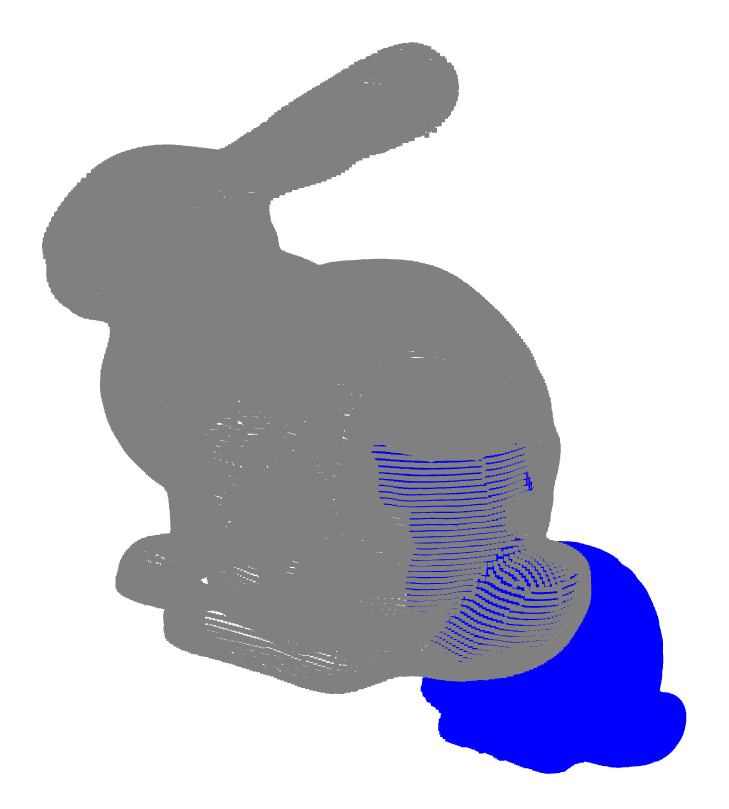
**三维视觉与理解课程**

**实验二报告【框架】**

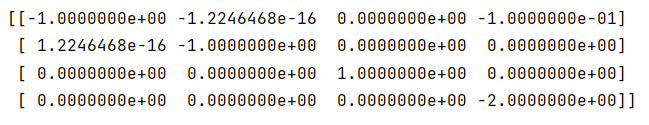
**1、点云的平移和旋转**

**旋转平移关键代码：**

|  |
| --- |
| Rotate = pcd\_affine.get\_rotation\_matrix\_from\_xyz(rotation=[0, 0, np.pi]) *# z轴顺时针转180°* Translate = np.array([-0.1, 0, 0]).reshape(3, 1) *# x轴平移0.1(向右)* |

**平移旋转后对比结果：**

**旋转矩阵：**

****

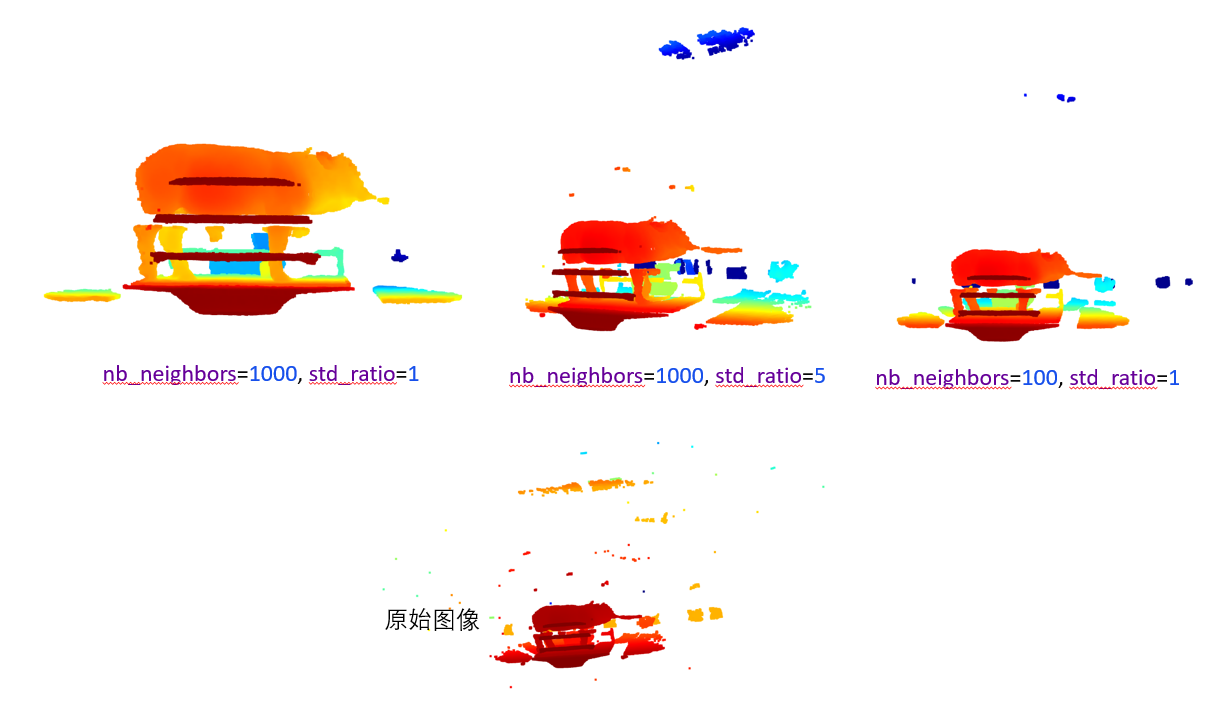
**2、点云滤波**

**滤波过程概述：(含关键参数、中间过程截图，应包含三种滤波方法)**

1. **统计学滤波**

**参数含义：**

**Nb\_neighbors 计算平均距离的邻域点数量 Std\_ratio:平均距离标准差阈值，越小滤波效果越明显.**

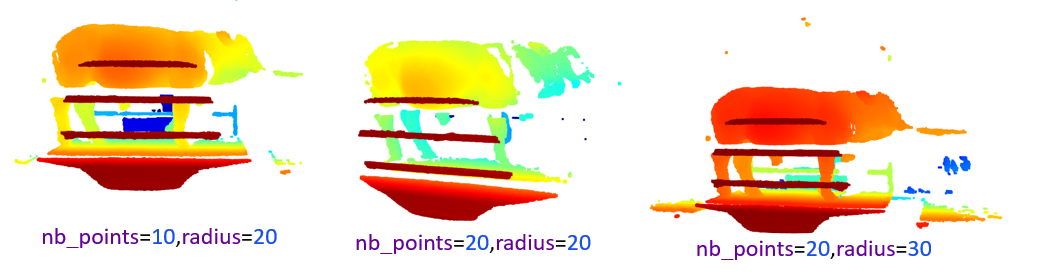
****

**最佳参数1：1000，参数2：1**

**效果：去掉了大部分离散的点，如上图最左边所示**

1. **半径滤波**

**参数含义： nb\_points：如果在指定半径范围内的邻居点数量少于 此值，则当前点将被标记为离群点。 radius：指定用于确定邻居点的搜索半径。**

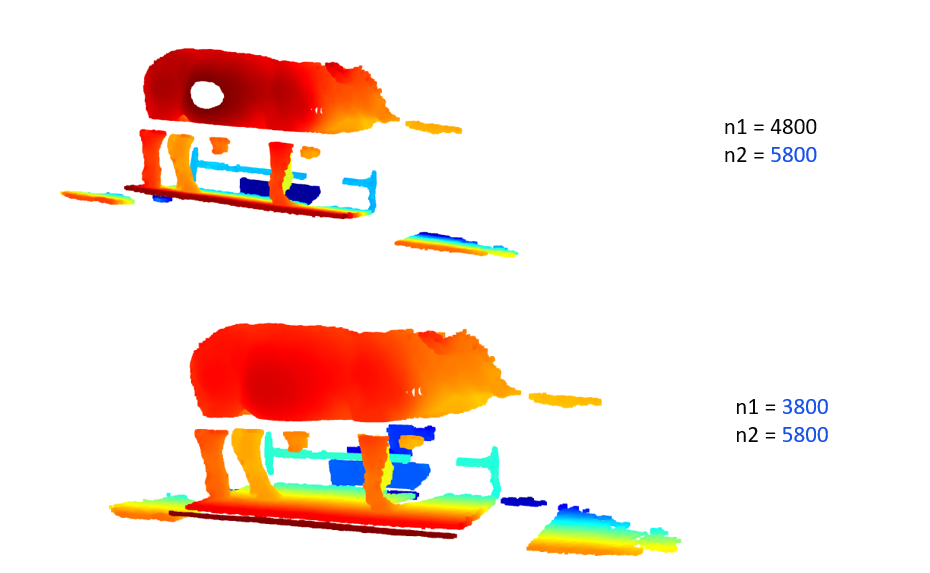
****

**效果：通过半径滤波，过滤掉了大部分不集中的点，左图效果较好**

1. **直通滤波**

|  |
| --- |
| *# 直通滤波* def pass\_through\_filter(pcd, axis, axis\_min, axis\_max):  *# 获取点云数据的Numpy数组* points = np.asarray(pcd.points)   *# 根据指定的坐标范围筛选点云* if axis == 'x':  mask = (points[:, 0] > axis\_min) & (points[:, 0] < axis\_max)  elif axis == 'y':  mask = (points[:, 1] > axis\_min) & (points[:, 1] < axis\_max)  elif axis == 'z':  mask = (points[:, 2] > axis\_min) & (points[:, 2] < axis\_max)  else:  raise ValueError("Invalid axis. Axis must be 'x', 'y', or 'z'.") *# 创建新的点云对象* filtered\_pcd = o3d.geometry.PointCloud()  filtered\_pcd.points = o3d.utility.Vector3dVector(points[mask])   *# 如果点云包含颜色信息，同样进行筛选* if pcd.has\_colors():  colors = np.asarray(pcd.colors)  filtered\_pcd.colors = o3d.utility.Vector3dVector(colors[mask])   *# 如果点云包含法线信息，同样进行筛选* if pcd.has\_normals():  normals = np.asarray(pcd.normals)  filtered\_pcd.normals = o3d.utility.Vector3dVector(normals[mask])   return filtered\_pcd |

**最终滤波结果截图：**

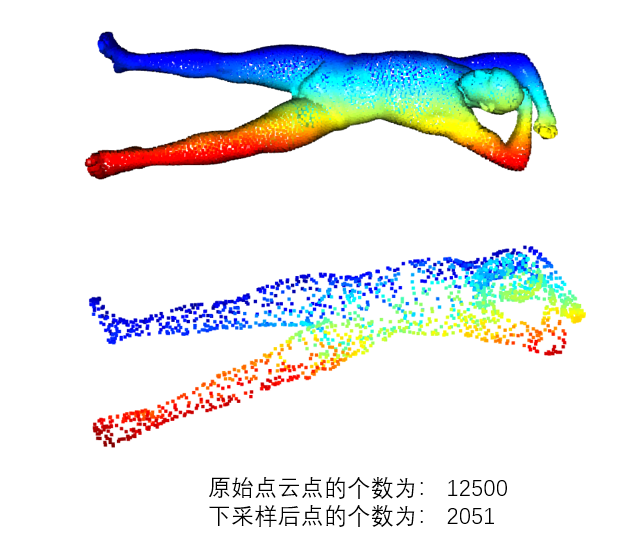
****

**另请附Q2.pcd文件。**

****

**3、点云的抽稀**

**抽稀前后对比截图：**

****

**简要描述曲率抽稀原理：**

**通过保留高曲率区域更多的点来减少点云数据的数量，同时尽量保持点云的几何特征。**

**大致过程：**

**· 加载点云数据。**

**· 计算每个点的法向量。**

**· 基于法向量计算每个点的曲率。**

**· 根据曲率将点云分为高曲率区域和低曲率区域。**

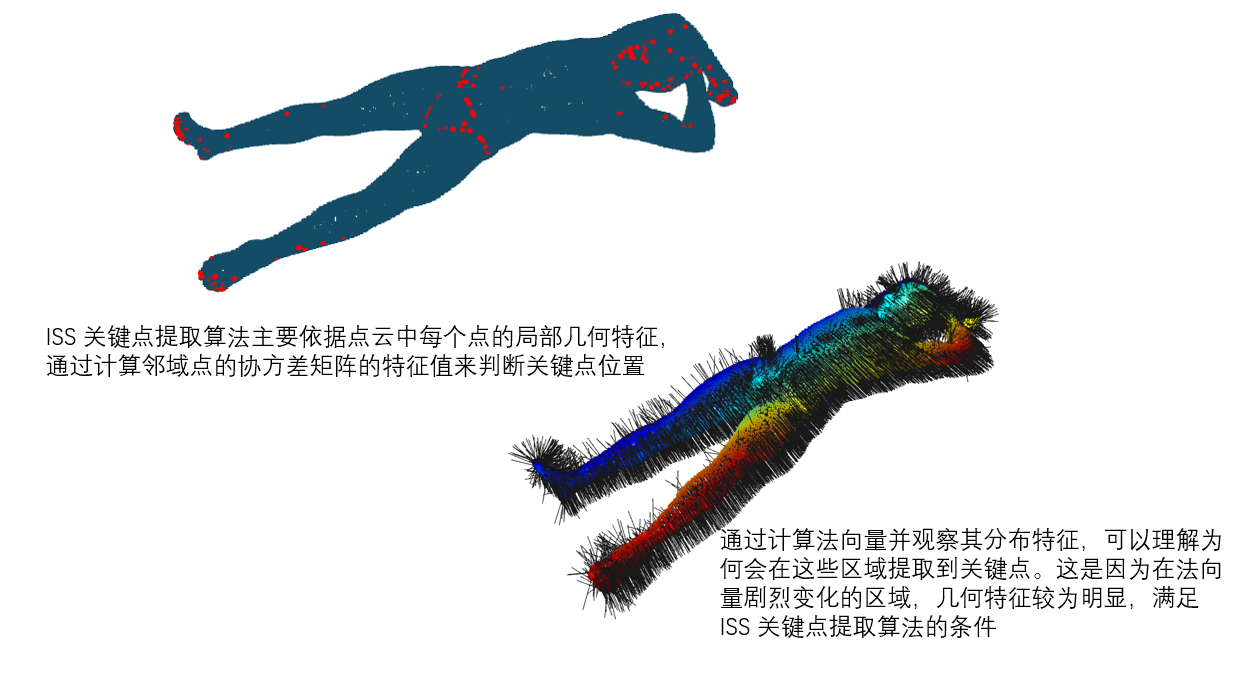
**· 分别对高曲率和低曲率区域进行均匀下采样。**

**· 合并下采样后的点云并进行可视化**

1. **点云关键点提取**
2. **使用代码 key\_point.py 提取点云 body.pcd 中的 ISS 关键点，将结果图进行截图，并说明提取到 了几个关键点**

****

1. **使用 Normal.py 程序，为 body.pcd 点云中的 所有点添加法向量。基于法向量的分布特征，简要解 释在问题(1)中为何会提取到这些关键点？**

****