

Article

L-PCM: Localization and Point Cloud Registration-Based Method for Pose Calibration of Mobile Robots

Dandan Ning [†]  and Shucheng Huang ^{*,†}

School of Computer Science, Jiangsu University of Science and Technology, Zhenjiang 212114, China; 211110701103@stu.just.edu.cn

* Correspondence: schuang@just.edu.cn

† These authors contributed equally to this work.

Abstract: The autonomous navigation of mobile robots contains three parts: map building, global localization, and path planning. Precise pose data directly affect the accuracy of global localization. However, the cumulative error problems of sensors and various estimation strategies cause the pose to have a large gap in data accuracy. To address these problems, this paper proposes a pose calibration method based on localization and point cloud registration, which is called L-PCM. Firstly, the method obtains the odometer and IMU (inertial measurement unit) data through the sensors mounted on the mobile robot and uses the UKF (unscented Kalman filter) algorithm to filter and fuse the odometer data and IMU data to obtain the estimated pose of the mobile robot. Secondly, the AMCL (adaptive Monte Carlo localization) is improved by combining the UKF fusion model of the IMU and odometer to obtain the modified global initial pose of the mobile robot. Finally, PL-ICP (point to line-iterative closest point) point cloud registration is used to calibrate the modified global initial pose to obtain the global pose of the mobile robot. Through simulation experiments, it is verified that the UKF fusion algorithm can reduce the influence of cumulative errors and the improved AMCL algorithm can optimize the pose trajectory. The average value of the position error is about 0.0447 m, and the average value of the angle error is stabilized at about 0.0049 degrees. Meanwhile, it has been verified that the L-PCM is significantly better than the existing AMCL algorithm, with a position error of about 0.01726 m and an average angle error of about 0.00302 degrees, effectively improving the accuracy of the pose.



Citation: Ning, D.; Huang, S. L-PCM: Localization and Point Cloud Registration-Based Method for Pose Calibration of Mobile Robots.

Information **2024**, *15*, 269. <https://doi.org/10.3390/info15050269>

Academic Editor: Xiao-Fang Liu

Received: 4 April 2024

Revised: 8 May 2024

Accepted: 9 May 2024

Published: 10 May 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: AMCL; UKF; PL-ICP point cloud registration; pose calibration

1. Introduction

With the continuous development of artificial intelligence and the Internet of Things (IoT), the demand for intelligence in all types of industries is increasing. At the same time, there is a growing demand for intelligent service robots, such as sweeping robots, delivery robots, and driverless vehicles, which are involved in all aspects of human daily life with intelligent mobile robots. The most important part of a mobile robot is the navigation system, and autonomous navigation is not a simple task for a robot. It first needs to perceive the information of its surroundings and build an accurate map. Secondly, it needs to know its position in the environment; that is, to realize autonomous global localization. Finally, it needs to realize autonomous navigation through global path planning and local path planning. In our daily life, the demand for precise navigation is gradually expanding. Accurate pose is extremely important for realizing autonomous navigation, so effective pose calibration is very necessary.

When the mobile robot is in an unknown environment, the navigation system will carry out real-time pose monitoring to obtain the latest estimated pose data at each moment. It matches and controls the pose data resolved by the navigation system through the optimal local path. Therefore, obtaining the real-time estimated pose data of the mobile robot

directly affects whether it accurately follows the optimal path planned by the navigation system, realizes real-time obstacle avoidance, and accurately arrives at the target point. Therefore, the study of how to obtain more accurate current pose data of mobile robots can provide more accurate pose information for autonomous navigation, generate more accurate environment maps, and reduce the uncertainty in the autonomous navigation of mobile robots.

For autonomous navigation, it is difficult to obtain accurate pose data. Researchers proposed the simultaneous localization and mapping (SLAM) technique [1,2], the core of which is to obtain the pose data of the current state of the mobile robot to achieve self-localization. The most stable, efficient, and widely used algorithm to achieve localization performance in a 2D environment is the adaptive Monte Carlo localization (AMCL) algorithm [3–5]. This algorithm obtains global localization of the mobile robot through a probabilistic model of the particle filter, and it is both real-time and computationally efficient. Although the AMCL algorithm has now become the only specified localization algorithm in the ROS (robot operating system) for mobile robots, the localization accuracy of this algorithm cannot meet the demand for the pose estimation accuracy of mobile robots. The motion model of this algorithm has errors, which directly affect the pose when initializing the particle swarm. Moreover, this algorithm cannot solve the problem of noise and cumulative errors of individual pose sensors, further affecting the pose accuracy. A typical pose calibration method is the Kalman filter algorithm, which has better robustness and adaptability in dynamic and complex environments. However, the error between the actual observed value and the estimated value of the system is too large when the traditional Kalman filter algorithm is used for data fusion. And the cumulative error of the calculation gradually becomes larger with the increase of the observation volume in the calculation process. In this case, Bucy and Y.Sunahara et al. proposed utilizing the extended Kalman filter (EKF) algorithm to fuse the IMU data and odometer data for pose calibration [6,7]. The algorithm efficiently estimates the system state by iterative means and realizes the recursive estimation of the system state, which can reduce the impact of low sensor accuracy and accumulated errors on the pose estimation. However, the algorithm has a low accuracy of the pose estimation when the uncertainty of the prior estimation of the system is high, and the numerical stability is not high. The pose calibration method can also utilize the point cloud registration algorithm of LiDAR, and the commonly used algorithm is the iterative closest point (ICP) algorithm [8], which can accurately find the conversion relationship between two radar scans. However, when the sample size of the point set is too large, the ICP algorithm will waste computational resources. With the continuous improvement and development of the theory and technology, the study of the pose calibration of mobile robots has better scalability and research value.

In order to solve the problems in the existing technology, this paper proposes a pose calibration method based on localization and point cloud registration, called L-PCM. The AMCL algorithm utilizes the wheel odometer model of the mobile robot for pose estimation, while the odometer has cumulative errors during robot motion, resulting in limited pose estimation accuracy. In order to improve this problem, we first use the UKF model to fuse the odometer and IMU data to estimate the pose, and then the estimated pose is used as the predicted pose to participate in the particle trajectory prediction part to improve the AMCL in order to obtain the global initial pose. An indoor simulation environment is built in GazeBo by comparing the traditional EKF and UKF algorithms as well as comparing the original AMCL and the improved AMCL to validate the improved algorithm's improvement effect and pose accuracy. Finally, PL-ICP point cloud registration is utilized to complete the pose correction of the L-PCM algorithm and verify the localization accuracy of the L-PCM algorithm. The contributions of this study are as follows:

- (1) The traditional algorithm for solving the problem of cumulative sensor errors is the EKF algorithm, while in this paper, the UKF fusion model is used to fuse the data from the odometer and the IMU. It achieves linearization through a weighted statistical linear regression process, replaces the one-step prediction of calculating the mean and covariance

in the Kalman filtering algorithm with the UT transform, which directly avoids the process of solving the Jacobi matrix, improves the filtering effect, and is able to effectively suppress the noise and improve the localization accuracy.

(2) The estimated pose obtained by fusing IMU and odometer data through UKF is used as the predicted pose of AMCL, which is involved in the particle trajectory prediction part, and the modified global initial pose of the mobile robot is obtained by improved AMCL. This gives a significant advantage to the linear localization of the mobile robot and greatly improves the stability of linear and angle localization.

(3) Usually point cloud registration algorithms are introduced to improve AMCL, whereas in this paper, the PL-ICP point cloud registration algorithm is utilized to calibrate the improved AMCL pose. In this paper, according to the pose difference obtained from the global initial pose at the previous and current moments, we use PL-ICP point cloud registration to iteratively update the minimization distance between the point clouds, so as to calibrate the global initial pose obtained from the improved AMCL. It combines the UKF pose estimation, the improved AMCL corrected pose, and the PL-ICP calibrated pose, which makes the angle localization of the pose estimation more accurate, and the yaw angle localization is always maintained at high precision. Meanwhile, this algorithm is less affected by noise and has greater stability.

The remainder of the paper is organized as follows. In Section 2, work related to mobile robots, localization, pose estimation, and point cloud registration is presented. Section 3 describes the pose estimation module based on UKF fusion with an odometer and IMU, the global initial pose module based on improved AMCL, and L-PCM-based pose calibration. The experiments with the UKF fusion module, improved AMCL module, and L-PCM pose calibration are described, and the results are analyzed in Section 4. In Section 5, the experimental results are discussed. The conclusions of the paper are summarized in Section 6.

2. Related Work

In this section, we introduce the current state of research on mobile robots, the existing problems and the current state of research on localization, the common methods and the current state of research on pose estimation, and the current state of research on point cloud registration.

2.1. Mobile Robots

The field of mobile robots developed earlier abroad, forming two camps of wheeled robot research led by the United States and humanoid two-legged robot research led by Japan. In 1972, the world's first robot, Shakey, was born at the Stanford Research Institute under the charge of Charlie Rosen, and Shakey marked the first time that the idea of artificial intelligence had been comprehensively applied. Although computers ran slowly at the time, and often took hours of computation for a robot to make the right decision, Shakey's practical results have far-reaching significance for the research of future generations [9]. In 1973, Japan's Ichiro Kato developed the world's first humanoid robot, WABOT. The robot could only perform simple walking but was capable of using vision to observe objects and using both hands to manipulate objects. Additionally, it could communicate with people through the Japanese language [10]. While the development of China's robotics field is relatively late, in 1986, China's 863 program was launched, and since then, China's robotics industry has entered a high-speed development stage [11]. National policy has provided a high degree of support for the mobile robot industry, significantly increasing researchers' enthusiasm. As a result, the field of mobile robots has achieved a series of fruitful results. In 1990, the National University of Defense Science and Technology developed the country's first bipedal walking robot. In 2000, the same university developed China's first humanoid robot, Pioneer, as documented in [12]. Pioneer can walk at a speed of up to two steps per second and can speak. It realized a number of technological breakthroughs such as

autonomous walking in unknown environments, marking China's entry into the world's leading level of mobile robot technology.

2.2. Localization

A mobile robot needs to obtain its pose in the global coordinate system in real time; that is, based on the construction of the navigation map, according to the environmental data obtained by the sensor and the pose estimation data obtained by the localization sensor to match with the current map environment, in order to confirm the real-time pose of the mobile robot in the global map. Generally speaking, the robot's localization is divided into three main problems according to the initial state [4]: (1) Local localization: It is the localization of the robot by appropriately increasing the motion noise in the case of knowing its own pose in the previous frame. (2) Global localization: The initial state of the robot does not know its own pose, and the robot needs to be localized in the world map. (3) Robot abduction problem: It is a kind of divergence problem of global localization problem, its initial position is known, but it is an incorrect pose. Therefore, the robot needs to determine by itself if the obtained pose is incorrect and then re-localize, which tests the localization failure recovery capabilities of the system. The commonly used robot localization methods include Markov localization and EKF localization [13,14], and AMCL [15–17].

Fox et al. first applied particle filtering to robot localization, and they created Monte Carlo localization. After that, Fox was inspired by Thrun and added KLD sampling to the localization algorithm, which is the adaptive version of particle filtering [18]. In 2016, DoHN et al. [19] proposed solving the localization problem using the GMRF model. Its simulation and experimental fruits well demonstrate the adaptation and flexibility of the full Bayesian approach for data processing. In 2018, Gimenez J et al. [20] applied Markov random fields to SLAM modeling, which were later solved by iterative computation. Their method can achieve collaborative multi-robot localization and localization in dynamic environments, and they demonstrated the superiority of their method with experimental data. G Ge [21] proposed a Text-MCL localization method based on the AMCL algorithm according to the environment semantic information and radar scanning information, which improves the speed and success rate of global localization of mobile robots. AA Fikri [22] integrated the cartographer mapping method proposed by Google with the AMCL algorithm and fused the data from IMU and the odometer, which reduces the mobile robots' systematic errors in the process of map building and localization.

The above studies have some advantages, but there are also some disadvantages. For example, some research methods only study the data error of the AMCL algorithm in the linear direction, ignoring the fact that the yaw angle error is also relatively large, resulting in the poor stability of the AMCL algorithm. Other research methods ignore the effect brought by the cumulative error of sensors. When the single sensor has problems, the normal operation of the robot cannot be guaranteed. Aiming at these problems, this paper utilizes the advantages of sensors complementing each other, and combines the odometer, IMU, and LiDAR to calibrate the initial pose results of AMCL, to improve the localization accuracy and stability of the AMCL algorithm.

2.3. Pose Estimation

Mobile robots usually move in unpredictable and complex environments, and obtaining their pose state at any given moment is difficult. However, when robots are working in well-structured environments, the level of uncertainty tends to be low. In this case, the problem of robot perception of the environment can be described by probability, which centers on estimating the robot state through a filtering algorithm in multiple sensor data, i.e., pose estimation [23]. In the pose estimation system of mobile robots, single pose sensor data have noise, and if the abnormal sensor data are collected by the pose estimation system, it will affect the precision of the pose estimation, and even lead to the failure of the pose estimation when the errors accumulate too much. To address the problem of the noise of a single attitude sensor, the fusion of multiple attitude sensor data, such as odometer

data and inertial measurement unit (IMU) data, is the most commonly used method to obtain the pose estimate [24].

The core algorithms for fusing data from multiple pose sensors all use Bayesian filtering, which is generally categorized into two classes. One is based on the Kalman filtering algorithm [25], which is widely used to quickly obtain the estimated pose and provide the initial value for the localization system due to its simplicity and stability. But the pose accuracy is lacking due to the influence of noise and model. Another class of algorithms is based on the particle filtering class [26–28], which is widely applicable and does not depend on noise or model, and is also widely used in mobile robot pose estimation and localization.

The typical Kalman filtering algorithm was proposed by Rudolf Emil Kalman [29], which can realize the optimal estimation of linear systems. However, the traditional Kalman filtering algorithm is only applicable to the linear Gaussian system, which has a large error between the state equation of the system and the observation information in the nonlinear state, resulting in an excessive error between the actual measured value and the estimated value. In addition to this, the cumulative error of the calculation increases gradually with the growth of time, which leads to a gradual increase in the error of the result of the pose estimation. To address this situation, Bucy and Y. Sunahara et al. [6,7] proposed the nonlinear Kalman filter method, the extended Kalman filter (EKF). The EKF has excellent stability when used for pose estimation, but the accuracy is low when the local nonlinearity is strong and the prior estimation uncertainty of the system is high. Martinez et al. [30] achieved pose estimation and localization by fusing the data from ranging and inertial navigation methods through the extended Kalman fusion (EKF) algorithm, which improved the accuracy of the pose estimation to a certain extent, but the accuracy of the pose estimation still did not meet the requirements. Rashidi H et al. [31] accomplished map construction for visual SLAM by fusing depth camera and odometer data based on a two-wheel differential robot model. However, the depth camera is susceptible to external ambient light, resulting in poor stability of this fusion strategy. In order to improve the stability of the position estimation system, Le-Anh T et al. [32], in order to alleviate the odometer error caused by the chassis wheel slippage of a two-wheeled differential robot, fused the IMU data and the odometer data through a Bayesian estimation algorithm. However, since no other sensors sensing the external environment were used, the fused position could not be corrected, resulting in increasing the cumulative error of the position estimation system at a later stage. S. Julier combined UT and KF (Kalman Filter) and proposed the unscented Kalman filter (UKF) [16,33], which filters through the framework of the UT transform and Kalman filter algorithm to accomplish the optimal estimation of the target state with high accuracy of pose estimation.

Compared with the UKF, the traditional EKF is linearized by Taylor's first-order expansion approximation for the nonlinear state transfer, which achieves a better linear transformation of the mean value. However, the nonlinearity of the variance is still not solved. Moreover, the EKF needs to specialize in calculating the nonlinear state transfer and Jacobi matrix, which not only multiplies the computational volume but also may lead to the dispersion of the results, poor numerical stability, and other problems. Based on the fusion of odometer and IMU data by UKF, this paper realizes linearization through a weighted statistical linear regression process and replaces the one-step prediction method of calculating the mean and covariance in the Kalman filtering algorithm with UT transformation, which directly avoids the process of solving the Jacobi matrix and improves the filtering effect.

2.4. Point Cloud Registration

Mobile robots for point cloud registration mainly rely on LiDAR, which aligns the point cloud data to obtain a high-precision laser odometer [34–36]. Currently, the most common method for point cloud registration is the iterative closest point (ICP) algorithm [8], which continuously optimizes the registration error between two point clouds through an

iterative process until the error is minimized. However, when the number of point clouds is too large, the ICP algorithm will waste a lot of computational resources. In order to solve this problem, the normal ICP (NICP) algorithm [37] adds constraints on the normal vector and the curvature of the surface where the point cloud is located, in addition to the constraints on the distance between the points in the point cloud registration. In recent years, researchers have proposed some improved algorithms for the ICP algorithm to improve the accuracy of the ICP alignment. The PL-ICP (point-to-line ICP) algorithm in this paper is an improvement of the ICP algorithm. The algorithm mainly improves the form of the error, and other aspects are basically the same, but the iteration rate and the accuracy are greatly improved, which is more suitable for nonlinear scenarios in mobile robot localization. In this paper, we utilize PL-ICP point cloud registration to calibrate the global initial pose obtained from the improved AMCL. The initial pose information of AMCL provides the initial pose matrix for the PL-ICP algorithm, and then the laser point cloud information from PL-ICP is used as feedback information to calibrate the AMCL's pose.

3. Materials and Methods

This section contains the following main parts: the pose estimation module based on the UKF fusion of the odometer and IMU, the global initial pose module based on the improved AMCL, and the L-PCM-based pose calibration. The architecture is shown in Figure 1. To address the problem of cumulative sensor errors, the UKF model fuses the data from the IMU and the odometer to estimate the pose. The estimated pose is then used as the predicted pose to participate in the particle trajectory prediction part to improve the AMCL in order to obtain the global initial pose. Finally, PL-ICP point cloud registration is utilized to complete the pose calibration of L-PCM.

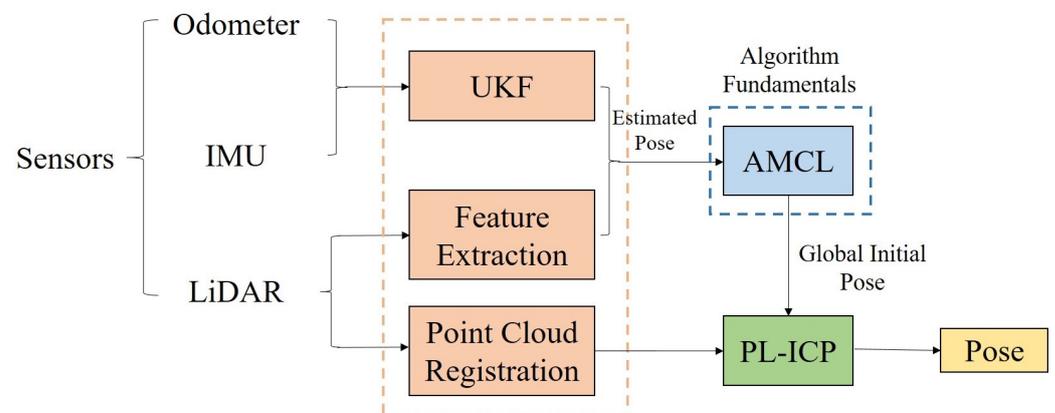


Figure 1. The architecture of the L-PCM pose calibration.

3.1. The Pose Estimation Module Based on UKF Fusion Odometer and IMU

In order to address the effect of cumulative sensor errors, such as odometers on the accuracy of the estimated pose, Kalman filtering is used to realize the pose estimation. Kalman filtering is an optimal estimation of a stochastic process based on a linear system, in which both the system noise and the measurement noise obey Gaussian distributions. However, in mobile robot localization, the transitions between state quantities and observations are nonlinear, and Kalman filtering is unable to deal with the Gaussian noise of nonlinear systems. Therefore, the UKF is used for this nonlinear system, and the nonlinear state and observation equations are linearized to obtain their approximate sub-merit solutions, and then the state estimation is completed by Kalman filtering algorithm.

Based on the state information of the mobile robot, UKF is used to filter and fuse the odometer data and IMU data. The motion model of the mobile robot is used as the prediction value of the UKF fusion algorithm, and the odometer and IMU data are used as

the observation values of the UKF fusion algorithm for calibration, realizing the fusion of the IMU and the odometer to obtain the estimated pose of the mobile robot.

The UKF algorithm improves the results when filtering nonlinear problems, and is centered on the UT transform, which is used to compute the statistical characteristics of random variables in a nonlinear transformation. Suppose we have the following nonlinear system:

$$x_{t+1} = f(x_t + u_t) + w_t \tag{1}$$

$$y_t = h(x_t) + v_t \tag{2}$$

The process of nonlinear state filtering consists of two steps: one-step prediction and measurement correction. One-step prediction is a minimum variance estimation of the state at moment t according to the measurement information at moment $t - 1$:

$$\bar{x} = E\{x_t | Y^{t-1}\} \tag{3}$$

The error covariance matrix is used to describe whether a one-step prediction is a good or bad estimate of the state at moment t :

$$\bar{P}_t = E\{(x_t - \bar{x}_t)(x_t - \bar{x}_t)^T | Y^{t-1}\} \tag{4}$$

Measurement correction is used to modify the one-step prediction estimate of the state at moment t after obtaining the measurement information at moment t , thus obtaining the optimal estimate at moment t .

$$\hat{x}_t = E\{x_t | Y^t\} = \bar{x}_t + K_t(y_t - \bar{y}_t) \tag{5}$$

$$K_t = P_{xy}(t)P_y^{-1}(t) \tag{6}$$

$$\bar{y}_t = E\{y_t | Y^{t-1}\} \tag{7}$$

$$P_y(t) = E\{(y_t - h(\bar{x}_t))(y_t - h(\bar{x}_t))^T | Y^{t-1}\} \tag{8}$$

$$P_{xy}(t) = E\{(x_t - \bar{x}_t)(y_t - h(\bar{x}_t))^T | Y^{t-1}\} \tag{9}$$

The superiority or inferiority of the quality of the state estimates of the measurement correction process is described utilizing the covariance matrix, as follows:

$$\hat{P}_t = E\{(x_t - \hat{x}_t)(x_t - \hat{x}_t)^T | Y^{t-1}\} = \bar{P}_t - K_t P_y(t) K_t^T \tag{10}$$

The UKF algorithm replaces the Kalman filtering algorithm with a one-step prediction to compute the mean and covariance with the UT transform. The UT transform first constructs the Sigma points. Assuming that the state of the mobile robot at moment $t - 1$ is $X_{t-1} = (x_{t-1}, y_{t-1}, \theta_{t-1}, v_{x,t-1}, v_{y,t-1})^T$, its prior distribution is Gaussian with mean (μ) and covariance (Σ). n -dimensional state vectors are dimensioned and $2n + 1$ Sigma points are extracted, as follows:

$$x_i = \begin{cases} \bar{x} = \mu & \\ \bar{x} + (\sqrt{(n + \kappa)\Sigma})_i & i = 1, 2, \dots, n \\ \bar{x} - (\sqrt{(n + \kappa)\Sigma})_{i-n} & i = n + 1, n + 2, \dots, 2n \end{cases} \tag{11}$$

where $\kappa = \alpha^2(n + \lambda) - n$, α , and λ are parameters that regulate how far the Sigma points are distributed from the mean, \bar{x} is the optimal estimate of the state vector at moment $t - 1$, that is, at the Gaussian mean, and Σ is the variance at moment $t - 1$. The Gaussian distribution of state x can be approximated using the sampled points x_i .

The transformed Sigma point set can be obtained by applying a $f(\cdot)$ nonlinear transformation to the point set $\{x_i\}$:

$$Y_i = f(x_i) \quad i = 0, 1, \dots, 2n \tag{12}$$

The transformed Sigma point set $\{y_i\}$ approximates the distribution of $y = f(x)$. The transformed Sigma point set $\{y_i\}$ is weighted to obtain the mean and variance of the output quantity y :

$$\bar{y} \approx \sum_{i=0}^{2n} W_i^{(m)} Y_i \tag{13}$$

$$P_y \approx \sum_{i=0}^{2n} W_i^{(c)} (Y_i - \bar{y})(Y_i - \bar{y})^T \tag{14}$$

where $W_i^{(m)}$ and $W_i^{(c)}$ are the weights used to calculate the mean and variance of y , respectively:

$$W_0^{(m)} = \kappa / (n + k) \tag{15}$$

$$W_0^{(c)} = \kappa / (n + k) + (1 - \alpha^2 + \beta) \tag{16}$$

$$W_i^{(m)} = W_i^{(c)} = \kappa / [2(n + k)] \quad i = 1, 2, \dots, 2n \tag{17}$$

The motion model and observation model of the mobile robot are assumed to be as follows, respectively:

$$x_t = f(x_{t-1}, u_t, w_t) \tag{18}$$

$$y_t = h(x_t) + v_t \tag{19}$$

where w_t is the noise during motion and v_t is the noise during measurement. First, the state vector x_t is reconstructed by generalizing it to a joint vector with the motion model noise, as follows:

$$x_t^a = \begin{bmatrix} x_t \\ w_t \end{bmatrix} \tag{20}$$

Then the motion of the mobile robot becomes

$$x_t^a = f(x_{t-1}^a, u_t) \tag{21}$$

Assuming that the mean of x_t is \hat{x}_t and the variance is $P_{t|t}$, the mean and variance of x_t^a are, respectively:

$$\hat{x}_t^a = \begin{bmatrix} \hat{x}_t \\ 0 \end{bmatrix} \tag{22}$$

$$P_{t|t}^a = \begin{bmatrix} P_{t|t} & 0 \\ 0 & Q_t \end{bmatrix} \tag{23}$$

The prediction process of the UKF algorithm is as follows: the Sigma point x_{t-1} is transformed by the motion model $y = f(x)$, and the mean and variance of the prediction from x_{t-1} are obtained as follows:

$$\begin{cases} x_{i,t|t-1} = f(x_{t-1}, u_t, w_t) \\ \hat{x}_{t|t-1} = \sum_{i=0}^{2n} W_i^{(m)} x_{i,t|t-1} \\ P_{t|t-1} = \sum_{i=0}^{2n} W_i^{(c)} (x_{i,t|t-1} - \hat{x}_{t|t-1})(x_{i,t|t-1} - \hat{x}_{t|t-1})^T \end{cases} \tag{24}$$

The process of updating the predicted state of the UKF is as follows: the new Sigma point $x_{i,t|t-1}^*$ is transformed by the observation model $y = h(x)$, which yields the corresponding mean and variance as follows:

$$\begin{cases} y_{i,t|t-1} = h(x_{i,t|t-1}^*) + v_t \\ \hat{\gamma}_{t|t-1} = \sum_{i=0}^{2n} W_i^{(m)} y_{i,t|t-1} \\ S_t = \sum_{i=0}^{2n} W_i^{(c)} (y_{i,t|t-1} - \hat{\gamma}_{t|t-1})(y_{i,t|t-1} - \hat{\gamma}_{t|t-1})^T \end{cases} \quad (25)$$

Then the corresponding mean and variance at moment t are as follows:

$$\begin{cases} \hat{x}_t = \hat{x}_{t|t-1} + K(\hat{\gamma}_{t|t-1} - \gamma_t) \\ P_{t|t} = P_{t|t-1} - KS_tK^T \\ P_{xy,t|t-1} = \sum_{i=0}^{2n} W_i^{(c)} (x_{i,t|t-1} - \hat{x}_{t|t-1})(y_{i,t|t-1} - \hat{\gamma}_{t|t-1})^T \end{cases} \quad (26)$$

where $K = P_{xy,t|t-1}S_t^{-1}$ denotes the Kalman gain and $P_{xy,t|t-1}$ is the intersecting covariance matrix.

3.2. Global Initial Pose Module Based on Improved AMCL

AMCL uses particle filters to track known robot poses for global localization. Compared to MCL (Monte Carlo localization), when a sudden decrease in the average particle score is detected (correct particles are lost in a certain iteration), particles are re-scattered on the map to solve the robot abduction problem. When the localization is accurate and the particles converge to the neighborhood of the true value, the number of particles is reduced to solve the fixed number of particles problem. Among them, the Augmented_MCL algorithm is utilized to solve the robot abduction problem, and the KLD_Sampling_MCL algorithm is used to solve the particle number fixing problem.

The improved AMCL model mainly includes the parts of particle initialization, particle trajectory prediction (including UKF fusion IMU and odometer module to update pose), particle weight calculation, particle resampling, and correcting the position. The process of pose calibration through the improved AMCL model is specified as follows:

(1) Make a judgment on whether the particle set is initialized or not. If it has been initialized, the particle pose ($X_{t-1} = (x_{t-1}, y_{t-1}, \theta_{t-1})^T$), control quantity (μ_{t-1}) and map information (m) at the moment $t - 1$ are directly obtained, and then enter into step (2). If there is no initialization, a large number of particle points are scattered randomly until the number of particles reaches the set value and then enters into step (3).

(2) After the particle swarm is initialized, the UKF fusion algorithm is used as the prediction model of AMCL, which directly replaces the original odometer prediction model, and the fused estimated pose is used as the predicted pose of the motion model of AMCL. Substituting the particle swarm at moment $t - 1$ into the motion model, the particle motion trajectory prediction at moment t is obtained. That is to say, the corrected particle pose at time t is $X_t = (x_t, y_t, \theta_t)^T$:

$$\begin{cases} x_t = x_{t-1} - \Delta S \sin \theta + \Delta S \sin(\theta + \Delta\theta) \\ y_t = y_{t-1} + \Delta S \cos \theta - \Delta S \cos(\theta + \Delta\theta) \\ \theta_t = \theta_{t-1} + \Delta\theta + \gamma\Delta t \end{cases} \quad (27)$$

where sampling from the difference between the estimated pose at moments $t - 1$ and t gives the amount of change in pose ($\Delta\theta$ and ΔS) at moment t . γ is an additional random disturbance. The particle swarm is updated based on the updated pose.

(3) Based on the known particle pose, the Gaussian sampling probability is calculated from the distance between the grid where the laser point is located and the obstacle according to the likelihood domain observation model of the laser point. Summing the likelihood domains of all laser beams emitted in all directions with this 2D grid coordinate as the center, the weight (η^k) of the k th particle point can be obtained as follows:

$$\eta^k = p(z_t^k | X_t^k, m), \quad (28)$$

where $p(z_t^k | X_t^k, m)$ is the expected probability of the laser observation model and X_t^k denotes the particle population at moment t after the update.

(4) Judge whether resampling is required. Long-term likelihood average estimation, short-term likelihood average estimation, long-term exponential filter decay rate, and short-term exponential filter decay rate are introduced for failure recovery. Failure is judged by comparing the long- and short-term weights of the particles. When the particle quality decreases, the average weight decreases, the particle short-term weight decreases faster than the long-term weight, and random probability is greater than 0, then resampling is required. However, when the particle quality improves and the particle's short-term weight decreases faster than the long-term weight, resampling is not required.

If resampling is needed, the invalid particles with particle weights less than the threshold are discarded, and the particles with particle weights greater than the threshold are copied; then the optimal number of particles is calculated and sorted by adding random particles to the particle swarm after adaptive sampling. If resampling is not needed, the particles are sorted according to their weights, and the average pose of the particle swarm with the highest weight is taken as the real pose of the mobile robot at the current moment.

3.3. L-PCM-Based Pose Calibration

The L-PCM pose calibration method is based on the improved AMCL localization algorithm with the addition of point cloud registration, and the point cloud registration uses the PL-ICP point cloud registration algorithm. PL-ICP is point cloud registration that considers two sets, a set of points in the current laser scan data and a set of points in the reference frame selected based on straight line segments. The points selected here are not the straight-line features extracted from the laser data, but the points on the connecting line between two consecutive points [38,39].

The main process of pose calibration for PL-ICP point cloud registration is as follows:

(1) Two laser-scanning point cloud collections at moments $t - 1$ and t , respectively:

$$\begin{aligned} X &= x_1, x_2, \dots, x_{N_x} \\ P &= p_1, p_2, \dots, p_{N_p} \end{aligned} \tag{29}$$

where x_i and p_i denote the point cloud coordinates, and N_x and N_p denote the size of the point cloud collection.

(2) The mobile robot poses corrected by the improved AMCL at moment $t - 1$ and at moment t are as follows:

$$\begin{aligned} X'_{t-1} &= (x_{t-1}, y_{t-1}, \theta_{t-1})^T \\ x'_t &= (x_t, y_t, \theta_t)^T \end{aligned} \tag{30}$$

Based on this pose difference, the initial values (R_0, T_0) of the rotation matrix (R) and translation matrix (T) iterations of the PL-ICP algorithm can be obtained. Based on the mobile robot pose ($X_{t-1} = (x_{t-1}, y_{t-1}, \theta_{t-1})^T$) at the moment $t - 1$, it is possible to directly find the corresponding point near the point cloud at the previous moment.

(3) Transform the point cloud P at moment t to the point cloud X coordinate at moment $t - 1$, and the distance is expressed as follows:

$$e_i = n_i^T [R(\theta_i)p_i + T_i - x_i] \tag{31}$$

The two points in the point cloud X that are closest in Euclidean distance to any point (p_i) in the point cloud P are x_m and x_n . n_i^T denotes the normal vector perpendicular to the line joining the points x_m and x_n . Minimizing the distance equation (e_i) yields the rotation matrix (R) and translation matrix (T); applying the rotation matrix (R) and translation matrix (T) to the point cloud P at time t , the new point cloud (p'_i) can be obtained by rotating and translating it.

(4) Update the distance between point cloud P and point cloud X :

$$d = \frac{1}{n} \sum_{i=1}^n \|p'_i - x_i\|^2 \quad (32)$$

The iteration continues when the distance (d) is greater than the threshold, and the iteration finishes when the distance (d) is less than the threshold to obtain the final (R, T). The pose transformation in the translation matrix (T) is represented as follows:

$$T = [T_x \quad T_y]^T \quad (33)$$

The rotation matrix (R) is denoted as follows:

$$R = \begin{bmatrix} \cos \theta_R & -\sin \theta_R \\ \sin \theta_R & \cos \theta_R \end{bmatrix} \quad (34)$$

The resulting angle of rotation (θ_R) can be obtained as follows:

$$\theta_R = \arctan(\cos \theta_R, \sin \theta_R) \quad (35)$$

Then the final pose ($X_{p,t}$) of the mobile robot at the moment t is obtained by L-PCM as follows:

$$X_{p,t} = \begin{bmatrix} x_{p,t} \\ y_{p,t} \\ \theta_{p,t} \end{bmatrix} = \begin{bmatrix} x_{p,t-1} \\ y_{p,t-1} \\ \theta_{p,t-1} \end{bmatrix} + \begin{bmatrix} \cos \theta_{p,t-1} & -\sin \theta_{p,t-1} & 0 \\ \sin \theta_{p,t-1} & \cos \theta_{p,t-1} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} T_x \\ T_y \\ \theta_R \end{bmatrix} \quad (36)$$

4. Results

We chose a computer using Ubuntu 18.04 (melodic version) and Robot Operating System (ROS). The simulation environment was built through GazeBo as shown in Figure 2.

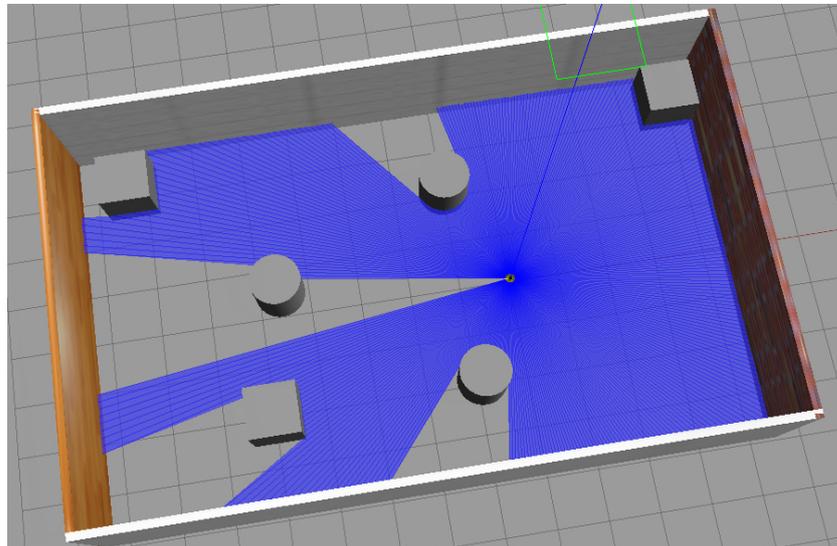


Figure 2. Simulation environment. This environment simulates an indoor environment; that is, surrounded by a closed environment. Cylinders and squares in a room are obstacles.

4.1. Experiments on the UKF Fusion Module

By directly setting the target point of the mobile robot in the GazeBo environment, the upper layer navigation algorithm plans the optimal path, and then directly subscribes to the pose topics of the mobile robot published by GazeBo. After subscribing to the topics, we obtain the timestamp-matched RosBag packets, obtain real-time pose information, and obtain the real pose trajectory.

We load both the traditional position estimation fusion algorithm (EKF) implemented on the underlying source code and the UKF position estimation fusion algorithm proposed in this paper at the time of running the mobile robot. We subscribe to the topics based on the EKF fusion odometer and IMU and the topics based on the UKF fusion odometer and IMU released by ROS in real time, respectively, and obtain the timestamp-matched RosBag packets after subscribing to the topics. The fused pose trajectories are plotted in Matlab as shown in Figure 3.

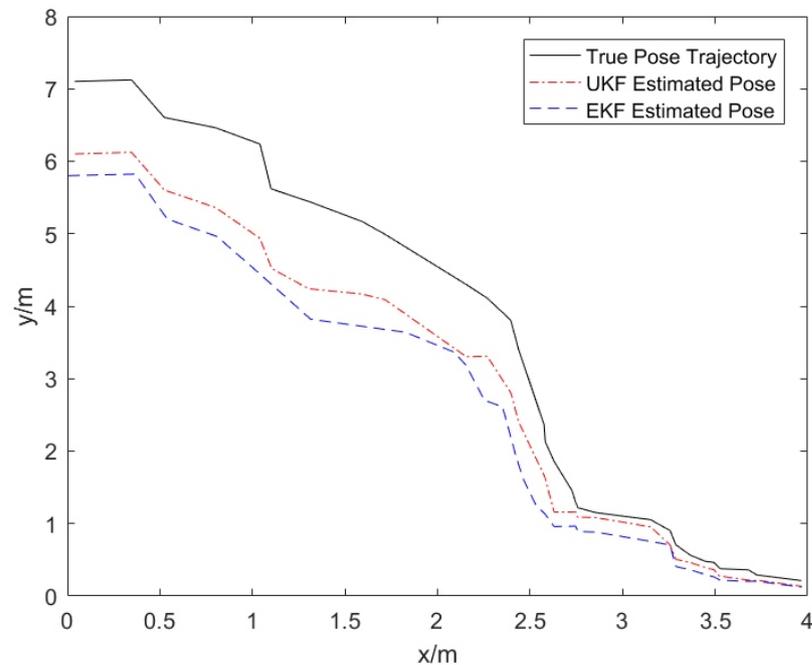


Figure 3. True and fused pose trajectories. Black is the true post trajectory, blue is the EKF estimated pose trajectory, and red is the UKF estimated pose trajectory.

From Figure 3, it can be seen that the error between the fused estimated pose and the true pose gradually increases as the mobile robot moves toward the target position, indicating that the cumulative error of the odometer and other sensors gradually becomes more influential on the accuracy of the estimated pose. Meanwhile, it can also be seen from Figure 3 that the estimated pose trajectory after UKF fusion is significantly better than the estimated pose trajectory after EKF fusion. Selecting 30 datasets from the above topics, the error map between the estimated pose and the real pose is obtained, as shown in Figure 4.

From Figure 4, it can be seen that the estimated pose error after UKF fusion is significantly smaller than the estimated pose error after EKF fusion. Therefore, the estimated pose obtained by the UKF fusion algorithm is more accurate than the estimated pose obtained by the EKF fusion algorithm. However, when the mobile robot is near the coordinate origin, the error between the UKF and the actual pose trajectory is large. This causes the mobile robot to misjudge the position and distance of obstacles, making it unable to avoid them. And the path planned at this time may be redundant or even invalid, which increases energy consumption and reduces the efficiency of task execution. Therefore, the UKF fusion algorithm still fails to meet the requirements for the localization accuracy of mobile robots, and the estimated pose accuracy is still poor.

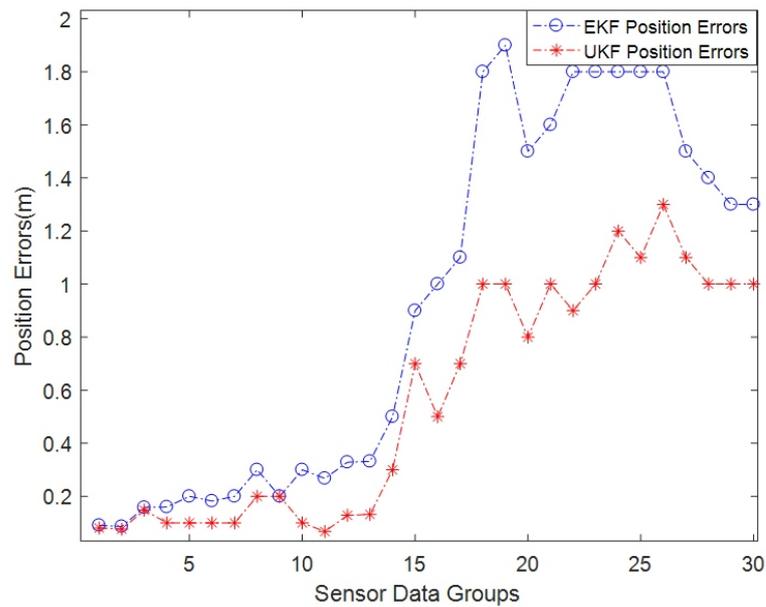


Figure 4. Error between the true and estimated postures. The blue color represents the error of the EKF estimated pose versus the true pose, and the red color represents the error of the UKF estimated pose versus the true pose.

4.2. Experiments on Improved AMCL Module

The AMCL localization function is implemented by configuring the navigation function package in ROS. In the configuration file `amcl.xml`, we set the initial position of the wheelchair as $(0, 0, 0)$, set the minimum number of particles as 500, and the maximum number of particles as 5000, and choose `diff` for the odometer motion model, which is the differential motion model. Two sets of experiments are designed to test the improved AMCL localization algorithm for the pose tracking and global localization problems, respectively.

The improved AMCL localization algorithm's pose tracking problem is shown in Figure 5.

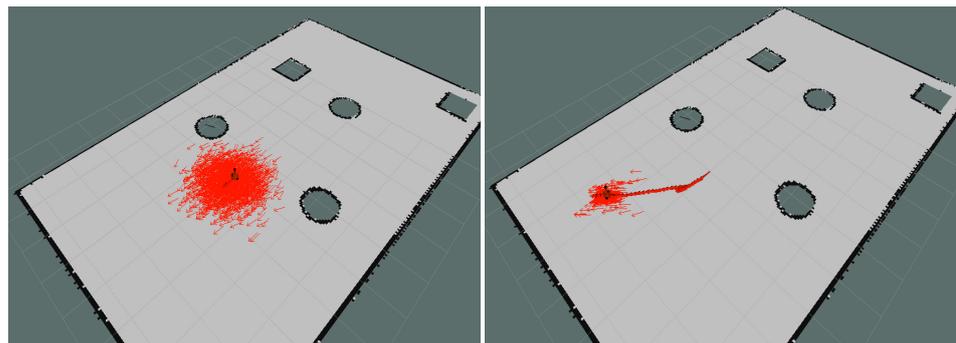


Figure 5. Pose tracking. The left figure shows the particle distribution state during initialization, and the right figure shows the particle distribution state after the mobile robot moves a certain distance. Black represents the obstacles. Red shows the improved AMCL algorithm particle swarm.

As shown in Figure 5, the red area particles are relatively scattered upon initialization, and the particle convergence is measured using the `rviz` tool to be approximately 1.485 m at the real position. When the mobile robot moves a distance, the particles begin to converge to the true position, while the number of particles decreases. Again, through the measurement tool to obtain the particle convergence in the true position of about 0.364 m, the convergence effect is significantly improved. Therefore, the improved AMCL localization algorithm has a significant convergence effect for pose tracking.

For the global localization problem of the improved AMCL localization algorithm, the initial pose in the `amcl.xml` configuration file is changed to $(1, 2, 0)$, so the real pose is shifted from the initial pose to a certain extent. When the mobile robot moves a certain distance, the particle convergence is shown in Figure 6.

As shown in Figure 6, the initial particle converges to the true position around 1.7448 m. The initial mobile robot does not conform to its true position, and with the slow movement of the mobile robot and the incoming environmental information from the sensors, the particles slowly converge. Eventually, it converges to about 0.3575 m and returns to the normal positional location, and the convergence effect is significantly improved. Therefore, the improved AMCL localization algorithm has a significant convergence effect on global localization.

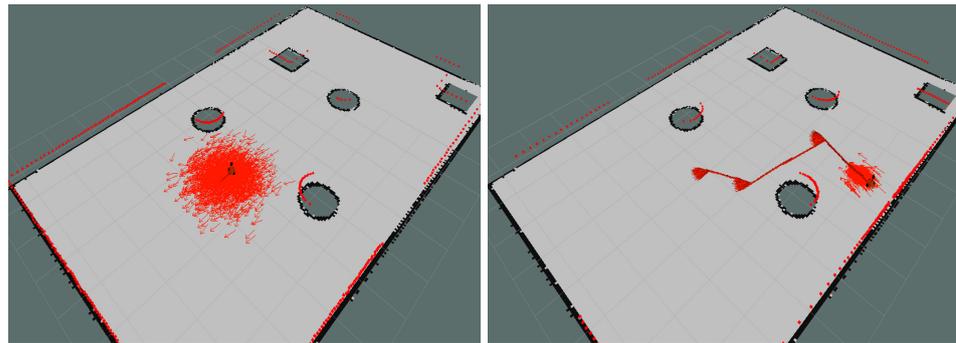


Figure 6. Global localization. The left figure shows the particle distribution state during initialization, and the right figure shows the particle distribution state after the mobile robot moves a certain distance. Red shows the improved AMCL algorithm particle swarm. Black represents the obstacles.

In order to compare the accuracy difference between the improved AMCL algorithm and the original algorithm, the original AMCL function package and the improved AMCL localization package are loaded. After subscribing to the pose topics released by both in real time, respectively, and parsing out the RosBag topic packets generated by both with matched timestamps, the pose trajectory is plotted as shown in Figure 7.

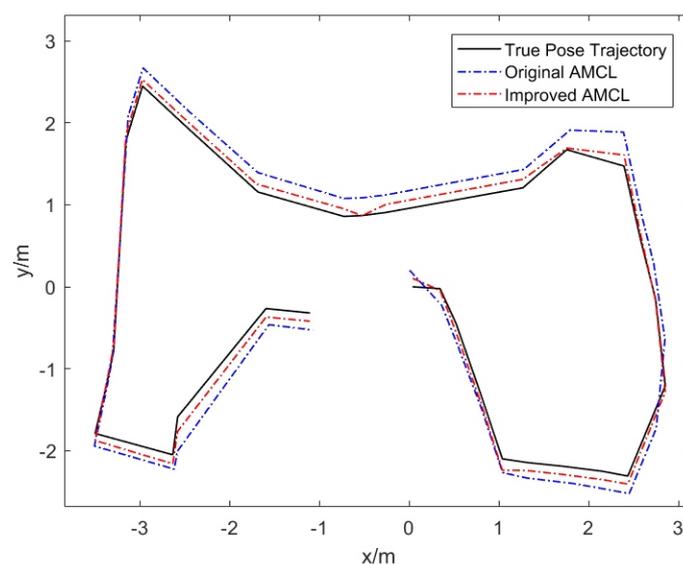
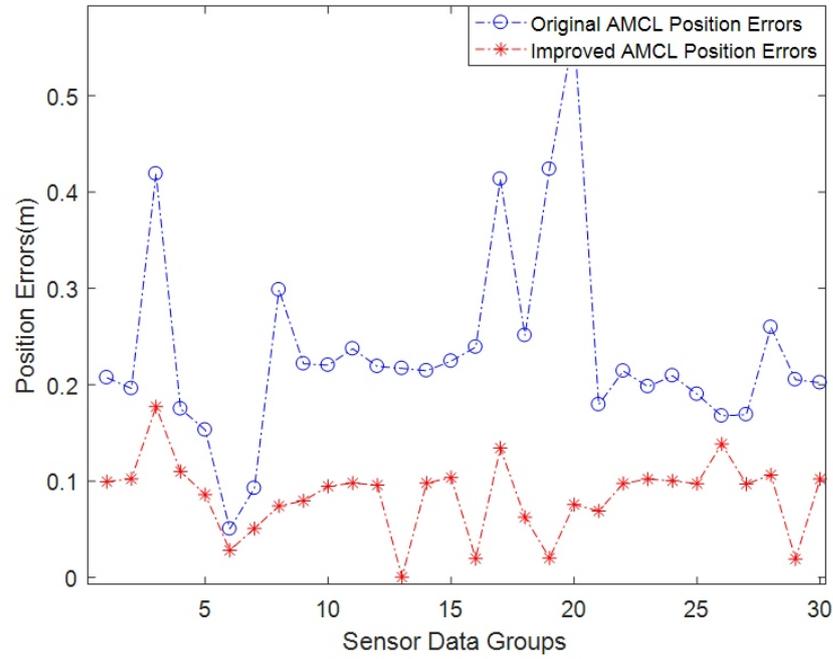


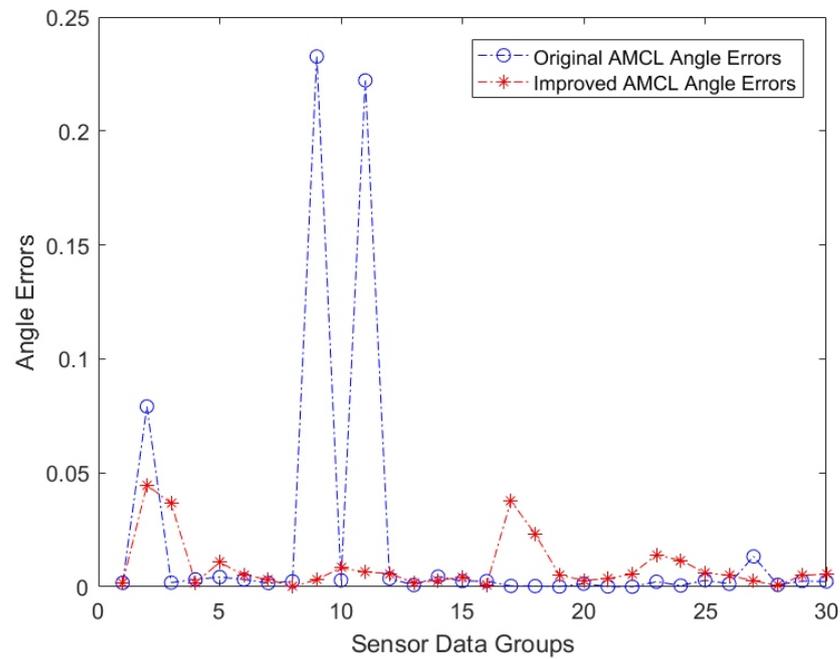
Figure 7. Original AMCL pose trajectory and improved AMCL pose trajectory. Black is the true pose trajectory, blue is the original AMCL pose trajectory, and red is the improved AMCL pose trajectory.

From Figure 7, it can be seen that as the mobile robot keeps moving, the improved AMCL modified pose trajectory is obviously better than the original AMCL pose trajectory,

and the improved AMCL pose trajectory also has good convergence and stability. Selecting 30 datasets from the above figure, the error plots of the improved AMCL modified pose and the original AMCL pose, respectively, with the real pose, are obtained as shown in Figure 8.



(a) Position errors.



(b) Angle errors.

Figure 8. Original AMCL pose error and modified AMCL pose error; (a) is the position error of the original AMCL and the improved AMCL and (b) is the angle error of the original AMCL and the improved AMCL. The blue color shows the error between the estimated and true pose of the original AMCL and the red color represents the error between the estimated and true pose of the improved AMCL.

As can be seen from Figure 8, the improved AMCL's corrected pose error is significantly smaller than the original AMCL's pose error. Using the original AMCL algorithm, the average value of its pose error is around 0.2346 m, and the average value of its angle error is around 0.0199 degrees. With the improved AMCL algorithm, the average value of its position error is around 0.0447 m and the average value of its angle error is around 0.0049 degrees. It can be obtained that the stability of linear and angle localization of the improved AMCL is greatly improved, especially the improved AMCL linear localization has a significant advantage, and its position error always remains lower than the position error of the original AMCL. The improved AMCL algorithm has better convergence and stability compared with the original AMCL. However, the improved AMCL algorithm still has some fluctuations in linear localization, and its linear localization stability is lower than the angle localization stability, so the improved AMCL algorithm is deficient in linear localization stability.

4.3. Experiments on L-PCM Pose Calibration

In order to verify the actual pose accuracy of the L-PCM pose calibration method, we subscribe to the pose topics published by the improved AMCL and the topics published by the L-PCM pose calibration, respectively. After parsing out the generated timestamp-matched RosBag topic packets, the pose trajectory is plotted as shown in Figure 9.

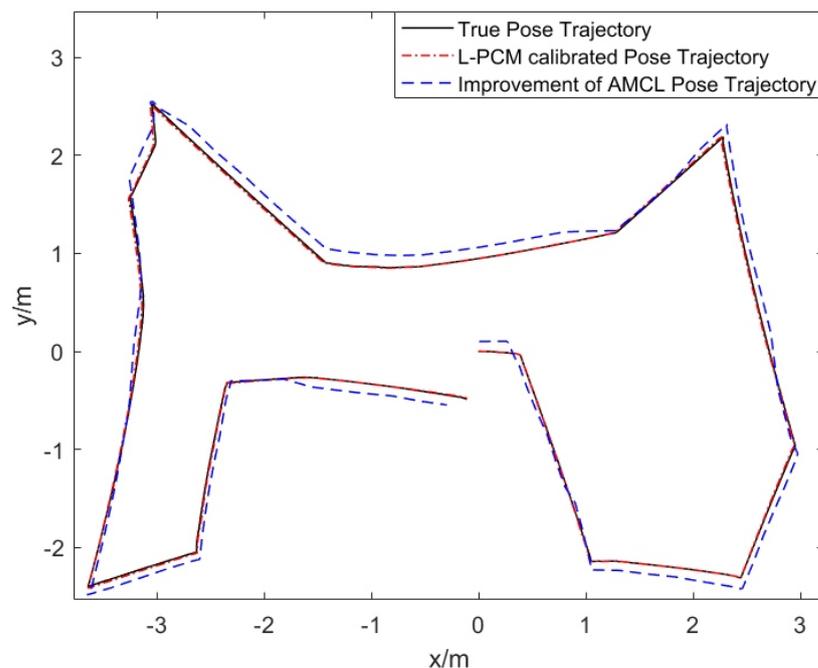


Figure 9. L-PCM calibrated pose trajectory and improved AMCL pose trajectory. The black color represents the real pose trajectory, the red color represents the L-PCM calibrated pose trajectory, and the blue color represents the improved AMCL pose trajectory.

As can be seen from Figure 9, the L-PCM-calibrated pose trajectory is significantly better than the improved AMCL pose trajectory as the mobile robot keeps moving. The L-PCM-calibrated pose trajectory in the figure almost completely coincides with the real pose trajectory, especially with the accumulation of cumulative error over time, which has less influence on the pose estimation and always maintains a better localization effect. The path similarity description (Fréchet distance) between the improved AMCL pose trajectory and the real pose trajectory is 0.1922, while the path similarity description between the L-PCM-calibrated pose trajectory and the real pose trajectory is 0.0479. This clearly demonstrates that the localization accuracy of the L-PCM pose calibration method

is significantly improved. Five sets of timestamp-matched pose data are selected from the above figure, as shown in Table 1.

Table 1. L-PCM pose calibration comparison.

	Improvement of AMCL Pose	L-PCM-Calibrated Pose	True Pose
1	(2.8513, −1.0408, 0.5687)	(2.8545, −1.1833, 0.5613)	(2.8556, −1.2076, 0.5616)
2	(2.7267, −0.0908, 0.7909)	(2.7453, −0.1647, 0.7836)	(2.7459, −0.1444, 0.7894)
3	(2.6047, 0.5160, 0.7781)	(2.5819, 0.5274, 0.7800)	(2.5823, 0.5648, 0.7819)
4	(2.3900, 1.5448, 0.7677)	(2.3958, 1.4058, 0.7693)	(2.3957, 1.4715, 0.7722)
5	(1.7737, 1.7000, 0.9274)	(1.7555, 1.6905, 0.9291)	(1.7586, 1.6709, 0.9249)

According to the pose data in Table 1, the localization accuracy error can be obtained as shown in Table 2.

Table 2. L-PCM-calibrated pose accuracy test.

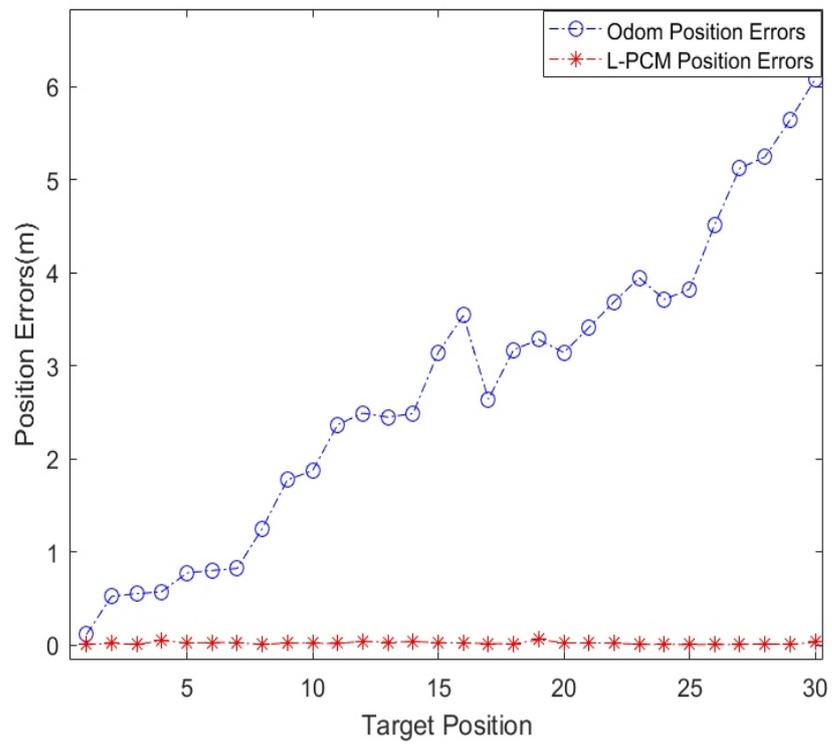
	Improvement of AMCL Pose Errors			L-PCM-Calibrated Pose Errors		
	X/m	Y/m	θ /rad	X/m	Y/m	θ /rad
1	−0.0043	0.1668	0.0071	−0.0011	0.0243	−0.0003
2	−0.0192	0.0536	0.0015	−0.0006	−0.0203	−0.0058
3	0.0224	−0.0488	−0.0038	−0.0004	−0.0374	−0.0019
4	−0.0057	0.0733	−0.0045	1×10^{-4}	−0.0657	−0.0029
5	0.0151	0.0291	0.0025	−0.0031	0.0196	0.0042
Maximum Values	0.0224	0.1668	0.0071	0.0031	0.0657	0.0058
Minimum Values	0.0043	0.0291	0.0015	1×10^{-4}	0.0196	0.0003
Average Values	0.01334	0.07432	0.00388	0.00106	0.03346	0.00302

From the sensor data in Table 2, it can be seen that the average value of the position error of the improved AMCL is around 0.04383 m and the average value of the angle error is around 0.00388. The pose error after L-PCM pose calibration is around 0.01726 m and the average angle error is around 0.00302. As a result, it can be concluded that both the linear and angle localization of the mobile robot are significantly improved after the L-PCM pose calibration, and at the same time, the L-PCM algorithm in this paper has a better advantage in linear localization accuracy, and the pose accuracy is obviously improved.

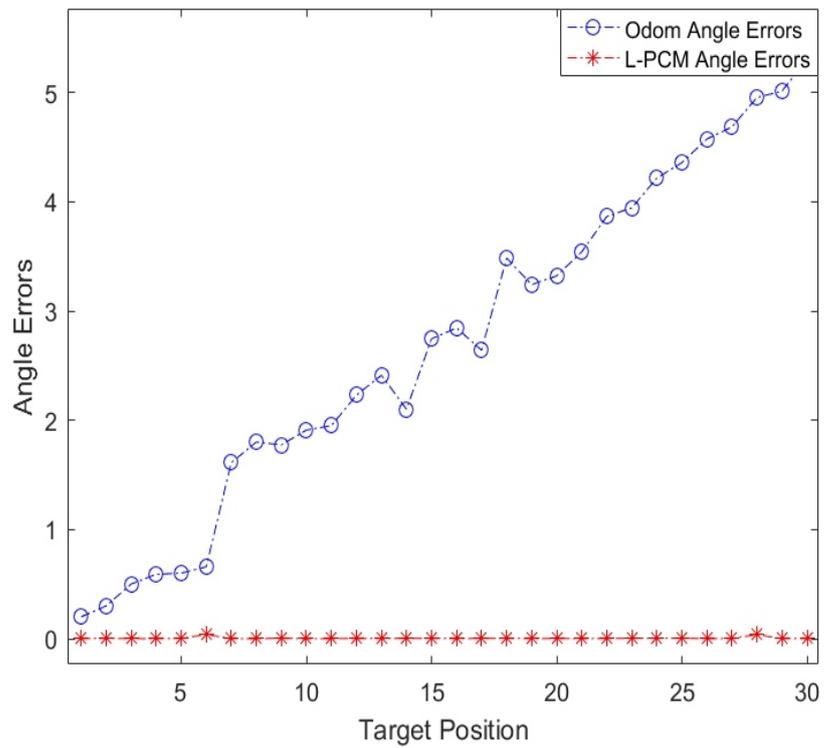
We perform ablation experiments by changing the sensors [23,40,41]. First, when the mobile robot only uses the odometer sensor for pose estimation, the pose information from the odometer and the L-PCM is collected and mathematically counted against the results of the true pose information to obtain the pose error data. The result of the pose error is shown in Figure 10.

As can be seen in Figure 10, the linear and angle localization results of the odometer have large errors from the actual values. At the same time, the pose error of the odometer gradually increases as the moving distance increases, which shows that the cumulative error of the odometer has a gradually larger impact on the accuracy of the estimated pose.

Second, when the mobile robot uses only the LiDAR sensor for pose estimation, two algorithms exist, Gmapping and cartographer, to collect the pose information. We collect the pose information from LiDAR and L-PCM and perform mathematical statistics with the results of the true pose information to obtain the pose error data. The result of the pose error is shown in Figure 11.

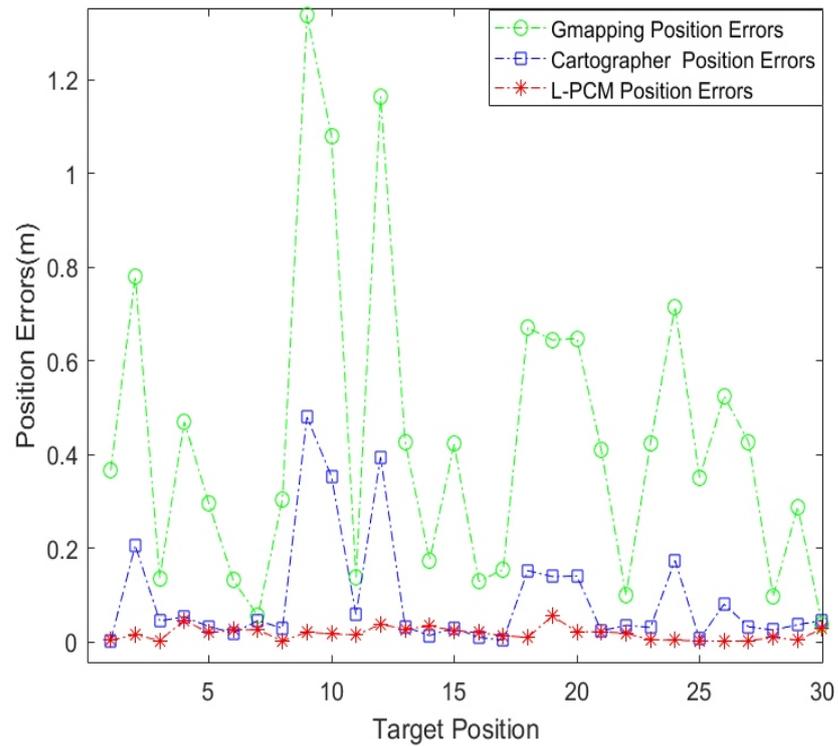


(a) Position errors.

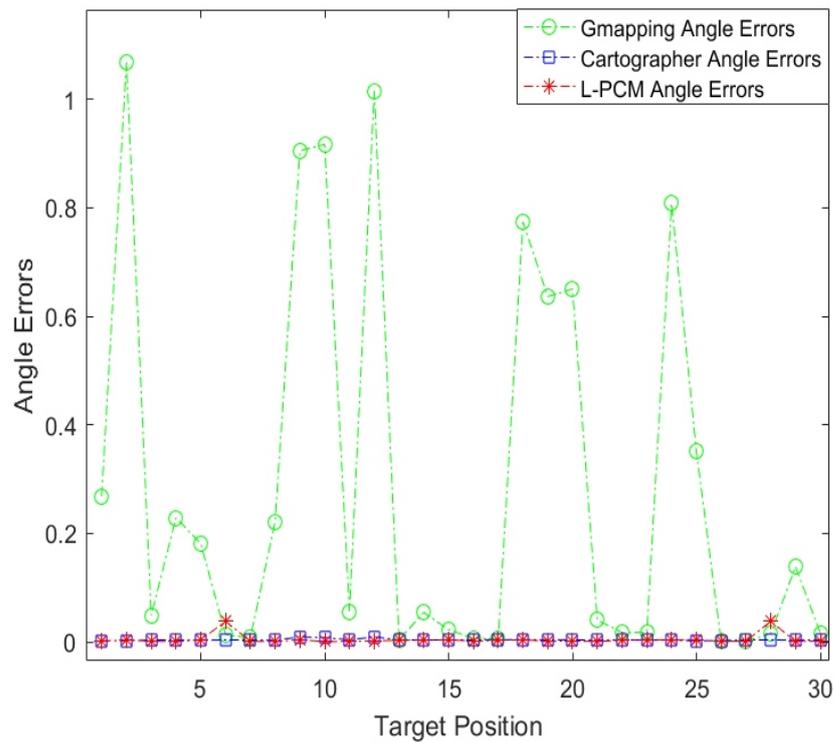


(b) Angle errors.

Figure 10. Odom pose error and L-PCM pose error; (a) is the position error of odom and L-PCM and (b) is the angle error of odom and L-PCM. The blue color indicates the error of odom and the red color indicates the error of L-PCM.



(a) Position errors.



(b) Angle errors.

Figure 11. Gmapping, cartographer, and L-PCM pose error; (a) is the position error of Gmapping, cartographer, and L-PCM and (b) is the angle error of Gmapping, cartographer, and L-PCM. The green color indicates the error of Gmapping, the blue color indicates the error of the cartographer, and the red color indicates the error of L-PCM.

From Figure 11, it can be seen that the linear and angle localization results of the Gmapping algorithm have a large error from the actual value, and the pose accuracy of the

cartographer algorithm is significantly higher than that of the Gmapping algorithm. At the same time, the linear and angle localization results of the cartographer algorithm are more stable, with less error from the actual values; in particular, the yaw angle localization is smooth with little fluctuation. The angle localization accuracy of the cartographer algorithm is almost the same as the L-PCM, and even its pose error and stability are slightly better than the L-PCM at times.

We simultaneously collected the pose information of odometer localization, EKF fusion, UKF fusion, Gmapping algorithm, cartographer algorithm, original AMCL algorithm, and L-PCM algorithm, and mathematically counted them with the results of the real pose information to obtain the data on the mobile robot's pose error, as shown in Table 3.

Table 3. Mobile robot pose error analysis.

	Position Error Mean (m)	Position Error Variance (m)	Angle Error Mean (rad)	Angle Error Variance (rad)
Odom [23]	2.3201	2.8716	2.1495	3.8112
EKF [7,16]	1.2079	2.7475	1.0254	3.7484
UKF	0.7732	2.4570	0.6142	3.1002
Gmapping [42]	0.4275	0.1023	0.2754	2.8970
Cartographer [40]	0.0827	0.0247	0.0037	0.4541
AMCL [41]	0.1146	0.0319	0.0199	1.7524
Ours	0.0163	0.0226	0.0040	0.4832

As can be seen from Table 3, the L-PCM algorithm proposed in this paper has a much smaller pose error, and the accuracy of both linear and angle localization is significantly improved. When only the odometer is used for localization, the linear and angle localization results have significantly larger errors from the actual values. When the odometer is combined with the IMU, the pose error is significantly reduced and the advantage of the UKF fusion algorithm is more obvious. When only LiDAR is used for localization, the cartographer algorithm is significantly more accurate than the Gmapping algorithm, and the linear and angle localization results of the cartographer algorithm are more accurate and stable.

Among these algorithms, the L-PCM algorithm has a clear advantage in linear localization accuracy. Compared with the original AMCL algorithm, the average localization error of the L-PCM algorithm is reduced by 0.0973 m in the linear direction and 0.0159 in the yaw angle. By comparing the standard deviation results of the two algorithms, it can be clearly seen that the standard deviation of the L-PCM algorithm proposed in this paper is smaller for both the linear localization value and the yaw angle localization value compared with the original AMCL algorithm. The proposed optimization measures play a great positive role in reducing the volatility of the localization results and enhancing localization stability. At the same time, we can see that the overall localization accuracy of the cartographer algorithm is better than that of the AMCL algorithm, and the angle localization accuracy and stability are slightly better than that of the L-PCM algorithm in this paper. Compared with the cartographer's algorithm, the average localization error of the L-PCM algorithm is reduced by 0.0654 m in the linear direction, and the average localization error in the yaw angle is only 0.0003. Although the L-PCM algorithm proposed in this paper achieves better results, compared with the cartographer's algorithm, the localization accuracy and stability in the yaw angle still need to be improved continuously.

5. Discussion

Combined with the above simulation experiments and the analysis of the localization accuracy, the L-PCM algorithm proposed in this paper has the best localization effect, significantly improves the localization accuracy of the original AMCL, and ensures the convergence and stability of the algorithm. UKF ensures the accuracy of the pre-position estimation during the localization process, which greatly improves the stability of linear and

angle localization of the improved AMCL, and also establishes the outstanding advantage of linear localization. After the L-PCM pose calibration, the advantage of linear localization is more obvious, the angle localization is more accurate, and the yaw angle localization is always kept at a high precision, which has great stability.

Analyzing the localization accuracy and stability of the seven algorithms mentioned above, the L-PCM algorithm proposed in this paper has a much smaller pose error, and the accuracy of both linear and angle localization is significantly improved. When using odometer localization, EKF fusion, UKF fusion, or Gmapping algorithm, the linear, and angle localization results have significantly larger errors from the actual values. Compared to the original AMCL algorithm, L-PCM has significantly lower localization errors in both linear direction and yaw angle and has higher stability. Compared with the cartographer algorithm, the average localization error of the L-PCM algorithm is reduced by 0.0654 m in the linear direction and only 0.0003 in the yaw angle.

6. Conclusions

In this paper, an L-PCM mobile robot pose calibration method is proposed. The method accomplishes UKF-based fusion of the odometer and IMU data to realize pose estimation, and experimentally obtains that the pose estimation accuracy of UKF fusion is better than that of EKF fusion. The original AMCL algorithm is improved by the UKF fusion algorithm and is obtained through experiments; the improved AMCL shows significant convergence in the pose tracking and global localization tasks, with the average value of the position error being about 0.0447 m and the average value of the angle error stabilized at about 0.0049. Finally, we complete the PL-ICP point cloud registration to directly calibrate the improved AMCL pose. Experimentally, we find that the corrected position error is approximately 0.01726 m, and the average value of the angle error is around 0.00302, which significantly improves the accuracy and stability of the pose.

Compared with the cartographer algorithm, the L-PCM algorithm has an obvious advantage in linear localization accuracy, but there are deficiencies in angle localization accuracy and stability. Therefore, the yaw angle localization accuracy and stability of the L-PCM algorithm proposed in this paper still need to be improved continuously. In addition, the inclusion of the point cloud registration technique in this paper increases the computation time of the algorithm, and the sparseness of the point cloud may impact the performance of the algorithm, leading to a decrease in matching accuracy. In order to shorten the localization time of L-PCM, subsequent optimization of the point cloud registration algorithm is needed. For point cloud sparsity, various strategies will be attempted to optimize future performance, such as increasing sensor sampling density, adopting feature enhancement techniques, and optimizing the matching algorithm to adapt to sparse data. These measures aim to improve the localization performance of L-PCM in sparse point cloud environments.

Author Contributions: Project administration, D.N.; writing—original draft preparation, D.N.; supervision, S.H.; formal analysis, D.N.; data curation, D.N. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China: 62276118.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: In this paper, the simulation test scenarios used GazeBo9 software. Data is contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Jia, Y.; Du, J.; Zhang, W. An Overview of SLAM. In *Proceedings of 2018 Chinese Intelligent Systems Conference*; Lecture Notes in Electrical Engineering; Springer: Singapore, 2019; Volume 528 (Volume I), pp. 673–681. [\[CrossRef\]](#)
2. Sinisa, M. Evaluation of SLAM Methods and Adaptive Monte Carlo Localization. Ph.D. Thesis, Vienna University of Technology, Vienna, Austria, 2022.
3. Hanten, R.; Buck, S.; Otte, S.; Zell, A. Vector-AMCL: Vector based Adaptive Monte Carlo Localization for Indoor Maps. In *Proceedings of the 14th International Conference on Intelligent Autonomous Systems (IAS-14)*, Shanghai, China, 3–7 July 2016.
4. Chung, M.A.; Lin, C.W. An Improved Localization of Mobile Robotic System Based on AMCL Algorithm. *IEEE Sens. J.* **2021**, *22*, 900–908. [\[CrossRef\]](#)
5. Garcia, A.; Martín, F.; Guerrero, J.M.; Rodríguez, F.J.; Matellán, V. Portable multi-hypothesis Monte Carlo localization for mobile robots. In *Proceedings of the 2023 IEEE International Conference on Robotics and Automation (ICRA)*, London, UK, 29 May–2 June 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 1933–1939.
6. Moravec, H.; Elfes, A. High resolution maps from wide angle sonar. In *Proceedings of the 1985 IEEE International Conference on Robotics and Automation*, St. Louis, MO, USA, 25–28 March 1985.
7. Hachemi, L.; Guiatni, M.; Nemra, A. Fault Diagnosis and Reconfiguration for Mobile Robot Localization Based on Multi-Sensors Data Fusion. *Unmanned Syst.* **2021**, *10*, 69–91. [\[CrossRef\]](#)
8. Maken, F.A.; Ramos, F.; Ott, L. Stein ICP for Uncertainty Estimation in Point Cloud Matching. *IEEE Robot. Autom. Lett.* **2022**, *7*, 1063–1070. [\[CrossRef\]](#)
9. Bienkowski, M.A.; Hoebel, L.J. Integrating AI Components for a Military Planning Application. In *Proceedings of the American Association for Artificial Intelligence*, Madison, MI, USA, 26–30 July 1998.
10. Hashimoto, K.; Takanishi, A. WABIAN and Other Waseda Robots. In *Humanoid Robotics: A Reference*; Springer: Dordrech, The Netherlands, 2019.
11. Bilesan, A.; Komizunai, S.; Tsujita, T.; Konno, A. Improved 3D Human Motion Capture Using Kinect Skeleton and Depth Sensor. *J. Robot. Mechatronics* **2021**, *33*, 1408–1422. [\[CrossRef\]](#)
12. Cheon, E.J.; Su, N.M. Integrating roboticist values into a Value Sensitive Design framework for humanoid robots. In *Proceedings of the ACM/IEEE International Conference on Human-robot Interaction*, Christchurch, New Zealand, 7–10 March 2016.
13. Lauttia, T. Adaptive Monte Carlo Localization in ROS. Bachelor’s Thesis, Tampere University, Tampere, Finland, 2022.
14. Wu, M.H.; Yu, J.C.; Lin, Y.C. Study of Autonomous Robotic Lawn Mower Using Multi-Sensor Fusion Based Simultaneous Localization and Mapping. In *Proceedings of the 2022 International Conference on Advanced Robotics and Intelligent Systems (ARIS)*, Taipei, Taiwan, 24–27 August 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 1–4.
15. Zhu, W.; Cheng, X. Indoor Localization Method of Mobile Educational Robot Based on Visual Sensor. *J. Internet Technol.* **2023**, *24*, 205–215.
16. Zhou, G.; Luo, J.; Xu, S.; Zhang, S.; Xiang, K. An EKF-based multiple data fusion for mobile robot indoor localization. *Assem. Autom.* **2021**. [\[CrossRef\]](#)
17. Muffert, M. Incremental Map Building with Markov Random Fields and its Evaluation. Ph.D. Thesis, Universitäts- und Landesbibliothek Bonn, Bonn, Germany, 2018.
18. Fox, D.; Burgard, W.; Dellaert, F.; Thrun, S. Monte Carlo Localization: Efficient Position Estimation for Mobile Robots. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence and Eleventh Conference on Innovative Applications of Artificial Intelligence*, Orlando, FL, USA, 18–22 July 1999.
19. Do, H.N.; Jadaliha, M.; Temel, M.; Choi, J. Fully Bayesian Field Slam Using Gaussian Markov Random Fields. *Asian J. Control* **2016**, *18*, 1175–1188. [\[CrossRef\]](#)
20. Gimenez, J.; Amicarelli, A.; Toibero, J.M.; Sciascio, F.D.; Carelli, R.O. Iterated Conditional Modes to Solve Simultaneous Localization and Mapping in Markov Random Fields Context. *Int. J. Autom. Comput.* **2018**, *15*, 310–324. [\[CrossRef\]](#)
21. Ge, G.; Zhang, Y.; Wang, W.; Jiang, Q.; Hu, L.; Wang, Y. Text-MCL: Autonomous mobile robot localization in similar environment using text-level semantic information. *Machines* **2022**, *10*, 169. [\[CrossRef\]](#)
22. Fikri, A.A.; Anifah, L. Mapping and Positioning System on Omnidirectional Robot Using Simultaneous Localization and Mapping (Slam) Method Based on Lidar. *J. Teknol* **2021**, *83*, 41–52. [\[CrossRef\]](#)
23. Kim, K. Integration of Real-Time Semantic Building Map Updating with Adaptive Monte Carlo Localization (AMCL) for Robust Indoor Mobile Robot Localization. *Appl. Sci.* **2023**, *13*, 909.
24. Kaczmarek, A.; Rohm, W.; Klingbeil, L.; Tchorzewski, J. Experimental 2D extended Kalman filter sensor fusion for low-cost GNSS/IMU/Odometers precise positioning system. *Measurement* **2022**, *193*, 110963. [\[CrossRef\]](#)
25. Alhamdi, E.; Hedjar, R. Comparative Study of Two Localization Approaches for Mobile Robots in an Indoor Environment. *J. Robot.* **2022**, *2022*, 1999082. [\[CrossRef\]](#)
26. Montemarlo, M. FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem. In *Proceedings of the American Association for Artificial Intelligence*, North Falmouth, MA, USA, 15–17 November 2002.
27. Woo, A.; Fidan, B.; Melek, W.W.; Zekavat, S.A.R.; Buehrer, R.M. *Localization for Autonomous Driving*; John Wiley & Sons, Ltd.: Hoboken, NJ, USA, 2019.
28. Zhu, D.; Sun, X.; Wang, L.; Liu, B.; Ji, K. Mobile Robot SLAM Algorithm Based on Improved Firefly Particle Filter. In *Proceedings of the International Conference on Robots & Intelligent System*, Macau, China, 4–8 November 2019.

29. Zhao, S.; Gu, J.; Ou, Y.; Zhang, W.; Peng, H. IRobot self-localization using EKF. In Proceedings of the 2016 IEEE International Conference on Information and Automation (ICIA), Ningbo, China, 1–8 August 2016.
30. Chen, C.L.; Li, J.; Li, M.; Xie, L. Model Predictive Trajectory Tracking Control of Automated Guided Vehicle in Complex Environments. In Proceedings of the 2018 IEEE 14th International Conference on Control and Automation (ICCA), Anchorage, AK, USA, 12–15 June 2018.
31. Rashidi, H.; Tsang, E.P.K. *A Complete and an Incomplete Algorithm for Automated Guided Vehicle Scheduling in Container Terminals*; Pergamon Press, Inc.: Oxford, UK, 2011.
32. Perez, R.J.I.; Álvarez Gloria Patricia Jaramillo. Application of mixed-integer linear programming in a car seats assembling process. *Pesqui. Oper.* **2011**, *31*, 593–610.
33. Julier, S.J.; Laviola, J.J. On Kalman Filtering With Nonlinear Equality Constraints. *IEEE Trans. Signal Process. Publ. IEEE Signal Process. Soc.* **2007**, *55*, 2774–2784. [[CrossRef](#)]
34. Zeng, Q.; Tao, X.; Yu, H.; Ji, X.; Chang, T.; Hu, Y. An Indoor 2D LiDAR SLAM and Localization Method Based on Artificial Landmark Assistance. *IEEE Sens. J.* **2023**, *24*, 3681–3692. [[CrossRef](#)]
35. He, S.; Song, T.; Wu, X. An improved adaptive monte carlo localization (amcl) for automated mobile robot (amr). In Proceedings of the 22nd COTA International Conference of Transportation Professionals, Changsha, China, 8–11 July 2022; pp. 171–181.
36. Huang, Y.H.; Lin, C.T. Indoor Localization Method for a Mobile Robot Using LiDAR and a Dual AprilTag. *Electronics* **2023**, *12*, 1023. [[CrossRef](#)]
37. Serafin, J.; Grisetti, G. *NICP: Dense Normal Based Point Cloud Registration*; IEEE: Piscataway, NJ, USA, 2015.
38. Nüchter, A. Parallelization of Scan Matching for Robotic 3D Mapping. In Proceedings of the 3rd European Conference on Mobile Robots, EMCR 2007, Freiburg, Germany, 19–21 September 2007.
39. Best, P.J. A method for registration of 3-D shapes. *IEEE Trans. Pattern Anal. Mach. Vis.* **1992**, *14*, 586–606.
40. Liu, Y.; Wang, C.; Wu, H.; Wei, Y.; Ren, M.; Zhao, C. Improved LiDAR localization method for mobile robots based on multi-sensing. *Remote Sens.* **2022**, *14*, 6133. [[CrossRef](#)]
41. He, S.; Song, T.; Wang, P.; Ding, C.; Wu, X. An Enhanced Adaptive Monte Carlo Localization for Service Robots in Dynamic and Featureless Environments. *J. Intell. Robot. Syst.* **2023**, *108*, 6. [[CrossRef](#)]
42. Sandoval-Castro, Y.; Ruiz-Torres, M.; Martínez, A.I.R.; Borade, Y. Comparison of different techniques of 2D simultaneous localization and mapping for a mobile robot by using ROS. In Proceedings of the XVI International Engineering Congress—CONiiN 2020, Hong Kong, China, 20–21 August 2020.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.