Mechatronic Systems and Control, Vol. 53, No. 10, 2025
An Open Access Paper

LASER SLAM ALGORITHM BASED ON MULTIPLE CONSTRAINTS IN DEGENERATE SCENARIOS

Songyi Lu,* Quande Yuan,** Wenhao Huang,* Guoyin Zhang,*** and Yifan Liu****

Abstract

In the factory environment, the spatial scale is large and there are many similar scenes. Traditional laser SLAM methods are prone to scene degradation, low positioning accuracy, and large mapping errors. This method is applied in the front-end by incorporating pre-integrated processed inertial measurement unit data and lidar data as constraints into subsequent optimisation. A keyframe is selected at regular intervals, and the best estimated pose is obtained through radar scanning matching. Then the best estimated pose is added to the sub map to estimate the robot's pose, and cumulative errors are eliminated through loop detection and global optimisation. A comparative experiment will be conducted between the laser SLAM method based on graph optimisation and the method proposed in this paper. The experiment shows that the improved SLAM method based on multiple constraint conditions in this paper can improve the mapping accuracy while reducing positioning errors compared to traditional SLAM methods.

Key Words

Laser slam, degradation scenario, graph optimisation, inertial measurement unit (IMU), multiple constraints

1. Introduction

In recent years, more and more factories are using intelligent equipment [1]. Through more intelligent equipment, the production efficiency can be greatly improved. At the same time, the demand for mobile robots in the factory is also constantly increasing. Transporting goods through mobile robots can not only save labor costs, reduce the risk of accidents for people when handling goods, but also improve the efficiency of goods handling. As a result, more and more companies and enterprises are increasing their investment in mobile robot research. Simultaneous localisation and mapping (SLAM) is an extremely important part of the study of mobile robots.

SLAM is a technique for simultaneous positioning and map construction [2], also known as the concurrent mapping and localisation (CML). SLAM technology allows robots to locate and map their surroundings in an unfamiliar environment. Implementation of SLAM methods can often be done using multiple different sensors. Therefore, we usually divide SLAM into laser SLAM [3] and the visual SLAM [4].

Visual SLAM uses visual sensors (depth cameras) to estimate the robot's own pose [5] and build a map of the surrounding environment. The robot captures the image of the environment through the camera and extracts the feature points in the image. Then, by comparing the feature points in different frame images, the robot can estimate its own motion and build a three-dimensional map of the environment. However, visual SLAM is influenced by environmental illumination, texture and other factors, so there may be visual feature extraction and tracking errors, so the accuracy and stability in some complex scenes are relatively low.

Different from visual SLAM, laser SLAM has a better performance in positioning accuracy and mapping stability, mainly due to the point cloud data derived from accurate lidar sensors. Lidar obtains information about the distance and angle of objects in the environment by transmitting a laser beam and receiving reflected light signals. Most of the current mobile robot projects also use lidar.

The SLAM system framework (as shown in Figure 1) is roughly divided into five parts: sensor data acquisition and processing, front-end odometer, back-end optimisation, loop detection, and map construction. The laser radar emits laser ranging information at a certain frequency, the front odometer calculates the corresponding machine position through the laser ranging information, which contains the cumulative error, the back end optimisation will optimise the global trajectory, and the loop detection

* Jilin Institute of Chemical Technology, Jilin City 132013, China; e-mail: lusongyi@jlict.edu.cn; huangwenhao@jlict.edu.cn
** Changchun Institute of Technology, Changchun 130012, China; e-mail: yuanqd@ccit.edu.cn
*** China National Heavy Duty Truck Manufacturing Company, Jinan, China; e-mail: zhangguoyin@jlict.edu.cn
**** Huali Integrated Circuit Manufacturing Co., Ltd., Shanghai, China; e-mail: liuyifan@jlict.edu.cn
Corresponding author: Quande Yuan

Recommended by Teng Li
(DOI: 10.2316/J.2025.201-0558)

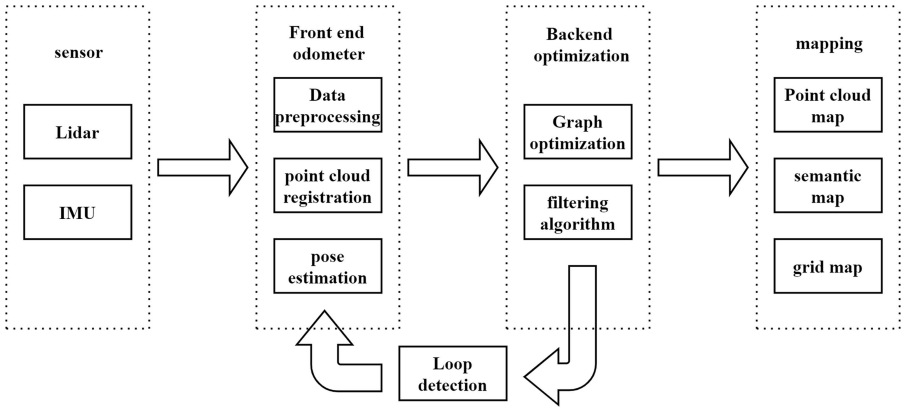

Figure 1. The SLAM system framework.

is also running to eliminate the cumulative error during the whole system operation.

In SLAM systems, data processing methods are also classified into loose coupling and tight coupling. In a loosely coupled system, each sensor is treated as an independent module, and each module uses the information it obtains to calculate and obtain its own results. Then put these results into the state optimisation equation for optimisation, in order to obtain the fused optimised pose. Zhang and Singh proposed the V-LOAM method [6], which utilises the results of loosely coupled visual inertial odometry as the initialisation prior for the lidar mapping system. Palieri *et al.* [7] proposed a multi-sensor fusion scheme LOCUS centered on lidar, which adds a state detection module to select the most optimal option for optimisation before lidar scanning matching.

A tightly coupled system is different, as it directly processes the raw or intermediate data obtained from various sensors together. The measured values of different sensors are put into the state optimisation equation, and then the fusion optimised pose is obtained. Han's team integrated the lidar inertial system and visual inertial system into a tightly coupled frame with a sliding window [8], where when one system detects a fault, the other frame can work independently or work together under good conditions. Zuo *et al.* [9] proposed an LIC fusion framework based on MSCKF, which utilises inertial measurement unit (IMU) measurements, sparse visual features, and radar features, supplemented by temporal and spatial calibration for state estimation. Compared with loosely coupled methods, tightly coupled methods can fully consider the inherent constraints between sensor data and jointly determine the final result.

The laser SLAM algorithm is currently mainly divided into filter based laser SLAM algorithm [10] and graph optimisation [11] based laser SLAM algorithm. The SLAM algorithm based on filters is suitable for scenarios with high real-time requirements due to its small computational complexity. However, compared to the laser SLAM algorithm based on graph optimisation, it lacks optimisation for global trajectories and may perform poorly in complex environments.

In graph optimisation algorithms, there are mainly Hector SLAM method and cartographer algorithm. By improving the generation of initial values in Hector SLAM, the reliability of the algorithm can be improved. For example, Wei [12] used the Near Point Matching (PL-ICP) method to locate and correct the direction and position of the robot using a reference frame during the initialisation stage of the algorithm. However, Hector SLAM only uses lidar in the process of building maps, without considering odometer information. The overall accuracy of the algorithm depends on lidar and has high frequency requirements. Xin and Huasong [13] improved the accuracy of point cloud matching and significantly enhanced the quality and trajectory accuracy of map construction by using velocity pre integration method on the original cartographer framework for pose enhancement. However, it is still prone to image feature extraction or matching failure, resulting in map ghosting or angle misalignment. Liang [14] designed a multi-sensor fusion scheme based on pose increment, which improved the effectiveness of cartographer mapping. However, if the data volume accumulates, the real-time performance may decrease, leading to problems, such as map ghosting and increased positioning errors. At present, the mainstream laser SLAM algorithm based on graph optimisation is difficult to provide correct poses in factory scenes with single, regular or symmetrical environmental textures. Laser matching algorithms are difficult to distinguish similar scenes and are prone to scene degradation caused by incorrect matching.

For the above issues, this article introduces IMU [15] to compensate for the impact of scene degradation, and treats the inertial coordinate system and radar coordinate system as a whole. The point cloud information of adjacent two frames is selected, and the initial value of the optimisation factor is used to map the point cloud information of the two frames to the inertial coordinate system to obtain new point cloud information. Then, the next frame is mapped to the inertial coordinate system corresponding to the previous frame. The accuracy of the optimisation factor is judged by the degree of overlap between the two frames, and the distance between the two points is calculated. This can reduce the errors generated in pose estimation, thereby improving the positioning accuracy. Through public dataset simulation experiments, it was verified that the method of using inter frame matching

error can improve the mapping accuracy of multi constraint laser SLAM method.

2. Related Work

The SLAM algorithm requires external sensors to input odometer information for pose estimation, and IMU and lidar data need to be processed first during pose estimation. Further fuse and calibrate the data from IMU and lidar sensors, and unify the data from both sensors into the vehicle coordinate system.

2.1 IMU Principle and Function

IMU, also known as inertial measurement unit, is a sensor that can provide effective local motion estimation and accurately determine the attitude of an object, including direction, angle, and tilt. IMU mainly consists of gyroscopes and accelerators, and may also include measurement units such as magnetometers.

A typical six axis IMU measures linear acceleration and rotational angular rate from three directions through its built-in gyroscope and accelerometer. Accelerometers can measure the linear acceleration of objects along the x, y, and z axes, while gyroscopes are responsible for measuring the angular velocity of objects around these three axes. Internally, they can calculate angular velocity and acceleration based on other physical quantities such as force or time, but externally, only the accuracy of their measurement of angular velocity and acceleration, as well as the relationship between these quantities and vehicle position and attitude, need to be considered.

IMU plays an important role in SLAM systems, estimating the position and attitude changes of robots during motion by measuring the acceleration and angular velocity on the three axes. This process is called odometer. In this method, the fusion of IMU data and laser sensor data to some extent compensates for the impact of scene degradation. When the laser sensor cannot obtain sufficient feature information, the IMU can provide continuous attitude and velocity information to help the system maintain stable positioning and map construction. The pre-integrated IMU data and lidar data are combined as constraint conditions to be added to the subsequent optimisation, generating sub maps and estimating the robot's pose. Then, cumulative errors are eliminated through loop detection and global optimisation, which improves the overall robustness and accuracy of the SLAM system to a certain extent.

2.2 IMU Pre-Integral Processing

Pre-integrating IMU data can process accumulated IMU data over a longer period of time, just add the previous integral and the motion increment between the two frames to reduce the number of integration in the optimisation process, thus saving computing resources and improving real-time performance [16]. In an IMU system, five variables are considered: rotation R, position p, angular velocity ω, linear velocity v, and acceleration a.

The position p, speed v and attitude q at time i and j are related as in 1:

$$p_j^w = p_{b_i}^w + v_{b_i}^w \Delta t_{\mathrm{ij}} + \frac{1}{2} g^w \Delta t_i^2 + \iint_{t\in[i,j]} R_t^w \left(\widehat{a}_t - b_{a_t}\right) dt^2 \quad (1)$$

$$v_{b_j}^w = v_{b_i}^w + g^w \Delta t_{\mathrm{ij}} + \int_{t\in[i,j]} [R_t^w \left(\widehat{a}_t - b_{a_t}\right)] dt \quad (2)$$

$$q_{b_j}^w = q_{b_i}^w \otimes \int_{t\in[i,j]} \frac{1}{2} \Omega[(\widehat{\omega}_t - b_{\omega_t})\, q_t^{b_k}] dt. \quad (3)$$

In the above equation, t is the time point, w and b are the world and IMU coordinates, g is gravity, $\widehat{a}_t$ is the observed value of the speedometer, $\widehat{\omega}_t$ is the gyroscope, b_{a_t} is the acceleration bias, b_{ω_t} is the gyroscope bias, and $\otimes$ is the quaternionic multiplication. Separating from the discretization equation of motion between time i and time j, the IMU pre-integral quantity is obtained as follows 2:

$$\alpha_{b_j}^{b_i} = \iint_{t\in[i,j]} R_t^{b_i} \left(\widehat{a}_t - b_{a_t}\right) dt^2 \quad (4)$$

$$\beta_{b_j}^{b_i} = \int_{t\in[i,j]} [R_t^{b_i} \left(\widehat{a}_t - b_{a_t}\right)] dt \quad (5)$$

$$\gamma_{b_j}^{b_i} = \int_{t\in[i,j]} \frac{1}{2} \Omega[(\widehat{\omega}_t - b_{\omega_t})\, \gamma_t^{b_i}] dt. \quad (6)$$

$\alpha_{b_j}^{b_i}, \beta_{b_j}^{b_i}, \gamma_{b_j}^{b_i}$ are position pre product component, velocity pre product component, and attitude pre product component, respectively. The state information at any time can be obtained by the IMU equation of state.

2.3 Lidar Data Preprocessing

The laser odometer needs to first derive the residual difference of the relative pose and the two optimisation variables. The laser odometer factor from time i to time j is shown in 3, and the neutralisation indicates the translation matrix and the rotation matrix in the laser odometer, respectively T_L, R_L:

$$\begin{aligned} r_{L_T} &= R_i^T (T_j - T_i) - T_L \\ r_{LR} &= \ln \left(R_j^T R_j R_L^T\right)^V. \end{aligned} \quad (7)$$

Then evaluate the translation and rotation changes at time i and j, respectively to obtain the corresponding Jacobian matrix, such in 4:

$$\begin{bmatrix} \dot{r_{LT}} \\ \dot{r_{LR}} \end{bmatrix} = \begin{bmatrix} -R_i^T & R_i^{-1}(t_j - t_i) \\ 0 & -J_r^{-1}(r_{LT})(\exp r_{LR}{}^{\wedge})^T \end{bmatrix} \begin{bmatrix} \delta T_i \\ \delta R_i \end{bmatrix} \quad (8)$$

$$\begin{bmatrix} \dot{r_{LT}} \\ \dot{r_{LR}} \end{bmatrix} = \begin{bmatrix} R_i^T & 0 \\ 0 & -J_r^{-1}(r_{LT}) R_0 \end{bmatrix} \begin{bmatrix} \delta T_i \\ \delta R_i \end{bmatrix}. \quad (9)$$

To obtain the results of the translation change and the rotation change between the two moments, after forming the Jacobian matrix, it can be added to the optimisation,

Table 1
Calibration Parameters of Inertial Navigation for Vehicle Coordinate System

Installation Parameters		Unit	Numerical Value
Inertial navigation angle installation deviation roll angle	Roll angle	°	0.461
	Pitch angle	°	−0.566
	Heading angle	°	0
Antenna angle installation deviation	Roll angle	°	0
	Pitch angle	°	0
	Heading angle	°	0
Inertial navigation—Rear axle centre position vector	longitudinal separation (x)	m	−0.34
	Left and right distance (y)	m	0
	Up and down distance (z)	m	0.45
Inertial navigation—Main antenna position vector	longitudinal separation (x)	m	−0.44
	Left and right distance (y)	m	0
	Up and down distance (z)	m	0.14
Track width	m	0.7	

and together with the IMU pre-integration factor as a constraint to obtain a more accurate pose, which further improves the accuracy of the subsequent map construction.

2.4 IMU and Lidar Coordinate System Calibration

Due to the fact that various sensor data is generated based on their own coordinate system, it is necessary to fuse and calibrate different sensor data when multiple sensor data are needed. This method unifies the data of different sensors to the car body coordinate system.

This method installs IMU sensors and lidar sensors on the platform. First, calibrate the inertial navigation coordinate system and the vehicle coordinate system, including the setting of the inertial navigation on the rear axle main antenna, the setting of the inertial navigation angle installation deviation, etc. Table 1 lists the specific parameters that need to be written into the equipment after measuring the relevant distances in each axis.

After completing the coordinate system calibration between inertial navigation and vehicle body, convert the coordinate systems of inertial navigation and radar. There is a six degree of freedom rigid body transformation between inertial navigation and radar. By solving the relative external parameters (R, T) between inertial navigation and radar, the coordinate information of the radar can also be transformed into the vehicle coordinate system. During the calibration process, this method uses $O-X_IY_IZ_I$ as the inertial coordinate system and $O-X_LY_LZ_L$ as the radar coordinate system, and then solves the external parameters through optimisation methods, using (R, T) as the factor that needs to be optimised.

Firstly, set the status of the starting point position of the system platform to A_1. At this time, the inertial coordinate system and radar coordinate system are integrated, and the information of a point cloud near the starting point is L_1. Then, let the platform travel a certain distance and stop at A_2. At this time, the information of a point cloud near the starting point is L_2. During this process, the inertial navigation obtains (R_1, T_1) for the rotation and translation during the motion process through the integration operation shown in Fig. 2. The initial value of the factors (R, T) that need to be optimised is set to (R_0, T_0).

In the process of optimising coordinate information, first use the initial value of optimisation factor (R_0, T_0) to map point cloud information L_1 and L_2 to states A_1 and A_2 corresponding to $O-X_IY_IZ_I$. The newly generated point clouds are L_1' and $L_2^{'}$. Based on the inertial integration operation, (R_1, T_1) is obtained, and then $L_2^{'}$ is mapped to state A_1 corresponding to $O-X_IY_IZ_I$, resulting in point cloud $L_2^{''}$. So the accuracy of the optimisation factors (R, T) can be determined based on the degree of overlap between L_1' and $L_2^{''}$. For each point in L_1', find the shortest point in $L_2^{''}$, consider it to be the corresponding point, and then calculate the distance between the two points. For disordered point clouds, using the kd-tree nearest neighbour search algorithm can improve search efficiency. Then, the square of the distances between each corresponding point is added to obtain the optimisation equation, as shown in 5, where q_i and p_i are points in point clouds $L_1^{'} = \{q_1, q_2, \ldots, q_n\}$ and $L_2^{'} = \{p_1, p_2, \ldots, p_n\}$. Finally, the rotation matrix R and the translation matrix T that minimise E are iteratively solved to complete the calibration:

$$E = \min \sum_{i=1}^{n} \left\| q_i - (Rp_i + T) \right\|^2 . \tag{10}$$

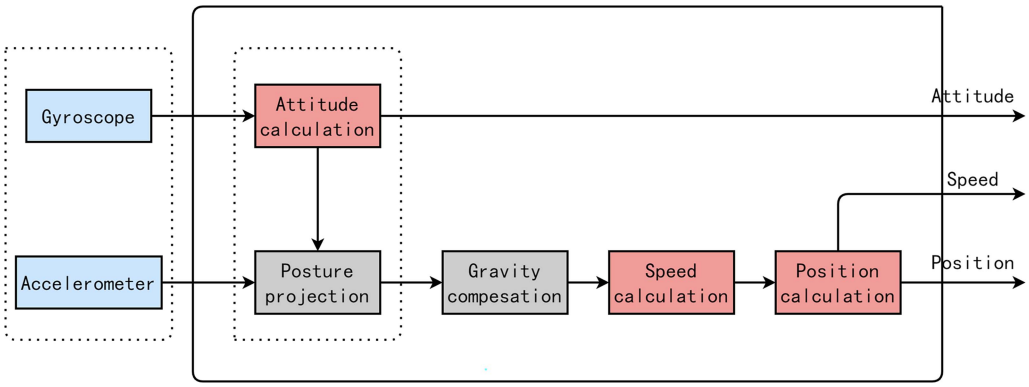

Figure 2. Inertial navigation pose calculation process.

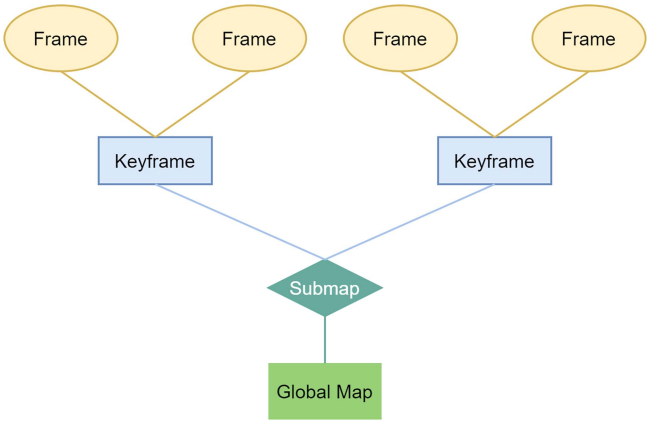

Figure 3. Submap selection.

3. A 2D Laser SLAM Based on Multiple Condit Constraints

The laser SLAM algorithm in this paper is improved by the current framework of cartographer algorithm, which performs better in accuracy, real-time, and robustness. In the front end, most SLAM systems need keyframes to store scanning data in different regions, and the selection of submaps is a more convenient way to manage scanning data. As shown in Fig. 3, this method selects a key frame at every other distance, and then matches by radar scanning (Scan Matching) [17]. Find the best estimated pose and add the best estimated pose in the keyframe to the subgraph (Submap). In this process, if the robot moves beyond the current submap, or the current submap contains more keyframes than a certain number, a new submap is generated again. At this time, there is no data generated at the submap. This paper takes the current frame as the centre of the submap. At this time, the relationship between the submap and the world coordinate system and the relationship between the current scan data and the world coordinate system are consistent. It also puts the keyframe of the old submap into the new submap, which also prevents the new submap from being registered due to lack of data. Finally, merging the raster map of each submap yields the global map.

The improved algorithm (as shown in Figure 4) takes IMU pre-integral data and lidar data as the constraint optimisation pose, and IMU pre-integral can be used as the observation data of the current map scan matching and the loaded prior map as the match. Iterative closest point (ICP) is mainly used for feature matching [18] algorithm. The point-to-line ICP algorithm is used here. The ICP algorithm divides the scan matching problem into data association and pose estimation, and it rotates these two steps until the results converge. In the nearest neighbour search, we also need to find multiple nearest neighbours, and fit a straight line, and finally calculate the distance between the target point and the straight line. Set k nearest neighbour points as $x_1, \ldots, x_k$, Set the line equation to:

$$ax + by = 0. \tag{11}$$

Where a, b, c are the parameters of the line, which is the least squares problem when the line fitting becomes the parameter estimation:

$$(a, b, c)^* = \arg\min \sum_{i=1}^{N} \left\| ax^i + by^i + c \right\|_2^2 . \tag{12}$$

Then arrange the point coordinates into a matrix:

$$A = \begin{bmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ & \cdots & \\ x_k & y_k & 1 \end{bmatrix} . \tag{13}$$

Then we find the minimum singular value vector of A. After obtaining the line parameters (a, b, c) of the nearest neighbour point, the vertical distance from any point (x, y) to this line can be expressed as:

$$d = \frac{ax + by + c}{\sqrt{a^2 + b^2}}. \tag{14}$$

The denominator part is a fixed constant, temporarily ignored first, so the residue can be left out:

$$e = ax + by + c. \tag{15}$$

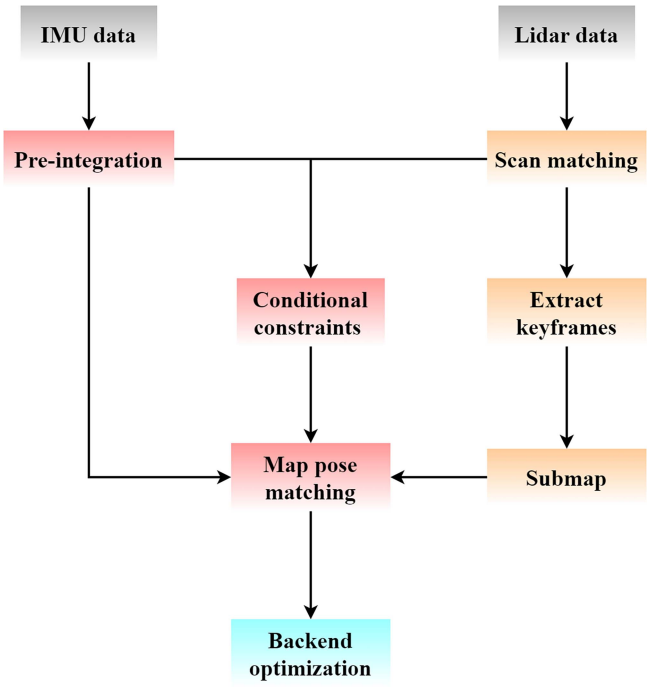

Figure 4. Improves on the SLAM framework.

As the target formula, the Jacobian matrix corresponding to the line equation is:

$$\frac{\partial e}{\partial x} = a, \quad \frac{\partial e}{\partial y} = b. \tag{16}$$

Add the above results to the position of the lidar itself, and set the position and angle of the lidar as $x = (x, y, \theta)$, so for a laser point, set the distance and Angle as, and transfer the point to the world coordinate system to obtain p_i^w. Its several nearest neighbour fitting line parameters are (a_i, b_i, c_i), and the Jacobian matrix of its residual pose can be expressed using the chain rule:

$$\frac{\partial e_i}{\partial x} = \frac{\partial e_i}{\partial p_i^W} \frac{\partial p_i^W}{\partial x}. \tag{17}$$

D-LIOM algorithm [19] propose to classify IMU data into both dynamic and static initialisations. In this paper, the timestamp of the lidar data of the current frame and the previous frame is selected to calculate the pre-integration of the IMU data between the two frames of lidar data. During scanning, the LIDAR and IMU data are matched to obtain the observed pose and used as the error generated by the constrained optimised pose estimation.

4. Experiment and Results Analysis

This article uses simulation software to analyse the experimental results, using the publicly available data set of the German Museum, which includes IMU data and lidar data. This article mainly studies the mapping effect of the method and cartographer algorithm in open and narrow scenes, which can meet the requirements of this experiment.

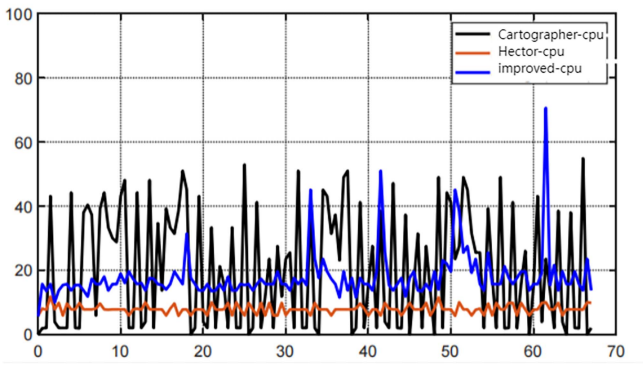

Figure 5. CPU usage during the mapping process.

Table 2
CPU Usage During the Mapping Process

SLAM Algorithm	Cartographer Algorithm	Hector Algorithm	Improved Algorithm
Average occupancy rate of CPU (%)	20.8950	8.5641	16.9126

4.1 Experimental Platform

The hardware equipment selects a laptop with Linux ubuntu 18.04 operating system; the corresponding robot operating system (ROS) [20] is melodic.

This paper uses Rviz to display the built maps. Rviz (ROS Visualisation) is a powerful visualisation tool in ROS, providing a 3D interface that can be used to display information about sensor data, robot status, and mapping results. Users can display a variety of information by adding different display panels (Display Panels). For example, by adding TF panels to display the transition relationships between different coordinate systems, by adding submaps panels to display the constructed maps, etc.

4.2 CPU Usage During the Mapping Process

Using the top analysis tool in Ubuntu, analyse the CPU usage of Hector algorithm [21], cartographer algorithm, and the algorithm proposed in this paper. The processor of the computer running the laser SLAM algorithm is an Intel Core i5-8300H CPU. The CPU utilisation of three laser SLAM algorithms is shown in Figure 5.

Compare the average CPU usage of three laser SLAM algorithms, as shown in Table 2.

From Fig. 5, it can be seen that the occupancy rate of our algorithm exhibits periodic peaks, and as the mapping process progresses, the peaks become higher and higher. The reason for this situation is that the algorithm in this article performs closed-loop detection every once in a while, and over time, more and more local submaps are created. Using the current lidar information to scan for loop detection requires matching more and more local submaps. Therefore, the CPU usage of loop detection is also increasing, and the peak of the CPU usage of the

algorithm in this article shows a growing trend. The CPU usage of the algorithm in this article is slightly higher than that of the Hector algorithm in the figure. This is because the algorithm in this article incorporates odometer data, reducing computational complexity. Additionally, map resolution, radar frequency, and the number of particles in the filter also affect the CPU usage of the algorithm in this article. According to Table 2, the average CPU usage of the cartographer algorithm is the highest, at 21.2326%; Hector has the lowest average CPU usage because it uses less information and does not have a closed-loop detection process; the average CPU usage of the algorithm in this article is 17.8750%, which is 3.3576% lower than that of the Cartographer algorithm. The running pressure of the algorithm in this article is moderate.

From the above experiments, it can be seen that the method proposed in this paper can still ensure good running occupancy while increasing the computational complexity of the mapping process.

4.3 Degeneration Scene Mapping Experiment

Current laser SLAM algorithm based on graph optimisation has the karto algorithm [22] and the cartographer algorithm [23]. Karto algorithm is the first graph-based algorithm with loop detection, but the real-time performance is poor, and cartographer algorithm is improved on the basis of karto algorithm with higher accuracy and stronger real-time performance. In this experiment, the cartographer algorithm and the improved graph optimisation algorithm were used to compare the established grid map and analyse the mapping effect of the two algorithms.

First, compare the drawing effect of the two algorithms in more small channel scenes. Figure 6(a) for the map built for the cartographer algorithm, it can be seen that the overall accuracy of the built map is relatively high, and the scene features are complete, but there are obvious double shadows at the edge of the map, and the map boundary is relatively blurred. Figure 6(b) shows the map constructed by the improved SLAM algorithm in this paper. It can be seen that the ghosting problem at the edge of the map is significantly improved and the map boundary is clearer.

Figure 7 shows the mapping effects of two algorithms in a wide scene. It can be seen in the figure that there is no big difference between the overall appearance of the map established by the two algorithms, but it can still be seen that the map edge built by cartographer algorithm has more weights and the map boundary has more burrs. However, the map edge built by the improved SLAM algorithm is relatively smooth and clearer.

Compared with the pictures, the specific data can show more obvious differences. In this experiment, seven positions were selected as the test points, recording the true value of each position in the test point in the actual map and the test value of the constructed map. Table 3 shows the measurement data of the cartographer algorithm, the minimum absolute error is 0.8 and the maximum absolute error is 4.2; the minimum relative error is 0.456% and the maximum relative error is 1.194%. Table 4 is the

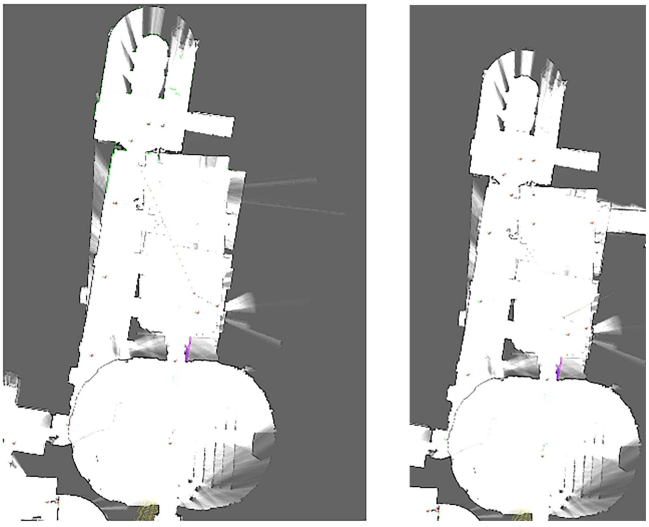

Figure 6. Maps constructed by both algorithms in narrow channel scenarios: (a) map constructed by the cartographer algorithm and (b) map constructed by the improved SLAM algorithm.

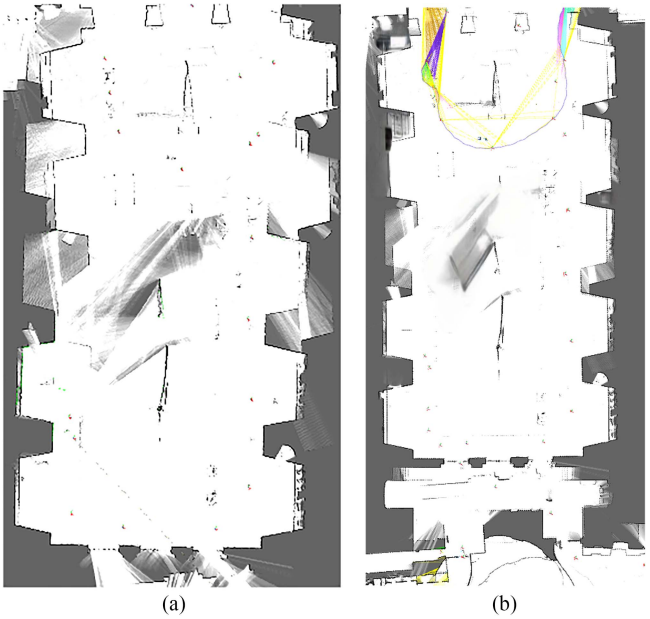

Figure 7. Map constructed by both algorithms in broad scenes: (a) Map constructed by the cartographer algorithm and (b) Map constructed by the improved SLAM algorithm in this paper.

measurement data of the improved SLAM algorithm, the minimum absolute error is 0.6 and the maximum absolute error is 3.3; the minimum relative error is 0.646% and the maximum relative error is 1.176%. It can be seen that the mapping error in a small range is not very different between the two algorithms, and even the error in individual measurement points cartographer is smaller. However, when the range and distance are constantly increasing, the error of cartographer is significantly larger than the improved algorithm in this paper.

Table 3
Map Measurement Data of the Cartographer Algorithm

Measurement Point Number	Ground Truth (cm)	Measured Value (cm)	Absolute Error (cm)	Fractional Error (%)
1	67	67.8	0.8	1.194
2	102	103.1	1.1	1.078
3	194	192.5	1.5	0.773
4	263	264.2	1.2	0.456
5	336	332.6	3.4	1.011
6	422	424.7	2.7	0.639
7	496	500.2	4.2	0.847
average	-	-	-	0.857

Table 4
The Improved SLAM Algorithm

Measurement Point Number	Ground Truth (cm)	Measured Value (cm)	Absolute Error (cm)	Fractional Error (%)
1	67	67.6	0.6	0.895
2	102	103.2	1.2	1.176
3	194	195.8	1.8	0.927
4	263	264.7	1.7	0.646
5	336	338.6	2.6	0.773
6	422	424.9	2.9	0.687
7	496	499.3	3.3	0.665
average	-	-	-	0.824

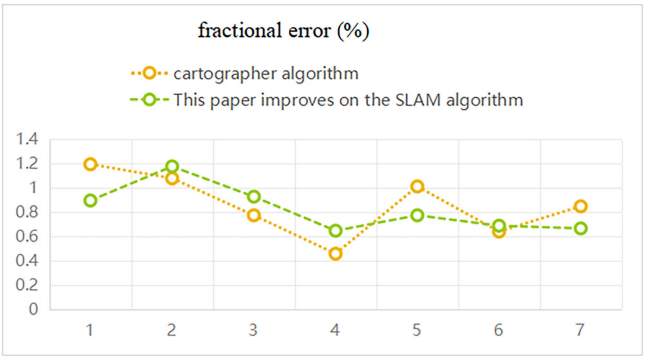

Figure 8. Relative error line plot of the cartographer algorithm and the improved algorithm in this paper.

The relative error line diagram of the map constructed by the two algorithms is shown in Fig. 8. It can be seen that with the increase of the measurement point distance, the improved SLAM algorithm can be better than the cartographer algorithm.

The experiment can be seen for small scene map two algorithm effect is very good, but when the scene becomes bigger, cartographer algorithm error will increase, increasing cumulative error, also can see the improved algorithm accuracy and higher stability, make the scene degradation problem to a certain extent, and the average relative error than cartographer algorithm reduced by about 3.8%.

5. Conclusion

This article investigates the differences in mapping accuracy between different laser SLAM algorithms in environments with large spatial scales and multiple similar scenes. This article introduces the IMU method, which can compensate for the impact of scene degradation. At the same time, the inertial navigation coordinate system and radar coordinate system are treated as a whole, and the method of frame matching error can solve the problem of large errors in pose estimation. The experimental results show that the cartographer algorithm can still achieve good mapping results in narrow scenes, but in wide scenes, it can be clearly seen that there are many heavy shadows on the map boundaries. In the future, we will focus more on practical applications and consider integrating UWB positioning methods to achieve high-precision, real-time robot trajectory estimation and map construction.

Acknowledgement

This work was supported by Jilin Province Science and Technology Development Plan Project (No.20210201049GX).

References

[1] F. Debbat and L. Adouane, Formation control and role allocation of autonomous mobile robots in unstructured environments, Mechatronic Systems and Control, *44*(2), 2016, 1–5.

[2] C. Weifeng, Z. Chengjun, S. Guangtao, W. Xiyang, L. Zhenxiong, X. Chonghui, H. Kai, SLAM Overview: From single sensor to heterogeneous fusion, *Remote Sensing, 14*(23), 2022, 6033.

[3] Y.K. Tee and Y.C. Han, Lidar-based 2D SLAM for mobile robot in an indoor environment: A review, in *Proc. International Conference on Green Energy, Computing and Sustainable Technology,* 2021, 1–7.

[4] C. Jianxian, G. Penggang, W. Yanxiong, and G. Zhitao, Mobile robot navigation using monocular visual-inertial fusion, *Mechatronic Systems And Control, 49*(1), 2021, 36–40.

[5] M.I. Idris, H. Hussin, S.A.M. Junos, W.H.M. Saad, M.N. Shah Zainudin, N.M. Yatim, A.S. Jaafar, M.R. Kamarudin1, and K.M. Saipullah, VSLAM analysis using various ORBSLAM parameters setting, *Przeglad Elektrotechniczny*, *19*(9), 2022, 40–45.

[6] J. Zhang, Laser-visual-inertial odometry and mapping with high robustness and low drift, Journal of Robotics & Machine Learning, 2018, 147.

[7] M. Palieri, B. Morrell, A. Thakur, K. Ebadi, J. Nash, A. Chatterjee, C. Kanellakis, L. Carlone, C. Guaragnella, and A.-A. Agha-mohammadi, , Corrections to LOCUS: A multi-sensor lidar-centric solution for high-precision odometry and 3D mapping in real-time, *IEEE Robotics and Automation Letters, 6*(2) 2021, 421–428.

[8] H. Lanyi, S. Zhiyong, and W. Huaiguang, A localization and mapping algorithm based on improved LVI-SAM for vehicles in field environments, *Sensors, 23*(7), 2023, 3744.

[9] X. Zuo, P. Geneva, W. Lee, Y. Liu, and G. Huang, LIC-fusion: LiDAR-inertial-camera odometry, 2019, *arXiv:1909.04102*.

[10] H. He, W. Xiang, H. Liu, and G. Huang, Autonomous navigation path planning of service robot based on multi-sensor fusion, Mechatronic Systems and Control, *52*(2), 2024, 76–82.

[11] G. Grisetti, R. Kümmerle, C. Stachniss, and W. Burgard, A tutorial on graph-based SLAM, *IEEE Intelligent Transportation Systems Magazine*, *2*(4), 2010, 31–43.

[12] W.C. Wei, B. Shirinzadeh, S. Esakkiappan, M. Ghafarian, and A. Al-Jodah, Orientation correction for hector SLAM at starting stage, in *Proc. of the 7th International Conference on Robot Intelligence Technology and Applications (RiTA), Daejeon,* 2019, 125–129.

[13] S. Xin and M. Huasong, Improved Cartographer algorithm based on velocity integral pose fusion, *Application of Laser*, *41*(5), 2021, 1063–1069.

[14] Z. Liang, Z. Liu, and J. Cao, Enhanced pose fusion of robotic vacuum cleaners cartographer algorithm and system implementation, *Journal of Software*, *31*(9), 2020, 2678–2690.

[15] J. Zhang and S. Singh, Low-drift and real-time lidar odometry and mapping, *Autonomous Robots*, *41*(2), 2017, 401–416.

[16] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, On-manifold preintegration for real-time visual: Inertial odometry, *IEEE Transactions on Robotics*, *33*(1), 2017, 1–21.

[17] E.B. Olson, Real-time correlative scan matching, in *Proc. of the IEEE International Conference on Robotics and Automation*, Kobe, 2009, 4387–4393

[18] A. Censi, An ICP variant using a point-to-line metric, in *Proc. of the IEEE International Conference on Robotics and Automation,* Pasadena, CA, 2008, 19–25.

[19] Z. Wang, L. Zhang, Y. Shen, and Y. Zhou, D-liom:Tightly-coupled direct lidar-inertial odometry and mapping, *IEEE Transactions on Multimedia*, *25*, 2022, 3905–3920.

[20] H. Li, L. Yang, and B. Zeng, Design and implementation of a mobile robot simulation experiment platform based on ROS, *Electronic Design Engineering*, *30*(14), 2022, 53–57.

[21] Q. Li and H. Zhu, Performance evaluation of 2D LiDAR SLAM algorithms in simulated orchard environments, *Computers and Electronics in Agriculture, 221*, 2024, 108994.

[22] K. Konolige, G. Grisetti, R. Kümmerle, W. Burgard, B. Limketkai, and R. Vincent, Efficient Sparse Pose Adjustment for 2D mapping, in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Taipei, 2010, 22–29.

[23] W. Hess, D. Kohler, H. Rapp, and D. Andor, Real-time loop closure in 2D LIDAR SLAM, in *Proc. of the IEEE International Conference on Robotics and Automation(ICRA)*, Stockholm, 2016, 1271–1278.

Biographies

Songyi Lu received the bachelor's degree in engineering from the Wenzheng College, Soochow University, in June 2022. He is currently pursuing the master's degree in electronic information engineering with Jilin Institute of Chemical Technology. His major research direction is laser SLAM.

Quande Yuan received the Ph.D. degree in computer application technology with Harbin institute of Technology. He is a Professor with Changchun Institute of Technology. His research interests include robots and artificial intelligence.

Wenhao Huang received the bachelor's degree in engineering from the Nanjing University of Technology in June 2021. He is currently pursuing the master's degree in electronic information engineering with Jilin Institute of Chemical Technology. His current research interest is robotic SLAM.

Guoyin Zhang received the master's degree in electronic information engineering from Jilin Institute of Chemical Technology in June 2024. Since July 2024, he has been an Engineer with China National Heavy Duty Truck Manufacturing Company. His research focuses mainly on robot positioning and navigation.

Yifan Liu received the master's degree in electronic information engineering from Jilin Institute of Chemical Technology in June 2024. Since July 2024, he has been an Engineer with Huali Integrated Circuit Manufacturing Co., Ltd. His research focuses mainly on visual SLAM and deep learning.