

Research Article

A Robot SLAM Improved by Quantum-Behaved Particles Swarm Optimization

Tao Zuo ^{1,2}, Huasong Min ¹, Quan Tang,¹ and Qiang Tao¹

¹College of Information Science and Engineering, Wuhan University of Science and Technology, Wuhan 430081, China

²Engineering Research Center for Metallurgical Automation and Detecting Technology of Ministry of Education, Wuhan University of Science and Technology, Wuhan 430081, China

Correspondence should be addressed to Tao Zuo; zuomu666@163.com

Received 23 January 2018; Revised 28 March 2018; Accepted 4 April 2018; Published 10 May 2018

Academic Editor: Oscar Reinoso

Copyright © 2018 Tao Zuo 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.

We propose a new SLAM method based on fast simultaneous localization and mapping (FastSLAM). The technique presented uses an improved quantum-behaved particles swarm optimization (QPSO) to improve the proposal distribution of particles and optimize the estimated particles. This method makes the sampled particles closer to the true pose of the robot and improves the estimation accuracy of robot poses and landmarks. In the QPSO algorithm, the Gaussian disturbance is added to increase the diversity of the particles. By using this technique the premature convergence of particles swarm is overcome. In the resample step, the threshold value is used to evaluate the particle diversity. When the particle diversity is below the threshold value, the linear optimization is used to produce new sample particles, which increases the particle diversity and eliminates the loss of diversity. Simulations and experiments show that the proposed approach improves the accuracy of SLAM. The accuracy of estimated poses and landmarks with the proposed method is better than that with the traditional SLAM method.

1. Introduction

Simultaneous localization and mapping (SLAM) is an active research area in mobile robotics [1, 2]. The task of SLAM is to build a map while estimating the robot pose relative to the map. In SLAM, the robot paths and maps are both unknown and the errors of robot path estimates correlate the errors in the maps.

The Extended Kalman Filter based SLAM (EKF-SLAM) approach was the most successful method in the early stage [3, 4]. However, the errors of state estimation are large when the EKF-SLAM is used in nonlinear systems, and the calculated amount increases exponentially along with the enlargement of maps.

Michael Montemerlo proposed FastSLAM to solve the SLAM problem, which had been proved to be an effective method [5, 6]. In FastSLAM, the particle filter is used to estimate the mobile robot poses and the EKF is used to estimate the features.

However, there exist drawbacks on FastSLAM. Several studies have demonstrated that the RBPF (Rao-Blackwellized

particle filter) used in FastSLAM present particle depletion problems. Minority high-weighted particles are duplicated many times after frequent resamples, which will result in loss of diversity and decrease of estimation accuracy. Moreover, the proposal distribution of sample particles influences the estimation accuracy in FastSLAM. It can improve position accuracy to a certain extent to increase the number of sample particles, but it will increase the computing time.

Some researches tried to improve the performance of FastSLAM and solve the sample impoverishment problem. Grisetti et al. [7] proposed an approach to compute an accurate proposal distribution taking into account not only the movement of the robot but also the most recent observation. Furthermore, resampling operations were carried out selectively which seriously reduced the problem of particle depletion. Lee et al. [8] proposed an approach to prevent the degeneracy by particle cooperation. The PSO is performed to update the robot position by means of particle cooperation. Zhao et al. [9] and Baifan et al. [10] applied PSO to driver all the particles to the regions with high likelihoods in

the nonlinear area. The number of particles was decreased and the computational complexity was reduced. Jia-cheng et al. [11] used PSO to acquire the proposal distribution of mobile robot's poses. The new distribution of particles maintained the diversity of mobile robot in a particular environment. Erliang et al. [12] proposed a RBPF SLAM algorithm based on PSO. Particles in high likelihood sampling set moved to the region with the maximum value of posterior probability distribution; meanwhile the algorithm maintained the multiplicity of the low likelihood particles. Kim et al. [13] provided a robust new algorithm based on the scaled unscented transformation called unscented FastSLAM (UFastSLAM). This approach improved the filter consistency and state estimation accuracy. Hanvangi et al. [14] proposed a square root unscented FastSLAM with improved proposal distribution and resampling. PSO was used to optimize the proposal distribution. Liu et al. [15] proposed a FastSLAM method based on PSO and unscented particle filter. The number of particles was seriously reduced and the computational time decreased.

These researches could improve the performance of FastSLAM to some extent, but the loss of diversity of particles was not solved. The PSO used in these researches could optimize the proposal distribution of particles, but PSO also had the premature convergence problem.

To overcome these problems, in this paper, we propose a new FastSLAM approach improved by Quantum-behaved particles swarm optimization (QPSO-FastSLAM). Compared with the PSO [16, 17], QPSO can search the globally optimal solution in the whole feasible solution spaces and has better global searching ability [18–20]. When QPSO is used to optimize the proposal distribution of the particles, the estimated particle pose is closer to the true pose. To overcome the depletion problem of particles, the threshold value is used to judge the particle diversity. When the particle diversity is below the threshold value, the linear optimization is used to produce new sample particles, which increase the particle diversity and eliminate the loss of diversity.

The rest of this paper is organized as follows. Section 2 describes the SLAM problem and the framework of FastSLAM. Section 3 presents the QPSO algorithm. Section 4 presents the framework of QPSO-FastSLAM. Section 5 gives the evaluation of diversity and Gaussian random disturbance approach for QPSO. Section 6 presents the linear optimization based resample method. Section 7 presents the steps of the proposed QPSO-FastSLAM approach. Section 8 shows the simulation and experiment results of the proposed method. Finally, Section 9 gives conclusions.

2. FastSLAM

2.1. FastSLAM Framework. In SLAM both the trajectory of the robot and the positions of landmarks are estimated simultaneously without any a priori knowledge of environments. The true positions of robots and landmarks are never known or measured directly. The estimations of the map and the trajectory rely on the observations between true robots and true landmarks.

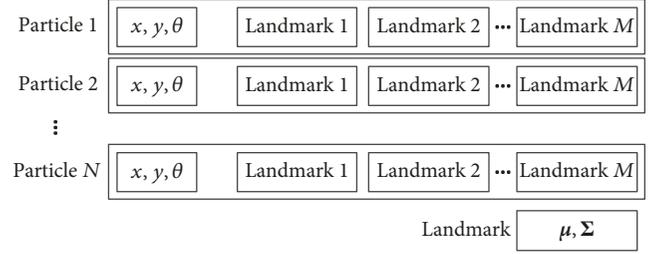


FIGURE 1: The structure of particles.

The FastSLAM proposed by Michael Montemerlo has now developed to FastSLAM 2.0 version. In FastSLAM, the state estimations of landmarks are considered mutually independent when the robot path is known. Then the estimations of poses and maps are separated. The robot poses are estimated by Rao-Blackwellized particle filters and the maps are estimated by EKF [1, 2, 21, 22]. The joint posterior probability distribution of FastSLAM is decomposed into

$$\begin{aligned}
 p(\Theta, s_{1:t} | z_{1:t}, u_{1:t}, c_{1:t}) \\
 = p(s_{1:t} | z_{1:t}, u_{1:t}, c_{1:t}) \prod_{n=1}^N p(\theta_n | s_{1:t}, z_{1:t}, u_{1:t}, c_{1:t}), \quad (1)
 \end{aligned}$$

where $s_{1:t}$ is the robot pose, θ_n denotes the n th landmark in environmental maps, $u_{1:t}$ is the control input, and $c_{1:t}$ is the data association; that is, the correspondence between known landmarks in the maps and observed information. N is the number of landmarks in each of the maps of the environment. In FastSLAM, the first part of (1) is estimated by the Rao-Blackwellized particle filter to get robot poses. Each particle carries pose information and saves a map, and each map is composed of N independent landmark features. In the second part of (1), the environmental maps are estimated by EKF with each landmark feature.

The structure of each particle in FastSLAM is shown in Figure 1. Each particle consists of its pose (x, y, θ) and M landmark estimates described by the mean μ and covariance Σ .

The i th particle $S_t^{[i]}$ at time t is $S_t^{[i]} = \{s_{1:t}^{[i]}, \mu_1^{[i]}, \Sigma_1^{[i]}, \dots, \mu_n^{[i]}, \Sigma_n^{[i]}\}$, where i is the sequence number of the particles, $s_{1:t}^{[i]}$ is the estimated pose of the i th particle, and $\mu_n^{[i]}$ and $\Sigma_n^{[i]}$ are the mean value and variance of the n th estimated landmark, respectively.

In FastSLAM2.0, the posterior probability distribution can only be sampled from a proposal distribution function which is approximate to the posterior probability. The following proposal distribution function is selected:

$$\begin{aligned}
 q(s_{1:t}^{[i]} | z_{1:t}, u_{1:t}, s_0) \\
 = q(s_t^{[i]} | s_{t-1}, z_t, u_t) q(s_{1:t-1}^{[i]} | z_{1:t-1}, u_{1:t-1}, s_0), \quad (2)
 \end{aligned}$$

where z_t is the observed information and u_t is the control input.

FastSLAM uses sequential important resampling (SIS) based particle filters. To overcome particle degeneracy, the

resampling method is used. The number of valid particles is used to judge whether the particle degeneracy occurs. If the number of valid particles is less than N_{eff} , the resampling should be carried out.

$$N_{\text{eff}} = \frac{1}{\sum_{j=1}^n (w_t^{[j]})^2}, \quad (3)$$

where $w_t^{[i]}$ is the normalized weight.

In FastSLAM2.0, EKF is used to calculate the proposal distribution of particles, which consist of two steps [4].

In estimating step, the current pose $s_{t|t-1}^{[i]} = g(s_{t-1}^{[i]}, u_t)$ is estimated by motion model, where $s_{t-1}^{[i]}$ is the pose at time $t-1$, and $g(\cdot)$ is the motion model.

In update step, the observed landmark information is matched with the known landmark information in maps. If there is not any matching information, the observed landmarks are added to the maps; otherwise, the poses are updated with the observed information to make the estimated poses more accurate.

$$\hat{z}_t^{[i]} = h(\mu_{c,t-1}^{[i]}, s_{t|t-1}^{[i]}), \quad (4)$$

where $h(\cdot)$ is the observation model, $\hat{z}_t^{[i]}$ is the estimation of the c th observed landmark from the observation model, and $\mu_{c,t-1}^{[i]}$ is the mean value of the c th landmark in maps at time $t-1$.

When the estimated landmarks are obtained, the estimated pose $s_t^{[i]}$ and variance $P_t^{[i]}$ are updated.

$$\begin{aligned} s_t^{[i]} &= s_{t|t-1}^{[i]} + K_t^{[i]} (z_t - \hat{z}_t^{[i]}), \\ P_t^{[i]} &= (I - K_t^{[i]} H_s^i) P_{t|t-1}^{[i]}, \end{aligned} \quad (5)$$

where $K_t^{[i]}$ is the Kalman gain and H_s^i is the Jacobian matrix. When the observed landmarks are matching with the landmarks in the known maps, the landmarks are updated by recursive procedures.

$$\hat{z}_{c,t}^{[i]} = h(\mu_{c,t-1}^{[i]}, s_t^{[i]}), \quad (6)$$

where $\hat{z}_{c,t}^{[i]}$ is the estimated observation of the landmark c from the observation equations.

2.2. Background of Particle Depletion Problem. Resample can solve the degeneration problem of particle filters, but it will result in the particle depletion phenomenon. In resample step, the high-weighted particles are duplicated and the low-weighted particles are discarded. As time passes, only high-weighted particles survive and are duplicated many times, and the low-weighted particles disappear together with their information about robot poses and features. The diversity of particles decreases; thus the estimated accuracy of robot poses and features decreases.

In particle filters based on SIS (Sequential Importance Sampling), the particle depletion problem is inevitable. Increasing the number of particles can alleviate the particle depletion problem to a certain extent, but it will increase

the computational cost. Furthermore, due to the lack of knowledge about the robot poses, it is difficult to maintain the enough effective particles to guarantee the estimated accuracy of robot poses.

3. Quantum-Behaved Particles Swarm Optimization

PSO is an effective global optimization algorithm. PSO imitates the process of birds foraging to search the global optimum. PSO algorithm has a powerful global search ability and fast speed to find exact or approximate solutions for optimization and search problems.

However, PSO still has the following major drawbacks: (1) When iterations are infinite, the global convergence cannot be realized. (2) The evolution equation of PSO is composed of velocities and positions, which may result in randomness and low intelligence. (3) The convergence performance of PSO relies on the setting of velocity upper limit and parameters. (4) The velocity of particles in PSO is limited, so the search area can not cover the whole feasible space and the global convergence can not be guaranteed.

The QPSO algorithm proposed by Sun et al. [18] brings the quantum behaviors into PSO algorithm. The QPSO uses the probability amplitude coding of quantum bits to increase the search ability for space solutions. The movement based on quantum revolving door makes the search more meticulous. The evolution and movement of the particles in QPSO are expressed as follows [18].

The mean best position m_{best} is defined as

$$\begin{aligned} m_{\text{best}}(t+1) &= \frac{1}{M} \sum_{i=1}^M P_i(t) \\ &= \left[\frac{1}{M} \sum_{i=1}^M P_{i1}(t), \frac{1}{M} \sum_{i=1}^M P_{i2}(t), \dots, \frac{1}{M} \sum_{i=1}^M P_{iD}(t) \right], \end{aligned} \quad (7)$$

where M is the population size in D -dimensional problem space and $P_i(t)$ is the personal best location of particle i at t th iteration. $P_{id}(t)$ is the d th dimension of vector $P_i(t)$. Vector $P_i(t) = [P_{i1}(t), P_{i2}(t), \dots, P_{iD}(t)]$ is the position with the best fitness found so far for particle i , which is called personal best location.

$$P_{id}(t) = \varphi P_{id}(t) + (1 - \varphi) P_{gd}(t). \quad (8)$$

The position of particle is updated according to the following equation:

$$\begin{aligned} X_{id}(t+1) &= P_{id}(t) + \beta a(t) | m_{\text{best}}(t+1) \\ &\quad - X_{id}(t) | \ln \left[\frac{1}{\mu_{id}(t)} \right]. \end{aligned} \quad (9)$$

Here, φ and $\mu_{id}(t)$ are uniformly distributed random numbers in the interval $[0, 1]$. And

$$\beta = \begin{cases} -1, & \mu_{id}(t) \leq 0.5 \\ 1, & \mu_{id}(t) > 0.5. \end{cases} \quad (10)$$

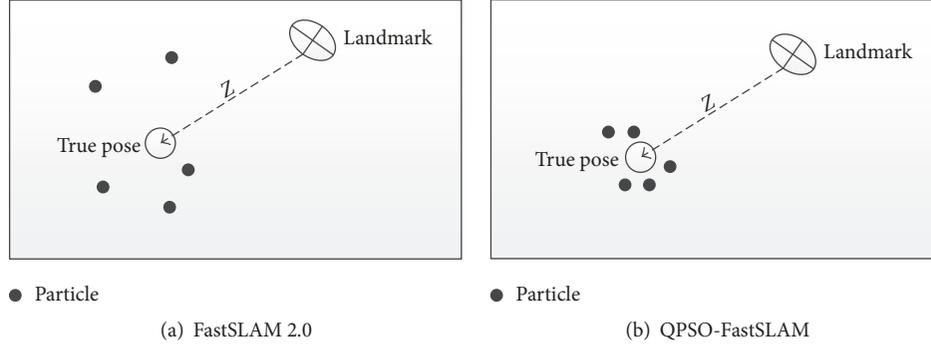


FIGURE 2: Sample comparison of QPSO-FastSLAM and FastSALM 2.0.

$P_g(t)$ is the global best location at t th iteration. Vector $P_g(t) = [P_{g1}(t), P_{g2}(t), \dots, P_{gD}(t)]$ is the best position with the best fitness found by the swam, which is called global best location. $X_i(t)$ is the position of particle i at t th iteration. $X_{id}(t)$ is the d th dimension of the vector $X_i(t)$. $a(t)$ is the Contraction-Expansion Coefficient, of which dynamic changes as follows:

$$a(t) = m - \frac{(m-n)t}{\text{Itermax}}; \quad (11)$$

usually $m = 1$ and $n = 0.5$ and Itermax is the maximum number of iterations. The renewal equations of $P_i(t)$ and $P_g(t)$ are as follows:

$$P_i(t) = \begin{cases} P_i(t), & f_{\text{fitness}}(X_i(t)) < f_{\text{fitness}}(P_i(t)) \\ X_i(t), & f_{\text{fitness}}(X_i(t)) > f_{\text{fitness}}(P_i(t)) \end{cases} \quad (12)$$

$$P_g(t) = \left\{ X_i(t) \mid X_i(t) = \max_{1 \leq i \leq M} (f_{\text{fitness}}(X_i(t))) \right\},$$

where $f_{\text{fitness}}(\cdot)$ is the fitness function.

QPSO has few parameters than PSO and has fast convergent rate and good robustness. Compared with PSO, QPSO has the following advantages:

- (1) QPSO has few parameters to be adjusted than PSO, and the complexity of parameter adjustment decreases.
- (2) QPSO has stronger searching ability than PSO. It can better converge to the global best position and has fast convergent rate.
- (3) QPSO has better robustness than PSO.
- (4) PSO searching is restrained by space constraint. QPSO can search in the whole feasible solution space.

4. FastSLAM Improved by QPSO

This paper uses QPSO to optimize the estimated poses of particles, which makes the estimated positions closer to the true positions. $s_{1:t}^{[i]}$ is the estimated pose of the i th particle, which is obtained from the motion model. We use the newest observed information z_{new} to make the estimated pose

closer to the true pose. z_{pred} is the estimated observed value according to the estimated pose $s_{1:t}^{[i]}$ and the existing map information.

Let the known landmark be (x_m, y_m) and the estimated pose of a particle be (x_s, y_s, θ_s) . Then the estimated observed value is

$$z_{\text{pred}} = \begin{pmatrix} r \\ \theta \end{pmatrix} = \begin{pmatrix} \sqrt{(x_s - x_m)^2 + (y_s - y_m)^2} \\ \arctan\left(\frac{y_m - y_s}{x_m - x_s}\right) - \theta_s \end{pmatrix}, \quad (13)$$

where (r, θ) is the range and azimuth relative to the observer.

By computing the difference between the true observed value and the estimated observed value, we can assess if the estimated pose of particles deviates from the true pose. If the deviation is large, we use the OPSO algorithm to discard the particles with large deviation and search new particles closer to the true pose.

The fitness function is selected as

$$f_{\text{fitness}} = \exp\left[-\frac{1}{2R_k} (z_{\text{new}} - z_{\text{pred}})^T \cdot (z_{\text{new}} - z_{\text{pred}})\right], \quad (14)$$

where R_k is the covariance of observation noises, z_{new} is the newest observed value, and z_{pred} is the predicted observed value.

As is shown in Figure 2, we improve FastSLAM 2.0 algorithm by observed information z and the sample particles are closer to the true position with QPSO. This approach will improve the estimated accuracy of SLAM.

5. Evaluation of Particle Diversity of QPSO

Similar to PSO, QPSO is also easy to fall into the premature convergence. The reason of the premature convergence is that the diversity of particles decreases after lots of iterations, and the local search capability deteriorates in subsequent searches [19].

This paper uses a fitness variance to estimate the degree of premature convergence. Because the position of a particle is corresponding to the fitness value, the same position means the same fitness value. Thus, we can use the fitness variance to estimate the aggregation degree of particle positions. After

repeated iterations, the positions of different particles will become increasing aggregation. The swarm tends to converge and the fitness value of each particle becomes closer and closer. Then the fitness variance becomes smaller and smaller. When the fitness variance is smaller than a threshold value, the particles are considered to be in the state of premature convergence.

The fitness variance ρ^2 is used as the criterion of premature convergence for particles. $f_{\text{fitness}}^{[i]}$ is the fitness of the i th particle and $f_{\text{fitness}}^{\text{avg}}$ is the average fitness of the present particle swarm. Then

$$\rho^2 = \frac{\sum_{i=1}^M (f_{\text{fitness}}^{[i]} - f_{\text{fitness}}^{\text{avg}})^2}{M}, \quad (15)$$

where M is the total number of particles. ρ^2 represents the convergence degree of particles. The smaller ρ^2 is, the more convergent the particle swarm is. λ is set as the threshold value. When $\rho^2 < \lambda$, the particles are considered in the state of premature convergence. We expand the particle diversity by adding Gaussian disturbance to each particle.

The new particle is produced with

$$s^{[i]}(t) = s^{[i]}(t) (1 + \alpha \cdot G), \quad (16)$$

where G is the Gaussian distributed random number with mean 0 and variance 1. α is an accommodation coefficient.

6. Improved Resample Method

In RBPF, resample can effectively eliminate the degeneration of particles, but it will result in the loss of diversity of particles. The linear optimization resample can effectively eliminate the loss of diversity, whose main idea is to produce new sample particles by linear combination [23, 24]. When a particle is duplicated, this particle and the discarded particle are used to produce a new sample particle by linear combination. The new particle is used to take the place of the duplicated particle; thus the loss of diversity is avoided. However, the existing linear combination method is carried out in each duplicated particle, so the amount of computation increases a lot.

This paper improved the linear combination based resample approach. First, the low-weighted particles are discarded and the high-weighted particles are duplicated according to the traditional resample approach. Second, the number of the particle species is counted. Assume the total number of particles is m and the particle species is n , then the assessment factor of particle diversity is n/m . Let the threshold value of particle diversity is a . If $n/m < a$, the particle depletion occurs and the linear combination is used to produce new particles. Some duplicated high-weighted particles are replaced by the new particles to increase the particle diversity.

The new particle is produced by linear optimization with the discarded particle and the duplicated particle. The combination formula is

$$x_n = x_s + KG(x_a - x_s), \quad (17)$$

where x_n is the new sample particle produced by linear optimization, x_s is the duplicated high-weighted particle, x_a is the discarded low-weighted particle, K is the accommodation coefficient, and G is the Gaussian distributed random number with mean 0 and variance 1.

Although the low-weighted particles are discarded after the linear optimization, whose information still exists in RBPF with a certain probability, with the proposed judgment method of particle diversity, the linear combination is only carried out when the particle diversity is below the threshold value. Thus, the computing time of the proposed resample method is less than that of [23].

7. Steps of QPSO-FastSLAM

The procedures of proposed QPSO-FastSLAM are as follows.

Step 1 (estimation). Sample particles from proposal distribution: estimate the particle $s_t \sim p(s_t | s_{t-1}, u_t)$ at the next time form the motion model.

Step 2 (improved by QPSO).

- (a) Initialize particle.
- (b) Compute the personal best position and the global best position of the particles.
- (c) Obtain the optimized estimated particle s_t^* .
- (d) Compute the fitness variance ρ^2 , if $\rho^2 < \lambda$, go to (e), else go to (f).
- (e) Gaussian disturbance according to (16): return to (b).
- (f) Compute the fitness function f_{fitness}^i ; if $f_{\text{fitness}}^i < \delta$, return to (b); else if $f_{\text{fitness}}^i \geq \delta$, go to Step 3; and $P_g(t) = s_t^*$.

Step 3. Compute weight $w_t = w_{t-1} p(z_t | s_t^*)$.

Step 4 (resample).

- (g) Sample s_t^* according to weight and adding them to particles.
- (h) Particle diversity judgment: if $n/m < a$, go to (i); else go to Step 5.
- (i) Linear combination according to (17).

Step 5. Update the estimated maps $p(m^{[i]} | s_{1:t}^{[i]}, z_{1:t})$ with the particle poses and observation z .

8. Simulations and Experiments

8.1. Simulations. The proposed algorithm is evaluated on simulated data. The simulation results are used to prove the applicability of the proposed approach and compare the performance with other methods. In the following simulations and experiments, the results of QPSO-FastSLAM are compared with other FastSLAM frameworks.

We use FastSLAM simulator developed by Bailey to simulate the proposed approach. QPSO-FastSLAM, PSO-FastSLAM, and FastSLAM 2.0 are implemented in the same

TABLE 1: Computational cost and estimation errors.

Number of particles	RMSE of estimated positions (m)		Processing time (s)	
	PSO-FastSLAM	QPSO-FastSLAM	PSO-FastSLAM	QPSO-FastSLAM
10	0.61	0.49	15.9	16.8
20	0.52	0.44	28.3	30.1
30	0.45	0.36	43.4	46.5
100	0.32	0.28	134.8	140.6

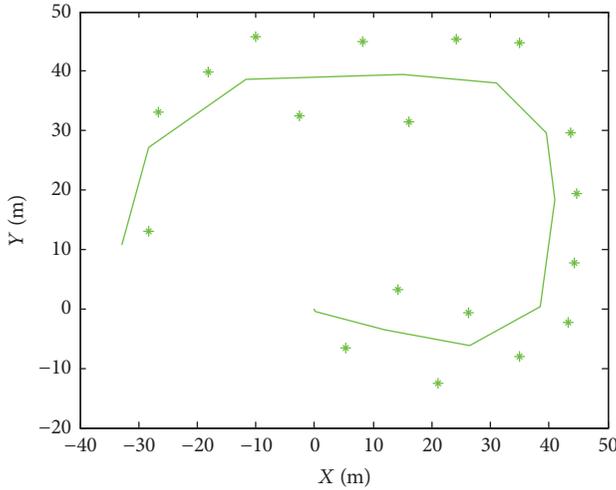


FIGURE 3: Simulation environment.

simulation environment (90 m \times 70 m and 18 landmarks). Figure 3 shows the robot trajectory and landmarks. The green star indicates the position of a landmark. The green line denotes the true robot trajectory. The robot moves at a speed of 3 m/s and the wheel base is 4 m. The maximum steering angle is 30°. The robot is equipped with a range-bearing sensor with a maximum range of 16 m and a 180° frontal field of view. The control frequency is 40 Hz and observation scans are obtained every 5 Hz. The control noise is (0.3 m, 3°) and the measurement noise is (0.1 m, 1°).

8.1.1. Performance Comparison. We compare the performance of the proposed method with that of PSO-FastSLAM and FastSLAM 2.0 in the same simulation environment. The number of particles for RBPF is set to 30. The number of particles for PSO and QPSO are all set to 10. Figure 4 shows the estimated paths and landmarks with the three methods, respectively. The red line denotes the estimated path and each black bordered rectangle denotes an estimated landmark. The RMSE (Root Mean Square Error) is used to evaluate the estimation accuracy, which is

$$\text{RMSE} = \frac{1}{N} \sum_{i=1}^N \sqrt{(\hat{x}^i - x)^2 + (\hat{y}^i - y)^2}, \quad (18)$$

where (\hat{x}^i, \hat{y}^i) and (x, y) are the estimated and true positions, respectively.

Figure 5 shows the RMSE of the estimated positions and features over steps. Each bar in Figure 5 represents the mean

and variance of RMSE for the estimated robot poses and landmarks. The mean and variance of RMSE are calculated over 20 independent Monte Carlo runs for each approach. It is obvious that the estimated accuracy of the proposed method is better than that of PSO-FastSLAM and FastSLAM 2.0. This is because the proposed method improves the proposal distribution of particles and decreases the depletion of particles. The proposed method uses QPSO to make the sampled particles closer to the true poses of the robot. In the resample steps, a new linear optimization method is used to produce new sample particles to increase the particle diversity and eliminate the loss of diversity.

We compare the performance of QPSO-FastSLAM and PSO-FastSLAM with various measurement noise levels. The results are obtained with 30 particles and over 20 independent Monte Carlo runs for each approach. The mean and variance of RMSE with various measurement noise levels are shown in Figure 6. As is shown in the results, the estimated errors of each approach increase as the measurement noise levels increase. However, the mean and variance of RMSE for QPSO-FastSLAM are smaller and increase more slowly than those of PSO-FastSLAM. It is obviously that the QPSO-FastSLAM has better estimation performance than PSO-FastSLAM while the measurement noise levels are varied.

8.1.2. Computational Cost. We use Matlab simulations to analyze the computational cost of the proposed approach. Our simulation platform is an Intel Core i5-4210M PC. When the same number of particles is used, the computational time of QPSO-FastSLAM is approximately close to that of PSO-FastSLAM and more than that of FastSLAM 2.0. This is because the PSO and QPSO increase the computational time. In QPSO-FastSLAM, the linear optimization and Gaussian disturbance also increase the computational time. However, the QPSO improves the accuracy of sampling particles and the proposed method increases the diversity of particles. Thus, QPSO-FastSLAM needs a smaller number of particles than PSO-FastSLAM to achieve the same estimation accuracy. As is shown in Table 1, to achieve the same estimation accuracy, the time-consuming of QPSO-FastSLAM is less than that of PSO-FastSLAM.

8.2. Experiments

8.2.1. Victoria Park Data Set. We use Victoria Park data set to compare the proposed algorithm with PSO-FastSLAM. The data set is a benchmark data set in Victoria Park [25], which is collected with an outdoor vehicle in experiments by Australian Centre for Field Robotics (ACFR). This data

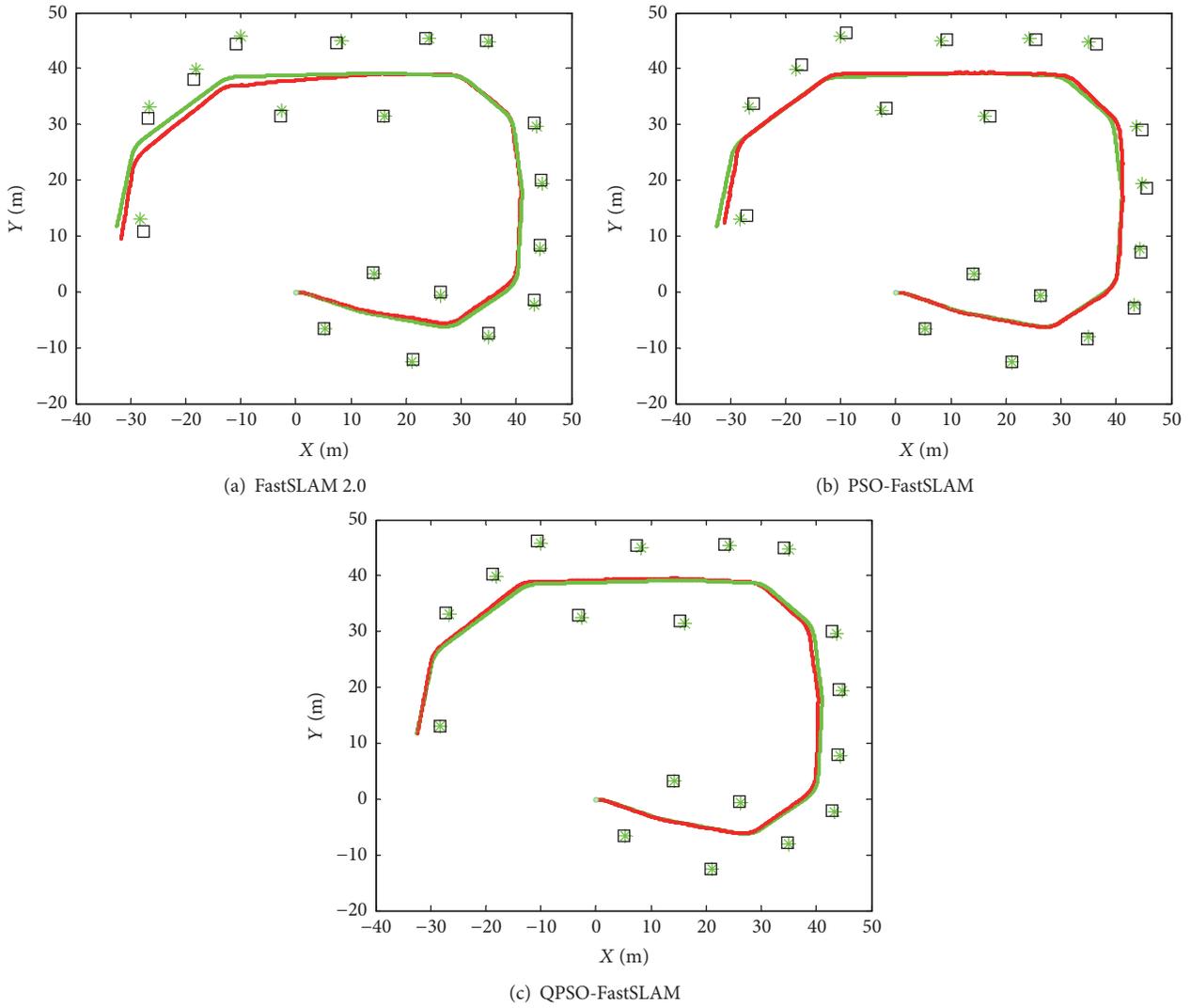


FIGURE 4: Estimated paths and landmarks; true paths and landmarks.

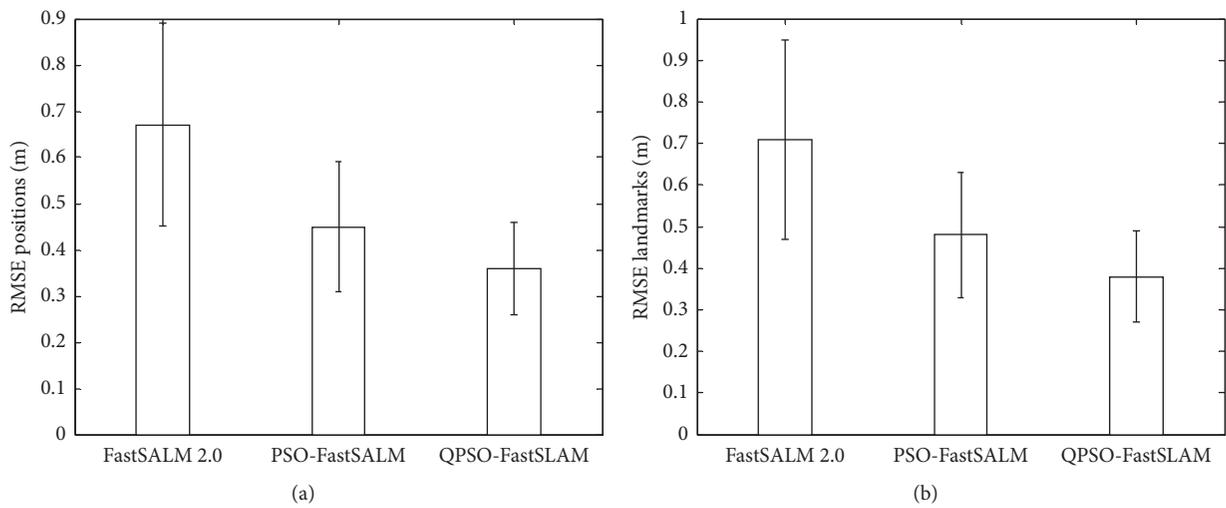


FIGURE 5: RMSE of the estimated positions and landmarks.

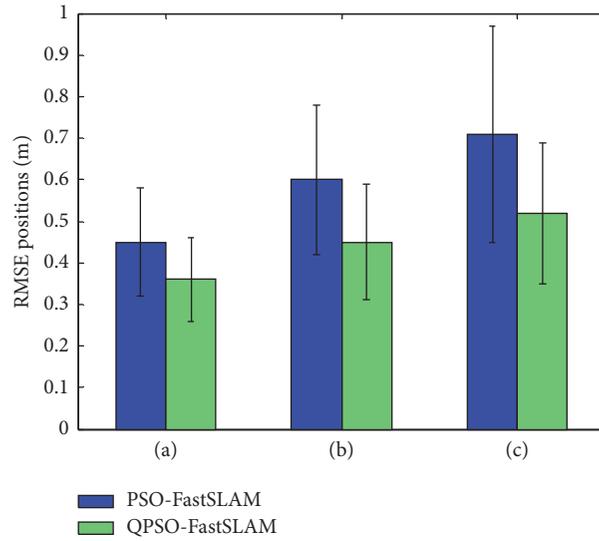


FIGURE 6: RMSE with various measurement noise levels. Measurement noises of each simulation are as follows: (a) (0.1 m, 1°), (b) (0.4 m, 2°), and (c) (0.7 m, 3°).

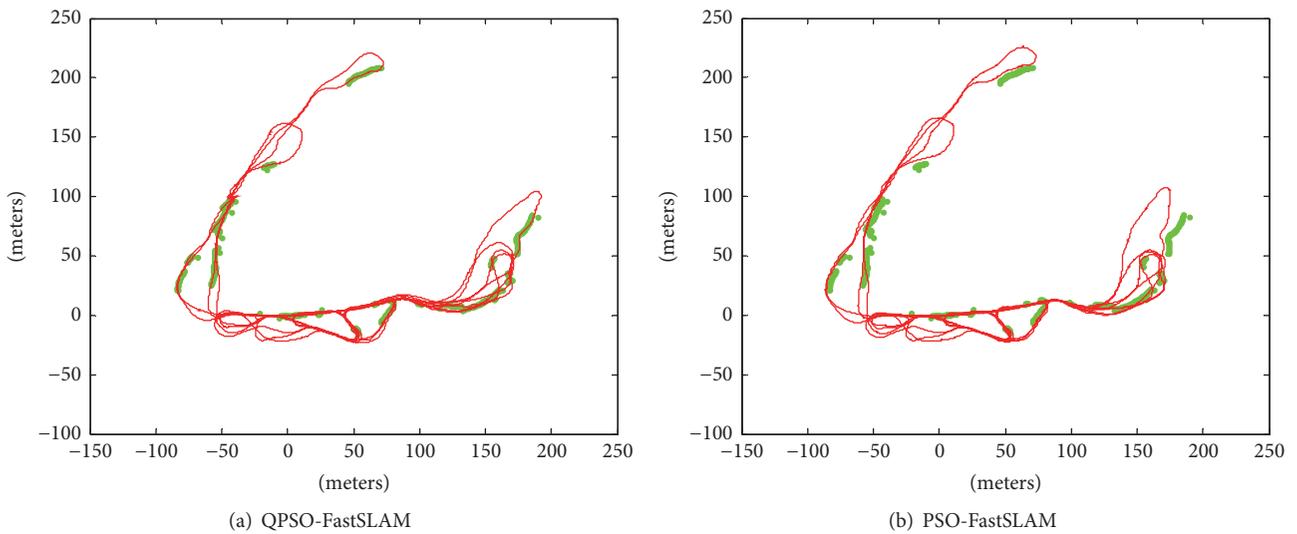


FIGURE 7: Victoria Park estimation results.

set has become a widely used standard data set in SLAM areas, which locates at a larger area. The positions of the car are obtained by GPS, which can be considered as the true positions. The true positions can be used to evaluate the performance of SLAM algorithms and are not involved in the SLAM operations. The velocity and the steering angle are measured by sensors.

In our experiments, the control noise is set as (1.6 m/s, 3.8°/s) and the observation noise is set as (0.7 m, 2.5°). The wheel base of the car is 2.83 m. The number of particles for RBPF is set to 20. The number of particles for PSO and QPSO are all set to 10.

Figure 7 shows the Victoria Park estimation results. The green dots denote the positions from GPS, which are considered as the true positions of the car. The red lines

indicate the trajectories of PSO-FastSLAM and the proposed approach. Each approach has been executed 10 times independently. From the trajectories, we can see that the estimated path of the proposed method is more consistent with the GPS path than that of PSO-FastSLAM. Thus the performance of the proposed approach is better than that of PSO-FastSLAM. The proposed approach improves the accuracy of state estimations.

8.2.2. Car Park Data Set. The second experiment is performed in the Car Park data set to compare the proposed algorithm with PSO-FastSLAM. The data set is also collected by ACFR in the top level of the car park building of the university campus [26]. The data set includes the information of high quality GPS, dead reckoning sensors, and a laser

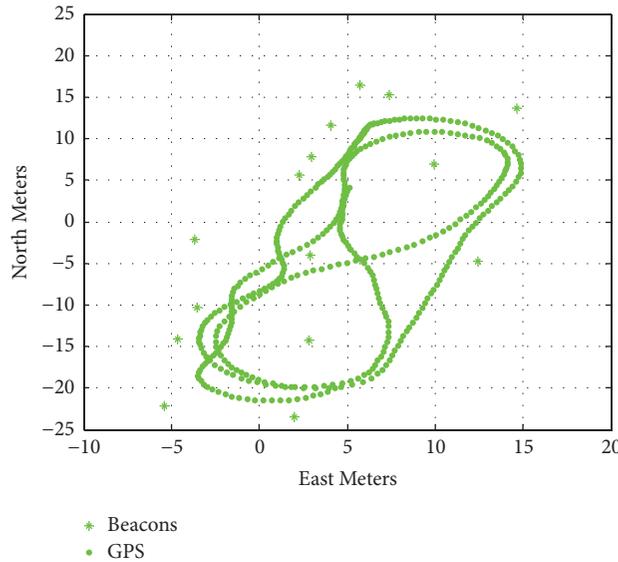


FIGURE 8: Car Park experimental environment.

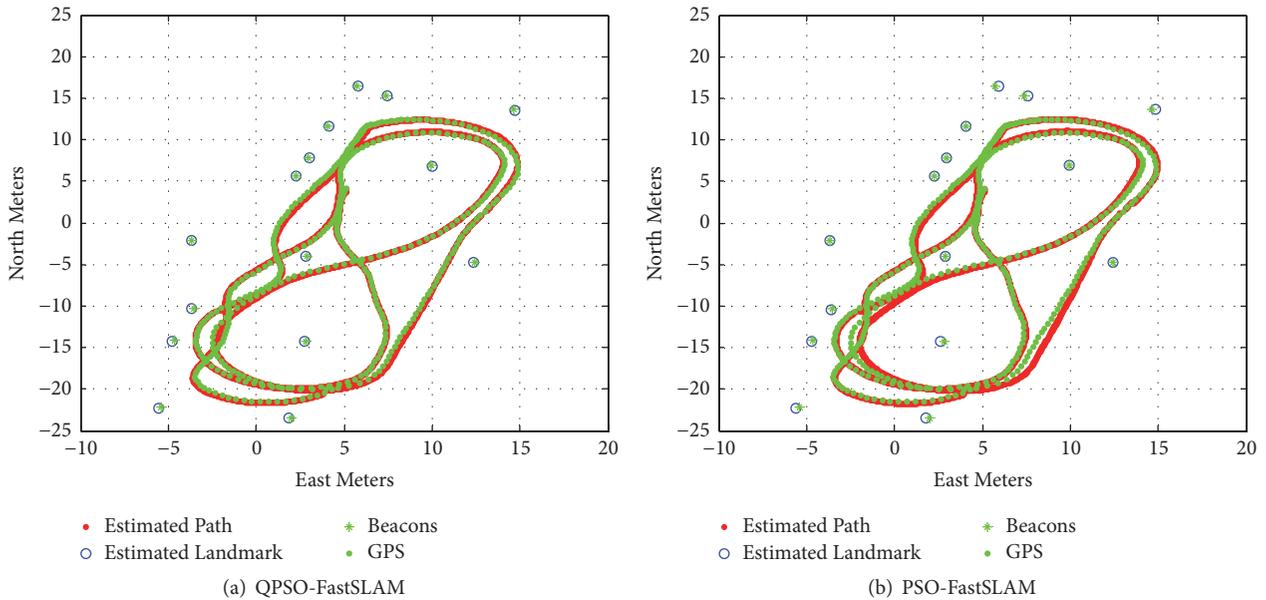


FIGURE 9: Car Park estimation results.

range sensor. The true positions of the landmarks and path are obtained with GPS for comparison purposes, which are shown in Figure 8.

In our experiments, the control noise is set as (0.7 m/s, 7°/s) and the observation noise is set as (0.1m, 1°). The number of particles for RBPf is set to 100. The number of particles for PSO and QPSO is all set to 30.

Figure 9 shows the path and landmark estimates produced by the proposed approach and PSO-FastSLAM. The red “.” is the estimated path, the blue “o” is the estimated beacon positions, the green “.” is the GPS path reference, and the green “*” is the beacon positions given by the GPS. Each approach has been executed 30 times independently. Figure 10 shows the RMSE error of the estimated positions

and landmarks for the two algorithms, respectively. It is obviously that the performance of the proposed method is better than that of the PSO-FastSLAM.

9. Conclusions

The proposal distribution of sample particles influences the estimation accuracy in FastSLAM. Moreover, FastSLAM has been proved to accuracy decrease due to the particle depletion in resampling steps. PSO has been proposed to improve the proposal distribution for FastSLAM, but it can not overcome the depletion problem of FastSLAM and it itself also has the premature convergence problem. In addition, the search ability of PSO is not good. In this paper, we

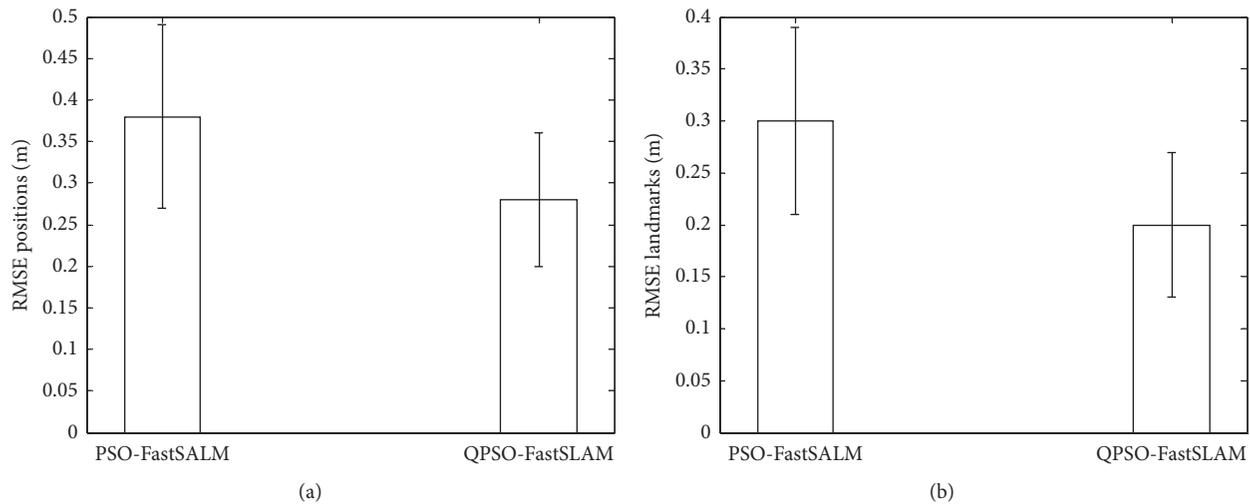


FIGURE 10: RMSE of the estimated positions and landmarks.

proposed the QPSO-FastSLAM approach to improve the estimated accuracy of SLAM. In the resample steps, the linear optimization is used to produce new sample particles, which increases the particle diversity. In the QPSO algorithm, the Gaussian disturbance is used to overcome the premature convergence of particles. The results of simulations and experiments indicate that the proposed approach can improve the estimated accuracy and overcome the particle depletion. The diversity of particles can be maintained by this method. The estimated pose of a particle is closer to the true pose. Meanwhile, compared with PSO-FastSLAM the time-consuming of QPSO-FastSLAM does not increase greatly. To achieve the same estimated accuracy, the time-consuming of QPSO-FastSLAM is less than that of PSO-FastSLAM.

Conflicts of Interest

The authors declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Acknowledgments

This work was supported by the project of National Natural Science Fund under Grant nos. 61175094 and 61673304, the National Defense Advance Researching Fund of Wuhan University of Science and Technology under Grant no. GF201702, and the Open Research Foundation of the Engineering Research Center for Metallurgical Automation and Detecting Technology under Grant no. MADT201703.

References

- [1] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: part I the essential algorithms," *IEEE Robotics Automation Magazine*, vol. 13, no. 2, pp. 99–110, 2006.
- [2] T. Bailey and H. Durrant-Whyte, "Simultaneous localization and mapping: part II the essential algorithms," *IEEE Robotics Automation Magazine*, vol. 13, no. 3, pp. 108–117, 2006.
- [3] J. Sola, T. Vidal-Calleja, J. Civera, and J. M. Montiel, "Impact of landmark parametrization on monocular EKF-SLAM with points and lines," *International Journal of Computer Vision*, vol. 97, no. 3, pp. 339–368, 2012.
- [4] Z. Li and Z. Chunxia, "A new FastSLAM algorithm based on iterated EKF," *Journal of Shandong University*, vol. 42, no. 4, pp. 41–47, 2012.
- [5] M. Montemerlo, T. Sebastian, D. Koller, and B. Wegbreit, "FastSLAM: a factored solution to the simultaneous localization and mapping problem," in *Proceedings of the Eighteenth National Conference on Artificial Intelligence*, pp. 593–598, Menlo Park, Calif, USA, 2002.
- [6] M. Montemerlo, S. Thrun, D. Roller, and B. Wegbreit, "FastSLAM 2.0: 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, August 2003.
- [7] G. Grisetti, C. Stachniss, and W. Burgard, "Improved techniques for grid mapping with rao-blackwellized particle filters," *IEEE Transactions on Robotics*, vol. 23, no. 1, pp. 34–46, 2007.
- [8] H.-C. Lee, S.-K. Park, J.-S. Choi, and B.-H. Lee, "PSO-FastSLAM: An improved FastSLAM framework using particle swarm optimization," in *Proceedings of the 2009 IEEE International Conference on Systems, Man and Cybernetics (SMC '09)*, pp. 2763–2768, San Antonio, TX, USA, October 2009.
- [9] Z.-S. Zhao, X. Feng, Y.-Y. Lin et al., "Improved rao-blackwellized particle filter by particle swarm optimization," *Journal of Applied Mathematics*, vol. 2013, Article ID 302170, 7 pages, 2013.
- [10] C. Baifan, C. Zixing, and Y. Cheng, "Mobile robot SLAM method based on particle swarm optimization," *Robot*, vol. 31, no. 6, pp. 513–517, 2009.
- [11] M. Jia-cheng, Q. Zhang, X. Wei, and M. Li-yong, "SLAM Method for Mobile Robot Based on Particle Swarm Optimization," *Transactions of Beijing Institute of Technology*, vol. 33, no. 11, pp. 1151–1154, 2013.
- [12] Y. Erliang, Z. Guoliang, T. Wenjun, and X. Jun, "Rao-Blackwellized particle filter simultaneous localization and mapping algorithm based on particle swarm optimization," *Journal of Computer Applications*, vol. 34, no. S2, pp. 37–40, 2014.

- [13] C. Kim, R. Sakhivel, and W. K. Chung, "Unscented FastSLAM: a robust and efficient solution to the SLAM problem," *IEEE Transactions on Robotics*, vol. 24, no. 4, pp. 808–820, 2008.
- [14] R. Havangi, H. D. Taghirad, M. A. Nekoui, and M. Teshnehlab, "A square root unscented fastSLAM with improved proposal distribution and resampling," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 5, pp. 2334–2345, 2014.
- [15] D. Liu, G. Liu, and M. Yu, "An improved FastSLAM framework based on particle swarm optimization and unscented particle filter," *Journal of Computational Information Systems*, vol. 8, no. 7, pp. 2859–2866, 2012.
- [16] N. A. Shrivastava, A. Khosravi, and B. K. Panigrahi, "Prediction interval estimation of electricity prices using pso-tuned support vector machines," *IEEE Transactions on Industrial Informatics*, vol. 11, no. 2, pp. 322–331, 2015.
- [17] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks (ICNN '95)*, vol. 4, pp. 1942–1948, Perth, Western Australia, November-December 1995.
- [18] J. Sun, B. Feng, and W. Xu, "Particle swarm optimization with particles having quantum behavior," in *Proceedings of the Congress on Evolutionary Computation (CEC '04)*, vol. 1, pp. 325–331, June 2004.
- [19] D. Chen, Q. Gong, Z. Jin, J. Zhang, and D. Wang, "Short-term wind power prediction based on support vector regression machine optimized by adaptive disturbance quantum-behaved particle swarm optimization," *Power System Technology*, vol. 37, no. 4, pp. 974–980, 2013.
- [20] H. Yu, X. Tiantian, and H. Pu, "An improved quantum particle swarm optimization and its application in system identification," in *Proceedings of the Control and Decision Conference IEEE*, vol. 31, no. 20, pp. 1132–1134, 2011.
- [21] T. Bailey, *Mobile Robot Localisation and Mapping in Extensive Outdoor Environments [Ph.D. Thesis]*, University of Sydney, 2002.
- [22] S. Bernard Williams, *Efficient Solutions to Autonomous Mapping and Navigation Problems [Ph.D. Thesis]*, University of Sydney, 2001.
- [23] G.-H. Zou, Z.-L. Jing, and H.-T. Hu, "Particle filter algorithm based on optimizing combination resampling," *Journal of Shanghai Jiaotong University*, vol. 40, no. 7, pp. 1135–1139, 2006.
- [24] C. Tian-qing, L. Yong, L. Zhong-ren, and D. Tian-zhao, "Particle filter algorithm based on improved resampling," *Application Research of Computers*, vol. 30, no. 3, pp. 748–750, 2013.
- [25] J. E. Guivant and E. M. Nebot, "Optimization of the simultaneous localization and map-building algorithm for real-time implementation," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 3, pp. 242–257, 2001.
- [26] J. Nieto, J. Guivant, E. Nebot, and S. Thrum, "Real time data association for FastSLAM," in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA '03)*, vol. 1, pp. 412–418, 2003.

