OPTIMIZING LOAM PERFORMANCE

TX.Cui
IMORPHEUS.AI
05/01/2018 12:00PM

iMorpheus.ai

Journal Club

Journal club介绍与自动驾驶中定位方案相关的论文， 主要关注的方向有：SLAM算法、点云数据的处理和压缩、 特征地图、传感器数据处理和融合、 GNSS信号处理等。我们一直关注领域前沿技术， 选取得到广泛认可的、或者是在我们的实际使用中结果比较好的论文， 与大家分享，共同学习成长。

每周五 北京时间12点

http://imorpheus.ai/journalclub

扫码加入无人驾驶技术群

iMorpheus.ai

THE MAIN CONTENTS

- Introduce the principle of LOAM
- Show optimization methods and results
- Other optimization ideas

iMorpheus.ai

KITTI-Visual Odometry / SLAM Evaluation 2012

Additional information used by the methods

- Stereo: Method uses left and right (stereo) images
- Laser Points: Method uses point clouds from Velodyne laser scanner
- Loop Closure Detection: This method is a SLAM method that detects loop closures
- Additional training data: Use of additional data sources for training (see details)

	Method	Setting	Code	Translation	Rotation	Runtime	Environment	Compare
1	V-LOAM			0.63 %	0.0014 [deg/m]	0.1 s	2 cores @ 2.5 Ghz (C/C++)	

J. Zhang and S. Singh: Visual-lidar Odometry and Mapping: Low drift, Robust, and Fast. IEEE International Conference on Robotics and Automation(ICRA) 2015.

2	LOAM			0.64 %	0.0014 [deg/m]	0.1 s	2 cores @ 2.5 Ghz (C/C++)	

J. Zhang and S. Singh: LOAM: Lidar Odometry and Mapping in Real-time. Robotics: Science and Systems Conference (RSS) 2014.

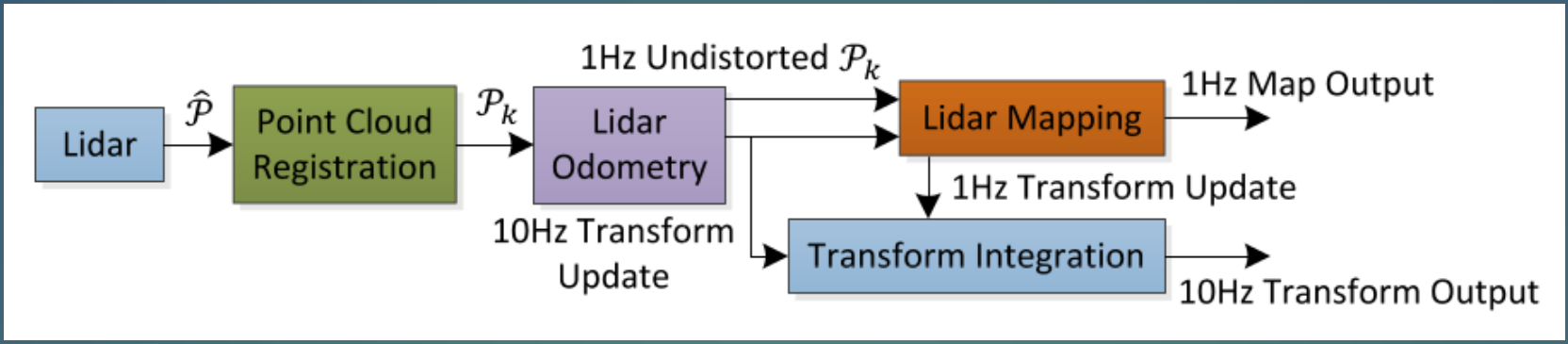
1) LOAM: Lidar Odometry and Mapping in Real-time.

2)On Degeneracy of Optimization-based State Estimation Problems.

iMorpheus.ai

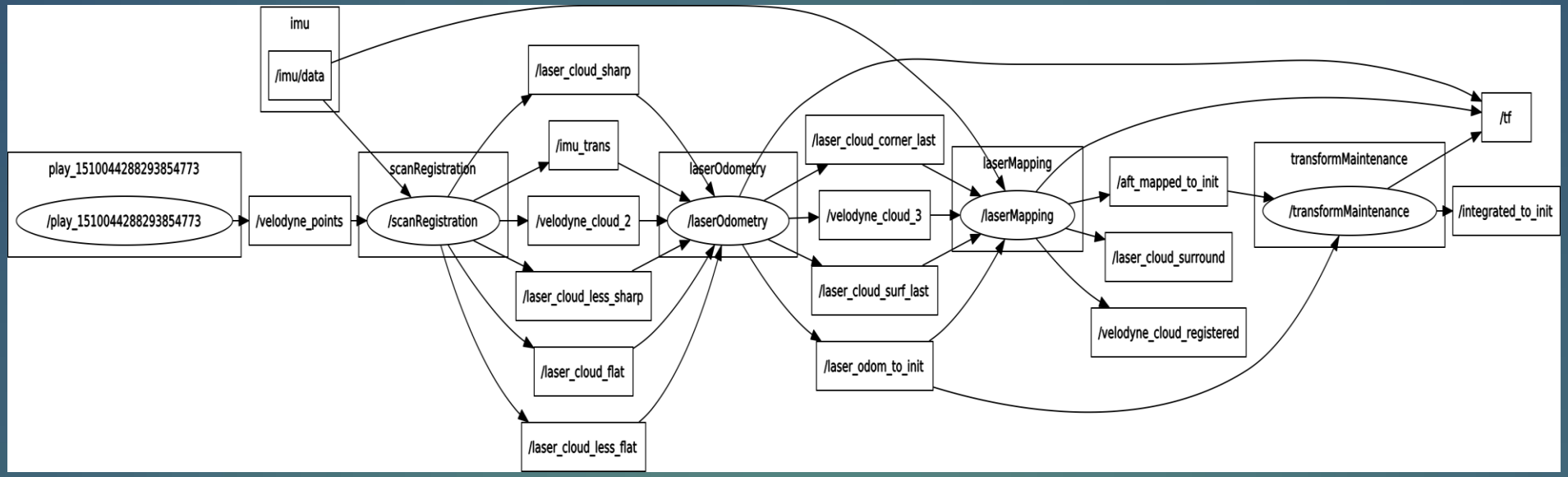
Structure and data flow of LOAM

The modules of LOAM:



Structure and data flow of LOAM

The data flow of LOAM:



the Optimization of LOAM

Purpose:

- Reducing the computation

- Improving the accuracy

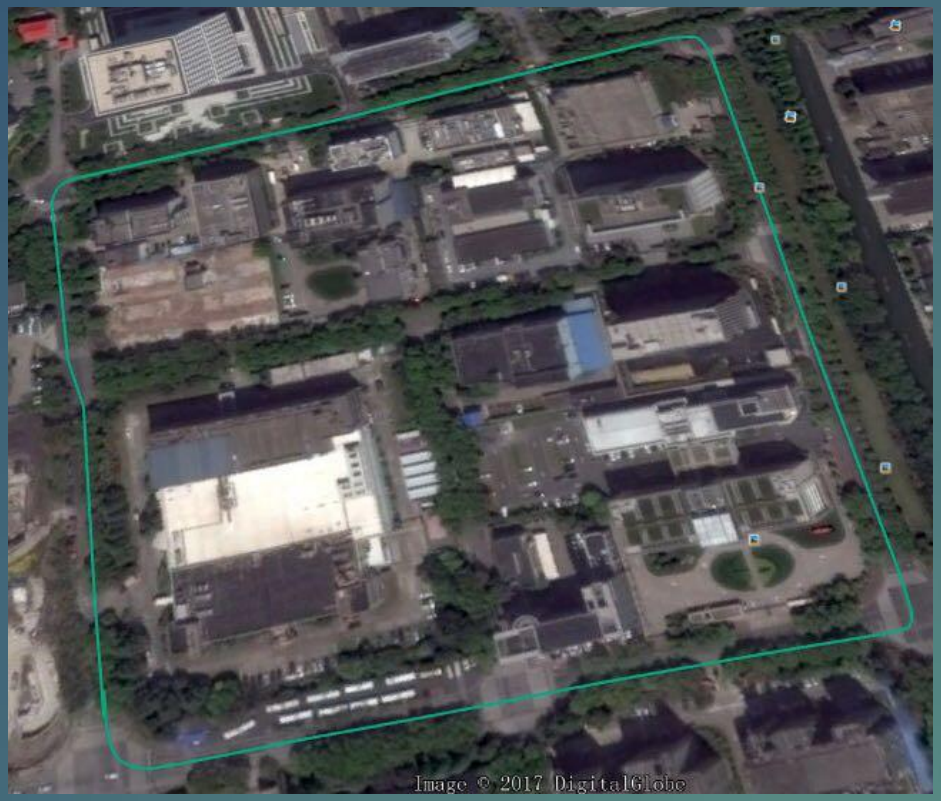
Means:

- Remove the lower scans

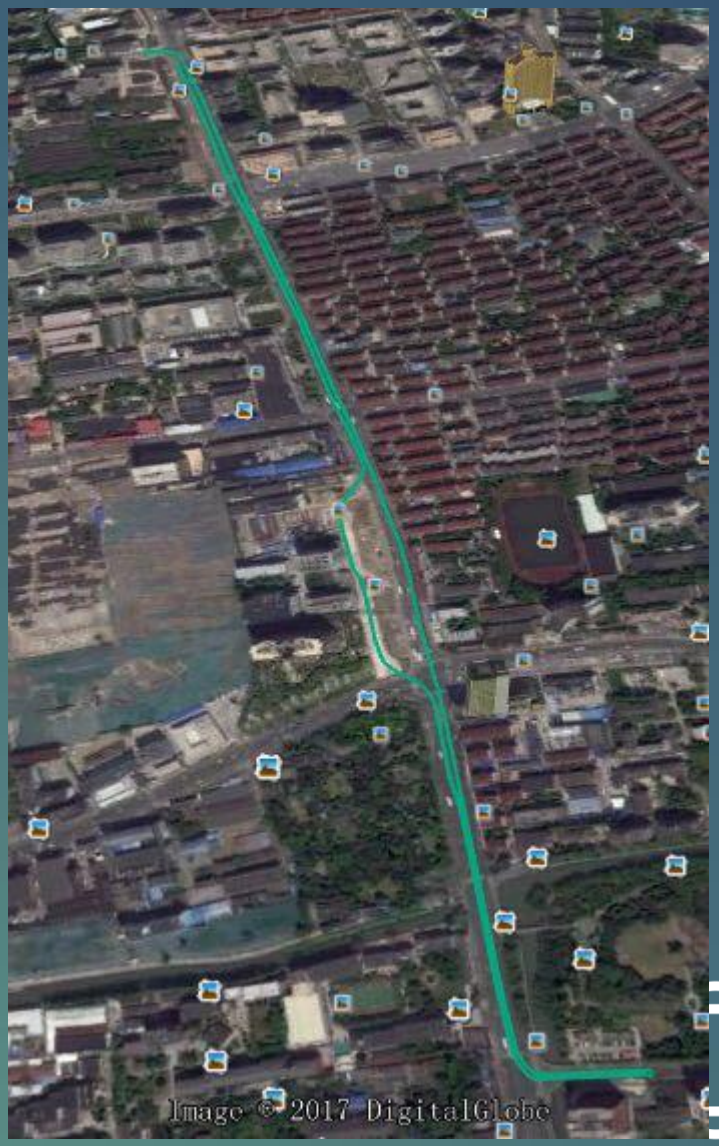
- Remove the dynamic noise--Filter out the points within the specified distance/Grid filtering points through height and quantity constraints

The routes used

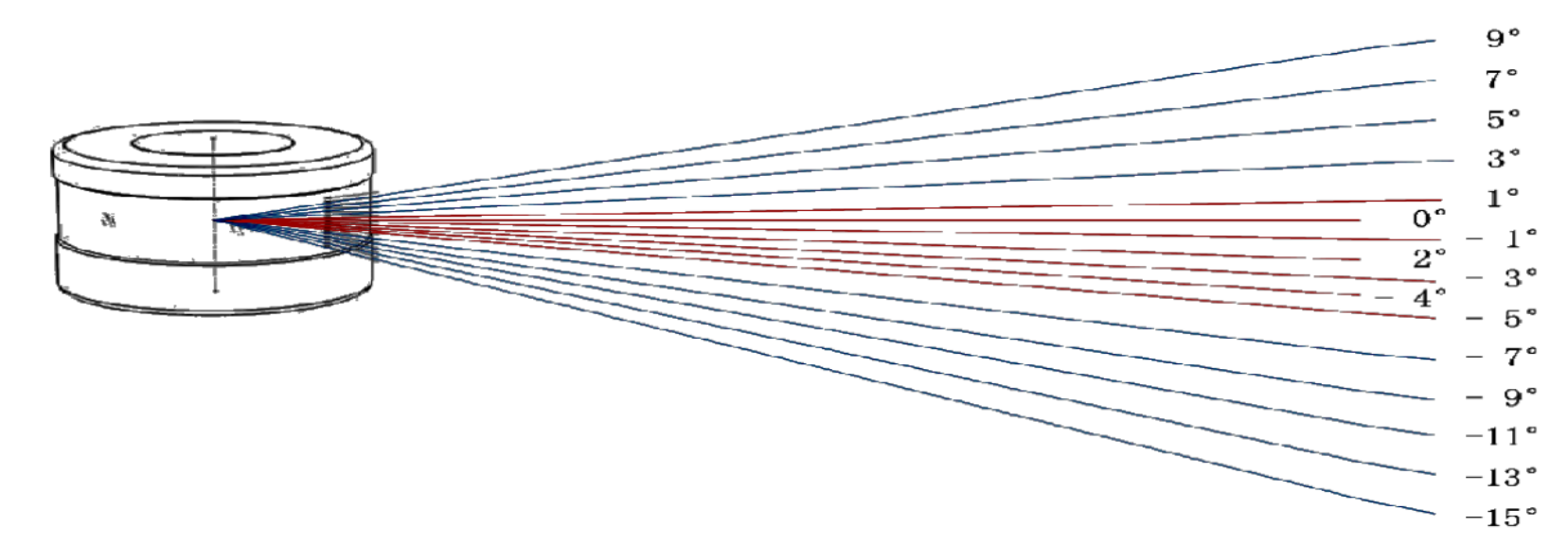
route 1(~1.2km):



route 2(~5km):

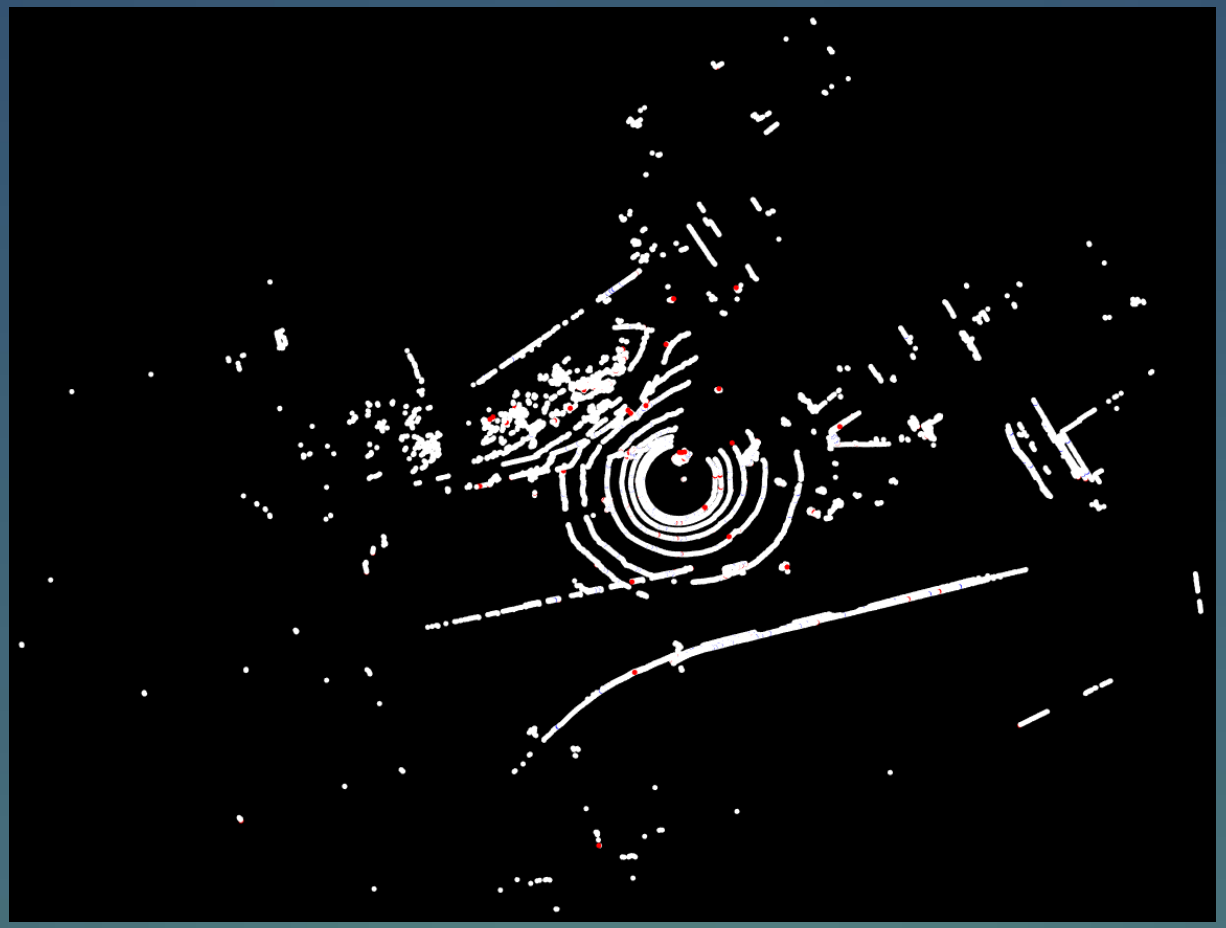


why remove the lower scans?



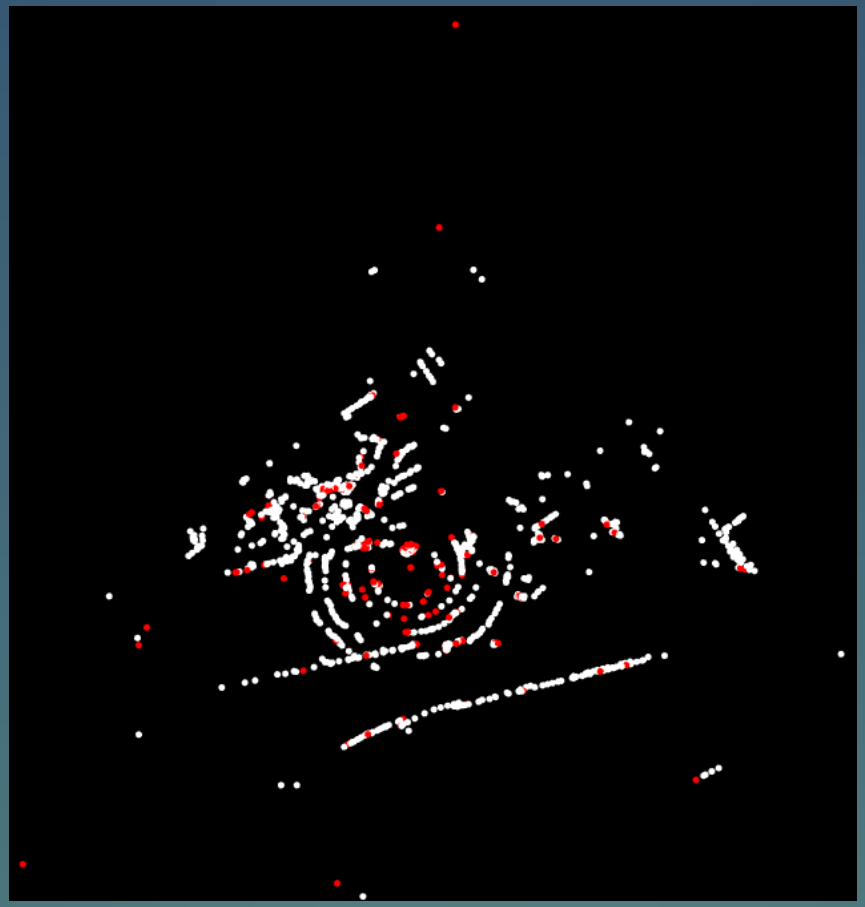
HDL-64E S3			
Lower Laser Block	Upper Laser Block	Vertical Field of View (VFOV)	Primary Application
32 lasers separated by 1/2° vertical spacing	32 lasers separated by 1/3° vertical spacing	+2° to 24.8°	Autonomous navigation

VFOV of lidar(r-fans-16 and velodyne)

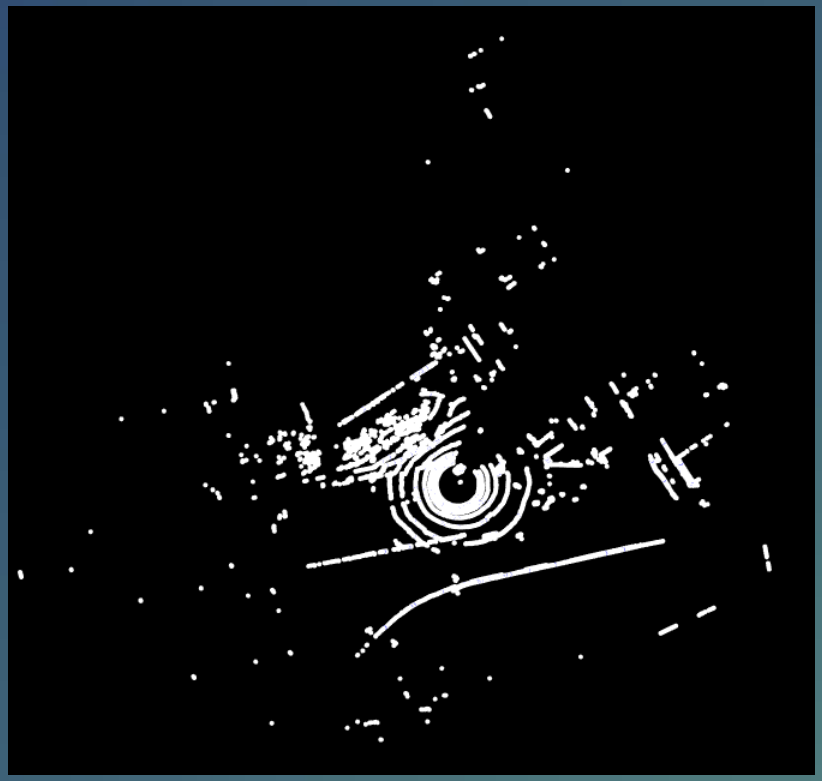


LOAM: All points in a frame

A frame:The points all scans received in one cycle
Red points:Sharp Points
Blue points:Flat Points



All LessSharp Points in the same frame(including Sharp Points)



All LessFlat Points in the same frame(including Flat Points)

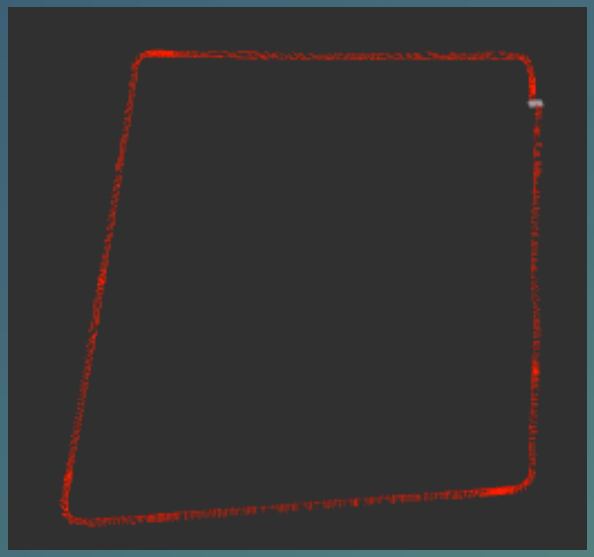
Discovery

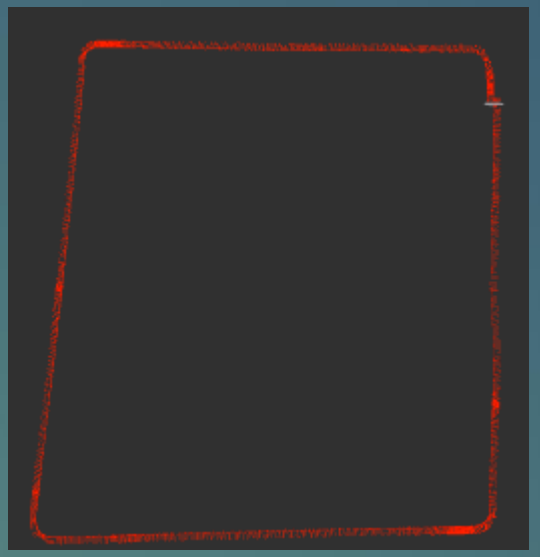
- Lower scans contains many feature points(Sharp Points and Flat Points), and these points ,lay on the road,will have a bad effect on odometry.

Removing these lower laser scans will reduce a lot of computation and help to improving the accuracy

Let`s remove the lower 5 scans, and keep 11 scans only.

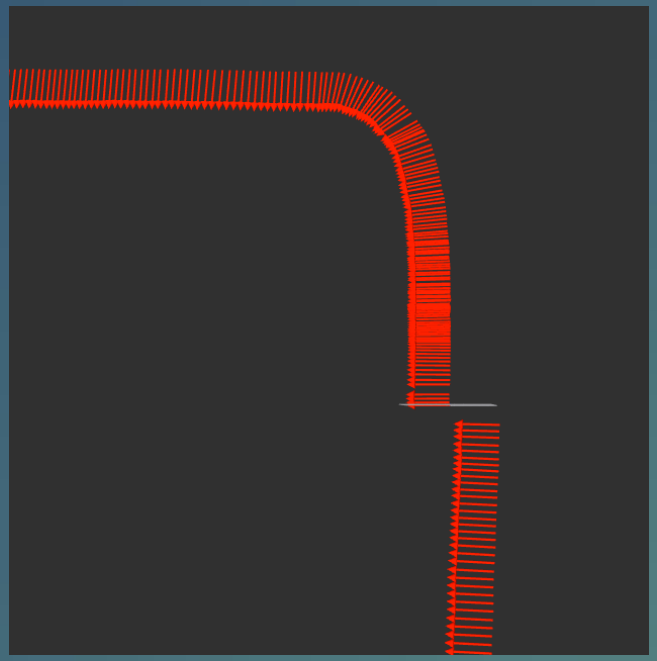
Odometry of route 1

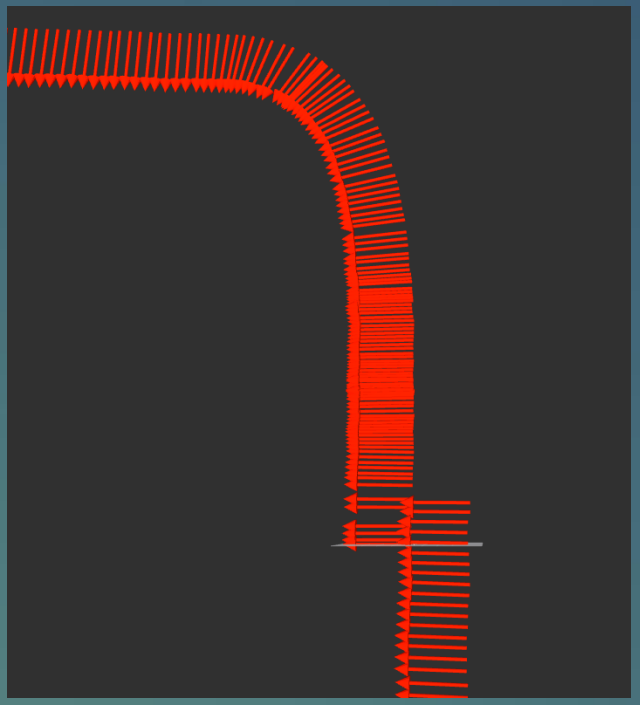




The left is LOAM odometry with 16 scans, and the right with 11 scans .

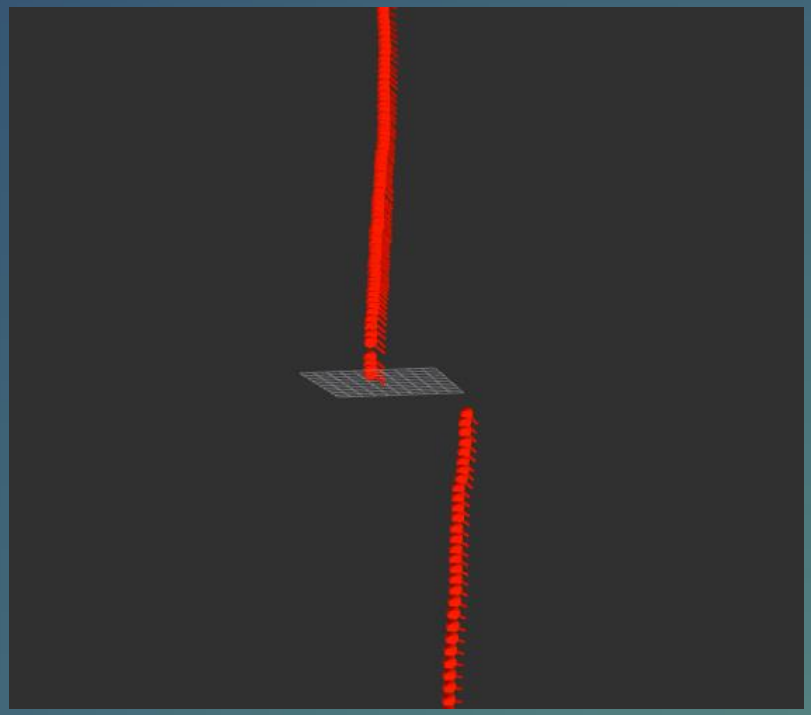
Closed local trajectories

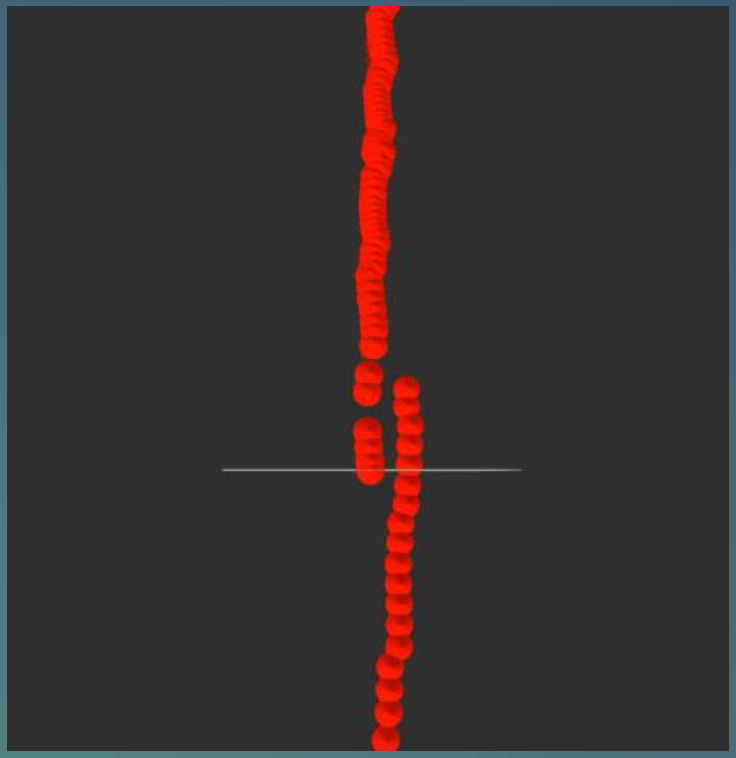




The translation of the upper 11 scans is better than that with all 16 scans.

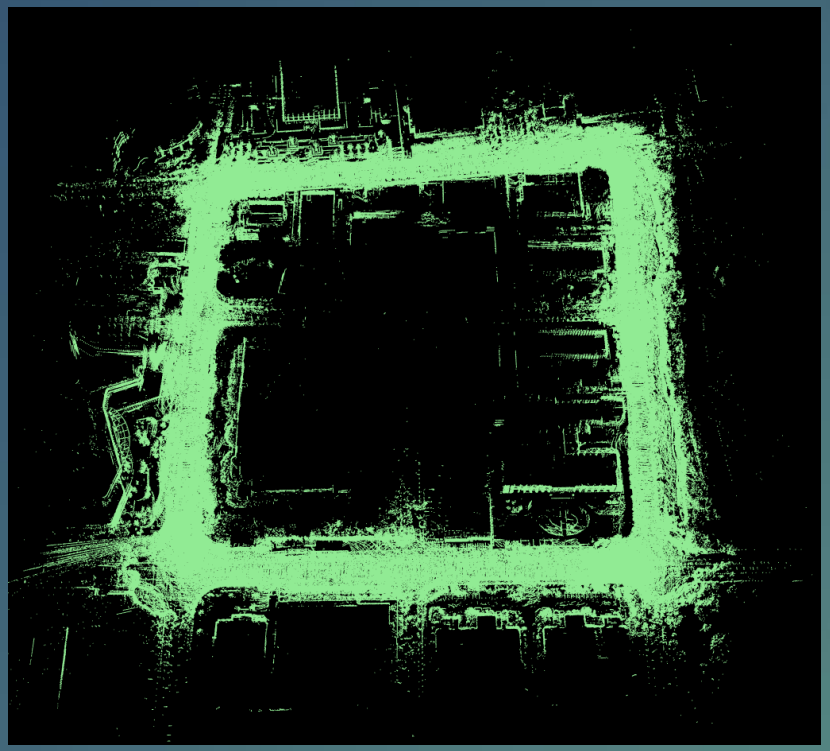
Closed local trajectories

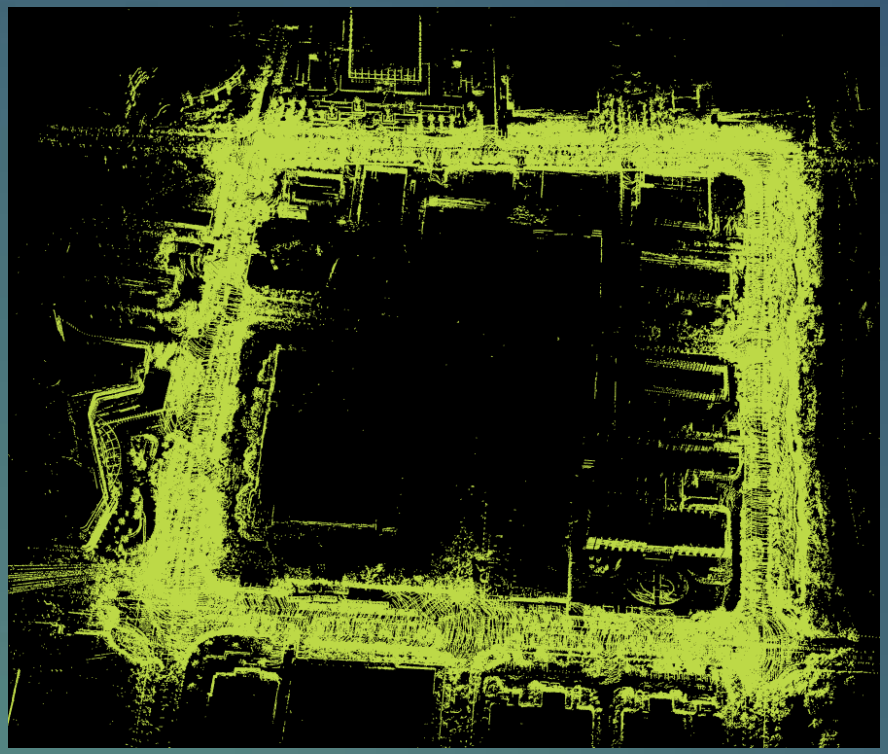




The rotation of the upper 11 scans is better than that with all 16 scans.

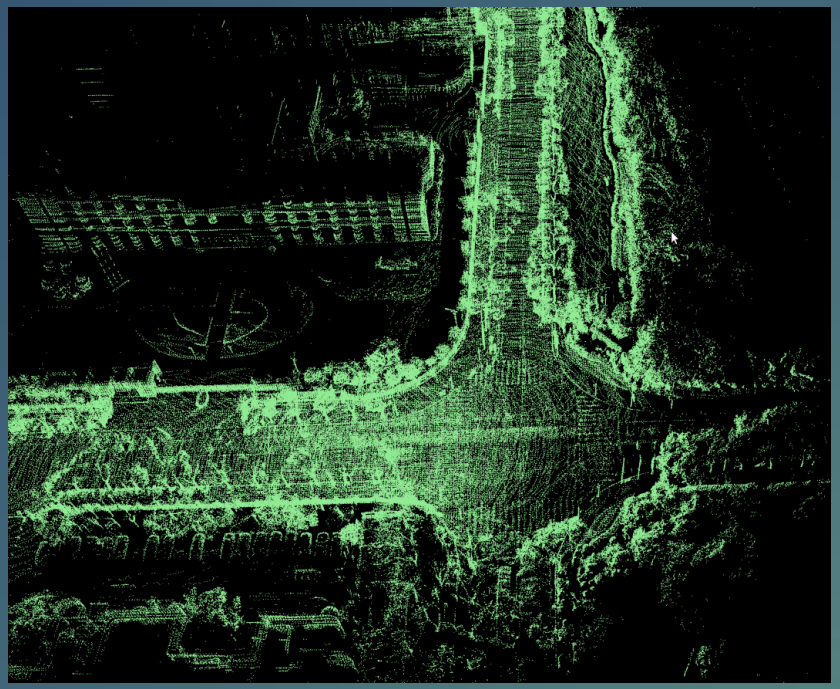
Mapping

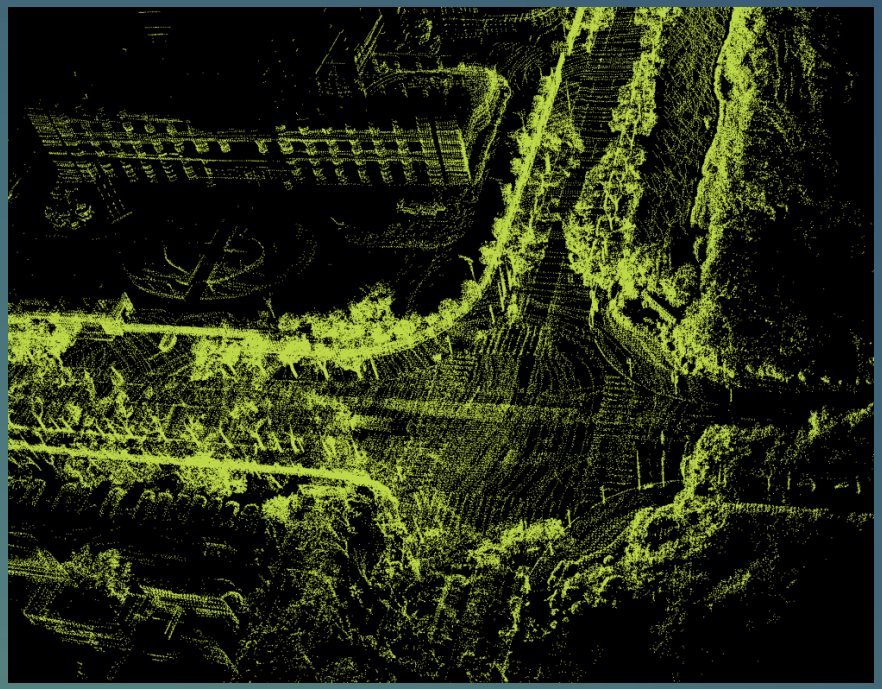




The left is LOAM mapping with 16 scans, and the right with 11 scans .

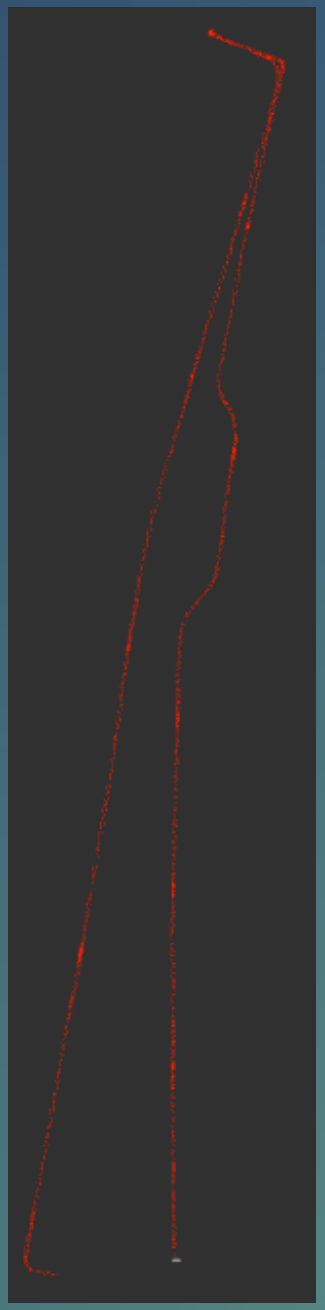
Local Mapping



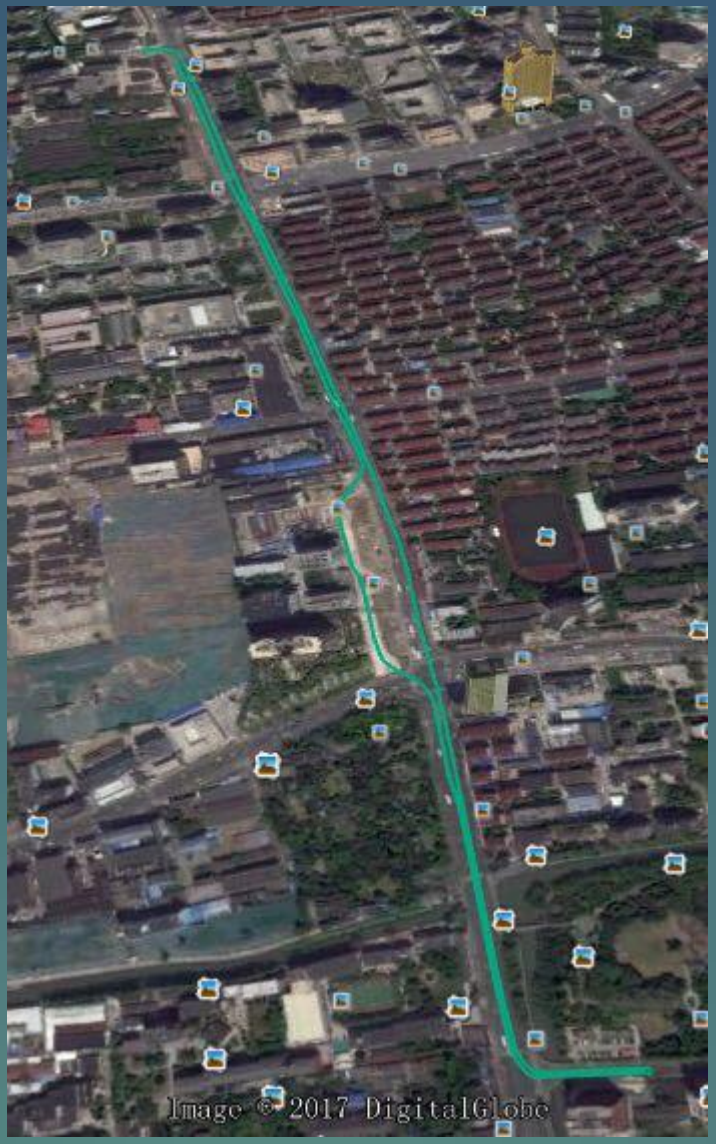


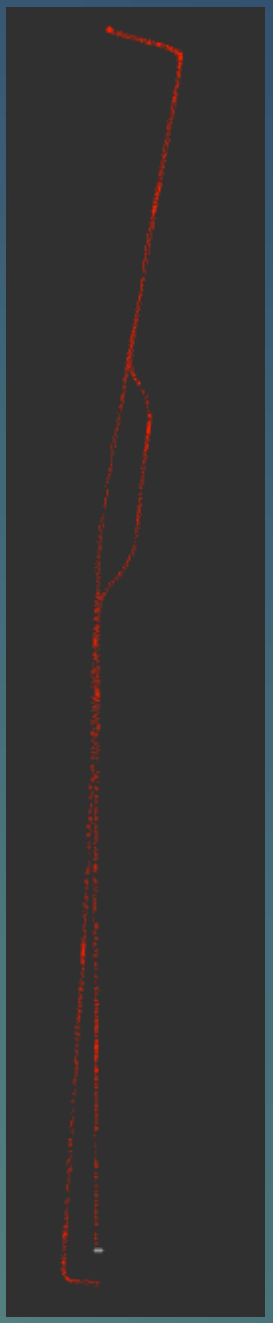
The mapping with 11 scans has less noise points.

route 2



16 scans





11 scans

iMorpheus.ai

Conclusion

- With the suitable selected scans,we can reduce lots of computation and get better results of odometry and mapping than all scans.

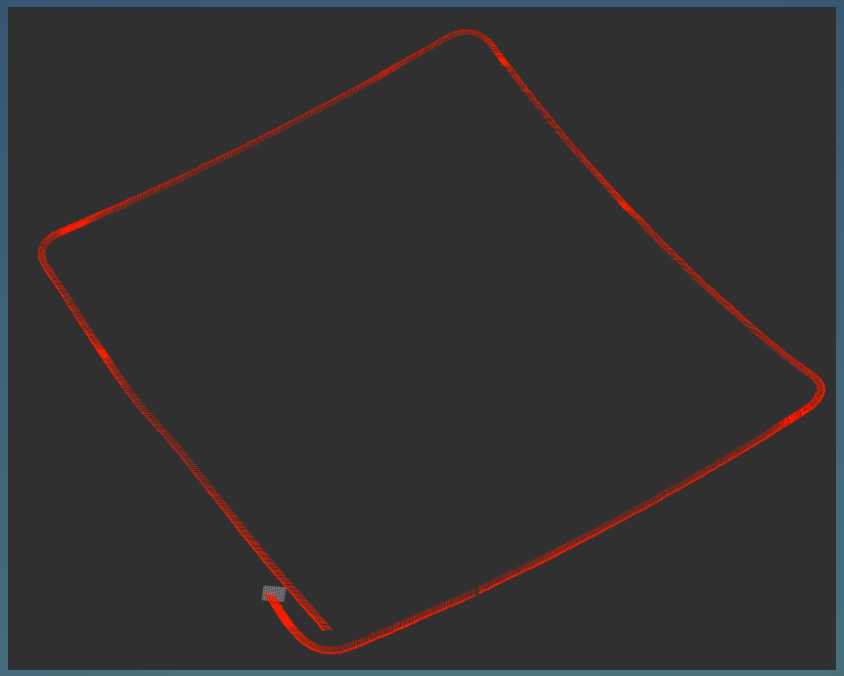
Grid filtering points through height and quantity constraints

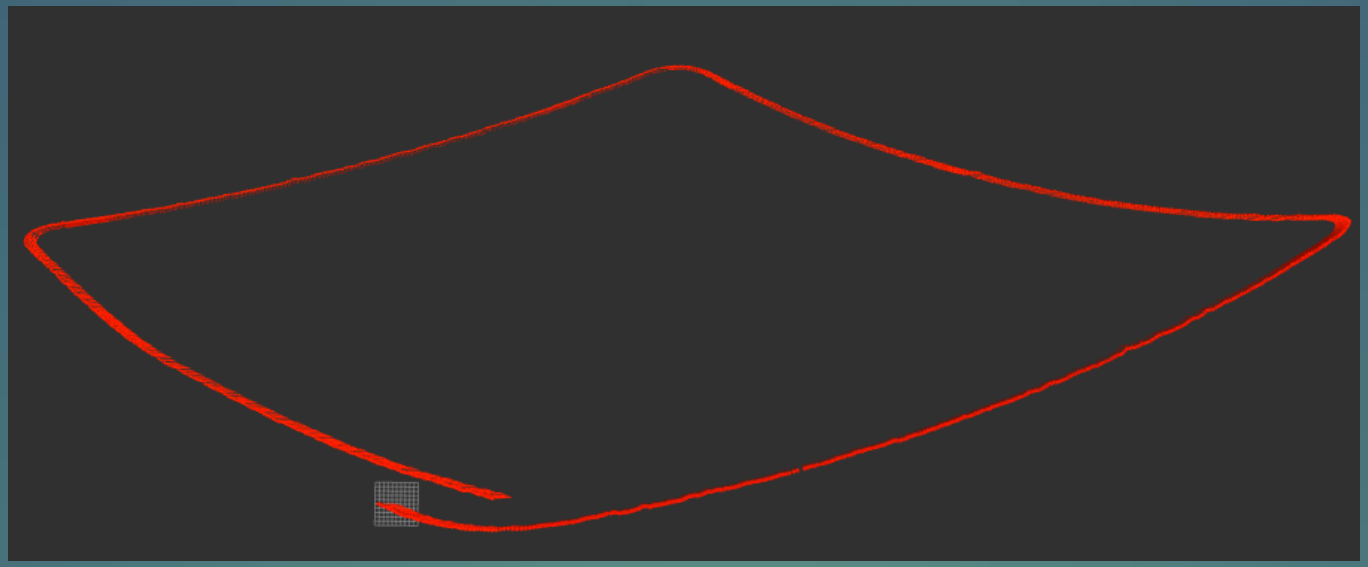
1.Every frame points are projected to 2-d grid (e.g. 0.5m*0.5m in x-y plane)

2.Statistics for the number of points that higher than a height(e.g. 1.7m in the direction of z) in each grid, noted n

3.The points in that grid ,its n value greater than a number(e.g. 3), will be kept

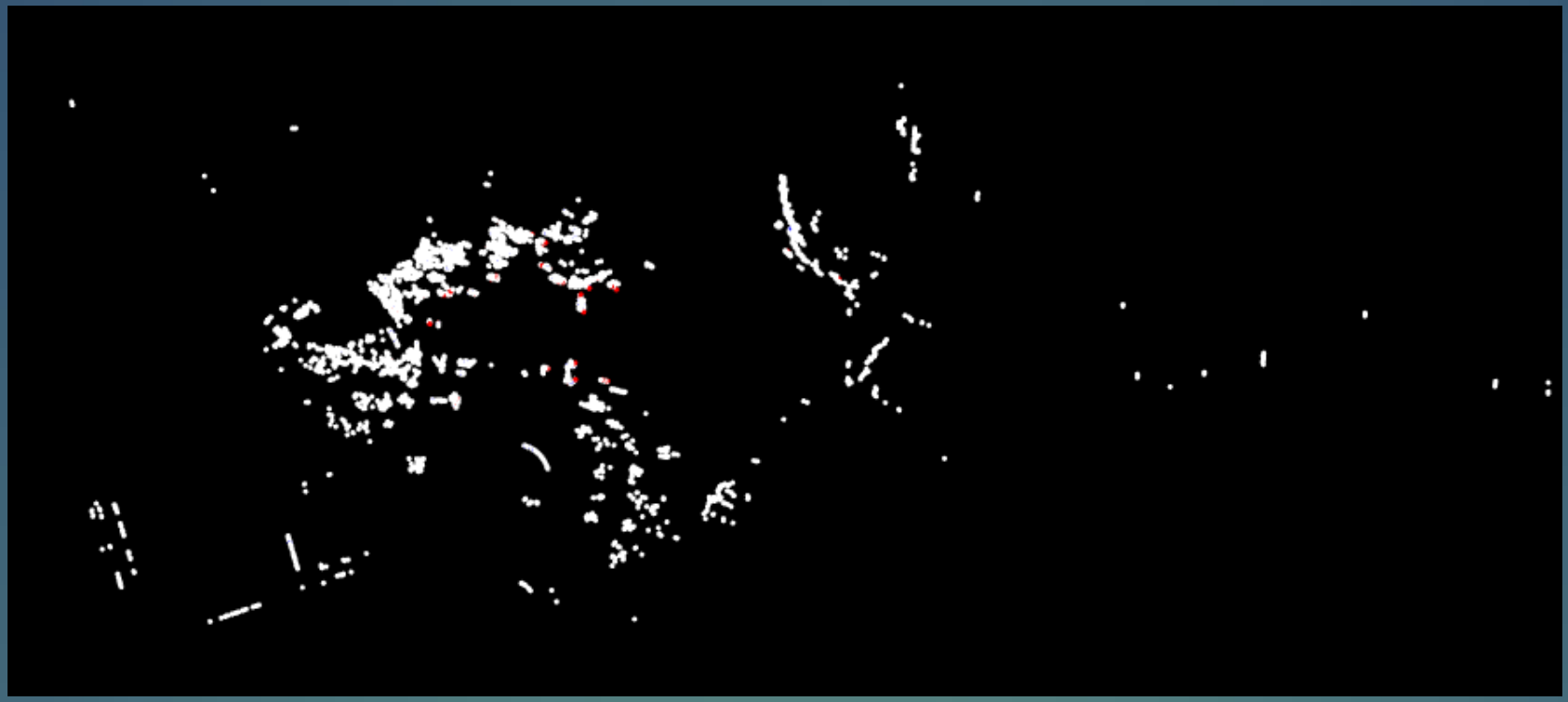
“The Classification of Laser Points and Recognition of Moving Target in Campus Environment”(WHUT, Du Hui, 2013)





The translation seems ok, but the rotation is not so good...

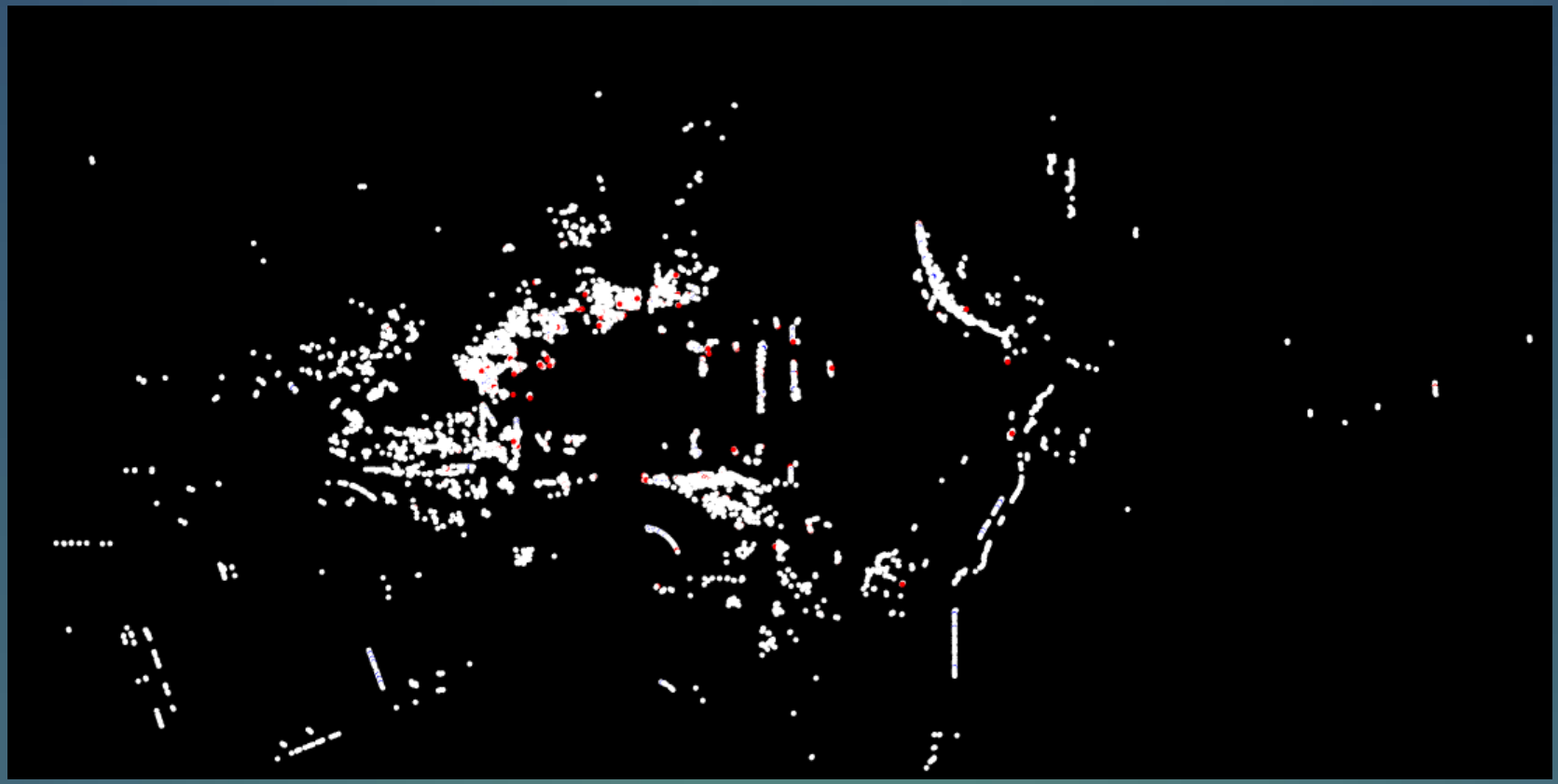
iMorpheus.ai



The filtered frame

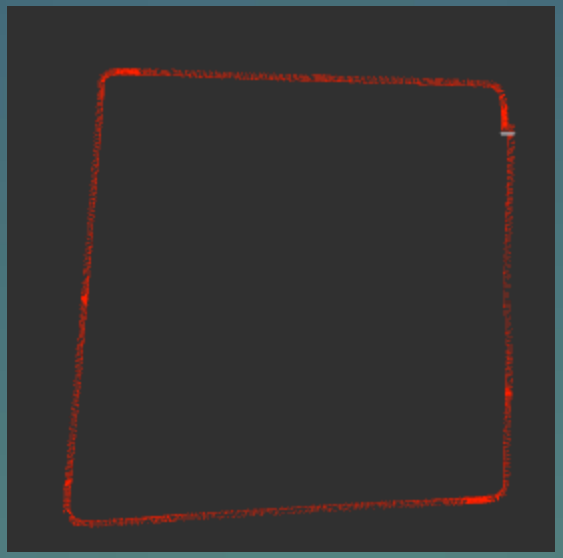
The number of points seems to be too little! Why not only filter out the points within the specified distance at the same time?

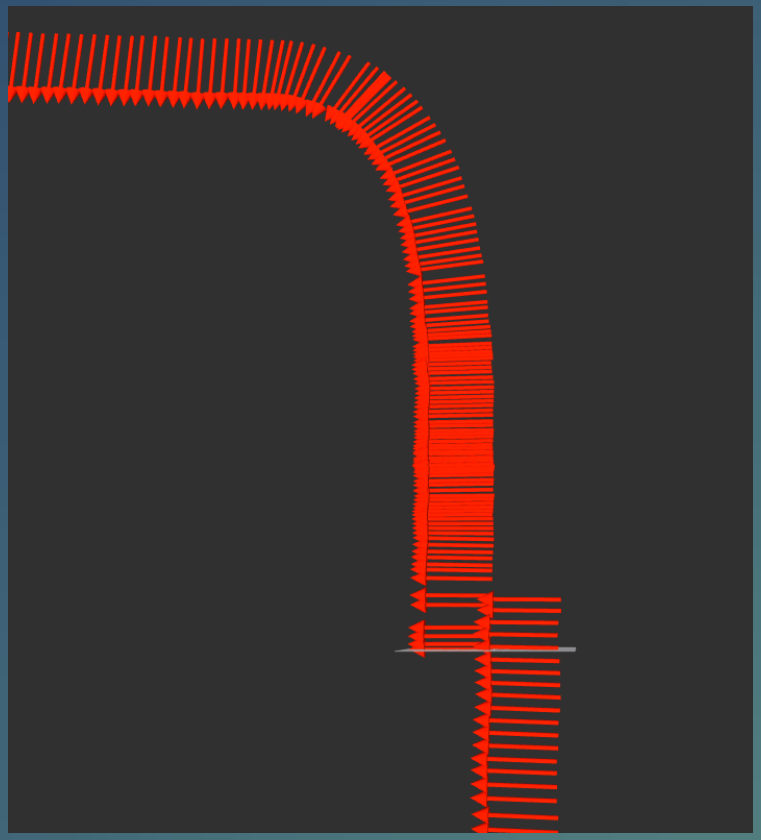
Filter out the points within the specified distance only

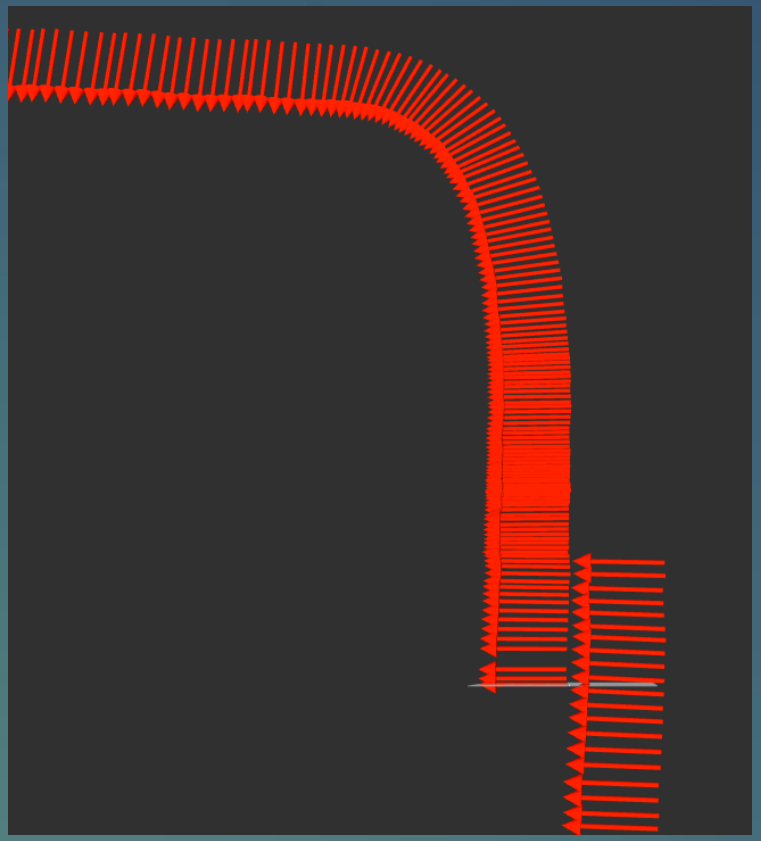


The filtered frame

route 1



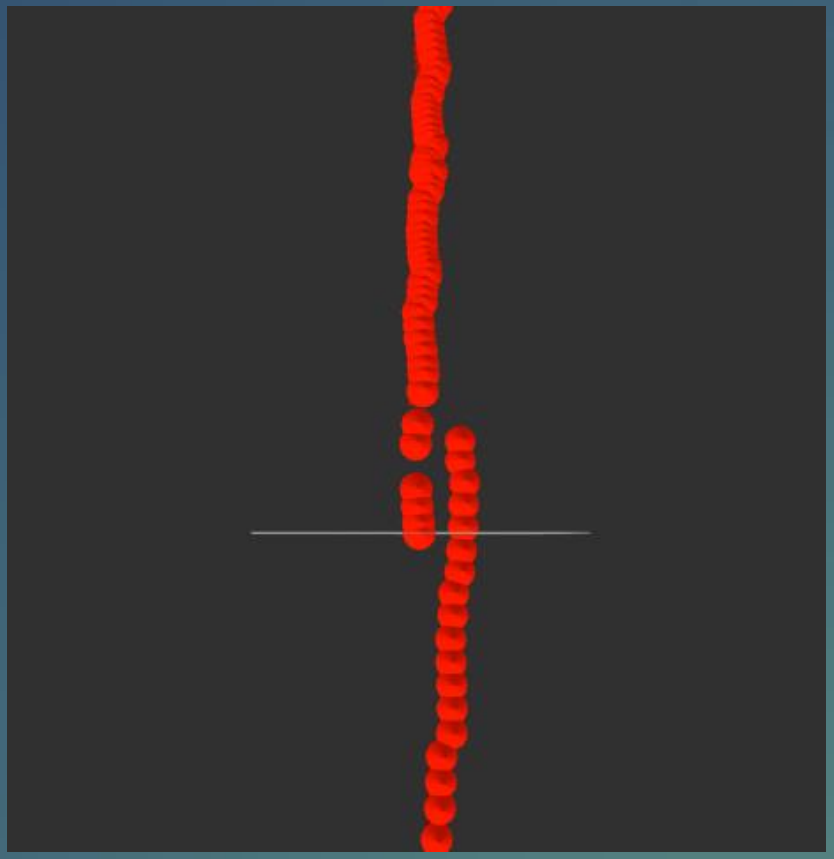


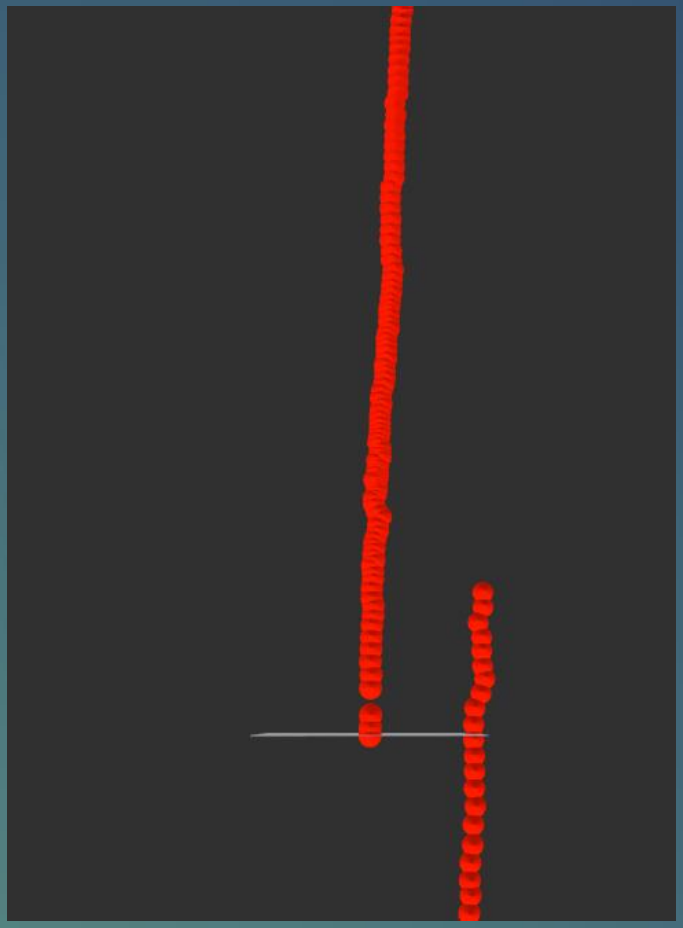


The translation is better than that with 11 scans.

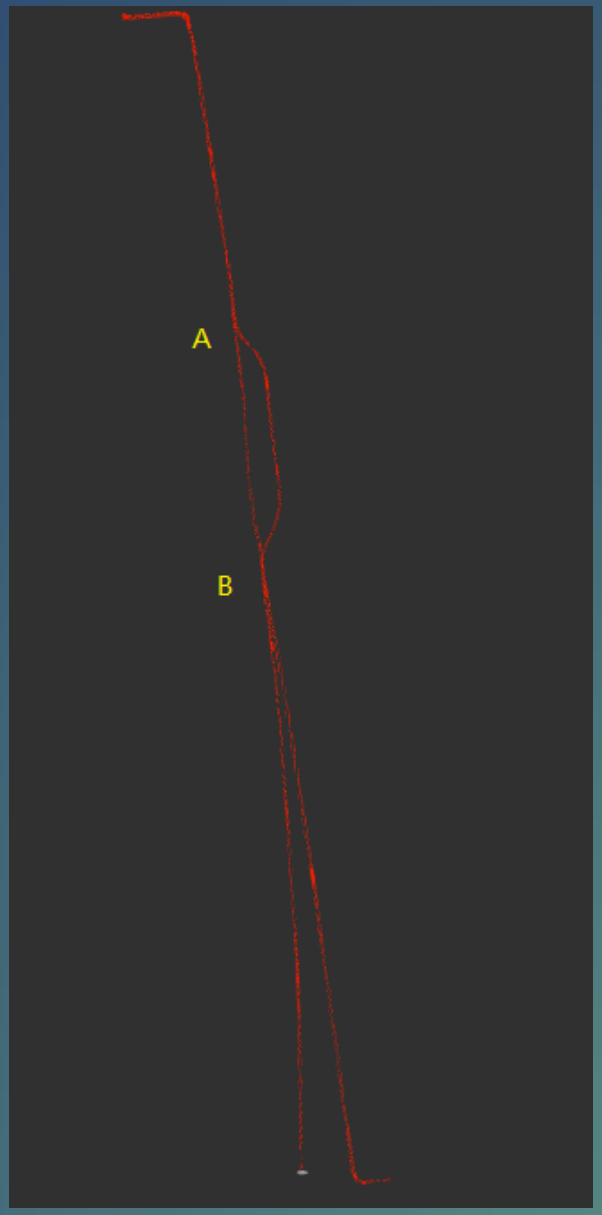
Analysis

The original frame





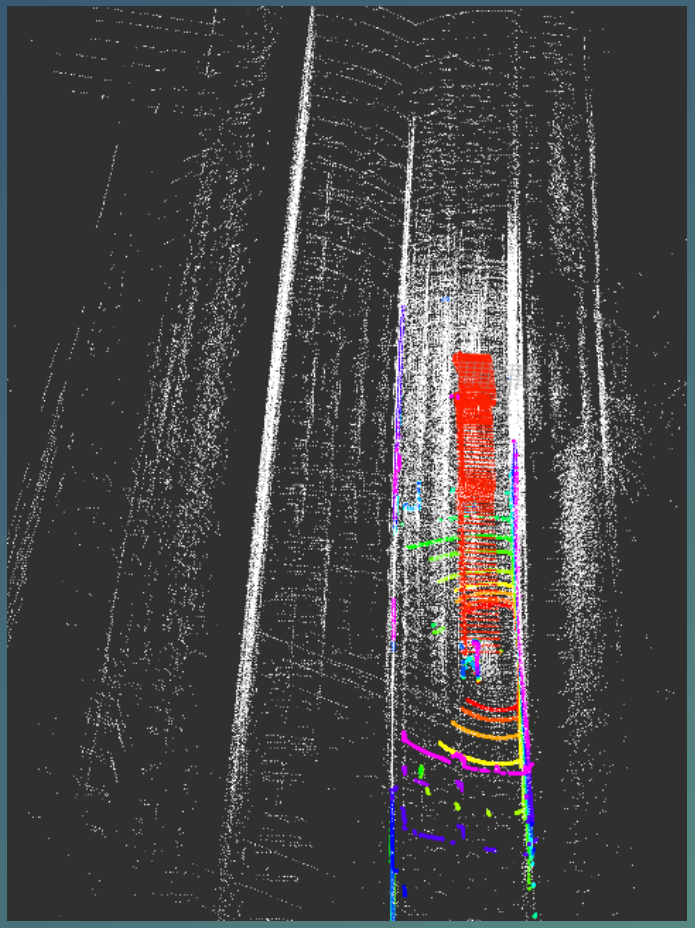
The rotation seems to be a little bit worse.

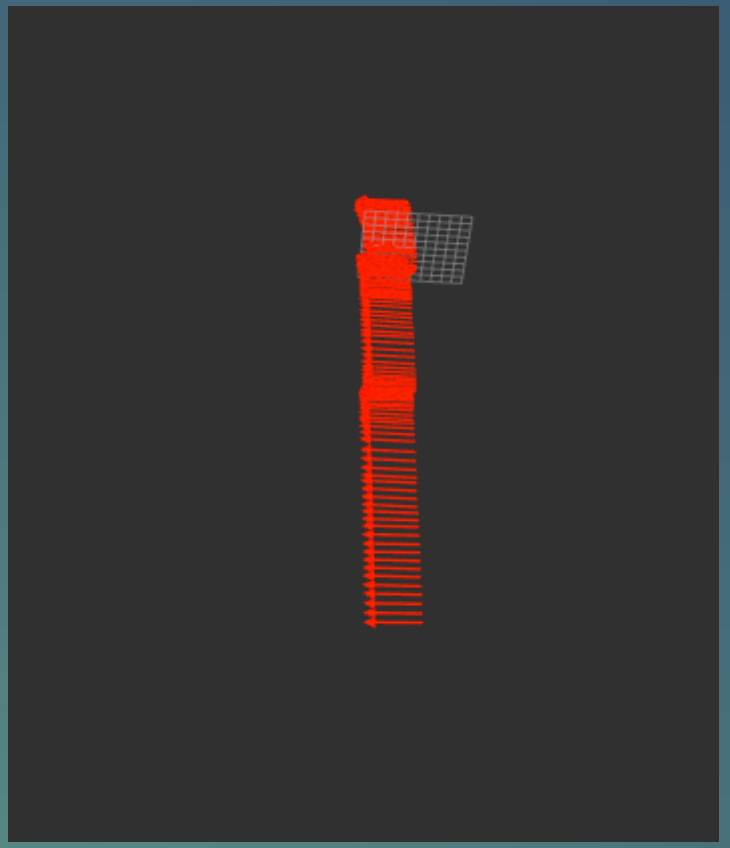


route 2

AB is beside the onstruction fence.

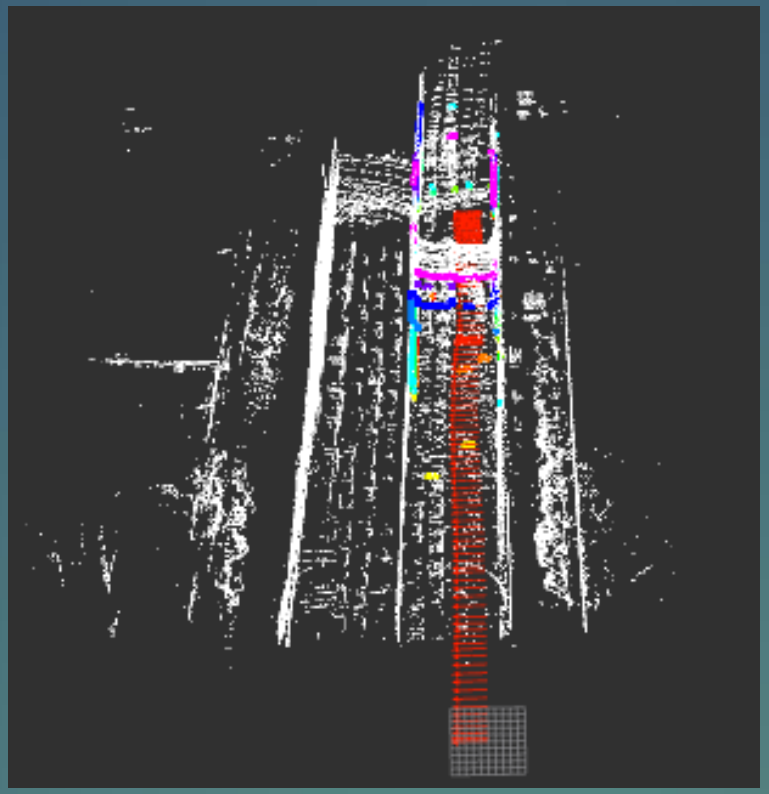
How about reversing problem?





Reversing in the tunnel !!!

Alleviated the problem of reversing



Conclusion

- Removing the dynamic points do well in translation ,but not in rotation. Although it can ease the problem of reversing, it seems hard to adapt to different environments well.
- It is not so hard to remove the points on the dynamic objects, but hard to remove the effect of these dynamic objects.

On degeneracy of Optimizaton-based State Estimation Problems

$$\mathbf{V}_p = [\boldsymbol{v}_1, ..., \boldsymbol{v}_m, 0, ..., 0]^T,$$
$$\mathbf{V}_u = [0, ..., 0, \boldsymbol{v}_{m+1}, ..., \boldsymbol{v}_n]^T,$$
$$\mathbf{V}_f = [\boldsymbol{v}_1, ..., \boldsymbol{v}_m, \boldsymbol{v}_{m+1}, ..., \boldsymbol{v}_n]^T.$$

$$\boldsymbol{x}_f = \boldsymbol{x}'_p + \boldsymbol{x}'_u,$$

where $\boldsymbol{x}'_p = \mathbf{V}_f^{-1}\mathbf{V}_p\boldsymbol{x}_p$ and $\boldsymbol{x}'_u = \mathbf{V}_f^{-1}\mathbf{V}_u\boldsymbol{x}_u$.

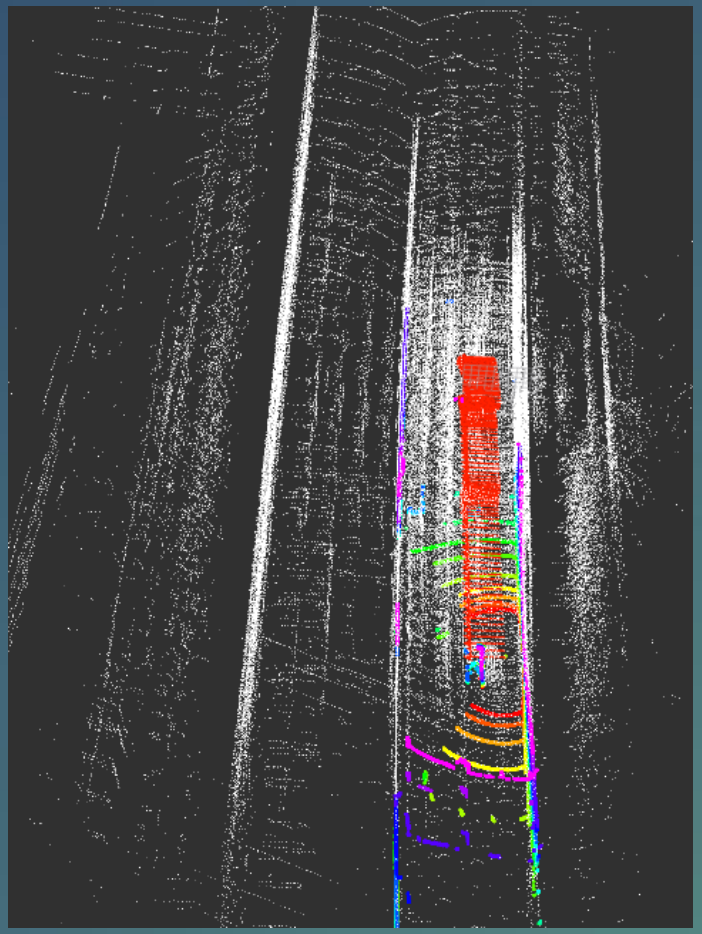
iMorpheus.ai

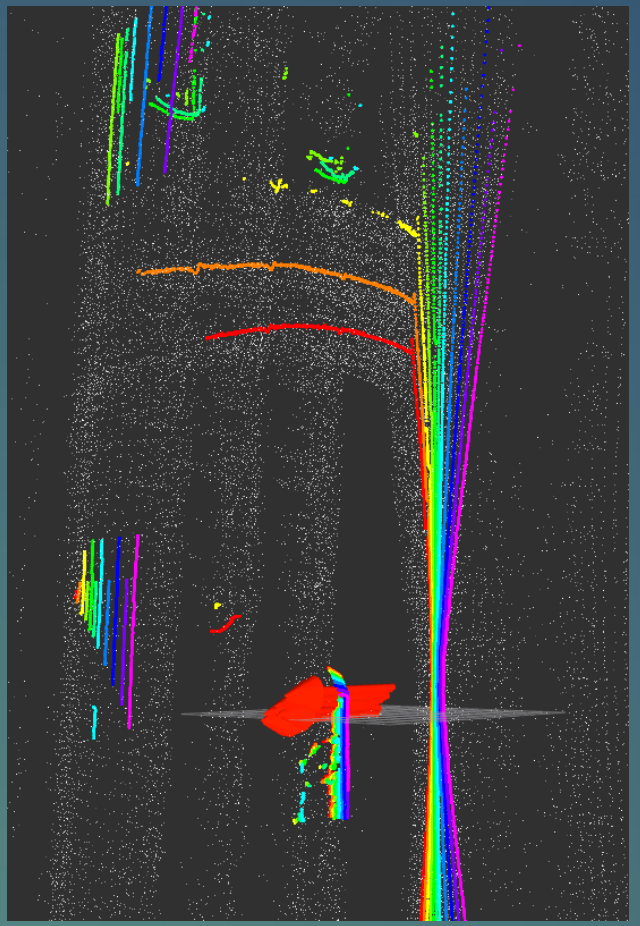
Algorithm

Algorithm 1: Nonlinear Solver with Solution Remapping

input : $\boldsymbol{f}$ (nonlinear function), $\boldsymbol{x}_p$ (predicted solution)
output : $\boldsymbol{x}_f$ (final solution)
begin
  $\boldsymbol{x}_f \leftarrow \boldsymbol{x}_p$; $O(*)$
  Linearize $\boldsymbol{f}$ at $\boldsymbol{x}_p$ to get $\mathbf{A}$, $\boldsymbol{b}$, and $\mathbf{A}^T\mathbf{A}$; $O(*)$
  Compute λ_i and $\boldsymbol{v}_i$ of $\mathbf{A}^T\mathbf{A}$, $i = 1, ..., n$; $O(n^3)$
  Determine a number m of λ_i smaller than a threshold, construct $\mathbf{V}_p$, $\mathbf{V}_u$, and $\mathbf{V}_f$ based on (15)-(17); $O(n^2)$
  while nonlinear iterations **do** $O(kn^2 + *)$
    Compute update $\Delta\boldsymbol{x}_u$; $O(*)$
    $\boldsymbol{x}_f \leftarrow \boldsymbol{x}_f + \mathbf{V}_f^{-1}\mathbf{V}_p\Delta\boldsymbol{x}_u$; $O(n^2)$
  end
  Return $\boldsymbol{x}_f$;
end

The solution is only updated in the well-conditioned directions, leaving x_p in the degenerate directions unchanged. From our experience it is not necessary to update the directions for each nonlinear iteration but the first iteration only, saving computation time.





It's very difficult to get a good adaptable value.

iMorpheus.ai

Other optimization ideas to be realized

- To improve the sampling rate to reduce point cloud distortion, and down sampling to reduce the amount of computation at the same time(e.g. frame skip)
- Remove the points on the dynamic objects, and use interpolation to restore the blocked points by dynamic to reduce the impact of dynamic points
- Dynamically modify some parameters according to the statistical distribution characteristics of lidar Points

Welcome to exchange ideas!

iMorpheus.ai

Journal Club

iMorpheus.ai Weekly Journal Club

Next Friday, 12/01/2018 12:00PM GMT+8

Semantics-Preserving Bag-of-Words Models and Applications

Website : **http://imorpheus.ai**
Email Address : **live@imorpheus.ai**

iMorpheus.ai