ISSN 1330-3651 (Print), ISSN 1848-6339 (Online)
https://doi.org/10.17559/TV-20240720001869
Original scientific paper

Synchronous Localization and Mapping of Mobile Robots Based on Pose Graph Optimization Algorithm

Xuan Rong ZHAO

Abstract: This study proposes an enhanced Simultaneous Localization and Mapping (SLAM) technique for mobile robots, integrating Inertial Measurement Unit (IMU) pre-integration theory with a Cardinality Balanced Multi-target Multi-Bernoulli (CBMeMBer) filter for pose graph optimization. The method aims to improve the accuracy of laser point cloud pose estimation and enhance posegraph optimization. Experimental results on public datasets demonstrate the model's superior performance, achieving 97% processing accuracy with a minimum of 180 iterations. Compared to existing advanced models, the proposed approach shows improved trajectory prediction and error control, with the lowest root mean square error of 1.86 m in positioning distance. This research contributes to advancing SLAM technology for more accurate and efficient robot navigation in complex environments.

Keywords: mapping; pose graph optimization; positioning; robots; trajectory

1 INTRODUCTION

Simultaneous Localization and Mapping (SLAM) technology plays a crucial role in autonomous robot navigation and environment sensing, which is widely used in autonomous driving, unmanned aerial vehicles, service robots, and other fields. However, SLAM systems are often interfered by sensor noise, unstable conditions, and dynamic objects when facing complex and dynamic environments, resulting in decreased localization accuracy and unsatisfactory map building [1]. Pose Graph Optimization (PGO) is a key technique in SLAM back-end optimization, which is used to accurately estimate the position and pose of a robot or sensor in the environment. PGO is used to construct a pose graph by representing the different poses of the robot as nodes in the graph and using motion constraints between nodes [2, 3]. By minimizing the error between nodes, PGO can effectively correct the noise and drift in sensor data, and improve the positioning accuracy and graph construction quality. However, the traditional PGO method still suffers from error accumulation and poor global optimization when dealing with multi-sensor fusion and highly dynamic scenes [4]. Therefore, improving PGO techniques to make them more adaptable and stable in complex environments has become the current research focus in SLAM. In order to solve these problems, the research proposes a new SLAM model based on improved PGO. First, the Inertial Measurement Unit-Pre Integration (IMU-PI) theory is used in the front-end to optimize the position estimation of the LiDAR point cloud, which improves the accuracy of the sensor data and solves data aberration triggered by time difference. Second, the back-end incorporates the Cardinality Balanced Multi-target Multi-Bernoulli (CBMeMBer) filter to further optimize bitmap errors and reduce error accumulation in dynamic environments. The contribution of the study is to optimize the front-end processing of the LiDAR point cloud to enhance the positioning accuracy in dynamic environments through IMU-PI. Combined with CBMeMBer, the PGO back-end is optimized to effectively reduce error accumulation and enhance system robustness. Based on this new SLAM model, this study aims to address the limitations of traditional SLAM methods in complex dynamic environments, and further apply and develop this technology in autonomous robot navigation. This research is divided into four parts. The first part reviews existing research. The second part describes how the new SLAM model is constructed. The third part tests the performance of this new model. The last part summarizes the article. The technical line diagram is shown in Fig. 1.

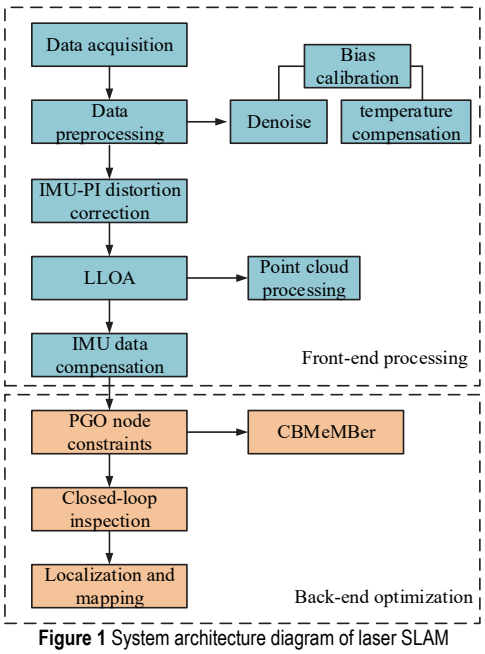

Figure 1 System architecture diagram of laser SLAM

2 LITERATURE REVIEW

In recent years, SLAM technology has been widely explored and applied in the fields of robot navigation and environment sensing. However, existing methods still face sensor noise, dynamic object interference, and error accumulation in dealing with complex environments. The study aims to provide a basis for the proposed method by critically analyzing the existing literature, identifying its strengths and weaknesses and proposing directions for improvement. The following section organizes the existing

literature by topics, focuses on the research progress in multi-sensor fusion, bitmap optimization, and dynamic object processing, and points out how this literature provides theoretical support for this research. Zhou H. et al. [5] proposed a SLAM model combining LiDAR and UWB sensors, which effectively accomplished accurate environment mapping under multi-robot collaboration. However, the method relied heavily on LiDAR data and was susceptible to sensor noise in highly dynamic environments, leading to accuracy degradation. Chen Y. et al. [6] optimized the bit position estimation in multi-robot SLAM by introducing a down-weighted Laplace matrix, which is a more limited method for handling dynamic objects in complex scenes, showing good performance on simulated and real datasets. Most existing multi-sensor fusion methods focus on static or semi-static environments and perform poorly when dealing with dynamic scenes. Therefore, this method optimizes LiDAR data by combining IMU pre-integration theory, which can better cope with sensor noise in dynamic environments, thereby improving the accuracy and robustness of data processing. Tsintotas K. A. et al. found that although there were accurate estimates of robot poses, there were still some issues with SLAM performance. To further leverage the advantages of SLAM technology, a low-complexity closed-loop inspection technique that combined visual perception with database accuracy optimization was proposed. Compared with traditional methods, this approach greatly improved computational efficiency and overall performance [7]. Zou Q. et al. observed that existing SLAM systems often exhibited fragile performance in tracking feature points, making reliable localization difficult. Therefore, a new 3D SLAM model that integrated Lidar point cloud technology with data mining was proposed. Experimental results indicated that this model provided more robust positioning in indoor environments and demonstrated superior compositional efficiency [8].

PGO, as an important tool for back-end optimization, has been widely used in recent years. Tian Y. et al. [9] proposed a sparse semi-definite relaxation method to improve the accuracy and speed of distributed PGO algorithms. However, the computational complexity of this method was high, which made it difficult to be applied to real-time scenarios. Bai F. et al. [10] solved the parameter optimization problem in sparse graphs by optimizing the cyclic space of PGO, but the convergence of this method in large-scale datasets was still limited. Most current methods were unable to effectively handle noise and dynamic object interference in complex environments. Therefore, this paper utilizes IMU pre-integration technology for motion distortion correction.

This technology is combined with the CBMeMBer filter to better handle uncertainty and multi-target tracking problems, in order to improve system robustness. In addition, in complex dynamic environments, sensor noise and dynamic objects have a great impact on SLAM performance. Yi X. et al. [11] investigated a SLAM system for fusing LiDAR and visual features.

A novel back-end optimization system that combined visual features with LiDAR point clouds was proposed for closed-loop detection and graph optimization. However, the method relied on a single wavelength LIDAR, which limited its performance in low-texture scenarios. Lin J. et al. [12] proposed a factor graph-based SLAM sensor framework that combined a Kalman filter and attitude map optimization to significantly improve the accuracy of both LIDAR and visual inertial odometry. However, the method still suffered from shortcomings in dealing with highly dynamic environments and complex noise, especially when dealing with multi-sensor data synchronization and dynamic objects.

In summary, although existing SLAM methods show good performance in static environments, they are not robust enough in dynamic environments and high-noise conditions, and most of them have too high computational complexity when facing complex sensor fusion problems. The method proposed in this study combines the advantages of these existing technologies, utilizing IMU pre-integration technology to optimize the position and pose of LiDAR point clouds, and using CBMeMBer filters to improve the accuracy and robustness of back-end PGO, thus overcoming the limitations of the existing methods in highly dynamic environments.

3 RESEARCH METHODOLOGY

In order to improve the efficiency and accuracy of robot localization and mapping, the study first optimizes the SLAM front-end mileage calculation. Based on the current popular laser SLAM, IMU-PI theory is introduced to improve the robot's state transitions to obtain more accurate preprocessed data. In addition, within the PGO framework, CBMeMBer is incorporated into the error calculation optimization. CBMeMBer combined with IMU further improves the robustness and accuracy of multi-sensor data, especially in multi-target tracking and complex environments. The combination of the two not only improves the accuracy of the front-end data, but also significantly improves the accuracy of global pose optimization at the back-end. Ultimately, this will develop a new and optimized robot SLAM model. The explanation of the key formula notation is shown in Tab. 1. In addition, a list of acronyms is shown in Tab. 2.

3.1 SLAM Front-end Laser Point Cloud Pose Optimization

SLAM was first introduced by Smith R. and Self M. et al. at the IEEE Robotics and Automation Conference in 1986, primarily to address the challenges of robot movement positioning and map construction [13, 14]. Over the years, SLAM has diversified into various categories, including visual SLAM, RGB-D SLAM, etc. Among them, laser SLAM stands out for its high accuracy and strong stability, enabling precise positioning and map construction even in complex environments. As a result, this research primarily focuses on laser SLAM. Furthermore, the typical robot software system is composed of an upper computer and a lower computer, each responsible for different aspects of task control and execution. The upper computer processes Lidar data in real-time, integrating it with existing map information to guide the robot in accurate positioning and mapping within unknown environments. The structural diagram of laser SLAM is illustrated in Fig. 2 [15, 16].

Table 1 Symbol interpretation table

Notation	Sense	Notation	Sense
X_i	Current robot status	$X_i^{'}$	Aberration-corrected robot positional locus
p_i	Robot coordinate system position	Z_t	Observed value
v_i	Robot movement speed	$h(x_t)$	Real value
q	Robot position status	e	Inaccuracies
b_a^T	Acceleration of robots	X_{t-1}	Postural state at time $t-1$
b_ω^T	Angular velocity displacement of the robot	X_t	Postural state at time t
p_j	Robot coordinate position in frame j	$\pi_{t/t-1}(X)$	Predictive probability distribution of the locus X at time t
v_j	Robot speed at frame j	p	State transfer probability
q_j	Robot position in frame j	$\pi_t(X)$	The posterior probability distribution of the robot's position X
Δt_{ij}	Length of time from frame i to frame j	z_k	Local observations
Δp_{ij}	Coordinate position of the transformed robot	f_j	Map features
Δv_{ij}	Converted robot speed	φ	Matching likelihood function
Δq	Transformed robot position	η	Normalizing factor
t_i	Moment of point i in the laser point cloud	X_t^*	Most likely robot position
T_i	The result of the positional transformation of the i th laser point	$r(X)$	Probability of the presence of a target at robot position X

Table 2 List of abbreviations for nouns

Abbreviations	Full name	Abbreviations	Full name
SLAM	Simultaneous Localization and Mapping	CARMEN	Cambridge University Robot Department Dataset
PGO	Pose Graph Optimization	VPD	Victoria Park Dataset
IMU-PI	Inertial Measurement Unit-Pre Integration	VIO-PGO	Visual-Inertial Odometry-Pose Graph Optimization
CBMeMBer	Cardinality Balanced Multi-target Multi-Bernoulli	LVI-SLAM	Laser-Vision-Inertial Simultaneous Localization and Mapping
LLOA	Lasers Lidar Odometry Algorithm	MSF-PGO	Multi-Sensor Fusion-Pose Graph Optimizatio
IMU	Inertial Measurement Unit	GBS	Graph-Based SLAM
FGS	Factor Graph SLAM	LOAM	Lidar Odometry and Multi-Sensor Fusion SLAM
RMSE	Root Mean Squared Error	*SSE*	Sum of Squared Errors
SD	Standard Deviation	/	/

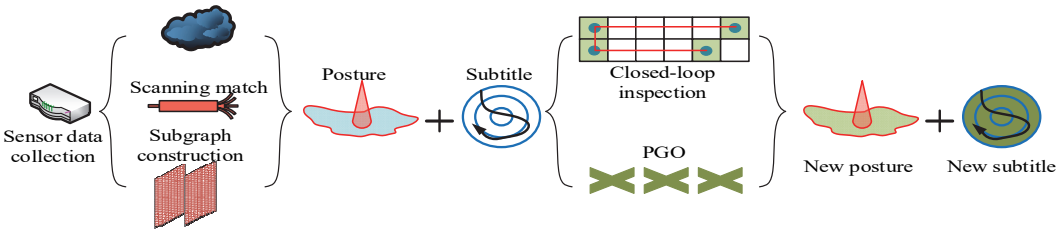

Figure 2 System architecture diagram of laser SLAM

In Fig. 2, the entire system is primarily divided into two components: front-end laser odometry and back-end optimization. The front-end takes the sensor data of the robot as input and outputs the robot's pose and sub-graphs. These are then fed into the back-end for optimization. After closed-loop inspection and PGO, the final robot pose and graph are produced. The existing front-end laser odometry commonly utilizes the Laser Lidar Odometry Algorithm (LLOA). This algorithm uses Lidar to scan the environment, capturing three-dimensional point cloud data, which are then processed and analyzed to deduce the robot's position and orientation [17, 18]. The structure of LLOA is depicted in Fig. 3.

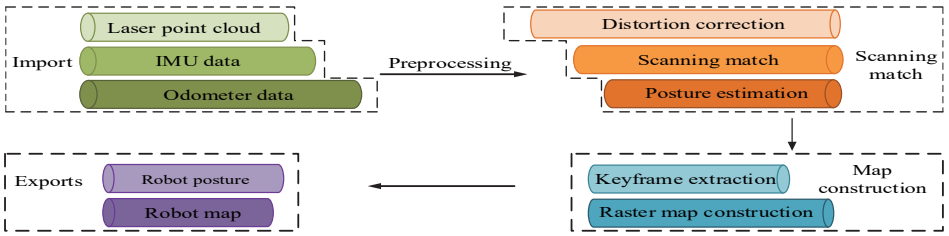

Figure 3 LLOA structure diagram

In Fig. 3, the LLOA is divided into three main stages: input, laser point cloud processing, scan matching and map construction, and output. First, laser point cloud data, Inertial Measurement Unit (IMU) data, and odometer data are input through sensors and processed using the LLOA. After processing, the data are scanned and matched with the database, and key-frames are generated through joint pose estimation. These key-frames are then used to construct a raster electronic map by extracting key-frame features, ultimately outputting the optimal robot pose and sub-graph. However, due to the time difference between the laser acquisition of the first and second data points, there are corresponding positional changes, leading to reduced positioning accuracy and composition quality of the robot [19, 20]. To address this issue, the study introduces distortion correction and IMU Pre-Integration (IMU-PI) theory for improvement. The initial robot state is defined in Eq. (1).

$$X_i = \left(p_i, v_i, q_i, b_a, b_a^T, b_\omega^T \right)^T \tag{1}$$

In Eq. (1), X_i represents the current state of the robot. p denotes the position of the robot within its coordinate system. v_i indicates the speed of the robot. q_i describes the pose of the robot. The terms b_a^T and b_ω^T represent the displacement caused by the acceleration and angular velocity of the robot, respectively. p_i, v_i and q_i are shown in Eq. (2).

$$\begin{cases} p_i = p_j + \sum_{k=i}^{j} \left[v_k \Delta t + \frac{1}{2} g^w \Delta t^2 + \frac{1}{2} q_k \left(\hat{a}_k - b_a \right) \Delta t^2 \right] \\ v_i = v_j + g^w \Delta t^2 + \sum_{k=i}^{j} q_k \left(\hat{a}_k - b_a \right) \Delta t \\ q_i = q_j \prod_{k=i}^{j} \mathrm{Exp} \left(\left(\hat{\omega}_k - b_\omega \right) \Delta t \right) \end{cases} \tag{2}$$

In Eq. (2), p_j, v_j, and q_j represent the coordinate position, velocity, and pose of the robot at frame j, respectively. Variables v_k, q_k, $\hat{a}_k$ and $\hat{\omega}_k$ represent the velocity, pose, acceleration vector, and angular velocity vector of the robot at frame k. g^w represents the current gravity vector, and Δt denotes the time interval between frames. Robot state transition with IMU-PI is a method to correct the time difference between sensors by integrating the acceleration and angular velocity data of IMU sensors, which is especially suitable for motion aberration correction in dynamic environments [21, 22]. The advantage of IMU-PI is that it can efficiently and accurately estimate the position of the robot without the need for frequent state calculations. Moreover, it can reduce the time difference caused by errors caused by inconsistent sensor frequencies, thereby improving the positioning accuracy and stability of the SLAM system. The IMU-PI performs the robot state transition calculation, as shown in Eq. (3).

$$\begin{cases} \Delta p_{ij} = p_j \left(p_j - p_i - v_i \Delta t_{ij} - \frac{1}{2} g^w \Delta t^2 \right) \\ \Delta v_{ij} = q_i \left(v_j - v_i - g^w \Delta t_{ij} \right) \\ \Delta q_{ij} = q_j q_i \end{cases} \tag{3}$$

In Eq. (3), Δt_{ij} represents the time length from frame i to frame j. Δp_{ij}, Δv_{ij} and Δq_{ij} represent the transformed robot coordinate position, velocity, and pose, respectively. Additionally, due to the different frequencies of LiDAR and IMU, such as multiple IMU data points appearing within a single LiDAR frame, there may be differences between the $k+1$ laser point cloud and IMU data input, resulting in decreased effectiveness of data preprocessing. To address this, the study assumes that the robot moves at a constant velocity within the k time frame and applies linear interpolation to transform the pose of the laser point relative to the previous point at any given moment, as described in Eq. (4).

$$T_i = \frac{t_{k+1} - t_i}{t_{k+1} - t_k} T_{t_k}^{t_{k+1}} \tag{4}$$

In Eq. (4), t_i represents the time of the i th point in the laser point cloud. T_i signifies the pose transformation result of the i th laser point relative to the previous laser point. Based on this transformation, the pose coordinates of the next laser point are determined, as shown in Eq. (5).

$$X_i^{'} = T_i X_i \tag{5}$$

In Eq. (5), $X_i^{'}$ represents the pose position after distortion correction. Following this calculation method, all laser point clouds are subjected to pose transformation, as illustrated in Fig. 4.

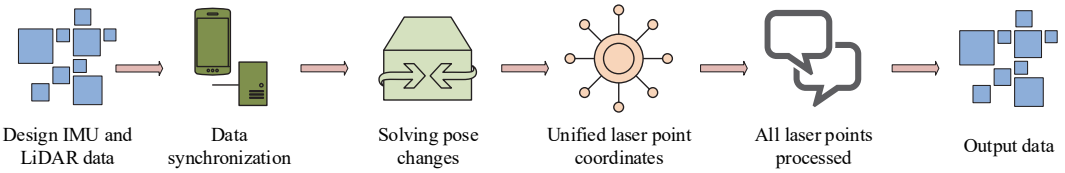

Figure 4 Laser point cloud pose optimization process

In Fig. 4, the process begins with the robot's sensors scanning and collecting IMU and Lidar data. Once these data are acquired, a linear interpolation method is applied to modulate the frequencies of the two data types. Next, the IMU-PI is used to preprocess and transform the robot's pose information, ensuring that the coordinate system of the laser points is unified. Following this method, all laser point cloud data undergo transformation, ultimately outputting the complete robot pose positioning result.

3.2 Optimization of Back-end Pose Diagram

After completing the pose calculation of the front-end laser point cloud data, the research shifts its focus to optimizing the back-end graph. This stage primarily enhances robot positioning and mapping by utilizing the data and constraints provided by the front-end. The PGO process is illustrated in Fig. 5 [23, 24].

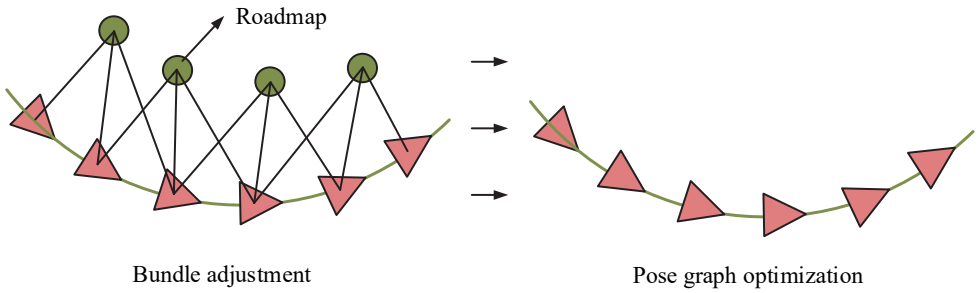

Figure 5 Schematic diagram of PGO model structure

In Fig. 5, PGO models the SLAM problem for robots using node-edge pose graphs. This approach does not account for the number or requirements of road signs along the pipeline, focusing solely on meeting the pose constraints. The method transforms the nonlinear optimization of least squares into a PGO problem by constructing a node-edge pose graph. In this model, each node represents the robot's pose at a specific moment, while the motion constraints between nodes describe their relative translation and rotation. This effectively converts the SLAM problem into a hyper-graph problem. At this stage, there is typically a discrepancy between the observed pose values at the robot's local points and the actual values, as expressed in Eq. (6).

$$e_{t,z} = Z_t - h(x_t) \tag{6}$$

In Eq. (6), Z_t represents the observed value. $h(x_t)$ represents the true value. $e_{t,z}$ signifies the error. When using a single PGO to estimate map features and pose, the dynamic changes of the robot may lead to a decrease in measurement accuracy. If the traditional least squares method is used for this estimation, it often only finds local optima and cannot reliably identify global optima. To address this limitation, the study introduces the Cardinality Balanced Multi-target Multi-Bernoulli (CBMeMBer) filter for optimization. CBMeMBer is a multi-objective tracking filter that operates within a Bayesian probability framework, allowing it to better manage uncertainty in estimation and the complexity of multi-objective environments, thereby reducing errors. CBMeMBer is able to effectively deal with target uncertainty in dynamic environments by estimating the states of multiple targets. Specifically, CBMeMBer uses Bernoulli's stochastic finite set theory to represent and process the states of multiple targets. It can estimate the number of targets at each moment and correlate it with the observation data, thereby reducing target recognition errors in position and attitude estimation [25, 26]. The advantage of CBMeMBer filter is that it can effectively handle complex multi-target environments. Especially when the SLAM system faces dynamic objects or environmental noise, it can optimize the estimation of the relative position between the robot and the environment through multi-objective probability distribution. By constantly updating the state of each target, CBMeMBer can improve the system's ability to recognize different targets and reduce the impact of sensor noise and environmental uncertainty on the accumulation of system errors. The principle of CBMeMBer is illustrated in Fig. 6.

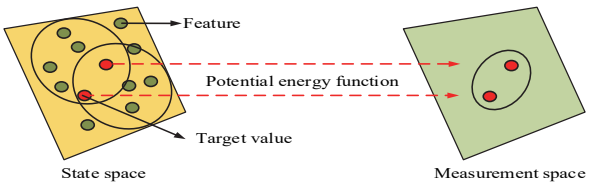

Figure 6 Schematic diagram of CBMeMBer principle

From Fig. 6, there are two sets of map spaces, each with different potential energy distributions. Three targets are identified in the real map and represented in the test map space as potential energy functions. The goal is to solve for the maximum potential energy function to achieve target detection and localization.

The multi-Bernoulli probability distribution of map features at time t is shown in Eq. (7).

$$\pi_{t|t-1}(X) = \int p(X_t|\, X_{t-1})\pi_{t-1}(X_{t-1})X_{t-1} \tag{7}$$

In Eq. (7), X_{t-1} and X_t denote the positional states at time $t-1$ and t, respectively. $\pi_{t|t-1}(X)$ signifies the predicted probability distribution of the pose X at time t given the previous time step. $p(X_t|\, X_{t-1})$ represents the probability of state transition from the previous state X_{t-1} to the current state X_t. $h(x_t)$ represents the

probability distribution of pose X at time $t-1$. The map update is shown in Eq. (8).

$$\pi_t(X)=\frac{p(Z_t|X_t)\pi_{t|t-1}(X_t)}{\int p(Z_t|X_t)\pi_{t|t-1}(X_t)X_t} \tag{8}$$

In Eq. (8), $\pi_t(X)$ represents the posterior probability distribution of the robot pose X at t. $p(Z_t|X_t)$ represents the likelihood function of the observed data Z_t in the pose state X_t. The data correlation likelihood function is expressed in Eq. (9).

$$p(Z_t|X_t)=\prod_{k\in Z_t}\sum_{j\in Features}\varphi(z_k,f_j) \tag{9}$$

In Eq. (9), z_k represents the local observation value. f_j represents map features. φ is the matching likelihood function. Additionally, the robot pose positioning optimized by the CBMeMBer algorithm is represented in Eq. (10).

$$\pi_t(X)=\eta\cdot p(Z_t|X_t)\int p(X_t|X_{t-1})\pi_{t-1}(X_{t-1})X_{t-1} \tag{10}$$

In Eq. (10), η represents the normalization factor. The fusion of pose updates is described in Eq. (11).

$$X_t^*=\arg\max_{X_t}\pi_t(X_t) \tag{11}$$

In Eq. (11), X_t^* represents the most likely robot pose, which is determined by maximizing the posterior probability distribution $\pi_t(X_t)$. The multi-objective estimation of the pose at this time is described in Eq. (12).

$$r_t(X)=1-e^{-\int\pi_t(X)X} \tag{12}$$

In Eq. (12), $r_t(X)$ represents the probability of the robot having a target at pose X.

In summary, the study enhances the LLOA in laser SLAM by incorporating IMU-PI. Additionally, it combines the CBMeMBer filter with PGO to propose a novel robot SLAM optimization model based on the improved PGO. The structure of this model is illustrated in Fig. 7.

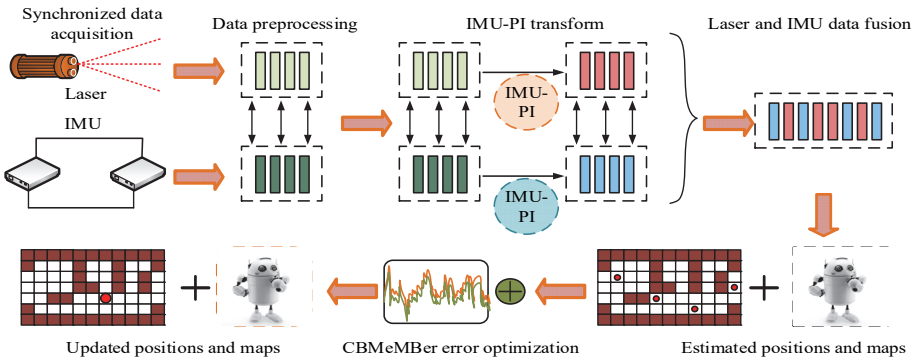

Figure 7 Structure diagram of LLOA-IMU-PI-CBMer-PGO

In Fig. 7, this model completes robot position estimation and environment mapping optimization through seven main stages. First, the LiDAR and IMU acquire data synchronously to ensure the time alignment of the laser point cloud data and the IMU data. At this stage, the IMU pre-integration technique is introduced to correct the motion aberration problem due to the inconsistency of the sensor frequencies, which improves the accuracy of the pose estimation. Next, the LIDAR and IMU data are preprocessed, including operations such as denoising, temperature compensation and bias calibration, to ensure that the quality of the sensor data meets the expected standards. Subsequently, after the initial LIDAR scanning data processing, the system performs the initial correction on the pose of the laser point cloud through IMU data and outputs key-frames and the sub-graphs. The key-frames ensure that the robot's motion state in known or unknown environments is accurately estimated by comparing them with the pre-collected data. In the back-end part, the CBMeMBer filter is introduced into the bitmap optimization process. Through multi-target tracking technique and Bayesian estimation framework, the system predicts and corrects the possible errors to reduce the uncertainty problem in multi-sensor data fusion. Finally, the CBMeMBer-optimized bitmap further reduces the error through graph optimization to generate more accurate bitmap estimation and environment map output.

4 RESULTS AND DISCUSSION

To validate its performance, a suitable experimental environment is established and the necessary experimental parameters are configured.

Then, the final model is validated through ablation testing, with trajectory prediction analysis conducted separately on pose estimation and localization at both the front-end and back-end stages. A simulation environment is also set up, and advanced algorithm models are introduced for comparison to demonstrate the mapping capability of the proposed model. Finally, localization metrics are tested across different models to assess and compare its performance.

4.1 Performance Testing of the Novel SLAM

To verify the performance of the model, a suitable experimental environment is established. The hardware setup includes an Intel Core i7 CPU, an NVIDIA GeForce GTX 1060 GPU, and 16 GB of memory. The operating system is Windows 10, and the software framework used is Python 3.8. The datasets employed are the Cambridge University Robot Department Dataset (CARMEN) and the Victoria Park Dataset (VPD). Among them, CARMEN is an open-source robot navigation dataset containing data from a wide range of indoor and outdoor environments, which is widely used for evaluating the performance of robots in tasks such as autonomous navigation, SLAM, and path planning. The data in this dataset comes from various sensors such as LiDAR, odometry, IMU, etc., which is suitable for evaluating the robustness of SLAM models under different sensor configurations and scenarios. VPD is a classic outdoor SLAM dataset collected by a mobile robot in Victoria Park. VPD is mainly used to test the performance of SLAM algorithms in open environments. The data includes point cloud data from LIDAR, sensor information such as IMU, GPS, etc. Robots are exposed to outdoor environments for extended periods of time to test the long-term robustness of SLAM algorithms and their ability to handle dynamic objects such as pedestrians and vehicles. The study first validates the final model through ablation testing, taking processing accuracy as the performance indicator. The test results are presented in Fig. 8.

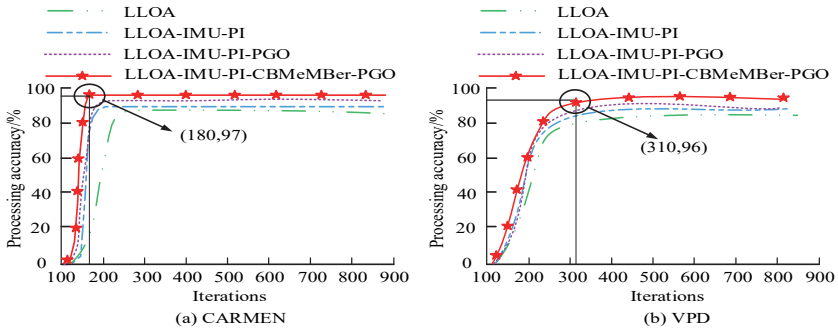

Figure 8 Model ablation test results

Fig. 8a displays the ablation test results of the final model on the CARMEN dataset, while Fig. 8b shows the results on the VPD dataset. As the number of iterations increased, each module achieved optimal performance in the later stages of training. The complete proposed model, i.e., LLOA-IMU-PI-PGO-CBMeMBer, performed best in terms of number of iterations and accuracy. On the CARMEN dataset, the minimum number of iterations was 180 and the processing accuracy was 97%. On the VPD dataset, the minimum number of iterations was 310, with 96% processing accuracy. In contrast, the model with CBMeMBer filter or IMU-PI pre-integration removed exhibited a significant performance degradation, with maximum accuracy of 94% and 92%, respectively. This suggests that the combination of these two modules is critical for the accuracy and robustness of the system. Because in the joint use of IMU pre-integration and CBMeMBer filter, IMU pre-integration can correct sensor time difference, significantly reduce the impact of motion distortion on attitude estimation, and make the system's performance more stable in dynamic environments. Meanwhile, the CBMeMBer filter effectively reduces the accumulation of errors in pose map optimization through its multi-target tracking and error prediction functions. Other algorithms, such as using only LLOA or deleting CBMeMBer combinations, are prone to accumulating errors when facing complex environments, leading to a decrease in accuracy. After completing the ablation test, the study conducts a comparative test on the front-end odometry of SLAM, introducing similar improved laser LLOA algorithms for testing, such as Visual-Inertial Odometry-Pose Graph Optimization (VIO-PGO), Laser-Vision-Inertial Simultaneous Localization and Mapping (LVI-SLAM), and Multi-Sensor Fusion-Pose Graph Optimization (MSF-PGO). Among them, VIO-PGO combines vision and inertial data for position estimation, which is able to provide a benchmark for comparison with the multi-sensor fusion scheme in this study. LVI-SLAM fuses laser, vision, and IMU, which can directly compare the effect of multi-sensor data processing. MSF-PGO performs position optimization through multi-sensor fusion, which is similar to the research methodology and provides a benchmark for performance comparison. The test results are presented in Fig. 9.

Fig. 9a shows the comparison results of actual mileage versus predicted mileage under the VIO-PGO algorithm. Fig. 9b presents the comparison results under the LVI-SLAM algorithm. Fig. 9c illustrates the comparison results under the MSF-PGO algorithm. Fig. 9d displays the comparison results for the proposed algorithm. The actual value represents the real time series data, i.e., the odometer data recorded by the sensor, which is the baseline value used to compare with the predicted results of the individual algorithms. The predicted values represent the time series values estimated by each SLAM algorithm based on the sensor data and the internal model. The comparison with the actual values shows the accuracy of each algorithm in terms of odometer prediction. In Fig. 9, in the 500 - 600 sample interval, the prediction errors of VIO-PGO and LVI-SLAM were large, especially in long time-series data, where the errors accumulated rapidly, leading to a significant decrease in positioning accuracy. In contrast, the proposed model performed stably over the entire test range with consistently low prediction errors, demonstrating its advantages in long-time localization. The reason why the proposed model outperforms the comparison algorithms is the collaborative work of IMU pre-integration technique and the CBMeMBer filter. This enables the proposed model

to effectively withstand the interference of sensor noise and external environmental changes in long-term positioning tasks, while other algorithms lack similar error correction mechanisms, resulting in a significant increase in prediction errors over time. However, in real-world scenarios, low-cost or low-precision sensors may lead to data distortion due to high noise and low precision, which in turn affects the overall performance of the system. Especially in dynamic environments, the decrease in sensor data quality may lead to error accumulation during IMU pre-integration, which in turn affects the accuracy of position estimation. Future research can explore robustness-based pre-processing algorithms, such as using adaptive filters or machine learning techniques to correct low precision sensor data. In addition, further research can be conducted on how to optimize the IMU pre-integration algorithm on low-cost hardware platforms to reduce its dependence on sensor quality and ensure its application in resource-constrained environments. After these tests, the study continued to evaluate the back-end data processing capabilities of the four algorithms, as shown in Fig. 10.

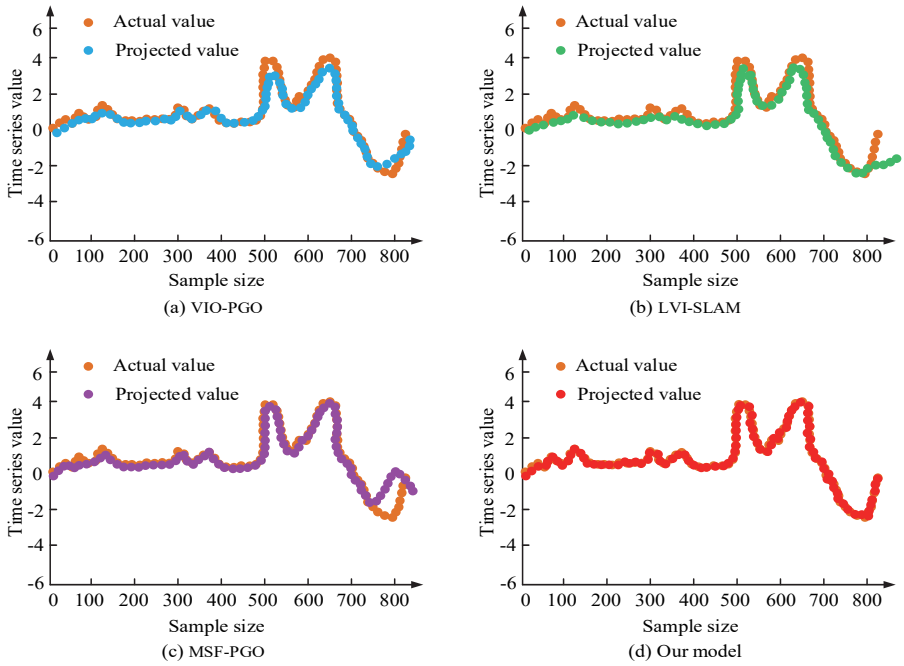

Figure 9 Mileage meter prediction results under different algorithms

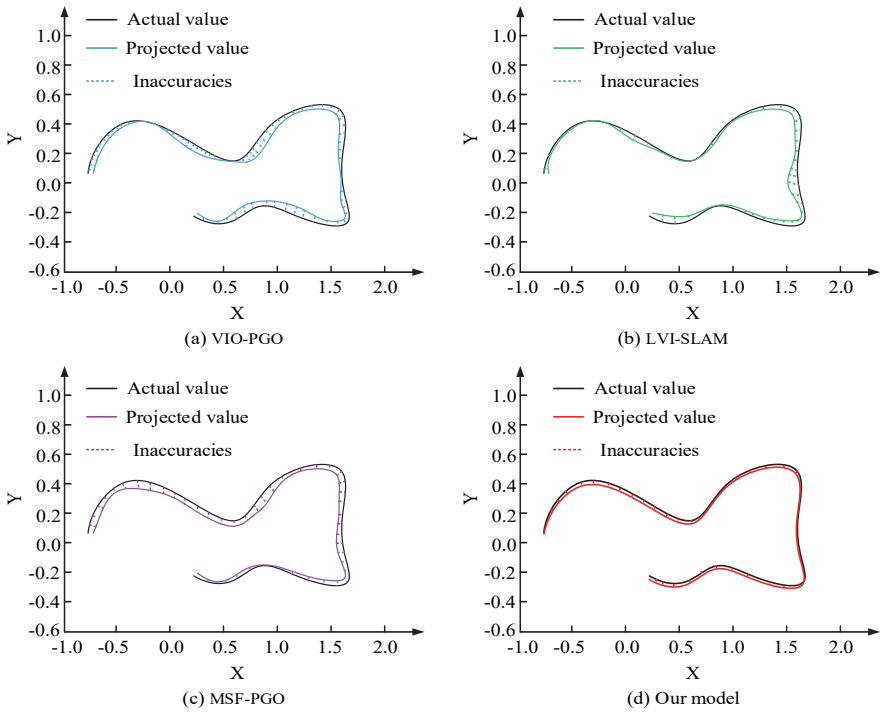

Figure 10 Comparison of back-end graph optimization trajectories for different algorithms

Fig. 10a presents the comparison results of back-end graph optimization trajectories under the VIO-PGO algorithm.

Fig. 10b shows the results under the LVI-SLAM algorithm. Fig. 10c illustrates the comparison results of the MSF-PGO algorithm. Fig. 10d displays the results of the proposed algorithm. Among them, the actual trajectory indicates the real robot motion trajectory in the 2D plane, which is a baseline trajectory generated by the sensor data. The predicted trajectory represents the robot motion trajectory estimated by each algorithm through graph optimization. The closer the trajectory is to the actual trajectory, the better the algorithm's position estimation and optimization ability.

From Fig. 10, the proposed model better overlapped the original trajectory in the two-dimensional plane, with a shorter trajectory error line, showing higher accuracy and consistency.

However, other algorithms, especially VIO-PGO and LVI-SLAM, have obvious trajectory deviations in some complex scenarios, with less trajectory overlapping area and longer error lines. The reason for this is that the proposed model effectively controls the error accumulation in multi-target environments through the CBMeMBer filter. The traditional PGO method is susceptible to noise and uncertainty when facing dynamic objects and multi-sensor data fusion, resulting in a significant increase in trajectory estimation error. In contrast, CBMeMBer is able to effectively reduce the uncertainty in the graph optimization process through multi-target tracking and Bayesian probabilistic modeling, thus significantly improving the accuracy of trajectory prediction.

4.2 Mobile Robot Simulation Testing

To verify the effectiveness of the designed model, the study sets the laser frequency to 20 Hz, the IMU frequency to 35 Hz, and the odometer frequency to 120 Hz. Additionally, the Freigurg dataset provided by the Computing Department of the Munich Institute of Technology in Germany is used as the source of real data scenarios. This dataset includes two categories: Fre1_desk and Fre2_desk. The measurement distance for Fre1_desk is 9.26 meters, with a frame rate of 627 fps and a testing duration of 25.47 seconds. For Fre2_desk, the testing distance is 11.25 meters, with a frame rate of 1135 fps and a testing duration of 19.86 seconds. A real scenario from the two subsets, Fre1_desk and Fre2_desk, is selected as the testing environment. The study provides real scene diagram of experimental environment used for testing, as shown in Fig. 11.

(a) Scene 1 (b) Scene 2

Figure 11 Simulation testing environment diagram

Fig. 11a shows Simulation Testing Environment 1, while Fig. 11b displays Simulation Testing Environment 2. Taking these two simulation test maps as the background, the study introduces advanced models of the same type as the proposed method for comparison. These models include Graph-Based SLAM (GBS), Factor Graph SLAM (FGS), and Lidar Odometry and Multi-Sensor Fusion SLAM (LOAM). Among them, GBS, as a classical graph optimization method, is chosen to compare with the accuracy and computational complexity of the proposed model in graph optimization. FGS excels in optimization efficiency, which is compared with the proposed graph optimization method. LOAM performs well in dynamic environments, which can be compared with the multi-sensor fusion effect of the research method. The test results are converted into a two-dimensional format for analysis. Taking Scenario 1 as an example, the mapping results of the four algorithms are shown in Fig. 12.

Fig. 12 presents the scene mapping effects of the four algorithms. Fig. 12a shows the results under GBS, Fig. 12b displays the results under FGS. Fig. 12c illustrates the mapping under Lidar Odometry and LOAM. Fig. 12d demonstrates the results using the proposed algorithm in this study. In Fig. 12, the proposed model outperformed the other three algorithms in terms of building accuracy and detail performance. Especially when dealing with complex structures, such as doors and walls, the proposed model accurately recognized and modeled them, while the GBS and FGS algorithms had higher misrecognition rates, and some of the structure build-ups are blurred or incomplete. The significant accuracy advantage demonstrated by the proposed model during the mapping process is mainly due to the optimization of front-end data processing by IMU pre-integration technology and the enhancement of back-end pose graph optimization by CBMeMBer filter. IMU pre-integration improves the temporal accuracy of the laser point cloud and ensures the high fidelity of the data in the dynamic scene, while the CBMeMBer further reduces the mapping process uncertainty and optimizes the position estimation results. In contrast, algorithms such as GBS and FGS lack an effective mechanism to cope with uncertainty when dealing with complex structures, resulting in insufficient map building accuracy. However, the combination of IMU pre-integration and CBMeMBer exacerbates the computational load of the system in complex dynamic scenes, affecting the real-time

performance. Subsequently, the combination of parallel computing, graph processing gas pedal or deep learning gas pedal can be considered to improve the overall processing efficiency and ensure that the system maintains a faster response speed in scenarios with high real-time requirements. To further evaluate the localization performance of the four methods, the study takes average running time, Root Mean Squared Error (*RMSE*), Sum of Squared Errors (*SSE*), and Standard Deviation (*SD*) as reference indicators. The results are presented in Tab. 3.

Table 3 Indicator test results for four algorithms

Scenes	Model	Average running time / s	*RMSE* / m	*SSE* / m^2	*SD* / m	*P*	Confidence interval (95%)	Standard error (*SE*)
1	GBS	44.27	7.99	664.82	7.86	0.002	7.55 - 8.43	0.22
	FGS	36.58	6.81	524.71	6.58	0.001	6.45 - 7.17	0.18
	LOAM	31.77	4.35	311.02	4.21	0.003	4.12 - 4.58	0.12
	Our model	21.06	2.01	189.52	1.86	0.002	1.92 - 2.10	0.06
2	GBS	51.36	6.87	654.73	6.54	0.001	6.43 - 7.31	0.21
	FGS	44.28	5.24	487.69	6.04	0.002	4.97 - 5.51	0.15
	LOAM	39.29	3.17	324.18	5.21	0.002	3.05 - 3.29	0.09
	Proposed model	21.11	1.86	210.17	2.18	0.003	1.76 - 1.96	0.05

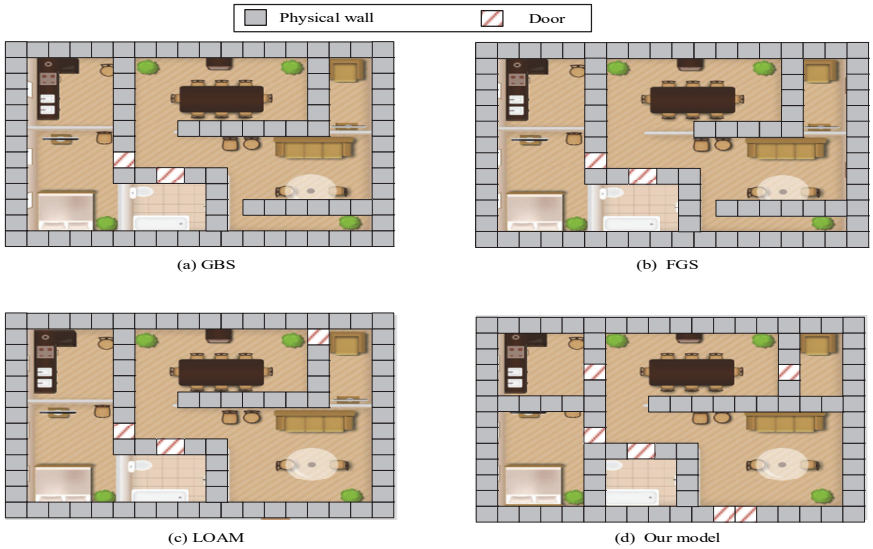

Figure 12 Comparison of mapping effects of different algorithms

According to Tab. 3, the average running time, *RMSE*, *SSE*, and *SD* values for GBS and FGS were lower than those for LOAM and the proposed model in both real scenarios. LOAM, leveraging its laser odometry and multi-sensor data collection, improved the accuracy of robot positioning operations compared with GBS and FGS. The minimum average running time of LOAM is 31.77 seconds, the minimum *RMSE* was 3.17 meters, the minimum *SSE* was 311.02 square meters, and the minimum *SD* was 4.21 meters. However, the proposed model outperformed all others, with the shortest running time of 21.06 seconds, the lowest *RMSE* of 1.86 meters, the lowest *SSE* of 189.52 square meters, and the lowest *SD* of 1.86 meters. In addition, the *P*-values in the table indicated that the differences between the models were statistically significant, further validating the significant improvement in performance of the model. The *RMSE* of the proposed model was 2.01 meters and 1.86 meters for the two scenarios, and its confidence intervals were 1.92 - 2.10 meters and 1.76 - 1.96 meters, respectively, which were significantly better than the other algorithms, and the *SE* values were small, indicating that the results were less volatile and had higher stability. Therefore, the SLAM optimization model significantly outperforms the existing methods in terms of positioning accuracy, operational efficiency and error control.

5 CONCLUSION

In this study, a SLAM optimization technique integrating IMU pre-integration and CBMeMBer filter was proposed, which significantly improved the robot's localization accuracy and mapping efficiency in dynamic environments. It especially performs well when dealing with complex, multi-target dynamic environments. Although the proposed SLAM optimization method performs well in dynamic environments, it still has some limitations. First, the method has a high dependence on high-quality IMU and LiDAR data, and low-cost sensors may lead to accuracy degradation. Second, the high computational complexity of the CBMeMBer filter may affect real-time performance in resource-constrained devices. The impact of this research on the SLAM field is to improve the robustness of traditional SLAM in dealing with dynamic objects and sensor noise, and to enhance the robot's navigation accuracy and error control in complex environments. Especially in multi-target tracking and dynamic object processing, the method demonstrates

significant advantages. Future research should further reduce the dependence on high-precision sensors and computational resources, optimize the computational efficiency of the CBMeMBer filter, and explore the integration of deep learning and SLAM techniques to enhance the model's adaptability to complex environments. The method has a wide range of potential real-world applications, including unmanned vehicles, UAV navigation, industrial automation, and intelligent service robots, especially in dynamic environments with improved navigation and obstacle avoidance capabilities, making it an important application prospect in future intelligent scenarios.

6 REFERENCE

[1] Tamantini, C., Luzio, F. S. D., Cordella, F., Pascarella, G., Agro. F., & Zollo L. (2021). A Robotic Health-Care Assistant for COVID-19 Emergency: A Proposed Solution for Logistics and Disinfection in a Hospital Environment. *IEEE Robotics & Automation Magazine*, *28*(1), 71-81. https://doi.org/10.1109/MRA.2020.3044953

[2] Wei, W., Zhu, X., & Wang, Y. (2022). Novel robust simultaneous localization and mapping for long-term autonomous robots. *Frontiers of Information Technology & Electronic Engineering*, *23*(2), 234-245. https://doi.org/10.1631/FITEE.2000358

[3] Bavle, H., Sanchez-Lopez, J. L., & Shaheer, M. (2023). S-graphs+: Real-time localization and mapping leveraging hierarchical representations. *IEEE Robotics and Automation Letters*, *8*(8), 4927-4934. https://doi.org/10.1109/LRA.2023.3290512

[4] Tian, Y., Chang, Y., Arias, F. H., Granda, C., How, J., & Carlone, L. (2021). Kimera-Multi: Robust, Distributed, Dense Metric-Semantic SLAM for Multi-Robot Systems. *IEEE Transactions on Robotics*, *38*(4), 2022-2038. https://doi.org/10.1109/TRO.2021.3137751

[5] Zhou, H., Yao, Z., Zhang, Z., Liu, P., & Lu, M. (2022). An Online Multi-Robot SLAM System Based on Lidar/UWB Fusion. *IEEE Sensors Journal*, *22*(3), 2530-2542. https://doi.org/10.1109/JSEN.2021.3136929

[6] Chen, Y., Zhao, L., Zhang, Y., Huang, S., & Dissanayake, G. (2021). Anchor Selection for SLAM Based on Graph Topology and Submodular Optimization. *IEEE Transactions on Robotics*, *38*(1), 329-350. https://doi.org/10.1109/TRO.2021.3078333

[7] Tsintotas, K. A., Bampis, L., & Gasteratos, A. (2021). Tracking-DOSeqSLAM: A dynamic sequence-based visual place recognition paradigm. *IET Computer Vision*, *15*(4), 258-273. https://doi.org/10.1049/cvi2.12041

[8] Zou, Q., Sun, Q., Chen, L., Nie, B., & Li, Q. (2021). A Comparative Analysis of LiDAR SLAM-Based Indoor Navigation for Autonomous Vehicles. *IEEE Transactions on Intelligent Transportation Systems*, *23*(7), 6907-6921. https://doi.org/10.1109/TITS.2021.3063477

[9] Tian, Y., Khosoussi, K., Rosen, D. M., & How, J. P. (2021). Distributed Certifiably Correct Pose-Graph Optimization. *IEEE Transactions on Robotics*, *37*(6), 2137-2156. https://doi.org/10.1109/TRO.2021.3072346

[10] Bai, F., Vidal-Calleja, T., & Grisetti, G. (2021). Sparse Pose Graph Optimization in Cycle Space. *IEEE Transactions on Robotics*, *37*(5), 1381-1400. https://doi.org/10.1109/TRO.2021.3050328

[11] Yi, X., Zhou, Y., Habermann, M., Golyanik, V., Pan, S., Theobalt, C., & Xu, F. (2023). Egolocate: Real-time motion capture, localization, and mapping with sparse body-mounted sensors. *ACM Transactions on Graphics (TOG)*, *42*(4), 1-17. https://doi.org/10.1145/3592099

[12] Lin, J., Zheng, C., Xu, W., & Zhang, F. (2021). R2LIVE: A Robust, Real-time, LiDAR-Inertial-Visual tightly-coupled state Estimator and mapping. *IEEE Robotics and Automation Letters*, *6*(4), 7469-7476. https://doi.org/10.1109/LRA.2021.3095515

[13] Hu, H., Pan, W., Gao, S., & Tang, X. (2023). An Optimization-Based Time-Optimal Velocity Planning for Autonomous Driving. *Studies in Informatics and Control*. *32*(3),45-56. https://doi.org/10.24846/v32i3y202304

[14] Lim, S., Jung, J., Kim, S. C., & Lee, S. (2021). Radar-Based Ego-Motion Estimation of Autonomous Robot for Simultaneous Localization and Mapping. *IEEE sensors journal*, *21*(19), 21791-21797. https://doi.org/10.1109/JSEN.2021.3101893

[15] Mokayed, H., Quan, T. Z., Alkhaled, L., & Sivakumar, V. (2023). Real-time human detection and counting system using deep learning computer vision techniques. *Artificial Intelligence and Applications*. *1*(4), 221-229. https://doi.org/10.47852/bonviewAIA2202391

[16] Iancu, D. T. & Florea, A. (2023). An Improved Vehicle Trajectory Prediction Model Based on Video Generation. *Studies in Informatics and Control*, *32*(1), 25-36. https://doi.org/10.24846/v32i1y202303

[17] Ding, H., Zhang, B., Zhou, J., Yan, Y., Tian, G., & Gu, B. (2022). Recent developments and applications of simultaneous localization and mapping in agriculture. *Journal of Field Robotics*, *39*(6),956-983. https://doi.org/10.1002/rob.22077

[18] Morales, M., Tapia, L., & Gildardo, S. A. (2021). A resource-aware approach to collaborative loop-closure detection with provable performance guarantees. *The International Journal of Robotics Research*, *40*(10-11),1212-1233. https://doi.org/10.1177/0278364920948594

[19] Placed, J. A., Strader, J., & Carrillo, H. (2023). A survey on active simultaneous localization and mapping: State of the art and new frontiers. *IEEE Transactions on Robotics*, *39*(3), 1686-1705. https://doi.org/10.1109/TRO.2023.3248510

[20] Lajoie, P. Y. & Beltrame, G. (2023). Swarm-slam: Sparse decentralized collaborative simultaneous localization and mapping framework for multi-robot systems. *IEEE Robotics and Automation Letters*, *9*(1), 475-482. https://doi.org/10.1109/LRA.2023.3333742

[21] Deng, T., Xie, H., & Wang, J. (2023). Long-term visual simultaneous localization and mapping: Using a Bayesian persistence filter-based global map prediction. *IEEE Robotics & Automation Magazine*, *30*(1), 36-49. https://doi.org/10.1109/MRA.2022.3228492

[22] Nazari, M. A., Seco-Granados, G., & Johannisson, P. (2023). MmWave 6D radio localization with a snapshot observation from a single BS. *IEEE Transactions on Vehicular Technology*, *72*(7), 8914-8928. https://doi.org/10.1109/TVT.2023.3246303

[23] Zhuang, Y., Wang, B., & Huai, J. (2023). 4d iriom: 4d imaging radar inertial odometry and mapping. *IEEE Robotics and Automation Letters*, *8*(6), 3246-3253. https://doi.org/10.1109/LRA.2023.3266669

[24] Sharafutdinov, D., Griguletskii, M., & Kopanev, P. (2023). Comparison of modern open-source visual SLAM approaches. *Journal of Intelligent & Robotic Systems*, *107*(3),43-47. https://doi.org/10.1007/s10846-023-01812-7

[25] Azagra, P., Sostres, C., Ferrández, Á., Luis, R., Clara, T., O. León, B., Javier, M., David, R., Víctor, M. B., Juan, J. G. R., Richard, E., Julia, L., Cristina, O., Javier, C., Juan, D. T., Ana,

C. M., Angel, L., & José, M. M. M. (2023). Endomapper dataset of complete calibrated endoscopy procedures. *Scientific Data*, 2023, *10*(1), 671-674. https://doi.org/10.1038/s41597-023-02564-7

[26] Waqas, A., Kang, D., & Cha, Y. J. (2024). Deep learning-based obstacle-avoiding autonomous UAVs with fiducial marker-based localization for structural health monitoring. *Structural Health Monitoring*, *23*(2), 971-990. https://doi.org/10.1177/14759217231177314

Contact information:

Xuan Rong ZHAO
(Corresponding author)
Si Chuan top ITVocational Institute,
Cheng Du, Si Chuan, China, 611743
E-mail: zhaoxuanrong@scetop.cn