

Research Article

A Location Predicting Method for Indoor Mobile Target Localization in Wireless Sensor Networks

Peng Gao,¹ Weiren Shi,¹ Wei Zhou,² Hongbing Li,¹ and Xiaogang Wang¹

¹ College of Automation, Chongqing University, Chongqing 400030, China

² School of Electrical and Information Engineering, Chongqing University of Science and Technology, Chongqing 401331, China

Correspondence should be addressed to Peng Gao; jallenwill12@yahoo.cn

Received 6 September 2012; Revised 29 January 2013; Accepted 30 January 2013

Academic Editor: Wei Meng

Copyright © 2013 Peng Gao 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.

Node position information is one of the important issues in many wireless sensor networks' usages. In this paper, based on path planning, a location predicting method (PPLP) for indoor mobile target localization is proposed. We first establish the path planning model to constrain the movement trajectory of the mobile target in indoor environment according to indoor architectural pattern. Then, one certain localization result can be obtained using MLE algorithm. After that, based on the path-planning model and some previous localization results, the most likely position of the target in the next time interval can be predicted with the proposed predicting approach. Finally, the MLE result and prediction result are weighted to obtain the final position. The simulation results demonstrate the effectiveness of the proposed algorithm.

1. Introduction

Wireless sensor networks (WSNs) are widely applied in monitoring, sensing, and collecting the information of interest in the environment [1]. Localization of target nodes is a fundamental problem in wireless sensor networks [2]. Up to now, the most existing localization algorithms of WSNs can be classified into two categories: range-based [3, 4] and range-free [5, 6]. Range-based algorithms use distance or angle estimates in their location estimations. Range-free algorithms use connectivity information between unknown nodes and anchor nodes. Range-based localization algorithms need to measure the actual distances or orientation between adjacent nodes, and then use the measured data to locate unknown nodes. Some ranging methods have been used for distance or orientation estimation, such as RSSI [7, 8], ToA [7, 9], TDOA [7, 10], and AoA [7, 11]. Whatever the ranging method is, there will be measurement errors in practical localization systems that result in noisy range estimations. Thus, accuracy in the position estimation phase is highly sensitive to range measurements [12]. Without improving range estimation or adding some other information related to localization, the accuracy of the current range-based algorithms cannot be improved obviously.

Indoor localization of WSNs has been a hot research topic for the last several years. Due to the randomness of targets moving and the complicated indoor environment, it is very different to locate indoor mobile target. In this paper, we proposed a location predicting method (PPLP) for indoor mobile target localization in WSNs based on path planning. We first establish the path-planning model to constrain the movement trajectory of the mobile target in indoor environment according to indoor architectural pattern. Then, we use MLE approach to get one certain location result of the target. After that, based on the path-planning model and some previous localization results of the target, the best possible position of the target in the next time interval can be predicted with the proposed predicting approach. Finally, the MLE result and prediction result are weighted to obtain the final position. In simulation process, we define three metrics to evaluate the performance of the proposed algorithm and compared with the MLE algorithm and PSO algorithm. Simulation results demonstrate that the proposed algorithm performed better than the other two algorithms.

The rest of this paper is organized as follows: in the next section, some related work is briefly introduced. Section 3 presents a detailed description of the main contribution of

this paper, the proposed algorithm PPLP. The simulation results on localization performance and error analysis are discussed in Section 4. Section 5 concludes.

2. Related Work

2.1. Maximum Likelihood Estimation. Maximum likelihood estimation (MLE) is widely used in many localization applications in wireless sensor networks [13–15]. In the localization process, the number of multiple measurement equations is usually more than the number of variables. Set r_i ($i = 1, 2, \dots, n$) is the estimated distance from anchor sensor node (x_i, y_i) to the target node, the target's position can be calculated as [16]:

$$\hat{\mathbf{U}} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{b}, \quad (1)$$

where

$$\mathbf{A} = 2 \begin{bmatrix} x_n - x_1 & y_n - y_1 \\ x_n - x_2 & y_n - y_2 \\ \vdots & \vdots \\ x_n - x_{n-1} & y_n - y_{n-1} \end{bmatrix}, \quad \hat{\mathbf{U}} = \begin{bmatrix} x_t \\ y_t \end{bmatrix},$$

$$\mathbf{b} = \begin{bmatrix} (r_1^2 - r_n^2) - (x_1^2 - x_n^2) - (y_1^2 - y_n^2) \\ (r_2^2 - r_n^2) - (x_2^2 - x_n^2) - (y_2^2 - y_n^2) \\ \vdots \\ (r_{n-1}^2 - r_n^2) - (x_{n-1}^2 - x_n^2) - (y_{n-1}^2 - y_n^2) \end{bmatrix}. \quad (2)$$

2.2. Particle Swarm Optimization for Localization. Particle swarm optimization (PSO) [17, 18] is a swarm bionic optimization algorithm, which models the behavior of flocks of birds and fish. This method converges to the most optimal solution in a larger probability. Its process does not depend upon the quality of the objective function. So, it is commonly used to solve the optimization problems.

Let $x_i = (x_{i1}, x_{i2})$ be the 2-dimensional vector representing the position of the i th particle in the swarm, $g = [g_1, g_2]$ the position of the best particle in the swarm, $p = [p_{i1}, p_{i2}]$ the current best optimal solution of the i th particle itself and $v_i = [v_{i1}, v_{i2}]$ is the velocity of the i th particle. The particles evolve according to the following equations:

$$v_{id} = \omega v_{id} + c_1 r_1 (p_{id} - x_{id}) + c_2 r_2 (g_d - x_{id}),$$

$$x_{id} = x_{id} + v_{id}, \quad (3)$$

where $d = 1, 2$; $i = 1, 2, \dots, K$; K is the size of the swarm population; ω is the inertial weight; c_1 determines how much a particle is influenced by the memory of its best solution; and c_2 is an indication of the impact of rest of the swarm on the particle. c_1 and c_2 are termed cognitive and social scaling parameters, respectively. r_1 and r_2 are uniform random numbers in the interval $[0, 1]$.

Reference [18] proposed an improved PSO algorithm with RSSI self-correcting localization algorithm for wireless sensor networks. Based on the RSSI ranging, the author combined

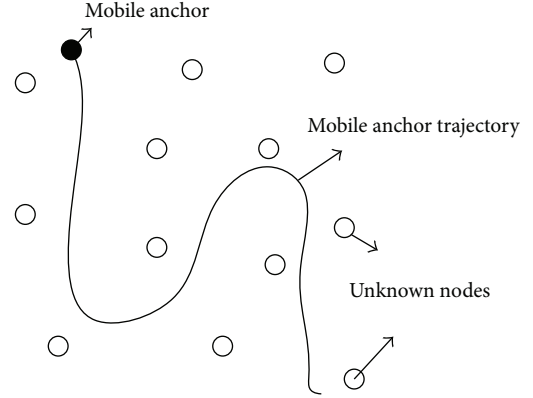


FIGURE 1: A mobile anchor assisting in the localization.

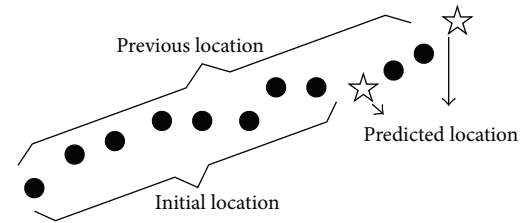


FIGURE 2: Prediction method for WSNs localization.

the proposed RSSI self-correction mechanism and improved PSO algorithm to optimize the nodes' localization for WSNs. Reference [12] proposed two novel and computationally efficient metaheuristic algorithms based on tabu search (TS) and particle swarm optimization (PSO) principles for locating the sensor nodes in a distributed wireless sensor network (WSN) environment. The author compared the performance of the proposed algorithms with each other and also against simulated annealing. The effects of range measurement error, anchor node density, and uncertainty in the anchor node position on localization performance are also studied through various simulations.

2.3. Path-Planning Method for WSNs Localization. Path planning is usually used for mobile anchor node in WSNs localization, where usually requires complex hardware support [19]. A mobile anchor node could be a small mobile robot equipped with a GPS and transmit its coordinate to the rest of the sensors to help them localize themselves. Figure 1 depicts a sensor network deployed over a geographical area. After the deployment, a mobile anchor traverses the sensor network while broadcasting its location packet. The packet contains the coordinates of the anchor, the current time, and some other information such as RSSI. Any node receiving the packet will be able to infer its location with several mobile anchors or one mobile anchor at different times.

2.4. Prediction Method for WSNs Localization. Prediction method is usually used to predict the possible locations of target in the next time interval based on the existing time series data [20] as Figure 2 shows.

Salamah and Doukhnich [9] proposed a new efficient algorithm based on time of arrival (ToA) to determine the position of a mobile object (MO) in a wireless environment. However, it is not suitable for indoor mobile target localization because of the non-line-of-sight (NLOS) propagation in indoor environment.

3. Proposed Algorithm

3.1. Assumptions. We assume that the whole network consists of some stationary anchor nodes (ANs) and a mobile target. The anchor nodes whose coordinates are known are randomly or artificially deployed in a 2-dimensional indoor flat environment. All anchor nodes have the same radio transmission range (R). A mobile target may be a human, a robot, or some object manipulated by some person. Turning point (TP) is the intersection of two subpaths. The target can move freely among various rooms. After encountering some turning points, the target may change or not change its motion path. The position of the target can be calculated periodical with the proposed algorithm. The trajectory of the target can be regarded as a series of discrete points called target nodes (TNs). So, the localization problem changes into solving the locations of the target nodes.

3.2. Path-Planning Model. Generally, the movement of the mobile target (such as a person) is driving by its intention with large randomness. But in indoor environment, the motion trajectory of the target is relatively fixed because of the spatial constraint. People often engage in some typical motion patterns. For example, if a person wants to go to another nonadjacent room, he/she must go out the door first, then cross the corridors, and finally reach his/her destination. It is impossible for him/her to go through walls directly to reach the final position. Target's indoor movement will be limited by the indoor architectural pattern, such as walls and doors. Suppose the location system knows the indoor architectural pattern beforehand, and use it to assist positioning, we can get a better localization accuracy and trajectory of the target.

As Figure 3 shows, any corridor/aisle or room can be viewed as a path. Assume that each path can be described using function $f(x, y)$, then all possible moving paths can be described using path function as follows:

$$F(x, y) = \begin{cases} f_1(x, y), \\ f_2(x, y), \\ \vdots \\ f_m(x, y), \end{cases} \quad x \in X; y \in Y, \quad (4)$$

where X and Y are the ranges of the x -coordinate and y -coordinate, respectively, and $f_m(x, y)$ is the path function for the m th path, called subpath function. All subpath functions form the total path function $F(x, y)$.

However, different buildings have different indoor architectural patterns. In order to make location computing more

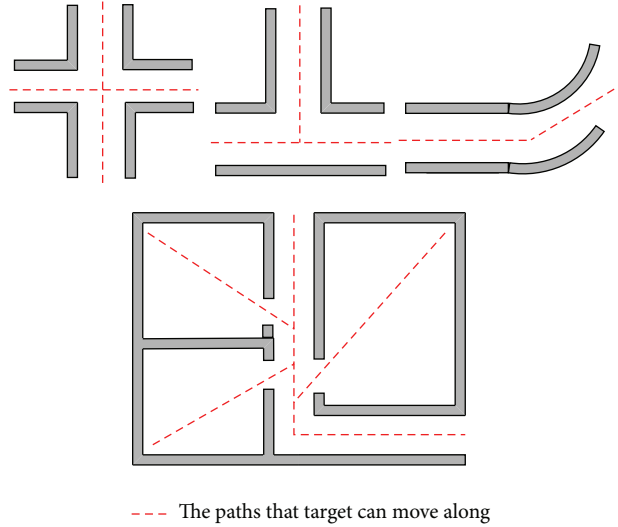


FIGURE 3: An indoor architectural pattern with some indoor paths.

effective, we use a straight line segment function to describe each subpath as follows:

$$(x_b - x_a)y = (y_b - y_a)x + (x_b y_a - y_b x_a), \quad (5)$$

$$\text{s.t. } \begin{cases} \min [x_a, x_b] \leq x \leq \max [x_a, x_b], \\ \min [y_a, y_b] \leq y \leq \max [y_a, y_b], \end{cases} \quad (6)$$

where (x_a, y_a) and (x_b, y_b) are the jumping-off point (JOP) and the end point (EP) of this straight line segment, respectively. A straight line segments subpath can be obtained once JOP and EP are determined. This function can completely (if the real subpath is straight) or approximately (if the real subpath is not straight) describe the real subpath. It will be useful to improve localization accuracy.

3.3. Location Predicting Method. We assume that the maximum velocity of human moving is v_{\max} , and localization is periodically with period being ΔT . It is difficult to determine TN's position according to the previous localization results, because the human moving is random and the localization error exists. However, the localization results can track target's trajectory with high possibility. So our strategy is, first, to compute localization results during a period of time T using some certain localization method (such as MLE); second, to predict the next possible positions according to these localization results; last, the localization result and prediction result are weighted to obtain the final position. In this paper, we use MLE algorithm to finish the first step. We only focused on step two and step three.

Let us use set $G = \{G_1, G_2, \dots, G_k\}$ to describe localization results of the first step during time T , where $G_i = (x^{(i)}, y^{(i)})$. The prediction problem can be described as follows: how to get the next position \hat{G}_{k+1} according to set G and the path-planning model.

For any subpath $f(x, y)$, a set Z can be used to describe all points on this subpath. Each element of set Z satisfied

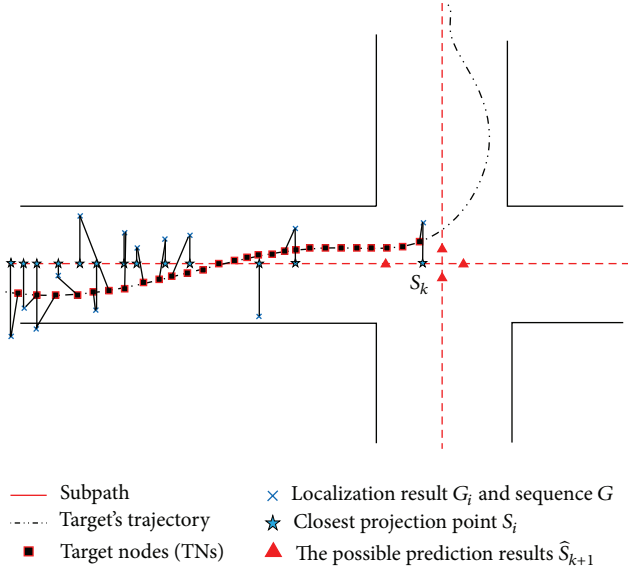


FIGURE 4: Predicting model in indoor environment.

function (5). We can also get that all elements of G are belong to set D which can be described as follows:

$$D = \left\{ (x_D, y_D) \mid \begin{cases} x_{\min} - \Delta x \leq x_D \leq x_{\max} + \Delta x \\ y_{\min} - \Delta y \leq y_D \leq y_{\max} + \Delta y \end{cases} \right\}, \quad (7)$$

where x_{\min} , y_{\min} , x_{\max} , and y_{\max} are the minimum x -coordinate, minimum y -coordinate, maximum x -coordinate, and maximum y -coordinate among all elements of G , respectively. Δx and Δy are threshold values which are related to accuracy of MLE algorithm.

One key point for predicting target's position is to find which subpath the target may move at time k . Some definitions are defined at first as follows:

Definition 1. Optional subpath that target may move on: for any subpath $f(x, y)$ described with set Z , if it is satisfied that $Z \cap D \neq \emptyset$, then this subpath is one optional subpath.

Definition 2. Closest projection point S_i and set S : S_i is the closest projection point of G_i which satisfies the following function (8), S is the set of $\{S_1, S_2, \dots, S_k\}$ whose element number is equal to G 's as

$$S_i = \left\{ \widehat{S}_x \mid \widehat{S}_x = \arg \min |S_x - G_i| \right\}, \quad (8)$$

where $S_x = (x_s, y_s)$ is any point on subpath $f_i(x, y)$. $f_i(x, y)$ is one of all optional subpaths.

The prediction model can be showed as in Figure 4.

Usually, the subpath $f_k(x, y)$ in which S_k is on is the most possible subpath that target may move on. In order to increase the predicting probability, we choose the subpath most of S_{k-p} to S_k are on as the k th subpath that targets moves on. Here p is constant which is determined by experiment. p should be satisfied during time $k - p$ to k ; the distance of target's moving is small. Then, we use the closet projection points

on $f_k(x, y)$ to form a new set $S' = \{S^{(1)}, S^{(2)}, \dots, S^{(k)}\}$. And the prediction problem based on the previous model can be written as follows:

$$\widehat{S}_{k+1} = S^{(k)} + v_k \cdot \Delta T, \quad (9)$$

where \widehat{S}_{k+1} is the position to be predicted, and v_k is the velocity at time k . For the randomness of target moving, the direction of vector v_k is hard to be determined. So, we rewrite it as follows:

$$\llbracket \widehat{S}_{k+1}, S^{(k)} \rrbracket = |v_k| \cdot \Delta T = \Delta S, \quad (10)$$

where $\llbracket \cdot, \cdot \rrbracket$ denotes the shortest distance from one point to another along some certain subpath. So, $\llbracket \widehat{S}_{k+1}, S^{(k)} \rrbracket$ is the shortest distance from $S^{(k)}$ to \widehat{S}_{k+1} along some certain subpath.

Obviously, the optional subpath targets are on at time $k + 1$ is very likely more than one. So, \widehat{S}_{k+1} may have one or multiple solutions. At time $k + 1$, target may still be on subpath $f_k(x, y)$ or turn to another adjacent subpath. Without loss of generality, we assume that $f_{k+1}(x, y)$ is the possible subpath at time $k + 1$. The key point to judge whether $f_{k+1}(x, y)$ exist is to find out whether there is a TP when target is moving ahead during time ΔT .

Let C be the set of all possible TPs that target may encounter, if there is a point C_r in C satisfying (10), then $f_{k+1}(x, y)$ exists as follows:

$$\llbracket C_r, S^{(k)} \rrbracket < \Delta S, \quad (11)$$

where $\llbracket C_r, S^{(k)} \rrbracket$ is the possible shortest distance from $S^{(k)}$ to C_r along the subpath $f_k(x, y)$ which can be obtained by as following:

$$\llbracket C_r, S^{(k)} \rrbracket = \int_{S^{(k)}}^{C_r} f_k(x, y). \quad (12)$$

Then the set of all possible predicting positions at time $k + 1$ can be written as follows:

$$M_{k+1} = \left\{ Q_r \mid \Delta S = \begin{cases} \int_{S^{(k)}}^{C_r} f_k(x, y) + \int_{C_r}^{Q_r} f_{k+1}(x, y), \\ \text{if } \llbracket C_r, S^{(k)} \rrbracket < \Delta S \\ \int_{S^{(k)}}^{Q_r} f_k(x, y), \text{ else} \end{cases} \right\}. \quad (13)$$

Set M_{k+1} contains all possible predicting positions. But the possibility of each element in M_{k+1} to become the final localization result is different. Let \widehat{U}_{k+1} be the localization result using MLE algorithm at time $k + 1$. Generally, \widehat{U}_{k+1} is close to real position with high possibility. The more accurate the MLE is, the higher the possibility will be. And the element in M_{k+1} nearby \widehat{U}_{k+1} has a higher possibility than the other elements. Predicting result in M_{k+1} that owns this feature can be treated as one final result, that is:

$$\widehat{S}_{k+1}^{(a)} = \left\{ M_j \mid \min_{M_j \in M_{k+1}} |M_j - \widehat{U}_{k+1}| \right\}. \quad (14)$$

On the other hand, for the randomness of human moving, different movement patterns may lead to different prediction possibilities. We can infer the next possible positions according to previous locations.

Definition 3. Direction value of $S^{(i)}$: for the i th point $S^{(i)}$ in S' , we use $\delta_{\text{orien}}(S^{(i)} | S^{(i-1)})$ to describe the target's moving direction at time i . If $\delta_{\text{orien}}(S^{(i)} | S^{(i-1)})$ equals to 1, the moving direction of $S^{(i)}$ is forward, otherwise backward. $\delta_{\text{orien}}(S^{(i)} | S^{(i-1)})$ can be calculated with the following:

$$\delta_{\text{orien}}(S^{(i)} | S^{(i-1)}) = \begin{cases} 1, & \text{if } \langle \vec{\psi}, \vec{l} \rangle \geq 0, \text{ where } \vec{\psi} = S^{(i)} - S^{(i-1)}, \\ \vec{l} \left(\frac{\partial f}{\partial x} \Big|_{S^{(i)}}, \frac{\partial f}{\partial y} \Big|_{S^{(i)}} \right), & \\ -1, & \text{else,} \end{cases} \quad (15)$$

where $f = f_k(x, y)$, $i \geq 2$.

So the possibility the target moving forward at time k can be described as follows:

$$P_{(k+1|k)}^{(\text{forward})} = \frac{\text{Num} \cdot \delta^{(+1)}}{m}, \quad (16)$$

where $\text{Num} \cdot \delta^{(+1)}$ is the amount of points whose direction value is 1; m is the amount of elements in S' .

From the previous description, we know that a TP may be encountered when target is moving forward or backward if $f_{k+1}(x, y)$ exists. So, in set M_{k+1} , some elements may reflect the prediction results that target moving forward; we use $n^{(\text{forward})}$ to denote the number of these elements. Others may reflect the prediction results that target moving backward; we use $n^{(\text{backward})}$ to denote the number of these elements. So, we can get another prediction result:

$$\begin{aligned} \widehat{S}_{k+1}^{(b)} = & \sum_{i=1}^{n^{(\text{forward})}} \left(\frac{P_{(k+1|k)}^{(\text{forward})}}{n^{(\text{forward})}} \cdot \widehat{Q}_{k+1}^{(i)} \right) \\ & + \sum_{j=1}^{n^{(\text{backward})}} \left(\frac{1 - P_{(k+1|k)}^{(\text{forward})}}{n^{(\text{backward})}} \cdot \widehat{Q}_{k+1}^{(j)} \right), \end{aligned} \quad (17)$$

where $\widehat{Q}_{k+1}^{(i)}$ is the i th prediction result in set M_{k+1} when target is moving forward, $\widehat{Q}_{k+1}^{(j)}$ is the j th prediction result in set M_{k+1} when backward.

Then the final prediction result can be written as follows:

$$\widehat{S}_{k+1} = \alpha \cdot \widehat{S}_{k+1}^{(a)} + (1 - \alpha) \cdot \widehat{S}_{k+1}^{(b)}, \quad (18)$$

where α is the weight of each predicting result. It can be obtained with some learning methods [20] when doing long-term prediction [21] applications. Generally, the long-term motion trajectory of the target usually comply with limited movement patterns, which is shown as a repetitive motion along one or several paths. In this paper, we only consider short-term predicting, and the value of α is set to 0.5.

3.4. Final Location Computing. The final localization result can be obtained as follows:

$$G_{k+1} = w \cdot \widehat{U}_{k+1} + w' \cdot \widehat{S}_{k+1}. \quad (19)$$

Here we defined w as follows:

$$w = \begin{cases} \frac{|v_{\max}| \cdot \Delta T - \llbracket S^{(k)}, \widehat{S}_{k+1} \rrbracket}{|\widehat{S}_{k+1} - \widehat{U}_{k+1}|}, & \\ 1, & \text{if } |\widehat{U}_{k+1} - S^{(k)}| \geq |v_{\max}| \cdot \Delta T, \\ & \text{else.} \end{cases} \quad (20)$$

3.5. Updating Rules. After getting the localization result at time $k + 1$, some updating rules are proposed for the next location predicting and computing. The updating rule for $|v_{k+1}|$ can be written as follows:

$$\begin{aligned} |v_{k+1}| = & \frac{\llbracket S_{k+1}, S_k \rrbracket^{(f_k(x,y))}}{\Delta T}, \quad \text{if } (|v_{k+1}| < |v_{\max}|), \\ & |v_{k+1}| = |v_{\max}|, \quad \text{else,} \end{aligned} \quad (21)$$

where $\llbracket S_{k+1}, S_k \rrbracket^{(f_k(x,y))}$ is the distance from S_k to S_{k+1} along the subpath $f_k(x, y)$; S_{k+1} is the closest projection point of G_{k+1} which can be obtained by Definition 2. $f_{k+1}(x, y)$ is the subpath that S_{k+1} is on at time $k + 1$.

The update rule for set S is to keep the length of S unchanged, remove the first element, and insert the new element S_{k+1} into the last of S .

3.6. Pseudoprocedure of PPLP Algorithm. Target's final location at time $k + 1$ can be obtained with the following pseudo procedure of PPLP algorithm.

Procedure begin:

//Path-planning begin:

- (1) input JOP and EP of each subpath according to actual indoor environment;
- (2) determine each subpath with function (5);
- (3) determine the total path with function (4); complete path-planning modeling.

//Path-planning end.

- (4) calculate the target's location \widehat{U}_{k+1} at time $k + 1$ using MLE algorithm;

//Predicting begin:

- (5) initialize G , $|v_k|$, $|v_{\max}|$, p , Δx , Δy ;
- (6) calculate set S with the path-planning model and functions (7) and (8);
- (7) calculate M_{k+1} with functions (9)–(13);
- (8) calculated $\widehat{S}_{k+1}^{(a)}$ with function (14);
- (9) calculated $\widehat{S}_{k+1}^{(b)}$ with function (15)–(17);
- (10) calculated \widehat{S}_{k+1} with function (18);

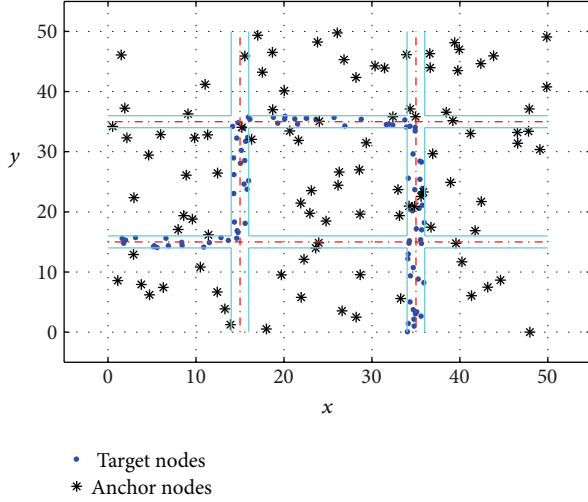


FIGURE 5: The nodes deployment and environment set up.

- (11) calculated the final position G_{k+1} with function (19) and (20).

//Predicting end.

//Updating begin:

- (12) update some with updating rules in Section 3.5.

//Updating end.

Procedure end.

4. Simulation and Analysis

In this section, we will evaluate the performance of the proposed localization algorithm through extensive simulations carried out using MATLAB.

4.1. Simulation Scenario and Settings. We set simulation scenario and some key parameters as follows.

All ANs are randomly deployed in a $50 * 50 \text{ m}^2$ area for the simulation. The total number of ANs is initially 100 and every AN is known its position. All members of G are initialized to $(0, 15)$. Some other parameters are shown in Table 1. The nodes deployment and the environment setup are shown in Figure 5.

In Figure 5, we use 4 dotted-line segments to represent 4 subpaths, respectively. The path width is set to be 2 m. We use some random discrete TNs (as shown in Figure 5 with blue dots) to simulate the randomness of human movement. In the proposed algorithm, we did not consider any particular ranging technique. In the simulation process, we use the following formula (22) [12, 22] to describe the measured distances between TNs and ANs with some certain ranging technique:

$$\widehat{d}_{ij} = d_{ij} + N_{ij}, \quad (22)$$

where \widehat{d}_{ij} and d_{ij} are the measured and real distance between the AN_{*i*} and the TN_{*j*}, respectively, and N_{ij} is assumed to

TABLE 1: Simulation parameters.

Type	Value
P	5
$ \mathbf{v}_k $	0.5
$ \mathbf{v}_{\max} $	5 m/s
T	2 s
ΔT	100 ms
Δx	3 m
Δy	3 m
R	10 m

be blurred by additive Gaussian random variables with zero mean and known variance σ_d^2 .

4.2. Evaluation Metrics. To analyze the simulation results, in this paper, we defined the following two metrics to evaluate the performance of the proposed algorithm.

- (a) Average localization error:

$$\text{err_aver} = \frac{1}{\text{NUM}} \sum_{i=1}^{\text{NUM}} \|X_i - \sigma_i\|, \quad (23)$$

where err_aver is the error mean of localization result which reflects the accuracy of the algorithm. X_i is the real coordinate of the TN_{*i*}; σ is the calculated coordinate of the TN_{*i*} using the proposed localization algorithm. $\|X_i - \sigma_i\|$ represents the localization error of TN_{*i*}. NUM is the number of TNs.

- (b) Standard variance of localization error:

$$\text{Loc_ver} = \sqrt{\frac{1}{\text{NUM}} \sum_{i=1}^{\text{NUM}} (\|X_i - \sigma_i\| - \text{err_aver})^2}, \quad (24)$$

where Loc_ver is the standard variance of localization results which can describe the degree of spread of the localization results. Other variables have the same meanings as metric (a).

- (c) Average distance to the correct subpath:

$$\text{deviate_value_aver} = \frac{1}{\text{NUM}} \sum_{j=1}^{\text{NUM}} \|\chi_j - \sigma_j\|, \quad (25)$$

where $\text{deviate_value_aver}$ is the average distance that the location results of the targets deviated from the correct subpath. χ_j is the closest projection point of the TN_{*j*}, σ_j is the calculate coordinate of TN_{*j*} using proposed algorithm. $\|\chi_j - \sigma_j\|$ represents the distance that TN_{*i*} departed from the correct subpath.

4.3. Simulation Results and Analysis. We firstly simulate 100 ANs to evaluate the performance of the proposed algorithm and the classical MLE algorithm [13–16]. The simulation results are shown in Figures 6 and 7.

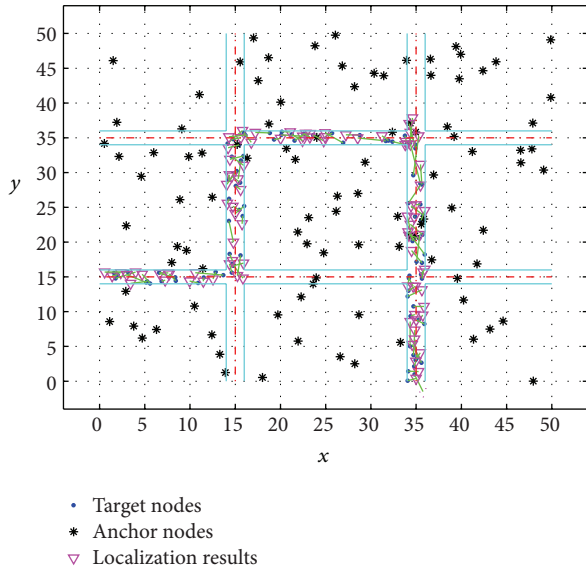


FIGURE 6: The simulation result of the proposed algorithm.

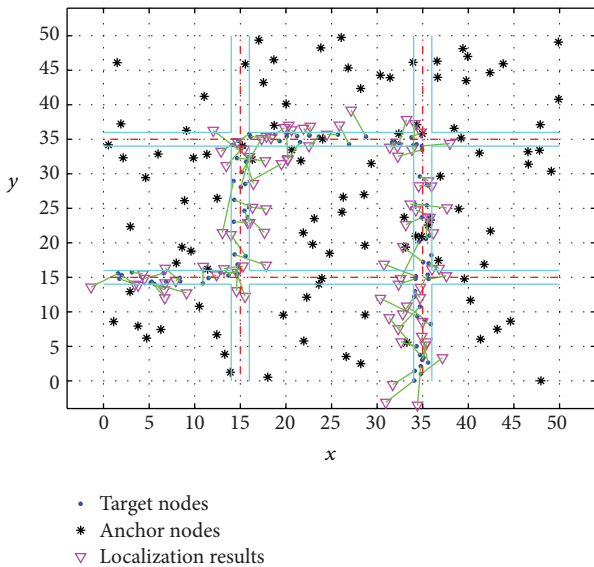


FIGURE 7: The simulation result of MLE algorithm.

Figures 6 and 7 show the simulation results of the proposed algorithm and MLE algorithm when the number of anchor nodes is 100 and transmission range (R) is 10 m. We can see that the performance of the proposed algorithm is better than MLE algorithm. To ease the understanding and analyzing of simulation results, we use average localization error (err_aver), standard variance of localization error (Loc_var), and average distance to the correct subpath ($deviate_value_aver$) as the evaluation metrics to evaluate the performance of these two algorithms. Finally, we get the following comparison results.

Figure 8 provides an intuitive comparison of the accuracy of the proposed localization algorithm and the MLE. The average localization error can be obtained using formula (23).

The result shows that the average localization error of MLE is 2.5621 m while the proposed algorithm is only 1.7624 m. We can see that the proposed localization algorithm has a better accuracy than MLE algorithm. Figure 9 shows the distance that localization results of TNs deviated from the correct trajectory when the target moves along the correct subpath as shown in Figure 5. The average distance to the correct subpath can be obtained using formula (25). The simulation result shows that the average distance to the correct subpath of the proposed algorithm is 0.5175 m, which is much smaller than MLE algorithm. We also calculate the standard variance of localization error with formula (24). The Loc_var of MLE is 1.4318, and the proposed algorithm's is 1.0972. Figure 10 gives the comparison results of these two algorithms with respect to the proposed three metrics. The result shows that the proposed algorithm (PPLP) has better performance in all evaluation metrics than MLE algorithm. The accuracy is high, the localization result is stable and concentrated, and it can always find the right way that the target moves on. This is very useful in some practical applications such as elders/children guarding, hospital patients care, indoor searching, and rescuing for trapped.

In order to further verify the effectiveness of the proposed algorithm, we also did some extensive simulations and compared it with the PSO algorithm [12, 17, 18]. By changing the transmission radius (R) and anchor nodes ratio, we get the following simulation results.

Figure 11 provides a comparison of the accuracy of the proposed localization approach, the MLE algorithm, and the PSO algorithm with respect to anchor nodes' ratio and average connectivity. We run the simulation with 90 TNs. The number of anchor nodes varied from 30 to 100 (as a result the average connectivity increased from 3.99 to 11.73). The simulation results show that the proposed algorithm has a higher accuracy than the other two algorithms. Figure 12 shows the results of these three algorithms with respect to the standard variance of localization error when the number of anchor nodes was changed from 30 to 100. The results show that the standard variance of localization error of the proposed algorithm is lower than the other two algorithms. Figure 13 gives the simulation results of average distance to the correct subpath when the simulation setting is the same as Figure 11. After running at least 100 times simulations, the average distance to the correct subpath can be obtained. As can be seen from Figure 13, it is obvious that the average distance to the correct subpath decreases when anchor nodes' ratio increase. But simulation result of the proposed algorithm changed within a narrow range from 0.51 m to 0.86 m, while the other two algorithms changed obviously. That is to say, the proposed algorithm is more stable than the other two algorithms in indoor environment.

Some more simulation results about the discussed three metrics of these three algorithms can be observed in Figures 14, 15, and 16. We run the simulation with 90 TNs and 100 ANs. The transmission range was increased from 5 m to 15 m (as a result, the average connectivity increased from 3.15 to 16.27). The transmission range of a sensor node varies with its transmission power. A better localization performance is expected with higher transmission range as the number

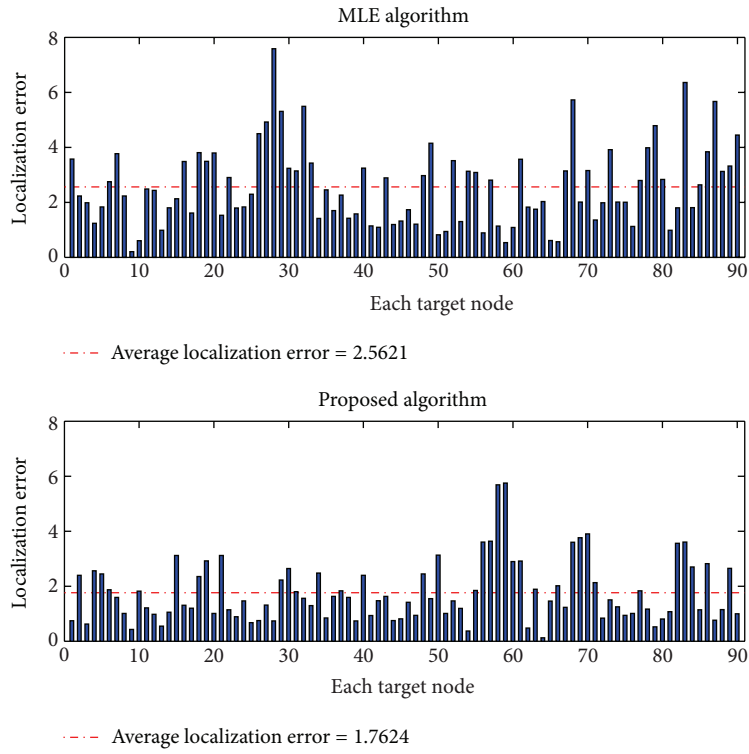


FIGURE 8: Accuracy comparison between the proposed algorithm and MLE.

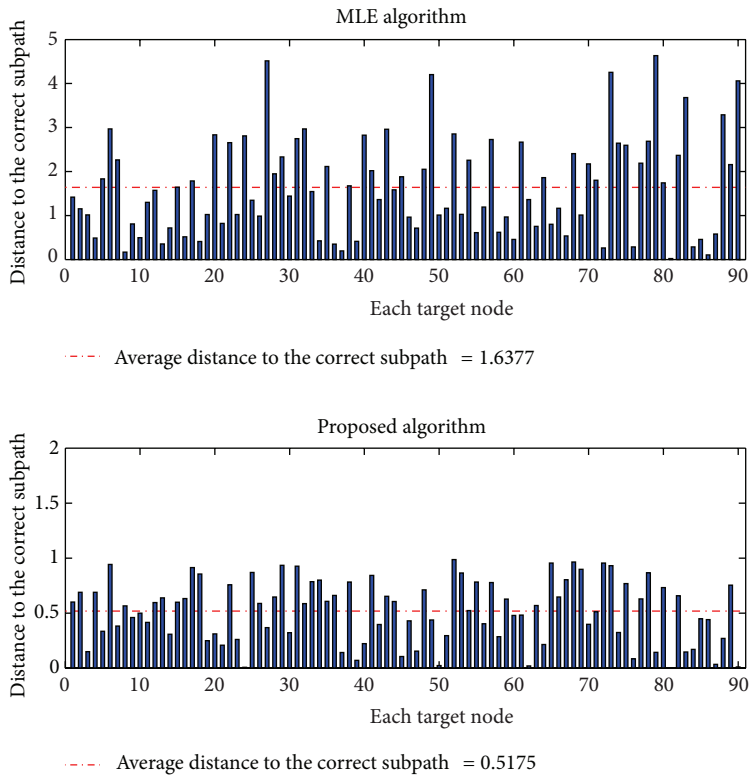


FIGURE 9: Distance to the correct subpath comparison between the proposed algorithm and MLE.

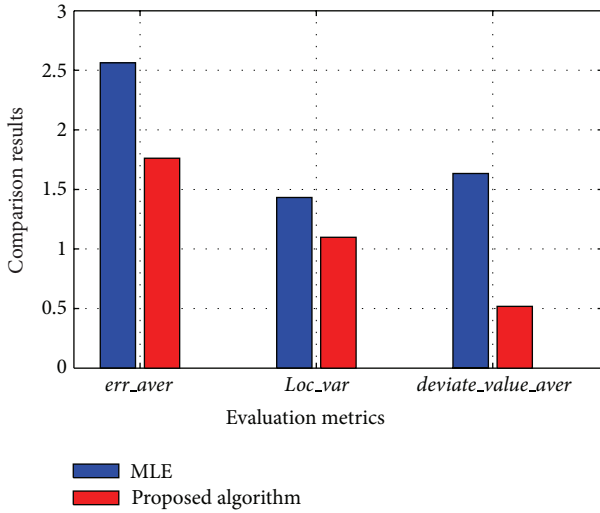


FIGURE 10: Comparison results of MLE and the proposed algorithm with respect to the proposed three evaluation metrics.

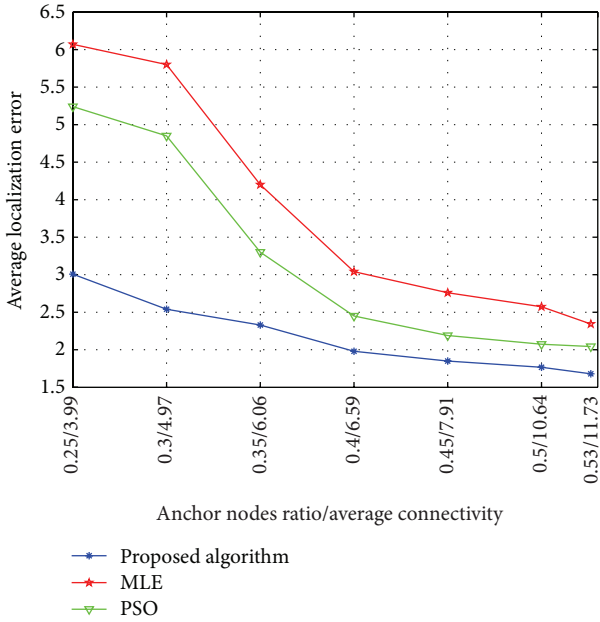


FIGURE 11: Average localization error versus anchor nodes' ratio/Average connectivity.

of one-hop ANs increases [14]. With the increase in transmission range, the average localization error, the standard variance of localization error, and the average distance to the correct subpath decrease. But the decrease scopes of these three metrics are not all obvious when transmission range is larger than 10 m (here the connectivity value is 11.60). That is because the connectivity is an essential factor that affects these algorithms' performance. When connectivity is greater than a certain value (such as 11.60 showed in Figures 14–16), the accuracy of the algorithm is changed little. Before that, connectivity can greatly affect algorithm's performance. However, the proposed algorithm can have an

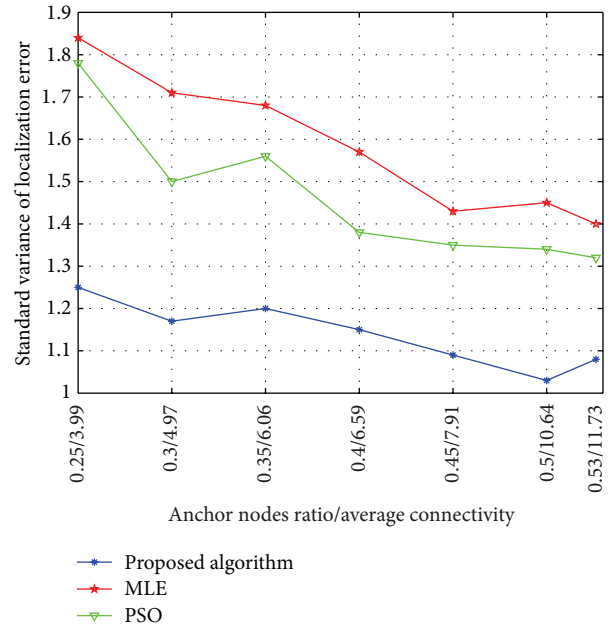


FIGURE 12: Standard variance of localization error versus anchor nodes' ratio/Average connectivity.

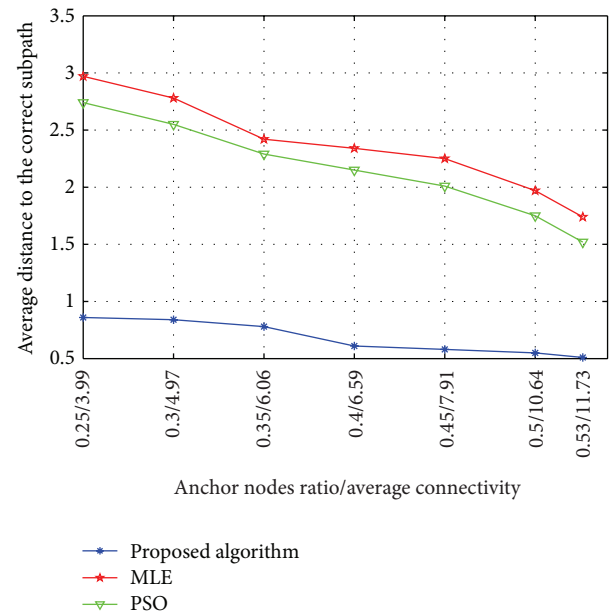


FIGURE 13: Average distance to the correct subpath versus anchor nodes' ratio/Average connectivity.

excellent performance even with low connectivity as showed in Figures 14–16.

All simulation results proved that PPLP algorithm is performed well in indoor environment. Furthermore, PPLP algorithm is a centralized computing method. The location calculation of the target can be done in some certain device with strong processing capacity such as personal computer. The proposed algorithm does not need to calculate large matrix; there is no iteration in localization computing,

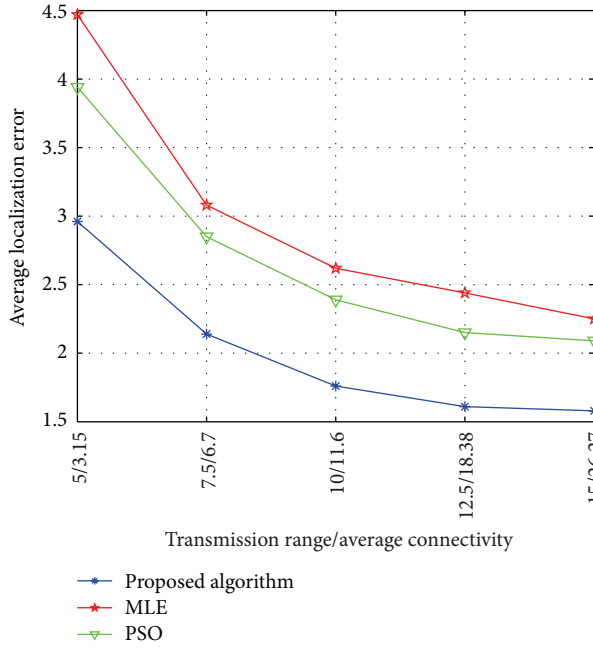


FIGURE 14: Average localization error versus transmission range/average connectivity.

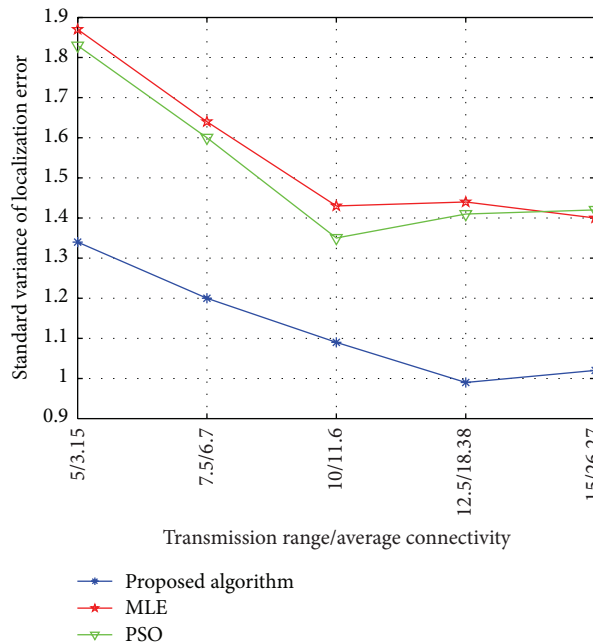


FIGURE 15: Standard variance of localization error versus transmission range/average connectivity.

the localization program is executed sequentially with high efficiency and low complexity.

5. Conclusion

Localization is one of the substantial issues in wireless sensor networks. In this paper, we presented a location predicting method (PPLP) for indoor mobile target localization in WSNs based on path-planning. We first analyzed the

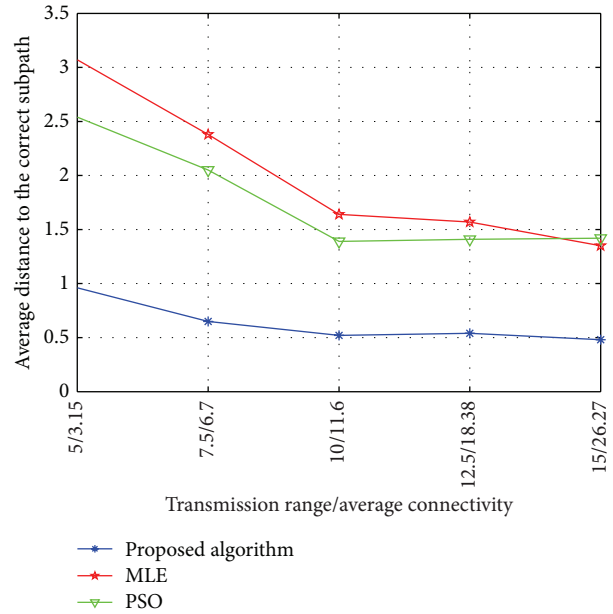


FIGURE 16: Average distance to the correct subpath versus transmission range/average connectivity.

common feature of indoor environment for most buildings and the motion pattern of most targets and established the path-planning model to constrain the movement trajectory of the mobile target according to indoor architectural pattern. Then, we used MLE algorithm to obtain one certain kind of location result of the target. After that, based on the path-planning model and some previous localization results of the target, the best possible position of the target in the next time interval was predicted with the proposed predicting approach. Finally, the MLE result and prediction result were weighted to obtain the final position. In simulation process, we defined three metrics to evaluate the performance of the proposed algorithm and compared it with the MLE algorithm and PSO algorithm. Simulation results showed that the proposed algorithm has a better performance in all these three evaluation indicators and can be very useful for some practical applications such as elders/children guarding, hospital patients care, indoor search, and rescue for trapped.

Acknowledgments

This work was supported by the National Key Technology R&D Program (2011BAK07B03), National Science and Technology Major Project (2009ZX07528-003-09), and the 2011 Internet of Things Development Special Fund.

References

- [1] T. Zhang, J. S. He, and H. Yu, "Secure localization in wireless sensor networks with mobile beacons," *International Journal of Distributed Sensor Networks*, vol. 2012, Article ID 732381, 11 pages, 2012.
- [2] J. Wang, R. K. Ghosh, and S. K. Das, "A survey on sensor localization," *Journal of Control Theory and Applications*, vol. 8, no. 1, pp. 2–11, 2010.

- [3] H. Chen, P. Huang, M. Martins, H. C. So, and K. Sezaki, "Novel centroid localization algorithm for three-dimensional wireless sensor networks," in *Proceedings of the International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM '08)*, pp. 1–4, Dalian, China, October 2008.
- [4] K. Z. Liu, S. Wang, F. Zhang, F. P. Hu, and C. C. Xu, "Efficient localized localization algorithm for wireless sensor networks," in *Proceedings of the 5th International Conference on Computer and Information Technology (CIT '05)*, pp. 517–523, Shanghai, China, September 2005.
- [5] S. Jian, L. Linlan, C. Yubin, and H. Haitao, "A novel three-dimensional localization algorithm in wireless sensor networks," in *Proceedings of the International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM '08)*, pp. 24–29, Dalian, China, October 2008.
- [6] S. Zheng, L. Kai, and Z. H. Zheng, "Three dimensional localization algorithm based on nectar source localization model in wireless sensor network," *Application Research of Computers*, vol. 25, no. 8, pp. 2512–2513, 2008.
- [7] K. Z. Lu, X. H. Xiang, D. Zhang, R. Mao, and Y. H. Feng, "Localization algorithm based on maximum a posteriori in wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2012, Article ID 260302, 7 pages, 2012.
- [8] M. S. Rahman, Y. Park, and K. D. Kim, "RSS-based indoor localization algorithm for wireless sensor network using generalized regression neural network," *Arabian Journal For Science and Engineering*, vol. 34, no. 4, pp. 1043–4053, 2012.
- [9] M. Salamah and E. Doukhnitch, "An efficient algorithm for mobile objects localization," *International Journal of Communication Systems*, vol. 21, no. 3, pp. 301–310, 2008.
- [10] L. Girod and D. Estrin, "Robust range estimation using acoustic and multimodal sensing," in *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, pp. 1312–1320, Maui, Hawaii, USA, October 2001.
- [11] D. Niculescu and B. Nath, "Ad hoc positioning system (APS) using AoA," in *Proceedings of the 22nd Annual Joint Conference on the IEEE Computer and Communications Societies*, vol. 3, pp. 1734–1743, San Francisco, Calif, USA, March 2003.
- [12] A. Gopakumar and L. Jacob, "Performance of some metaheuristic algorithms for localization in wireless sensor networks," *International Journal of Network Management*, vol. 19, no. 5, pp. 355–373, 2009.
- [13] M. M. Noel, P. P. Joshi, and T. C. Jannett, "Improved maximum likelihood estimation of target position in wireless sensor networks using particle swarm optimization," in *Proceedings of the 3rd International Conference on Information Technology: New Generations (ITNG '06)*, pp. 274–278, Las Vegas, Nev, USA, April 2006.
- [14] A. J. Weiss and J. S. Picard, "Maximum likelihood localization of wireless networks using biased range measurements," in *Proceedings of the International Symposium on Communications and Information Technologies*, pp. 865–870, Sydney, Australia, October 2007.
- [15] Y. T. Chan, H. Y. C. Hang, and P. C. Ching, "Exact and approximate maximum likelihood localization algorithms," *IEEE Transactions on Vehicular Technology*, vol. 55, no. 1, pp. 10–16, 2006.
- [16] X. Chen and B. L. Zhang, "Improved DV-Hop node localization algorithm in wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2012, Article ID 213980, 7 pages, 2012.
- [17] J. J. Yao, J. Li, L. Wang, and Y. Ha, "Wireless sensor network localization based on improved particle swarm optimization," in *Proceedings of the International Conference on Computing, Measurement, Control and Sensor Network (CMCSN '12)*, pp. 72–75, Taiyuan, China, July 2012.
- [18] M. Y. Shen, Y. J. Lu, and M. S. Zhao, "Study of node localization algorithm based on improved particle swarm optimization and RSSI for WSNs," in *Informatics in Control, Automation and Robotics*, vol. 2 of *Lecture Notes in Electrical Engineering*, pp. 185–193, 2011.
- [19] H. Li, J. Wang, X. Li, and H. Ma, "Real-time path planning of mobile anchor node in localization for wireless sensor networks," in *Proceedings of the IEEE International Conference on Information and Automation (ICIA '08)*, pp. 384–389, Zhangjiajie, China, June 2008.
- [20] M. Jafari, N. Abdollahi, and H. Mohammadi, "Predicating the location of nodes in ad hoc network by lazy learning method," *Communications in Computer and Information Science*, vol. 241, pp. 336–345, 2011.
- [21] M. J. Jahromi, A. I. Maswood, and K. J. Tseng, "Long term prediction of tidal currents," *IEEE Systems Journal*, vol. 5, no. 2, pp. 146–155, 2011.
- [22] E. Niewiadomska-Szynkiewicz and M. Marks, "Optimization schemes for wireless sensor network localization," *International Journal of Applied Mathematics and Computer Science*, vol. 19, no. 2, pp. 291–302, 2009.

