

三维视觉与理解课程

实验三报告【框架】

1、手动点云分割与提取

(1) 记录三个点的选择过程，含截图

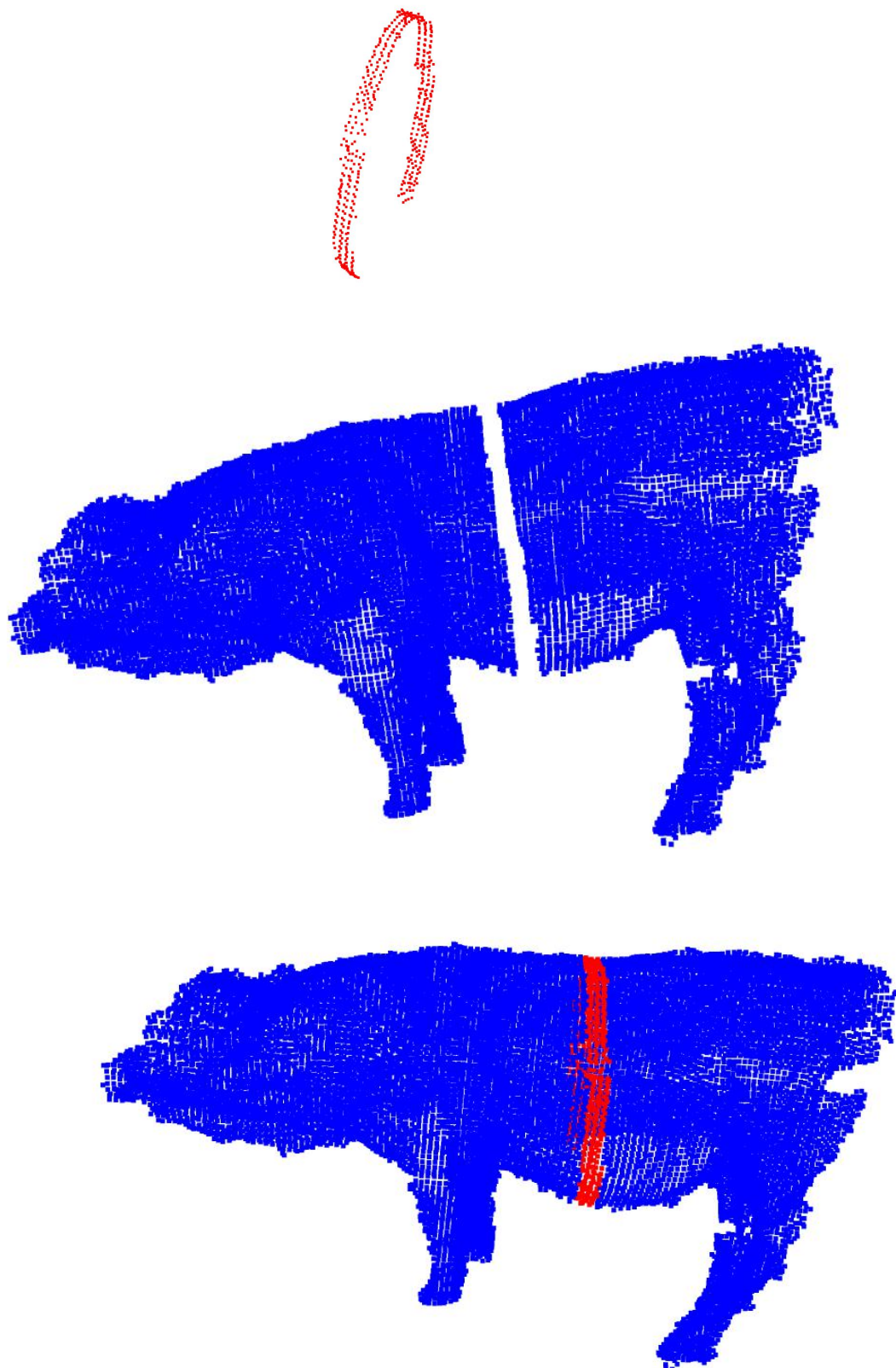
```
[Open3D INFO] Picked point #5566 (26., -1.3e+02, -1.6e+02) to add in queue.  
[Open3D INFO] Picked point #1713 (-5.5e+02, -2.8e+02, -2.1e+02) to add in queue.  
[Open3D INFO] Picked point #16046 (-52., 2.4e+02, -1.5e+02) to add in queue.
```

(2) 平面切割的关键代码说明

如果在平面之间则加入

```
if __name__ == '__main__':  
    pcd=o3d.io.read_point_cloud("cow2.pcd")  
    #点个数  
    num_points = len(pcd.points)  
    #计算平面参数  
    A,B,C,D = computePlane(pcd.points[8454], pcd.points[3783], pcd.points[2385])  
    #两个平面  
    d1=D+0.1  
    d2=D-0.1  
    idx=[]  
    for i,point in enumerate(pcd.points):  
        #处在两个平面之间  $(Ax+By+Cz+D1) * (Ax+By+Cz+D2) < 0$   
        #请补充判定条件与操作  
        x, y, z = point  
        if (A * x + B * y + C * z + d1) * (A * x + B * y + C * z + d2) < 0:idx.append(i)  
        #请补充判定条件与操作  
    #提取切片的点  
    slice= pcd.select_by_index(idx)  
    slice.paint_uniform_color([1,0,0])  
    #提取剩余的点  
    rest_points= pcd.select_by_index(idx,invert=True)  
    rest_points.paint_uniform_color([0,0,1])  
    o3d.visualization.draw_geometries([slice,rest_points])  
    o3d.io.write_point_cloud("4.pcd",slice)  
    o3d.io.write_point_cloud("5.pcd",rest_points)
```

(3) 牛腹部点云截图、切除腹部后剩余牛体点云截图，及相应的 pcd 文件



(4) 提交全部代码文件

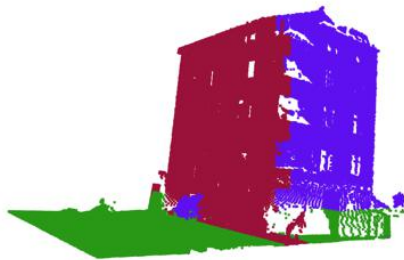
2、自动点云分割与提取

(1) 基于 RANSAC 的分割

提供使用的参数和分割结果截图放入实验报告，并简要说明你对参数选择的任何理解或想法。



min_num = 9000
distance_threshold = 1.2



min_num = 6000
distance_threshold = 1.2



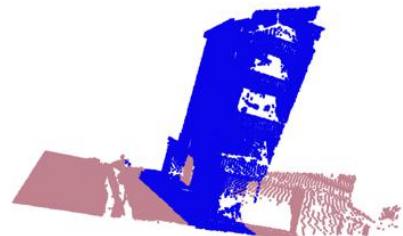
min_num = 3000
distance_threshold = 0.5



min_num = 8200
distance_threshold = 1.2



min_num = 8200
distance_threshold = 1.6



min_num = 8200
distance_threshold = 2.4

`min_num` 参数代表每个分割平面所需的最小点数，当这个值过小 (< 3000) 容易检测到多个面

`distance_threshold` 参数表示 Ransac 分割的距离阈值，当阈值过小 (< 1)，越难找到一个面，当阈值过大 (> 5) 容易把本不是一个面识别成一个面。

(2) 基于 DBSCAN 的分割 【自动分割】

提供使用的参数和分割结果截图放入实验报告，并简要说明你对参数选择的任何理解或想法。



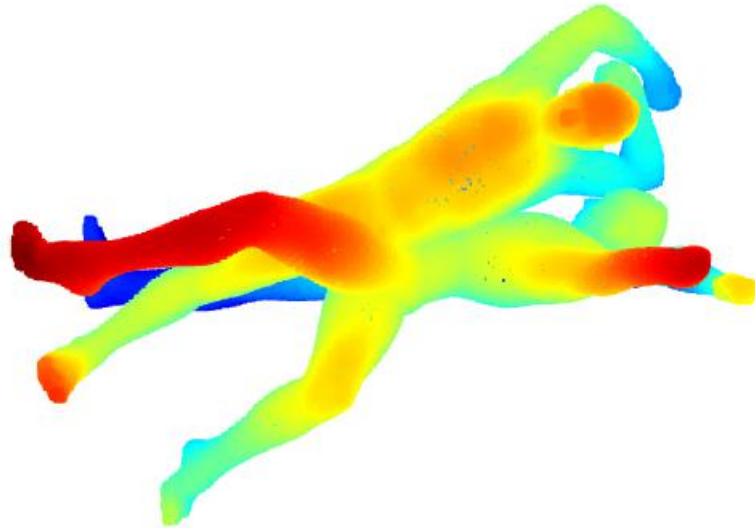
eps 领域距离参数，距离越小（数值），越容易被聚类成多个类（领域），例如左图恐龙的两条腿识别成不同的类了；

min_points 最小点数，聚类成一个类满足的最小点数，点数越小（数值），识别出的类越多，例如左图有 11 个类，而右边的图有 4 个类。

3、点云配准任务

(1) 提供配准前后对比图

配准前：



配准后：



调用 RANSAC 执行配准：红绿点云重叠，结果很准确

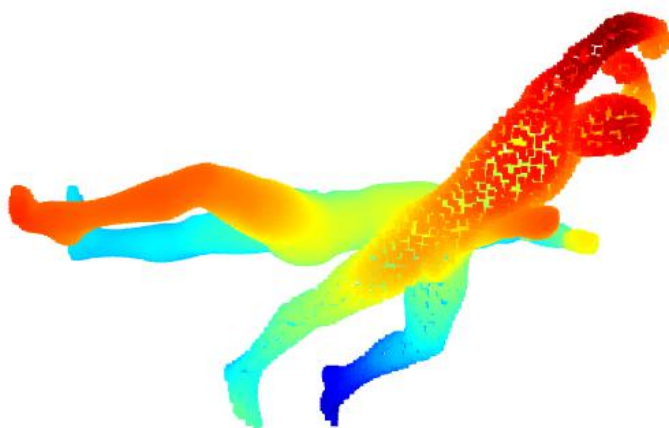


调用 ICP 执行配准：红绿点云重叠，结果很准确

(2) 提供配准前后对比图

说明：body3 和 body 相比，仅仅是位置不同，而 body2 和 body 相比，不仅是位置不同，还多点云做了抽稀

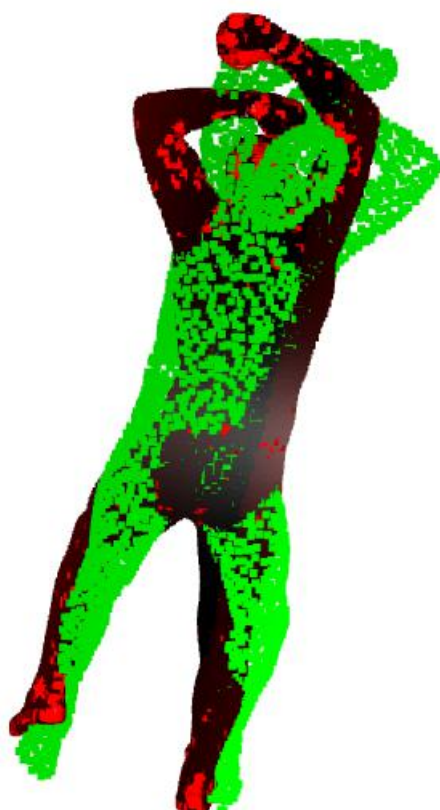
配准前：



配准后:



调用 RANSAC 执行配准: 整体上配准较差, 原因是 body2 做了抽稀。



调用 ICP 执行配准：整体上配准较差，原因是 body2 做了抽稀。

(3) 观察(1)、(2)的配准结果，你有什么想法或启发？

答：预处理步骤如去噪和下采样对配准效果有显著影响。(2)中的实验结果与(1)相比差距较大，主要原因在于抽稀操作导致点云信息量的减少和特征点的丢失。抽稀后的点云密度降低，关键特征点变得稀疏，使得算法在匹配和对齐过程中难以找到足够的对应点，尤其对于 ICP 算法，对点云密度和初始位置的依赖较大。因此，在处理抽稀后的点云时，配准精度和鲁棒性都会受到明显影响，导致配准结果不如未抽稀时准确。