A Range-Based Monte Carlo Box Algorithm for Mobile Nodes Localization in WSNs

Dan Li[1], Xianbin Wen[2]
[1]School of Computer Science and Technology, Tianjin University
Tianjin 300072 - China
[e-mail: luyaocyan@163.com]
[2]Key Laboratory of Computer Vision and System, Ministry of Education, Tianjin University of Technology
Tianjin 300384 - China
[e-mail: wenxianbin@163.com]
*Corresponding author: Dan Li

Received October 22, 2016; revised February 6, 2017; accepted May 1, 2017; published August 31, 2017

Abstract

Fast and accurate localization of randomly deployed nodes is required by many applications in wireless sensor networks (WSNs). However, mobile nodes localization in WSNs is more difficult than static nodes localization since the nodes mobility brings more data. In this paper, we propose a Range-based Monte Carlo Box (RMCB) algorithm, which builds upon the Monte Carlo Localization Boxed (MCB) algorithm to improve the localization accuracy. This algorithm utilizes Received Signal Strength Indication (RSSI) ranging technique to build a sample box and adds a preset error coefficient in sampling and filtering phase to increase the success rate of sampling and accuracy of valid samples. Moreover, simplified Particle Swarm Optimization (sPSO) algorithm is introduced to generate new samples and avoid constantly repeated sampling and filtering process. Simulation results denote that our proposed RMCB algorithm can reduce the location error by 24%, 14% and 14% on average compared to MCB, Range-based Monte Carlo Localization (RMCL) and RSSI Motion Prediction MCB (RMMCB) algorithm respectively and are suitable for high precision required positioning scenes.

Keywords: Monte Carlo, mobile, localization, range-based, wireless sensor network

1. Introduction

Wireless sensor networks (WSNs) are kind of networks with distributed autonomous nodes that can sense their environment cooperatively which have been used in a wide range of applications in recent years such as military target tracking and surveillance, medical treatments, hazardous environment exploration, the monitoring of animal activity and so on. However, the basic assumption of these applications is that all nodes in WSNs know their own location because without location information all the other information would be meaningless. Therefore, node localization algorithms in WSNs have important theoretical and practical significance.

Localization methods in WSNs can be roughly divided into two types: range-based and range-free. The range-based algorithms use some ranging techniques such as Time of Arrival [1] (TOA), Angle of Arrival [2] (AOA), Time Difference of Arrival [3] (TDOA) and Received Signal Strength Indication [4] (RSSI) to measure distances or angles between target nodes and their neighboring anchors. Moreover, it provides more precise localization results than the range-free algorithms, which only use the connectivity between sensor nodes without any ranging techniques to estimate the locations of unknown nodes. The typical range-free algorithms mainly include the Centroid [5], the convex position estimation [6] (CPE), the approximate point in triangulation [7] (APIT), the distance vector-hop [8] (DV-hop) and the multidimensional scaling-MAP [9] (MDS-MAP). In particular, the MDS-MAP generates the most accurate positioning results among range-free algorithms. However, this algorithm has a computational overhead of $O(n^3)$ in a network of n nodes and not suitable for large-scale WSNs. It also yields significantly larger localization error if there are holes in the sensor field. Therefore, Shon, et al proposed a cluster-based MDS [10] (CMDS) that overcomes the above shortcomings and yields smaller localization error in all environments even with holes.

A wide variety of localization algorithms have been proposed in static WSNs, which provide good performance. However, the growing interest in mobile devices and the emergence of new applications such as mobile targets tracking and monitoring in mobile environments require us to develop new localization methods. To improve the accuracy of mobile targets tracking, some estimation and prediction algorithms such as Bayesian [11] and Kalman filtering [12-13] have been proposed by scholars. However, these methods bring complicated calculation and increasing overhead of information transmission. Tran-Quang, et al [14] proposed a collaborative data processing and dynamic clustering method, combining with the Lateration-localizing algorithm to solve the mobile target tracking problem. It achieved good balance of energy consumption, delay, accuracy and improved the practicability of mobile target tracking method.

Mobile nodes localization in WSNs is more difficult than static nodes localization since the nodes mobility brings more data. The information of neighboring anchors for each node and the topology of the whole network change over time. Although static nodes localization algorithms in WSNs can be extended to mobile nodes localization problems, they will generate high communication and computation costs, thus bringing more energy consumption. Nevertheless, the positioning accuracy is not satisfactory. In order to improve the positioning precision by exploiting the mobility of sensor nodes, Hu and Evans [15] applied the sequential Monte Carlo (SMC) method to the mobile nodes localization in WSNs for the first time. Their algorithm which is named Monte Carlo Localization (MCL) divided the time into discrete intervals. The location of mobile nodes had to be constantly updated in each time interval by rerunning the algorithm. This algorithm includes two phases: prediction phase and filtering phase. Firstly, each target node randomly draws a set of samples in the sampling area to represent its possible locations in the prediction phase. Secondly, invalid samples which are not inside the transmission range of anchor nodes are filtered out in the filtering phase. Finally, the estimated location of the target node is the average of all valid samples. Although the MCL algorithm has disadvantages such as the localization error with low anchor density is large, it provides a new way to solve the localization problem in mobile WSNs.

From then on, more and more scholars developed their own improved scheme on the basis of the MCL algorithm. Baggio and Langendoen [16] proposed the Monte Carlo Localization Boxed (MCB) algorithm which built upon the MCL algorithm to improve the success rate of sampling by defining an anchor box and sample box. Different from the MCL algorithm using information about one-hop/two-hop anchors in the filtering phase only, the MCB algorithm applied the anchor information to constrain the area from which the samples are drawn in the prediction phase also. The Mobile and Static sensor network Localization (MSL) and the MSL* algorithms proposed in [17] were faster in converge and sampling procedure than the MCL algorithm. They improved the localization accuracy by using the estimated location of all neighboring nodes (not only neighboring anchors). Since the algorithms generate more communication and computation costs, they are more suitable for sensors that can support extra communication. A distributed Improved Monte Carlo Localization [18] (IMCL) scheme was proposed to improve the localization accuracy by adding neighboring constraint and moving direction constraint which is suitable for application on the resource-limited sensor nodes. The above algorithms are all range-free and do not meet the requirement of high precision positioning. Literature [19] presented a Range-based Monte Carlo Localization (RMCL) algorithm which improved the localization accuracy and efficiency by introducing the RSSI ranging technique to restrict the sampling area. A new filtering method in [20] which divided the transmission range into n evenly spaced concentric circles strengthened the filter conditions. An overview of the localization algorithms in mobile WSNs was proposed in [21] which researched the principles and characteristics of existing work in this field and classified the important algorithms proposed recently. Moreover, it pointed out the future directions and challenges of the localization problem in mobile WSNs.

The remaining part of this paper is organized as follows. The next section briefly introduces some related works. In section 3, we propose our Range-based Monte Carlo Box (RMCB) algorithm for nodes localization in mobile WSNs. Simulation experiments and results analysis are presented in section 4. Section 5 concludes this paper and points out the future work that lies ahead.

2. Related Work

2.1 RSSI Ranging Model

While the anchor node moves in the sensing field, it can broadcast its position and the RSSI value to any other node within its transmission range since it has a radio frequency chip and needs no additional hardware. The RSSI value increases when the anchor node comes close and decreases when the anchor node moves away. Multi-path, obstacles, diffraction and other factors make the transmission model in actual application environments very complicated. In this paper, we use a logarithmic normal distribution model [22] and the RSSI value at a distance d from the transmitter is given by formula (1).

$$RSSI(d) = P_T - P_L(d_0) - 10\alpha \log_{10} \frac{d}{d_0} + X_\sigma \tag{1}$$

where P_T is the transmission power, $P_L(d_0)$ is the path loss at a reference distance d_0 and α is the path loss exponent. The random variation in RSSI is modeled as a Gaussian random variable $X_\sigma = N(0, \sigma^2)$. The values of α and σ can be set differently depending on the actual propagation environment.

2.2 Monte Carlo Localization Boxed (MCB) Algorithm

Let's briefly introduce the MCB algorithm [16] which includes two phases: the prediction phase and the filtering phase. Firstly, each target node randomly draws a set of samples in the sample box (sometimes is anchor box) to represent its possible locations in the prediction phase. Secondly, invalid samples, which are not inside the transmission range of anchor nodes are filtered out in the filtering phase. Finally, the estimated location of the target node is the average of all valid samples. The specific process of the MCB algorithm is as follows.

Initialiaztion: At the beginning, a node has no knowledge of its location. N is a constant that denotes the maximum number of samples to maintain in a set. L_0 is the initial set of samples, Box_0 is the initial anchor box, O_0 is the initial set of observations, x_{range} and y_{range} are the maximum x and y coordinates of the deployment area, respectively.

if $O_0=\phi$ then
  $Box_0=\{(0, x_{range}); (0, y_{range})\}$
  $L_0=\{ l_0^0, l_0^1, \ldots, l_0^N \}$ *//Set of N random locations in deployment area*
else
  $Box_0=\{(x_{min}, x_{max}); (y_{min}, y_{max})\}$ *//Anchor box built from one-hop and two-hop anchors*
  $L_0=\{ l_0^0, l_0^1, \ldots, l_0^N \}$ *//Set of N random locations within the anchor box Box_0 filtered with O_0*
Fi

Step: Compute a new possible location set L_t based on both L_{t-1}, the set of possible locations from the previous time step t-1 and the new observations O_t, the position information obtained from both the one-hop and two-hop anchors between time t-1 and time t.

if $O_t=\phi$ then $Box_t=\{(0, x_{range}); (0, y_{range})\}$
else $Box_t=\{(x_{min}, x_{max}); (y_{min}, y_{max})\}$ *//Anchor box building*
fi
$L_t=\phi$
while (size(L_t)<N) do
  foreach $l_{t-1}^i \in L_{t-1}$ with $1 \le i \le N$
    $Box_t^i=\{(x_{min}^i, x_{max}^i); (y_{min}^i, y_{max}^i)\}$ for l_{t-1}^i *//Sample box building*
    $R=\{ l_t^i \mid l_t^i$ is selected within Box_t^i with $p(l_t \mid l_{t-1}^i)>0\}$ *//Prediction*
    $R_{filtered}=\{l_t^i \mid l_t^i$ where $l_t^i \in R$ and $p(O_t \mid l_t^i)>0\}$ *//Filtering*
    L_t=choose($L_t \cup R_{filtered}$, N)
  done
done

Location estimate$=\frac{1}{N}\sum_{i=1}^{N} l_t^i$

Anchor box building:

$Box_t=\{(x_{min}, x_{max}); (y_{min}, y_{max})\}$ with (x_j , y_j) being the coordinates of the considered anchor j and n being the total number of anchors heard. We replace r by $2r$ in the following formulas when using the two-hop anchors:

$$x_{\min} = \max_{j=1}^{n}(x_j - r),\ x_{\max} = \min_{j=1}^{n}(x_j + r),\ y_{\min} = \max_{j=1}^{n}(y_j - r),\ y_{\max} = \min_{j=1}^{n}(y_j + r)$$

Sample box building:

$Box_t=\{(x^i_{min}, x^i_{max}); (y^i_{min}, y^i_{max})\}$ with (x^i_{t-1}, y^i_{t-1}) being the coordinates of the old sample l^i_{t-1} in t-1 moment:

$$x^i_{\min} = \max(x_{\min}, x^i_{t-1} - v_{\max}),\ x^i_{\max} = \min(x_{\max}, x^i_{t-1} + v_{\max})$$

$$y^i_{\min} = \max(y_{\min}, y^i_{t-1} - v_{\max}),\ y^i_{\max} = \min(y_{\max}, y^i_{t-1} + v_{\max})$$

Prediction:

$p(l_t|l^i_{t-1})=1$ if $x^i_{min}\le x^i_t\le x^i_{max}$ and $y^i_{min}\le y^i_t\le y^i_{max}$; 0 otherwise, assumes a node is equally likely to move in any direction with any speed between 0 and $v_{\max}$ within Box^i_t.

Filtering:

$p(o_t|l^i_t)=1$ if $\forall s\in S, d(l_t, s) \le r \wedge \forall s\in T, r<d(l_t, s)\le 2r$; 0 otherwise, where r is the radio range, S is the set of one-hop anchors and T is the set of two-hop anchors, $d(l_t, s)$ is the Euclidean distance between the anchor s and the sample l_t.

3. Range-based Monte Carlo Box (RMCB) Algorithm

In this section, a Range-based Monte Carlo Box (RMCB) algorithm, which builds upon the MCB algorithm, is proposed. Although the MCB algorithm has reached a certain precision requirements, its localization result is not as accuracy as those range-based methods. However, those algorithms only use ranging methods such as RSSI also have the following problems: (1) One target node needs at least three anchor nodes that are expensive within its transmission range to localize itself. It is difficult to meet the requirement of quantity. (2) Even if the anchor node number is enough, it is possible that the RSSI value cannot be received resulting in the failure of localization due to obstructions. Our proposed algorithm combines the ranging method with range-free MCB method to solve the above problems. It not only improved the precision of MCB algorithm, but also relaxed the requirement of anchor node number. The specific content of our algorithm is described in the following five subsections.

3.1 Sampling Phase

The method used for constraining the area from which we draws samples is as follows. A target node that has heard anchors (one-hop or two-hop anchors) builds a box that covers the region where the anchors' transmission ranges overlap. In other words, this box is the region of the deployment area where the node is localized. We call such a box an anchor box [16]. When building the anchor box, the original MCB algorithm uses the transmission range R/2R for one-hop/two-hop anchors. In our RMCB algorithm, the RSSI value are utilized to estimate the distances between the target node and its one-hop/two-hop anchors.

Fig. 1 shows the method of building an anchor box based on RSSI. The shaded area in this figure denotes the range of the anchor box.

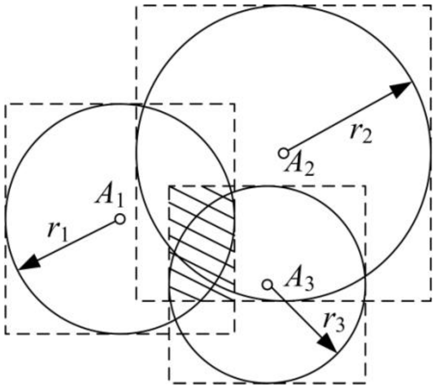

Fig. 1. Anchor box based on RSSI and preset error coefficient

The range of the anchor box can be expressed as ([x_{min}, x_{max}], [y_{min}, y_{max}]). The value of x_{min}, x_{max}, y_{min}, y_{max} is calculated by formula (2) and (3).

$$x_{\min} = \max_{j=1}^{M}(x_j - r_j),\ x_{\max} = \min_{j=1}^{M}(x_j + r_j),\ y_{\min} = \max_{j=1}^{M}(y_j - r_j),\ y_{\max} = \min_{j=1}^{M}(y_j + r_j) \tag{2}$$

$$r_j = \begin{cases} \min((1+\delta)d_j, R), & for\ one-hop\ anchors \\ \min(D_j, 2R), & for\ two-hop\ anchors \end{cases} \tag{3}$$

Where M is the number of anchor nodes, (x_j, y_j) denotes the coordinate of anchor A_j (j=1,2,…,M), R is the transmission range, d_j is the measure distance to a one-hop anchor by RSSI, D_j is the approximate distance to a two-hop anchor (distance to a one-hop node plus distance from a one-hop node to a two-hop anchor). δ is a preset error coefficient since the RSSI ranging model has a certain degree of error.

The sample box is built with an additional constraint: for each old sample l_{t-1}^{i} from the sample set L_{t-1}, an additional square of size $2*v_{\max}$ centered at the old sample is added.

Fig. 2 shows the method of building a sample box. The shaded area in this figure denotes the range of the sample box.

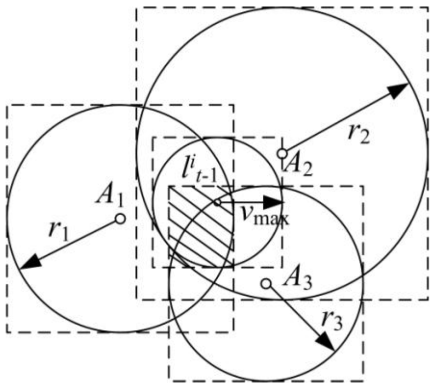

Fig. 2. Sample box

The range of the sample box in t moment for the old sample l_{t-1}^i with coordinates (x_{t-1}^i, y_{t-1}^i) can be expressed as ([x_{min}, x_{max}], [y_{min}, y_{max}]) (i=1, 2,…, N). The value of x_{min}^i, x_{max}^i, y_{min}^i, y_{max}^i is calculated by formula (4) and (5).

$$x_{\min}^i = \max(x_{\min}, x_{t-1}^i - v_{\max}),\ x_{\max}^i = \min(x_{\max}, x_{t-1}^i + v_{\max}) \tag{4}$$

$$y_{\min}^i = \max(y_{\min}, y_{t-1}^i - v_{\max}),\ y_{\max}^i = \min(y_{\max}, y_{t-1}^i + v_{\max}) \tag{5}$$

Where N is the sample number and $v_{\max}$ is the maximum velocity of all mobile nodes. In particular, when there is no old sample set (like the initial time) or no overlap between the anchor box and the area per old sample can move in one time interval at maximum, the anchor box is directly considered as the sample box. When a node has an old sample set but can hear no anchor, we build the sample box only based on the node maximum velocity and the positions of old samples.

3.2 Filtering Phase

The purpose of the filtering phase is to remove the impossible samples from the sample set. The original MCB algorithm uses the transmission range R/2R for one-hop/two-hop anchors as the filter condition. In our RMCB algorithm, the RSSI value are employed to form the new filter condition.

Fig. 3 shows the new filter condition in our proposed algorithm. The shaded area in this figure denotes the effective sample area. In this figure, the target node has two one-hop anchors A_1 and A_2 which distance measured by RSSI between itself and the target node is d_1 and d_2 respectively. Since the RSSI ranging model has a certain degree of error, we designed a preset error coefficient δ to limit the range of error which can be expressed as (1-δ)*d_1/d_2 to min((1+δ)*d_1/d_2, R).

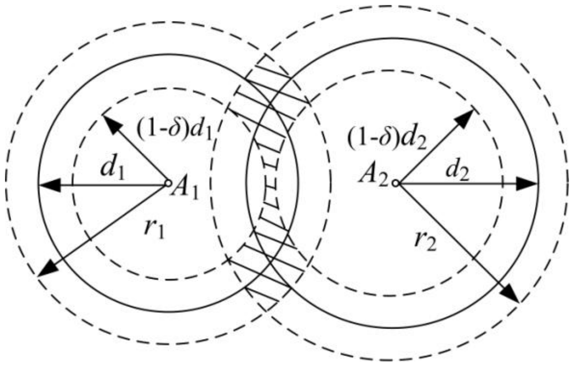

Fig. 3. Filter condition

For two-hop anchors, the distance between itself and the target node is the sum of two one-hop distance measured by RSSI and always larger than the actual distance. Therefore, the filter condition for two-hop anchors is not as strict as for one-hop anchors. The specific condition in the filtering phase is shown in formula (6).

$$filter(l_t^i) = \begin{cases} \forall s_j \in S,\ (1-\delta)d_j \le d(l_t^i, s_j) \le r_j \\ \forall s_j \in T,\ R \le d(l_t^i, s_j) \le r_j \end{cases} \tag{6}$$

Where S is the set of one-hop anchors and T is the set of two-hop anchors, $d(l_t^i, s_j)$ is the Euclidean distance between the sample l_t^i and the anchor s_j, r_j is defined in formula (3).

3.3 Generating New Samples

Because of the strict sampling and filtering process, the quality of samples is much higher than before. However, the sample number is hard to meet the requirement. Constantly repeated sampling and filtering will increase time cost and energy consumption. Here we introduce the simplified Particle Swarm Optimization (sPSO) algorithm [23] to generate new samples. We select the new samples which objective function (7) value is less than a certain threshold and remove those which are unqualified. This process is completed when the sample set is full or the iteration number reaches the preset maximum number.

1) Objective function

In each iteration, we only select the new samples which the objective function (7) value is less than a certain threshold into the next iteration.

$$F(x_t^i, y_t^i) = \frac{1}{M}\sum_{j=1}^{M}\omega_j\left|\sqrt{\left(x_t^i - x_j\right)^2 + \left(y_t^i - y_j\right)^2} - d_j\left(or\ D_j\right)\right| \tag{7}$$

Where M is the number of anchor nodes, (x_j, y_j) $(j=1,2,\ldots,M)$ is the coordinate of anchor A_j, the coordinate of ith sample l_t^i in t moment is (x_t^i, y_t^i) $(i=1, 2,\ldots, N)$. d_j (or D_j) is the distance between the one-hop (or two-hop) anchor node A_j and sample l_t^i.

2) The value of ω_j

In objective function (7), for one-hop anchors, ω_j is 1; for two-hop anchors, the distance D_j between itself and the target node is the sum of two one-hop measure distance by RSSI and always larger than the actual distance, so ω_j is not appropriate to be a constant. Therefore, we associate the value of ω_j with the estimated position E_j calculated after the sampling and filtering phases. The value of ω_j is given by formula (8) and (9).

When the target node can hear both one-hop and two-hop anchors:

$$\omega_j = \begin{cases} 1 & & for\ one-hop\ anchors \\ 0.8 & D_j - E_j \le 0.2 * D_j & \\ 0.2 & D_j - E_j \ge 0.5 * D_j & for\ two-hop\ anchors \\ 0.5 & otherwise & \end{cases} \tag{8}$$

When the target node can hear only two-hop anchors:

$$\omega_j = \begin{cases} 1 & D_j - E_j \le 0.2 * D_j \\ 0.2 & D_j - E_j \ge 0.5 * D_j \\ 0.6 & otherwise \end{cases} \tag{9}$$

3) Simplified Particle Swarm Optimization

Particle Swarm Optimization (PSO) is a popular bio-inspired stochastic global search algorithm proposed by Kennedy and Eberhart [24] that models the social behavior of a flock of birds. In this algorithm, all individuals in a population are seen as particles in a multi-dimensional solution space. First of all, randomly initialize a group of particles in a population. Each of them is a feasible solution and its fitness value is determined by its position in the search space. Each particle moves in the solution space towards the randomly weighted average of the historical personal best position and the historical global best position, and finds the current global solution. In this paper, we utilize the simplified Particle Swarm Optimization (sPSO) algorithm without a velocity term as described in formula (10) for simplicity to generate new samples and choose those that meet the objective function (7) as valid samples to join the next iteration.

$$X_i^{k+1} = X_i^k + c_1 \times rand_1 \times \left(pbest_i - X_i^k\right) + c_2 \times rand_2 \times \left(gbest - X_i^k\right) \tag{10}$$

Where X_i^k (i=1, 2,…, N) is the coordinate of ith sample l_t^i in t moment in kth iteration. X_i^{k+1} is the new sample coordinate after the kth iteration, $pbest_i$ is the historical personal best position of sample i where it had the smallest fitness, and $gbest$ is the global best position of all samples, c_1 and c_2 are acceleration constants, $rand_1$ and $rand_2$ are random numbers uniformly distributed in [0, 1].

3.4 The Standard of Using Anchor Nodes

As we know, more anchor nodes bring more information meanwhile one-hop anchors provide more accurate location information than two-hop anchors. However, the excessive use of anchor nodes may cause sampling dried up and waste of nodes' energy. Therefore, in this paper, we use anchor nodes according to the following standards:

1) The maximum number of anchor nodes we use to localize a target node is four.
2) When the number of anchor nodes is less than four, we use all of them.
3) When the number of one-hop anchors is more than four, we do not use two-hop anchors any more.
4) Priority in use of nearest one-hop or two-hop anchor nodes.

3.5 Algorithm Steps

1) Choose anchor nodes that we use to localize target nodes as described in Section 3.4.
2) Build sample box and then randomly draw N samples in the sample box.
3) Filter out those samples which do not meet the filter condition (6).
4) Generate new samples as described in Section 3.3 till the sample set is full or iteration number reaches the preset maximum number.
5) Calculate the average coordinates of all samples as the final estimated position:

$$(x_t, y_t) = \frac{1}{N}\sum_{i=1}^{N}(x_t^i, y_t^i) \tag{11}$$

6) In particular, when the target node has no anchor around but successful localized in last moment, repeat step 2) ~ 3) to obtain valid sample set and then estimated the position of target nodes as described in step 5), otherwise the localization fails.

Fig. 4 shows the flowchart of our proposed algorithm more clarity.

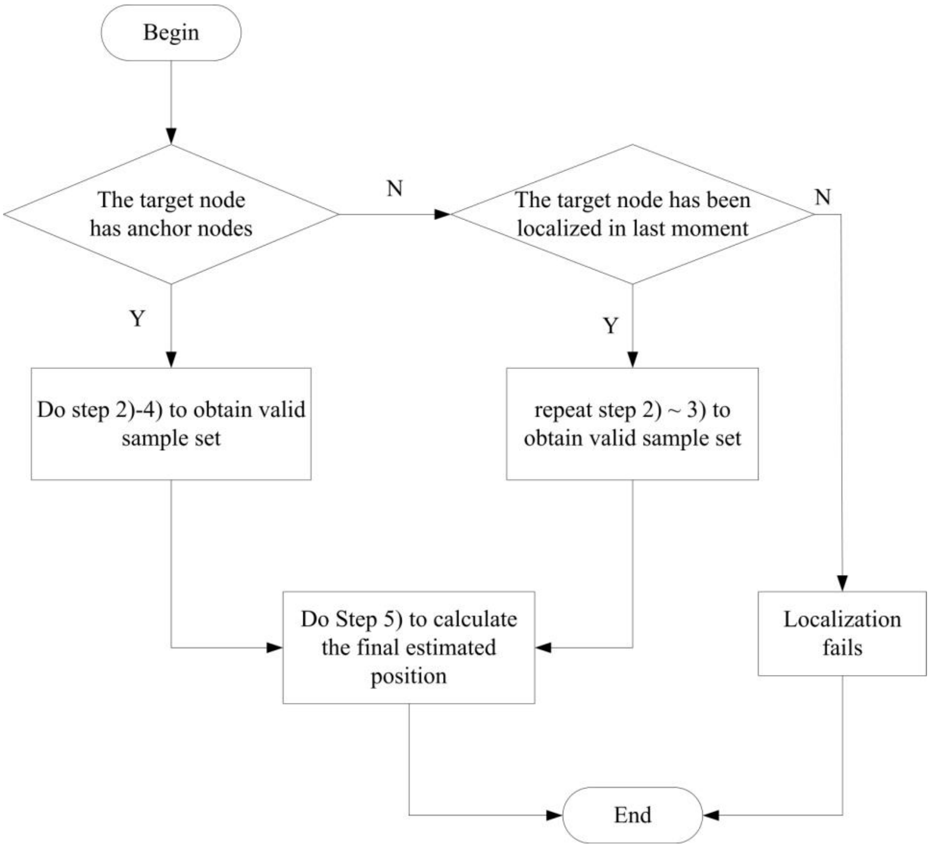

Fig. 4. Algorithm flowchart

4. Simulation and Analysis

4.1 Simulation Parameters

The RMCL method [19] introduce the RSSI value only in the process of building the sample box, the filter condition is exactly the same as the MCL algorithm. The RMMCB method [20] which divided the transmission range into n evenly spaced concentric circles strengthens the filter conditions by using the ranging information. However, the re-sampling method is too rough and the frequent use of motion prediction model to get weighted samples will cause an energy burden of sensor nodes. In our simulation, we compare the MCB, RMCL, RMMCB and our proposed RMCB algorithm under the same experimental setup. Simulation was carried out in MATLAB environment and the results are the average of 20 independent experiments. The parameters of the node localization algorithm are set as follows:

- In a 300×300 square units sensor field, 300 sensor nodes which have the same transmission range R=30 units are deployed randomly. Among them, there are 50 anchor nodes knowing their location information, others are the target nodes that need to be positioned.
- we assume all nodes move in the sensor field according to the random waypoint mobile model [25] (RWP). The maximum velocity v_{max} of all nodes is 10 units/s, each node can vary its velocity less than v_{max} at each time step before it reaches its destination.
- The maximum number of samples N for each target node is 50.

● Since the RSSI ranging model has a certain degree of error, we assume the measured distance is the actual distance d plus a Gaussian additive white noise which can be expressed as $d \cdot (1 \pm P_n/100)$, P_n=5 is the percentage noise.

● Some special parameters in our RMCB algorithm: preset error coefficient δ=0.1, acceleration constants c_1=c_2=2.0, the threshold of objective function is 0.1R, the maximum iteration number of sPSO algorithm is 30.

4.2 Localization Error

Fig. 5 shows the localization error of four different localization approaches at the previous 50 time steps. In mobile WSNs, the localization error of a node (the distance between its real position and estimated position) is changed over time duo to the mobility of the nodes. Accuracy of all algorithms are improved as time goes on and gradually stable since Monte Carlo methods utilize previous position information to localize current nodes. However, the localization error of our proposed RMCB algorithm is always lower than the MCB, RMCL and RMMCB algorithm by about 24%, 14% and 14% respectively on average.

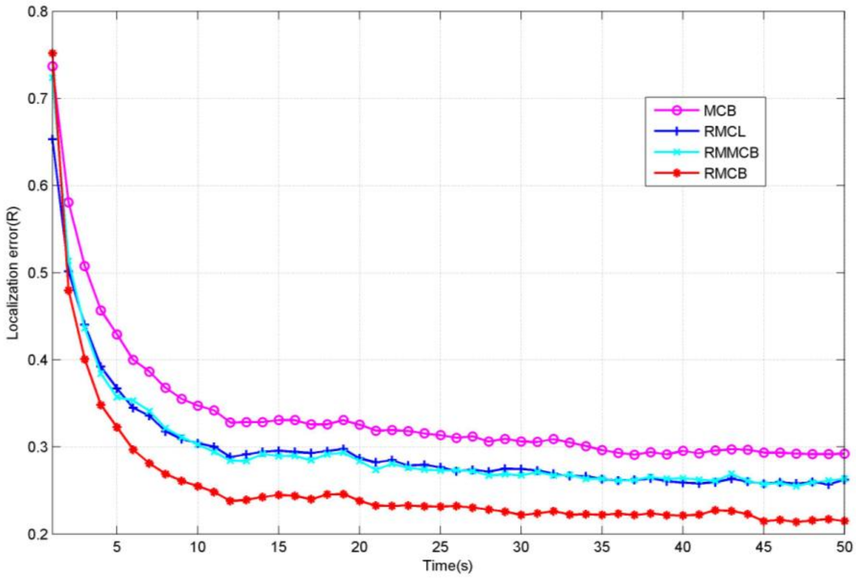

Fig. 5. Localization error with time steps

4.3 The Effect of Maximum Velocity

When we build the sample box, the maximum velocity v_{max} is used to limit the sample location. As the value of v_{max} increases, the target node can hear more anchors to localize itself more accurately. In the mean time, the prediction area of the samples is larger which leads to low precision. Therefore, the maximum velocity has a great influence on positioning accuracy. In **Fig. 6**, we contrastive analysis the impact of v_{max} on positioning accuracy of four different algorithms. The variation tendencies of four curves are very similar. The localization error decreases firstly when v_{max} changes from 2.5 units/s to 7.5 units/s, and then gradually increases. The simulation results show that the localization errors of our RMCB algorithm is lower than the MCB, RMCL and RMMCB algorithm by about 22%, 11% and 14% respectively on average.

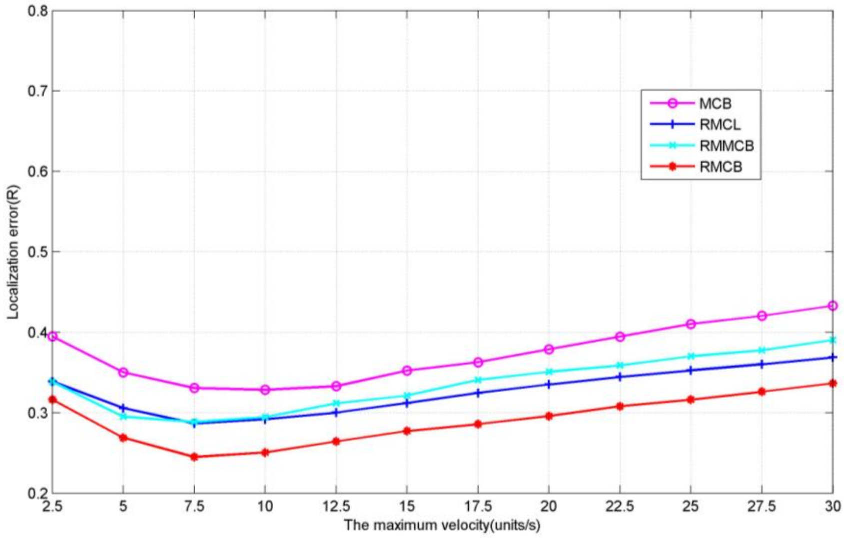

Fig. 6. Localization error with maximum velocity

4.4 The Effect of Anchor Nodes Number

As we know, the positioning accuracy can always be improved with the increase in the number of anchor nodes. However, we cannot use as many anchor nodes as we want since it is expensive. Therefore, even use a limited number of anchor nodes, the localization algorithm should also be able to achieve sufficient accuracy. The effect of the anchor nodes number on localization error is shown in **Fig. 7**. When the anchor nodes number is more than 50, the localization error of the four algorithms is gradually stable. However, the localization error of our proposed RMCB algorithm is always lower than the other three algorithms regardless of how many the anchor nodes number is.

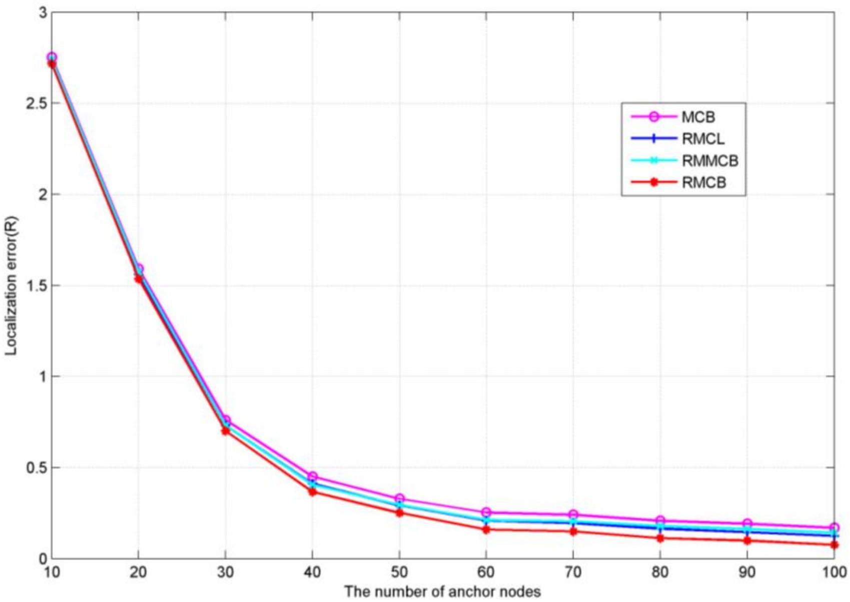

Fig. 7. Localization error with anchor nodes number

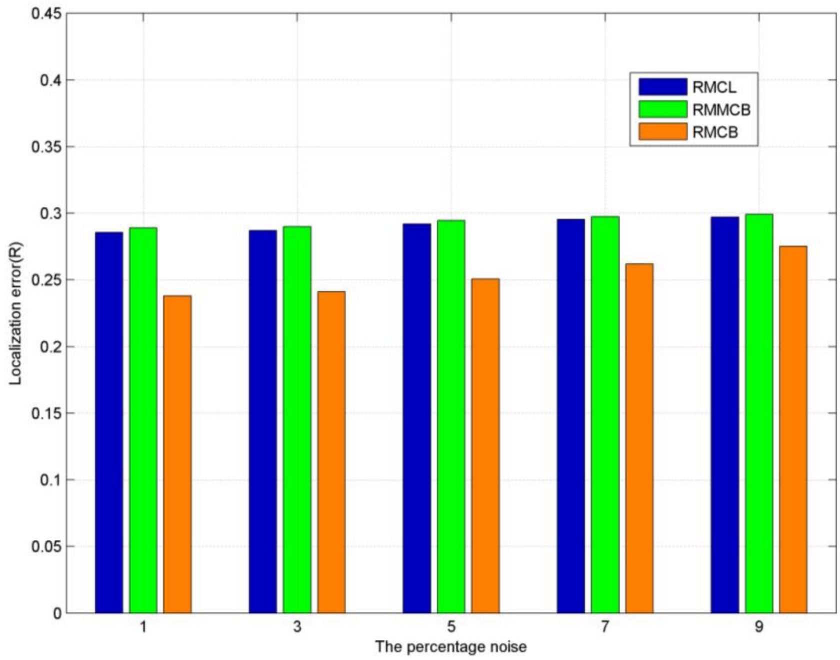

Fig. 8. Localization error with percentage noise

4.5 The Effect of Percentage Noise

Average of localization error for P_n (percentage noise in distance measurement) = 1, 3, 5, 7, 9 are computed and analyzed in **Fig. 8**. Because of the accuracy of the MCB algorithm is not affected by distance measurement, we only contrast the other three algorithms here. As shown in this figure, the effect of P_n can be clearly seen. The localization error increases with the increasing noise P_n whatever the method is. However, our RMCB method is more precise than the RMCL and RMMCB method by about 13% and 14% on average.

5. Conclusion and Future Work

In this paper, we propose a Range-based Monte Carlo Box (RMCB) algorithm for mobile nodes localization in WSNs. The main contributions of this paper are the following four parts: (1) It improves the efficiency and success rate of sampling by using the RSSI ranging technique to build the sample box. (2) It enhances the accuracy of valid samples by adding a preset error coefficient in the sampling and filtering phase since the RSSI ranging model has a certain degree of error. (3) It introduces the sPSO algorithm to generate new samples and avoid constantly repeated sampling and filtering process. (4) The dynamic weight instead of constant for two-hop anchors in objective function makes the sample generation process more accurately and quickly. Simulation results denote that our proposed RMCB algorithm can reduce the localization error by 24%, 14% and 14% on average compared to the MCB, RMCL and RMMCB algorithm respectively and always achieves higher precision under the various conditions of changing maximum velocity, anchor numbers and percentage noise. Thus, it is suitable for high precision required positioning scenes. For simplicity, most of the mobile node localization algorithms are considered in two-dimensional space, which not conforms to the actual situation. Therefore, how to localize mobile sensor nodes in three-dimensional space is our research direction in the future.

Acknowledgment

This work is supported by National Natural Science Foundation of China (No. 61472278, 60872064 and 61102125), the Natural Science Foundation of Tianjin (No.12JCYBJC10200).

References

[1] Andy Ward, Alan Jones and Andy Hopper, "A new location technique for the active office," *IEEE Personal Communications*, vol. 4, no. 5, pp. 42-47, 1997. Article (CrossRef Link)
[2] Dragos Niculescu and Badri Nath, "Ad hoc positioning system (APS) using AOA," in *Proc. of 22nd Annual Joint Conference of the IEEE Computer and Communications*, pp. 1734-1743, 30 March-3 April, 2003. Article (CrossRef Link)
[3] Andreas Savvides, Chih-Chieh Han and Mani B. Strivastava, "Dynamic fine-grained localization in ad-hoc networks of sensors," in *Proc. of the 7th Annual International Conference on Mobile Computing and Networking*, pp. 166-179, July 16-21, 2001. Article (CrossRef Link)
[4] U. Bischoff, M. Strohbach, M. Hazas and G. Kortuem, "Constraint-based distance estimation in ad-hoc wireless sensor networks," in *Proc. of European Workshop on Wireless Sensor Networks*, pp. 54-68, February 13-15, 2006. Article (CrossRef Link)
[5] N. Bulusu, J. Heidemann and D. Estrin, "GPS-less low cost outdoor localization for very small devices," *IEEE Personal Communications*, vol. 7, no. 5, pp.28-34, 2000. Article (CrossRef Link)
[6] L. Doherty, K.S.J. Pister and L.E. Ghaoui, "Convex position estimation in wireless sensor networks," in *Proc. of 20th Annual Joint Conference of the IEEE Computer and Communications Societies*, pp. 1655-1663, April 22-26, 2001. Article (CrossRef Link)
[7] T. He, C. Huang, B.M. Blum, J.A. Stankovic and T. Abdelzher, "Range-free localization schemes for large scale sensor networks," in *Proc. of the 9th Annual International Conference on Mobile Computing and Networking*, pp. 81-95, September 14-19, 2003. Article (CrossRef Link)
[8] D. Niculescu and B. Nath, "DV based positioning in ad hoc networks," *Telecommunication Systems*, vol. 22, no. 1, pp. 267-280, 2003. Article (CrossRef Link)
[9] Y. Shang, W. Ruml, Y. Zhang and M. Fromhertz, "Localization from mere connectivity," in *Proc. of the 4th ACM International Symposium on Mobile Ad Hoc Networking & Computing*, pp. 201-212, June 1-3, 2003. Article (CrossRef Link)
[10] M. Shon, M. Jo and H. Choo, "An interactive cluster-based MDS localization scheme for multimedia information in wireless sensor networks," *Computer Communications*, vol. 35, no. 15, pp. 1921-1929, 2012. Article (CrossRef Link)
[11] V. Fox, J. Hightower, L. Liao, D. Schulz and G. Borriello, "Bayesian filtering for location estimation," *IEEE Pervasive Computing*, vol. 2, no. 3, pp. 24-33, 2003. Article (CrossRef Link)
[12] V. Tran-Quang, H. Nguyen-Khanh, and T. Ngo-Quynh, "Target tracking system using lateration estimation method in wireless sensor networks," in *Proc. of 5th International Conference on Ubiquitous and Future Networks*, pp. 264-269, July 2-5, 2013. Article (CrossRef Link)
[13] Y. Hamouda and C. Phillips, "Adaptive sampling for energy-efficient collaborative multitarget tracking in wireless sensor networks," *IET Wireless Sensor Systems*, vol. 1, no. 1, pp. 15-25, 2011. Article (CrossRef Link)
[14] V. Tran-Quang, T. Ngo-Quynh and M. Jo, "A Lateration-localizing Algorithm for Energy-efficient Target Tracking in Wireless Sensor Networks," *Ad Hoc & Sensor Wireless Networks*, vol. 34, no. 1-4, pp. 191-220, 2016.

[15] Ling-xuan Hu and David Evans, "Localization for mobile sensor networks," in *Proc. of the 10th Annual International Conference on Mobile Computing and Networking*, pp. 45-57, September 26-October 1, 2004. Article (CrossRef Link)
[16] Aline Baggio and Keon Langendoen, "Monte-carlo localization for mobile wireless sensor networks," in *Proc. of International Conference on Mobile Ad-Hoc and Sensor Networks*, pp. 317-328, December 13-15, 2006. Article (CrossRef Link)
[17] Masoomeh Rudafshani and Suprakash Datta, "Localization in wireless sensor networks," in *Proc. of 6th International Symposium on Information Processing in Sensor Networks*, pp. 51-60, April 25-27, 2007. Article (CrossRef Link)
[18] Jang-Ping Sheu, Wei-Kai Hu and Jen-Chiao Lin, "Distributed localization scheme for mobile sensor networks," *IEEE Transactions on Mobile Computing*, vol. 9, no. 4, pp. 516-526, 2010. Article (CrossRef Link)
[19] H. Zhu, X. Zhong, Q. Yu and Y. Wan, "A localization algorithm for mobile wireless sensor networks," in *Proc. of 3rd International Conference on Intelligent System Design and Engineering Applications*, pp. 81-85, Jane 16-18, 2013. Article (CrossRef Link)
[20] Xiao-lin Wu, Zhi-long Shan, Shu-lin Cao and Chu-qun Cao, "Monte carlo boxed localization algorithm for mobile nodes based on received signal strength indication ranging," *Journal of Computer Applications*, vol. 35, no. 4, pp. 916-920, 2015. Article (CrossRef Link)
[21] M. Khelifi, I. Benyahia, S. Moussaoui and F. Naït-Abdesselam, "An overview of localization algorithms in mobile wireless sensor networks," in *Proc. of International Conference on Protocol Engineering and International Conference on New Technologies of Distributed Systems*, pp. 1-6, July 22-24, 2015. Article (CrossRef Link)
[22] Z. Fang, Z. Zhao, D. Geng, Y. Xuan, L. Du and X. Cui, "RSSI variability characterization and calibration method in wireless sensor network," in *Proc. of IEEE International Conference on Information and Automation*, pp. 1532-1537, June 20-23, 2010. Article (CrossRef Link)
[23] Wang Hu and Zhi-shu Li, "A simpler and more effective particle swarm optimization algorithm," *Journal of software*, vol. 18, no. 4, pp. 861-868, 2007. Article (CrossRef Link)
[24] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. of IEEE International Conference on Neural Networks*, pp. 1942-1948, November 27-December 1, 1995. Article (CrossRef Link)
[25] Tracy Camp, Jeff Boleng and Vanessa Davies, "A survey of mobility models for ad hoc network research," *Wireless Communications and Mobile Computing*, vol. 2, no. 5, pp. 483-502, 2002. Article (CrossRef Link)

Dan Li is currently working toward PhD degree in the School of Computer Science and Technology, Tianjin University, China.

Xianbin Wen received his M.S. degree (April 1992) in applied mathematics and PhD degree (May 2005) in Computer Science from Northwestern Polytechnical University, China, respectively. He is currently a professor of computer science and technology of Tianjin University of Technology. His research interests include fusion of network information, computer vision, pattern recognition, statistical learning theory, Intelligence computing and mutiscale system theory.