Indoor Mobile Target Localization Based on Path-planning and Prediction in Wireless Sensor Networks

PENG GAO, WEI-REN SHI, HONG-BING LI AND WEI ZHOU
Department of College of Automation,
Chongqing University,
Chongqing China 400030,
Jallenwill12@yahoo.cn

Abstract: Node position information is one of the important issues in many wireless sensor networks usages. In this paper, based on path-planning and prediction, an indoor mobile target localization algorithm (PPIMT) 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.

Keywords: Wireless sensor networks, Localization, Path-planning, Prediction.

1. Introduction

Wireless sensor networks (WSNs) has been broadly discussed and studied in recent years [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 locations 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 [9, 10], TDOA [11], AoA [12,13]. 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 [14]. Without improve ranging estimation or add some other information related to localization, the accuracy of the current range-based algorithms can’t be improved obviously.

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 PPIMT. 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 [15-18]. In the localization process, the number of multiple measurement equations is usually more than the number of variables. Set r_i (i=1,2,.,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:

$$\mathbf{X} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b} \tag{1}$$

where

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

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

2.2. Particle swarm optimization for localization

Particle Swarm Optimization (PSO) [19, 20] is a swarm bionic optimization algorithm, which models the behavior of flocks of birds and fish. Its process does not depend upon the quality of the objective function, and then converge to the most optimal solution in a larger probability. So it is commonly used to solve the optimization problems.

Let $\mathbf{x}_i$ = (x_{i1}, x_{i2}) be the *2*-dimensional vector representing the position of the *i*-th particle in the swarm, $\mathbf{g}$ = [g_1, g_2] the position vector of the best particle in the swarm, $\mathbf{p}$ = [p_{i1}, p_{i2}] the position vector of the *i*-th particle's personal best and $\mathbf{v}_i$ = [v_{i1}, v_{i2}] the velocity of the *i*-th particle. The particles evolve according to the following equations:

$$\begin{cases} v_{id} = \omega v_{id} + c_1 r_1 (p_{id} - x_{id}) + c_2 r_2 (g_d - x_{id}) \\ x_{id} = x_{id} + v_{id} \end{cases} \tag{2}$$

where d = 1, 2; i = 1, 2, . . . , K; and 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; whereas 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].

[20] proposed an improved PSO algorithm with RSSI self-correcting localization algorithm for wireless sensor networks. Based on the RSSI ranging, the author combined with the proposed RSSI self-correction mechanism and an improved PSO algorithm optimize the nodes localization for WSNs.[14] 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 [21]. 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. Fig. 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 time.

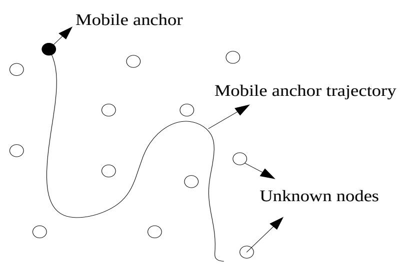

Fig.1. A mobile anchor assisting in the localization

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 [22]. As fig.2 shows:

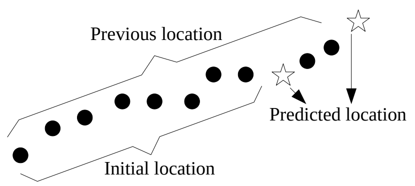

Fig.2. Prediction method for WSNs localization

M. Salamah and E. Doukhnitch [10] 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

Indoor localization of WSNs has been a hot research topic for the last several years. Due to the randomness of target's moving and the complicated indoor environment, it is very different to locate indoor mobile target. In this paper, we proposed an indoor mobile target localization algorithm based on path-planning and prediction (PPIMT algorithm) in WSNs. 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 two metrics to evaluate the performance of the proposed algorithm and compared with the MLE algorithm and PSO algorithm with these two evaluation indicators. Simulation results demonstrate that the proposed algorithm performed better than the other two algorithms.

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 sub-paths. The target can move freely among various rooms. After encountering some turning point, the target may change or not change its motion path. The position of the target can be calculated periodically 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. People's indoor movement will be limited by the indoor architecture 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.

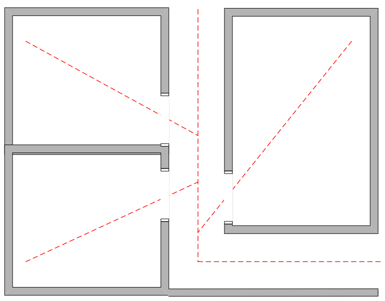

Fig.3. An indoor architectural pattern with some indoor paths

As fig.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 in equation (3):

$$F(x,y)=\begin{cases} f_1(x,y) \\ f_2(x,y) \\ \vdots \\ f_m(x,y) \end{cases},x\in X;y\in Y \tag{3}$$

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

However, different buildings have different indoor architectural patterns. In order to make location computing more effectively, we use a straight line segment function to describe each sub-path:

$$(x_b-x_a)y=(y_b-y_a)x+(x_by_a-y_bx_a) \tag{4}$$

$$s.t.\ \begin{cases} \min[x_a,x_b]\le x\le \max[x_a,x_b] \\ \min[y_a,y_b]\le y\le \max[y_a,y_b] \end{cases}$$

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

3.3. Location predicting and computing

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, compute localization results during a period of time T using some certain localization method (such as MLE); second, predicting the next possible positions according 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 compete the first step. We only focused on step two and step three.

3.3.1. Location predicting

Let us use set G={G_1, G_2, …, 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: how to get the next position $\hat{\mathbf{G}}_{\mathbf{k+1}}$ according to set G and the path-planning model.

For any sub-path f(x,y), a set Z can be use to describe all points on this sub-path. Each element of set Z satisfied function (4). We can also get that all elements of G are belong to set D which can be described as:

$$D=\left\{(x_D,y_D)\left|\begin{cases}x_{\min}-\Delta x\le x_D\le x_{\max}+\Delta x\\ y_{\min}-\Delta y\le y_D\le y_{\max}+\Delta y\end{cases}\right.\right\} \qquad (5)$$

Where $x_{\min}$, $y_{\min}$, $x_{\max}$, $y_{\max}$ 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 value which related to accuracy of MLE algorithm.

One key point for predicting target's position is to find which sub-path the target may move on at time k. Some definitions are defined at first:

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

Definition 2. Closest projection point S_i and set S: S_i is Closest projection point of G_i which satisfy the following function (8), S is the set of { S_1, …, S_k } whose element number is equal to G's.

$$\mathbf{S_i}=\{\hat{\mathbf{S}}_\mathbf{x}\left|\hat{\mathbf{S}}_\mathbf{x}=\arg\min|\mathbf{S_x}-\mathbf{G_i}|\right.\} \qquad (6)$$

where $\mathbf{S_x}=(x_s,y_s)$ is any point on sub-path f_i(x,y) which is satisfied $f_i(x_s, y_s) = 0$. f_i(x,y) is one of all optional sub-paths.

The prediction model can be showed as fig.4.

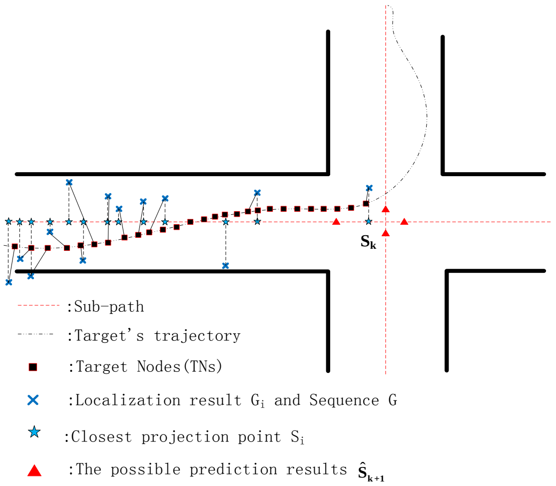

Fig.4. Prediction model in indoor environment

Usually, the sub-path f_k(x,y) that S_k is on is the most possible sub-path that target may move on. In order to increase the predicting probability, we choose the sub-path that most of S_{k-t} to S_k are on as the k-th sub-path that target moves on. Here t is constant which is determined by experiment. t should be satisfied that during time k-t 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'={ $S^{(1)}$,$S^{(2)}$,….$S^{(k)}$}. And the prediction problem based on the previous model can be written as:

$$\hat{\mathbf{S}}_{\mathbf{k+1}}=\mathbf{S}^{(k)}+\mathbf{v_k}\cdot\Delta T \qquad (7)$$

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

$$\left[\!\left[\hat{\mathbf{S}}_{\mathbf{k+1}},\mathbf{S}^{(k)}\right]\!\right]=|\mathbf{v_k}|\cdot\Delta T=\Delta S \qquad (8)$$

where $[\![\bullet,\bullet]\!]$ denotes the shortest distance from one point to anther along some certain sub-path, So $[\![\hat{\mathbf{S}}_{\mathbf{k+1}},\mathbf{S}^{(k)}]\!]$ is the shortest distance from $S^{(k)}$ to $\hat{\mathbf{S}}_{\mathbf{k+1}}$ along some certain sub-path.

Obviously, the optional sub-path that target moving on at time k+1 is very likely more than one. So $\hat{\mathbf{S}}_{\mathbf{k+1}}$ may have one or multiple solutions. At time k+1, target may still on sub-path f_k(x,y) or turn to another adjacent sub-path. Without loss of generality, we assume f_{k+1}(x, y) is the possible sub-path at time k+1. The key point to judge whether f_{k+1}(x, y) is exist is to find out whether there is a TP when target 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 satisfy equation (12), then f_{k+1}(x,y) is exist.

$$\left[\!\left[\mathbf{C}_r,\mathbf{S}^{(k)}\right]\!\right]<\Delta S \qquad (9)$$

where $[\![\mathbf{C}_r,\mathbf{S}^{(k)}]\!]$ is the possible shortest distance from $\mathbf{S}^{(k)}$ To C_r along the sub-path f_k (x,y) which can be obtained by:

$$[\![\mathbf{C}_r, \mathbf{S}^{(k)}]\!] = \int_{s^{(k)}}^{C_r} f_k(x,y) \tag{10}$$

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

$$M_{k+1} = \left\{ Q_{\mathbf{r}} \middle| \Delta S = \begin{cases} \int_{\mathbf{S}^{(k)}}^{\mathbf{C}_r} f_k(x,y) + \int_{\mathbf{C}_r}^{Q_r} f_{k+1}(x,y), if\, [\![\mathbf{C}_r, \mathbf{S}^{(k)}]\!] < \Delta S \\ \int_{\mathbf{S}^{(k)}}^{Q_r} f_k(x,y), else \end{cases} \right\} \tag{11}$$

Set M_{k+1} contains all possible predicting positions. But the possibility of each element in M_{k+1} becomes the final localization result is different. Let $\hat{\mathbf{U}}_{\mathbf{k+1}}$ be the localization result using MLE algorithm at time k+1. Generally, $\hat{\mathbf{U}}_{\mathbf{k+1}}$ is close to real position with high possibility. The more accuracy the MLE is, the higher the possibility will be. And the element in M_{k+1} nearby $\hat{\mathbf{U}}_{\mathbf{k+1}}$ 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:

$$\hat{\mathbf{S}}_{k+1}^{(a)} = \{ \mathbf{M_j} \mid \min_{\mathbf{M_j} \in M_{k+1}} \left| \mathbf{M_j} - \hat{\mathbf{U}}_{\mathbf{k+1}} \right| \} \tag{12}$$

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

$$\delta_{orien}(\mathbf{S^{(i)}} \mid \mathbf{S^{(i-1)}}) = \begin{cases} 1, if\, < \vec{\psi}, \vec{l} >= 0, where, \vec{\psi} = \mathbf{S^{(i)}} - \mathbf{S^{(i-1)}}, \vec{l} = (\left.\frac{\partial f}{\partial x}\right|_{\mathbf{S}^{(i)}}, \left.\frac{\partial f}{\partial y}\right|_{\mathbf{S}^{(i)}}) \\ -1, else \end{cases} \tag{13}$$

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

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

$$p_{(k+1|k)}^{(forward)} = \frac{Num\_\delta^{(+1)}}{m} \tag{14}$$

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

From the previous description, we known that a TP may be encountered when target 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 $n^{(forward)}$ to denote the number of these elements. Others may reflect the prediction results that target moving backward, we use $n^{(backward)}$ to denote the number of these elements. So we can get another prediction result:

$$\hat{\mathbf{S}}_{\mathbf{k+1}}^{(b)} = \sum_{i=1}^{n^{(forward)}} \left(\frac{p_{(k+1|k)}^{(forward)}}{n^{(forward)}} \cdot \hat{Q}_{\mathbf{k+1}}^{(\mathbf{i})} \right) + \sum_{j=1}^{n^{(backward)}} \left(\frac{1 - p_{(k+1|k)}^{(forward)}}{n^{(backward)}} \cdot \hat{Q}_{\mathbf{k+1}}^{(\mathbf{j})} \right) \tag{15}$$

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

Then the final predication result can be written as:

$$\hat{\mathbf{S}}_{\mathbf{k+1}} = \alpha \cdot \hat{\mathbf{S}}_{k+1}^{(a)} + (1-\alpha) \cdot \hat{\mathbf{S}}_{k+1}^{(b)} \tag{16}$$

where α is the weight of each predicting result. It can be obtained with some learning methods [23] when doing long-term prediction [24] applications. Generally, the long-term motion trajectory of the target usually comply with limited movement patterns, which is shown as 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.

3.3.2 Final localization computing

The final localization result can be obtained as:

$$\mathbf{G}_{\mathbf{k+1}} = w \cdot \hat{\mathbf{U}}_{\mathbf{k+1}} + w' \cdot \hat{\mathbf{S}}_{\mathbf{k+1}} \tag{17}$$

Here we defined w as:

$$w = \begin{cases} \dfrac{\left| \left|\mathbf{v}_{\max}\right| \cdot \Delta T - [\![\mathbf{S}^{(k)}, \hat{\mathbf{S}}_{k+1}]\!] \right|}{\left| \hat{\mathbf{S}}_{k+1} - \hat{\mathbf{U}}_{k+1} \right|}, if \left| \hat{\mathbf{U}}_{k+1} - \mathbf{S}^{(k)} \right| \geq \left|\mathbf{v}_{\max}\right| \cdot \Delta T \\ 1, else \end{cases} \tag{18}$$

After getting the localization result at time k+1, some updating rules are proposed for the localization computing. The updating rule for $|\mathbf{v}_{k+1}|$ can be written as:

$$\begin{cases} \left|\mathbf{v}_{k+1}\right| = \dfrac{[\![\mathbf{S}_{k+1}, \mathbf{S}_{k}]\!]^{(f_k(x,y))}}{\Delta T}, if \left(\left|\mathbf{v}_{k+1}\right| < \left|\mathbf{v}_{\max}\right| \right) \\ \left|\mathbf{v}_{k+1}\right| = \left|\mathbf{v}_{\max}\right|, else \end{cases} \tag{19}$$

where $[\![\mathbf{S}_{k+1}, \mathbf{S}_{k}]\!]^{(f_k(x,y))}$ is the distance from S_k to S_{k+1} along the sub-path $f_k(x,y)$; S_{k+1} is the closest projection point of $\mathbf{G}_{k+1}$ which can be obtained by definition 2. $f_{k+1}(x,y)$ is the sub-path that $\mathbf{S}_{k+1}$ is on at time k+1. The update rule for set S is: Keep the length of S unchanged, remove the first element, insert the new element $\mathbf{S}_{k+1}$ into the last of S.

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*50m^2 area for the simulation. The total number of ANs is initially 100 and every AN known its position. The initial value of $|\mathbf{v}_k|$ is 1 m/s. $|\mathbf{v}_{\max}|$ is set to 5m/s. The transmission range (R) of all nodes is set to the same and initialized to 10m. T is set to 2s and is set to 100ms. So the length (k) of G is 20. t is set to 5. All members of G are initialized to (0, 15). Δx and Δy are both initialized to 3m. The nodes deployment and the environment set up are shown in fig.5.

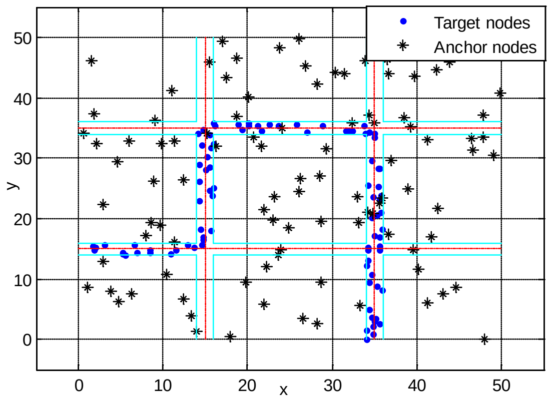

Fig.5. The nodes deployment and environment set up

In fig.5, we use 4 dotted line segments to represent 4 sub-paths respectively. The path width is set to be 2m. We use some random discrete TNs (as shown in fig.5 with blue dots) to simulate the randomness of human movement. In the proposed algorithm we did not considered any particular ranging technique. In the simulation process, we use the following formula (20) [14 25] to describe the measured distances between TNs and ANs with some certain ranging technique:

$$\hat{d}_{ij} = d_{ij} + N_{ij} \tag{20}$$

where $\hat{d}_{ij}$ and d_{ij} are the measured and real distance between the AN_i and the TN_j, respectively; N_{ij} is assumed to 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.

(1) Average localization error

$$err\_aver = \frac{1}{NUM} \sum_{i=1}^{NUM} \left\| \mathbf{X}_i - \boldsymbol{\sigma}_i \right\| \tag{21}$$

where *err_aver* is average localization error reflecting the accuracy of the algorithm. **X** is the ture coordinate of the TN_i, $\boldsymbol{\sigma}$ is the calculated coordinate of the TN_i using the proposed localization algorithm. $\|\mathbf{X}_i - \boldsymbol{\sigma}_i\|$ represents the localization error of TN_i. *NUM* is the number of TNs. The smaller the *err_aver*, the better performance the algorithm.

(2) Average distance to the correct sub-path

$$deviate\_value\_aver = \frac{1}{NUM}\sum_{j=1}^{NUM}\|\boldsymbol{\chi}_j - \boldsymbol{\sigma}_j\| \qquad (22)$$

where $deviate\_value\_aver$ is the average distance that the location results of the targets deviated from the correct sub-path. The smaller the $deviate\_value$, the better performance the algorithm. $\boldsymbol{\chi}_j$ is the closest projection point of the TN_j, $\boldsymbol{\sigma}_j$ is the calculate coordinate of TN_j using proposed algorithm localization algorithm. $\|\boldsymbol{\chi}_j - \boldsymbol{\sigma}_j\|$ represents the distance that TN_i departed from the correct sub-path .

4.3. Simulation results and analysis

We firstly simulate with 100 ANs to evaluate the performance of the proposed algorithm and the classical MLE algorithm [15-18]. The simulation results are shown in fig.6 and fig.7.

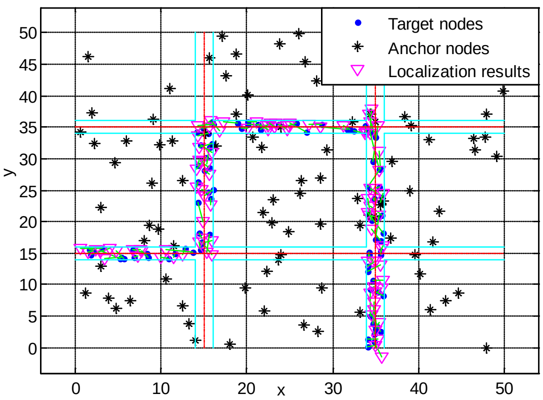

Fig.6. The simulation result of the proposed algorithm

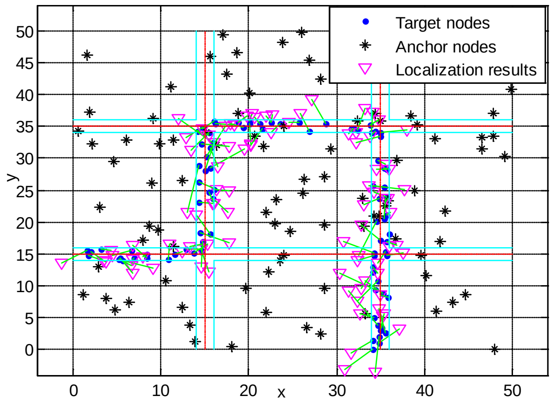

Fig.7. The simulation result of MLE algorithm

Fig.6 and Fig.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 10m, respectively. 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 and average distance to the correct sub-path as the evaluation metric to evaluate the performance of these two algorithms. Finally we get the following comparison results:

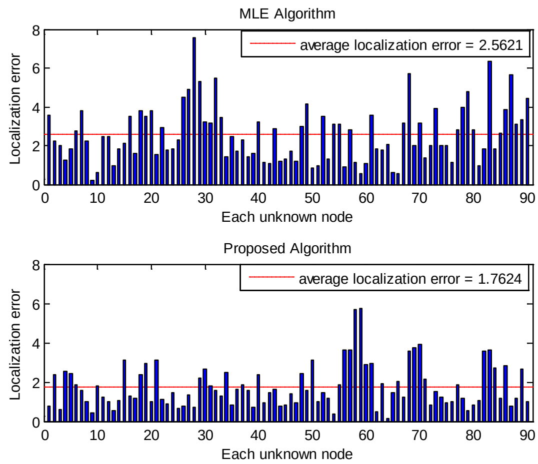

Fig.8. Accuracy comparison between the proposed algorithm and MLE.

Fig.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 (21). The results show that the average localization error of MLE is 2.5621m while the proposed algorithm is only 1.7624m. We can see that the proposed localization algorithm has a better accuracy than MLE algorithm.

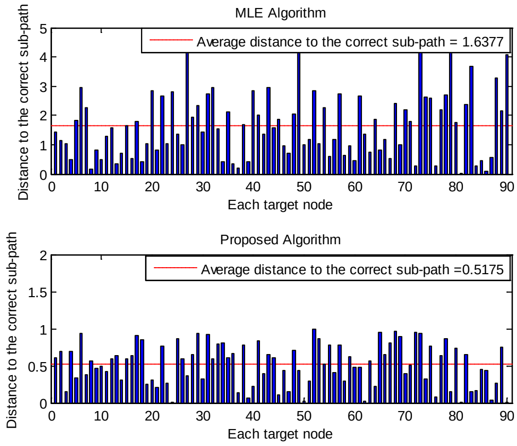

Fig.9. Distance to the correct sub-path comparison between the proposed algorithm and MLE.

Fig.9 shows the distance that localization results of TNs deviated from the correct trajectory when the target moves along the correct sub-path as shown in fig.5. The average distance to the correct sub-path can be obtained using formula (22). The simulation results show that the average distance to the correct sub-path of the proposed algorithm is 0.5175m, which is much smaller than MLE algorithm. That is to say, even in the case of poor positioning accuracy (sometimes even up to 6m with the proposed algorithm as fig.8 shows), We can still find the right sub-path the target is on. This is very useful in some practical applications such as elders/children guarding, hospital patients care, indoor searching and rescuing for trapped and so on.

In order to further verify the effectiveness of the proposed algorithm, we also did some extensive simulations, and compared it with the PSO algorithm [14, 19, 20]. By changing the transmission radius, anchor nodes ratio, we get the following simulation results.

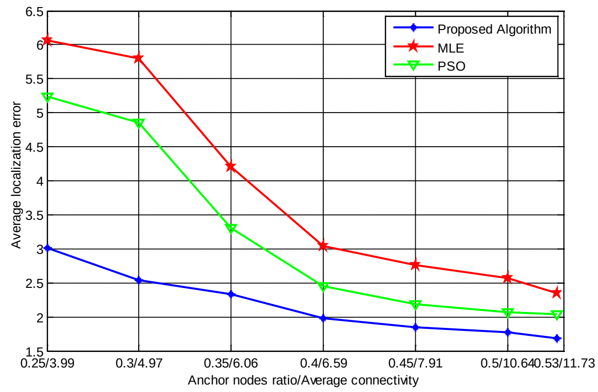

Fig.10. The average localization error vs. Anchor nodes ratio/Average connectivity.

Fig.10 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, and the number of anchor nodes varying 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.

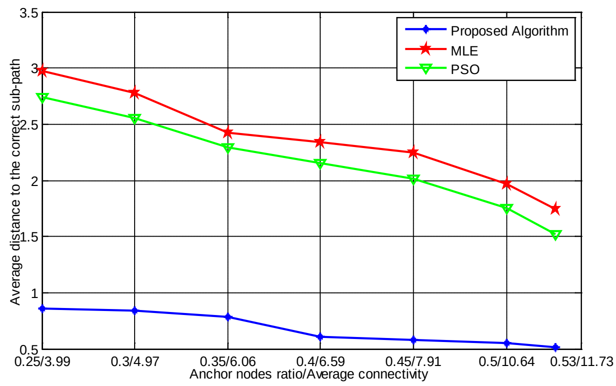

Fig.11. The average distance to the correct sub-path vs. Anchor nodes ratio/Average connectivity.

Fig.11 gives the simulation results of average distance to the correct sub-path at the same simulation settings as Fig.10. After running at least 100 times simulation, the average distance to the correct sub-path can be obtained. As can be seen from fig.11, it is obvious that the average distance to the correct sub-path decrease when anchor nodes ratio increase. But simulation result of the proposed algorithm changed within narrow range from 0.51m to 0.86m, while the other two algorithms decreased obviously.

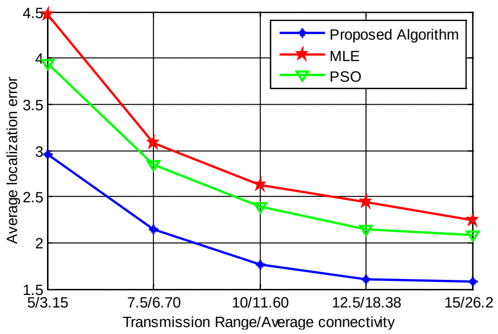

Fig.12. The average localization error vs. Transmission range /Average connectivity.

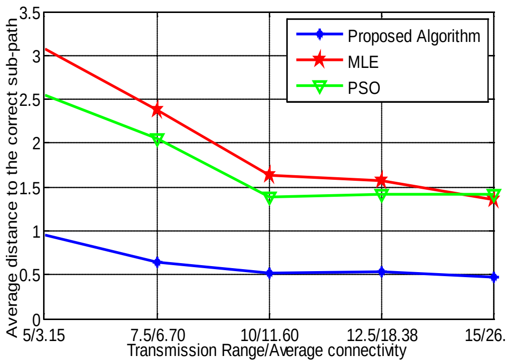

Fig.13. The average distance to the correct sub-path vs. Transmission range /Average connectivity.

Fig.12 and Fig.13 shows the simulation results of average localization error and average distance to the correct sub-path with using MLE, PSO and the proposed algorithm, respectively. We run the simulation with 90 TNs and 100 ANs, and the transmission range increasing from 5m to 15m (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 of one-hop ANs increases [14]. As the increase of the transmission range, the average localization error and average distance to the correct sub-path both decreased, but not obviously when transmission is larger than 10m (the connectivity value is 11.60). In this case, the essential factor to improve accuracy is the improvement of the connectivity, because the connectivity of TNs also increases when transmission range increases. The accuracy of almost all algorithms is not obviously improved when connectivity is greater than a certain value (such as 11.60 in fig.12 and fig.13). And when the connectivity does not reach the value, there will be a great influence on the accuracy of the algorithms. However, the proposed algorithm can have an excellent performance even with low connectivity. It can remain in a narrow range in both the two evaluation indicators. The simulation results show that the proposed algorithm is much better than the other two algorithms in both localization accuracy and average distance to the correct sub-path.

5. Conclusion

Localization is one of the substantial issues in wireless sensor networks. In this paper, we presented an indoor mobile target localization algorithm for wireless sensor networks based path-planning and prediction. We first analyzed the 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 two metrics, average localization error and average distance to the correct sub-path, to evaluate the performance of the proposed algorithm and compared with the MLE algorithm and PSO algorithm with these two evaluation indicators. Simulation results showed that the proposed algorithm has a better performance in both these two 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 and so on.

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] Shih T.-F. and Chang W.-T., 'Hierarchical Localization Strategy for Wireless Sensor Networks', *WSEAS Transactions on Computers* , Vol. 7, No.8, 2008, pp. 1260-1269.

[2] Wang J., Ghosh R.K. and Das S.K. 'A survey on sensor localization', *Journal of Control Theory and Applications*, Vol. 8, 2010, pp. 2-11.

[3] Chen H.Y., Huang P., Martins M., So H.C. and Sezaki K. 'Novel centroid localization algorithm for three-dimensional wireless sensor networks', *Proceedings of the International Conference on Wireless Communications, Networking and Mobile Computing*, Dalian, China, 2008, pp. 1-4.

[4] Liu K.Z., Wang S., Zhang F., Hu F.P. and Xu C.C. 'Efficient localized localization algorithm for wireless sensor networks', *Proceedings of the 5th international conference on computer and information technology* , Shanghai, China, 2005, pp. 517–523.

[5] Shu J., Liu L. and Chen Y. 'A novel three-dimensional localization algorithm in wireless sensor networks, wireless communications, networking and mobile computing', *Proceedings of the International Conference on Wireless Communications, Networking and Mobile Computing*, Dalian, China, 2008, pp. 24-29.

[6] Zheng S., Kai L. and Zheng Z.H. 'Three dimensional localization algorithm based on nectar source localization model in wireless sensor network', *Application Research of Computers*, Vol.25, No.8, 2008, pp. 2512–2513.

[7] Qiu T., Zhou Y. , Xia F. Jin N.G. and Feng L. 'A localization strategy based on *n*-times trilateral centroid with weight', *International Journal of Communication System*s, Vol. 25, 2012, pp. 1160-1177.

[8] Rahman M.S., Park Y. and Kim K.-D. 'RSS-Based Indoor Localization Algorithm for Wireless Sensor Network Using Generalized Regression Neural Network', *Arabian Journal for Science and Engineering*, Vol. 34, No. 4 , 2012, pp.1043-4053.

[9] Harter A., Hopper A., Steggles P., Ward A. and Ward P. 'The anatomy of a context-aware application', *Wireless Networks*, Vol. 8, 2002, pp. 187-197.

[10] Salamah M. and Doukhnitch E. 'An efficient algorithm for mobile objects localization', *International Journal of Communication Systems*, Vol. 21, 2008, pp. 301-310.

[11] Girod L. and Estrin D. 'Robust range estimation using acoustic and multimodal sensing', *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, Maui, HI, United states, 2001, pp. 1312–1320.

[12] Niculescu D. and Nath B. 'Ad hoc positioning system (APS) using AoA', *Proceedings of the 22nd Annual Joint Conference on the IEEE Computer and Communications Societies*, San Francisco, CA, United states, Vol. 3, 2003, pp. 1734–1743.

[13]Sharawi M.S. and Aloi D.N. 'Characterizing the performance of single-channel Pseudo-Doppler direction finding systems at 915MHz for vehicle localization', *International Journal of Communication System*s, Vol. 24, 2011, pp. 27-39.

[14] Gopakumar A. and Jacob L. (2009) 'Performance of some metaheuristic algorithms for localization in wireless sensor networks', *International Journal of Distributed Sensor Networks*, Vol. 19, 2009, pp. 355-373.

[15] Noel M.M., Joshi P.P. and Jannett T.C. 'Improved maximum likelihood estimation of target position in wireless sensor networks using particle swarm optimization', *Proceedings of the 3rd International Conference on Information Technology: New Generations*, Las Vegas, NV, USA, 2006, pp.

274-279.

[16] Weiss A.J. and Picard J.S. 'Maximum likelihood localization of wireless networks using biased range measurements', *Proceedings of International Symposium on Communications and Information Technologies*, Sydney, Australia, 2007, pp. 865-870.

[17] Chan Y.-T., Hang H.Y.C. and Ching P.-C. 'Exact and approximate maximum likelihood localization algorithms', *IEEE Transactions on Vehicular Technology*, Vol. 55, No. 1, 2006, pp. 10-16.

[18] Neri A. and Jacovitti G. (2004) 'Maximum likelihood localization of 2-D patterns in the Gauss-Laguerre transform domain: Theoretic framework and preliminary results', *IEEE Transactions on Image Processing*, Vol. 13, No. 1, 2004, pp. 72-86.

[19] Yao J.J., Li J., Wang L. and Ha Y. 'Wireless sensor network localization based on improved particle swarm optimization', *Proceedings of International Conference on Computing, Measurement, Control and Sensor Network*, Taiyuan, China, 2012, pp. 72-75.

[20] Shen M.-Y., Lu Y.-J. and Zhao M.-S. 'Study of node localization algorithm based on improved particle swarm optimization and RSSI for WSNs', *Lecture Notes in Electrical Engineering*, Vol. 2, 2011, pp. 185-193.

[21] Li H.J., Wang J.W., Li X. and Ma X.X. 'Real-time path planning of mobile anchor node in localization for wireless sensor networks', *Proceedings of the 2008 IEEE International Conference on Information and Automation*, Zhangjiajie, China, June, 2008, pp. 384-389.

[22] Jafari M., Abdollahi N. and Mohammadi H. 'Predicating the Location of Nodes in Ad Hoc Network by Lazy Learning Method', *Proceedings of the 1st International Conference on Innovative Computing Technology*, Tehran, Iran, December, 2011, pp. 336-345.

[23] Jafari M., Abdollahi N. and Mohammadi H. 'Predicating the location of nodes in ad hoc network by lazy learning method', *Communications in Computer and Information Science*, Vol. 241 CCIS, 2011, pp. 336-345.

[24] Jahromi M.J., Maswood A.I. and Tseng K.-J. 'Long term prediction of tidal currents', IEEE Systems Journal, Vol. 5, 2011, pp. 146-155.

[25] Niewiadomska-Szynkiewicz E. and Marks M. 'Optimization schemes for wireless sensor network localization', *International Journal of Applied Mathematics and Computer Science*, Vol. 19, 2009, pp. 291-302.