**Imu\_lidar手眼标定算法总结**

手眼标定是featureless的一种标定方法，利用车体上两个位置的传感器记录数据并算出各自的里程计A和B。 令这两个传感器之间的外参转换为X， 那么存在AX=XB 的关系，从而（通过batch optimization）算出X。

Imu里程计用gnss+imu容易得到。

Lidar里程计现有的方法 icp及其变种gicp, ndt及其变种ndt-tku。

标定数据包一般为跑8字。

（具体使用方法请参见文件夹下的README，

<http://192.168.103.77:30000/calib/imu-lidar/tree/chenxi_development/imu-lidar>）

模块一： imu-lidar/examples/handeyeDataExporter.cpp

这个模块可以选择ndt ，icp 以及gicp三种模式， 同是也可以使用ndt—tku来代替此模块。其中ndt，ndt-tku是不断构建地图， 将降维点云与地图进行匹配，当满足一定条件时把点云加入地图。 而icp和gicp使用单帧匹配（如果和地图匹配，会由于点云的累积很快导致匹配变慢）。

目前 ndt 和ndt-tku 效果最好，且比较稳定。icp和gicp单针匹配的参数设置还没有调好，得到的轨迹存在很大偏差。 其中ndt的参数设置参考值列在结果评估的部分。

模块二： imu-lidar/examples/handeyeExamples.cpp

此模块使用单步法（dual quaternion）来解算手眼标定方程 （AX=XB），理论上相较于两步法更为精确。 此算法首先随机选取K个对应的手眼位姿，在满足一定条件下解算出一个粗略估计值。如此重复N次，得到N（实际小于等于N）个初始值。 然后对N个初值进行评估排序，留下误差小的部分初值。然后对这些留下的初值进行再优化，筛选出最好的结果最为最后的输出结果。

其中这些参数可调

结果评估：

1. ndt算法 （根据autoware的算法逻辑修改）

可供参考算法参数：

<arg name="min\_scan\_range" default="8.0" />

<arg name="max\_scan\_range" default="80.0" />

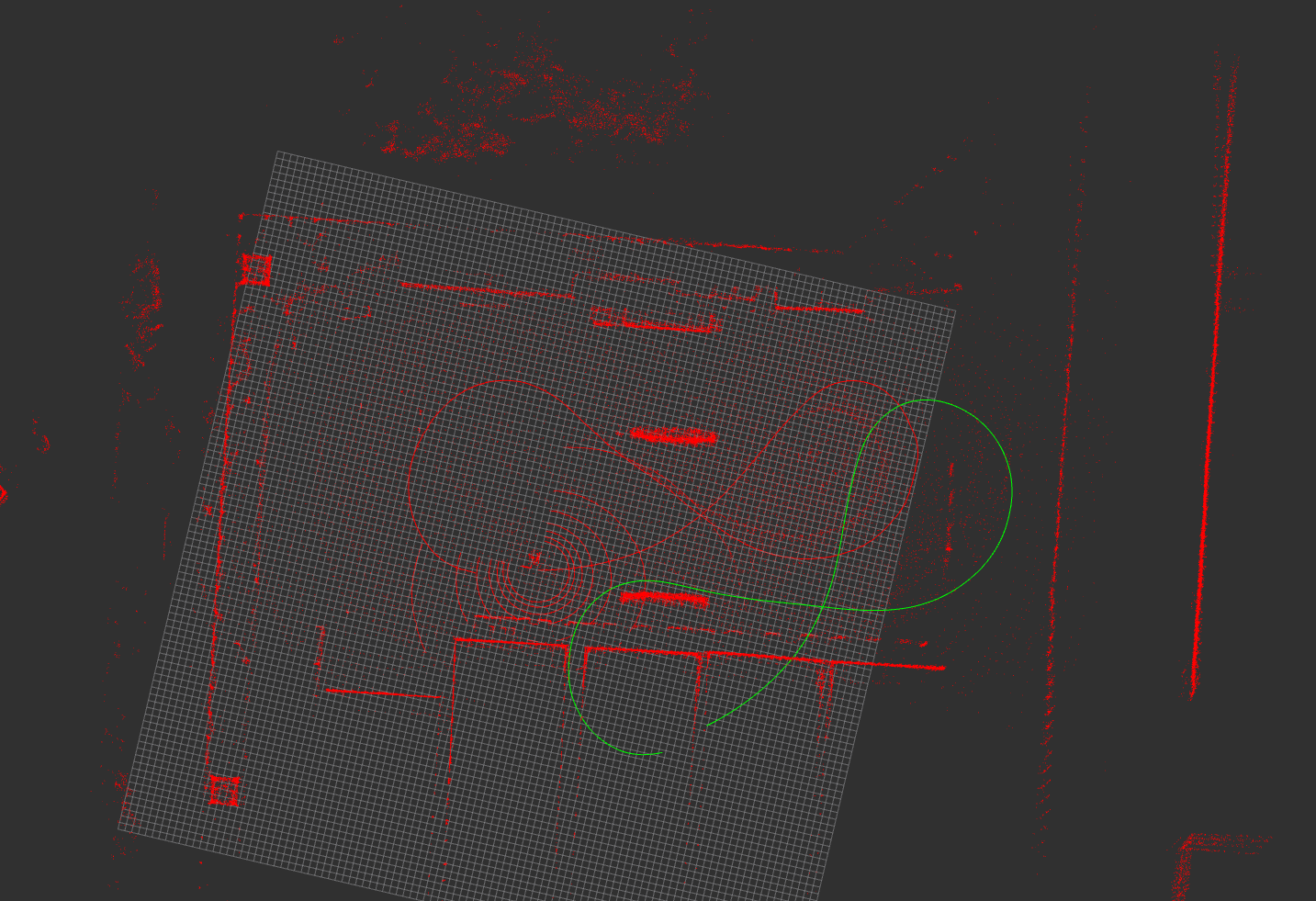
<arg name="min\_add\_scan\_shift" default="1.5" />

<arg name="voxel\_leaf\_size" default="2.0" />

<arg name="ndt\_resolution" default="3.5" />

<arg name="fitness\_tresh" default="10.0" />

图中红色为点云构建的地图，红色轨迹是对应的点云匹配得到的轨迹， 绿色为gnss得到的轨迹。 图中构建的点云地图墙面不够精确，存在分叉的现象。这种匹配多出现在快速转弯。



问题分析：

转弯时原始点云（绿色）因为转弯太快出现断层的现象，可能是导致误差的主要原因。

远处点云断层明显， 近处点云也有细微的断层。

08.02

更新使用的数据包, 较慢速度录制8字， localization-rr2-2019-08-02-10-35-21.bag

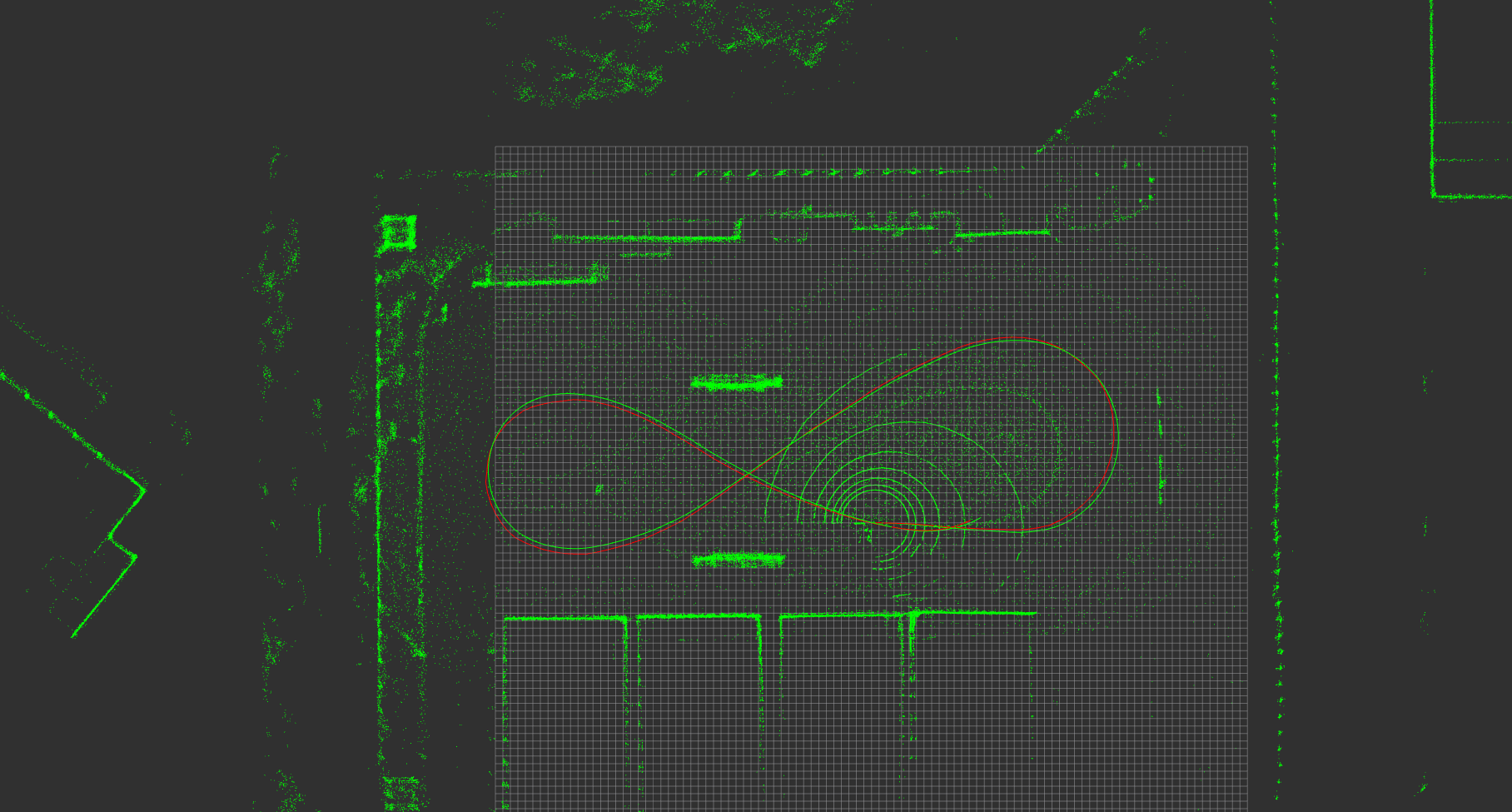
图中绿色点云为ndt mapping得到的结果，

rr2

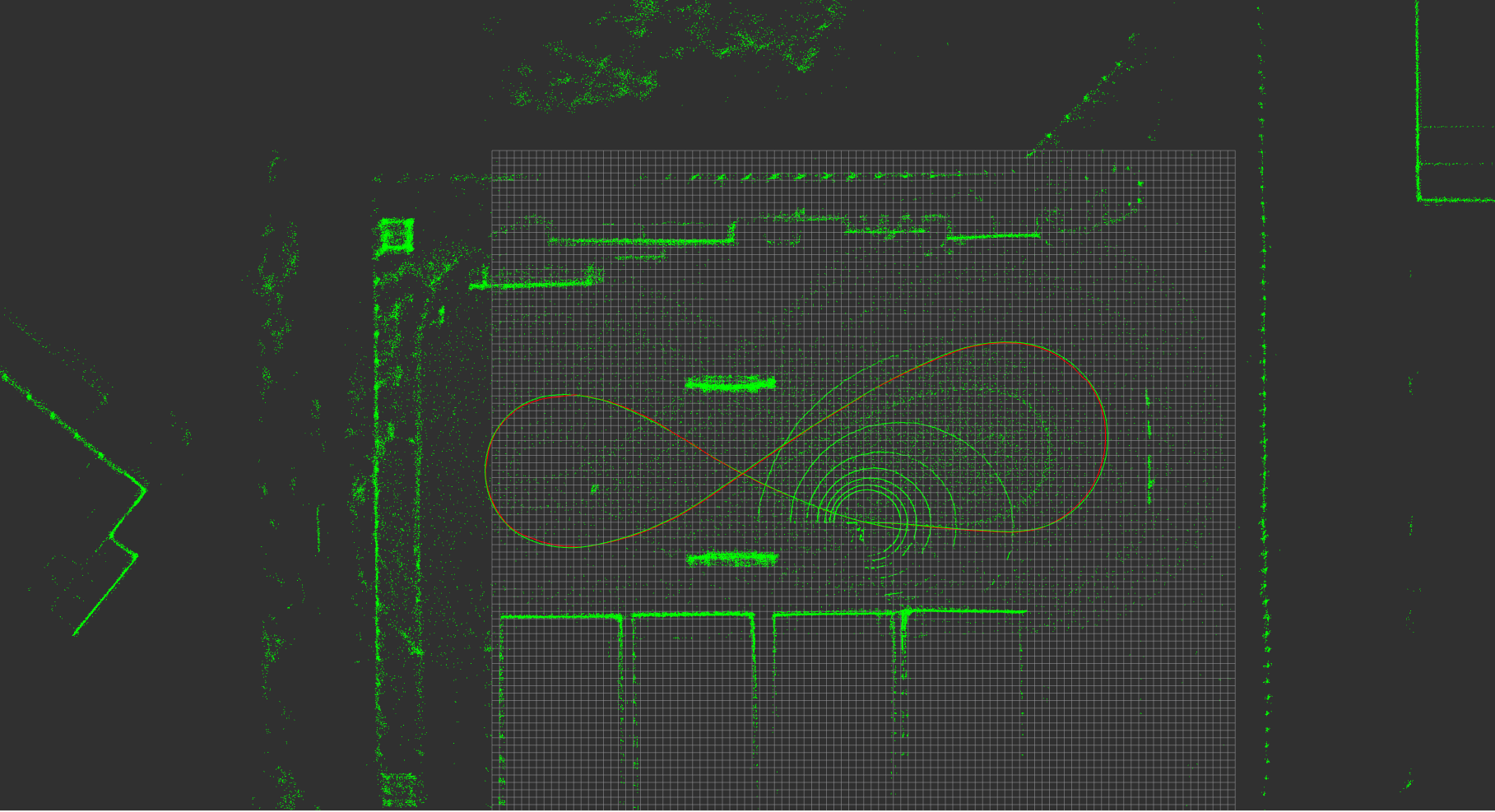
左激光

绿色轨迹为gps的结果， 红色轨迹为ndt path根据初始猜测转换到gps坐标系下的轨迹。

初始猜测： 1.53, 1.3125, -0.67, 0.0, -0.017, -0.02

绿色轨迹为gps的结果， 红色轨迹为ndt path优化后转换到gps坐标系下的轨迹。

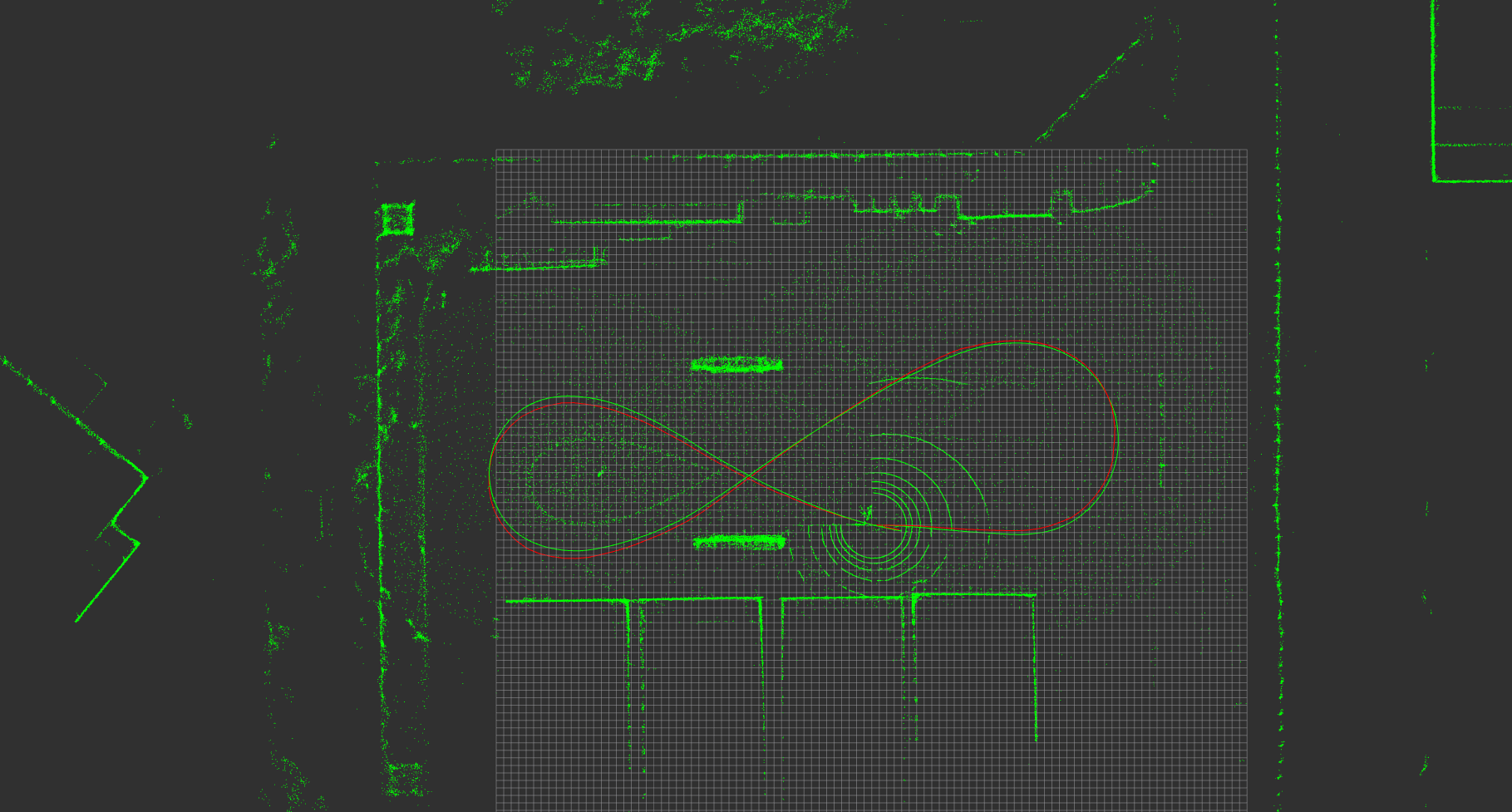
优化后参数 x: 1.7 y: 1.323 z: -0.7 r: 0.03448 p: 0.00328 y: -0.0213285



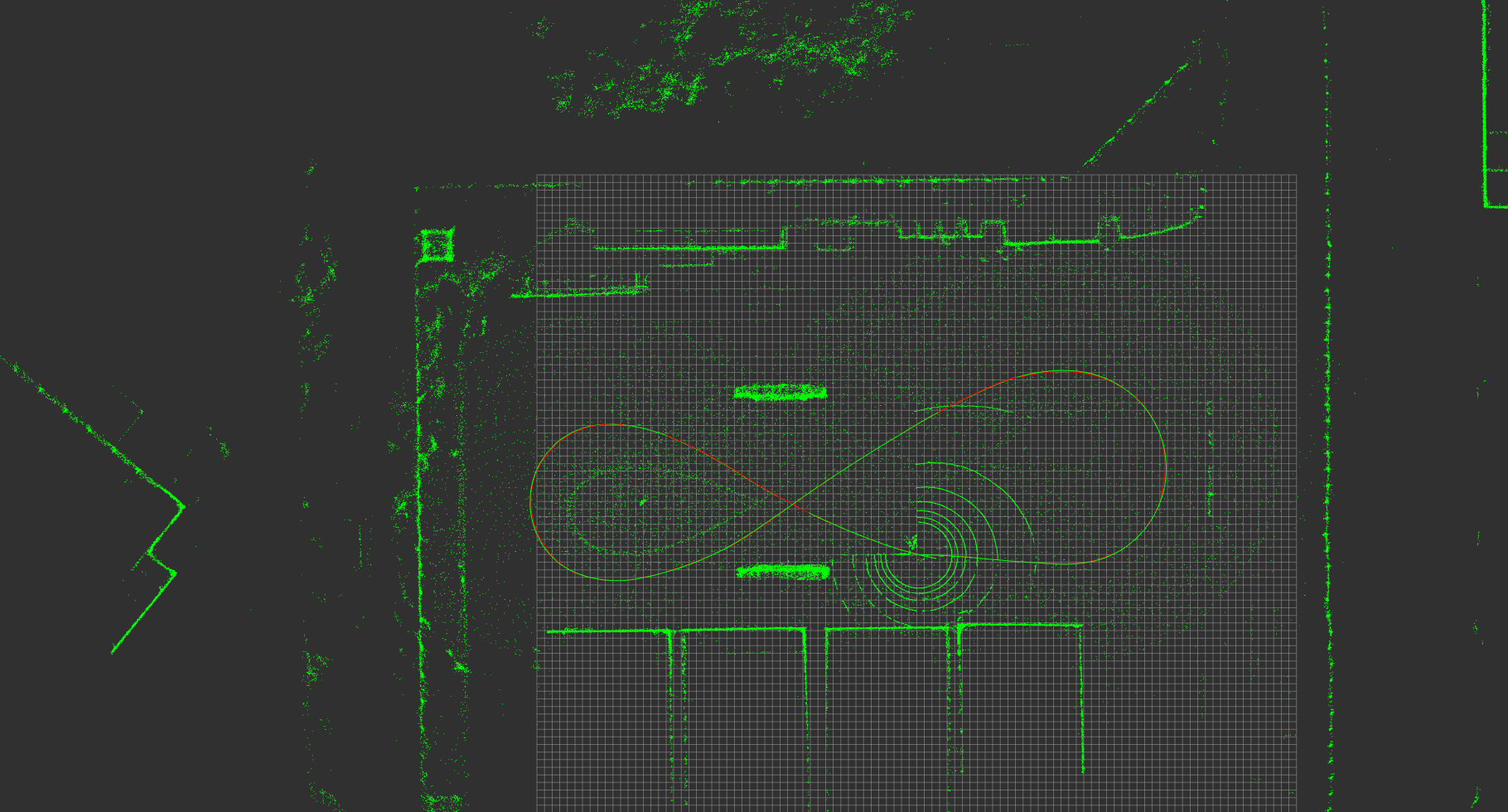
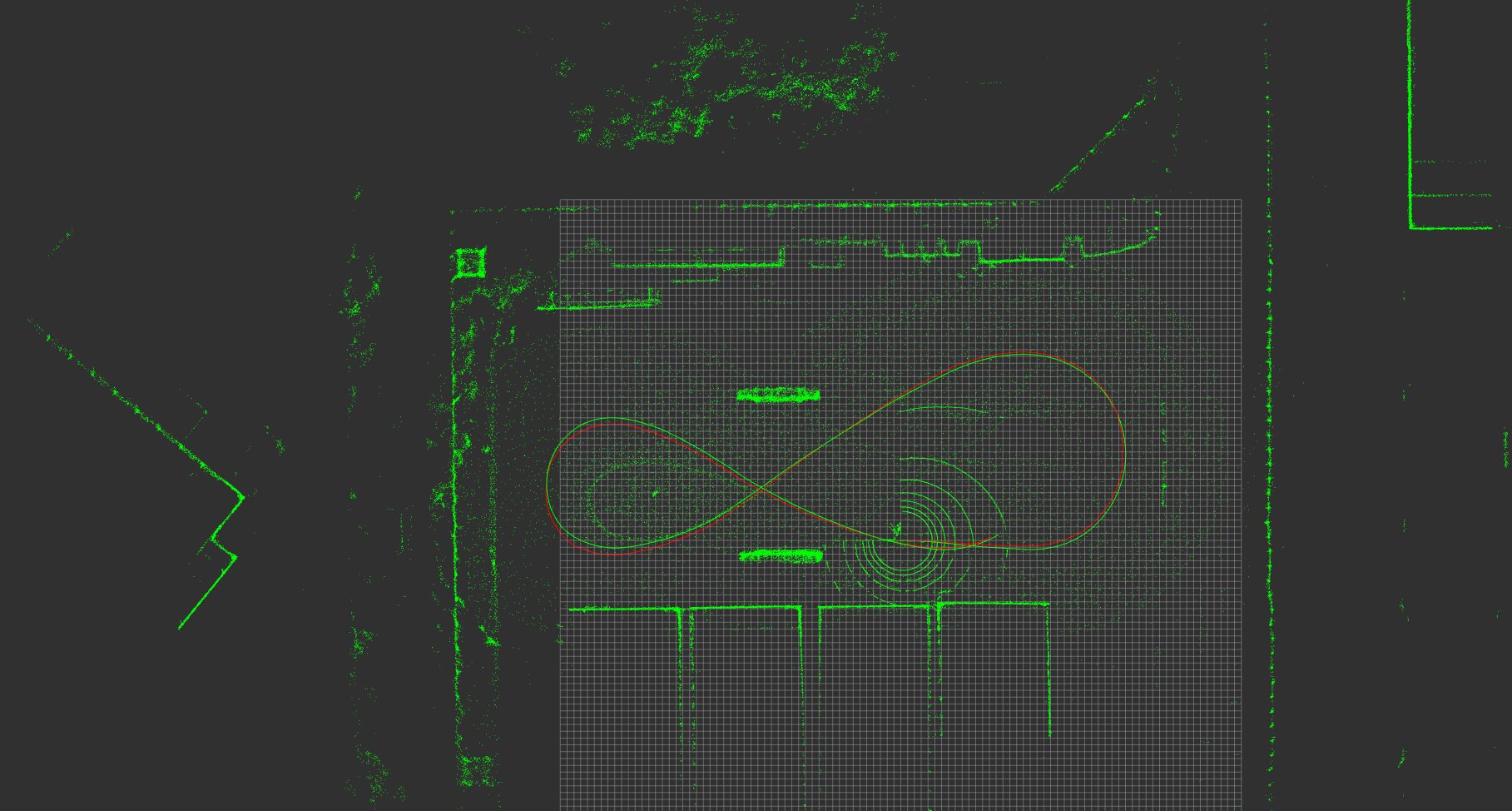
右激光

初始猜测：x1.60, y -1.3125, z -0.66, r 0, p 0, y -0.02

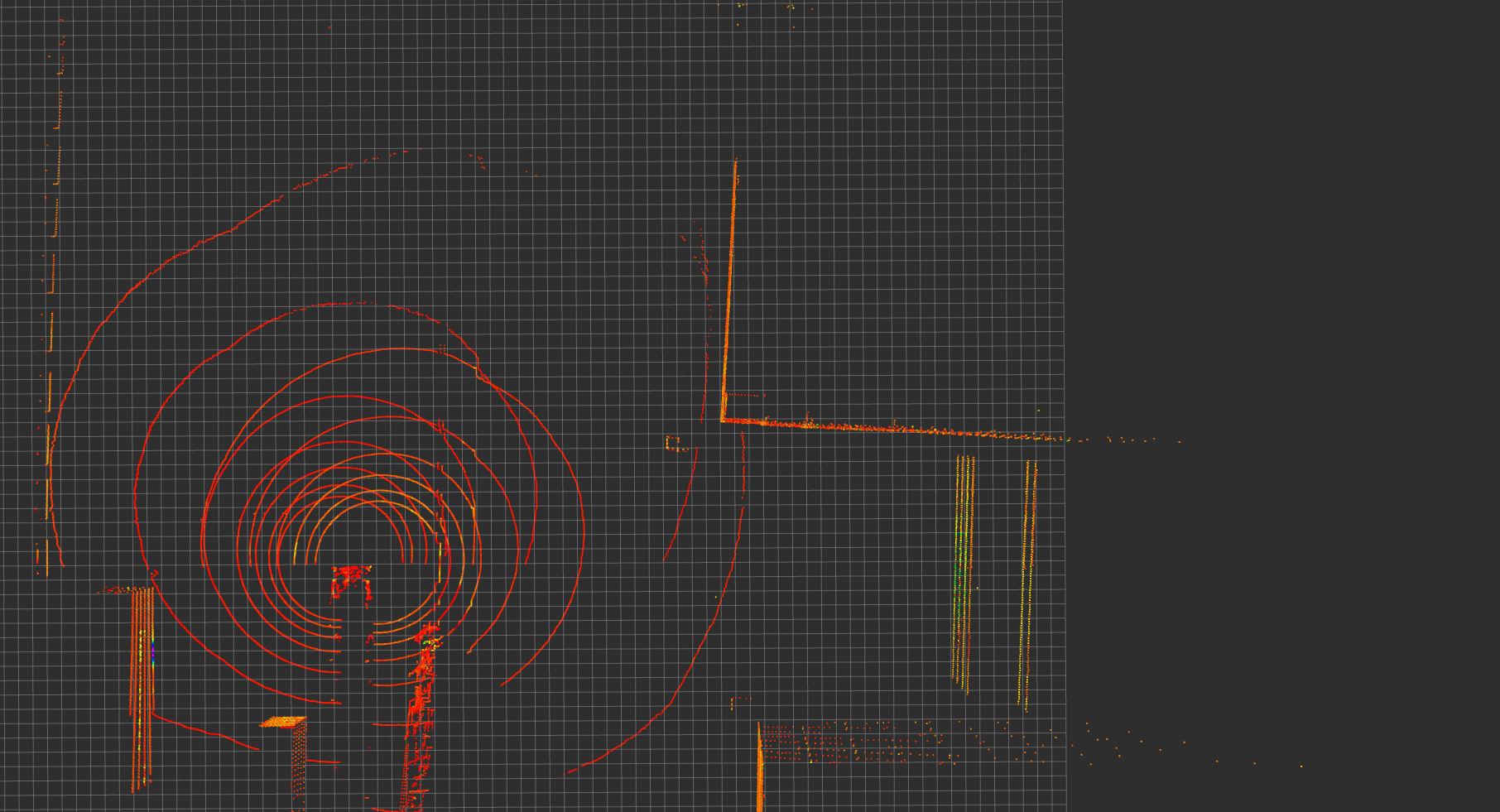
绿色轨迹为gps的结果， 红色轨迹为ndt path根据初始猜测转换到gps坐标系下的轨迹。

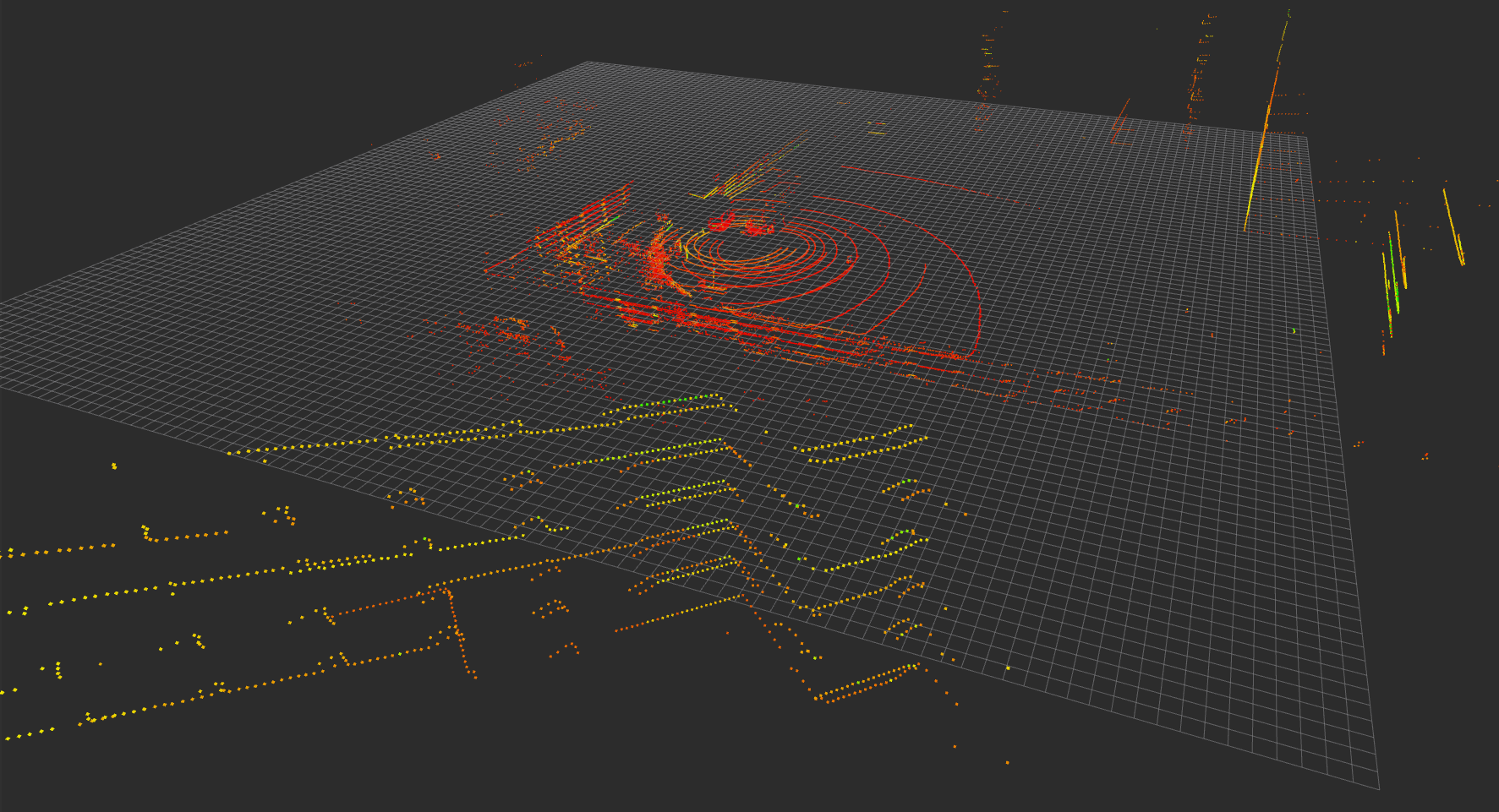
绿色轨迹为gps的结果， 红色轨迹为ndt path优化后转换到gps坐标系下的轨迹。

优化参数： x: 1.6 y: -1.253 z: -0.7 r: 0.0336113 p: 0.0032 y: -0.0237



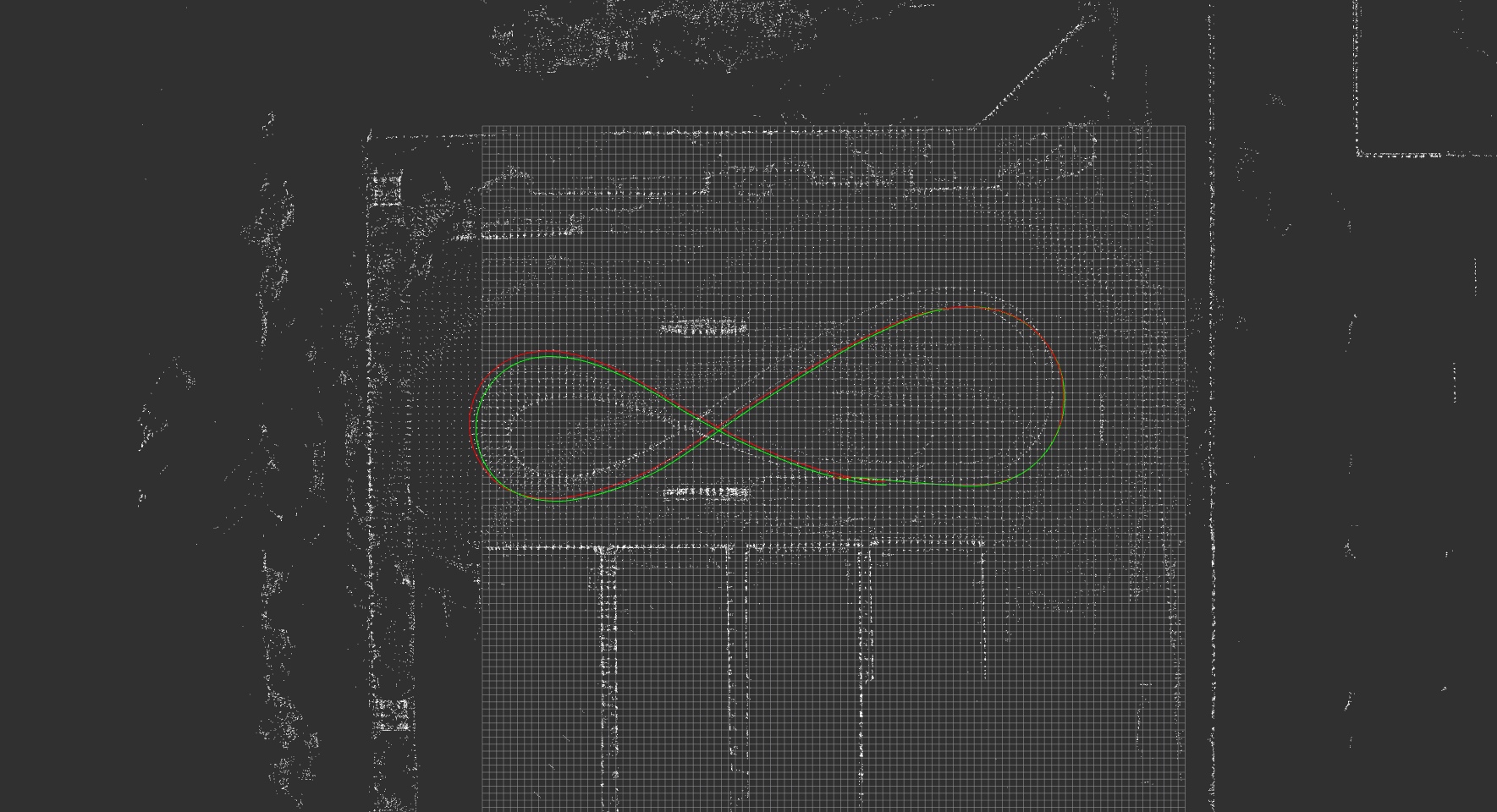
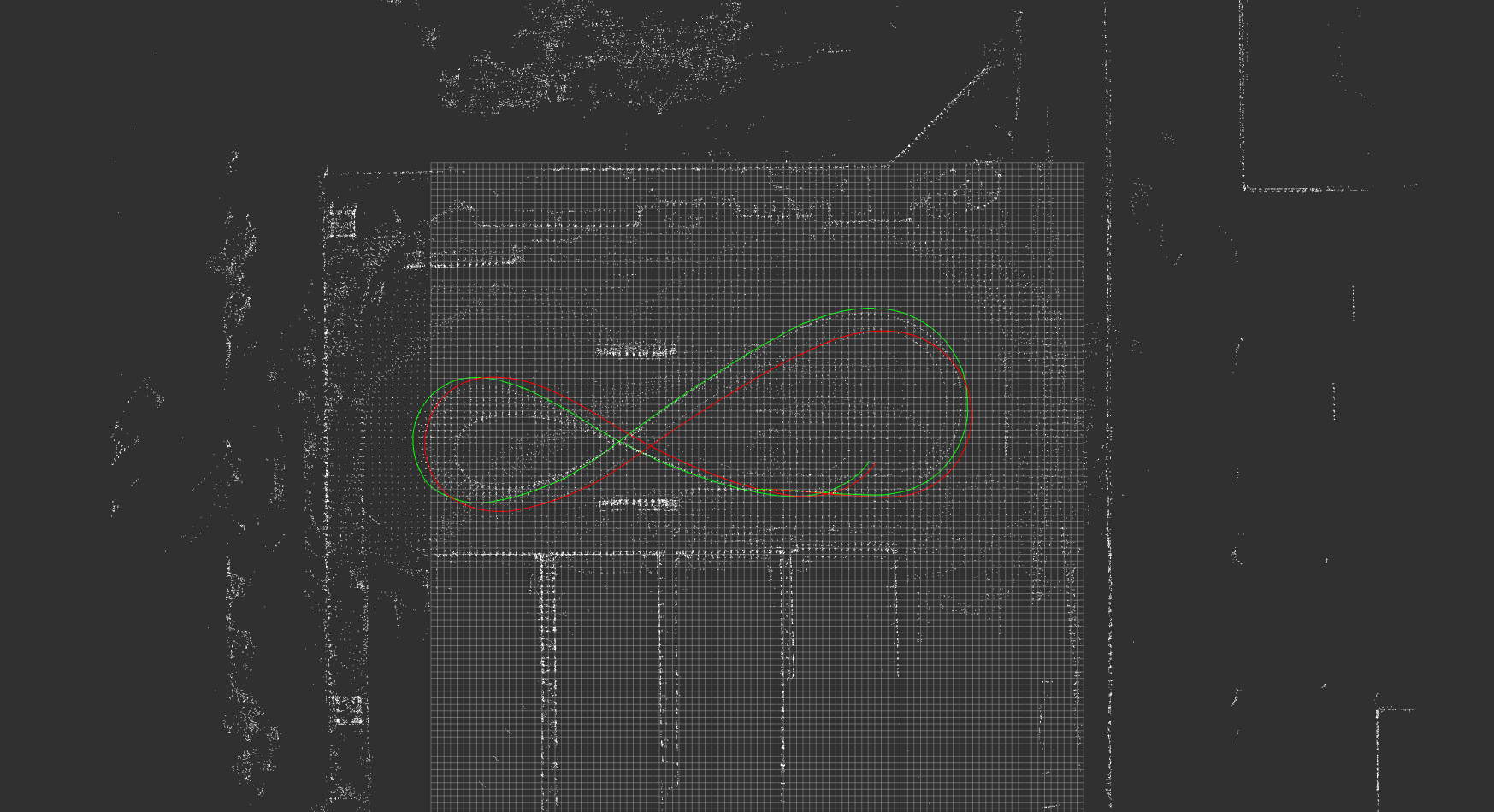
激光融合参数结果



2. ndt-tku

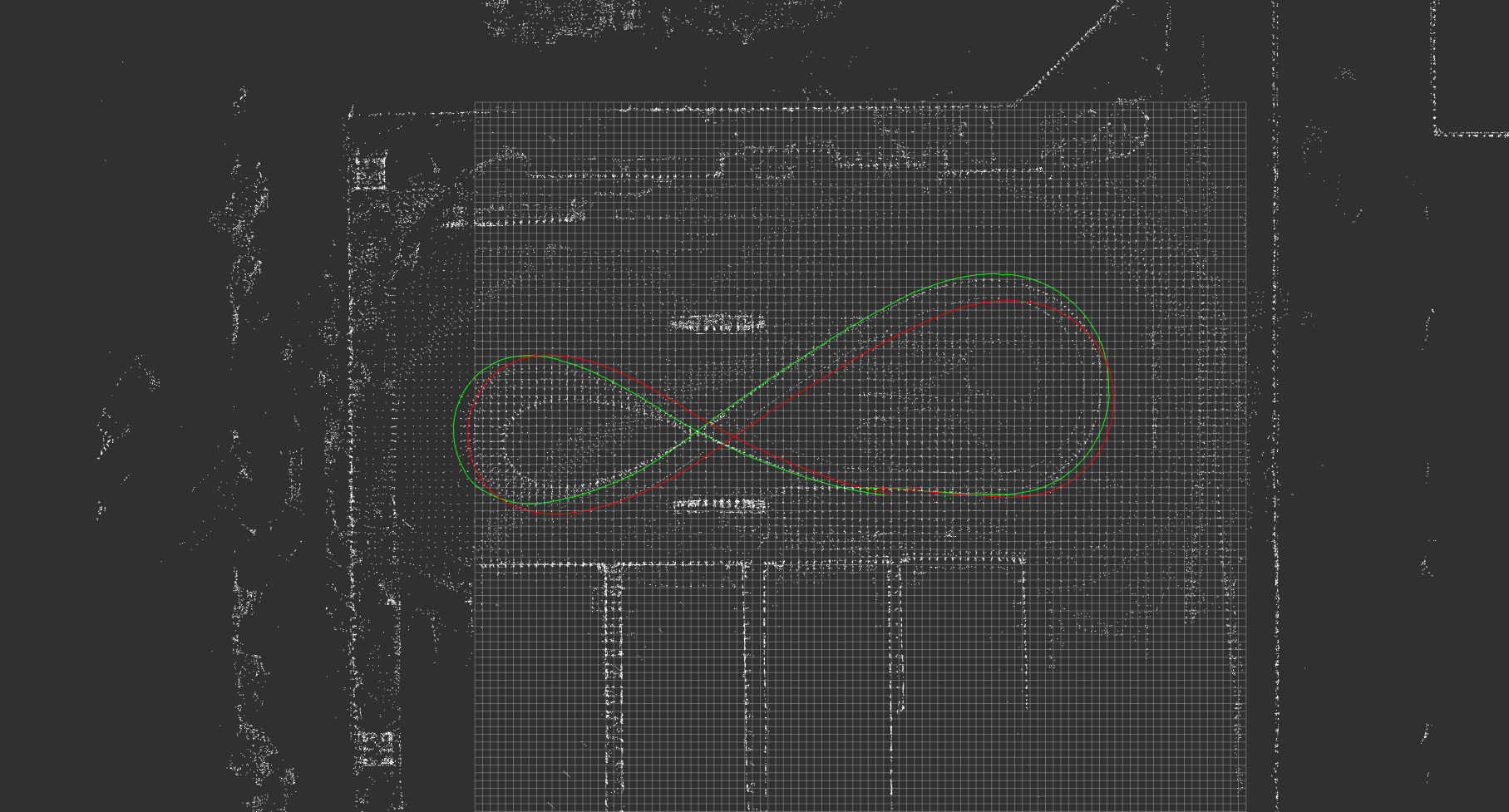
rr2 左激光：

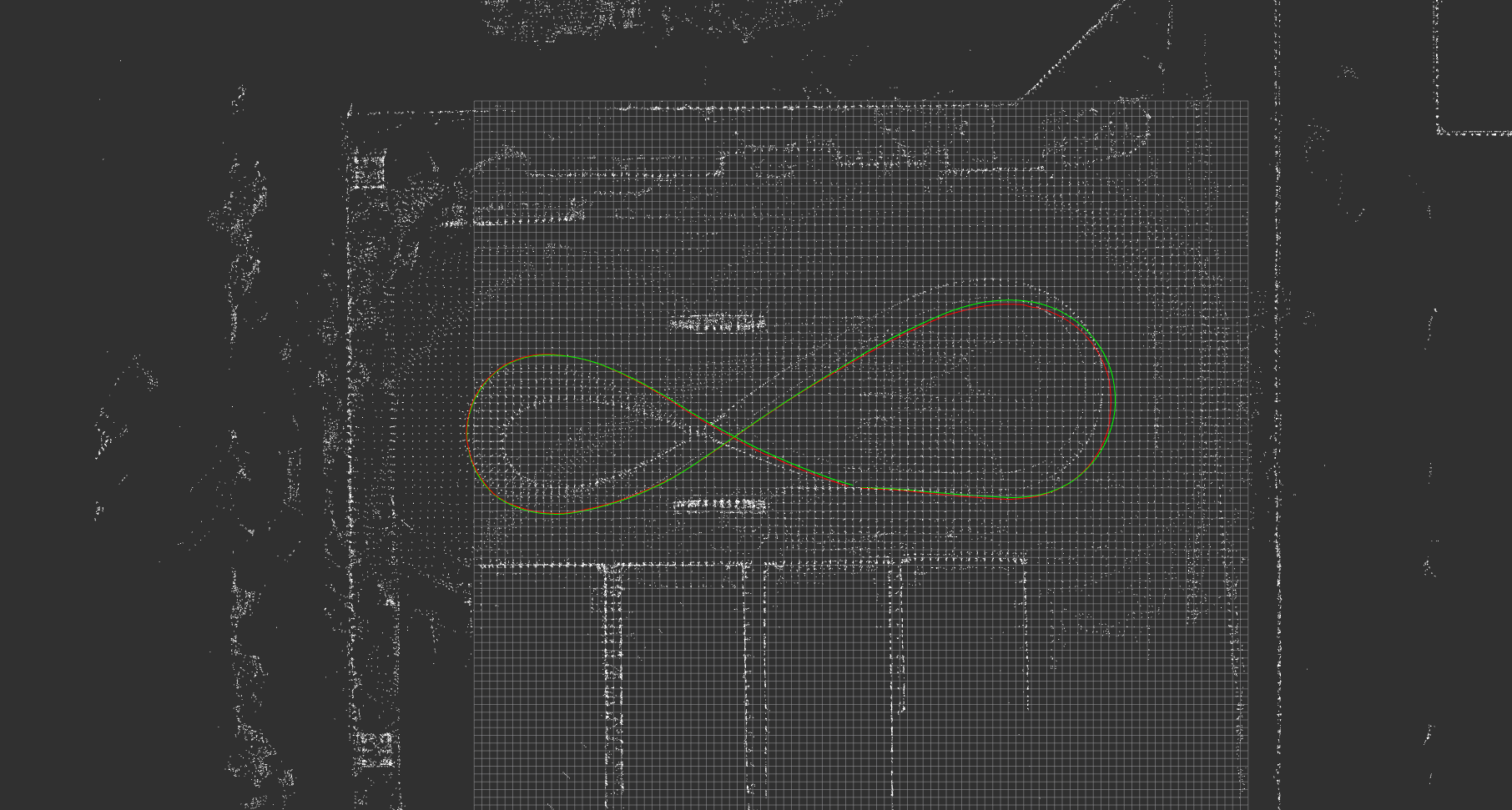
绿色轨迹为gps的结果， 红色轨迹为ndt-tku path在 lidar 坐标系下的轨迹。

绿色轨迹为gps的结果， 红色轨迹为ndt-tku path优化后在 imu 坐标系下的轨迹。

右激光：

绿色轨迹为gps的结果， 红色轨迹为ndt-tku path在 lidar 坐标系下的轨迹。

绿色轨迹为gps的结果， 红色轨迹为ndt-tku path优化后在 imu 坐标系下的轨迹。

优化参数： x: 1.51 y: -1.197 z: -0.7 r: 0.0101 p: 0.0164 y: -0.0328

对比融合后的激光点云， ndt-tku 表现稍差于ndt，具体原因尚不清楚。

未来工作方向：

在模块一的激光里程计部分，可以通过几个方面提高精度：

a. 录制数据包时保持2m/s左右 车速， 宜慢不宜快，减少运动造成的误差

b. 在匹配前加入点云smoothing 的部分，减少噪点

c. 在硬件条件允许的情况下，接入gnss，用精确地时间戳进行运动补偿

d. pcl 的gicp存在bug(<http://www.pcl-users.org/ICP-converges-to-better-solution-than-GICP-td4019799.html>), 可能需要使用segal原始的library去做(<https://github.com/avsegal/gicp>)

e. 在激光里程计后加入pose-graph optimization,以提高每一个位姿的准确度。 其中node为每一个时间点的位姿，vertex为激光点云匹配得到的相对转换关系，用g2o来进行优化.

在模块二单步法求解手眼标定的部分，该算法通过产生大量粗略估计值，来在统计学上达到较优解。 因此可以通过如下方面提升最后的结果：

a. 通过调参达到一个合理的统计学结果，提高产生最优解的概率

b. 研究dual quaternion 解算方法， 想办法提高单次结算结果的准确度