TR on LiDAR Odometry

Di Wang

Xi'an Jiaotong University

Oct 25^{th}, 2018

1. Motivation

For GraphSLAM system, the most typical settings are vertex representing individual pose and edge representing connection between vertices, whether be it from IMU, LiDAR odometry, loop closure (involving two vertices) or GPS (involving single vertex).

In nature, the back-end optimization in GraphSLAM can be considered as to find a way to distribute accumulative transformation error into vertices proportionally based on covariance matrix. Recall that the quality of LiDAR odometry plays an important role in accumulative error.

In this report, the popular point cloud registration algorithms are validated in one sequence of KITTI dataset. This is of course different from the point cloud generated from HDL-64E-S3 in Berkeley, but it still provides with us some sensible considerations on how to fine-tune the parameters in registration algorithms.

Dataset: KITTI SLAM/Odemetry dataset indexed by 07. This is captured from urban environment, the time duration is 43 seconds, and the trajectory length is 690 meters.

Programming Platform: Visual Studio 2015.

Registration algorithms:

ompNDT: ompNDT is employed in hdl_graph_slam, which is developed by Kenji Koide in Toyohashi University of Technology.

rosNDT: which is implemented by the author of NDT. It consists of point-to-distribution and distribution-to-distribution NDT, which are shorthanded as p2dNDT and d2dNDT.

PCL: point cloud library (PCL) contains ICP variants like point-to-point, point-to-plane, plane-to-plane ICP, also p2pNDT is included.

Error metric: the error is defined by the Euclidean distance between the end point of estimated trajectory and ground truth trajectory.

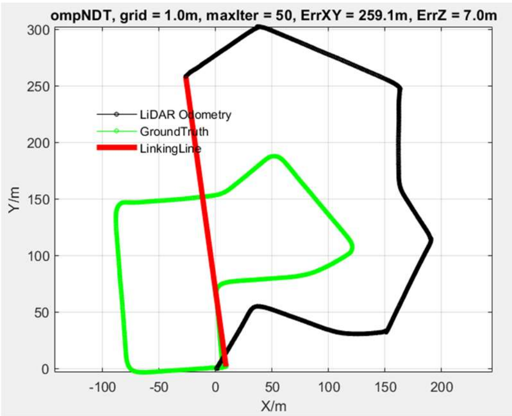
Notice that the NDT implementations in ompNDT, rosNDT and pclNDT are different with respect on optimization methods and other internal parameter selection. Also, d2dNDT only exists in rosNDT.

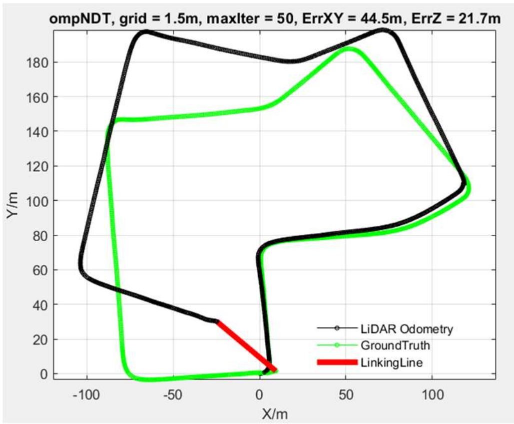
A special note of thanks also goes to Mr. Weisong Wen from the Hong Kong Polytechnic University, who introduces the fabulous NDT algorithm to me and we have a fruitful discussing on LiDAR-based SLAM for autonomous driving.

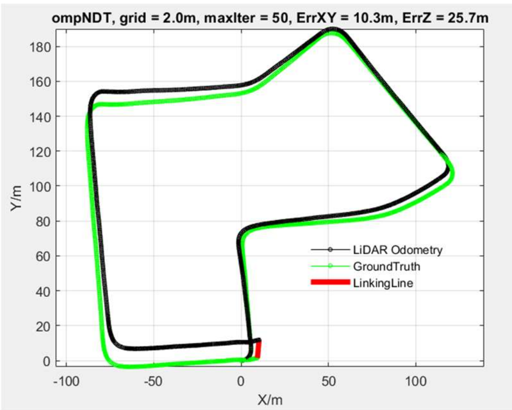
2. ompNDT

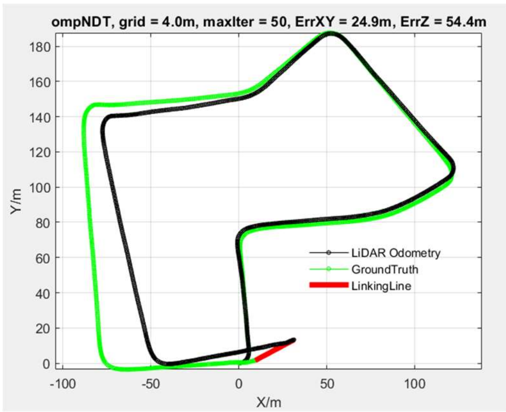
2.1 Different Grid Size

Grid Size/m	ErrXY/m	ErrZ/m
1.0	259.1	7.0
1.5	44.5	21.7
2.0	10.3	25.7
4.0	24.9	54.4



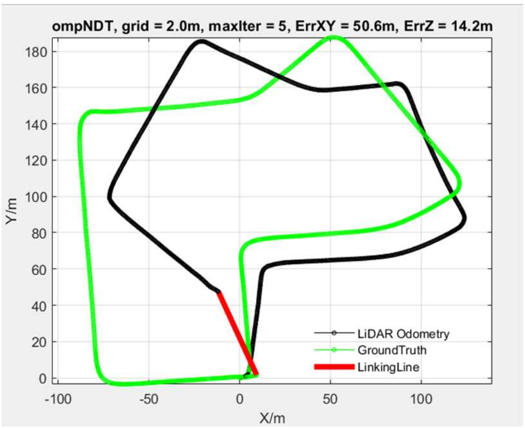


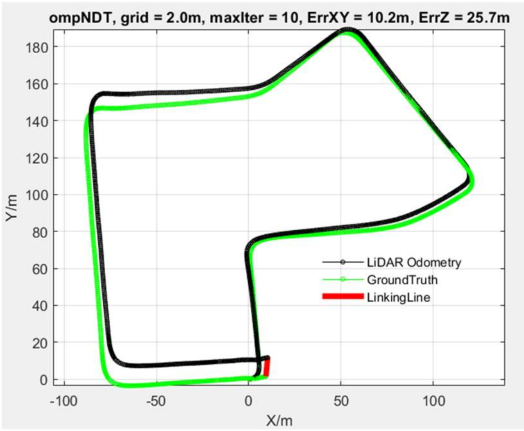


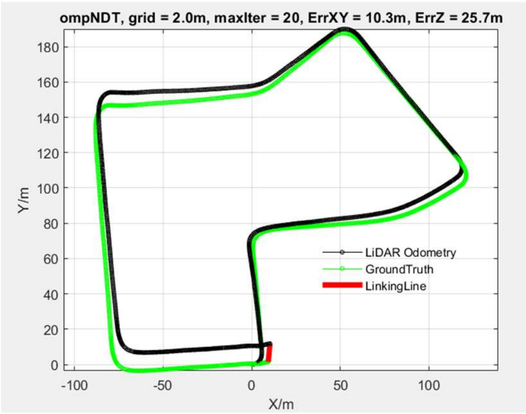


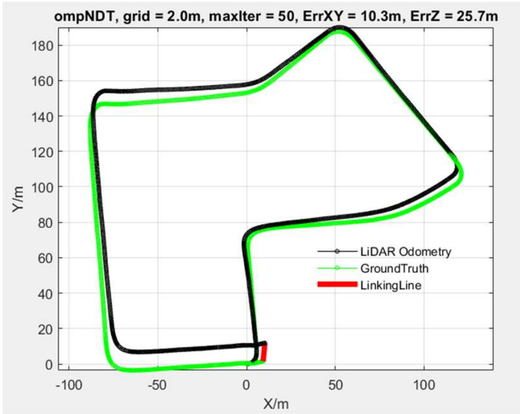
The grid size of NDT should be properly tuned. In this scenario, 2.0m is the best choice.

2.2 Different Maximal Iterations





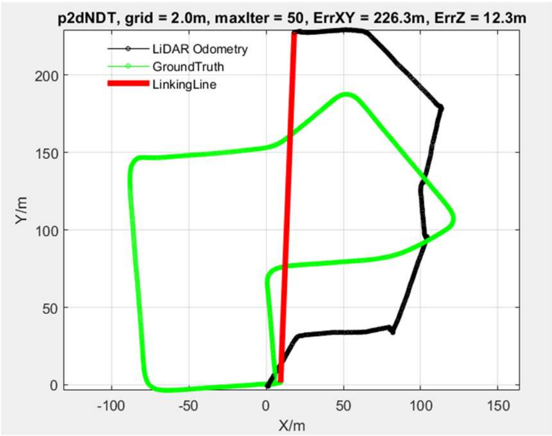


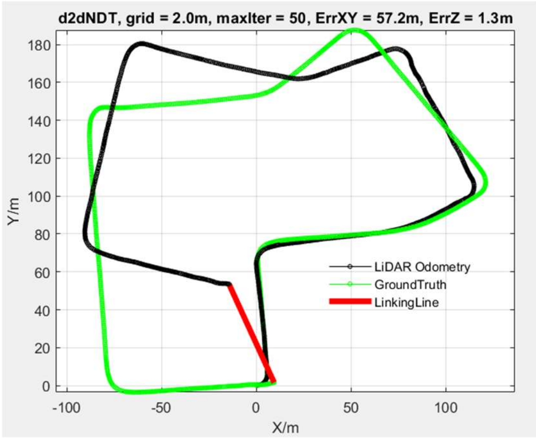


With increasing maximal iterations, the results are becoming better. However, the saturation occurs when maximal iteration equals to 10. However, the mapping process itself is off-line, and 20 iterations will guarantee that NDT converges normally.

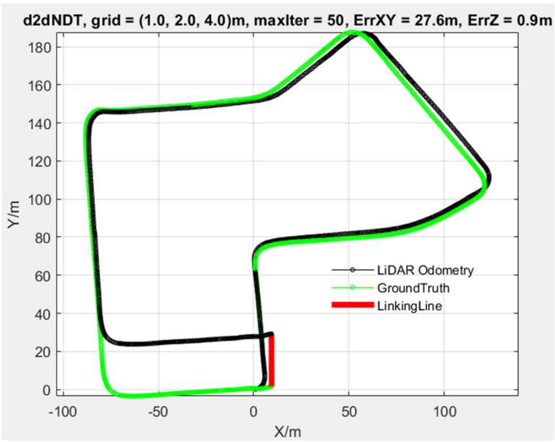
3. rosNDT

3.1 With Single Grid Size





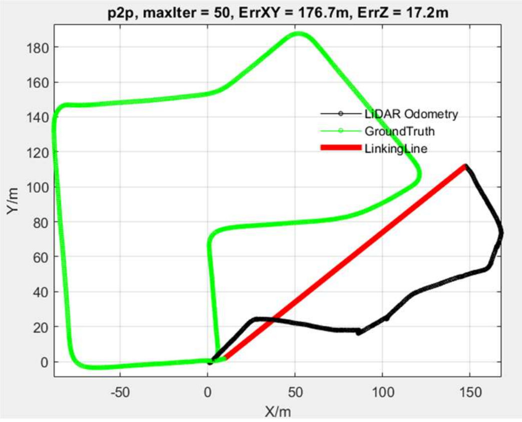
3.2 With Multiple Grid Size

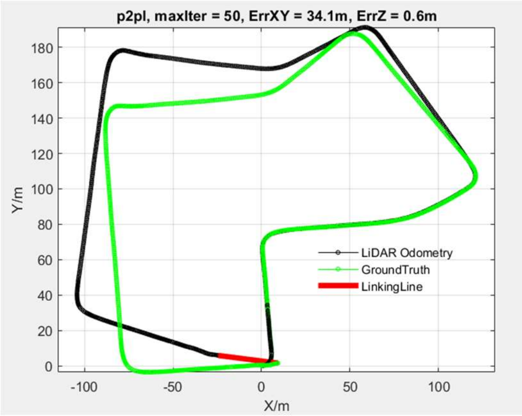


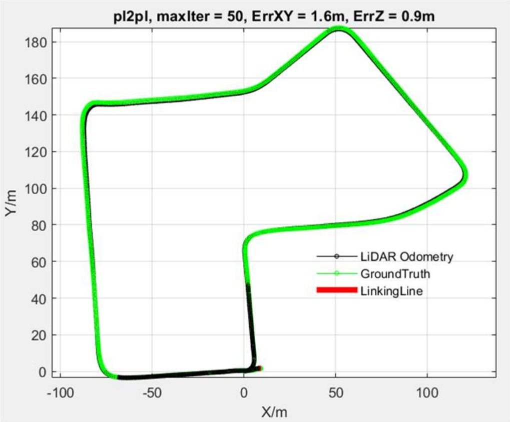
The multiple grid size initial strategy is better than single grid size.

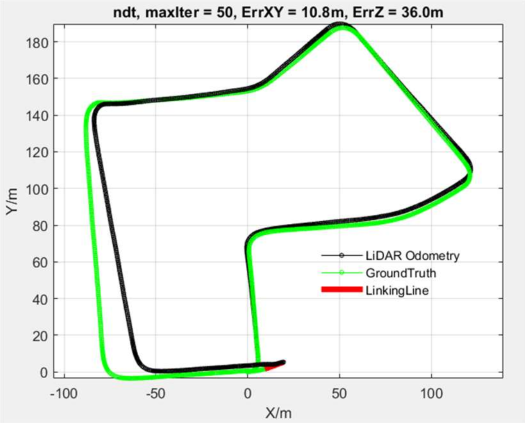
4. PCL Results: ICP and pclNDT

Method	ErrXY/m	ErrZ/m
Point-to-point ICP	176.7	17.2
Point-to-plane ICP	34.1	0.6
Generalized ICP	1.6	0.9
pclNDT(grid size = 2.0m)	10.8	36.0









5. Conclusion

Actually, there exists no single algorithms outperforms all the other algorithms in all the traffic scenarios. All the possible algorithms should be validated in order to find the best registration algorithms Still, some conclusions relating to NDT can be attained:

1. The maximal iteration should be set with 10~50.
2. The grid size influences the performance in a major manner. Choosing proper grid size for different environments.
3. The initial strategy, single grid size or multiple grid size also plays an important role. Multiple grid size initialization will overcome local minimum to some extent.

For ICP, it seems that GICP is good. In fact, many papers report that GICP is better than most of ICP variants on generalized datasets.