Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 3340529, 10 pages
https://doi.org/10.1155/2022/3340529

Research Article

Research on Robot Positioning and Navigation Algorithm Based on SLAM

Yue Dai

Experimental Training Center,Chuzhou Vocational and Technical College, Chuzhou 239000, China

Correspondence should be addressed to Yue Dai; daiyue@chzc.edu.cn

Received 15 December 2021; Revised 2 January 2022; Accepted 3 January 2022; Published 31 January 2022

Academic Editor: Xin Ning

In the industrial field, industrial robots have taken over the heavy lifting that used to be done by traditional handicraft assembly lines, greatly freeing up human resources and improving production efficiency and safety. As a result, the focus of this paper is on the SLAM-based robot localization and navigation algorithm (simultaneous localization and mapping). An attitude estimation algorithm based on KF (Kalman filtering) information fusion of vision SLAM and IMU (Inertial Measurement Unit) is proposed, and the ORB-SLAM algorithm is studied and perfected. The fusion of the two postures improves the accuracy and frequency of the robot's attitude estimation during motion. In addition, PSO (Particle Swarm Optimization) technology is used to optimize the resampling process, and PSO optimizes the particle set to alleviate the problem of particle degradation and exhaustion caused by resampling in the FastSLAM algorithm. Finally, the algorithm is verified to meet the requirements of positioning and composition accuracy, as well as the feasibility and effectiveness of robot autonomous navigation, using the open simulation platform.

1. Introduction

Intelligent mobile robot is a kind of robot with self-planning, self-organization, and self-adaptability that works in complex environment [1]. The goal of intelligent navigation research is to make robots move purposefully, complete specific tasks, and perform specific operations without human intervention. SLAM (Simultaneous Localization and Mapping) technology is the foundation of mobile robot's environmental exploration and intelligent navigation, and the determination of robot's own position and pose is the premise of its autonomous navigation [2, 3]. Therefore, how to build the current environmental map independently in uncertain space, and update and determine the robot's own position and pose cyclically and conduct high-precision navigation at the same time of incremental mapping, that is, how to accurately realize the SLAM technology of mobile robot, is a hot issue that needs to be solved and improved urgently in the field of robotics.

The current focus of SLAM research is on increasing computational efficiency, data association, and loop closure [4]. Although domestic research on this problem was delayed, many advances have been made in map representation, map creation, positioning and navigation, and other areas. SLAM based on sonar and visual information fusion is studied in the literature [5, 6], which effectively improves the accuracy and robustness of positioning and map creation. The position of the carrier in an inertial navigation system is determined by twice integrating the acceleration with time. The autonomy and concealment of this navigation method are significant advantages. Simultaneously, as robot technology advances, so do robot-related algorithms, which are constantly evolving and updating. These technological advancements and hardware upgrades have laid the groundwork for autonomous indoor robot navigation and positioning [7]. Indoor mobile robots used for mapping, navigation, and positioning in unfamiliar environments have not been widely adopted due to a variety of technical limitations. As a result, obtaining the mobile robot's own pose information in real time and accurately indoors is critical for fully autonomous composition and navigation in any indoor environment [8].

In this paper, the SLAM based on lidar is studied with the mobile robot as the carrier, so as to realize the autonomous positioning and navigation of the mobile robot. By analyzing the principle and system structure of SLAM, the motion model of mobile robot and the sensor observation model based on point features and line features are established. On the basis of PF (particle filter), an unscented Fast SLAM algorithm based on particle swarm optimization is proposed, and the feasibility and effectiveness of the algorithm are verified by experiments. Firstly, the algorithm combines UPF (unscented particle filter) with UKF (unscented Kalman filter) algorithm, uses UPF to estimate the posterior probability of robot pose, and uses UKF to deal with the map estimation part, which avoids the operation of Jacobian matrix and the linearization of nonlinear function, and greatly improves the accuracy of the algorithm.

According to the main contents of the above research, the main contributions and innovations of this paper are as follows: the fusion system is studied by loose coupling method, and the pose fusion algorithm of visual SLAM and IMU (Inertial Measurement Unit) is realized, which improves the output frequency, accuracy, and robustness of robot posture and pose.

2. Related Work

There are many kinds of mobile robot navigation and positioning methods, mainly including visual navigation and positioning, electromagnetic navigation and positioning, GPS global positioning, and light reflection navigation and positioning. These positioning methods use different sensors and cooperate with the robot to move dynamically, so that the robot can realize high-precision positioning and navigation under the restriction of different environments. Literature [9, 10] puts forward the square-root unscented KF (Kalman filter) based on UKF, which expands the restricted range of UKF and can be applied to Gaussian regression process. Literature [11] puts forward a new idea; that is, traditional UKF is combined with a new hotspot algorithm ParticleFilter, thus forming a high-precision unscented particle filter algorithm. Reference [12] puts forward that UKF is used to deal with the location problem of nonlinear transformation. Literature [13, 14] uses particle filter combined with mobile robot odometer and sensor model to locate. The probability is expressed by the scattered particle set distribution. By looking for a series of randomly scattered particles in the working environment, the robot position and the distribution of obstacles are approximated by the function of probability density, and the position distribution is accurately approximated after repeated iterations. Literature [15] optimizes the sampling process by reducing the number of algorithmic sampling and then puts forward the Monte Carlo method based on Gaussian mixture model. Literature [16] proposes improving the particle filter algorithm by combining the matching process of fuzzy map and the resampling process of particle filter. This algorithm can achieve the purpose of reducing the influence of various uncertain factors. However, due to the relatively complex actual environment, the demand for the number of sampling samples increases, which virtually greatly increases the calculation amount of the algorithm and leads to the slow implementation of the algorithm. In addition, sample dilution caused by resampling stage is also an urgent problem to be solved.

There are still few mobile robots that can truly realize autonomous positioning and navigation. In the navigation of mobile robots, autonomous line inspection is the main way in industry. The mobile robot developed in literature [17] for logistics storage or large-scale automatic production line can move the shelves full of goods back and forth between any points, control the robot to put the best-selling goods to the front through the centralized control system of the warehouse, and plan the optimal path of the mobile robot. Literature [18] using SLAM technology to provide map update for automobiles can make it unimpeded in any range of the world and even find the destination for automobiles. Literature [19] discusses SLAM based online features, and the reliability of the algorithm is verified by simulation experiments. Literature [20] studies SLAM based on laser ranging and monocular vision in indoor environment to realize autonomous navigation task of mobile robot in unknown environment. Literature [21] uses laser rangefinder to obtain information, and completes the autonomous navigation of mobile robot in indoor environment without any prior conditions. FastSLAM is a combination algorithm, the central idea of which is to use recursive algorithm to estimate robot posture and calculate the complete posterior distribution of map signposts, so as to realize map creation and location. Literature [22] puts forward an optimized resampling algorithm to make the sub-plasma move towards the superior particles. This algorithm not only greatly reduces the degradation of particle swarm, but also is of great significance to preserve the diversity of particle swarm. With the independence of data association, FastSLAM is widely used in complex environment because of its fast computing speed and strong anti-interference ability.

To calculate its own pose, computer vision-based positioning technology primarily uses the image frames collected by the camera. The feature detection algorithm will extract feature points and descriptors for each image frame. The rotation matrix and translation vector between the object coordinate system and the camera coordinate system can be obtained under certain constraints, and the corresponding visual features can be extracted, based on the corresponding relationship between the positions of these feature points and descriptors in the image, that is, through the matching relationship between two images. Finally, a specific algorithm can be used to find the target location.

3. Research Method

3.1. SLAM Description. SLAM is to build an environmental map according to the robot's position and then determine the robot's position from the known map. The two complement each other. It can be described as using sensor information to simultaneously solve the problem of establishing an environmental map and determining its own orientation.

The motion model of the mobile robot indicates that the robot's posture changes with time under the influence of control input u_t and process noise δ_t. The motion model of the robot can be expressed as

$$x_t = h(x_{t-1}, u_t) + \delta_t. \tag{1}$$

The robot's state vector x_t represents the robot's posture state at time t, including various parameters such as position, heading angle, and speed. h is a state transition model, which is usually nonlinear. u_t is the control input information, and δ_t represents the process random noise in the process of robot movement, which is independent of time, assuming that its mean value is 0 and its covariance is Q_t.

In an unknown environment, the mobile robot starts from an unknown position, uses its own sensors to obtain environmental information during the movement, and uses this information to create and update the environmental map constantly, and at the same time determines its position and posture according to the sensor information and the created environmental map. The location and mapping of SLAM can be simplified as a process of "estimation"-"observation"-"correction." The system state diagram is shown in Figure 1.

x_t represents the state of the mobile robot at time t, m_j represents the environmental characteristics, u_t is the control input of the robot, and z_t is the observed measurement of the sensor.

The state x_t of the robot at time t is related to the system state x_{t-1} at the previous time, the input u_t at the current time, and the observed measurement $z_{t,k}$ of the sensor at the current time.

Because the sensor will have some errors, it is necessary to correct the pose by observing external environmental landmarks, calculating the robot's pose based on the landmarks observed, and using the pose calculated by landmarks to estimate the robot's original estimated pose. The more environmental features observed, the smaller the estimation error. The system framework diagram of SLAM is shown in Figure 2.

The core of SLAM is to estimate the position and pose of mobile robot and the position of environmental landmark, which mainly depends on the perception of environmental information by measuring sensors. At present, the sensors that can sense environmental information in SLAM mainly include lidar, camera, sonar sensor, and so on. Among them, laser and sonar equidistant sensors were used to solve SLAM problem earlier.

3.2. Robot Positioning and Navigation Algorithm

3.2.1. Attitude Fusion Algorithm Based on KF. ORB-SLAM is a SLAM system that can operate in large-scale, small-scale, indoor and outdoor environments and supports monocular, binocular, and RGB-D modes. This system is similar to the PTAM algorithm in that it uses a multi-thread approach to separate tracking, local mapping, and loop detection into three threads, resulting in increased efficiency and performance.

To begin, compare the current frame to all of the frames in the candidate closed-loop frames; then compare and score based on the algorithm's conditions. The closed-loop frame can be obtained once the score exceeds the threshold, and then the back-end optimization can begin. Finally, enter the closed-loop correction stage, which requires stopping the local mapping thread and global bundle adjustment to complete the closed-loop fusion and essence map optimization.

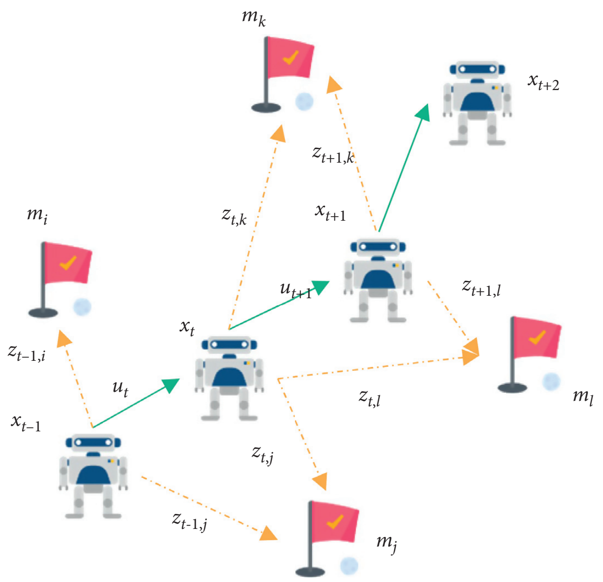

FIGURE 1: SLAM system state diagram.

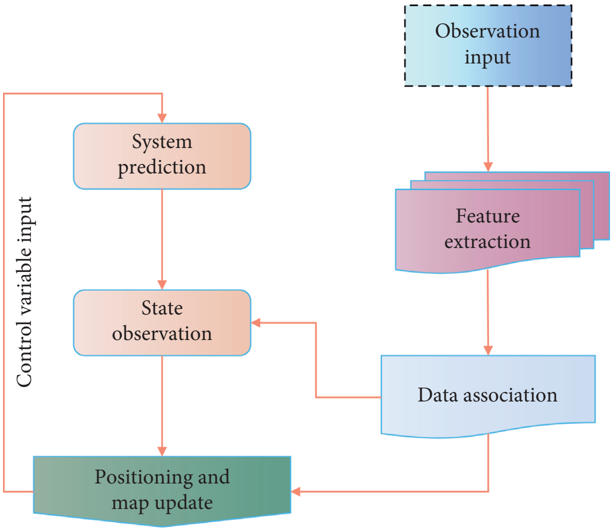

FIGURE 2: SLAM system framework diagram.

In this paper, the improved ORB-SLAM algorithm based on RGB-D interface is used, and the main functions added are real-time map building, 3D dense point cloud map building, loose coupling attitude estimation with IMU (Inertial Measurement Unit), map saving and reading, and visual repositioning after overloading the map [23].

Through the visual sensor, the position and attitude of the robot at any position can be obtained by using the visual SLAM algorithm, with a total of six degrees of freedom. Through the output of IMU, the six-degree-of-freedom information of the robot can also be obtained simultaneously.

Visual SLAM has the characteristics of low running frequency, dependence on lighting environment, and easy to lose objects, which leads to large attitude error. IMU's attitude calculation frequency is high, and its accuracy is higher than that of visual SLAM. However, for position calculation, it is easy to increase the accumulated error with time. In this section, the attitude fusion estimation algorithm of visual inertial navigation based on KF will be introduced in detail. The algorithm flow chart is shown in Figure 3.

In the practical application of inertial navigation, its output conforms to a nonlinear system. In order to process the KF more intuitively, it is necessary to obtain the state equation which accords with the linear system, that is, needing to linearize KF [24]. However, using error state quantity to represent system modeling can reduce data redundancy and improve operation efficiency, which is a common practice at present.

Error quaternion is used to define the error quantity, instead of using the arithmetic error between rotation and rotation estimation. Here, the error quaternion two represents the slight rotation between an estimated quantity and the real state quantity, and this error is defined by multiplication, as shown in the following formula:

$$\delta\hat{q} = q \otimes \hat{q}^{-1}. \tag{2}$$

As the error quaternion two is a tiny rotation, it can be expressed by an approximately small angle error, and define a small angle error $\delta\theta$ as follows:

$$\delta q = \begin{bmatrix} \delta q \\ \delta q_w \end{bmatrix} = \begin{bmatrix} \hat{k}\sin\left(\dfrac{\delta\theta}{2}\right) \\ \cos\left(\dfrac{\delta\theta}{2}\right) \end{bmatrix} \approx \begin{bmatrix} \dfrac{1}{2}\delta\theta \\ 1 \end{bmatrix}, \tag{3}$$

where δq is the imaginary part of micro rotation and δq_w is the real part of micro rotation. At this time, the error state quantity δx can be expressed by the following formula, and Δb is the error:

$$\delta x = \begin{bmatrix} \delta\theta \\ \Delta b \end{bmatrix}. \tag{4}$$

Since then, the state equation of inertial navigation system in real situation can be approximately expressed by the error state equation.

Because the frequency of the IMU's attitude estimation output is so high, it is easy to accumulate larger attitude accumulation errors over time. The robustness of the entire system can be improved by using the low frequency output result of visual SLAM as the attitude output value of the IMU.

The visual attitude estimation value q_c^w represents the attitude transformation from the camera coordinate system to the world coordinate system and also represents the observation value z of the filter. The expression of the available observation equation H of the joint observation matrix is as follows:

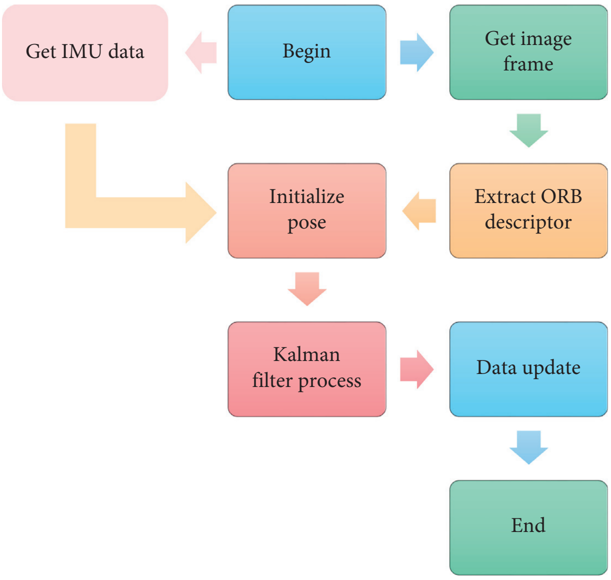

Figure 3: Flow chart of vision inertial navigation attitude fusion.

$$H = \left[\, 0_{3\times 3} \quad I_{3\times 3} \,\right]. \tag{5}$$

From then on, the updating process of KF can be obtained from the observation matrix and observation noise variance matrix [25], and the specific steps are shown in the following formula:

$$\begin{aligned} K_k &= P_{k|k-1}H_k^T\left(H_k P_{k|k-1}H_k^T + R_k\right)^{-1}, \\ \hat{x}_k &= \hat{x}_{k|k-1} + K_k\left(z_k - H_k\hat{x}_{k|k-1}\right), \\ P_k &= \left(I - K_k H_k\right)P_{k|k-1}. \end{aligned} \tag{6}$$

A distinct feature that is fixed in the environment is referred to as a feature. Corners, surfaces, rectangles, and points are just a few examples of physical features. Natural and man-made topographical features can both be considered features as long as sensors can measure them consistently and reliably. Physical visible and sensible features are not always the characteristics, according to this definition. A feature is mathematically represented by a vector that defines its position and other properties.

In this paper, point features are used to describe the environmental features measured by sensors. The i th feature in the environment can be expressed by p_i, which is defined as

$$p_i = \begin{bmatrix} x_i \\ y_i \end{bmatrix}. \tag{7}$$

From time k to time $k+1$, the change of feature state is negligible, so the feature can be regarded as static, which is shown as follows:

$$p_i(k+1) = p_i(k) = p_i. \tag{8}$$

Compared with the robot model, the feature model has no additional uncertainties. Although the location of the feature may be uncertain, this uncertainty does not increase with time.

3.2.2. Unscented FastSLAM Based on PSO. FastSLAM is an algorithm combining PF and EKF (extended Kalman filter). Its basic idea is to use Rao-Blackwellized decomposition to decompose the SLAM problem of mobile robot into a single robot positioning problem and an environmental map construction problem based on robot pose estimation.

FastSLAM mainly estimates the pose of mobile robot by PF and estimates the position of environmental landmark by EKF. Under the condition that the path of the mobile robot is known, the estimation of each environmental landmark is independent of each other, so the estimation of the environmental map can be decomposed into several independent feature estimates; that is, the pose of the mobile robot and the posterior probability distribution of the map can be decomposed as follows:

$$\begin{aligned} p\left(s^k, \Theta | z^k, u^k, n^k\right) &= p\left(s^k | z^k, u^k, n^k\right) p\left(\Theta |^k, z^k, u^k, n^k\right) \\ &= p\left(s^k | z^k, u^k, n^k\right)^k \prod_{i=1}^{N} p\left(m_i, s | z^k, u^k, n^k\right). \end{aligned} \tag{9}$$

The unscented Kalman filter (UKF) approximates a nonlinear function's probability distribution rather than the nonlinear function itself. UKF does not use the EKF linearization process and does not calculate the Jacobian matrix. Due to the increased state vector dimension, UKF will increase the amount of computation. According to the latest research, the amount of computation performed by UKF and EKF is on par, but the algorithm's consistency and accuracy are superior to those of EKF.

In UKF, the gain K_t is defined as

$$K_t = P_{x_t z_t} P_{z_t z_t}^{-1}. \tag{10}$$

The posterior estimation of the state vector can be obtained by using Kalman gain and the predicted state vector and observation vector obtained by UT (unscented transformation) calculation:

$$\widehat{X}_t = X_{t|t-1} + K_t\left(z_t - z_{t|t-1}\right). \tag{11}$$

The error covariance matrix of state posterior probability estimation is

$$P_t = P_{t|t-1} - K_t P_{z_t z_t} K_t^T. \tag{12}$$

UKF uses a few Sigma points to obtain the statistical characteristics of the probability distribution of nonlinear functions based on UT and KF. As a result, UKF can handle nonlinear functions better than EKF and has higher estimation accuracy and algorithm consistency. Particle swarm optimization (PSO) tracks a target using particle cooperation technology, which improves target estimation accuracy. Furthermore, the number of particles correctly estimated for robot position increases while the number of particles incorrectly estimated decreases. The problem of particle degradation is solved by preventing incorrect particle replacement.

Using the expansion formula, the state vector of the robot in the i th particle is expanded and initialized to $X_{t-1}^{a(i)} = \left[X_{t-1}^{(i)T} \;\; 0 \;\; 0 \right]^T = \left[X_{x,t-1}^{(i)} \;\; X_{y,t-1}^{(i)} \;\; X_{\theta,t-1}^{(i)} \;\; 0 \;\; 0 \right]^T$. The covariance matrix of the expanded state vector is

$$P_{t-1}^{a(i)} = \begin{bmatrix} P_{t-1}^{(i)} & 0 & 0 \\ 0 & Q_t & 0 \\ 0 & & R_t \end{bmatrix}, \tag{13}$$

where $X_{t-1}^{(i)T}$ is the original state vector of the robot in the i th particle at time $t-1$, and $X_{x,t-1}^{(i)}, X_{y,t-1}^{(i)}, X_{\theta,t-1}^{(i)}$ corresponds to the x coordinate, y coordinate, and heading angle of the robot state vector, respectively. The corresponding covariance matrix of is $P_{t-1}^{(i)}$. Q_t, R_t are the process noise covariance matrix and the observation noise covariance matrix, respectively.

PSO-UFastSLAM uses UKF to update the environmental features. If the observed features are the existing features in the map, the features will be updated. If the observed features are new features, expand the new features into the map.

Sigma points are defined by the latest feature mean and covariance, assuming that the mean and covariance of the ith feature in the environment are $\mu_{n,t-1}^{(i)}, \sum_{n_t,t-1}^{(i)}$, respectively, where the covariance is a 2×2-dimensional matrix and the dimension of the feature state is $n = 2$, so the symmetric sampling strategy is adopted to sample $2n + 1$ Sigma points:

$$\begin{aligned} \chi^{(0)(i)} &= \mu_{n_t,t-1}^{(i)}, \\ \chi^{(j)(i)} &= \mu_{n_t,t-1}^{(i)} +_i, \left(\sqrt{(n+\lambda) \sum_{n_t,t-1}^{(i)}} \right) i = 1, 2, \ldots, n, \\ \chi^{(j)(i)} &= \mu_{n_t,t-1}^{(i)} - \left(\sqrt{(n+\lambda) \sum_{n_t,t-1}^{(i)}} \right)_{i-n} \quad i = n+1, \ldots, 2n, \end{aligned} \tag{14}$$

$\lambda = \alpha^2 (n + \kappa) - n$, where $\alpha = 0.01$ and $\kappa = 0$ is usually set in feature estimation.

In PSO-UFastSLAM, UKF is used to generate the suggested distribution, which has higher estimation accuracy, but the same resampling process and standard FastSLAM have the same particle degradation phenomenon. In this paper, PSO algorithm is introduced into resampling, and particles are processed by PSO after resampling.

If the target selection is wrong, although the diversity of particles is maintained, the particles gather around the wrong target particles. Select that PSO target can determine the local optimal value p_{best} of each particle and the global optimal value g_{best} of the whole particle:

$$p_{\text{best}} = \arg\min_{x_i^k} \left\| x_i^k - m_x \right\|, \tag{15}$$

where $m_x = 1/N \cdot \sum_{i=1}^{N} x_i^1$ is the PSO target and k is the number of iterations. x_i^1 is the initial position of the i th particle at time t.

$$g_{\text{best}} = \arg\min_{x_i^k} \left\| x_i^k - m_x \right\|. \tag{16}$$

$i = 1, 2, \ldots, N$ represents the i-th particle.

Optimization makes particle positioning more accurate, and then by redistributing particle weights, the accurate distribution of particles can be obtained, thus making particles converge and improving the performance of the whole algorithm.

4. Analysis and Discussion

4.1. Comparative Analysis of Vision Inertial Navigation Attitude. The fusion algorithm adopted in this paper is applied to pose estimation of mobile robot in indoor environment. Among them, the camera of the mobile robot is Kinect2 depth camera, and the true value of attitude estimation is obtained by Optitrack system. In this section, the attitude results of the experiment of fusion algorithm are analyzed and compared, and the effect of system mapping after attitude fusion is given.

In the process of fusion, first of all, pay attention to coordinate system alignment caused by IMU and Kinect depth camera installation position. Because of the relative translation and rotation between them, it is necessary to unify their coordinate systems through the transformation of coordinate systems; otherwise the accuracy of the fusion algorithm will be substantially affected.

Because the output frequency of the Kinect depth camera is only 30 Hz, which is significantly lower than the output frequency of the IMU, and because the Kinect depth camera lacks a hardware trigger mechanism, the time stamps of topics published by the IMU and Kinect are mapped one by one using the time filter mechanism in ROS (Robot Operating System), and the data is aligned by software to complete the time synchronization. The experiment was completed in an indoor environment, demonstrating that the attitude estimation algorithm could be implemented on an indoor mobile robot. At the same time, RMSE (root mean square error) and maximum error are used for error analysis in order to accurately evaluate the experimental results of attitude fusion.

Among them, Figure 4 shows the visual collection process of robot mapping.

From the results of Figure 4, the accuracy of the attitude information output by IMU can meet the requirements in a short time, but after long-term operation, the output value drifts due to the increase of accumulated error. However, the ORB-SLAM algorithm based on depth vision has some errors after a long run, but its attitude error value is stable in an acceptable range.

The attitude fusion algorithm designed in this paper uses the attitude information obtained by vision as the observed value to correct the output value of IMU. The error statistics results are also given in Figure 5. Compared with the improved ORB-SLAM algorithm, the fusion algorithm designed in this paper has a certain improvement in accuracy.

4.2. Analysis of Simulation Results. In order to verify the effectiveness and correctness of PSO-UFastSLAM algorithm, this paper studies the open simulation environment of Tim Bailey and carries out Monte Carlo experiments in sparse signpost environment. The purpose of SLAM is to accurately locate the robot and build the map.

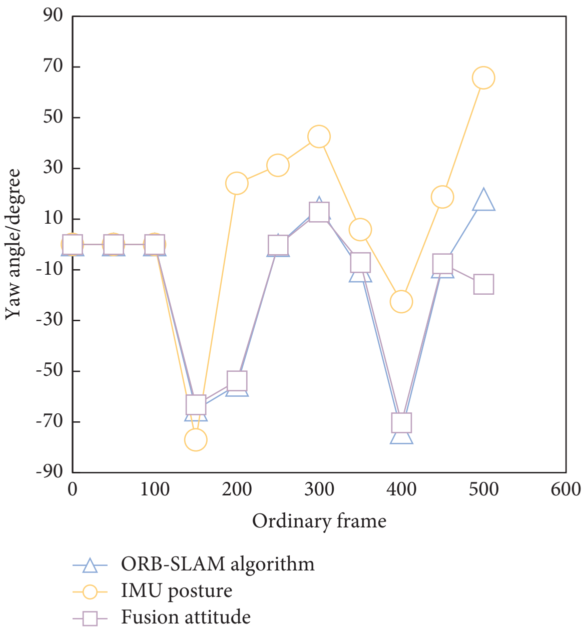

Figure 4: Visual inertial navigation attitude comparison chart.

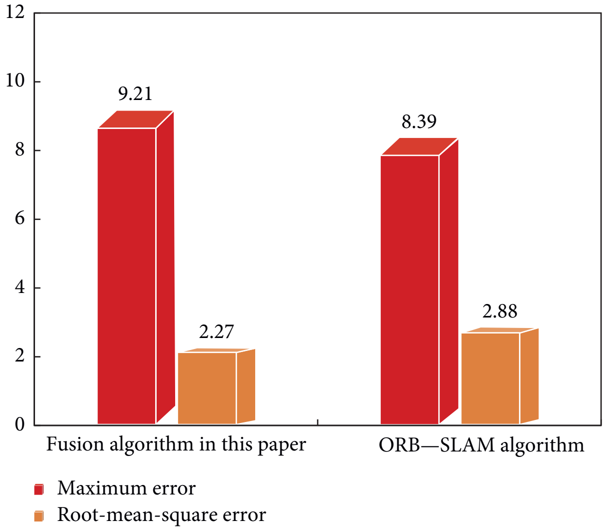

Figure 5: Attitude error result of visual inertial navigation experiment.

In this paper, use RMS (root mean square) p_{rms} of robot pose and RMS p_{rms} of environmental feature estimation to measure the estimation performance of the algorithm, and RMS of robot pose and environmental feature of the three algorithms, as shown in Figure 6.

It can be seen from Figure 6 that UFastSLAM can better estimate robot pose and environmental characteristics, and UFastSLAM algorithm has better accuracy for solving SLAM problem, among which PSO-UFastSLAM has the highest accuracy for robot pose and environmental characteristics estimation.

In order to express it more clearly, the environmental characteristics of RMSE and FastSLAM, UFastSLAM, and PSO-UFastSLAM algorithms of robot pose are compared in

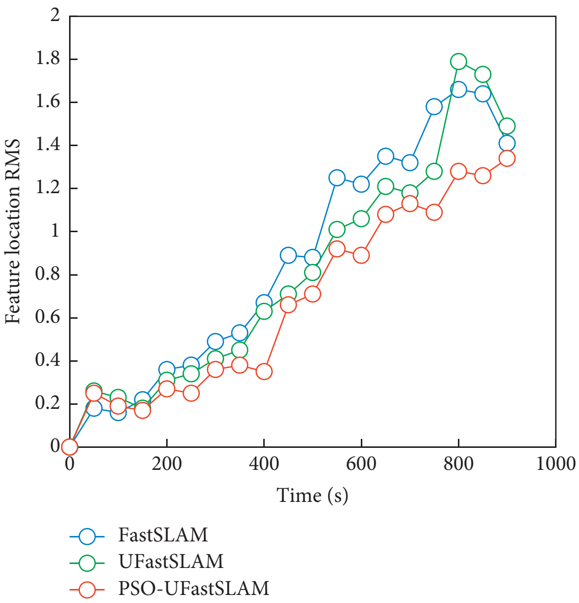

Figure 6: Environmental characteristic RMS.

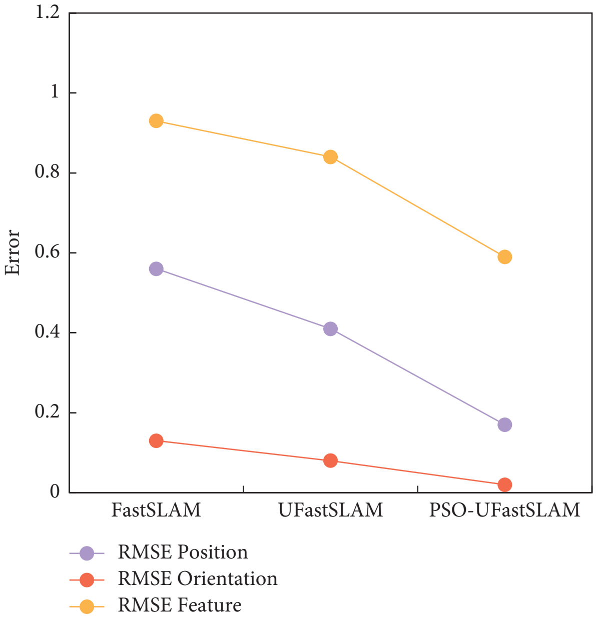

Figure 7: RMSE comparison.

turn. As shown in Figure 7, it can be seen that the error of PSO-UFastSLAM algorithm is smaller than that of other algorithms.

As can be seen from the above description, the estimation accuracy of UFastSLAM and PSO-UFastSLAM is higher than that of FastSLAM, and UT produces more accurate estimation results by avoiding the calculation of Jacobian matrix and nonlinear function linearization. At the same time, the particle set is concentrated in the true posterior probability distribution in the PSO-UFastSLAM algorithm due to the improved resampling algorithm, allowing the nonlinear system to be estimated more accurately and the estimation effect to be obtained more accurately. The PSO-UFastSLAM algorithm improves estimation accuracy and solves the problem of particle degradation by optimizing the resampling process, using particle cooperation technology to obtain the particle target position, and then using the PSO algorithm to gather the particles in the high likelihood region, keeping enough particles to accurately estimate the robot pose. Figure 8 shows the particle diversity of the three algorithms at different times.

It can be seen from Figure 8 that, in the three algorithms, the number of different particles decays exponentially. The particle diversity of FastSLAM obviously declines, followed by UFastSLAM, and PSO-UFastSLAM can keep the particle diversity of pose estimation well in the whole process.

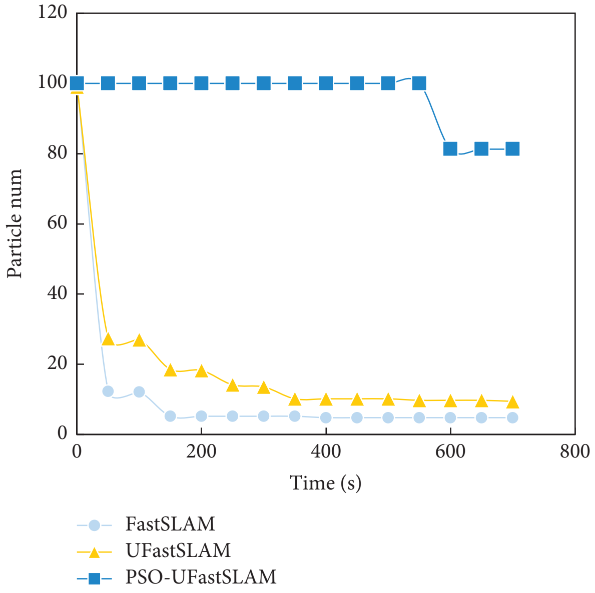

Figure 8: Different particle numbers.

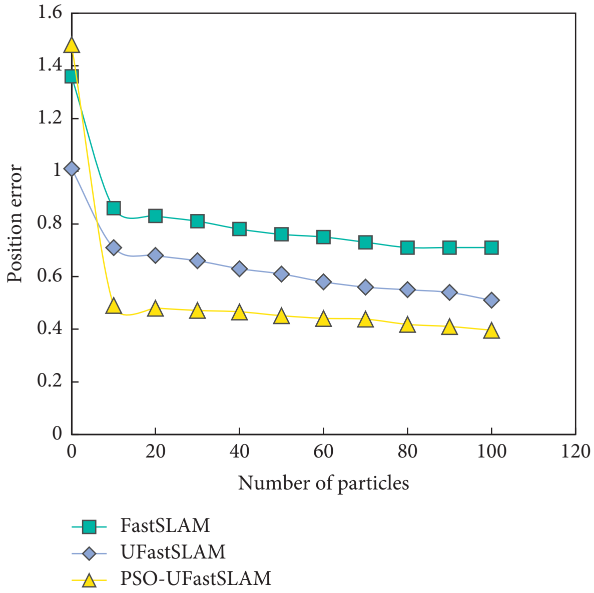

Figure 9: RMS error of pose under different particle numbers.

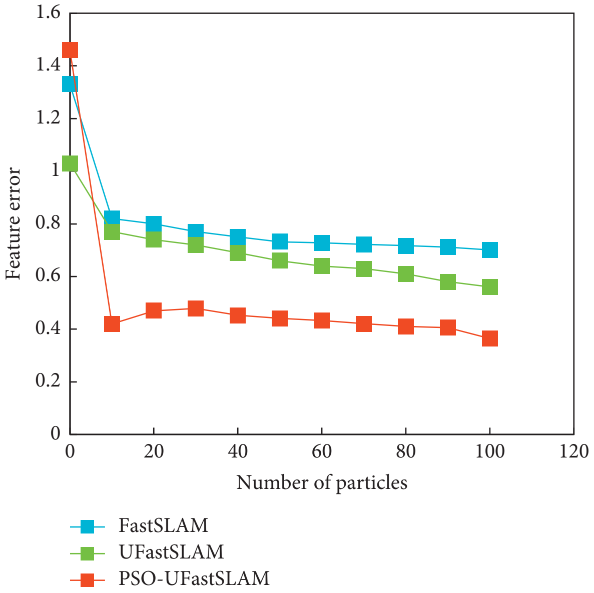

Figure 10: RMS error of environmental characteristics under different particle numbers.

Taking different particle numbers, the RMS of robot pose estimation and RMS of environmental characteristics are calculated and compared. Each algorithm has to carry out 50 Monte Carlo experiments, and the comparison results are shown in Figures 9 and 10.

Obviously, when the number of particles is greater than 10, the estimation performance of PSO-UFastSLAM algorithm is better than other algorithms.

Figure 11 shows the comparison of the running time of the three algorithms under different particle numbers. It can be seen from the figure that the calculation time of the three algorithms is on the same order of magnitude. Under the condition of the same particle number, PSO-UFastSLAM has the longest running time, followed by UFastSLAM.

PSO-UFastSLAM has a longer positioning time than the other two algorithms, but its positioning accuracy is much better with the same number of particles. The PSO-UFastSLAM algorithm can obtain the same error level as the FastSLAM algorithm with only ten particles. As a result, even though the PSO-UFastSLAM algorithm's calculation amount grows, by reducing the number of particles, the algorithm's calculation amount decreases and its real-time performance improves.

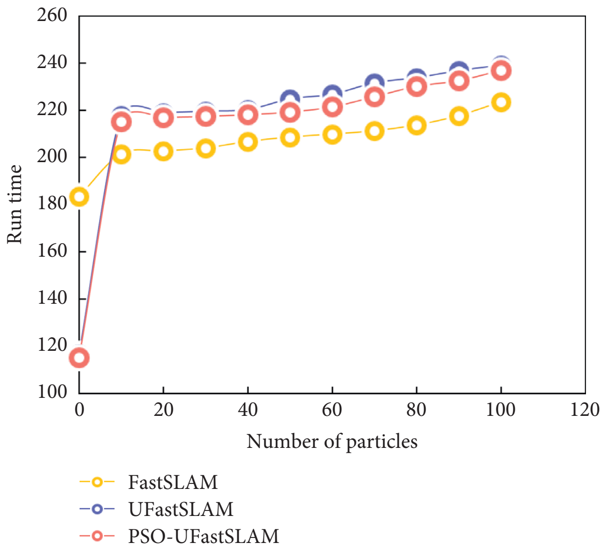

Figure 11: Algorithm running time under different particle numbers.

5. Conclusion

Localization and map creation for mobile robots is a hot and difficult topic in robot research, and it is the foundation for autonomous robot navigation in unfamiliar environments. This paper investigates the indoor mobile robot's navigation and positioning system, which is primarily based on visual sensors and supplemented by multi-sensor fusion. It primarily investigates the robot's mapping, obstacle avoidance, and navigation functions. The improved ORB-SLAM algorithm's attitude estimation results are used as filtered observations, correcting the attitude values output by the IMU. The vision and IMU fusion algorithm proposed in this paper can correct the IMU's accumulated error while also addressing the issues of low camera output frequency and easy tracking failure when rotating. In the FastSLAM algorithm, UKF is used to replace EKF, and particle cooperation technology is used to track the target, resulting in more accurate target estimation. The experimental results also show that the proposed algorithm meets the SLAM accuracy requirements.

Although the vision and IMU pose fusion algorithm proposed in this paper improved accuracy, its accuracy improvement effect was not as good as the optimization-based fusion algorithm due to its loosely coupled framework. The optimization-based fusion algorithm, on the other hand, has the drawbacks of low real-time performance and high resource usage. The next research direction of this paper is to see how to improve the system's real-time performance and robustness while also increasing accuracy.

Data Availability

The dataset used to support the findings of this study is available from the author upon request.

Conflicts of Interest

The author declares no conflicts of interest.

Acknowledgments

This study was supported by Natural Science Research Project of Anhui Universities in 2020, "Research on Robot Positioning and Navigation Algorithm Based on Image Recognition Project," under No. KJ2020A0995.

References

[1] J. Li, Z. Li, Y. Feng, Y. Liu, and G. Shi, "Development of a human-robot hybrid intelligent system based on brain teleoperation and deep learning SLAM," *IEEE Transactions on Automation Science and Engineering*, vol. 16, no. 4, pp. 1664–1674, 2019.

[2] Y. P. Yan and S. F. Wong, "A navigation algorithm of the mobile robot in the indoor and dynamic environment based on the PF-SLAM algorithm," *Cluster Computing*, vol. 22, no. 6, pp. 14207–14218, 2019.

[3] H. Wu, Y. Chen, and H. Qin, "MPC based trajectory tracking for an automonous deep-sea tracked mining vehicle," *ASP Transactions on Internet of Things*, vol. 1, no. 2, pp. 1–13, 2021.

[4] F. Kun, L. Ma, and Y. Sun, "Navigation of substation inspection robot based on laser sensor[J]," *Sensors and Microsystems*, vol. 38, no. 2, pp. 118–120, 2019.

[5] Z. Li, G. Zhang, and J. Xie, "Implementation of SLAM and navigation function of indoor mobile robot based on ROS," *Shandong Industrial Technology*, vol. 10, no. 22, pp. 119-120, 2019.

[6] F. Haihua, B. Luo, and L. Guoming, "Research report on autonomous navigation system based on SLAM-Lidar technology," *Engineering Technology and Development*, vol. 1, no. 7, p. 1, 2019.

[7] F. Zhong and Z. Huang, "Trajectory planning of capacitor loading robot based on genetic algorithm [J]. Packaging engineering," vol. 40, no. 9, pp. 185–191, 2019.

[8] N. Yu, Y. Zhai, Y. Yuan, and Z. Wang, "A bionic robot navigation algorithm based on cognitive mechanism of Hippocampus," *IEEE Transactions on Automation Science and Engineering*, vol. 16, no. 4, pp. 1640–1652, 2019.

[9] Y. Liu, H. Zhang, and C. Huang, "A novel RGB-D SLAM algorithm based on cloud robotics," *Sensors*, vol. 19, no. 23, p. 5288, 2019.

[10] S. Wen, X. Hu, J. Ma, F. Sun, and B. Fang, "Autonomous robot navigation using Retinex algorithm for multiscale image adaptability in low-light environment," *Intelligent Service Robotics*, vol. 12, no. 4, pp. 359–369, 2019.

[11] M. N. A. Wahab, C. M. Lee, M. F. Akbar, and F. H. Hassan, "Path planning for mobile robot navigation in unknown indoor environments using hybrid PSOFS algorithm," *IEEE Access*, vol. 8, pp. 161805–161815, 2020.

[12] F. Demim, A. Nemra, A. Boucheloukh, E. Kobzili, M. Hamerlain, and A. Bazoula, "SLAM based on adaptive SVSF for cooperative unmanned vehicles in dynamic environment," *IFAC-PapersOnLine*, vol. 52, no. 8, pp. 73–80, 2019.

[13] S. Hadian Jazi, "Map-merging using maximal empty rectangles in a multi-robot SLAM process," *Journal of Mechanical Science and Technology*, vol. 34, no. 6, pp. 2573–2583, 2020.

[14] X. Wang, Y. Mizukami, M. Tada, and F. Matsuno, "Navigation of a mobile robot in a dynamic environment using a point cloud map," *Artificial Life and Robotics*, vol. 26, no. 1, pp. 10–20, 2021.

[15] Y. T. Liu, R. Z. Sun, and X. N. Zhang, "An autonomous positioning method for fire robots with multi-source sensors," *Wireless Networks*, vol. 2, pp. 1–13, 2021.

[16] S.-H. Park and S.-Y. Yi, "Least-square matching for mobile robot SLAM based on line-segment model," *International Journal of Control, Automation and Systems*, vol. 17, no. 11, pp. 2961–2968, 2019.

[17] X. Li, S. Du, G. Li, and H. Li, "Integrate point-cloud segmentation with 3D LiDAR scan-matching for mobile robot localization and mapping," *Sensors*, vol. 20, no. 1, p. 237, 2019.

[18] C. Wang, J. Wang, C. Li et al., "Safe and robust mobile robot navigation in uneven indoor environments," *Sensors*, vol. 19, no. 13, p. 2993, 2019.

[19] W. Deng, K. Huang, and X. Chen, "Semantic RGB-D SLAM for rescue robot navigation," *IEEE Access*, vol. 99, p. 1, 2020.

[20] Z. Hu, B. Qi, and Y. Luo, "Improved point cloud registration algorithm and mobile robot experiment in V-SLAM system [J]," *Journal of Harbin Institute of Technology*, vol. 51, no. 1, pp. 170–177, 2019.

[21] X. Li, S. Xu, S. H. Cheng, and D. Sun, "Simultaneous localization and mapping-based in vivo navigation control of microparticles," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 5, pp. 2956–2964, 2020.

[22] A. Handa, A. Suzuki, A. Suzuki et al., "Navigation based on metric route information in places where the mobile robot visits for the first time," *Journal of Robotics and Mechatronics*, vol. 31, no. 2, pp. 180–193, 2019.

[23] H. Yang, J. Qi, Y. Miao, H. Sun, and J. Li, "A new robot navigation algorithm based on a double-layer ant algorithm and trajectory optimization," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 11, pp. 8557–8566, 2019.

[24] S. F. Wong and Z. Yu, "The mobile robot anti-disturbance vSLAM navigation algorithm based on RBF neural network," *Procedia Manufacturing*, vol. 38, pp. 400–407, 2019.

[25] M. Khazaee, M. Sadedel, and A. Davarpanah, "Behavior-based navigation of an autonomous hexapod robot using a hybrid automaton," *Journal of Intelligent and Robotic Systems*, vol. 102, no. 2, pp. 1–24, 2021.