

目录

一、滤波去噪..... 1

 1.1 PCL 去噪..... 1

 2.2 深度学习去噪 p2p-bridge..... 2

二、配准..... 2

三、分割..... 3

 3.1 PCL 分割机械壁支座 3

 3.2 PCL 平面分割 4

 3.3 PCL 其他分割 5

 3.4 OpenIns3D 5

 3.5 SAM2Point..... 6

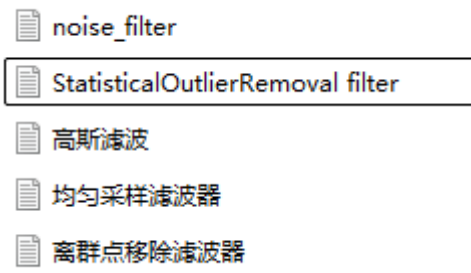
四、PCL mesh..... 8

PCL 说明:我的电脑路径在
D:\HXJ\code\pcl-learning\mypcl_segmentation\bulid\cloud_viewer.sln,双击即可打开,代码只需要把 txt 文件的内容复制到本地的 mypcl_segmentation.cpp 里,点击运行即可,如需部署在其他环境,可参考 CMakeLists.txt 进行配置环境.

一、滤波去噪

1.1 PCL 去噪

PCL 运行的时候不能含中文名的路径
修改路径，代码不长，具体参数修改可问 chatgpt



2.2 深度学习去噪 p2p-bridge

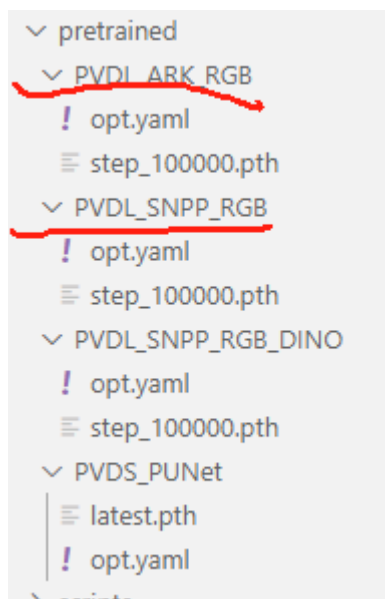
4090 Anaconda 环境: p2pb

纯 3D 文件输入

```
python denoise_room.py --room_path <ROOM PATH> --model_path <MODEL PATH> --out_path <OUTPUT PATH>
```

```
python denoise_room.py --room_path /home/hanglok/Desktop/HXJ/code/P2P-Bridge/data/hxj/data/cloud.ply --model_path /home/hanglok/Desktop/HXJ/code/P2P-Bridge/pretrained/PVDL_ARK_RGB/step_100000.pth --out_path /home/hanglok/Desktop/HXJ/code/P2P-Bridge/data/hxj/output/denoised_cloud.ply --overwrite
```

`--overwrite` (可选) 当 `out_path` 已经存在输出的点云时, 再运行代码会报错, `Overwrite` 可以直接覆盖原来输出的去噪点云, 如果之前没有存在, 也可以生成新的, `--overwrite` 最好写上去。



`/home/hanglok/Desktop/HXJ/code/P2P-Bridge/pretrained` 下面有 `model_path`, 在 `xyzrgb` 输入时只能选红色划线部分的, 下面两个一个需要 `dino`, 一个是输入是 `XYZ` 形式的, 不符合要求。

二、配准

我的电脑环境

Python 解释器: Python 3.7 (base) D:\pycharm\python.exe

代码位置: D:\HXJ\code\probreg-master

安装好 probreg 环境, 修改 D:\HXJ\code\probreg-master\probreg\callbacks.py

修改 callbacks.py, 这样源点云和目标点云保持原来的颜色, 配准后的结果点云显示蓝色。

```
91     self._save = save
92     self._keep_window = keep_window
93     if not self._source.has_colors():
94         self._source.paint_uniform_color([1, 0, 0])
95     if not self._target.has_colors():
96         self._target.paint_uniform_color([0, 1, 0])
97     # if not self._result.has_colors():
98     #     self._result.paint_uniform_color([0, 0, 1])
99     self._result.paint_uniform_color([0, 0, 1])
100     self._vis.add_geometry(self._source)
101     self._vis.add_geometry(self._target)
102     self._vis.add_geometry(self._result)
```

myRT.py

source, target = prepare_source_and_target_3d(source_filename, target_filename, 0.01)

只需要修改源点云和目标点云的路径, voxel_size 根据点云的数量来调整。

- ☒ bcpd_nonrigid
- ☒ cpd_affine3d_cuda
- ☐ cpd_nonrigid3d_cuda
- ☐ cpd_rigid
- ☐ cpd_rigid_cuda
- ☐ filterreg_feature
- ☐ filterreg_rigid
- ☐ filterreg_rigid_pt2pl
- ☐ gmmtree_rigid
- ☐ icp_test
- ☐ myRT
- ☐ svr_rigid

三、分割

3.1 PCL 分割机械壁支座

PCL 运行的时候不能含中文名的路径

半人工分割-去除 mesh 的代码如果没有进行 Manual cutting stage 不报错

半人工分割代码如果没有进行 Manual cutting stage 报错, 需要进行这个操作。(如果需要修改该问题可以参考半人工分割-去除 mesh 的代码进行修改)

这个是颜色区域生长分割, 如果部件分割效果不好可以调整此参数

```
411 // 参数设置
412 int MinClusterSize = 1000, KN_normal = 20;
413 float DistanceThreshold = 10.0, ColorThreshold = 4.3, RegionColorThreshold = 3.5, SmoothnessThreshold = 30.0, CurvatureThreshold = 0.05;
```

这是计算法面信息, mls_radius 越大运行速度会越慢, polynomial_order 可不用调

```
498 // 计算表面元素
499 std::cout << "Computing surface ... " << std::endl;
500 double mls_radius = 0.05;
501 int polynomial_order = 2;
502 auto surfels = reconstructSurface(cloud, mls_radius, polynomial_order);
```

这个参数是控制 mesh 之后的平滑效果的, 数字越大越平滑。

```
508 // 对网格进行拉普拉斯平滑处理
509 std::cout << "Applying Laplacian smoothing ... " << std::endl;
510 pcl::PolygonMesh smoothedMesh = smoothMesh(mesh, 3); // 迭代5次
```

这个是滤波参数, 可以通过观察输出的点数来看看滤波多少

```
127 {
128     pcl::StatisticalOutlierRemoval<pcl::PointXYZRGB> sor_stat;
129     sor_stat.setInputCloud(input_cloud);
130     sor_stat.setMeanK(20);
131     sor_stat.setStddevMulThresh(0.01);
132     sor_stat.filter(*output_cloud);
133 }
```

Mesh 阶段三角网格参数, 一般调整 setSearchRadius, setMu, setMaximumNearestNeighbors

```
376
377 gp3.setSearchRadius(0.05); // 设置搜索半径
378 gp3.setMu(20.0); // 设置内侧点搜索距离
379 gp3.setMaximumNearestNeighbors(100); // 设置最大邻居点数
380 gp3.setMaximumSurfaceAngle(M_PI / 4); // 设置最大表面角 (45度)
381 gp3.setMinimumAngle(M_PI / 18); // 设置最小角度 (10度)
382 gp3.setMaximumAngle(2 * M_PI / 3); // 设置最大角度 (120度)
383 gp3.setNormalConsistency(true); // 设置法线一致性
```

在三角化之后进行拉普拉斯平滑

```
395 laplacianVtk.setInputMesh(mesh);
396 laplacianVtk.setNumIter(iterationsNumber);
397 laplacianVtk.setConvergence(0.5f);
398 laplacianVtk.setRelaxationFactor(0.25f);
399 laplacianVtk.setFeatureAngle(360.f);
400 laplacianVtk.setEdgeAngle(180.f);
```

3.2 PCL 平面分割

PCL 运行的时候不能含中文名的路径

```

17 // 平面分割初始化
18 pcl::SACSegmentation<pcl::PointXYZRGB> seg(true);
19 seg.setOptimizeCoefficients(true);
20 seg.setModelType(pcl::SACMODEL_PLANE);
21 seg.setMethodType(pcl::SAC_RANSAC);
22 seg.setMaxIterations(1000);
23 seg.setDistanceThreshold(0.06);

36 // 提取输入平面
37 while (cloud_copy->points.size() > 0.3 * nr_points) {
38     pcl::PointCloud<pcl::PointXYZRGB>::Ptr cloud_temp(new pcl::PointCloud<pcl::PointXYZRGB>);
39     pcl::PointCloud<pcl::PointXYZRGB>::Ptr cloud_remaining(new pcl::PointCloud<pcl::PointXYZRGB>);
40     pcl::ModelCoefficients::Ptr coeff(new pcl::ModelCoefficients);
41








77
78 // 设置颜色聚类参数
79 reg.setInputCloud(planes[i]);
80 reg.setDistanceThreshold(1); // 空间距离阈值
81 reg.setPointColorThreshold(5); // 点颜色差异阈值
82 reg.setRegionColorThreshold(5); // 区域颜色差异阈值
83 reg.setMinClusterSize(0.06 * nr_points); // 最小聚类大小

```

3.3 PCL 其他分割

PCL 运行的时候不能含中文名的路径

修改路径，代码不长，具体参数修改可问 chatgpt

名称
 cluster_extraction
 conditional_euclidean_clustering
 cylinder_segmentation
 min_cut_segmentation
 region_growing_rgb_segmentation
 region_growing_segmentation
 supervoxel_clustering

3.4 OpenIns3D

4090 Anaconda 环境: openins3d

纯 3D 文件输入

需要把点云放到指定文件夹下，修改路径（蓝色），detector 选择"odise"/"yoloworld",

一般"odise"效果较好。vocabulary 可以是一句话,也可以是一个词语,用来后续进行 all mask 后的 detect 操作。

```
94 if __name__ == "__main__":
95     v = viz.Visualizer()
96
97     name_of_scene = "d_cloud"
98     path_3d_scans = f"/home/hanglok/Desktop/HXJ/code/OpenIns3D/data/hxj/scenes/{name_of_scene}.ply"
99     path_masks = None
100     path_images = None
101     vocabulary = "sit"
102     detector="odise"
103     xyz, rgb, mask_final, mask, v = single_vocabulary_detection(path_3d_scans, vocabulary, path_masks, path_images, detector)
104     plot_mask(mask, mask_final, xyz, rgb, name_of_scene, v)
105     v.save(f'output/hxj/viz')
106
```

如果不在乎 detect 操作,只需要 all mask,则不需要修改下面的,需要进行 detect,可修改路径, [800, 800]为顶部虚拟摄像机拍摄的图像大小, [2, 0.5, 1.0]为 self.lift_cam, self.zoomout, self.remove_lip = adjust_camera, 调整相机参数, [相机抬起高度, 缩放, 去除天花板高度], Lookup 的 0.5 为 self.remove_lip = remove_lip, 需要和上面的 snap 保持一致。

```
39 snap_class = Snap([800, 800], [2, 0.5, 1.0], "output/hxj_snap")
40 lookup_class = Lookup([800, 800], 0.5, "output/hxj_snap", text_input=[vocabulary], results_folder="output/hxj_results")
41
```

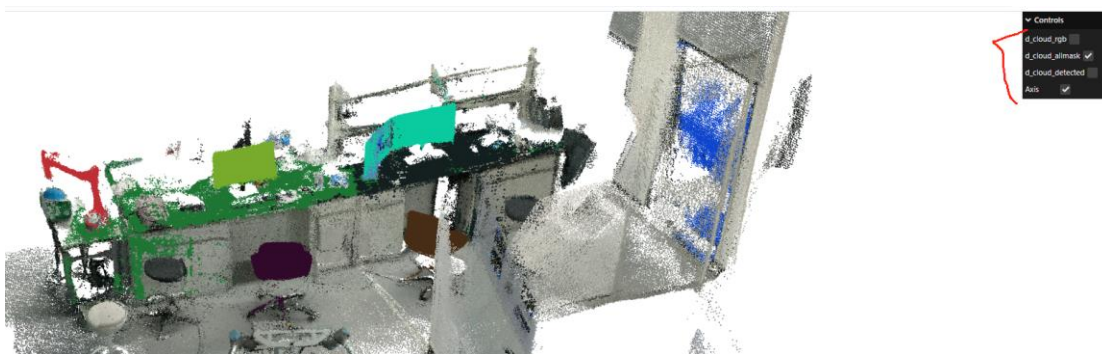
可以通过调整 threshold 来增加检测能力

```
61 if path_images:
62     mask_classification, score = lookup_class.lookup_pipeline(xyz_rgb, mask, name_of_scene, threshold = 0.6, use_2d=True, single_detection=True)
63 else:
64     mask_classification, score = lookup_class.lookup_pipeline(xyz_rgb, mask, name_of_scene, threshold = 0.6, use_2d=False, single_detection=True)
65
```

只需要点击运行即可, 或者 python single_voc.py

```
*****
1) Start local server:
   cd /home/hanglok/Desktop/HXJ/code/OpenIns3D/output/hxj/viz; python -m http.server 6008
2) Open in browser:
   http://localhost:6008
*****
● (openins3d) hanglok@hanglok-Z790-UD-AX:~/Desktop/HXJ/code/OpenIns3D$ cd /home/hanglok/Desktop/HXJ/code/OpenIns3D/output/hxj/viz
○ (openins3d) hanglok@hanglok-Z790-UD-AX:~/Desktop/HXJ/code/OpenIns3D/output/hxj/viz$ python -m http.server 6008
Serving HTTP on 0.0.0.0 port 6008 (http://0.0.0.0:6008/) ...
127.0.0.1 - - [29/Oct/2024 17:17:47] "GET /nodes.json HTTP/1.1" 200 -
127.0.0.1 - - [29/Oct/2024 17:17:47] "GET /d cloud reb.bin HTTP/1.1" 200 -
```

根据步骤进行操作, 在浏览器打开页面, 通过勾选右上角来选择想要看到的画面



中间过程的结果图可以在/home/hanglok/Desktop/HXJ/code/OpenIns3D/output 中进行查看

3.5 SAM2Point

实验效果一般般, 可舍弃。

4090 Anaconda 环境: sam2point

用这个进行归一化 data.py

/home/hanglok/Desktop/HXJ/code/SAM2Point/sam2point/configs.py

需要填入归一化之后的点

```
41
42 sample_my = {'path': 'data/S3DIS/d_cloud.txt',
43              'point_prompts': [ [0.77367154, 0.34661436, 0.37428309]],
44              'box_prompts': [[0.14766599, 0.36918767, 0.38502234, 0.19011664, 0.46423318, 0.49241477]],
45              }
46
47 # sample_my = {'path': 'data/S3DIS/denoised_cloud3.txt',
48 #              'point_prompts': [ [0.61904955, 0.71906361, 0.56666444]],
49 #              'box_prompts': [ [0.53020026, 0.63451371, 0.49860599, 0.69802669, 0.78670353, 0.70278135]],
50 #              }
51
52 S3DIS_samples = [sample_2, sample_3, sample_4, sample_1, sample_0, sample_my ]
53
54
```

需要填入（未归一化前）原始 box 和 point 的位置

box.py

```
129 args.dataset = 'S3DIS'
130 args.prompt_type='box'
131 args.sample_idx=5
132 args.prompt_idx =0
133
134
135 point, color = point_color[:, :3], point_color[:, 3:]
136 new_color = color * point_mask_not.numpy() + (color * 0 + np.array([[0., 1., 0.]]) * point_ma
137
138 point=point*(point0.max(axis=0)-point0.min(axis=0))+point0.min(axis=0)
139 prompt_box=np.array([-4.0, 0.7, -0.19, -3.7, 1.5, 0.23])
140 file_path = visualize_point_cloud(point,new_color,prompt_box, title="Segmented Point Cloud")
141
142
```

point.py


```
105 args.dataset = 'S3DIS'
106 args.prompt_type='point'
107 args.sample_idx=5
108 args.prompt_idx =0
109
110
111 point, color = point_color[:, :3], point_color[:, 3:]
112 new_color = color * point_mask_not.numpy() + (color * 0 + np.array([[0., 1., 0.]]) * point_mask.nump
113
114 point=point*(point0.max(axis=0)-point0.min(axis=0))+point0.min(axis=0)
115 prompt_point=np.array([0.424, 0.51, -0.232])
116 file_path = visualize_point_cloud(point,new_color,prompt_point, title="Segmented Point Cloud")
117
118 # 检查 file path 是否有效
119
```


只需要点击运行即可，或者 python box.py/python point.py


四、PCL mesh


PCL 运行的时候不能含中文名的路径


修改路径，代码不长，具体参数修改可问 chatgpt

 Greedy+MLS+Laplacian3

 GreedyTriangulation+MLS

 泊松曲面重建

 贪心三角化

 移动立方体