LiDAR Matching Strategies For HD Point Cloud Map Generation In Urban Area

Jou-An Chen [1, a*], Surachet Srinara [1, b], Yu-Ting Chiu [1, c], Kai-Wei Chiang [1, d]

[1] Department of Geomatics, National Cheng Kung University, Tainan, Taiwan (R.O.C.) –
[a] p66114080@gs.ncku.edu.tw, [b] surachetsrinara@gmail.com, [c] p66081106@gs.ncku.edu.tw, [d] kwchiang@geomatics.ncku.edu.tw

KEY WORDS: LiDAR Odometry; SLAM; GNSS/INS fusion; Direct Georeference; Autonomous Driving; HD Map.

ABSTRACT

Autonomous driving relies on high accuracy point vector map, which was generated by the point cloud map, and pre-provides the vehicle preliminary road environment information. Lidar Odometry and Mapping (LOAM) has always been a promising research topic in the field of robotics, environment sensing, and currently autonomous driving. However, in certain urban environments like basement parking lot, tunnels, highways, or other similar settings, the geometric features are not clearly discernible. As a result, algorithms resembling the LOAM framework may encounter difficulties in accurately mapping these areas. The paper utilized relative low-cost LiDAR expecting to propose a state-of-the-art point cloud mapping/update scheme. We compared the GNSS-challenge area with straight line and loop area separately, simultaneously considered the DG, ICP, NDT matching algorithm for the low-cost mapping/update strategy. With the realistic experiment conduction, our result evaluated by point to point corresponding mean error and standard error. For the straight line environment, ICP has the fastest convergence in empirical cumulative distribution under 0.4 meters. For the loop scenario, point-to-point ICP still has the fastest convergence in empirical cumulative distribution under 0.22 meters. Yet both of them still suffer from the fault matching.

1. INTRODUCTION

Accurate state estimation and building maps of the surrounding environment are crucial for intelligent mobile robots operating in environments where GNSS (Global Navigation Satellite System) localization signals are not available. In such scenarios, SLAM (Simultaneous Localization and Mapping) emerges as a valuable sensing technology for mobile robots. SLAM assist without GNSS (Global Navigation Satellite System) localization signalss in determining the robot's localization, pose information, and motion control by simultaneously constructing a map of the environment. This map becomes a valuable resource for mobile robots to plan optimal routes and navigate around obstacles by providing essential information about the surroundings. By leveraging SLAM, mobile robots can effectively operate and adapt to complex environments, enhancing their autonomy and overall performance.

The LOAM (Lidar Odometry and Mapping) approach utilizes a horizontal lidar sensor along with an inertial sensor to achieve efficient localization and mapping capabilities. Since its inception, several researchers have proposed enhancement strategies building upon the LOAM framework, such as Lego LOAM (Shan and Englot, 2018) and LIO-SAM (Shan et al., n.d.) methods. While the LOAM scheme has enjoyed considerable success, it still exists some limitations.

There are two primary limitations associated with the LOAM scheme. Firstly, the lack of a loop closure detection module results in lower accuracy in localization and mapping during actual testing. Loop closure detection is a crucial component for recognizing previously visited locations and correcting accumulated errors. Its absence in the LOAM scheme contributes to reduced accuracy.

Secondly, the LOAM algorithm relies on a uniform motion model assumption, which can lead to localization and mapping failures when the robot or carrier exhibits rapid and vigorous motion. The algorithm's robustness in handling such scenarios is insufficient, resulting in compromised performance.

To address the aforementioned challenges, we present a novel framework for simultaneous localization and mapping (SLAM) that combines a rotating lidar and an inertial measurement unit (IMU). Our proposed framework offers the following solutions:

1. Motion Distortion Compensation: We introduce a nonlinear motion distortion compensation method specifically designed for rotating lidar systems. By fusing low-frequency lidar data and high-frequency inertial data, we effectively compensate for motion-induced distortions, resulting in more accurate localization and mapping outcomes.
2. Improved Matching: We introduce a point cloud matching effect evaluation module, which assesses the quality of point cloud matching. Additionally, we incorporate a module based on sub-map to sub-map matching using a key-frame strategy. This module includes a double judgment candidate loop-frame strategy to enhance the reliability of loop closure detection.

By integrating these advancements into our SLAM framework, we aim to overcome the limitations of existing approaches, achieve higher localization and mapping accuracy, and enhance the robustness of the system under challenging motion conditions.

2. RELATED WORK

2.1 Broad Lidar Odometry and Mapping framework

Several existing methods have been proposed to improve upon the LOAM framework and overcome its limitations. For instance:

1. Lego-LOAM (Shan and Englot, 2018): Lego-LOAM addresses the limitations of LOAM by introducing lightweight components. It extracts ground feature points to participate in point cloud matching and performs L-M optimization using line and surface feature points in two steps. It also incorporates a loop closure detection module based on European distance to mitigate accumulated drift.

This contribution has been peer-reviewed.
https://doi.org/10.5194/isprs-archives-XLVIII-1-W2-2023-785-2023 | © Author(s) 2023. CC BY 4.0 License.

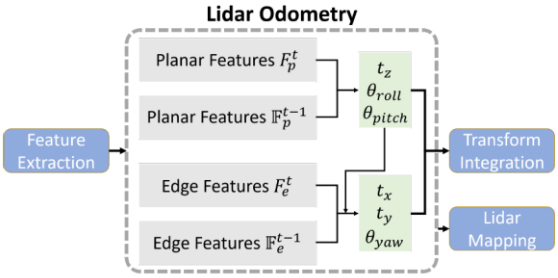

Figure 2-1 Two-step optimization for the lidar odometry module.

2. LIO-SAM (Shan et al., n.d.): LIO-SAM leverages 9-axis inertial data for system initialization and front-end laser odometry. It integrates GPS factors and loop closure factors in the back-end to improve the accuracy of localization and mapping.

3. ALOAM (Zhang and Singh, n.d.) : ALOAM focuses solely on lidar sensors and employs the Ceres-Solver and Eigen libraries to reconstruct and optimize the LOAM code. It aims to enhance the performance and accuracy of the original LOAM algorithm.

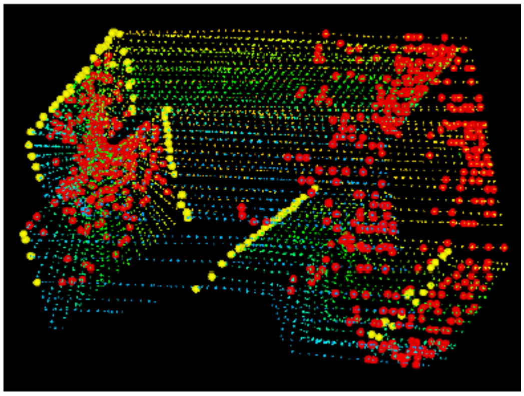

Figure 2-2 An example of extracted edge points (yellow) and planar points (red) from lidar cloud taken in a corridor.

4. F-LOAM (Wang et al., 2021): F-LOAM employs a two-step procedure to correct motion distortion in the original point cloud. It deviates from LOAM's parallel processing of laser odometry and laser mapping and instead utilizes a weighted feature point constraint during nonlinear optimization.

5. R-LIO (Chen et al., 2022): R-LIO is mainly composed of four sequential modules, namely nonlinear motion distortion compensation module, frame-to-frame point cloud matching module based on normal distribution transformation by self-adaptive grid, frame-to submap point cloud matching module based on line and surface feature, and loop closure detection module based on submap-to-submap point cloud matching.

6. IN2LLAMA (Le Gentil et al., 2021): IN2LLAMA is an offline framework that combines probabilistic methods to perform localization, mapping, and extrinsic calibration tasks. This framework utilizes a 3D lidar and a six-degree-of-freedom inertial measurement unit (IMU) to achieve accurate and reliable results.

7. CT-ICP (Dellenbach et al., 2021): Continuous-Time ICP extend this method to encompass a full SLAM system, which incorporates a unique loop detection procedure. The key innovation of their approach lies in the integration of both continuity and discontinuity in scan matching.

8. G-ICP (Ren et al., 2019): GICP-based 3D point cloud registration is based on the point-to-plane algorithm to optimize laser odometry constraints between consecutive frames and key frames without the need for additional sensors like an IMU. Additionally, G-ICP automatically remove noise from the point cloud data to enhance the consistency of the underground roadway map. Overall, G-ICP improves the accuracy and reliability of the map by combining precise registration, innovative roadway plane extraction, and noise removal techniques. However, these methods demand high computing resources for matching a full point cloud.

3. METHODOLOGY

3.1 System Overview

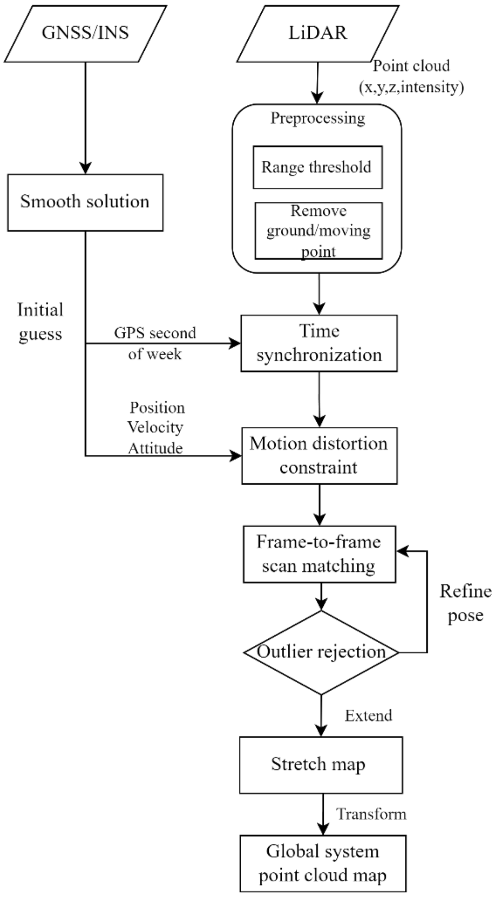

Figure 3-1 System overview of proposed method

3.2 LiDAR data preprocessing

LiDAR data after collection will be stored in the pcap or ROS BAG file. To parse them into pcd file and range the threshold or remove the moving point, we utilized the open-source parser called “VLP-16 parser” with the self revision cross-hour code to deal with the time system inconsistent or error issues.
To illustrate the geomatic relationship shown in **Figure 3-2**, we named Lidar as L-frame, and scanned point cloud as P-frame. r_l^p is the translation vector between LiDAR and point cloud. R_l^p is the rotation matrix between the LiDAR and point cloud.

Range the threshold is to filter out the extremly near and far point cloud. For example: min range is 3.5 meters and max range is 75 meters.

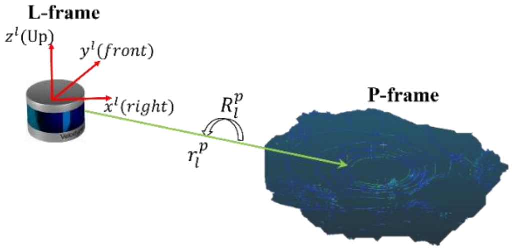

Figure 3-2 LiDAR preprocessing

3.3 Time synchronization

Firstly, in **Figure 3-3** we define the INS/GNSS as a Body frame so-called B-frame, LiDAR as L-frame, and mapping frame as M-frame. Secondly, make sure the LiDAR timestamp and INS/GNSS are recorded as the same time system, i.e., GPS second of week, or UNIX or UTC. Noted that cross-hour is the base principal to be solved, it usually happened when using the open-source parser with the experiment time was crossing-hour at that time. Thirdly, once the timestamp is consistent, interpolate the nan value.

As the INS/GNSS integrated solution can provide the translation vector at epoch time $r_m^{b_k}$, $R_m^{b_k}$ which is calculated by commercial software IE.

3.4 Motion distortion constraint

Except for the mounting parameters (i.e., lever arms r_b^l and boresight angles R_b^l) between the LiDAR and INS/GNSS. The other parameter following right-hand rules be calculated below: Let t_k be the current time stamp, and recall that t_{k+1} is the starting time of the sweep. The translation vector of the M-frame and B-frame which changed time by time named $r_m^{l_k} = [t_x, t_y, t_z]$, and LiDAR pose rotation matrix $R_m^{l_k}$ (3*3 matrix) from interpolated attitude $[\theta_x(heading), \theta_y\ (roll), \theta_z(pitch)]$

$$r_m^l = R_m^b r_b^l + r_m^b \qquad (3.3.1)$$

$$R_m^l = R_m^b R_b^l \qquad (3.3.2)$$

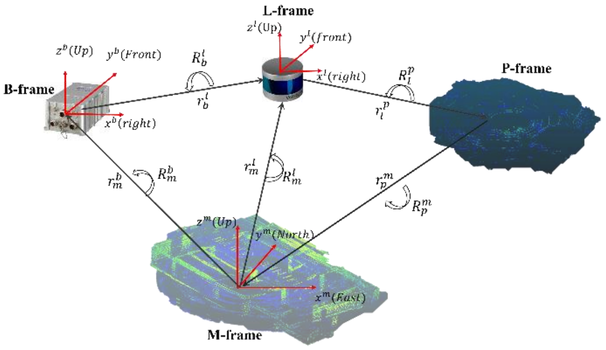

Figure 3-3 Direct georeference system

3.5 Feature point extraction

The given paragraph describes the process of extracting feature points from a scan plane obtained by a rotating laser scanner. The scanner rotates at an angular speed of 180 degrees per second and generates scans at a frequency of 10Hz. As a result, the resolution in the direction perpendicular to the scan planes is 180 degrees / 10Hz = 18 degrees.

The feature points are selected based on information from individual scans and their co-planar geometric relationship. Two types of feature points are chosen: sharp edges and planar surface patches.

Let "i" represent a point in the scan plane "P_K," and "S" be the set of consecutive points of "i" returned by the laser scanner in the same scan. The scanner generates point returns in either clockwise (CW) or counterclockwise (CCW) order, and "S" contains half of its points on each side of "i" with 0.25-degree intervals between two points.

$$c = \frac{1}{|\mathcal{S}| \cdot ||\boldsymbol{X}_{(k,i)}^L||} || \sum_{j \in \mathcal{S}, j \neq i} (\boldsymbol{X}_{(k,i)}^L - \boldsymbol{X}_{(k,j)}^L)||. \qquad (3.5.1)$$

In the scanning process, the points obtained are organized based on their "c" values. Subsequently, specific feature points are chosen: those with the highest "c" values, known as edge points, and those with the lowest "c" values, known as planar points. To achieve an even distribution of these feature points across the scanned environment, the scan is divided into four equal subregions. Each subregion can yield a maximum of 2 edge points and 4 planar points. However, for a point to be selected as an edge or planar point, it must meet two conditions: its "c" value should be either higher or lower than a certain threshold, and the total number of selected points cannot exceed the predetermined maximum. This selection process ensures a balanced distribution of feature points for further analysis and utilization.

3.6 Frame to frame scan matching

1. ICP (“a method for registration of 3D shape,” n.d.) : The Iterative Closest Point (ICP) algorithm refines the relative pose of two overlapping scans by minimizing the sum of squared distances between corresponding points in the two scans. Corresponding point pairs are identified based on their point-to-point distance.

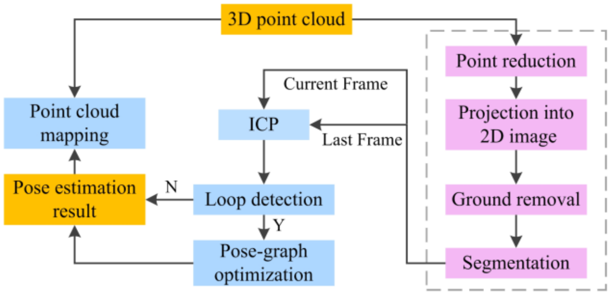

Figure 3-4 Overview of the proposed LiDAR localization and mapping system architecture (Li et al., 2020)

2. 2D-NDT (Biber, 2003): NDT involves dividing the 2D plane into cells, similar to an occupancy grid. Each cell is assigned a normal distribution, representing the local probability of measuring a point within that cell. The resulting transformation yields a piecewise continuous and differentiable probability density. This density can be leveraged to match another scan using Newton's algorithm, eliminating the need for establishing explicit correspondences between points. By utilizing this probabilistic representation, we achieve a smooth and differentiable matching process, improving the efficiency and accuracy of the algorithm. The function from Equation 2.3.1 is the most straightforward 2D NDT transformation function. Let $\vec{p}$

https://doi.org/10.5194/isprs-archives-XLVIII-1-W2-2023-785-2023 |

$= [t_x, t_y, \theta]^T$, where t_x and t_y are the translation vectors and θ is the rotation angle.(Magnusson, 2009)

$$T_{2D}(\vec{p}, \vec{x}) = \begin{bmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{bmatrix} \vec{x} + \begin{bmatrix} t_x \\ t_y \end{bmatrix} \qquad (3.6.1)$$

3. 3D-NDT: The Normal Distribution Transform (NDT) algorithm divides into 2D and 3D registration methods. The main difference between 2D and 3D registration with NDT lies in the spatial transformation function T($\vec{p}, \vec{x}$) its partial derivatives. Rotation can be shown in two dimensions with a solitary value for the rotation angle with respect to the origin.

Despite the possible issues related to the representation of rotation using 3D Euler angles, they will be utilized below. The benefits, such as not having to constrain the numerical optimization process and having slightly simpler derivatives, are deemed more important than the danger of gimbal lock, which would only happen at angles so large that the local registration procedure would almost certainly fail. The pose's six-dimensional parameter vector includes six parameters to optimize: $\overrightarrow{p_6} = [t_x, t_y, t_z, \theta_x, \theta_y, \theta_z]^T$ three for translation and three for rotation. The 3D NDT transformation equation (3.6.2) can be obtained by employing the Euler sequence z-y-x, where $c\theta_i = cos\theta_i$, and $s\theta_i = sin\theta_i$:

$$T_{3D}(\overrightarrow{p_6}, \vec{x}) = R_x R_y R_z \cdot \vec{x} + \vec{t}$$

$$= \begin{bmatrix} c\theta_y c\theta_z & -c\theta_y s\theta_z & s\theta_y \\ c\theta_x s\theta_z + s\theta_x s\theta_y c\theta_z & c\theta_x c\theta_z - s\theta_x s\theta_y s\theta_z & -s\theta_x c\theta_y \\ s\theta_x s\theta_z - c\theta_x s\theta_y c\theta_z & c\theta_x s\theta_y s\theta_z + s\theta_x c\theta_z & c\theta_x c\theta_y \end{bmatrix} \vec{x} + \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} \qquad (3.6.2)$$

4. RESULT

4.1 Experiment setup

The experimental setup employed in this study is depicted in. The reference system consisted of the IMU (iNAV-RQH) and GNSS (PwrPak). The vehicle utilized an VLP-16 LiDAR, iNAV-RQH and PwrPak as an IMU, and PwrPak as a GNSS antenna and receiver. The navigation solution was post-processing utilizing the tightly-coupled INS/GNSS scheme with forward and backward smoothing through the commercial INS/GNSS software, Inertial Explorer (IE).

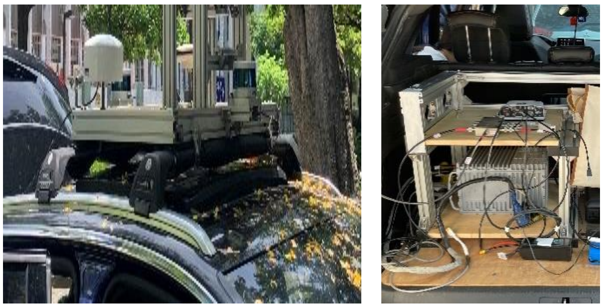

Figure 4-1 Experiment setup

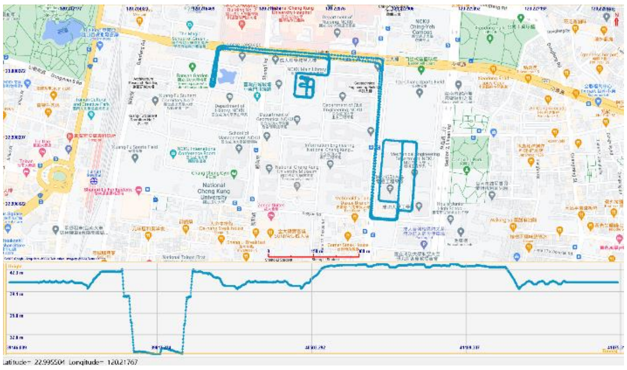

Figure 4-2 Experiment trajectory

Table 4-1. Experiment setup Test system and reference system

	Test system	***Reference system***
IMU	NovAtel PwrPak7-E2	NovAtel PwrPak7-E2
GNSS	NovAtel OEM5	iNAV-RQH
LiDAR	Velodyne VLP-16	Velodyne VLP-16

Table 4-2 The specification of Velodyne LiDAR (VLP-16)

Velodyne VLP-16	
Max Measurement Range	100 m
Range Accuracy	± 3cm (typical)
FOV (Vertical)	+15° to -15° (30°)
FOV (Horizontal)	360°
Angular Resolution (Vertical)	2°
Angular Resolution (Horizontal/ Azimuth)	0.1° - 0.4°

4.2 Mapping result

4.2.1 Direct Georeference Point Cloud Map

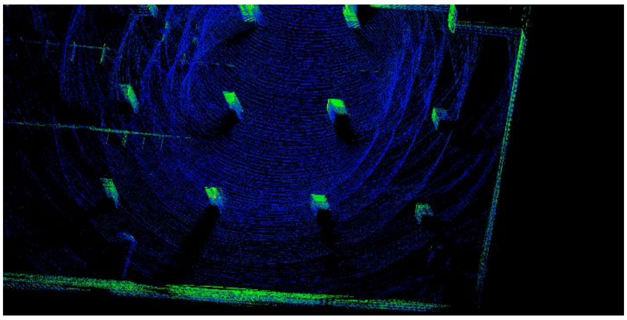

Figure 4-3 DG frame 6200-6269

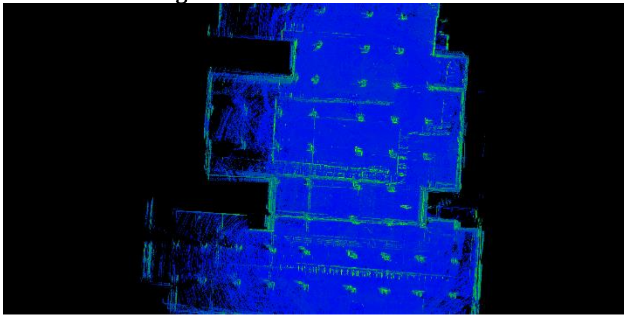

Figure 4-4 DG frame 6200-7000

4.2.2 ICP Point Cloud Map

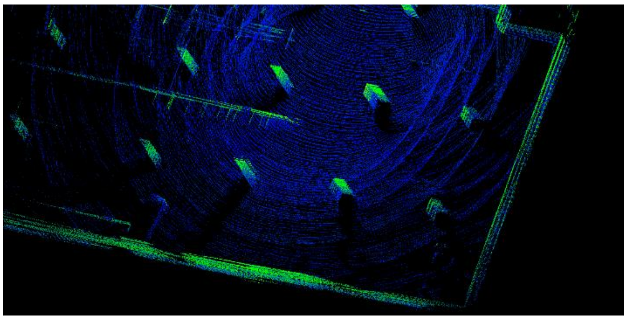

Figure 4-5 ICP frame 6200-6269

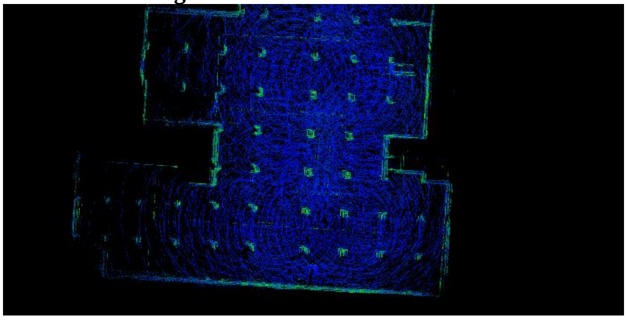

Figure 4-6 ICP frame 6200-7000

Table 4-3 The specification of ICP point to point

ICP point to point configuration	
Downsample Grid Step	0.05
Downsample Method	Grid Average
regInlierRatio	0.5

4.2.3 NDT Point Cloud Map

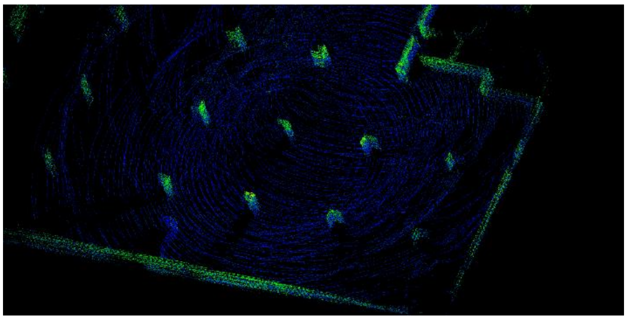

Figure 4-7 NDT frame 6200-6269

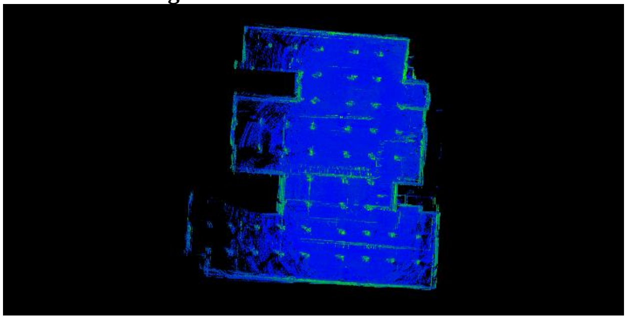

Figure 4-8 NDT frame 6200-7000

Table 4-4 The specification of NDT point to point

NDT configuration	
Outlier Ratio	0.9
Downsample Method	Grid Average
regInlierRatio	0.5

4.3 Accuracy evaluation

4.3.1 Trajectory accuracy

Table 4-5 E2 VS. RQH position error analysis

	E	***N***	***U***	***Horizontal***	***3D***
Initial	0.880	0.357	0.112	0.950	0.956

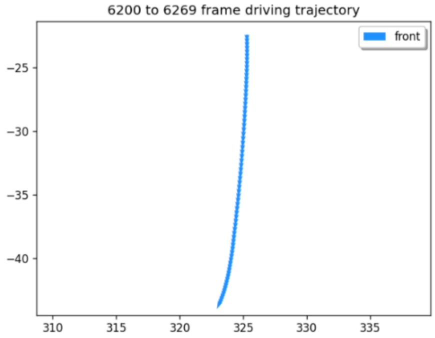

Figure 4-9 trajectory frame 6200-6269

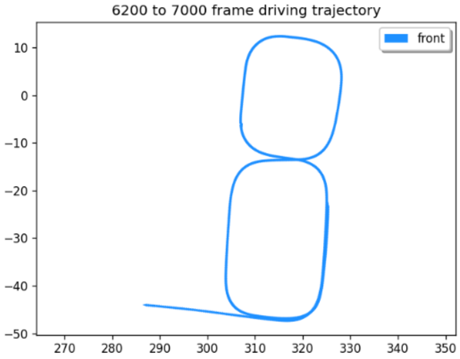

Figure 4-10 trajectory frame 6200-7000

4.3.2 Point cloud matching accuracy

Table 4-6 Frame 6200- 6269 Error analysis

(meter)	***Mean error***		***Std error***	
	Horizontal	***Vertical***	***Horizontal***	***Vertical***
DG	0.3406	-0.1472	0.4854	0.5435
DG-ICP	0.3376	-0.2462	0.5595	0.4017
DG-NDT	0.3654	-0.1888	0.5958	0.4233

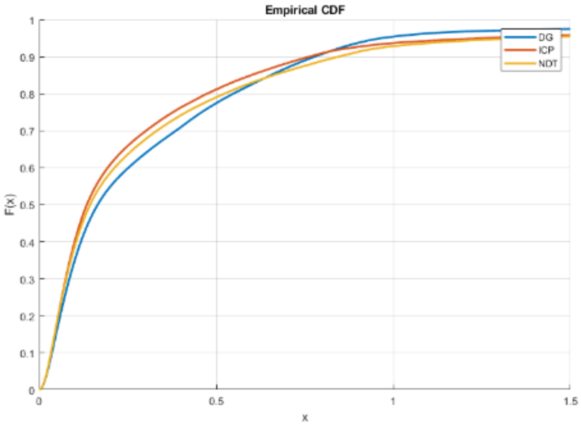

Figure 4-11 Horizontal Empirical cumulative distribution frame 6200-6269

Empirical cumulative distribution, which computes the function values (f) and the confidence bounds using different algorithms, depending on the censorship information. Frame period 6200 to 6269, the tested algorithm claimed that if we expect the 80% of horizontal accuracy under 0.4 meters, ICP algorithms in our experiment scenario will perform better than NDT and DG. As the result of the evaluation, when it comes to ICP and NDT algorithm, we still need to improve fault matching detection, and avoid the large outlier over the long time drift.

Table 4-7 Frame 6200-7000 Error analysis

(meter)	*Mean error*		*Std error*	
	Horizontal	*Vertical*	*Horizontal*	*Vertical*
DG	0.2139	-0.016	0.4269	0.2308
DG-ICP	0.2061	-0.017	0.4644	0.2125
DG-NDT	0.2344	-0.020	0.5237	0.2255

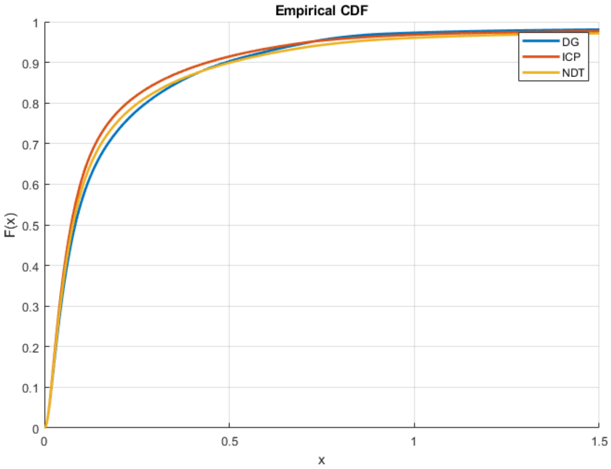

Figure 4-12 Horizontal Empirical cumulative distribution frame 6200-7000

When it comes to loop detection (Frame number 6200-7000), we could get the other discovery. From **Figure 4-12** the tested algorithm claimed that ICP algorithm achieved 80% of horizontal accuracy under 0.22 meters, outperform amomg DG and NDT. Even though we enlarge the period of time, DG always maintains the advantage under this scenario.

5. CONCLUSION

Due to the generation in large-scale LiDAR dense HD point cloud map required high-cost LiDAR, relied on sufficient laser beams to acquire abundant information in the complicated urban area. This paper considered the HD map required the latest mapping information, therefore we compared the most popular LiDAR scan matching algorithm with low-cost LiDAR and find out that ICP has the fastest convergence in empirical cumulative distribution under 0.5 meters. For the loop area scenario, ICP has the fastest convergence as well in empirical cumulative distribution under 0.22 meters. Yet both of them still suffer from the fault matching.

The paper proposed the scenario in basement parking lot where the GNSS is limited yet the artificial object information is abundant for LiDAR scanning. Technically speaking, our method still has amounts of discussion and improvement. For example, to eliminate most of the height error, fitting the plane and remove most of the ground points would be a suitable option. As for HD point cloud mapping results, it may reflect on the trajectory accuracy. The mapping issue would be trajectory drift over the distance, therefore, for the future work, author would still work on LOAM algorithm using loop closure, scan context, or pose graph to meet the same position and reduce the accumulation error.

REFERENCES

Biber, P., 2003. The Normal Distributions Transform: A New Approach to Laser Scan Matching, in: IEEE International Conference on Intelligent Robots and Systems. pp. 2743–2748. https://doi.org/10.1109/iros.2003.1249285

Chen, K., Zhan, K., Pang, F., Yang, X., Zhang, D., 2022. R-LIO: Rotating Lidar Inertial Odometry and Mapping. Sustainability (Switzerland) 14. https://doi.org/10.3390/su141710833

Dellenbach, P., Deschaud, J.-E., Jacquet, B., Goulette, F., 2021. CT-ICP: Real-time Elastic LiDAR Odometry with Loop Closure.

Le Gentil, C., Vidal-Calleja, T., Huang, S., 2021. IN2LAAMA: Inertial Lidar Localization Autocalibration and Mapping. IEEE Transactions on Robotics 37, 275–290. https://doi.org/10.1109/TRO.2020.3018641

Li, X., Du, S., Li, G., Li, H., 2020. Integrate point-cloud segmentation with 3d lidar scan-matching for mobile robot localization and mapping. Sensors (Switzerland) 20. https://doi.org/10.3390/s20010237

Magnusson, M., 2009. The Three-Dimensional Normal-Distributions Transform-an Efficient Representation for Registration, Surface Analysis, and Loop Detection.

Ren, Z., Wang, L., Bi, L., 2019. Robust GICP-based 3D LiDAR SLAM for underground mining environment. Sensors (Switzerland) 19. https://doi.org/10.3390/s19132915

Shan, T., Englot, B., 2018. LeGO-LOAM: Lightweight and Ground-Optimized Lidar Odometry and Mapping on Variable Terrain, in: IEEE International Conference on Intelligent Robots and Systems. Institute of Electrical and Electronics Engineers Inc., pp. 4758–4765. https://doi.org/10.1109/IROS.2018.8594299

Shan, T., Englot, B., Meyers, D., Wang, W., Ratti, C., Rus, D., n.d. LIO-SAM: Tightly-coupled Lidar Inertial Odometry via Smoothing and Mapping.

Wang, H., Wang, C., Chen, C.-L., Xie, L., 2021. F-LOAM: Fast LiDAR Odometry And Mapping. https://doi.org/10.1109/IROS51168.2021.9636655

Zhang, J., Singh, S., n.d. LOAM: Lidar Odometry and Mapping in Real-time.