

深度图像

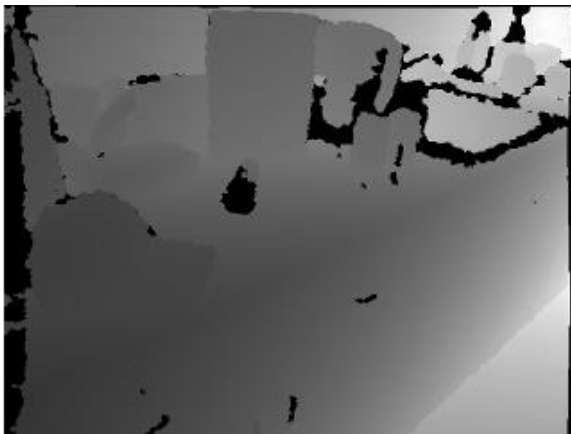
Rangeldimage

分享内容

- **深度图的基础理论**
- PCL中模块RangeImage相关类的介绍
- 从点云创建深度图
- 从深度图中提取边界
- 点云到深度图的变换与曲面重建

深度图的基础理论

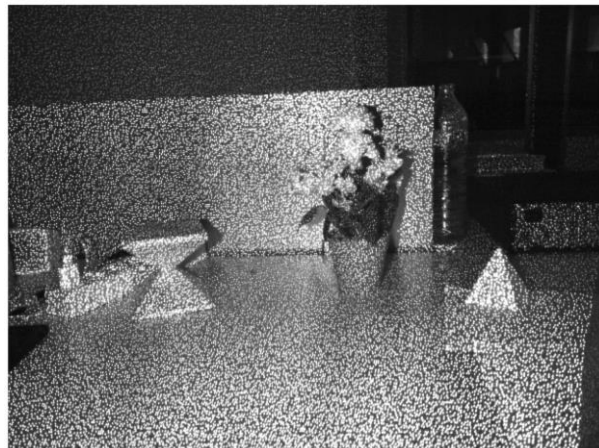
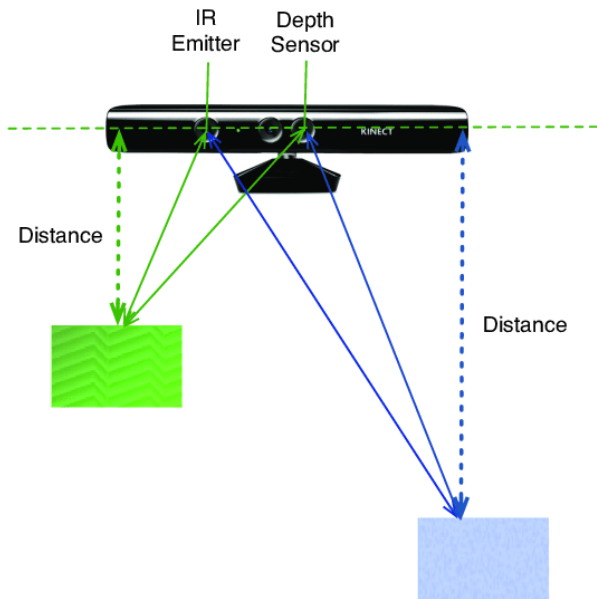
RGB-D Image = RGB Image + Depth Image



深度图的基础理论

获取方法

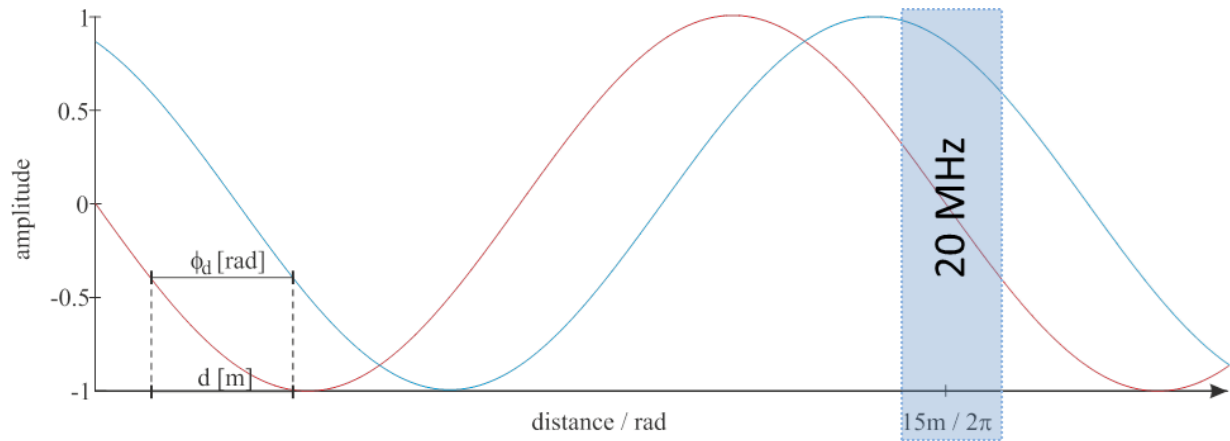
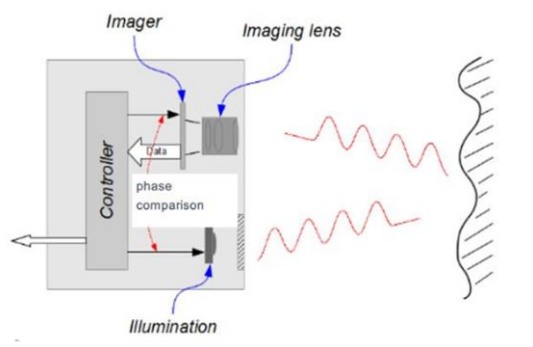
1. 结构光 (Structured Light)



深度图的基础理论

获取方法

2. 飞行时间 (Time-of-Flight)

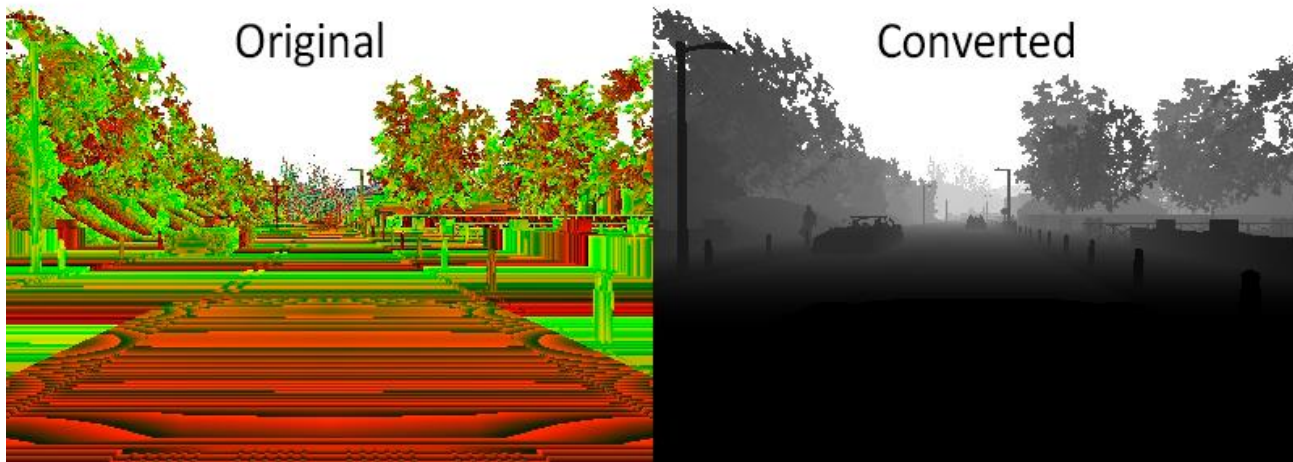


$$d = \frac{c}{f_{\text{mod}}} \cdot \frac{1}{2} \cdot \frac{\phi_d}{2\pi}$$

深度图的基础理论

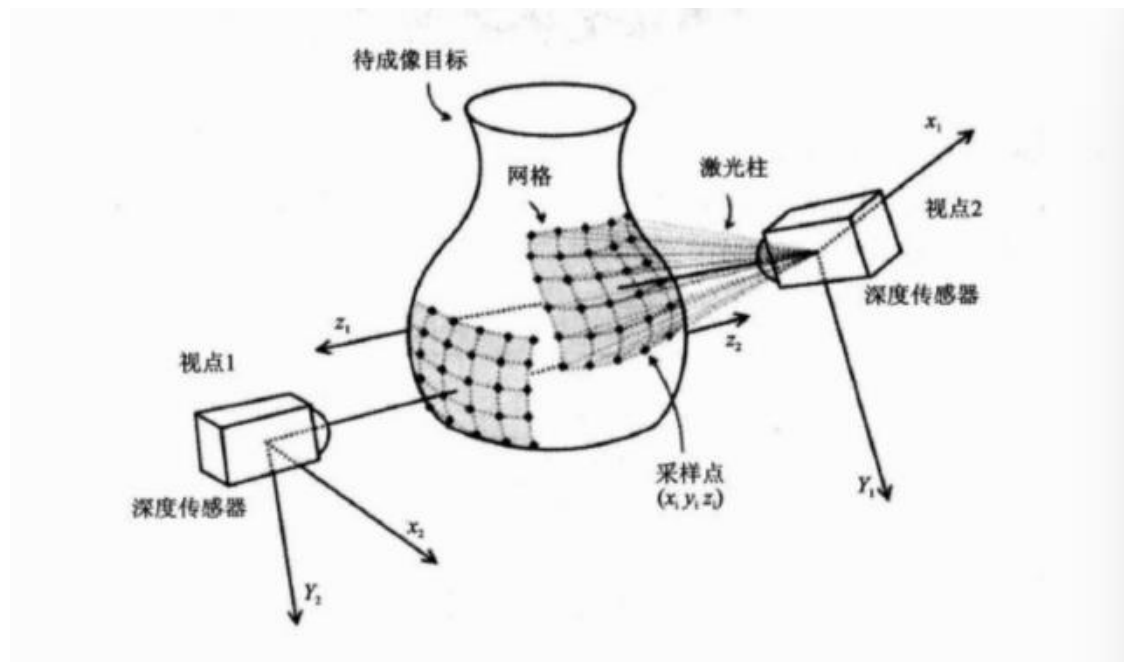
获取方法

3. 激光雷达 (Time-of-Flight)



深度图的基础理论

深度图与点云的区别



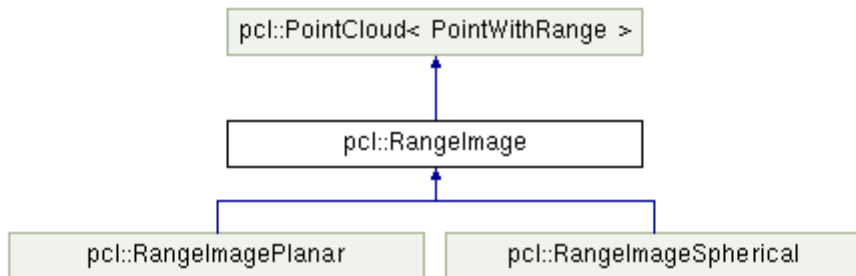
分享内容

- 深度图的基础理论
- **PCL中模块Rangemage相关类的介绍**
- 从点云创建深度图
- 从深度图中提取边界
- 点云到深度图的变换与曲面重建

PCL中Rangemage相关类

`pcl::RangeImage`

`RangeImage`类继承于`PointCloud`，主要功能是实现一个特定视点得到一个三维场景的深度图像。



`PointWithRange`:

```
pcl::PointWithRange::PointWithRange ( float _x,  
                                       float _y,  
                                       float _z,  
                                       float _range = 0.f  
                                       )
```

PCL中Rangemage相关类

Public Member Functions

```
template<typename PointCloudType >
```

void

[createFromPointCloud](#)

(const PointCloudType &point_cloud, float angular_resolution=[pcl::deg2rad](#)(0.5f), float max_angle_width=[pcl::deg2rad](#)(360.0f), float max_angle_height=[pcl::deg2rad](#)(180.0f), const Eigen::Affine3f &sensor_pose=Eigen::Affine3f::Identity(), CoordinateFrame coordinate_frame=CAMERA_FRAME, float noise_level=0.0f, float min_range=0.0f, int border_size=0)

从点云创建深度图像，
point_cloud为指向创建深度图像所需要的点云的引用，
angular_resolution为模拟的深度传感器的角度分辨率，即深度图像中一个像素对应的角度大小，
max_angle_width为模拟的深度传感器的水平最大采样角度，
max_angle_height为模拟传感器的垂直方向最大采样角度，
sensor_pose设置模拟的深度传感器的位姿是一个仿射变换矩阵，默认为4*4的单位矩阵变换，
coordinate_frame定义按照那种坐标系统的习惯默认为CAMERA_FRAME，
noise_level获取深度图像深度时，近邻点对查询点距离值的影响水平，
min_range设置最小的获取距离，小于最小获取距离的位置为传感器的盲区，
border_size获得深度图像的边缘的宽度 默认为0
该函数中涉及的角度单位都是弧度

PCL中Rangemage相关类

Public Member Functions

template<typename PointCloudType >

void [createFromPointCloudWithKnownSize](#)
(const PointCloudType &point_cloud, float angular_resolution, const Eigen::Vector3f &point_cloud_center, float point_cloud_radius, const Eigen::Affine3f &sensor_pose=Eigen::Affine3f::Identity(), [CoordinateFrame](#) coordinate_frame=[CAMERA_FRAME](#), float noise_level=0.0f, float min_range=0.0f, int border_size=0)

从点云创建深度图像，其中参数中有关场景大小的提示，提高了获取深度图像时的计算速度。

point_cloud为指向创建深度图像所需要的点云的引用，

angular_resolution为模拟的深度传感器的角度分辨率，弧度表示，

point_cloud_center为点云外接球体的中心，默认为 (0, 0, 0)

point_cloud_radius为点云外接球体的半径，

sensor_pose设置模拟的深度传感器的位姿是一个仿射变换矩阵，默认为4*4的单位矩阵变换，

coordinate_frame定义按照那种坐标系统的习惯默认为CAMERA_FRAME，

noise_level获取深度图像深度时，近邻点对查询点距离值的影响距离，以米为单位，

min_range设置最小的获取距离，小于最小获取距离的位置为传感器的盲区，

border_size获得深度图像的边缘的宽度 默认为0

该函数中涉及的角度单位都是弧度

PCL中Rangemage相关类

Public Member Functions	
template<typename PointCloudTypeWithViewpoints>	
void	<div><div>createFromPointCloudWithViewpoints</div><div>(const PointCloudTypeWithViewpoints &point_cloud, float angular_resolution, float max_angle_width, float max_angle_height, CoordinateFrame coordinate_frame=CAMERA_FRAME, float noise_level=0.0f, float min_range=0.0f, int border_size=0)</div></div>
	<div>从点云创建深度图像，点云中包含深度信息，其中， point_cloud为指向创建深度图像所需要的点云的引用， angular_resolution为模拟的深度传感器的角度分辨率，即深度图像中一个像素对应的角度大小， max_angle_width为模拟的深度传感器的水平最大采样角度， max_angle_height为模拟传感器的垂直方向最大采样角度， sensor_pose: Average viewpoints as sensor pose, 无需设置sensor pose coordinate_frame定义按照那种坐标系统的习惯默认为CAMERA_FRAME， noise_level获取深度图像深度时，近邻点对查询点距离值的影响水平，如果该值比较小，则常用Z-缓冲区中深度平均值作为查询点的深度， min_range设置最小的可视深度，小于最小获取距离的位置为传感器的盲区， border_size获得深度图像的边缘的宽度 默认为0 该函数中涉及的角度单位都是弧度</div>

PCL中Rangemage相关类

Public Member Functions	
template<typename PointCloudTypeWithViewpoints>	
void	<div><div><div>createEmpty</div></div><div>(float angular_resolution, const Eigen::Affine3f &sensor_pose=Eigen::Affine3f::Identity(), Rangemage::CoordinateFrame coordinate_frame=CAMERA_FRAME, float angle_width=pcl::deg2rad(360.0f), float angle_height=pcl::deg2rad(180.0f))</div></div>
	<div>创建一个空的深度图像，以当前视点不可见点填充，其中， angle_width为模拟的深度传感器的水平采样角度，默认为PI*2（360）； angle_height垂直方向的采样角度默认为PI（180） *****其他参数同上</div>

PCL中Rangemage相关类

Public Member Functions

template<typename PointCloudType>

void	integrateFarRanges (const PointCloudType &far_ranges)
------	---

	将已有的远距离测量结果融合到深度图像中 （向深度图像中添加远距离测量点？）
--	---------------------------------------

PCL中Rangemage相关类

Public Member Functions

template<typename PointCloudType>

void [cropImage](#)(int border_size=0, int top=-1, int right=-1, int bottom=-1, int left=-1)

裁剪深度图像到最小尺寸，使这个最小尺寸包含所有点云，
其中，board_size设置裁剪后深度图像的边界尺寸， top为裁剪框的边界
*****默认都为-1

PCL中Rangemage相关类

Public Member Functions

template<typename PointCloudType>

void	<u>setTransformationToRangeImageSystem</u> (const Eigen::Affine3f &to_range_image_system)
------	---

	设置从深度图像坐标系（传感器的坐标系）转换到世界坐标系的变换矩阵
--	----------------------------------

PCL中Rangemage相关类

Public Member Functions

template<typename PointCloudType>

float [getAngularResolution\(\)](#) const

获得深度图像X和Y方向的角分辨率 弧度表示

PCL中Rangemage相关类

Public Member Functions

template<typename PointCloudType>

void [setAngularResolution](#)(float angular_resolution)

设置深度图像在X方向和Y方向的新的角分辨率， angular_resolution即每个像素所对应的弧度

PCL中Rangemage相关类

Public Member Functions

template<typename PointCloudType>

void	calculate3DPoint (float image_x, float image_y, float range, PointWithRange &point) const
------	--

	根据深度图像点（X Y）和距离（range）计算返回场景中的3D点的point
--	---

PCL中Rangemage相关类

Public Member Functions

template<typename PointCloudType>

void	calculate3DPoint (float image_x, float image_y, PointWithRange &point) const
------	---

	根据给定的深度图像点和离该点最近像素上的距离值计算返回场景中的3D点point
--	---

分享内容

- 深度图的基础理论
- PCL中模块Rangemage相关类的介绍
- **从点云创建深度图**
- 从深度图中提取边界
- 点云到深度图的变换与曲面重建

从点云创建深度图

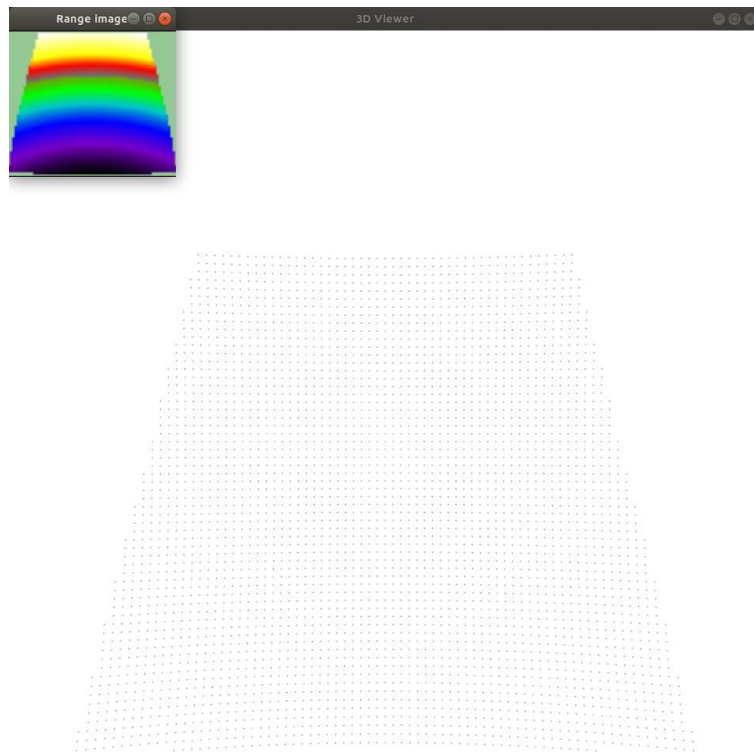
函数：通过pointCloud点云创建深度图

```
- createFromPointCloud(pointCloud,angularResolution,maxAngleWidth,maxAngleHeight,  
sensorPose,coordinate_frame,noiseLevel,minRange,borderSize)
```

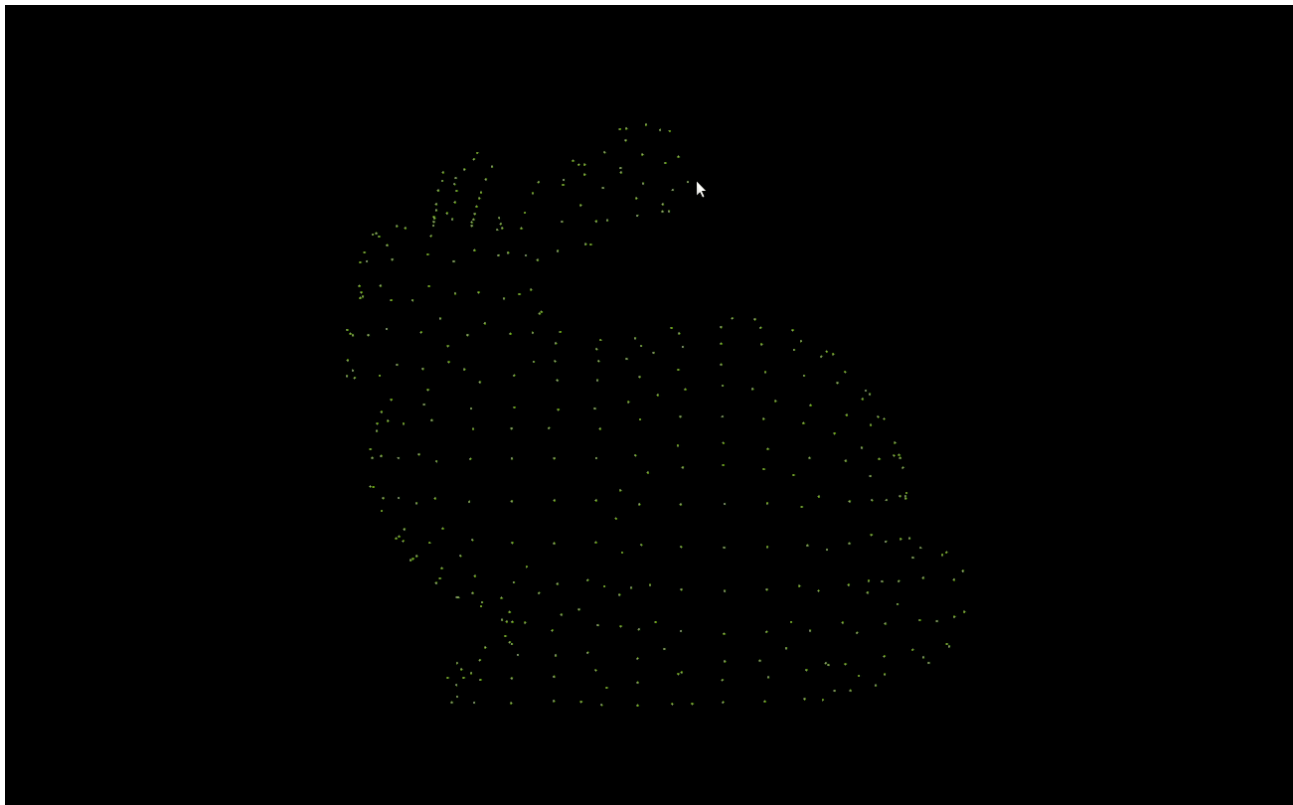
参数：

- **pointCloud**：被检测点云
- **angularResolution=1**：邻近的像素点所对应的每个光束之间相差 1°
- **maxAngleWidth=360**：进行模拟的距离传感器对周围的环境拥有一个完整的 360° 视角，无论任何数据集都推荐使用此设置，因为最终获取的深度图像将被裁剪到有空间物体存在的区域范围
- **maxAngleHeight=180**：当传感器后面没有可以观测的点时，设置一个水平视角为 180° 的激光扫描仪即可，因为需要观察距离传感器前面就可以了。
- **sensorPose**：定义了模拟深度图像获取传感器的6DOF（6自由度）位置，其原始值为横滚角roll、俯仰角 pitch、偏航角 yaw 都为 0
- **coordinate_frame**：设置为CAMERA_FRAME说明系统的X轴是向右的、Y轴是向下的、Z轴是向前的，另外参数值是LASER_FRAME，其X轴向前、Y轴向左、Z轴向上
- **noiseLevel=0**：是指使用一个归一化的 Z缓存区来创建深度图像，如果想让邻近点集都落在同一个像素单元，可以设置一个较高的值，例如 noiseLevel = 0.05 可以理解为深度距离值是通过查询点半径为 5cm 的圆内包含的点用来平均计算而得到的。
- **minRange=0**：如果设置>0则所有模拟器所在位置半径 minRange 内的邻近点都将被忽略，即为盲区。
- **borderSize=1**：如果设置>0,在裁剪图像时，将在图像周围留下当前视点不可见点的边界。

从点云创建深度图



从点云创建深度图



从点云创建深度图

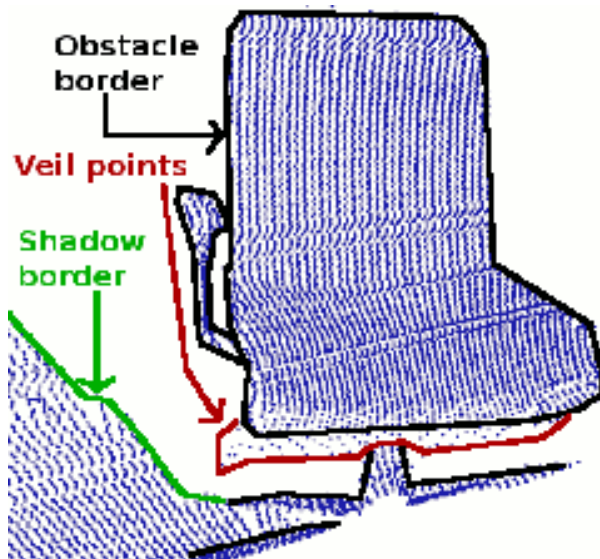


分享内容

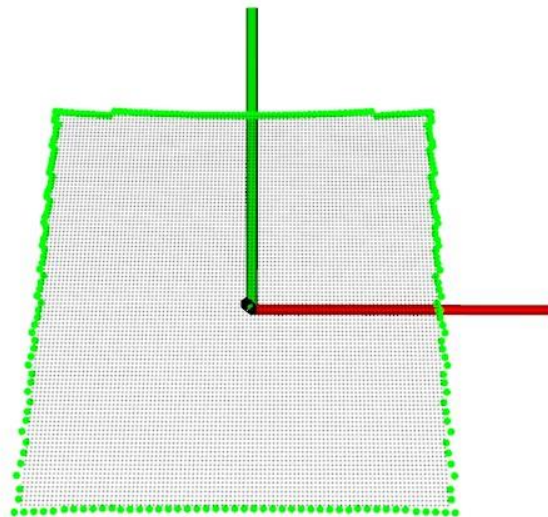
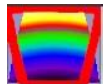
- 深度图的基础理论
- PCL中模块Rangemage相关类的介绍
- 从点云创建深度图
- **从深度图中提取边界**
- 点云到深度图的变换与曲面重建

从深度图中提取边界

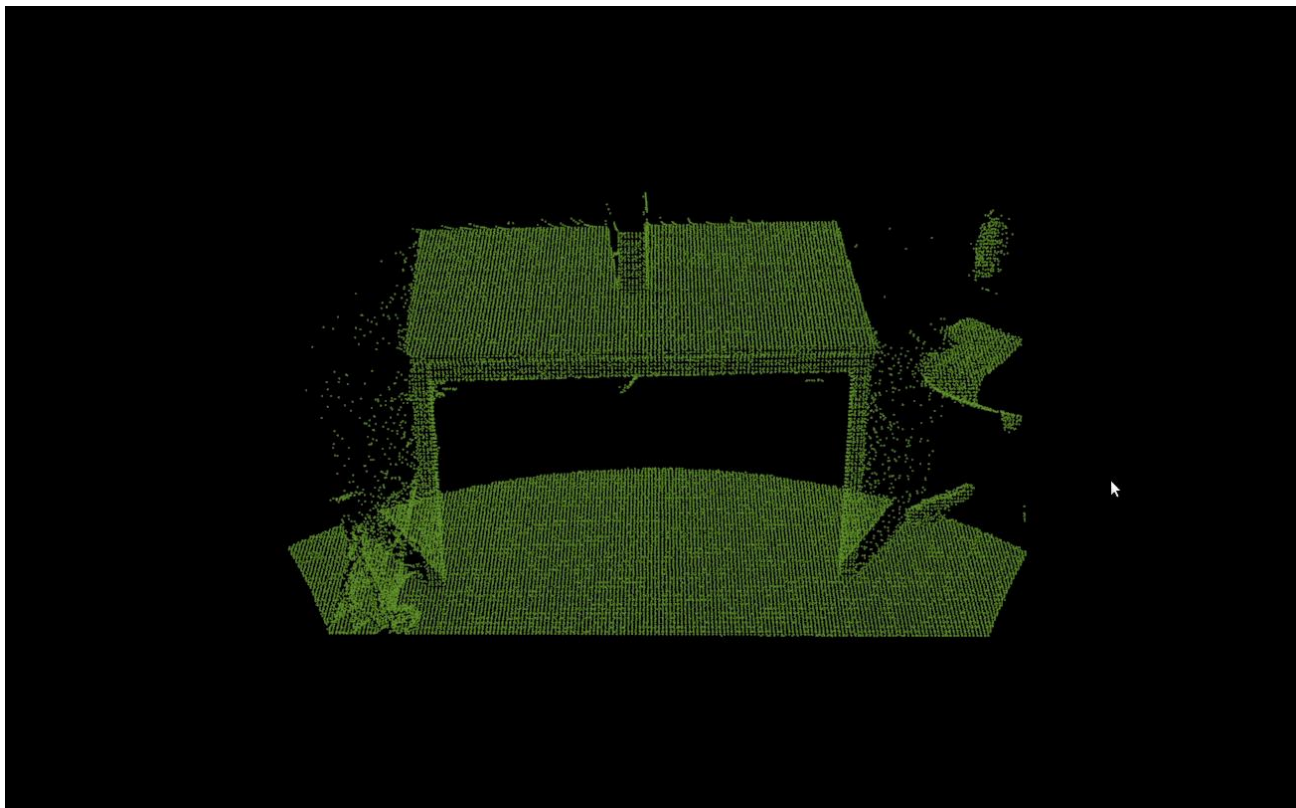
1. obstacle border : 对象边界 (属于对象的最外面的可见点)
2. shadow border : 阴影边界 (在背景中与遮挡物相邻的点)
3. Veil points : 面纱点集 (对象边界边界与阴影边界之间的内插点)



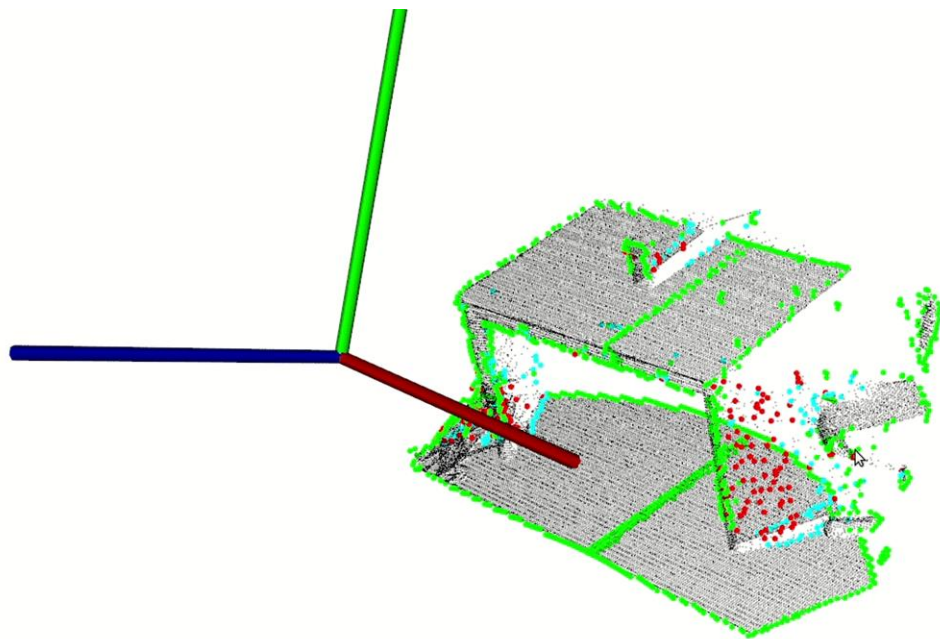
从深度图中提取边界



从深度图中提取边界



从深度图中提取边界



分享内容

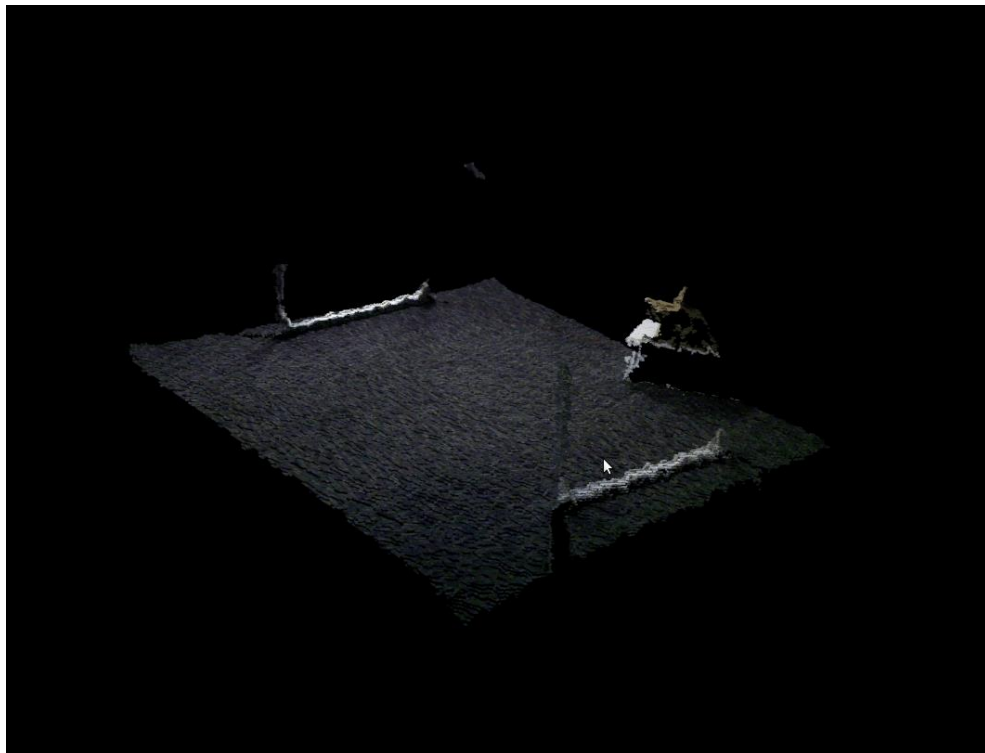
- 深度图的基础理论
- PCL中模块Rangemage相关类的介绍
- 从点云创建深度图
- 从深度图中提取边界
- **点云到深度图的变换与曲面重建**

点云到深度图的变换与曲面重建

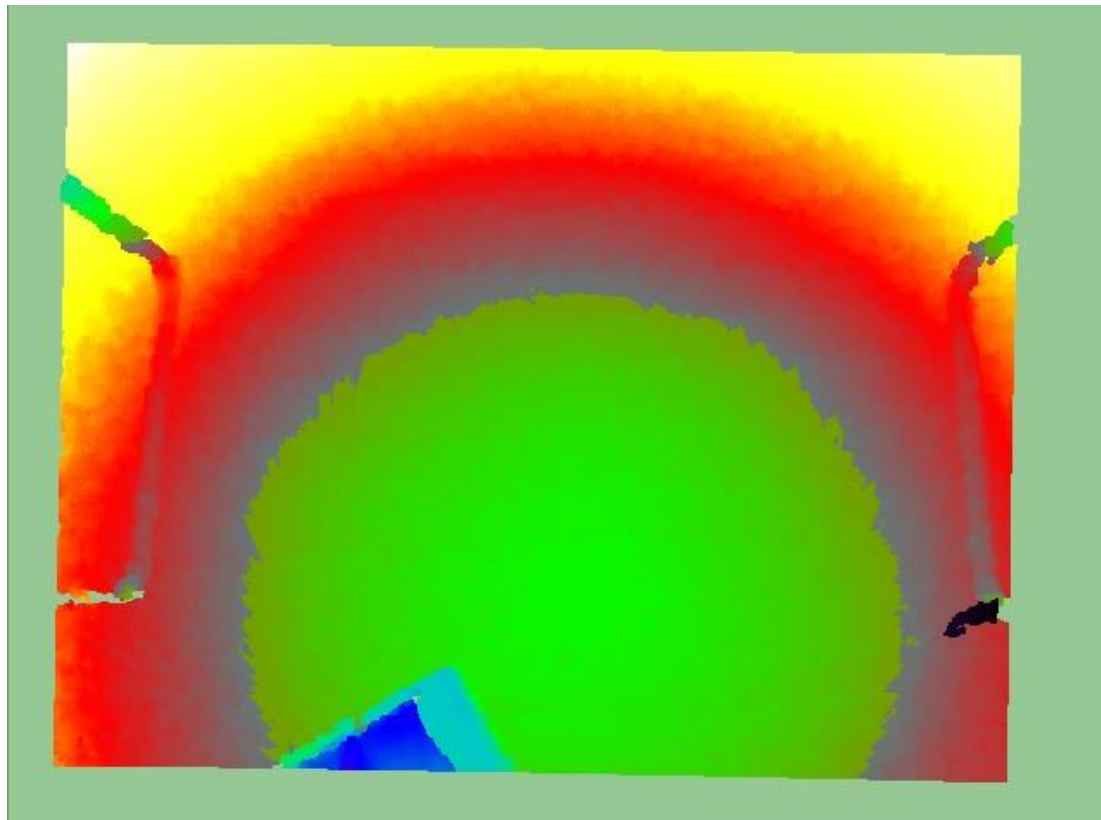
问题描述：有时候我们获取的点云数据是从一个视点获取的，为了使用深度图相关的计算方法，我们需要将点云数据转换为深度图。

内容：将点云转换为深度图像，进而使用PCL内部只适用于深度图的算法来进行曲面重建等。

点云到深度图的变换与曲面重建



点云到深度图的变换与曲面重建



点云到深度图的变换与曲面重建

