

Retraction

Retracted: SLAM of Mobile Robot for Wireless Communication Based on Improved Particle Filter

Wireless Communications and Mobile Computing

Received 28 November 2023; Accepted 28 November 2023; Published 29 November 2023

Copyright © 2023 Wireless Communications and Mobile Computing. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This article has been retracted by Hindawi, as publisher, following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of systematic manipulation of the publication and peer-review process. We cannot, therefore, vouch for the reliability or integrity of this article.

Please note that this notice is intended solely to alert readers that the peer-review process of this article has been compromised.

Wiley and Hindawi regret that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

References

- [1] D. Zhu, M. Wang, M. Su, S. Liu, and P. Guo, "SLAM of Mobile Robot for Wireless Communication Based on Improved Particle Filter," *Wireless Communications and Mobile Computing*, vol. 2021, Article ID 9263277, 14 pages, 2021.

Research Article

SLAM of Mobile Robot for Wireless Communication Based on Improved Particle Filter

Daixian Zhu¹, Mingbo Wang¹, Mengyao Su¹, Shulin Liu², and Ping Guo¹

¹College of Communication and Information Engineering, Xi'an University of Science and Technology, Xi'an 710054, China

²College of Electrical and Control Engineering, Xi'an University of Science and Technology, Xi'an 710054, China

Correspondence should be addressed to Daixian Zhu; zhudaixian@xust.edu.cn

Received 3 July 2021; Revised 23 July 2021; Accepted 9 August 2021; Published 30 August 2021

Academic Editor: Balakrishnan Nagaraj

Copyright © 2021 Daixian Zhu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The mobile robot is moved by receiving instructions through wireless communication, and the particle filter is used to simultaneous localization and mapping. Aiming at the problem of the degradation of particle filter weights and loss of particle diversity, which leads to the decrease of filter accuracy, this paper uses the plant cell swarm algorithm to optimize the particle filter. First of all, combining the characteristics of plant cells that affect the growth rate of cells when the auxin content changes due to light stimulation realizes the optimization of the particles after importance sampling, so that they are concentrated in the high-likelihood area, and the problem of particle weight degradation is solved. Secondly, in the process of optimizing particle distribution, the auxin content of each particle is different, which makes the optimization effect on each particle different, so it effectively solves the problem of particle diversity loss. Finally, a simulation experiment is carried out. During the experiment, the robot moves by receiving control commands through wireless communication. The experimental results show that the algorithm effectively solves the problem of particle weight degradation and particle diversity loss and improves the filtering accuracy. The improved algorithm is verified in the simultaneous localization and mapping of the robot, which effectively improves the robot's performance at the same time positioning accuracy. Compared with the classic algorithm, the robot positioning accuracy is increased by 49.2%. Moreover, the operational stability of the algorithm has also been improved after the improvement.

1. Introduction

The particle filter (PF) algorithm is a filtering method based on Monte Carlo and Bayesian estimation. The basic idea is to use a set of weighted samples randomly drawn from the probability density to approximate the posterior probability density. Since the proposed particle filter algorithm, it has been widely used in the field of nonlinear system parameter estimation, such as target tracking [1], system state detection [2], and simultaneous localization and mapping (SLAM) of robots [3].

Although the classical particle filter algorithm has achieved good filtering effect in practical application, it uses the prior probability density function as the important density function, which will lead to the increase of important weight variance with the increase of time. The phenomenon that the weights are concentrated on a few particles occurs,

that is, the degradation of particle weights. In order to solve the problem of particle weight degradation, the resampling strategy is introduced in the classical particle filter algorithm, and the corresponding algorithm is called Rao-Blackwellized particle filters (RBPF); Rao-Blackwellized particle filter is applied to robotic SLAM and is named FastSLAM1.0 algorithm [4]. The FastSLAM2.0 algorithm [5] fuses the noise of the sensor on the sampling function of the predicted pose to improve the positioning accuracy caused by the reduction of particle diversity.

Frequent resampling steps lead to gradual dissipation of particles; the current popular solution is to solve the problem of particle weight degradation and particle diversity loss through the efficient combination of one or several meta-heuristic algorithms [6] and particle filtering. Shiming and others [7] use the rules of planet formation in the universe to take the particles sampled from the importance function

as the dust in the universe and the global optimal value as the central dust. Through, the attraction and repulsion of the central dust, the particle set is distributed near the high-likelihood region, which alleviated the weight degradation and improved the diversity of particles [8]. This improved algorithm has also been successfully applied to SLAM and achieved a good positioning effect. However, due to the strong optimization ability of the gravitational field algorithm, there is a risk of premature maturity in the iterative process of the algorithm. The literature [9–11] optimizes the original particle filter improvement algorithm and improves the particle weight degradation and diversity loss jointly through the efficient combination of several algorithms, so as to improve the filtering performance of the algorithm. Although the above algorithm improves the performance of particle filtering, a large number of particles are needed to realize state estimation in the scene with complex environment and strong noise. The fusion of multiple algorithms will inevitably cause the problems of long running time and low operating efficiency.

In this paper, an improved plant cell swarm algorithm (PCSA) improves PF by optimizing the particle distribution after importance sampling and omit the resampling process in particle filter algorithm, by initializing the plant cell swarm, determining the position of the strongest light, calculating the growth rate, and updating the plant cell swarm, and the PCSA-PF method is obtained. The PCSA-PF is applied to FastSLAM 2.0 framework, which can be called PCSA-FastSLAM. Experimental results show that the improved algorithm can effectively improve the particle degradation and particle diversity loss in FastSLAM 2.0 algorithm and improve the accuracy of robot positioning mapping.

The rest of the paper is organized as follows. Section 1 introduces briefly the PF and FastSLAM. Section 2 introduces the principle of plant cell swarm algorithm. Section 3 discusses the proposed PCSA-PF and PCSA-FastSLAM in detail. Section 4 gives the simulation results and analyzes the performance of the PCSA-PF and PCSA-FastSLAM in detail. Finally, Section 5 concludes the paper and gives some orientations for future work.

2. Particle Filter Algorithm and Its Application in Robot Autonomous Mobile Mapping

2.1. Particle Filter Algorithm. Particle filtering is a method to realize recursive Bayesian filtering through Monte Carlo thought [12], which uses Monte Carlo method to solve the integral problem in Bayesian estimation.

The main steps of the classical PF algorithm are as follows: initialization particle set, importance sampling, importance weight calculation, resampling, and state estimation.

Initialization: at $k=0$, sample N particles are from the prior distribution $p(x_0)$, the set weight of each particle is $1/N$, and the sampled particle set is $\{x_0^i, 1/N\}_{i=1}^N$.

Importance sampling: at $k \geq 1$, the particles are sampled from the importance density function. The importance density function is

$$q(x_k^i | x_{k-1}^i, z_k) = p(x_k^i | x_{k-1}^i). \quad (1)$$

Calculation of importance weight: calculate the weight of each particle for the sampled particles. The weight calculation formula is

$$\tilde{w}_t^i = \tilde{w}_{t-1}^i \frac{p(z_t | \tilde{x}_t^i) p(\tilde{x}_t^i | \tilde{x}_{t-1}^i)}{q(\tilde{x}_t^i | \tilde{x}_{t-1}^i)}. \quad (2)$$

Weight normalization: normalize weights for particles with different weights:

$$w_t^i = \frac{\tilde{w}_t^i}{\sum_{i=1}^N \tilde{w}_t^i}. \quad (3)$$

Resampling: in order to solve the problem of particle weight degradation, the resampling strategy is introduced. For the evaluation of the degree of particle weight degradation, the effective particle number is adopted as the criteria:

$$\tilde{N}_{\text{eff}} = \frac{1}{\sum_{i=1}^N (\tilde{w}_t^i)^2}, \quad (4)$$

by comparing the effective particle number \tilde{N}_{eff} with the set threshold N_{thres} . If $\tilde{N}_{\text{eff}} \leq N_{\text{thres}}$, the resampling strategy is carried out: that is, the particles with heavier weight are copied multiple times, and the particles with lower weight are copied to reduce the number of times or even without copying. The number of particles after resampling is equal to the original particles, and the weight is the same as $1/N$. If not, continue to perform the state estimation operation.

State estimation:

$$\tilde{x}_t = \sum_{i=1}^N w_t^i x_t^i. \quad (5)$$

2.2. FastSLAM2.0 Algorithm. PF algorithm is applied to SLAM of robots. The core idea is to decompose the system state estimation through the Rao-Blackwellized decomposition method and decompose it into path estimation and environment map estimation. The decomposition formula is

$$\begin{aligned} p(x_{1:k}, m | z_{1:k}, u_{0:k-1}) \\ = p(x_{1:k} | z_{1:k}, u_{0:k-1}) \prod_{i=1}^M p(m_i | x_{1:k}, z_{1:k}, u_{0:k-1}). \end{aligned} \quad (6)$$

In the above formula, $p(x_{1:k}, m | z_{1:k}, u_{0:k-1})$ represents the mathematical formula expression of the SLAM problem, $p(x_{1:k} | z_{1:k}, u_{0:k-1})$ represents the robot path estimation, and $\prod_{i=1}^M p(m_i | x_{1:k}, z_{1:k}, u_{0:k-1})$ represents the robot environment map estimation.

Due to the different selection methods of the proposed distribution function, the FastSLAM algorithm can be divided into the FastSLAM1.0 algorithm and the FastSLAM2.0 algorithm. Compared with the FastSLAM1.0

algorithm that uses the prior distribution $p(x_{k+1}|x_k^i, u_k)$ as the proposal distribution function, the proposal distribution function of the FastSLAM2.0 algorithm is updated by EKF from $p(x_{k+1}|x_k^i, u_k)$. This improvement makes the proposal distribution function contain historical information and observation information at the current moment. Therefore, it will effectively alleviate the problem of algorithm degradation and improve the filtering accuracy of the algorithm [13].

In order to solve the problem of particle weight degradation, the FastSLAM2.0 algorithm adopts a resampling strategy, which will cause the loss of particle diversity and reduce robot positioning accuracy. Therefore, the key to improving the accuracy of robot positioning and mapping is to effectively solve the particle weight degradation and loss of particle diversity [14].

3. Plant Cell Swarm Algorithm

On the basis of studying the process of plant positive growth, plant positive cell swarm algorithm is proposed, hereinafter referred to as plant cell swarm algorithm (PCSA) [15].

3.1. Growth Mechanism of Positive Plants. Positivity is the result of long-term evolution of plants, which helps them absorb more light energy. Under the stimulation of light, the distribution of auxin in the plant changes. The auxin is transported from the strong light side to the backlight side. The auxin concentration on the backlight side is higher than the strong light side, so the growth rate of plant cells on the backlight side is faster than the strong light side and makes the plant body bend toward the sun, so the plant grows toward the sun.

Auxin plays a crucial role in plant growth, which is manifested in duality: auxin can not only promote cell growth but also inhibit cell growth. The effect of auxin on plant cells is shown in Figure 1.

3.2. Plant Cell Swarm Algorithm. The plant cell swarm algorithm simulates the growth behavior of sunny plants, takes the location of plant cells with the highest light intensity and the distance from other cells as the standard of auxin concentration distribution, determines the growth rate of different plant cells based on auxin concentration, and finds the maximum location of light intensity through multiple iterations.

The plant cell swarm algorithm is mainly divided into the following five steps: initializing the plant cell swarm, determining the position of the strongest light, allocating auxin, calculating the growth rate, and updating the plant cell swarm.

Step 1. Initialize the plant cell swarm.

Randomly generate a plant cell swarm with a population of N . In the process of generating, keep the cell swarm covering all feasible solutions. Assuming the search in the D -dimensional search space, the k -dimensional coordinate is x_k , the position of the m th cell is $X_m = (x_{m1}, x_{m2}, \dots, x_{mk}, \dots, x_{md})$, and there are

$$x_{mk} = x_{k \min} + \text{rand} (x_{k \max} - x_{k \min}). \quad (7)$$

In formula (1), $x_{k \min}$ is the minimum value of the k -dimension coordinate in the domain of definition, $x_{k \max}$ is the maximum value of the k -dimension coordinate, and rand is a random number in the interval $[0,1]$.

Step 2. Determine the location of the strongest light.

Assuming the function is $P_i = f(X_i) (i = 1, 2, \dots, d)$, p_i represents the total amount of sunlight that each plant cell is exposed to in the D -dimensional space, and f represents the mapping of the plant cell in the D -dimensional space and the total amount of light intensity. By calculating the total amount of light exposed to each cell in the plant cell swarm and comparing it, the maximum light intensity in a cell is recorded as sun ; the position of the plant cell with the maximum light intensity is X_{sun} ; that is, the formula is expressed as

$$\text{sun} = \max \{f(X_1), f(X_2) \dots f(X_D)\}. \quad (8)$$

Step 3. Allocate auxin.

First, calculate the distance between the position of each cell in the plant cell swarm and the maximum light intensity of X_{sun} . The distance was calculated using the European formula, and the calculation formula is

$$L(X_i, X_{\text{sun}}) = \left[\sum_{k=1}^d (X_i - X_{\text{sun}})^2 \right]^{1/2}. \quad (9)$$

In the above formula, X_i represents the position of the i th cell, d represents the dimension of the plant cell, and $L(X_i, X_{\text{sun}})$ represents the distance between the i th cell and maximum light intensity.

The distance value calculated by the above formula is used to calculate the auxin content of each cell location. The calculation formula is

$$Q_i = 1 + \frac{L(x_i, X_{\text{sun}})}{(1 + P_i)}. \quad (10)$$

Step 4. Calculate the growth rate.

From the above biological mechanism of the effect of auxin on the growth of plant cells, it can be seen that for positive organisms, auxin will move when it receives light stimulation. The farther the plant cell is from the maximum light intensity, the higher the auxin content is. In addition, the different content of auxin has different promoting effects on plant cell growth. The mechanism of action is as follows:

- (1) $0 < Q_i \leq \alpha$, with the increase of auxin, the growth promotion effect on plant cells gradually increases
- (2) $\alpha < Q_i \leq \beta$, with the increase of auxin, the growth promotion effect on plant cells gradually weakens
- (3) $\beta < Q_i \leq \gamma$, with the increase of auxin, it has an inhibitory effect on the growth of plant cells

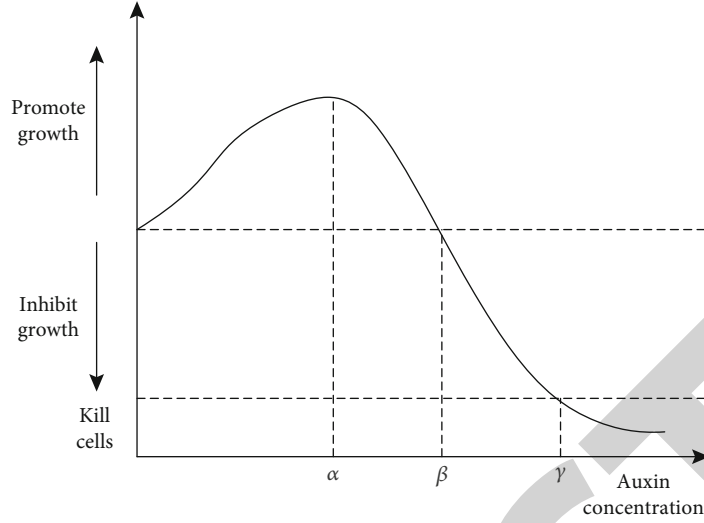


FIGURE 1: The effect of auxin concentration on plant cells.

(4) $\gamma < Q_i$, auxin kills cells

In summary, auxin with different contents has different effects on the growth of plant cells. When the plant receives light stimulation and completes auxin distribution, the calculation formula for the growth rate of each plant cell V_i is

$$V_i = \begin{cases} \frac{aQ_i}{b} L(X_i, X_{\text{sun}}), & Q_i \leq \alpha, \\ \frac{-aQ_i + 2a\alpha}{c} L(X_i, X_{\text{sun}}), & \alpha < Q_i \leq \gamma, \\ 0, & Q_i > \gamma. \end{cases} \quad (11)$$

In the above formula, the $a > 0, b > 0, c > 0$ value is large enough to make $V_i < L(X_i, X_{\text{sun}})$; that is, the growth distance of plant cells under the action of auxin is less than the distance between plant cells and the maximum light intensity before they move.

The direction of V_i is the vector direction from the position of the i cell to the position of the strongest light.

Step 5. Update the plant cell swarm.

Calculate the position X_i^* of each cell in the plant cell group after growth; the calculation formula is

$$X_i^* = X_i + V_i. \quad (12)$$

Calculate the total amount of sunlight P_i^* received by the i cell after its position update in the d space:

$$P_i^* = f(x_{i1}^*, x_{i2}^*, \dots, x_{ik}^*, \dots, x_{id}^*). \quad (13)$$

If $P_i^* > P_i$, then $P_i^* = P_i$, and $X_i = X_i^*$; otherwise, the growth is considered invalid.

When the auxin concentration of the i cell is greater than γ , the cell does not grow. At this point, a new cell is generated according to formula (1), and let $P_i^* = P_i$ and $X_i = X_i^*$.

When all cells are updated, calculate the amount of sunlight at the location of each cell to find the maximum light value sun^* , $\text{sun}^* > \text{sun}$, and $X_{\text{sun}} = X_{\text{sun}^*}$.

4. Optimized Particle Filter Based on Plant Cell Swarm Algorithm (PCSA-PF)

The detailed process of plant cell swarm algorithm is introduced above. Regarding the process of iteration to find the optimal solution among all feasible solutions in PCSA, it is based on the research of the sunny plant growth process. When the sunny plants receive light stimulation, the auxin moves from the backlight side to the strong light side. The auxin concentration on the backlight side is higher than the strong light side, so the plants bend to the strong light side. In this process, the plant cells move toward the maximum light intensity; that is, the feasible solution of the algorithm gradually approaches the optimal solution in the iterative process. Therefore, particle distribution after importance sampling is optimized by PCSA. PCSA optimizes the particle distribution after importance sampling and uses the PCSA's strong optimization ability to distribute the particle set in the high likelihood region, thus alleviating the particle degradation problem. At the same time, the distribution of plant cells is determined by the concentration of auxin. Different concentrations of auxin have different effects on the growth of plant cells, which will enhance the diversity of particles.

In summary, it can be considered to apply the PCSA to the improvement of PF algorithm.

4.1. Particle Weight Is Used as the Light Intensity Value in the PCSA. It can be seen from the above that after initializing the plant cell swarm, the PCSA evaluates the light intensity of the location of each plant cell through the correlation function. In the PF algorithm, the particle weight after importance sampling is used as the standard to evaluate the performance of each particle filter. Therefore, in this paper, the particle weight is taken as the light intensity value in

the PCSA. This step not only completes the PF performance evaluation but also establishes the relationship between the PCSA and the PF. This process is the basis for improving the particle filter of the PCSA.

4.2. Choose the Appropriate Growth Rate Formula. In the PCSA, the growth rate depends on the concentration of auxin. Under different auxin concentrations, the growth rate of plant cells is different, which is manifested as follows: as the auxin increases, the growth rate of plant cells is accelerated, the growth rate of plant cells is inhibited, or plant cells are killed. The PF is improved by the PCSA. For the important sampled particles, the particle set should move to the high-likelihood region under the optimization of the PCSA. Therefore, this article selects the $Q_i \leq \alpha$ section; that is, with the increase of auxin, the growth promotion effect on plant cells gradually increases. Since the particles sampled from the importance function are not distributed near the global optimal value. Improving by this method, the farther the particles are, the more effective the optimization is. In this paper, the plant cell growth formula used in the process of optimizing particle filtering is

$$V_i = a \frac{Q_i}{b} L(X_i, X_{\text{sun}}). \quad (14)$$

The PCSA-PF process is as follows.

Step 1. At $k = 0$, N particles were sampled from the prior distribution, and the weight of each particle was set as $1/N$, the particle set after sampling was set as $\{x_0^i, 1/N\}_{i=1}^N$, and the number of population in the PCSA was set as N .

Step 2. When $k \geq 1$, particles are sampled from the importance density function, the weight of each particle is calculated, and the weight is normalized.

Step 3. PCSA optimized particle distribution:

- (1) The weight of the sampled particles was taken as the light intensity of the plant cell population algorithm to determine the maximum point of the current light intensity, that is, the position of the global optimal value
- (2) The auxin content of each particle was calculated
- (3) Cell growth
- (4) Update the plant cell population to determine whether the iteration is over. If the iteration is over, go to Step 4; otherwise, recalculate the weight of each particle after optimization and return to (1).

Step 4. Particle weights are normalized.

Step 5. The state estimation is realized by formula (5).

5. Experimental Results and Analysis

The hardware environment of the experiment is a desktop computer (IntelCorei5 processor, 4GB memory), and the experiment environment is MATLAB 2016b.

5.1. Analysis of Particle Filter Simulation Experiment. The improved algorithm is compared and analyzed by experiments to verify its filtering performance and whether the improved algorithm can effectively alleviate the problem of particle weight reduction and diversity loss. This experiment is based on a univariate dynamic change filter model. When the particle number is 100, the classical PF algorithm, the gravitational field-based optimization PF algorithm (GFA-PF) [8], and the improved algorithm proposed in this paper (PCSA-PF) are compared.

The system equations and state equations used in PF simulation are

$$\begin{aligned} x(t) &= 0.5x(t-1) + \frac{2.5x(t-1)}{1+x(t-1)^2} + 8 \cos [1.2x(t-1)] + w(t-1), \\ z(t) &= \frac{x(t)^2}{20} + v(t). \end{aligned} \quad (15)$$

This model is a typical nonlinear non-Gaussian system model; $w(t)$, $v(t)$ is a zero mean Gaussian noise.

The performance of the filtering algorithm is expressed by the root mean square error (RMSE), and the formula is

$$\text{RMSE} = \left[\frac{1}{T} \sum_{i=1}^T (x_k - \tilde{x}_k)^2 \right]^{1/2}, \quad (16)$$

where k is the number of iterations, \tilde{x}_k is the state true value, and x_k is the filtering estimate.

K is the filtering estimate.

5.1.1. Algorithm Convergence Analysis. The PCSA-PF is improved on the classic PF algorithm. So the improved algorithm theoretically has similar convergence properties to the classic algorithm. Therefore, the RMSE comparison is performed under the condition of different numbers of particles. The simulation results are shown in Figure 2.

Figure 2 shows that as the number of particles gradually increases, the RMSE of both the PCSA-PF and classic PF algorithm gradually decreases, and when the number of particles reaches 200, the RMSE value tends to stabilize. It indicates that the PCSA-PF and PF algorithm has the same convergence characteristics; that is, as the number of particles continues to increase, and the filtering accuracy of the algorithm gradually stabilizes while improving the algorithm convergence.

5.1.2. Comparison of Filtering Accuracy. In order to verify the filtering accuracy of the PCSA-PF, the sampling points were set to 100 and the sampling period was $T = 100$. Then, the GFA-PF-related parameters are set by reference [8], the iteration number of the PCSA-PF is 8, and the ratio between

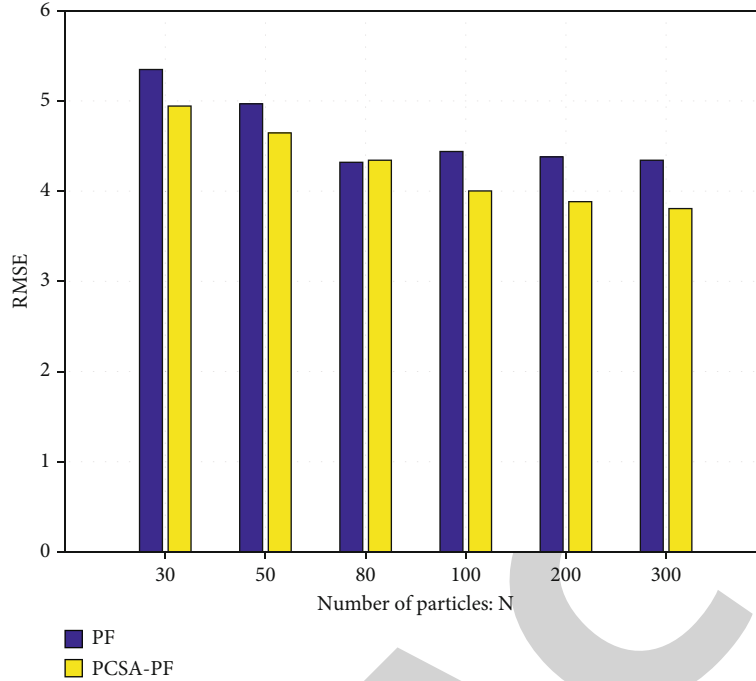


FIGURE 2: Algorithm convergence comparison.

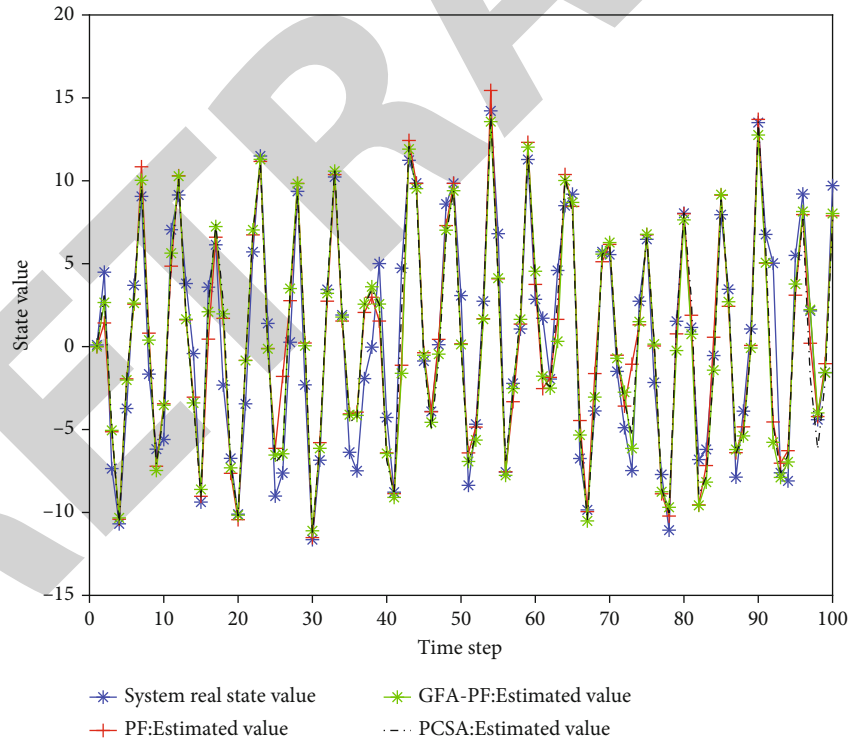


FIGURE 3: System state estimation.

a and b in the above equation (11) was set to 0.002. Set the noise variance of the system state equation as 10 and the noise variance of the system observation equation as 1. Then, compare among the PF, GFA-PF, and PCSA-PF algorithm. The experimental results are shown in Figures 3 and 4.

From Figures 3 and 4, it can be seen that the PCSA-PF proposed in this paper and the GFA-PF proposed in [8] have better filtering accuracy than the classical PF algorithm, and the PCSA-PF has better filtering accuracy than the GFA-PF. This is because after the unique optimization mechanism of

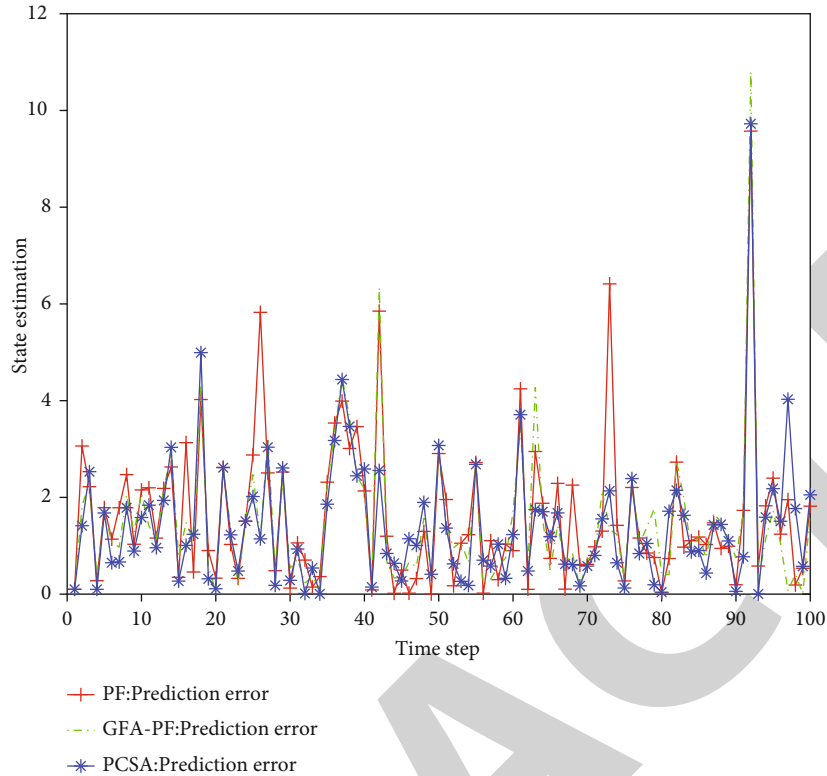


FIGURE 4: Absolute value of estimated error.

PCSA-PF is combined with PF; the particle set will move to the high likelihood area under the effect of the global optimal value, which makes the filtering accuracy improve; and the PCSA has a more obvious promotion effect on particles farther from the global optimal value, which is also the reason for the improved filtering accuracy.

From Tables 1 and 2, it can be seen that under the same number of particles, the filtering accuracy of GPA-PF and PCSA-PF is higher, and the PCSA-PF has higher filtering accuracy when there is less number of particles $N = 50$. It is proved that the PCSA and PF algorithm are efficiently combined. The PCSA plays a vital role in optimizing the particles after the importance sampling and effectively optimizes the distribution of the particles to make them concentrated in the high likelihood region. The PCSA-PF achieves higher filtering accuracy through a smaller number of particles in a strong noisy environment. In addition, compared with the GFA-PF algorithm, the PCSA-PF algorithm efficiency is high; the reason for this phenomenon is that the proposed algorithm's time complexity is $O(N)$, namely, complete particle through a loop optimization, the GFA-PF algorithm firstly by the global optimal value to attract the operations of particles, and then throw, partly to increase the time complexity of the algorithm.

5.1.3. Spatial Distribution Characteristics of Particle Samples.

For the evaluation of the degradation degree of particle weights, adopt effective particle number on the measurement. The experiment set the particle number is 100. In order to avoid contingency, 10 experiments were compared.

TABLE 1: Comparison of mean square error of three algorithms.

Algorithm	$N = 30$	$N = 50$	$N = 80$	$N = 100$
PF	6.565690	5.201521	4.724141	4.442274
GFA-PF	5.069986	4.389173	4.183473	4.005711
PCSA-PF	4.820985	4.116689	4.062849	3.838158

TABLE 2: Comparison of running time of three algorithms.

Algorithm	$N = 30$	$N = 50$	$N = 80$	$N = 100$
PF time/s	0.002856	0.003664	0.004355	0.004355
GFA-PF time/s	0.006497	0.009463	0.014663	0.016842
PCSA-PF time/s	0.007014	0.009079	0.011923	0.014418

According to the Table 3, we can find that the PCSA-PF has greatly improved the degradation degree of particle weights compared with the classic PF. Therefore, it can be concluded that the PCSA-PF effectively solves the problem of particle weight degradation.

In order to verify the impact of PCSA-PF on the diversity of particles, when the number of particles is 80, take $t = 10$, $t = 40$, and $t = 75$, that is, compare the results of three experiments in the initial, middle, and final stages of the algorithm. The experimental results are shown in Figure 5.

From Figure 5, it can be seen that the particles are concentrated in the high likelihood area at the end of the PF algorithm, which loses the diversity of particles to a certain

TABLE 3: Comparison of effective particle number.

Time	1	2	3	4	5	6	7	8	9	10
PF	51.44	51.69	51.19	51.76	51.78	51.07	51.56	51.81	51.38	51.64
PCSA-PF	71.05	71.13	70.74	70.73	70.63	70.40	70.23	70.30	70.95	70.30

extent, thus leading to a decrease in filtering accuracy. In the later stage of the PCSA-PF, while most of the particles are concentrated in the high-likelihood area, some particles are distributed around the true value of the particles. This shows that PCSA-PF has improved the diversity of particles to a certain extent which is also the reason of PCSA-PF precision promotion.

5.2. Experiment of Robot Localization and Mapping. Experiments are performed in order to verify the effectiveness of the algorithm proposed in this paper, thus, comparing the classic FastSLAM2.0, GFA-FastSLAM2.0, and the PCSA-FastSLAM2.0.

Firstly, establish the mobile robot simulation model. The model is a mobile robot motion model:

$$\begin{bmatrix} x_k^v \\ y_k^v \\ \Phi_k^v \end{bmatrix} = \begin{bmatrix} x_{k-1}^v + \Delta T \cdot \cos(\Phi_{k-1}^v + \alpha_k) \\ y_{k-1}^v + \Delta T \cdot \sin(\Phi_{k-1}^v + \alpha_k) \\ \Phi_{k-1}^v + \frac{(\Delta T \cdot v_k \cdot \sin(\alpha_k))}{D} \end{bmatrix} + \begin{bmatrix} v_x \\ v_y \\ v_\phi \end{bmatrix}. \quad (17)$$

In the formula, (x_k^v, y_k^v) represents the position posture state of the robot in the two-dimensional environment at the k moment; Φ_k^v represents the heading angle, and the value range is $[-180^\circ, 180^\circ]$; v_k represents the movement speed of the robot; α_k represents its steering angle; α_k is the robot odometer sampling time; v_x, v_y, v_ϕ is the noise during the robot movement; and D is the distance between the drive shafts. The mobile robot observation model is

$$\begin{bmatrix} r_k \\ \theta_k \end{bmatrix} = \begin{bmatrix} \sqrt{(x_i - x_k^v)^2 + (y_i - y_k^v)^2} \\ \arctan \frac{y_i - y_k^v}{x_i - x_k^v} - \Phi_k^v \end{bmatrix} + \omega_k. \quad (18)$$

In the formula, r_k, θ_k , respectively, represents the distance between the detected environmental feature and the mobile robot, and the angle of movement direction; ω_k is the observation noise.

First, simulate the robot localization and mapping; establish the working environment and mobile robot motion parameters and noise parameters, as shown in Figure 6 and in Table 4; set the 17 heading points and 35 road marking points; and the mobile robot movement range is 100 m \times 80m. The mobile robot starts from the coordinate origin (red point in Figure 6) and moves counterclockwise by the movement command obtained through wireless communication, the green * represents the road marking point, red* represents the heading point, and cyan line represents the prescribed robot path.

Figures 7–9, respectively, show the robot localization mapping effects of the three algorithms. The robot is set to move for two circles and compare the predicted trajectory with the actual trajectory during the second circle. Compared to the classic FastSLAM2.0 algorithm, the predicted trajectory of GFA-FastSLAM2.0 and PCSA-FastSLAM2.0 overlaps with the real trajectory in many parts, and the PCSA-FastSLAM2.0 has the most overlapped part; that is, the PCSA-FastSLAM2.0 algorithm has the good prediction effect. The reason is that compared to the particle adjustment way of GFA-FastSLAM2.0 after importance sampling, the PCSA-FastSLAM2.0 has the higher degree of attraction for particles far from the global optimal value; thus, the PCSA-FastSLAM2.0 for the robot localization accuracy has been significantly improved. And it is through the prediction of road signs in Figures 8–10 (the red dot represents the predicted location of the road signs). In Figure 8, the predicted road sign accuracy of FastSLAM2.0 gradually decreases as the running time increases. This is due to the degradation of particle weights and loss of particle diversity, resulting in a decrease in localization accuracy, which in turn affects the estimation of road signs. As illustrated in the PCSA-FastSLAM2.0 shown in Figure 10, it can be seen that the predicted value of the road sign and the actual value have a higher degree of coincidence, which indicates that the PCSA-FastSLAM2.0 improves the degradation of particle weight, loss of particle diversity, and other problems and further illustrates the superiority of the PCSA-FastSLAM2.0.

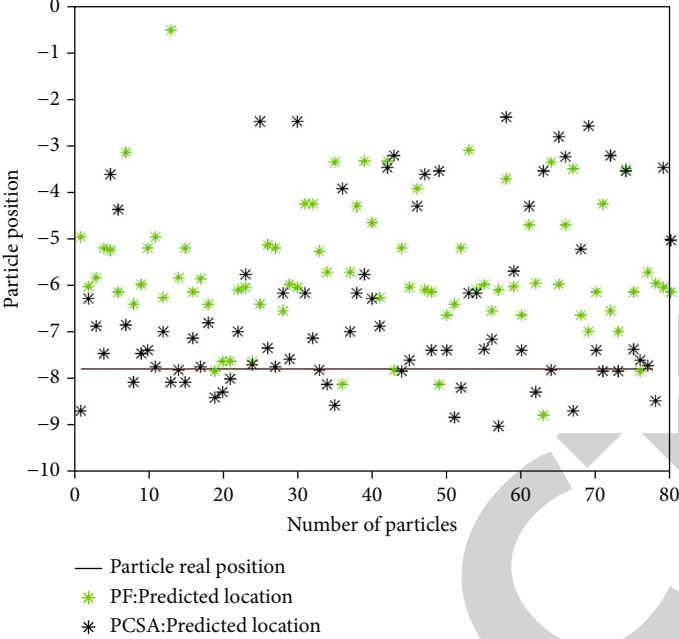
In order to further verify the optimization effect of the PCSA-FastSLAM2.0 on robot localization and mapping, at 8000 sampling moments, respectively, compare the Euclidean distance between the predicted value and the estimated value of the three algorithms, namely, the localization error. The formula is:

$$\rho = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}. \quad (19)$$

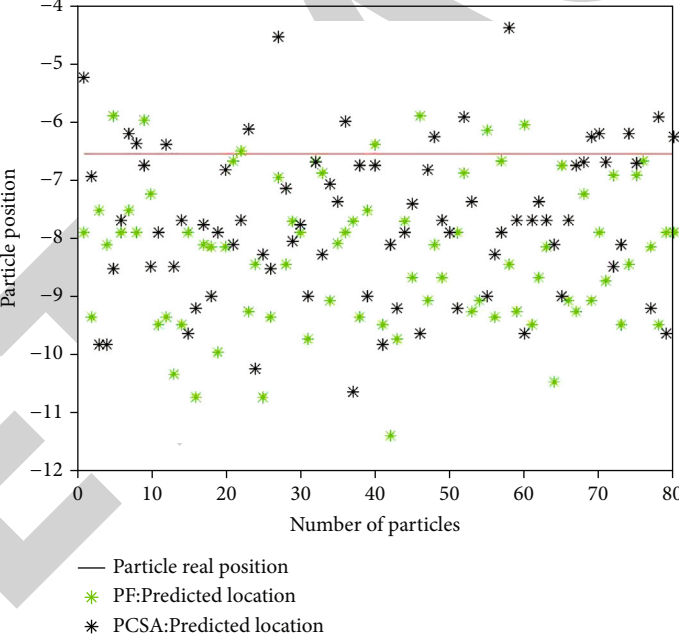
In the formula, $(x_1, y_1), (x_2, y_2)$ represents the coordinates of the predicted position and actual position.

The comparison of the robot localization accuracy errors of the three algorithms is shown in Figure 10.

From the above Figure 10, it can be seen the PCSA-FastSLAM2.0 has the highest localization accuracy and is relatively stable compared to FastSLAM2.0 and GFA-FastSLAM2.0. While the localization accuracy of the classic FastSLAM2.0 increases, the running time gradually increases, and the localization accuracy error also gradually increases. This is because in the later iterations of the algorithm, the particles are severely degraded and the diversity is lost, resulting in lower localization accuracy. The PCSA-FastSLAM2.0 solves this problem effectively and increases



(a) Particle distribution ($t = 10, N = 80$)



(b) Particle distribution ($t = 40, N = 80$)

FIGURE 5: Continued.

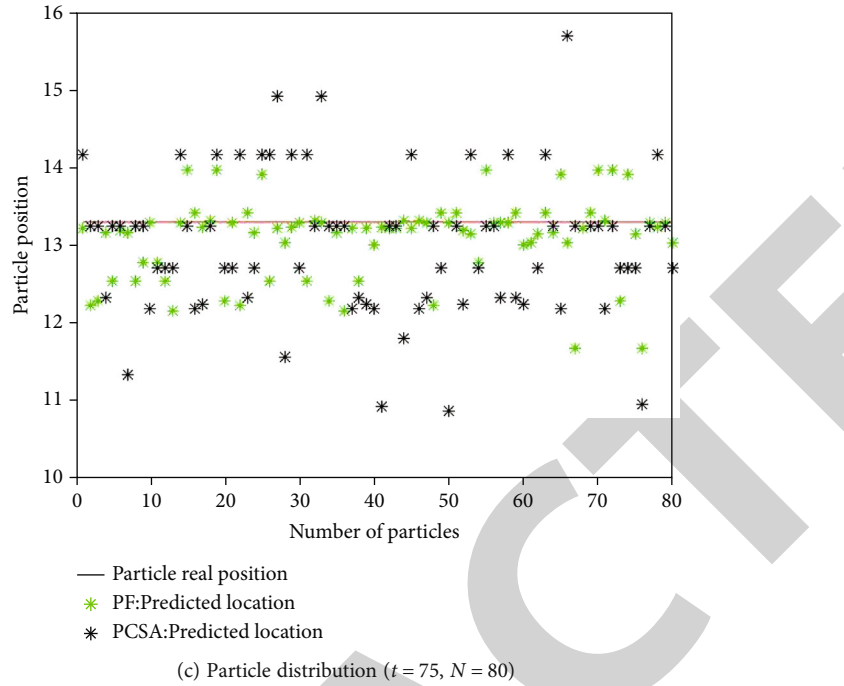


FIGURE 5: Particle diversity experiment.

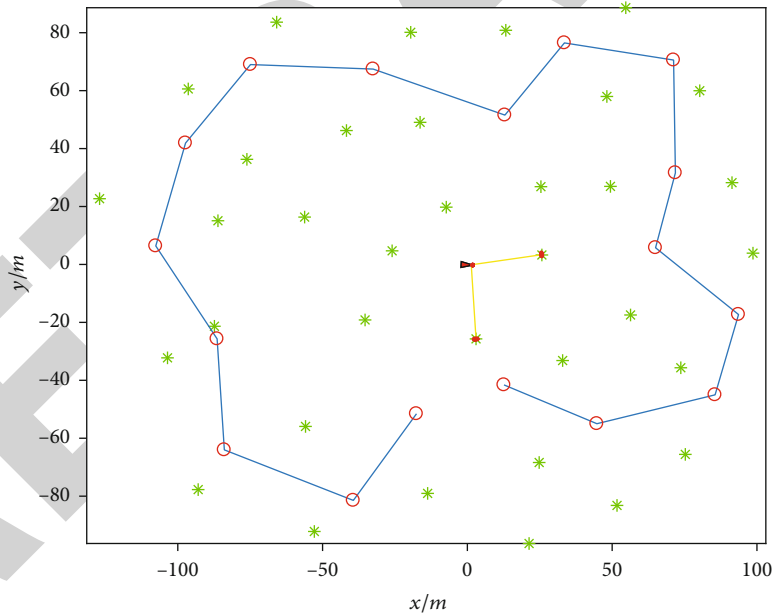


FIGURE 6: Simulation environment.

TABLE 4: Mobile robot motion parameters and noise parameters.

Motion parameters	Value	Noise parameters	Value
Speed	3 m/s	Motion noise	0.3 m/s
Maximum steering angle	10°		1.5°
Maximum steering angle speed	15°/s	Observation noise	0.1 m/s
Wheel spacing	4 m		1°
Sampling interval	0.025 s		

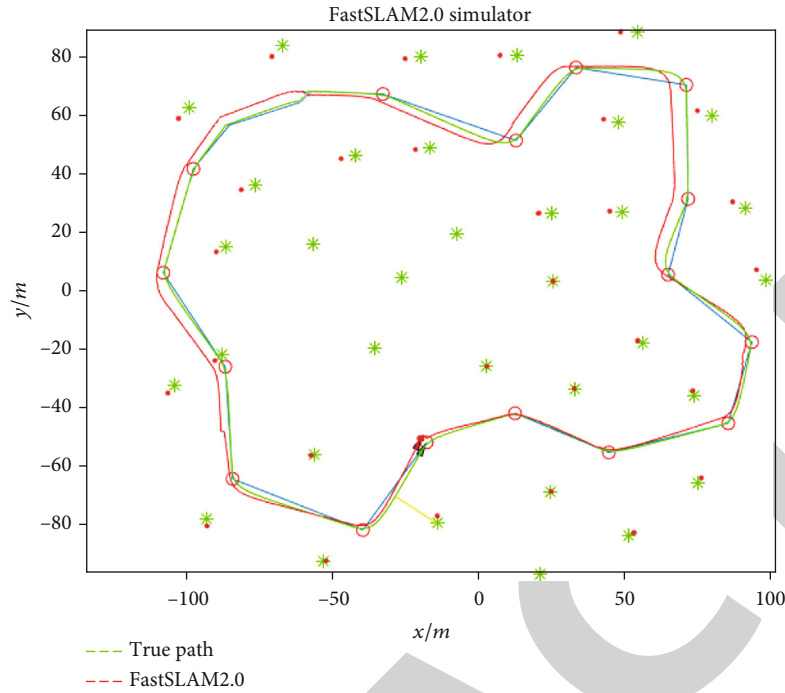


FIGURE 7: FastSLAM2.0 simulation result.

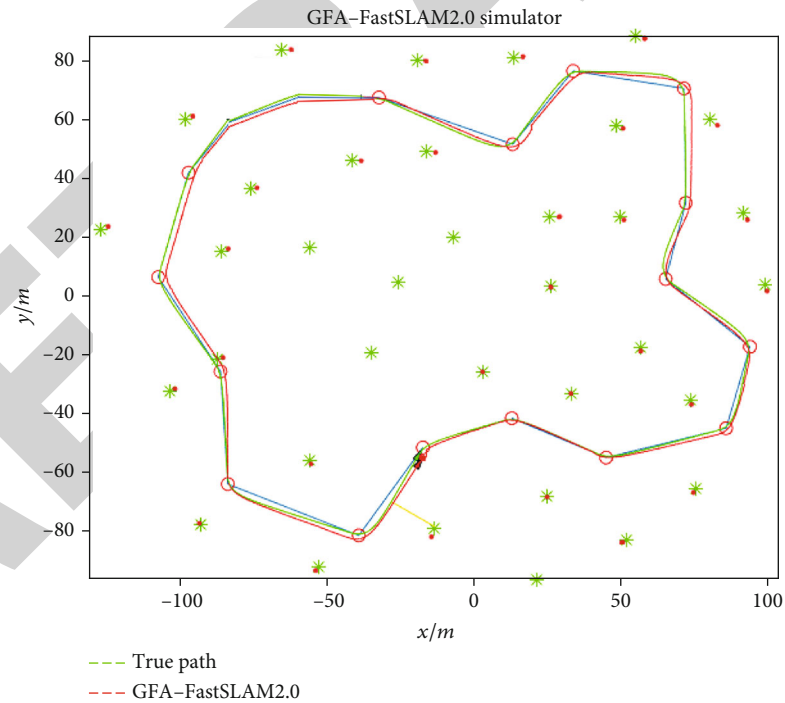


FIGURE 8: GFA-FastSLAM2.0 simulation result.

the localization accuracy. For the GFA-FastSLAM2.0, its positioning accuracy is lower than the PCSA-FastSLAM2.0, but better than the classic FastSLAM2.0. This is because the GFA has a certain degree of optimization effect on importance sampled particles, and GFA-PF has improved the diversity of particles to a certain extent. Therefore, the

GFA-FastSLAM2.0 has a certain degree of improvement on the robot localization accuracy. But compared with the idea of PCSA-FastSLAM2.0, the PCSA has a strong degree of optimization for particles far from the global optimal value; thus, the PCSA-FastSLAM2.0 is better than GFA-FastSLAM2.0 in the robot localization accuracy improvement.

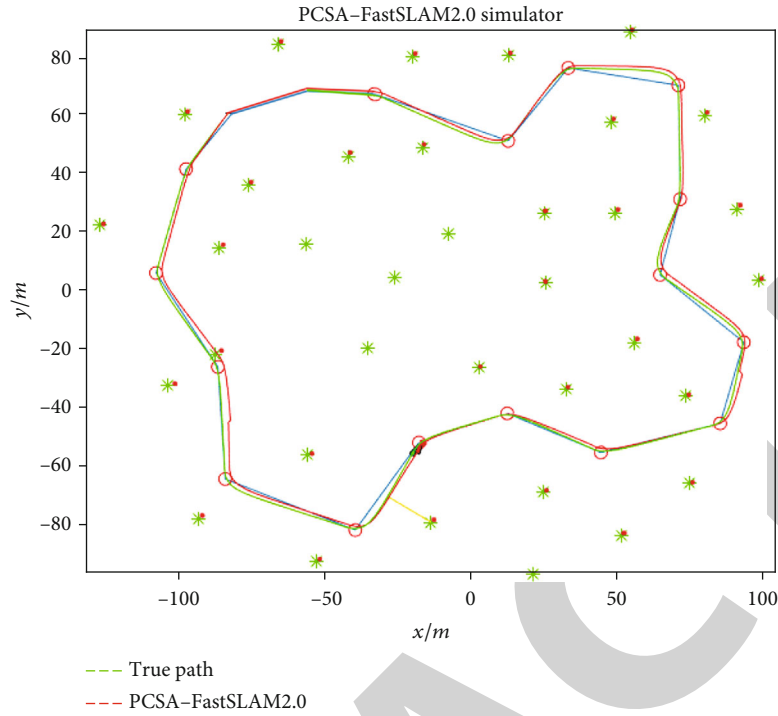


FIGURE 9: PCSA-FastSLAM2.0 simulation result.

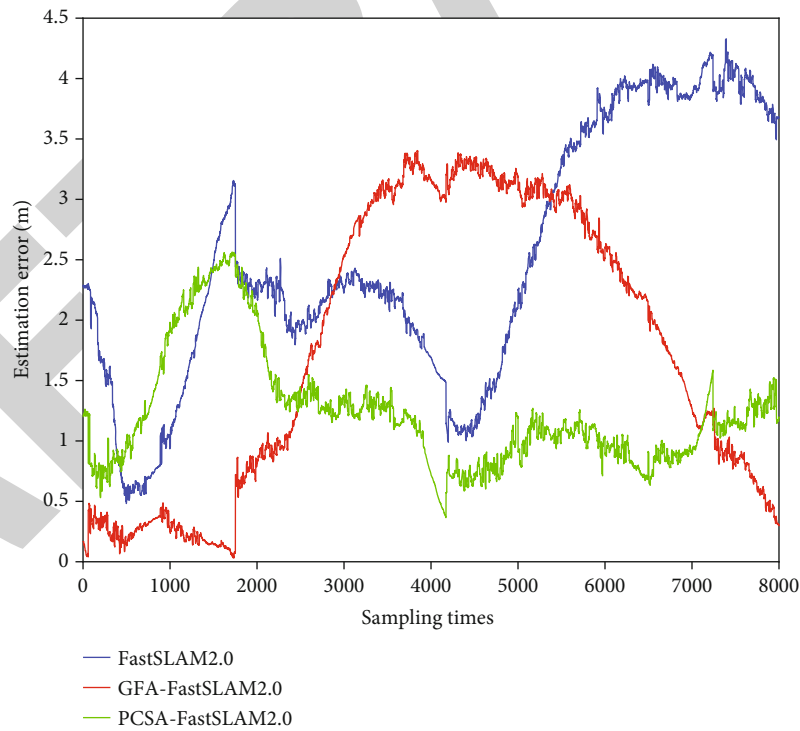


FIGURE 10: Comparison chart of robot localization accuracy error.

In order to verify the improvement effect of the improved algorithm on the robot localization accuracy, compare the mean error and variance of the localization accuracy of the three algorithms, as shown in Table 5.

In the above table, the PCSA-FastSLAM2.0 has the lowest mean error value and the smallest variance of positioning accuracy, which indicates that the PCSA-FastSLAM2.0 improves the localization accuracy, also improving the

TABLE 5: Comparison of mean error and variance of localization accuracy of three algorithms.

Algorithm	Mean localization accuracy error (m)	Variance of localization accuracy error	Improvement rate of localization accuracy (%)	Improvement rate of stability (%)
FastSLAM2.0	2.5479	1.0754		
GFA-FastSLAM2.0	1.6728	1.3949	35%	-29%
PCSA-FastSLAM2.0	1.2309	0.2101	62%	81%

TABLE 6: Comparison of x -axis, y -axis and road sign RMSE estimation of three algorithms.

Algorithm	RMSE of x -axis (m)	RMSE of y -axis (m)	RMSE of road sign estimation (m)
FastSLAM2.0	1.9634	2.2563	2.8910
GFA-FastSLAM2.0	1.3695	0.9363	1.6485
PCSA-FastSLAM2.0	0.8036	0.7235	1.0415

stability at the same time, while in the strategy of adjusting the sampled particles in GFA-FastSLAM2.0, it makes the sampled particles centrally distributed near the global optimal value, because of the unique strategy of simulating the gravitational field; that is, the central dust attracts and repels the surrounding dust and adjusts the particle distribution. This is the reason why the accuracy of the GFA-FastSLAM2.0 is improved. At the same time, the stability of the algorithm decreases due to the unique optimization mechanism.

In order to verify the degree of improvement in the accuracy of robot localization and mapping, respectively, compare the RMSE of the x -axis, y -axis, and road signs, as shown in Table 6.

From Table 6, it can be seen that the PCSA-FastSLAM2.0 is better than FastSLAM2.0 and GFA-FastSLAM2.0 algorithms in the x -axis, y -axis, and road sign estimation. This is because the improved algorithm efficiently optimizes the particles after importance sampling and produces the best effect, and this also shows that the optimization of particles in PCSA-FastSLAM2.0 is better than GFA-FastSLAM2.0. The main reason is that the unique particle optimization method in PCSA-FastSLAM2.0 simulates the stimulation of auxin on plant cells, and the more distant the particle is from the global optimal value, the more obvious the optimization effect is. Therefore, PCSA-FastSLAM2.0 has the highest prediction accuracy.

6. Conclusion

This paper proposes an improved particle filter algorithm based on the plant cell swarm algorithm (PCSA-PF) and applies it to the robot autonomous mobile mapping. By improving the particle distribution after importance sampling, this algorithm guides the particle set to be distributed in the high-likelihood area, and the farther the particle is from the global optimal value, the stronger the optimization effect of the algorithm. In the experimental verification of particle filtering, the improved algorithm effectively allevi-

ates the problem of particle weight degradation and improves the particle diversity, so as to improve the filtering accuracy and achieve a higher filtering accuracy through a small number of particles in the strong noise scene. In the experiment of SLAM, the robot receives the wireless communication command and moves; the localization accuracy and the stability greatly are improved. However, in the simulator used in this paper, the environment of the robot is a static environment, which is different from the actual working environment of the robot. In the future, the feasibility of the algorithm proposed in this paper will be further verified through robot experiments.

Data Availability

The datasets used and analyzed during the current study are available from the corresponding author on reasonable request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is partially supported by the National Natural Science Foundation of China (No. 61302133), Key Research and Development Program of Shaanxi Province of China (No. 2021GY-338), and Xi'an Beilin District Science and Technology plan project (No. GX1930).

References

- [1] S. S. Moghaddasi and N. Faraji, "A hybrid algorithm based on particle filter and genetic algorithm for target tracking," *Expert Systems with Applications*, vol. 147, article 113188, 2020.
- [2] J. A. Nunez, D. I. Singham, and M. P. Atkinson, "A particle filter approach to estimating target location using Brownian bridges," *Journal of the Operational Research Society*, vol. 71, no. 4, pp. 589–605, 2020.
- [3] A. Stentz, D. Fox, M. Montemerlo et al., "FastSLAM: a factored solution to the Simultaneous localization and mapping problem with unknown data association," in *Proceedings of AAAI National Conference on Artificial Intelligence*, pp. 240–248, Edmonton, AB, USA, 2002.
- [4] D. Jiahui, X. Pengcheng, and L. Xiaobo, "FastSLAM algorithm of second-order central difference particle filter," *Control Theory and Applications*, vol. 35, no. 9, pp. 1382–1390, 2018.
- [5] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "FastSLAM2.0: an improved particle filtering algorithm for simultaneous localization and mapping that provably converges," in

- Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, pp. 1151–1156, Acapulco, Mexico, 2003.
- [6] M. Micev, M. Čalasan, Z. M. Ali, H. M. Hasanien, and S. H. E. Abdel Aleem, “Optimal design of automatic voltage regulation controller using hybrid simulated annealing - manta ray foraging optimization algorithm,” *Ain Shams Engineering Journal*, vol. 12, no. 1, pp. 641–657, 2021.
- [7] C. Shiming, X. Juan, L. Haiying, and N. Sen, “Particle filter algorithm based on gravitational field,” *Control and Decision*, vol. 32, no. 4, pp. 709–714, 2017.
- [8] C. Shiming, L. Junkai, and X. Juan, “Unscented FastSLAM2.0 algorithm based on gravitational field optimization,” *Control Theory and Applications*, vol. 35, no. 8, pp. 1186–1193, 2018.
- [9] C. Haoyang, Z. Yu, Z. Kun, F. Hu, and Y. Xu, “Study on accuracy prediction of SLAM algorithm based on bionic algorithm and improved particle filter,” *Control and Decision*, vol. 36, no. 1, pp. 166–172, 2021.
- [10] L. Cheng, X. Jin, H. Xiao, X. S. Gao, Z. Zhang, and H. Zhang, “Research on robot FastSLAM based on particle swarm optimization algorithm,” *Journal of Shenyang Normal University*, vol. 37, no. 4, pp. 335–339, 2019.
- [11] S. Pan, H. Liu, N. Li, and Y. Nie, “FastSLAM algorithm based on quantum PSO immune particle filter,” *Computer Simulation*, vol. 35, no. 8, pp. 202–205, 2018.
- [12] C. Jie, Z. Wang, and L. Wei, “Anti-occlusion particle filter target tracking method based on chicken swarm optimization,” *Microelectronics and Computer*, vol. 36, no. 6, pp. 40–44, 2019.
- [13] C. Shiming, Y. Junfeng, C. Xiaoling, and F. Huajing, “FastSLAM2.0 algorithm based on electromagnetic-like mechanism optimization,” *Control Theory and Applications*, vol. 32, no. 1, pp. 127–132, 2015.
- [14] Z. Tingjun, G. Yifeng, and H. Limin, “Improved resampling mobile robot SLAM algorithm,” *Computer Engineering and Design*, vol. 40, no. 11, pp. 3276–3281, 2019.
- [15] L. Zhaoxian and Y. Wenpeng, “A new swarm intelligence algorithm: plant cell swarm algorithm,” *Science Technology and Engineering*, vol. 19, no. 4, pp. 166–172, 2019.