三维点云算法

Date: 2020.07.26

Author: MagicTZ

Homework: 4

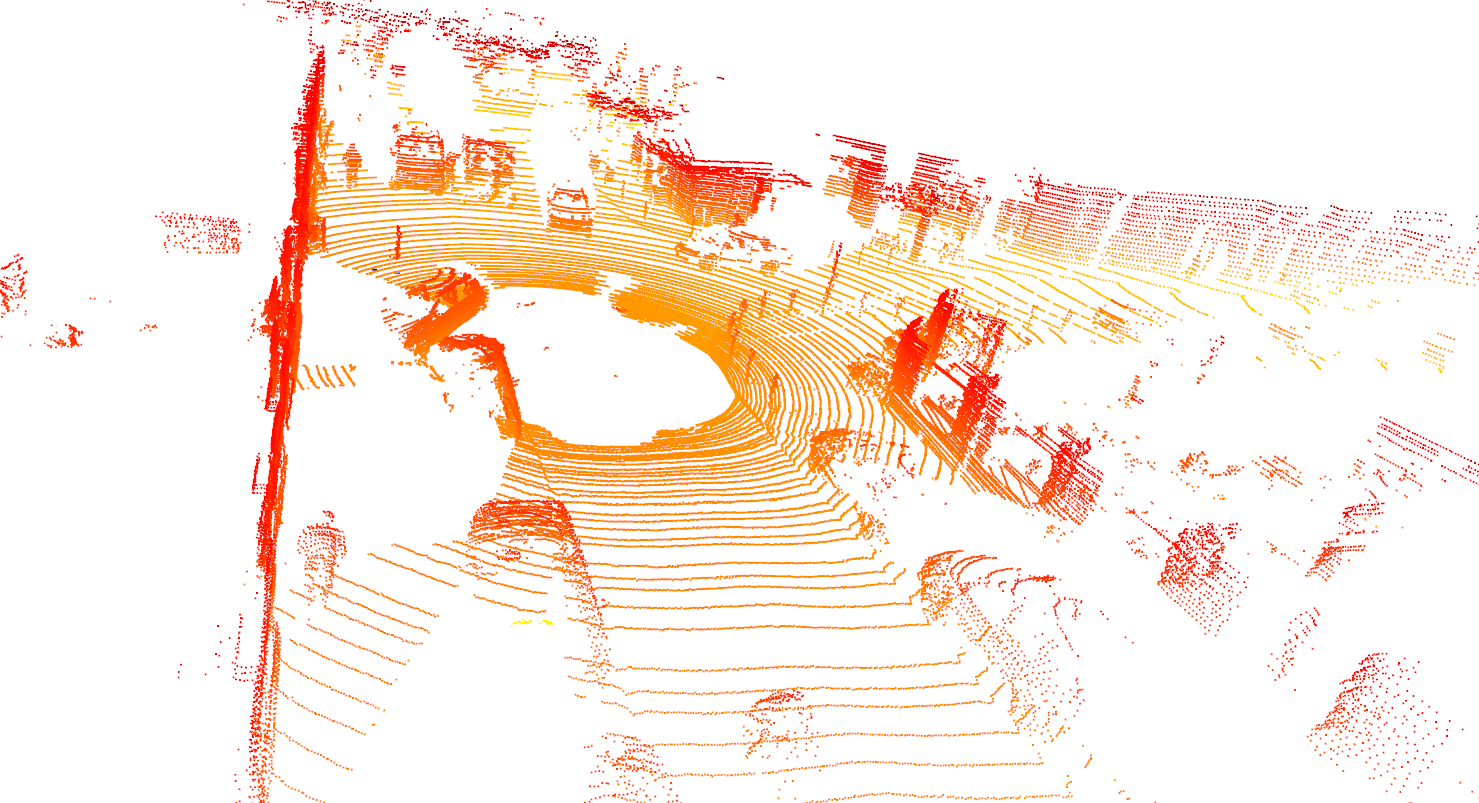
Description:

The implementation of clustering algorithm regarding:

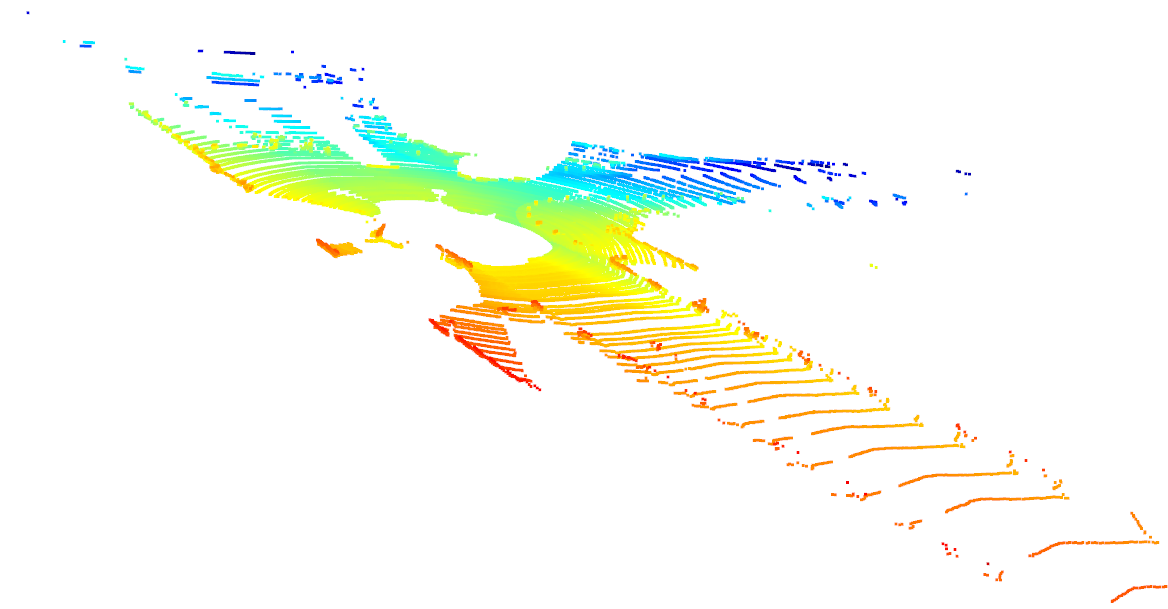
1. Using RANSAC to fit the ground
2. Removing ground and downsampling the non-ground points
3. Clustering the non-ground points by DBSCAN

1 数据处理（以一组数据为例，其他两组数据附在后面）

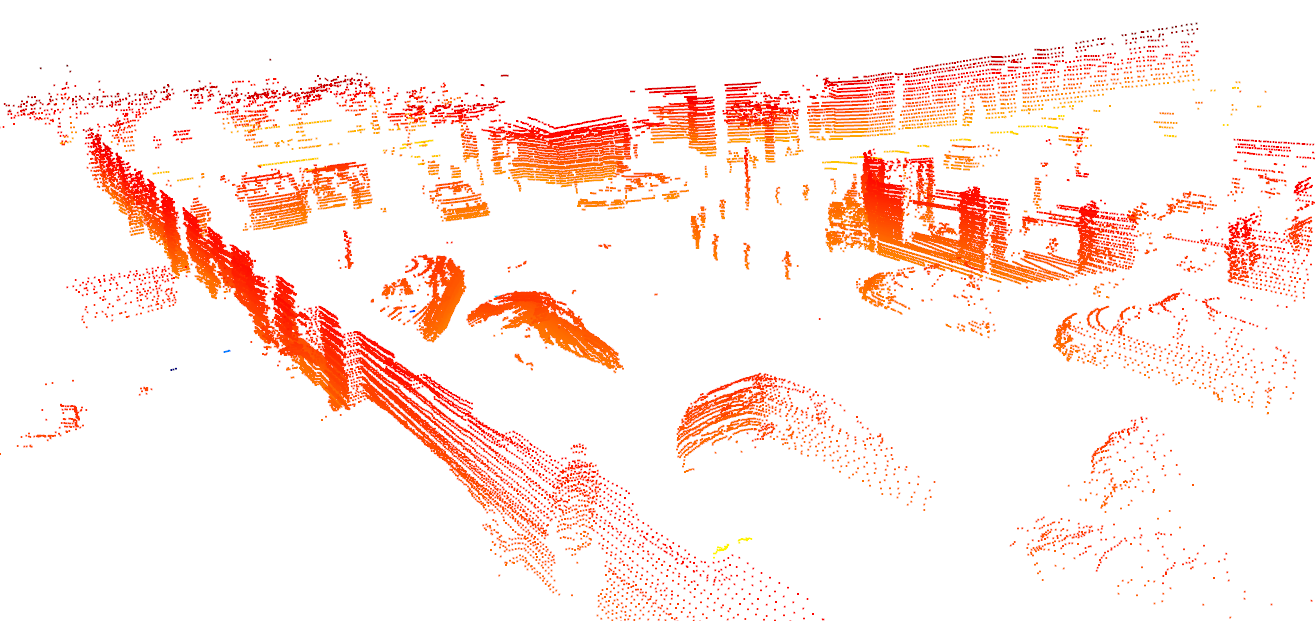
* 1. 原始数据



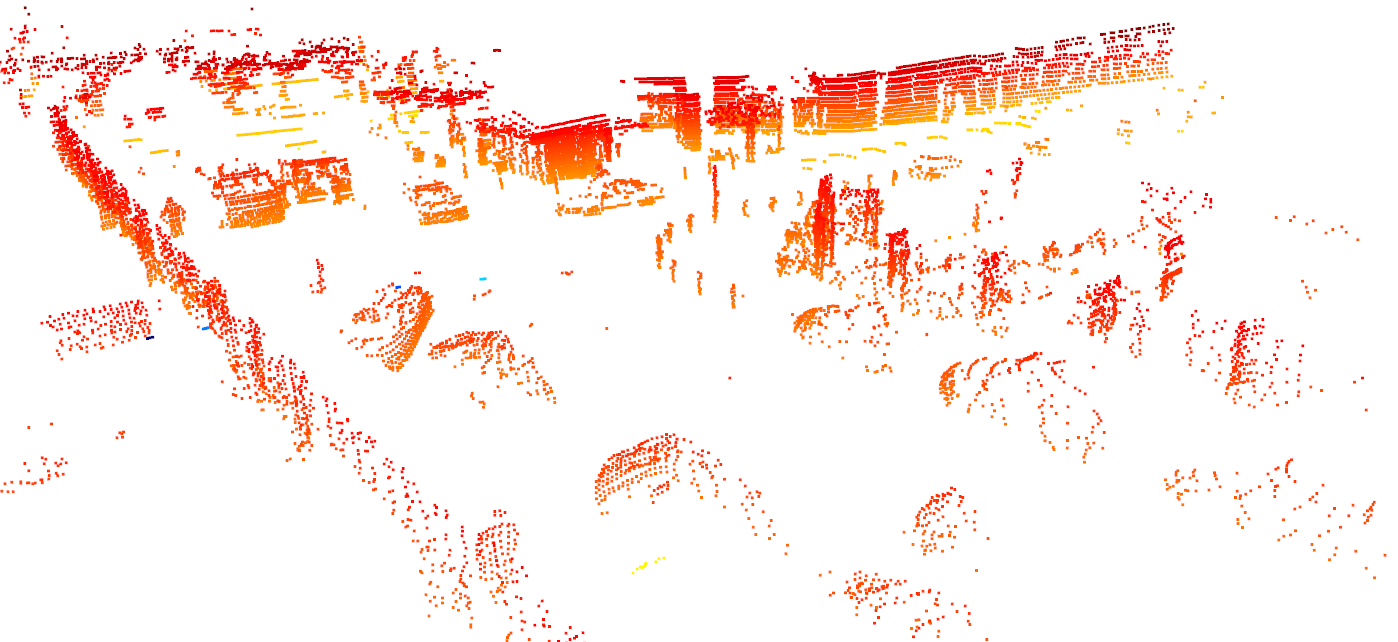
* 1. 地面点



* 1. 非地面点

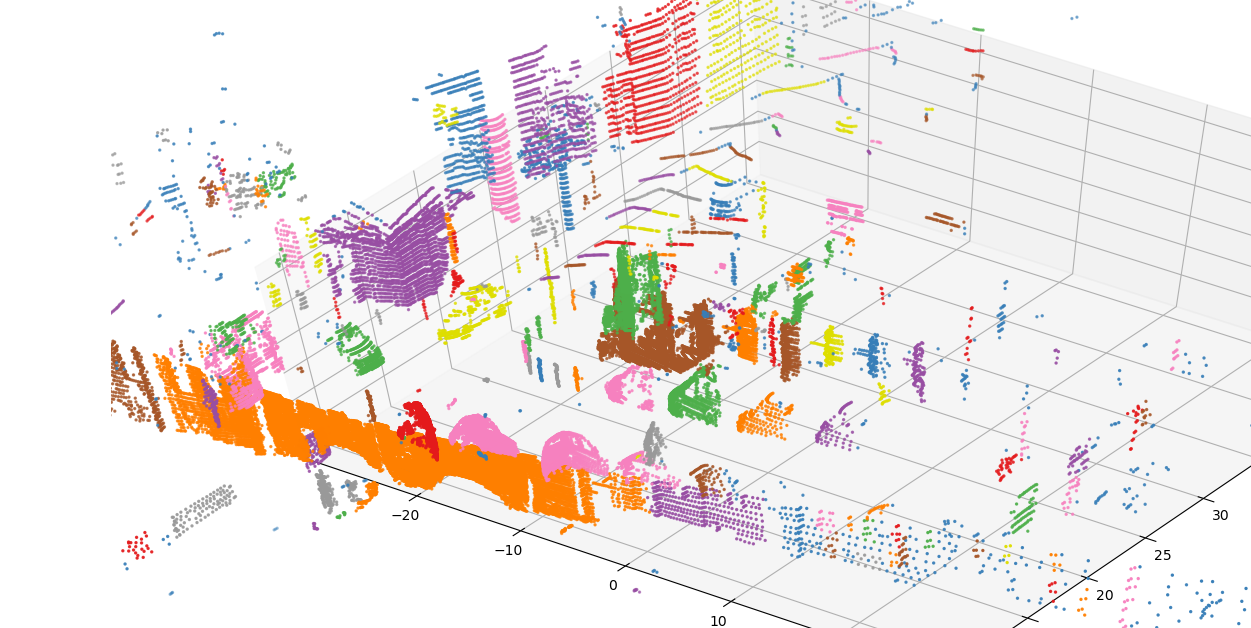


* 1. 下采样

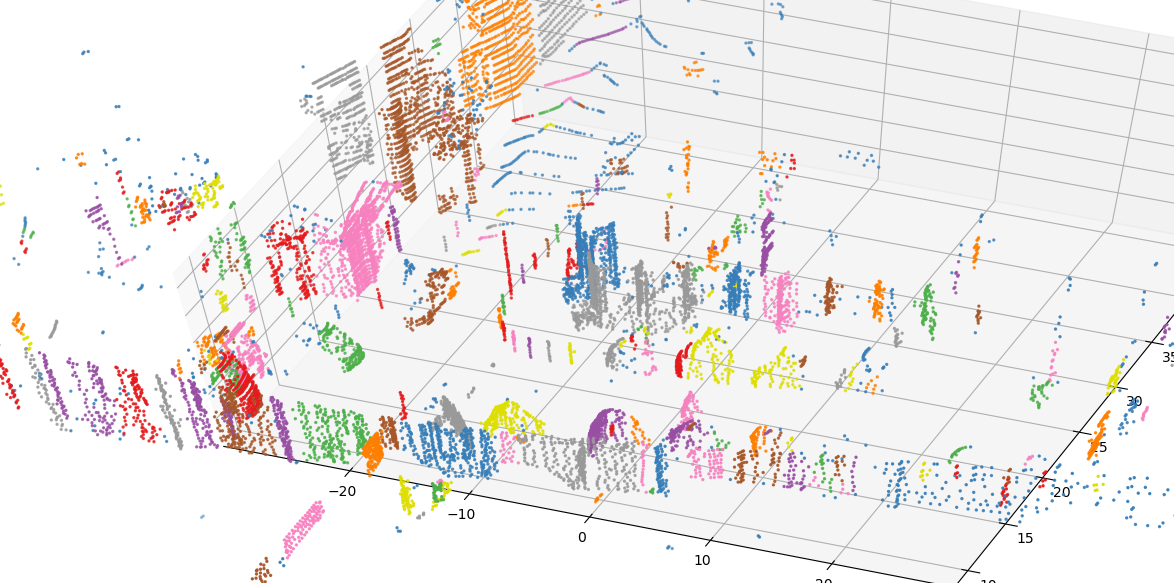


* 1. 聚类

1. 采样前



1. 采样后



1. 数据分析

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 |
| 原始点云(个数) | 115839 | 123397 | 121691 | 123178 |
| 非地面点（个） | 66863 | 49324 | 68779 | 63323 |
| 非地面点(采样后) | 15509 | 25769 | 16190 | 23890 |
| 采样前聚类[s] | 28.116 | 13.3229 | 34.976 | 23.5575 |
| 采样后聚类[s] | 1.542 | 3.764 | 1.581 | 3.677 |

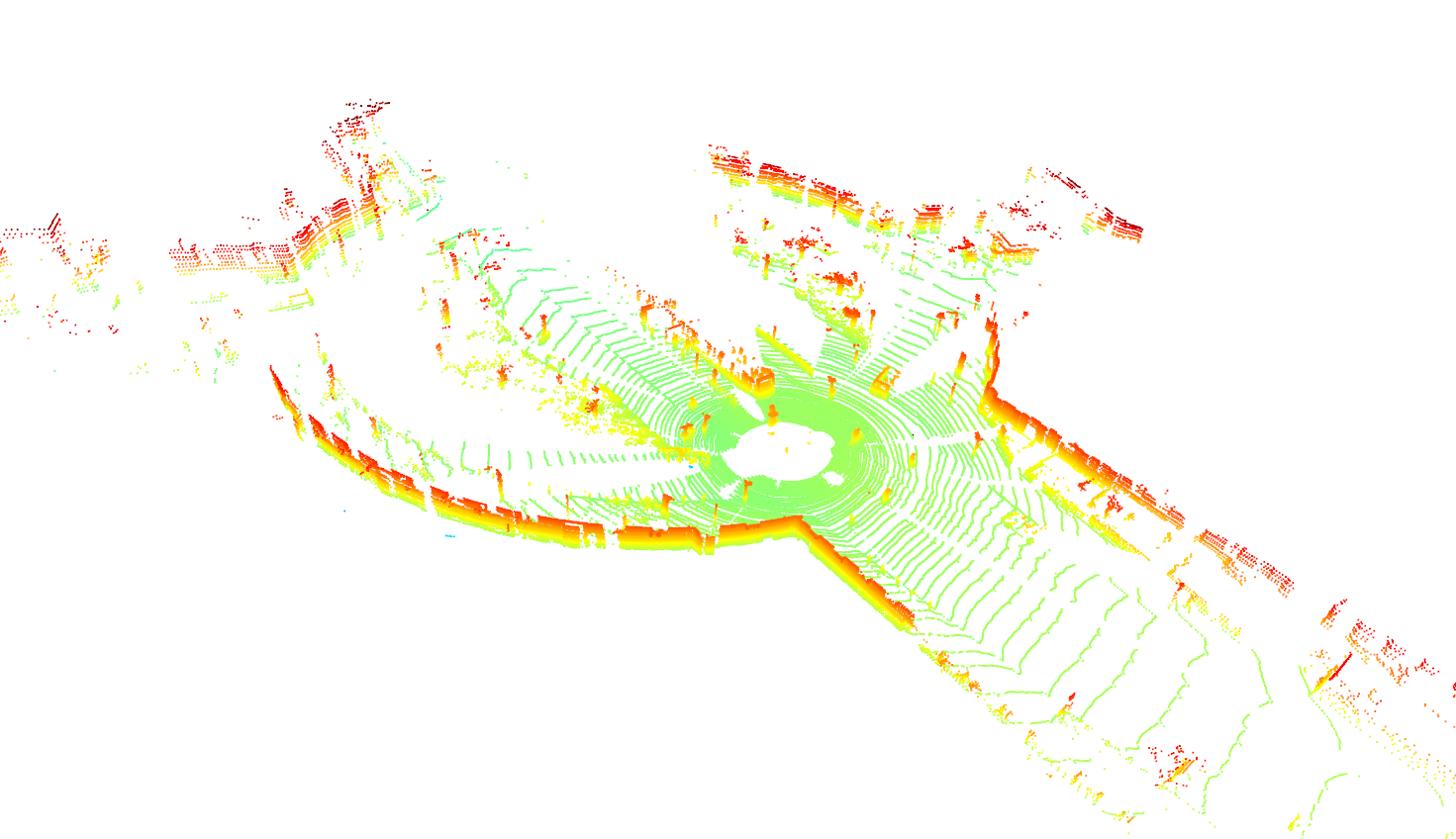
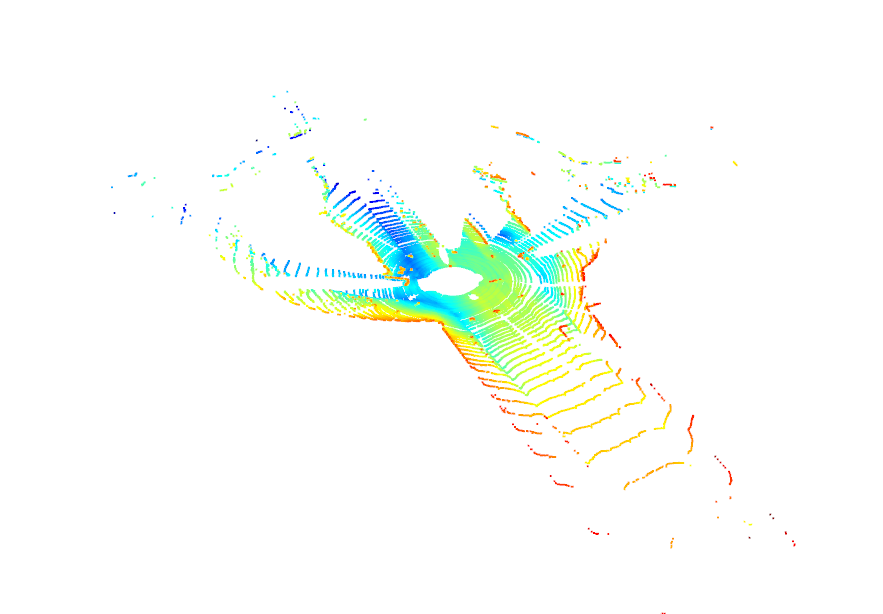
上面的表格结合所有的图片结果可以发现，原始点云往往数据量非常大，如果仅仅对原始点云进行处理，效率会非常低，虽然结果比较好。因此在本次实验中，首先使用RANSAC的方法获取地面点数据，然后计算非地面点点云并进行下采样（使用之前voxel\_filter函数），降低数据的规模，然后再使用DBSCAN的方式对点云进行聚类（首先需要利用kdtree找到所有核心点，不能使用欧式距离，矩阵太大，矩阵运算速度巨大），可以发现，采样后的结果不会对最后聚类的结果产生很大影响，但却使聚类速度有了巨大的提高。

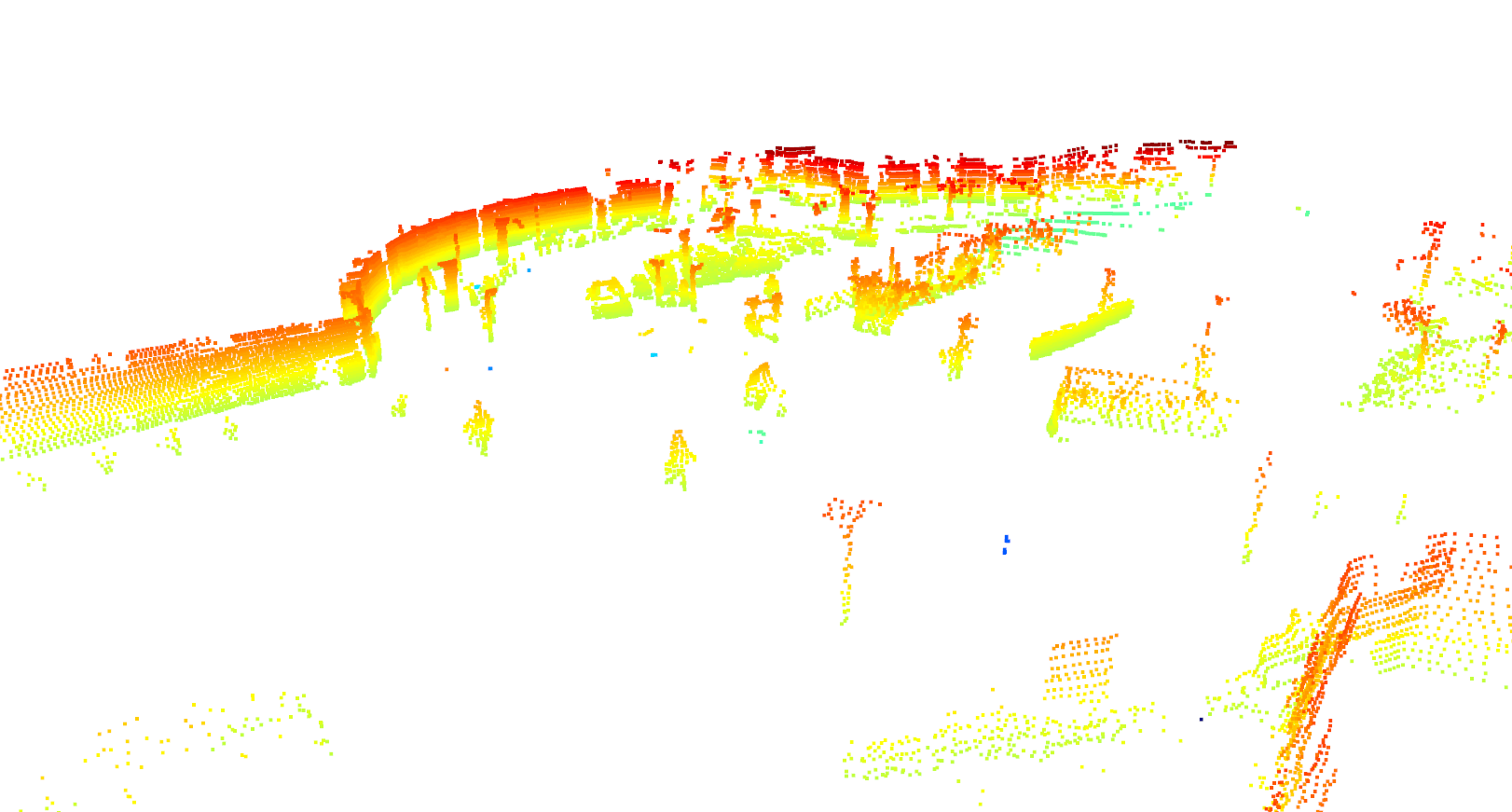
3 问题

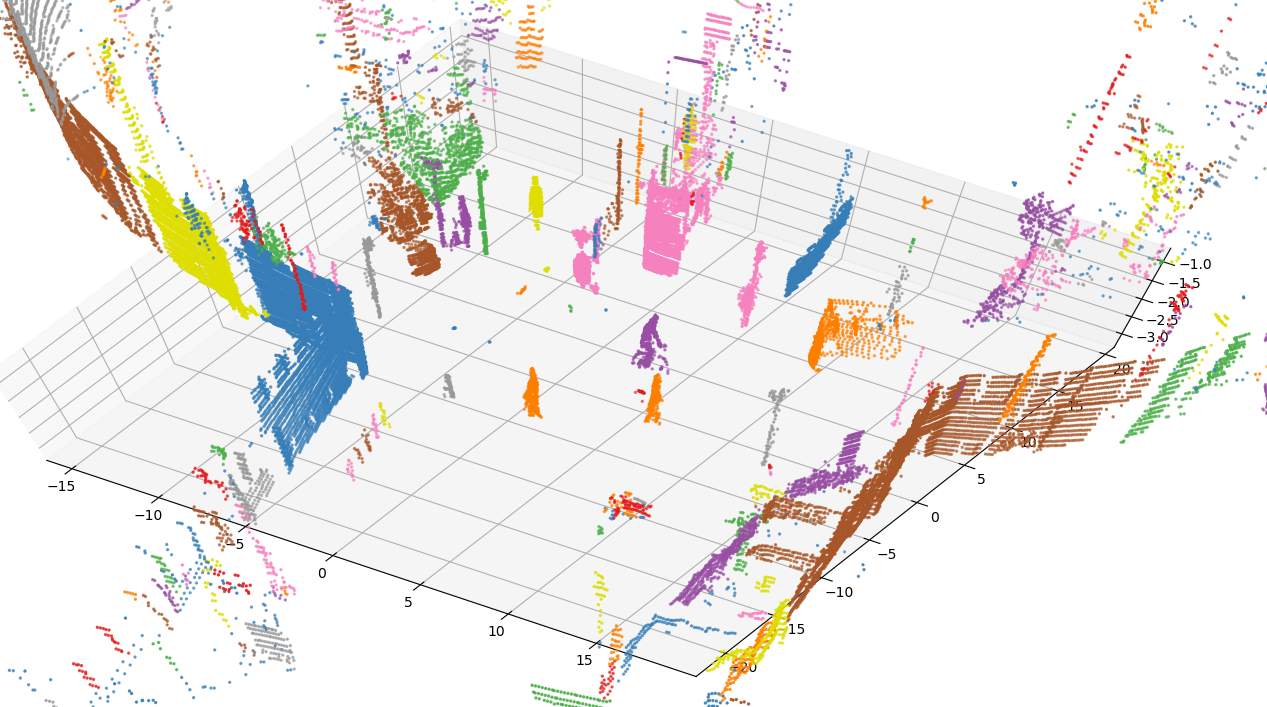
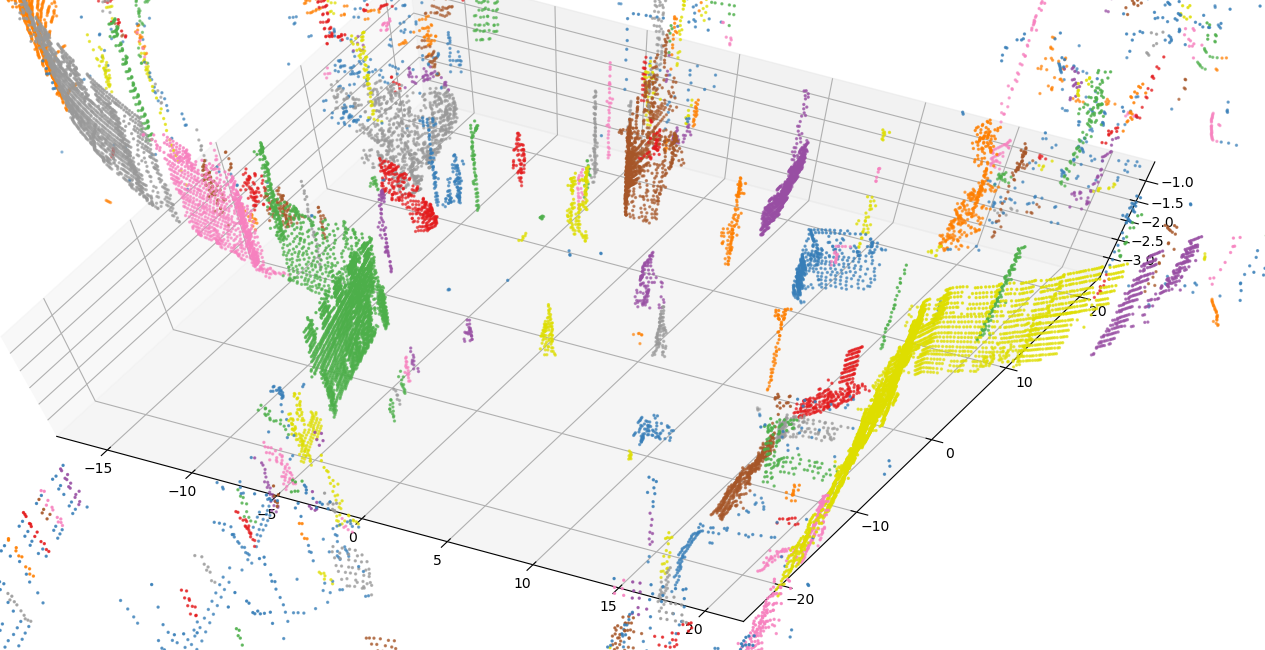
1. 当前的算法无法很好的检测出斜坡，因为如果阈值过大，会对路沿进行无检测，优化思路是对路面进行分段检测，但时间上肯定会大打折扣。
2. 如何平衡算法的速度和性能。如果参数设置过大，误检测的概率会小很多，但是搜索的速度往往影响会更大，而且数据特点的不同，参数的设置也会随之改变，一组参数不能对应所有的情况。

4 其他几组结果

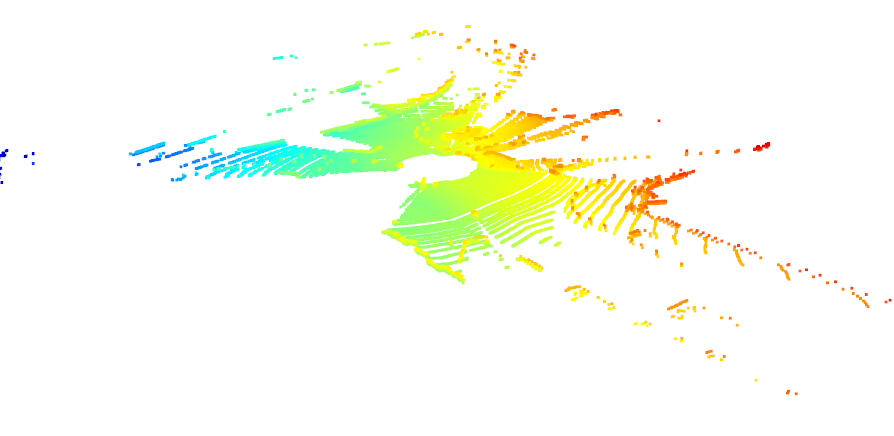
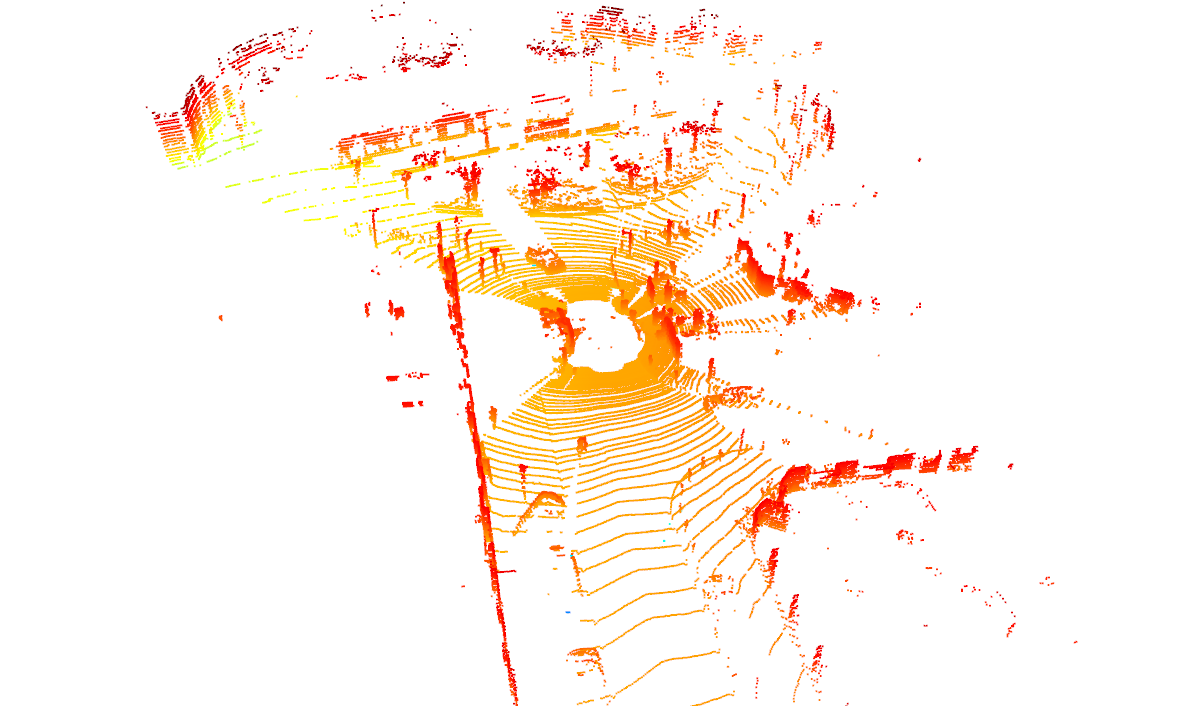
（1）第二组数据

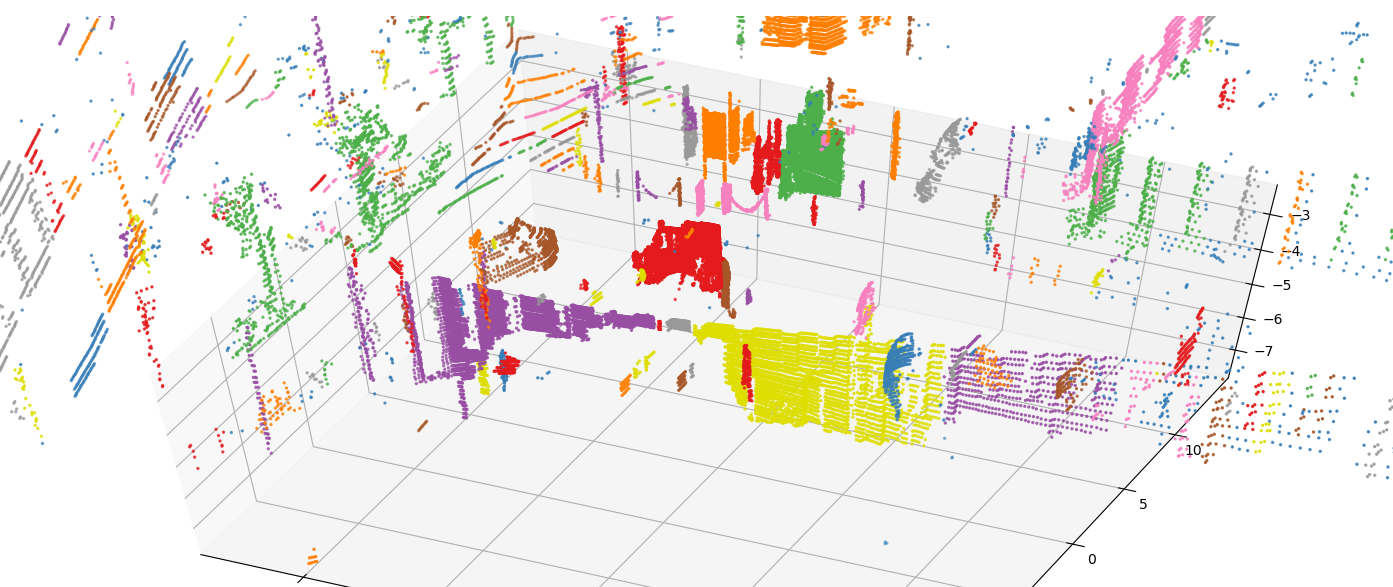


（2）第三组数据







1. 第四组数据

