作业说明

一、作业完成内容

1. 特征点提取

* 尝试了自己实现的 ISS 特征点提取函数和第 8 节课提供的 PCL 库中的 Sift 特征点、ISS 特征点提取函数完成了特征点提取。
* 最终测试下来发现 Sift 表现稳定些，因此最终提交的结果采用了 PCL 库中的 Sift 特征点。
* 在 init_rt 函数中修改注释可以控制选择哪种特征点提取方式。

2. 特征描述与匹配

* 采用了 PCL 库中的 SHOT352 函数进行特征描述的提取。匹配方法是采用的暴力匹配，匹配规则是两个特征点的 description 互为彼此的最近邻。

3. 旋转和平移矩阵初始化

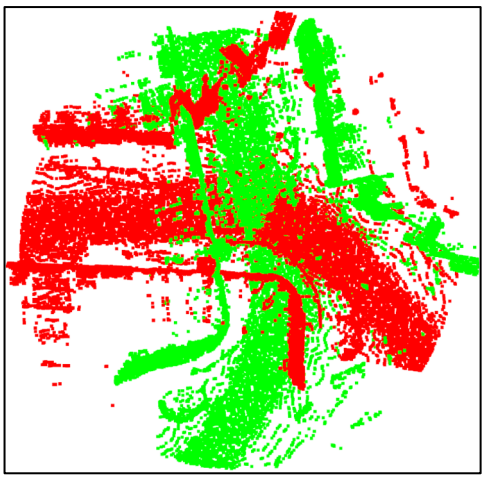
* 利用 ransac 方法：随机选取 3 个点，求解 procrustes 问题后得到一组 r 和 t，利用该组解对特征点进行空间变换；若某个特征点在转换后与其匹配点的空间位置小于阈值（10m），则记一次投票；投票最多的一组解作为初值。
* 存在初始化失败的情况，原因是特征点匹配没有正确匹配上。问题的根源分析如下：
  * 特征点的提取不够好，两帧点云提取到的大部分特征点不在相同的位置，导致怎么匹配都匹配不上对的位置。
  * ransac 方法利用 tau 作为投票的分界线，假设 tau 为 2.0，两组距离分别为：[0.1，0.5，0.7，0.3，3.0，0.7，5.0]和[0.1，1.0，1.8，1.5，1.7，1.4，3.0]。从投票结果上来看，第二组得票更高，但实际匹配效果可能第一组更好。代码中给的 tau = 10m 是拍脑袋定的，没有怎么调试过，多试几次可能会减少一些初始化失败的情况。

4. ICP 迭代

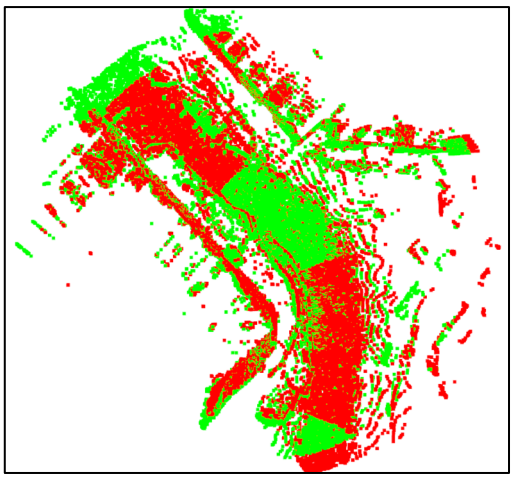
* 采用 NN 方法进行点与点的匹配，然后求解 procrustes 问题，不断迭代，直到收敛。

二、作业完成效果

Sample 1：　35.bin & 490.bin

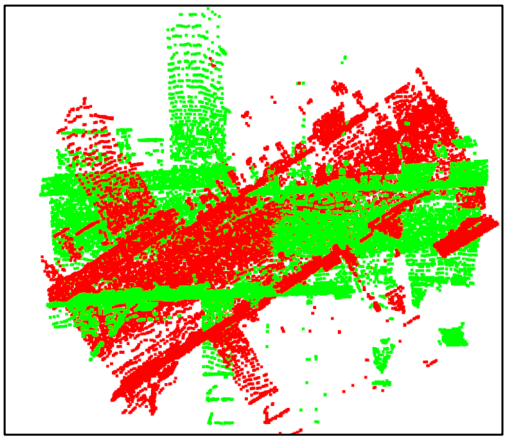


左：初始状态

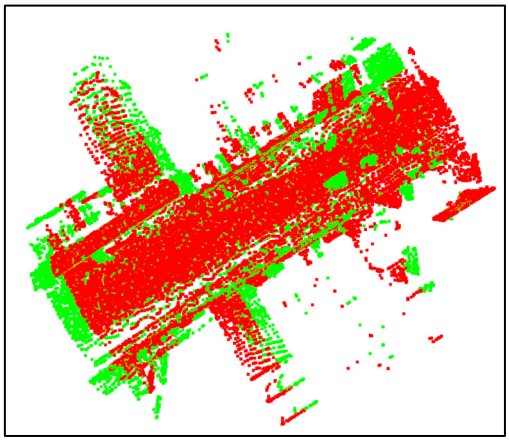


右：最终状态

Sample 2： 164.bin & 432.bin

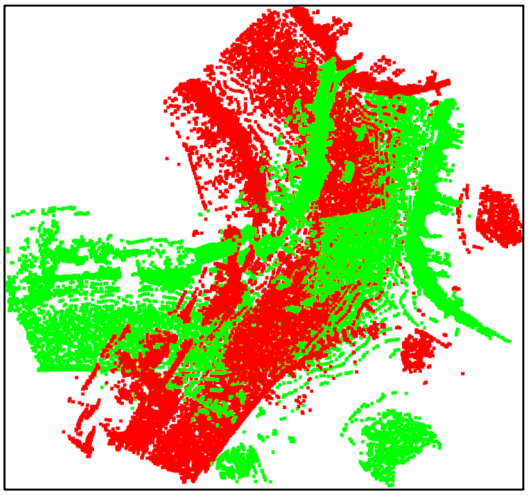


左：初始状态

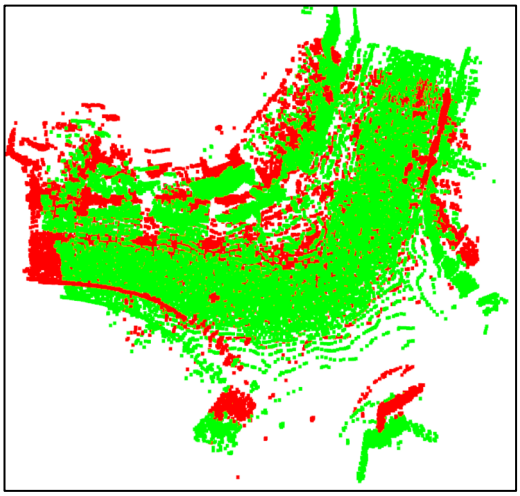


右：最终状态

Sample 3： 2.bin & 458.bin

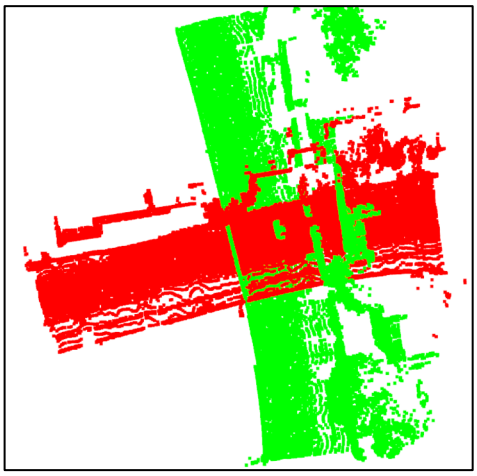


左：初始状态

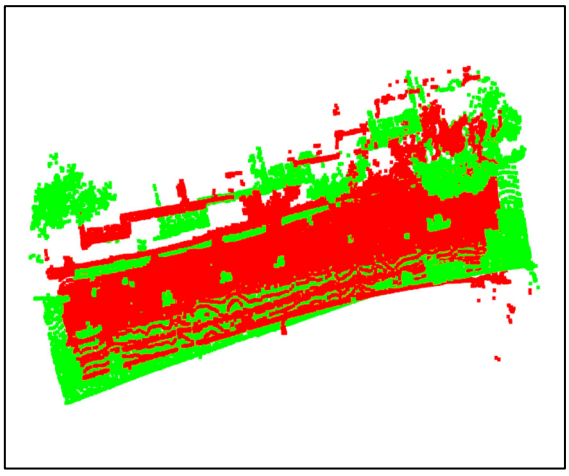


右：最终状态

Sample 4： 177.bin & 463.bin



左：初始状态



右：最终状态

三、代码说明

main.py：遍历 reg_result.txt 中所有行，读取每一行记录的两个点云文件，计算好 R、T 之后写入 result.txt 中（交上去的 reg_result.txt 是 result.txt 重命名得到的）。

iss.py：自己实现的 ISS 特征点检测函数

libPCLKeypoint.cpython-37m-x86_64-linux-gnu.so：PCL 库 C++代码的 python 调用库

evaluate_rt.py：一定要和 main.py 放在一个文件夹下才能运行 main.py，因为 main.py 调用了其中的部分函数（点云读取之类的工具函数）。
（注：可以修改该文件中的 visualize_row_idx 变量，运行该文件，可以看到第 24 对点云在旋转前和旋转后的可视化。）