Modified ICP based Framework for 3D Localization and Mapping in GNSS-denied Environments using LiDAR

Mansi Koshti, Surbhi Barnwal, Salil Goel

Department of Civil Engineering, IIT Kanpur, Kanpur, India - (mansipk22, surbhib20, sgoel)@iitk.ac.in

Keywords: LiDAR, Odometry, Registration, Iterative Closest Point, Simultaneous Localization and Mapping

Abstract

Lidar odometry and mapping has emerged as fundamental technology in various fields, including accurate positioning and mapping, autonomous vehicles, robotics, and environmental monitoring. The ability to create accurate maps and calculate trajectories is essential for localization and navigation in complex environments, particularly in scenarios where GNSS signal are unavailable. In this context, we propose a novel method for simultaneous localization and mapping (SLAM) specifically designed for GNSS-denied environments and without the need for high-accuracy ranging or inertial measurement units (IMUs). Our approach aims to address the challenges posed by GNSS-denied environments and the cost constraints associated with using multiple sensors. By using only lidar sensors, specifically the Velodyne VLP 16 sensor, we offer a cost-effective solution. Our method utilizes a modified version of the Iterative Closest Point (ICP) algorithm to handle consecutive lidar scans seamlessly. In addition to our method, we conducted thorough accuracy assessments on the created map. The reason behind extensive accuracy evaluation is to ensure the quality of registration. By comparing alignment between the features of different frames within the map and determining the error between the registered features and the original, we gain a more comprehensive understanding of registration quality.

1. Introduction

With the increasing demand for large-scale data acquisition with minimum efforts, light detection and ranging (LiDAR) emerged as remote sensing technology (Wang, Z. et al., 2021). Lidar sensors capture highly detailed, three-dimensional maps of the Earth’s surface. The resulting dataset of millions of data points creates a three-dimensional representation known as a point cloud. Point clouds mainly capture the geometric structure of surroundings, excluding environmental texture features. As the LiDAR sensor collects the raw data in the form of fragmented and unorganized point clouds, there is a necessary step called point cloud registration also known as alignment or fusion, which is the process of aligning multiple point clouds obtained from different viewpoints or sensors into a unified coordinate system. The point cloud registration acts as a vital bridge between raw scan data and actionable maps. Without it, mapping efforts may suffer from inaccuracies and inconsistencies, limiting the effectiveness of autonomous systems. There are plenty of point cloud registration methods (Gu, X. et al. 2020). One of the widely used methods is ICP-based point cloud registration. ICP (Iterative Closest Point) based point cloud registration is favoured due to its simplicity, efficiency, and robustness in aligning noisy and incomplete data. Its iterative nature enables it to effectively handle imperfections in real-world scenarios while offering good accuracy, making it versatile for various applications such as autonomous vehicles, object reconstruction, and localization (Li P. et.al., 2020).

In the context of simultaneous localization and mapping (SLAM), LOAM (Lidar Odometry and Mapping), ICP plays a crucial role (Khan et al., 2021; Mendes E. et al., 2016). Lidar odometry and mapping involve estimating the motion of the sensor using consecutive point cloud scans while simultaneously creating a 3D map of the environment (Ji X. et al., 2019). Within this framework, we propose a novel approach for localization and mapping in GNSS-denied environments using solely LiDAR sensors. While existing systems often utilize GNSS and IMU data to enhance accuracy, our focus is on environments lacking GNSS signals, such as dense urban areas, indoor spaces, remote regions, or dense forests. Our proposed system operates without an IMU and aims for cost-effectiveness and simplicity by relying solely on low-cost LiDAR sensors. Additionally, our approach allows for accuracy validation without the need for high-precision measurement units.

In addition, we conduct thorough accuracy assessments of the created map. We used techniques to extract geometric features from the generated map and compared these features with the original data features. The reason behind extensive accuracy evaluation is to ensure the quality of registration. Root mean square error gives a measure of overall accuracy. It does not provide insights into whether the correct correspondence is being determined between data points. By comparing features within the map and determining the error between the registered and original features, we gain a more comprehensive understanding of registration quality.

2. Related Work

Lidar-based SLAM is revolutionizing mapping and navigation research, offering efficient solutions. By integrating Lidar scans, it tracks sensor movement and creates detailed maps simultaneously, transforming navigation and analysis. The LOAM algorithm (Ji Zhang et al., 2014), introduced by Ji Zhang and Sanjiv Singh in 2014, achieves minimal error accumulation and low computational demand without relying on highly accurate measurements. It splits problem into two algorithms: one estimates high-frequency sensor velocity for odometry, while the other focuses on precise point cloud registration, enabling real-time mapping. Subsequent advancements have done to develop more efficient algorithms using LiDAR for localization and mapping integrating GNSS and IMU. But research is actively focused on mapping in GNSS-Denied environments. The authors in G. He (2021) proposed a system which can be used in partially GNSS denied environment. Author develops mobile mapping systems for outdoor environments that do not rely on expensive inertial systems. Map construction is achieved through GNSS-LiDAR integration where RTK positioning is available, and solely through LiDAR in GNSS-denied environments. Jian Tang and Xiaoji Niu have introduced a Lidar scan matching-aided inertial navigation system for GNSS-denied environments (Tang J. et al., 2015). They integrate INS with LiDAR SLAM into a unified navigation framework

using an extended Kalman filter to achieve stable long-term navigation. For precise positioning, they rely on INS to compensate for the absence of GNSS. De Paula Veronese, L. has proposed Lidar-only systems for GNSS-denied environments, introduced a single-sensor system for mapping (de Paula Veronese, L. et al., 2019). Their approach utilizes Lidar odometry based on particle filter localization and global positioning derived from virtual GNSS to detect loop closures. Additionally, they employ graphSLAM to fuse outcomes from Lidar odometry and virtual GNSS.

Given the limited research on localization and mapping relying solely on LiDAR systems, our primary focus is on proposing an effective and a simplistic solution for GNSS-denied environments. We aim to develop a cost-efficient system that utilizes low-cost LiDAR technology without the requirement for high-cost inertial measurement units or other complex technologies.

3. Data Sets and Methodology

The VLP 16 sensor is chosen for data collection. The reason behind this selection serves two key purposes: to thoroughly evaluate the accuracy of our proposed system and to maintain cost-effectiveness in its implementation. Also, the environment where data collection can be done should be rich in diverse mapping features includes prominent structures such as buildings, vehicles, and vegetation such as trees, bushes. Also, this environment extends to outdoor areas, assuring complete coverage and representation of urban landscape for precise and detailed mapping purposes. The data collection is conducted within the premises of faculty building parking area at Indian institute of Technology, Kanpur, shown in figure 1. This location has prominent features for accurate data registration and precise alignment, additionally, this location has plenty of geometric features, which are pivotal for registration process and also the evaluating the accuracy of registration. These features provide good reference point to compare and verify alignment of data.

Figure 1. Selected Location for Data Collection.

After data collection, the raw data is stored as PCAP (Packet Capture) file format. The data goes through conversion from PCAP to PCD (Point cloud Data) file format using Veloview software (Paraview, 2024). This conversion process extracts the data into point cloud data file containing coordinates (x, y, z) for analysis and visualization.

3.1 Registration of Two Point Clouds

After the data collection, there is necessary step of noise removal. Noise removal in point cloud data includes data filtering which removes unwanted or noisy points to improve the data quality and accuracy. Usually, it is achieved by techniques like statistical outlier removal and voxel grid down sampling.

After preprocessing, we pass the data to a registration module that uses a point to plane ICP approach for registration of two consecutive frames. We employ a modified GICP (Segal et al., 2009) algorithm with LM (Levenberg Marquadt) optimization to estimate the translation and rotation parameters between the two frames.

The basic ICP introduced by chen and medioni (Besl, P. J., 1992) in which convergence is done by minimizing Euclidean distance between 2 point clouds. Point to plane ICP is a modification to basic ICP which takes advantage of surface normal information for registration process.

Let the source and target point clouds are denoted as $A = \{a_1,..., a_N\}$ and $B = \{b_1,...,b_N\}$, respectively. Hence, the relationship between the two correspondences a_i and b_i is given by equation (1).

$$a_i = T.b_i \tag{1}$$

where, $T = [R: t]$ (transformation Matrix).
The notation R denotes a 3×3 rotation matrix and t denotes a 3×1 translation vector.

To compute transformation matrix, basic ICP minimizes $\sum ||T.b_i - a_i||^2$, however, Point to plane ICP takes the different approach by minimizing the distance from each point in target to tangent plane of its corresponding (source) point cloud. Instead of computing point to point distance, point to plane ICP computes distance to the tangent plane, which provides a more accurate representation of surface geometry.

Our method uses a modified generalized ICP (Segal et al., 2009) for registration. It incorporates a probabilistic model for minimization. Correspondences are determined using kd-tree. It efficiently finds the closest points (correspondences) between two sets of points. For each point in the source point cloud, the kd-tree is queried to quickly locate its nearest neighbor in the target point cloud, significantly speeding up the correspondence search process. This probabilistic approach can incorporate surface information and assess the reliability of correspondences, making it more robust against erroneous matches. Consequently, for N correspondences, the transformation T is determined in the equation (2).

$$\mathrm{T} = \operatorname{argmax} \sum_{1}^{N} \log\left(p(d_i)\right) \tag{2}$$

where $d_i = a_i - T.b_i$ denotes the transformation error.

As these correspondences a_i and bi are inaccurate and are assumed to be sampled using gaussian distribution. i.e. $a_i \sim N(a_i, C_i^A)$, $b_i \sim N(b_i, C_i^{B)}$ and indexed according to their correspondence (a_i correspondence with b_i). In this case, C_i^A and C_i^B are covariance matrices related to the measured points. Since both a_i and b_i are following gaussian distribution, d_i

will also follow a gaussian distribution with mean $\widehat{d_i}$ and covariance matrix ($C_i^A + TC_i^BT^T$), shown in equation (3).

$$d_i^{(T*)} \sim N(a_i - (T^*)b_i, C_i^A + (T^*)C_i^B\,(T^*)^T) \quad (3)$$

The estimator di is an unbiased estimator of di and hence, $di = 0$.

$$d_i^{(T*)} = N(0, C_i^A + (T^*)C_i^B(T^*)^T \quad (4)$$

By substituting, equation (4) in equation (2), we get.

$$T = argmin \sum_{i=1}^{N} (d_i^T (C_i^A + TC_i^BT^T)^{-1} d_i \quad (5)$$

The values of the covariance matrices C_i^A and C_i^B are depends on the properties of the correspondences.

After the determination of correspondence pair, to get better accurate results, our method uses L-M optimization to achieve a wider convergence basin. The Levenberg-Marquardt (L-M) optimization technique is used in point cloud registration to address challenges like low overlap, noise, outliers, and complex environments. By solving non-linear least squares problems, the L-M algorithm enhances point cloud alignment accuracy. It also mitigates the issue of local minima in ICP registration, aiming to find global minima. This robust and efficient approach significantly improves the results of point cloud registration, making it more reliable in various challenging scenarios.

From registration of two point clouds, we transformed this modified Generalized ICP (G-ICP) approach for the simultaneous registration of frames. This method enhances the robustness and accuracy of aligning multiple point clouds.

3.2 Registration of Simultaneous Frames

Simultaneous registration of frames is an important step for accurate alignment of features and mapping of the environment. This process involves continuous registration of point cloud data frames for creation of up-to-date map of surrounding features. Here, as simultaneous registration of frames means to convert all frames one by one into coordinate system of 1st frame. Registration depends on percentage of overlap between the frames. In order to enhance the efficiency of our proposed method, we conducted experiments using various conditions to optimize our approach and identify potential enhancements. Our goal was to simultaneously register all frames with respect to the coordinate system of the first frame. Initially, we attempted to individually register each frame with the first frame to determine the optimal overlap required for registration. However, this approach proved unsuccessful as the overlap decreased over time.

Subsequently, we explored an alternative method where we used the solution from the previous registration as the initial parameters for the registration of the next frame. Unfortunately, this approach led to the overall solution becoming trapped in a local optimum, resulting in incorrect alignment between the frames. Later, we performed the approach in which provided in below pseudo code.

Input: Number of consecutive frames N, Consecutive Frames, Initial transformation matrix [r,t]

Output: Final transformation matrices [r,t], Registered Frames

iter ← 0
tr_1 = [0,0,0,0,0,0]
for *i* in [1, *N*] **do**
  Frame *i* is target point cloud
  Frame *i+1* is source point cloud
  Preprocesses both point clouds with voxel grid
  Transformation matrix $tr_{i,i+1}$ = point cloud registration(target, source, *tr*[0,0,0,0,0,0])
  Final transformation matrix, $tr_{i+1} = tr_i \times tr_{i,i+1}$
  Final source point cloud = transform original source point cloud by applying transformation matrix tr_{i+1}
  Save tr_{i+1} and transformed source point cloud
end
return all transformation matrices and registered point clouds

Using this approach, results are optimized throughout the trajectory, involving fewer points cloud for computation, and significantly taking less time for computation. The final map is generated by merging all frames together. The trajectory is determined by translation vectors extracted from the resulting transformation matrices.

3.3 Accuracy Evaluation

Extensive accuracy evaluation is performed by extracting planes, lines, cylinder, and corner planes from Registered data and comparing them with original data. The reason behind extensive accuracy evaluation is to ensure the quality of registration. Root mean square error gives a measure of overall accuracy. It does not provide insights into whether the correct correspondence is being determined between data points. By comparing features within the map and determining the error between the registered features and the original, we gain a more comprehensive understanding of registration quality.

1. Standard deviation between fitted planes
2. Angle between Normal of planes (e.g., Plane of Buildings)
3. Angle between axis of the cylinder (e.g., Electric Pole)
4. Angle between fitted line in 2 corner planes (e.g., Corner of Building)
5. Deviation between planes

4. Results and Discussion

4.1 Data Collection

Data was collected using the VLP 16 LiDAR sensor employing a stop and go approach without loop closure. In the STOP and GO approach, the LiDAR sensor remains stationary at one position for 3 to 4 seconds, collects the data, and proceeds to the following position. Data collection was carried out for about 22 meters distance with the sensors positioned at intervals of 1.5 to 2 meters. At each location, the sensor remains stationary for 3 to 4 seconds. With a rotation speed of 600 RPM, the sensor collected approximately 30 to 40 frames at each location. Out of which, one frame is taken for the registration process. Figure 2 shows the set up for the data collection.

Figure 2. Data Collection using VLP 16 Sensor.

4.2 Data Preprocessing and Final Map Creation

As data is collected, it is essential to remove noise from the data, to avoid wrong correspondence matching between two frames, and remove unnecessary points from the frames. Noise removal is achieved using Voxel Grid Down sampling techniques. The parameters for Voxel Grid Down-sampling are set as 0.1, number of neighbours taken as 6. Data collected at 12 consecutive locations, one frame is taken for processing from each location. All frames were registered with respect to the first frame. The final map is shown in figure 3.

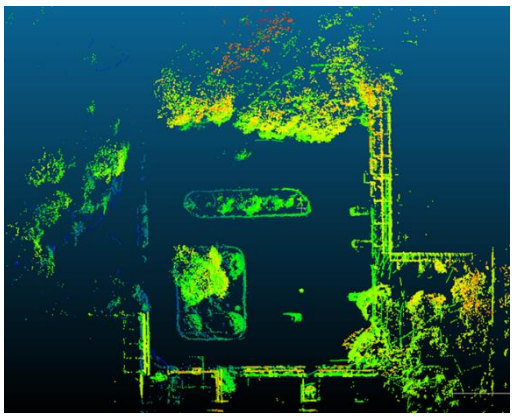

Figure 3. Top view of generated Map.

Figure 3 shows the plan view of the created map. This map is a combination of 12 consecutive frames. We can see the features present in the map such as cars, bushes, trees, buildings, electric poles, etc. All features from the different frames are perfectly aligned with each other. This accurate alignment between the features makes the accurate 3D map. The figure 4 shows the oblique view of the generated map.

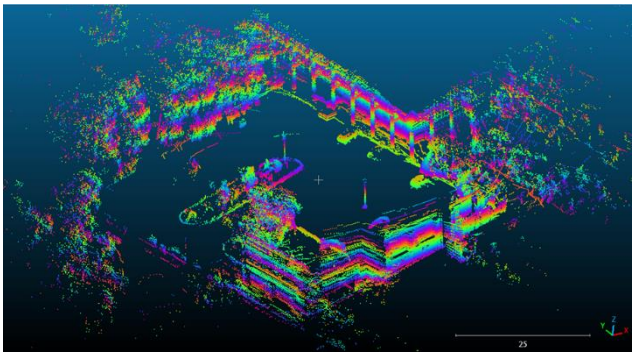

Figure 4. Oblique view of generated map.

Additionally, the trajectory is determined to track the movement of the sensor during data collection. The coordinates of the sensor’s position at 1st location are set as [0,0,0], and subsequent coordinates are computed with respect to 1st location of the sensor. Figure 5 shows the plotted trajectory in the generated map.

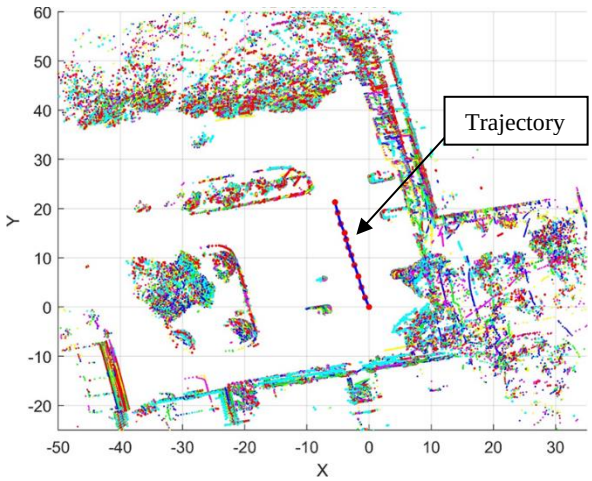

Figure 5. Generated map with the trajectory.

4.3 Accuracy Evaluation of Created Map

To determine the accuracy of a map, it is essential to check the relative difference between features shown in the map. Ideally, if features from various frames are perfectly align with each other, it indicates that there is no error in the map. However, since registration errors can occur, it is necessary to compute the relative errors between the features shown in the map, by which accuracy is determined.

To compute the accuracy of the map, we extracted geometric features such as planes, lines, cylinders, corner planes, etc within the map and compared them with each other. Using the novel method of accuracy determination, ensure precise determination of accuracy.

4.3.1 Standard Deviation between Fitted Planes

By fitting the plane in the extracted data, standard deviation can be determined. It talks about the deviation of points around the plane. Lower the standard deviation, higher the accuracy in the map.

The figure 6 shows the extracted planes, Plane A, Plane B, Plane C from the map for standard deviation computation.

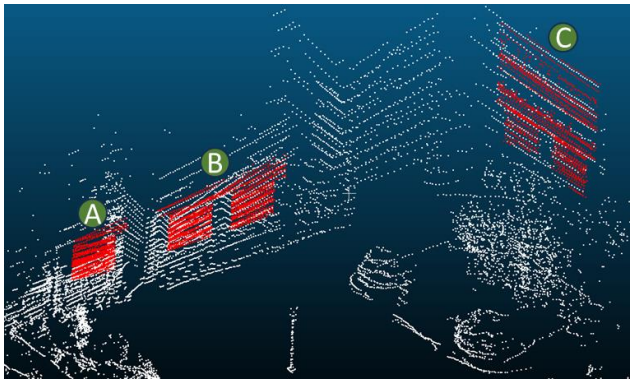

Figure 6. Extracted planes from the data.

Imaginary planes are fitted using least square method (Mathematics , 2024) in these extracted data points. Figure 7 shows the fitted planes in data points.

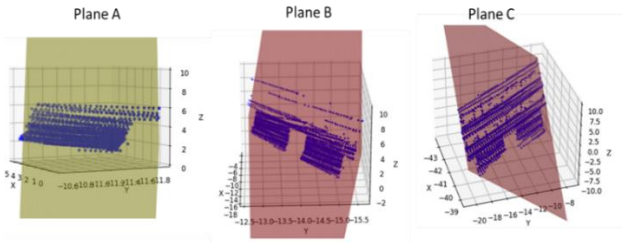

Figure 7. Fitted planes in three extracted data sets.

Subsequently, Standard deviation is computed for each data sets with respect to the fitted planes. The below table 1 shows the standard deviation value.

Planes	A	B	C
Max Std dev. (in cm) from data points to fitted plane in all individual frames (frame 1 to frame 12)	1.87	2.16	6.18
Std. dev. (in cm) From total data points to fitted plane of all frames (Total data)	1.96	2.22	8.23

Table 1. Computed standard deviation in extracted planes.

In table 1, planes A, B, C exhibit different standard deviation values compared to each other. This discrepancy arises due to their respective distance from the sensor’s position along the trajectory. Plane A is at 4 to 5 meters from the sensor’s starting point. Plane B is at 7 to 8 meters, and Plane C is at around 13 meters. As the distance from plane to sensor’s position is increasing, there is a corresponding increase in the standard deviation among the point cloud data. The sensor's range accuracy, typically around ± 3 cm, ensures that Plane A and Plane B maintain an acceptable level of error. However, the slightly increased distance between Plane C and the sensor results in a marginally higher error which demonstrates the error in the registration or the quality of registration.

4.3.2 Angle between the Normal of Planes

Another approach to achieve the registration accuracy involves comparing the angles between computed normal for the fitted planes. The plane is fitted in extracted data sets and normal of fitted planes is computed. Subsequently, the angles between original and registered frames data sets are compared. Computing the normal of each plane and comparing the angles between them gives the error in rotation among the features within the map. If the angle between normal of planes is zero, indicates the planes are perfectly aligned. Therefore, by comparing the angle between normal from registered data with original data determines the error in alignment between planes, thereby the quality of the registration.

The normals are computed for 1st frame (original) and 2nd frame (registered) of plane 2 data sets are shown figure 8.

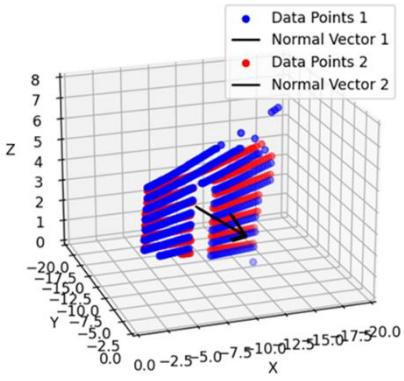

Figure 8. Extracted datasets with computed normals of original and registered frame.

As shown above figure, normals have been computed for 1st frame (original) and 2nd frame (registered). The angle between these normal measures 0.59 milli radians. Similarly, angle compared in all other registered frame with original frame for Plane A, Plane B, and Plane C in figure 9.

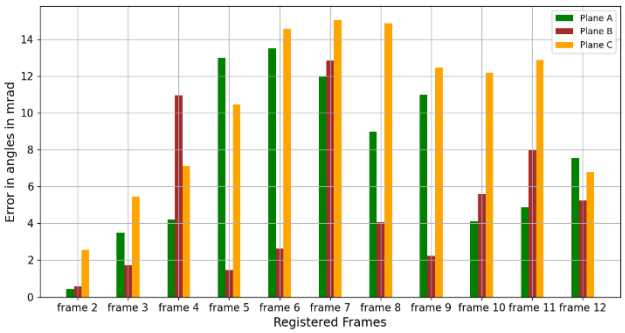

Figure 9. Bar plot between computed angles and registered frames for Plane A, B and C.

The RMS value of Error in angle between the normals for each plane is computed and shown in table 2. These metrics indicate the registration quality by assessing the alignment accuracy of planes. With all planes exhibiting errors within the range of 8 to 12 milliradians, or equivalently 0.45 to 0.68 degrees, our results suggest that the registration error falls within the acceptable range considering the sensor resolution accuracy.

Planes	RMS values (in mrad)
Plane A	8.66
Plane B	9.34
Plane C	11.17

Table 2. Root mean square value for angles for each extracted plane.

4.3.3 Angle between Axis of Fitted Lines

Features like electric pole, kerb are cylindrical in shape, but due to spare density of the sensor’s point cloud data. These features appear as line features. Therefore, line can be fitted by extracting these features from the map and angles can be compared. Extracted data of electric poles, Pole A and Pole B are shown in figure 10.

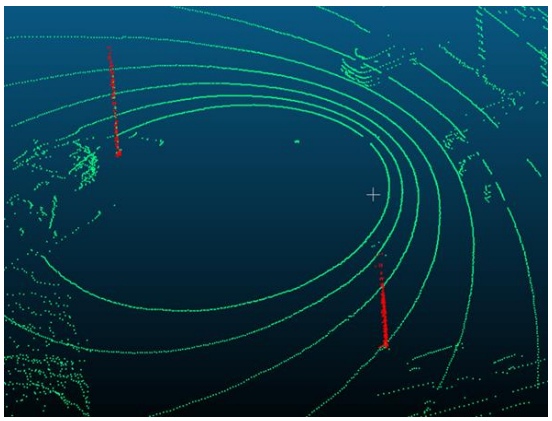

Figure 10. Extracted electric poles from the map.

Line is fitted in each extracted data from original as well as registered frames and angle is compared. The below figure 11 shows the fitted line in data of 1st frame (original frame) and data of 2nd frame (registered frame). Angle is compared among original and all registered frames and plotted in bar graph, shown in figure 12.

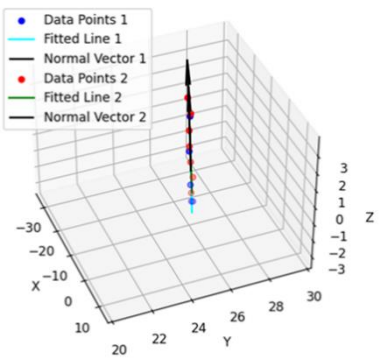

Figure 11. Fitted lines in extracted data.

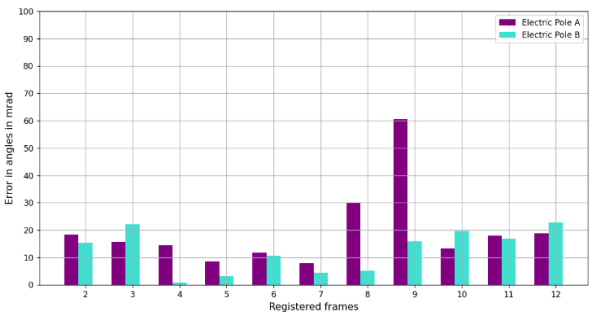

Figure 12. Computed Error in angles between original and registered frames.

Electric Pole	RMS Value (in mrad)
Electric Pole A	24
Electric Pole B	14

Table 3. RMS value of line fitted in electric pole.

RMS value is computed in error in angles of axis of line for both poles in table 3. Here you can see, compared to error in angles of normal vectors of planes, error in angle of electric poles is little bit on higher side. This is because electric poles have fewer points than plane of buildings. As the feature has a denser point cloud, registration tends to occur more accurately. Or we can say if feature has less points registration error tends to be higher. The reason behind this is that registration process relies on the matching corresponding points between frames, and if points are less to determine the feature, the alignment between the features becomes less precise.

4.3.4 Angle between Fitted Line in Two Corner Planes

Planes intersect at corner, where line can be fitted and angle between the lines can be compared. Figure 13 shows the extracted two corner planes from the map.

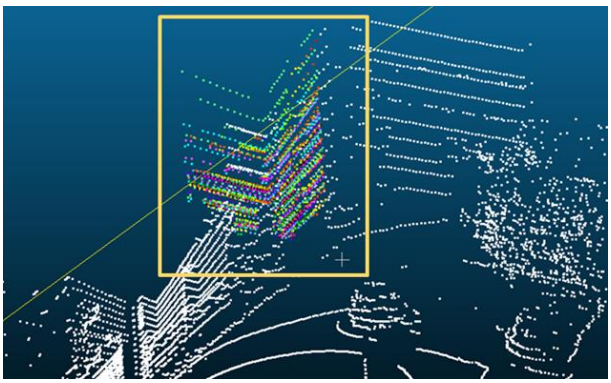

Figure 13. Extracted corner planes from the map.

Line is fitted in the extracted data from original as well as registered frames. The fitted line between data of 1st frame (original) and 2nd frame (registered) is shown in figure 14. (a) shows the fitted line in data 1 and (b) shows the fitted line in data 2.

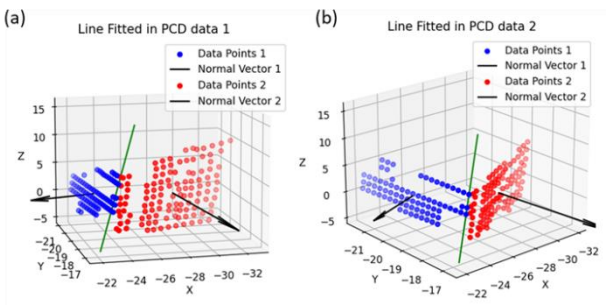

Figure 14. (a) Line fitted in original Data set (b) Line fitted in registered data set.

Subsequently angle is computed between in the fitted lines of original and registered data sets and compared in figure 15.

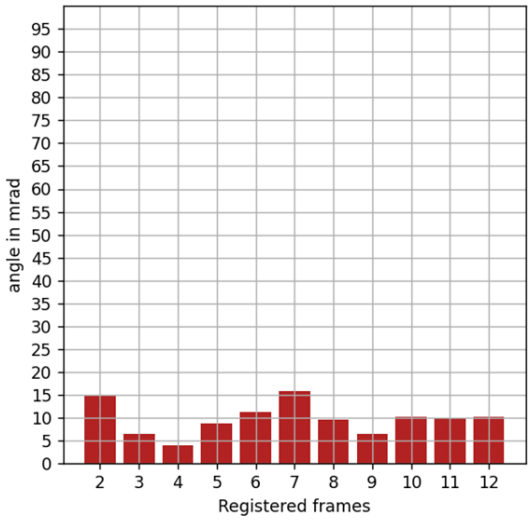

Figure 15. Bar plot between error in angles vs registered frames.

The RMS values of error in all different registered frames is 10 milli radians. This means there is average deviation between the corner planes is about 10 milli radians or half a degree. VLP 16 has horizontal resolution of 0.20 degrees for angular speed of 600 RPM. If we neglect the resolution from the error value, we will get error due to registration process, that is about 0.30.

4.3.5 Deviation between the Planes

Till now, errors in angles or in angular alignment have been identified. However, it’s also essential to check for error in normal deviation or distance between the points. It can be achieved by computing distance between the planes. Ideally, if the planes are completely overlapping, distance between the planes should be zero. But due to registration error, it is important to check deviation between the planes of original as well as registered frames. To check the deviation between the planes, some plane features are extracted from the map and compared. figure 16 shows the top view of one single column feature extracted from map. The deviation between plane of the column is compared between the original and registered frames.

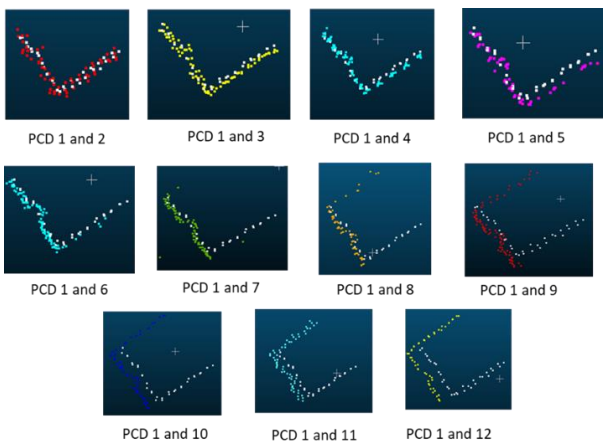

Figure 16. Extracted data points from original and registered frames.

In figure 16, data points extracted from 1st frame (original) are depicted in white, while data points from the 2nd frame (registered) are shown in different colors. It is clearly shown that from 2nd frame to 12th frame, there is an observable increase in deviation. It occurs due to drift error in simultaneous registration of frames. Also same as column 1, planes from other columns are also extracted shown in figure 17 and deviation is computed in figure 18.

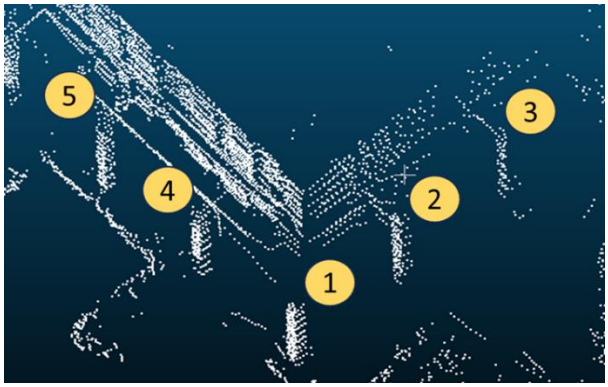

Figure 17. Extracted planes from column features from the map.

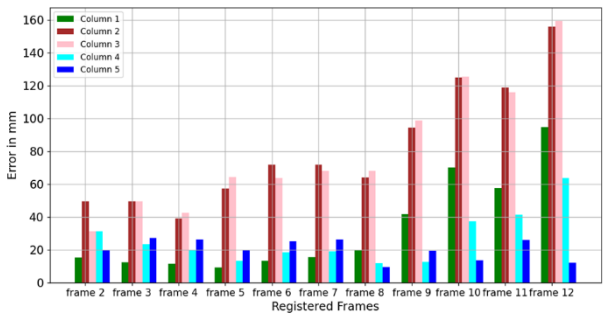

Figure 18. Bar plot: Deviation in mm vs registered frames.

In figure 21, the plot is shown between calculated errors in mm and registered frames. As we can see, the deviation is increasing as senor is moving further. In the 2nd frame the maximum deviation is 50 mm. The maximum deviation in the 12th frame is 158 mm. The increase in error from the initial frame to the last frame is called drift error, which is cumulatively increasing as more frames are registered.

We have conducted thorough accuracy evaluations of the generated map. The reported error serves as a metric for assessing the registration quality. The maximum error in rotation and translation currently stands at 60 milli radians and 10 cm, respectively. Given our utilization of the VLP 16, the coverage is limited, resulting in incomplete capture of features due to lower point cloud density. In future we are planning to perform mapping using the VLP 32, will provide broader coverage and higher point cloud density, are anticipated to yield reduced error compared to the current reported figures.

5. Conclusion and Future Work

This research focuses on transforming a modified ICP registration algorithm into a crucial application, specifically Lidar odometry and mapping. By leveraging this algorithm, the study advances the efficiency and effectiveness of the mapping process and contributes to the evolution of novel LOAM techniques. Our work stands as a novel contribution to the field of mapping, alongside a thorough accuracy evaluation process. This method offers valuable insights for evaluating the alignment between frames, effectively identifying errors in translation and

rotation. Up to this point, our emphasis has been on utilizing the VLP-16 LiDAR sensor without loop closure and with restricted data gathering. In the future, our strategy involves expanding data collection using the VLP-32 LiDAR sensor with good coverage and dense point cloud data across diverse environments, incorporating loop closure conditions to assess trajectory error accurately. Additionally, we plan to integrate advanced machine learning techniques to further refine the alignment process and enhance the overall robustness of the system in dynamic and complex settings. Through these advancements, we aim to push the boundaries of current LOAM methodologies and contribute significantly to the field of autonomous navigation and mapping.

References

Besl, P. J., 1992: A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *14*(2), 239-256. https://doi.org/10.1109/34.121791[1]

De Paula Veronese, L., Badue, C., Auat Cheein, F., Guivant, J., & Ferreira de Souza, A., 2019: A single sensor system for mapping in GNSS-denied environments. *Cognitive Systems Research, 56*, 246-261.

Mendes, E., Koch, P., & Lacroix, S., 2016: ICP-based pose-graph SLAM. 2016 *IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR)*, 195-200. https://doi.org/10.1109/SSRR.2016.7784298.

He, G., Yuan, X., Zhuang, Y., & Hu, H., 2021: An integrated GNSS/LiDAR-SLAM pose estimation framework for large-scale map building in partially GNSS-denied environments. *IEEE Transactions on Instrumentation and Measurement, 70*, 1-9.

Gu, X., Wang, X., Guo, Y., 2020: A Review of Research on Point Cloud Registration Methods. *IOP Conference Series: Materials Science and Engineering, 782*. 022070. https://doi.org/10.1088/1757-899X/782/2/022070.

Khan, Misha U., Zaidi, Syed Azhar Ali, 2021: A Comparative Survey of LiDAR-SLAM and LiDAR based Sensor Technologies. *Mohammad Ali Jinnah University International Conference on Computing (MAJICC)*, 1-8.

Li, P., Wang, Ruisheng, Wang, Yanxia, Tao, Wuyong, 2020: Evaluation of the ICP Algorithm in 3D Point Cloud Registration. *IEEE Access. 8*, 47045-47055.

Mathematics, 2024. Fit plane to 3D data using least squares. https://math.stackexchange.com/questions/1657030/fit-plane-to-3d-data-using-least-squares (13 March 2024).

Paraview, 2024. VeloView: The Velodyne Lidar Viewer based on Paraview Lidar. https://www.paraview.org/veloview/ (5 March 2024).

Wang, Z., & Menenti, M., 2021. Challenges and opportunities in Lidar remote sensing. *Frontiers in Remote Sensing, 2*, 641723. https://doi.org/10.3389/frsen.2021.641723

Segal, A., Hähnel, D., & Thrun, S., 2009. Generalized-ICP. *Proceedings of Robotics: Science and Systems, 5*, 435-442. https://doi.org/10.15607/RSS.2009.V.021.

Tang, J., Chen, Y., Niu, X., Wang, L., Chen, L., Liu, J., Shi, C., & Hyyppä, J., 2015: LiDAR Scan Matching Aided Inertial Navigation System in GNSS-Denied Environments. *Sensors*, *15(7)*, 16710-16728.

Ji, X., & Li, Z., 2019: LLOAM: LiDAR odometry and mapping with loop-closure detection based correction. *IEEE International Conference on Mechatronics and Automation (ICMA)*, 2475-2480.

Zhang, J., & Singh, S., 2014: LOAM: Lidar odometry and mapping in real-time. *Robotics: Science and systems*, *2*, 1-9.