-type: none"> <li>• 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。</li> <li>• 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。</li> </ul>

其他条件相同，仅噪点去除强度不同的点云对比如下：







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。


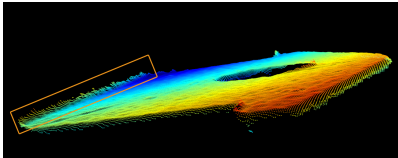
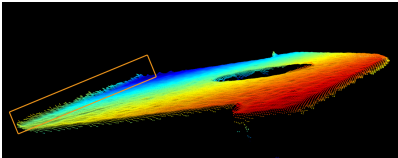
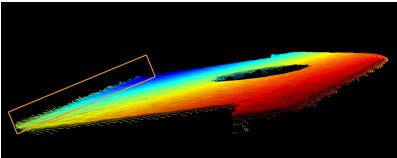
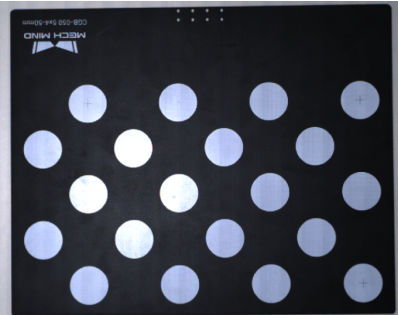
## 失真校正

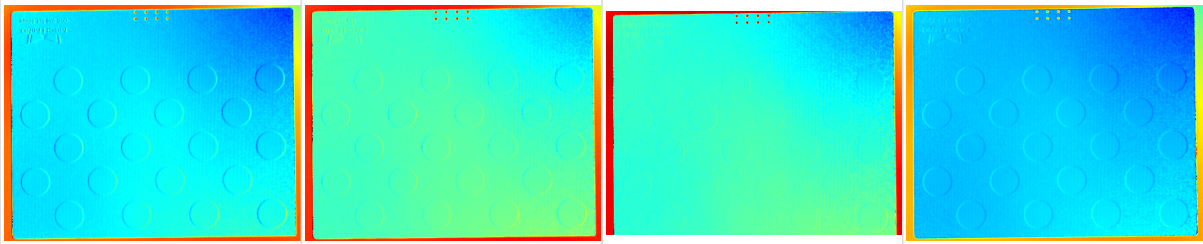
<b>参数说明</b>	校正深度数据中由物体颜色或高度的剧烈变化导致的失真。 请勾选 <b>启用失真校正</b> ，然后调节 <b>当前值</b> 。
<b>可见级别</b>	专家、大师
<b>参数取值</b>	1~10
<b>调节说明</b>	根据深度图或点云中的失真程度调节 <b>当前值</b> ，重新采集图像确认效果。



- 启用失真校正将降低采集速度。
- **当前值**过大可能导致反效果。请在调节后重新采集数据，确认该参数对深度图和点云的影响。

其他条件相同，仅**失真校正**的**当前值**不同的深度图对比如下：

目标物体为金属工件，表面平整，边缘的高度变化剧烈。			
			
不开启	7	9	
			
目标物体为标定板，表面平整，有从黑色到白色的颜色剧烈变化。			
			
不开启	3	7	10



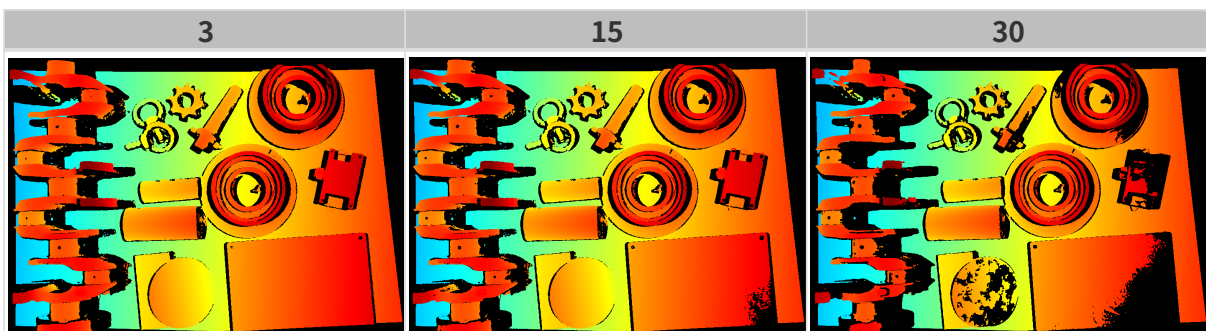
### 边缘保持

参数说明	在进行表面平滑时保持物体边缘的锐利度。
可见级别	大师
参数取值	<ul style="list-style-type: none"> <li>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。</li> <li>Normal: 在保持边缘的同时达到较好的表面平滑效果。</li> <li>Smooth: 不进行边缘保持。表面平滑效果最好, 但物体边缘会失真。</li> </ul>
调节说明	根据对工件边缘特征的需求调节。

### 条纹对比度阈值

参数说明	用于去除点云中的噪点。如调节 <b>离群点去除</b> 与 <b>噪点去除</b> 后, 仍无法获得需要的点云, 再调节该参数。
可见级别	初级、专家、大师
参数取值	1~100
调节说明	<ul style="list-style-type: none"> <li>该参数值越大, 去除的点越多; 值越小, 去除的点越少。</li> <li>调大该参数可以去除点云中的噪点, 但也可能造成较暗物体对应的点丢失。</li> </ul>

其他条件相同, 仅**条纹对比度阈值**不同的点云对比如下:

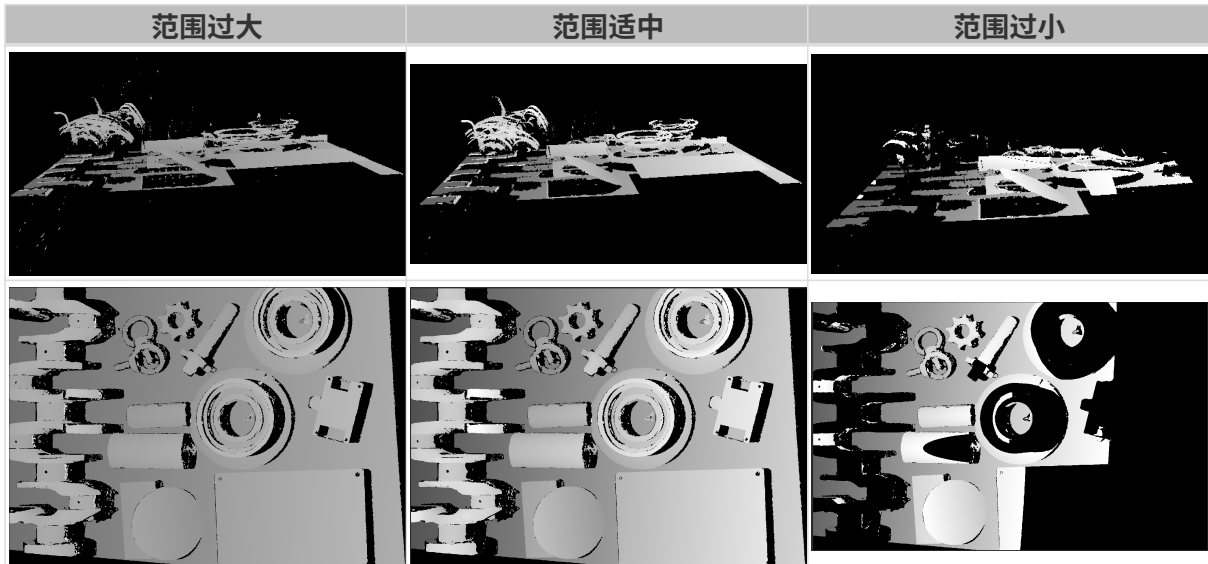


### 深度范围

参数说明	设置Z向感兴趣区域。在相机的工作距离范围内设置 <b>深度范围</b> , 可滤除 <b>深度范围</b> 外的数据。
可见级别	初级、专家、大师

参数取值	<ul style="list-style-type: none"> <li>• 下限：1~4000mm</li> <li>• 上限：1~5000mm</li> </ul>
调节说明	<ul style="list-style-type: none"> <li>• <b>深度范围</b>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。</li> <li>• 如何设置详见下方的<a href="#">设置深度范围</a>。</li> </ul>

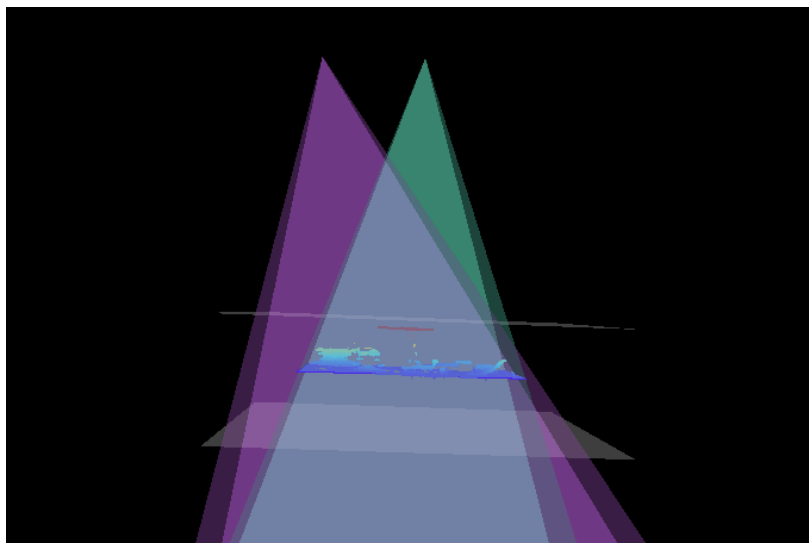
不同**深度范围**效果对比：



### 设置深度范围

执行以下步骤调节**深度范围**：

1. 双击**深度范围**右侧的[编辑]，打开设置深度范围界面。
2. 点击右边栏最上方的[更新点云]，获取最新的点云。
3. **调整点云的位置**：调整到可看到代表**深度范围**上下限的两个灰色长方形即可。



4. 调节深度范围：拖动右侧滑动条上的滑块，大致调节**深度范围**。再输入数值，精准调节**深度范围**。



判断深度范围是否合适：所有必需的物体特征均位于两个灰色长方形之间，大多数噪点和离群点位于该区域之外。

5. 设置完成后，单击右下角的[ **保存** ]。



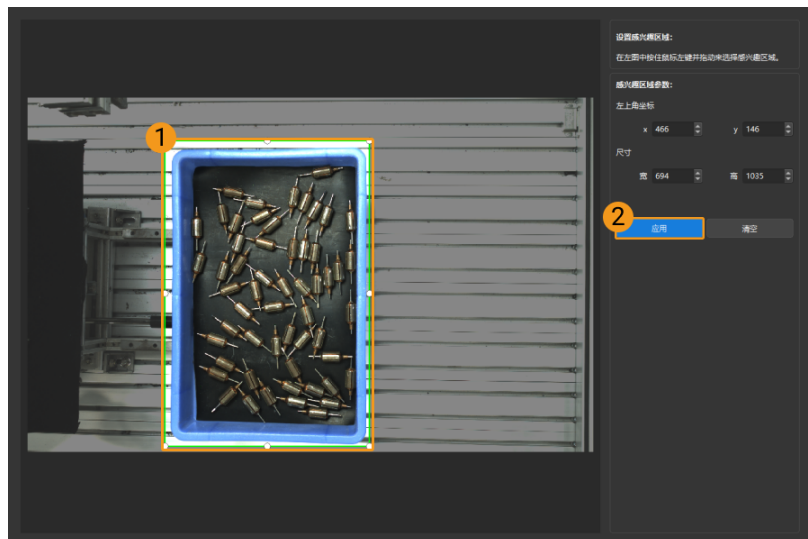
- 单击[ **推荐值** ]，可将**深度范围**调整为当前相机的推荐工作距离。
- 单击[ **重置** ]，可将**深度范围**恢复为上次保存的数值。

## 感兴趣区域

参数说明	设置深度图和点云在XOY平面上的感兴趣区域，所选区域外的点将被移除。
可见级别	初级、专家、大师
参数取值	无
调节说明	详见下方 <b>设置感兴趣区域</b> 。

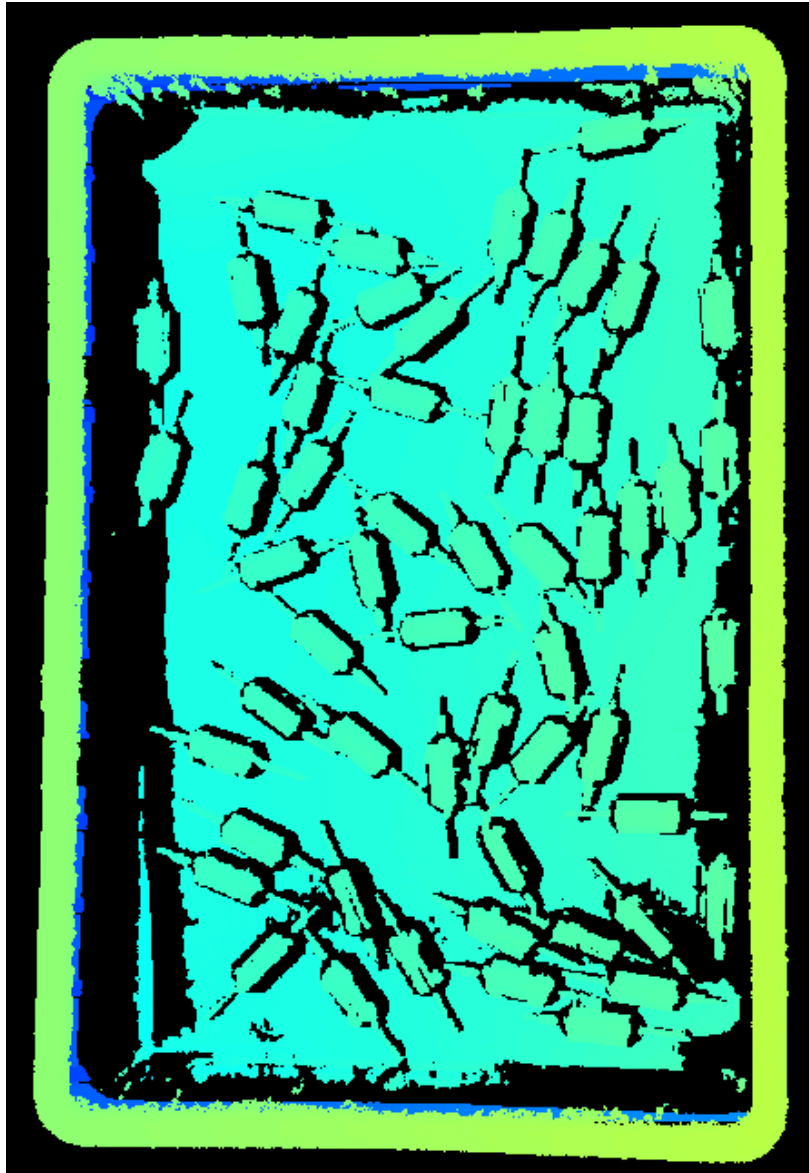
## 设置感兴趣区域

1. 双击**感兴趣区域**右侧的[ **编辑** ]，进入**设置感兴趣区域**页面。
2. 在左侧选择并调节感兴趣区域。拖拽选框可调节位置，拖拽白色锚点可调节大小。
3. 单击[ **应用** ]，应用设置的感兴趣区域。



- 单击[ **清空** ]，可清除当前设置的感兴趣区域。
- DEEP (V4) 与LSR (V4) 系列，在此界面显示的是2D图（深度源）。如果图像过暗或过亮，请调节**2D图（深度源）曝光模式**。

4. 重新采集图像，查看深度图或点云，确认所设感兴趣区域的效果。



### 5.3.5. DEEP / LSR系列参数

本章介绍DEEP / LSR系列相机的参数。参数按照影响的数据类型拆分为2D图参数及深度图与点云参数。

#### 2D图参数

DEEP与LSR系列提供两种2D图：2D图（纹理）和2D图（深度源）。其使用场景如下：

2D图类型	使用场景
2D图（纹理）	为点云添加纹理
	内参检查
2D图（深度源）	设置感兴趣区域
	进行手眼标定

2D图应不过亮或过暗，可看清目标物体的表面特征。

2D参数分组下的参数及3D参数分组下的**相机增益**影响2D图质量。



彩色相机采集图像时，如因现场光照条件导致图像颜色与实际差别较大，请调节**白平衡**。详细操作请参考[调节白平衡](#)。

## 2D参数

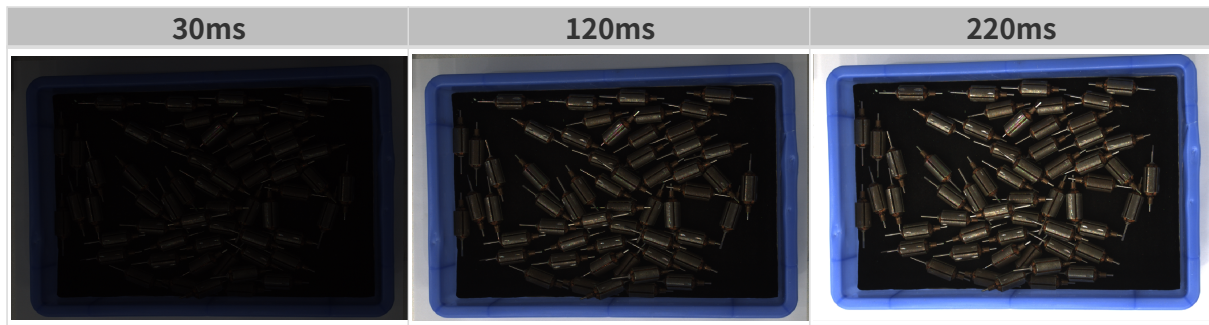
### 2D图（纹理）曝光模式

<b>参数说明</b>	设置拍摄2D图（纹理）时的曝光模式。
<b>可见级别</b>	初级、专家、大师
<b>参数取值</b>	<ul style="list-style-type: none"> <li>固定曝光：设置固定曝光时间，常用于稳定的光照条件。</li> <li>自动曝光：自动调整曝光时间，常用于多变的光照条件。</li> <li>HDR：设置多个曝光时间并融合所有图像，常用于颜色或材质多样的物体。</li> </ul>
<b>调节说明</b>	选择不同选项后， <b>2D参数</b> 分组中将显示不同的参数供调节： <ul style="list-style-type: none"> <li>固定曝光：显示<b>曝光时间</b>。</li> <li>自动曝光：显示<b>灰度值与自动曝光感兴趣区域</b>。</li> <li>HDR：显示<b>色调映射与曝光时间序列</b>。</li> </ul>

#### 固定曝光：曝光时间

<b>参数说明</b>	影响图像亮度。 <ul style="list-style-type: none"> <li>曝光时间越长，图像亮度越高。</li> <li>曝光时间越短，图像亮度越低。</li> </ul>
<b>可见级别</b>	初级、专家、大师
<b>参数取值</b>	0.1~999ms
<b>调节说明</b>	根据2D图的质量调节。2D图应不过亮或过暗，可看清目标物体的表面特征。 <ul style="list-style-type: none"> <li>较暗的环境下通常使用较长曝光时间。</li> <li>较亮的环境下通常使用较短曝光时间。</li> </ul>

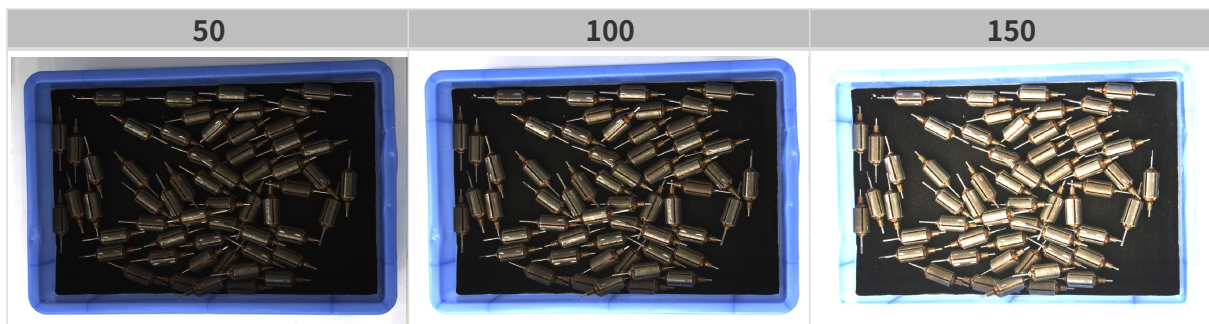
其他条件相同，仅**曝光时间**不同的2D图对比如下：



### 自动曝光：灰度值

参数说明	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。
可见级别	初级、专家、大师
参数取值	0~255
调节说明	无

其他条件相同，仅灰度值不同的2D图对比如下：

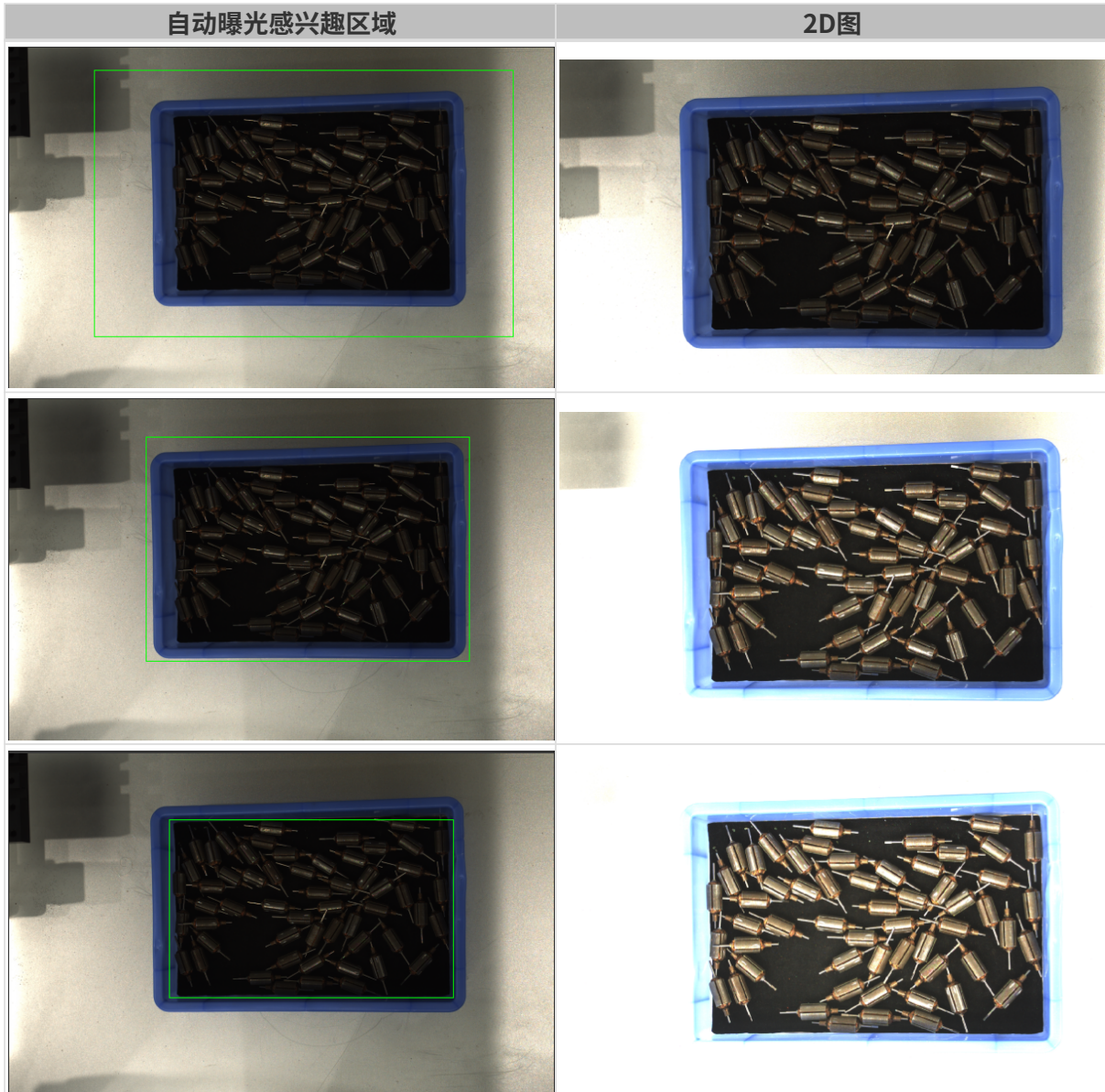


黑白图像的灰度值相当于图像亮度；彩色图像的灰度值相当于每个颜色通道的亮度。

### 自动曝光：自动曝光感兴趣区域

参数说明	<ul style="list-style-type: none"> <li>相机参照该区域内的光照、物体颜色等自动调整曝光时间。</li> <li>未设置感兴趣区域时，相机根据整个视野自动调整曝光时间。</li> </ul>
可见级别	初级、专家、大师
参数取值	无
调节说明	详见下方 <a href="#">设置自动曝光感兴趣区域</a> 。

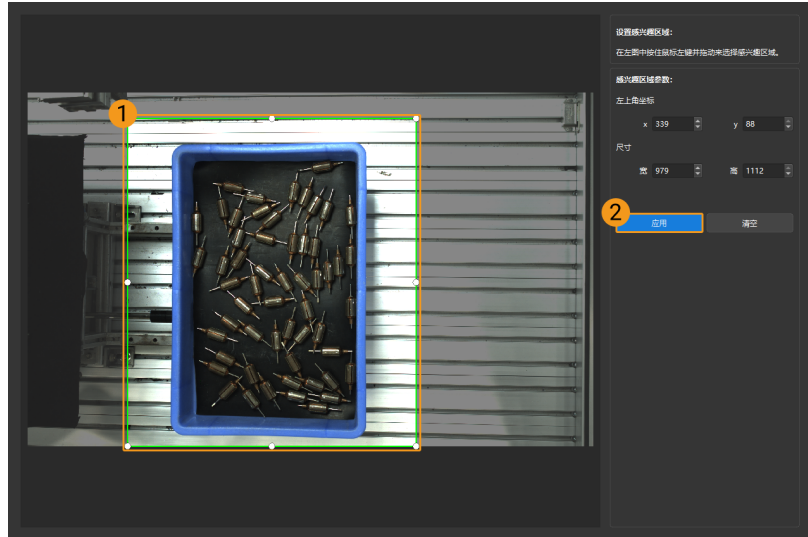
其他条件相同，仅自动曝光感兴趣区域不同的2D图对比如下：




### 设置自动曝光感兴趣区域

1. 双击自动曝光感兴趣区域右侧的[编辑]，进入设置感兴趣区域页面。
2. 在左侧选择并调节感兴趣区域。拖拽选框可调节位置，拖拽白色锚点可调节大小。
3. 单击[应用]，应用设置的自动曝光感兴趣区域。





 单击[清空], 可清除当前设置的自动曝光感兴趣区域。

4. 重新采集图像并查看2D图, 确认自动曝光的效果。

### HDR: 色调映射


参数说明	可使图像看起来更自然。如2D图和真实物体相差较大, 可勾选该参数。
可见级别	初级、专家、大师
参数取值	<ul style="list-style-type: none"> <li>False</li> <li>True</li> </ul>
调节说明	勾选即可开启色调映射。

色调映射开启前后的2D图对比如下:

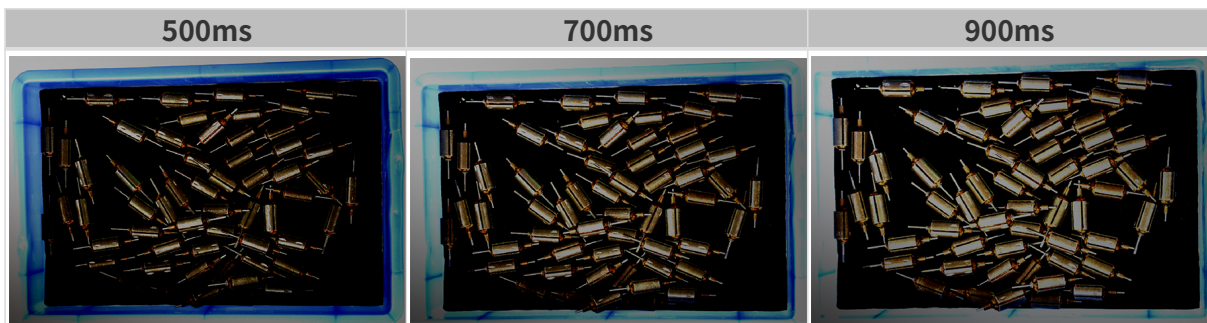


### HDR: 曝光时间序列

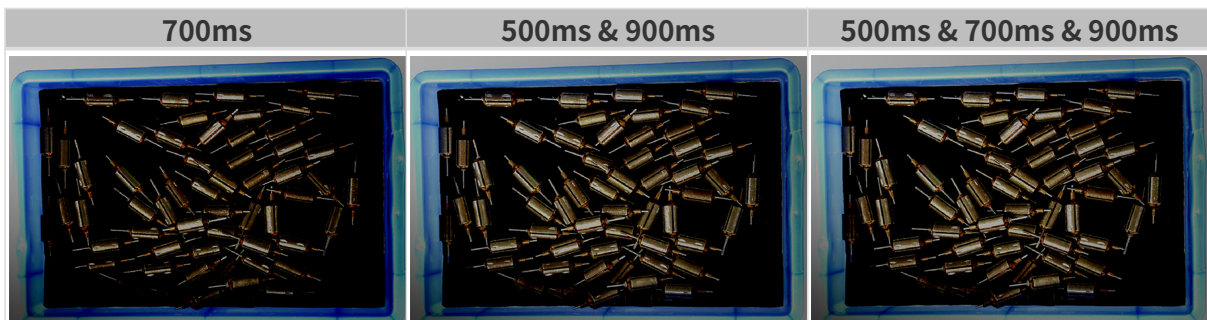
参数说明	设置多个曝光时间, 并将所拍摄的图像融合为一张保留更多明暗细节的2D图。
可见级别	初级、专家、大师
参数取值	无

调节说明	<ol style="list-style-type: none"> <li>1. 双击<b>曝光时间序列</b>右侧的[ <b>编辑</b> ]，进入曝光时间序列的编辑界面。</li> <li>2. 单击[ <b>+</b> ]，新增并设置曝光时间。</li> <li>3. 如需删除某个曝光时间，选中该曝光时间并单击[ <b>-</b> ]。</li> <li>4. 编辑完成后，单击[ <b>应用</b> ]使曝光时间序列生效。</li> </ol>
	<div style="display: flex; align-items: center;"> <div style="margin-right: 10px;">  </div> <ul style="list-style-type: none"> <li>◦ [ <b>取消</b> ]: 不保存当前设置，直接退出该页面。</li> <li>◦ [ <b>重置</b> ]: 清空所有的曝光时间。</li> </ul> </div>

设置单个曝光时间时，不同曝光时间的2D图对比如下：



使用以上曝光时间组合成不同的曝光时间序列时，不同序列的2D图对比如下：



## 2D图（深度源）曝光模式

参数说明	设置拍摄2D图（深度源）时的曝光模式。
可见级别	初级、专家、大师
参数取值	<ul style="list-style-type: none"> <li>• 固定曝光：设置固定曝光时间，常用于稳定的光照条件。</li> <li>• Flash：使用投影机补光，常用于较暗的环境。</li> </ul>
调节说明	选择不同选项后， <b>2D参数</b> 分组中将显示不同的参数供调节： <ul style="list-style-type: none"> <li>• 固定曝光：显示<b>2D图（深度源）曝光时间</b>。</li> <li>• Flash：无需调节其他参数。采集图像时将自动补光。</li> </ul>

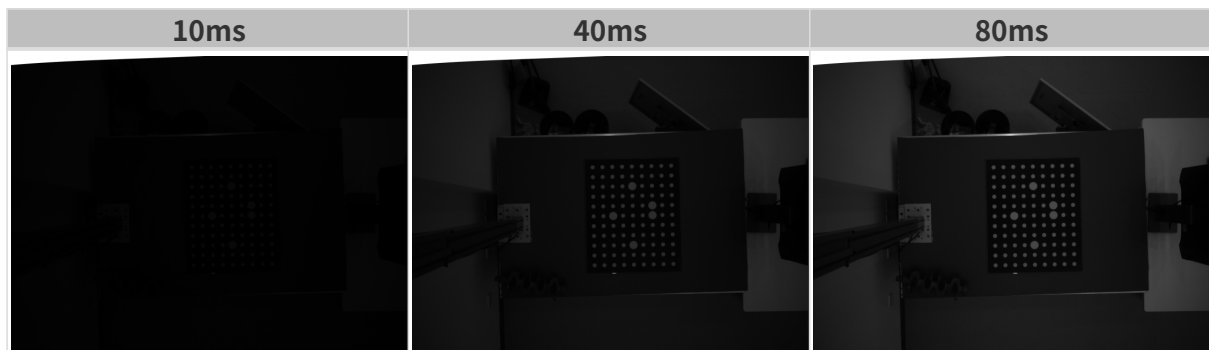


Mech-Vision连接LSR与DEEP系列相机时，Mech-Vision中的“彩色图”端口获取的是2D图（深度源）。如需调节Mech-Vision中的“彩色图”，请调节**2D图（深度源）曝光模式**。

### 固定曝光：2D图（深度源）曝光时间

	影响图像亮度。
参数说明	<ul style="list-style-type: none"> <li>曝光时间越长，图像亮度越高。</li> <li>曝光时间越短，图像亮度越低。</li> </ul>
可见级别	初级、专家、大师
参数取值	0.1~999ms
调节说明	根据2D图的质量调节。2D图应不过亮或过暗，可看清目标物体的表面特征。 <ul style="list-style-type: none"> <li>较暗的环境下通常使用较长曝光时间。</li> <li>较亮的环境下通常使用较短曝光时间。</li> </ul>

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：

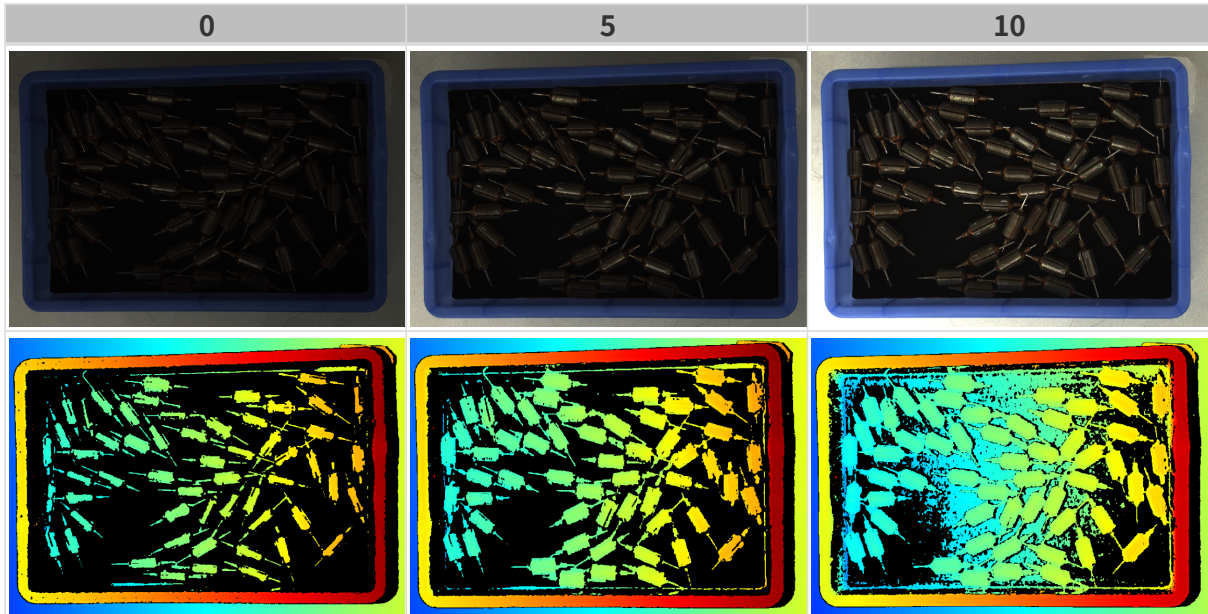


### 3D参数

#### 相机增益

参数说明	用于增加图像亮度，但可能会引入噪点。 <div style="margin-top: 10px;"> <span style="font-size: 1.2em; color: #0070C0;">i</span> 该参数通过改变<b>用于计算深度数据的图像</b>的亮度影响深度图和点云质量。                 </div>
可见级别	专家、大师
参数取值	0~16dB
调节说明	当设置 <b>曝光时间</b> 无法达到期望亮度时，可以使用该参数。

其他条件相同，仅**相机增益**不同的2D图和深度图对比如下：



### 深度图与点云参数

深度图与点云中，所需的数据应完整。



根据实际需求判断所需数据的范围。例如，需通过碗沿抓取碗口向上摆放的金属碗时，一般只需确保碗沿部分的数据完整。

以下分组中的参数影响深度图与点云质量。

参数分组	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

该分组下的参数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。

使用**曝光助手**可获得推荐的曝光参数组合。双击**3D参数**右侧的[自动设置]，即可打开**曝光助手**。

### 曝光次数

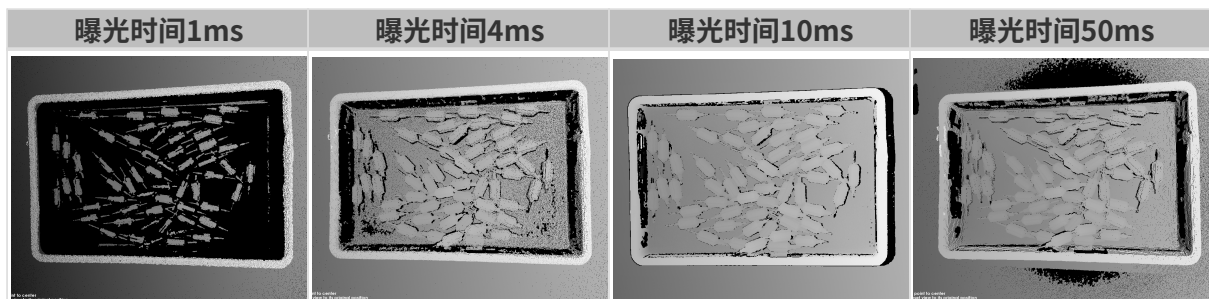
参数说明	设置 <b>曝光时间</b> 的个数。
可见级别	初级、专家、大师
参数取值	1~3

调节说明	<ul style="list-style-type: none"> <li>曝光次数大于1时，需设置多个<b>曝光时间</b>。</li> <li>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，但也将延长处理时间。</li> <li>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。</li> </ul>
------	---

## 曝光时间

参数说明	设置采集深度信息时的曝光时间，需设置的 <b>曝光时间</b> 个数取决于 <b>曝光次数</b> 的值。
可见级别	初级、专家、大师
参数取值	<ul style="list-style-type: none"> <li>取值范围：0.1~99ms</li> </ul>
调节说明	<ul style="list-style-type: none"> <li>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。</li> <li>曝光时间过长或过短都会导致信息缺失。</li> <li>激光相机的曝光时间须为4的倍数，输入值不为4的倍数时将被自动调整。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。</li> </ul>

不同**曝光时间**对比图如下：

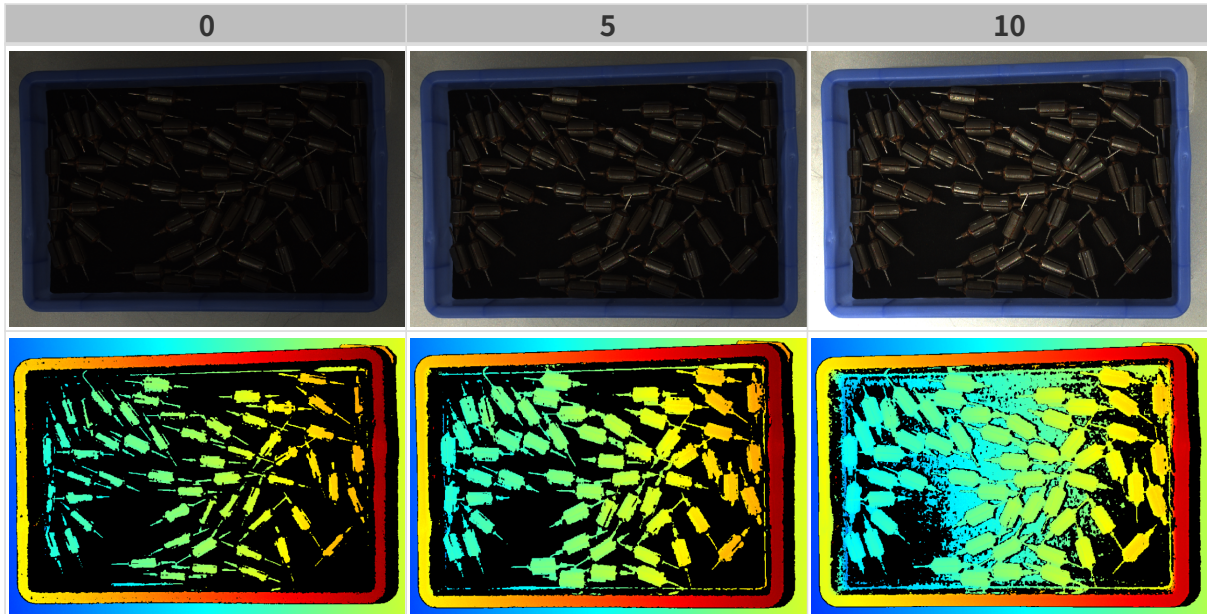


黑色部分缺少对应的物体点云。

## 相机增益

参数说明	用于增加图像亮度，但可能会引入噪点。
可见级别	专家、大师
参数取值	0~16dB
调节说明	当设置 <b>曝光时间</b> 无法达到期望亮度时，可以使用该参数。

其他条件相同，仅**相机增益**不同的2D图和深度图对比如下：



## 激光

### 编码模式

参数说明	影响采集速度与深度数据质量。
可见级别	专家、大师
参数取值	<ul style="list-style-type: none"> <li>Fast: 采集速度快, 但深度图和点云质量较差。</li> <li>Accurate: 采集速度慢, 但深度图和点云质量较好。</li> </ul>
调节说明	根据对数据质量和采集速度的实际需求调节。

其他条件相同, 仅**编码模式**不同的点云对比如下:



### 激光强度

参数说明	设置激光器的投影强度, 影响结构光的亮度。
可见级别	专家、大师
参数取值	50~100%

调节说明	一般情况下使用默认值即可。 <ul style="list-style-type: none"> <li>数值越大，结构光亮度越高；数值越小，结构光亮度越低。</li> <li>深色物体建议使用高强度，反光物体建议使用低强度。</li> </ul>
------	---

## 点云后处理

调节点云后处理分组下的参数，可提升点云质量。

### 调参原则

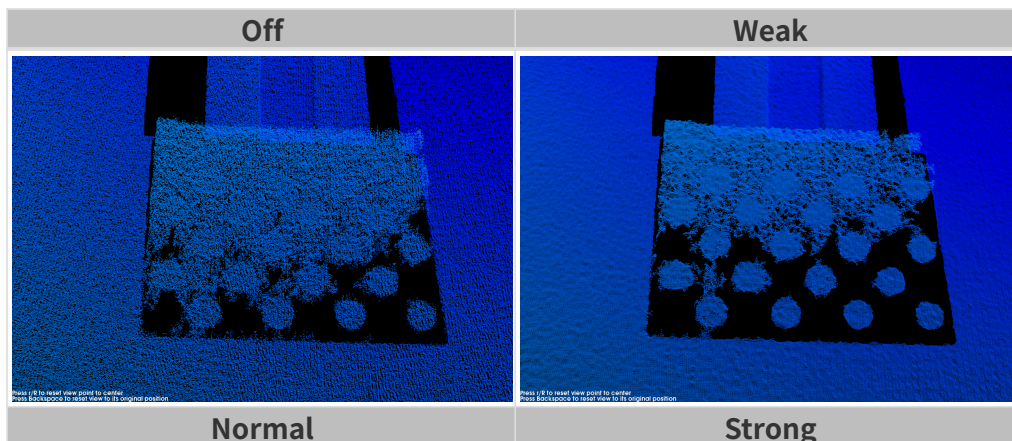
调节点云后处理参数时，遵循以下调参原则可减少相机采集时间，优化节拍。

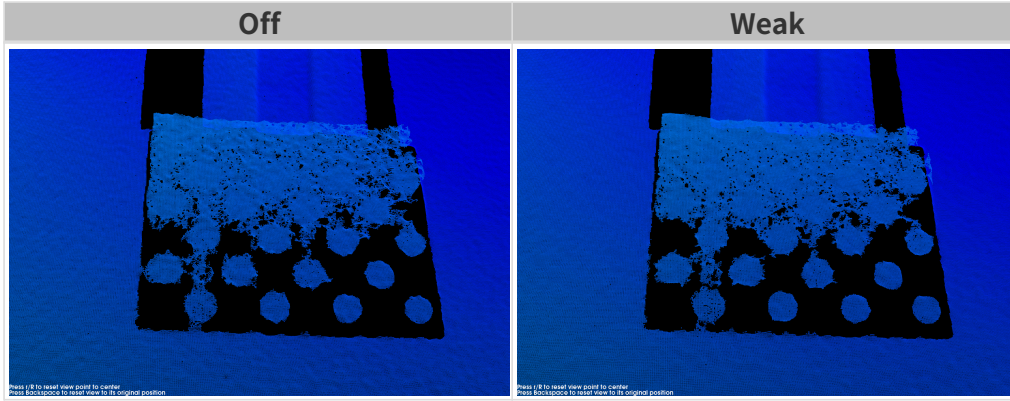
1. 优先调节**离群点去除**。该参数各强度计算时间基本相同，即使使用较高强度也不会增加太多计算时间。
2. 建议使用低强度**表面平滑与噪点去除**。这两个参数强度越高，计算时间越长。

### 表面平滑

参数说明	可减少点云的深度波动，使点云更接近真实的物体表面，但会损失部分物体表面细节。
可见级别	初级、专家、大师
参数取值	<ul style="list-style-type: none"> <li>Off</li> <li>Weak</li> <li>Normal</li> <li>Strong</li> </ul>
调节说明	<ul style="list-style-type: none"> <li><b>表面平滑</b>强度越高，损失的物体表面细节越多；强度越低，损失的物体表面细节越少。</li> <li><b>表面平滑</b>强度越高，计算时间越长；强度越低，计算时间越短。</li> </ul>

其他条件相同，仅**表面平滑**强度不同的点云对比如下：

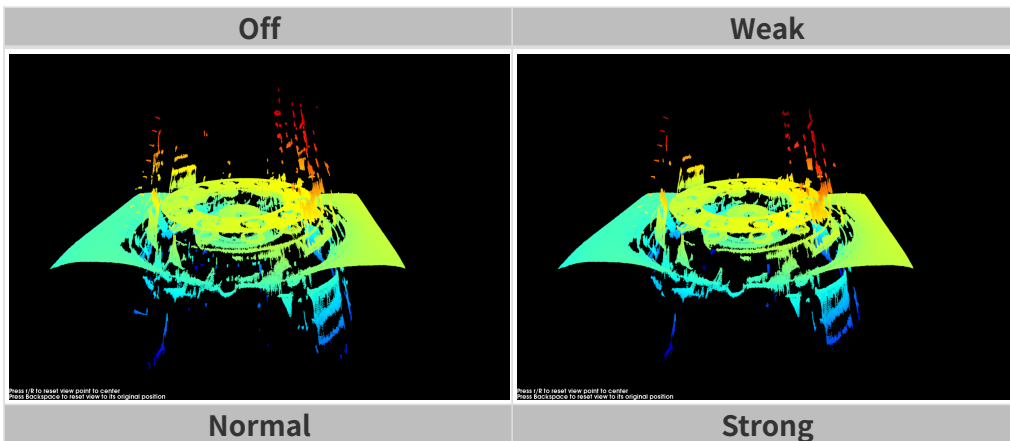




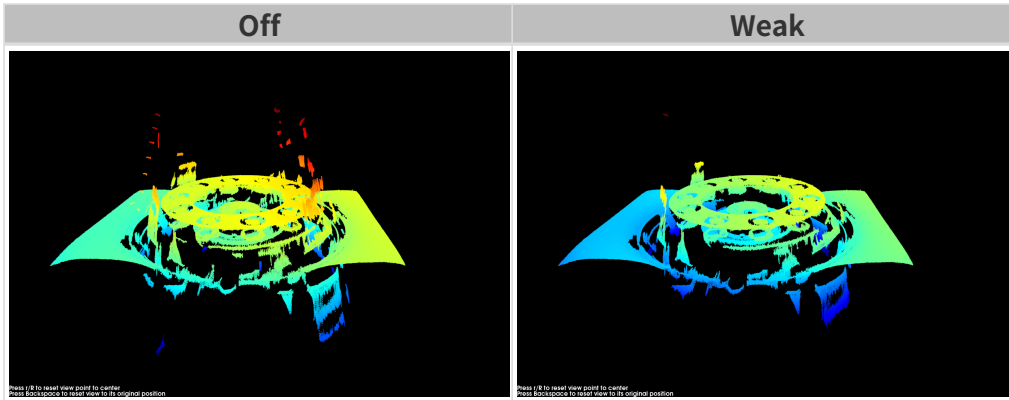
### 离群点去除

<b>参数说明</b>	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。
<b>可见级别</b>	初级、专家、大师
<b>参数取值</b>	<ul style="list-style-type: none"> <li>• Off</li> <li>• Weak</li> <li>• Normal</li> <li>• Strong</li> </ul>
<b>调节说明</b>	<ul style="list-style-type: none"> <li>• 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。</li> <li>• 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。</li> </ul>

其他条件相同，仅离群点去除强度不同的点云对比如下：



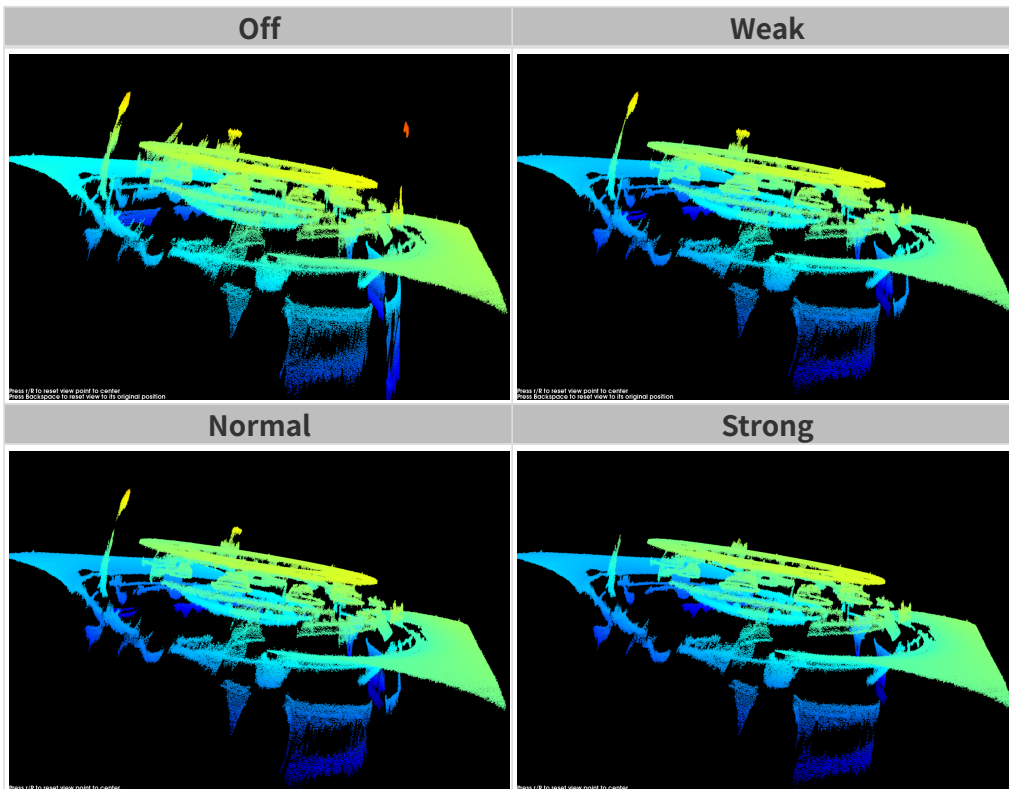




### 噪点去除

参数说明	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。
可见级别	初级、专家、大师
参数取值	<ul style="list-style-type: none"> <li>• Off</li> <li>• Weak</li> <li>• Normal</li> <li>• Strong</li> </ul>
调节说明	<ul style="list-style-type: none"> <li>• 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。</li> <li>• 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。</li> </ul>

其他条件相同，仅噪点去除强度不同的点云对比如下：





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。

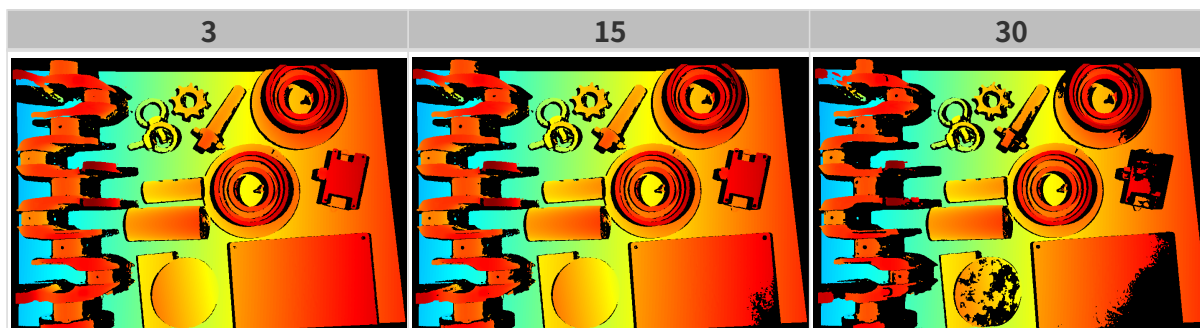
## 边缘保持

参数说明	在进行表面平滑时保持物体边缘的锐利度。
可见级别	大师
参数取值	<ul style="list-style-type: none"> <li>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。</li> <li>Normal：在保持边缘的同时达到较好的表面平滑效果。</li> <li>Smooth：不进行边缘保持。表面平滑效果最好，但物体边缘会失真。</li> </ul>
调节说明	根据对工件边缘特征的需求调节。

## 条纹对比度阈值

参数说明	用于去除点云中的噪点。如调节 <b>离群点去除</b> 与 <b>噪点去除</b> 后，仍无法获得需要的点云，再调节该参数。
可见级别	初级、专家、大师
参数取值	1~100
调节说明	<ul style="list-style-type: none"> <li>该参数值越大，去除的点越多；值越小，去除的点越少。</li> <li>调大该参数可以去除点云中的噪点，但也可能造成较暗物体对应的点丢失。</li> </ul>

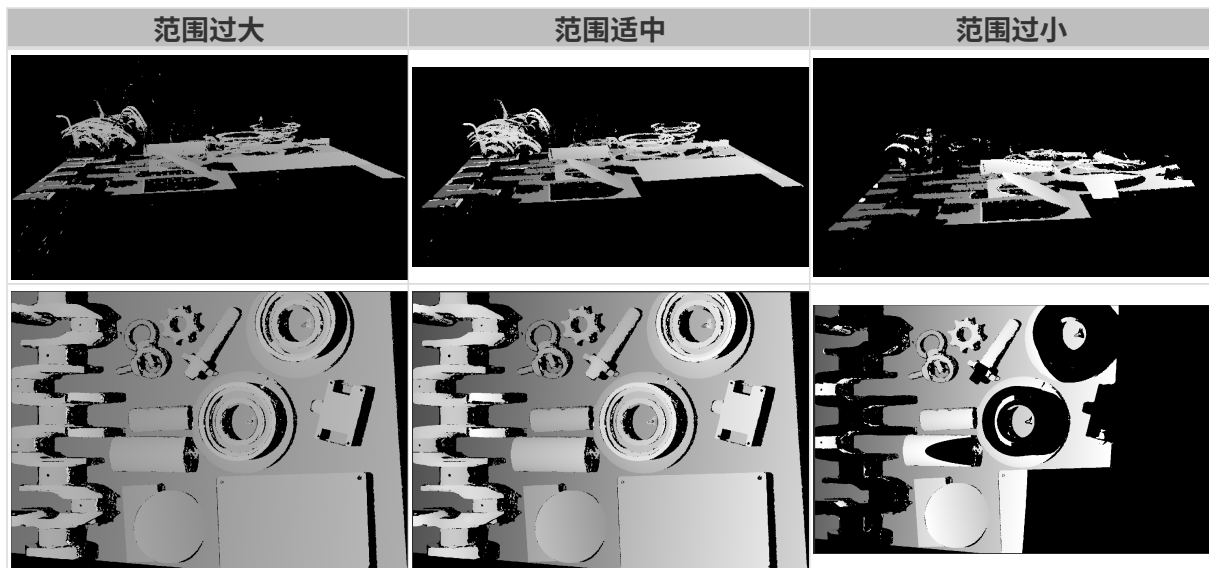
其他条件相同，仅**条纹对比度阈值**不同的点云对比如下：



## 深度范围

参数说明	设置Z向感兴趣区域。在相机的工作距离范围内设置 <b>深度范围</b> ，可滤除 <b>深度范围</b> 外的数据。
可见级别	初级、专家、大师
参数取值	<ul style="list-style-type: none"> <li>下限：1~4000mm</li> <li>上限：1~5000mm</li> </ul>
调节说明	<ul style="list-style-type: none"> <li><b>深度范围</b>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。</li> <li>如何设置详见下方的<a href="#">设置深度范围</a>。</li> </ul>

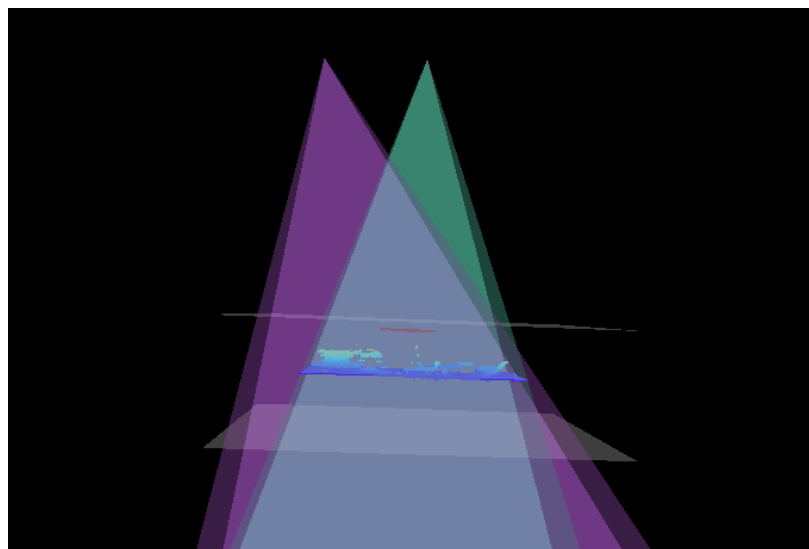
不同深度范围效果对比：



### 设置深度范围

执行以下步骤调节深度范围：

1. 双击深度范围右侧的[编辑]，打开设置深度范围界面。
2. 点击右边栏最上方的[更新点云]，获取最新的点云。
3. 调整点云的位置：调整到可看到代表深度范围上下限的两个灰色长方形即可。



4. 调节深度范围：拖动右侧滑动条上的滑块，大致调节深度范围。再输入数值，精准调节深度范围。



判断深度范围是否合适：所有必需的物体特征均位于两个灰色长方形之间，大多数噪点和离群点位于该区域之外。

5. 设置完成后，单击右下角的[保存]。



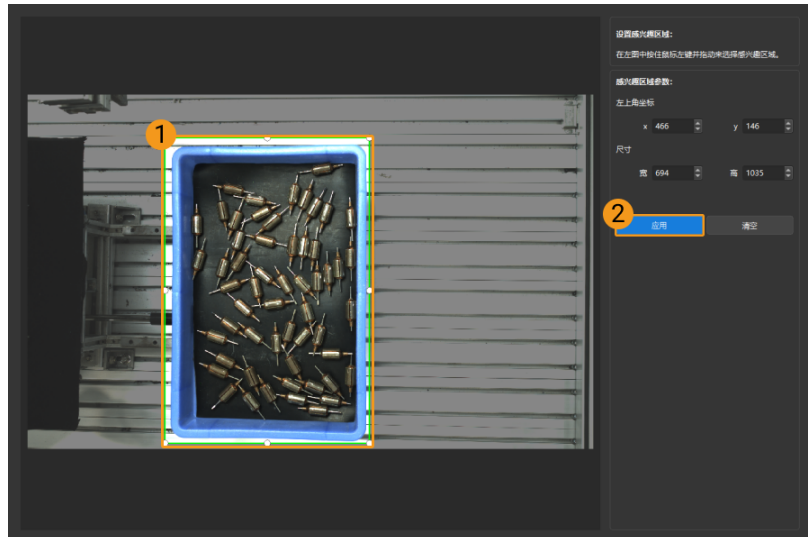
- 单击[ **推荐值** ], 可将**深度范围**调整为当前相机的推荐工作距离。
- 单击[ **重置** ], 可将**深度范围**恢复为上次保存的数值。

## 感兴趣区域

参数说明	设置深度图和点云在XOY平面上的感兴趣区域, 所选区域外的点将被移除。
可见级别	初级、专家、大师
参数取值	无
调节说明	详见下方 <a href="#">设置感兴趣区域</a> 。

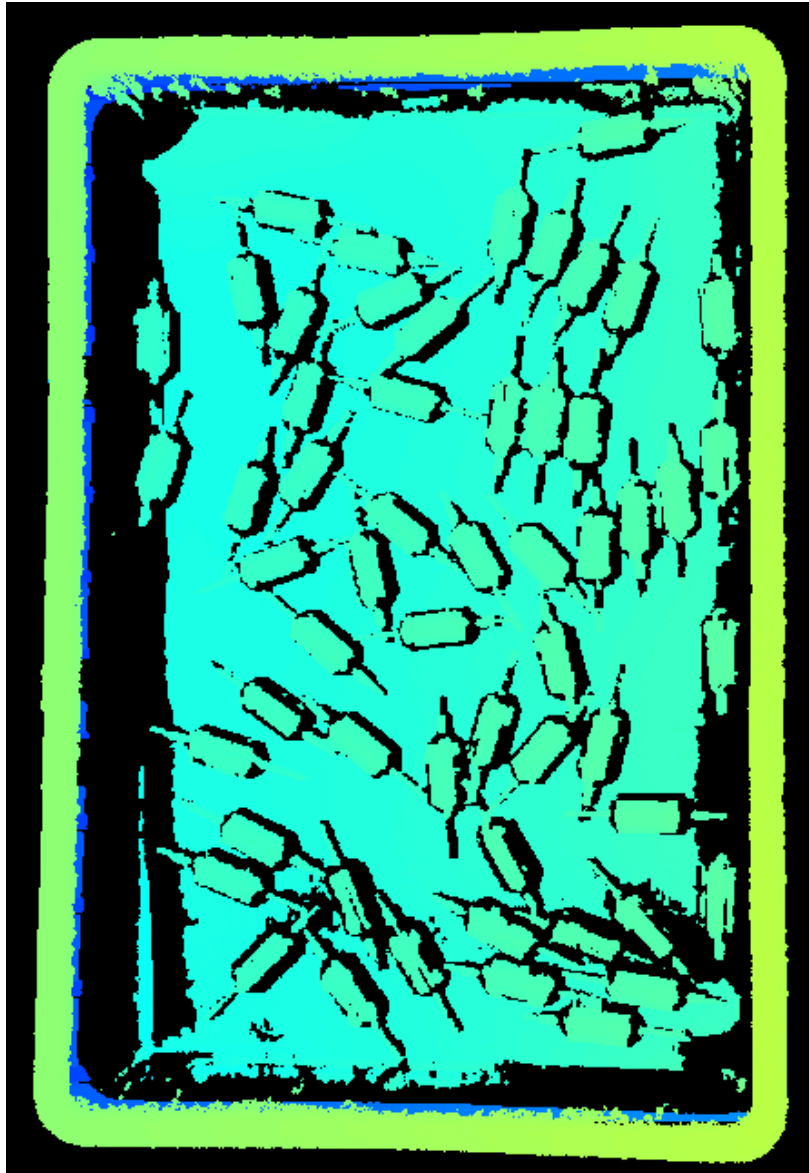
## 设置感兴趣区域

1. 双击**感兴趣区域**右侧的[ **编辑** ], 进入**设置感兴趣区域**页面。
2. 在左侧选择并调节感兴趣区域。拖拽选框可调节位置, 拖拽白色锚点可调节大小。
3. 单击[ **应用** ], 应用设置的感兴趣区域。



- 单击[ **清空** ], 可清除当前设置的感兴趣区域。
- DEEP (V4) 与LSR (V4) 系列, 在此界面显示的是2D图 (深度源)。如果图像过暗或过亮, 请调节**2D图 (深度源) 曝光模式**。

4. 重新采集图像, 查看深度图或点云, 确认所设感兴趣区域的效果。



### 5.3.6. V3相机参数

本章介绍V3相机的参数。参数按照影响的数据类型拆分为2D图参数及深度图与点云参数。

#### 2D图参数

2D图应不过亮或过暗，可看清目标物体的表面特征。

2D参数分组下的参数及3D参数分组下的**相机增益**影响2D图质量。



彩色相机采集图像时，如因现场光照条件导致图像颜色与实际差别较大，请调节**白平衡**。详细操作请参考[调节白平衡](#)。

#### 2D参数

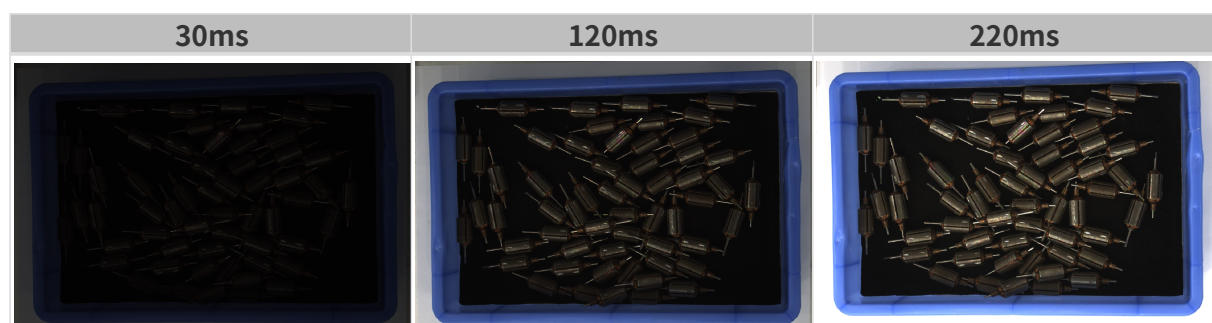
## 曝光模式

<b>参数说明</b>	设置拍摄2D图时的曝光模式。
<b>可见级别</b>	初级、专家、大师
<b>参数取值</b>	<ul style="list-style-type: none"> <li>固定曝光：设置固定曝光时间，常用于稳定的光照条件。</li> <li>自动曝光：自动调整曝光时间，常用于多变的光照条件。</li> <li>HDR：设置多个曝光时间并融合所有图像，常用于颜色或材质多样的物体。</li> <li>Flash：使用投影机补光，常用于较暗的环境。</li> </ul>
<b>调节说明</b>	选择不同选项后， <b>2D参数</b> 分组中将显示不同的参数供调节： <ul style="list-style-type: none"> <li>固定曝光：显示<b>曝光时间</b>。</li> <li>自动曝光：显示<b>灰度值</b>与<b>自动曝光感兴趣区域</b>。</li> <li>HDR：显示<b>色调映射</b>与<b>曝光时间序列</b>。</li> <li>Flash：无需调节其他参数。采集图像时将自动补光。</li> </ul>

### 固定曝光：曝光时间

<b>参数说明</b>	影响图像亮度。 <ul style="list-style-type: none"> <li>曝光时间越长，图像亮度越高。</li> <li>曝光时间越短，图像亮度越低。</li> </ul>
<b>可见级别</b>	初级、专家、大师
<b>参数取值</b>	0.1~999ms
<b>调节说明</b>	根据2D图的质量调节。2D图应不过亮或过暗，可看清目标物体的表面特征。 <ul style="list-style-type: none"> <li>较暗的环境下通常使用较长曝光时间。</li> <li>较亮的环境下通常使用较短曝光时间。</li> </ul>

其他条件相同，仅**曝光时间**不同的2D图对比如下：

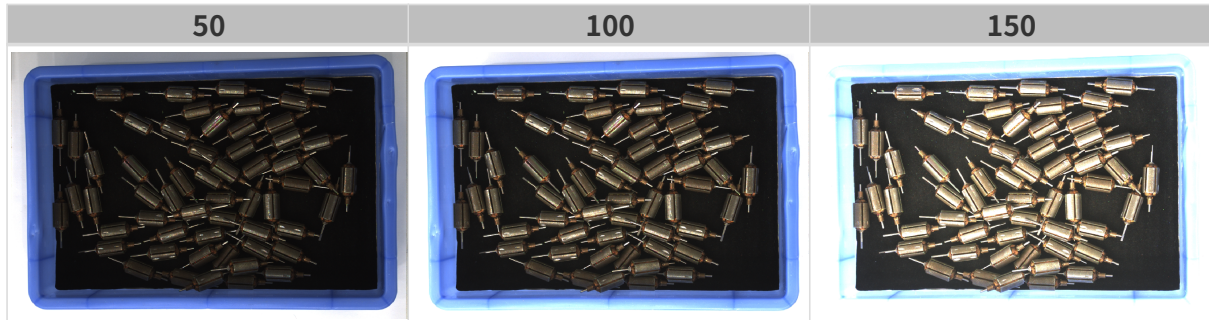


### 自动曝光：灰度值

<b>参数说明</b>	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。
-------------	-------------------------------

可见级别	初级、专家、大师
参数取值	0~255
调节说明	无

其他条件相同，仅灰度值不同的2D图对比如下：

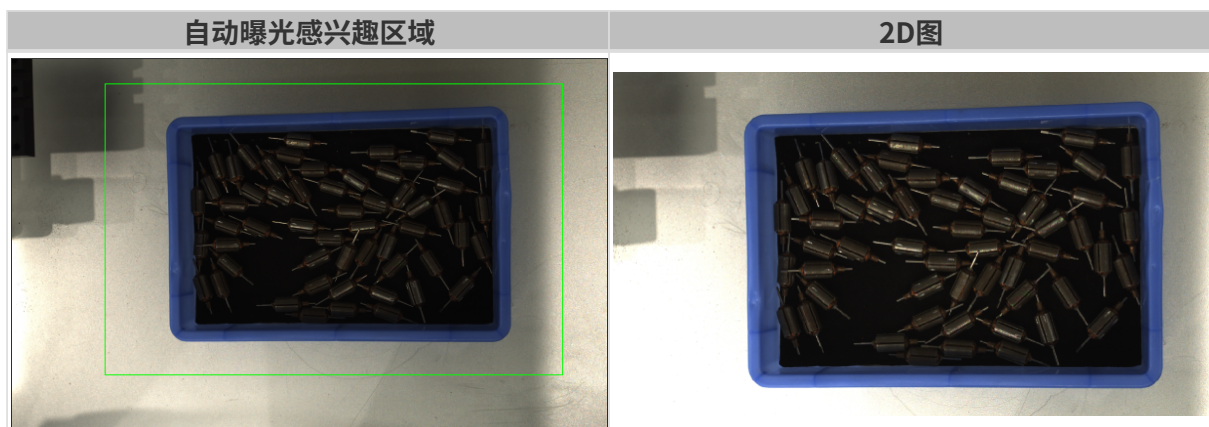


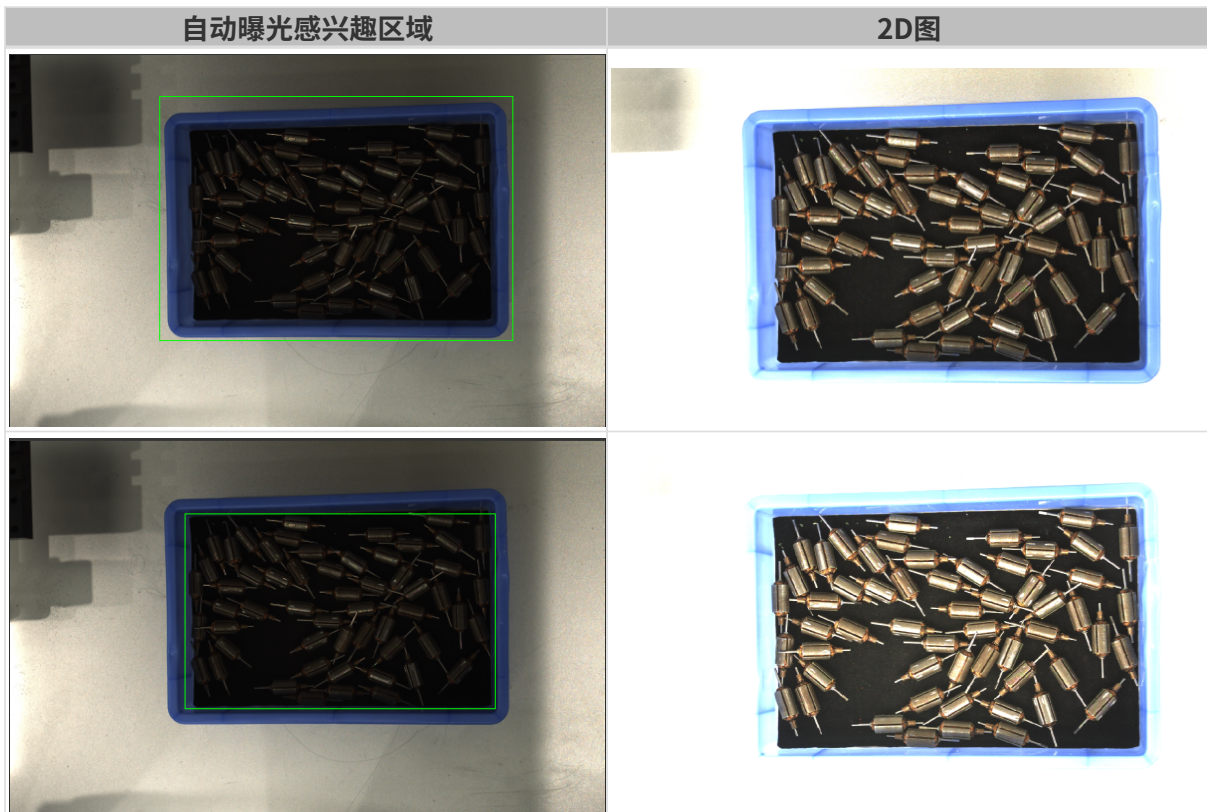
黑白图像的灰度值相当于图像亮度；彩色图像的灰度值相当于每个颜色通道的亮度。

### 自动曝光：自动曝光感兴趣区域

参数说明	<ul style="list-style-type: none"> <li>相机参照该区域内的光照、物体颜色等自动调整曝光时间。</li> <li>未设置感兴趣区域时，相机根据整个视野自动调整曝光时间。</li> </ul>
可见级别	初级、专家、大师
参数取值	无
调节说明	详见下方 <a href="#">设置自动曝光感兴趣区域</a> 。

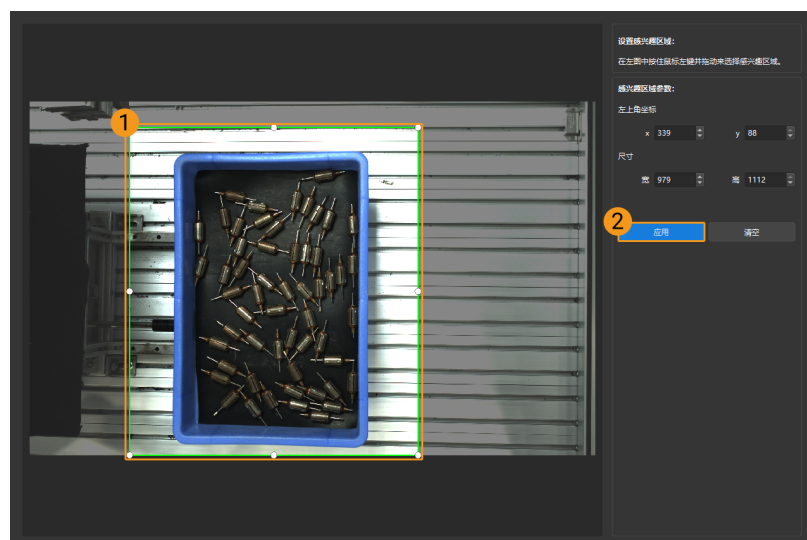
其他条件相同，仅自动曝光感兴趣区域不同的2D图对比如下：





### 设置自动曝光感兴趣区域

1. 双击自动曝光感兴趣区域右侧的[编辑]，进入设置感兴趣区域页面。
2. 在左侧选择并调节感兴趣区域。拖拽选框可调节位置，拖拽白色锚点可调节大小。
3. 单击[应用]，应用设置的自动曝光感兴趣区域。



单击[清空]，可清除当前设置的自动曝光感兴趣区域。

4. 重新采集图像并查看2D图，确认自动曝光的效果。




## HDR：色调映射

参数说明	可使图像看起来更自然。如2D图和真实物体相差较大，可勾选该参数。
可见级别	初级、专家、大师
参数取值	<ul style="list-style-type: none"> <li>• False</li> <li>• True</li> </ul>
调节说明	勾选即可开启色调映射。

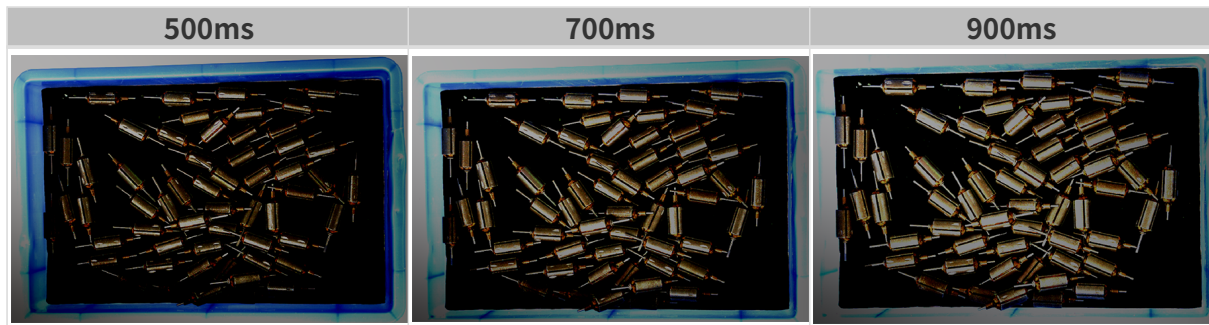
色调映射开启前后的2D图对比如下：



## HDR：曝光时间序列

参数说明	设置多个曝光时间，并将所拍摄的图像融合为一张保留更多明暗细节的2D图。
可见级别	初级、专家、大师
参数取值	无
调节说明	<ol style="list-style-type: none"> <li>1. 双击<b>曝光时间序列</b>右侧的[编辑]，进入曝光时间序列的编辑界面。</li> <li>2. 单击[+]，新增并设置曝光时间。</li> <li>3. 如需删除某个曝光时间，选中该曝光时间并单击[-]。</li> <li>4. 编辑完成后，单击[应用]使曝光时间序列生效。</li> </ol> <div style="display: flex; align-items: center; margin-top: 10px;"> <div style="margin-right: 10px;">  </div> <ul style="list-style-type: none"> <li>◦ [取消]：不保存当前设置，直接退出该页面。</li> <li>◦ [重置]：清空所有的曝光时间。</li> </ul> </div>

设置单个曝光时间时，不同曝光时间的2D图对比如下：



使用以上曝光时间组合成不同的曝光时间序列时，不同序列的2D图对比如下：



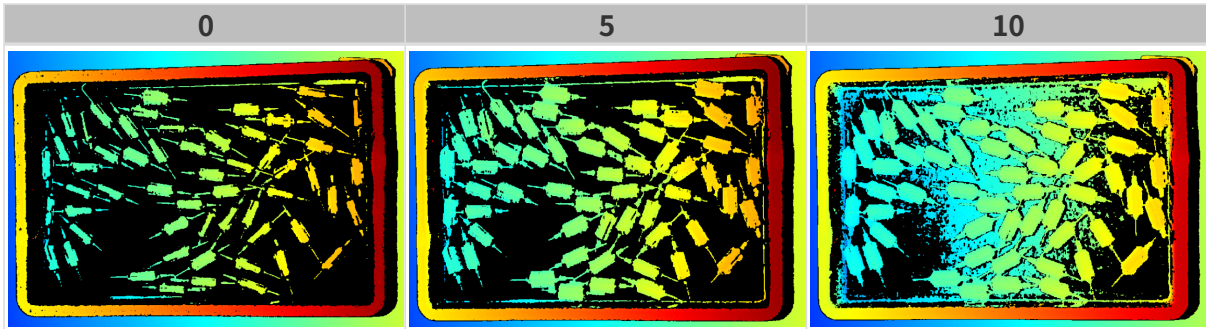
### 3D参数

#### 相机增益

参数说明	用于增加图像亮度，但可能会引入噪点。 <div style="display: flex; align-items: center; margin-top: 5px;"> <span style="font-size: 1.2em; margin-right: 5px;">i</span> <span>该参数通过改变用于计算深度数据的图像的亮度影响深度图和点云质量。</span> </div>
可见级别	专家、大师
参数取值	0~16dB
调节说明	当设置曝光时间无法达到期望亮度时，可以使用该参数。

其他条件相同，仅相机增益不同的2D图和深度图对比如下：





## 深度图与点云参数

深度图与点云中，所需的数据应完整。



根据实际需求判断所需数据的范围。例如，需通过碗沿抓取碗口向上摆放的金属碗时，一般只需确保碗沿部分的数据完整。

以下分组中的参数影响深度图与点云质量。

参数分组	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

该分组下的参数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。

使用**曝光助手**可获得推荐的曝光参数组合。双击**3D参数**右侧的[自动设置]，即可打开**曝光助手**。

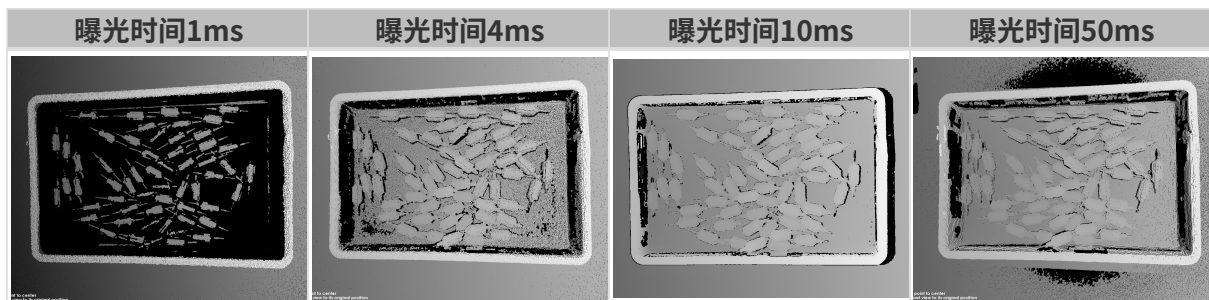
## 曝光次数

<b>参数说明</b>	设置 <b>曝光时间</b> 的个数。
<b>可见级别</b>	初级、专家、大师
<b>参数取值</b>	1~3
<b>调节说明</b>	<ul style="list-style-type: none"> <li>曝光次数大于1时，需设置多个<b>曝光时间</b>。</li> <li>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，但也将延长处理时间。</li> <li>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。</li> </ul>

## 曝光时间

参数说明	设置采集深度信息时的曝光时间，需设置的 <b>曝光时间</b> 个数取决于 <b>曝光次数</b> 的值。
可见级别	初级、专家、大师
参数取值	<ul style="list-style-type: none"> <li>取值范围：0.1~99ms</li> </ul>
调节说明	<ul style="list-style-type: none"> <li>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。</li> <li>曝光时间过长或过短都会导致信息缺失。</li> <li>激光相机的曝光时间须为4的倍数，输入值不为4的倍数时将被自动调整。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。</li> </ul>

不同**曝光时间**对比图如下：

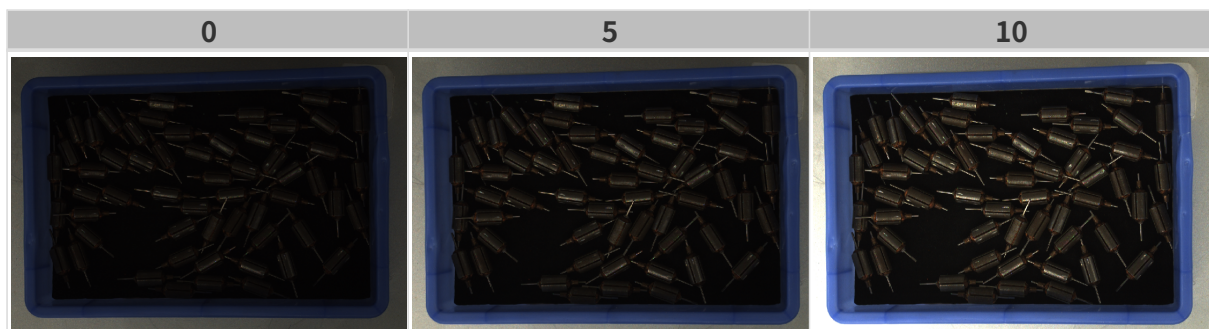


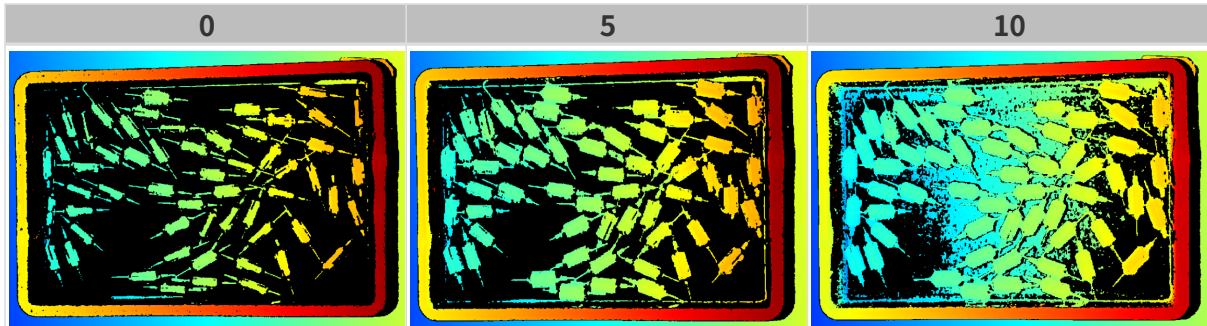
黑色部分缺少对应的物体点云。

## 相机增益

参数说明	用于增加图像亮度，但可能会引入噪点。
	 该参数通过改变 <b>用于计算深度数据的图像</b> 的亮度影响深度图和点云质量。
可见级别	专家、大师
参数取值	0~16dB
调节说明	当设置 <b>曝光时间</b> 无法达到期望亮度时，可以使用该参数。

其他条件相同，仅**相机增益**不同的2D图和深度图对比如下：





## 投影

适用相机型号：Nano (V3)、Pro (V3) 和Log (V3) 系列。

### 投影光亮度

参数说明	投影仪投射的结构光亮度。
可见级别	专家、大师
参数取值	<ul style="list-style-type: none"> <li>High: 高亮度, 适用于深色物体。</li> <li>Normal: 正常亮度, 适用于普通物体。</li> <li>Low: 低亮度, 适用于反光物体。</li> </ul>
调节说明	根据物体类型调节。

## 编码模式

适用相机型号：Nano (V3)、Pro XS (V3)。

参数说明	选择投射的结构光的样式。
可见级别	专家、大师
参数取值	<ul style="list-style-type: none"> <li>Fast: 适用于不透明的物体, 采集速度快, 但深度数据质量稍差。</li> <li>Accurate: 适用于不透明的物体, 深度数据质量高, 但采集速度较慢。</li> </ul>
调节说明	根据目标物体类型以及对数据质量和采集速度的实际需求调节。

目标物体为非透明物体时, 其他条件相同, **编码模式**分别选择**Accurate**和**Fast**时获取的点云如下:

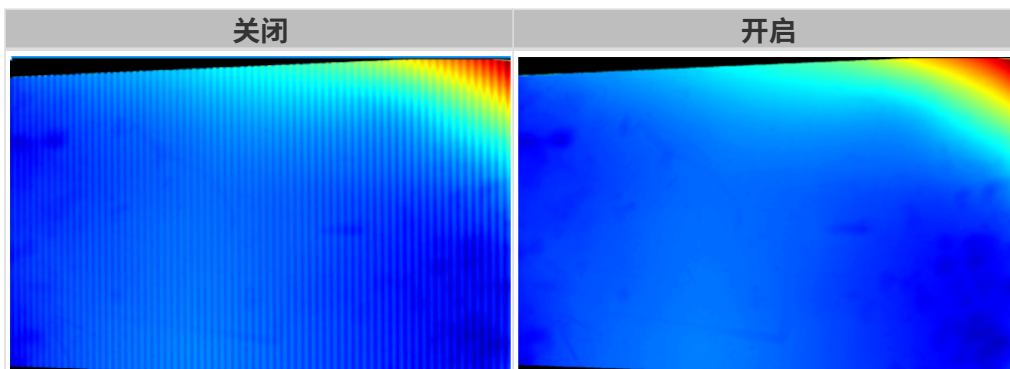


### 抗频闪模式

适用相机型号：Nano (V3)。

参数说明	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。
可见级别	专家、大师
参数取值	<ul style="list-style-type: none"> <li>• Off</li> <li>• AC50Hz</li> <li>• AC60Hz</li> </ul>
调节说明	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。

抗频闪模式开启前后的深度图对比如下：



### 激光

适用相机型号：Laser (V3) 系列。

### 编码模式

参数说明	影响采集速度与深度数据质量。
可见级别	专家、大师

参数取值	<ul style="list-style-type: none"> <li>Fast: 采集速度快, 但深度图和点云质量较差。</li> <li>Accurate: 采集速度慢, 但深度图和点云质量较好。</li> </ul>
调节说明	根据对数据质量和采集速度的实际需求调节。

其他条件相同, 仅**编码模式**不同的点云对比如下:



### 激光强度

参数说明	设置激光器的投影强度, 影响结构光的亮度。
可见级别	专家、大师
参数取值	50~100%
调节说明	一般情况下使用默认值即可。 <ul style="list-style-type: none"> <li>数值越大, 结构光亮度越高; 数值越小, 结构光亮度越低。</li> <li>深色物体建议使用高强度, 反光物体建议使用低强度。</li> </ul>

### 点云后处理

调节点云后处理分组下的参数, 可提升点云质量。

#### 调参原则

调节点云后处理参数时, 遵循以下调参原则可减少相机采集时间, 优化节拍。

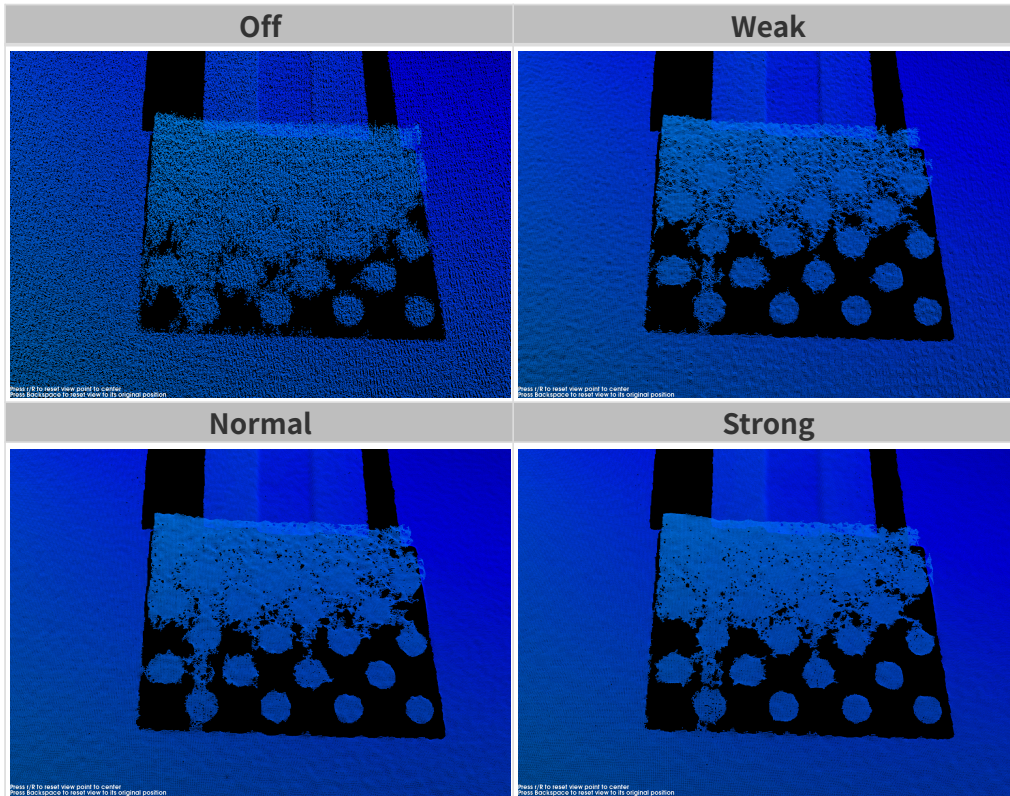
1. 优先调节**离群点去除**。该参数各强度计算时间基本相同, 即使使用较高强度也不会增加太多计算时间。
2. 建议使用低强度**表面平滑与噪点去除**。这两个参数强度越高, 计算时间越长。

### 表面平滑

参数说明	可减少点云的深度波动, 使点云更接近真实的物体表面, 但会损失部分物体表面细节。
可见级别	初级、专家、大师

参数取值	<ul style="list-style-type: none"> <li>• Off</li> <li>• Weak</li> <li>• Normal</li> <li>• Strong</li> </ul>
调节说明	<ul style="list-style-type: none"> <li>• 表面平滑强度越高，损失的物体表面细节越多；强度越低，损失的物体表面细节越少。</li> <li>• 表面平滑强度越高，计算时间越长；强度越低，计算时间越短。</li> </ul>

其他条件相同，仅表面平滑强度不同的点云对比如下：

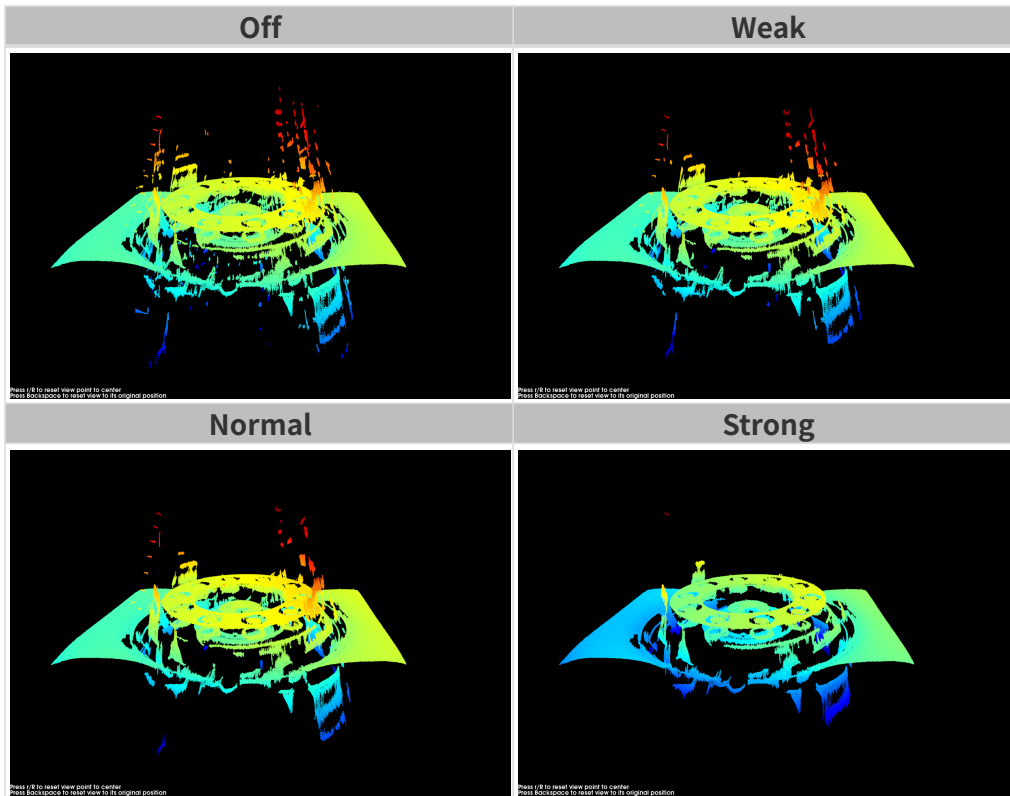


### 离群点去除

参数说明	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。
可见级别	初级、专家、大师
参数取值	<ul style="list-style-type: none"> <li>• Off</li> <li>• Weak</li> <li>• Normal</li> <li>• Strong</li> </ul>
调节说明	<ul style="list-style-type: none"> <li>• 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。</li> <li>• 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。</li> </ul>



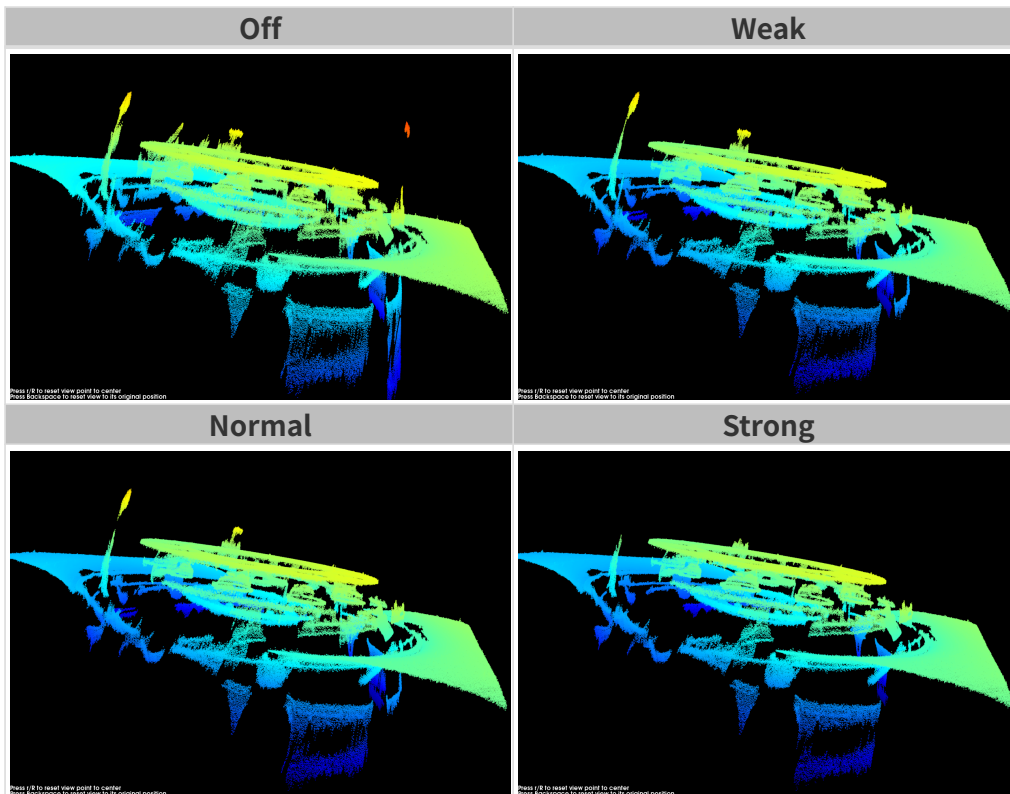
其他条件相同，仅离群点去除强度不同的点云对比如下：



### 噪点去除

参数说明	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。
可见级别	初级、专家、大师
参数取值	<ul style="list-style-type: none"> <li>• Off</li> <li>• Weak</li> <li>• Normal</li> <li>• Strong</li> </ul>
调节说明	<ul style="list-style-type: none"> <li>• 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。</li> <li>• 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。</li> </ul>

其他条件相同，仅噪点去除强度不同的点云对比如下：



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。

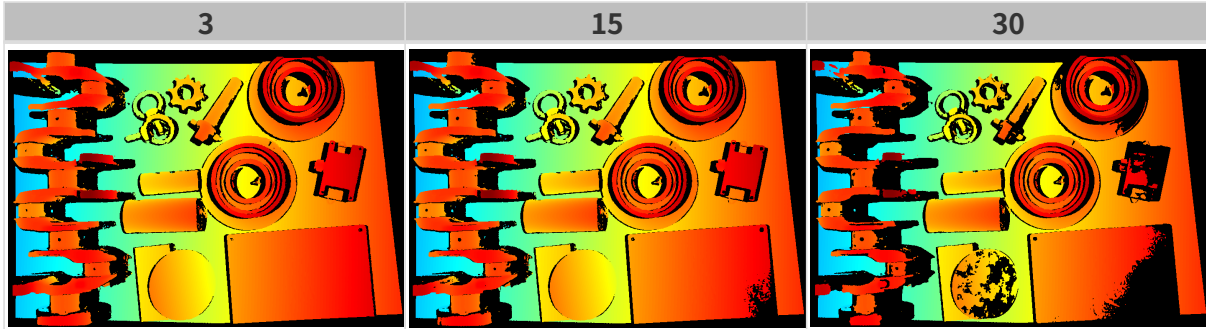
## 边缘保持

参数说明	在进行表面平滑时保持物体边缘的锐利度。
可见级别	大师
参数取值	<ul style="list-style-type: none"> <li>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。</li> <li>Normal: 在保持边缘的同时达到较好的表面平滑效果。</li> <li>Smooth: 不进行边缘保持。表面平滑效果最好，但物体边缘会失真。</li> </ul>
调节说明	根据对工件边缘特征的需求调节。

## 条纹对比度阈值

参数说明	用于去除点云中的噪点。如调节 <b>离群点去除</b> 与 <b>噪点去除</b> 后，仍无法获得需要的点云，再调节该参数。
可见级别	初级、专家、大师
参数取值	1~100
调节说明	<ul style="list-style-type: none"> <li>该参数值越大，去除的点越多；值越小，去除的点越少。</li> <li>调大该参数可以去除点云中的噪点，但也可能造成较暗物体对应的点丢失。</li> </ul>

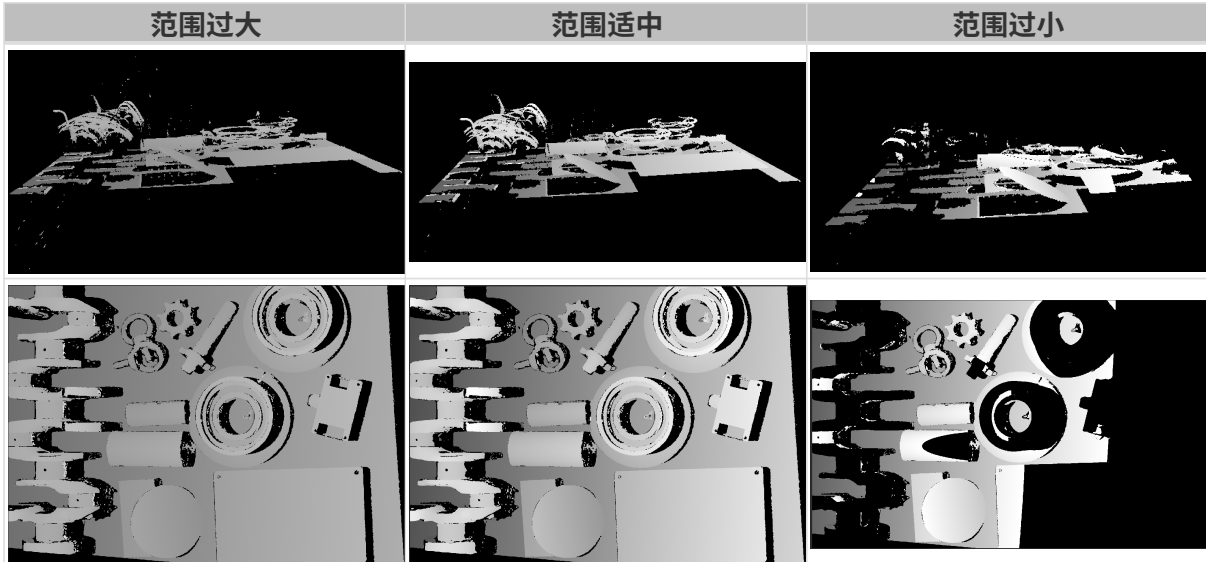
其他条件相同，仅**条纹对比度阈值**不同的点云对比如下：



### 深度范围

参数说明	设置Z向感兴趣区域。在相机的工作距离范围内设置 <b>深度范围</b> ，可滤除 <b>深度范围</b> 外的数据。
可见级别	初级、专家、大师
参数取值	<ul style="list-style-type: none"> <li>• 下限：1~4000mm</li> <li>• 上限：1~5000mm</li> </ul>
调节说明	<ul style="list-style-type: none"> <li>• <b>深度范围</b>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。</li> <li>• 如何设置详见下方的<a href="#">设置深度范围</a>。</li> </ul>

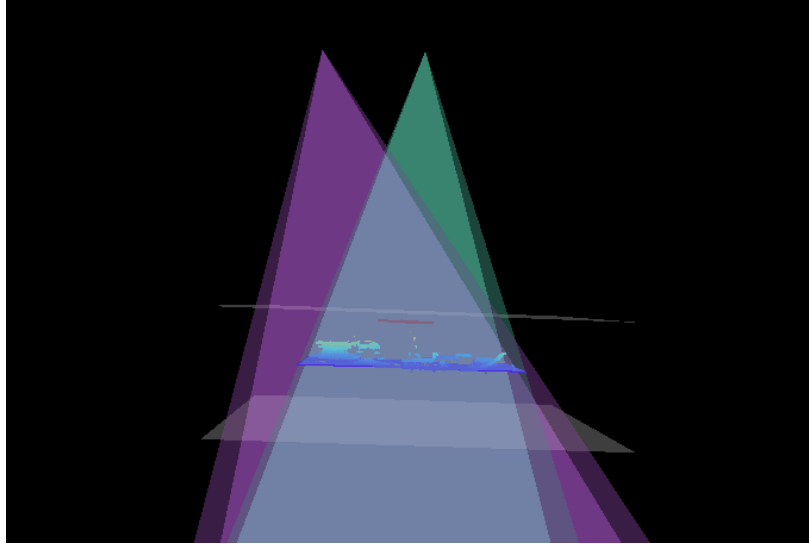
不同**深度范围**效果对比：



### 设置深度范围

执行以下步骤调节**深度范围**：

1. 双击**深度范围**右侧的[编辑]，打开设定**深度范围**界面。
2. 点击右边栏最上方的[更新点云]，获取最新的点云。
3. **调整点云的位置**：调整到可看到代表**深度范围**上下限的两个灰色长方形即可。



4. 调节深度范围：拖动右侧滑动条上的滑块，大致调节**深度范围**。再输入数值，精准调节**深度范围**。



判断深度范围是否合适：所有必需的物体特征均位于两个灰色长方形之间，大多数噪点和离群点位于该区域之外。

5. 设置完成后，单击右下角的[ **保存** ]。



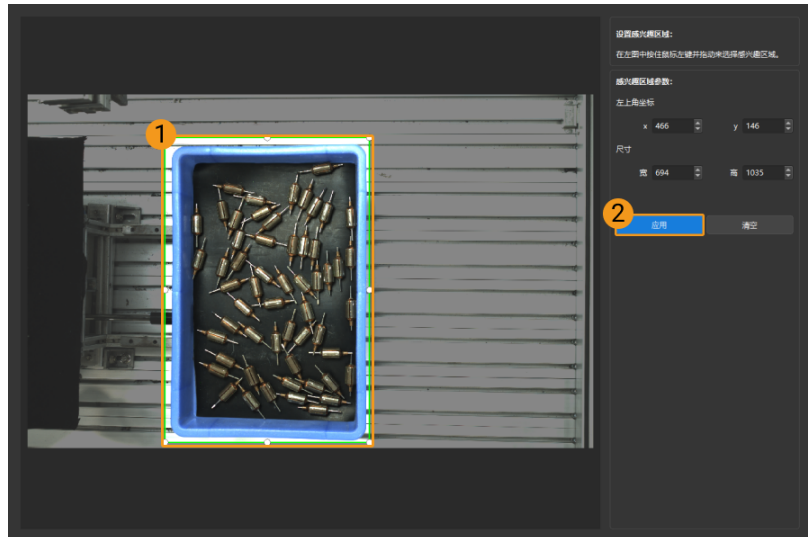
- 单击[ **推荐值** ]，可将**深度范围**调整为当前相机的推荐工作距离。
- 单击[ **重置** ]，可将**深度范围**恢复为上次保存的数值。

## 感兴趣区域

参数说明	设置深度图和点云在XOY平面上的感兴趣区域，所选区域外的点将被移除。
可见级别	初级、专家、大师
参数取值	无
调节说明	详见下方 <a href="#">设置感兴趣区域</a> 。

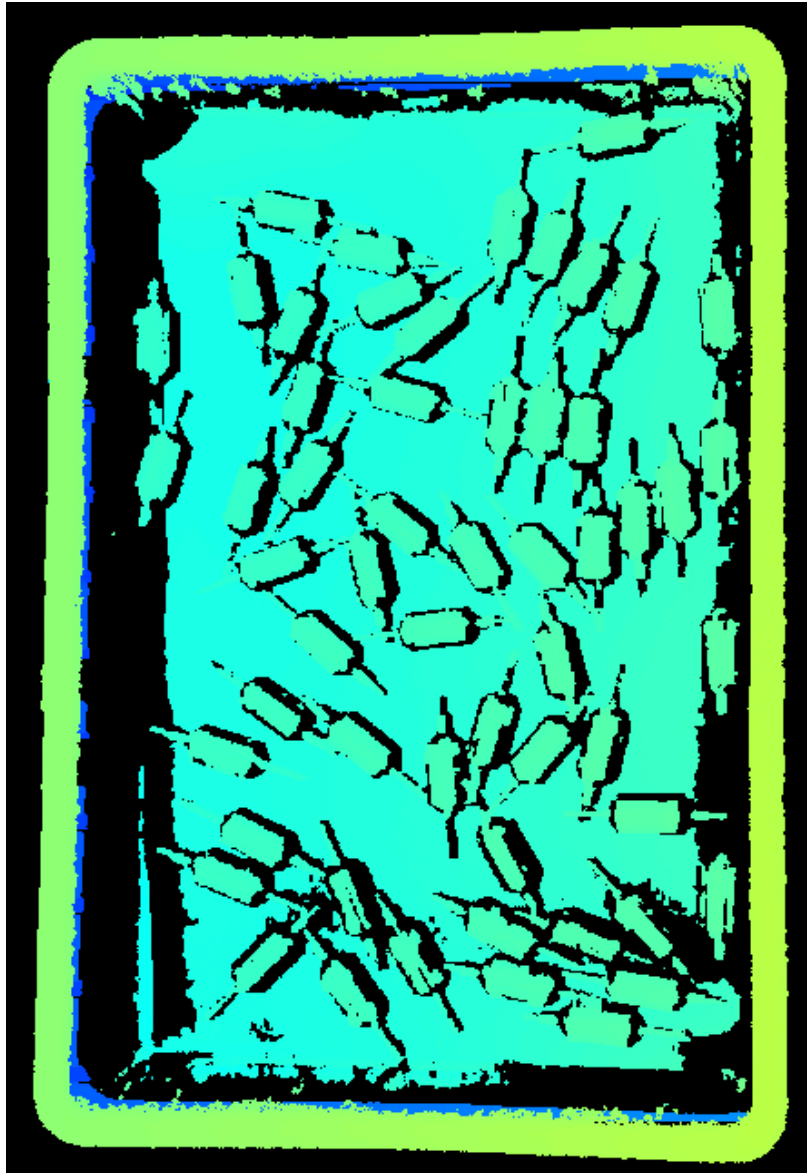
## 设置感兴趣区域

1. 双击**感兴趣区域**右侧的[ **编辑** ]，进入**设置感兴趣区域**页面。
2. 在左侧选择并调节感兴趣区域。拖拽选框可调节位置，拖拽白色锚点可调节大小。
3. 单击[ **应用** ]，应用设置的感兴趣区域。



- 单击[ 清空 ], 可清除当前设置的感兴趣区域。
- DEEP (V4) 与LSR (V4) 系列, 在此界面显示的是2D图 (深度源)。如果图像过暗或过亮, 请调节**2D图 (深度源) 曝光模式**。

4. 重新采集图像, 查看深度图或点云, 确认所设感兴趣区域的效果。



## 5.4. 工具使用指南

Mech-Eye Viewer提供一系列工具辅助使用相机。本章介绍各工具的用途和功能，并提供详细的操作指南。

工具名称	功能描述
<a href="#">内参工具</a>	检查相机的内参是否准确、矫正内参、恢复出厂内参。
<a href="#">3D曝光助手</a>	获取推荐的 <b>3D参数</b> 中曝光参数的组合。
<a href="#">深度图分析器</a>	检验深度数据的质量。
<a href="#">查看并配置2D相机</a>	查看和设置2D相机的参数。
<a href="#">视野计算器</a>	根据工作距离计算相机的视野。
<a href="#">自定义坐标系</a>	自定义坐标系，用于显示深度图和点云。
<a href="#">相机管理器</a>	查看相机的投光单元固件版本、时间、温度等信息及更改图像分辨率。

工具名称	功能描述
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**选项，可显示包含常用工具的工具栏。

单击工具栏中的[ **更多工具** ]，可选择需在工具栏中显示的工具。单击[ **OK** ]保存修改。

### 5.4.1. 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。只有内参准确，才能保证计算得出的各点坐标准确。

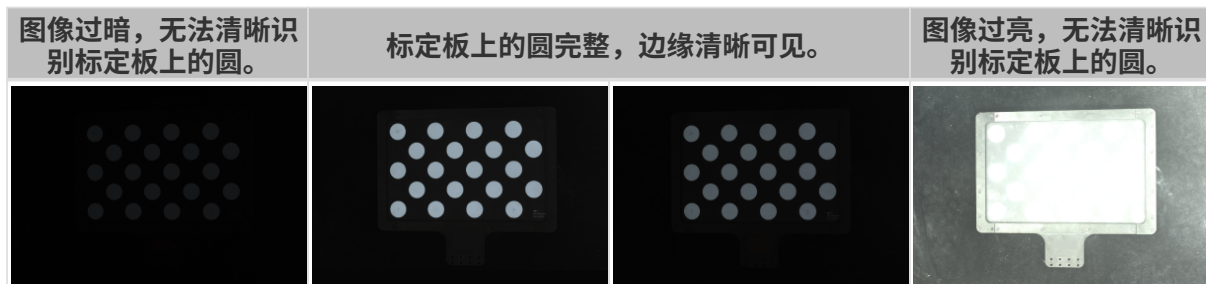
#### 准备工作

检查内参前，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图相关参数**。

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](#)。

## 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。



如出现错误提示，请根据下一节的内容排查问题。

## 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。

### 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图相关参数](#)；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。具体操作说明详见[画辅助圆](#)。
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**，请查看下一步。
5. 使用**编辑特征检测参数**功能辅助检测。具体操作说明详见[编辑特征检测参数](#)。
  6. 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。

### 辅助功能

**内参工具**提供以下功能辅助解决常见问题。

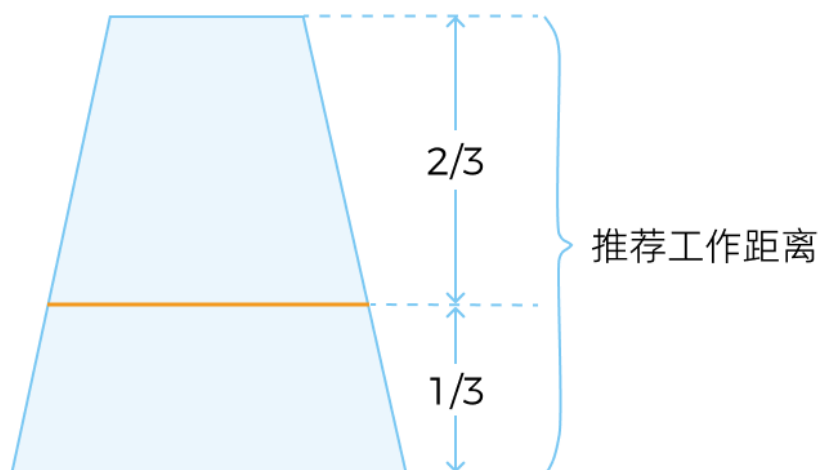
#### 矫正内参

该功能可辅助解决内参误差较大的问题。

-  UHP系列不可使用**矫正内参**功能。如内参误差较大，请联系技术支持。

使用**矫正内参**功能时，应按以下要求放置标定板：

- 将标定板放置在下图中橙线所在高度：



- 至少采集标定板位于视野中心和四个角的5组数据。



请执行以下步骤使用**矫正内参**功能：

- 按上述要求放置标定板。
- 勾选**矫正内参**后，单击[采集数据]。
- 移动标定板，再次单击[采集数据]。
- 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。

### 画辅助圆

该功能可辅助解决**未检测到标定板上的圆**的问题。

请执行以下步骤使用**画辅助圆**功能：

- 单击**3. 检查内参**中的[画辅助圆]。
- 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**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。



- 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。

### 编辑特征检测参数（高级）

该功能可辅助解决**未检测到标定板上的圆**的问题。

请执行以下步骤使用**编辑特征检测参数**功能：

- 单击**3. 检查内参**中**编辑特征检测参数（高级）**。
- 请根据实际情况调整显示的特征检测参数。
- 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。
- 在**3. 检查内参**中单击[检查内参]，重新检查内参。

### 5.4.2. 3D曝光助手

此工具用于获取**3D参数**分组下**曝光时间**参数的推荐值。



当前仅能计算单次曝光时间。

请执行以下步骤使用**3D曝光助手**：

1. 在**参数**标签页中确认**深度范围**和**感兴趣区域**参数的设置：应保证目标物体在深度范围和感兴趣区域内。
2. 双击**参数**标签页中**3D参数**右侧的[ **自动设置** ]，或单击菜单栏的**工具**菜单中的**3D曝光助手**，打开**3D曝光助手**。
3. 调节感兴趣区域，**3D曝光助手**将根据该感兴趣区域自动计算曝光时间。请框选目标物体所在范围，尽量避免选中无关物体。完成后，单击[ **下一步** ]。



拖拽选框可调节其位置，拖拽选框上的白色锚点可调节选框大小。

4. 单击[ **计算曝光时间** ]，计算单次曝光的时间。结果将显示在**计算结果**区中。
5. 在左侧图像区中确认深度图和点云的质量：
  - 如质量符合要求，单击[ **应用** ]，将计算结果直接应用至**3D参数**分组。

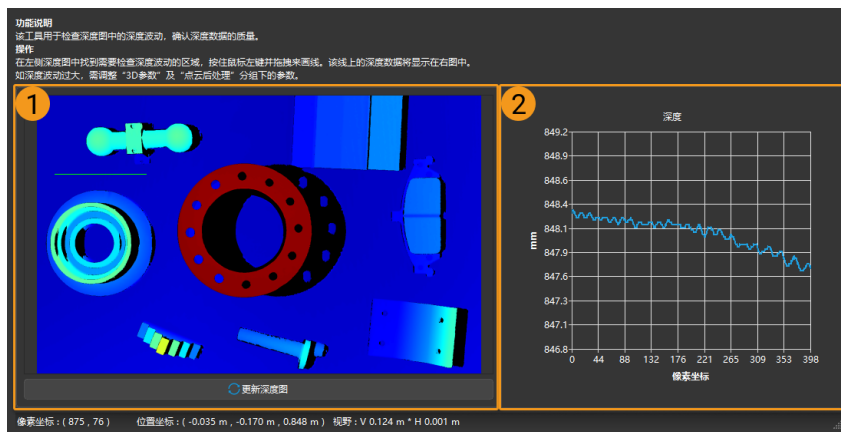


如**曝光次数**参数已设置为大于1的值，应用计算结果后，**曝光次数**参数将被设置为1。

- 如质量不符合要求，单击[ **上一步** ]，并重复步骤2至4。

### 5.4.3. 深度图分析器

此工具用于检查深度图中的深度波动，帮助确认深度数据的质量。



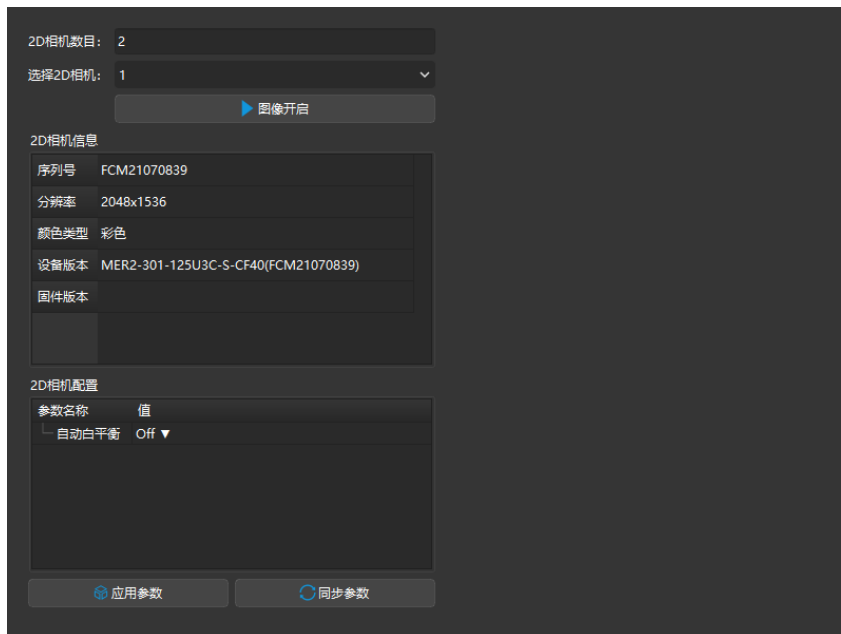
请执行以下步骤检查深度图中的深度波动：

1. 单击**工具** > **深度图分析器**，打开**深度图分析器**。
2. 在左侧的深度图中（上图①），找到需要检查深度波动的区域。
3. 按住鼠标左键并拖拽以画线。该线上的深度数据将显示在右图中（上图②）。
4. 查看右图中深度值的上下波动情况：
  - 如果波动相对于场景实际深度较小，则深度数据质量良好；
  - 如果波动相对于场景实际深度较大，则深度数据质量较差。请调节**3D参数**和**点云后处理**分

组下的参数，提升深度数据的质量。

#### 5.4.4. 查看并配置2D相机

该工具用于查看和配置Mech-Eye工业级3D相机中的2D相机。



#### 查看2D相机信息

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

在**选择2D相机**菜单中选择要查看的2D相机后，可查看该2D相机的上述信息。

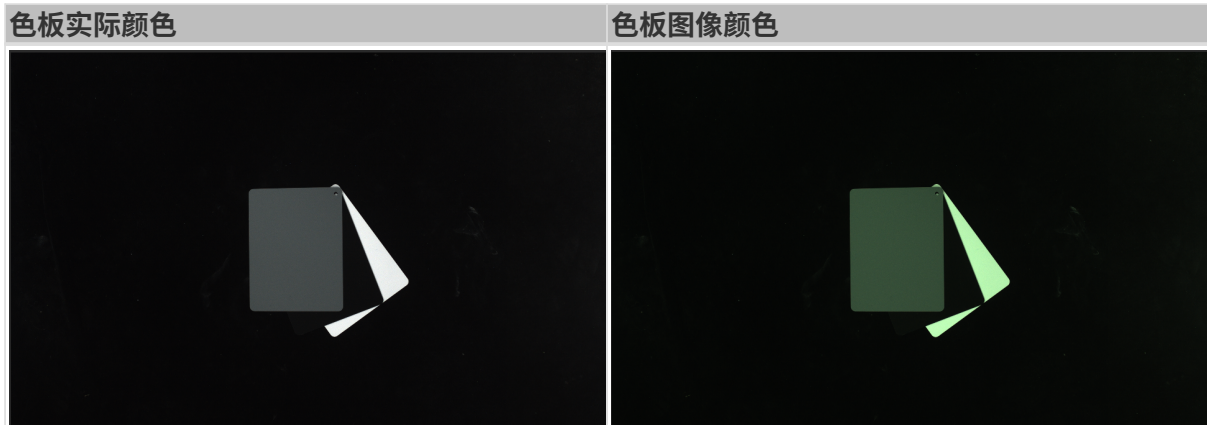
#### 调节白平衡



仅用于彩色2D相机。查看**2D相机信息**中的**颜色类型**可确认是否为彩色2D相机。

如果获取的2D图颜色与实际物体差别较大，需调节白平衡。否则，2D图颜色失真将导致纹理点云的颜色失真并影响后续处理。在深度学习中，如果颜色失真的2D图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。

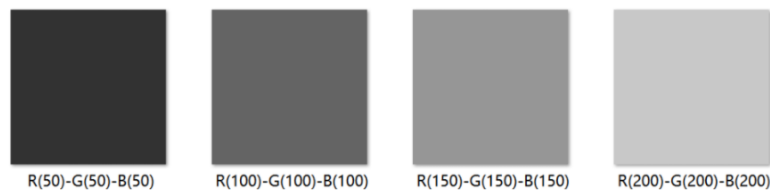
#### 对比示例



## 准备工作

调节白平衡前，需进行如下工作：

1. 确认2D图不欠曝或过曝。如2D图欠曝或过曝，请调节2D图参数。
2. 准备中性灰色板，颜色请参考下图。然后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. 查看灰色色板的图像颜色。
  - 如图像颜色存在明显偏差，请移动灰色色板，直至色板颜色无明显偏差。
  - 如图像颜色无明显偏差，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。

### 5.4.5. 视野计算器

该工具用于基于对相机视野的需求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。

计算结果仅供参考，请以实际情况为准。

## 计算相机视野

请执行以下步骤计算相机视野：

### 1. 在选择相机型号中选择相机型号。



如所用型号不在下拉菜单中，请单击菜单最下方的**其他型号**查看更多型号。

### 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长度与视野宽度。如计算结果不符合实际视野需求，请执行下一步。



- 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，直至计算结果符合需求。根据该工作距离可确定相机的安装高度。

## 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	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

### 5.4.6. 自定义坐标系

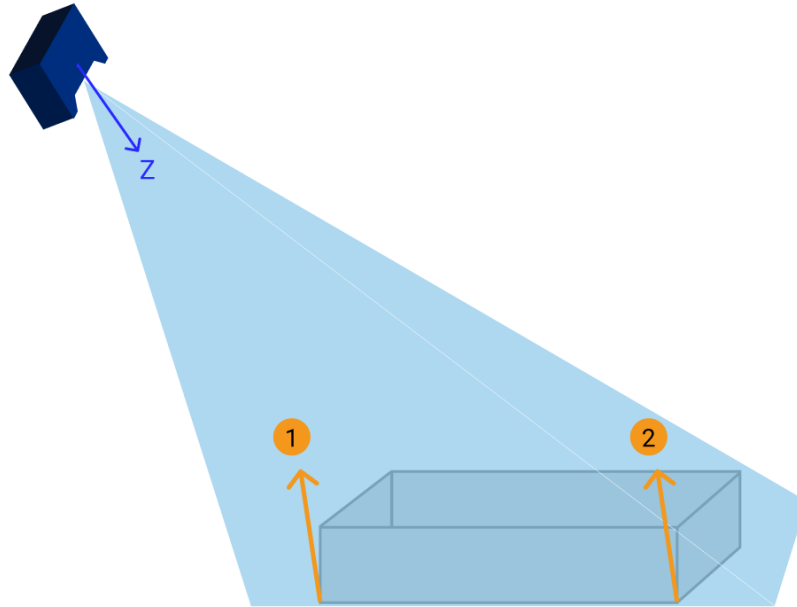
该工具用于自定义显示坐标系。可在自定义的坐标系下显示深度图和点云。



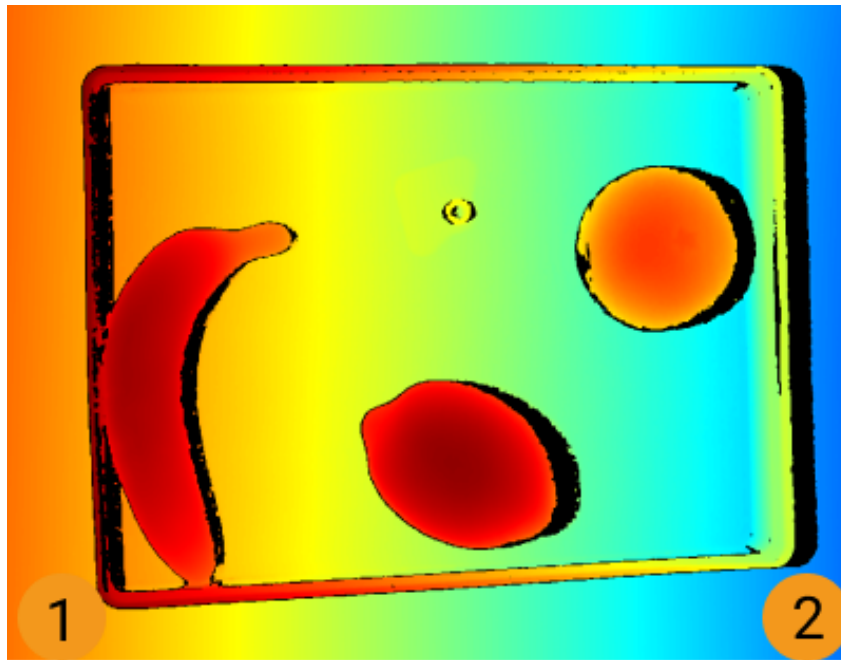
自定义坐标系仅影响Mech-Eye Viewer中的深度值显示，不影响深度图和点云中的实际深度数据。

### 适用场景

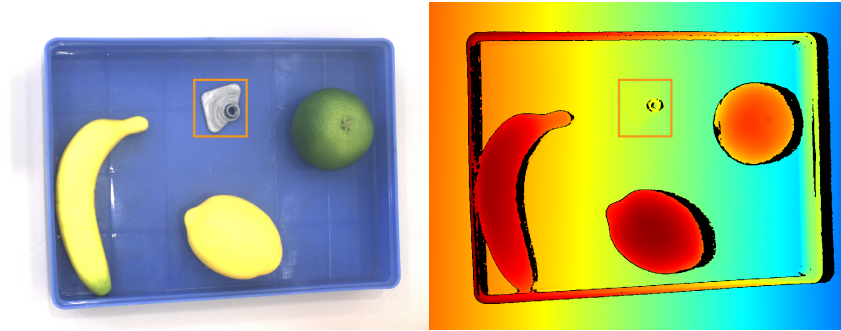
适用于相机未安装在目标物体正上方的情况，如下图所示。



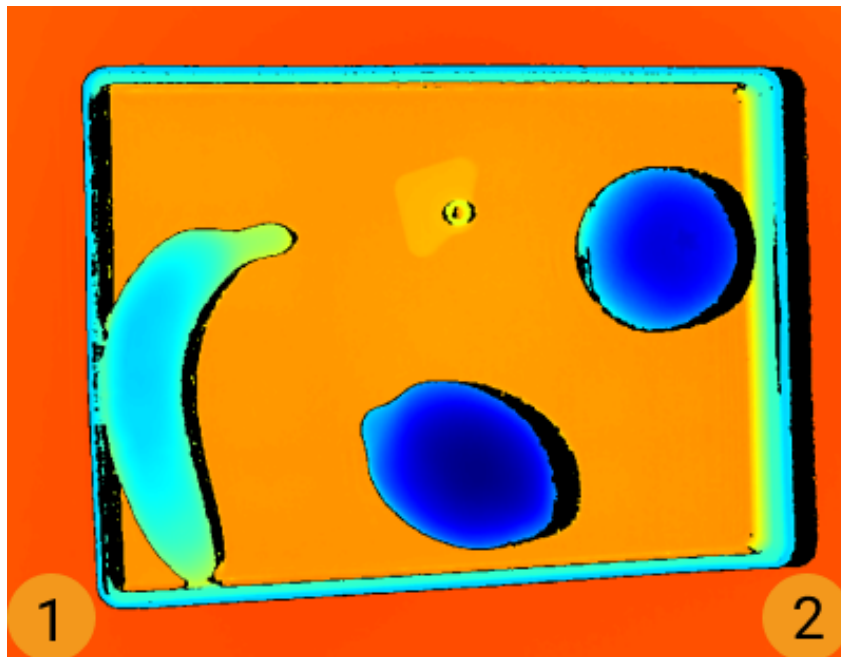
上图中①和②处，对于一个观察者来说，位于同一高度。但因为相机坐标系的Z轴不垂直于物体所在平面，所获得的深度图和点云中，这两点的深度值是不同的。



另外，在相机坐标系下，很薄的目标物体的自身厚度可能被所在平面的深度值变化掩盖，从而无法轻易观察到。



通过自定义坐标系，可调整深度图和点云的显示，使①和②处的深度值显示为相同的，且能够观察到很薄的目标物体，如下图所示。



### 操作步骤

请执行以下步骤自定义坐标系：

1. [调整点云的位置](#)，找到应作为自定义坐标系XY平面的表面。
2. 按住 **Shift** 键并在点云上单击来选择三个点。三个点的作用如下：
  - 第一个点为坐标系原点；
  - 第二个点用于确定X轴正方向；
  - 第三个点用于确定Y轴正方向。
3. 三个点选择完成后，将根据右手原则自动生成坐标系。旋转和缩放点云，检查生成的坐标系是否符合需求：
  - 确认三点是否位于同一表面。
  - 确认Z轴方向是否正确。



如不符合，请单击右下角的[重置]后，重复步骤2。

4. 坐标系设置完成后，单击右下角的[确定]。

### 使用自定义坐标系

如需在自定义坐标系下显示深度图和点云，请执行以下步骤：

1. 在数据查看区，切换至**深度图**或**点云**。
2. 单击左边栏最上方的**坐标系**下拉菜单，选择**自定义**。深度图和点云将在自定义坐标系下显示。



坐标系的设置对深度图和点云同时生效。

### 5.4.7. 相机管理器

该工具用于查看相机的投光单元固件版本、时间、温度等信息及更改图像分辨率。



### 查看相机信息

**相机管理器**中可查看相机的以下信息。

- 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。




仅DLP相机显示此选项。

- 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。
- 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电脑系统时间。
- 温度：显示相机CPU和投光单元的温度。

单击[ **刷新参数** ]可更新温度数据。

## 更改图像分辨率

 该功能在**管理员**用户下可见。

部分相机型号的2D图或深度图分辨率可调节。选择较低分辨率可有效提升相机的采集速度，更好地应对高节拍场景。

- LSR (V4) 系列：2D图分辨率可选4000 × 3000或2000 × 1500；
- DEEP (V4) 系列：深度图分辨率可选2048 × 1536或1024 × 768。

请执行以下步骤更改相机图像的分辨率：


1. 切换至**管理员**用户后，打开**相机管理器**。
2. 在**分辨率**中的**2D图**或**深度图**下拉菜单中，选择需要的分辨率。
3. 单击下方的[ **应用设置并重启相机** ]，并在弹窗中单击[ **确定** ]，重启相机。

 重启相机需花费几分钟，请耐心等待。可单击左上角的刷新相机列表。

4. 相机重新出现在列表中后，连接相机。再次采集的2D图或深度图将为所选分辨率。

### 5.4.8. 满垛模拟器

该工具用于确认满垛时垛顶层是否全部位于相机视野内，可根据确认结果调节相机安装高度。

- 
  - 仅适用于相机垂直于托盘安装的场景。
  - 显示结果仅供参考。

## 准备工作

使用**满垛模拟器**前，请做好以下准备工作：

- 确认满垛时垛的尺寸。
- 确认相机安装位置是否有任何限制条件，如最大安装高度。
- 准备好托盘并放置于相机视野内，采集托盘的点云并确认质量。
- 调整托盘位置，使托盘点云的长短边与相机视野长短边分别平行。
- 设置**感兴趣区域**，只保留托盘对应的点云。

## 操作步骤

1. 单击工具 > 满垛模拟器，打开**满垛模拟器**。
2. **调整点云的位置**，直至可以从侧面看到整个点云。
3. 单击[ **创建长方体** ]，将自动创建模拟垛的长方体。



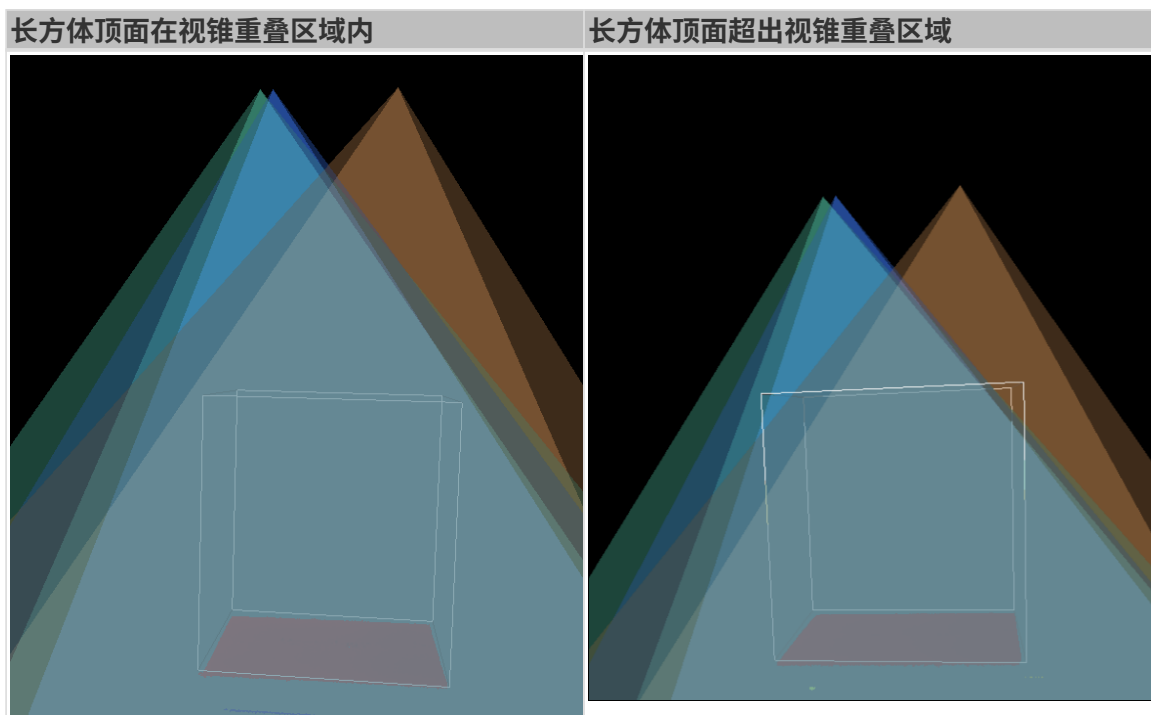
创建的长方体无法旋转。如长方体的边和托盘点云的边不平行，请退出**满垛模拟器**，调整托盘的位置，并重新采集图像。

4. 在**长方体尺寸**中输入满垛时垛的尺寸。
5. (可选) 调节长方体的位置：如长方体的底面不在托盘点云上，需调节**长方体底面中心位置**的 $z$ 值，直至长方体底面位于托盘点云上。



请记录 $z$ 的调整量。

6. 旋转点云，确认长方体顶面是否位于各视锥的重叠区域内。以下为示例。



7. 评估相机安装高度：

- 如长方体顶面在视锥重叠区域内，则相机安装高度合适。
- 如长方体顶面超出视锥重叠区域，则相机安装高度过低。请查看下一步。

8. 增加**长方体底面中心位置**的 $z$ 值，直至长方体顶面完全位于视锥重叠区域内。



请记录 $z$ 的调整量。

9. 计算相机安装高度调整量：步骤8中 $z$ 的调整量减去步骤5中 $z$ 的调整量，即为相机安装高度应向上提高的量。



满垛模拟器中的更改不会保存，请确保你记录了 $z$ 的调整量。

## 6. Mech-Eye API

你可使用Mech-Eye API连接相机，调节参数，获取并保存2D图、深度图和点云。

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。



Mech-Eye API 2.2.0进行了重构，在保留原有的全部功能的基础上，调整为更加清晰的结构，并提供新增功能。

- 如希望在已有客户端程序中使用2.2.0版本的Mech-Eye API，请参考[迁移指南](#)修改客户端程序。
- 如需将相机固件版本升级至2.2.0，但需继续在客户端程序中使用2.1.0版本的Mech-Eye API，请参考[向下兼容性声明](#)完成对应的处理。

查看以下内容，了解如何**安装Mech-Eye SDK**。

[Mech-Eye SDK安装指南（Windows）](#)

[Mech-Eye SDK安装指南（Ubuntu）](#)

查看以下内容，了解**Mech-Eye API的整体结构**。

[类的功能层级](#)

查看以下内容，了解**使用Mech-Eye API控制相机的流程**。

[使用流程](#)

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，了解如何**安装、配置、编译、运行Mech-Eye API例程**。

[例程使用指南](#)

查看以下内容，了解**辅助使用Mech-Eye API的工具**。

[固件升级工具](#)

[IP配置工具](#)

查看以下内容，了解2.0.0版本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](#)

 不适用于2.0.0之前的Mech-Eye API。

## 6.1. Mech-Eye SDK安装指南（Ubuntu）

本章介绍如何在Ubuntu操作系统中下载、安装、升级和卸载Mech-Eye SDK。

 如使用Windows系统，请参考[Mech-Eye SDK安装指南](#)。

### 下载Mech-Eye SDK

Mech-Eye SDK提供AMD64和ARM64两种架构的安装包：

- AMD64架构：**Mech-Eye\_API\_2.2.2\_amd64.deb**
- ARM64架构：**Mech-Eye\_API\_2.2.2\_arm64.deb**

请根据需要在[Mech-Eye SDK安装包](#)。

 使用`arch`命令可查看系统架构。

### 安装Mech-Eye SDK

 如已安装Mech-Eye SDK，请[升级Mech-Eye SDK](#)。

请执行以下步骤在Ubuntu系统上安装Mech-Eye SDK：

1. 进入安装包所在路径后，执行以下命令安装Mech-Eye SDK：

- 架构为AMD64时，执行以下命令：

```
sudo dpkg -i 'Mech-Eye_API_2.2.2_amd64.deb'
```

- 架构为ARM64时，执行以下命令：

```
sudo dpkg -i 'Mech-Eye_API_2.2.2_arm64.deb'
```

2. 执行以下命令确认安装是否成功。如打印了Mech-Eye SDK版本，则安装成功。

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

## 升级Mech-Eye SDK

如已安装Mech-Eye SDK，请执行以下步骤升级Mech-Eye SDK：

1. 执行以下命令查询是否已安装Mech-Eye SDK：

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

2. 执行以下命令卸载已安装的Mech-Eye SDK：

```
sudo dpkg -P mecheyeapi
```

3. 进入安装包所在路径后，执行以下命令安装Mech-Eye SDK：

- 架构为AMD64时，执行以下命令：

```
sudo dpkg -i 'Mech-Eye_API_2.2.2_amd64.deb'
```

- 架构为ARM64时，执行以下命令：

```
sudo dpkg -i 'Mech-Eye_API_2.2.2_arm64.deb'
```

4. 执行以下命令确认安装是否成功。如打印了Mech-Eye SDK版本，则安装成功。

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

## 卸载Mech-Eye SDK

请执行以下步骤卸载Mech-Eye SDK：

1. 执行以下命令卸载已安装的Mech-Eye SDK：

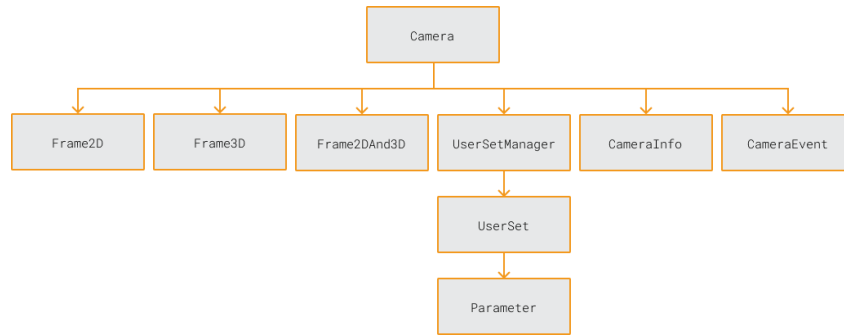
```
sudo dpkg -P mecheyeapi
```

2. 执行以下命令确认卸载是否成功。如未打印任何信息，则卸载成功。

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

## 6.2. 类的功能层级

本章介绍Mech-Eye API中类的功能层级。在下图中，位于高层级的类提供访问位于低层级的类对象的方法。因此，在客户端程序中应先实例化高层级的类。



以下小节介绍上图中各类的功能。

### Camera

**Camera**类代表一台相机，为Mech-Eye API中功能层级最高的类。在客户端程序中，应首先实例化该类。通过**Camera**类，可访问其余类的对象。

**Camera**类提供的方法主要包含：

- 发现相机
- 连接相机
- 采集数据
- 获取相机的信息
- 访问**UserSetManager**和**UserSet**类的对象

### Frame2D

**Frame2D**类用于存储2D数据。使用该类中的数据可生成2D图。

### Frame3D

**Frame3D**类用于存储3D数据。使用该类中的数据可生成深度图和无纹理点云。

### Frame2DAnd3D

**Frame2DAnd3D**类用于同时存储2D和3D数据。使用该类中的数据可生成2D图、深度图和纹理点云。



## UserSetManager

**UserSetManager**类提供管理参数组所需的方法，主要包含：

- 查看所有可用参数组
- 选择参数组
- 创建参数组
- 删除参数组
- 导入参数组
- 导出参数组
- 访问**UserSet**类的对象

## UserSet

**UserSet**类代表一个参数组，提供与当前所选参数组相关的方法。这些方法主要包含：

- 获取当前参数组中所有参数
- 重置当前参数组
- 保存参数至当前参数组
- 获取参数信息
- 获取和调节参数值
- 重命名当前参数组

## Parameter

**Parameter**类代表单独的参数，提供与单个参数相关的方法。这些方法主要包含：

- 获取参数信息（包含参数名称、参数说明、数据类型、最大值、最小值、最小调节单位、读写权限等）
- 获取和调节参数值

## CameraInfo等

以下类用于存储相机的各类信息和属性：

- **CameraInfo**：型号、序列号、硬件和固件版本、IP配置等
- **CameraStatus**：相机投影单元和CPU温度
- **CameraResolutions**：2D图和深度图的分辨率
- **CameraIntrinsics**：相机内参

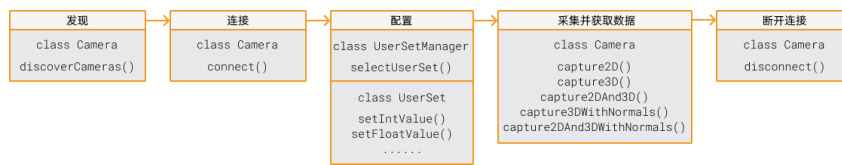
## CameraEvent

**CameraEvent**类提供用于检测相机相关事件的方法，当前可以检测相机断开连接的事件。

本章介绍了Mech-Eye API中类的功能层级。下一章将介绍使用Mech-Eye API控制相机的基本流程。

## 6.3. 使用流程

本章介绍使用Mech-Eye API控制相机的基本流程。本章图片及描述中使用C++ Mech-Eye API的类和方法名称，不同语言的对应名称详见代码示例。



### 发现

调用**Camera**类中的**discoverCameras()**方法枚举当前可连接的全部相机，并获取各个相机的信息。

#### C++

```
std::vector<mmind::eye::CameraInfo> cameraInfoList = mmind::eye::Camera
::discoverCameras();
```

#### C#

```
var cameraInfoList = Camera.DiscoverCameras();
```

#### Python

```
camera_info_list = Camera.discover_cameras()
```

### 连接

实例化**Camera**类后，调用该类中的**connect()**方法，通过IP地址或由**discoverCameras()**方法获取的设备信息连接对应的相机。

#### C++

```
mmind::eye::Camera camera;
mmind::eye::ErrorStatus status = camera.connect(cameraInfoList[inputIndex]);
```

```
mmind::eye::Camera cameras;
mmind::eye::ErrorStatus status = camera.connect("192.168.0.10");
```

### C#

```
var camera = new Camera();
var status = camera.Connect(cameraInfoList[inputIndex]);
```

```
var camera = new Camera();
var status = camera.Connect("192.168.0.10");
```

### Python

```
camera = Camera()
camera.connect("192.168.0.10")
```

## 配置

调用**UserSetManager**类和**UserSet**类中的方法，选择相机的参数组并调节参数。

1. 调用**UserSetManager**类中的**selectUserSet**方法选择需使用的参数组。



可通过该类中的**getAllUserSetNames()**方法获取相机所有可用参数组。

### C++

```
mmind::eye::UserSetManager userSetManager = camera.userSetManager();
std::vector<std::string> userSets;
auto status = userSetManager.getAllUserSetNames(userSets);
status = userSetManager.selectUserSet(userSets.front());
```

### C#

```
var userSetManager = camera.UserSetManager();
List<string> userSets = new List<string>();
var status = userSetManager.GetAllUserSetNames(ref userSets);
status = userSetManager.SelectUserSet(userSets[0]);
```

### Python

```
user_set_manager = camera.userSetManager()
error, user_sets = user_set_manager.get_all_user_set_names()
user_set_manager.select_user_set(user_sets[0])
```

- 调用 `UserSet` 类中的 `getAvailableParameters()` 方法获取当前参数组中的全部参数信息，并确认各个参数的数据类型。

#### C++

```
mmind::eye::UserSet currentUserSet = camera.currentUserSet();
std::vector<Parameter*> parameters = currentUserSet.getAvailableParameters();
```

#### C#

```
var currentUserSet = camera.CurrentUserSet();
var parameters = currentUserSet.GetAvailableParameters();
```

#### Python

```
current_user_set = camera.current_user_set()
arameters = current_user_set.get_available_parameters()
```

- 调用 `UserSet` 类中的 `getIntValue()` 及类似方法，获取某一参数当前的值。

#### C++

```
int fringeContrastThreshold = 0;
currentUserSet.getIntValue(mmind::eye::pointcloud_processing_setting::FringeContrastThreshold::name, fringeContrastThreshold);
```

#### C#

```
var fringeContrastThreshold = new int();
camera.CurrentUserSet().GetIntValue(MMind.Eye.PointCloudProcessingSetting.FringeContrastThreshold.Name, ref fringeContrastThreshold);
```

#### Python

```
error, fringe_contrast_threshold = current_user_set.get_int_value(
    PointCloudFringeContrastThreshold.name)
```

- 调用 `UserSet` 类中的 `setIntValue()` 及类似方法，设置某一参数的值。

#### C++

```
auto status = currentUserSet.setIntValue(mmind::eye::pointcloud_processing_setting::FringeContrastThreshold::name, 15);
```

**C#**

```
var status = currentUserSet.SetIntValue(MMind.Eye.PointCloudProcessingSetting.FringeContrastThreshold.Name, 15);
```

**Python**

```
status = current_user_set.set_int_value(PointCloudFringeContrastThreshold.name, 15)
```

5. 调用UserSet类中的saveAllParametersToDevice()方法，将设置的参数值保存至相机中。

**C++**

```
auto status = currentUserSet.SaveAllParametersToDevice();
```

**C#**

```
var status = currentUserSet.SaveAllParametersToDevice();
```

**Python**

```
status = current_user_set.save_all_parameters_to_device()
```

## 采集和获取数据

Mech-Eye API提供多种方法用于采集和获取数据，可单独采集2D或3D数据，也可同时采集2D和3D数据。

- 采集并获取用于生成2D图的数据：

**C++**

```
mmind::eye::Frame2D frame2D;
camera.capture2D(frame2D);
mmind::eye::Color2DImage color = frame2D.getColorImage();
mmind::eye::GrayScale2DImage gray= frame2D.getGrayScaleImage();
```

**C#**

```
var frame = new Frame2D();
camera.Capture2D(ref frame);
var color = frame.GetColorImage();
var gray = frame.GetGrayScaleImage();
```

### Python

```
frame_2d = Frame2D()
camera.capture_2d(frame_2d)
color = frame.get_color_image()
gray = frame.get_gray_scale_image()
```

- 采集并获取用于生成深度图和无纹理点云的数据：

### C++

```
mmind::eye::Frame3D frame3D;
camera.capture3D(frame3D);
mmind::eye::DepthMap depth = frame3D.getDepthMap();
mmind::eye::PointCloud cloud= frame3D.getUntexturedPointCloud();
```

### C#

```
var frame = new Frame3D();
camera.Capture3D(ref frame);
var depth = frame.GetDepthMap();
var cloud = frame.GetUntexturedPointCloud();
```

### Python

```
frame_3d = Frame3D()
camera.capture_3d(frame_3d)
depth = frame.get_depth_map()
cloud = frame_3d.get_untextured_point_cloud()
```

- 采集并获取用于生成2D图、深度图和纹理点云的数据：

### C++

```
mmind::eye::Frame2DAnd3D frame2DAnd3D;
camera.capture2DAnd3D(frame2DAnd3D);
mmind::eye::Color2DImage color = frame2DAnd3D.frame2D().getColorImage();
mmind::eye::DepthMap depth = frame2DAnd3D.frame3D().getDepthMap();
mmind::eye::TexturedPointCloud cloud = frame2DAnd3D.getTexturedPointCloud();
```

### C#

```
var frame = new Frame2DAnd3D();
camera.Capture2DAnd3D(ref frame);
var color = frame2DAnd3D.Frame2D().GetColorImage();
```

```
var depth = frame2DAnd3D.Frame3D().GetDepthMap();
var cloud = frame2DAnd3D.GetTexturedPointCloud();
```

### Python

```
frame = Frame2DAnd3D()
camera.capture_2d_and_3d(frame)
color = frame.frame_2d().get_color_image()
depth = frame.frame_3d().get_depth_image()
cloud = frame.get_textured_point_cloud()
```

- 采集并获取用于生成深度图和含法向量的无纹理点云的数据：

### C++

```
mmind::eye::Frame3D frame3D;
camera.capture3DWithNormal(frame3D);
mmind::eye::DepthMap depth = frame3D.getDepthMap();
PointCloudWithNormals pointCloud = frame3D.getUntexturedPointCloudWithNormals();
```

### C#

```
var frame3D = new Frame3D();
camera.Capture3DWithNormal(frame3D);
var depth = frame3D.GetDepthMap();
var pointCloud = frame3D.GetUntexturedPointCloudWithNormals();
```

### Python

```
frame_3d = Frame3D()
camera.capture_3d_with_normal(frame_3d)
depth = frame_3d.get_depth_map()
point_cloud = frame_3d.get_untextured_point_cloud_with_normals()
```

- 采集并获取用于生成2D图、深度图和含法向量的纹理点云的数据：

### C++

```
mmind::eye::Frame2DAnd3D frame2DAnd3D;
camera.capture2DAnd3DWithNormal(frame2DAnd3D);
mmind::eye::Color2DImage color = frame2DAnd3D.frame2D().getColorImage();
mmind::eye::DepthMap depth = frame2DAnd3D.frame3D().getDepthMap();
TexturedPointCloudWithNormals pointCloud = frame2DAnd3D
    .getTexturedPointCloudWithNormals();
```

**C#**

```
var frame2DAnd3D = new Frame2DAnd3D();
camera.Capture2DAnd3DWithNormal(ref frame2DAnd3D);
var color = frame2DAnd3D.Frame2D().GetColorImage();
var depth = frame2DAnd3D.Frame3D().GetDepthMap();
var pointCloud = frame2DAnd3D.GetTexturedPointCloudWithNormals();
```

**Python**

```
frame_2d_and_3d = Frame2DAnd3D()
camera.capture_2d_and_3d_with_normal(frame_2d_and_3d)
color = frame_2d_and_3d.frame_2d().get_color_image()
depth = frame_2d_and_3d.frame_3d().get_depth_map()
point_cloud = frame_2d_and_3d.get_textured_point_cloud_with_normals()
```

**断开连接**

调用Camera类中的disconnect()方法，断开与当前相机的连接。

**C++**

```
camera.disconnect();
```

**C#**

```
camera.Disconnect();
```

**Python**

```
camera.disconnect()
```

本章介绍了使用Mech-Eye API控制相机的基本流程。下一章提供Mech-Eye API的参考手册。

## 6.4. 迁移指南（从2.1.0版本到2.2.0版本）

Mech-Eye API 2.2.0进行了重构，在保留原有的全部功能的基础上，调整为更加清晰的结构，并提供新增功能。

本章列出Mech-Eye API 2.2.0相对于Mech-Eye API 2.1.0的主要变更，如希望在已有客户端程序中使用2.2.0版本的Mech-Eye API，可参考本章修改客户端程序。





- 关于Mech-Eye API 2.2.0的整体结构，请参阅[类的功能层级](#)。
- 关于使用Mech-Eye API 2.2.0控制相机的基本流程，请参阅[类的功能层级](#)。

## 导入模块

Mech-Eye API 2.2.0中，导入模块的语句发生以下变更：

2.1.0	2.2.0
<p><b>C++</b></p> <pre>#include "MechEyeApi.h"  using namespace mmind::api;</pre> <p><b>C#</b></p> <pre>using mmind.apiSharp;</pre> <p><b>Python</b></p> <pre>from MechEye import Device</pre>	<p><b>C++</b></p> <pre>#include "Camera.h"  using namespace mmind::eye;</pre> <p><b>C#</b></p> <pre>using MMind.Eye;</pre> <p><b>Python</b></p> <pre>from mecheye.shared import * from mecheye.area_scan_3d_camera import * from mecheye.area_scan_3d_camera_utils import *</pre>

## 创建相机对象

Mech-Eye API 2.2.0中，创建一个代表相机的对象的语句发生以下变更：

2.1.0	2.2.0
<p><b>C++</b></p> <pre>mmind::api::MechEyeDevice device;</pre> <p><b>C#</b></p> <pre>var device = new MechEyeDevice();</pre> <p><b>Python</b></p> <pre>device = Device()</pre>	<p><b>C++</b></p> <pre>mmind::eye::Camera camera;</pre> <p><b>C#</b></p> <pre>var camera = new Camera();</pre> <p><b>Python</b></p> <pre>camera = Camera()</pre>

## 发现与连接相机

Mech-Eye API 2.2.0中，发现与连接相机的语句发生以下变更：

2.1.0	2.2.0
<p><b>C++</b></p> <pre>std::vector&lt;mmind::api::MechEyeDeviceInfo&gt; deviceInfoList = mmind::api::MechEyeDevice::enumerateMechEyeDeviceList();  mmind::api::MechEyeDevice device; device.connect(deviceInfoList[0]);</pre> <p><b>C#</b></p> <pre>var deviceInfoList = MechEyeDevice     .EnumerateMechEyeDeviceList();  var device = new MechEyeDevice(); device.Connect(deviceInfoList[0]);</pre> <p><b>Python</b></p> <pre>device_list = Device     .get_device_list()  device = Device() device.connect(device_list[0])</pre>	<p><b>C++</b></p> <pre>std::vector&lt;mmind::eye::CameraInfo&gt; cameraInfoList = mmind::eye::Camera::discoverCameras();  mmind::eye::Camera camera; camera.connect(cameraInfoList[0]);</pre> <p><b>C#</b></p> <pre>var cameraInfoList = Camera     .DiscoverCameras();  var camera = new Camera(); camera.Connect(cameraInfoList[0]);</pre> <p><b>Python</b></p> <pre>camera_infos = Camera     .discover_cameras()  camera = Camera() camera.connect(camera_infos[0])</pre>

## 管理参数组

Mech-Eye API 2.2.0通过类对参数组和参数相关功能进行了分类和层级划分，详见[类的功能层级](#)。此处列举出常用功能的变更。



**2.1.0**
**C++**

```
std::vector<std::string> userSets;
device.getAllUserSets(userSets);

std::string currentUserSetName;
device.getCurrentUserSet(currentUserSet);

device.setCurrentUserSet(userSets.front());

device.addUserSet("NewSetting");

device.saveAllSettingsToUserSets();
```

**C#**

```
var userSets = new List<string>();
device.GetAllUserSets(ref userSets);

std::string currentUserSetName =
"";
device.GetCurrentUserSet(ref currentUserSetName);

device.SetCurrentUserSet(userSets[0])

device.AddUserSet("NewSetting");

device.SaveAllSettingsToUserSets();
```

**Python**

```
# Obtain the names of all parameter
groups.
user_sets = device
.get_all_user_sets()

# Get the name of the current
parameter group.
current_user_set = device
.get_current_user_set()

# Select a parameter group.
device.set_current_user_set(user_se
```

**2.2.0**
**C++**

```
std::vector<std::string> userSets;
mmind::eye::UserSetManager&
userSetManager = camera
.userSetManager();
userSetManager.getAllUserSetNames(u
serSets);

mmind::eye::UserSet& curSettings =
userSetManager.currentUserSet();
std::string currentName;
curSettings.getName(currentName);

userSetManager.selectUserSet(userSe
ts.front());

userSetManager.addUserSet("NewSetti
ng");

camera.currentUserSet().saveAllPara
metersToDevice();
```

**C#**

```
var userSetManager = camera
.UserSetManager();
var userSets = new List<string>();
userSetManager.GetAllUserSetNames(r
ef userSets);

var currentName = userSetManager
.CurrentUserSet().GetName();

userSetManager.SelectUserSet(userSe
ts[0]);

userSetManager.AddUserSet("NewSetti
ng");

camera.CurrentUserSet().SaveAllPara
metersToDevice();
```

**Python**

```
# Obtain the names of all parameter
groups.
user_set_manager = camera
.userSetManager()
```

## 设置与获取参数值

Mech-Eye API 2.2.0通过类对参数组和参数相关功能进行了分类和层级划分，详见[类的功能层级](#)。

同时，Mech-Eye API 2.2.0按照参数的数据类型提供通用的方法，用于设置与获取参数值。

Mech-Eye API 2.2.0区分了以下参数数据类型：

- `_Int`
- `_Float`
- `_Bool`
- `_Enum`
- `_Roi`
- `_Range`
- `_FloatArray`

以下小节给出各数据类型的示例。

### `_Int`类型

以点云后处理分组下的**条纹对比度阈值**参数为例。

2.1.0	2.2.0
<p><b>C++</b></p> <pre>device.setFringeContrastThreshold(15);  int fringeContrastThreshold = 0; device.getFringeContrastThreshold(fringeContrastThreshold);</pre> <p><b>C#</b></p> <pre>device.SetFringeContrastThreshold(15);  var fringeContrastThreshold = new int(); device.GetFringeContrastThreshold(ref fringeContrastThreshold);</pre> <p><b>Python</b></p> <pre>device.set_fringe_contrast_threshold(15)  fringe_contrast_threshold = device.get_fringe_contrast_threshold()</pre>	<p><b>C++</b></p> <pre>#include "area_scan_3d_camera/parameters/PointCloudProcessing.h"  camera.currentUserSet().setIntValue(mmind::eye::pointcloud_processing_setting::FringeContrastThreshold::name, 15);  int fringeContrastThreshold = 0; camera.currentUserSet().getIntValue(mmind::eye::pointcloud_processing_setting::FringeContrastThreshold::name, fringeContrastThreshold);</pre> <p><b>C#</b></p> <pre>camera.CurrentUserSet().SetIntValue(MMind.Eye.PointCloudProcessingSetting.FringeContrastThreshold.Name, 15);  var fringeContrastThreshold = new int(); camera.CurrentUserSet().GetIntValue(MMind.Eye.PointCloudProcessingSetting.FringeContrastThreshold.Name, ref fringeContrastThreshold);</pre> <p><b>Python</b></p> <pre>camera.current_user_set().set_int_value(PointCloudFringeContrastThreshold.name, 15)  error, fringe_contrast_threshold = camera.current_user_set().get_int_value(PointCloudFringeContrastThreshold.name)</pre>

## Float类型

以2D参数分组下的曝光时间参数为例。

2.1.0	2.2.0
<p><b>C++</b></p> <pre>device.setScan2DExposureTime(100);  double scan2DExposureTime = 0; device.getScan2DExposureTime(scan2DExposureTime);</pre> <p><b>C#</b></p> <pre>device.setScan2DExposureTime(100);  var scan2DExposureTime = new double(); device.GetScan2DExposureTime(ref scan2DExposureTime);</pre> <p><b>Python</b></p> <pre>device.set_scan_2d_exposure_time(100)  scan_2d_exposure_time = device .get_scan_2d_exposure_time()</pre>	<p><b>C++</b></p> <pre>#include "area_scan_3d_camera/parameters/Scanning2D.h"  camera.currentUserSet().setFloatValue(mmind::eye::scanning2d_setting::ExposureTime::name, 100);  double scan2DExposureTime = 0; camera.currentUserSet().getFloatValue(mmind::eye::scanning2d_setting::ExposureTime::name, scan2DExposureTime);</pre> <p><b>C#</b></p> <pre>camera.CurrentUserSet().SetFloatValue(MMind.Eye.Scanning2DSetting.ExposureTime.Name, true);  var scan2DExposureTime = new double(); camera.CurrentUserSet().GetFloatValue(MMind.Eye.Scanning2DSetting.ExposureTime.Name, ref scan2DExposureTime);</pre> <p><b>Python</b></p> <pre>camera.current_user_set().set_float_value(Scanning2DExposureTime.name, 100)  error, scan_2d_exposure_time = camera.current_user_set().get_float_value(Scanning2DExposureTime.name)</pre>

## Bool类型

以2D参数分组下的色调映射参数为例。

2.1.0	2.2.0
<p><b>C++</b></p> <pre>device.setScan2DToneMappingEnable(true);  bool toneMappingEnable = false; device.getScan2DToneMappingEnable(toneMappingEnable);</pre> <p><b>C#</b></p> <pre>device.SetScan2DToneMappingEnable(true);  var toneMappingEnable = new bool(); device.GetScan2DToneMappingEnable(ref toneMappingEnable);</pre> <p><b>Python</b></p> <pre>device.set_scan_2d_tone_mapping_enable(True)  tone_mapping_enable = device.get_scan_2d_tone_mapping_enable()</pre>	<p><b>C++</b></p> <pre>#include "area_scan_3d_camera/parameters/Scanning2D.h"  camera.currentUserSet().setBoolValue(mmind::eye::scanning2d_setting::ToneMappingEnable::name, true);  bool toneMappingEnable = false; camera.currentUserSet().getBoolValue(mmind::eye::scanning2d_setting::ToneMappingEnable::name, toneMappingEnable);</pre> <p><b>C#</b></p> <pre>camera.CurrentUserSet().SetBoolValue(MMind.Eye.Scanning2DSetting.ToneMappingEnable.Name, true);  var toneMappingEnable = new bool(); camera.CurrentUserSet().GetBoolValue(MMind.Eye.Scanning2DSetting.ToneMappingEnable.Name, ref toneMappingEnable);</pre> <p><b>Python</b></p> <pre>camera.current_user_set().set_bool_value(Scanning2DToneMappingEnable.name, True)  error, tone_mapping_enable = camera.current_user_set().get_bool_value(Scanning2DToneMappingEnable.name)</pre>

## \_Enum类型

以点云后处理分组中的表面平滑参数为例。



## 2.1.0

**C++**

```
device.setCloudSurfaceSmoothingMode
(mmind::api::PointCloudProcessingSe
ttings::PointCloudSurfaceSmoothing:
:Normal);

mmind::api::PointCloudProcessingSet
tings::PointCloudSurfaceSmoothing
surfaceSmoothing;
device.getCloudSurfaceSmoothingMode
(surfaceSmoothing);
```

**C#**

```
device.SetCloudSurfaceSmoothingMode
(PointCloudSurfaceSmoothing.Normal)
;

var surfaceSmoothingMode = new
PointCloudSurfaceSmoothing();
device.GetCloudSurfaceSmoothingMode
(ref surfaceSmoothingMode);
```

**Python**

```
device.set_cloud_surface_smoothing_
mode("Normal")
cloud_surface_smoothing_mode =
device.get_cloud_surface_smoothing_
mode()
```

## 2.2.0

**C++**

```
#include
"area_scan_3d_camera/parameters/Poi
ntCloudProcessing.h"

camera.currentUserSet().setEnumValu
e(mmind::eye::pointcloud_processing
_setting::SurfaceSmoothing::name,
static_cast<int>(mmind::
eye::pointcloud_processing_setting:
:SurfaceSmoothing::Value::Normal));

int surfaceSmoothing = 0;
camera.currentUserSet().currentUser
Set.getEnumValue(mmind::eye::pointc
loud_processing_setting::SurfaceSmo
othing::name, surfaceSmoothing);
```

**C#**

```
camera.currentUserSet().SetEnumValu
e(MMind.Eye.PointCloudProcessingSet
ting.SurfaceSmoothing.Name, (int
)MMind.Eye.PointCloudProcessingSett
ing.SurfaceSmoothing.Value.Normal);

var surfaceSmoothingMode = new
int();
camera.currentUserSet().GetEnumValu
e(MMind.Eye.PointCloudProcessingSett
ing.SurfaceSmoothing.Name, ref
surfaceSmoothingMode);
```

**Python**

```
camera.current_user_set().set_enum_
value(PointCloudSurfaceSmoothing.na
me, PointCloudSurfaceSmoothing
.Value_Normal)
error, cloud_surface_smoothing_mode
= camera.current_user_set
().get_enum_value(PointCloudSurface
Smoothing.name)
```

## \_Roi类型

以感兴趣区域参数为例。

2.1.0	2.2.0
<p><b>C++</b></p> <pre>device.setScan3DROI(mmind::api::ROI(0, 0, 500, 500));  mmind::api::ROI roi; device.getScan3DROI(roi);</pre> <p><b>C#</b></p> <pre>device.SetScan3DROI(new ROI(0, 0, 500, 500));  var roi = new ROI(); device.GetScan3DROI(ref roi);</pre> <p><b>Python</b></p> <pre>device.set_scan_3d_roi(0, 0, 500, 500)  roi = device.get_scan_3d_roi()</pre>	<p><b>C++</b></p> <pre>#include "area_scan_3d_camera/parameters/Scanning3D.h"  camera.currentUserSet().setRoiValue(mmind::eye::scanning3d_setting::ROI::name, mmind::eye::ROI(0, 0, 500, 500));  mmind::eye::ROI roi; camera.currentUserSet().getRoiValue(mmind::eye::scanning3d_setting::ROI::name, roi);</pre> <p><b>C#</b></p> <pre>camera.CurrentUserSet().SetRoiValue(MMind.Eye.Scanning3DSetting.ROI.Name, new ROI(0, 0, 500, 500));  var roi = new ROI(); camera.CurrentUserSet().GetRoiValue(MMind.Eye.Scanning3DSetting.ROI.Name, ref roi);</pre> <p><b>Python</b></p> <pre>camera.current_user_set().set_roi_value(Scanning3DROI.name, ROI(0, 0, 500, 500))  error, roi = camera.current_user_set().get_roi_value(Scanning3DROI.name)</pre>

## \_Range类型

以深度范围参数为例。

## 2.1.0

**C++**

```
device.setDepthRange(mmind::api::DepthRange(100, 2000));

mmind::api::DepthRange depthRange;
device.getDepthRange(depthRange);
```

**C#**

```
device.SetDepthRange(new DepthRange(100, 2000));

val depthRange = new DepthRange();
device.GetDepthRange(ref depthRange);
```

**Python**

```
device.set_depth_range(100, 2000)

depth_range = device
.get_depth_range()
```

## 2.2.0

**C++**

```
#include "area_scan_3d_camera/parameters/Scanning3D.h"

camera.currentUserSet().setRangeValue(mmind::eye::scanning3d_setting::DepthRange::name, mmind::eye::Range<int>{100, 2000});

mmind::eye::Range<int> rangeValue;
camera.currentUserSet().getRangeValue(mmind::eye::scanning3d_setting::DepthRange::name, rangeValue);
```

**C#**

```
camera.CurrentUserSet().SetRangeValue(MMind.Eye.Scanning3DSetting.DepthRange.Name, new IntRange(100, 2000));

var range = new IntRange();
camera.CurrentUserSet().GetRangeValue(MMind.Eye.Scanning3DSetting.DepthRange.Name, ref range);
```

**Python**

```
camera.current_user_set().set_range_value(Scanning3DDepthRange.name, RangeInt(100, 1000))

error, range = camera
.current_user_set().get_range_value(Scanning3DDepthRange.name)
```

## \_FloatArray类型

以3D参数分组中的多个曝光时间为例。

2.1.0	2.2.0
<p><b>C++</b></p> <pre>device.setScan3DExposure(std::vector&lt;double&gt;{5, 10});  std::vector&lt;double&gt; exposureSequence; device.getScan3DExposure(exposureSequence);</pre> <p><b>C#</b></p> <pre>device.SetScan3DExposure(new List&lt;double&gt; { 5, 10 });  var exposureSequence = new List&lt;double&gt;(); device.GetScan3DExposure(ref exposureSequence);</pre> <p><b>Python</b></p> <pre>device.set_scan_3d_exposure([5.0, 10.0])  exposure_sequence = device.get_scan_3d_exposure()</pre>	<p><b>C++</b></p> <pre>#include "area_scan_3d_camera/parameters/Scanning3D.h"  camera.currentUserSet().setFloatArrayValue(mmind::eye::scanning3d_setting::ExposureSequence::name, std::vector&lt;double&gt;{5, 10});  std::vector&lt;double&gt; exposureSequence; camera.currentUserSet().getFloatArrayValue(mmind::eye::scanning3d_setting::ExposureSequence::name, exposureSequence);</pre> <p><b>C#</b></p> <pre>camera.CurrentUserSet().SetFloatArrayValue(MMind.Eye.Scanning3DSetting.ExposureSequence.Name, new List&lt;double&gt; { 5, 10 });  var exposureSequence = new List&lt;double&gt;(); camera.CurrentUserSet().GetFloatArrayValue(MMind.Eye.Scanning3DSetting.ExposureSequence.Name, ref exposureSequence);</pre> <p><b>Python</b></p> <pre>camera.current_user_set().set_float_array_value(Scanning3DExposureSequence.name, [5, 10])  error, exposure_sequence = camera.current_user_set().get_float_array_value(Scanning3DExposureSequence.name)</pre>

## 采集数据

Mech-Eye API 2.2.0中，采集数据的语句发生以下变更：

## 采集2D图

2.1.0	2.2.0
<p><b>C++</b></p> <pre>mmind::api::ColorMap color; device.captureColorMap(color);</pre> <p><b>C#</b></p> <pre>var color = new ColorMap(); device.CaptureColorMap(ref color);</pre> <p><b>Python</b></p> <pre>color_map = device.capture_color()</pre>	<p><b>C++</b></p> <pre>mmind::eye::Frame2D frame2D; camera.capture2D(frame2D); mmind::eye::Color2DImage color = frame3D.getColorImage();</pre> <p><b>C#</b></p> <pre>var frame = new Frame2D(); camera.Capture2D(ref frame); var color = frame.GetColorImage();</pre> <p><b>Python</b></p> <pre>frame_2d = Frame2D() camera.capture_2d(frame_2d) color = frame.get_color_image()</pre>

## 采集深度图

2.1.0	2.2.0
<p><b>C++</b></p> <pre>mmind::api::DepthMap depth; device.captureDepthMap(depth);</pre> <p><b>C#</b></p> <pre>var depth = new DepthMap(); device.CaptureDepthMap(ref depth);</pre> <p><b>Python</b></p> <pre>depth_map = device.capture_depth()</pre>	<p><b>C++</b></p> <pre>mmind::eye::Frame3D frame3D; camera.capture3D(frame3D); mmind::eye::DepthMap depth = frame3D.getDepthMap();</pre> <p><b>C#</b></p> <pre>var frame = new Frame3D(); camera.Capture3D(ref frame); var depth = frame.GetDepthMap();</pre> <p><b>Python</b></p> <pre>frame_3d = Frame3D() camera.capture_3d(frame_3d) depth = frame_3d.get_depth_map()</pre>

## 采集无纹理点云

2.1.0	2.2.0
<p><b>C++</b></p> <pre data-bbox="268 376 759 562">mmind::api::PointXYZMap pointXYZMap; device.capturePointXYZMap(pointXYZM ap);</pre> <p><b>C#</b></p> <pre data-bbox="268 636 759 822">var pointXYZMap = new PointXYZMap(); device.CapturePointXYZMap(ref pointXYZMap);</pre> <p><b>Python</b></p> <pre data-bbox="268 896 759 1010">points_xyz = device .capture_point_xyz()</pre>	<p><b>C++</b></p> <pre data-bbox="874 376 1366 562">mmind::eye::Frame3D frame3D; camera.capture3D(frame3D); mmind::eye::PointCloud cloud = frame3D.getUntexturedPointCloud();</pre> <p><b>C#</b></p> <pre data-bbox="874 636 1366 822">var frame = new Frame3D(); camera.Capture3D(ref frame); var cloud = frame .GetUntexturedPointCloud();</pre> <p><b>Python</b></p> <pre data-bbox="874 896 1366 1093">frame_3d = Frame3D() camera.capture_3d(frame_3d) cloud = frame_3d .get_untextured_point_cloud()</pre>

## 采集纹理点云

2.1.0	2.2.0
<p><b>C++</b></p> <pre>mmind::api::PointXYZBGRMap pointXYZBGRMap; device.capturePointXYZBGRMap(pointX YZBGRMap);</pre> <p><b>C#</b></p> <pre>var pointXYZBGRMap = new PointXYZBGRMap(); device.CapturePointXYZBGRMap(ref pointXYZBGRMap)</pre> <p><b>Python</b></p> <pre>points_xyz_bgr = device .capture_point_xyz_bgr()</pre>	<p><b>C++</b></p> <pre>mmind::eye::Frame2DAnd3D frame2DAnd3D; camera.capture2DAnd3D(frame2DAnd3D) ; mmind::eye::TexturedPointCloud cloud = frame2DAnd3D .getTexturedPointCloud();</pre> <p><b>C#</b></p> <pre>var frame = new Frame2DAnd3D(); camera.Capture2DAnd3D(ref frame); var cloud = frame .GetTexturedPointCloud();</pre> <p><b>Python</b></p> <pre>frame = Frame2DAnd3D() camera.capture_2d_and_3d(frame) cloud = frame .get_textured_point_cloud()</pre>

## 获取相机的信息和属性

Mech-Eye API 2.2.0通过类对相机的各类信息和属性进行了分类。详见[类的功能层级](#)。

### 获取相机信息

2.1.0	2.2.0
<p><b>C++</b></p> <pre> mmind::api::MechEyeDeviceInfo deviceInfo; device.getDeviceInfo(deviceInfo);  std::string model = deviceInfo .model; std::string serialNumber = deviceInfo.id; std::string firmwareVersion = deviceInfo.firmwareVersion;                     </pre> <p><b>C#</b></p> <pre> var deviceInfo = new MechEyeDeviceInfo(); device.GetDeviceInfo(ref deviceInfo);  var model = deviceInfo.model; var serialNumber = deviceInfo.id; var firmwareVersion = deviceInfo .firmwareVersion;                     </pre> <p><b>Python</b></p> <pre> device_info = device .get_device_info()                     </pre>	<p><b>C++</b></p> <pre> mmind::eye::CameraInfo cameraInfo; camera.getCameraInfo(cameraInfo);  std::string model = cameraInfo .model; std::string serialNumber = cameraInfo.serialNumber; std::string firmwareVersion = cameraInfo.firmwareVersion.toString ();                     </pre> <p><b>C#</b></p> <pre> var cameraInfo = new CameraInfo(); camera.GetCameraInfo(ref cameraInfo);  var model = cameraInfo.Model; var serialNumber = cameraInfo .SerialNumber; var firmwareVersion = cameraInfo .FirmwareVersion.ToString();                     </pre> <p><b>Python</b></p> <pre> camera_info = CameraInfo() camera.get_camera_info(camera_info)                     </pre>

## 获取相机温度



2.1.0	2.2.0
<p><b>C++</b></p> <pre> mmind::api::DeviceTemperature devicetemperature; device.getDeviceTemperature(devicet emperature);  float cpu = devicetemperature .cpuTemperature; float projector = devicetemperature.projectorModuleTe mperature;                     </pre> <p><b>C#</b></p> <pre> var temperature = new DeviceTemperature(); device.GetDeviceTemperature(ref temperature);  var cpu = temperature.cpu; var projector = temperature .projectorModule;                     </pre> <p><b>Python</b></p> <pre> device_temperature = device .get_device_temperature()  cpu = device_temperature .cpu_temperature() projector = device_temperature .projector_module_temperature()                     </pre>	<p><b>C++</b></p> <pre> mmind::eye::CameraStatus cameraStatus; camera.getCameraStatus(cameraStatus );  float cpu = cameraStatus .temperature.cpuTemperature; float projector = cameraStatus .temperature.projectorTemperature;                     </pre> <p><b>C#</b></p> <pre> var cameraStatus = new CameraStatus(); camera.GetCameraStatus(ref cameraStatus);  var cpu = cameraStatus.Temperature .CpuTemperature; var projector = cameraStatus .Temperature.ProjectorModuleTempera ture;                     </pre> <p><b>Python</b></p> <pre> camera_status = CameraStatus() camera.get_camera_status(camera_sta tus)  cpu = camera_status.temperature .cpu_temperature projector = camera_status .temperature.projector_module_tempe rature                     </pre>

## 获取图像分辨率

2.1.0	2.2.0
<p><b>C++</b></p> <pre> mmind::api::DeviceResolution deviceResolution; device.getDeviceResolution(deviceRe solution);  unsigned int textureWidth = deviceResolution.colorMapWidth; unsigned int textureHeight = deviceResolution.colorMapHeight; unsigned int depthWidth = deviceResolution.depthMapWidth; unsigned int depthHeight = deviceResolution.depthMapHeight;                     </pre> <p><b>C#</b></p> <pre> var deviceResolution = new DeviceResolution(); device.GetDeviceResolution(ref deviceResolution);  var textureWidth = deviceResolution.colorMapWidth; var textureHeight = deviceResolution.colorMapHeight; var depthWidth = deviceResolution .depthMapWidth; var depthHeight = deviceResolution .depthMapHeight;                     </pre> <p><b>Python</b></p> <pre> device_resolution = device .get_device_resolution()  texture_width = device_resolution .color_width() texture_height = device_resolution .color_height() depth_width = device_resolution .depth_width() depth_height = device_resolution .depth_height()                     </pre>	<p><b>C++</b></p> <pre> mmind::eye::CameraResolutions cameraResolutions; camera.getCameraResolutions(cameraR esolutions);  unsigned int textureWidth = cameraResolutions.texture.width; unsigned int textureHeight = cameraResolutions.texture.height; unsigned int depthWidth = cameraResolutions.depth.width; unsigned int depthHeight = cameraResolutions.depth.height;                     </pre> <p><b>C#</b></p> <pre> var cameraResolutions = new CameraResolutions(); camera.GetCameraResolutions(ref cameraResolutions);  var textureWidth = cameraResolutions.Texture.Width; var textureHeight = cameraResolutions.Texture.Height; var depthWidth = cameraResolutions .Depth.Width; var depthHeight = cameraResolutions.Depth.Height;                     </pre> <p><b>Python</b></p> <pre> camera_resolutions = CameraResolutions() camera.get_camera_resolutions(camer a_resolutions)  texture_width = camera_resolutions .texture.width texture_height = camera_resolutions.texture.height depth_width = camera_resolutions .depth.width depth_height = camera_resolutions .depth.height                     </pre>

## 获取相机内参



## 2.1.0

## C++

```

mmind::api::DeviceIntri
deviceIntri;
device.getDeviceIntri(deviceIntri);

const mmind::eye::CameraIntri
textureIntri = deviceIntri
.textureCameraIntri;

const double fx = textureIntri
.cameraMatrix[0];
const double fy = textureIntri
.cameraMatrix[1];
const double cx = textureIntri
.cameraMatrix[2];
const double cy = textureIntri
.cameraMatrix[3];

const double k1 = textureIntri
.distortion[0];
const double k2 = textureIntri
.distortion[1];
const double p1 = textureIntri
.distortion[2];
const double p2 = textureIntri
.distortion[3];
const double k3 = textureIntri
.distortion[4];
    
```

## C#

```

var deviceIntri = new
DeviceIntri();
device.GetDeviceIntri(ref
deviceIntri);

var textureCameraIntri =
deviceIntri.textureCameraIntri;

const double fx =
textureCameraIntri.fx;
const double fy =
textureCameraIntri.fy;
const double cx =
textureCameraIntri.cx;
const double cy =
textureCameraIntri.cy;

const double k1 =
    
```

## 2.2.0

## C++

```

mmind::eye::CameraIntrinsics
intrinsics;
camera.getCameraIntrinsics(intrinsi
cs);

const mmind::eye
::Intrinsics2dCamera
textureIntrinsics = intrinsics
.texture;
const mmind::eye::CameraMatrix
cameraMatrix = textureIntrinsics
.cameraMatrix;
const double fx = cameraMatrix.fx;
const double fy = cameraMatrix.fy;
const double cx = cameraMatrix.cx;
const double cy = cameraMatrix.cy;

const mmind::eye::CameraDistortion
distCoeffs = intrinsics.texture
.cameraDistortion;
const double k1 = distCoeffs.k1;
const double k2 = distCoeffs.k2;
const double p1 = distCoeffs.p1;
const double p2 = distCoeffs.p2;
const double k3 = distCoeffs.k3;
    
```

## C#

```

var intrinsics = new
CameraIntrinsics();
camera.GetCameraIntrinsics(ref
intrinsics);

var textureIntrinsics = intrinsics
.Texture;

const double fx =
textureIntrinsics.CameraMatrix.Fx;
const double fy =
textureIntrinsics.CameraMatrix.Fy;
const double cx =
textureIntrinsics.CameraMatrix.Cx;
const double cy =
textureIntrinsics.CameraMatrix.Cy;

const double k1 =
textureIntrinsics.CameraDistortion.
K1;
    
```

## 新增功能

Mech-Eye API 2.2.0新增了多个便捷的新功能。



如需使用以下新增功能，则必须参考本章修改客户端程序，以应用2.2.0版本的Mech-Eye API。

## 计算点云法向量

调用以下新增方法，可直接获取包含法向量的3D数据，减少后续数据处理的工作量。

### 获取带法向量的无纹理点云

#### C++

```
mmind::eye::Frame3D frame3D;
camera.capture3DWithNormal(frame3D);

PointCloudWithNormals pointCloud = frame3D.getUntexturedPointCloudWithNormals();
```

#### C#

```
var frame3D = new Frame3D();
camera.Capture3DWithNormal(frame3D);

var pointCloud = frame3D.GetUntexturedPointCloudWithNormals();
```

#### Python

```
frame_3d = Frame3D()
camera.capture_3d_with_normal(frame_3d)

point_cloud = frame_3d.get_untextured_point_cloud_with_normals()
```

### 获取带法向量的纹理点云

#### C++

```
mmind::eye::Frame2DAnd3D frame2DAnd3D;
camera.capture2DAnd3DWithNormal(frame2DAnd3D);

TexturedPointCloudWithNormals pointCloud =
frame2DAnd3D.getTexturedPointCloudWithNormals();
```

#### C#

```

var frame2DAnd3D = new Frame2DAnd3D();
camera.Capture2DAnd3DWithNormal(ref frame2DAnd3D);

var pointCloud = frame2DAnd3D.GetTexturedPointCloudWithNormals();
    
```

### Python

```

frame_2d_and_3d = Frame2DAnd3D()
camera.capture_2d_and_3d_with_normal(frame_2d_and_3d)

point_cloud = frame_2d_and_3d.get_textured_point_cloud_with_normals()
    
```

## 保存点云

调用以下新增方法，可直接保存PLY、PCD或CSV格式的点云，无需依赖第三方软件库。

### 保存无纹理点云

#### C++

```

mmind::eye::Frame3D frame3D;
camera.capture3D(frame3D);

frame3D.saveUntexturedPointCloud(mmind::eye::FileFormat::PLY, "PointCloud.ply");
    
```

#### C#

```

var frame3D = new frame3D();
camera.Capture3D(frame3D);

frame3D.SaveUntexturedPointCloud(FileFormat.PLY, "PointCloud.ply");
    
```

#### Python

```

frame_3d = Frame3D()
camera.capture_3d(frame_3d)

frame_3d.save_untextured_point_cloud(FileFormat.PLY, "PointCloud.ply")
    
```

### 保存纹理点云

#### C++

```

mmind::eye::Frame2DAnd3D frame2DAnd3D;
camera.capture2DAnd3D(frame2DAnd3D);

frame2DAnd3D.saveTexturedPointCloud(mmind::eye::FileFormat::PLY,
"TexturedPointCloud.ply");
    
```

### C#

```

var frame2DAnd3D = new Frame2DAnd3D();
camera.Capture2DAnd3D(ref frame2DAnd3D);

frame2DAnd3D.SaveTexturedPointCloud(FileFormat.PLY, "TexturedPointCloud.ply");
    
```

### Python

```

frame_2d_and_3d = Frame2DAnd3D()
camera.capture_2d_and_3d(frame_2d_and_3d)

frame_2d_and_3d.save_textured_point_cloud(FileFormat.PLY, "TexturedPointCloud.ply")
    
```

## 导入与导出参数组

调用以下新增方法，可将所有参数组保存为JSON文件，或从JSON文件导入并替换所有参数组。

### 导入参数组

#### C++

```
camera.userSetManager().loadFromFile("camera_config.json");
```

#### C#

```
camera.UserSetManager().LoadFromFile("camera_config.json");
```

#### Python

```
camera.user_set_manager().load_from_file("camera_config.json")
```

#### C++

```
camera.userSetManager().saveToFile("camera_config.json");
```



**C#**

```
camera.UserSetManager().SaveToFile("camera_config.json");
```

**Python**

```
camera.user_set_manager().save_to_file("camera_config.json")
```

## 检测相机连接状态

新增心跳信号发送机制，可用于检测相机断开连接的问题。

### 设置心跳信号的频率

通过以下方法可设置心跳信号的频率：

**C++**

```
camera.setHeartbeatInterval(1000);
```

**C#**

```
camera.SetHeartbeatInterval(1000);
```

**Python**

```
camera.set_heartbeat_interval(1000)
```

## 注册回调函数

通过以下方法注册回调函数，可自动检测到相机断开连接的情况并提供报错：

**C++**

```
mmind::eye::CameraEvent::EventCallback callback = [](mmind::eye::CameraEvent::Event
event, void* pUser) {
    std::cout << "A camera event has occurred. The event ID is " << event << "."
<< std::endl;
};

mmind::eye::CameraEvent::registerCameraEventCallback(camera, callback, nullptr,
mmind::eye::CameraEvent::CAMERA_EVENT_DISCONNECTED);
```

### C#

```
private static void CallbackFunc(CameraEvent.Event cameraEvent, IntPtr pUser)
{
    Console.WriteLine("A camera event has occurred. The event ID is {0}.",
        cameraEvent);
}

Utils.ShowError(CameraEvent.RegisterCameraEventCallback(ref camera, CallbackFunc,
    IntPtr.Zero, (uint)CameraEvent.Event.CAMERA_EVENT_DISCONNECTED));
```

### Python

```
class CustomCallback(EventCallbackBase):
    def __init__(self):
        super().__init__()

    def run(self, event):
        print("A camera event has occurred. The event ID is {0}.".format(event))

camera_event = CameraEvent()
callback = CustomCallback()
show_error(camera_event.register_camera_event_callback(camera, callback, CameraEvent
    .CAMERA_EVENT_ALL))
```

## 6.5. Mech-Eye API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](#)。

## 6.6. 例程使用指南

本章提供不同平台和语言的例程使用指南, 以及调用Mech-Eye API的第三方软件例程的使用指南。

- [C++ \(Windows\)](#)
- [C++ \(Ubuntu\)](#)
- [C#](#)
- [Python \(Windows\)](#)
- [Python \(Ubuntu\)](#)
- [VisionPro](#)

### 6.6.1. C++ (Windows)

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

例程分为以下类别：**Basic**、**Advanced**、**Util**、**Calibration**、**Pcl**和**Halcon**。

- **Basic**例程：连接和采集数据。
- **Advanced**例程：通过复杂、高阶的方式采集数据，设置部分型号特有参数。
- **Util**例程：获取相机信息和设置通用参数。
- **Calibration**例程：通过Mech-Eye API进行手眼标定。
- **Pcl**例程：利用PCL库进行数据格式转换、可视化等。
- **Halcon**例程：通过Mech-Eye API获取HALCON可读取的点云。

各分类中包含的例程及其简介如下。

### ▼ Basic

- **ConnectToCamera**：连接相机。
- **ConnectAndCaptureImages**：连接相机并获取2D图、深度图及点云数据。
- **Capture2DImage**：从相机获取并保存2D图。
- **CaptureDepthMap**：从相机获取并保存深度图。
- **CapturePointCloud**：从相机获取并保存无纹理点云和纹理点云。
- **CapturePointCloudHDR**：设置多个曝光时间，然后从相机获取并保存点云。
- **CapturePointCloudWithNormals**：计算法向量，并保存合法向量的点云。

### ▼ Advanced

- **ConvertDepthMapToPointCloud**：从深度图生成并保存点云。
- **MultipleCamerasCaptureSequentially**：使用多台相机按序获取并保存2D图、深度图及点云。
- **MultipleCamerasCaptureSimultaneously**：使用多台相机同时获取并保存2D图、深度图及点云。
- **CapturePeriodically**：在设定时间内，定时获取并保存2D图、深度图和点云。
- **Mapping2DImageToDepthMap**：从覆盖掩膜的2D图和深度图生成并保存无纹理点云和纹理点云。
- **SetParametersOfLaserCameras**：设置激光相机特有的参数。
- **SetParametersOfUHPCameras**：设置UHP系列相机特有的参数。
- **RegisterCameraEvent**：定义并注册检测相机连接状态的回调函数。
- **CaptureStereo2DImages**：获取Deep (V3)、Laser L Enhanced (V3)、PRO XS (V4)、LSR L (V4)、LSR S (V4) 和DEEP (V4) 的两个2D相机的2D图像。

### ▼ Util

- **GetCameraIntrinsics**: 获取并打印相机内参。
- **PrintCameraInfo**: 获取并打印相机型号、序列号、固件版本、温度等信息。
- **SetScanningParameters**: 设置**3D参数**、**2D参数**和**感兴趣区域**分组下的参数。
- **SetDepthRange**: 设置**深度范围**参数。
- **SetPointCloudProcessingParameters**: 设置**点云后处理**参数。
- **ManageUserSets**: 管理参数组，如获取所有参数组的名称、新增参数组、切换参数组和保存参数设置至参数组。
- **SaveAndLoadUserSet**: 从JSON文件导入并替换所有参数组，将所有参数组保存为JSON文件。

### ▼ Calibration

- **HandEyeCalibration**: 进行手眼标定。

### ▼ Pcl

- **ConvertPointCloudToPcl**: 从相机获取点云数据并转换为PCL数据结构。
- **ConvertPointCloudWithNormalsToPcl**: 从相机获取含法向量的点云数据并转换为PCL数据结构。

### ▼ Halcon

- **ConvertPointCloudToObjectModel3D**: 从相机获取点云数据，通过HALCON C++接口转换并保存点云。

## 使用前提

使用Mech-Eye API的C++例程，需先满足以下使用前提：

- [正确连接相机和电脑](#)。
- 获取例程。
- 安装必需软件。
- （可选）安装例程依赖的软件库，并添加相关的环境变量。

## 获取例程

C++例程包含在Mech-Eye SDK的安装路径中，也可从[GitHub](#)克隆获取。安装路径中的例程为Mech-Eye SDK发布时的版本，GitHub上的例程可能包含最新修改。

- 安装路径中的例程位于xxx/*Mech-Eye SDK-2.2.2/API/samples/cpp/area\_scan\_3d\_camera*路径下；
- 通过GitHub克隆获取的例程位于xxx/*mecheye\_cpp\_samples/area\_scan\_3d\_camera*路径下。

## 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:

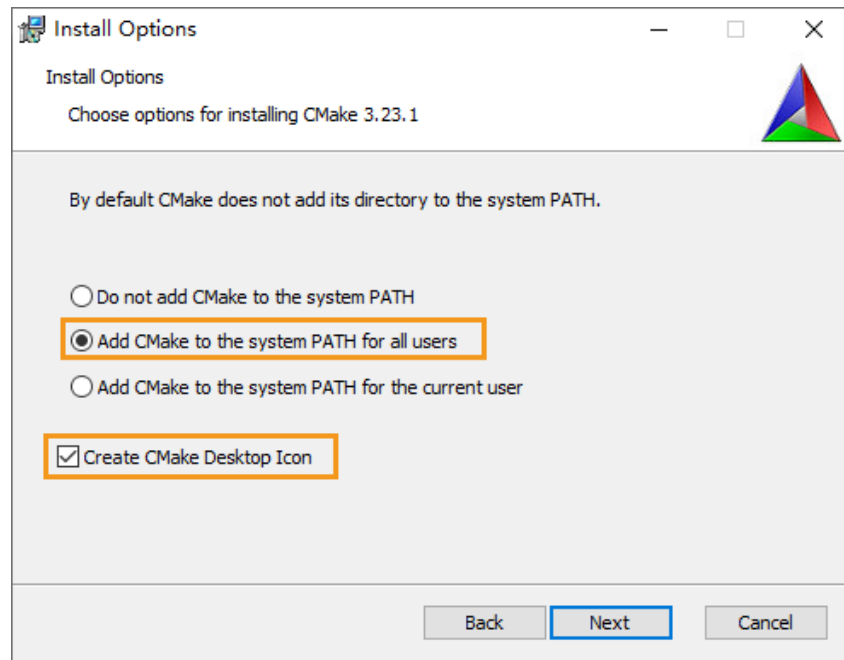
Platform	Files
Unix/Linux Source (has \n line feeds)	cmake-3.23.1.tar.gz
Windows Source (has \r\n line feeds)	cmake-3.23.1.zip

Binary distributions:

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

2. 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


 安装完成后，请重启电脑。如仍需安装例程依赖的软件库，可在[添加环境变量](#)后再重启。

### （可选）安装依赖的软件库

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库。

- 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

 低于20.11版本的HALCON未经完全验证。

 如未安装可选软件库，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
Capture2DImage	√		
CaptureDepthMap	√		
CapturePeriodically	√		
CaptureStereo2DImages	√		
ConvertPointCloudToObjectModel3D			√
ConvertPointCloudToPCL		√	
ConvertPointCloudWithNormalsToPCL		√	
HandEyeCalibration	√		
MultipleCamerasCaptureSequentially	√		
MultipleCamerasCaptureSimultaneously	√		

### 安装OpenCV

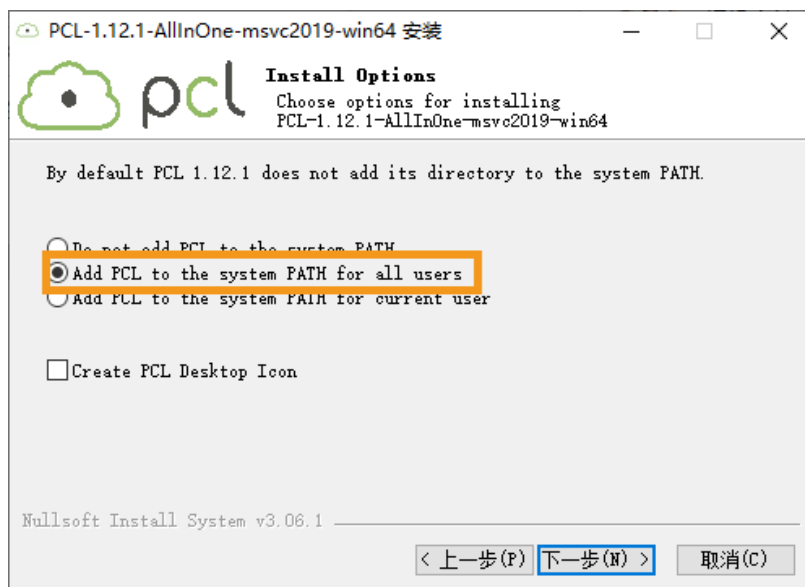
1. [下载并安装OpenCV](#)。请记录OpenCV的安装路径。
2. 将OpenCV添加至环境变量。详见[添加环境变量](#)。

### 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	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。



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[添加环境变量](#)。

4. 将PCL依赖的OpenNI2添加至环境变量。详见[添加环境变量](#)。

## 安装HALCON

安装 [HALCON20.11](#)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*
    - *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`



添加环境变量后，请重启电脑，否则可能导致添加的环境变量不生效。

## 构建及运行例程

你可以一次性构建全部例程，也可以仅构建单独一个例程。



本节以安装路径中包含的例程为例提供相关路径，如使用从GitHub克隆获取的例程，需对应更改路径。

## 禁用不需要的例程（可选）



未安装可选软件库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-2.2.2/API/samples/cpp/area_scan_3d_camera`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库的例程。

示例：不需要构建依赖HALCON的例程，则将**option(USE\_HALCON...)**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。

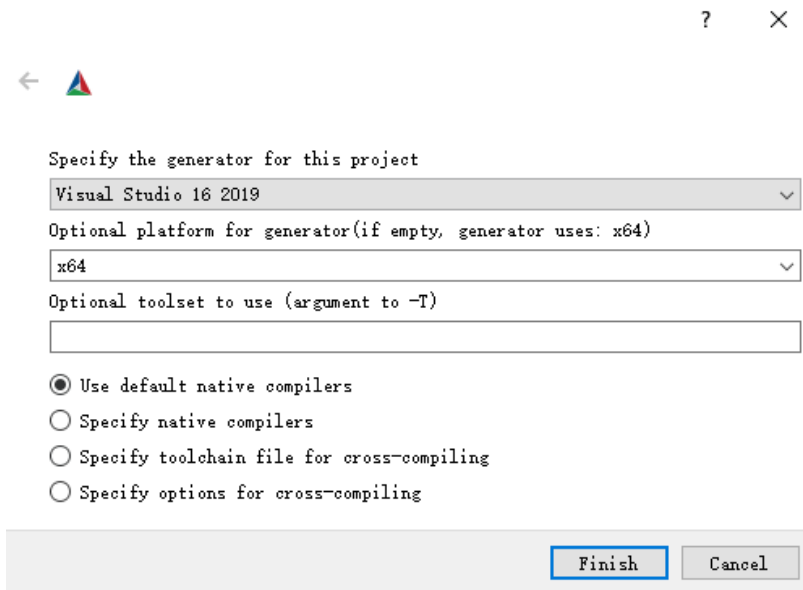
<b>Where is the source code</b>	<code>xxx/Mech-Eye SDK-2.2.2/API/samples/cpp/area_scan_3d_camera</code>
<b>Where to build the binaries</b>	<code>xxx/Mech-Eye SDK-2.2.2/API/samples/cpp/area_scan_3d_camera/build</code>

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

<b>Where is the source code</b>	<code>xxx/Mech-Eye SDK-2.2.2/API/samples/cpp/area_scan_3d_camera/Category/sampleName</code>
<b>Where to build the binaries</b>	<code>xxx/Mech-Eye SDK-2.2.2/API/samples/cpp/area_scan_3d_camera/Category/sampleName/build</code>



- 单击[ **Configure** ]，进入配置页面。选择Visual Studio的版本，并将平台设置为**x64**，然后单击[ **Finish** ]。配置成功后，日志最末行将显示**Configuring done**。



- 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

- 在Visual Studio工具栏中，将解决方案配置从**Debug**改为**Release**模式。
- 在菜单栏中选择生成 > 生成解决方案。每个例程会生成对应的可执行文件 (.exe)，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

你可以在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

- 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**。
- 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 (index)，并按 **Enter** 键连接相机。

- 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

- 进入在CMake中输入的**Where to build the binaries**目录，打开其中的**Release**文件夹。
- 运行与例程同名的可执行文件。

例程运行过程中，请根据提示输入待连接相机的编号（index），并按 **Enter** 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。

## 6.6.2. 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。

### 例程简介

例程分为以下类别：**Basic**、**Advanced**、**Util**、**Calibration**、**Pcl**和**Halcon**。

- **Basic**例程：连接和采集数据。
- **Advanced**例程：通过复杂、高阶的方式采集数据，设置部分型号特有参数。
- **Util**例程：获取相机信息和设置通用参数。
- **Calibration**例程：通过Mech-Eye API进行手眼标定。
- **Pcl**例程：利用PCL库进行数据格式转换、可视化等。
- **Halcon**例程：通过Mech-Eye API获取HALCON可读取的点云。

各分类中包含的例程及其简介如下。

#### ▼ Basic

- **ConnectToCamera**：连接相机。
- **ConnectAndCaptureImages**：连接相机并获取2D图、深度图及点云数据。
- **Capture2DImage**：从相机获取并保存2D图。
- **CaptureDepthMap**：从相机获取并保存深度图。
- **CapturePointCloud**：从相机获取并保存无纹理点云和纹理点云。
- **CapturePointCloudHDR**：设置多个曝光时间，然后从相机获取并保存点云。
- **CapturePointCloudWithNormals**：计算法向量，并保存合法向量的点云。

#### ▼ Advanced

- **ConvertDepthMapToPointCloud**：从深度图生成并保存点云。
- **MultipleCamerasCaptureSequentially**：使用多台相机按序获取并保存2D图、深度图及点云。
- **MultipleCamerasCaptureSimultaneously**：使用多台相机同时获取并保存2D图、深度图及点云。
- **CapturePeriodically**：在设定时间内，定时获取并保存2D图、深度图和点云。
- **Mapping2DImageToDepthMap**：从覆盖掩膜的2D图和深度图生成并保存无纹理点云和纹理点云。

- **SetParametersOfLaserCameras**: 设置激光相机特有的参数。
- **SetParametersOfUHPCameras**: 设置UHP系列相机特有的参数。
- **RegisterCameraEvent**: 定义并注册检测相机连接状态的回调函数。
- **CaptureStereo2DImages**: 获取Deep (V3)、Laser L Enhanced (V3)、PRO XS (V4)、LSR L (V4)、LSR S (V4) 和DEEP (V4) 的两个2D相机的2D图像。

#### ▼ Util

- **GetCameraIntrinsics**: 获取并打印相机内参。
- **PrintCameraInfo**: 获取并打印相机型号、序列号、固件版本、温度等信息。
- **SetScanningParameters**: 设置**3D参数**、**2D参数**和**感兴趣区域**分组下的参数。
- **SetDepthRange**: 设置**深度范围**参数。
- **SetPointCloudProcessingParameters**: 设置**点云后处理**参数。
- **ManageUserSets**: 管理参数组，如获取所有参数组的名称、新增参数组、切换参数组和保存参数设置至参数组。
- **SaveAndLoadUserSet**: 从JSON文件导入并替换所有参数组，将所有参数组保存为JSON文件。

#### ▼ Calibration

- **HandEyeCalibration**: 进行手眼标定。

#### ▼ Pcl

- **ConvertPointCloudToPcl**: 从相机获取点云数据并转换为PCL数据结构。
- **ConvertPointCloudWithNormalsToPcl**: 从相机获取含法向量的点云数据并转换为PCL数据结构。

#### ▼ Halcon

- **ConvertPointCloudToObjectModel3D**: 从相机获取点云数据，通过HALCON C++接口转换并保存点云。



该例程无法在ARM64架构的平台上使用。

### 使用前提

使用Mech-Eye API的C++例程，需先满足以下使用前提：

- [正确连接相机和电脑](#)。
- Ubuntu系统版本为18或以上。
- 获取例程。
- 安装必需软件。
- (可选) 安装例程依赖的软件库。



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。

## 获取例程

C++例程包含在Mech-Eye SDK的安装路径中，也可从[GitHub](#)克隆获取。安装路径中的例程为Mech-Eye SDK发布时的版本，GitHub上的例程可能包含最新修改。

- 安装路径中的例程位于`/opt/mech-mind/mech-eye-sdk/samples/cpp/area_scan_3d_camera`路径下；
- 通过GitHub克隆获取的例程位于`xxx/mecheye_cpp_samples/area_scan_3d_camera`路径下。

## 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[Mech-Eye SDK安装指南（Ubuntu）](#)。

## （可选）安装依赖的软件库

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库。

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



低于20.11版本的HALCON未经完全验证。



如未安装可选软件库，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
Capture2DImage	√		
CaptureDepthMap	√		
CapturePeriodically	√		

例程	OpenCV	PCL	HALCON
CaptureStereo2DImages	√		
ConvertPointCloudToObjectModel3D			√
ConvertPointCloudToPCL		√	
ConvertPointCloudWithNormalsToPCL		√	
HandEyeCalibration	√		
MultipleCamerasCaptureSequentially	√		
MultipleCamerasCaptureSimultaneously	√		

## 安装OpenCV

请执行以下命令安装最新版本的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

请执行以下命令安装PCL:

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



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

请执行以下步骤安装HALCON:

1. 下载HALCON安装包后，执行以下命令安装HALCON（以20.11版本为例）：

```
tar zxvfHALCON-20.11.3.0-linux.tar.gz
sudo sh install-linux.sh #Note down the installation directory ofHALCON.
```

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; exportHALCONARCH
HALCONROOT="/opt/halcon"; exportHALCONROOT
```

```

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
    
```



- 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。

## 构建及运行例程

你可以一次性构建全部例程，也可以仅构建单独一个例程。



本节以安装路径中包含的例程为例提供相关路径，如使用从GitHub克隆获取的例程，需对应更改路径。

## 禁用不需要的例程（可选）



未安装可选软件库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/cpp/area_scan_3d_camera/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库的例程。

示例：不需要构建依赖HALCON的例程，则将 `option(USE_HALCON...)` 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

请执行以下步骤构建例程：

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/cpp/area_scan_3d_camera/
```

- 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/cpp/area_scan_3d_camera/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。

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

3. 配置并构建例程：

```
sudo cmake ..  
sudo make
```



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和第3步，重新构建例程。

## 运行例程

构建例程后，请执行以下步骤运行例程：

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/cpp/area_scan_3d_camera/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/cpp/area_scan_3d_camera/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（index），并按 **Enter** 键连接相机。

```
sudo ./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。

### 6.6.3. C#

本章介绍如何在Windows系统中使用Visual Studio构建Mech-Eye API的C#例程。

## 例程简介

例程分为以下类别：**Basic**、**Advanced**和**Util**。

- **Basic**例程：连接和采集数据。
- **Advanced**例程：通过复杂、高阶的方式采集数据，设置部分型号特有参数。
- **Util**例程：获取相机信息和设置通用参数。

各分类中包含的例程及其简介如下。

### ▼ Basic

- **ConnectToCamera**：连接相机。
- **ConnectAndCaptureImages**：连接相机并获取2D图、深度图及点云数据。
- **Capture2DImage**：从相机获取并保存2D图。
- **CaptureDepthMap**：从相机获取并保存深度图。
- **CapturePointCloud**：从相机获取并保存无纹理点云和纹理点云。
- **CapturePointCloudHDR**：设置多个曝光时间，然后从相机获取并保存点云。
- **CapturePointCloudWithNormals**：计算法向量，并保存合法向量的点云。

### ▼ Advanced

- **ConvertDepthMapToPointCloud**：从深度图生成并保存点云。
- **MultipleCamerasCaptureSequentially**：使用多台相机按序获取并保存2D图、深度图及点云。
- **MultipleCamerasCaptureSimultaneously**：使用多台相机同时获取并保存2D图、深度图及点云。
- **CapturePeriodically**：在设定时间内，定时获取并保存2D图、深度图和点云。
- **Mapping2DImageToDepthMap**：从覆盖掩膜的2D图和深度图生成并保存无纹理点云和纹理点云。
- **SetParametersOfLaserCameras**：设置激光相机特有的参数。
- **SetParametersOfUHPCameras**：设置UHP系列相机特有的参数。
- **RegisterCameraEvent**：定义并注册检测相机连接状态的回调函数。
- **CaptureStereo2DImages**：获取Deep (V3)、Laser L Enhanced (V3)、PRO XS (V4)、LSR L (V4)、LSR S (V4) 和DEEP (V4) 的两个2D相机的2D图像。

### ▼ Util

- **GetCameraIntrinsics**：获取并打印相机内参。
- **PrintCameraInfo**：获取并打印相机型号、序列号、固件版本、温度等信息。
- **SetScanningParameters**：设置**3D参数**、**2D参数**和**感兴趣区域**分组下的参数。



- **SetDepthRange**: 设置深度范围参数。
- **SetPointCloudProcessingParameters**: 设置点云后处理参数。
- **ManageUserSets**: 管理参数组，如获取所有参数组的名称、新增参数组、切换参数组和保存参数设置至参数组。
- **SaveAndLoadUserSet**: 从JSON文件导入并替换所有参数组，将所有参数组保存为JSON文件。

## 使用前提

使用Mech-Eye API的C#例程，需先满足以下使用前提：

- [正确连接相机和电脑](#)。
- 获取例程。
- 安装必需软件。

## 获取例程

C#例程包含在Mech-Eye SDK的安装路径中，也可从[GitHub](#)克隆获取克隆获取。安装路径中的例程为Mech-Eye SDK发布时的版本，GitHub上的例程可能包含最新修改。

- 安装路径中的例程位于xxx/Mech-Eye SDK-2.2.2/API/samples/csharp/area\_scan\_3d\_camera路径下；
- 通过GitHub克隆获取的例程位于xxx/mecheye\_csharp\_samples/area\_scan\_3d\_camera路径下。

## 安装必需软件

使用Mech-Eye API的C#例程，必须安装Mech-Eye SDK、Visual Studio和Emgu CV。

## 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Visual Studio（2019或以上版本）

1. 下载[Visual Studio安装包](#)。
2. 安装时，勾选以下工作负荷和单个组件，再点击[ **安装** ]。
  - **桌面应用和移动应用**分类的工作负荷：
    - **.NET桌面开发**
    - **使用C++的桌面开发**
    - **通用Windows平台开发**

- 单个组件：**.NET Framework 4.8目标包**



C#的Mech-Eye API基于.NET Framework 4.8。如未安装.NET Framework 4.8，将无法构建例程。

## 安装Emgu CV

**CaptureDepthMap**例程中的功能依赖OpenCV软件库，因此需通过Visual Studio的NuGet管理器安装Emgu CV（即OpenCV的.NET封装）。

具体安装方式请参考 [Microsoft提供的指南](#)。

## 构建及运行例程

在Visual Studio中构建例程后，可选择合适的方式运行例程。

### 使用Visual Studio构建例程

1. 双击`area_scan_3d_camera`文件夹中的**MechEyeCSharpSamples.sln**。
2. 在Visual Studio工具栏中，将解决方案配置从**Debug**改为**Release**模式。
3. 在菜单栏中选择生成 > 生成解决方案。每个例程会生成对应的可执行文件（.exe），保存在**Build**文件夹中，该文件夹位于`area_scan_3d_camera`文件夹中。

### 运行例程

你可以在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（index），并按 **Enter** 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。

### 运行例程可执行文件

1. 进入`area_scan_3d_camera`文件夹中的**Build**文件夹。
2. 运行与例程同名的可执行文件。

例程运行过程中，请根据提示输入待连接相机的编号（index），并按 **Enter** 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。

## 6.6.4. Python (Windows)

本章介绍如何在Windows系统中获取并运行Mech-Eye API的Python例程。

### 例程简介

例程分为以下类别：**basic**、**advanced**和**util**。

- **basic**例程：连接和采集数据。
- **advanced**例程：通过复杂、高阶的方式采集数据，设置部分型号特有参数。
- **util**例程：获取相机信息和设置通用参数。

各分类中包含的例程及其简介如下。

#### ▼ basic

- **connect\_to\_camera**：连接相机。
- **connect\_and\_capture\_images**：连接相机并获取2D图、深度图及点云数据。
- **capture\_2d\_image**：从相机获取并保存2D图。
- **capture\_depth\_map**：从相机获取并保存深度图。
- **capture\_point\_cloud**：从相机获取并保存无纹理点云和纹理点云。
- **capture\_point\_cloud\_hdr**：设置多个曝光时间，然后从相机获取并保存点云。
- **capture\_point\_cloud\_with\_normals**：计算法向量，并保存含法向量的点云。

#### ▼ advanced

- **convert\_depth\_map\_to\_point\_cloud**：从深度图生成并保存点云。
- **multiple\_cameras\_capture\_sequentially**：使用多台相机按序获取并保存2D图、深度图及点云。
- **multiple\_cameras\_capture\_simultaneously**：使用多台相机同时获取并保存2D图、深度图及点云。
- **capture\_periodically**：在设定时间内，定时获取并保存2D图、深度图和点云。
- **mapping\_2d\_image\_to\_depth\_map**：从覆盖掩膜的2D图和深度图生成并保存无纹理点云和纹理点云。
- **set\_parameters\_of\_laser\_cameras**：设置激光相机特有的参数。
- **set\_parameters\_of\_uhp\_cameras**：设置UHP系列相机特有的参数。
- **register\_camera\_event**：定义并注册检测相机连接状态的回调函数。
- **capture\_stereo\_2d\_images**：获取Deep (V3)、Laser L Enhanced (V3)、PRO XS (V4)、LSR L (V4)、LSR S (V4) 和DEEP (V4) 的两个2D相机的2D图像。

#### ▼ util

- **get\_camera\_intrinsics**：获取并打印相机内参。

- **print\_camera\_info**: 获取并打印相机型号、序列号、固件版本、温度等信息。
- **set\_scanning\_parameters**: 设置**3D参数**、**2D参数**和**感兴趣区域**分组下的参数。
- **set\_depth\_range**: 设置**深度范围**参数。
- **set\_point\_cloud\_processing\_parameters**: 设置**点云后处理**参数。
- **manage\_user\_sets**: 管理参数组，如获取所有参数组的名称、新增参数组、切换参数组和保存参数设置至参数组。
- **save\_and\_load\_user\_set**: 从JSON文件导入并替换所有参数组，将所有参数组保存为JSON文件。

## 使用前提

使用Mech-Eye API的Python例程，需先满足以下使用前提：

- [正确连接相机和电脑](#)。
- 确保安装的Python为64位，且版本在3.7至3.11之间。



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/`

- 获取例程。
- 安装必需软件。
- （可选）安装例程依赖的软件库。

## 获取例程

Python例程包含在Mech-Eye SDK的安装路径中，也可从[GitHub](#)克隆获取。安装路径中的例程为Mech-Eye SDK发布时的版本，GitHub上的例程可能包含最新修改。

- 安装路径中的例程位于`xxx/Mech-Eye SDK-2.2.2/API/samples/python/area_scan_3d_camera`路径下；
- 通过GitHub克隆获取的例程位于`xxx/mecheye_python_samples/area_scan_3d_camera`路径下。

## 安装必需软件

使用Mech-Eye API的Python例程，必须安装Mech-Eye SDK及Python Mech-Eye API。

## 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。

## 安装最新版本的Python Mech-Eye API

请执行以下步骤安装最新版本的Python Mech-Eye API:

1. 以管理员身份运行Windows PowerShell。
2. 执行以下命令安装Python Mech-Eye API:

```
pip install MechEyeAPI
```

### (可选) 安装依赖的软件库

部分例程中的功能依赖OpenCV。如需使用下表中列出的例程，则必须安装OpenCV（推荐最新版本）。

- capture\_2d\_image
- capture\_depth\_map
- capture\_periodically
- capture\_stereo\_2d\_images
- multiple\_cameras\_capture\_sequentially
- multiple\_cameras\_capture\_simultaneously

请执行以下命令安装最新版本的OpenCV:

```
pip install opencv-python
```

## 运行例程



本节以安装路径中包含的例程为例提供相关路径，如使用从GitHub克隆获取的例程，需对应更改路径。

请执行以下步骤运行例程:

1. 进入例程所在的文件夹。将路径中的**category**替换为示例所在类别的名称。

```
cd xxx/Mech-Eye SDK-2.2.2/API/samples/python/area_scan_3d_camera/category
```

2. 运行例程: 将**sample\_name**替换为示例名称。例程运行过程中, 请根据提示输入待连接相机的编号(index), 并按 **Enter** 键连接相机。

```
python sample_name.py
```

3. 如例程保存了图像或点云, 将在例程所在文件夹中得到保存的文件。

## 6.6.5. Python (Ubuntu)

本章介绍如何在Ubuntu系统中获取并运行Mech-Eye API的Python例程。

### 例程简介

例程分为以下类别：**basic**、**advanced**和**util**。

- **basic**例程：连接和采集数据。
- **advanced**例程：通过复杂、高阶的方式采集数据，设置部分型号特有参数。
- **util**例程：获取相机信息和设置通用参数。

各分类中包含的例程及其简介如下。

#### ▼ basic

- **connect\_to\_camera**：连接相机。
- **connect\_and\_capture\_images**：连接相机并获取2D图、深度图及点云数据。
- **capture\_2d\_image**：从相机获取并保存2D图。
- **capture\_depth\_map**：从相机获取并保存深度图。
- **capture\_point\_cloud**：从相机获取并保存无纹理点云和纹理点云。
- **capture\_point\_cloud\_hdr**：设置多个曝光时间，然后从相机获取并保存点云。
- **capture\_point\_cloud\_with\_normals**：计算法向量，并保存含法向量的点云。

#### ▼ advanced

- **convert\_depth\_map\_to\_point\_cloud**：从深度图生成并保存点云。
- **multiple\_cameras\_capture\_sequentially**：使用多台相机按序获取并保存2D图、深度图及点云。
- **multiple\_cameras\_capture\_simultaneously**：使用多台相机同时获取并保存2D图、深度图及点云。
- **capture\_periodically**：在设定时间内，定时获取并保存2D图、深度图和点云。
- **mapping\_2d\_image\_to\_depth\_map**：从覆盖掩膜的2D图和深度图生成并保存无纹理点云和纹理点云。
- **set\_parameters\_of\_laser\_cameras**：设置激光相机特有的参数。
- **set\_parameters\_of\_uhp\_cameras**：设置UHP系列相机特有的参数。
- **register\_camera\_event**：定义并注册检测相机连接状态的回调函数。
- **capture\_stereo\_2d\_images**：获取Deep (V3)、Laser L Enhanced (V3)、PRO XS (V4)、LSR L (V4)、LSR S (V4) 和DEEP (V4) 的两个2D相机的2D图像。

#### ▼ util

- **get\_camera\_intrinsics**：获取并打印相机内参。

- **print\_camera\_info**: 获取并打印相机型号、序列号、固件版本、温度等信息。
- **set\_scanning\_parameters**: 设置**3D参数**、**2D参数**和**感兴趣区域**分组下的参数。
- **set\_depth\_range**: 设置**深度范围**参数。
- **set\_point\_cloud\_processing\_parameters**: 设置**点云后处理**参数。
- **manage\_user\_sets**: 管理参数组，如获取所有参数组的名称、新增参数组、切换参数组和保存参数设置至参数组。
- **save\_and\_load\_user\_set**: 从JSON文件导入并替换所有参数组，将所有参数组保存为JSON文件。

## 使用前提

使用Mech-Eye API的Python例程，需先满足以下使用前提：

- [正确连接相机和电脑](#)。
- 确保安装的Python为64位，且版本在3.7至3.11之间。
- 获取例程。
- 安装必需软件。
- （可选）安装例程依赖的软件库。

## 获取例程

Python例程包含在Mech-Eye SDK的安装路径中，也可从[GitHub](#)克隆获取。安装路径中的例程为Mech-Eye SDK发布时的版本，GitHub上的例程可能包含最新修改。

- 安装路径中的例程位于`/opt/mech-mind/mech-eye-sdk/samples/python/area_scan_3d_camera`路径下；
- 通过GitHub克隆获取的例程位于`xxx/mecheye_python_samples/area_scan_3d_camera`路径下。

执行以下指令从GitHub克隆Python例程：

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

## 安装必需软件

使用Mech-Eye API的Python例程，必须安装依赖库、Mech-Eye SDK及Python Mech-Eye API。

1. 安装Mech-Eye SDK：详见[Mech-Eye SDK安装指南（Ubuntu）](#)。
2. 安装pip3或升级pip3，保证其版本为20.3.1或以上。

#### ▼ 操作步骤

- 执行以下指令查看pip3是否安装及其版本：

```
pip3 -V
```

- 执行以下指令安装pip3：

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

- 执行以下指令升级pip3：

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

### 3. 升级g++，保证选中的版本为12或以上。



如g++版本低于12，将导致import失败。

#### ▼ 操作步骤

1. 执行以下指令安装新版本g++（以g++ 13为例）：

```
sudo add-apt-repository ppa:ubuntu-toolchain-r/test  
sudo apt-get update  
sudo apt install g++-13
```

2. 通过ls指令查看已安装的g++的版本：

```
ls usr/bin/g++*
```

3. 执行以下指令将各版本的g++添加为alternative（以g++ 9和g++ 13为例）：



命令末尾的数字为优先级。数字越大，优先级越高。

```
sudo update-alternatives --install /usr/bin/g++ g++ /usr/bin/g++-9 10  
sudo update-alternatives --install /usr/bin/g++ g++ /usr/bin/g++-13 20
```

4. 执行以下指令，选择g++版本。输入新版本g++对应的数字以选中该版本。

```
sudo update-alternatives --config g++
```

5. 执行以下指令，确认是否成功选中新版本g++：



```
g++ --version
```

#### 4. 安装Python Mech-Eye API。

```
sudo pip3 install MechEyeApi
```

#### (可选) 安装依赖的软件库

部分例程中的功能依赖OpenCV。如需使用下表中列出的例程，则必须安装OpenCV（推荐最新版本）。

- capture\_2d\_image
- capture\_depth\_map
- capture\_periodically
- capture\_stereo\_2d\_images
- multiple\_cameras\_capture\_sequentially
- multiple\_cameras\_capture\_simultaneously

请执行以下命令安装最新版本的OpenCV:

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

### 运行例程



本节以安装路径中包含的例程为例提供相关路径，如使用从GitHub克隆获取的例程，需对应更改路径。

请执行以下步骤运行例程：

1. 进入例程所在的文件夹。将路径中的**category**替换为例程所在类别的名称。

```
cd ~/opt/mech-mind/mech-eye-sdk/samples/python/area_scan_3d_camera/category
```

2. 运行例程：将**sample\_name**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（index），并按 **Enter** 键连接相机。

```
sudo python3 sample_name.py
```

3. 如例程保存了图像或点云，将在例程所在文件夹中得到保存的文件。



自行编写程序时，请先使用import指令导入Mech-Eye API：

```
python3
import MechEye
import open3d
```

### 6.6.6. VisionPro

本章介绍如何使用梅卡曼德提供的VisionPro例程通过C# Mech-Eye API连接相机并获取2D和3D数据。

#### 例程简介

目前提供以下例程：

**AcquireColorAndRangeImages**：获取2D和深度数据，并生成康耐视格式的2D图（CogImage24PlanarColor）和深度图（CogImage16Range）。

#### 使用前提

使用VisionPro例程，需先满足以下使用前提：

- [正确连接相机和电脑](#)。
- 获取例程。
- 安装必需软件。

#### 获取例程

VisionPro例程包含在Mech-Eye SDK的安装路径中，也可从[GitHub](#)克隆获取。安装路径中的例程为Mech-Eye SDK发布时的版本，GitHub上的例程可能包含最新修改。

- 安装路径中的例程位于xxx/Mech-Eye SDK-2.2.2/API/samples/visionpro/area\_scan\_3d\_camera路径下；
- 通过GitHub克隆获取的例程位于xxx/mecheye\_visionpro\_samples/area\_scan\_3d\_camera路径下。

#### 安装必需软件

使用VisionPro例程，必须安装Mech-Eye SDK及VisionPro。

- Mech-Eye SDK：请根据[Mech-Eye SDK安装指南](#)安装最新版本或升级。
- 康耐视VisionPro：9.8 SR1版本经过测试。

#### 配置及运行例程

完成必要的配置后，即可运行例程。

## 复制动态链接库

在VisionPro软件中调用Mech-Eye API，需先将Mech-Eye SDK的动态链接库复制至VisionPro的安装路径下。

请执行以下步骤复制动态链接库：

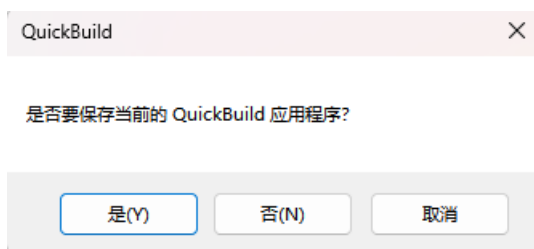
1. 进入Mech-Eye SDK的安装路径并打开以下文件夹：*xxx/Mech-Eye SDK-2.2.2/API/dll*。
2. 复制上述文件夹中的以下动态链接库文件（.dll）：
  - **MechEyeApi**
  - **MechEyeApiNet**
  - **MechEyeApiWrapper**
3. 进入VisionPro的安装路径，打开其中的**bin**文件夹，并粘贴复制的动态链接库文件。

## 配置脚本参考程序集

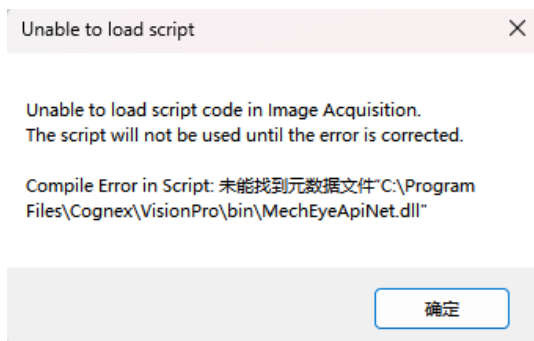
复制动态链接库后，需在例程中配置脚本参考程序集的路径。

请执行以下步骤配置脚本参考程序集：



1. 打开VisionPro QuickBuild，选择**文件**菜单中的**打开 QuickBuild 应用程序**。
2. 在弹出的窗口中，选择需使用的VisionPro例程，并单击[ **打开** ]按钮。
3. 将弹出以下窗口。单击窗口中的[ **否** ]按钮关闭该弹窗。



4. 可能弹出以下窗口。如弹出，单击窗口中的[ **确定** ]按钮关闭该弹窗。



5. 在VisionPro QuickBuild的左边栏中，双击**Mech-Eye Industrial 3D Camera**。
6. 在弹出的作业编辑器窗口的左边栏中，双击**Image Acquisition**。

7. 在弹出的**Image Acquisition**窗口的工具栏中，单击  按钮。
8. 在弹出的**Image Acquisition 脚本**窗口的工具栏中，单击  按钮。
9. 在弹出的**添加/移除参考程序集**窗口中，滚动至最下并双击**MechEyeApiNet.dll**条目。
10. 在弹出的**输入程序集名称**窗口中，单击[浏览...]按钮，导航至Mech-Eye SDK安装路径下动态链接库所在的文件夹 (`xxx/Mech-Eye SDK-2.2.2/API/dll`)，并选择**MechEyeApiNet**动态链接库文件。
11. 单击**输入程序集名称**窗口中的[确定]按钮关闭该窗口，然后单击**添加/移除参考程序集**窗口中的[确定]按钮关闭该窗口。

## 设置相机IP地址

运行例程前，需设置脚本中相机的IP地址，以便成功连接相机。



可通过[Mech-Eye Viewer](#)或[IP配置工具](#)查看并更改相机的IP地址。

请执行以下步骤设置相机的IP地址：


1. 在**Image Acquisition 脚本**窗口中，找到以下行，并将其中的IP地址更改为待连接相机的实际IP地址。

```
status = camera.Connect("192.168.20.15", 10000)
```

2. 单击工具栏中的  按钮使修改生效。
3. 关闭**Image Acquisition 脚本**和**Image Acquisition**窗口，返回至作业编辑器窗口。

## 运行例程

请执行以下步骤运行例程并查看获取的2D图和深度图：

1. 在作业编辑器窗口的工具栏中，单击  按钮运行例程。
2. 在左边栏中，双击对应的工具查看获取的2D图或深度图：
  - 2D图：双击**CogAffineTransformTool1**。
  - 深度图：双击**Cog3DVisionDataReRenderTool1**。
3. 如果获取的图像不满足要求，请根据下一节的内容调节相机的参数，以提升数据质量。

## 调节相机参数

相机的参数可通过Mech-Eye Viewer调节，或在例程中通过C# Mech-Eye API调节。推荐使用Mech-Eye Viewer调节参数，可在图形化界面中即时采集数据确认参数调节的效果。

- 使用Mech-Eye Viewer调节参数：请参考[操作指南](#)在Mech-Eye Viewer中连接相机、采集数据、查看当前参数值并调节参数。

- 在例程中通过C# Mech-Eye API调节参数：使用例程调节相机参数，需在例程脚本中添加对应的C# Mech-Eye API方法。请参考以下内容编辑例程脚本。
  - C# Mech-Eye API例程：例程简介、获取方式和使用指南请参考[C#](#)。
  - [Mech-Eye API参考手册](#)：提供C++ Mech-Eye API方法的说明。

## 6.7. 工具

### 6.7.1. 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，简化固件升级的流程。

固件升级工具可从[梅卡曼德下载中心](#)获取。

固件升级工具包含两部分：名为**firmware\_upgrader**的可执行文件和名为**mmind\_eye\_v2.2.2**的压缩文件。

请执行以下步骤升级固件：

1. 执行以下命令，为固件升级工具增加可执行权限。

```
chmod 777 xxx/firmware_upgrader
```

2. 执行以下命令，进行固件升级。

```
sudo xxx/firmware_upgrader xxx/mmind_eye_v2.2.2.zip
```



- 应先输入**firmware\_upgrader**的路径，再输入**mmind\_eye\_v2.2.2.zip**的路径，否则固件升级工具无法运行。
- 两个路径之间需加空格，否则固件升级工具无法运行。

3. 根据提示输入需升级固件的相机编号（index），并按 **Enter** 键连接相机。如显示**Upgrade Completed**，固件升级成功。



如固件升级失败，请重新尝试或联系技术支持。

### 6.7.2. IP配置工具

使用该工具可直接在Ubuntu系统中更改相机的IP配置，简化IP配置的流程。

#### 使用前提

使用IP配置工具，需先满足以下使用前提：

[下载并安装2.2.0或以上版本的Mech-Eye SDK](#)，以获取IP配置工具。

IP配置工具名为`ip_configurator`的可执行文件，位于`/opt/mech-mind/mech-eye-sdk/bin`路径下。

IP配置工具独立于Mech-Eye SDK。你可以将该可执行文件移动至其他位置保存并单独运行。

## 操作步骤

请执行以下步骤更改相机的IP配置：

1. 执行以下命令，运行IP配置工具。请将路径替换为`ip_configurator`可执行文件实际所在的路径。

```
sudo /opt/mech-mind/mech-eye-sdk/bin/ip_configurator
```

2. 根据提示输入需更改IP配置的相机编号（index），并按 `Enter` 键连接相机。
3. 根据提示选择IP地址分配方式，`0`为动态分配IP地址，`1`为设置静态IP地址。
  - 如选择动态分配IP地址，请执行步骤6。
  - 如选择设置静态IP地址，请执行下一步。
4. 输入需设置的IP地址，并按 `Enter` 键。
5. 输入需设置的子网掩码，并按 `Enter` 键。如需将子网掩码设置为`255.255.255.0`，输入`0`后按 `Enter` 键即可。
6. 相机的新旧IP配置将打印在窗口中，请检查新的IP配置是否正确：
  - 如确认无误，请输入`1`并按 `Enter` 键，IP配置更改完成。
  - 如有误，请输入`0`并按 `Enter` 键。程序将重新列出所有可连接的相机，请返回步骤2重新操作。

## 6.8. 向下兼容性相关

### 6.8.1. 2.2.0版本的向下兼容性

Mech-Eye API 2.2.0进行了重构，在保留原有的全部功能的基础上，调整为更加清晰的结构，并提供新增功能。

如需将相机固件版本升级至2.2.0，但需继续在客户端程序中使用2.1.0版本的Mech-Eye API，请参考本章完成所用语言对应的处理。



如希望在已有客户端程序中使用2.2.0版本的Mech-Eye API，请参考[迁移指南](#)修改客户端程序。

### C++

如需继续在客户端程序中使用2.1.0版本的C++ Mech-Eye API，请根据所使用的操作系统完成对应的处理。

## Windows系统

Mech-Eye SDK 2.2.0包含了2.1.0版本的C++ Mech-Eye API。

请[升级Mech-Eye SDK](#)，并通过Mech-Eye Viewer 2.2.0升级相机固件。升级后，请重新编译客户端程序。

## Ubuntu系统

Mech-Eye SDK 2.2.0包含了2.1.0版本的C++ Mech-Eye API。

请[升级Mech-Eye SDK](#)，并通过[固件升级工具](#)升级相机固件。升级后，请重新编译客户端程序。

## C#

如需继续在客户端程序中使用2.1.0版本的C# Mech-Eye API，请注意以下事项：

- [升级Mech-Eye SDK](#)时，请选择[ **升级，保留历史版本** ]，并取消勾选**添加至环境变量**。
- 请通过Mech-Eye Viewer 2.2.0升级相机固件。

## Python

如需继续在客户端程序中使用2.1.0版本的Python Mech-Eye API，请根据所使用的操作系统完成对应的处理。

## Windows系统

如使用的是Windows系统，请注意以下事项：

- 请勿通过pip安装最新版本的Python Mech-Eye API。
- [升级Mech-Eye SDK](#)时，请选择[ **升级，保留历史版本** ]，并取消勾选**添加至环境变量**。
- 请通过Mech-Eye Viewer 2.2.0升级相机固件。

## Ubuntu系统

如使用的是Ubuntu系统，请注意以下事项：

- 请勿通过pip3安装最新版本的Python Mech-Eye API。
- 请通过[固件升级工具](#)升级相机固件。

### 6.8.2. C++参数名称更新

Mech-Eye API 2.0.0的C++接口更新了参数名称。如果你已经安装Mech-Eye SDK 2.0.0，但是仍需使用之前编写的程序，请参照下表更新程序中的参数名称。

2.0.0之前	2.0.0
Scanning3DSettings::ExposureSequence	Scanning3DSettings::exposureSequence

2.0.0之前	2.0.0
Scanning3DSettings::Gain	Scanning3DSettings::gain
Scanning3DSettings::Scan3DROI	Scanning3DSettings::scan3DROI
Scanning3DSettings::DepthRange	Scanning3DSettings::depthRange
Scanning2DSettings::Scan2DExposureMode::ExposureMode	Scanning2DSettings::Scan2DExposureMode::exposureMode
Scanning2DSettings::ExposureTime	Scanning2DSettings::exposureTime
Scanning2DSettings::SharpenFactor	Scanning2DSettings::sharpenFactor
Scanning2DSettings::ExpectedGrayValue	Scanning2DSettings::expectedGrayValue
Scanning2DSettings::Scan2DROI	Scanning2DSettings::scan2DROI
Scanning2DSettings::ToneMappingEnable	Scanning2DSettings::toneMappingEnable
Scanning2DSettings::HDRExposureSequence	Scanning2DSettings::hdrExposureSequence
PointCloudProcessingSettings::FringeContrastThreshold	PointCloudProcessingSettings::fringeContrastThreshold
PointCloudProcessingSettings::FringeMinThreshold	PointCloudProcessingSettings::fringeMinThreshold
PointCloudProcessingSettings::CloudOutlierFilterMode::OutlierFilterMode	PointCloudProcessingSettings::CloudOutlierFilterMode::outlierFilterMode
PointCloudProcessingSettings::CloudSmoothMode::SmoothMode	PointCloudProcessingSettings::CloudSmoothMode::smoothMode
LaserSettings::LaserFringeCodingMode::FringeCodingMode	LaserSettings::LaserFringeCodingMode::fringeCodingMode
LaserSettings::FrameRangeStart	LaserSettings::frameRangeStart
LaserSettings::FrameRangeEnd	LaserSettings::frameRangeEnd
LaserSettings::FramePartitionCount	LaserSettings::framePartitionCount
LaserSettings::PowerLevel	LaserSettings::powerLevel
UhpSettings::UhpCaptureMode::CaptureMode	UhpSettings::UhpCaptureMode::captureMode
UhpSettings::UhpFringeCodingMode::FringeCodingMode	UhpSettings::UhpFringeCodingMode::fringeCodingMode

### 6.8.3. C#方法名称与参数名称更新

Mech-Eye API 2.0.0 的C#接口更新了方法与参数的名称。如果你已经安装Mech-Eye SDK 2.0.0，但是仍需使用之前编写的程序，请参照下表更新程序中的方法与参数名称。

#### 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()



2.0.0之前	2.0.0
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)

2.0.0之前	2.0.0
MechEyeDevice.getScan2DHDRExposureSequence(System.Collections.Generic.List<double>)	MechEyeDevice.GetScan2DHDRExposureSequence(System.Collections.Generic.List<double>)
MechEyeDevice.getScan2DROI(mmMind.ApiSharp.ROI)	MechEyeDevice.GetScan2DROI(mmMind.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(mmMind.ApiSharp.ROI)	MechEyeDevice.GetScan3DROI(mmMind.ApiSharp.ROI)
MechEyeDevice.getUhpCaptureMode(mmMind.ApiSharp.UhpCaptureMode)	MechEyeDevice.GetUhpCaptureMode(mmMind.ApiSharp.UhpCaptureMode)
MechEyeDevice.getUhpFringeCodingMode(mmMind.ApiSharp.UhpFringeCodingMode)	MechEyeDevice.GetUhpFringeCodingMode(mmMind.ApiSharp.UhpFringeCodingMode)
MechEyeDevice.getUhpSettings(mmMind.ApiSharp.UhpSettings)	MechEyeDevice.GetUhpSettings(mmMind.ApiSharp.UhpSettings)
MechEyeDevice.saveAllSettingsToUserSets()	MechEyeDevice.SaveAllSettingsToUserSets()
MechEyeDevice.setCloudOutlierFilterMode(mmMind.ApiSharp.CloudOutlierFilterMode)	MechEyeDevice.SetCloudOutlierFilterMode(mmMind.ApiSharp.CloudOutlierFilterMode)
MechEyeDevice.setCloudSmoothMode(mmMind.ApiSharp.CloudSmoothMode)	MechEyeDevice.SetCloudSmoothMode(mmMind.ApiSharp.CloudSmoothMode)
MechEyeDevice.setCurrentUserSet(string)	MechEyeDevice.SetCurrentUserSet(string)
MechEyeDevice.setDepthRange(mmMind.ApiSharp.DepthRange)	MechEyeDevice.SetDepthRange(mmMind.ApiSharp.DepthRange)
MechEyeDevice.setFringeContrastThreshold(int)	MechEyeDevice.SetFringeContrastThreshold(int)
MechEyeDevice.setFringeMinThreshold(int)	MechEyeDevice.SetFringeMinThreshold(int)
MechEyeDevice.setLaserSettings(mmMind.ApiSharp.LaserSettings)	MechEyeDevice.SetLaserSettings(mmMind.ApiSharp.LaserSettings)
MechEyeDevice.setScan2DExpectedGrayValue(int)	MechEyeDevice.SetScan2DExpectedGrayValue(int)
MechEyeDevice.setScan2DExposureMode(mmMind.ApiSharp.Scan2DExposureMode)	MechEyeDevice.SetScan2DExposureMode(mmMind.ApiSharp.Scan2DExposureMode)
MechEyeDevice.setScan2DExposureTime(double)	MechEyeDevice.SetScan2DExposureTime(double)
MechEyeDevice.setScan2DHDRExposureSequence(System.Collections.Generic.List<double>)	MechEyeDevice.SetScan2DHDRExposureSequence(System.Collections.Generic.List<double>)
MechEyeDevice.setScan2DROI(mmMind.ApiSharp.ROI)	MechEyeDevice.SetScan2DROI(mmMind.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>)

2.0.0之前	2.0.0
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()

### 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

2.0.0之前	2.0.0
Scanning3DSettings.ExposureSequence	Scanning3DSettings.exposureSequence
Scanning3DSettings.Gain	Scanning3DSettings.gain
Scanning3DSettings.Scan3DROI	Scanning3DSettings.scan3DROI
UhpSettings.CaptureMode	UhpSettings.captureMode
UhpSettings.FringeCodingMode	UhpSettings.fringeCodingMode

## 7. GenICam接口

### 7.1. 什么是GenICam?

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（如HALCON）便捷地连接和控制相机，而无需对相机进行特定的配置。

#### 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：[GenICam](#)。

#### 相机对GenICam标准的支持

将相机固件升级至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[HALCON——连接相机、调节参数、采集图像](#)。

### 7.2. HALCON

使用梅卡曼德提供的HALCON例程，你可以完成连接相机、采集数据、手眼标定等多种流程。

HALCON例程包含在Mech-Eye SDK的安装路径中，也可从[GitHub](#)克隆获取。安装路径中的例程为Mech-Eye SDK发布时的版本，GitHub上的例程可能包含最新修改。

- 安装路径中的例程位于xxx/Mech-Eye SDK-2.2.2/API/samples/halcon/area\_scan\_3d\_camera路径下；
- 通过GitHub克隆获取的例程位于xxx/mecheye\_halcon\_samples/area\_scan\_3d\_camera路

径下。

以下章节提供各个例程的使用指导：

- [HALCON](#)
- [HALCON——手眼标定](#)
- [HALCON——获取纹理点云](#)
- [HALCON——仅获取Z值](#)
- [HALCON——设置IP地址](#)
- 获取合法向量的点云（敬请期待）







如使用相机时遇到问题，请访问[梅卡曼德在线社区](#)获取帮助。

### 7.2.1. HALCON——连接相机、调节参数、采集图像

本章介绍使用HALCON助手和梅卡曼德提供的HALCON例程连接Mech-Eye工业级3D相机、调节参数并采集图像。

#### 使用前提

- [相机和电脑已正确连接。](#)
  -  Nano (V3) 和Pro XS (V3) 相机推荐与电脑直接连接，不使用交换机。
- 电脑上已安装[HALCON 20.11](#)或以上。
  -  低于20.11版本的HALCON未经完全验证。
- 电脑上已安装Mech-Eye SDK 2.0.0或以上。
  -  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](#)。
- 相机与电脑IP地址在同一网段。
  -  建议将相机与电脑设置为静态IP地址，IP地址设置请参考[使用Mech-Eye Viewer设置相机IP地址](#)。

在HALCON中，你可以通过HALCON助手或梅卡曼德提供的HALCON例程连接相机。

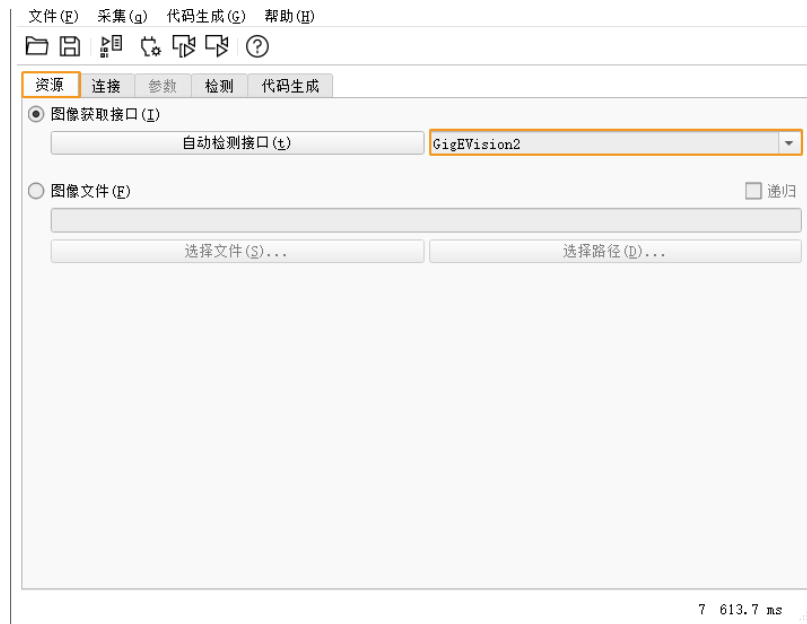
#### 使用HALCON助手

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。

在电脑上运行HALCON软件，选择助手 > 打开新的 Image Acquisition，即可打开图像采集助手。

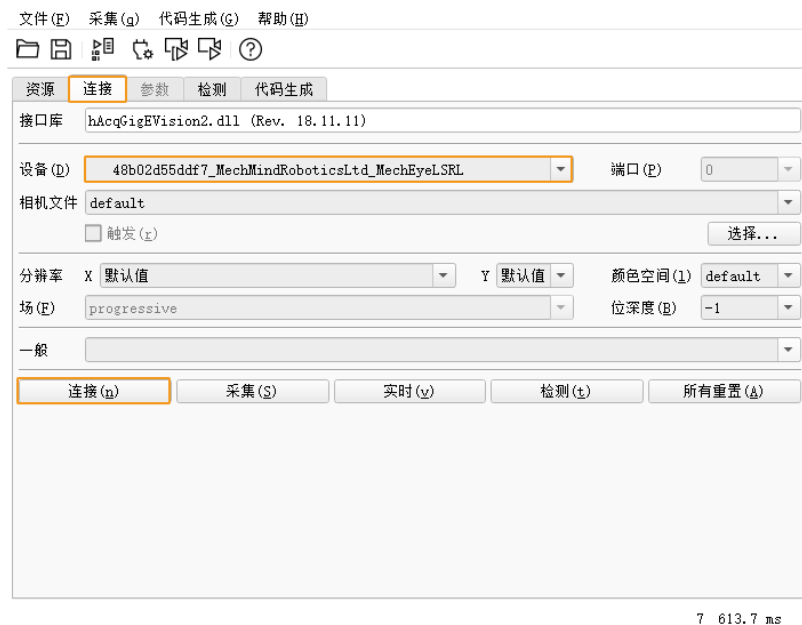
## 连接相机

1. 在Image Acquisition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。



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。



- 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，单击[断开]。
- 如相机已被其他客户端连接，则连接一定不成功。请先在其他客户端中断开连接，再使用HALCON连接。

- 设备中的相机名称可通过Mech-Eye Viewer自定义，便于查找需要连接的相机。在Mech-Eye Viewer中连接相机后，[自定义相机备注](#)即可。推荐设置为英语名称。

## 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。



如相机采图时间较长，可调大相机的MTU值，并开启电脑上的[巨型帧功能](#)。

你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：
  - 切换至[参数](#)选项卡，将**AcquisitionMode**参数设置为**MultiFrame**。
  - 单击右上角的[ **刷新** ]后，在**AcquisitionFrameCount**中设置需进行的图像采集次数。
  - 切换至[连接](#)选项卡，单击[ **实时** ]进行图像采集。
  - 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：
  - 切换至[参数](#)选项卡，将**AcquisitionMode**参数设置为**Continuous**。
  - 切换至[连接](#)选项卡，单击[ **实时** ]进行图像采集。
  - [ **实时** ]按钮变为[ **停止** ]。单击[ **停止** ]停止图像采集。



- **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。

## 选择数据类型

相机连接后，默认采集的数据类型是2D图。你可通过调节**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。



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。

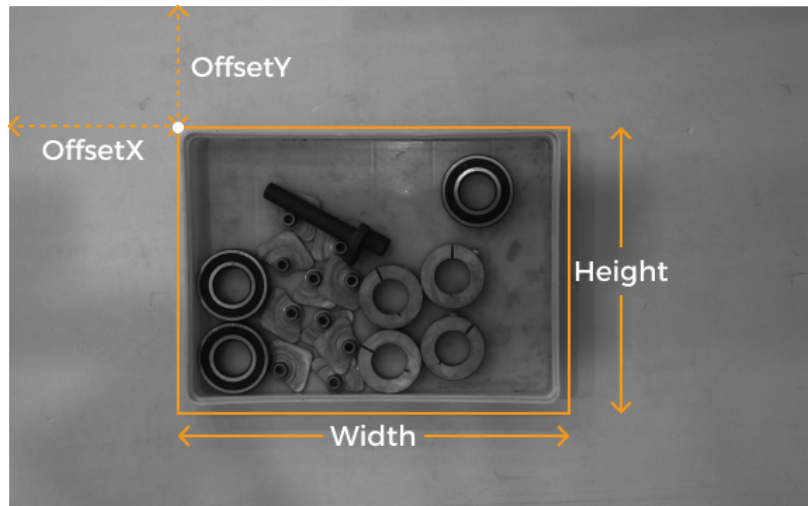
## 设置采集区域

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



请按照以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

以上四个参数的设置需满足以下要求：

- $(Width + OffsetX)$  不超过原始图像的宽度
- $(Height + OffsetY)$  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。



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。

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



- 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 <a href="#">Mech-Eye Viewer中的可视化工具设置</a>

### 调节参数

如获取的数据不满足需求，可在**参数**选项卡下调节相机参数。



- 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。需先选择用于保存参数值的参数组，再调节参数。
- 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助手的**参数**选项卡下，单击右上角的[刷新]按钮读取相机最新配置。

执行以下步骤调节参数：

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。



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

- 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，以及GenICam支持的相机参数的描述和解释，请参考[GenICam支持的相机参数](#)。
- 设置自动曝光感兴趣区域、深度范围及感兴趣区域等相机参数时，通常需要借助可视化工具不断调节从而获取理想的值。由于HALCON并未提供辅助参数设置的可视化工具，你可以借助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](#)。



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。

### 使用HALCON例程


GitHub上提供了一系列[HALCON例程](#)。本章介绍如何使用其中的connect\_to\_camera\_and\_capture\_images例程连接相机、采集数据、修改参数并保存数据。你可以直接使用例程或者根据实际需要进行修改以满足项目需求。



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。

### 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_MGE134GN2   user_name:P022206030802H   interface:Esen_ITF_b07b2528849ac0a81425fffff00   producer:Esen'




user\_name为自定义的相机备注。可在Mech-Eye Viewer中设置。

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

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

- 单击工具栏中的  按钮或单击键盘上的 **F5**，运行例程。
- 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**。



- 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。
- Canvas**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，可调大相机的MTU值，并开启电脑上的**巨型帧功能**。
- 网络较差导致数据丢失时，可以使用以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)
```

## 调节参数

如获取的数据不满足需求，可调节相应参数。



- 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。

执行以下步骤调节参数：

- 使用以下算子获取相机的参数组列表。

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

- 在**控制变量**区域的**ParameterValues**中，查看所有的参数组名称。
- 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')
```

- 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，可以根据需要更改。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')
```

### 参考信息

- 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，以及GenICam支持的相机参数的描述和解释，请参考[GenICam支持的相机参数](#)。
- 设置**自动曝光感兴趣区域**、**深度范围**及**感兴趣区域**等相机参数时，通常需要借助可视化工具不断调节从而获取理想的值。由于HALCON并未提供辅助参数设置的可视化工具，你可以借助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](#)。



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。

### 7.2.2. HALCON——手眼标定

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**：用于获取进行手眼标定时时的标定位姿。
- **perform\_hand\_eye\_calibration**：用于进行手眼标定。



- 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

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hand\_eye\_calibration例程，依次单击Code > Download ZIP即可下载。

## 修改例程

使用determine\_calibration\_poses获取标定位姿后，需使用perform\_hand\_eye\_calibration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，获取位姿前，需设置robot\_pose.json文件中的欧拉角类型。

## 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_MGE134GN2   user_name:P022206030802H   interface:Esen_ITF_b07b2528849ac0a81425fffff00   producer:Esen'



user\_name为自定义的相机备注。可在Mech-Eye Viewer中设置。

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

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

## 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')
```

### 设置欧拉角类型和单位

使用**determine\_calibration\_poses**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"
```

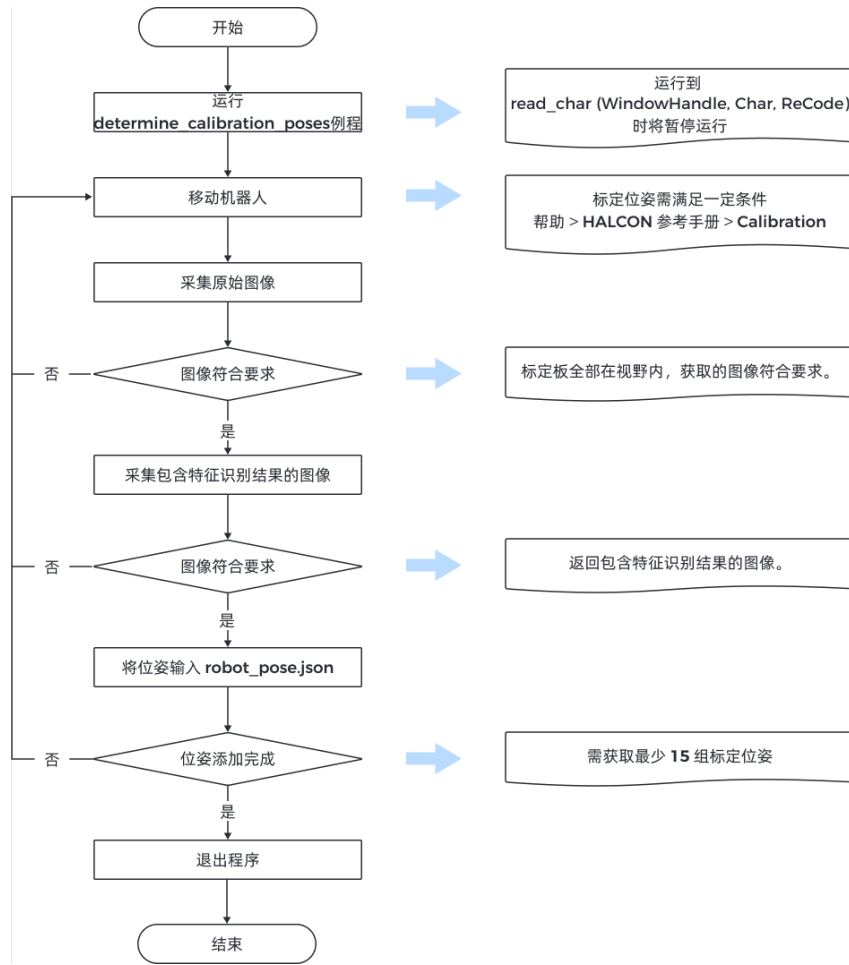
3. 设置欧拉角单位：如需输入以弧度为单位的欧拉角，定位至如下命令行，将**true**替换为**false**。

```
"FromDegree": true
```

4. 保存**robot\_pose.json**文件。


### 获取标定位姿

#### 流程图



## 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。



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。



标定位姿需满足一定条件，相关指导可从HALCON参考手册中获取：打开HALCON的帮助 > HALCON参考手册，查看 **Calibration** 章节。

3. 输入 **P**，相机采集图像。

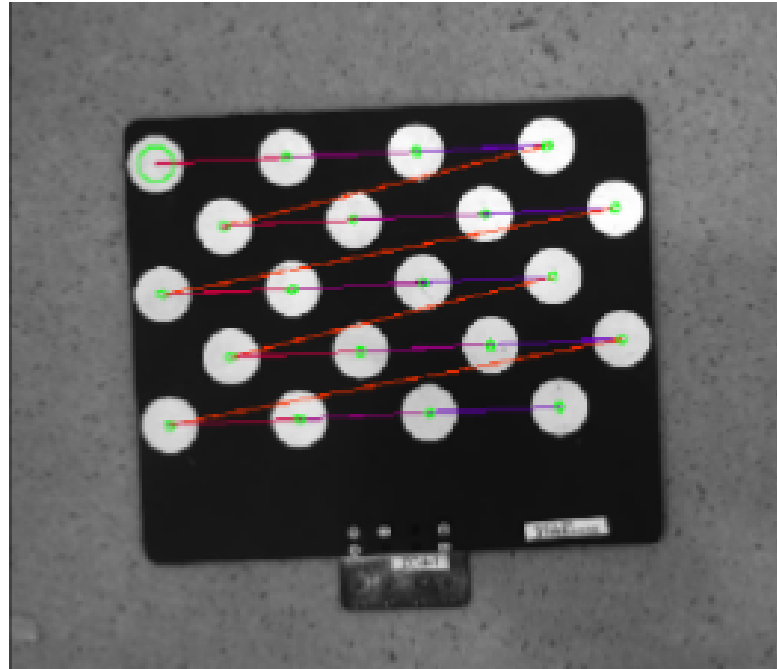
- 如标定板只有部分在视野之内或不在视野之内，请移动机器人，并重新采集图像。
- 如标定板全部在视野之内，请执行下一步。

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。位姿中的平移量应以mm为单位，旋转量（欧拉角）的单位与`robot_pose.json`文件中的设置保持一致。



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。

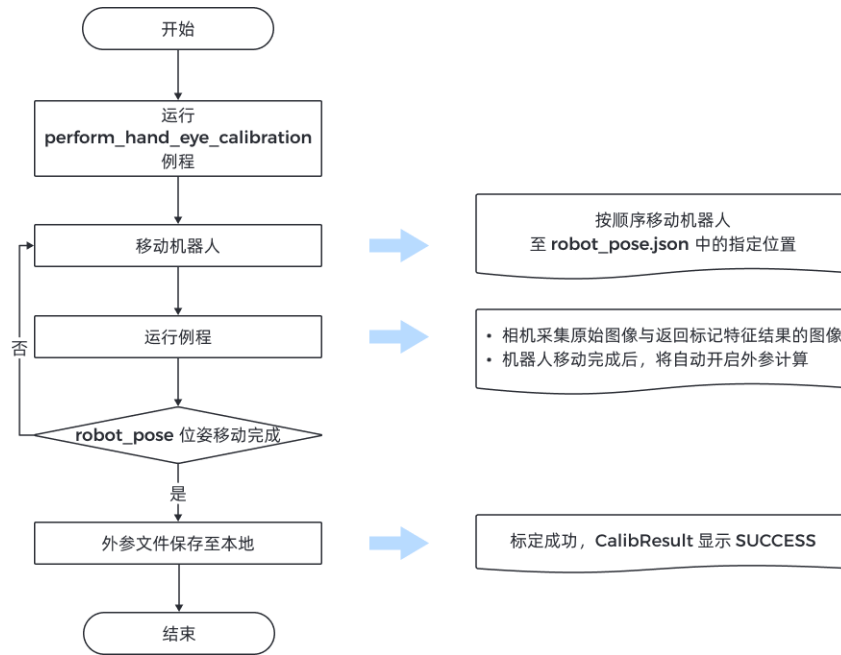
6. 重复步骤2~5，继续获取位姿。
7. 确认获取至少15组位姿后，采集数据结束，输入`Q`，退出程序。
8. 设置`robot_pose.json`文件中标定位姿个数：打开`robot_pose.json`文件，定位至如下命令行，将`15`替换为实际获取的标定位姿个数。

```
"pose_count":15
```

## 进行手眼标定

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

## 流程图



## 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')
```

## 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，通过该算子可使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)
```

## 步骤

手眼标定需执行以下步骤：

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。



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。



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与获取的点云；
- 如发生报错，请根据错误码排查问题，再重新获取标定位姿。

## 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。

常见表示方法	欧拉角类型	机器人品牌
Z-Y'-Z"/OAT	rzyz	Kawasaki (川崎)
Z-Y'-X"/yaw, pitch, roll	rzyx	ABB
X-Y-Z/WPR	sxyz	KUKA (库卡)
		FANUC (发那科)
		YASKAWA (安川)
		Rokae (珞石)
X-Y'-Z"	rxyz	/
Z-X'-Z"	rxz	/



- 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，需要你自行添加该欧拉角类型与四元数的转换。请在perform\_hand\_eye\_calibration例程的euler\_to\_quad函数中，参照已有代码添加转换。

## 标定参数说明

本节介绍进行手眼标定时使用的参数。

### 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

### 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图相关参数</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="#">深度图相关参数</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。

### 7.2.3. HALCON——获取纹理点云

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

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。



如需为两通道中的图像设置采集区域（**Width**、**Height**、**OffsetX**和**OffsetY**参数），需保障参数一致，否则将导致两通道中的图像像素不对齐，无法生成纹理点云。

## 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_b07b2528849ac0a81425fffff00   producer:Esen'



**user\_name**为自定义的相机备注。可在[Mech-Eye Viewer](#)中设置。

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

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

7. **Canvas**窗口中将显示获取的纹理点云，单击该窗口中的[ **Continue** ]继续执行程序。
8. 在**ObjectModel3D**变量中查看获取的纹理点云。



DEEP (V4) 与LSR (V4) 系列默认使用2D图（纹理）生成纹理点云。如需使用2D图（深度源），请定位至如下命令行，并取消该行注释。


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

### 7.2.4. HALCON——仅获取Z值

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

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'



**user\_name**为自定义的相机备注。可在[Mech-Eye Viewer](#)中设置。

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

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


7. 在**DepthInM**变量中查看获取的深度图。

## 7.2.5. HALCON——设置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

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'




**user\_name**为自定义的相机备注。可在[Mech-Eye Viewer](#)中设置。

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

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

## 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




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。

## 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. 单击工具栏中的  重置程序执行。
2. 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'
```


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

## 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. 单击工具栏中的  重置程序执行。
2. 定位至如下命令行，将**true**替换为**false**。

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

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



相机与电脑直接连接时，不可设置为动态分配IP地址。

## 7.3. 参考手册

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](#)

### 7.3.1. 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。

#### 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你在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	



GenICam支持的参数	Mech-Eye Viewer参数	可见级别	备注
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	大师	
Scan2DPatternRoleExposureMode	2D参数 • 2D图（深度源）曝光模式	初级	LSR L (V4)
Scan2DPatternRoleExposureTime	2D参数 • 2D图（深度源）曝光时间	初级	DEEP (V4)
Scan3DROIHeight	感兴趣区域	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

GenICam支持的参数	Mech-Eye Viewer参数	可见级别	备注
ProjectorFringeCodingMode	3D参数 • 投影 <ul style="list-style-type: none"> <li>◦ 编码模式</li> </ul>	专家	Nano (V3) Pro XS (V3) PRO M和PRO S (V4) UHP-140
AntiFlickerMode	3D参数 • 投影 <ul style="list-style-type: none"> <li>◦ 抗频闪模式</li> </ul>	专家	PRO M和PRO S (V4) NANO (V4)
ProjectorSelectionMode	3D参数 • 投影 <ul style="list-style-type: none"> <li>◦ 投影机模式</li> </ul>	大师	Deep (V3)
ProjectorSelector	3D参数 • 投影 <ul style="list-style-type: none"> <li>◦ 选择投影机</li> </ul>	大师	Pro L Enhanced (V3)
LaserFringeCodingMode	3D参数 • 激光 <ul style="list-style-type: none"> <li>◦ 编码模式</li> </ul>	专家	
LaserPowerLevel	3D参数 • 激光 <ul style="list-style-type: none"> <li>◦ 激光强度</li> </ul>	专家	Laser L (V3)
LaserFrameAmplitude	3D参数 • 激光投影画幅控制 <ul style="list-style-type: none"> <li>◦ 激光投影范围幅值</li> </ul>	大师	Laser L Enhanced (V3) LSR L (V4)
LaserFrameOffset	3D参数 • 激光投影画幅控制 <ul style="list-style-type: none"> <li>◦ 激光投影范围偏移值</li> </ul>	大师	DEEP (V4)
LaserFramePartitionCount	3D参数 • 激光投影画幅控制 <ul style="list-style-type: none"> <li>◦ 激光投影分块数</li> </ul>	大师	

GenICam支持的参数	Mech-Eye Viewer参数	可见级别	备注
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	
UhpCaptureMode	3D参数 • UHP ◦ 拍摄模式	初级	UHP-140
AcquisitionMode	 和 	-	
DepthLowerLimit	深度范围	初级	
DepthUpperLimit			
DeviceScanType	-	-	
PointCloudSurfaceSmoothing	点云后处理 • 表面平滑	初级	固件版本2.1.0及以上可用
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	

GenICam支持的参数	Mech-Eye Viewer参数	可见级别	备注
CloudOutlierFilterMode	点云后处理 • 噪点去除	初级	固件版本2.0.2及以下可用
CloudSmoothMode	点云后处理 • 点云平滑	初级	
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	参数组下拉菜单选项	-	

### 7.3.2. 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、深度范围及感兴趣区域等相机参数时，通常需要借助可视化工具不断调节从而获取理想的值。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，你可以借助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，你可以直接在GenICam客户端中读取最新的相机配置。

#### 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右侧的**参数**标签页中选择参数组。



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。

#### 调节参数

查看以下内容，分别设置自动曝光感兴趣区域、深度范围及感兴趣区域。

#### 设置自动曝光感兴趣区域

请执行以下步骤设置自动曝光感兴趣区域：

1. 在Mech-Eye Viewer右侧的**参数**标签页中，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**右侧的[编辑]，进入**设置感兴趣区域**页面。
4. 设置**自动曝光感兴趣区域**，详细信息请参考[设置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

请执行以下步骤设置**深度范围**：

1. 在Mech-Eye Viewer右侧的**参数**标签页中，选择**深度范围**。
2. 双击**深度范围**右侧的[编辑]，进入**设置深度范围**页面。
3. 设置**深度范围**，详细信息请参考[设置深度范围](#)。

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

请执行以下步骤设置**感兴趣区域**：

1. 在Mech-Eye Viewer右侧的**参数**标签页中，选择**感兴趣区域**。
2. 双击**感兴趣区域**右侧的[编辑]，**设置感兴趣区域**页面。
3. 设置**感兴趣区域**，详细信息请参考[设置感兴趣区域](#)。

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

## 同步配置修改

在Mech-Eye Viewer中设置完**自动曝光感兴趣区域**、**深度范围**及**感兴趣区域**后，执行以下步骤使GenICam客户端读取参数设置。

1. 在Mech-Eye Viewer右侧的**参数**标签页中，单击最上方的[ **保存** ]或者使用快捷键 **Ctrl+S**，保存参数至参数组。
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，通过**UserSetSelector**和**UserSetLoad**参数选择并读取参数组。

### 7.3.3. 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。



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。

#### 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 APIC++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。

#### 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 APIC++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。

## 8. 相机硬件

### 8.1. 相机型号对比

本章介绍相机的型号及各型号的特点与适用场景。

#### 基础信息

相机当前在售型号分为V3和V4两代。以下表格列出两代型号的基础信息。



- 使用[视野计算器](#)，可计算各型号在不同工作距离下的视野范围。
- 如需下载PDF格式的硬件用户手册和技术参数，或需获取相机3D模型、安装图纸、配件信息、产品宣传册等相关资料，请访问[梅卡曼德下载中心](#)。

#### V4相机

型号	2D图颜色	物料编号	对焦距离 (mm)	推荐工作距离 (mm)
UHP-140	黑白	UHP-140-MP30D300M	300	280~320
NANO	黑白	NANO-V4D350M	350	300~450
		NANO-V4D550M	550	450~600
	彩色	NANO-V4D350C	350	300~450
		NANO-V4D550C	550	450~600
PRO S	黑白	PRO S-V4D500M	500	500~600
		PRO S-V4D700M	700	600~800
		PRO S-V4D1000M	1000	800~1000
	彩色	PRO S-V4D500C	500	500~600
		PRO S-V4D700C	700	600~800
		PRO S-V4D1000C	1000	800~1000
PRO M	黑白	PRO M-V4D1200M	1200	1000~1300
		PRO M-V4D2000M	2000	1300~2000
	彩色	PRO M-V4D1200C	1200	1000~1300
		PRO M-V4D2000C	2000	1300~2000
LSR L	彩色 <sup>(1)</sup>	LSR L-V4D1500A	1500	1200~1800
		LSR L-V4D3000A	3000	1800~3000
LSR S	彩色 <sup>(1)</sup>	LSR S-V4D800A	800	500~900
		LSR S-V4D1400A	1400	900~1500
DEEP	彩色	DEEP-V4D3000A	3000	1200~3500

(1) 指2D图(纹理)的颜色。详细说明请参考[数据类型](#)。



更多相机技术参数，请参阅[技术参数](#)。



## V3相机

型号	2D图颜色	对焦距离 (mm)	推荐工作距离 (mm)
Pro XS	黑白	350	300~450
		550	450~650
Log S	彩色	700	500~750
		1000	750~1000
Log M	彩色	1200	1000~1300
		2000	1300~2000
Laser L Enhanced	黑白	1500	1200~1700
		3000	1700~3000



更多相机技术参数，请参阅[V3相机技术参数](#)。

## 特点与适用场景

不同型号相机的特点与适用场景详见下表。

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 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。
Pro XS (V3)		
DEEP (V4)	大视野，大景深，高速度，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。
PRO S (V4)	高精度，高速度，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 M (V4)		
Log M (V3)	高速度，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。
Log S (V3)		

## 8.2. 硬件用户手册

### 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，每个部分均经过检查。如您发现任何错误或有任何疑问，请随时与梅卡曼德联系。
- 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，再使用相机。

### 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；请勿将产品放入火中，或机械粉碎，否则可能导致爆炸。
- 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。自行维修或拆卸等导致的损坏或损失，梅卡曼德不承担任何责任。
- 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。相机的工作温度范围详见技术参数。
- 请在室内使用相机（除LSR S外）。
- 请在海拔4000米以下的环境中使用相机。
- 请勿直视相机发射的光束。
- 相机需安装在通风且开阔的地方。

### 使用前检查

- 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/导轨电源使用须知

- 请不要在电源接口、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。
- 请使用24V的隔离适配器/导轨电源，且输出功率应不小于36W（NANO和PRO XS）或90W（其他型号）。
- 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。
- 请勿进入激光束及其反射范围。
- 本产品接通电源后，请等待至少30分钟再使用，否则可能导致数据质量不稳定。调节参数时，请勿关闭电源，否则可能导致丢失部分或全部的修改。

## 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。请勿将旧电池扔进生活垃圾中；请勿随意丢弃废旧相机，不当处置可能污染环境。

## 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。如果您需要更多相关信息，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。

## 包装清单

相机	
配件袋	
使用说明书	
标定板 (仅UHP-140)	
DC电源线 (20m) CBL-PWR-20M-LU	
网线 (20m) CBL-ETH-20M-LU	



- 使用前，请确认包装完好，相机无损坏，配件无缺失。如有损坏或物品缺失，请联系梅卡曼德。
- 如需使用其他长度的线缆，请联系梅卡曼德。
- 如需查看线缆的技术参数，请联系梅卡曼德。
- 除UHP-140外，标定板为可选配件。

## 可选配件

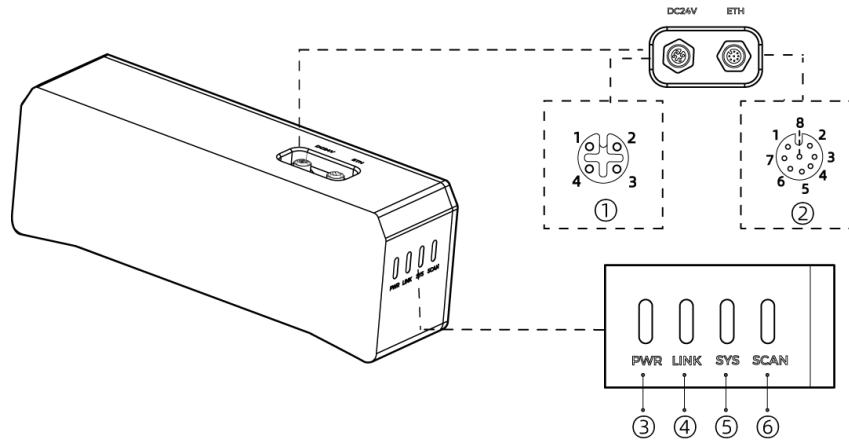
导轨电源	
标定板 (除UHP-140外)	



如需查看导轨电源或标定板的技术参数，请联系梅卡曼德。

## 功能示意图

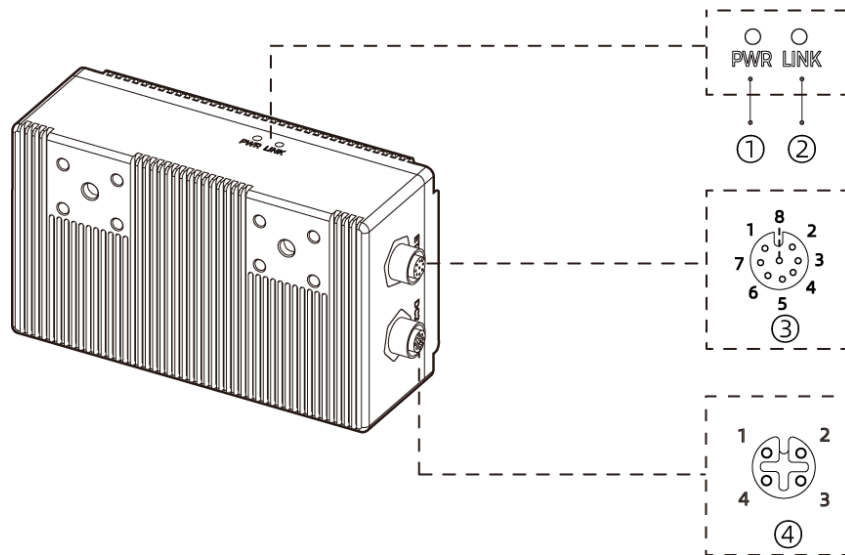
### PRO S/PRO M、DEEP、UHP-140、LSR S和LSR L



No.	名称	功能
①	DC 24V电源接口	1: GND 2: GND 3: 24V DC 4: 24V 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
⑥	SCAN指示灯	常亮：采集和计算中
		熄灭：未采集

### NANO和PRO XS




上图以NANO为例。

No.	名称	功能
①	PWR指示灯	熄灭：电源未连接 绿灯亮：电压正常
②	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
③	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
④	DC 24V电源接口	1: GND                         2: GND 3: 24V DC                      4: 24V DC

## 安装

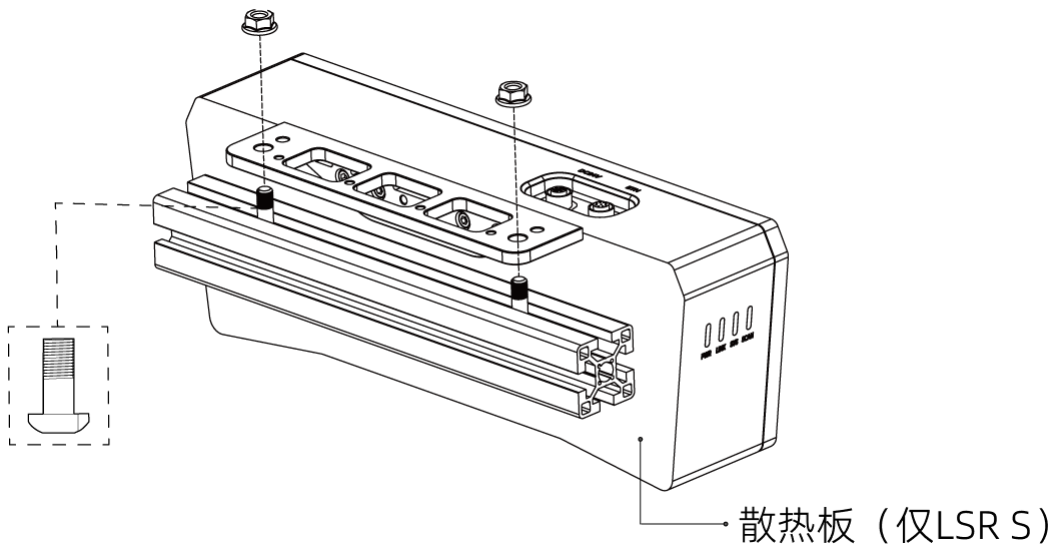
-  相机尺寸信息详见[技术参数](#)。请自行准备扳手。

### 使用L型转接件安装

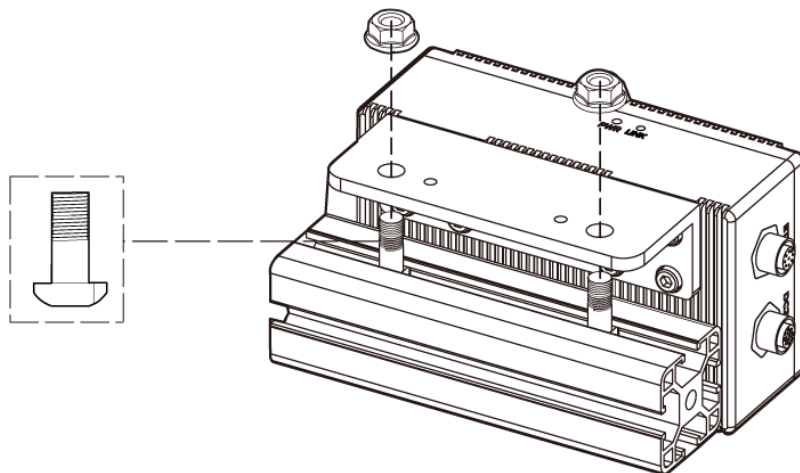
-  发货时，L型转接件与散热板（仅LSR S）已安装至相机背部。

如下图所示，使用扳手拧紧两颗螺母，固定相机。



- PRO S/PRO M、DEEP、UHP-140、LSR S和LSR L:



- NANO和PRO XS:

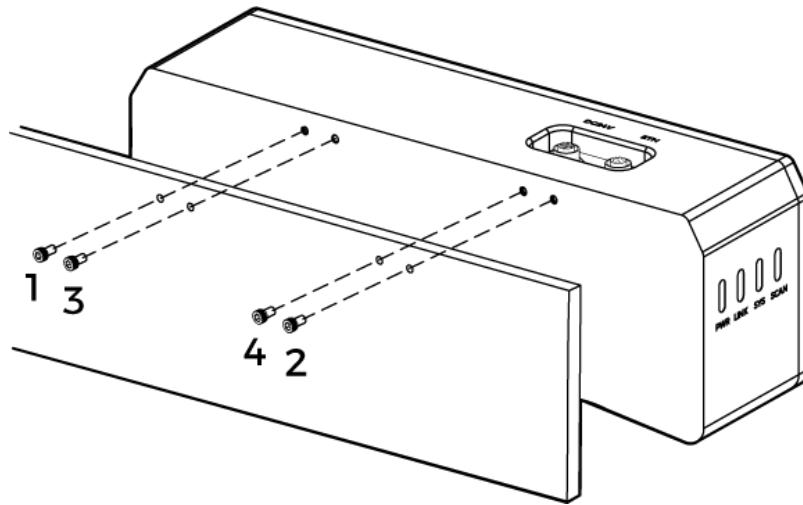


### 使用相机背面螺纹孔安装

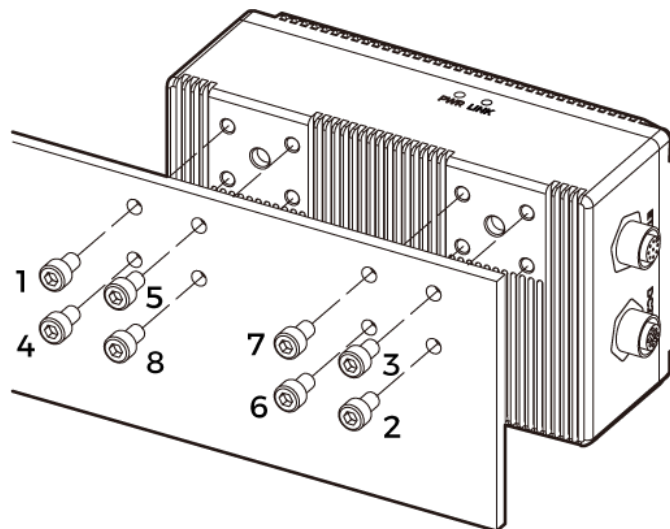
-  安装前请拆卸L型转接件与散热板（仅LSR S）。
-  LSR S使用此安装方式时，需将相机背面贴紧金属散热面安装，防止相机散热不良，影响正常工作。

如下图所示，使用扳手按顺序先预紧，再按顺序拧紧螺钉。


- PRO S/PRO M、DEEP、UHP-140、LSR S和LSR L:



- NANO和PRO XS:

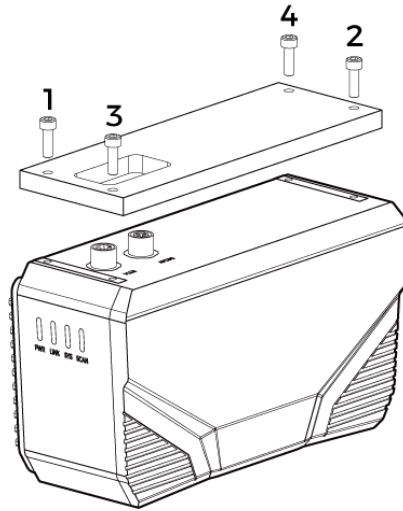


### 使用相机顶部螺纹孔安装（LSR S和UHP-140）

-  安装前请拆卸L型转接件。

如下图所示，使用扳手按顺序先预紧，再按顺序拧紧螺钉。





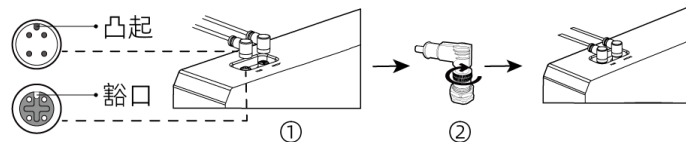
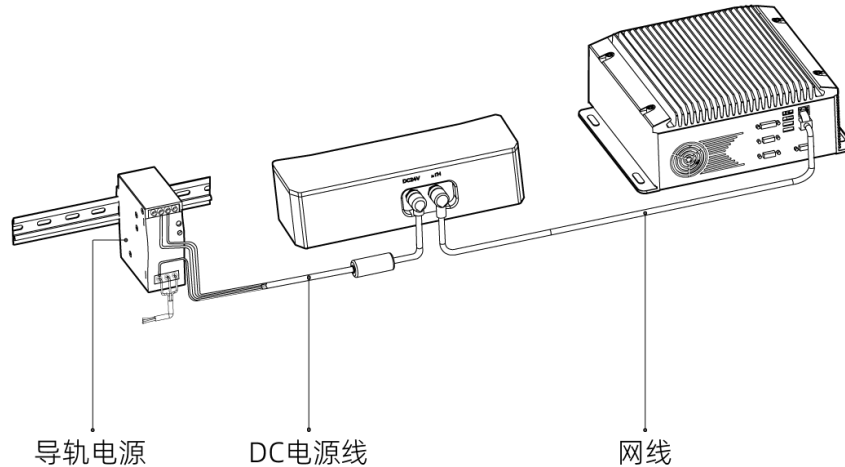
## 连接

### 网线

如下图所示，将网线的航插头插入相机的ETH网口，RJ45接头插入工控机的网口。

### DC电源线

如下图所示，将DC电源线的航插头插入相机的DC 24V电源接口。



连接网线与DC电源线时：

1. 将航插的凸起对准对应接口的豁口插入。
2. 拧紧紧固螺母，推荐扭矩为 $0.7\text{N}\cdot\text{m}$ 。拧紧后，约有2mm间隙。



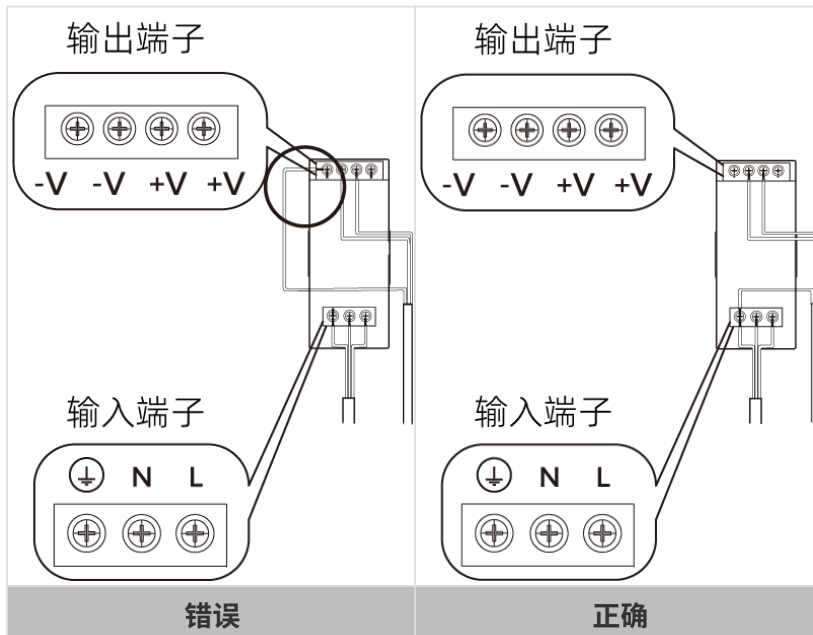
- 连接相机与工控机时，可使用交换机连接。

- 连接时，请最后接通电源。接通电源后，PWR指示灯应绿色常亮。否则，请联系梅卡曼德。
- 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。

## 导轨电源



- 请自行准备AC电源线。
- 本节的导轨电源为梅卡曼德提供的导轨电源。如您自行准备导轨电源，请查看对应的说明书进行连接。



AC电源线共有三股接线插头，分别为：L、N、PE (⊕)。

DC电源线共有三股接线插头，分别为：+V、-V、PE (⊕)。

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。

- 导轨电源须放在配电箱中使用。
- 导轨电源或连接导轨电源的导轨应可靠接地。如多台导轨电源安装在同一导轨时，各导轨电源应保持一定间距。
- AC电源插座应采用有保护接地线（PE线）的单相三线电源插座。

## 维护保养

### 清洁

清洁相机机身时，请使用干净软布擦去灰尘和浮屑。若要去除镜头上的污渍，可使用滴有镜头清洁剂或玻璃清洁剂的干净柔软无绒布小心擦拭，以免造成镜头划伤。



- 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。由于进水而导致的损害和损失，梅卡曼德不承担任何责任。

## 存储

- LSR S：产品防护等级为IP67。可有效防止粉尘和水进入其内部，影响其功能。使用中应避免将产品长时间浸入水中。不使用产品时，请将其存于室内阴凉干燥、通风良好的地方。产品存储温度范围：-20~60°C。
- LSR S以外的型号：产品防护等级为IP65。可有效防止粉尘进入其内部，影响其功能。使用中应避免将产品长时间浸入水中，置于高湿度的环境中，或长时间放在户外。不使用产品时，请将其存于室内阴凉干燥、通风良好的地方。产品存储温度范围：-20~60°C。



- 存放前请断开与导轨电源/电源适配器的连接，以免火灾发生。
- 请勿将镜头正对太阳或对准强光源。强光可能会损坏影像传感器，导致图像出现白色模糊现象。

## 免责声明

强烈建议为本产品使用梅卡曼德配套的电源和线缆，以确保符合安全和EMC辐射及抗扰度标准。使用第三方电源和线缆产生的任何问题，梅卡曼德不承担任何责任。


## 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	X	○	○	○	○	○
电源	X	○	○	○	○	○
线缆	X	○	○	○	○	○
配件	X	○	○	○	○	○

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。  
 X：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。



## 商标与法律声明

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。

© 2024，梅卡曼德（北京）机器人科技有限公司

未经梅卡曼德（北京）机器人科技有限公司（以下简称梅卡曼德）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与销售。

凡侵犯梅卡曼德商标权的，梅卡曼德必依法追究其法律责任。

梅卡曼德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。对于购买并使用本产品的用户，您可以下载并打印或复制本使用说明书作为个人或组织内部使用。未经梅卡曼德的授权，禁止将本使用说明书的内容用于任何其他用途；且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。

## 8.3. 技术参数

以下章节提供V4各型号的技术参数：

- [NANO](#)
- [PRO S / PRO M](#)
- [UHP-140](#)
- [DEEP](#)
- [LSR S](#)
- [LSR L](#)

V3型号的技术参数，详见[V3相机技术参数](#)。

### 8.3.1. NANO

#### 技术参数

产品名称	Mech-Eye工业级3D相机	
型号	NANO	
对焦距离 <sup>(1)</sup>	350mm	550mm
推荐工作距离	300~450mm	450~600mm
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>(2)</sup>	0.1mm @ 0.5m	
VDI/VDE测量精度 <sup>(3)</sup>	0.1mm @ 0.5m	
重量	0.7kg	
基线长度	68mm	
尺寸	145 × 51 × 85mm	

典型采集时间	0.6~1.1s	
2D图颜色	黑白	彩色
光源	蓝光LED (459nm, RG2)	白光LED (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/KC/ISED/NRTL	
防护等级 <sup>(4)</sup>	IP65	
散热	被动	

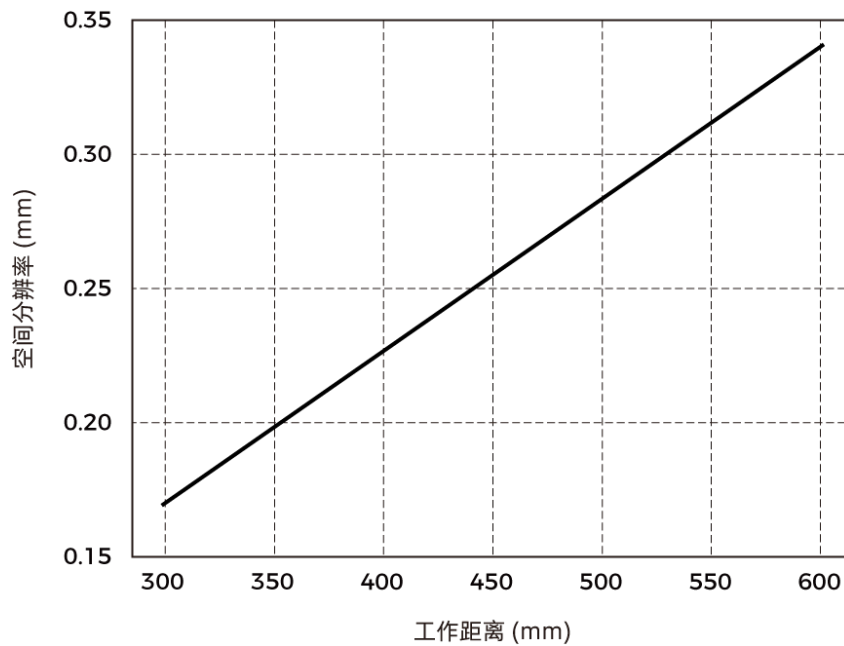
(1) 相机有两种规格的对焦距离，且每种对焦距离对应不同的推荐工作距离。请根据实际情况进行选择。

(2) 单点Z值100次测量的一倍标准差，测量目标为陶瓷板。

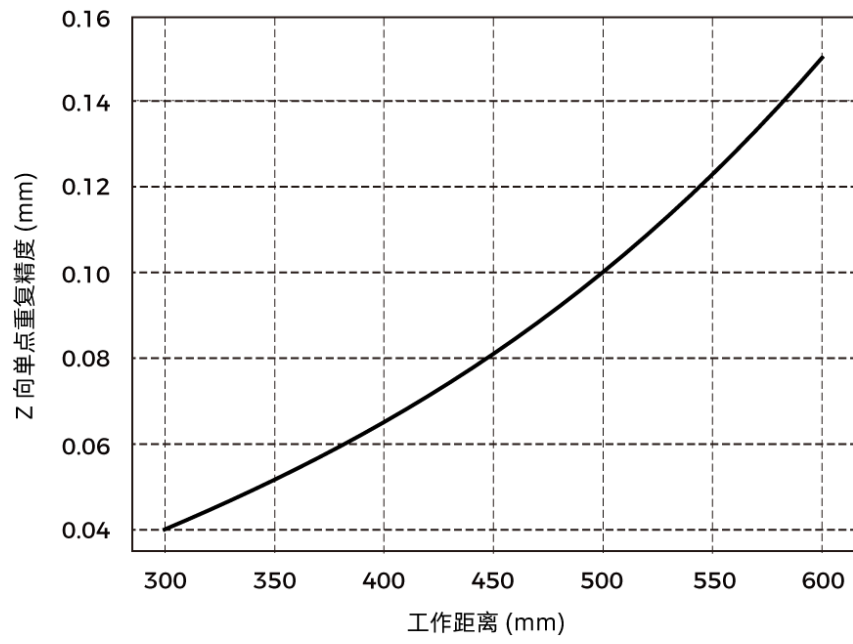
(3) 基于VDI/VDE 2634 Part II标准。

(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。

### 空间分辨率

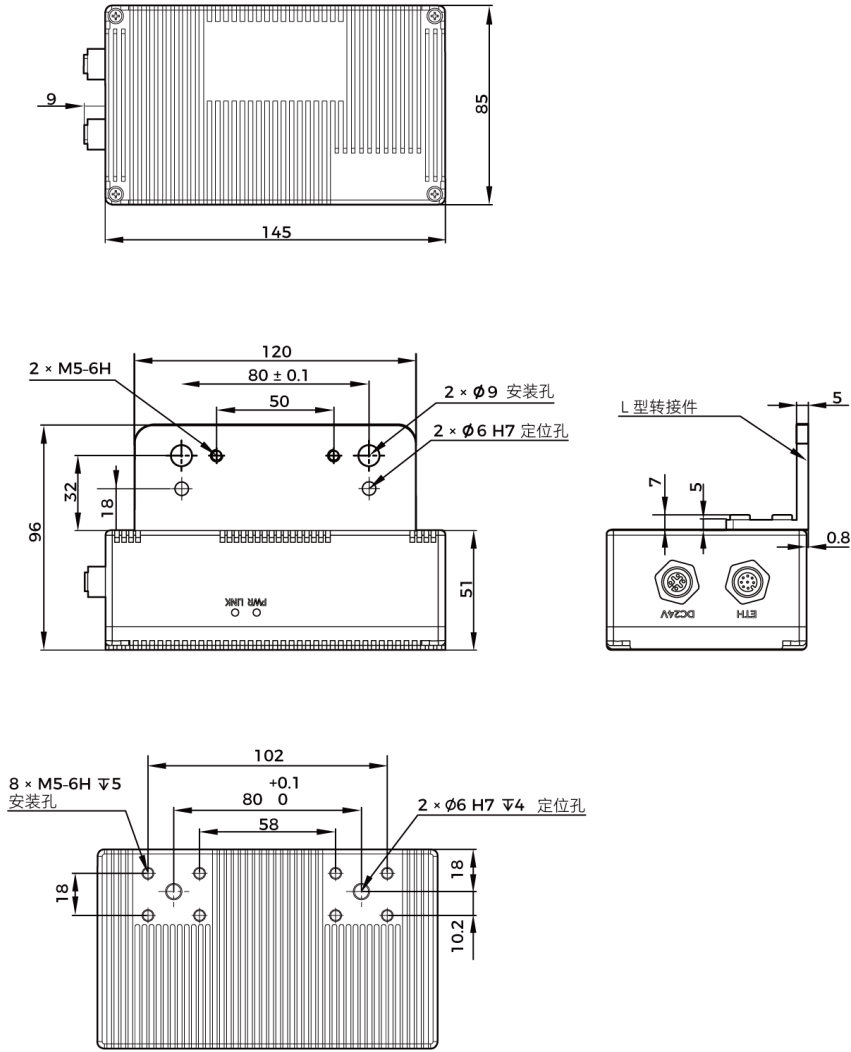


### Z向单点重复精度



### 相机尺寸

单位：mm



### 8.3.2. PRO S / PRO M

#### 技术参数

产品名称	Mech-Eye工业级3D相机		
型号	PRO S		
对焦距离 <sup>(1)</sup>	500mm	700mm	1000mm
推荐工作距离	500~600mm	600~800mm	800~1000mm
近端视场	370 × 240mm @ 0.5m		
远端视场	800 × 450mm @ 1m		
分辨率	1920 × 1200		
像素数	2.3MP		
Z向单点重复精度( $\sigma$ ) <sup>(2)</sup>	0.05mm @ 1m		
VDI/VDE测量精度 <sup>(3)</sup>	0.1mm @ 1m		
重量	1.6kg		
基线长度	180mm		

尺寸	265 × 57 × 100mm	
典型采集时间	0.3~0.6s	
2D图颜色	黑白	彩色
光源	蓝光LED (459nm, RG2)	白光LED (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/ISED/VCCI/UKCA/KC	
防护等级 <sup>(4)</sup>	IP65	
散热	被动	

(1) 相机有三种规格的对焦距离，且每种对焦距离对应不同的推荐工作距离。请根据实际情况进行选择。

(2) 单点Z值100次测量的一倍标准差，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。

产品名称	Mech-Eye工业级3D相机	
型号	PRO M	
对焦距离 <sup>(1)</sup>	1200mm	2000mm
推荐工作距离	1000~1300mm	1300~2000mm
近端视场	800 × 450mm @ 1m	
远端视场	1500 × 890mm @ 2m	
分辨率	1920 × 1200	
像素数	2.3MP	
Z向单点重复精度( $\sigma$ ) <sup>(2)</sup>	0.2mm @ 2m	
VDI/VDE测量精度 <sup>(3)</sup>	0.2mm @ 2m	
重量	1.9kg	
基线长度	270mm	
尺寸	353 × 57 × 100mm	
典型采集时间	0.3~0.6s	
2D图颜色	黑白	彩色
光源	蓝光LED (459nm, RG2)	白光LED (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/KC/ISED/NRTL
防护等级 <sup>(4)</sup>	IP65
散热	被动

(1) 相机有两种规格的对焦距离，且每种对焦距离对应不同的推荐工作距离。请根据实际情况进行选择。

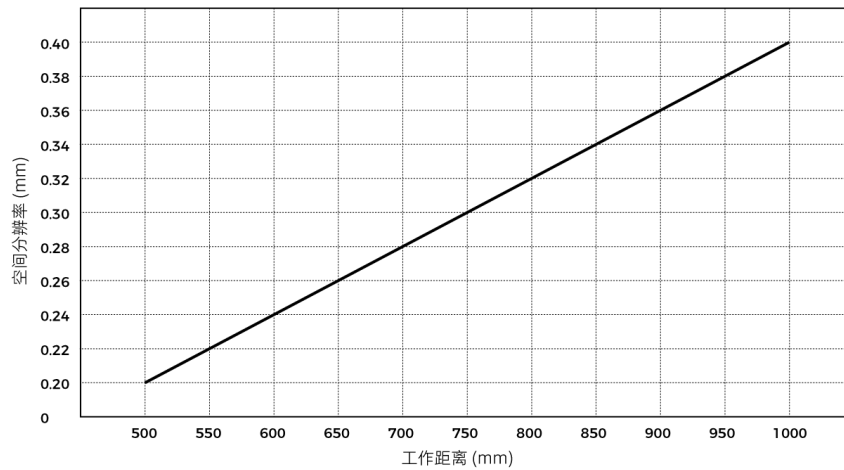
(2) 单点Z值100次测量的一倍标准差，测量目标为陶瓷板。

(3) 基于VDI/VDE 2634 Part II标准。

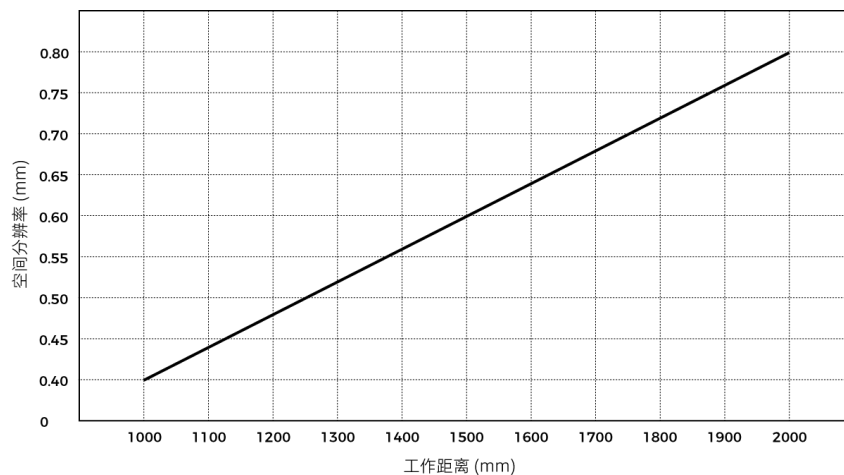
(4) 根据IEC 60529标准测试。其中，6代表防尘等级，5代表防水等级。

## 空间分辨率

### PRO S

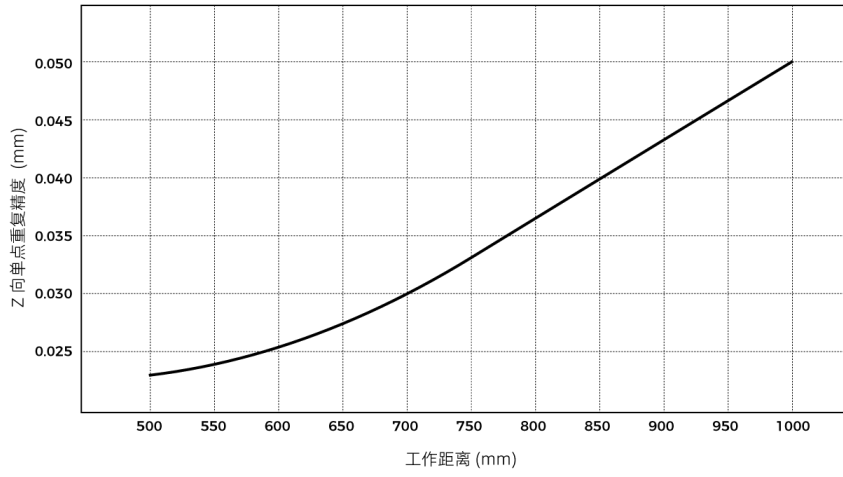


### PRO M

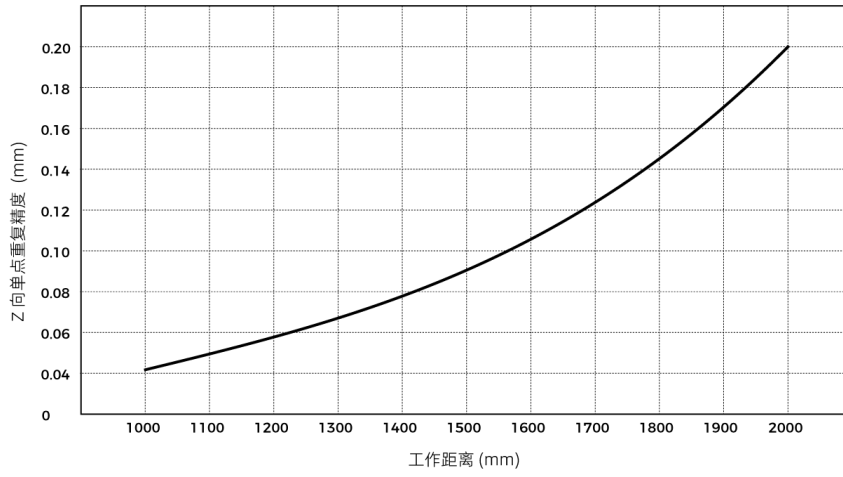


## Z向单点重复精度

### PRO S



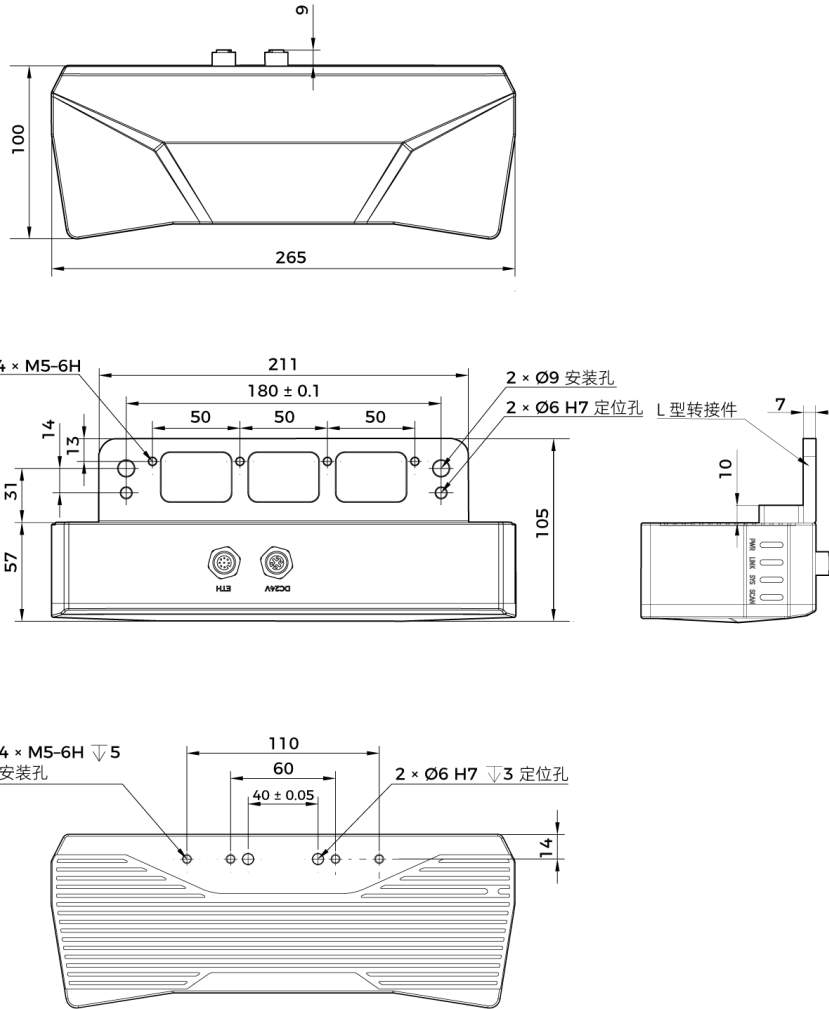
### PRO M



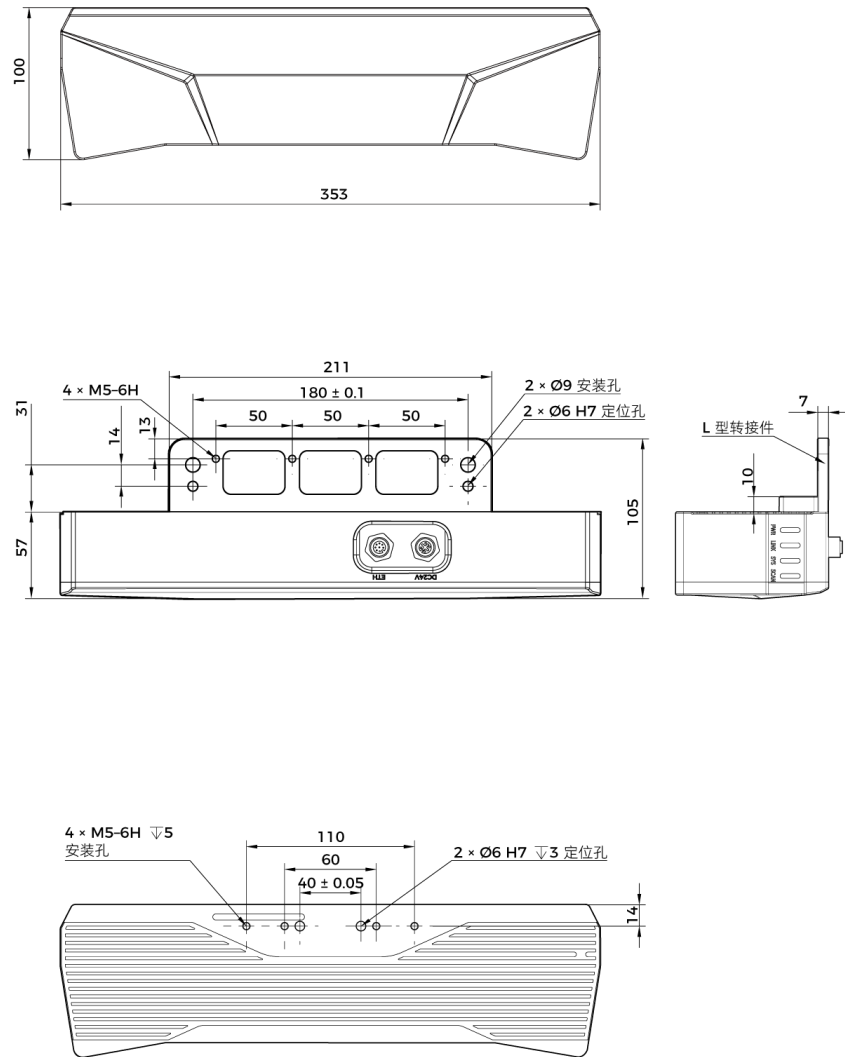
### 相机尺寸

单位：mm

### PRO S



## PRO M



### 8.3.3. UHP-140

#### 技术参数

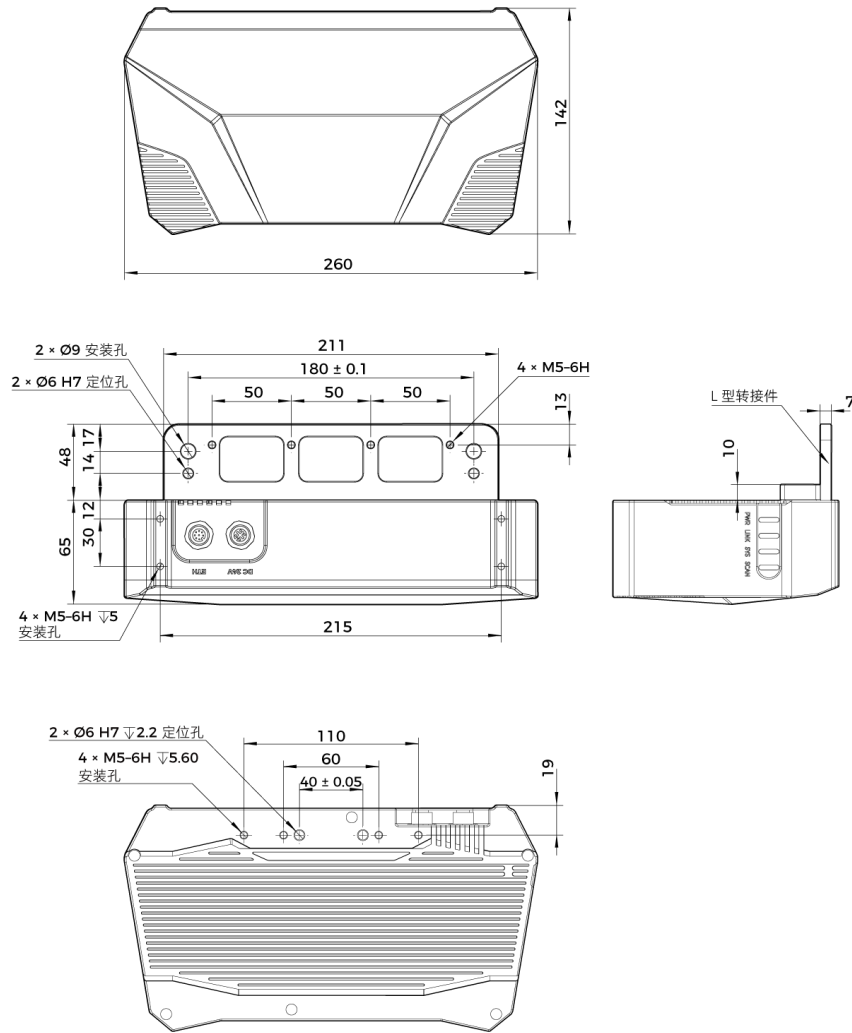
产品名称	Mech-Eye工业级3D相机
型号	UHP-140
对焦距离	300mm
推荐工作距离	280~320mm
近端视场	135 × 90mm @ 280mm
远端视场	150 × 100mm @ 320mm
分辨率	2048 × 1536
像素数	3MP
Z向单点重复精度( $\sigma$ ) <sup>(1)</sup>	2.6 $\mu$ m @ 0.3m
Z向区域重复精度( $\sigma$ ) <sup>(2)</sup>	0.09 $\mu$ m @ 0.3m
VDI/VDE测量精度 <sup>(3)</sup>	0.03mm @ 0.3m
典型采集时间	0.6~0.9s

产品名称	Mech-Eye工业级3D相机
重量	约1.9kg
基线长度	约80mm
尺寸	约260 × 65 × 142mm
2D图颜色	黑白
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/KC/ISED/NRTL
防护等级 <sup>(4)</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) 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



### 8.3.4. DEEP

#### 技术参数

产品名称	Mech-Eye工业级3D相机
型号	DEEP
对焦距离	3000mm
推荐工作距离	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
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
重量	2.4kg
基线长度	300mm

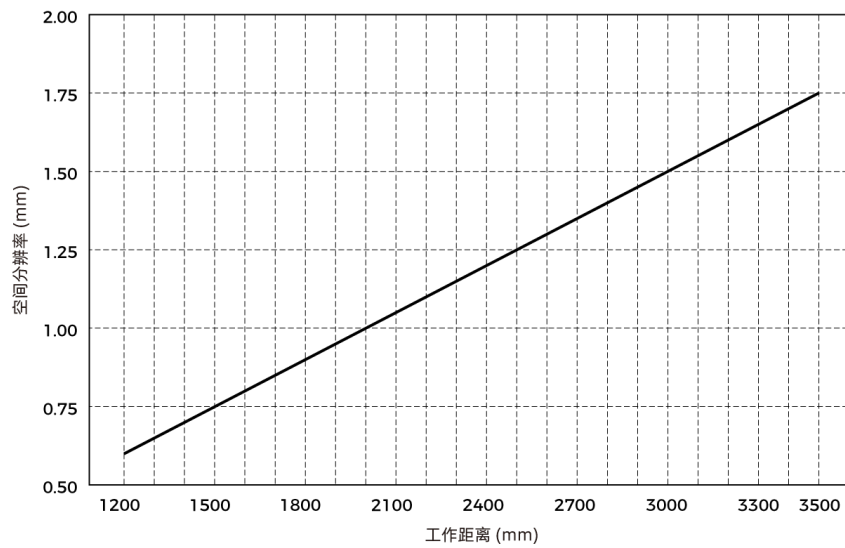
尺寸	366 × 77 × 92mm
典型采集时间	0.5~0.9s
光源	红色激光 (638nm, 2类)
工作温度范围	-10~45°C
通讯接口	千兆以太网
输入	24V DC, 3.75A
安全与电磁兼容	CE/FCC/VCCI/KC/ISED/NRTL
防护等级 <sup>(3)</sup>	IP65
散热	被动

(1) 单点Z值100次测量的一倍标准差，测量目标为陶瓷板。

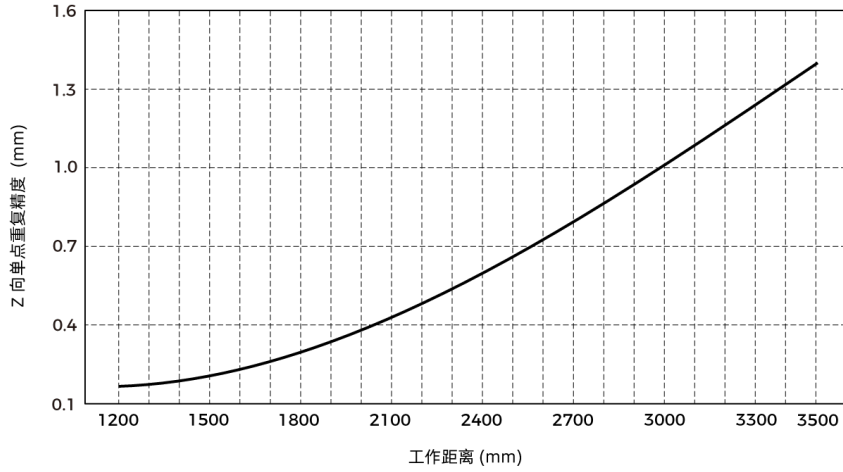
(2) 基于VDI/VDE 2634 Part II标准。

(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

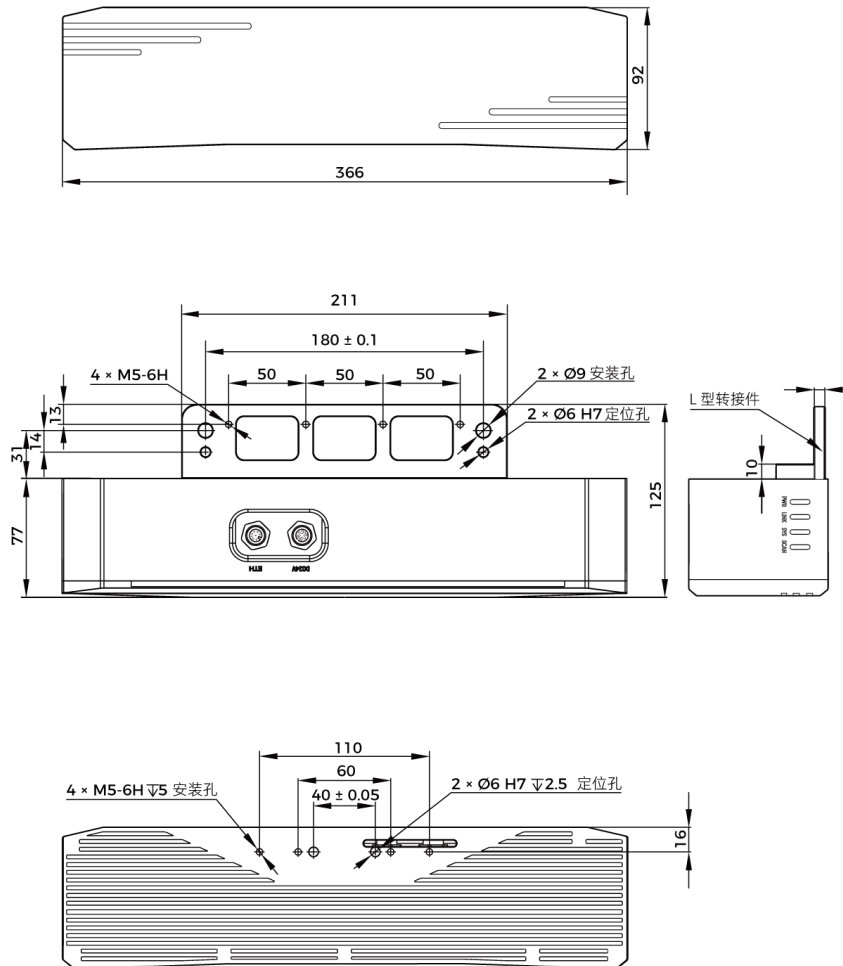


### Z向单点重复精度



## 相机尺寸

单位：mm



## 激光产品安全

依据GB 7247.1-2012标准施行激光等级分级。

型号	波长	最大输出功率	激光等级
----	----	--------	------

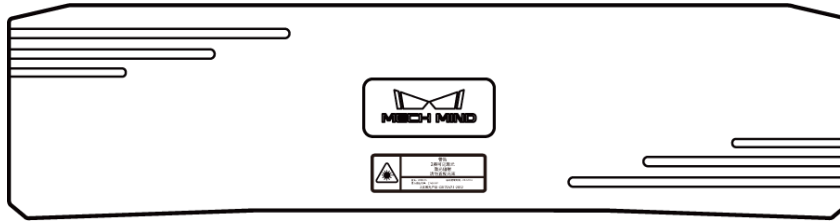


DEEP	638nm	2.46mW	Class 2
------	-------	--------	---------

### 标签



### 标签粘贴位置



## 8.3.5. LSR S

### 技术参数

产品名称	Mech-Eye工业级3D相机	
型号	LSR S	
对焦距离 <sup>(1)</sup>	800mm	1400mm
推荐工作距离	500~900mm	900~1500mm
近端视场	480 × 360mm @ 0.5m	
远端视场	1500 × 1200mm @ 1.5m	
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>(2)</sup>	0.2mm @ 1.5m	
VDI/VDE测量精度 <sup>(3)</sup>	1.0mm @ 1.5m	
重量 <sup>(4)</sup>	1.9kg	
基线长度	140mm	
尺寸 <sup>(4)</sup>	228 × 77 × 126mm	
典型采集时间	0.5~0.9s	
光源	红色激光 (638nm, 2类)	
工作温度范围 <sup>(5)</sup>	-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/KC/ISED/NRTL
防护等级 <sup>(6)</sup>	IP67
散热	被动

(1) 相机有两种规格的对焦距离，且每种对焦距离对应不同的推荐工作距离。请根据实际情况进行选择。

(2) 单点Z值100次测量的一倍标准差，测量目标为陶瓷板。

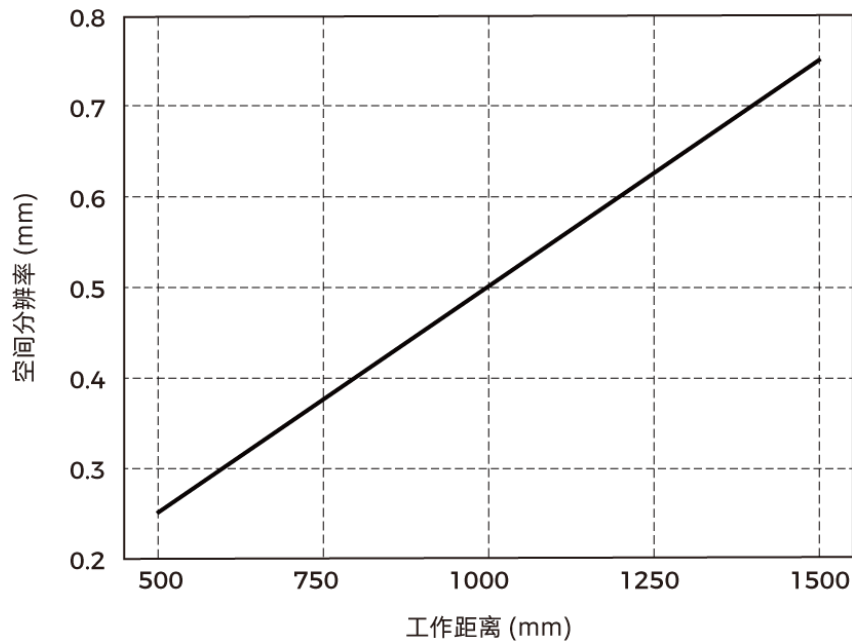
(3) 基于VDI/VDE 2634 Part II标准。

(4) 不含L型转接件与散热板。

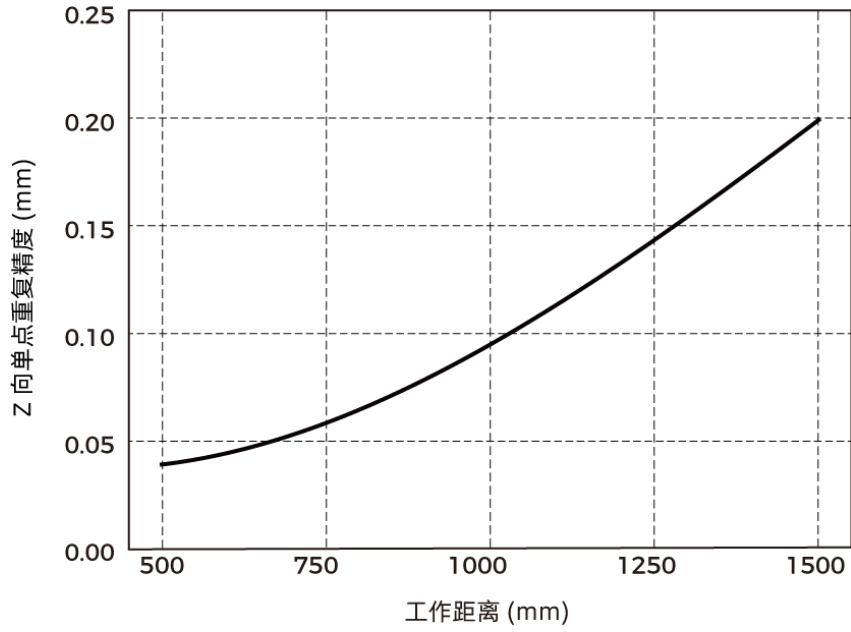
(5) 相机本体（不含散热板）的工作温度范围。

(6) 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

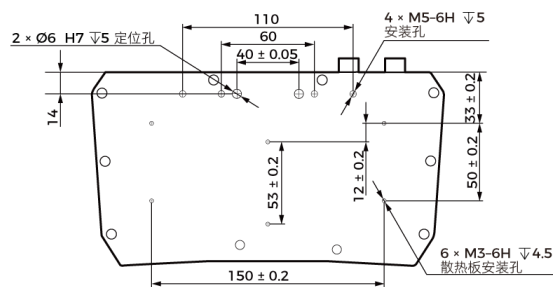
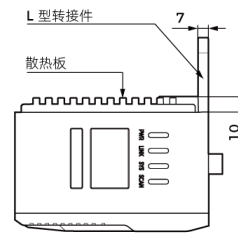
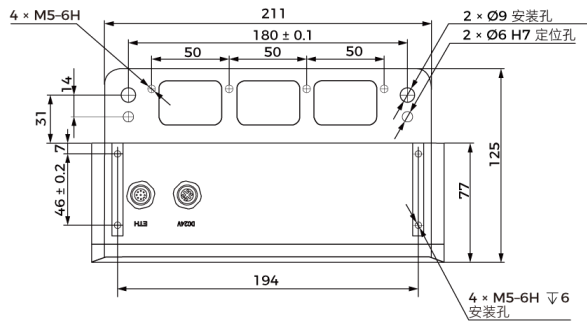
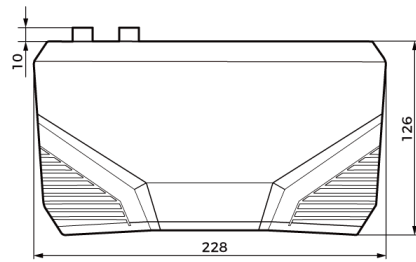


### Z向单点重复精度



### 相机尺寸

单位：mm



## 激光产品安全

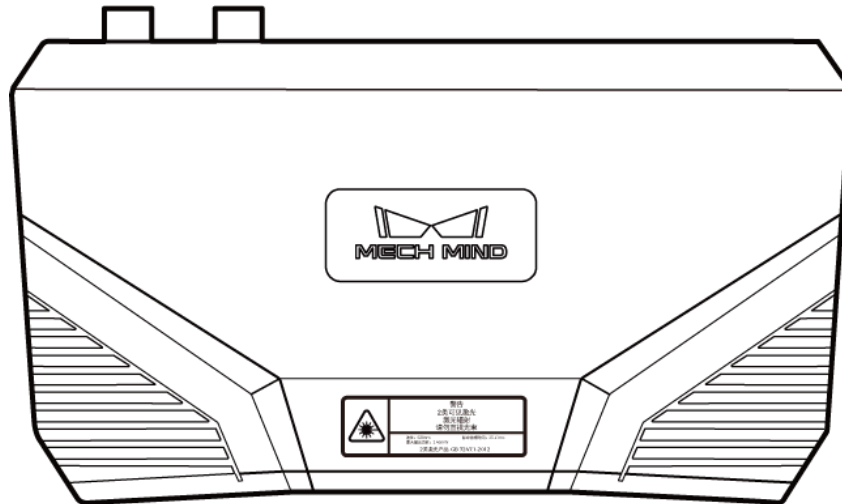
依据GB 7247.1-2012标准施行激光等级分级。

型号	波长	最大输出功率	激光等级
LSR S	638nm	2.46mW	Class 2

## 标签



## 标签粘贴位置



## 8.3.6. LSR L

### 技术参数

产品名称	Mech-Eye工业级3D相机	
型号	LSR L	
对焦距离 <sup>(1)</sup>	1500mm	3000mm
推荐工作距离	1200~1800mm	1800~3000mm
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>(2)</sup>	0.5mm @ 3m
VDI/VDE测量精度 <sup>(3)</sup>	1.0mm @ 3m
重量	2.9kg
基线长度	380mm
尺寸	459 × 77 × 86mm
典型采集时间	0.5~0.9s
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/KC/ISED/NRTL
防护等级 <sup>(4)</sup>	IP65
散热	被动

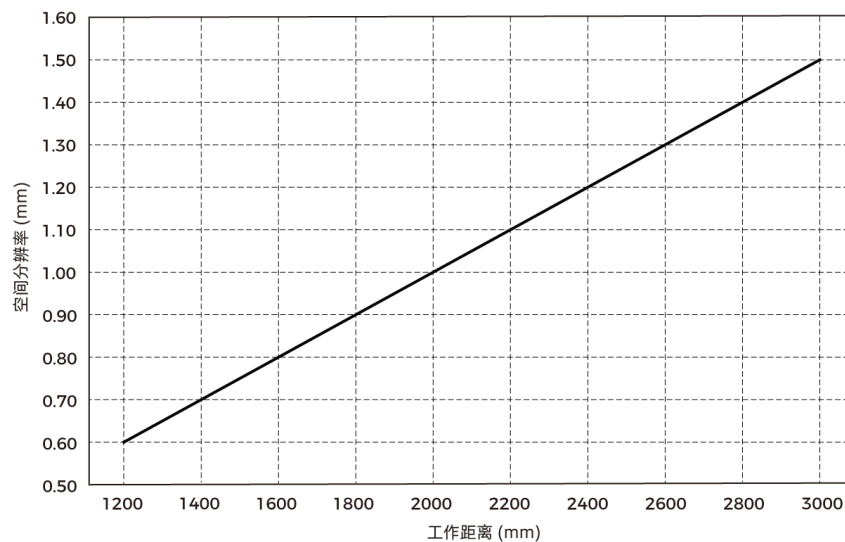
(1) 相机有两种规格的对焦距离，且每种对焦距离对应不同的推荐工作距离。请根据实际情况进行选择。

(2) 单点Z值100次测量的一倍标准差，测量目标为陶瓷板。

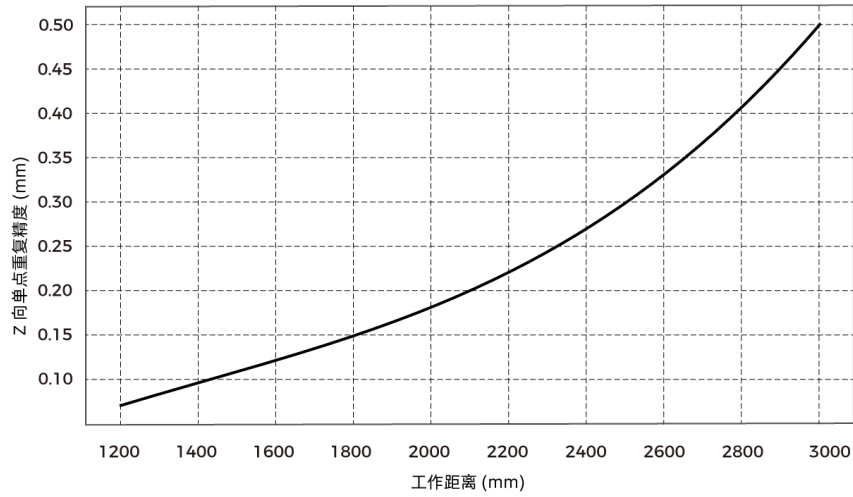
(3) 基于VDI/VDE 2634 Part II标准。

(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。

### 空间分辨率

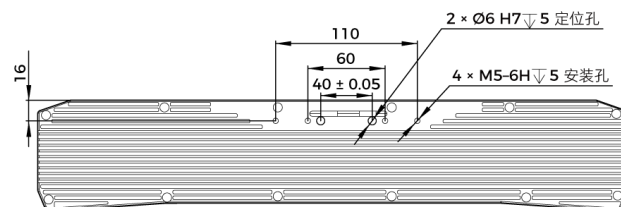
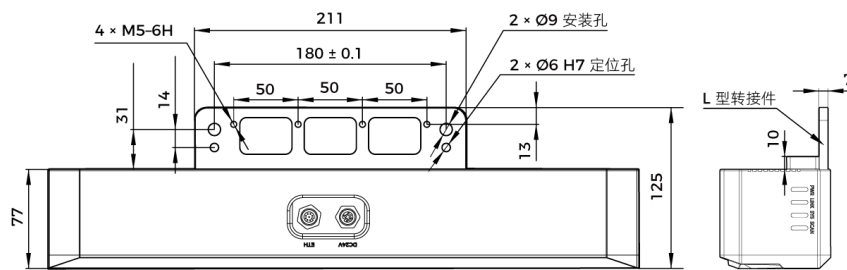
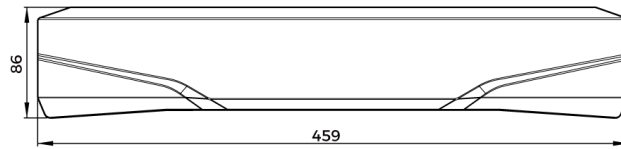


### Z向单点重复精度



## 相机尺寸

单位：mm



## 激光产品安全

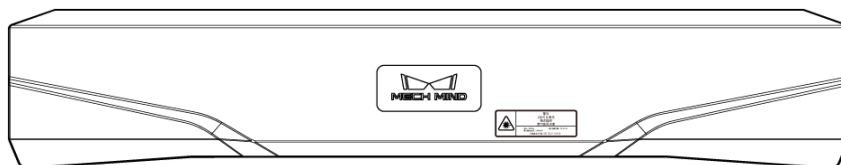
依据GB 7247.1-2012标准施行激光等级分级。

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

### 标签



### 标签粘贴位置



## 8.3.7. V3相机技术参数

### 技术参数

产品名称	Mech-Eye工业级3D相机			
型号	Pro XS	Log S	Log M	Laser L Enhanced
对焦距离	350	700	1200	1500
	550	1000	2000	3000
推荐工作距离范围	300~450mm	500~750mm	1000~1300mm	1200~1700mm
	450~650mm	750~1000mm	1300~2000mm	1700~3000mm
近端视场	220 × 160mm @ 0.3m	360 × 250mm @ 0.5m	520 × 390mm @ 0.8m	1500 × 1200mm @ 1.5m
远端视场	430 × 320mm @ 0.6m	710 × 490mm @ 1.0m	1410 × 960mm @ 2.0m	3000 × 2400mm @ 3.0m
分辨率	1280 × 1024			4096 × 3000
像素数	1.3MP			12.0MP
Z向重复精度 (σ)	0.1mm @ 0.5m	0.1mm @ 1m	0.3mm @ 2m	0.5mm @ 3m
标定精度	0.1mm @ 0.5m	0.2mm @ 1m	0.3mm @ 2m	0.5mm @ 3m
典型采集时间	0.7~1.1s	0.3~0.5s	0.3~0.5s	1.4~1.7s
基线长度	约93mm	约150mm	约280mm	约400mm
外形尺寸	约160 × 52 × 87mm	约270 × 72 × 130mm	约387 × 72 × 130mm	约459 × 89 × 145mm

重量	约0.8kg	约2.2kg	约2.4kg	约3.9kg
工作温度范围	0~45°C			-10~45°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			



## 9. 扩展阅读

---

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，了解相机内预置的参数组。

[预置参数组](#)

---

以下章节提供使用相机时可能需要的操作指导。

- [设置电脑IP地址](#)
- [启用巨型帧](#)
- \*\*

### 9.1. 结构光相机工作原理

Mech-Eye工业级3D相机（简称3D相机）是一种结构光3D相机，由2D相机与投光单元组成。投光单元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，计算出物体表面的深度数据。

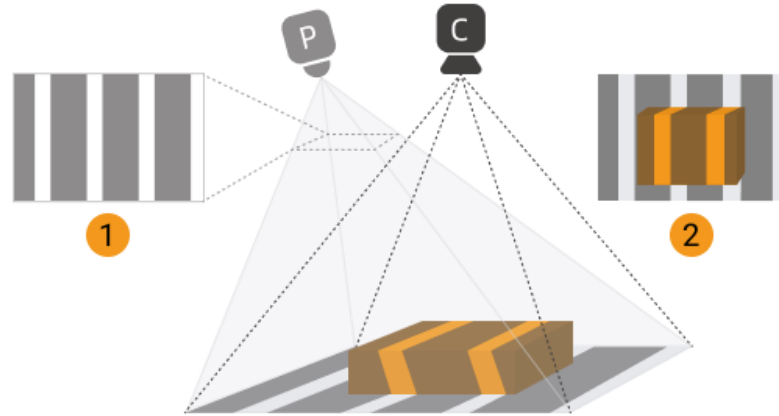
3D相机也可像普通2D相机一样，获取物体的2D图像。

3D相机的完整工作过程，可分为三个阶段：图像采集、计算与处理、输出。

#### 图像采集

使用Mech-Eye SDK触发3D相机采集图像，采集过程为：

1. 投光单元（下图中P）投射条纹样式的光至物体表面。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。



同时，3D相机也采集了未被投射结构光的物体的普通2D图像。

## 计算与处理

3D相机根据条纹样式的变形，计算出物体表面的深度数据。经过进一步处理，生成深度图与点云。

## 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。

- 2D图：投光单元不投光时采集的图像。
- 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。

## 9.2. 预置参数组

将V4相机固件升级至2.2.0版本后，相机中新增适用于不同场景或工件的预置参数组。根据实际场景选择对应的参数组后，微调参数值即可获取质量满足要求的数据，简化参数调节的过程。

本章介绍各型号的预置参数组适用的场景或工件。

型号	参数组名	适用场景/工件
PRO M	反光 + 不反光	同时有易反光和不易反光部分的工件
	反光金属件	易反光的金属件
	纸盒	纸盒
	透明物体	透明物体
PRO S	反光轴棒	易反光的轴棒形工件
	小零件	小且形状复杂的工件，如齿轮
	螺柱	有螺纹的轴棒形工件
	透明物体	透明物体

型号	参数组名	适用场景/工件
LSR L	反光金属件	易反光的金属件
	非反光金属件	不易反光的金属件
DEEP	麻袋	麻袋
	纸箱	纸箱
	塑料周转箱	塑料周转箱
UHP-140	车架	汽车车架
	磁控管	磁控管
	非反光铝材	不易反光的铝材
NANO	金属焊接	金属焊接
	充电口定位	电动汽车充电口定位

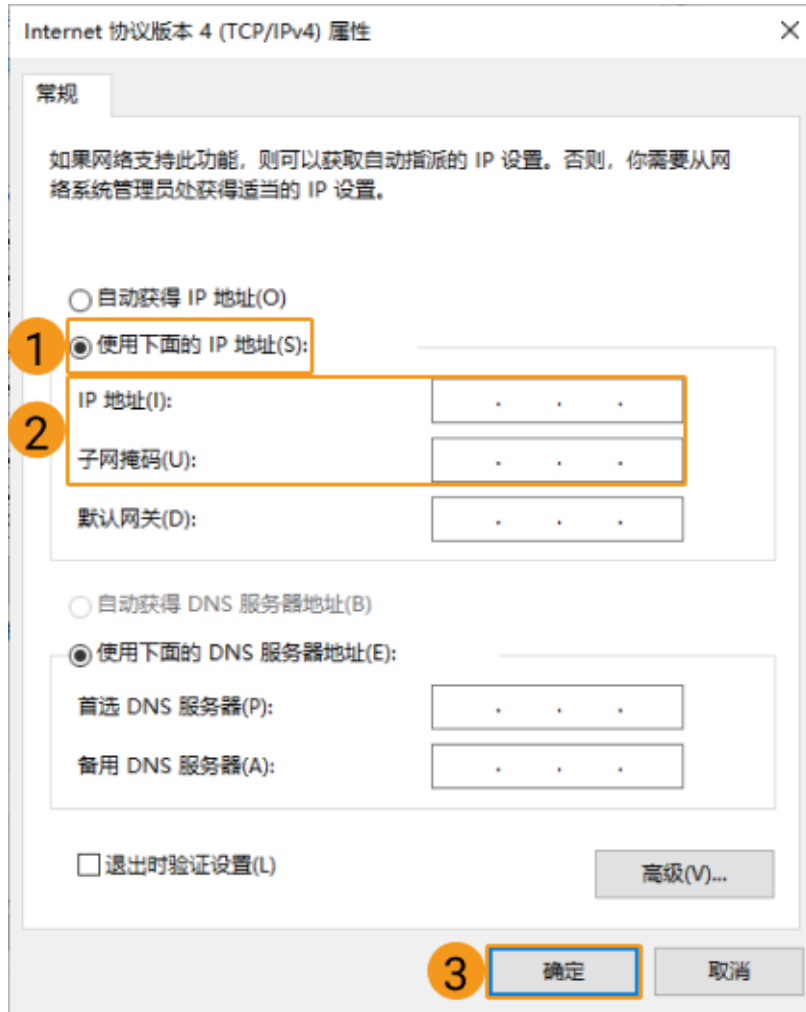
### 9.3. 设置电脑IP地址

本章介绍如何设置与相机相连的电脑网口的IP地址。

1. 在电脑上，选择控制面板 > 网络和Internet > 网络和共享中心 > 更改适配器设置，**网络连接**页面将打开。
2. 右键单击与相机相连的网口，选择**属性**，进入**以太网属性**页面。
3. 选中**Internet协议版本4 (TCP/IPv4)**，然后单击[ **属性** ]按钮进入**Internet协议版本4 (TCP/IPv4) 属性**界面。



4. 选择使用下面的IP地址，并输入IP地址和子网掩码。单击[确定]，设置完成。



 如需自动分配IP地址，请选择**自动获得IP地址**。

## 9.4. 启用巨型帧

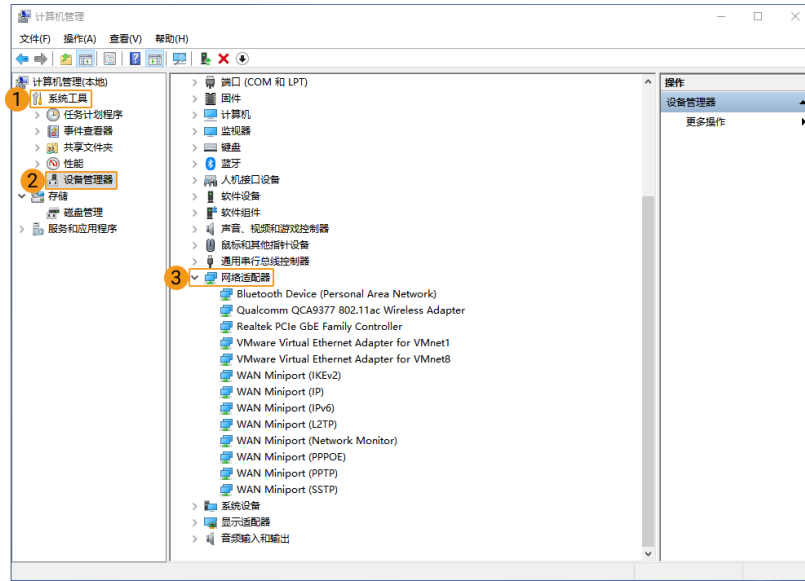
标准以太网帧最多可负载约1500字节的数据，巨型帧是最多可负载约9000字节数据的以太网帧。使用巨型帧可以降低CPU计算量、加快数据传输。

如相机和电脑直接连接或通过交换机连接，且相机的MTU值设置为9000，则需启用电脑上的巨型帧功能以提升数据传输速度。

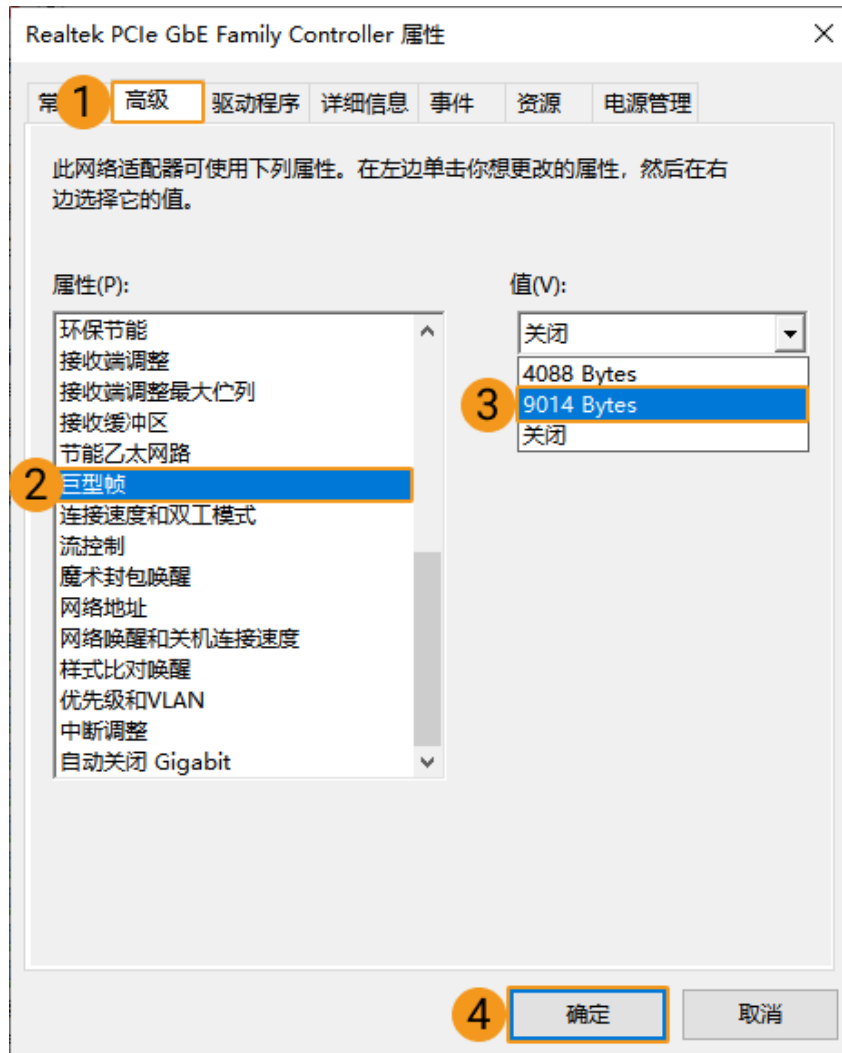
 相机的MTU值在**管理员**用户下可设置。如需更改，请咨询技术支持。

请执行以下步骤启用巨型帧：

1. 右键单击电脑中的**此电脑**，选择**管理**，进入**计算机管理**页面。
2. 在左栏中选择系统工具 > 设备管理器，再在中间区域中选中**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**，并将右侧的**值**设置为**9014 Bytes**。单击[ **确定** ]完成修改。



## 10. 售后支持

### 10.1. 故障排查

本节提供各类常见问题的故障排查指导。



如遇到以下章节未涉及的问题，请访问[梅卡曼德在线社区](#)获取帮助。

#### 相机连接相关

- [接通电源后，相机指示灯指示异常情况](#)
- [Mech-Eye SDK搜索不到相机](#)
- [Mech-Eye SDK无法连接相机](#)
- [Mech-Eye SDK无法连接相机 \(Windows Server\)](#)

#### 数据采集相关

- [相机不投光，且未生成深度图](#)
- [图像采集时间过长](#)

#### 特殊型号和版本相关

- [V2相机升级固件后采集的点云有缺失](#)

#### 10.1.1. 接通电源后，相机指示灯指示异常情况

相机接通电源后，通过相机上的指示灯可确认相机是否处于正常工作状态。

本章提供三种通过指示灯可发现的异常问题的故障排查指导。



- 本章适用于V4相机。如V3相机出现类似问题，请联系技术支持。
- 正常工作状态下指示灯的显示，请参考[相机硬件用户手册](#)。

#### 所有指示灯均不亮

##### 问题现象

接通电源后，相机的四个指示灯均不亮。

##### 可能原因

- 相机线缆连接不正确。

- 电源电压异常。
- 导轨电源未接地。

## 解决方案

请执行以下步骤排查问题：

1. 参考[相机硬件用户手册](#)，检查相机各线缆是否连接正确、插头的紧固螺母是否拧紧。
  - 如线缆连接有误，请更正连接，然后再次检查相机指示灯。
    - 如相机指示灯指示正常情况，故障排查结束。
    - 如仍指示异常情况，请执行步骤2。
  - 如线缆连接无误，请执行步骤2。
2. 断开相机DC电源线与相机的连接，并使用万用表检查航插的电压是否达到24V。
  - 如未达到24V，请检查供电线路中其他线缆的连接等，并做出相应调整。然后，将DC电源线重新连接至相机，并检查相机指示灯。
    - 如相机指示灯指示正常情况，故障排查结束。
    - 如仍指示异常情况，请执行步骤3。
  - 如达到24V，请执行步骤3。
3. 检查电源是否正确接地：如使用导轨电源为相机供电，参考[相机硬件用户手册](#)，确认导轨电源的PE线连接是否正确。如使用电源适配器为相机供电，请检查适配器AC电源线的三相插头连接的插座是否正确接地。
  - 如不正确，相机可能已被损伤，请联系技术支持。
  - 如正确，但问题现象仍然存在，请联系技术支持。

## PWR指示灯红色常亮

### 问题现象

接通电源后，相机的PWR指示灯红色常亮。

### 可能原因

电源电压异常。

### 解决方案

请执行以下步骤排查问题：

1. 参考[相机硬件用户手册](#)，检查相机DC电源线是否连接正确、插头的紧固螺母是否拧紧。
  - 如相机DC电源线连接有误，请更正连接，然后再次检查相机指示灯。



- 如相机指示灯指示正常情况，故障排查结束。
  - 如仍指示异常情况，请执行步骤2。
  - 如相机DC电源线连接无误，请执行步骤2。
2. 断开相机DC电源线与相机的连接，并使用万用表检查航插的电压是否达到24V。
- 如未达到24V，请检查供电线路中其他线缆的连接等，并做出相应调整。然后，将相机DC电源线重新连接至相机，并检查相机指示灯。
    - 如相机指示灯指示正常情况，故障排查结束。
    - 如仍指示异常情况，请联系技术支持。

## PWR指示灯绿色常亮，LINK指示灯不亮

### 问题现象

接通电源后，相机的PWR指示灯绿色常亮，LINK指示灯不亮。

### 可能原因

网络连接异常。

### 解决方案

请执行以下步骤排查问题：

1. 参考[相机硬件用户手册](#)，检查相机网线是否连接正确、插头的紧固螺母是否拧紧。
  - 如相机网线连接有误，请更正连接，然后再次检查相机指示灯。
    - 如相机指示灯指示正常情况，故障排查结束。
    - 如仍指示异常情况，请执行步骤2。
  - 如相机网线连接无误，请执行步骤2。
2. 更换相机网线，然后再次检查相机指示灯。
  - 如相机指示灯指示正常情况，故障排查结束。
  - 如仍指示异常情况，请联系技术支持。

### 相关问题

以下问题可能由相同的原因导致。如同时遇到这些问题，在解决本章涉及的问题后，可确认其他问题是否也已解决。

- [相机不投光，且未生成深度图](#)
- [图像采集时间过长](#)
- [相机偶发连接断开](#)

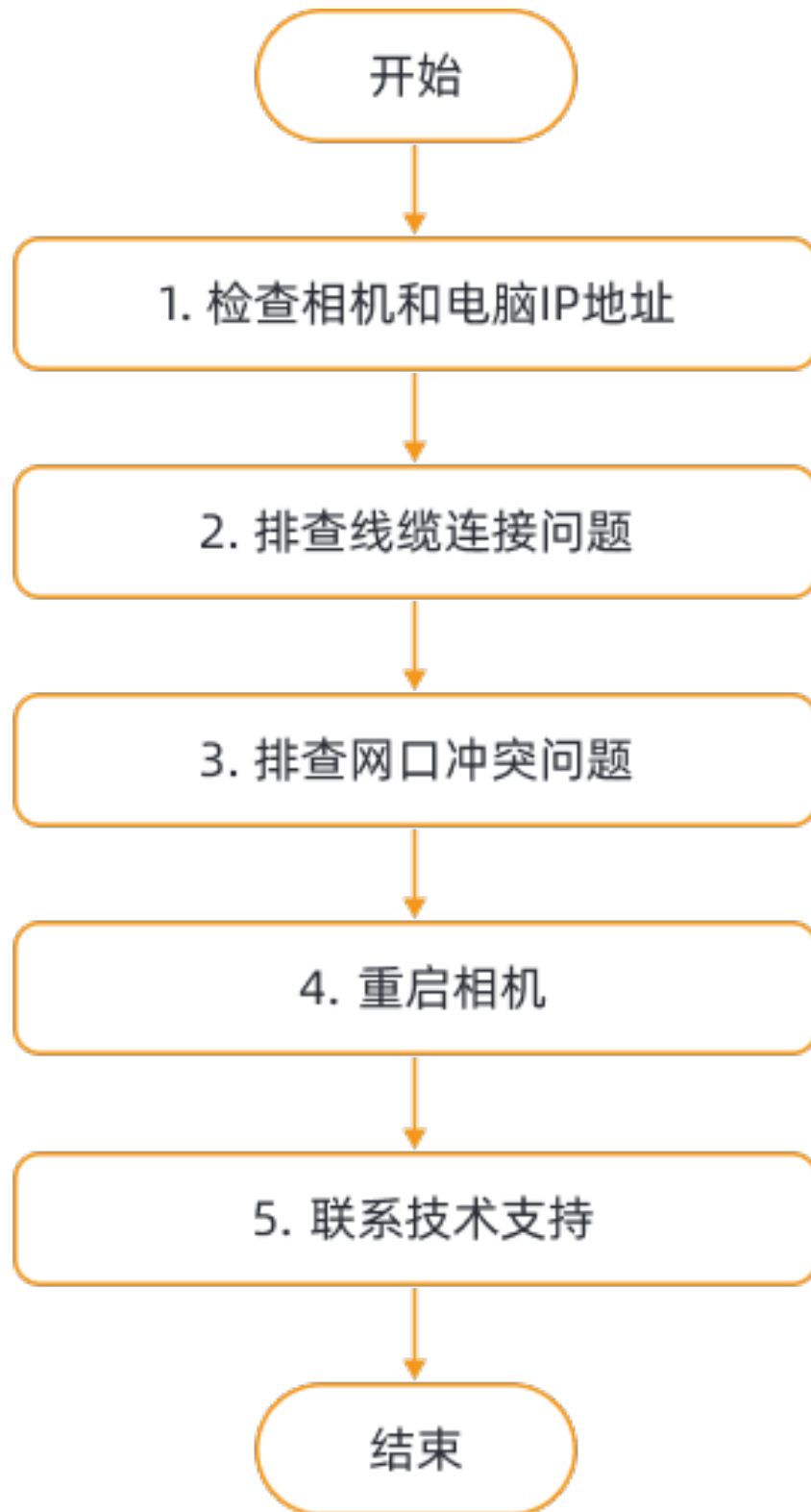
### 10.1.2. Mech-Eye SDK搜索不到相机

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。

#### 问题原因

- 相机和电脑的IP地址冲突
- 相机、电脑、交换机线缆连接异常
- 网口冲突
- 其他

#### 排查思路



Mech-Eye SDK搜索不到相机时的排查思路为：

1. 如Mech-Eye SDK仅能搜索到一台相机，**检查该相机和电脑的IP地址。**
2. 其次，**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。

3. 排查网口冲突问题。
4. 如解决网口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。

## 排查方法

### 检查相机和电脑的IP地址

如电脑同时连接了多台相机，但Mech-Eye SDK仅能搜索到一台相机，请执行以下步骤排查问题：

1. 检查以下两个IP地址是否相同：
  - 该相机的IP地址
  - 与该相机相连的电脑网口的IP地址
2. 如两个IP地址相同，需修改其一：
  - 修改相机的IP地址：使用[Mech-Eye Viewer](#)或[IP配置工具](#)修改。
  - 修改电脑网口的IP地址：请参考[设置电脑IP地址](#)。
3. 运行Mech-Eye SDK，重新搜索相机。
  - 如能搜索到全部相机，故障处理结束。
  - 如仍无法搜索到全部相机，请查看下一节。

### 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 <a href="#">功能示意图</a> 。
检查电脑的线缆连接	网线	网口指示灯常亮
检查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



- 相机与电脑直接连接时，不需要检查交换机的连接。
- 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、电脑的连接。通常，该现象是网线连接不良导致的。

### 排查网口冲突问题

#### 排查步骤：

1. 禁用电脑中连接相机网口外的其它网口。

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。

选中网口，右键单击，选择**禁用**以禁用该网口。

2. 运行Mech-Eye SDK，重新搜索相机。
3. 如成功搜索到相机，则为网口冲突。启用其他网口，并将其设置到不同网段。



如无法搜索到相机，请尝试重启相机网口；如重启网口后仍无法搜索到，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。

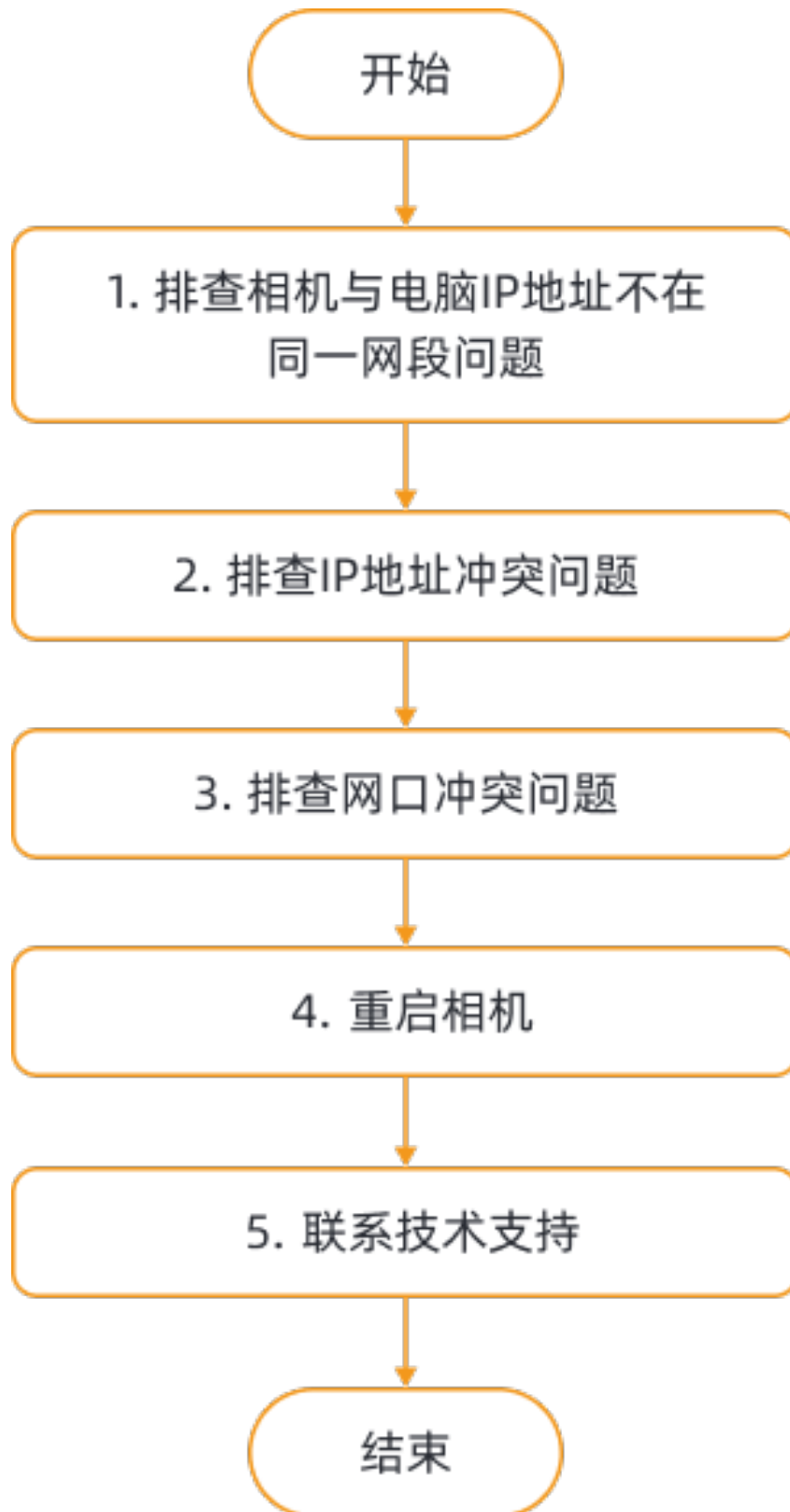
### 10.1.3. Mech-Eye SDK无法连接相机

使用Mech-Eye SDK连接相机时，无法连接搜索到的相机。本章将介绍问题的可能原因，排查问题的思路及排查方法。

#### 问题原因

- 相机与电脑IP地址不在同一网段
- IP地址冲突
- 网口冲突
- 其他

#### 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. **首先排查相机与电脑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. 排查网口冲突问题。
4. 如解决网口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。

## 排查方法

### 排查相机与电脑IP地址不在同一网段问题

#### 排查步骤：

1. 查看电脑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. 确保相机与电脑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地址与电脑在同一网段。
4. 运行Mech-Eye SDK，重新扫描并连接相机。

### 排查IP地址冲突问题

电脑与交换机连接的设备中，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. 在电脑上，仅保留与相机连接的网线，拔掉电脑和交换机的其他网线。
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。

### 排查网口冲突问题

#### 排查步骤：

1. 禁用电脑中连接相机网口外的其它网口。

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。  
选中网口，右键单击，选择**禁用**以禁用该网口。

2. 运行Mech-Eye SDK，重新连接相机。
3. 如成功连接到相机，则为网口冲突。启用其他网口，并将其设置到不同网段。



如无法连接相机，请尝试重启相机网口；如重启网口后仍无法连接，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

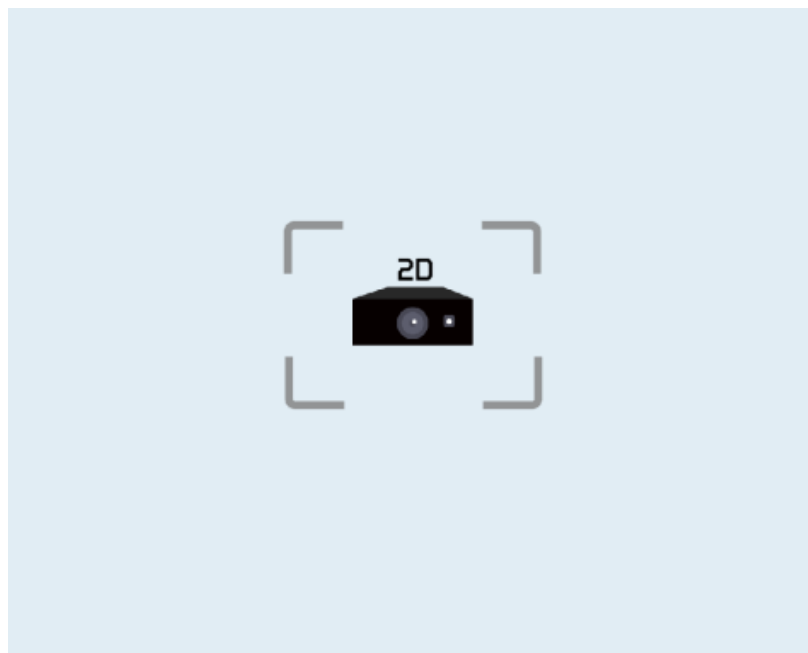
如以上步骤全部尝试后，仍无法连接相机，请联系技术支持。

### 10.1.4. 相机不投光，且未生成深度图

#### 问题现象

通过Mech-Eye SDK控制相机采集图像后，相机未投光，且软件未收到深度图。

- Mech-Eye Viewer：深度图显示界面仅显示如下的图像。



- Mech-Eye API：调用采集图像的方法后返回错误码-6。
- GenICam接口：可能发生超时错误。

#### 可能原因

- 相机线缆连接不正确。
- 导轨电源的DC电源线过长。



- 电源电压不足。

## 解决方案



本节内容仅适用于Windows系统。如果你使用的是Ubuntu系统，请联系技术支持。

1. 根据问题发生的时间，在Mech-Eye Viewer中[打开问题发生前最新生成的日志](#)。
2. 在该日志中，根据左侧的时间，找到问题发生时记录的日志信息。根据信息中的错误码，查看下方对应的解决方案。

### 错误码0x0105、0x0107或0x0109

请执行以下步骤排查问题：

1. 参考[相机硬件用户手册](#)和[相机线缆敷设规范](#)，检查相机线缆是否正确连接和敷设，没有插头紧固螺母未拧紧、与其他线缆缠绕等问题。
  - 如相机线缆未正确连接和敷设，请更正后确认问题现象是否仍存在。
    - 如不存在，故障排查结束。
    - 如仍存在，请执行步骤2。
  - 如相机线缆已正确连接和敷设，请执行步骤2。
2. 根据相机的供电方式，执行相应的步骤：
  - 如直接使用导轨电源或电源适配器为相机供电，请执行步骤3。
  - 如通过自动导向车（AGV）为相机供电，请执行步骤4。
3. 检查电源是否正确接地：如使用导轨电源为相机供电，参考[相机硬件用户手册](#)，确认导轨电源的PE线连接是否正确。如使用电源适配器为相机供电，请检查适配器AC电源线的三相插头连接的插座是否正确接地。
  - 如不正确，请更正接地，然后确认问题现象是否仍存在。
    - 如不存在，故障排查结束。
    - 如仍存在，请联系技术支持。
  - 如正确，但问题现象仍然存在，请联系技术支持。
4. 确认AGV是否使用了延时继电器以保证电压稳定。
  - 如未使用，请增加延时继电器，然后确认问题现象是否仍存在。
    - 如不存在，故障排查结束。
    - 如仍存在，请联系技术支持。
  - 如已使用，请联系技术支持。

### 错误码0x020c

请执行以下步骤排查问题：

确认导轨电源的DC电源线长度是否超过20米。

- 如超过，请调节导轨电源的电压，使其提供给相机的电压升高2V。
- 如未超过，请联系技术支持。

### 其他错误码

请联系技术支持。

### 相关问题

以下问题可能由相同的原因导致。如同时遇到这些问题，在解决本章涉及的问题后，可确认其他问题是否也已解决。

- [接通电源后，相机指示灯指示异常情况](#)
- [图像采集时间过长](#)
- [相机偶发连接断开](#)

### 10.1.5. 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的相机信息卡片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。

#### 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。

#### 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。

### 10.1.6. 图像采集时间过长

#### 问题现象

通过Mech-Eye SDK控制相机采集图像后，图像采集时间远超正常采集时间。使用Mech-Eye Viewer连接相机并查看左上角的数据传输速度，发现传输速度低于700Mbps（NANO和Pro XS低于500Mbps）。

#### 可能原因

- 所用交换机或网线不支持千兆带宽。

- 电脑网口的驱动需更新或网口设置有误。
- 相机网线连接不正确或需更换。

## 解决方案

请执行以下步骤排查问题：

1. 确认所用的交换机或网线是否支持千兆带宽。
  - 如不支持，请更换为支持千兆带宽的交换机或网线。百兆带宽无法保证数据传输的稳定性。更换后，请确认问题现象是否仍存在。
    - 如不存在，故障排查结束。
    - 如仍存在，请执行步骤2。
  - 如支持，请执行步骤2。
2. 检查相机网线是否有磨损、断裂等问题。
  - 如有此类问题，请联系技术支持更换相机网线。然后，请确认问题现象是否仍存在。
    - 如不存在，故障排查结束。
    - 如仍存在，请执行步骤3。
  - 如无此类问题，请执行步骤3。
3. 参考[相机硬件用户手册](#)和[相机线缆敷设规范](#)，检查相机网线是否正确连接和敷设，没有插头紧固螺母未拧紧、与其他线缆缠绕等问题。
  - 如相机网线未正确连接和敷设，请更正后确认问题现象是否仍存在。
    - 如不存在，故障排查结束。
    - 如仍存在，请执行步骤4。
  - 如相机网线已正确连接和敷设，请执行步骤4。
4. 检查电脑上与相机相连的网口，确认其驱动是否需要更新，或其设置是否需要更改（例如，可能需要取消传输速度限制）。
  - 如需要，请更新驱动或更改设置后确认问题现象是否仍存在。
    - 如不存在，故障排查结束。
    - 如仍存在，请执行步骤5。
  - 如不需要，请执行步骤5。
5. 尝试更换与相机相连的网口，[为该网口设置与其他网口不同的网段](#)，或禁用其他未使用的网口。然后，请确认问题现象是否仍存在。
  - 如不存在，故障排查结束。
  - 如仍存在，请联系技术支持。

## 相关问题

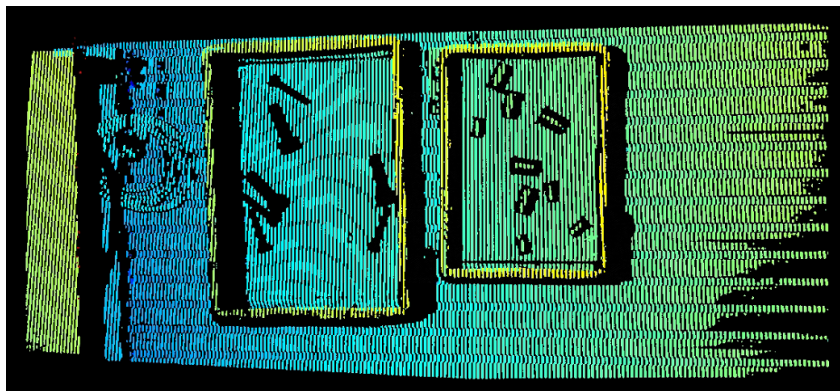
以下问题可能由相同的原因导致。如同时遇到这些问题，在解决本章涉及的问题后，可确认其他问题是否也已解决。

- [接通电源后，相机指示灯指示异常情况](#)
- [相机不投光，且未生成深度图](#)
- [相机偶发连接断开](#)

### 10.1.7. V2相机升级固件后采集的点云有缺失

#### 问题现象

Pro S Enhanced (V2) 与Pro M Enhanced (V2) 升级固件后，采集的点云有明显缺失，如下图所示。



#### 可能原因

Pro S Enhanced (V2) 与Pro M Enhanced (V2) 在升级后，相机实际型号和显示型号不一致。


#### 解决方案

请执行以下步骤排查问题：

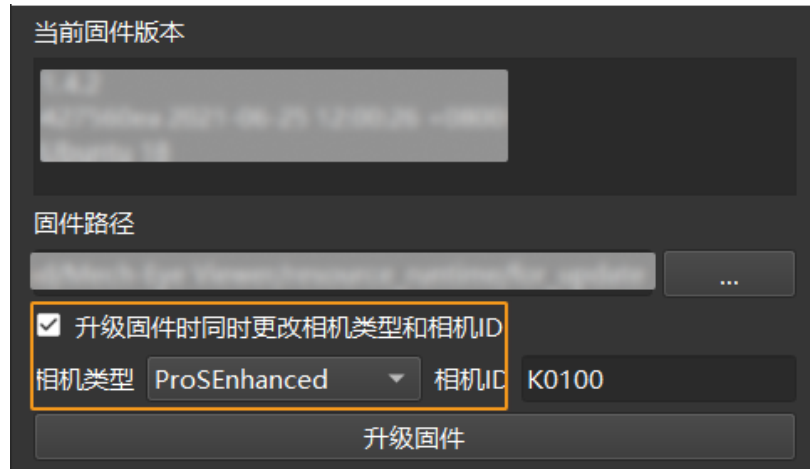
1. 根据IP地址、序列号等信息，找到Mech-Eye Viewer中显示为Log S或Log M的相机，并确认机身上粘贴的标签中的相机型号：
  - 如为 Log S 或 Log M，可直接升级固件。
  - 如为Pro S Enhanced或Pro M Enhanced，请执行下一步。



如机身上的标签不易查看，可通过[查看分辨率](#)判断相机型号。

2. 联系技术支持，获取Mech-Eye Viewer中**管理员**用户的密码。
3. 在Mech-Eye Viewer中连接相机。单击  切换到**管理员**用户，输入密码。
4. 在**工具**菜单中，选择**相机固件升级**。

5. 在弹出的窗口中，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**中选择正确的相机型号。



6. 单击[ **升级固件** ]，重新升级相机的固件。

### 通过分辨率判断相机型号

请执行以下步骤查看相机分辨率：

1. 使用Mech-Eye Viewer连接相机。
2. 在**工具**菜单中，选择**查看2D相机并设置参数**。
3. 确认相机的分辨率，并根据下表判断相机型号：

显示的型号	分辨率	实际型号
Log S	1280 × 1024	Log S
	1920 × 1200	Pro S Enhanced (V2)
Log M	1280 × 1024	Log M
	1920 × 1200	Pro M Enhanced (V2)

## 10.2. FAQ

本章提供常见疑问的解答。

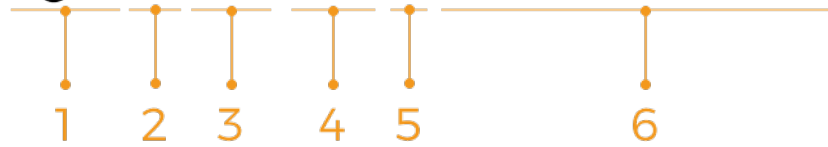
- [如何解读相机序列号？](#)

### 10.2.1. 如何解读相机序列号？

相机上粘贴的标签上印有相机序列号（SN），为每台相机的唯一识别编号。

序列号共16位，从序列号可得知相机的型号、2D相机类型、对焦距离和生产年月等信息。

# QAA30228A4030008



### 1. 第1至2位代表相机型号：

编码	型号
QA	Deep (V3) 或DEEP (V4)
NA/KA	Log M (V3) 或Log S (V3)
NE/KE	Pro M Enhanced (V3)、Pro S Enhanced (V3)、PRO S (V4) 或PRO M (V4)
TA	Nano (V3) 或NANO (V4)
RA	Pro XS (V3)
WA	Laser L (V3) 或LSR L (V4)
VA	LSR S (V4)
WU	Laser L Enhanced-12MP (V3)
QE	Pro L Enhanced (V3)
UA	UHP-140 (V4)

### 2. 第3位代表产品中包含的2D相机的类型：

编码	2D相机类型
C	彩色2D相机
M	黑白2D相机
A	同时包含黑白与彩色2D相机

### 3. 第4至5位代表相机的对焦距离：

型号	对焦距离 (mm)
UHP-140 (V4)	编码 × 10
Nano (V3) 或NANO (V4)	
Pro XS (V3)	
其他型号	编码 × 100

### 4. 第6至7位代表相机生产年份：例如，“23”代表生产年份为2023年。

### 5. 第8位代表相机生产月份：

编码	月份
1~9	1~9月

编码	月份
A	10月
B	11月
C	12月

6. 第9至16位为相机的生产编号。