

## Research Article

# Multirobot FastSLAM Algorithm Based on Landmark Consistency Correction

Shi-Ming Chen,<sup>1</sup> Jun-Feng Yuan,<sup>1</sup> Fang Zhang,<sup>1</sup> and Hua-Jing Fang<sup>2</sup>

<sup>1</sup> School of Electrical and Electrical Engineering, East China Jiaotong University, Nanchang 330013, China

<sup>2</sup> Department of Control Science and Engineering, Huazhong University of Science and Technology, Wuhan 430074, China

Correspondence should be addressed to Jun-Feng Yuan; [yesyjf817@163.com](mailto:yesyjf817@163.com)

Received 19 December 2013; Revised 16 April 2014; Accepted 20 April 2014; Published 12 May 2014

Academic Editor: Piermarco Cannarsa

Copyright © 2014 Shi-Ming Chen 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.

Considering the influence of uncertain map information on multirobot SLAM problem, a multirobot FastSLAM algorithm based on landmark consistency correction is proposed. Firstly, electromagnetism-like mechanism is introduced to the resampling procedure in single-robot FastSLAM, where we assume that each sampling particle is looked at as a charged electron and attraction-repulsion mechanism in electromagnetism field is used to simulate interactive force between the particles to improve the distribution of particles. Secondly, when multiple robots observe the same landmarks, every robot is regarded as one node and Kalman-Consensus Filter is proposed to update landmark information, which further improves the accuracy of localization and mapping. Finally, the simulation results show that the algorithm is suitable and effective.

## 1. Introduction

Simultaneous localization and map building (SLAM) problem considers the situation in which an autonomous mobile robot moves through an unknown environment and incrementally builds a map of this space while simultaneously making use of this map to refresh its absolute location [1–3]; SLAM algorithm is deemed to be the key factor of the real sense of the mobile robot in autonomous navigation. Compared with single-robot SLAM, multirobot SLAM algorithm can not only get more accurate and faster environmental information but also have better self-positioning ability, which is mainly reflected in two aspects: (1) the multirobot can cover the entire environment faster via communication and coordination in parallel; explore [4]; (2) the localization and mapping errors can be reduced through integrating measurement and navigation information of multiple robots [5]. In recent years, research on multirobot SLAM has drawn more and more attention, and it gradually becomes a hot issue in robot field [6, 7].

At present, the existing approaches for multirobot SLAM mainly focus on expanding the single-robot SLAM algorithm

to the multirobot case. Huang et al. [8] study the consistent fusion of multirobot relative measurements based on Extended Kalman Filtering (EKF); they employ the method in which multirobots jointly maintain the state of global map and its covariance matrix and use consistency protocol to fuse multirobot measurements information for the global update. However, system linearization error and high complexity are the biggest drawbacks of this kind of algorithm based on EKF; Yuan et al. [9] introduce a way about local submap, in which robots build their maps independent of each other and then get the global map by EKF traverse map fusion; Howard [10] proposes an effective online MCSLAM algorithm, which extends the Blackwellised particle filter, and multirobots share measurement information with each other for collaborative SLAM. However, the algorithm has no matching fusion for local map when two robots meet but rather employs the process of time reverse to share information about the history of measurements after robots encounters, which makes the global map update delay and reduces the efficiency of building map due to the same measurement and motion control information reused by multirobot SLAM process. Cai et al. [11] adopt auxiliary

robot to provide the leader robot with measurement and put forward an online multirobot FastSLAM algorithm based on auxiliary measurements, which can improve the accuracy of localization and mapping about lead robot, but the SLAM problem of auxiliary robot cannot be solved. Gil et al. [12] and Lee et al. [13] study multirobot SLAM algorithm based on particle filter; they consider the uncertainty of localization, but no attention is paid to uncertainty of map information.

Considering uncertainty of the map information, this paper presents a new multirobot FastSLAM algorithm based on landmark consistency correction (MBLCC-FastSLAM). Firstly, electromagnetism-like mechanism is introduced to resampling procedure in single-robot FastSLAM; attraction-repulsion mechanism in electromagnetism field is used to simulate interactive force between particles, where attractions drive the particles approach to high likelihood region; moreover, repulsions drive the particles separate from each other and maintain a certain distance in moving process to assure diversity. Then, when multiple robots observe the same landmarks, every robot is regarded as one node and Kalman-Consensus Filter (KCF) is used to update landmark location, which further improves the accuracy of localization and mapping.

## 2. Problem Statement

Suppose the prediction and measurement model of robot system at time  $k$  are represented by prior probability function  $p(x_k | x_{k-1}, u_k)$  and measurement probability function  $p(z_k | x_k, m)$ , where the state vector  $x_k$  and map  $m$  denote the pose of mobile robot and the position of environmental features of mobile robot at time  $k$ , respectively,  $u_k$  is control variable, and  $z_k$  is observed quantity at time  $k$ . The SLAM problem can be expressed as the joint posterior probability distribution function  $p(x_{1:k}, m | z_{1:k}, u_{1:k})$ , where  $x_{1:k}$  is robot trajectory,  $z_{1:k}$  is measurement sequence, and  $u_{1:k}$  is robot control sequence.

The FastSLAM algorithm is based on the idea of Rao-Blackwellised; the posterior joint probability distribution is decomposed into robot trajectory estimation and maps estimation [14, 15]. It states that if one knew the path of the robot, the landmark positions could be estimated independently; so, under Bayesian formula and a Markov assumption, the joint probability distribution of SLAM can make decomposition as follows:

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

where  $M$  is the number of landmarks. Particle filter is used to estimate the pose of robot in FastSLAM algorithm; each particle can save a map along with  $M$  Gaussian landmarks; the remaining posterior of landmark locations is analytically calculated by EKF filter. Each particle in FastSLAM is of the form  $\{X_k^i, w_k^i, \mu_1, \mu_2, \mu_3, \dots, \mu_M, \Sigma_M\}_1^N$ , where  $X_k^i = (x_k^i, y_k^i, \theta_k^i)$

denotes the position of particle  $i$ ,  $w_k^i$  is the weight of particle  $i$ ,  $\mu_M$  and  $\Sigma_M$  denote the mean and covariance of landmark  $M$ , respectively. The posterior probability distribution is generally hard to obtain. So particles are sampled from proposal distribution function  $q(x_{1:k} | z_{1:k}, u_{1:k})$  and it satisfies the following equation which can approximate the posterior probability distribution:

$$\begin{aligned} q(x_{1:k}^i | z_{1:k}, u_{1:k}) &= q(x_k^i | x_{k-1}^i, z_k, u_k) q(x_{1:k-1}^i | z_{1:k-1}, u_{1:k-1}). \end{aligned} \quad (2)$$

Then the weight of sampling particles can be calculated by

$$w_k^i = \frac{p(x_k^i | z_{1:k}, u_{1:k})}{q(x_k^i | z_{1:k}, u_{1:k})} = \bar{w}_{k-1}^i \frac{p(z_k | x_k^i) p(x_k^i | x_{k-1}^i, u_k)}{q(x_k^i | x_{k-1}^i, z_k, u_k)}, \quad (3)$$

where  $\bar{w}_k^i = w_k^i / \sum_{j=1}^N w_k^j$  is the weight normalized value.

The differences between FastSLAM2.0 and FastSLAM1.0 are as follows: FastSLAM1.0 uses only prior distribution  $p(x_k | x_{k-1}, u_k)$  as the proposal distribution function and thus ignores the current measurement, which is particularly troublesome, if the measurement is too accurate to the robot's motion noise. FastSLAM2.0 is an improved version; it estimates robot pose by EKF algorithm at first and then uses the estimating mean and variance to construct a Gauss distribution function as proposal distribution, and, as a result, FastSLAM2.0 is greatly superior to FastSLAM1.0.

FastSLAM algorithm is based on sequential important sampling (SIS) method; as a result, the particle degradation and sample dilution are its major problems [16]. In order to solve the problems and improve the estimation performance of FastSLAM2.0, this paper presents a new algorithm called MBLCC-FastSLAM, as will be shown in the next paragraphs.

## 3. MBLCC-FastSLAM

*3.1. Electromagnetism-Like Mechanism Is Introduced to Single-Robot Resampling.* Electromagnetism-like mechanism algorithm is a new kind of natural heuristic global optimization algorithm [17]; its main idea is to assume that every solution is an electron, and then attraction-repulsion mechanism in electromagnetic field is simulated to drive search particles to move to the optimal solution by certain rules. The four basic procedures of electromagnetism-like mechanism algorithm are as follows (to calculate the maximum value of function  $f(x)$  as an example).

- (a) Initialization:  $N$  points are sampled randomly from known feasible region; then the objective function value (OFV) of each particle is calculated by  $f(x)$ , and the point that has best OFV is stored in  $x^{\text{best}}$ .
- (b) Local search: it is a neighborhood search procedure, which can be applied to one or many points for local refinement at each iteration.

(c) Calculation of total force vector: in this procedure, a charged-like value is assigned to each point; the charges are calculated by

$$Q_i = \exp \left\{ -n \frac{f(x^{\text{best}}) - f(x^i)}{\sum_{k=1}^N (f(x^{\text{best}}) - f(x^k))} \right\}, \quad (4)$$

where  $n$  is the dimension of point  $x$ . Then, the total force  $F_i$  is calculated by the calculation formula of force and the superposition principle in electromagnetism field:

$$F_i = \begin{cases} \frac{\sum_{j \neq i}^N (x^j - x^i) Q_i Q_j}{\|x^j - x^i\|^2} & (f(x^j) > f(x^i)); \\ \frac{\sum_{j \neq i}^N (x^i - x^j) Q_i Q_j}{\|x^j - x^i\|^2} & (f(x^j) < f(x^i)); \end{cases} \quad (5)$$

as can be seen in (5), between two points, the point that has a larger OFV attracts the other one; contrary to it, the point with worse OFV repels the other.

(d) Movement along total force vector: points are moved along the total force vector  $F_i$ ; the movement is according to

$$x^i = \frac{x^i + \lambda R F_i}{\|F_i\|}, \quad (6)$$

where  $\lambda$  is the random step length, which is uniformly distributed between 0 and 1;  $R$  is the mobile coefficient, which denotes the allowed range of movement toward the lower bound or the upper bound for the corresponding dimension; through the above procedures, points complete electromagnetism-like mechanism iteration and move toward a better set of points.

Electromagnetism-like mechanism algorithm has advantages of less parameters and a good ability of global optimization; when the mobile coefficient  $R$  is selected appropriately, the stochastic points can be driven close to  $x^{\text{best}}$  quickly according to (6); it also has a relatively fast convergence speed [18]. In order to take both particle degradation and sample dilution into account, we introduce electromagnetism-like mechanism algorithm into FastSLAM2.0 resampling procedure for improving the particle distribution and overall quality of particles.

At a given time instant  $k$ , FastSLAM2.0 algorithm based on electromagnetism-like mechanism (EM-FastSLAM) performs the following steps.

*Step 1* (sampling). Considering both the control variable  $u_k$  and the current measurement  $z_k$ , EKF algorithm is used to estimate robot pose at first and obtain the improved proposal distribution, from which samples particles set  $\{x_k^i\}_1^N$ .

*Step 2* (weights calculation). The weight of particles is calculated by (3).

*Step 3* (the electromagnetism-like mechanism optimization sampling process). Each particle is regarded as an electron. First, the charge of each particle is calculated by (4). Here,  $N$  is the number of particles,  $f(x)$  is the weight calculation formula, and  $x^{\text{best}}$  denotes the optimal location of the robot. Then, the total force vector  $F_i$  is calculated; we use the following total force calculation model in this paper:

$$\begin{aligned} F_{ij} &= (x^j - x^i) Q_i Q_j, & f(x^j) > f(x^i), \\ F_{ij} &= (x^i - x^j) Q_i Q_j, & f(x^j) \leq f(x^i), \end{aligned} \quad (7)$$

$$F_i = \sum F_{ij}, \quad j = 1, 2, \dots, N, \quad j \neq i,$$

where  $F_{ij}$  is the force particle  $x^j$  on particle  $x^i$ ; finally, particles are moved along the total force vector  $F_i$  by (6) and thus get a new batch of high quality particles. Equation (3) is reused to calculate the weight of new particles and weight normalized by  $\bar{w}_k^i = w_k^i / \sum_{j=1}^N w_k^j$ .

*Step 4* (resampling). The effective particle number is  $N_{\text{eff}} = 1 / \sum_{i=1}^N (\bar{w}_k^i)^2$ . If  $N_{\text{eff}}$  is less than the given threshold, then the algorithm starts resampling and gets a new particles set  $\{x_k^{i*}, w_k^{i*}; i^* = 1, 2, \dots, N\}$  or does not resample.

*Step 5* (map updating). Each particle updates its observed landmarks by EKF with current state estimate  $x_k^i$  and the previous map.

*Step 6*. Repeat the above steps until there are no new measurements.

As can be seen, after the electromagnetism-like mechanism optimization sampling process, between two particles, the particle that has a greater weight attracts the other one; contrary to it, the particle with worse weight repels the other. So the particles are driven to high likelihood region by electromagnetism-like attractions, which can reduce particle degradation. Simultaneously, electromagnetic-like repulsions drive the particles to separate from each other and maintain a certain distance in moving process to assure diversity.

*3.2. Landmark Consistency Correction.* When robot explores an unknown area and gets measurements of the landmark feature, all of the landmark position estimates are related. For the reason that they are based on the same pose state estimate of mobile robots, the accuracy of landmark estimation directly affects the positional accuracy and consistent estimation of the map. For each robot, the landmark estimated values are inaccurate. To make things worse, when calculating the particle weights, the uncertainty is brought into next moment. Aiming to decrease the uncertainty of map information, we adopt three robots to work at the same time and suppose each robot applies an EM-FastSLAM algorithm. Furthermore, each robot is regarded as a node and Kalman-Consensus Filter (KCF) is used to update landmark location, which corrects the estimation of landmark for further improving the accuracy of robot localization and map

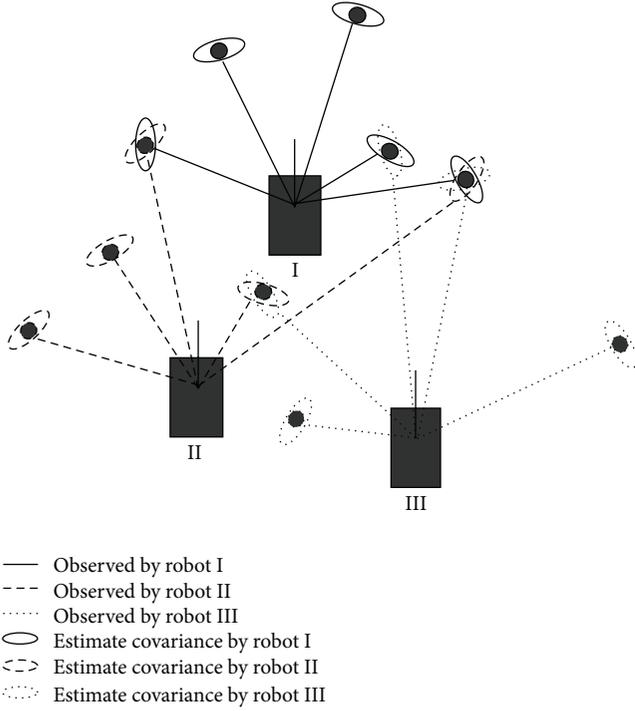


FIGURE 1: Illustration of the MBLCC-FastSLAM algorithm.

building. The illustration of MBLCC-FastSLAM is as shown in Figure 1.

The so-called core process of the KCF [19] is represented by

$$\hat{x}_{k|k}^i = \bar{x}_{k|k-1}^i + K(z_k^i - H\bar{x}_{k|k-1}^i) + C_i \sum (\bar{x}_{k|k-1}^j - \bar{x}_{k|k-1}^i), \quad (8)$$

where  $\hat{x}_{k|k}^i$  and  $\bar{x}_{k|k-1}^i$  are estimate and prior estimate of the node  $i$ , respectively, and  $K$  and  $C_i$  are the Kalman and consensus gains of node  $i$ , respectively. We refer to  $z_k^i$  and  $H$  as sensor data and measurement matrix. From (8), we can know that it includes a Kalman filtering part and a consensus part. The Kalman filtering updates target node prior estimate values with measurements. Meanwhile, because of the capacity of multiagent consistency, the node estimation error can be decreased to some extent and allow all nodes to maintain an approximate value. Based on this idea, each robot is regarded as a node and we use the KCF algorithm to estimate landmarks location for fusing measurements of other robots.

Now, a method for landmark consistency correction is presented.

Firstly, each robot calculates its prior estimate of the landmark:

$$\hat{z}_{j,k}^{t,i} = h(\mu_{j,k-1}^{t,i}, x_k^{t,i}), \quad (9)$$

where  $\hat{z}_{j,k}^{t,i}$  is the prior estimate of robot  $t$  with particle  $i$  to landmark  $j$  at time  $k$  and  $h(\bullet)$  is measurement equation.

$$H_{j,k}^{t,i} = h'(\mu_{j,k-1}^{t,i}, x_k^{t,i}) \quad (10)$$

with (10) is the Jacobian matrix. And the Kalman gain is given by

$$K_{j,k}^{t,i} = \Sigma_{j,k-1}^{t,i} (H_j^{t,i})^T (H_j^{t,i} \Sigma_{j,k-1}^{t,i} (H_j^{t,i})^T + R_k^t)^{-1}, \quad (11)$$

where  $\Sigma_{j,k-1}^{t,i}$  is the covariance of robot  $t$  with particle  $i$  to landmark  $j$  at time  $k-1$  and  $R_k^t$  is measurement noise.

It is assumed that the landmark is stationary; then its prior estimate value is itself at time  $k$ . Landmark location update step by consensus-based filtering is thus given by

$$\begin{aligned} \mu_{j,k}^{t,i} &= \mu_{j,k-1}^{t,i} + K_{j,k}^{t,i} (z_{j,k}^{t,i} - \hat{z}_{j,k}^{t,i}) \\ &+ \varepsilon \sum_{n \in N_{j,k-1}} (\mu_{j,k-1}^{n,\max} - \mu_{j,k-1}^{t,i}), \end{aligned} \quad (12)$$

$$\begin{aligned} \Sigma_{j,k}^{t,i} &= \Sigma_{j,k-1}^{t,i} - \Sigma_{j,k-1}^{t,i} (H_j^{t,i})^T \\ &\times (H_j^{t,i} \Sigma_{j,k-1}^{t,i} (H_j^{t,i})^T + R_k^t)^{-1} H_j^{t,i} \Sigma_{j,k-1}^{t,i}, \end{aligned} \quad (13)$$

where  $\varepsilon$  is the step length,  $N_{j,k-1}$  is the number of robots for getting the same observed landmark  $j$  at time  $k-1$ , and  $\mu_{j,k-1}^{n,\max}$  denotes the maximum weight particle at time  $k-1$ . Notice that, in (12), this method only requires prior estimates exchange between robots which observe the same landmarks. In practical applications, considering the limited network communication broadband, packet-dropping, delay, and other sorts of interference, the algorithm may be of weak anti-interference and poor robustness, if there was too much information exchange between robots [19].

## 4. Simulation Analysis

4.1. *Simulation Model.* In this paper, we adopt three robots to work simultaneously; robot motion model is as follows:

$$X_k^v = \begin{bmatrix} x_k^v \\ y_k^v \\ \Phi_k^v \end{bmatrix} = \begin{bmatrix} x_{k-1}^v + \Delta T \cdot \cos(\Phi_{k-1}^v + a_k) \\ y_{k-1}^v + \Delta T \cdot \sin(\Phi_{k-1}^v + a_k) \\ \Phi_{k-1}^v + \frac{\Delta T \cdot V_k \cdot \sin(a_k)}{D} \end{bmatrix} + \begin{bmatrix} v_x \\ v_y \\ v_\Phi \end{bmatrix}, \quad (14)$$

where  $(x_k^v, y_k^v)$  shows pose of robot in global coordinate,  $\Phi_k^v$  represents yaw angle,  $V_k$  is motion speed,  $a_k$  denotes deflection angle,  $\Delta T$  is sampling time of odometer,  $v_x, v_y$ , and  $v_\Phi$  are noise terms, and  $D$  denotes spacing of two axes.

The measurement model can be expressed as

$$Z_k = \begin{bmatrix} r \\ \theta \end{bmatrix} = \begin{bmatrix} \sqrt{(x_i - x_k^v)^2 + (y_i - y_k^v)^2} \\ \arctan \frac{y_i - y_k^v}{x_i - x_k^v} - \Phi_k^v \end{bmatrix} + \omega_k, \quad (15)$$

where  $r$  and  $\theta$  refer to the distance and direction angle between landmark and robot, which are measured by sensor,  $(x_i, y_i)$  denotes observed position coordinates of the  $i$ th landmark, and  $\omega_k$  is measurement noise.

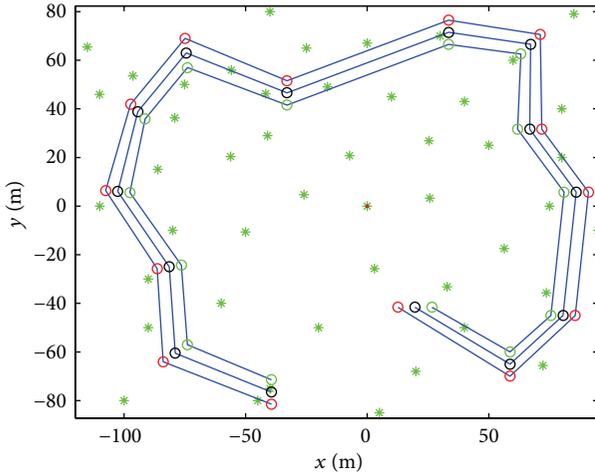


FIGURE 2: The simulated environment.

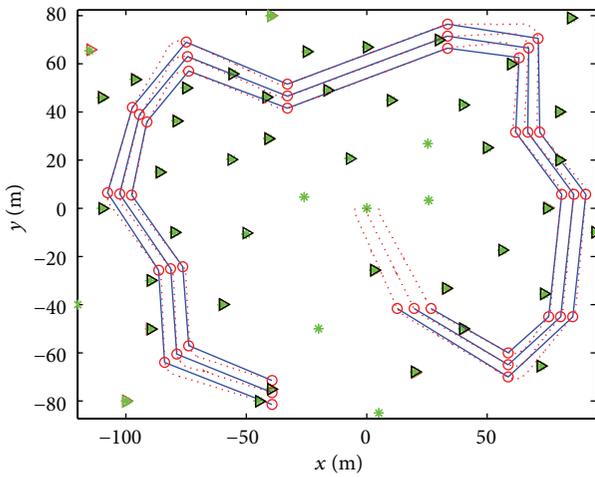


FIGURE 3: MBLCC-FastSLAM estimation.

**4.2. Result Analysis.** In order to verify the effectiveness of the improved algorithm, the proposed MBLCC-FastSALM, multirobot SLAM [10] (with known initial pose), single-robot EM-FastSLAM, and traditional FastSLAM2.0 are simulated for SLAM experiment, respectively, and the estimation accuracy, error, and stability are analyzed to contrast.

The environment simulated is shown in Figure 2, where the asterisk represents a random set of landmarks and the three curves represent the three preset motion paths of robots; the robots run along this path from starting points to terminal points; the speed of robots is  $V_1 = 4$  m/s,  $V_2 = 3.8$  m/s, and  $V_3 = 3.5$  m/s; the effective measurement distance of sensor is 30 m. Assume that the data association is known; using MBLCC-FastSLAM algorithm to obtain the results which are shown in Figure 3 in the simulation environment, where the triangle represents the position of estimated landmarks, the dotted line indicates the estimated trajectory. The process noise is  $(\sigma_V = 0.3$  m/s,  $\sigma_G = 3^\circ)$ , measurement noise is  $(\sigma_r = 0.2$  m,  $\sigma_\theta = 2^\circ)$ , mobile coefficient  $R$  is  $[0.008, 0.005, 0.001]^T$ , landmark update step length is  $\varepsilon = 0.1$ ,

the number of particles  $N$  is 10, and the resampling, when  $N_{\text{eff}}$ , is less than 75% of the total particles.

As shown in Figure 3, the estimated path and landmarks of the robot utilizing MBLCC-FastSLAM algorithm are basically consistent with the true value, which performs with very good estimation ability.

Three robots are simulated in multirobot SLAM, EM-FastSLAM, and FastSLAM2.0 algorithm under the same conditions, respectively. Figure 4 represents the three robots' estimation errors under four different algorithms in a simulation run, from which we can see that the proposed algorithm MBLCC-FastSLAM performs better than the other 3 algorithms in all three robots, particularly in robot 1 and robot 2 in which the estimated errors of MBLCC-FastSLAM are significantly lower both in pose error and landmark error. And EM-FastSLAM is lower than FastSLAM2.0 most of the time.

In order to eliminate randomness and more directly see the results of the four algorithms which are applied 50 times independently in Table 1, the simulation experiment of error statistics is given. By comparing EM-FastSLAM algorithm to FastSLAM2.0 algorithm, the estimated error of the former is lower. For that, the electromagnetism-like mechanism is introduced to resample process in FastSLAM2.0 algorithm, which improves the quality of the particles; similarly, the estimated errors of the proposed MBLCC-FastSLAM algorithm are the lowest; for that, the proposed algorithm gets more correct measurement results by revising landmark based on the consistency. As the correlation of landmark estimation and the pose estimation, the robot localization estimation and map building precision are improved further. Furthermore, we compute the averaging RMSE (ARMSE) in landmark with respect to different sampling particles  $N$ . The simulation results are shown in Figure 5. From Figure 5, it can be seen that the accuracy of four algorithms is improved as the particles increase. On the other hand, when particles exceed about 150, it has little effect on the algorithms when particles continue to increase. At the given noise level, the ARMSE of MBLCC-FastSLAM is the lowest in four algorithms; it can maintain a good precision even when  $N$  is small. In this case, the filtering accuracy of MBLCC-FastSLAM takes 10 particles approximately which is equal to FastSLAM2.0 taking 100 particles.

## 5. Conclusion

In this paper, we presented a multirobot FastSLAM algorithm based on landmark consistency correction. Firstly, electromagnetism-like mechanism is introduced to resampling procedure in single-robot FastSLAM, which improves the distribution of particles by the rule that greater weight particles attract the worse weight particles and the worse weight particles repel greater weight particles. And, then, the landmark location is updated with KCF, when it is observed by multiple robots, which further improves the robot localization and map building precision. Under the same conditions, simulation results show that the MBLCC-FastSLAM algorithm performs more accurately in the localization and map

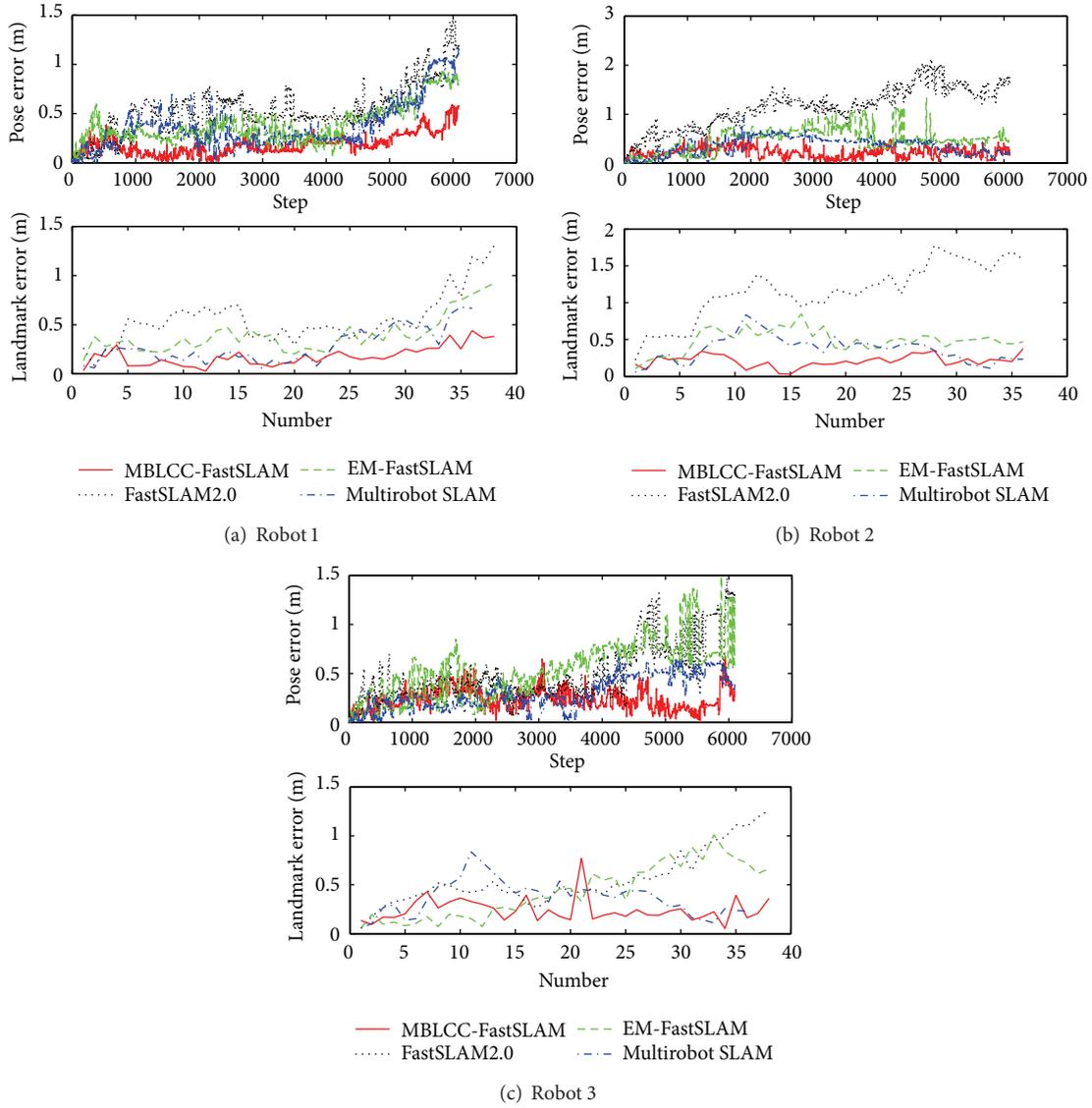


FIGURE 4: Comparison of estimated errors.

TABLE 1: Error statistics for each robot by four algorithms.

Robot	Algorithm	The mean of pose/m	The variance of pose/m	The mean of landmark/m	The variance of landmark/m	The mean of pose in Figure 4/m	The mean of landmark in Figure 4/m
1	FastSLAM2.0	0.7029	0.4280	0.7003	0.4305	0.7521	0.6508
	EM-FastSLAM	0.4101	0.2206	0.4701	0.2159	0.5103	0.4345
	Multirobot SLAM	0.3872	0.3010	0.2808	0.2039	0.5409	0.3811
	MBLCC-FastSLAM	0.2501	0.1850	0.2048	0.1206	0.3772	0.2907
2	FastSLAM2.0	1.2852	0.5213	1.4001	0.6827	1.3212	1.3597
	EM-FastSLAM	0.7469	0.3352	0.5861	0.2925	0.6734	0.4870
	Multirobot SLAM	0.4561	0.2014	0.5031	0.2022	0.4508	0.4421
	MBLCC-FastSLAM	0.2873	0.1769	0.3551	0.1527	0.2883	0.2474
3	FastSLAM2.0	0.7690	0.5263	0.6539	0.5502	0.6113	0.5322
	EM-FastSLAM	0.6889	0.4470	0.5960	0.5019	0.5847	0.3680
	Multirobot SLAM	0.4501	0.2803	0.4391	0.3618	0.3715	0.3952
	MBLCC-FastSLAM	0.3541	0.2061	0.3789	0.3284	0.3003	0.2493

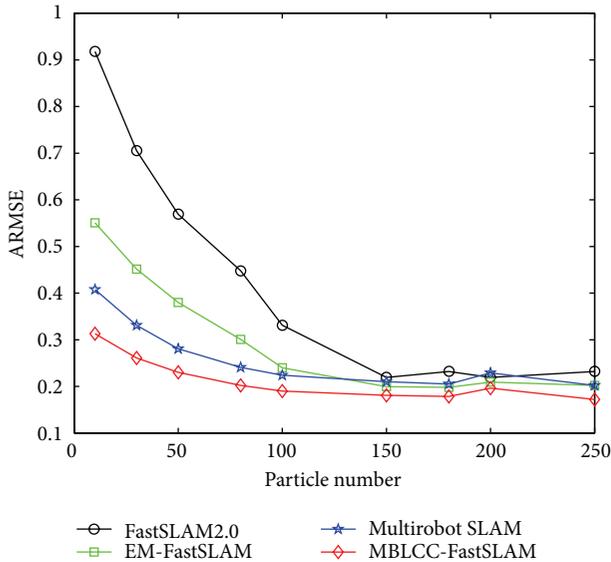


FIGURE 5: ARMSE curves of four algorithms when their numbers change.

building precision compared to multirobot SLAM algorithm [1], EM-FastSLAM algorithm, and traditional FastSLAM2.0 algorithm, which is proved to be effective and feasible.

## Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

## Acknowledgments

This work is supported by National Natural Science Foundation of China (no. 61364017, no. 61034006, and no. 60804066) and the Jiangxi Education Department Scientific and Technological Project (no. GJJ12286).

## References

- [1] S. J. Julier and J. K. Uhlmann, "Using covariance intersection for SLAM," *Robotics and Autonomous Systems*, vol. 55, no. 1, pp. 3–20, 2007.
- [2] J. Leonard and P. Newman, "Consistent, convergent and constant-time SLAM," in *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 1143–1150, 2003.
- [3] B. J. Xiao, K. Xu, X. Chen et al., "A wavelet transform assisted extended kalman filter-based approach for simultaneous localization and mapping problems," *Sensor Letters*, vol. 10, no. 8, pp. 1814–1818, 2012.
- [4] P. Fazli, A. Davoodi, and K. M. Alan, "Multi-robot repeated area coverage," *Autonomous Robots*, vol. 34, no. 4, pp. 251–276, 2013.
- [5] Y. F. Cai, Z. M. Tang, and C. X. Zhao, "A new approach of formation navigation derived from multi-robots cooperative online FastSLAM," *Journal of Control Theory and Applications*, vol. 10, no. 4, pp. 451–457, 2012.
- [6] S. Thrun and Y. F. Liu, "Multi-robot SLAM with sparse extended information filters," *Springer Tracts in Advanced Robotics*, vol. 15, no. 2, pp. 254–266, 2005.
- [7] M. Pfingsthorn, B. Slamet, and A. Visser, "A scalable hybrid multi-robot SLAM method for highly detailed maps," *Lecture Notes in Computer Science*, vol. 50, no. 1, pp. 457–464, 2008.
- [8] G. P. Huang, N. Trawny, A. I. Mourikis, and S. I. Roumeliotis, "Observability-based consistent EKF estimators for multi-robot cooperative localization," *Autonomous Robots*, vol. 30, no. 1, pp. 99–122, 2011.
- [9] J. Yuan, Y.-L. Huang, T. Tao, and B.-Y. Xi, "Multi-robot active simultaneous localization and mapping based on local submap approach," *Robot*, vol. 31, no. 2, pp. 97–103, 2009.
- [10] A. Howard, "Multi-robot simultaneous localization and mapping using particle filters," *International Journal of Robotics Research*, vol. 25, no. 12, pp. 1243–1256, 2006.
- [11] Y. F. Cai, Z. M. Tang, and C. X. Zhao, "An improved algorithm of multi-robots cooperative online fastSLAM," *Journal of Computer Research and Development*, vol. 49, no. 4, pp. 763–769, 2012.
- [12] A. Gil, Ó. Reinoso, M. Ballesta, and M. Juliá, "Multi-robot visual SLAM using a Rao-Blackwellized particle filter," *Robotics and Autonomous Systems*, vol. 58, no. 1, pp. 68–80, 2010.
- [13] H. C. Lee, S. H. Lee, H. Choi et al., "Probabilistic map merging for multi-robot RBPF-SLAM with unknown initial poses," *Robotica*, vol. 30, no. 2, pp. 205–220, 2012.
- [14] M. Montemerlo, S. Thrun, S. T. D. Koller et al., "FastSLAM2.0: an improved particle filtering algorithm for simultaneous localization and mapping that provably converges," in *Proceedings of the International Conference on Artificial Intelligence*, pp. 1151–1156, IJCAI, California, Calif, USA, 2003.
- [15] Q. Zhang, J. C. Ma, and Q. Liu, "Improved fastslam2.0 based on the H4 filter for intelligent mobile robot," *Research Journal of Applied Sciences, Engineering and Technology*, vol. 4, no. 16, pp. 2748–2754, 2012.
- [16] T. Bailey, J. Nieto, and E. Nebot, "Consistency of the FastSLAM algorithm," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '06)*, pp. 424–429, May 2006.
- [17] S. I. Birbil and S.-C. Fang, "An electromagnetism-like mechanism for global optimization," *Journal of Global Optimization*, vol. 25, no. 3, pp. 263–282, 2003.
- [18] S. I. Birbil, S.-C. Fang, and R.-L. Sheu, "On the convergence of a population-based global optimization algorithm," *Journal of Global Optimization*, vol. 30, no. 2-3, pp. 301–318, 2004.
- [19] S. Wang, W. Yang, and H.-B. Shi, "Consensus-based filtering algorithm with packet-dropping," *Acta Automatica Sinica*, vol. 36, no. 12, pp. 1689–1696, 2010.



# Hindawi

Submit your manuscripts at  
<http://www.hindawi.com>

