Robotica (2025), 1–14
doi:10.1017/S0263574725000463

RESEARCH ARTICLE

Tightly coupled SLAM system for indoor complex scenes

Chen Da[1], Zailiang Chen[1,2], Tianlin Song[1] and Yaping Lu[1]

[1]Applied Technology College, Soochow University, Suzhou, China
[2]School of Mechanical and Electric Engineering, Soochow University, Suzhou, China
Corresponding author: Chen Da; E-mail: 1425513566@qq.com

Received: 15 August 2024; **Revised:** 30 December 2024; **Accepted:** 26 February 2025

Keywords: pre-integration; cartographer; hybrid filter; tightly-coupled

Abstract

Cartographer is an algorithm that was open sourced by Google in 2016 and adapted to multiple sensors. To address issues of the original algorithm, such as the negative impact of outlier point cloud on the scan matching, and low accuracy of position fusion. This paper preprocesses the sensor data and presents **HT-Carto**, an improved **h**ybrid point-cloud filtering system, and a **t**ightly coupled LiDAR/IMU framework based on **Carto**grapher's front-end. The inertial measurement unit (IMU) provides initial values for the point cloud, and the IMU pre-integration combines the scan-matched pose to construct the factors, which are added as constraints to the factor graph. The result is used to update the current pose and work as odometer residuals at the back-end. The optimization of the selected strategy during point cloud preprocessing, PassThrough, and RadiusOutlierRemoval are combined to ensure quality. An actual vehicle is used in complex indoor environment to verify the stability and robustness of HT-Carto. Compared to the Cartographer, Karto, Hector, and GMapping, HT-Carto demonstrates better localization and mapping, it can obtain a more precise trajectory.

1. Introduction

Safety is a priority owing to the high frequency of laboratory accidents. A low-cost and versatile robot is required to replace daily manual inspections. Automated guided vehicles (AGV) are widely used in complex environments, such as large warehouse, disaster area rescue, and mine exploration [1, 2]. High-precision mapping and localization are important prerequisites for patrols. Simultaneous localization and mapping (SLAM) is the key technology to this [3].

Scholars often study laser and visual SLAM, visual SLAM contains semantic information but is susceptible to environmental changes, it is an angle-measurement sensor and cannot obtain distance information directly. It is necessary to reconstruct the feature distance from multiple views, which is unsuitable for applications in embedded platforms owing to the large computation [4, 5]. In contrast, light detection and ranging (LiDAR) is more accurate and resistant to interference, but prones to distortion when rotating, leading to a map offset [6]. The prices of 3D LiDAR, vision sensors, and 2D LiDAR show a downward trend. This study focuses on the localization and mapping of 2D LiDAR in indoor environment.

A complete SLAM application cannot be supported using a single sensor. The use of multi-sensor fusion can enhance the positioning accuracy of the AGV. By integrating different sensors, the drawbacks of a single sensor can be compensated. The IMU provides high-frequency accurate position information in a short time; however, its cumulative error and acceleration drift are unsuitable for long-term use. The Global Positioning System offers meter-level positioning; however, the indoor signal is weak. Wheel odometry (W-Odom) is a cost-effective option that utilizes the difference in wheel speed to obtain linear and angular velocities; however, it loses accuracy when the wheels spin or the mechanical structure wears out. The characteristics of the proposed HT-Carto and our contributions are summarized as follows:

(1) The preprocessing of IMU and LiDAR data is carried out using the front-end framework of Cartographer, which addresses the issues of loosely coupled sensors, inadequate utilization of observation information, and low point cloud quality.

(2) This study proposes a Hybrid Filter that enhances the point selection strategy based on adaptive VoxelGrid, integrates Passthrough, and RadiusOutlierRemoval to eliminate outliers and accelerates the speed of subsequent algorithms.

(3) Devise a "plug-and-play" independent module for seamless integration. Initially, a preintegrated model of the IMU is derived, incorporating LiDAR observations into the IMU odometry, enabling the correction of the IMU bias and enhancing computational efficiency. Subsequently, the selection approach for the linear and angular velocities in the PoseExtrapolator is refined by updating the starting point of each frame, thereby improving the overall localization accuracy.

(4) The application of the pre-integration technique in 3D SLAM to 2D SLAM allowed the Cartographer to obtain a new attempt and test the HT-Carto algorithm in several environments.

2. Related work

SLAM was introduced at the IEEE Robotics and Automation Conference in San Francisco in 1986. GMapping was proposed in 2007 and constructed a SLAM system based on Rao-Blackwellized particle filters (RBPF) [7]. State-space vectors have been applied in analytical methods under certain conditions, and when the odometer model is propagated, we use optimal particles to reduce the number and prevent degradation. GMapping is highly dependent on W-Odom, making it unsuitable for constructing large-scene maps. Hector SLAM solves the least squares problem by the Gauss-Newton method in 2013 [8]. The optimization procedure ensures the solution's independence from the W-Odom, while Cartographer incorporates and executes the scan matching methodology.

Researchers have increasingly relied on factor graph to tackle SLAM problems as hardware performance has advanced. Cartographer employs a hierarchical optimization approach using the unscented Kalman filter (UKF) to integrate multi-source data for position estimation and construct a submap in the front-end. The submap serves as the fundamental unit for constructing the optimization problem, a branch-and-bound algorithm expedites the establishment of constraints between the submaps. Although data fusion is loose, it is well suitable for scenarios with less stringent accuracy requirements. Upon evaluating multiple indoor SLAM, Zou [9] and Herranz [10] determined that Cartographer exhibits distinct superiority, it was designed as an engineered project with a comprehensive balance of configuration parameters. Sobczak identified the optimal configuration for Cartographer in the simulation by adjusting the parameters [11]. Gao proposed a lightweight and efficient neighborhood encoding-based [12] to achieve better map. Simanek exploited an extended Kalman filter to reckon mobile robots [13]. Another idea is to use a neural network in 3D LiDAR [14, 15], [16] proposed geometry and intensity features based on the scan context for loop closure [17].

Currently, 3D LiDAR and visual SLAM offer new approaches. LOAM [18], LIO-SAM [19], Lego-LOAM [20] apply the pre-integration to tackle the issue of substantial cumulative errors and copious sensor data. D-LIOM employs the pre-integration and gravity constraints during submap construction in 3D Cartographer applications [21]. Numerous researchers employ this approach as a foundation for various enhancements. Tightly coupled sensor data are mainly used in visual SLAM, [22] presented a probabilistic monocular visual-odometric SLAM and so on.

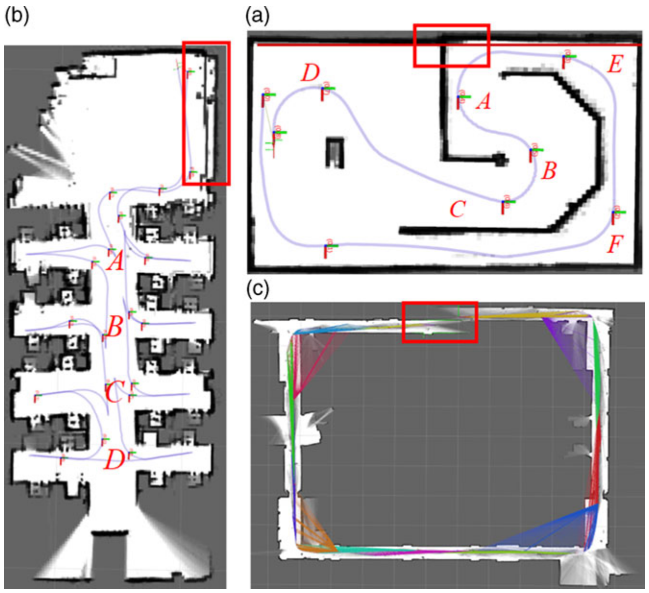

Figure 1. *Defect maps. (a) denotes 9.0 m×5.3 m rotated small map, the rotation angle is over* 180°*(i.e., A,B,C,D) while (i.e., E,F) is over* 90°*; (b): 21.2 m×8.1 m symmetrical middle map, (i.e., A,B,C,D) share the same point cloud characteristics; (c): 65 m×42 m big unfeatured map. Images a and b with local SLAM while c with global SLAM(back-end).*

As can be seen, Cartographer employs four primary optimization methods: (1) parameter adjustment; (2) utilization of loosely coupled sensors based on Kalman and particle filtering; (3) incorporation of tightly coupled pre-integration and factor graph; and (4) integration of scan context to enhance scan matching at the front-end. This study cites the third idea for application to 2D Cartographer.

3. Problem formulation

In the front-end (local slam) of the Cartographer, the laser data are filtered by an adaptive VoxelGrid, and the PoseExtrapolator provides the initial position through the IMU and W-Odom.

3.1. VoxelGrid

LiDAR collects environmental data to create a point cloud and evaluates suitable points by specifying distance criteria. This process involves multiple iterations of the adaptive VoxelGrid to minimize the number of point clouds while preserving the features. However, despite these efforts, noise points may persist owing to factors such as line-of-sight occlusion and the surface materials of obstacles.

3.2. Scan matching

The robot can obtain an accurate position when traveling slowly without violent rotation. However, laboratory environments are complex and unpredictable. Figure 1 shows the mapping results for common laboratories with map offsets in red boxes.

Figure 1(a) depicts multiple violent rotations, where the robot is prone to losing its position during rotation, leading to an offset in the map, which can be corrected through loop closure. In the symmetric structure environment shown in Figure 1(b), if we turn on the back-end, the map shows erroneous

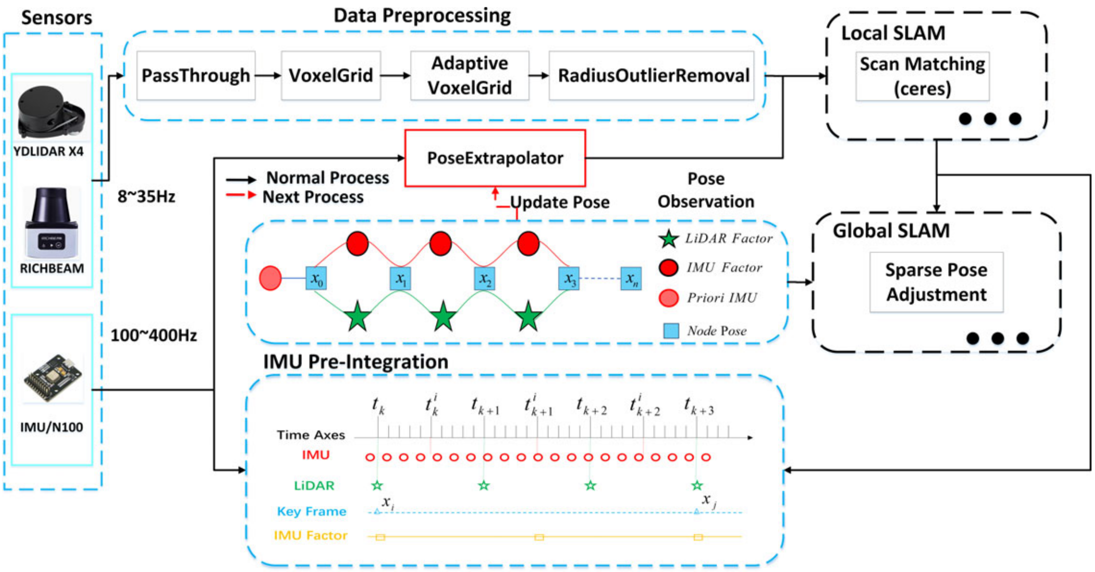

Figure 2. *Framework overview of HT-Carto. In "Sensors," HT-Carto provides two LiDARs to be selected, they need to match the IMU to get suitable frequency. In "Data preprocessing," we use a new approach to handle the point cloud. No alterations were made to the local or global SLAM component. In contrast, we developed IMU pre-integration and LiDAR odometry factors to facilitate IMU odometry acquisition through a factor graph. This odometry is then fed into the PoseExtrapolator for the current position updates.*

matching. Figure 1(c) illustrates a corridor with a degraded structure. Owing to the lack of features, even with loop closure, an error of 2 m×2.5 m occurs in the red box. The back-end benefits the accuracy of Cartographer's maps, but it has limitations in a symmetric structural environment.

3.3. PoseExtrapolator

Table I illustrates the hierarchical algorithm used to prioritize angular and linear velocity data during position renewing. W-Odom is not utilized due to its inferior performance, while high-frequency IMU data will consume substantial computation. The Pose estimation signifies the position data following the last scan matching, and we can also predict the current position by using the data from the preceding frame. However, the position accuracy decreases. If none of these is reachable, Cartographer will use a uniform rectilinear model to estimate.

4. Theory and methodology

4.1. Improved framework

We optimize the front-end to enhance the accuracy of robot localization and mapping in unknown environment.

The overall framework of HT-Carto is shown in Figure 2 and comprises two main components. The first component involves a hybrid filter of the point cloud, the second initializes the IMU data statically and dynamically with reference to D-LIOM [21]. In theory, the IMU has no bias, and the integral of the angular velocity is ideally equal to the value estimated by the LiDAR odometer. However, in practical application, the external noise and internal bias of the IMU cannot be ignored. To address this

Algorithm 1: Hybrid point-cloud filtering for HT-Carto

```
   Data: sensor_msgs::LaserScan old, raw data from LiDAR
         sensor::PointCloud& temp, temporary variable
         pars, configuration parameters
   Result: sensor::PointCloud& new, Cartographer's data type
1  InitialPar(pars); // Initial parameters
2  temp = DataTypeTransform(old);
3  temp = PassThrough(temp, pars); // Step 1
4  temp = VoxelGrid(temp, pars); // Step 2
   // Step 3
5  if temp.size <= pars.minimum then
6      return temp;
7  end
8  temp = VoxelGrid(temp, pars.max_length);
9  if temp.size > pars.minimum then
10     return temp;
11 end
12 for pars.max_length/2 do
13     temp = VoxelGrid(temp, pars.max_length/2);
14     temp = getTheSuitableLength(temp, pars.max_length/2);
15     return temp;
16 end
17 new = RadiusOutlierRemoval(temp, pars); // Step 4
18 return new;
```

problem, the system acquires timestamps from two consecutive LiDAR frames and uses scan matching as observation to constrain the IMU noise and bias.

4.2. Hybrid filter

Preprocessing of the point cloud can reduce the quantity and noise, which improves the quality of the point cloud and reduces the effect of noise in the subsequent scan matching. In the original algorithm, Cartographer employs distance judgment, VoxelGrid, and Adaptive VoxelGrid. VoxelGrid can preserve most environmental characteristics, but its indoor environment is complex. The point cloud emanating from the interstices between the tables and chairs is not conducive to map construction, and is thus classified as a noise point. A RadiusOutlierRemoval was used to remove these points. A hybrid filter was designed, as shown in Figure 2, based on PassThrough to remove a specific range of points. As shown in Algorithm.1 line 3, VoxelGrid achieves down sampling to preserve the geometric features of the point cloud, while RadiusOutlierRemoval filters outlier points.

4.3. IMU pre-integration

An IMU typically comprises a gyroscope and accelerometer, each containing a three-axis angular velocity and three-axis acceleration, respectively. Theoretically, the robot's position can be determined by integrating the angular velocity and acceleration; however, bias and noise impair the accuracy. Moreover, long-term integration not only accumulates errors but also increases computational resources.

Assuming that the IMU coordinate system is coincident with the robot system *Body*(b), the world coordinate system is *World*(w), and the z-axis is aligned with the direction of gravity in the w. The physical measurement model of the IMU is as follows [4]:

$$\begin{aligned} &\widetilde{w}^b(t) = w^b(t) + b^g(t) + \eta^g(t) \\ &\widetilde{a}^b(t) = (R_b^w)^T(t)\left[a^w(t) - g^w\right] + b^a(t) + \eta^a(t) \end{aligned} \tag{1}$$

In Eq. (1), $\widetilde{w}^b(t)$ and $w^b(t)$ denote the measurement and actual values of the angular velocity in the b system at moment t, $\widetilde{a}^b(t)$ is the measured value of acceleration in the b system, while $a^w(t)$ is the true value in the w system. $(R_b^w)^T$ refers to the rotational attitude matrix under the b system relative to the w system. $b^g(t)$ and $b^a(t)$ are the bias of accelerometer and gyroscope measurements. g^w indicates the gravity of the current area. $\eta^g(t)$ and $\eta^a(t)$ are measurement noises at the time t.

$$\begin{aligned} &\dot{R}_b^w = R_b^w(\omega^b)^\wedge \\ &\dot{v}^w = a^w \\ &\dot{p}^w = v^w \end{aligned} \tag{2}$$

Combined with the different form of the kinematic equation in Eq. (2) for further derivation. First, the forward Euler method is applied to obtain the discrete form of kinematics. The discretized integral recursion is then performed on the attitude (R), velocity (v), and position (p) after the Δt transformation. To simplify the equations, sign substitution like the Eq. (3) was applied to the w system, and Eq. (4) can be obtained. Where η^{ad}, η^{gd} denote discrete Gaussian noise. Knowledge of Lie Groups and Lie Algebra [23] was used to derive the formula.

$$R(t) \triangleq R_b^w(t), v(t) \triangleq v^w(t), p(t) \triangleq p^w(t) \tag{3}$$

$$\begin{aligned} &R_{k+1} = R_k Exp\left[\left(\tilde{\omega}_k - b_k^g - \eta_k^{gd}\right)\Delta t\right] \\ &v_{k+1} = v_k + R_k\left(\tilde{a}^k - b_k^a - \eta_k^{ad}\right)\Delta t + g\Delta t \\ &p_{k+1} = p_k + v_k\Delta t + \tfrac{1}{2}g\Delta t^2 + \tfrac{1}{2}R_k\left(\tilde{a}^k - b_k^a - \eta_k^{ad}\right)\Delta t^2 \end{aligned} \tag{4}$$

The IMU operates at a frequency range of 100–400 Hz. If each data need to be integrated during motion solving, the resulting nodes that are inserted into the factor graph require extensive CPU time. Additionally, the new data are highly correlated with the historical data, leading to repeated calculations. The LiDAR data between two frames (moments i and j) are selected as observations, and the pre-integration model is utilized to determine the zero bias of the IMU. The variation is subsequently applied to update the IMU data in the next frame, with the aim of reducing the computation. Furthermore, if the radar data are deemed unreliable, IMU pre-integration will provide the initial position of the subsequent scan.

Equation (4) determines the changes in R, v, p over two intervals. However, each calculation must start from its initial value, which requires the construction of the pre-integration term in Eq. (5).

$$\begin{aligned} &\Delta R_{ij} = R_i{}^T R_j = \prod_{k=i}^{j-1} Exp\left[\left(\tilde{\omega}_k - b_k^g - \eta_k^{gd}\right)\Delta t\right] \\ &\Delta v_{ij} = R_i{}^T\left(v_j - v_i - g\Delta t_{ij}\right) = \sum_{k=i}^{j-1} \Delta R_{ik}\left(\tilde{a}^k - b_k^a - \eta_k^{ad}\right)\Delta t \\ &\Delta p_{ij} = R_i{}^T\left(p_j - p_i - v_i\Delta t_{ij} - \tfrac{1}{2}g\Delta t_{ij}^2\right) \\ &= \sum_{k=i}^{j-1}\left[\Delta v_{ik}\Delta t + \tfrac{1}{2}\Delta R_{ik}\left(\tilde{a}^k - b_k^a - \eta_k^{ad}\right)\Delta t^2\right] \end{aligned} \tag{5}$$

For b(bias) and η(white noise) in the pre-integration, the value of each individual moment is theoretically unequal. Consequently, the recursion of noise, as described in Eq. (5), is necessary to determine the residuals of R, v, p.

Algorithm 2: Tightly coupled LiDAR/IMU framework for HT-Carto

```
Data: sensor_msgs::Imu imu, raw data from IMU
      nav_msgs::Odometry odom, laser odometry from Cartographer
      pars, IMU related parameters
Result: nav_msgs::Odometry imuOdom, IMU odometry after immediate pre-integration
        velocity, linear_velocity and angular_velocity in the PoseExtrapolator
  // Step1: Get the imuOdom
1 InitialSystem(pars, imu, odom, gtsam);
2 imuOdom = gtsam_solve(imu, odom);
  // Step2: Update the PoseExtrapolator
3 TrimImuOdomData(30);
4 $v_{new}$ = imuOdom.linear_velocity_new_;
5 $v_{last}$ = imuOdom.linear_velocity_last_;
6 imuTracker ->AddImuLinearAccelerationObservation($v_{new}$, $v_{last}$);
7 calcCoorTrans($v_{new}$, $v_{last}$);
8 $w_i$ = ExtrapolateRotation(imuTracker);
9 $v_i$ = ExtrapolateTranslation();
```

4.4. Factor graph

A factor graph is used for model estimation by transforming the problems involved in solving a joint probability density function and can make full use of the sensor's information from historical times [24]. Variable nodes represent the quantity to be optimized, such as the robot's position, whereas factor nodes represent observation constraints, including IMU pre-integration factors and position factors from scan matching. The process involves solving for the optimal value of the objective function, as shown in Eq. (6), which consists of the residuals of each item in the system, and determines the optimal value of the objective function through nonlinear optimization.

$$\min_X \left\{ \sum ||r_L \left(L_k^{k+1}, X\right)||^2 + \sum ||r_B \left(B_k^{k+1}, X\right)||^2 \right\} \tag{6}$$

Where r_L denotes the residuals from LiDAR odometry and r_B denotes the residuals after IMU pre-integration.

4.5. Tightly coupled LiDAR/IMU

Both IMU pre-integration and factor graphs are utilized to obtain IMU Odometry. As indicated in Algorithm 2, the data acquired from GTSAM are employed to update the PoseExtrapolator. Initially, through trimming to reduce the data in the cache, the first and last data points in the queue are utilized to calculate the linear and angular velocities. Subsequently, the rotation and translation values in the PoseExtrapolator are updated.

5. Experimental results

Two economical single-line LiDARs matching frequency-adjustable IMU are assembled on an Ackermann car to complete tests in diverse intricate settings, as shown in Figure 3.

5.1. Experimental protocols

(1) Implements:

This study adds two external modules to improve the performance of the Cartographer. All modules in HT-Carto are implemented in C++, which runs in the robot operating system [25] (ROS 1.0) in

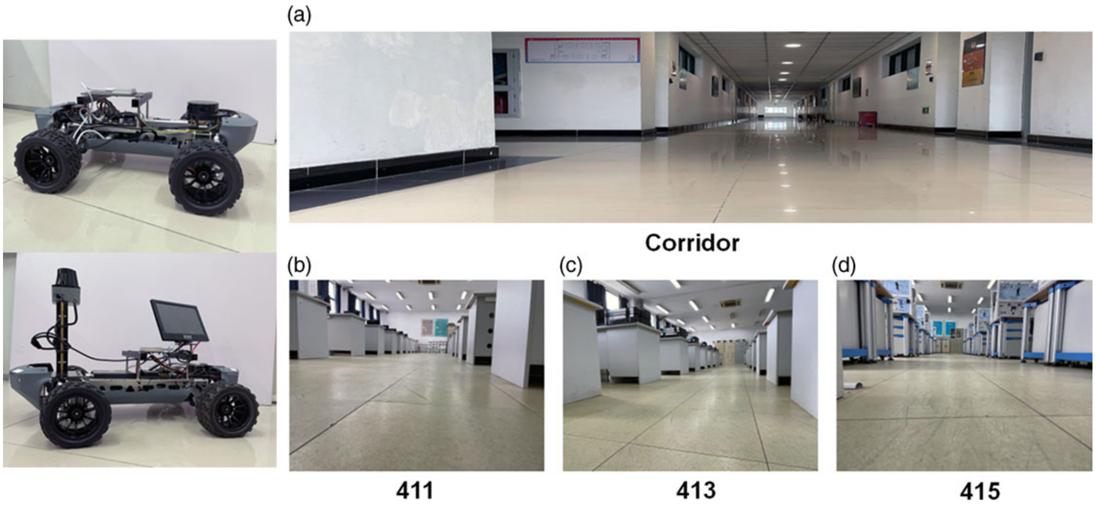

***Figure 3.** Experimental environment. The left panel shows the Ackermann car with LiDAR(YDLIDAR X4, RICHBEAM 1). Right is the test environment, which consists a corridor and three symmetry structure laboratories. (a) With the size of 63.2 m×2.1 m; (b) 15.8 m×7.9 m; (c) 15.7 m×7.9 m; (d) 15.6 m×7.9 m.*

Ubuntu Melodic. The configurations of the two LiDARs are presented in Table II, and the corresponding experimental scenarios are illustrated in Figure 3 (right). To ensure the replicability of the experiments, numerous datasets are recorded by the Jeston Nano 4 g vehicle platform. These datasets are subsequently transferred to an Intel i7-12 64 g to test HT-Carto, and the experimental results are graphically represented using Matplotlib.

(2) Datasets:

In the Hybrid Filter of HT-Carto test, as shown in Figure 3(b), two lidars were used to record the datasets for a few minutes while the robot remained stationary.

For the localization and mapping of HT-Carto, we used the fourth floor of the laboratory building as the experimental site, as shown in Figure 3(right). To obtain a suitable match between the LiDAR and IMU frequencies and verify the feasibility in a strongly rotating environment, an additional small map is added, as shown in Figure 1(a).

As a uniform note, the parameters in HT-Carto were consistently maintained, and the back-end was disabled. YDLIDAR and RICHBEAM were used as acronyms for their respective radar carts. "Cartographer," "Loop Closure" marking the mapping results of the local SLAM and global SLAM. The result of Loop Closure was used as a reference trajectory if it could be opened. Verification of HT-Carto's characteristics was accomplished by examining the similarity and size of trajectories and maps. First, the similarity between Cartographer, HT-Carto, GMapping, Hector, Karto, and Loop Closure trajectories were assessed using dynamic time warping (DTW) [26]. Subsequently, the EVO trajectory evaluation tool [27] was employed to conduct a comparative analysis of the absolute trajectory error (ATE) and relative pose error (RPE) between trajectories. Finally, several distinct points were marked on each map, the discrepancies between the map and reality were evaluated by comparing the field measurements and utilizing the measurement tool in the RVIZ [28].

5.2. Hybrid filter of HT-Carto

Figure 4 demonstrates the filtering effects of the two LiDARs in diverse environments. Gray serves as the initial point cloud, while green and red represent the refined outcomes. A comparison of the colors demonstrates that the number of individual points is diminished following the Hybrid Filter. As shown

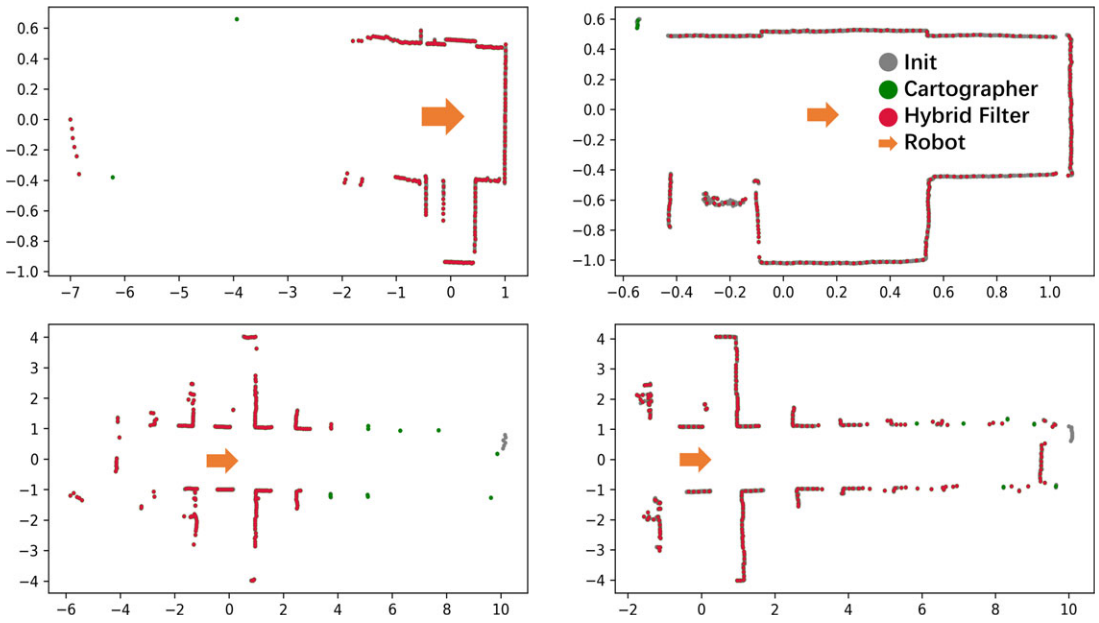

Figure 4. *Hybrid filter results. The horizontal pictures are the same feature environment as shown in Figure 3(b), while the vertical images are the same LiDAR. The left pictures are RPLIDAR X4 and the right are RICHBEAM 1.*

in Table III, the point cloud was obtained from different areas: one was small, and the other was large. The filtering results are shown in Figure 4, after several thousand scans of point clouds were tested on both HT-Carto and Cartographer, the number was reduced by 4%, whereas the time was increased by only a few hundred microseconds.

5.3. Localization and mapping of HT-Carto

5.3.1. Strongly rotating environment

Table IV lists the experimental findings for the two LiDARs shown in Figure 3(b). The trajectories and maps from the Loop Closure were utilized as reference values for comparison with HT-Carto and others. Five different sensor combinations with varying frequencies were selected for the experiment to evaluate the plane errors and DTW values between trajectories. A smaller DTW value signifies a higher degree of similarity between the trajectories.

The cross-sectional analysis indicates that the errors between HT-Carto and Loop Closure are smaller and more similar than those of Cartographer across all levels, as shown in Table IV. However, the magnitude of the error cannot be used as a criterion for directly selecting the frequency. From a longitudinal comparison of the data, the similarity of the trajectories between HT-Carto and Loop Closure was higher at frequencies of "8 + 100" and "30 + 100". These two sets of frequencies are selected for further testing. Table V lists the root mean square error (RMSE) results of the ATE and RPE from GMapping, Hector, Cartographer, and HT-Carto which compared to Loop Closure respectively. Except for GMapping, the remaining algorithms exhibited minimal differences and demonstrated excellent performance in map construction within small-scale environments.

Our findings indicate that the Cartographer and HT-Carto demonstrated satisfactory performance in the trajectory comparison. Furthermore, it is imperative to conduct a comparative analysis of their mapping efficacy. The beginning of trajectories in Figure 5 are highly overlapped and the errors accumulated

Table I. *Velocity selected rules in poseExtrapolator.*

Priority type	**Linear velocity**	**Angular velocity**
0	W-Odom	IMU
1	Pose estimation	W-Odom
2	No	Pose estimation
3	Uniform rectilinear model	Uniform rectilinear model

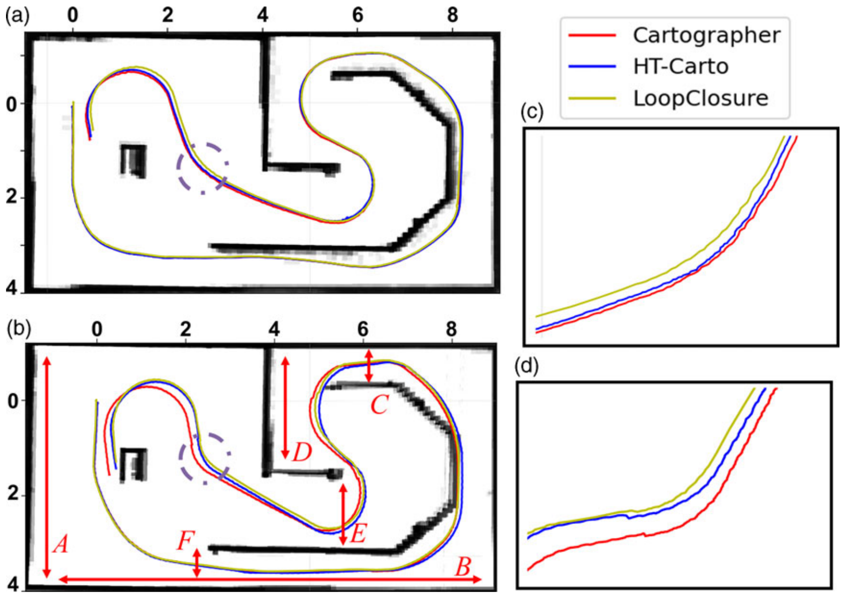

Figure 5. *Trajectories and maps (Figure 3b) of the two LiDARs. The red line represents the trajectory of Cartographer which closes the back-end. The blue line is HT-Carto, whereas the yellow line is Cartographer with loop closure, which is close to the true path. (a) notes YDLIDAR, (b) notes RICHBEAM, (c) and (d) represents the Partial Zoom from the left's dashed circle boxes.*

with the robot's motion. The deviation of trajectories would be larger if there is no back-end, whereas the high overlap between HT-Carto and Loop Closure indicates that HT-Carto is closer to the real path.

Then, a map size comparison is carried out; six distances are marked in Figure 5, and Table VI shows the results of the real and test values under the two algorithms. Refs. A and B illustrate the full map's size, which is important than the others.

5.3.2. Symmetry environment

The preceding section demonstrated the effectiveness of HT-Carto in optimizing the front-end. As shown in Figure 3(b), the environment is symmetric; therefore, Loop Closure can not be utilized. Figure 6 depicts the results of the map construction, and the red arrows indicate the three optional marker distances. In a comparison of the true values, HT-Carto reduced the map size error by 4.5 cm in Figure 6(left) and 2.4 cm in Figure 6(right) compared to the Cartographer.

Table II. *Parameters of LiDAR.*

Feature name (Unit) **LiDAR**	**YDLIDAR X4**	**RICHBEAM 1**
Scan frequency (Hz)	8	10,20,25,30
Measurement distance (m)	<10	<25
Angular resolution (deg)	0.5	0.25
Scan angle (deg)	360	270
Repeatability (cm)	<2%	2
Point numbers	720	2880

Table III. *Point cloud comparison.*

			Cartographer		**Hybrid Filter**	
LiDAR	**Area**	**Scans**	**Number**	**Time/ms**	**Number**	**Time/ms**
YDLIDAR	small	1780	205	1.6	203	1.9
	big	1030	207	1.5	198	1.8
RICHBEAM	small	6664	204	2.4	202	2.8
	big	5199	207	2.9	199	3.1

Table IV. *Trajectory comparison.*

		Cartographer[2]			**HT-Carto**[2]			
		STD (cm)			**STD (cm)**			
LiDAR	**Freq**[1] **(Hz)**	***x***	***y***	**DTW**	***x***	***y***	**DTW**	**DTW_ratio (%)**
YDLIDAR	$8+100$	2.8	2.6	0.07	1.7	2	0.06	16%
RICHBEAM	$10+100$	1.4	1.7	0.06	0.8	0.9	0.04	23%
	$20+100$	7.3	5.4	0.13	6.2	3.7	0.11	14%
	$30+100$	4.8	5.7	0.1	2.5	2.5	0.07	26%
	$30+200$	3.9	3.7	0.09	4.2	3.5	0.09	1%

[1]Freq represents the combination of the LiDAR + IMU frequency.
[2]Standard deviation in the difference of x,y directions, compare trajectory differences between Cartographer (without back-end), HT-Carto and Loop Closure.

Table V. *EVO comparison.*

	8 + 100 RMSE			**30 + 100 RMSE**		
Type	**ATE**	**RTE**	**DTW**	**ATE**	**RTE**	**DTW**
GMapping	0.58	0.047	8	1.51	0.01	7.21
Hector	0.07	0.008	7	0.07	0.01	7.61
Cartographer	0.05	0.003	0.034	0.1	0	0.07
HT-Carto	0.04	0.006	0.03	0.07	0	0.06

Table VI. *Comparation of the map size.*

		YDLIDAR 8 + 100			RICHBEAM 30 + 100		
Ref[1]	**Truth**	**Carto**[2]	**HT-Carto**	**Optimal**	**Carto**	**HT-Carto**	**Optimal**
A	525.5	520.1	525.6	5.3	530.6	524.1	3.7
B	893.8	885.1	896.8	5.7	901.8	895.4	6.3
C	84.6	70.5	75.3	4.8	85.5	83.2	−0.6[3]
D	274.5	268.9	270.3	1.4	275.8	274.8	1
E	162.3	164.9	165.7	−0.8[3]	160.5	162.7	1.4
F	87.5	74.4	80.3	5.9	75.3	80.5	5.2

[1] All units in the table are in cm. Ref: A-F note the red marks from Figure 5b. Assume Optimal=("Cartographer"- "True")- ("HT-Carto"- "True");

[2] Cartographer, the widths of the table columns are too small.

[3] The discrepancy was less than 1 cm and can be considered negligible.

Table VII. *Trajectory comparison of the dynamic environment.*

	ATE		PRE		
	RMSE	**STD**	**RMSE**	**STD**	**DTW**
Hector	3.29	1.1	0.016	0.014	6.75
Karto	14.84	7.39	0.036	0.028	27.17
Cartographer	0.93	0.58	0.013	0.011	0.97
HT-Carto	0.19	0.09	0.01	0.007	0.16

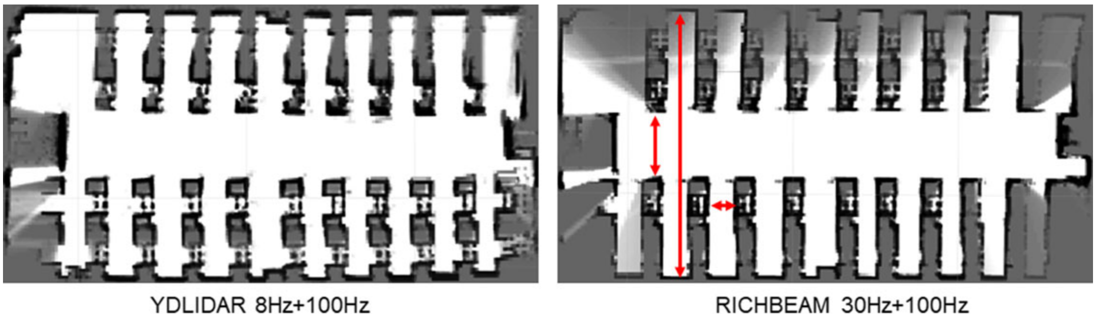

Figure 6. *Symmetry environment mapping results in HT-Carto.*

5.3.3. Dynamic environment

The corridor is illustrated in Figure 3(a). Without sufficient feature points, a robot is susceptible to losing its current positional attitude. Consequently, a sensor combination of 30 and 100 Hz from Table IV was selected for testing, which enabled HT-Carto to reduce the error by 10 cm.

To further simulate the authentic laboratory environment, we incorporate a dynamic scenario experiment by placing books on the floor to represent items left behind by students and positioning four persons to ambulate randomly within the environment. The Cartographer drifts significantly in the red box, as shown in Figure 7(a), and HT-Carto performs better in Figure 7(b). Figure 7(c) displays a map generated from the Loop Closure, and HT-Carto clearly performs better on the trajectory. Figure 7(d) and (e) show the details of the map; Loop Closure leads to map overlaps, whereas HT-Carto is more complete in the structural symmetry scene. Table VII illustrates the comparative outcomes of trajectory analysis for the four algorithms. As the environmental area expanded, GMapping became impractical, and the Hector exhibited excessive reliance on the laser data. Cartographer(without back-end) and Karto demonstrated loosely coupled sensor data.

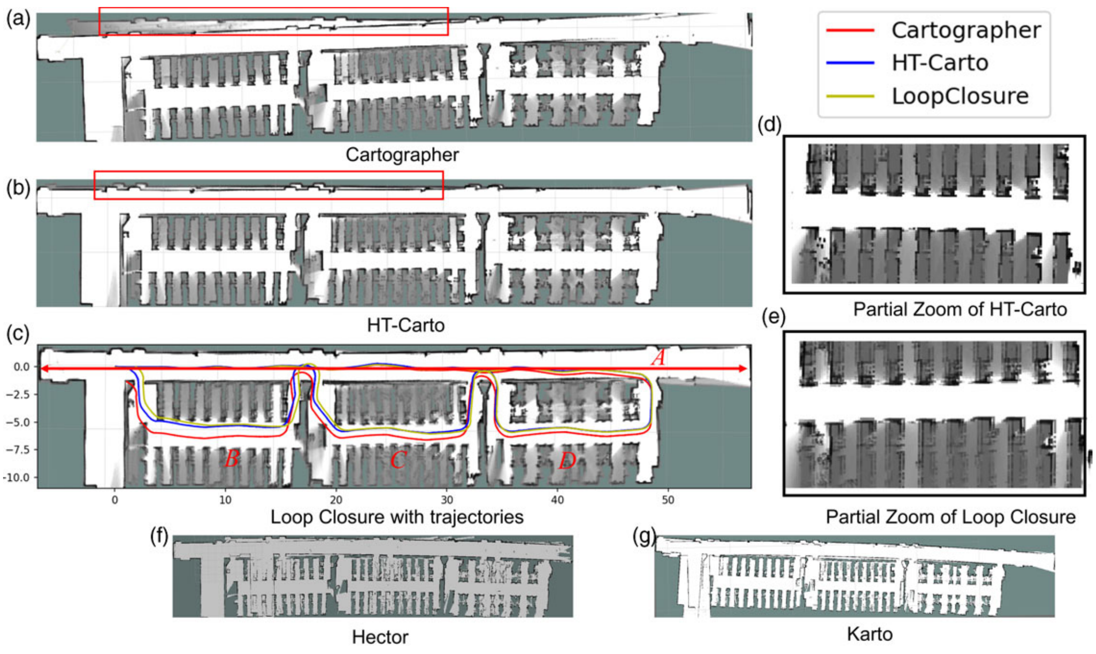

Figure 7. *Dynamic mapping results. The left panel includes three images representing the results of the Cartographer, HT-Carto, and Loop Closure. The red letter A is a corridor and (i.e.,B,C,D) are a symmetry laboratory. Right is the Partial Zoom like the C. The last line shows the maps for both Hector and Karto.*

6. Conclusion

HT-Carto is proposed as a solution for AGV in situations where the back-end is unavailable in certain environments when using the Cartographer for localization and mapping. The two low-cost 2D LiDARs examined in this study provide cost savings while maintaining accuracy compared with 3D LiDAR and vision cameras. By applying pre-integration to the Cartographer to obtain a new attempt, an independent factor graph module was integrated into the front-end of the system to determine its position more accurately. To guarantee the quality of the point cloud and minimize the impact of noise, a Hybrid Filter algorithm was proposed. The outcomes of these environments indicate that HT-Carto can achieve similar or better accuracy when compared with Cartographer(back-end) and others. HT-Carto can employ standard CPU and memory resources on an embedded platform; however, this work only set the 100 Hz of the IMU and 3D SLAM to 500 Hz. In the future, we will increase the frequency and consider the computation. We can also activate the back-end in HT-Carto; however, because of the loss of a more precise sensor, we cannot obtain a reference trajectory. Perhaps it performs more effectively than Loop Closure.

Author contributions. Author contributions Conceptualization was contributed by C.D., Z.C. and T.S.; methodology was contributed by C.D., T.S. and Y.L.; software was contributed by C.D.; data curation was contributed by T.S. and Y.L.; validation was contributed by C.D.; formal analysis was contributed by C.D. and Z.C.; investigation was contributed by Z.C., T.S., Y.L.; writing – original draft preparation, was contributed by C.D; writing – review and editing, was contributed by Z.C., T.S., Y.L.; visualization was contributed by Y.L; supervision was contributed by T.S.; project administration was contributed by Z.C.; all authors have read and agreed to the published version of the manuscript.

Financial support. This research received no specific grant from any funding agency, commercial, or not-for-profit sectors.

Competing interests. The authors declare no conflicts of interest exist.

Ethical approval. Not applicable.

References

[1] M. Aizat, N. Qistina and W. Rahiman, "A comprehensive review of recent advances in automated guided vehicle technologies: Dynamic obstacle avoidance in complex environment toward autonomous capability," *IEEE Trans. Inst. Meas.* **73**, 1–25 (2024).

[2] G. Winkelmaier, R. Battulwar, M. Khoshdeli, J. Valencia, J. Sattarvand and B. Parvin, "Topographically guided uav for identifying tension cracks using image-based analytics in open-pit mines," *IEEE Trans. Ind. Electron.* **68**(6), 5415–5424 (2021).

[3] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid and J. J. Leonard, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Trans. Rob.* **32**(6), 1309–1332 (2016).

[4] T. Qin, P. Li and S. Shen, "Vins-mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Trans. Rob.* **34**(4), 1004–1020 (2018).

[5] T. Qin and S. Shen, "Robust Initialization of Monocular Visual-Inertial Estimation on Aerial Robots," **In:** *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2017) pp. 4225–4232.

[6] 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)* (2016) pp. 1271–1278.

[7] G. Grisetti, C. Stachniss and W. Burgard, "Improved techniques for grid mapping with rao-blackwellized particle filters," *IEEE Trans. Rob.* **23**(1), 34–46 (2007).

[8] S. Kohlbrecher, O. von Stryk, J. Meyer and U. Klingauf, "A Flexible and Scalable Slam System with Full 3D Motion Estimation," **In:** *2011 IEEE International Symposium on Safety, Security, and Rescue Robotics* (2011) pp. 155–160.

[9] Q. Zou, Q. Sun, L. Chen, B. Nie and Q. Li, "A comparative analysis of lidar slam-based indoor navigation for autonomous vehicles," *IEEE Trans. Intell. Transp.* **23**(7), 6907–6921 (2022).

[10] F. Herranz, A. Llamazares, E. Molinos, M. Ocaña and M. A. Sotelo, "Wifi slam algorithms: An experimental comparison," *Robotica* **34**(4), 837–858 (2016).

[11] S. Łukasz, F. Katarzyna, D. Joanna and D. Adam, "Finding the best hardware configuration for 2d slam in indoor environments via simulation based on google cartographer," *Sci. Rep-UK* **12**(1), 18815–18815 (2022).

[12] H. Gao, X. Zhang, J. Yuan and Y. Fang, "Negl: Lightweight and efficient neighborhood encoding-based global localization for unmanned ground vehicles," *IEEE Trans. Veh. Technol.* **72**(6), 7111–7122 (2023).

[13] J. Simanek, M. Reinstein and V. Kubelka, "Evaluation of the ekf-based estimation architectures for data fusion in mobile robots," *IEEE/ASME Trans. Mechatron.* **20**(2), 985–990 (2015).

[14] H. Yin, Y. Wang, X. Ding, L. Tang, S. Huang and R. Xiong, "3d lidar-based global localization using siamese neural network," *IEEE Trans. Intell. Transp.* **21**(4), 1380–1392 (2020).

[15] F. Alkhawaja, M. A. Jaradat and L. Romdhane, "Low-cost depth/imu intelligent sensor fusion for indoor robot navigation," *Robotica* **41**(6), 1689–1717 (2023).

[16] G. Kim and A. Kim, "Scan Context: Egocentric Spatial Descriptor for Place Recognition within 3D Point Cloud Map," **In:** *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2018) pp. 4802–4809.

[17] H. Wang, C. Wang and L. Xie, "Intensity-slam: Intensity assisted localization and mapping for large scale environment," *IEEE Rob. Autom. Lett.* **6**(2), 1715–1721 (2021).

[18] J. Zhang and S. Singh, "LOAM: Lidar Odometry and Mapping in Real-Time," **In:** D. FoxL. E. Kavraki and H. Kurniawati eds. *Robotics: Science and Systems X, University of California, Berkeley, USA, July 12-16, 2014* (2014).

[19] T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti and D. Rus, "Lio-Sam: Tightly-Coupled Lidar Inertial Odometry via Smoothing and Mapping," **In:** *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2020) pp. 5135–5142.

[20] T. Shan and B. Englot, "Lego-Loam: Lightweight and Ground-Optimized Lidar Odometry and Mapping on Variable Terrain," **In:** *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2018) pp. 4758–4765.

[21] Z. Wang, L. Zhang, Y. Shen and Y. Zhou, "D-liom: Tightly-coupled direct lidar-inertial odometry and mapping," *IEEE Trans. Multimedia* **25**, 3905–3920 (2023).

[22] M. Quan, S. Piao, M. Tan and S.-S. Huang, "Tightly-coupled monocular visual-odometric slam using wheels and a mems gyroscope," *IEEE Access* **7**, 97374–97389 (2019).

[23] H. A. Hashim and A. E. E. Eltoukhy, "Nonlinear filter for simultaneous localization and mapping on a matrix lie group using IMU and feature measurements," *IEEE Trans. Syst. Man Cybern. Syst.* **52**(4), 2098–2109 (2022).

[24] P. Lyu, B. Wang, J. Lai, S. Bai, M. Liu and W. Yu, "A factor graph optimization method for high-precision imu-based navigation system," *IEEE Trans. Inst. Meas.* **72**, 1–12 (2023).

[25] R. F. Carpio, C. Potena, J. Maiolini, G. Ulivi, N. B. Rosselló, E. Garone and A. Gasparri, "A navigation architecture for Ackermann vehicles in precision farming," *IEEE Rob. Autom. Lett.* **5**(2), 1103–1110 (2020).

[26] W. Li, R. He, B. Liang, F. Yang and S. Han, "Similarity measure of time series with different sampling frequencies based on context density consistency and dynamic time warping," *IEEE Signal Proc. Lett.* **30**, 1417–1421 (2023).

[27] Y. Zhou, Y. Wang, F. Poiesi, Q. Qin and Y. Wan, "Loop closure detection using local 3d deep descriptors," *IEEE Rob. Autom. Lett.* **7**(3), 6335–6342 (2022).

[28] D. T. Fasiolo, L. Scalera and E. Maset, "Comparing lidar and imu-based slam approaches for 3d robotic mapping," *Robotica* **41**(9), 2588–2604 (2023).

Cite this article: C. Da, Z. Chen, T. Song and Y. Lu, "Tightly coupled SLAM system for indoor complex scenes", Robotica. https://doi.org/10.1017/S0263574725000463