WEP Warwick Evans Publishing

International Journal of Computer Science and Information Technology

ISSN: 3005-9682 (Print), ISSN: 3005-7140 (Online) | Volume 6, Number 2, Year 2025
DOI: https://doi.org/10.62051/ijcsit.v6n2.01
Journal homepage: https://wepub.org/index.php/IJCSIT/index

Research on Laser SLAM Mapping and Navigation Algorithm Based on Improved Cartographer Algorithm and Path Planning Algorithm

Jingzhi Zhang

School of Management Science and Engineering, Anhui University of Finance and Economics, Bengbu, Anhui, 233030, China
2390150198@qq.com

ABSTRACT

Aiming at the problems of high environmental sensitivity, poor mapping effect and poor autonomous navigation and positioning accuracy of traditional laser SLAM mapping in complex environments, a lidar mapping algorithm based on the improved Cartographer algorithm and a path planning algorithm based on multi-algorithm fusion are proposed. Firstly, the radar data is dedistorted by using the feature extraction method of the curvature size in LIO-SAM to generate high-quality point clouds. Then, the adaptive untraced Kalman filtering algorithm is used to fuse the IMU and odometer data to obtain more accurate pose estimation. When establishing the path planning algorithm, the RRT-Connect algorithm is used as the global path planning, and the artificial potential field method is used as the local path planning. The RRT-Connect algorithm is utilized to construct an optimal path from the initial point to the target point, and the artificial potential field method is used for real-time obstacle avoidance and local path correction during the operation process. By integrating the two algorithms, autonomous navigation operations in complex environments are achieved. Finally, the mapping algorithm and navigation algorithm were verified in the gazebo environment and the outdoor real environment. Experiments show that the mapping accuracy of the optimized Cartographer algorithm is higher. Meanwhile, when using the path planning algorithm to achieve autonomous navigation, the positioning of navigation is more accurate when the speed is less than 0.3m/s, and it is suitable for the navigation requirements of complex outdoor environments.

KEYWORDS

Cartographer algorithm; Lidar; Multi-sensor fusion; Path planning algorithm

1. INTRODUCTION

With the development of autonomous driving technology, map construction and positioning, as a core technical field, has attracted many scientific researchers to devote themselves to it. Since the Simultaneous Localization and Mapping (SLAM) technique was first proposed from a probabilistic perspective at the ICRA conference in 1986, Smith [1] and others established the statistical basis for describing the relationships between landmarks and handling the relationships between set uncertainties in 1988, marking the official entry of this technology into a steady development stage. At present, the mainstream SLAM technologies are divided into laser SLAM based on lidar and visual SLAM based on monocular or binocular cameras. The core idea is to utilize sensors to achieve the construction of real-time maps and the position estimation of robots in the maps.

For laser SLAM, the current mainstream methods are gampping [2], Hector SLAM [3], Karto [4], Cartographer [5], etc. As for visual SLAM, the main ones are ORB-SLAM3 [6], VINS-Fusion [7],

InfiniTAM [8], etc. Aiming at the high environmental sensitivity and radar data distortion problems of laser SLAM, Fu [9] et al. adopted a wheeled odometer to assist the lidar in motion distortion correction, and utilized the untraceless Kalman filter to integrate multiple sensors for pose optimization estimation to improve the mapping accuracy. However, the complexity of the final algorithm is relatively high and the time consumption is long. Liu [10] et al. utilized the inertial measurement unit (IMU) tightly coupled with lidar for positioning mapping, which improved the robustness of the algorithm, but it was highly dependent on the accuracy of IMU data. Li [11] et al. improved the backloop by using the GICP algorithm to enhance the matching accuracy of the backloop point cloud. However, it consumed too many resources and had certain disadvantages in real-time mapping. Li [12] et al. improved the ORB-SLAM3 framework by using a wheeled odometer to reduce the motion drift of the sensor, but still had the characteristic that visual SLAM was greatly affected by environmental features.

SLAM mapping provides a technical basis for the implementation of subsequent path planning algorithms. The optimal path from the current point to the target point is planned on the previously established map through the path planning algorithm, thereby achieving the navigation operation. At present, the mainstream of intelligent cars adopts the Nav navigation framework for navigation operations. In Nav, the global path planner is used to generate the best path of the current environment, and then the local path planner is used to handle the complex situation of the current running section in real time. Thus, the global path planning algorithm and the local path planning algorithm are integrated. The commonly used path planning algorithms at present are Dijkstra [13], A* [14], DWA [15], RRT [16], artificial potential field [17], etc. Tang [18] utilized the improved ant colony algorithm as the global path planning algorithm and DWA as the local path planning algorithm to achieve the path planning of the wheeled robot with the shortest path length and smooth corners. Miao [19] et al. utilized the improved RRT algorithm to enhance the robustness and search ability of the path planning algorithm, and simultaneously introduced the artificial potential field method to expand the search direction of the algorithm. Jia [20] presented the detailed theoretical derivation ideas of A* as the global path planning algorithm and DWA algorithm as the local path planning.

To sum up, this paper draws on the feature extraction operation in LIO-SAM [21] to extract feature points for two-dimensional lidar, in order to reduce the influence of lidar distortion on mapping. Then, the radar data after feature extraction is fused with multi-sensor data using the adaptive traceless Kalman filtering algorithm (AUKF) to improve the Cartographer algorithm for mapping operations. And RT-Connect is utilized as the global path planning algorithm, and the artificial potential field is used as the local path planning algorithm to improve the path planning algorithm. Finally, the effectiveness of the improved path planning algorithm is verified in the map established by the optimized mapping algorithm.

2. RADAR DATA PROCESSING

In this paper, two-dimensional lidar is adopted for experiments. Its basic working principle is that a pulsed laser beam is emitted through the emission system. The laser beam propagates in a specific direction. When encountering an object, part of the laser energy will be reflected. The intensity and characteristics of the reflected light depend on the material of the object, its surface properties and the incident Angle of the laser. Then the receiving system will receive the reflected laser, process it, obtain the distance and Angle information of the object, and finally determine the position of the object on the two-dimensional plane. Lidar continuously measures in a certain scanning mode, and a two-dimensional point cloud map composed of a group of observation points can be obtained. Since the lidar itself is in motion when obtaining data, it will cause the coordinate system of each frame of data to be different. However, when processing, it will be treated as the same coordinate system, which will cause errors and result in the distortion of lidar data [22]. This paper draws on the method of feature extraction in LIO-SAM to extract feature points from radar data.

2.1. LIO-SAM method

LIO-SAM, whose full name is Tightly coupled Lidar Inertial Odometry via Smoothing and Mapping, was developed by Tixiao Shan based on LeGO-LOAM. The idea of factor graph is used to optimize laser SLAM, and the IMU pre-integration factor, lidar range factor, GPS factor and closed-loop factor are introduced respectively. The initial value of the radar odometer is optimized by integrating a large number of relative measurement values, absolute measurement values, loops and other data as factors into the radar inertial odometer system. The feature extraction process of LIO-SAM is as follows.

2.1.1. Point cloud data preprocessing

Due to reasons such as the Angle between the object and the radar ray being too large or the reflection intensity of the laser being very low, the radar data may contain many inf and nan values. As invalid points in the point cloud, errors may occur in subsequent mathematical operations. Therefore, the invalid values should be removed at the beginning. Meanwhile, LIO-SAM also considers the possible outliers, including the points that have been selected in the surrounding area, the points approximately parallel to the laser beam on the local plane, and the points located at the boundary of the occlusion area. Abnormal points are filtered out by detecting the distance difference between two points of the laser beam.

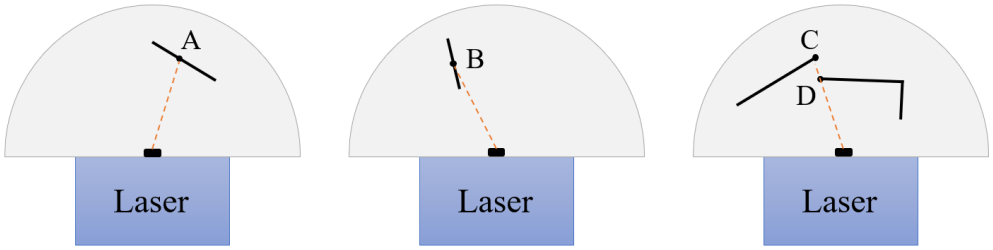
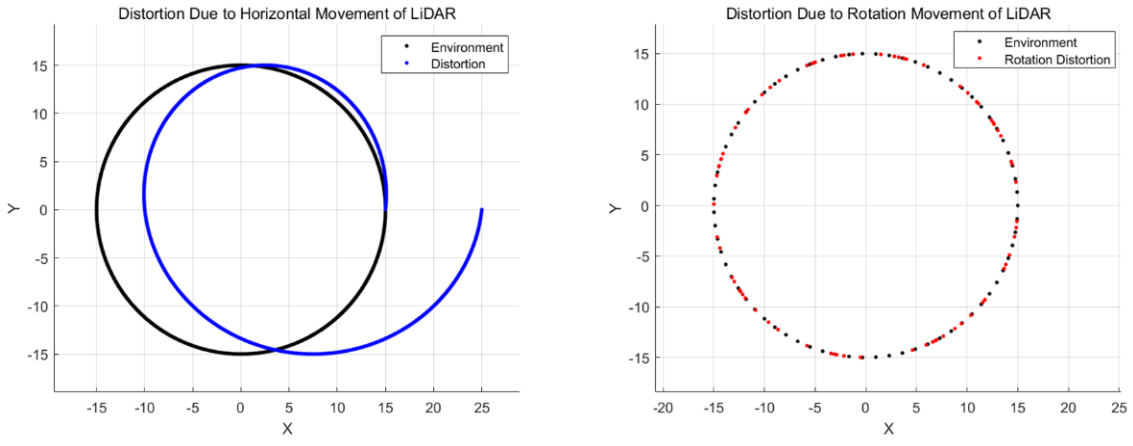


Figure 1. Radar anomaly point

2.1.2. Point cloud dedistortion

Since the point cloud coordinates obtained by the lidar are all moving relative to the radar during the movement process, the origin of the coordinate axes of the different point clouds is different. The radar was simulated using Matlab, and the results are shown in Figure 2. (a) represents the distortion during horizontal movement, and (b) represents the distortion during rotational movement.



(a) Distortion Due to Horizontal Movement

(b) Distortion Due to Rotation Movement

Figure 2. Distortion

The compensation method of LIO-SAM is to record the coordinate transformation of the acquisition moment of each point relative to the initial moment, and then use time for interpolation, thereby converting all point clouds to the same coordinate system.

Let the pose at the start time of collection (time t) be $(R,T)_t$ and the pose at the end time (time t+1) be $(R,T)_{t+1}$. Then, the phase pose transformation between the start and end times can be calculated as $(R,T)_{t+1}^{t}$. By using the horizontal Angle of the point cloud, the relative time of a certain point i in a frame of the point cloud can be obtained as follows.

$$T_i = \frac{ori - startori}{endori - startori} * (T_{t+1} - T_t) \tag{1}$$

Among them, ori represents the Angle at the current moment, startori represents the Angle at the beginning moment, and endori represents the Angle at the end moment. Then it can be concluded that the pose change of a certain point i relative to the starting moment is blow.

$$(R,T)_i = \frac{T_i}{T_{t+1} - T_t} * (R,T)_{t+1}^{t} \tag{2}$$

Then the corrected point cloud coordinates can be obtained as

$$P_{result} = (R,T)_i * P_i \tag{3}$$

Among them, P_{result} is the corrected coordinate and P_i is the coordinate at time i.

2.1.3. Curvature calculation

In LIO-SAM, five points before and after the same laser beam are taken to form the set S, and the average distance of the points is used as the estimated value of the curvature. Then for the point collection $P_t = \{p_i, i \in | P_t |\}$, let the distance from each point to the origin be r_i, and the curvature c of point p_i can be expressed as

$$c = \frac{1}{|S| \cdot \| r_i \|} \left\| \sum_{j \in S, j \neq i} (r_j - r_i) \right\| \tag{4}$$

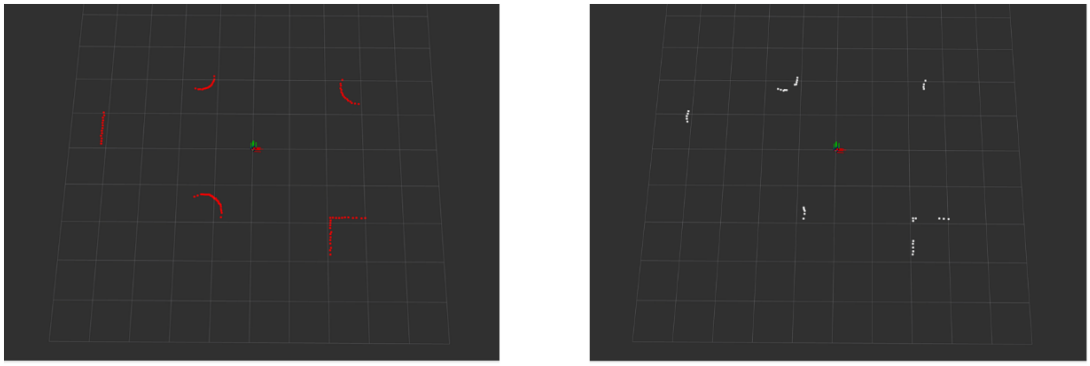
2.1.4. Feature point extraction

The extraction of feature points is judged by comparing the curvatures of each point. When the curvature of a certain point cloud is higher than the set threshold, it is regarded as a corner point; when the curvature is less than the threshold, it is identified as a plane point. In order to evenly distribute the feature points in the environment, LIO-SAM evenly divides the environment into 6 identical sub-regions. At most 20 corner points can be extracted from each sub-region, and there is no limit to the number of planar points. The conditions for the selection of feature points are that the number of feature points does not exceed the maximum value of the sub-region, the surrounding points of the feature points cannot be repeatedly selected, and the outliers in 1) cannot be selected as feature points.

2.2. Radar Data Feature Extraction

By drawing on the feature extraction method of LIO-SAM, feature extraction is carried out on the original radar data.

Firstly, a large number of inf and nan values in the radar data are filtered out, and only the valid distance points are retained. Then, the curvature value is calculated, and the deviation degree of the values of the five points before and after the current point is taken as the estimated curvature value. Finally, feature point extraction is carried out. The original radar data is divided into six parts. The indexes of the starting and ending points of these six data segments are calculated respectively. The data of each segment is sorted from small to large according to the curvature value, and then up to 20 points are selected as feature points through traversal from back to front. After implementation through code, the effect is shown in Figure 3.



(a) Radar raw data (b) The radar data after feature extraction

Figure 3. Illustration of Radar Feature Extraction

3. IMPROVEMENTS TO THE CARTOGRAPHER ALGORITHM

3.1. Analysis of the Cartographer Algorithm

Cartographer is a set of graph-based SLAM algorithms launched by Google, which can provide cross-platform 2D and 3DSLAM methods. The main goal of this algorithm is to achieve real-time SLAM operations while realizing low resource consumption.

The Cartographer algorithm is divided into two parts. The first part is local SLAM, which adopts the scan-map matching algorithm. That is, a Submap is established through frame by frame of radar scanning. The submap is composed of a series of Grid maps. When new radar data is generated, it will be inserted into the best position in the sub-map through the Ceres Scan Matching method. The second part is the global SLAM section. When radar scan data is continuously added to the subgraph, the accumulation of cumulative errors will occur. In this part, Loop Closure is used for loopback detection to achieve the purpose of eliminating cumulative errors. When a Submap is no longer updated, the algorithm will add it to the loop for optimization processing.

When the Cartographer algorithm builds a map, the entire process is associated through pose relationships. If $\xi = (\xi_x, \xi_y, \xi_\theta)$ is used to represent the pose coordinate, then the initial pose coordinate is set as $\xi_1 = (0,0,0)$, the radar scan frame at this pose is set as $Scan_1$, and the first submap is initialized as $Submap_1$ through $Scan_1$. Then, the scan-map matching algorithm is used to calculate the pose ξ_2 of $Scan_2$, and $Scan_2$ is added to $Submap_1$ based on the pose ξ_2. Continuously add radar frames to subfigure 1 until the new radar frames fail to detect new data other than $Submap_1$, and then complete the creation of $Submap_1$. Then repeat the above steps to construct $Submap_2$. Integrate all the local subgraphs $Submap_m$ to form the final global map map. Each radar frame corresponds to the global coordinates under a global map and the local coordinates under a local submap.The global pose $\xi_j^s, j = 1,2,\ldots,s$ corresponding to all radar frames and the global pose

$\xi_i^m, i=1,2,\ldots,m$ in the local subgraph are correlated through the local pose ξ_{ij} of the scan-map algorithm to form the pose map.During the loop detection, the pose quantities of the entire pose map are corrected, and the corresponding coordinates of each pose on the map will also be corrected, thus achieving the global mapping operation.

Let $\{h_k\}, k=1,2,\ldots,K$ represent the distance points obtained by the radar scanning one circle, $T_\xi = (R_\xi, t_\xi)$ represent the relative pose transition matrix of the radar frame, and the $M_{nearest}$ function represent the probability value corresponding to the coverage grid of the radar data point $T_\xi \bullet h_k$. Then, when conducting loopback detection, the corresponding search results are as follows.

$$\xi^* = \arg\max_{\xi \in W} \sum_{k=1}^{K} M_{nearest}(T_\xi \bullet h_k) \tag{5}$$

If the search result is the true pose corresponding to the current radar frame, and the matching degree between the contour of the current radar frame and the map is high, then each value of the $M_{nearest}$ function is large, and the search result is the largest. Loopback detection is to make the value of Equation (5) maximum through search. Cartographer uses the branch and bound method for efficient search matching, which is essentially a depth-first search. First, the low-resolution maps are matched, and then the resolution is gradually increased. The optimal solution is selected by vertically comparing the matching scores at different resolutions. The score calculation formula is as follows.

$$score \leftarrow \sum_{k=1}^{K} M_{nearest}(T_{\xi_0+\Delta\xi} \bullet h_k), \Delta\xi = \begin{bmatrix} j_x \bullet \Delta x \\ j_y \bullet \Delta y \\ j_\theta \bullet \Delta\theta \end{bmatrix} \tag{6}$$

Loopback detection is carried out after the arrival of each frame of radar data. When the matching score in the detection exceeds the set threshold, a closed loop is determined and added to the global constraint. An optimization operation is performed on the global pose using the sparse pose map. The structure of the final constructed Cartographer map is shown in Figure 4.

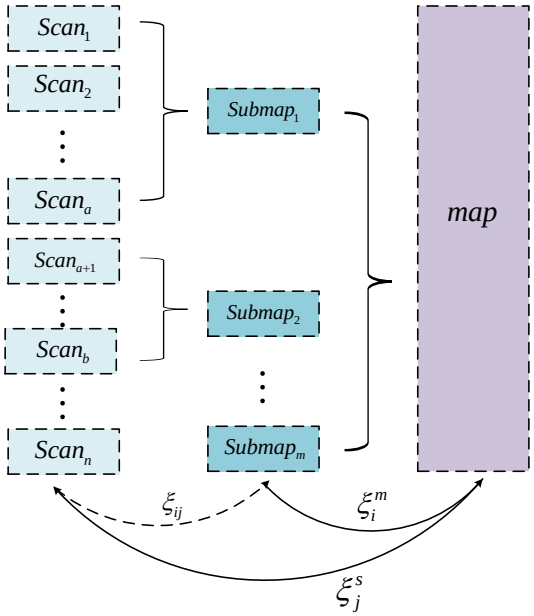


Figure 4. Cartographer map structure

3.2. Improvements to the Cartographer Algorithm

The original Cartographer algorithm adopts loose coupling when performing multi-sensor fusion. That is, when optimizing the initial pose, if it contains an IMU, the angular velocity integral of the IMU is used as the initial pose, and no acceleration information of the IMU is trusted. If an odometer is included, the linear velocity integral of the odometer is adopted as the initial translation. Specifically, when positioning, it mainly relies on the data of lidar, while the data of IMU and odometer are not truly used to optimize the objective function. Therefore, the adaptive traceless Kalman filtering algorithm is adopted to fuse the data obtained by IMU and odometer, and combined with the lidar point cloud information after feature extraction to obtain the current pose estimate value for mapping operation.

3.3. Improvement Methods of the AUKF Algorithm

When conducting pose estimation, the motion and measurement models of the current state of the robot system are initialized and modeled first. Its motion model is as follows.

$$x_k = f(x_{k-1}) + \omega_k \tag{7}$$

Among them, x_k is the state vector, $f()$ is the state transition equation, and ω_k is the process noise of the system.

Let the measured value of the current state obtained through the sensor be z_k, and Equation (8) is its measurement model.

$$z_{k+1} = h(x_k) + \gamma_k \tag{8}$$

In the formula, $h()$ is the observation equation and γ_k is the measurement noise.

Then, by combining the noise distribution of the IMU and the odometer, a set of Sigma point sets is generated on the Gaussian distribution of the state vector x_k, containing a total of 2n+1 points, where n is the dimension of x_k. Let E_k represent the mean of the state vector x_k and C_k represent the covariance of the state vector x_k. Then the selected points are as follows

$$\begin{cases} x_k^0 = E_k \\ x_k^i = E_k + (\sqrt{(n+\lambda)C_k})_i, i = 1, 2, \ldots, n \\ x_k^j = E_k - (\sqrt{(n+\lambda)C_k})_j, j = n+1, n+2, \ldots, 2n \end{cases} \tag{9}$$

Among them, x_k^0, x_k^i , and x_k^j represent the state variables at the corresponding moments, and i and j are the loop iteration variables. λ is the scale adjustment factor, which is used to control the distance between the Sigma point and the mean value.

If the point set output of Sigma is represented by y_i , then the predicted Sigma point set y_k and the predicted measurement value z_k' can be obtained by using the equation of state and the observation equation.

$$\begin{aligned} y_k &= f(x_k) \\ z_k' &= h(y_k) \end{aligned} \tag{10}$$

Then calculate the mean $\bar{E}_k$ and covariance matrix $\bar{C}_k$ of the predicted state. The formula is as follows

$$\begin{cases} \bar{E}_k = \sum_{i=0}^{2n} W_m^i y_k^i \\ \bar{C}_k = \sum_{i=0}^{2n} [W_c^i (y_k^i - \bar{E}_k)(y_k^i - \bar{E}_k)^T + \sigma_k] \end{cases} \tag{11}$$

Among them, W_m^i and W_c^i are the weight coefficients of the collected Sigma points respectively, and σ_k is the process noise covariance matrix during the generation of IMU and odometer data. The weights are as follows.

$$\begin{cases} W_m^0 = \dfrac{\lambda}{\lambda + n} \\ W_c^0 = \dfrac{\lambda}{\lambda + n} + (1 - \alpha^2 + \beta) \\ W_m^i = W_c^i = \dfrac{1}{2(n+\lambda)}, i = 1, 2, \ldots, 2n \end{cases} \tag{12}$$

In the formula, α is the proportional parameter of the Sigma distribution points, and β is the non-negative weight coefficient. The value of α is generally between 0 and 1, while β takes 0 as a univariate parameter and 2 under a Gaussian distribution.

If φ_k is used to represent the covariance matrix of the sensor's observed noise, the mean $\bar{E}_{z|k}$ of the observed values and the covariance matrix $\bar{C}_{z|k}$ can be obtained.

$$\begin{cases} \bar{E}_{z|k} = \sum_{i=0}^{2n} W_m^i z_k'^i \\ \bar{C}_{z|k} = \sum_{i=0}^{2n} [W_c^i (z_k'^i - \bar{E}_{z|k})(z_k'^i - \bar{E}_{z|k})^T + \varphi_k] \end{cases} \tag{13}$$

The covariance matrix between the state at the time of prediction and the measurement quantity is fused to calculate the cross-covariance matrix.

$$\bar{C}_{xz|k} = \sum_{i=0}^{2n} [W_c^i (y_k^i - \bar{E}_k)(z_k'^i - \bar{E}_{z|k})^T] \tag{14}$$

The Kalman gain can be calculated based on the observed covariance matrix and the cross-covariance matrix.

$$K_k = \bar{C}_{xz|k} [\bar{C}_{z|k}]^{-1} \tag{15}$$

Then, the filtered state estimation is carried out based on the predicted values. The state $x_{k|k}$ and covariance matrix $C_{k|k}$ corresponding to time k are as follows.

$$\begin{cases} x_{k|k} = \bar{E}_k + K_k (z_k'^i - \bar{E}_{z|k}) \\ C_{k|k} = \bar{C}_{xz|k} - K_k \bar{C}_{z|k} K_k^T \end{cases} \tag{16}$$

Finally, based on the noise covariance matching method, the adaptive update operation is carried out on the process noise covariance matrix σ_k and the observation noise covariance matrix φ_k. In this paper, the adaptive forgetting factor method is adopted to update the noise covariance matrix. The specific implementation method is as follows.

$$\begin{cases} \sigma_k = \rho\sigma_k + (1-\rho)(x_i - \hat{x}_i)(x_i - \hat{x}_i)^T \\ \varphi_k = \rho\varphi_k + (1-\rho)(z_i - \hat{z}_i)(z_i - \hat{z}_i)^T \end{cases} \tag{17}$$

In the formula, ρ is the forgetting factor, with a value range of [0, 1]. The specific values are adjusted according to different experimental situations. x_i and z_i represent the true states and observed values at the current moment, while $\hat{x}_i$ and $\hat{z}_i$ represent the predicted states and predicted observed values at the current moment. By repeating the above steps until no new data is generated, the robot data processed by the adaptive untraced Kalman filtering algorithm can be obtained once.

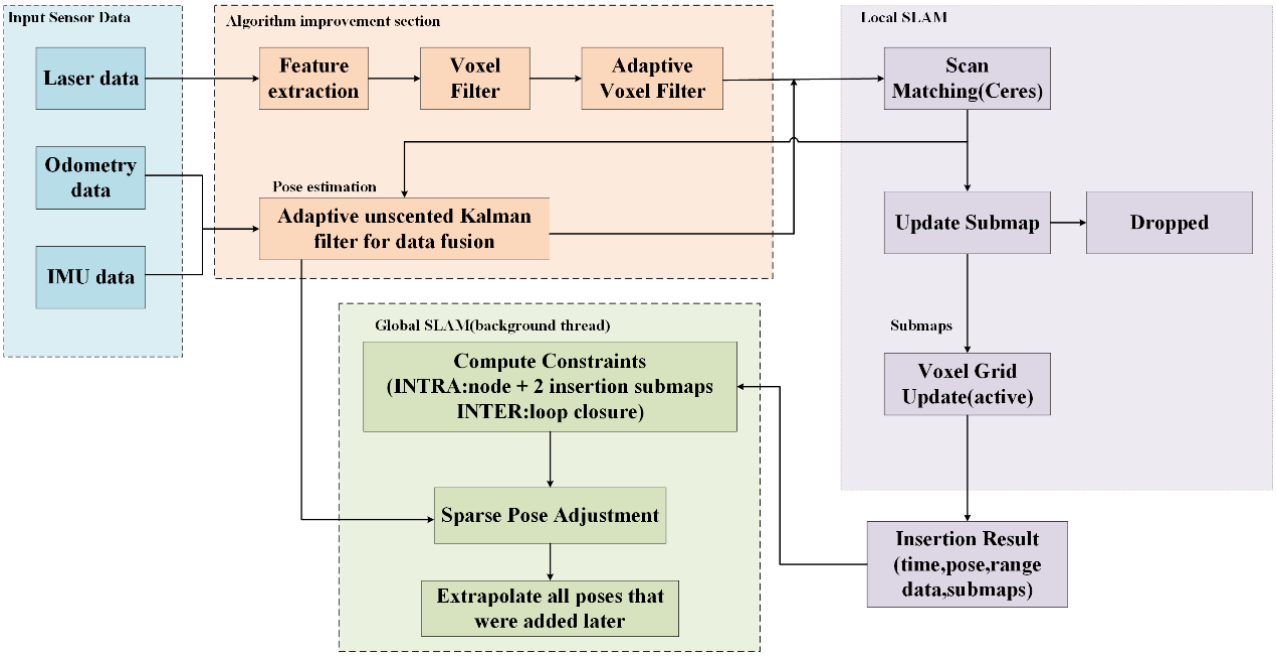


Figure 5. Improve the structure of the cartographer algorithm

During the mapping operation, by combining the pose estimation result obtained from the adaptive untraced Kalman filtering as a constraint condition with the radar point cloud information after feature extraction, an optimized map and robot motion trajectory are obtained. The structure of the improved Cartographer algorithm is shown in Figure 5.

4. MAPPING VERIFICATION

4.1. Simulation Verification Based on Gazebo

Gazebo is a commonly used simulation software for robot technology, which can provide nearly realistic physical simulations and sensor data simulations in detail [23]. Build an indoor environment 10 meters long and 10 meters wide in gazbo, place two folded walls to simulate the walls in a real indoor environment, and put two obstacles at the corner at the same time. The simulation environment is shown in the following figure.

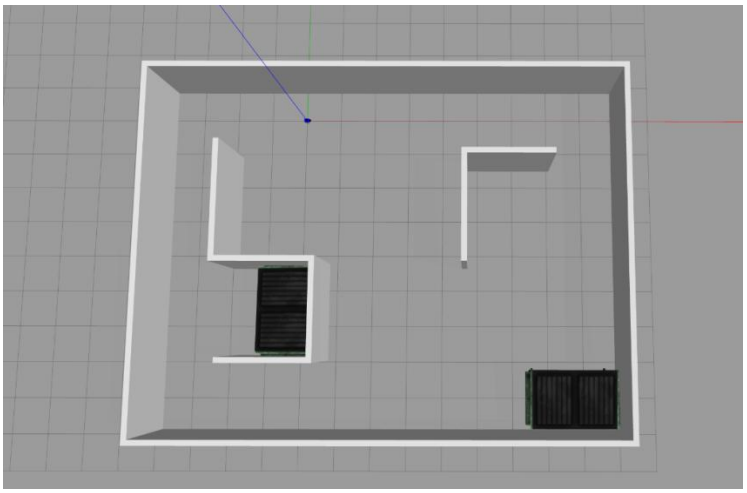
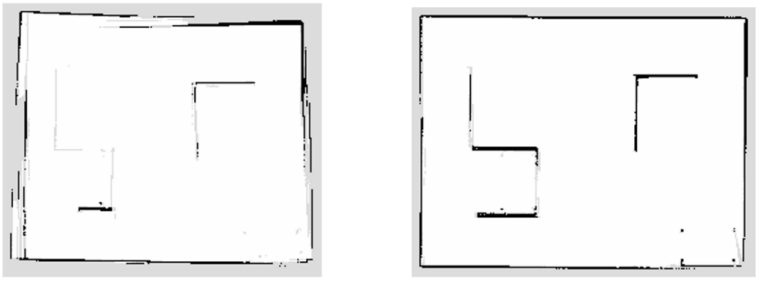


Figure 6. Gazebo simulation environment

The simulation car is equipped with lidar, IMU and odometer, and is controlled to run at a speed of 0.3m/s. Mapping operations are carried out respectively using the original Cartographer algorithm and the improved Cartographer algorithm, and the effects are as follows.



(a) Original algorithm (b) Improved algorithm

Figure 7. Comparison of mapping results

By comparing the mapping effects of the two algorithms, it can be found that the mapping effect of the Cartographer algorithm optimized by combining the point cloud information after radar feature extraction with the AUKF algorithm is significantly better than that of the original Cartographer algorithm. It is indicated that the radar point cloud data extracted through feature values can accurately contain environmental information. Meanwhile, the AUKF algorithm can better fuse the data of each sensor together and can represent the motion information of the robot in the environment more accurately. So as to ensure that more accurate construction can be achieved in the subsequent SLAM submap construction part and better mapping effects can be obtained.

4.2. Mapping Experiments in Real Environments

The experimental platform used in this paper is an intelligent car equipped with high-precision sensors such as lidar, intelligent processors and imu. As shown in the following figure.

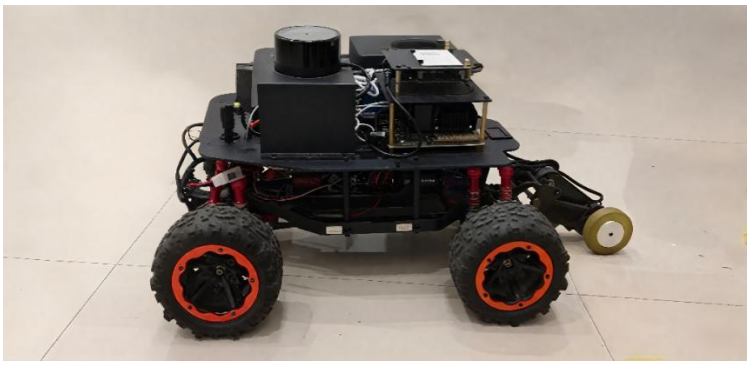


Figure 8. Experimental platform

The upper computer control system is ubuntu 22.04 version. Among them, the processor adopts the Huawei Ascend 310 series AI processor, which can provide a relatively high level of computing power. A 9-axis IMU attitude sensor is adopted to form a combined mode that integrates a 3-axis gyroscope, a 3-axis accelerometer and a 3-axis magnetometer. It can provide the deflection Angle by utilizing the Earth's magnetic field with relatively high precision. The specific parameter information of the lidar is as follows:

Table 1. Laser parameters

Parameters	Indicator
Angle/o	360
Scanning frequency/Hz	20
Measure frequency/Hz	10000
Measurement range/m	25

The experimental environment was a circular track formed by conical barrels. 68 conical barrels with specifications of 630*280*280mm were placed at the positions shown in Figure (a), and the physical object is shown in Figure (b).Among them, the yellow area is surrounded by yellow adhesive tape.

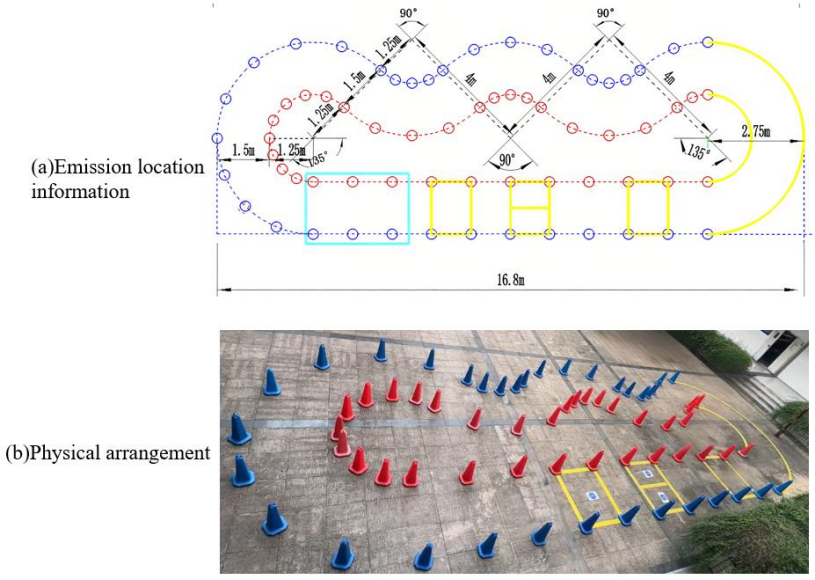
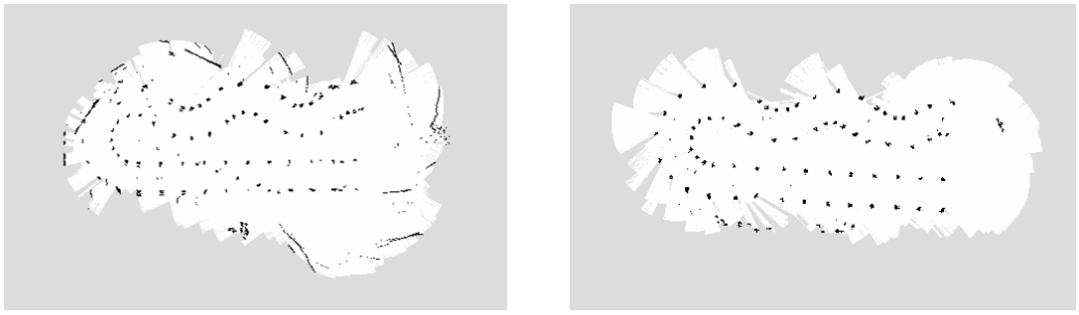


Figure 9. Setup of experimental environment

In the environment shown in the above figure, the experimental platform is controlled to perform mapping operations at a speed of 0.3m/s in the experimental environment. The original Cartographer algorithm and the improved Cartographer algorithm were adopted respectively to compare the mapping effects of the two.



(a) Original algorithm (b) Improved algorithm

Figure 10. Comparison of physical mapping results

By comparing the mapping results in the above figure, it can be found that when the original Cartographer algorithm builds a map, due to the disordered radar data and the influence of ground integrity, weeds and dust in the outdoor environment, there are many noise points. Some points have relatively large offsets, resulting in poor mapping effects. The improved algorithm can better extract the features contained in the radar point cloud in the complex outdoor environment, fuse multi-sensor data for more accurate pose estimation, and the final mapping effect is more accurate. In order to display the mapping effect more clearly, the IMU and odometer data during the mapping process were collected and visualized using MATLAB. Select a section in the map as the experimental section for data extraction. By comparing the real path of the experimental section, the pose data of the original Cartographer algorithm, and the pose data of the improved algorithm, the result of pose estimation can be visually reflected.

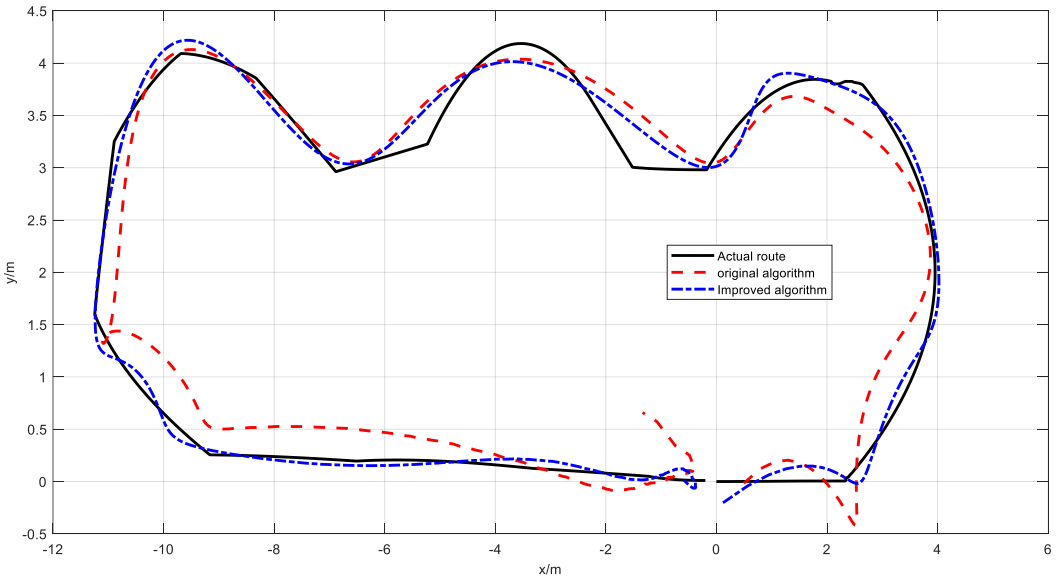


Figure 11. Comparison of position estimation

By comparing the pose data, it can be intuitively seen that the original algorithm has a large offset when performing pose estimation, and its estimation accuracy for poses is poor. The maximum offset reaches 0.536m. The offset of the optimized algorithm is significantly reduced, and the obtained pose estimation is closer to the original motion trajectory, with a maximum offset of only 0.329m. It can be seen from the above data that the improved algorithm is more suitable for real-time mapping operations in complex environments.

5. AUTONOMOUS NAVIGATION EXPERIMENT

Based on the SLAM mapping technology and combined with the path planning algorithm, the autonomous navigation of intelligent robots is realized[24]. In the current research related to autonomous driving technology, how to achieve autonomous navigation control has always been a hot issue. The most mainstream approach at present is to adopt the A* algorithm as the global path planning algorithm to plan the target path. The DWA algorithm is adopted as the local path planning algorithm for real-time obstacle avoidance and path detail adjustment. In this question, it is proposed to adopt the RT-Connect algorithm as the global path planning algorithm and the artificial potential field method as the autonomous navigation mode of the local path planning algorithm. The specific implementation process is as follows.

5.1. Construction of Path Planning Algorithm

The RRT-Connect algorithm is an improved version of the Fast Random Tree Algorithm (RRT). The RRT algorithm generates a tree at the initial position, and then continuously expands the tree structure using random sampling to cover the state space until the generation stops when the target position is reached. The RRT-Connect algorithm generates two trees simultaneously at the initial position and the target position, and accelerates the search process by connecting the two trees.

First, determine the initial position P_{start} and the target position P_{goal} in the initial stage of the algorithm. Then generate two trees T_{start} and T_{goal} respectively at the two positions.

Then random sampling is carried out. Randomly select a point P_{rand} in the entire space and use (x_{rand}, y_{rand}) to represent its coordinates. For the two previously constructed random trees, search for point P_{near} which has the closest Euclidean distance to point P_{rand}. If the coordinates of point P_{near} are represented by (x_{near}, y_{near}), then the calculation formula for the Euclidean distance is as follow.

$$d(P_{near}, P_{rand}) = \sqrt{(x_{near} - x_{rand})^2 + (y_{near} - y_{rand})^2} \tag{18}$$

Then generate new nodes. Let the step size be Δd, then extend Δd from P_{near} to P_{rand} to generate a new node P_{new}.

$$P_{new} = P_{near} + \frac{\Delta d}{d(P_{near}, P_{rand})}(P_{rand} - P_{near}) \tag{19}$$

By using the above method, two points of P_{near} can be generated, namely P_{new_start} of the initial position tree T_{start} and P_{new_goal} of the target position tree T_{goal}. First, extend from node P_{new_start} to the nearest node P_{near_goal} of tree T_{goal}, and then extend from node P_{new_goal} to the nearest node P_{near_start} of tree T_{start}. Calculate the corresponding Euclidean distance.

$$\begin{aligned} &d(P_{new\_start}, P_{near\_goal}) \\ &d(P_{new\_goal}, P_{near\_start}) \end{aligned} \tag{20}$$

When either of them is less than the step size Δd, it is considered that the two trees can be connected. Combine the two trees and generate a target path.

A global path $P=[q_0,q_1,\ldots,q_n]$ from the initial position to the target position is generated by using the RRT-connect algorithm, and the value range of n is generated according to the specific situation. The robot moves along the global path, and during the movement, the artificial potential field method is adopted for local path planning.

The artificial potential field method was proposed by Khatib in 1985. The potential function composed of the gravitational potential field and the repulsive potential field is constructed to guide the robot to perform obstacle avoidance operations and move towards the target point.

First, set the next target point in the global path as the target point q_{goal} of the current artificial potential field method, and calculate the attraction field of the target point to the robot.

$$V_{attr}(q)=\frac{1}{2}k_{attr}d(q,q_{goal})^2,\quad d(q,q_{goal})\le r_{goal} \tag{21}$$

Among them, k_{attr} is the weight coefficient of the gravitational field, representing the gravitational intensity of the target point on the robot. $d(q,q_{goal})$ represents the Euclidean distance from the current point q to the target point q_{goal}. r_{goal} represents the radius of the target area. Once entering the range of the target area, it can be considered that the target point has been reached.

The magnitudes of the gravitational force corresponding to the gravitational field are as follows.

$$F_{attr}(q)=-\nabla V_{attr}(q)=-k_{attr}d(q,q_{goal})\frac{q-q_{goal}}{d(q,q_{goal})} \tag{22}$$

When encountering obstacles, calculate the repulsion field of the obstacles on the robot.

$$V_{req}(q)=\begin{cases}\frac{1}{2}k_{req}\left(\frac{1}{d(q,b_i)}-\frac{1}{d_0}\right)^2 & ,d(q,b_i)\le d_0\\ 0 & else\end{cases} \tag{23}$$

In the formula, k_{req} is the weight coefficient of the repulsion field, representing the repulsive force intensity of the obstacle on the robot. $d(q,b_i)$ is the Euclidean distance from the current point q to the obstacle b_i. d_0 represents the influence range of the obstacle. When the distance is greater than d_0, it is considered that the obstacle does not affect the robot.

The magnitude of the repulsive force corresponding to the repulsive field is as follows.

$$F_{req}(q,b_i)=-\nabla V_{req}(q)=\begin{cases}-\frac{k_{req}\left(\frac{1}{d(q,b_i)}-\frac{1}{d_0}\right)}{d(q,b_i)^2}\frac{q-b_i}{d(q,b_i)} & ,d(q,b_i)\le d_0\\ 0 & else\end{cases} \tag{24}$$

Combining the above two equations, the total potential field $V_{all}(q)$ can be obtained as

$$V_{all}(q)=V_{attr}(q)+\sum_{i=1}^{m}V_{req}(q,b_i) \tag{25}$$

In the formula, m represents the number of corresponding obstacles in the environment.

The final corresponding resultant force is

$$F_{all}(q) = F_{attr}(q) + \sum_{i=1}^{m} F_{req}(q, b_i) \tag{26}$$

After obtaining the resultant force, the position and speed of the robot are updated by using the resultant force.

$$\begin{aligned} v_{new} &= v_{old} + \mu F_{all}(q) \\ q_{new} &= q + v_{new} \cdot \Delta t \end{aligned} \tag{27}$$

In the formula, v_{old} represents the current speed and v_{new} represents the updated speed. μ is the velocity update constant. Δt is the time step size. Then repeat the above steps to calculate the resultant force and update the motion state of the robot. When the target point is reached, update the next target point from the global path and repeat the local path planning until the final target point is reached.

When during the movement process, due to the existence of obstacles, the movement path of the robot deviates significantly from the global path previously planned by the RT-CONNECT algorithm, the current position of the robot is taken as the new initial position to re-plan the global path and generate a new global path. By combining the RT-Connet algorithm and the artificial potential field method, the fusion algorithm for road force planning is established.

Figure 12. Flow chart of path planning fusion algorithm

5.2. Verification of Autonomous Navigation Algorithms

In the map constructed based on gazebo in Article 3.1 of this paper, the simulation experiment of the path planning algorithm is carried out. The navigation speed is set at 0.3m/s. Starting from the initial position, a corner is selected as the final position. The coordinates of the target position are published in rviz. The car will achieve path planning and navigate there autonomously. The following figure shows the release location and the global path diagram of the car planning.

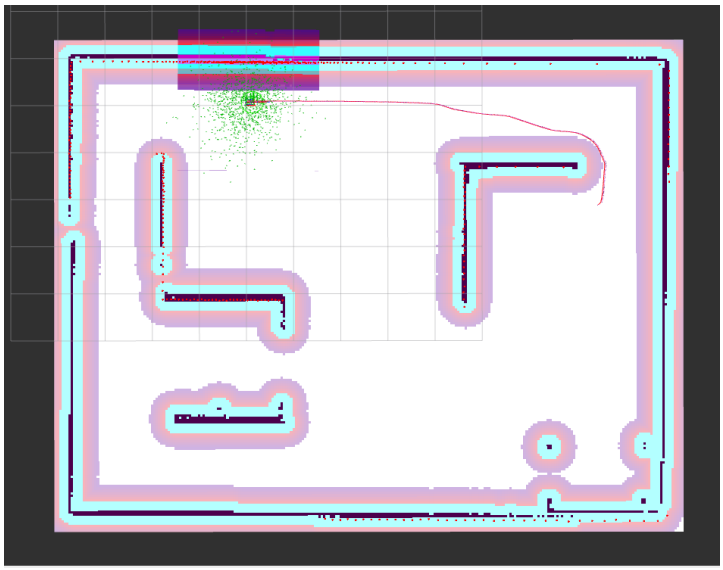
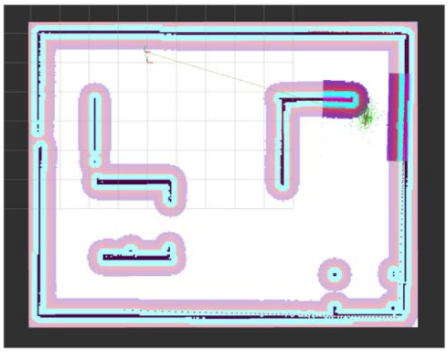
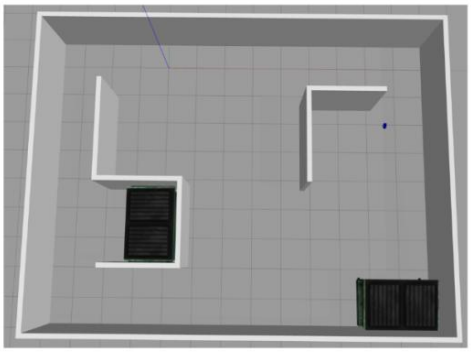


Figure 13. The global path planned by the algorithm

It can be seen that the path planned by the algorithm is close to the corner when turning, the selected path distance is short, and the effect is better. Finally, the car stopped at the target point position and was observed in gazebo at the same time. It was found that the positioning was relatively accurate and the coordinate deviation was small.



(a)The final position in RViz



(a) The final position in Gazebo

Figure 14. Final position

Using the experimental platform shown in 3.2 of this paper, the autonomous navigation test was conducted on the circular runway. By recording the coordinates of the positions of the navigation target points and the actual reached positions at the speeds of 0.1m/s, 0.2m/s, 0.3m/s, 0.4m/s and 0.5m/s of the platform, the deviations of the coordinate positions were compared. The final experimental data are shown in the following table.

Table 2. Motion deviation

Speed/(m/s)	Deviation/m
0.1	0.06
0.2	0.09
0.3	0.14
0.4	0.17
0.5	0.17

Since the coordinates in the experiment are converted into coordinate representations and include the x and y axes, linear distance is adopted to process the collected data and convert it into linear distance. It can be seen from the above table that as the speed increases, the error of navigation accuracy gradually increases, and the error basically remains within 0.2m. Due to the presence of numerous interferences in the outdoor actual environment, the data from radar, IMU and odometer will have certain deviations each time they run. It cannot be guaranteed that the position during navigation operation is exactly the same as the coordinates of the corresponding position during mapping, so there will be a certain coordinate offset. It can be seen from the experiment that the navigation accuracy is relatively high when the speed is lower than 0.3m/s, and it is suitable for navigation operations in complex outdoor environments.

6. CONCLUSION

Aiming at the problem of data distortion of lidar during the motion process, drawing on the part of feature extraction in the LIO-SAM algorithm, feature extraction is carried out on the point cloud information of lidar. Then, the adaptive untraceable Kalman filtering method is used to fuse the pose information output by the IMU and the odometer. Based on the above two points, the Cartographer algorithm is optimized and verified on the gazebo simulation platform and in the real outdoor environment. The experimental mapping results show that the improved algorithm has higher mapping accuracy, the data obtained through each sensor are fused more closely, and the pose estimation effect is better.

Then, the path planning algorithm is constructed by integrating the RRT-connect algorithm with the artificial potential field method. The autonomous navigation operation using the path planning algorithm was verified in the gazebo simulation environment and the real outdoor environment. The results show that when the running speed of the robot is lower than 0.3m/s, the navigation accuracy and positioning accuracy are relatively high. This paper presents a feasible scheme for mapping navigation based on multi-sensor fusion.

REFERENCES

[1] Smith, R., M. Self, and P. Cheeseman. "Estimating Uncertain Spatial Relationships in Robotics." Machine Intelligence and Pattern Recognition 5 (1988): 435-461.

[2] Grisetti, G., C. Stachniss, and W. Burgard. "Improving Grid-based SLAM with Rao-Blackwellized Particle Filters by Adaptive Proposals and Selective Resampling." IEEE International Conference on Robotics and Automation (2005): 2432-2437.

[3] Kohlbrecher, S., et al. "A Flexible and Scalable SLAM System with Full 3D Motion Estimation." 2011 IEEE International Symposium on Safety, Security, and Rescue Robotics (2011): 155-160.

[4] Olson, E. B. "Real-Time Correlative Scan Matching." IEEE International Conference on Robotics and Automation (2009).

[5] Hess, W., et al. "Real-Time Loop Closure in 2D LIDAR SLAM." 2016 IEEE International Conference on Robotics and Automation (ICRA) (2016): 1271-1278.

[6] Campos, C., et al. "ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual–Inertial, and Multimap SLAM." IEEE Transactions on Robotics 37.6 (2021): 1874-1890.

[7] Qin, T., et al. "A General Optimization-Based Framework for Global Pose Estimation with Multiple Sensors." Computer Science (2019).

[8] Prisacariu, V. A., et al. "A Framework for the Volumetric Integration of Depth Images." Computer Science (2014).

[9] Fu, H., et al. "Multi-Sensor Fusion Mapping and Navigation Research for Tomato Greenhouse Robot." Journal of Chinese Agricultural Mechanization 46.04 (2025): 171-178+185.

[10] Liu, T., and W. Changshui. "A Tightly Coupled Algorithm for the LiDAR and IMU in the Context of Autonomous Parking Vehicle." Journal of Electronic Measurement and Instrumentation (2025): 1-8.

[11] Li, F., et al. "Lidar SLAM Algorithm Based on Improved Loop Closure and Pose Graph Optimization." Laser & Optoelectronics Progress (2025): 1-15.

[12] Li, F., et al. "Research on Indoor Positioning Method of Mobile Robot Based on Multi-Sensor Fusion." Machinery Design & Manufacture (2025): 1-4.

[13] Dijkstra, E. W. "A Note on Two Problems in Connexion with Graphs." Numerische Mathematik 1.1 (1959): 269-271.

[14] Hart, Peter E., N. J. Nilsson , and B. Raphael. "A Formal Basis for the Heuristic Determination of Minimum Cost Paths." IEEE Transactions on Systems Science & Cybernetics 4.2(1972):28-29.

[15] Fox, D., W. Burgard, and S. Thrun. "The Dynamic Window Approach to Collision Avoidance." IEEE Robotics & Automation Magazine 4.1 (1997): 23-33.

[16] Karaman, S., et al. "Anytime Motion Planning Using the RRT*." 2011 IEEE International Conference on Robotics and Automation (2011): 1478-1483.

[17] Khatib, O. "Real-Time Obstacle Avoidance for Manipulators and Mobile Robots." Proceedings. 1985 IEEE International Conference on Robotics and Automation (1985): 500-505.

[18] Tang Qiang. “Indoor Mobile Robot Based on Improved Ant Colony Algorithm Research on Navigation Technology”. 2024. Southwest University of Science and Technology, MA thesis.

[19] MIAO Hongxia, CHEN Jialin, QI Bensheng, et al. "Path planning algorithm base on improved RRT and artificial potential field method." Automation and Instrumentation (2023): 9-14.

[20] Jia hao. "Design of SLAM and Navigation Robot Based on Cartographer Algorithm." 2019. Shandong University. MA thesis.

[21] Shan, T., et al. "LIO-SAM: Tightly-Coupled Lidar Inertial Odometry via Smoothing and Mapping." 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 2020, pp. 5135-5142.

[22] Pu Wenhao, et al. "LiDAR Point Cloud Correction and Location Based on Multisensor Fusion." Laser & Optoelectronics Progress 60. 24 (2023): 275-282.

[23] Singh, M., et al. "Comparative Study of Digital Twin Developed in Unity and Gazebo." Electronics 14.2 (2025): 276.

[24] Pan Shuaijiang, et al. "Research on Indoor Positioning and Map Construction of Mobile Robot Based on ROS." Mechanical Management and Development 40. 04 (2025): 251-254.