

1、第一题

题目：要求使用前面讲过的 scan context，实现此功能。并和已有的图优化功能完成对接，实现修正。并最终给出修正前后的轨迹精度对比。

算法参考 SC-Lego-Loam 实现，将 SC-Lego-Loam 算法中的 scan context 部分提取出来，来进行回环约束。

编译代码

```
cd /home/qz-dell/test/04_Multi-sensor
source devel/setup.bash
catkin_make -j4
```

测试 bag 包的效果，需要开启三个终端

终端一：roslaunch lidar_localization mapping.launch

终端二：rosbag play kitti_lidar_only_2011_10_03_drive_0027_synced.bag

在运行过程中，可以执行如下的命令，保存地图与回环帧数据。

```
source devel/setup.bash
rosservice call /optimize_map
```

保存地图：rosservice call /save_map

然后就是通过 bag 包测试，保存的地图在位置下，利用这个图测试定位问题。保存地图的

效果如下：



2、第二题

题目：要求实现地图中任意位置的位姿初始化，可以从三种难度等级中任选一种，难度越高，得分越高。

完成了两种方式的全局任意位置定位，一种是已知位姿的，另一种是只知道位置不知道姿态的。

2.1、前提准备

用于定位的地图

这个直接用建图时候生成的 `filtered_map.pcd` 就可以

地图建立的时候的原始 `gps` 经纬度坐标

用于初始化 `data_pretrat_node` 节点里的 `gps` 坐标，方便后面计算相对的位姿（在上一步生成的 `map_origion.txt` 中）

2.2、基于位姿的定位

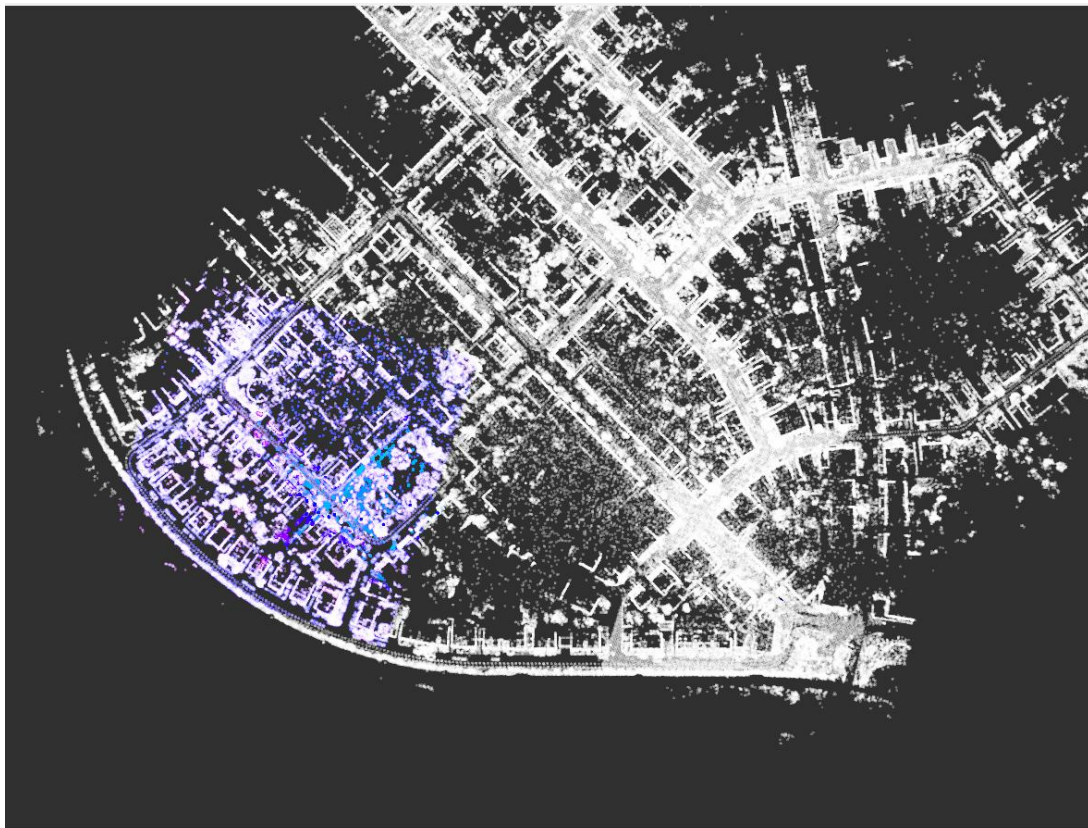
此时 matching 收到的 gnss_pose 已经是相对地图原点的坐标了，所以只需要更新发布一次子地图，即可，剩下的不需要更改其它代码，就可以直接运行定位，这里做了几个测试，测试步骤如下：

首先设置参数文件中为 FullPose

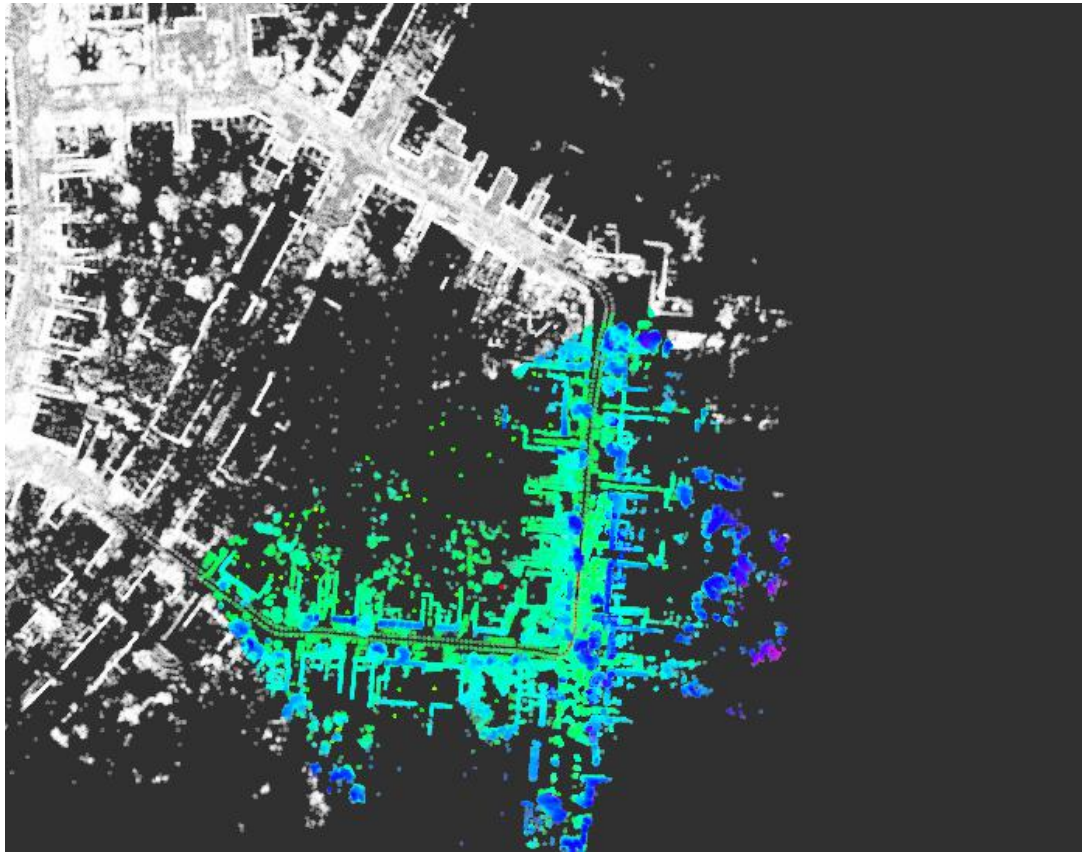
启动节点:roslaunch lidar_localization matching.launch

播放 bag 包，使用 rosbag play xxx.bag -s time 来从 time 时刻播放，选择 100s，200s，300s 来测试，每次测试前需要重新启动 matching.launch 不然会出现时间戳倒回或者跳动很大的问题

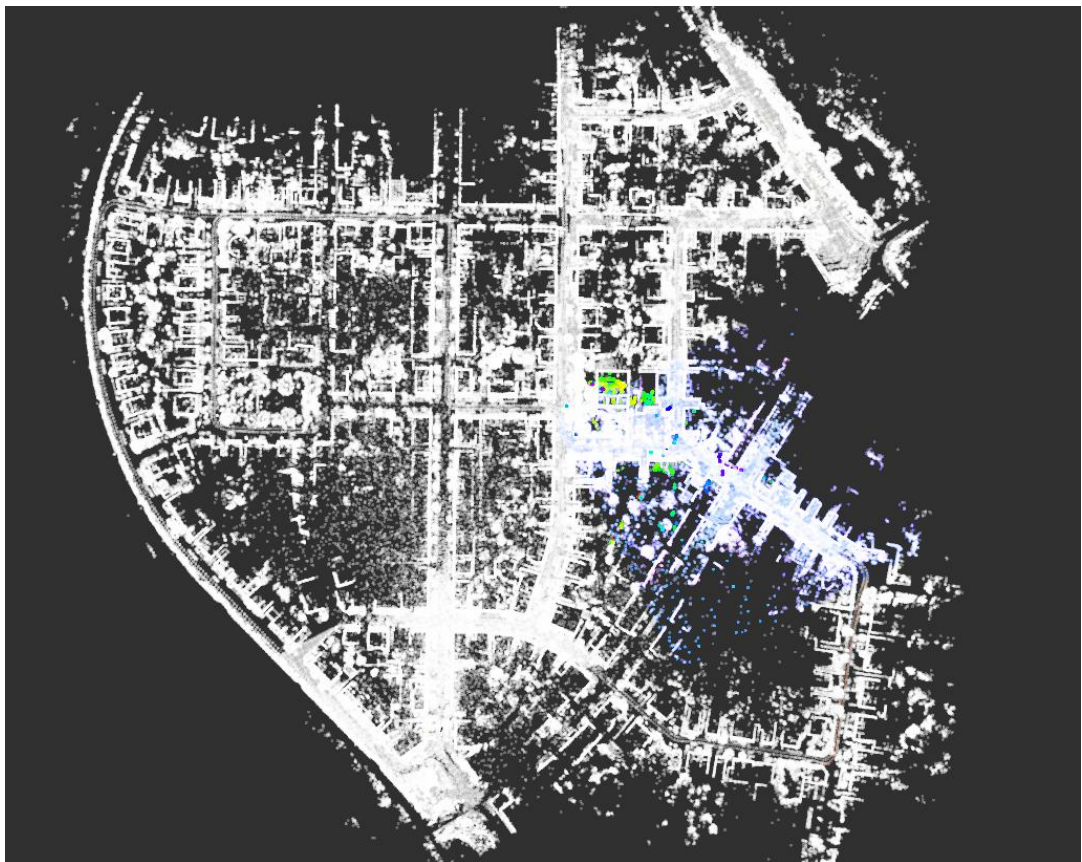
100s



200s



300s



2.3、基于位置的定位

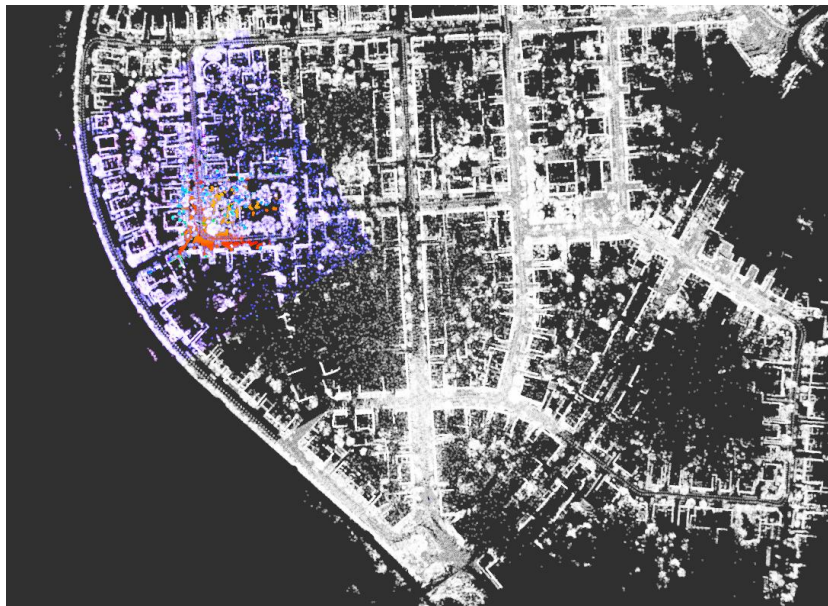
因为在这种情况下，只知道位置，位置可以用来更新子地图，但是如果只用位置去匹配，因为初始化给的不准，有可能会匹配失败，另外，由于俯仰和滚转角比较小，主要是偏航的差距比较大，所以，需要有一种方法能够给出偏航角初始值。利用高度的高斯分布进行初始 yaw 角度的估计，有了初始的 yaw 角。就可以采用 2.2 的方式进行地图中的定位了。测试步骤如下：

首先设置参数文件中为 OnlyPosition

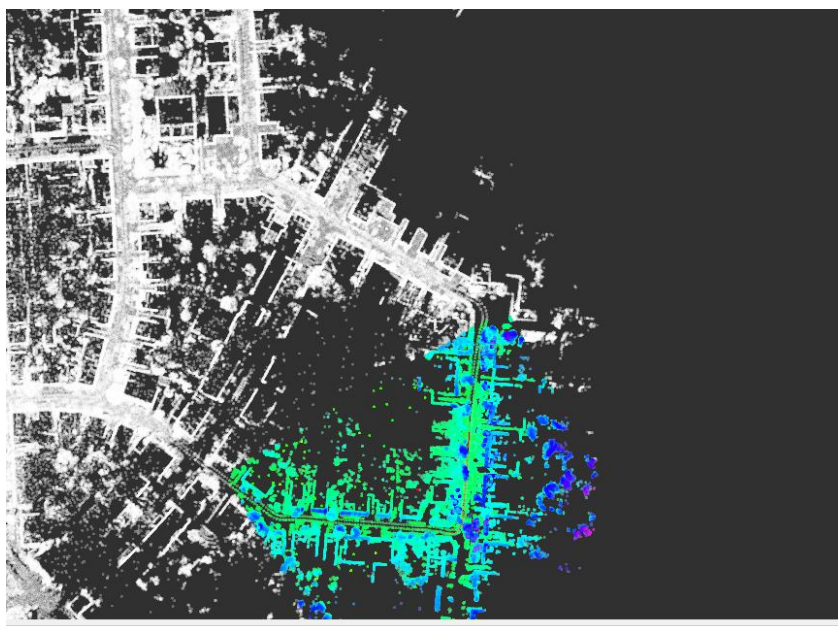
启动节点: `roslaunch lidar_localization matching.launch`

播放 bag 包，使用 `rosbag play xxx.bag -s time` 来从 time 时刻播放，选择 100s，200s，300s 来测试。

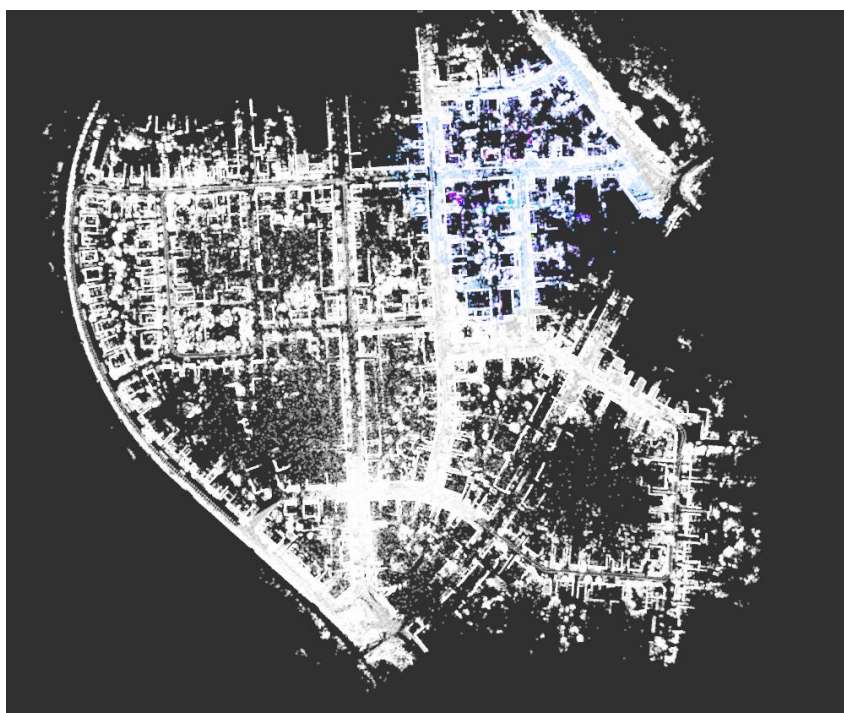
100s



200s



300s



注：先通过 data_pretreat_node 将数据处理好，然后转发出来，后面 matching_node 得到的就是相对于原点 gps 坐标系的 ENU 坐标了。data_pretreat_node 数据处理坐标转换层。