38th International Electric Vehicle Symposium and Exhibition (EVS38) Göteborg, Sweden, June 15-18, 2025

A Study on the Integration of LiDAR and IMU for Safer Autonomous Vehicle Navigation

Bo-Wei Chen[1], Pin-Yung Chen[1, *], Jinn-Feng Jiang[2], Hui-Ting Liang[2],

Hao-Chu Lin[3], Yung-Chuan Chen[1]

[1] *Department of Vehicle Engineering of National Pingtung University of Science and Technology, Taiwan, kelvin_chen@mail.npust.edu.tw*

[2] *Metal Industries Research and Development Centre, Taiwan.*

[3]*Department of Marketing and Supply Chain Management of Overseas Chinese University, Taiwan.*

Abstract

With the rapid advancement of science and technology, there is an increasing consensus that efforts should be aligned with the Sustainable Development Goals (SDGs). In this study, we construct a comprehensive campus-wide map using LiDAR (Light Detection and Ranging) and IMU (Inertial Measurement Unit) sensors. This approach aims to enhance the scientific and technological capabilities of enterprises across various countries through applied research. The objective of this paper is to explore methods for improving the performance and reliability of autonomous vehicles and their sensor technologies. Ultimately, through our experimental approach, we achieved a LiDAR map accuracy of less than 30 cm.

Keywords: LiDAR, Sensing fusion, LIO-SAM, Navigation, Autonomous Vehicle.

1 Introductions

In this paper, the sensor system is integrated to address the specific problem at hand. LiDAR (Light Detection and Ranging) is initially employed to gather environmental information. Before constructing the autonomous driving system, all features of the driving environment, such as buildings, sidewalks, safety islands, trees, etc., are mapped using LiDAR. This approach eliminates the need for continuous data acquisition and processing during autonomous driving, thereby enhancing computational efficiency and prioritizing the acquisition of relevant environmental data.

2 System Configuration

2.1 System communication architecture

The communication architecture of the system is illustrated in Figure 1, highlighting the interactions between the NVIDIA Jetson AGX Orin, the 32-beam LiDAR, an IMU (Inertial Measurement Unit), and a millimeter-wave radar. The LiDAR transmits data via an Ethernet cable to the LiDAR control box through UART (Universal Asynchronous Receiver-Transmitter) communication, which is then forwarded to the NVIDIA Jetson AGX Orin for processing via a second Ethernet connection. The IMU is connected to the USB interface of the NVIDIA Jetson AGX Orin using an RS232-to-USB transmission line.

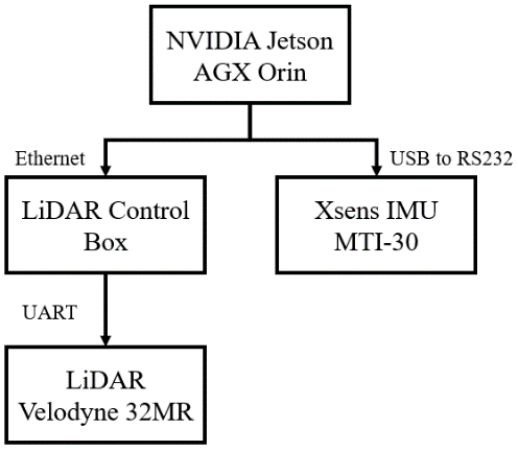

Figure 1: System Communication Diagram

2.2 Robot Operating System

ROS (Robot Operating System) is a software framework designed for robotic system development, which can be broadly divided into three layers: the community layer, the computational graph layer, and the file system layer. The computational graph layer plays a crucial role in data processing and program execution within ROS. This layer primarily includes four key concepts: node manager, node, message, and topic. These concepts, along with other important components, are essential for the communication system. A diagram illustrating the communication system operation is shown in Figure 2.

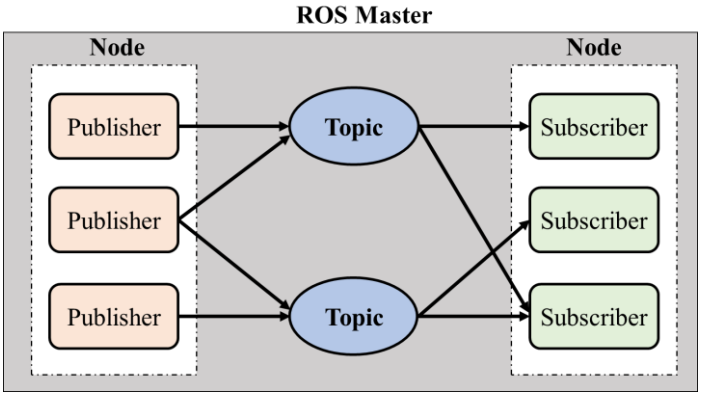

Figure 2: ROS Communication System Operation

2.3 Coordinate Transform Tree

In a robotic system, the TF (Transform) Tree is a fundamental concept used to represent the relationships between coordinate transformations within the robot or system, as illustrated in Figure 3. It describes the relationship between various coordinate frames in the system, typically represented as a Directed Acyclic Graph (DAG). In this graph, each node corresponds to a coordinate frame, and the edges represent the transformation relationships between them. The primary purpose of the TF Tree is to provide a mechanism for coordinate transformations, enabling different parts of the system to describe positions and orientations in various coordinate frames, and facilitating the coordination and merging of data across these frames.

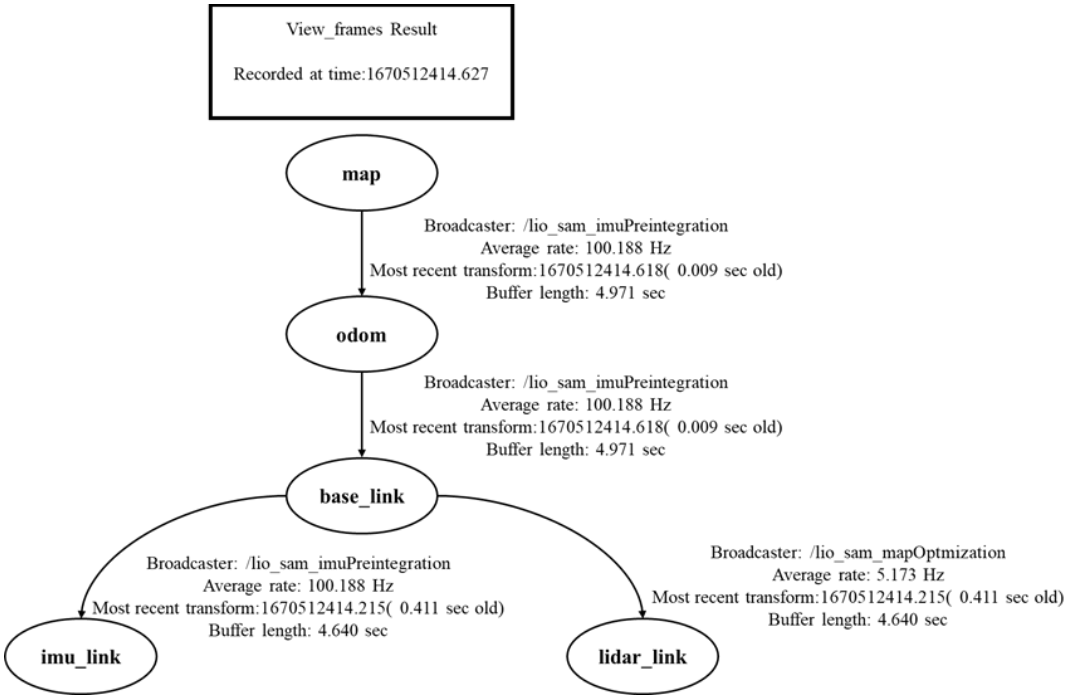

Figure 3: TF Coordinates

2.4 Simultaneous localization and mapping

Simultaneous Localization and Mapping (SLAM) refers to the process by which a robot, starting from an unknown location in an unfamiliar environment, continuously detects features of objects in its surroundings—such as curbs, traffic lights, roadside signs, and other landmarks—using sensors while in motion. In the context of autonomous vehicles, SLAM allows the vehicle to simultaneously determine its position and orientation while incrementally constructing a map based on its current location, thereby achieving both localization and mapping in real time.

3 Lidar-Inertial Odometry via Smoothing and Mapping

3.1 LIO-SAM framework

LIO-SAM (Lidar-Inertial Odometry via Smoothing and Mapping) is a SLAM method that integrates LiDAR and an IMU to facilitate robot localization and environmental mapping. This approach enables the generation of high-precision three-dimensional point cloud maps. The overall framework of LIO-SAM is illustrated in Figure 4 below.

- Point cloud distortion: If the lidar is moving, then the point cloud becomes inaccurate due to the movement of the lidar, so it needs to be corrected for motion distortion.
- Feature extraction: Feature points are extracted by the smoothness of each point.
- IMU pre-integration: The high-frequency data released by the IMU is used to correct the attitude of the vehicle, so that the LiDAR odometer can achieve higher accuracy.
- Mapping optimization: Mapping optimization is to iteratively update the image data through the Gauss-Newton Method (GNM).

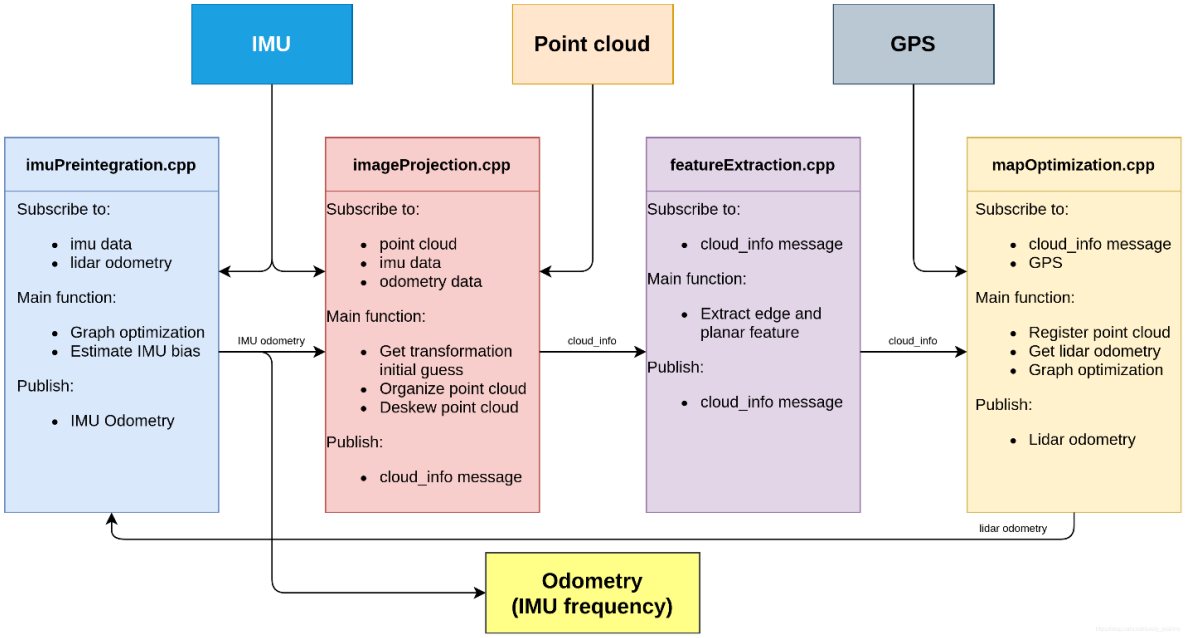

Figure 4: LIO-SAM Overall Framework

3.2 LIO-SAM process

As shown in Figure 5, Matches are made using keyframes, and the frames between keyframes are discarded. (thresholds set at 1m and 10 degrees), there are four factors in total:

- Orange: IMU pre-integration factor.
- Green: LiDAR odometer factor, laser "key frame" and "voxel map composed of the previous N key frames" are matched.
- Yellow: Global Positioning System (GPS) factor is added when the variation of the estimated pose is greater than the variation of the GPS position.
- Black: Loopback detection factor, which is obtained by matching 2m+1 key grid diagram adjacent to both the key grid and the candidate key grid.

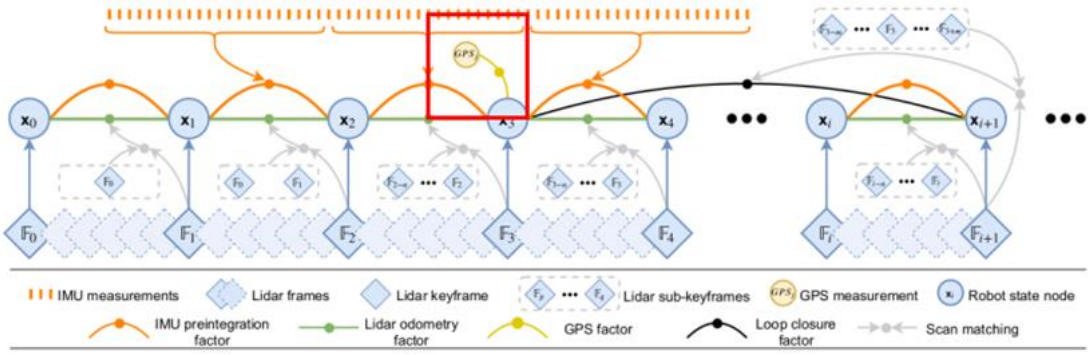

Figure 5: LIO-SAM Overall Process

3.3 Point cloud distortion

Since LiDAR captures a frame of point clouds by emitting laser beams in a 360-degree circle, the points within a given frame are detected at different times. If the LiDAR is moving during the emission of the laser beams, the point cloud becomes distorted due to the movement of the sensor. Therefore, it is necessary to correct for this motion distortion.

As illustrated in Figure 6, based on the timestamp of each point in the point cloud for each frame, the rotational and translational transformations from each point to the starting point of the frame are calculated. Subsequently, each point is transformed into the coordinate system of the starting point.

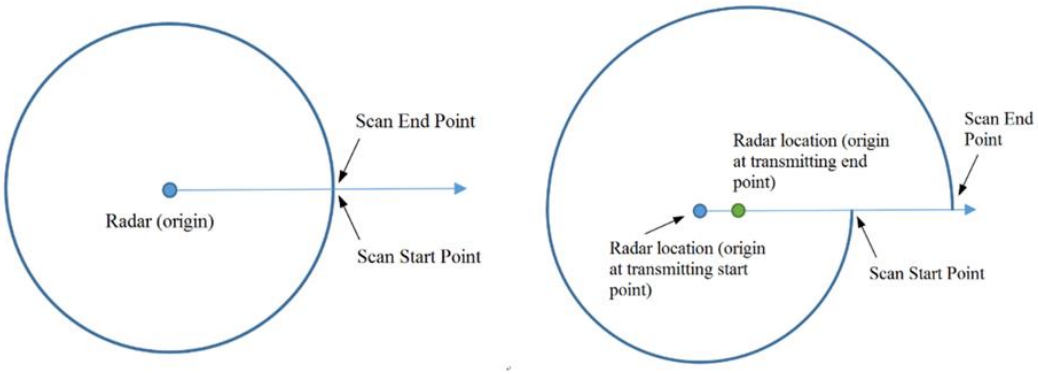

Figure 6: Movement Deformation

3.4 Feature extraction

Before extracting the feature points, clustering is performed on the point cloud. Point cloud clusters containing fewer than 30 data points are considered noise, while the remaining points represent relatively static objects. Feature points are then extracted based on the smoothness of each point. Points with smoothness above a certain threshold are classified as edge points, while points with smoothness below the threshold are considered planar points. The smoothness of each point is estimated based on the average distance to neighboring points before and after the laser line is imaged.

To ensure a more uniform distribution of feature points, the entire image is divided into six horizontal sub-images, and feature points are extracted from each sub-image. For each laser beam in each sub-image, the number of edge points is limited to 20, while there is no restriction on the number of planar points. The smoothness of a point can be calculated using the following formula:

$$c = \frac{1}{|S| \cdot \|r_i\|} \left\| \sum_{j \in S, j \neq i} (r_j - r_i) \right\| \quad (1)$$

S A collection of point clouds of the same harness

r The coordinate distance from a point cloud to the origin

3.5 GTSAM

- Variables: Each vertex in the factor graph is a substitute variable. Suppose we only need to solve the robot pose at each moment, then each vertex is the robot pose at that moment, as shown in Figure 7, X_1, X_2, X_3.
- Values: The values are the values for each variable. When calling the optimizer to optimize the factor graph, we need to set the initial value for each variable first, and then take out the optimized value of each variable from the optimizer after the optimization is completed.
- Factors: Factors are edges in a graph of factors, and each factor can be treated as a constraint. For example, between two consecutive poses, the IMU can calculate the transformation of two poses, which is connected by adding a factor graph as an edge (binary factor). Or at some point there is a GPS data input, the GPS data is an observation, or it can be added to the factor plot as a factor, and only one end of the GPS edge is connected to the variable (unary factor).

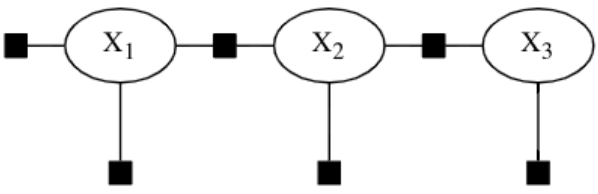

Figure 7: Factor Plots

3.6 IMU pre-integration

The IMU can output three-axis acceleration, three-axis angular velocity, and three-axis geomagnetometer data (nine-axis IMU) in the vehicle coordinate system. It is known that the integral of acceleration provides velocity, the integral of velocity gives distance, the integral of angular velocity gives the angle, and the geomagnetometer can calculate the declination angle relative to the geomagnetic north pole. However, regardless of the precision of the IMU, it is challenging to maintain high accuracy over extended periods, as integration accumulates errors. This limits the IMU's odometry to short-term pose corrections.

In contrast, the frequency of LiDAR data output is approximately 10 Hz, which is much lower than the IMU's frequency. Therefore, for every LiDAR data release, the IMU may output data up to 10 times. As shown in Figure 8, the high-frequency data from the IMU is used to correct the vehicle's attitude, thereby enhancing the accuracy of the LiDAR odometry.

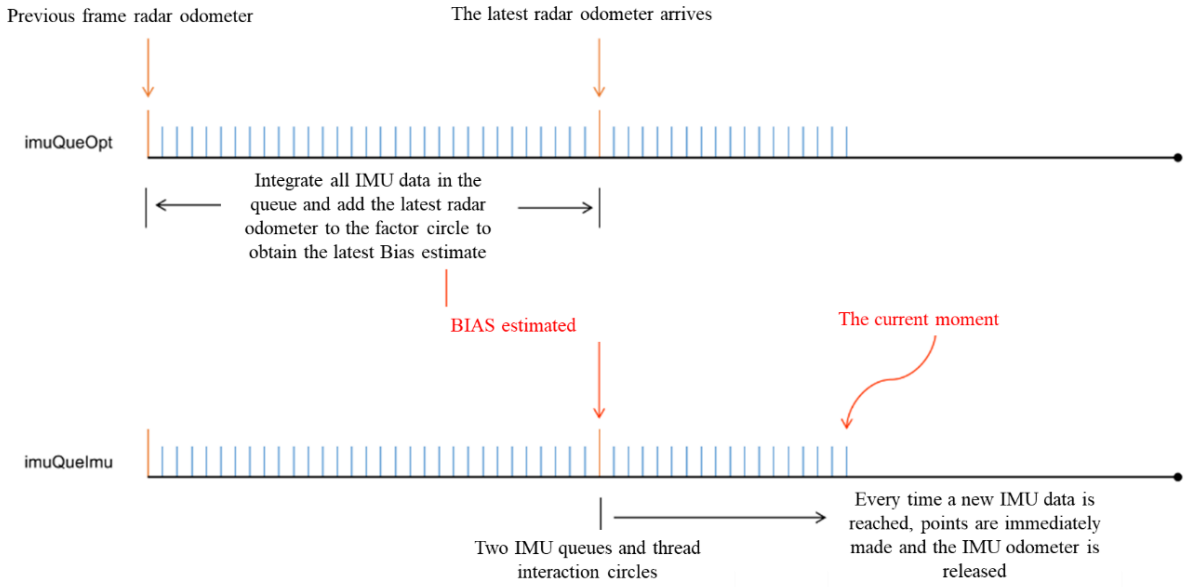

Figure 8: IMU Pre-Integrated Squad

3.7 Mapping optimization

The point cloud matching of the LiDAR odometer is based on point-to-line and point-to-surface distances, and most of them are solved using the least squares method. The GNM also updates the algorithm through iterative updates, and the focus of each step of the algorithm is to find the update direction and step size of the unknown quantity, and each step superimposes the updated vector on the unknown quantity, so that the objective function gradually converges. As shown by the following formula:

$$F_i^m = R_i F_i + t_i \tag{2}$$

$$T_i = \left[t_x, t_y, t_z, \alpha, \beta, \gamma\right] = [R, t] \tag{3}$$

F_i^m Map point cloud coordinate system

F_i Lidar point cloud coordinate system

R Rotation matrix

t translate vector

After obtaining the lidar odometer, it will give priority to judging whether there is a set IMU pre-integration data in the received message, and if so, the data will be used as the initial value of the point cloud, otherwise the IMU angle increment will be used as the initial value, and not all radar frames will be regarded as keyframes, and only if the pose difference from the previous keyframe is large enough will it be adopted as keyframes; Once the keyframes are found, loopback detection can be performed.

3.8 Loopback detection

By checking whether the current position is already the position that has been reached, and matching the poses of the repeated positions. As shown in Figure 9, X5 and X2 form a loop, and through the constraints between the loops, it provides strong information to the factor graph optimizer, so that the overall pose estimation reaches the global optimal state. However, the distance between X5 and X2 needs to be less than or equal to 15 meters to achieve the loopback condition; Loopback detection uses the Iterative Closest Point (ICP) algorithm to calculate the latest keyframe to match the point cloud at the location where the loop may form a loop, and the loopback result is only accepted if the match score is very good.

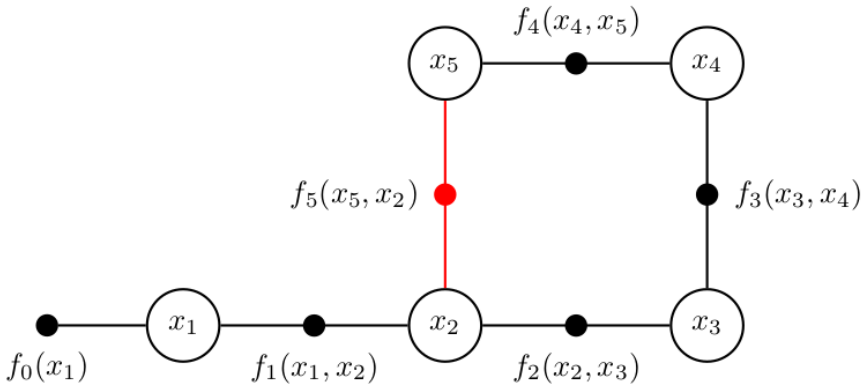

Figure 9: Loopback Detection

Loopback detection steps

Step 1: Find the historical pose and find the point cloud that is closest to the current pose and has a long interval from the current pose.

Step 2: Take the historical pose as a candidate, iterate with the ICP algorithm, correct the pose, compared with the Cartographer, Lightweight and Ground Optimized LiDAR Odometry and Mapping (Lego-LOAM) algorithm, its default offset is smaller, if the offset is larger, it cannot be corrected, the distance of the offset needs to be less than or equal to 15 meters, and the loop detection does not have the function of repositioning, because the premise of this algorithm is that you need to know your approximate position, and match the nearby position in history, and then correct the pose.

3.9 Iterative Closest Point

The basic concept of this algorithm is to continuously adjust the pose of one-point cloud or model in an iterative manner so that it is as aligned as possible with another point cloud or model. In each iteration, the ICP algorithm makes them closer together by finding the most matching pairs of points in two-point clouds and then calculating the transformations between them (usually translation and rotation). This process continues to iterate until a stop condition is met, such as reaching the maximum number of iterations or reaching a small enough error.

IPC- Euclidean distance formula

$$e_{ij} = \|T_0 p_i - q_i\| \tag{4}$$

$$T_0 = [t_x, t_y, t_z, \alpha, \beta, \gamma] = [R, t] \tag{5}$$

R Rotation matrix

t translate vector

p_i Previous point cloud

p_j Behind point cloud

3.10 CloudCompare Point cloud processing software

CloudCompare is a 3D point cloud (mesh) editing and processing software that utilizes a specialized octree structure. It is capable of handling large-scale point clouds, often exceeding 10 million points. Additionally, CloudCompare offers display enhancement tools (such as custom color gradients, color and normal processing, calibrated image processing, Open Graphics Library (OpenGL) shaders, and plug-ins) and filtering algorithms (including low-pass filtering, pass-through filtering, bilateral filtering, Gaussian filtering, statistical filtering, and cloth simulation filtering). In this paper, CloudCompare is used to remove noise and align the map data from different areas of the campus by matching and superimposing point clouds, enabling the detection of errors when map data overlap.

4 Experimental results

4.1 LiDAR range test

In this study, a range test of LiDAR is performed to evaluate its measurement accuracy and error characteristics. By placing the test object at different distances, the test starts at 1 meter and ends at 15 meters, measuring every 1 meter. The test results show that the actual distance and the average distance measured by the LiDAR are 2.91 cm, and the maximum error distance is 4.77 cm, both of which are within the ±3 cm of the average value of the LiDAR measurement, so there is no need to make accuracy correction. The measurement result is shown in Figure 10, and the error distance is shown in Figure 11.

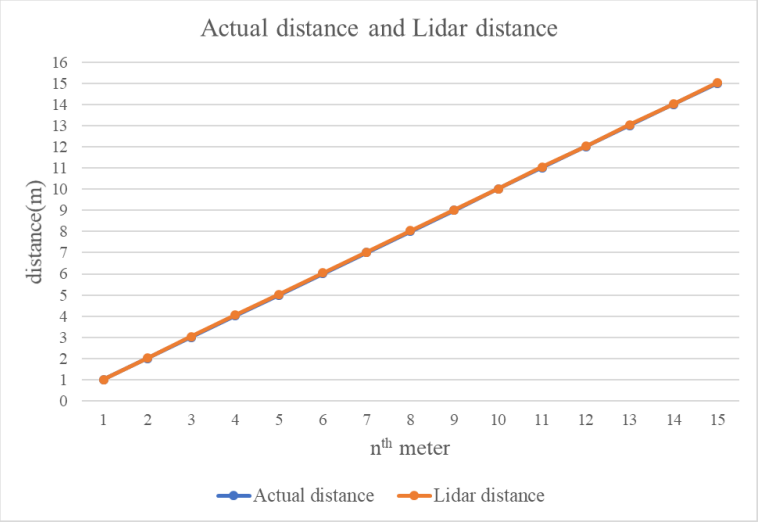

Figure 10: Actual distance and Lidar distance

Figure 11: Measure distance error

4.2 Static testing of IMU

In this study, the acceleration, angular velocity and vector values of the IMU were analyzed by performing a static test of the IMU. The measured analysis data is imported into the LIO-SAM algorithm, because the IMU is a high-frequency published message, which can improve the accuracy and reliability of the algorithm. As shown in Figure 12 and Figure 13 and Figure 14, the acceleration, angular velocity, and vector values for the three axes, respectively.

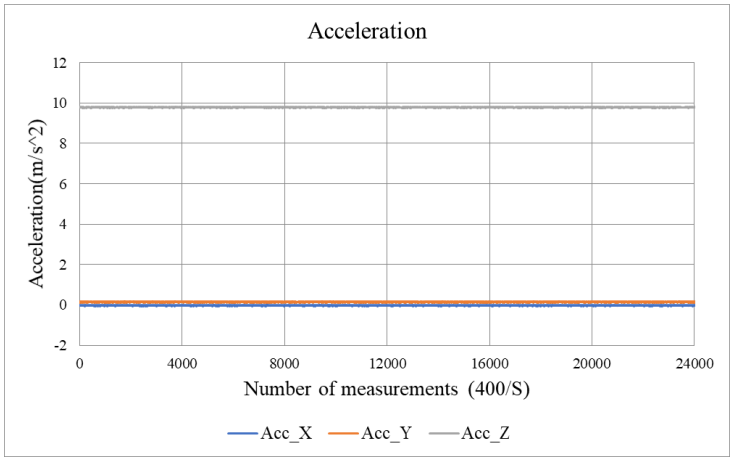

Figure 12: Static Measurement of Acceleration

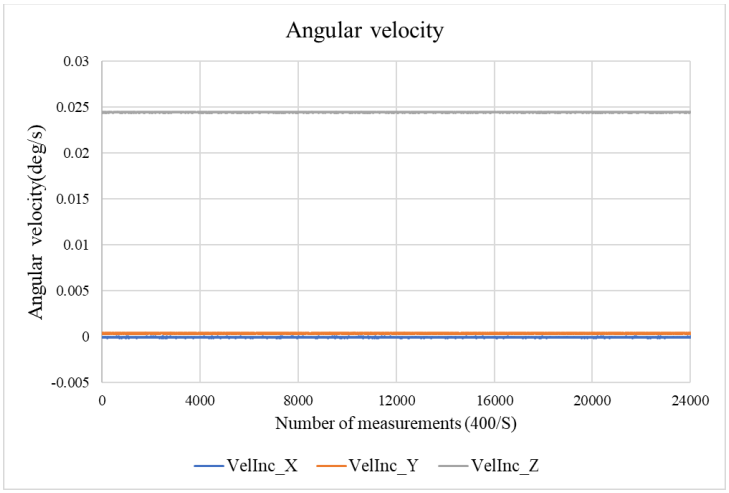

Figure 13: Static Measurement of Angular Velocity

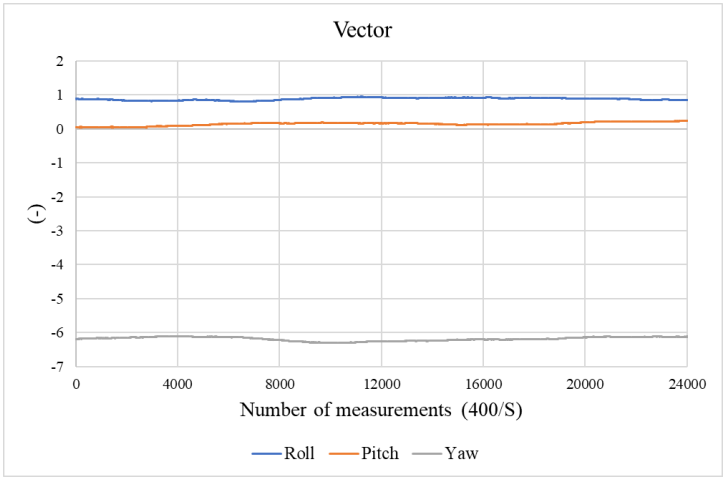

Figure 14: Static Measurement of Vector

4.3 Overlapping images

In this study, the CloudCompare point cloud processing software is used to extract and match the LiDAR point cloud map of each region, and compare the LiDAR point cloud map of each region with the actual navigation map, as shown in Figure 15. In the matching process, the root mean square (RMS) value of the overlapping of the map data is 0.151257 meters. According to the classification standards of the cited map data, the error of high-precision map data should be within 0.3 meters. As shown in Figure 16, six zones exceeded the overall mean error, with three zones (zones 5, 13 and 14) exceeding the 0.3 m specification.

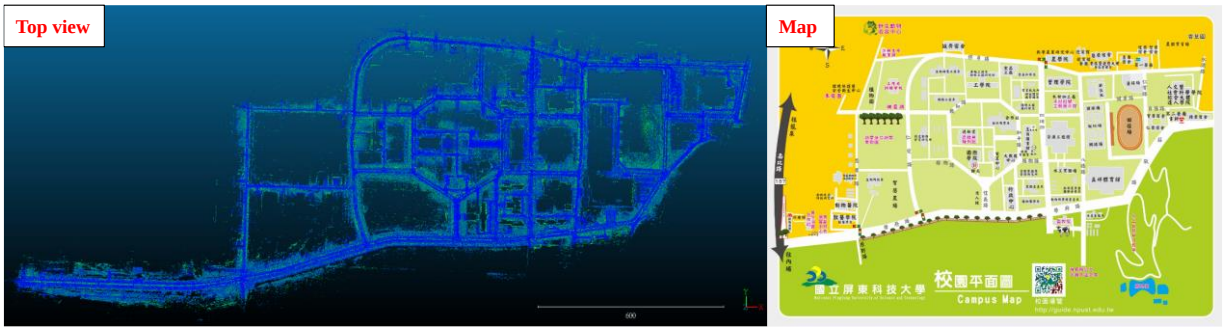

Figure 15: Top View and Map

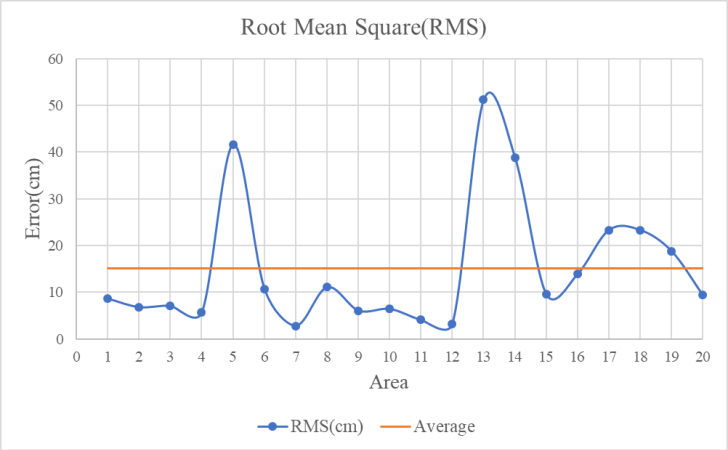

Figure 16: Root Mean Square Error Value

5 Conclusion

During the course of the project, we utilized two types of sensors and, through program development and implementation, obtained physical layer information of both the map and objects. We validated the accuracy of the sensing methods and algorithms through repeated experimental measurements, reducing the average error of the generated map to below 30 cm. Finally, by employing sensor fusion techniques, we integrated and aligned data from multiple sensors with the vehicle's coordinate system.

At present, the overall error of the image data produced by this method is 15 centimeters on average, and it is necessary to further increase the accuracy of the map data and improve the factors that may affect the accuracy of the map data, such as: vibration of the vehicle when the map is established, the correction problem of the sensor itself, filtering of the unfiltered sensor when the sensor data output, and the time accuracy of the sensor fusion. In the future, these problems can be improved or solved, for the time being, the vibration of the vehicle can be solved by improving the tires or chassis to solve the vibration problem during mapping; In the future, the time of the sensor time alignment problem can be used to align the time of the data release of the two data through the method of time stamping, so as to improve the output result.

Acknowledgments

The authors would like to thank the National Pingtung University of Science and Technology of Taiwan and Metal Industries Research and Development Centre for Resource support.

References

[1] Z. Zhang, Y. Zeng, 2022, Applying Deep Learning to MSD Point Cloud Semantics for Segmentation and Image Classification of High-Definition Maps, *Journal of Photogrammetry and Remote Sensing*, Vol. 27, No.1, pp. 15-28.

[2] J. Zeng, P. Hsu, C. Chang, Y. Li, K. Chiang, J. Wang, J. Huang, J. Wu, 2019, Feasibility Assessment of High-precision Map Specification Formulation and Data Collection and Processing for Self-driving Cars, *Journal of Photogrammetry and Remote Sensing*, Vol. 24, No.3, 2019, pp. 173- 182.

[3] C. Chang, Y. Tseng, 2022, Applying Deep Learning to MMS Point Cloud Semantic Segmentation and Image Classification for HD Map Generation, *Journal of Photogrammetry and Remote Sensing*, Vol. 27, No.1, 2022, pp. 15-28.

[4] T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti, and D. Rus, 2020, LIO-SAM: Tightly-coupled Lidar Inertial Odometry via Smoothing and Mapping, *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Las Vegas, NV, USA (Virtual).

[5] J. Jo, J. Ha, S. Kim, J. Lee, J. Lee, K. Jo, 2024, Enhancing Object Detection Accuracy in Autonomous Vehicles through LiDAR Point Cloud Registration, *EVS37 Symposium COEX*, Seoul, Korea, April 23-26, 2024.

[6] K. Kang, D. Kim, H. Ryu, Y. Han, 2024, Angle Offset Calibration Method to Improve LiDAR Angle Accuracy, *EVS37 Symposium COEX*, Seoul, Korea.

Presenter Biography

Bo-Wei Chen, Graduate Student
He received his B. S. degree in the Department of Vehicle Engineering from the National Pingtung University of Science and Technology, Taiwan, in 2024. He is currently pursuing his Master's degree in the Department of Vehicle Engineering of the National Pingtung University of Science and Technology.

Pin-Yung Chen, Assistant Professor
Pin-Yung Chen received his Ph.D. degree in the Department of Power Mechanical Engineering from the National Tsing Hua University, Hsinchu, Taiwan, in 2018. Before 2023, Dr. Chen was been a senior researcher at the Industrial Technology Research Institute(ITRI), Hsinchu, Taiwan. Currently, Dr. Chen is presently an assistant professor in the Department of Vehicle Engineering, National Pingtung University of Science and Technology, Pingtung, Taiwan. His research interests are intelligent vehicles and autonomouscontrol systems. Dr. Chen is a member of the IEEE Control Systems, Chinese Institute of Engineers Society of Taiwan, Intelligent Transportation Society (ITS) of Taiwan, and SAE Taipei Section.

Jinn-Feng Jiang, Deputy Director
Jinn-Feng Jiang received his Ph.D. degree in the Department of Mechanical Engineering from the National Taiwan University of Science and Technology. With many years of dedicated service in Taiwan's automotive industry, possesses extensive experience in automotive industry R&D. Also served as a research scientist in the DoIT of MOEA, and played a crucial role in advancing Taiwan's smart EV industry and facilitating the production of Taiwan's brand, Luxgen. Recently focusing on the research and promotion of autonomous vehicles and wire-controlled chassis vehicle technology.

Hui-Ting Liang, Manager
Liang Hui-Ting received her master's degree in Management from National Cheng Kung University. With several years of experience in research and management in the automotive industry, her primary research area focuses on product reliability and testing verification. Recently placed an emphasis on the development of testing verification techniques and regulations for industry promotion related to components of electric vehicles, such as suspension, steering, and braking systems.

Hao-Chu Lin, Associate Professor
Hao-Chu Lin received his Ph.D. degree in management science from the National Taiwan University of Science and Technology in 2011. He was working in the Department of Industrial Technology, Ministry of Economic Affairs, for over twenty years. and has participated include Industrial Technology Development Program, Innovative R&D Centers in Taiwan Program, International Cooperation Program, Promotion of the local industries Innovative Technology Applications and Service Program, etc. Currently, he is an associate professor in the department of Marketing & Supply Chain Management of Overseas Chinese University, Taiwan. His research interests include public administration, technology strategy, foresight research management of innovation and ICT application.

Yung-Chuan Chen, Professor
Yung-Chuan Chen received his M.S. and Ph.D. degrees in Mechanical Engineering from National Sun Yat-sen University, Kaohsiung, Taiwan, in 1991 and 1996, respectively. He is currently a Professor in the Department of Vehicle Engineering at National Pingtung University of Science and Technology, Pingtung, Taiwan. His primary research focuses on the areas of structural optimization design and stress analysis.