Mapping Method Based on Improved Cartographer Algorithm

Huilin Zhang
School of Computer Science and Engineering
Xi'an Technological University
Xi'an, China
E-mail: 3952226890@qq.com

Xiaohui Su
School of Computer Science and Engineering
Xi'an Technological University
Xi'an, China
E-mail: 287099144@qq.com

Yao Zhang
School of Computer Science and Engineering
Xi'an Technological University
Xi'an, China
E-mail: 2044828475@qq.com

Shuping Xu
School of Computer Science and Engineering
Xi'an Technological University
Xi'an, China
E-mail: 563937848@qq.com

***Abstract*—This research introduces an enhanced version of the Cartographer algorithm for laser-based simultaneous localization and mapping (SLAM) to address common challenges in traditional LiDAR methods, such as incomplete point cloud features, low-quality data, and pose drift induced by measurement noise. The proposed approach incorporates an Adaptive Unscented Kalman Filter (AUKF) during sensor fusion, which effectively predicts and updates sensor measurements with adaptive noise optimization to mitigate interference during pose estimation. Additionally, in the point cloud processing stage, we improve upon standard voxel filtering by integrating weighted secondary screening inspired by particle filtering concepts, resulting in reduced redundancy and enhanced point cloud accuracy. Experimental evaluations were performed in both indoor and outdoor settings compared to the original Cartographer algorithm. Results from indoor tests showed a notable decrease of 17.2% in absolute translation error and 30.1% in absolute rotation error. Similarly, outdoor experiments demonstrated improvements of 25.8% and 28.9%respectively. These quantitative findings validate the effectiveness of the proposed algorithm in achieving significantly lower errors and superior mapping accuracy, showcasing its practical applicability.**

Keywords-Cartographer; Sensor Fusion; Adaptive Unscented Kalman Filter; Improved Voxel Filtering (Key Words)

I. INTRODUCTION

SLAM, or Simultaneous Localization and Mapping, is a groundbreaking technology that was first introduced by Smith and his colleagues back in 1988. This innovative concept involves the use of various sensors, such as cameras, LiDAR, IMU, and wheel odometers, mounted on robots to gather environmental data. Through sophisticated filtering or fusion methods, real-time map construction and precise robot position estimation are achieved. What sets SLAM apart is its ability to allow robots to navigate and explore their surroundings without any prior map information, thanks to advanced algorithms. This technology serves as the cornerstone of robotic advancements, with different variations like visual SLAM, LiDAR SLAM, and multi-sensor SLAM offering diverse applications. In particular, LiDAR SLAM remains a pivotal field with popular algorithms like Gmapping, Hector, and Karto still widely used. Of these, Cartographer stands out as a powerful graph optimization-based LiDAR SLAM algorithm developed by Google in 2016, boasting support for multiple sensors like IMU, wheel odometers, and 2D/3D LiDAR for simultaneous data collection and processing.

Traditional cartographers typically rely on the conventional Unscented Kalman Filter (UKF) approach to predict and fuse data from various sensors like IMU, wheel odometers, and LiDAR in order to create observation models for submaps. Unfortunately, this approach is not without its challenges. Issues such as delayed pose detection and LiDAR detection errors often plague large-scale model fusion during the mapping process. Furthermore, sensor noise can greatly impact the accuracy of the measurements obtained, leading to potential inaccuracies in the final map. The traditional UKF algorithm assumes that sensor noise follows a Gaussian distribution, which may not always hold true in real-world scenarios. Additionally, manual selection of sampling points introduces uncertainty into the mapping process, as different sampling points can yield varying results. Several studies have proposed solutions to address these issues and improve the accuracy of mapping techniques. One approach suggested in a specific study involves a multi-strategy pose estimation method that combines sensor fusion with visual information to enhance mapping accuracy. However, this method heavily relies on the extraction of visual feature points and may struggle to provide precise pose motion information. Another study introduced a multi-level distance scheduler approach to optimize point cloud recognition and LiDAR data acquisition speed within the Cartographer algorithm. While this method aims to improve the identification and replacement of point cloud data to remove pose disturbances, challenges such as low point cloud quality and data redundancy persist. In an effort to enhance the traditional UKF algorithm, a different study updated the covariance matrix of sensor noise to predict noise trends and utilized fading factors to suppress filter divergence. While this adjustment helped to reduce the algorithm's reliance on Gaussian noise distribution for prediction, challenges with precise sampling point selection and remaining uncertainties still persisted. Another proposed method focused on eliminating point cloud distortion and enhancing quality by fusing wheel odometer and laser point cloud data. However, issues related to point cloud redundancy remained unresolved. Further improvements were suggested with the introduction of pass-through and radius filtering techniques to process point cloud information effectively. By fusing these filters and incorporating velocity integral pose fusion for IMU and wheel odometer data, problems like drift and slippage during mapping were addressed. Despite these advancements, the overall enhancement in mapping quality was modest. Efforts to refine the back-end graph optimization component of the Cartographer algorithm have also been made to bolster mapping accuracy. While these improvements have demonstrated enhancements in submap fusion and construction, the acquisition of sensor information at the front-end remains unchanged. Additionally, enhancements to radius filtering have helped to eradicate large-scale noise errors during fusion, thereby improving point cloud quality and maintaining features. Nonetheless, the statistical analysis of point cloud parameters continues to result in substantial data redundancy that impacts mapping outcomes.

The existing methods for addressing the low performance of Cartographer's sensor information acquisition have shown some improvements, but there are still unresolved issues in sensor information fusion and LiDAR point cloud acquisition. This article introduces a novel approach to overcome the limitations of the

Cartographer algorithm. The proposed method involves using an adaptive Unscented Kalman Filter (UKF) to process data from IMU and wheel odometers separately. By predicting and updating the data using covariance matrices and performing a weighted fusion of both sources, this approach effectively improves sensor noise optimization. Additionally, the method involves selecting appropriate discrete sampling data based on noise conditions, thereby avoiding the limitations of traditional UKF. In the context of point cloud information processing, the article introduces an enhanced voxel filtering technique that incorporates particle filtering concepts. This technique helps to identify and filter out redundant and low-quality point cloud information, ultimately enhancing the quality of the point cloud data and improving the accuracy of mapping processes. These advancements contribute to enhancing the overall effectiveness and precision of sensor information acquisition in Cartographer systems.

II. CARTOGRAPHER ALGORITHM ANALYSIS

A. Cartographer Algorithm

The process of cartography can be broken down into two main components, front-end local optimization and back-end global optimization. In the front-end phase, poses are initialized and updated using data from IMU and wheel odometers. Submaps are then constructed and updated with information gathered from LiDAR point clouds. This phase involves creating and updating submaps based on radar scan data. The back-end phase focuses on global optimization, continually updating and merging submaps using Ceres to correct any cumulative errors that may arise. These two parts work together until no new data or submaps are detected, ensuring accurate and reliable mapping results. The detailed workflow can be found in Figure 1 for a clear visualization of the process.

In Cartographer, the LiDAR-collected data is represented as H in a certain frame, with the collection quantity denoted as k, $H=\{h_k\}$, $k=1,2,\ldots,K,h_k \in \mathrm{R}^2$. The LiDAR pose is represented by $\xi=\left(\xi_x,\xi_y,\xi_\theta\right)$,where ξ_x and ξ_y are the specific pose information positions in the point cloud coordinate system, i.e., offsets, and ξ_θ is the yaw angle. The mapping process from the point cloud coordinate system to the specific submap can be expressed by Equation (1).

$$T_\xi H=\begin{pmatrix}\cos\xi_\theta & -\sin\xi_\theta\\ \sin\xi_\theta & \cos\xi_\theta\end{pmatrix}H+\begin{pmatrix}\xi_x\\ \xi_y\end{pmatrix} \tag{1}$$

Continuous mapping is an ongoing process in Cartographer, resulting in the creation of a unified probability grid. This grid consists of submaps, with each grid cell indicating the likelihood of a point being occupied. The optimization of these submaps is carried out by a Ceres-based matcher, which conducts pose recognition on LiDAR-scanned point clouds. Utilizing the least squares method, iterations are employed to determine the optimal pose state with the highest probability. This process involves the use of bicubic interpolation functions to fine-tune the submaps. Ultimately, the goal is to refine the mapping process and improve the accuracy of the generated maps M_{smooth} within Cartographer.

$$\arg\min_{\xi}\sum_{k=1}^{K}\left[1-M_{\text{smooth}}\left(T_\xi H\right)\right]^2 \tag{2}$$

The Cartographer algorithm enhances accuracy and efficiency in matching by optimizing the best matching solution. Loop closure constraints are utilized through branch and bound methods to enhance the detection efficiency of loop closures. In this process, W represents the search window range, and M_{nearest} symbolizes the nearest neighbor difference function as detailed in Equation (3). This approach ultimately leads to improved results in mapping and navigation systems.

$$\xi^* = \arg\max_{\xi \in W} \sum_{k=1}^{K} M_{\text{nearest}}\left(T_\xi H\right) \tag{3}$$

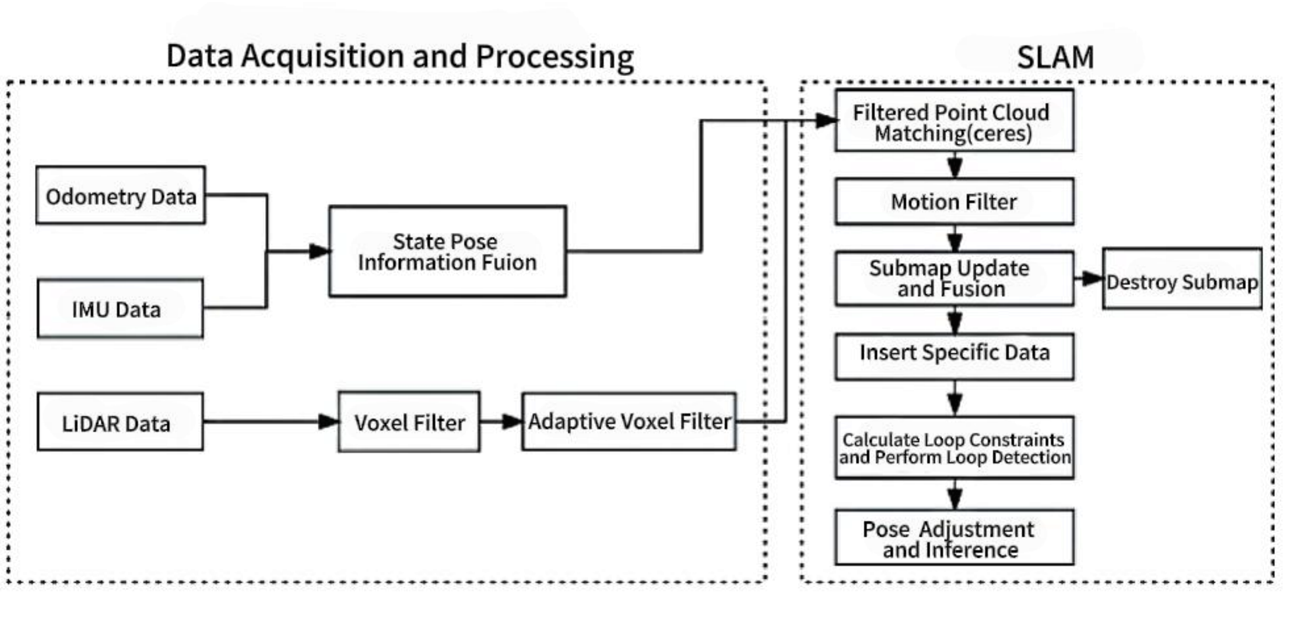


Figure 1. Cartographer Algorithm Workflow.

B. Improved Cartographer Algorithm Workflow

The enhancements to the Cartographer algorithm discussed in this paper involve two key aspects. The original algorithm combined IMU and odometer data through pose information fusion, resulting in significant noise and errors during subsequent fusion with LiDAR data. To address this, an adaptive unscented Kalman filter method is proposed to weight the fusion of the two data sources. The IMU's angular velocity, linear velocity, and yaw angle data are used to predict and update specific pose estimates with wheel odometer data. This approach includes adaptive adjustments to control noise levels after fusion, reducing errors in the process. Additionally, the original algorithm utilized voxel filtering to create a 3D grid when processing LiDAR-collected point clouds, choosing a centroid to represent each block of points. However, this method often led to redundancies and poor point cloud quality. To improve this, a new voxel filtering technique is introduced in this paper, integrating the concept of weighting from particle filtering. Point cloud information is weighted based on repetition and proportion levels before undergoing a secondary screening process. The secondarily filtered point cloud's density and state information are then updated in the algorithm to ensure higher quality and reliability during mapping. The revised Cartographer algorithm workflow, as detailed in Figure 2, showcases these advancements in data fusion and point cloud filtering techniques. Through these improvements, the algorithm can produce more accurate and precise mapping results, reducing errors and enhancing overall performance in various applications.

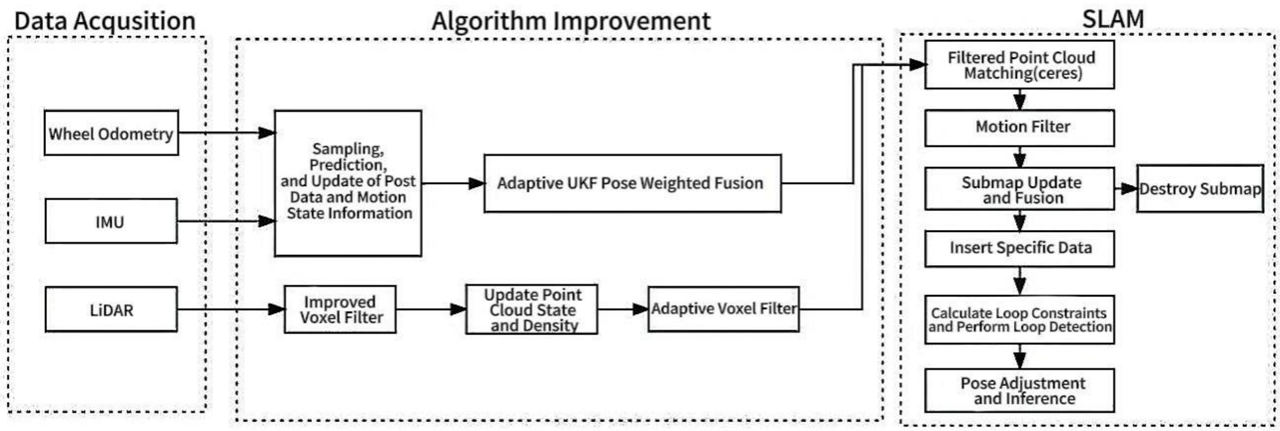


Figure 2. Improved Cartographer Algorithm Workflow

III. SPECIFIC IMPROVEMENT METHODS

A. Adaptive UKF Fusion

When integrating data from IMU and wheel odometer sensors, noise is a significant factor affecting the fusion process. The original Cartographer algorithm utilizes the Unscented Kalman Filter (UKF), which assumes a zero-mean Gaussian distribution for noise. However, real-world data often deviates from this distribution, resulting in errors in the fused data. To address this issue, this study introduces the Adaptive Unscented Kalman Filter (AUKF) method for fusing IMU and wheel odometer data. By incorporating adaptive techniques, the AUKF is able to effectively handle non-Gaussian noise, thereby improving the accuracy of the fused data. This approach enhances the reliability and robustness of sensor fusion algorithms in navigation systems.

Step 1: Initialization

Initialize the system state equation and observation vector:

$$\begin{cases} Z_{k+1} = h\left(X_{k+1}\right) + \psi_k \\ X_{k+1} = f\left(X_k\right) + \omega_k \end{cases} \tag{4}$$

Where X_{i+1} represents the state vector, Z_{i+1} represents the observation vector, $f(\ )$ and $h(\ )$ denote the state equation and measurement equation respectively, and ω_k and ψ_i represent the noise means.

Initialize the filter:

$$\begin{cases} \hat{X}_0 = E\left(X_0\right) \\ P_0 = E\left[\left(X_0 - \hat{X}_0\right)\left(X_0 - \hat{X}_0\right)^T\right] \end{cases} \tag{5}$$

Assume $\hat{X}_0$ as the initial state expectation, X_0 as the initial value, and P_0 as the initial covariance matrix.

Step 2: State Prediction

Combining the noise distribution characteristics of the IMU and odometer, a corresponding set of Sigma points is generated through Gaussian integration, where the IMU input vector is u_k :

$$X_k^i = f\left(X_{k-1}, u_k, w_k\right), i = 0,1,\ldots,2n \tag{6}$$

Given the IMU predicted state vector $\hat{X}_k^i$ and its corresponding covariance matrix p_u the

Sigma point set can be selected according to the following method:

$$X_k^i = \begin{cases} \hat{X}_k^0, & i=0 \\ \hat{X}_k^i + \left(\sqrt{(n+\lambda)P_u}\right)_i, & i=1,2,\ldots,n \\ \hat{X}_k^i - \left(\sqrt{(n+\lambda)P_u}\right)_i, & i=n+1,n+2,\ldots,2n \end{cases} \tag{7}$$

Where X_k^i represents the selected Sigma points, and λ denotes the scale adjustment factor.

Subsequently, the mean and covariance matrix of the predicted state need to be calculated:

$$\bar{X}_k = \sum_{i=0}^{2n}\left(W_m^{(i)} X_k^i\right) \tag{8}$$

$$P_k^- = \sum_{i=0}^{2n}\left[W_c^{(i)}\left(X_k^i - \bar{X}_k\right)\left(X_k^i - \bar{X}_k\right)^T + \Phi_k\right] \tag{9}$$

Where $\bar{X}_k$ and P_k^- represent the mean and covariance matrix of the predicted state of the Sigma point set, respectively, W_m and W_c are the weight coefficients for collecting Sigma points, and Φ_k is the process noise covariance matrix generated by the IMU.

The values of W_m and W_c are respectively:

$$W_m^{(i)} = \begin{cases} \dfrac{\lambda}{n} + \lambda, & i=0 \\ [4pt]\dfrac{1}{2(n+\lambda)}, & i=1,2,\ldots,2n \end{cases} \tag{10}$$

$$W_c^{(i)} = \begin{cases} \dfrac{\lambda}{n+\lambda} + \left(1-\alpha^2+\beta\right), & i=0 \\ [4pt]\dfrac{1}{2(n+\lambda)}, & i=1,2,\ldots,2n \end{cases} \tag{11}$$

Where α is the Sigma distribution point scale parameter, and β is a non-negative weight coefficient.

Step 3: Update and Fusion

In order to effectively integrate and update the IMU data points with the motion data from the odometer, it is crucial to first calculate the mean and covariance matrix Ψ_k of the observation state. By incorporating the relevant IMU observation noise covariance matrix, the fusion process can be accurately executed. This step is essential in ensuring the seamless integration of the various data sets for a comprehensive analysis of the overall motion dynamics.

$$\bar{Z}_k = \sum_{i=0}^{2n}\left(W_m^{(i)} Z_k^i\right) \tag{12}$$

$$P_{z,k}^- = \sum_{i=0}^{2n}\left[W_c^{(i)}\left(Z_k^i - \bar{Z}_k\right)\left(Z_k^i - \bar{Z}_k\right)^T + \Psi_k\right] \tag{13}$$

The covariance matrix of the observation state and the covariance matrix of the predicted state need to be fused and calculated:

$$P_{x,z}^- = \sum_{i=0}^{2n}\left[W_c^{(i)}\left(X_k^i - \bar{X}_k\right)\left(Z_k^i - \bar{Z}_k\right)^T\right] \tag{14}$$

Calculate the Kalman gain after fusion:

$$K_k = P_{x,z}^-\left(P_{z,k}^-\right)^{-1} \tag{15}$$

State prediction needs to be performed by combining the measurement values with the fused observation data, while simultaneously updating the covariance matrix after fusing the IMU and odometer:

$$X_k^i = \bar{X}_k + K_k\left(Z_k^i - \bar{Z}_k\right) \tag{16}$$

$$P_k = P_{x,z}^- - \left(K_k P_{z,k}^-\right) - \left(K_k\right)^T \tag{17}$$

Step 4: Adaptive Adjustment

This step performs the most critical adaptive detection and adjustment of noise. Based on the measurement residual and Kalman gain, the noise parameters are updated:

$$\Phi_k = \Phi_k + \varphi \sum_{i=0}^{2n} \left[\left(K_k \left(Z_k^i - \bar{Z}_k \right) \right) \left(K_k \left(Z_k^i - \bar{Z}_k \right) \right)^T \right] \tag{18}$$

$$\Psi_k = \Psi_k + \varphi \left[\left(Z_k^i - \bar{Z}_k \right) \left(Z_k^i - \bar{Z}_k \right)^T - \Psi_k \right] \tag{19}$$

The adaptive gain parameter, denoted as λ, plays a crucial role in adjusting fusion noise based on specific conditions. This adaptive adjustment helps in mitigating unpredictable noise, leading to a more accurate model and improved algorithm robustness. The process involves repeating steps 2 to 4 until no new data is generated, ensuring that the model continues to adapt effectively to changing circumstances. By fine-tuning the adaptive gain parameter φ , the algorithm can better handle various noise conditions and enhance overall performance.

B. Weight-Based Improved Voxel Filtering

When the Cartographer algorithm processes LiDAR data, voxel filtering plays a crucial role in refining the collected point cloud information. This involves dividing the point cloud into a Voxel Grid and retaining the centroid point within each voxel while discarding excess data. Although this method helps reduce noise and maintain the overall characteristics of the point clouds, it has its limitations. Due to the inability of centroids to fully capture the internal dynamics of each block, certain blocks may contain redundant information while others may lack essential details post-filtering. To address this issue, this paper introduces a novel approach that combines particle filtering to assign weights to the point cloud data and conducts a secondary screening process to enhance the voxel filtering results. By integrating these techniques, the algorithm can effectively refine the point cloud data while mitigating the shortcomings of traditional voxel filtering methods. This innovative approach promises improved accuracy and reliability in processing LiDAR data for various applications.

LiDAR scanning provides point cloud data which is then processed through voxelization in order to create a three-dimensional voxel grid for each point. Centroids are designated to represent the points within each voxel grid. Sampling and statistics are then conducted on the centroid points within each larger voxel grid, and projected onto a two-dimensional plane to create an index of the voxel grid. This process allows for easier analysis and visualization of the point cloud data, making it more manageable and understandable. Overall, voxelization helps to simplify complex point cloud data and facilitate further analysis and interpretation.

$$\Phi_{\text{voxel}}(\hat{x}, \hat{y}, \hat{z}) = f\left(\varphi_{\text{voxel}}(x, y, z), \varphi_{\text{point}}(i, j, k)\right) \tag{20}$$

Where $\Phi_{\text{voxel}}(\hat{x}, \hat{y}, \hat{z})$ represents the projection of the three-dimensional voxel grid onto a two-dimensional plane, $\varphi_{\text{voxel}}(x, y, z)$ represents the original coordinates of the three-dimensional voxel grid centroid, $\varphi_{\text{point}}(i, j, k)$ represents the coordinates of each three-dimensional point cloud, and $f(\ )$ is the projection function.

State estimation is then performed on the point cloud projected onto the two-dimensional grid to predict the state information for the next step:

$$h_t = H\left(\Phi_{\text{voxel}}(\hat{x}, \hat{y}, \hat{z}) + K_t\right) + \delta \tag{21}$$

The paragraph discusses the relationship between the state pose,control input K_t ,state transition function h_t ,and noise disturbance δ in a point cloud. These elements define how the point cloud changes over time and how external factors can affect its state via the observation model $h(\ )$.

Weights are assigned to predicted point cloud states based on their similarity to previous data. Lower weights are given to point clouds that match closely with previous data, while higher weights are assigned to non-overlapping point clouds. This process involves calculating the observation value of each point cloud using the observation model function $V(\ )$.By comparing probabilities, weights are carefully assigned to ensure accurate predictions. This method allows for the efficient evaluation of point cloud data and helps improve the overall predictive accuracy of the system.

$$v_t = V(h_t) + \delta \tag{22}$$

$$w_t = P(v_t | h_t) \sum_{i=0}^{t} (v_t \times \Delta t) \tag{23}$$

In conclusion, the fusion screening process involves projecting the weighted point cloud data onto the world coordinate system and combining it with the voxelized data of the grid. The point cloud weight data is carefully adjusted before the screening takes place. The first step is to update the state information and point cloud weights. This ensures that the screening process runs smoothly and effectively, leading to accurate and reliable results.

$$h_{t+1} = \sum_{i=0}^{t} h_t \times \Delta t + \varepsilon_t \tag{24}$$

$$w_{t+1} = w_t \times P(v_t | h_{t+1}) \times \phi_t \tag{25}$$

After filtering, the state and density of the filtered point cloud information can be updated:

$$\begin{cases} \theta_{\text{new}}(\hat{x}, \hat{y}, \hat{z}) = \theta_{\text{old}}(x, y, z) \times w_{t+1} \\ \vartheta_{\text{new}}(\hat{x}, \hat{y}, \hat{z}) = \vartheta_{\text{old}}(x, y, z) \times h_{t+1} + \dfrac{\vartheta_E}{\theta_{\text{new}}(\hat{x}, \hat{y}, \hat{z})} \end{cases} \tag{26}$$

The updated point cloud density value, represented by Ω, is calculated by considering the point cloud weight at the new stage, denoted as Ψ. Additionally, the point cloud state information, symbolized by Φ, is updated based on the state information at the new stage and the point cloud distribution value within a unit space, denoted as $\theta_{\text{new}}(\hat{x}, \hat{y}, \hat{z})$ together with the updated weight w_{t+1} new point cloud correlation quantity $\vartheta_{\text{new}}(\hat{x}, \hat{y}, \hat{z})$ recursive state variable h_{t+1} and error term ϑ_E .

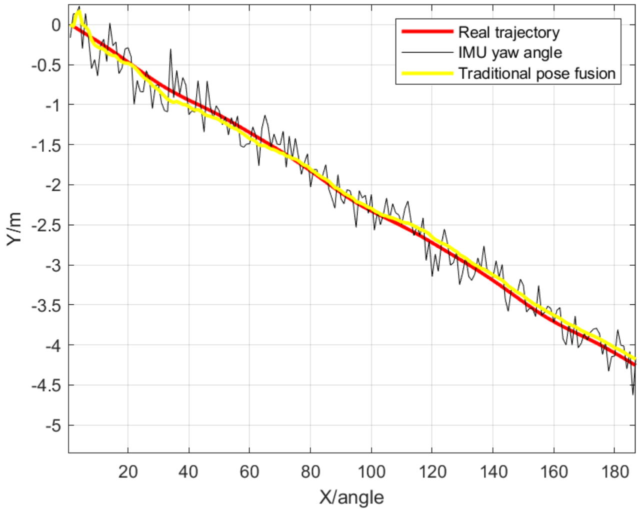
IV. Simulation and Real-world Environment Verification

A. AUKF Simulation Experiment Based on MATLAB

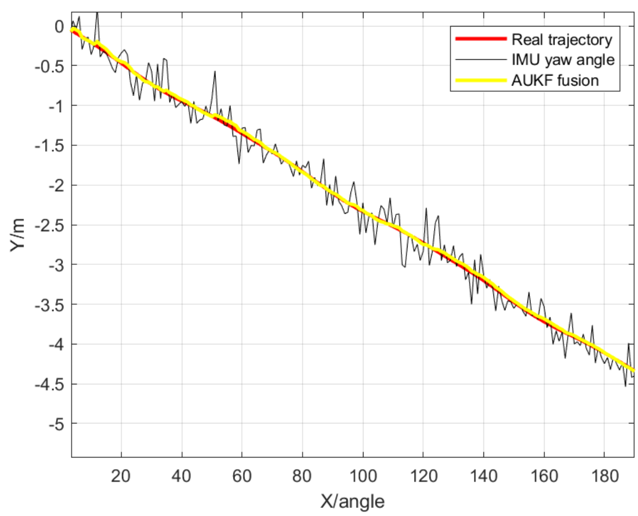
Research has confirmed the effectiveness of the AUKF algorithm in data fusion. Using MATLAB software, a study was conducted to compare the fusion performance of IMU and wheel odometer data using traditional UKF versus adaptive UKF. Three sets of experiments were undertaken to analyze the results. The comparison focused on the IMU-measured yaw angle and odometer information using both filtering methods. Results showed that the AUKF algorithm outperformed the traditional pose fusion algorithm in accurately filtering data. This was demonstrated by comparing the maximum yaw angle of 190°, with red denoting the true trajectory and yellow representing data filtered by both algorithms.

The data presented shows a clear difference between the traditional pose fusion algorithm (Figure a) and the optimized AUKF algorithm

(Figure b). The traditional algorithm displays notable discrepancies at the front and rear ends due to noise uncertainties. In contrast, the AUKF algorithm effectively addresses and improves this issue, showcasing its superior performance during implementation.



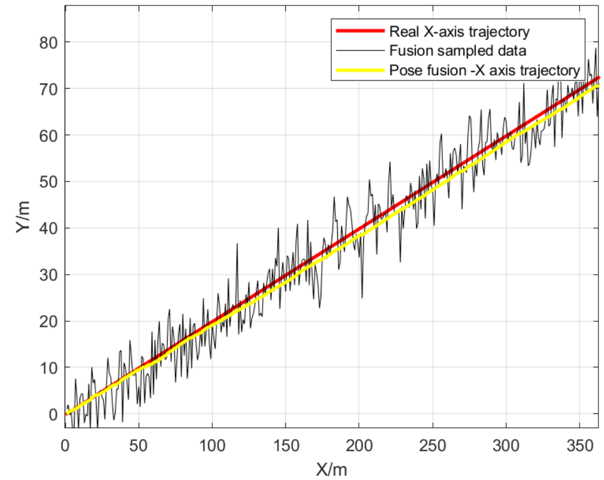
(a) Yaw angle trajectory of pose fusion



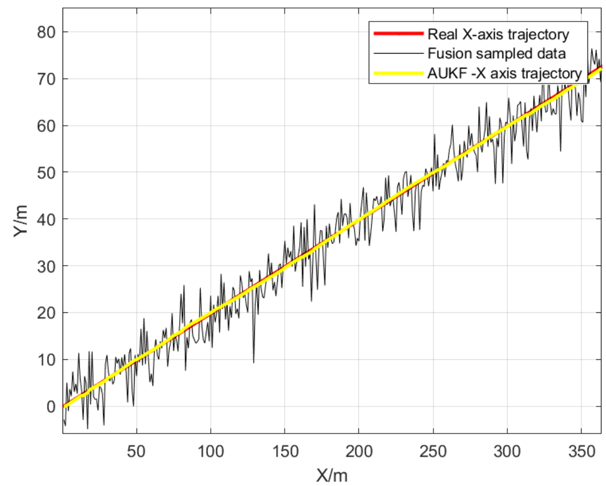
(b) Yaw angle trajectory of AUKF algorithm

Figure 3. Comparison of yaw angle trajectories

In Figure 4, the impact of sampling filtering on the fused IMU and odometer data is illustrated. The simulation results show differences in filtering conditions for the X and Y axes, highlighting the importance of accurate sensor data fusion in navigation systems.

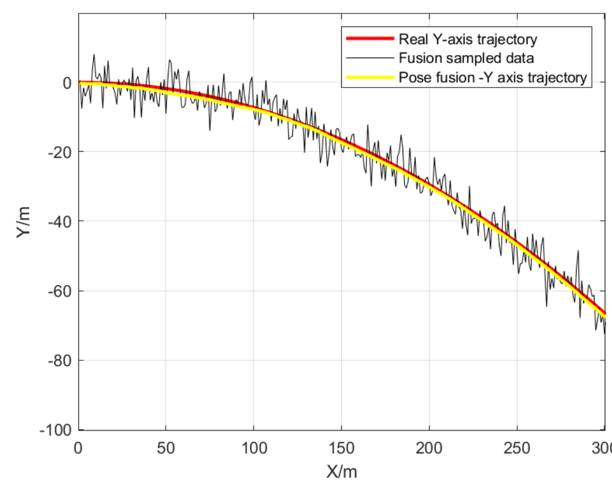


(a) X-axis trajectory of pose fusion algorithm

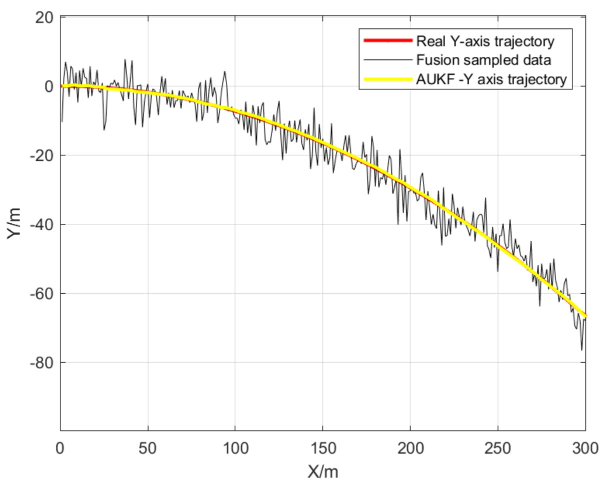


(b) X-axis trajectory of AUKF algorithm

Figure 4. Comparison of fused X-axis trajectories



(a) Y-axis trajectory of pose fusion algorithm



(b) Y-axis trajectory of AUKF algorithm

Figure 5. Comparison of fused Y-axis trajectories

The comparison between AUKF's optimization of IMU and odometer on operational trajectories and traditional pose fusion algorithms shows significant advantages.

TABLE I. TRAJECTORY DEVIATION CONDITIONS

Coordinate Axis	Trajectory Error	Pose Fusion Algorithm	AUKF Fusion Algorithm
X-axis Deviation	Maximum Value/m	3.84	0.65
	Mean Value/m	3.79	0.33
	Mean Value/m	0.41	0.26
Y-axis Deviation	Maximum Value/m	2.64	0.61
	Mean Value/m	2.15	0.28
	Mean Value/m	0.33	0.17

Traditional pose fusion had maximum deviations of 3.84m and 2.64m on two trajectories, while AUKF maintained trajectory error between 0.1-0.7mclosely aligning with the true trajectory. AUKF outperforms traditional pose fusion in noise processing and trajectory restoration accuracy. The table illustrates the specific trajectory deviation conditions. AUKF's superior performance in maintaining trajectory stability and accuracy highlights its effectiveness in improving navigation systems. Overall, AUKF's optimized algorithms lead to more precise and reliable trajectory estimations compared to traditional methods.

From the above results, it can be observed that in fusing IMU and odometer, the use of AUKF can significantly reduce potential path deviations and improve overall pose accuracy.

B. Real-world Environment Mapping Verification

The ROSMASTER X3 experimental verification platform utilized in the study features a Mecanum four-wheel robot with LiDAR technology, showcased in Figure 6. Comprising the robot chassis, core controller, and LiDAR component, the system incorporates the RPLIDAR A1 model. The chassis is powered by a Raspberry Pi 4B, while the host computer operates on Ubuntu 18.04. This setup provides a comprehensive foundation for testing and validating various robotic functionalities in real-time applications.

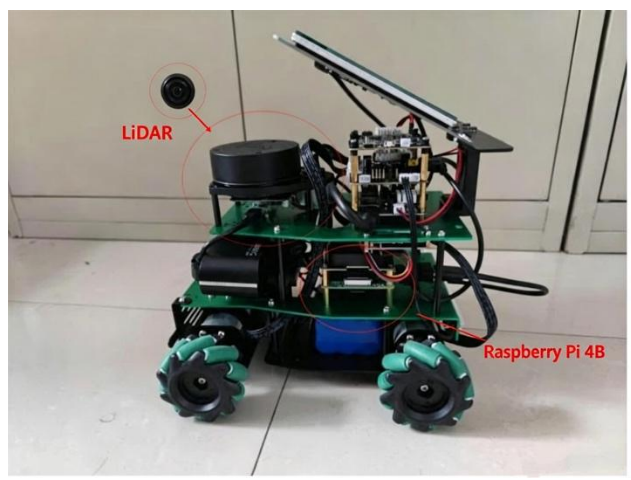


Figure 6. Experimental platform configuration

The RPLIDAR A1 is a single-line LiDAR system designed to efficiently convert obstacle data into two-dimensional information. By rotating the laser emitter and receiver in a continuous manner, the device utilizes triangulation ranging to capture obstacle details on a flat surface. To collect obstacle data at a specific height, the device is installed 20cm above ground level with an unobstructed front view, providing a full 360° scanning capability. The detailed performance specifications of the RPLIDAR A1 can be found in Table 2, showcasing its impressive capabilities in obstacle detection and data conversion.

TABLE II. RPLIDAR A1 PARAMETER SPECIFICATIONS

Parameter	Scanning Range/°	Measurement Range/r	Scanning Frequency/Hz	Measurement Frequency(times/s)
Specification	360	12	5.5-10	8000

1) *Indoor Environment Mapping Verification*

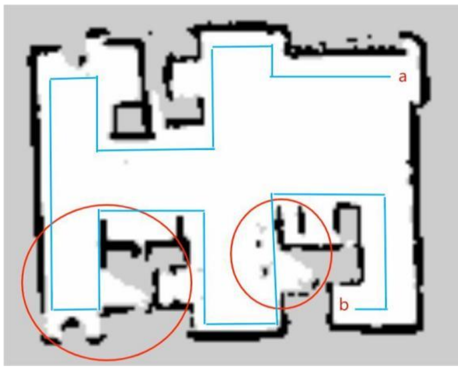
The initial mapping tests were carried out in a confined room measuring 4.5 meters in length and width, with obstacles strategically placed within the space. The purpose of this test was to assess the performance of the updated algorithm in indoor settings, starting from the entrance of the room. The results are detailed in Figure 7, demonstrating the algorithm's efficacy in navigating indoor environments with obstacles.

(a)Rear Half of Environment

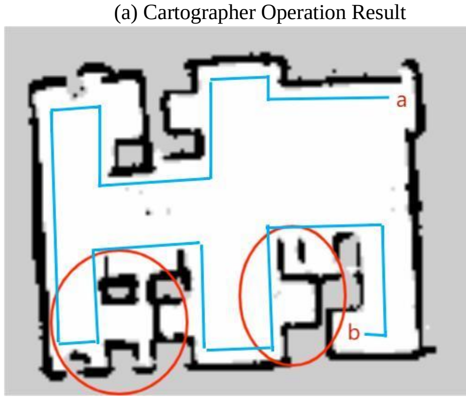
(b) Front Half of Environment

Figure 7. Indoor Mapping Environment

In a simulated environment, chairs and tables were strategically placed to create obstacles, with some tables having hollow undersides. The LiDAR system failed to detect obstacles in the hollow condition. Two mapping algorithms, the traditional Cartographer and a proposed improved version, were evaluated separately. The results of the mapping process revealed differences in the accuracy and efficiency of both algorithms. This study highlights the importance of using advanced mapping techniques to accurately navigate complex environments with varying obstacles.



(a) Cartographer Operation Result



(b) Improved Cartographer Operation Result

Figure 8. Indoor Environment Mapping Comparison

The indoor experimental results displayed in Figure 8 depict the mapping journey from point a to point b, starting at the door position. Comparing the outcomes of traditional Cartographer algorithm with a new method, it is evident that in intricate environments, the former shows incomplete mapping due to challenges in processing point cloud information. By applying the evaluation formula suggested in Reference [17], the mapping performance of both algorithms is critically analyzed. This study underlines the significance of utilizing advanced mapping techniques for better accuracy and efficiency in complex indoor environments. The findings highlight the importance of innovative approaches in overcoming mapping challenges.

$$f(\delta)=\frac{1}{N}\sum_{i,j=0}\left(\mathrm{trans}\left(\delta_{i,j}\Theta\delta^{*}_{i,j}\right)^{2}+\mathrm{rot}\left(\delta_{i,j}\Theta\delta^{*}_{i,j}\right)^{2}\right) \quad (26)$$

Where N is the relative relation value, $trans(\ )$ represents the component of separated weighted translation, $rots(\ )$ represents the component of separated weighted rotation, δ and δ^{*} represent the estimated value and true value respectively, θ is the operator for inverse standard motion composition, and $\delta_{i,j}$ represents the transformation relationship from i to j under the coordinate system.

After verifying the operation results in Figure 8 through the above formula, quantitative data analysis was performed, and the results are shown in Table 3.

Traditional pose fusion had maximum deviations of 3.84m and 2.64m on two trajectories, while AUKF maintained trajectory error between 0.1-0.7mclosely aligning with the true trajectory. AUKF outperforms traditional pose fusion in noise processing and trajectory restoration accuracy. The table illustrates the specific trajectory deviation conditions. AUKF's superior performance in maintaining trajectory stability and accuracy highlights its effectiveness in improving navigation systems. Overall, AUKF's optimized algorithms lead to more precise and reliable trajectory estimations compared to traditional methods.

From the above results, it can be observed that in fusing IMU and odometer, the use of AUKF can significantly reduce potential path deviations and improve overall pose accuracy.

TABLE III. QUANTITATIVE ANALYSIS OF INDOOR MAPPING PERFORMANCE

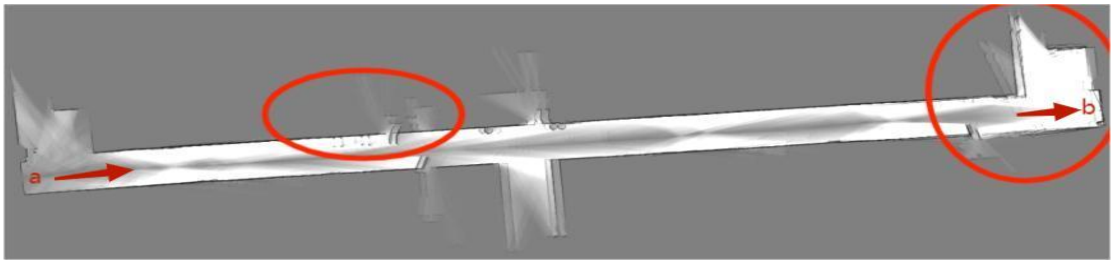
Parameter	**Cartographer Algorithm**	**Improved Cartographer Algorithm**
Absolute Translation Error/m	0.71574±1.12082	0.59267±0.64481
Root Mean Square Translation Error/m²	1.13662±0.49230	0.77144±0.23769
Absolute Rotation Error/°	1.29337±0.88421	0.90386±0.57340
Root Mean Square Rotation Error/(°)²	1.11335±0.76452	0.78441±0.59208

2) Outdoor Environment Mapping Verification

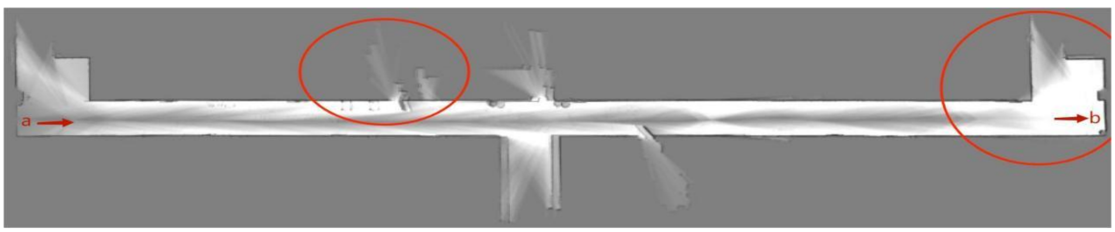
The subsequent set of experiments detailed in this paper took place outdoors within a 65m enclosed corridor, as depicted in Figure 9. The purpose was to assess the enhanced algorithm's performance in an outdoor setting. Commencing from the western edge and concluding at the eastern end of the corridor, the test sought to confirm the algorithm's efficacy in real-world conditions.

Figure 9. Outdoor Mapping Environment

Maintaining accurate and reliable detection data in outdoor environments with complex situations like pedestrians and intermittently opening doors requires walking in straight lines at a consistent speed. This ensures stable IMU data throughout the process. It's important to note that the surrounding areas connected to the corridor are not considered in this verification scope to maintain focus. By comparing the performance of the traditional and improved Cartographer algorithms, we can see the impact of these advancements on mapping accuracy. The figure below illustrates this comparison, highlighting the benefits of utilizing improved algorithms for better navigation and data collection in outdoor settings.



(a) Cartographer Operation Result



(b) Improved Cartographer Operation Result

Figure 10. Outdoor Environment Mapping Comparison

The outdoor environment experiment progressed from point a to point b, with both mapping processes experiencing blurring phenomena during the middle section of the journey. This was due to pose information deviation caused by the unmapped middle corridor section, resulting in map overlay effects. In the latter half of the mapping, the traditional Cartographer showed significant oscillations attributed to noise errors accumulating from extended IMU and odometer operation without correction. Conversely, the improved Cartographer algorithm demonstrated noteworthy correction in this area. Quantitative comparisons of absolute translation error and absolute rotation error were carried out, the results of which can be seen in Table 4. Overall, the findings highlight the importance of accurate correction mechanisms in mapping processes to minimize errors and improve overall efficiency.

TABLE IV. QUANTITATIVE ANALYSIS OF OUTDOOR MAPPING PERFORMANCE

Parameter	**Cartographer Algorithm**	**Improved Cartographer Algorithm**
Absolute Translation Error/m	0.44702±1.61382	0.33169±0.83511
Root Mean Square Translation Error/m²	0.51387±0.76447	0.39785±0.50639
Absolute Rotation Error/°	1.00824±0.28720	0.98431±0.46181
Root Mean Square Rotation Error/(°)²	0.87335±1.53962	0.62105±1.13429

Table 3 data reveal a significant enhancement in mapping accuracy achieved by the improved Cartographer algorithm in outdoor settings. The algorithm successfully minimized translation error by 25.8% and rotation error by 28.9% compared to the traditional version. This marks a significant improvement in mapping precision and overall performance.

V. CONCLUSIONS

The development of Cartographer mapping algorithms has been pivotal in the realm of mapping technology. However, the original algorithm has encountered challenges such as trajectory drift, odometer slippage, and poor point cloud filtering performance, leading to issues like point cloud redundancy, high computational load, and subpar mapping quality. In response to these limitations, an enhanced Cartographer mapping algorithm has been proposed to mitigate these challenges effectively. This improved algorithm tackles the root causes behind these problems through two key methods targeting distinct components. Firstly, in the pose information fusion component, an Adaptive Unscented Kalman Filter (AUKF) is utilized to fuse Inertial Measurement Unit (IMU) and odometer data. This fusion process minimizes noise impact on odometer information and mapping accuracy, enabling real-time noise estimation updates to enhance trajectory precision and mapping quality. Secondly, in the point cloud data processing component, a novel weighting process is introduced alongside the original voxel filtering technique. This process prioritizes point clouds based on their weights, removing lower-weighted clouds for further screening and filtering. Additionally, density and state data of the remaining point clouds are updated before adaptive voxel filtering, reducing cumulative mapping errors and ensuring comprehensive mapping outcomes. Experimental tests conducted in indoor and outdoor environments have demonstrated the efficacy of the enhanced algorithm. In indoor settings, the algorithm reduced absolute translation error by 17.2% and absolute rotation error by 30.1% compared to the original Cartographer algorithm. Similarly, in outdoor environments, the improved algorithm achieved a 25.8% reduction in absolute translation error and a 28.9% decrease in absolute rotation error. These results signify the algorithm's ability to minimize mapping errors and enhance overall mapping quality, showcasing its potential for advancement in the mapping technology landscape.

ACKNOWLEDGMENT

The authors express their gratitude to their collaborators. This research has received partial funding from the National-Level College Student Innovation and Entrepreneurship Fund Project (202510702057) and the Shaanxi Province College Students Innovation and Entrepreneurship Project (S202510702093X).

REFERENCES

[1] SMITH R, SELF M, CHEESEMAN P. Estimating uncertain spatial relationships in robotics[J]. Machine Intelligence and pattern Recognition, 1988, 5: 435-461.

[2] Li Y B. Research on Indoor SLAM and Path Planning for Omnidirectional Mobile Robot [D]. North University of China, 2023.

[3] Wang G C. Design and Research on Omnidirectional Mobile Robot System Based on ROS [D]. Anhui University of Science and Technology, 2019.

[4] Li X K, Li G Y, Suo S H, et al. Progress in Laser SLAM Technology [J]. Journal of Navigation and Positioning, 2023, 11(04): 8-17.

[5] The article discusses a laser SLAM algorithm that utilizes an enhanced version of Cartographer for better performance in mapping environments. The study explores how this improved algorithm can enhance the efficiency and accuracy of Simultaneous Localization and Mapping (SLAM) systems.

[6] Liu Z C, Wang H L, Ma X L. Review of Laser Simultaneous Localization and Mapping Algorithms [J/OL]. Computer Measurement and Control: 1-12 [2023-10-04].

[7] Yao and colleagues investigated the utilization of the Unscented Kalman Filter Algorithm for integrating

Ultra-Wideband and Inertial Measurement Unit technologies in positioning systems. Their study in Mechanical Science and Technology provides valuable insights for improved position accuracy in various applications.

[8] A study on implementing an Improved Particle Filter for more accurate state estimation in non-Gaussian forecasting in the energy sector, focusing on enhancing reliability and efficiency. The research explores methods to optimize forecasting-aided state estimation processes.

[9] Qi J C, He L, Yuan L, et al. SLAM Method Based on Monocular Camera and LiDAR Fusion [J]. Electronics Optics & Control, 2022, 29(2): 99-102.

[10] Title, Optimization of Cartographer Algorithm with Multi-Resolution Layered Search Strategy. The study presents an improved algorithm for cartographers that enhances front-end matching efficiency through a multi-resolution layered search strategy. The optimized approach aims to increase accuracy and speed of map creation in various applications.

[11] An integration method combining ultra-wideband, inertial measurement unit, and wheel odometer for improved positioning using an enhanced unscented Kalman filter is proposed. Results in the Journal of Chinese Inertial Technology demonstrate its effectiveness.

[12] Wang X, Li D F, Huang S S, et al. Algorithm Design for Motion Distortion Removal of LiDAR [J]. Process Automation Instrumentation, 2021, 42(5): 89-91.

[13] Shen X, Min H S. Improved Cartographer Algorithm Based on Velocity Integration and Pose Fusion [J]. Applied Laser, 2021, 41(5): 1063-1069.

[14] Lin G C, Xue B Q, Wang D Q. Research on SLAM Back-end Optimization Algorithm Based on Graph Optimization [J]. Electronic Design Engineering, 2020, 28(24): 6-10.

[15] DWIJOTOMO A, RAHMAN M A, ARIFF M H, et al. Cartographer SLAM method for optimization with anadaptive multidistance scan scheduler[J]. Applied Sciences,2020,10(1): 347

[16] HESS W, KOHLER D, RAPP H, et al. Real-time loop closure in 2D LIDAR SLAM[C]//2016 IEEE International Conferenceon Robotics and Automation (ICRA), StockholmSweden: IEEE,2016:1271-1278

[17] Kvmmerle R, Steder B, Dornhege C, Ruhnke M, Grisetti G, Stachniss C, Kleiner A. On measuring the accuracy of SLAM algorithms. Autonomous Robots. 2009. 27(4): 387-407.