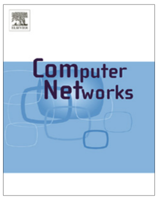
Computer Networks xxx (2013) xxx–xxx

Contents lists available at SciVerse ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet



Efficient localization for mobile sensor networks based on constraint rules optimized Monte Carlo method

Q1 Ze Wang [a,*], Yunlong Wang [a], Maode Ma [b], Jigang Wu [a]

[a] School of Computer Science and Software, Tianjin Polytechnic University, Tianjin, China
[b] School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore

ARTICLE INFO

Article history:
Received 18 December 2012
Received in revised form 18 June 2013
Accepted 25 June 2013
Available online xxxx

Keywords:
Mobile sensor networks
Localization
Sequential Monte Carlo methods

ABSTRACT

Wireless sensor networks (WSNs) have been widely used in many fields. The issue of node localization is a fundamental problem in WSNs. And it is the basis and prerequisite for many applications. Due to the mobility of the sensor nodes, it is more challenging to locate nodes in the mobile WSNs than in the static ones. The existing localization schemes for mobile WSNs are almost based on the Sequential Monte Carlo (SMC) localization method. The SMC-based schemes may suffer from low sampling efficiency resulted from a large sampling area, which makes them difficult to achieve high localization accuracy and efficiency. Some schemes try to reduce the sampling area by further employing position relationship with neighbor common nodes, while we have found that the movements of the neighbor beacon nodes have not been fully exploited. Addressing this issue, in this paper, some new constraint rules are developed and some existing constraint rules are optimized with the consideration of the moving distance and direction of neighbor beacons. A series of distance constraint conditions are further created, by which, the scope/size of the sampling area can be further reduced, and the samples can be filtered more accurately. The performance of our algorithm is evaluated by extensive simulation experiments. The simulation results show that the localization error and computation cost of our proposed algorithm are lower than those of the existing ones, even when the speed of the sensor nodes is relative high.

1. Introduction

Wireless sensor networks (WSNs) consist of large numbers of sensing devices to collect information in a monitored field. Such networks have been used in many fields, including object tracking, disaster rescue, environmental monitoring, coverage calculation and so on. Location awareness is very important because many applications of WSNs depend on the information of locations of sensor nodes. Localization is to obtain the nodes positions. In most existing sensor networks, sensors are static, while some modern applications require sensors to be mobile. The mobile beacons are aware of their own location information with the use of the global positioning, or other location support system. They move and broadcast their coordinates at some certain location for improving the location accuracy of the sensors nodes. The sensor nodes can calculate their new locations by intersection of all possible region based on the different coordinates received from the mobile beacons.

Many localization algorithms have been proposed in the past several years [1–15]. According to whether the actual distance measurements between nodes are used or not, the localization algorithms can be divided into two categories: range-based and range-free. Range-based algorithms calculate the location of unknown nodes by measuring the actual distance or direction between nodes, which also needs some special hardware to obtain accurate absolute range

Q2 * Corresponding author. Tel.: +86 18622509157.
E-mail addresses: wangze.cn@gmail.com, wangze@tjpu.edu.cn (Z.
Q1 Wang).

1389-1286/$ - see front matter © 2013 Published by Elsevier B.V.
http://dx.doi.org/10.1016/j.comnet.2013.06.010

measurements. Range-based approaches have exploited time of arrival [16], received signal strength [17], time difference of arrival of two different signals (TDOA) [18], and angle of arrival (AOA) [19]. Though they can achieve higher localization accuracy than the range-free algorithms, the required hardware is usually not economical. Range-free algorithms, which do not need special hardware, are more attractive due to their low cost in recent years. Range-free algorithms usually require that each node knows which nodes are within its radio range as well as the (ideal) radio range of common sensors or beacons. Many range-free approaches have been proposed including centroid method [20], approximate point-in-triangulation test (APIT) method [21], amorphous localization method [22], distance vector-hop (DV-HOP) method [23] and so on. For example, the APIT method isolates the environment into triangular regions among beacon nodes, and uses a grid algorithm to calculate the maximum area in which a node most likely resides. APIT typically assumes a larger radio range for seed nodes and hence requires higher seed density.

Existing localization algorithms for mobile sensor networks are usually based on the Sequential Monte Carlo (SMC) method [6,10,13,24–28,31]. By such localization algorithms, the position distribution of a sensor node is represented with a set of weighted samples. In order to obtain enough valid samples to characterize the distribution of its position, a sensor node needs to repeat the sampling step (generate candidate samples) and the filtering step (evaluate candidate samples and filter out invalid samples) many times. However, generating candidate samples is usually a very costly operation. On the Micaz [32] platform, the cost of generating a candidate sample is much higher than evaluating it. Since sensor nodes usually have limited computational ability, it is necessary to improve the sampling efficiency in order to reduce the computational cost. Another problem of most existing SMC-based localization algorithms is that some of them rely on increasing beacon density to improve the localization accuracy. However, beacon nodes are usually more expensive than sensor nodes. Because there are much more sensor nodes than beacon nodes in a sensor network, it will be very beneficial if sensor nodes can be used to improve the localization accuracy. In some existing algorithms, a technique called bounding box [24] is used to improve the sampling efficiency by reducing the scope from which the candidate samples are selected. The WMCL [31] algorithm uses the estimated position information of sensor nodes and further reduces the size of a sensor node's bounding box to improve localization accuracy. However, we find that there are still spaces left for building new constraint rules or optimizing the existing ones. In this paper, we focus on finding new constraint rules from observation on the beacon nodes in time unit $t-1$ and to optimize existing rules from observation on the beacons' movements from time unit $t-1$ to t. Based on it, we propose a new SMC-based localization scheme, a Constraint rules Optimized Monte Carlo Localization Scheme (COMCL), which can adopt more effective constraint rules on building an effective sampling area and filtering the candidate samples efficiently.

The remainder of this paper is organized as follows. In Section 2, the related works of our scheme are described. In Section 3, the optimization methods of the constraint rules are introduced. In Section 4, the COMCL scheme is described in detail. In Section 5, the performance analysis is presented. And finally, in Section 6, the paper is concluded with a summary.

2. Related works

The SMC-based algorithms assume that all the nodes are moving and their maximum speeds are known. The time is divided into discrete time units and in each time unit each sensor node calculates its position through three steps, initialization, sampling and filtering. Samples are the possible locations randomly drawn from a sampling area which represents a region where the sensor node to be localized most probably resides. In the initialization step, N samples are randomly drawn from the deployment area. In the importance sampling step, candidate samples are drown from a sampling area related to the samples in the previous time unit $t-1$ and the motion of the node. The weights of the samples are computed using the observations collected in time unit t. The observations are usually the location and connectivity information of the neighbor nodes obtained by the sensor node to be localized, e.g. a sensor node can directly hear a beacon node A with location (x_A, y_A) and a beacon node B with location (x_B, y_B). Based on the above observations, samples of locations which are beyond the communication ranges of the node A and B will be weighted as 0. In the filtering step, samples with weight 0 will be filtered out. The importance sampling step and the filtering step may repeat several times in order to obtain enough number of effective samples with weight greater than 0. At last, the weighted average of the N samples is used as the estimation of the position of the sensor node in the current time unit. Next, we will first review some schemes and then propose the optimization direction.

In 2004, Hu and Evans for the first time applied the SMC method into the location problems of the mobile sensor network nodes, and put forward the Monte Carlo Localization (MCL) [6] scheme. By the MCL algorithms, the time is divided into many discrete time units and in each time unit all the sensor nodes estimate their own positions by calculate a series of weighted sampling coordinate values, and the weight of each sample is either 0 or 1. After updating sampling coordinates set used for localization in every time unit, the estimated positions of sensor nodes will then be recalculated.

In MCL scheme, the three main steps are as follows:

Initialization: All the nodes are deployed uniformly in the whole deployment area. And all of the sensor nodes have no knowledge of their location. N is a constant that denotes the largest number of valid samples, v_{max} denotes the maximum speed of a sensor node, r denotes communication radius of the nodes. A sensor node selects randomly the N positions within the deployment area to initialize the initial sampling set L_0.

Prediction: In time unit t, a sensor node generates the sampling set L_t based on L_{t-1}, which is the sampling set in the previous time unit, and the new observations O_t.

Assuming that the sampling set L_{t-1} includes a position l_{t-1}^i, so adding the new position l_t^i that obtained from the position based on the probability $p\left(l_t^i|l_{t-1}^i\right)$ to the sampling set L_t. $p\left(l_t^i|l_{t-1}^i\right)$ is defined as follows:

$$p\left(l_t^i|l_{t-1}^i\right) = \begin{cases} \frac{1}{\pi v_{max}^2} & d\left(l_t^i, l_{t-1}^i\right)) < v_{max} \\ 0 & d\left(l_t^i, l_{t-1}^i\right)) \geqslant v_{max} \end{cases} \tag{1}$$

Therein, $d\left(l_t^i, l_{t-1}^i\right)$ denotes the distance between l_t^i and l_t^i. If l_{t-1}^i is one possible position of a node in previous time unit, the possible current positions l_t^i are contained in the circular region with center l_{t-1}^i and radius v_{max}.

Filtering: Unqualified samples are filtered out based on the current observations O_t at time unit t. Let S denotes the set of one-hop beacon neighbors of the sensor node and T denotes the set of two-hop beacon neighbors of the sensor node. Every candidate samples l_t is filtered by the following formula.

$$filter(l_t) = \forall s \in S, d(l_t, s) \leqslant r \wedge \forall s \in T, r < d(l_t, s) \leqslant 2r \tag{2}$$

While achieving high localization accuracy, MCL left spaces for improvement in two aspects: its sampling efficiency is low and it relies on high beacon density to achieve high localization accuracy. The Monte Carlo localization Boxed (MCB) algorithm [24] which is proposed by Baggio and Langendoen in 2006 improves MCL's sampling efficiency by using bounding box to restrict the scope from which the candidate samples are drawn. Fig. 1 shows how the bounding box is built in MCB. Assuming that a sensor node has some one-hop and/or two-hop neighbor beacon nodes, then the possible positions of the sensor node are contained in the circular region with the beacon node as the center and r (or $2r$) as the radius, namely inevitable exist in the corresponding external rectangular area. When there are multiple rectangular areas, the public rectangular cross-region is the coverage region of an anchor box which is used as the sampling area.

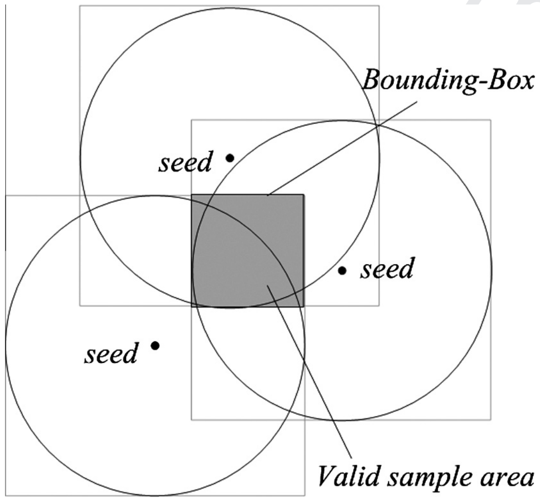


Fig. 1. Building bounding box.

After building an anchor box defined by (x_{min}, y_{min}) and (x_{max}, y_{max}), The borders of the sampling boxes will be calculated for each of the samples in L_{t-1}:

$$\begin{cases} x_{min}^i = max\{x_{min}^i, x_{t-1}^i - v_{max}\} \\ y_{min}^i = max\{y_{min}, y_{t-1}^i - v_{max}\} \\ x_{max}^i = min\{x_{max}, x_{t-1}^i + v_{max}\} \\ y_{max}^i = min\{y_{max}, y_{t-1}^i + v_{max}\} \end{cases} \tag{3}$$

The candidate samples are drawn from these sample boxes. The MCB scheme reduces the size of sampling area, improves the sampling efficiency, and also improves the localization accuracy within a certain scope. However, the filtering phase in MCB just follows the filtering conditions as that of the MCL scheme.

Another SMC-based localization scheme, the Weighted Monte Carlo Localization algorithm (WMCL) [31], is proposed by Zhang et al. to further improve the localization performance by introducing the observation to the common neighbor nodes. Being different from the MCB method, WMCL method makes full use of the positioning information of the node itself and its common neighbor nodes in last time unit, including the estimated locations, the corresponding sample set, and the maximum possible error of estimated locations on the x-axis and y-axis, to further reduce the size of the bounding box. At the same time, a kind of effective weight calculation method is put forward to improve localization accuracy.

Assumes l_t^i is a sample obtained by random sampling within the bounding box. S and T denote the set of one-hop beacon neighbors of the sensor node and the set of two-hop beacon neighbors of the sensor node respectively. US denotes the set of its one-hop neighboring common sensor nodes. The set of all nodes that the node can observe can be expressed as $O_t = S \cup T \cup US$. s denotes an arbitrary node observed, L_{t-1}^s denotes the sample set of the node s in the last time unit. Then the weight of sample l_t^i is computed as follows:

$$\tilde{w}_t^i = p\left(O_t|l_t^i\right) = \prod_{s\in S,T} p\left(s|l_t^i\right) \prod_{s\in US} p\left(s|l_t^i\right) \tag{4}$$

The first step, when beacon node $s \in S \cup T$, the calculation of $p\left(s|l_t^i\right)$ is the same as MCL method, when the filtering conditions of the Eq. (2) are satisfied, $p\left(s|l_t^i\right)$ is 1, otherwise is 0. The second step, concerning the neighbor common node s, when $s \in US$, $p\left(s|l_t^i\right)$ is computed approximately as

$$p\left(s|l_t^i\right) \approx \frac{\sum_{L\in L_{t-1}^s}\left[d\left(l, l_t^i\right) \leqslant r + v_{max}\right]}{\left|L_{t-1}^s\right|} \tag{5}$$

Thus the observation to the common neighbor nodes are adopted to evaluate the weight of each sample. The second step makes the sample weight be in a range [0, 1] which changes the situation in MCL that each valid sample is equally involved (weighted 1) in the calculation of the final location.

The WMCL scheme makes use of some potential effective information, such as the sampling sets of common neighbor nodes in the last time unit, to further cut the bounding box to improve the localization efficiency.

Please cite this article in press as: Z. Wang et al., Efficient localization for mobile sensor networks based on constraint rules optimized Monte Carlo method, Comput. Netw. (2013), http://dx.doi.org/10.1016/j.comnet.2013.06.010

However, we find that some potential movement information are not exploited yet. Since the positions of a beacon neighbor in time unit $t-1$ and t are known to the node, the moving distance and direction of the beacon will be known too. The introduce of this kind of observation on the beacon's movement in a time unit will reduce the uncertainty of the node's position so that the corresponding constraint rules built on this observation can further reduce the sampling area. Therefore, it is possible for us to build some more efficient constraint rules based on the movements of the nodes.

3. Constraint rules optimization

3.1. Network model and notation

Unit Disk Graph (UDG) connectivity model is adopted just like most of the existing SMC-based localization schemes. By the UDG model, two nodes p and q can directly communicate with each other if and only if they are within the communication range defined by a radius r. If two nodes p and q are within r, we say that q is a one-hop neighbor of p. It is true that the probability to receive a frame decreases with the SNR, which is related to the distance between correspondents. We consider one hop transmission range as a distance over which the receiver can successfully receive signals with a tolerable SNR, which is the threshold of SNR. When the received signal has lower SNR than the threshold, the receiver node will be considered as out of the transmission range of the transmitter. Therefore, we are more likely to treat transmission range as a distance to receive a nodes message with a probability very close to 1 just like most of the existing SMC-based schemes. This is a crude model of real radio ranges, which are affected by noise, fading, and many other factors. A more detailed model of radio ranges are illustrated in [30]. We refer the interested readers to some of the existing SMC-based localization schemes [29,31] to find how the variation and irregularity of communication ranges affect the localization process.

For ease of description, a beacon node is called a *seed* which can obtain its exactly coordinate and broadcast it in the beginning of every time unit. A normal sensor node is called a *node* which need to be localized once a time unit. Notation is defined as follows:

- $seed_t$, $seed_{t-1}$: the real position of a beacon node *seed* in time unit t and $t-1$.
- $node_t$, $node_{t-1}$: the real position of a sensor node *node* in time unit t and $t-1$.
- x_{range}, y_{range}: the width and height of the deployment area.
- v_{max}: the maximum speed of sensor nodes (including beacon nodes).
- d_n: the distance which a sensor node moved from time unit $t-1$ to time unit t.
- d_s: the distance which a beacon node moved from time unit $t-1$ to time unit t.
- r: the radius of the communication range of sensor nodes.
- n_d, s_d: the average number of sensor nodes and of beacon nodes in a node's communication range respectively.
- $d(node_i, seed_j)$: the Euclidean distance between a *node*'s position in time unit i and a *seed*'s position in time unit j.
- $d(n_i, s_j)$: the alias of $d(node_i, seed_j)$.
- $U_{i,j}$, $L_{i,j}$: the upper bound and lower bound of $d(node_i, seed_j)$.

3.2. Existing constraint rules

The existing SMC-based localization schemes try to localize a node by building constraint rules obtained from the distance relations between a sensor node and one of it is one-hop or two-hop beacon neighbors. The basic constraint rule is defined as:

$$\forall s_t \in S_t, 0 \leqslant d(n_t, s_t) \leqslant r \wedge \forall s_t \in T_t, r < d(n_t, s_t) \leqslant 2r \tag{6}$$

Therein, S_t is the set of node p's one-hop neighbor seeds in time unit t and T_t is the set of two-hop neighbor seeds in time unit t.

Similarly, in time unit $t-1$, we have a constraint rule:

$$\begin{aligned}&\forall s_{t-1} \in S_{t-1}, \quad 0 \leqslant d(n_{t-1}, s_{t-1}) \leqslant r \wedge \forall s_{t-1} \in T_{t-1},\\&\quad r < d(n_{t-1}, s_{t-1}) \leqslant 2r\end{aligned} \tag{7}$$

In the time unit t, the existing constraint rules can be unified into $L_{t,t} \leqslant d(n_t, s_t) \leqslant U_{t,t}$. Therefore, for each beacon node $s_t \in S_t \cup T_t$, we can form a concrete constraint condition which represents a certain area with minimum and maximum distance to a determined position. Since the area of the sampling region is determined by the intersections of these areas, the localization accuracy will be impacted by the size of the intersections. For the existing constraint rules, if the number of neighbor beacon nodes is few or zero, the sampling area will be too large for an accurate position prediction due to lack of enough constraint conditions. If we can try to build some new constraint rules or optimize the existing ones, the localization accuracy could be improved. Next, we will first try to build new constraint rules by deducing the upper bound and lower bound of $d(n_t, s_{t-1})$ and then to optimize the existing rules.

3.3. Building new rules based on $seed_{t-1}$

3.3.1. Time going backwards

The existing rule $L_{t,t} \leqslant d(n_t, s_t) \leqslant U_{t,t}$ could be used for deducing the upper bound and lower bound in $L_{t,t-1} \leqslant d(n_t, s_{t-1}) \leqslant U_{t,t-1}$. Since $d(n_t, s_{t-1})$ is different from $d(n_t, s_t)$ on seed moving from the time unit t to the last time unit $t-1$, we call this case as 'Time Going Backwards'. In this case, the concrete value of $seed_t$ and $seed_{t-1}$ is known to the neighbor sensor nodes to be localized, thus the moving distance d_s of the seed is a determined value $d(seed_t, seed_{t-1})$. Let L and U denote the lower bound and upper bound in the referenced existing constraint rules. In this case, L and U denotes $L_{t,t}$ and $U_{t,t}$ respectively. After the time going backwards for $seed_t$, the position relationship between $node_t$ and $seed_{t-1}$ is showed in Fig. 2.

In Fig. 2, the shadowed area represents the possible positions of $node_t$, and we try to deduce the bounds of $d(n_t, s_{t-1})$ in three conditions, (a) $d_s \leqslant L$; (b) $L < d_s \leqslant U$; (c) $d_s > U$. To solve the upper and lower bounds of $d(n_t, s_{t-1})$, we should observe the possible moving distance and direction of the *seed*.

Suppose the *seed* moves away from the *node*, $d(n_t, s_{t-1})$ will reach the maximum value $d(n_t, s_t) + d_s$, so the upper bound $U_{t,t-1}$ is $U_{t,t} + d_s$, which is showed in each condition of Fig. 2.

Suppose the *seed* moves towards the *node*, $d(n_t, s_{t-1})$ will probably reach the minimum value. We discuss the lower bound under the three conditions.

In Fig. 2a, the $seed_{t-1}$ cannot fall into the shadowed area under the condition $d_s < L$, so the lower bound $L_{t,t-1}$ is $L_{t,t} - d_s$.

In Fig. 2b, the $seed_{t-1}$ will fall into the shadowed area under the condition $L < d_s \leqslant U$, so the lower bound $L_{t,t-1}$ should be zero.

In Fig. 2c, the $seed_{t-1}$ will fall outside the outer circular border of the shadowed area under the condition $d_s > U$, so the lower bound $L_{t,t-1}$ is $d_s - U_{t,t}$.

Consequently, the above results can be combined into the following equation:

$$\begin{cases} U_{t,t-1} = U_{t,t} + d_s \\ L_{t,t-1} = max\{L_{t,t} - d_s, 0, d_s - U_{t,t}\} \end{cases} \tag{8}$$

3.3.2. Time going forwards

The existing rule $L_{t-1,t-1} \leqslant d(n_{t-1}, s_{t-1}) \leqslant U_{t-1,t-1}$ could also be used for deducing the upper bound and lower bound in $L_{t,t-1} \leqslant d(n_t, s_{t-1}) \leqslant U_{t,t-1}$. Since $d(n_t, s_{t-1})$ is different from $d(n_{t-1}, s_{t-1})$ on node moving from last time unit $t - 1$ to the current time unit t, we call this case 'Time Going Forwards'. In this case, the real value of $node_{t-1}$ and $node_t$ is not known, thus the moving distance d_n of the node is not a determined value. However, we can determine the range of d_n as $[0, v_{max}]$. In the case of 'Time Going Forwards', L and U denotes $L_{t-1,t-1}$ and $U_{t-1,t-1}$ respectively. After time going forwards for $node_{t-1}$, the position relationship between $node_t$ and $seed_{t-1}$ is showed in Fig. 3.

In Fig. 3, the shadowed area represents the possible positions of $seed_{t-1}$, and we try to deduce the bounds of $d(n_t, s_{t-1})$ in three conditions, (a) $d_n \leqslant L$; (b) $L < d_n \leqslant U$; (c) $d_n > U$. To solve the upper and lower bounds of $d(n_t, s_{t-1})$, we need to observe the possible moving distance and direction of the *node*.

Suppose the *node* moves away from the *seed*, $d(n_t, s_{t-1})$ will reach the maximum value $d(n_{t-1}, s_{t-1}) + d_n$, so the upper bound $U_{t,t-1}$ is $U_{t-1,t-1} + d_n$, which is showed in each condition of Fig. 3.

Suppose the *node* moves towards the *seed*, $d(n_t, s_{t-1})$ will probably reach the minimum value. We discuss the lower bound in the three conditions.

In Fig. 3a, the $node_t$ cannot fall into the shadowed area under the condition $d_n < L$, so the lower bound $L_{t,t-1}$ is $L_{t-1,t-1} - d_n$.

In Fig. 3b, the $node_t$ will fall into the shadowed area under the condition $L < d_n \leqslant U$, so the lower bound $L_{t,t-1}$ should be zero.

In Fig. 3c, the $node_t$ will fall outside the outer circular border of the shadowed area under the condition $d_n > U$, so the lower bound $L_{t,t-1}$ is $d_n - U_{t-1,t-1}$.

Since we only know the value of d_n is in a range $[0, v_{max}]$, the above results can be combined into the following equation:

$$\begin{cases} U_{t,t-1} = U_{t-1,t-1} + v_{max} \\ L_{t,t-1} = max\{L - v_{max}, 0\} \end{cases} \tag{9}$$

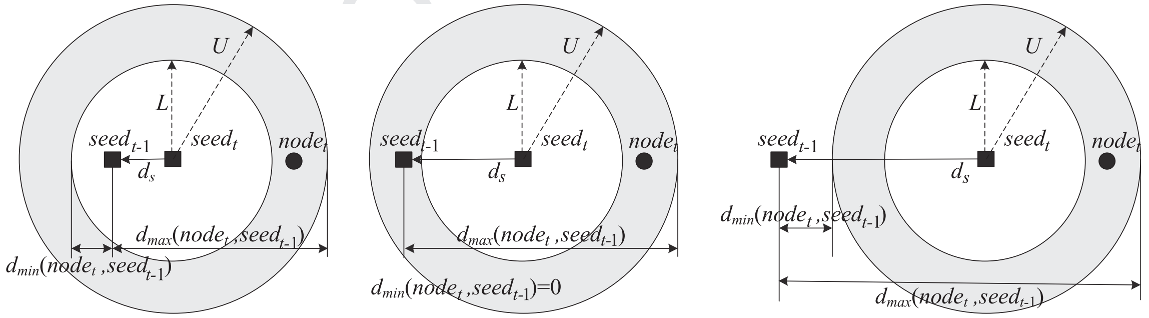
3.4. Optimization of existing constraint rule

Since the upper and lower bounds of $d(n_{t-1}, s_{t-1})$ has been determined in time unit $t - 1$, we try to deduce the bounds of $d(n_t, s_t)$ based on $U_{t-1,t-1}$ and $L_{t-1,t-1}$. After the *node* and the *seed* moves from the time unit $t - 1$ to the time unit t, the position relationship between $node_t$ and $seed_t$ is showed in Fig. 4.

As in Fig. 4a, if the *node* and the *seed* moves away from each other, $d(n_t, s_t)$ will reach the maximum value $d(n_{t-1}, s_{t-1}) + d_n + d_s$, so the upper bound $U_{t,t}$ is $U_{t-1,t-1} + d_n + d_s$.

As in Fig. 4b, if the *node* and the *seed* moves towards each other, $d(n_t, s_t)$ will probably reach the minimum value $max\{d(n_{t-1}, s_{t-1}) - d_n - d_s, 0\}$, so the lower bound $L_{t,t}$ is $max\{L_{t-1,t-1} - d_n - d_s, 0\}$.

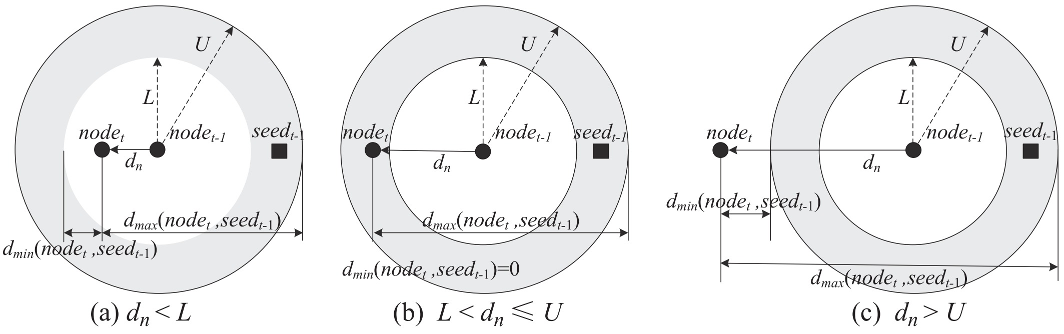
Since we only know the value of d_n is in a range $[0, v_{max}]$, the above results can be combined into the following equation:


U
L
$seed_{t\text{-}1}$
$seed_t$
$node_t$
d_s
$d_{max}(node_t, seed_{t\text{-}1})$
$d_{min}(node_t, seed_{t\text{-}1})$
U
L
$seed_{t\text{-}1}$
$seed_t$
$node_t$
d_s
$d_{max}(node_t, seed_{t\text{-}1})$
$d_{min}(node_t, seed_{t\text{-}1})=0$
U
L
$seed_{t\text{-}1}$
$seed_t$
$node_t$
d_s
$d_{min}(node_t, seed_{t\text{-}1})$
$d_{max}(node_t, seed_{t\text{-}1})$

(a) $d_s < L$ (b) $L < d_s \leqslant U$ (c) $d_s > U$

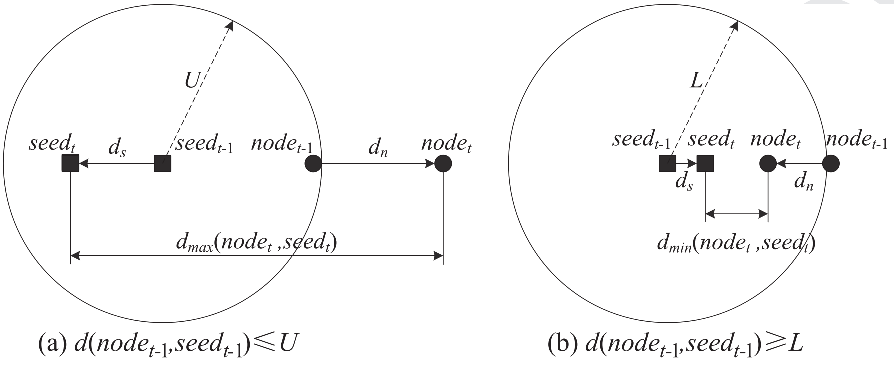
Fig. 2. Position relationship when time going backwards.

Q1 Please cite this article in press as: Z. Wang et al., Efficient localization for mobile sensor networks based on constraint rules optimized Monte Carlo method, Comput. Netw. (2013), http://dx.doi.org/10.1016/j.comnet.2013.06.010


U
L
node_t
node_{t-1}
seed_{t-1}
d_n
d_max(node_t ,seed_{t-1})
d_min(node_t ,seed_{t-1})
d_min(node_t ,seed_{t-1})=0

(a) $d_n < L$ (b) $L < d_n \leqslant U$ (c) $d_n > U$

Fig. 3. Position relationship when time going forwards.



(a) $d(node_{t\text{-}1},seed_{t\text{-}1}) \leqslant U$ (b) $d(node_{t\text{-}1},seed_{t\text{-}1}) \geqslant L$

Fig. 4. Position relationship between *node*$_t$ and *seed*$_t$.

$$\begin{cases} U_{t,t} = U_{t-1,t-1} + d_s + v_{max} \\ L_{t,t} = max\{L_{t-1,t-1} - d_s - v_{max}, 0\} \end{cases} \tag{10}$$

3.5. Building a solutions table of constraint rules

To apply the new or optimized constraint rules, the specific value of the upper and lower bounds of them should be determined after the bounds of the referenced rules are given. The bounds of the existing rules are determined by whether the neighbor seed is a one-hop neighbor or a two-hop neighbor. For the convenience of implementation, we work out the solutions of constraint rules and list them in Table 1. When the bounds of $d(n_{t-1}, s_{t-1})$ and $d(n_t, s_t)$ are given, we can quickly track the bounds of new or optimized constraint rules from the table.

It should be noticed that in the line Nos. 3 and 6 of Table 1 the *seed* is not the neighbor of the *node* in time unit $t - 1$ and in the line Nos. 7 and 8 the *seed* is not the neighbor of the *node* in time unit t. In these cases, the corresponding constraint rule will not be applicable unless the *seed* has taken some special method to transmit its positions in time unit $t - 1$ and t to the nodes which are not its one-hop or two-hop neighbors. However, at least six more new constraint conditions derived from the solutions of $d(n_t, s_{t-1})$ and two more optimized constraint conditions derived from the solutions of new $d(n_t, s_t)$ can be employed to reduce the size of the sampling area.

4. Proposed scheme

From the new observations of the position relationship between the *node* and the *seed* in different adjacent time units, the set of constraint rules could be enriched or optimized to achieve better localization. The proposed COMCL

Table 1
Solutions of constraint rules.

No.	$d(n_{t-1}, s_{t-1})$	$d(n_t, s_t)$	$d(n_t, s_{t-1})$	$d(n_t, s_t)$ (new)
1	$[0, r]$	$[0, r]$	$[max(d_s - r, 0), d_s + r]$	$[0, r]$
2	$(r, 2r]$	$[0, r]$	$[max(r - v_{max}, d_s - r, 0), d_s + r]$	$[max(r - v_{max-d_s}), r]$
3	N/A	$[0, r]$	$[max(d_s - r, 0), d_s + r]$	$[0, r]$
4	$[0, r]$	$(r, 2r]$	$[max(d_s - 2r, r - d_s, 0), min(d_s + 2r, r + v_{max})]$	$(r, min(r + d_s + v_{max}, 2r)]$
5	$(r, 2r]$	$(r, 2r]$	$[max(r - d_s, d_s - 2r, 0), d_s + 2r]$	$(r, 2r]$
6	N/A	$(r, 2r]$	$[max(r - d_s, d_s - 2r, 0), d_s + 2r]$	$(r, 2r]$
7	$[0, r]$	N/A	$[0, r + v_{max}]$	$[0, r + d_s + v_{max}]$
8	$[r, 2r]$	N/A	$[max(r - v_{max}, 0), 2r + v_{max}]$	$(r, 2r + d_s + v_{max}]$

localization scheme includes three main parts: (1) generation of the set of constraint conditions; (2) building sampling area; (3) sampling and optimized filtering. The set of constraint conditions is generated after the upper and lower bounds of the optimized constraint rules have been solved for each neighbor *seed* of the *node*. After that, a bounding box could be built as the sampling area according to the constraint conditions. The constraint conditions can also be used in the filtering phase in the proposed scheme so that the COMCL adapts more strict filtering conditions than the WMCL and other previous SMC-based localization schemes.

4.1. Generation of the set of constraint conditions

For a *node* to be localized, after finding all of its neighbor *seeds* in time unit t and $t-1$, we can determine the upper and lower bounds of $d(n_t, s_t)$ and $d(n_{t-1}, s_{t-1})$ and then track the bounds of $d(n_t, s_{t-1})$ and new $d(n_t, s_t)$ from Table 1. Thus a constraint condition can be stored as $(seed, L, U)$ which means $L \leqslant d(node_t, seed_i) \leqslant U$ with $i \in \{t, t-1\}$.

For ease of implementation, we define two *seeds* sets SS_{last} and SS_{now}. An element in SS_{last} includes a *seed*'s identification, coordinate and neighboring hops (one or two) to the *node* in time unit $t-1$. An element in SS_{now} includes a *seed*'s identification, coordinate and neighboring hops to the *node* in time unit t. Let $Constraint_{now}$ denote the set of $(seed, L, U)$ used for a *node*'s localization in time unit t. Let $Constraint_{last}$ denote the set of $(seed, L, U)$ used for a *node*'s localization in time unit $t-1$. The steps of Constraint Conditions Generation are as follows:

1. Copy SS_{now} to SS_{last}, copy $Constraint_{now}$ to $Constraint_{last}$, clear SS_{now} and $Constrant_{now}$.
2. After receiving the broadcast packets from the *seed*'s, store each *seed*'s identification, coordinate and neighboring hops into SS_{now}.
3. Generate a temp *seed* set $TS = SS_{now} \cup SS_{last} - SS_{now} \cap SS_{last}$.
4. For each element i in TS: Find i in SS_{last}, and record the neighboring hops as $H_{t-1}(i)$; Find i in SS_{now}, and record the neighboring hops as $H_t(i)$; According to $H_{t-1}(i)$ and $H_t(i)$, find the solutions from Table 1; Generate constraint conditions in the form of $(seed, L, U)$; Add constraint conditions into $Constraint_{now}$.

To better make use of the set of the constraint conditions, we try to employ $Constraint_{last}$ in the current time unit t. For each element $(seed, L, U)$ in $Constraint_{last}$, we modify it to $(seed, max\{L - v_{max}, 0\}, U + v_{max})$ and put it into a set *lastConstraint*.

4.2. Building sampling area

Both MCB and WMCL use bounding box as a sampling area to decrease the cost of random generation of sampling coordinates. By the COMCL, we try to build a bounding box in two steps: building basic sampling area (anchor box) and building optimized sampling area (refined box). In the first step, an anchor box is built according to the constraint conditions in $Constraint_{now}$ and *lastConstraint*. In the second step, the anchor box is optimized to a refined box by utilizing the location information in time unit $t-1$ obtained from the node itself and its neighbor sensor nodes.

4.2.1. Building anchor box

Each condition in $Constraint_{now}$ and *lastConstraint* is used to build a anchor box in the following method:

$$\begin{cases} x_{min} = max_{i=1}^{n}(x_{min}, x_i - L^i); & x_{max} = min_{i=1}^{n}(x_{max}, x_i + U^i) \\ y_{min} = max_{i=1}^{n}(y_{min}, y_i - L^i); & y_{max} = min_{i=1}^{n}(y_{max}, y_i + U^i) \end{cases} \tag{11}$$

Therein, (x_{min}, y_{min}) and (x_{max}, y_{max}) represents the scope of the anchor box. n denotes the number of conditions in $Constraint_{now}$ and *lastConstraint*. (x_i, y_i) denotes a *seed*'s coordinate in the ith condition and L^i and U^i denote the lower bound and upper bound of the distance to (x_i, y_i).

The anchor box could be further reduced by the shadowed method showed in Fig. 5. Since the *node* cannot locate in the circle with (x_i, y_i) as the center and L^i as the radius, we can eliminate the shadowed region showed in Fig. 5. If two vertices (Fig. 5a) or three vertices (Fig. 5b) of the anchor box locate in the circle, we can eliminate the shadowed region effectively and remain a smaller anchor box. In WMCL, only those conditions built from two-hop beacon neighbors are utilized to eliminate the shadowed region within the scope of a circle with r as the radius. Different from WMCL, we use the lower bounds in every constraint condition to achieve better effect on reducing the anchor box.

4.2.2. Building refined box

Although we do not know the exact real position of a sensor node or the exact distance difference between the prediction position and the real position, we do know that the real position of each sensor node is within the scope of its bounding box. After a time unit's movement, the new position of a node cannot be beyond the scope of the bounding box enlarged with the maximum distance v_{max}.

We try to utilize the bounding box in the last time unit to refine the bounding box in the current time unit. The refining process is as follows:

$$\begin{cases} x_{min} = max\{x_{min}, x_{t-1,min} - v_{max}\} \\ x_{max} = min\{x_{max}, x_{t-1,max} + v_{max}\} \\ y_{min} = max\{y_{min}, y_{t-1,min} - v_{max}\} \\ y_{max} = min\{y_{max}, y_{t-1,max} + v_{max}\} \end{cases} \tag{12}$$

Furthermore, the bounding boxes of the neighbor sensor nodes could be used to refine the bounding box in the current time unit. The refining process is as follows:

$$\begin{cases} x_{min} = max\{x_{min}, max_{i=1}^{n}(x_{min}^{i} - v_{max} - r)\} \\ x_{max} = min\{x_{max}, max_{i=1}^{n}(x_{max}^{i} + v_{max} + r)\} \\ y_{min} = max\{y_{min}, max_{i=1}^{n}(y_{min}^{i} - v_{max} - r)\} \\ y_{max} = min\{y_{max}, max_{i=1}^{n}(y_{max}^{i} + v_{max} + r)\} \end{cases} \tag{13}$$

Therein, (x_{min}^i, y_{min}^i) and (x_{max}^i, y_{max}^i) represent the bounding box of the ith neighbor sensor node in the time unit $t-1$.

Q1 Please cite this article in press as: Z. Wang et al., Efficient localization for mobile sensor networks based on constraint rules optimized Monte Carlo method, Comput. Netw. (2013), http://dx.doi.org/10.1016/j.comnet.2013.06.010

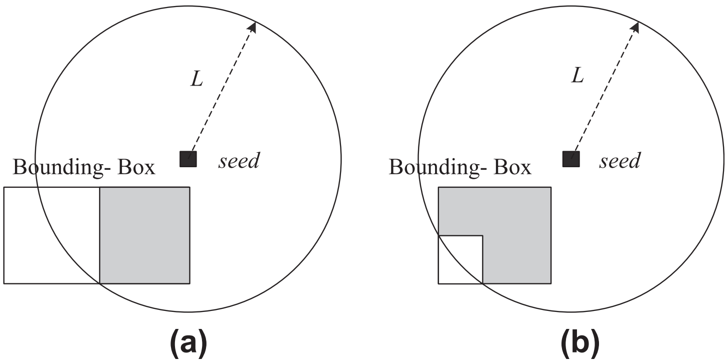


Fig. 5. Using shadowed method to reduce bounding box.

The refining process is superior to the one adopted in WMCL which will be showed in the following section.

4.2.3. Theory analysis to the refining process

Let (x_e^i, y_e^i) denote the ith neighbor node's prediction position in time unit $t-1$. Let ER_x^i and ER_y^i denote the prediction error of (x_e^i, y_e^i) in x-axis and y-axis. Let (x_{min}^i, y_{min}^i) and (x_{max}^i, y_{max}^i) define the ith neighbor node's bounding box in time unit $t-1$. We deduce the x-axis bounds of bounding box in WMCL after refining as:

$$\begin{cases} x_{min}(WMCL) = max\left\{x_{min}, max_{i=1}^{n}\left(x_e^i - v_{max} - r - ER_x^i\right)\right\} \\ \quad = max\{x_{min}, max_{i=1}^{n}(x_e^i - v_{max} - r - max(x_e^i - x_{min}^i, x_{max}^i - x_e^i))\} \\ \quad = max\{x_{min}, max_{i=1}^{n}(-max(-x_{min}^i, x_{max}^i - 2x_e^i) - v_{max} - r)\} \\ \quad = max\{x_{min}, max_{i=1}^{n}(min(x_{min}^i, 2x_e^i - x_{max}^i) - v_{max} - r)\} \\ \quad = max\{x_{min}, max_{i=1}^{n}(min(x_{min}^i, 2x_e^i - x_{max}^i)) - v_{max} - r\} \\ x_{max}(WMCL) = min\left\{x_{max}, min_{i=1}^{n}\left(x_e^i + v_{max} + r + ER_x^i\right)\right\} \\ \quad = min\{x_{max}, min_{i=1}^{n}(x_e^i + v_{max} + r + max(x_e^i - x_{min}^i, x_{max}^i - x_e^i))\} \\ \quad = min\{x_{max}, min_{i=1}^{n}(max(x_{max}^i, 2x_e^i - x_{min}^i) + v_{max} + r)\} \\ \quad = min\{x_{min}, min_{i=1}^{n}(max(x_{min}^i, 2x_e^i - x_{max}^i)) + v_{max} + r\} \end{cases} \tag{14}$$

With a similar process, the y-axis bounds can be deduced as:

$$\begin{cases} y_{min}(WMCL) = max\{y_{min}, max_{i=1}^{n}(min(y_{min}^i, 2y_e^i - y_{max}^i)) - v_{max} - r\} \\ y_{max}(WMCL) = min\{y_{min}, min_{i=1}^{n}(max(y_{min}^i, 2y_e^i - y_{max}^i)) + v_{max} + r\} \end{cases} \tag{15}$$

From the refining process adopted in COMCL, we can deduce the bounds of bounding box as:

$$\begin{cases} x_{min}(COMCL) = max\{x_{min}, max_{i=1}^{n}(x_{min}^i) - v_{max} - r\} \\ y_{min}(COMCL) = max\{y_{min}, max_{i=1}^{n}(y_{min}^i) - v_{max} - r\} \\ x_{max}(COMCL) = min\{x_{max}, min_{i=1}^{n}(x_{max}^i) + v_{max} + r\} \\ y_{max}(COMCL) = min\{y_{max}, min_{i=1}^{n}(y_{max}^i) + v_{max} + r\} \end{cases} \tag{16}$$

After we compare the bounds of WMCL with COMCL, we have:

$$\begin{cases} x_{min}(COMCL) \geqslant x_{min}(WMCL) \\ y_{min}(COMCL) \geqslant y_{min}(WMCL) \\ x_{max}(COMCL) \leqslant x_{max}(WMCL) \\ y_{max}(COMCL) \leqslant y_{max}(WMCL) \end{cases} \tag{17}$$

The final bounding box built in COMCL is not bigger than the one built in WMCL even if the same anchor box has been built for both schemes. As a result, the sampling area in our scheme is more accurate than that of the WMCL scheme.

4.3. Sampling and optimized filtering

The random sampling process is performed just like MCB. The samples are randomly generated in the area of a bounding box. The samples should be filtered by the optimized constraint conditions in *Constraint*$_{now}$ and *lastConstraint*. The filtering method is:

$$filter(l_t) = L^i \leqslant d(l_t, s) \leqslant U^i \text{ for each } C_i = (s, L^i, U^i) \tag{18}$$

The samples which cannot pass the filtering equation will be discarded. The weighted method for the samples passed the filtering examination is taken to improve the localization accuracy just like WMCL.

5. Performance evaluation through simulation

Our simulation, which adopts the same network model and motion model as those in MCB [24] and in WMCL [31] for a fair comparison, reuses and extends the simulation models and the parameters settings used in MCL [6]. The time is divided into discrete time units and in each time unit all the sensor nodes calculate their positions after the beacon nodes broadcast their real positions. The value of the maximum speed v_{max} is abbreviated as the maximum distance a node travels in one time unit. In order to have a fair comparison between the proposed scheme and other localization schemes, the duration of a time unit in the real time unit in terms of second has not been specified and the distance is also measured with the relevant unit in the simulation experiments. A modified Random Waypoint mobility model proposed for the MCL algorithm is used as the motion model. Nodes are unaware of their velocities and directions, but have a known maximum velocity v_{max}. Instead of choosing a certain speed for the destinations, nodes randomly vary their speed at each movement. The pause time is set to 0, so the average speed is $v_{max}/2$ when speed is chosen randomly between 0 and v_{max}. We assume that the communication range r is set to be 50 units and all the nodes are uniformly deployed in a $500 * 500$ region.

Q1 Please cite this article in press as: Z. Wang et al., Efficient localization for mobile sensor networks based on constraint rules optimized Monte Carlo method, Comput. Netw. (2013), http://dx.doi.org/10.1016/j.comnet.2013.06.010

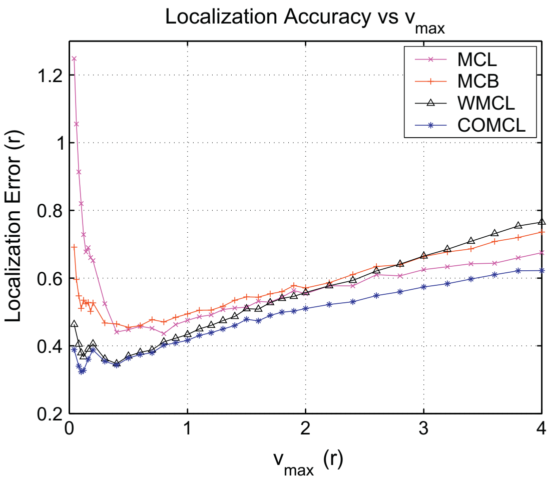


Fig. 6. How v_{max} affects the localization accuracy.

The default values of the node and beacon densities are: $n_d = 10$, $s_d = 1$. Thus, in most cases, altogether 320 nodes, which includes 288 sensor nodes and 32 beacon nodes (*seeds*), are deployed. The maximum number of effective samples is set to 50. For each of different parameters, the simulation has been repeated for 9 times. And in each time of the execution, the simulation program has been run for 1000 time units. And only the localization errors between time unit 600 and time unit 1000 have been averaged. The same way to obtain the statistical results can be found in [31], which can produce the smooth curves in the simulations. In the extensive simulations, it is found that above settings may yield stable results.

5.1. Localization accuracy

Localization accuracy is a very important metric in evaluating localization schemes. The localization accuracy in a mobile networks can be impacted by the maximum moving speed v_{max}, the node degree n_d and the seed degree s_d. When the moving speed increases, on the one hand, the size of the possible area which a node locates in may increase so that the effective samples are too scattered to make an accurate position prediction, on the other hand, the moving node may meet more neighbor beacon nodes and get more constraint conditions to reduce the bounding box or filer out more useless samples to improve localization accuracy. Fig. 6 shows how the localization accuracy of different algorithms varies when v_{max} increases.

When v_{max} is in a range of $0 - 0.4r$, the localization error of all of the schemes declines more or less. This is because the node can meet more neighbor seeds to increase the constraint conditions. COMCL yields the least localization error even when v_{max} ranges in very small values due to the accurate bounding box built with the optimized constraint rules.

When $v_{max} > 0.5r$, the localization error of different schemes rises gradually. Since the effect of enlarged possible area overwhelms the effect of increased neighbor seeds gradually, the localization accuracy trends to decline. When $v_{max} > 2.5r$, the localization error of WMCL tends to be the largest one among the four schemes and COMCL still yields the least localization error. This is due to the optimization adopted in WMCL, which mainly comes from the localization information from the neighbor sensor nodes, cannot bring more benefit when v_{max} reaches a relative high value. However, COMCL utilizes the information in time unit $t - 1$ from neighbor seeds, which are aware of their accurate positions, to construct more effective constraint rules.

Fig. 7 shows how n_d and s_d affects the localization accuracy when $v_{max} = 2r$. The localization error trends to decline as the n_d or s_d increases. When n_d varies from 2 to 10, the localization error declines from about $0.9r$ to about $0.5r$. When s_d varies from 0.5 to 3.5, the localization error declines from 1r to about $0.25r$. Higher density of nodes in certain horizon yields better localization accuracy. However, when the nodes density goes up to some value, the increase of localization accuracy becomes insignificant. The effect of the density of the seeds on localization accuracy is more important than the density of the nodes. The COMCL has introduced more effective constraint rules from the neighbor seeds so as to achieve better localization accuracy.

The convergence trend of the localization error is showed in Fig. 8. MCL spends about 10 time units to reach a stable value. However, the other three schemes spend only 4 time units to reach a stable value, which shows the effect of the constraint rules on building a more compact sampling area.

When considering packets loss, if the MAC layer protocol is the CSMA/CA scheme, the acknowledgment and retransmission mechanism will ensure the reliable transmission of the unicasting packets. So we mainly focus on the loss of broadcasting packets which delivering localization information from the neighbor nodes. In the simulation, among all the broadcasting packets which delivering localization information, we set a certain packets loss rate and let the packets drops randomly as the packets loss rate so that the constraint rules built on the observations will not be as sufficient as in the ideal scenario.

When the node density n_d varies from 6 to 20, the influence of packet loss on the localization accuracy of the WMCL and the proposed scheme are showed in Fig. 9. For the referenced scheme WMCL, when packet loss rate is 5%, the localization error rises about $0.01 - 0.02r$ for different node density n_d. When packet loss rate goes up from 5% to 10%, the localization error rises slightly when n_d values are 10, 12, 14, 18, and 20 while keep nearly the same value when n_d values are 6, 8 and 16. The packet loss rates do incur the rise of the localization error, but the influence is trivial especially when the constraint rules are nearly not affected by the lost packets. For the proposed scheme, the influence of packet loss on the localization error is weaker than the referenced scheme. The newly added constrain rules of our scheme and the employment of the rules on the filtering step reduces the impact of packet loss.

5.2. Size of the sampling area

For the schemes using bounding box technique, the size of the bounding box represents the scope of the sampling

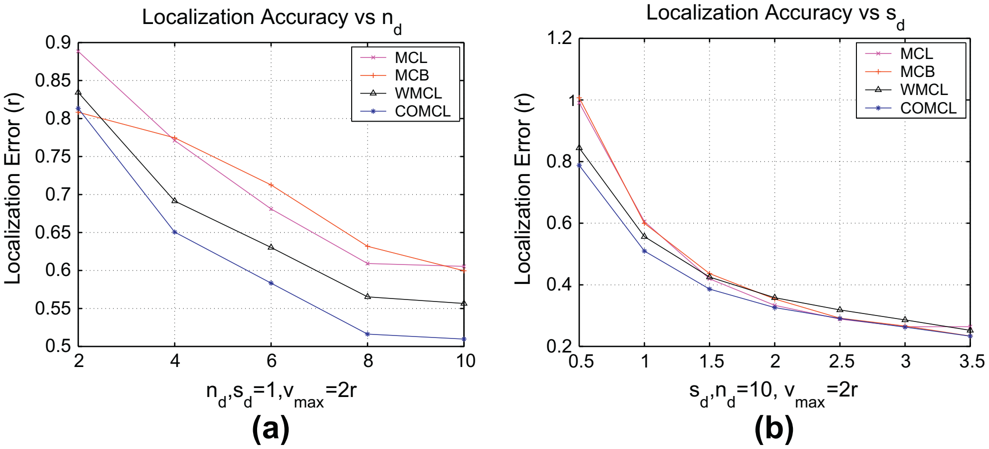


Fig. 7. How n_d, s_d affects the localization accuracy.

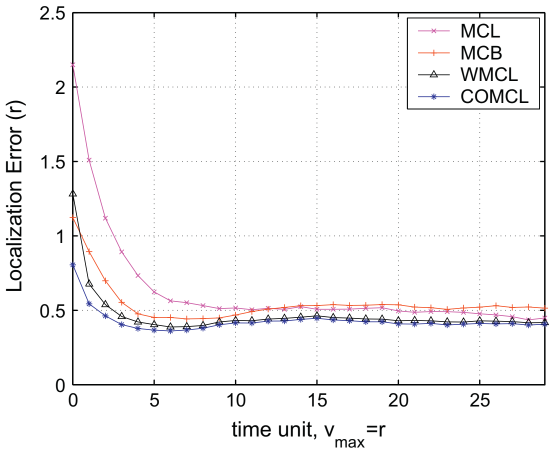


Fig. 8. Convergence of localization error.

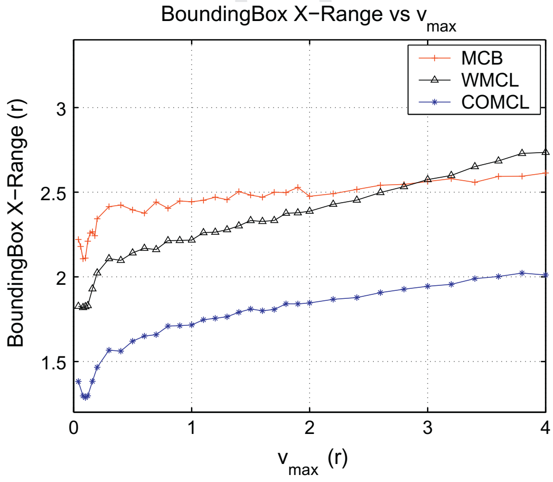


Fig. 10. Size of the bounding box when v_{max} varies.

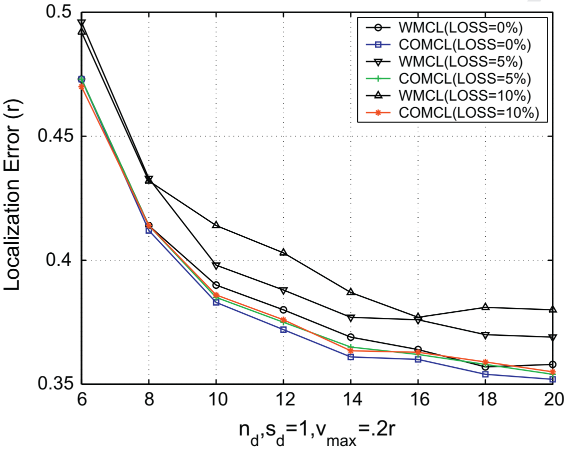


Fig. 9. Influence of packet loss on localization error.

area. If the size of the bounding box could be reduced, the scope of the samples' coordinates would be reduced which means the possible improvement of the localization accuracy and efficiency. In the following, we try to evaluate the size of the bounding box of different schemes. The average sizes of the bounding boxes in x-axis are compared in Fig. 10.

For all of the schemes, the average size of the bounding box increases as v_{max} increases. When v_{max} is in the range of $(0.1r, 4r)$, the average size of the bounding box of the COMCL is about $0.67r$ less than MCB and $0.58r$ less than WMCL. When v_{max} is below $0.1r$, the average size of the bounding box of the COMCL is about $0.82r$ less than MCB and $0.48r$ less than WMCL.

Fig. 11a and b show how n_d and s_d affects the average size of the bounding box when $v_{max} = 2r$. The average size of the bounding box decreases as n_d decreases. When $n_d = 4$ in COMCL, the size is as small as the situation when $n_d = 10$ in WMCL. No matter how n_d and s_d varies, COMCL always yields the least value of the average size of the bounding box, which verifies the effect of the optimized constraint rules in our scheme.

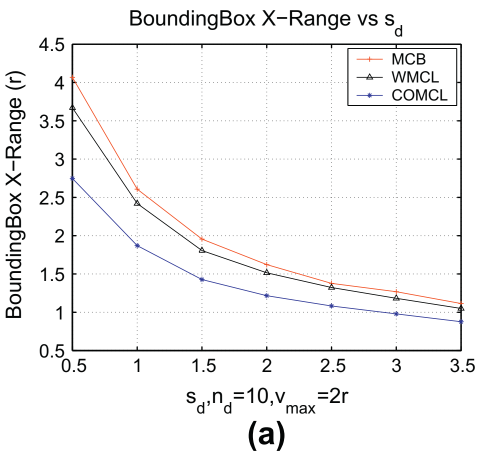


Fig. 11. Size of the bounding box when n_d or s_d varies.

5.3. Number of sampling

The numbers of sampling of different schemes are displayed in Fig. 12a. MCL yields the largest number of sampling because it has not used a bounding box as the sampling area. The detailed comparison on the schemes using bounding box is displayed in Fig. 12b. When v_{max} > 0.5r, the number of sampling in COMCL is the least one of the three schemes. The number of sampling in COMCL is about 20–30% less than the other two schemes. Fig. 12c and d show COMCL always yields the least number of sampling no matter how n_d and s_d varies.

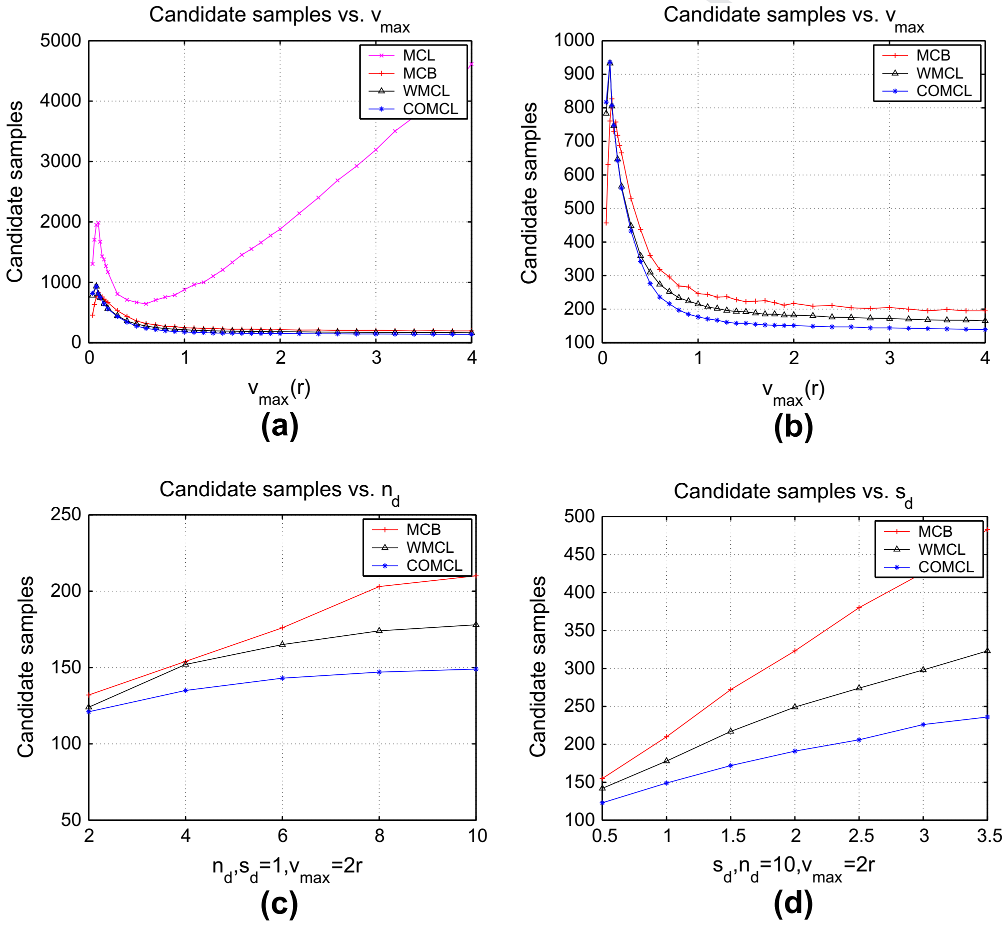


Fig. 12. Comparison on the number of sampling.

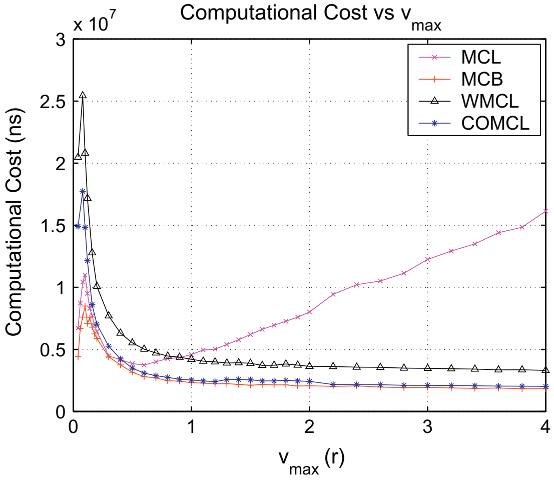


Fig. 13. Comparison on computation cost.

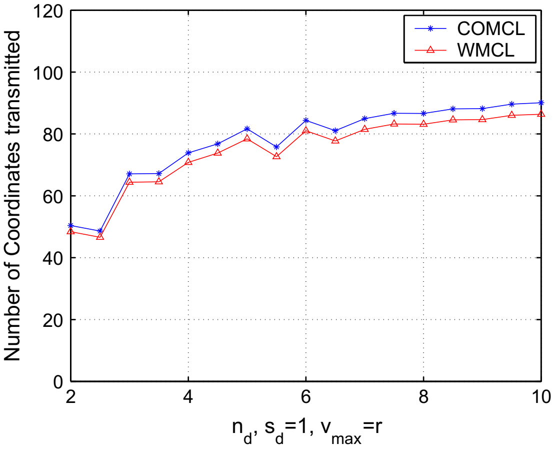


Fig. 14. Comparison on communication cost.

To generate 50 valid samples, MCL always takes the largest number of attempts to randomly generate candidate samples. As far as the sampling efficiency is concerned, we sort the four schemes as *COMCL > WMCL > MCB > MCL* according to the results of our simulation.

5.4. Computation cost

In SMC-based localization schemes, the computational cost mainly contains the cost in generating candidate samples and the cost in filtering the candidate samples. There are two key operations in filtering the candidate samples: computing the distance between two positions and comparing the distance with a predefined value (e.g., the lower or upper bound to a *seed*).

According to the experiments on Micaz platform [32] in WMCL, the cost of generating a candidate sample is much higher than filtering it. If the number of sampling can be reduced effectively, the overall computation cost will decrease accordingly. As what Fig. 12 has displayed, COMCL wins the advantage in computation cost due to the least number of sampling.

The average computation time spent in localization process of each time unit is introduced for comparison on computation cost. Fig. 13 shows the computation cost of different schemes in the metric of computation time. Although both COMCL and WMCL employs more optimization steps to reduce the size of the bounding box than MCB, the corresponding computation overhead does not lead to a significant rise of the overall cost. When $v_{max} > 0.5r$, the computation time of COMCL is about 1.1 times of MCB, and 0.6 times of WMCL. MCL yields the largest computation time among the schemes, which is consistent with the results in Fig. 12a.

5.5. Communication cost

The communication cost for the SMC-based schemes comes mainly from messages carrying position information. Each seed needs to broadcast its position information and each node needs to rebroadcast the information it received from its neighbor seeds. In addition, a node in WMCL needs to broadcast the positions of the samples in last time unit. Furthermore, a node in COMCL needs to broadcast the scope of the bounding box in last time unit. Thus, we try to compare the average number of positions transmitted per node in COMCL and WMCL. The communication cost in the metric of number of positions is compared in Fig. 14. The average number of positions transmitted by a node in COMCL is about 1.04 times of WMCL.

6. Conclusion

Most of the existing SMC-based localization schemes build their sampling areas according to the constraint rules within the coverage scopes of one-hop or two-hop beacon neighbors. They filter the candidate samples with the same set of constraint rules. The WMCL employs position information of neighbor sensor nodes to reduce the sampling area but uses the same set of constraint rules in filtering. In this paper, as the major contribution, we have deduced new constraint rules based on $seed_{t-1}$ by exploiting the position relationship between a node and its beacon neighbor after the beacon has finished a time-going-backward or time-going-forward movement. Furthermore, we have deduced the change of the position relationship between a node and its beacon neighbor after moving from time unit $t-1$ to t to optimize the existing constraint rules. Since the positions of a beacon neighbor in time unit $t-1$ and t are known to the node, the moving distance and direction of the beacon can be known too. The introduction of this kind of observation on the beacon's movement in a time unit reduce the uncertainty of the node's position so that the corresponding constraint rules built on this observation can reduce the sampling area effectively. The new and optimized constraint rules are not only employed in the sampling stage but also in the filtering stage to supply more strict filtering conditions. By the extensive simulations, the performance of our scheme has been verified. The proposed COMCL can achieve more accurate localization results even when v_{max} is high up to 4*r*. The optimized

Q1 Please cite this article in press as: Z. Wang et al., Efficient localization for mobile sensor networks based on constraint rules optimized Monte Carlo method, Comput. Netw. (2013), http://dx.doi.org/10.1016/j.comnet.2013.06.010

constraint rules are able to cut the sampling area and reduce the number of sampling. The proposed COMCL algorithm has been implemented and evaluated in a real sensor network composed of nodes built with Micaz Motes in [31]. It will be more valuable to compare the proposed scheme with the existing solutions in a larger scaled mobile WSN, which will be a part of our future work.

References

[1] L. Doherty, K.S. Pister, L.E. Ghaoui, Convex position estimation in wireless sensor networks, Proc. IEEE INFOCOM 3 (2001) 1655–1663.

[2] A. Nasipuri, K. Li, A directionality based location discovery scheme for wireless sensor networks, in: Proc. ACM International Workshop Wireless Sensor Networks and Applications (WSNA '02), 2002.

[3] T. He, C. Huang, B.M. Blum, J.A. Stankovic, T. Abdelzaher, Range-free localization schemes for large scale sensor networks, in: Proc. ACM MobiCom, 2003, pp. 81–95.

[4] P. Biswas, Y. Ye, Semidefinite programming for ad hoc wireless sensor network localization, in: Proc. Third Int'l Symp. Processing in Sensor Networks (IPSN '04), 2004.

[5] D.K. Goldenberg, P. Bihler, M. Cao, J. Fang, B.D. Anderson, A.S.Morse, Y.R. Yang, Localization in sparse networks using sweeps, in: Proc. ACM MobiCom, 2006, pp. 110–121.

[6] L. Hu, D. Evans, Localization for mobile sensor networks, in: Proc. ACM MobiCom, 2004, pp. 45–57.

[7] M. Li, Y. Liu, Rendered path: range-free localization in anisotropic sensor networks with holes, in: Proc. ACM MobiCom, 2007, pp. 51–62.

[8] G. Wu, S. Wang, B. Wang, Y. Dong, S. Yan, A novel range-free localization based on regulated neighborhood distance for wireless ad hoc and sensor networks, Computer Networks 56 (16) (2012) 3581–3593.

[9] D. Niculescu, B. Nath, Ad-hoc positioning system (APS), in: Proc. GLOBECOM, 2001, pp. 2926–2931.

[10] M. Rudafshani, S. Datta, Localization in wireless sensor networks, in: Proc. Sixth Int'l Conf. Information Processing in Sensor Networks (IPSN '07), 2007, pp. 51–60.

[11] W. Liu, D. Wang, H. Jiang, W. Liu, C. Wang, Approximate convex decomposition based localization in wireless sensor networks, in: Proceedings of IEEE INFOCOM, 2012.

[12] Y. Shang, W. Ruml, Y. Zhang, F. MPJ, Localization from mere connectivity, in: Proc. ACM MobiHoc, 2003.

[13] D. He, L. Cui, H. Huang, M. Ma, Design and verification of enhanced secure localization scheme in wireless sensor networks, IEEE Transactions on Parallel and Distributed Systems 20 (7) (2009) 1050–1058.

[14] Y. Wang, S. Lederer, J. Gao, Connectivity-based sensor network localization with incremental delaunay refinement method, in: Proc. IEEE INFOCOM, 2009.

[15] H. Wu, C. Wang, N.-F. Tzeng, Novel self-configurable positioning technique for multihop wireless networks, IEEE/ACM Transactions Networking 13 (3) (2005) 609–621.

[16] B.H. Wellenhoff, H. Lichtenegger, J. Collins, Global Positioning System: Theory and Practice, fourth ed., Springer Verlag, 1997.

[17] P. Bahl, V.N. Padmanabhan, RADAR: an in-building RF-based user location and tracking system, in: IEEE InfoCom 2000, 2000.

[18] A. Savvides, C. Han, M.B. Strivastava, Dynamic fine-grained localization in Ad-Hoc networks of sensors, in: MobiCom, 2001.

[19] D. Niculescu, B. Nath, Ad hoc positioning system (APS) using AoA, in: Proc. IEEE InfoCom 2003, 2003, pp.1734–1743.

[20] N. Bulusu, J. Heidemann, D. Estrin, GPS-less low cost outdoor localization for very small devices, IEEE Personal Communications Magazine (2000).

[21] T. He, C. Huang, B.M. Blum, J.A. Stankovic, T. Abdelzaher, Range-free localization schemes for large scale sensor networks, in: MobiCom, 2003.

[22] R. Nagpal, H. Shrobe, J. Bachrach, Organizing a global coordinate system from local information on an ad hoc sensor network, in: 2nd Int'l Workshop on Information Processing in Sensor Networks (IPSN), 2003.

[23] D. Niculescu, B. Nath, DV based positioning in ad hoc networks, Journal of Telecommunication Systems (2003).

[24] A. Baggio, K. Langendoen, Monte-carlo localization for mobile wireless sensor networks, in: Proc. Conf. Mobile Ad-Hoc and Sensor Networks (MSN '06), 2006, pp. 317–328.

[25] B. Dil, S. Dulman, P. Havinga, Range-based localization in mobile sensor networks, in: Proc. Third European Workshop Wireless Sensor Networks (EWSN '06), 2006, pp. 164–179.

[26] W. Wang, Q. Zhu, Varying the sample number for monte carlo localization in mobile sensor networks, in: Proc. IEEE Int'l Multi-Symp. Computer and Computational Sciences, 2007, pp. 490–495.

[27] J. Yi, S. Yang, H. Cha, Multi-hop-based monte carlo localization for mobile sensor networks, in: Proc. Fourth Ann. IEEE Comm. Soc. Conf. Sensor, Mesh, and Ad Hoc Comm. and Networks (SECON '07), 2007, pp. 162–171.

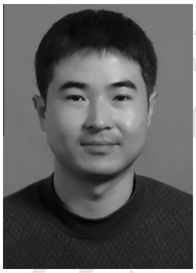
[28] E. Stevens-Navarro, V. Vivekanandan, V. Wong, Dual and mixture monte carlo localization algorithms for mobile wireless sensor networks, in: Proc. IEEE Wireless Comm. and Networking Conf. (WCNC '07), 2007, pp. 4024–4028.

[29] M. Rudafshani, S. Datta, Localization in wireless sensor networks, in: Proc. Sixth Intl Conf. Information Processing in Sensor Networks (IPSN 07), 2007, pp. 51–60.

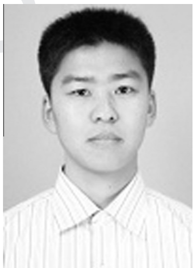
[30] M. Torrent-Moreno, F. Schmidt-Eisenlohr, H. Fussler, H. Hartenstein, Effects of a realistic channel model on packet forwarding in vehicular ad hoc networks, in: Proc. Wireless Communications and Networking Conference (WCNC), vol. 1, 2006, pp. 385C391.

[31] S. Zhang, J. Cao, L. Chen, D. Chen, Accurate and energy-efficient range-free localization for mobile sensor networks, IEEE Transactions Mobile Computing 9 (6) (2010) 897–910.

[32] http://www.memsic.com/products/wireless-sensor-networks/wireless-modules.html



Ze Wang received the BE degree and the ME degree both from Xi'an Jiaotong University, China, in 1998 and 2001 respectively, and the PhD degree in computer applications technology from Northeastern University, China, in 2004. He has been a visiting scholar in the School of Electrical and Electronic Engineering at Nanyang Technological University in Singapore in 2011. He has been an associate professor in the School of Computer Science and Software at Tianjin Polytechnic University in China since November 2006. He has been the head of Department of Network Engineering in the School of Computer Science and Software at Tianjin Polytechnic University in China since July 2011. His primary research interests include network security, mobile computing and distributed systems.



Yunlong Wang received the BE degree of computer software and theory and the ME degree of computer applications technology both from Tianjin Polytechnic University, China, in 2009 and 2012 respectively. His primary research interests include wireless sensor networks and mobile computing.

Maode Ma received his BE degree from Tsinghua University in 1982, his ME degree from Tianjin University in 1991 and his Ph.D. degree in computer science from Hong Kong University of Science and Technology in 1999. Now, Dr. Ma is a tenured Associate Professor in the School of Electrical and Electronic Engineering at Nanyang Technological University in Singapore. He has extensive research interests including wireless networking and wireless network security. He has been a member of the technical program committees for more than 100 international conferences. He has been a general chair, technical symposium chair, tutorial chair, publication chair,

Q1 Please cite this article in press as: Z. Wang et al., Efficient localization for mobile sensor networks based on constraint rules optimized Monte Carlo method, Comput. Netw. (2013), http://dx.doi.org/10.1016/j.comnet.2013.06.010

publicity chair and session chair for more than 50 international conferences. Dr. Ma has more than 180 international academic publications. Dr. Ma is the Editor-in-Chief of International Journal of Electronic Transport and International Journal of Wireless Networking and Security. He currently serves as an Associate Editor for IEEE Communications Letters, a Senior Editor for IEEE Communications Surveys and Tutorials, and an Associate Editor for International Journal of Network and Computer Applications, International Journal of Security and Communication Networks, and International Journal of Wireless Communications and Mobile Computing. Dr. Ma is a senior member of IEEE Communication Society and a member of a few technical committees in the IEEE Communication Society.

Jigang Wu received the BSc degree in computational mathematics from Lanzhou University, China, in 1983, and the doctoral degree in computer software and theory from the University of Science and Technology of China (USTC) in 2000. He was with the Department of Computer Science of Lanzhou University, China, from 1983 to 1993, as an assistant professor followed by lecturer. He was with the Department of Computer Science and Engineering of Yantai University, China, from 1993 to 2000, as a lecturer followed by associate professor. He was with the Center for High Performance Embedded Systems, School of Computer Engineering, Nanyang Technological University, Singapore, from 2000 to 2009, as a postdoctoral fellow followed by research fellow. In 2009, he joined as a full professor in the School of Computer Science and Software, Tianjin Polytechnic University, China. He has published more than 100 technical papers including journals in the IEEE Transactions, IEE Proceedings, and other reputed international journals. His research interests include in distributed and parallel computing, reconfigurable VLSI design, hardware/software codesign and combinatorial search.