

SLAM Algorithm Based on Heterogeneous Sensor Data Fusion

Jianbin Dou

School of Artificial Intelligence
Anhui University
HeFei, China
wa22301065@stu.ahu.edu.cn

Chen Xu

School of Artificial Intelligence
Anhui University
HeFei, China
xuchen_98@163.com

Zhicheng Tang

School of Artificial Intelligence
Anhui University
HeFei, China
394005481@qq.com

Abstract—With the rapid development of artificial intelligence technology, simultaneous localization and mapping (SLAM), as a key technology, has attracted widespread attention. However, laser-based SLAM technology suffers from significant drift issues. A heterogeneous sensor data fusion algorithm aimed at improving the performance of SLAM is proposed in this paper. First, data from laser radar and the Inertial Measurement Unit (IMU) is collected. Then, the Cartographer algorithm itself employs the Extended Kalman Filter (EKF) method to fuse the data from both sources, ultimately constructing a 2D SLAM framework for environmental mapping. Laser radar data and IMU data from three different experimental scenarios are collected, and the mentioned algorithm is utilized to fuse the data for map construction. The error between the constructed map and the real map is compared, demonstrating that the heterogeneous sensor data fusion-based SLAM algorithm in this paper outperforms single lidar Cartographer. It overcame the issue of rotational drift and exhibited promising application potential.

Keywords—*SLAM, Cartographer, EKF, IMU, LIDAR*

I. INTRODUCTION

SLAM represents a pivotal technology for realizing intelligent robot navigation and comprehending environmental surroundings [1–4]. The origins of this concept can be traced back to 1988, when Smith, Self, and Cheeseman initially advocated for it [5]. Through continuous scholarly research, Dissanayake demonstrated for the first time the feasibility of SLAM technology [6]. Presently, the SLAM field primarily delineates into Visual SLAM [7–8] and Laser SLAM [9–12]. Visual SLAM boasts advantages such as applicability in expansive environmental contexts, incorporation of semantic information, and cost-effectiveness. However, it exhibits sensitivity to lighting conditions and necessitates a requisite level of textural diversity. Laser SLAM excels in terms of unmatched reliability and accuracy compared to Visual SLAM. Nonetheless, it contends with higher expenses and the limitation of environmental information. In sum, Visual SLAM finds its niche in cost-effective, texture-rich settings, while Laser SLAM thrives in high-precision tasks, particularly under low-light conditions. As of the present, Laser SLAM remains the dominant method for localization and navigation.

The key to mapping in Laser SLAM technology lies in scan matching techniques, with the establishment of this technique based on the Iterative Closest Point (ICP) algorithm [13] and its improved versions. Scan matching mainly involves two methods: scan-to-scan matching [14] and scan-to-map matching [15–16]. Scan-to-scan matching methods calculate the pose by aligning laser scans with one another, but errors can accumulate rapidly over time. Scan-to-map matching methods first generate a local result from laser scans

and then align that result with the overall map. This approach is more efficient and robust. While scan-map matching methods have seen significant improvements compared to the former, the issue of errors still persists. When the robot returns to the starting area, the poses of the starting and ending points are often inconsistent, leading to non-closure of the trajectory, and this issue results in loop closure problems [17]. Currently, the majority of solutions address this problem through loop closure detection.

According to mapping principles, 2d Laser SLAM can be primarily divided into three approaches: Gmapping, Hector, and Cartographer.

The Gmapping algorithm is a SLAM algorithm that estimates a robot's pose through particle filtering. It integrates laser radar data with the robot's motion model and updates particle weights through resampling. Subsequently, a map is constructed using interpolation and optimization methods based on the information from the weights of these particles [18]. The Gmapping algorithm offers advantages such as ease of implementation, cost-effectiveness, and minimal computational demands [19]. However, it also presents limitations, including high demands on application scenarios and odometry, as well as a lack of loop closure detection.

The Hector algorithm framework is founded on the Gauss-Newton method and is bifurcated into two primary phases: the front end and the back end [20]. In the front end phase, the Hector algorithm initiates by conducting laser scans and transforming data obtained from the laser radar into a grid map. By utilizing bilinear interpolation, a continuous probability grid map is generated, representing the likelihood of obstacles or open space within each grid of the environment. During the back end phase, the Hector algorithm aligns the most recent laser radar data with the existing map to estimate the current position and orientation of the robot [21]. The Hector algorithm presents two key advantages: it doesn't necessitate odometry and is applicable to non-flat terrain and aerial scenarios. However, it also exhibits certain limitations, including high demands on the radar, susceptibility to drift during rapid rotations, and the absence of loop closure detection.

In this research, the selected algorithm is Cartographer, a SLAM framework rooted in graph optimization that is introduced by Google in 2016 [26]. This framework is divided into two main components: the front end and the back end. The front end, the perception component, is responsible for the real-time collection and processing of sensor data, as well as the estimation of the robot's pose and position within the environment [22–23]. The back end, the decision-making component, is tasked with optimizing the robot's trajectory and path in the environment while simultaneously

constructing maps. Collaboration and information exchange between the front end and back end occur through sensor data. The positioning data provided by the front end (robot pose and position) serves as observation data for the back end, and the optimized trajectory path and map created by the back end are then fed back to the front end for more precise positioning estimation. This iterative process empowers Cartographer to continuously enhance its positioning accuracy and map quality during real-time operations [24-25]. Despite Cartographer's outperformance of similar SLAM algorithms, drift issues persist when exclusively employing laser radar mapping in environments with analogous features.

To tackle this concern, an scheme employing the Cartographer algorithm based on heterogeneous sensor data fusion is proposed in this paper. The primary works of this study encompass the following aspects:

1. In response to the drift issue observed in laser-based SLAM mapping, data from two distinct heterogeneous sensors are gathered, and this data is subjected to fusion through the Extended Kalman Filter (EKF) algorithm. This fusion significantly augments the information content and yields more precise pose estimation outcomes for the SLAM algorithm. This establishes a robust foundation for subsequent map construction.

2. After comparing the performance of commonly used SLAM algorithms, the Cartographer algorithm is selected based on the experimental scenarios to construct a 2D SLAM framework and build maps using the fused data.

This paper be divided into five sections. The first section introduce SLAM algorithms and provide an overview of the main content and structure of the article. The second section elucidate the basic principles and key technologies of data fusion. The third section provide a detailed explanation of the principles and implementation methods of the Cartographer algorithm. The fourth section cover experimental design, data collection, experimental results, and error analysis. Finally, the fifth section summarize the work presented in this paper and propose future research directions.

II. HETEROGENEOUS SENSOR DATA FUSION

The process of fusing laser radar data and Inertial Measurement Unit (IMU) data using Extended Kalman Filtering primarily encompasses four key stages: data synchronization, Extended Kalman Filter initialization, prediction step, and update step, as depicted in Figure 1.

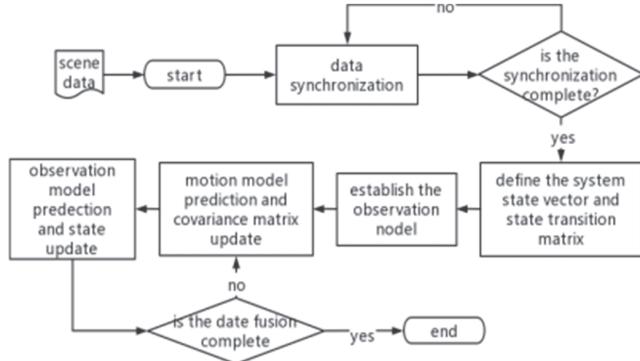


Fig. 1. Data Fusion Flowchart

A. Lidar Data

Lidar employs laser beams for the measurement of object distances and spatial positions. Common data types generated by lidar systems encompass point cloud data, intensity data, and reflectance data, which contain information such as distance, angle, and more.

$$[\text{Lidar Data}] = \begin{bmatrix} d \\ \theta_h \\ \theta_v \\ I \\ H \end{bmatrix}$$

$$\text{Distance: } d = c * \frac{t}{2} \quad (1)$$

$$\text{Horizontal angle: } \theta_h = \tan^{-1}\left(\frac{x_1 - x_2}{d}\right) \quad (2)$$

$$\text{Vertical angle: } \theta_v = \tan^{-1}\left(\frac{y_1 - y_2}{d}\right) \quad (3)$$

$$\text{Intensity: } I = \frac{i}{D^2} \quad (4)$$

Where c represents the speed of light, t stands for time, (x_1, y_1) denotes the target's coordinates, (x_2, y_2) represents the radar's coordinates, i signifies the reflection intensity, and D corresponds to the distance from the target to the radar. Point cloud data H constitutes a collection of three-dimensional coordinate points, typically represented as $\{(x_1, y_1, z_1), (x_2, y_2, z_2), \dots, (x_n, y_n, z_n)\}$. This data offers information about the position, geometry, and spatial relationships of objects within a three-dimensional space.

B. IMU Data

The data type of an IMU is determined by the data measured and output by its sensors. The commonly used IMU data types include the following three:

$$[\text{IMU Data}] = \begin{bmatrix} a \\ \omega \\ \text{Pose data} \end{bmatrix}$$

1) Acceleration Data a : Acceleration data is represented as $a = [a_x, a_y, a_z]$, can be calculated using Formula 5:

$$a = b_a + s_a * (a_r + n_a) \quad (5)$$

Where a denotes the output of the accelerometer, b_a signifies the bias of the accelerometer, s_a represents the scale factor of the accelerometer, a_r stands for the actual acceleration value, and n_a represents the noise of the accelerometer.

2) Angular velocity data ω : Angular velocity data is represented as $\omega = [\omega_x, \omega_y, \omega_z]$, can be calculated using Formula 6:

$$\omega = b_\omega + s_\omega * (\omega_r + n_\omega) \quad (6)$$

Where ω represents the output of the angular velocity sensor, b_ω denotes the bias of the angular velocity sensor, s_ω signifies the scale factor of the angular velocity sensor, ω_r stands for the actual angular velocity value, and n_ω represents the noise associated with the angular velocity sensor.

3) Pose data: The accelerometer and gyroscope data from the Inertial Measurement Unit (IMU) can be utilized to determine the pose of an object. This is achieved by integrating the acceleration and angular velocity data over a specific time period. The integration formula is presented in

equation 7, where Δt represents the time interval between two consecutive measurements.

$$\Delta x = \int_0^{\Delta t} x(t) dt \quad (7)$$

These data provide crucial input for applications that require precise attitude and motion information.

C. Extended Kalman Filtering

The Extended Kalman Filter (EKF)[27] possesses the capability to fuse IMU and radar data within nonlinear systems, yielding the most accurate estimate of the robot's state. It achieves this by approximating and addressing nonlinear issues through the linearization of the nonlinear system. The primary stages of EKF encompass initialization, prediction, and update. In the prediction phase, IMU data and the system model are employed to anticipate the state and covariance at the subsequent time instant. The IMU supplies data related to acceleration and angular velocity, enabling the estimation of the robot's motion state. The system model characterizes the robot's motion patterns and alterations in pose. The update phase involves the correction of the predicted state and covariance using radar data and the Kalman gain. Radar provides measurement data, including distance and angle, which enhances the precision of state estimation. The iterative process of EKF prediction and update allows the continuous enhancement of state estimation accuracy while accommodating various sensor data and the system's nonlinear characteristics. This proves invaluable for applications such as real-time map construction and navigation.

EKF exploits the local linear properties of nonlinear functions to approximate them with linear models. The discrete-time nonlinear system is described by equations 8 and 9.

$$x_{k+1} = f(x_k, u_k) + w_k \quad (8)$$

$$z_k = h(x_k) + v_k \quad (9)$$

Where $k \in N$ serves as the temporal index, $x_k \in R^n$ denotes the system state vector at time k, encompassing the fused data derived from both laser radar and IMU. u_k pertains to the control input vector, encompassing external inputs or control data originating from the IMU, such as linear acceleration and angular velocity, for instance. w_k represents the process noise, signifying inherent noise within the system model, typically associated with errors in IMU data. $z_k \in R^m$ stands for the observation vector of the system at time "k," generally inclusive of measurement data from the laser radar, such as target distance and angle information. Finally, v_k represents the observation noise, reflecting the inherent noise within laser radar measurements.

The state estimate, denoted as \hat{x}_k , encompasses information such as position, velocity, and IMU attitude, and can be expressed using the following equation:

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H_k\hat{x}_k^-) \quad (10)$$

Where \hat{x}_k^- denotes the predicted estimation, intrinsically linked to the acceleration and angular velocity data derived from the IMU. H_k signifies the observation matrix, while K_k designates the gain matrix.

The nonlinear functions $f(*)$ and $h(*)$ are expanded around the filter value \hat{x}_k using a first-order :

$$x_{k+1} = f(\hat{x}_k, u_k) + \frac{\partial f}{\partial \hat{x}_k}(x_k - \hat{x}_k) + w_k \quad (11)$$

$$z_k = h(\hat{x}_k) + \frac{\partial h}{\partial \hat{x}_k}(x_k - \hat{x}_k) + v_k \quad (12)$$

Let $\frac{\partial f}{\partial \hat{x}_k} = F_k$ (system state transition matrix), $\frac{\partial h}{\partial \hat{x}_k} = H_k$. By substituting equations 11 and 12 into the linear Kalman filtering recursion process, can derive the extended Kalman filter.

Compared to the basic equations of linear Kalman filtering, in the system equations of extended Kalman filtering after linearization, the state transition matrix F_k and observation matrix H_k can be replaced by the Jacobian matrices of f and h . Assuming the state vector has n dimensions, $x = [x_1 \ x_2 \ \dots \ x_n]^T$, the method for calculating the corresponding Jacobian matrices is as follows:

$$F_k = \frac{\partial f}{\partial \hat{x}_k} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \dots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \dots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \dots & \frac{\partial f_n}{\partial x_n} \end{bmatrix} \quad (13)$$

$$H_k = \frac{\partial h}{\partial \hat{x}_k} = \begin{bmatrix} \frac{\partial h_1}{\partial x_1} & \frac{\partial h_1}{\partial x_2} & \dots & \frac{\partial h_1}{\partial x_n} \\ \frac{\partial h_2}{\partial x_1} & \frac{\partial h_2}{\partial x_2} & \dots & \frac{\partial h_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial h_n}{\partial x_1} & \frac{\partial h_n}{\partial x_2} & \dots & \frac{\partial h_n}{\partial x_n} \end{bmatrix} \quad (14)$$

III. PREPARE YOUR PAPER BEFORE STYLING

Cartographer is an excellent SLAM framework developed and maintained by Google. It can achieve localization and mapping in indoor or outdoor environments using various sensor data. The simplified workflow of Cartographer is shown in Figure 2.

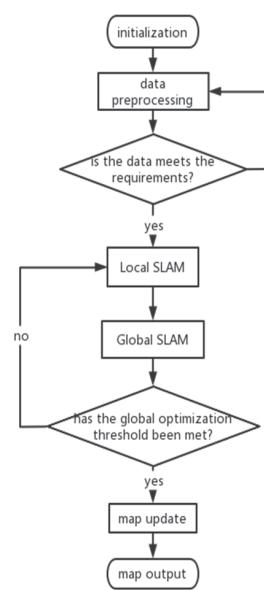


Fig. 2. Cartographer Workflow Diagram

Cartographer consists of two steps: local SLAM and global SLAM. In the local SLAM step, each frame's scan data is transformed into multiple submaps. In the global SLAM step, optimization is performed to integrate these submaps into a complete map.

A. Scans

The starting point is defined as the "origin", and the endpoint is defined as the "scan point". The combination of the starting point and the endpoint forms a scan. Assuming that the origin of the Submap's coordinate system is $0 \in R^2$, and each radar scan point is represented as p , the data from this frame of radar can be represented as the set H:

$$H = \{h_k\}_{k=1, \dots, K}, h_k \in R^2 \quad (15)$$

Pose $\xi = (\xi_x, \xi_y, \xi_\theta)$ consists of a translation (x, y) and a rotation ξ_θ . The coordinate transformation formula for the transformation matrix T_ξ used to insert scans into submaps is as follows:

$$T_\xi p = \underbrace{\begin{pmatrix} \cos \xi_\theta & -\sin \xi_\theta \\ \sin \xi_\theta & \cos \xi_\theta \end{pmatrix}}_{R_\xi} p + \begin{pmatrix} \xi_x \\ \xi_y \end{pmatrix} \quad (16)$$

B. Submaps

A "submap" is a probabilistic grid map composed of consecutive frames of scans, which can be divided into multiple probability grids, as shown in Figure 3. The 'r' in the figure represents the grid map's resolution, and each grid cell's value indicates whether there is data in that grid. Grid probabilities less than p_{min} indicate that the point is empty, probabilities between p_{min} and p_{max} represent an unknown state, and probabilities greater than p_{max} indicate the presence of data.

When inserting each scan frame into a submap, it is imperative to ascertain the sets of hits and misses. In Figure 4, the shaded regions marked with "×" represent hit points, the shaded regions without "×" represent miss points. Each hit point is initialized with a value of p_{hit} and miss points are initialized with a value of p_{miss} . For the grids that haven't been observed yet, a custom p value is assigned (usually set to 0.5), indicating that the grid is in an unknown state. For grids that have been observed, odds are defined as the probability of hits divided by the probability of misses:

$$odds(p) = \frac{p}{1-p} \quad (17)$$

Each grid's probability M is updated by the formula 18:

$$M_{new}(x) = clamp\left(odds^{-1}\left(odds\left(M_{old}(x)\right) \cdot odds\left(p_{hit}\right) \right) \right) \quad (18)$$

Where "clamp" refers to a function used to constrain the value within a predefined range.

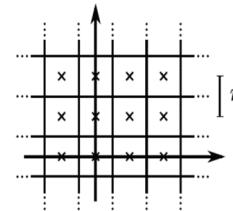


Fig. 3. Submap

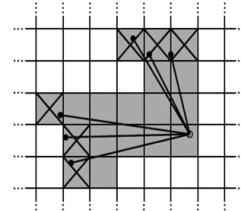


Fig. 4. Construction of submaps

C. Ceres scan matching

Before inserting each scan into the submap, a "Ceres scan matching" is conducted to refine the pose of every scan. By the scan matching optimizer, an optimal pose is sought to maximize the probability of matching the scan to the submap. Mapping optimization is treated as a non-linear least squares problem and is addressed using equation 19.

$$\operatorname{argmin}_\xi \sum_{k=1}^K \left(1 - M_{smooth}(T_\xi h_k) \right)^2 \quad (19)$$

Where T_ξ represents the coordinate transformation matrix, and M_{smooth} implements $R^2 \rightarrow R$.

D. Loop closure

The optimization problem for loop closure detection in Cartographer is a nonlinear least squares problem:

$$\operatorname{argmin}_{\Xi^m, \Xi^s} \frac{1}{2} \sum_{ij} \rho\left(E^2(\xi_i^m, \xi_j^s; \Sigma_{ij}, \xi_{ij})\right) \quad (20)$$

$$\Xi^m = \{\xi_i^m\}_{i=1, \dots, m} \quad (21)$$

$$\Xi^s = \{\xi_j^s\}_{j=1, \dots, n} \quad (22)$$

Where Ξ^m denotes the pose of the submap, Ξ^s signifies the pose of the scan, ξ_{ij} characterizes the pose of the j-th scan within the i-th submap, and Σ_{ij} corresponds to the covariance matrix.

The formula for calculating the residual E is as follows:

$$E^2(\xi_i^m, \xi_j^s; \Sigma_{ij}, \xi_{ij}) = e(\xi_i^m, \xi_j^s; \xi_{ij})^T \Sigma_{ij}^{-1} e(\xi_i^m, \xi_j^s; \xi_{ij}) \quad (23)$$

$$e(\xi_i^m, \xi_j^s; \xi_{ij}) = \xi_{ij} - \begin{pmatrix} R_{\xi_i^m}^{-1} (t_{\xi_i^m} - t_{\xi_j^s}) \\ \xi_{i;\theta}^m - \xi_{j;\theta}^s \end{pmatrix} \quad (24)$$

IV. EXPERIMENTS AND RESULTS ANALYSIS

To evaluate the mapping performance of the Cartographer algorithm based on heterogeneous sensor data fusion, an experimental platform is established, and environmental data are gathered. The experimental platform is configured with Ubuntu 18.04 and ROS 1 system, enabling the direct implementation of the Cartographer algorithm for map construction. Data packets are recorded using the "rosbag" method, which are subsequently processed for map construction. This study conducts experiments in three distinct scenarios, including indoor, corridor, and outdoor environments.

A. Setting up the experimental platform

The experimental mobile robot is equipped with an outdoor heavy-duty operation drive and control integrated

chassis. The system comprises the main body with a four-wheel independent suspension chassis, a control system, a multi-sensor system, a power system, and a power management system. The controller is part of the Intel Core i7 series, featuring an I7 processor and an Intel HD Graphics 5500 graphics card. The laser radar is mounted at the front of the entire vehicle and is of the model "RPLiDAR A2." The IMU is installed directly above the radar and belongs to the WHEELTEC N100 series. It includes a three-axis gyroscope, a three-axis accelerometer, a three-axis magnetometer, and a temperature sensor, belonging to the nine-axis attitude sensor. The complete platform is depicted in Figure 5.

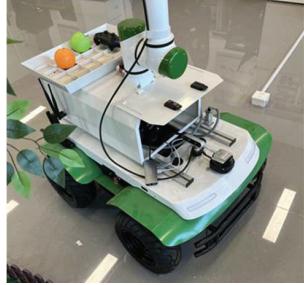


Fig. 5. Experimental Robot

The chassis parameters are presented in Table 1.

TABLE I. CHASSIS PARAMETERS

	components	main technical performance indicators
vehicle body	drive wheels	4
	gear ratio	40
	wheel diameter	305mm
	length * width * height	917*710*410mm
	ground clearance	110mm
	wheelbase	500mm
	front/rear track	555mm
	drive mode	four-wheel independent drive
	steering	four-wheel differential steering
	maximum climbing ability	$\leq 15^\circ$
	empty vehicle maximum speed (m/s)	≤ 1.0
	minimum turning radius	capable of making zero-radius turns

The specific parameters of the laser radar can be found in Table 2.

TABLE II. PRLIDAR A2 LASER RADAR PARAMETERS

main technical performance indicators	value or range
measurement radius	0.2m - 12m
sampling frequency	8K
scanning frequency	10Hz (5Hz-15Hz)
angular resolution	0.45°
pitch angle	$\pm 1.5^\circ$
scanning range	360°
distance resolution	$\leq 1\%$ of the actual distance (for distance measurements $\leq 12m$)
	$\leq 2\%$ of the actual distance (for distance measurements between 12m and 16m)
distance accuracy	1% of the actual distance (for distances $\leq 3m$)
	2% of the actual distance (for distances between 3m and 5m)
	2.5% of the actual distance (for distances between 5m and 16m)

The specific parameters of the IMU can be found in Table 3.

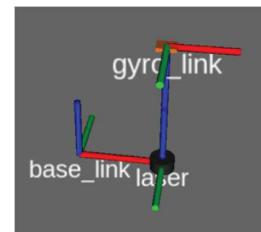
TABLE III. WHEELTEC N100 PARAMETERS

main technical performance indicators	IMU		
	Accelerometer	Gyroscope	Magnetometer
measurement range	$\pm 16 g$	$\pm 2000^\circ/s$	$\pm 4900 \mu T$
zero bias stability	<0.04 mg	<10°/hr (5°/hr typ.)	/
linearity	<0.1 % FS	<0.1 % FS	<0.1 %
noise density	75ug/ \sqrt{Hz}	0.0028/s/ \sqrt{Hz}	140 μ Gauss/ \sqrt{Hz}
bandwidth	260 Hz	256 Hz	200 Hz
orthogonality error	$\pm 0.05^\circ$	$\pm 0.05^\circ$	$\pm 0.05^\circ$
resolution	<0.5 mg	<0.02°/s	1.5 Milligauss

The installation configuration of the radar and IMU is depicted in Figure 6 (a). The radar is positioned at the front of the vehicle, while the IMU is directly mounted above the radar. Figure 6 (b) illustrates the interrelation between the coordinate systems of these components. In this illustration, "base_link" signifies the vehicle's coordinate system, with its origin located at the vehicle's center. "gyro_link" designates the IMU's coordinate system, and "laser" signifies the radar's coordinate system. The red, green, and blue axes correspond to the x-axis, y-axis, and z-axis of their respective coordinate systems.



(a) Physical attributes



(b) Coordinate system

Fig. 6. Experimental Robot

B. Corridor experiment

First, environmental data of the corridor scene is collected using the equipped sensor devices, and a map is generated using the Cartographer algorithm. Subsequently, several positions within the corridor, as depicted in Figure 7, are chosen, and a measuring tool is employed to obtain the real-world distances. Next, software is used to compute the pixel distances between the corresponding positions on the map and to convert these pixel distances to actual distances with the assistance of scaling factors. Finally, the absolute error and relative error between the actual distance and the mapped distance are determined. An analysis of this error data allowed for the evaluation of the mapping performance of the Cartographer algorithm based on the fusion of heterogeneous sensor data. The constructed 2D corridor map is depicted in Figure 8.

The experimental scene is shown in Figure 7.



Fig. 7. Corridor

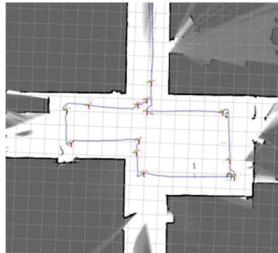


Fig. 8. Corridor (this paper's scheme)

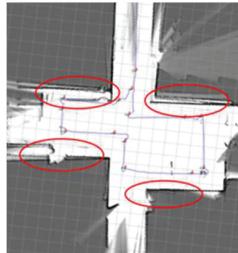


Fig. 9. Corridor (Single LiDAR Cartographer)

To validate the mapping performance of the scheme presented in this paper in a corridor scenario, a comparison is made with the mapping results of the single lidar Cartographer using only the lidar data recorded in the data bag. The resulting map is shown in Figure 9.

After observing Figure 9, it becomes apparent that the mapping quality of the single lidar Cartographer algorithm is unsatisfactory, marked by significant overlap issues within the circled area. This is primarily attributed to the vehicle executing turns at these locations during data recording, resulted the issue of rotation drift. In contrast, in Figure 8, this issue has been significantly improved. Subsequently, distance measurements taken for the portion marked in Figure 10 for a more comprehensive analysis of the mapping outcomes.

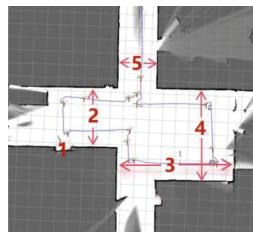


Fig. 10. Marked Area in the Corridor

In the corridor scene, measurements are taken at the five marked positions shown in Figure 10. The data obtained is presented in Tables 4 and 5, as well as Figures 11 and 12.

TABLE IV. DATA FROM SINGLE LIDAR CARTOGRAPHER

position	actual measurement value/m	figure measurement value/m	absolute error/m	relative error/%
1	1.22	1.09	0.12	10.7
2	4.41	5.18	0.77	17.4
3	8.78	9.14	0.36	4.1
4	7.01	6.78	0.23	3.2
5	2.94	3.39	0.45	15.3

TABLE V. DATA FROM THIS PAPER'S SCHEME

position	actual measurement value/m	figure measurement value/m	absolute error/m	relative error/%
1	1.22	1.25	0.03	2.5
2	4.41	4.60	0.19	4.3

3	8.78	8.98	0.20	2.3
4	7.01	7.09	0.08	1.1
5	2.94	3.01	0.07	2.4

Analyzed the data presented in Table 4, it becomes evident that in the corridor scenario, the relative errors in the map generated by the single LiDAR Cartographer consistently exceed 3%, with a majority of them surpassing 10%. In contrast, Table 5 illustrates that the map produced by the scheme outlined in this paper exhibits relative errors mostly below 3%, with the error at location 2 being the highest at 4.3%.

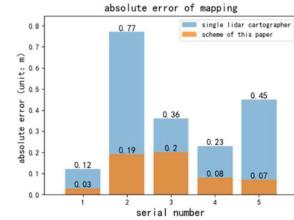


Fig. 11. Absolute error of mapping

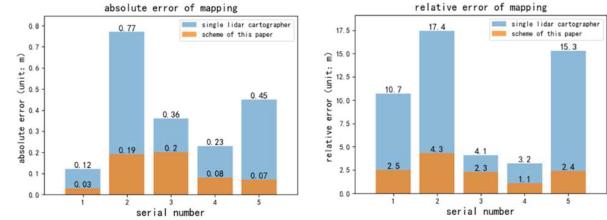


Fig. 12. Relative error of mapping

After examination of Figures 11 and 12, it becomes evident that, in the corridor scene, the absolute and relative errors of the scheme presented in this paper are significantly lower than those of the single lidar Cartographer. Specifically, at each marked location, the scheme proposed in this paper exhibited relative errors over 2 percentage points lower than those of the single lidar Cartographer. Especially at positions 1, 2, and 5, due to rotation-induced drift, the difference in relative error between the two becomes more evident. At these three marked locations, the scheme presented in this paper reduced the relative error by 8.2%, 13.1%, and 12.9%, respectively, in comparison to the single lidar Cartographer within the lobby corridor scene. This outcome suggests that the scheme presented in this paper effectively reduces the impact of drift due to rotation on mapping accuracy. Through comparative analysis, it can be concluded that in the corridor scenario, the mapping scheme proposed in this paper exhibited higher accuracy compared to the single lidar Cartographer.

C. Indoor experiments

The indoor environment is more complex compared to corridors, with more obstacles and details. The presence of these obstacles makes indoor mapping more challenging. At the same time, indoor mapping demands higher precision and detail. The following indoor experiment is conducted, and the experimental process is similar to the corridor experiment. The constructed 2D indoor map is shown in Figure 14.

The experimental scene is shown in Figure 13.

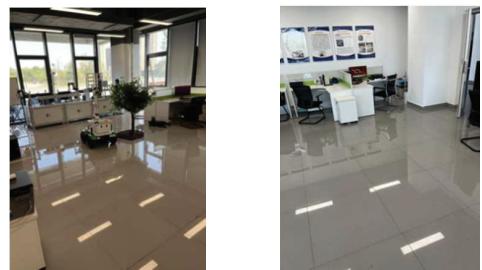


Fig. 13. Indoor scene

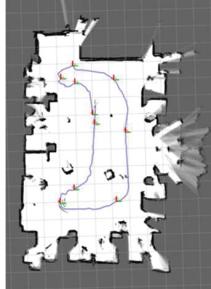


Fig. 14. Indoor (this paper's scheme) Fig. 15. Indoor (Single LiDAR Cartographer)



To validate the mapping performance of the scheme presented in this paper in an indoor scenario, a comparison is made with the mapping results of the single lidar Cartographer using only the lidar data recorded in the data bag. The resulting map is shown in Figure 15.

By observing Figure 15, it becomes evident that the single lidar Cartographer algorithm outperforms the corridor mapping in indoor environments. However, issues of overlap persist within the circled area. This occurrence is attributed to relatively more obstacles within these regions, resulting in signal interference and confusion during radar signal reflection and propagation, consequently impacting map construction accuracy. Nevertheless, upon observing the mapping results of this paper's scheme in Figure 14, a noticeable improvement in addressing the aforementioned overlap issue is observed. Subsequently, distances at the marked locations in Figure 16 were measured to conduct a more in-depth analysis of the mapping results.

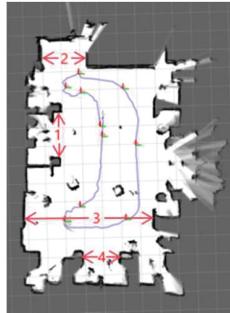


Fig. 16. Marked Area Indoors

In the indoor environment, measurements are taken at four locations marked in Figure 16, and the obtained data is shown in Table 6, Table 7, Figure 17, and Figure 18.

TABLE VI. DATA FROM SINGLE LIDAR CARTOGRAPHER

position	actual measurement value/m	figure measurement value/m	absolute error/m	relative error/%
1	2.38	2.27	0.11	4.6
2	2.43	2.51	0.08	3.3
3	6.88	6.78	0.10	1.5
4	1.99	1.92	0.07	3.5

TABLE VII. DATA FROM THIS PAPER'S SCHEME

position	actual measurement value/m	figure measurement value/m	absolute error/m	relative error/%
1	2.38	2.41	0.03	1.3
2	2.43	2.40	0.03	1.2
3	6.88	6.82	0.06	0.9
4	1.99	1.97	0.02	1.0

Based on the data in Table 6, it can be observed that in the indoor environment, the relative errors of the maps generated by the single lidar Cartographer are mostly above 3%, with the exception of a 1.5% error at position 3. Upon examination of Table 7, it becomes apparent that the relative errors of the maps generated by the scheme proposed in this paper consistently approximate 1%, with the maximum error registering at a mere 1.3%.

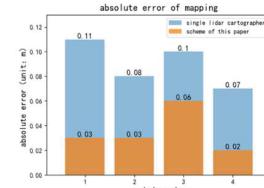


Fig. 17. Absolute error of mapping

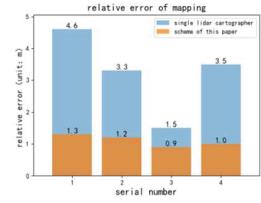


Fig. 18. Relative error of mapping

After examining Figures 17 and 18, it is evident that in the indoor environment, the scheme presented in this paper demonstrated reduced absolute and relative errors compared to the single lidar Cartographer. In the four marked locations selected, this paper's scheme showcases a relative error decrease of 3.3%, 2.1%, 0.6%, and 2.5% when compared to the single lidar Cartographer. While the reduction in error magnitude is not as significant as observed in the corridor scenario, it still represented a noteworthy enhancement. Hence, it could be inferred that within indoor scenarios, the map construction scheme proposed in this paper provides higher accuracy compared to the single lidar Cartographer.

D. Outdoor experiments

The scale of outdoor and indoor environments differs significantly, and outdoor mapping is more susceptible to influences. Subsequent experiments are conducted to validate whether the proposed scheme could enhance mapping in outdoor scenarios. The 2D outdoor map constructed is depicted in Figure 20.

The experimental scene is shown in Figure 19.



Fig. 19. Outdoor scene

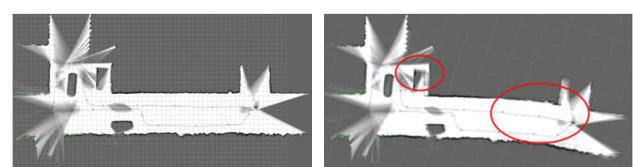


Fig. 20. Outdoor (this paper's scheme)

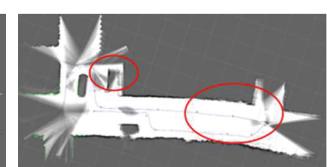


Fig. 21. Outdoor (Single LiDAR Cartographer)

To validate the mapping performance of the scheme presented in this paper in an outdoor scenario, a comparison is made with the mapping results of the single lidar Cartographer using only the lidar data recorded in the data bag. The resulting map is shown in Figure 21.

In outdoor environments, the utilization of GPS data is a common practice. However, this paper do not utilize GPS data, outdoor scenarios are not the primary focus of this study. As a result, the steps for measuring outdoor scenario data are omitted here. It is evident from the portion highlighted in Figure 21 that due to the long distances and limited feature information, the single lidar Cartographer exhibits curvature in the constructed map. In contrast, as seen in Figure 20, the scheme presented in this paper has effectively mitigated this issue. This result indicates that in an outdoor environment, the map construction scheme proposed in this paper offers higher accuracy compared to the single lidar Cartographer.

CONCLUSION

This paper focuses on the study of SLAM and the fusion of heterogeneous sensor data, proposing a SLAM algorithm based on heterogeneous sensor data fusion to address drift issues in single lidar Cartographer mapping. Initially, laser radar data and IMU data are collected. Through the Cartographer algorithm, these two data types are integrated, which construct a 2D SLAM framework for mapping the environment. Unlike conventional single lidar Cartographer algorithms, the scheme adopted in this paper maximized the advantages offered by laser radar and IMU sensors. To verify the efficacy of the proposed solution, mapping experiments are conducted, featuring three distinct experimental scenarios: extended corridors, indoor environments, and outdoor environments. A comparative analysis of mapping performance between the proposed scheme and single lidar Cartographer is undertaken for each scenario. The measurement data from the maps are visually presented in tabular and bar chart formats. The experimental outcomes validate that the map generated by the proposed scheme effectively eliminates drift issues and surpasses the accuracy of maps constructed using single lidar Cartographer. Furthermore, this paper's scheme exhibits superior practicality, delivering commendable mapping results across various scenarios, indicating promising application potential. The method described in this paper has shown strong performance. However, it is important to note that the experiments conducted in this study utilized a single-line radar for data collection. If a multi-line radar were to be employed, it could capture a more comprehensive point cloud dataset, leading to improved map quality. Additionally, the integration of a wider range of heterogeneous data for fusion has the potential to further enhance mapping performance.

REFERENCES

- [1] A. Macario Barros, M. Michel, Y. Moline, G. Corre, and F. Carrel, "A comprehensive survey of visual slam algorithms," *Robotics*, vol. 11, no. 1, p. 24, 2022.
- [2] H. Taheri and Z. C. Xia, "Slam; definition and evolution," *Engineering Applications of Artificial Intelligence*, vol. 97, p. 104032, 2021.
- [3] Y. Fan, Q. Zhang, Y. Tang, S. Liu, and H. Han, "Blitz-slam: A semantic slam in dynamic environments," *Pattern Recognition*, vol. 121, p. 108225, 2022.
- [4] Y. Liu and J. Miura, "Rds-slam: Real-time dynamic slam using semantic segmentation methods," *Ieee Access*, vol. 9, pp. 23 772–23 785, 2021.
- [5] R. C. Smith and P. Cheeseman, "On the representation and estimation of spatial uncertainty," *The international journal of Robotics Research*, vol. 5, no. 4, pp. 56–68, 1986.
- [6] M. G. Dissanayake, P. Newman, S. Clark, H. F. Durrant-Whyte, and M. Csorba, "A solution to the simultaneous localization and map building (slam) problem," *IEEE Transactions on robotics and automation*, vol. 17, no. 3, pp. 229–241, 2001.
- [7] B. Fang, G. Mei, X. Yuan, L. Wang, Z. Wang, and J. Wang, "Visual slam for robot navigation in healthcare facility," *Pattern Recognition*, vol. 113, p. 107822, 2021.
- [8] I. A. Kazerouni, L. Fitzgerald, G. Dooly, and D. Toal, "A survey of state-of-the-art on visual slam," *Expert Systems with Applications*, vol. 205, p. 117734, 2022.
- [9] L. Huang, "Review on lidar-based slam techniques," in *2021 International Conference on Signal Processing and Machine Learning (CONF-SPML)*. IEEE, 2021, pp. 163–168.
- [10] J. Zhao, L. Zhao, S. Huang, and Y. Wang, "2d laser slam with general features represented by implicit functions," *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 4329–4336, 2020.
- [11] M. Wu, C. Cheng, and H. Shang, "2d lidar slam based on gauss-newton," in *2021 International Conference on Networking Systems of AI (INSAI)*. IEEE, 2021, pp. 90–94.
- [12] Z. Ren, L. Wang, and L. Bi, "Robust gicp-based 3d lidar slam for underground mining environment," *Sensors*, vol. 19, no. 13, p. 2915, 2019.
- [13] S. Rusinkiewicz, "A symmetric objective function for icp," *ACM Transactions on Graphics (TOG)*, vol. 38, no. 4, pp. 1–7, 2019.
- [14] E. Olson, "M3rsm: Many-to-many multi-resolution scan matching," in *2015 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2015, pp. 5815–5821.
- [15] J. Behley and C. Stachniss, "Efficient surfel-based slam using 3d laser range data in urban environments," in *Robotics: Science and Systems*, vol. 2018, 2018, p. 59.
- [16] K. Ji, H. Chen, H. Di, J. Gong, G. Xiong, J. Qi, and T. Yi, "Cpfslam: A robust simultaneous localization and mapping based on lidar in off-road environment," in *2018 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2018, pp. 650–655.
- [17] Y. C. Tong, W. Cui, and H. Y. Fu, "Slam closed-loop detection method of 2d laser fusion vision in a larger space," in *Journal of Physics: Conference Series*, vol. 1948, no. 1. IOP Publishing, 2021, p. 012036.
- [18] M. Liao, D. Wang, and H. Yang, "Deploy indoor 2d laser slam on a raspberry pi-based mobile robot," in *2019 11th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)*, vol. 2. IEEE, 2019, pp. 7–10.
- [19] H. Ye, G. Chen, W. Chen, L. He, Y. Guan, and H. Zhang, "Mapping while following: 2d lidar slam in indoor dynamic environments with a person tracker," in *2021 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE, 2021, pp. 826–832.
- [20] S. Nagla, "2d hector slam of indoor mobile robot using 2d lidar," in *2020 International Conference on Power, Energy, Control and Transmission Systems (ICPECTS)*. IEEE, 2020, pp. 1–4.
- [21] W. Weichen, B. Shirinzadeh, M. Ghafarian, S. Esakkiaappan, and T. Shen, "Hector slam with icp trajectory matching," in *2020 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*. IEEE, 2020, pp. 1971–1976.
- [22] A. Dwijotomo, M. A. A. Rahman, M. H. M. Ariff, and H. Zamzuri, "Cartographer local slam optimization using multistage distance scan scheduler," in *Proceedings of the 6th International Conference and Exhibition on Sustainable Energy and Advanced Materials: ICE-SEAM 2019*, 16–17 October 2019, Surakarta, Indonesia. Springer, 2020, pp. 201–213.
- [23] W. Hess, D. Kohler, H. Rapp, and D. Andor, "Real-time loop closure in 2d lidar slam," in *2016 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2016, pp. 1271–1278.
- [24] L. Weerakoon, G. S. Herr, J. Blunt, M. Yu, and N. Chopra, "Cartographer glass: 2d graph slam framework using lidar for glass environments," *arXiv preprint arXiv:2212.08633*, 2022.
- [25] W. Ali, L. Sheng, and W. Ahmed, "Robot operating system-based slam for a gazebo-simulated turtlebot2 in 2d indoor environment with cartographer algorithm," *International Journal of Mechanical and Mechatronics Engineering*, vol. 15, no. 3, pp. 149–157, 2021.
- [26] I. Aloise and G. Grisetti, "Chordal based error function for 3-d pose-graph optimization," *IEEE Robotics and Automation Letters*, vol. 5, no. 1, pp. 274–281, 2019.
- [27] B. Zheng, Z. Zhang et al., "An improved ekf-slam for mars surface exploration," *International Journal of Aerospace Engineering*, vol. 2019, 2019.