An Adaptive Localization Approach for Wireless Sensor Networks Based on Gauss-Markov Mobility Model

ZHONG Zhi[1] LUO Da-Yong[1] LIU Shao-Qiang[1] FAN Xiao-Ping[1] QU Zhi-Hua[1]

Abstract This paper proposes an adaptive localization approach for wireless sensor networks based on Gauss-Markov mobility model. In the approach, the perpendicular bisector strategy, the virtual repulsive strategy, and the velocity adjustment strategy are properly combined to enhance localization efficiency. The velocity adjustment strategy causes that the mobile anchor node automatically tunes its velocity. The perpendicular bisector strategy locally adjusts trajectory for the mobile anchor node, which ensures that unknown nodes obtain enough non-collinear anchor coordinates as soon as possible. The virtual repulsive strategy impels that the mobile anchor node rapidly leaves the communication range of location-aware nodes or returns to the surveillance region after the mobile anchor node was out of the boundary. Both theoretical analysis and simulation studies show that this approach can increase localization accuracy, consume less energy, and cover more surveillance region during the same period than virtual beacons-energy ratios localization scheme using the Gauss-Markov mobility model.

Key words Wireless sensor network, perpendicular bisector strategy, virtual repulsive force strategy, velocity adjustment strategy, Gauss-Markov mobility model

DOI 10.3724/SP.J.1004.2010.01557

In wireless sensor networks, the sensor localization plays a critical role because the location information is typically useful for coverage, deployment, routing, location service, target tracking, and rescue[1]. In addition, some routing algorithms with the aid of geographic location, including the geographical adaptive fidelity (GAF)[2] routing algorithm, the geographic and energy-aware (GEAR)[3] routing algorithm, have been proposed.

We classify the existing localization algorithms into two categories: the stationary anchor localization algorithms and the mobile anchor localization algorithms. The deployment and number of anchor nodes could greatly influence localization accuracy[3−4]. However, the more anchor nodes are, the larger the cost of deployment network is. Once all the nodes are located, anchor nodes will be not so important. We, therefore, could use a mobile anchor node (MN) instead of all stationary anchor nodes, which can reduce cost. Furthermore, since the MN can move to blind areas where static anchor nodes do not cover, it may communicate with all the nodes directly, which could enhance localization accuracy. In addition, employing an MN may eliminate the need of deploying stationary anchors. The nodes of initially unknown location will be called unknown nodes.

In the paper, we present an adaptive localization approach for wireless sensor networks based on Gauss-Markov mobility model. The MN can adaptively adjust its velocity and direction based on the perpendicular bisector strategy, the virtual repulsive force strategy, and the velocity adjustment strategy. Receiving the acknowledgement packets, the MN speeds down and moves based on the perpendicular bisector strategy; otherwise, the MN speeds up to move to other parts of the surveillance region. After becoming a location-aware node, this node will exert virtual repulsive force on the MN, which impels MN to move away from this location-aware node as soon as possible. When the MN has left the surveillance region, the virtual repulsive force strategy could ensure the MN go back quickly. Hence, the proposed algorithm can provide lower residue unknown node ratio, less running time, and higher localization accuracy in a short time. Finally, we utilize acoustic energy measurements[5] to evaluate the performance of the proposed algorithm.

The rest of the paper is organized as follows. Section 1 summarizes similar works in previous localization researches, especially about methods based on the mobile anchor node. In Section 2, we describe the proposed algorithm, including the perpendicular bisector strategy, the virtual repulsive force strategy, and the velocity adjustment strategy. The simulation results and conclusion are provided in Sections 3 and 4, respectively.

1 Related work

Stationary anchor localization algorithms utilize stationary anchor information for localization, which can be classified as range-based and range-free. The range-based algorithm uses absolute node-to-node distance or angle between neighboring sensors to estimate location. Common approaches for distance or angle estimation include received signal strength indicator (RSSI)[6], time of arrival (TOA)[7], time difference of arrival (TDOA)[8], angle of arrival (AOA)[9], and distributed weighted-multidimensional scaling based on relative error (DWMDS(E))[10]. Range-based approaches can obtain more accurate measurements, but they require complex and expensive hardware. The range-free algorithm only depends on the connectivity of the network and the received messages. Typical range-free algorithms include centroid algorithm[11], distance vector-hop (DV-Hop) algorithm[12−13], approximate point-in-triangulation test (APIT) algorithm[14], and amorphous algorithm[15]. Although the range-free approach cannot be accomplished with higher precision as the range-based, it is more economical for the large-scale sensor networks.

Mobile anchor localization algorithms also compute node-to-node distances by RSSI, TOA, TDOA, and AOA. Zhang et al.[16] proposed very low energy consumption wireless sensor localization for dangerous environments with single mobile anchor node. But this algorithm cannot ensure each node receives three non-collinear anchor coordinates. Koutsonikolas et al.[17] studied the problem of path planning for mobile anchor to reduce localization error. Zhang et al.[18] proposed a range-free localization scheme using mobile anchor nodes. When running once,

Manuscript received September 18, 2009; accepted August 1, 2010
Supported by National Natural Science Foundation of China (60776834, 60870010)
1. School of Information Science and Engineering, Central South University, Changsha 410083, P. R. China

this algorithm only located a part of nodes. In order to increase localization efficiency, the movement mode of the MN needs to be improved. Kuang et al.[19] proposed a virtual beacons-energy ratios localization (VB-ERL) scheme using the Gauss-Markov mobility model, which was fully distributed and did not need inter-sensor communication. The Gauss-Markov mobility model provided movement patterns which might be expected in the real-world for the mobile anchor node, which took full advantage of the correlation between the current velocity and location of the MN and its future velocity and location. But random variables brought certain randomness to the Gauss-Markov mobility model, which caused the probabilities the MN moved to any locations within the surveillance region were equivalent, i.e., the Gauss-Markov mobility model could make the MN traverse all the surveillance region when running time was long enough. However, it was this randomness that could not ensure all unknown nodes obtained enough non-collinear anchor coordinates, which had adverse effect on localization efficiency. Moreover, the VB-ERL did not consider how to shorten the unnecessary movement time for the MN, i.e., 1) When entering into the communication range of a location-aware node, the MN was unable to leave this location-aware node immediately for the purpose of moving to other parts of the surveillance region; 2) After being out of the boundary, the MN would take a long time to return to the surveillance region; 3) When receiving an acknowledgement or not, the MN cannot automatically adjust its velocity in order to ensure that unknown nodes receive anchor coordinates as many as possible. The above mentioned aspects are precisely what this paper aims to improve.

2 Proposed approach

2.1 Assumptions

The whole network consists of stationary unknown nodes and a mobile anchor node; the MN obtains its location coordinates by GPS; as receiving a packet sent by the MN, unknown nodes send an acknowledgement packet immediately and both of them have the same communication radius and periodicity; we regard the location where the MN sends packets as an anchor; after being a location-aware node, this node will attach its coordinate in the acknowledgement message, accordingly, the MN receives this acknowledgement and conserves the coordinate of location-aware node.

2.2 Acoustic energy measurement localization mechanism

In [5], localizations based on the acoustic energy measurement have several potential advantages, including low intersensor communication requirement, robustness with respect to parameter perturbations and measurement noise, and low-complexity implementation.

The signal energy measured on the i-th unknown node over a time interval, denoted by g_i, can be expressed as follows:

$$g_i = \frac{B}{|\boldsymbol{X} - \boldsymbol{r}_i|^{\lambda}} + \varepsilon_i \tag{1}$$

where B is a scalar denoting the energy emitted by the mobile node; $\boldsymbol{X}$ is a vector denoting the coordinates of the unknown node; $\boldsymbol{r}_i$ is a vector denoting the i-th coordinates of the anchor; $\lambda\,(\approx 2)$ is an energy decay factor; and ε_i is measurement noise, based on the central limit theorem, it can be approximated as a Gaussian distribution.

Approximating the additive noise term ε_i in (1) by its mean value μ_i, we can compute the energy ratio k_{ij} of the i-th and j-th unknown nodes as follows:

$$k_{ij} = \left(\frac{B - \mu_i}{B - \mu_j}\right)^{\frac{-1}{\lambda}} = \frac{|\boldsymbol{X} - \boldsymbol{r}_i|}{|\boldsymbol{X} - \boldsymbol{r}_j|} \tag{2}$$

Note that for $0 < k_{ij} \neq 1$, all the possible coordinates of unknown nodes $\boldsymbol{X}$ that satisfy (2) reside on a d-dimensional hypersphere described by the equation

$$|\boldsymbol{X} - \boldsymbol{c}_{ij}|^2 = \rho_{ij}^2 \tag{3}$$

where the center $\boldsymbol{c}_{ij}$ and the radius ρ_{ij} of this hypersphere associated with unknown node i and j are given by

$$\boldsymbol{c}_{ij} = \frac{\boldsymbol{r}_i - k_{ij}^2 \boldsymbol{r}_i}{1 - k_{ij}^2}, \quad \rho_{ij} = \frac{k_{ij}|\boldsymbol{r}_i - \boldsymbol{r}_j|}{1 - k_{ij}^2} \tag{4}$$

When the dimension is 2, such a hypersphere is a circle. When the dimension is 3, it is a sphere.

We apply (1) $\sim$ (4) to the node localization. The potential unknown nodes location can be restricted to a hypersphere whose center and radius are function of the energy ratio and the two unknown nodes location. We apply the least square criterion to solve cost function which is defined as

$$\begin{cases} J = \sum\limits_{m_1=1}^{M_1} | \, ||\boldsymbol{X} - \boldsymbol{c}_{m_1}|| - \rho_{m_1}|^2 + \sum\limits_{m_2=1}^{M_2} |\omega_{m_2}^{\mathrm{T}} \boldsymbol{X} - \psi_{M_2}|^2 \\ M_1 + M_2 = M \\ \omega_{ij}^{\mathrm{T}} = \boldsymbol{r}_i - \boldsymbol{r}_j \\ \psi_{ij} = \dfrac{|\boldsymbol{r}_i|^2 - |\boldsymbol{r}_j|^2}{2} \end{cases} \tag{5}$$

where m_1 and m_2 are indices of the energy ratios computed between different pairs of unknown node energy readings. M_1 is the number of hyperspheres and M_2 is the number of hyperplanes.

Since the cost function J is a nonlinear equation, we can minimize it by nonlinear optimization method, such as the Newton method.

2.3 Velocity adjustment strategy

The Gauss-Markov mobility model was originally presented for the simulation of a personal communication service network (PCS)[20−21]. It can be described by[22]

$$s_n = \alpha s_{n-1} + (1 - \alpha)\overline{s} + \sqrt{(1 - \alpha^2)}s_{x_{n-1}} \tag{6}$$

$$d_n = \alpha d_{n-1} + (1 - \alpha)\overline{d} + \sqrt{(1 - \alpha^2)}d_{x_{n-1}} \tag{7}$$

where s_n and d_n are the new velocity and direction of the MN at time interval n; α, where $0 \leq \alpha \leq 1$, is the tuning parameter used to vary the randomness; $\overline{s}$ and $\overline{d}$ are constants representing the mean value of velocity and direction as $n \to \infty$; $s_{x_{n-1}}$ and $d_{x_{n-1}}$ are random variables from a Gauss distribution.

At each time interval, the next location is calculated based on the current location, velocity, and direction of movement. Specifically, at time interval n, the location of MN is given by the equation:

$$x_n = x_{n-1} + s_{n-1}\mathrm{cos}d_{n-1} \tag{8}$$

$$y_n = y_{n-1} + s_{n-1}\mathrm{sin}d_{n-1} \tag{9}$$

where (x_n, y_n) and (x_{n-1}, y_{n-1}) are the x and y coordinates of the MN at the n-th and $(n-1)$-th time intervals, respectively, and s_{n-1} and d_{n-1} are the velocity and direction of the MN, respectively, at the $(n-1)$-th time interval.

We can recursively expand (6) to express s_n explicitly in terms of the initial velocity s_0,

$$s_n = \alpha^n s_0 + (1-\alpha^n)\overline{s} + \sqrt{(1-\alpha^2)}\sum_{i=0}^{n-1}\alpha^{n-i-1}s_{x_i} \quad (10)$$

If $t = 0\,\mathrm{s}$, the MN is static, i.e., $s_0 = 0$. The new velocity at n-th is only related to the mean value of the velocity and the random variable. We set $\alpha = 0.75$[21] as $t = 10\,\mathrm{s}$, $(1-\alpha^n) = 0.94$. The larger the value of n is, the smaller the value of the polynomial $(1-\alpha^n)$ is. So, (10) can be simplified to

$$s_n \approx \overline{s} + \sqrt{(1-\alpha^2)}\sum_{i=0}^{n-1}\alpha^{n-i-1}s_{x_i}, \quad t > 10\,\mathrm{s} \quad (11)$$

Since s_{x_i} is Gaussian with zero mean for any constant $\overline{s}$, s_n is a Gaussian process with mean $\overline{s}$, i.e., the value of new velocity is in the neighboring range of $\overline{s}$ when running time is more than 10 s.

According to the introduction offered above, we can conclude the MN cannot automatically adjust its velocity in accordance with environmental change. So how to choose a new velocity for the MN in different environments is a very important question. In this strategy, the velocity will be automatically adjusted and details are as follows:

1) Receiving acknowledgements, the MN moves at a low speed at next time interval, which ensures the unknown node can receive anchor coordinates as many as possible. We modify $\overline{s}$ in (6) to $\overline{S}_{\min}$, where $\overline{S}_{\min}$ is the minimum mean value of the velocity. And then adding a polynomial $(1-\alpha)s_{n-1}$ and subtracting a polynomial $(1-\alpha)s_{n-1}$ in (6), we obtain another expression

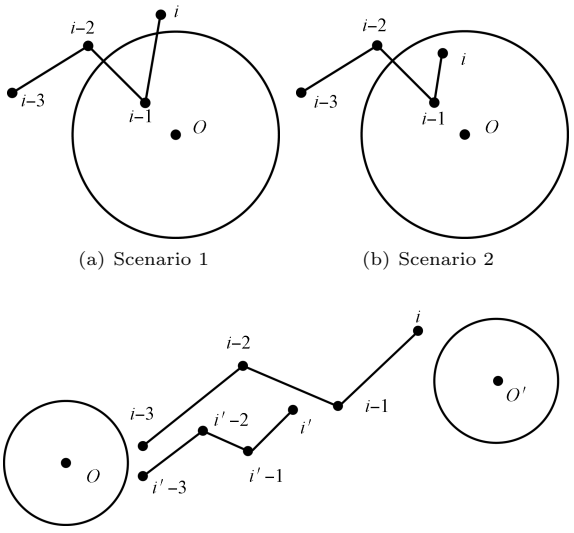
$$s_n = s_{n-1} + (1-\alpha)(\overline{S}_{\min} - s_{n-1}) + \sqrt{(1-\alpha^2)}s_{x_{n-1}} \quad (12)$$

As shown in (12), in order to decelerate immediately, we must adjust the polynomial $(1-\alpha)(\overline{S}_{\min} - s_{n-1})$ to minimum. If and only if $\overline{S}_{\min} = 0$, $(1-\alpha)(\overline{S}_{\min} - s_{n-1})$ realizes minimum, which achieves the best performance for this strategy. The simulation will verify this conclusion in Subsection 3.3. Fig. 1 (a) shows a situation where the velocity of the MN does not slow down after entering into the communication range of an unknown node. O denotes an unknown node; the circle denotes the communication range of O; $i-3$, $i-2$, $i-1$, and i denotes the coordinates of the MN in continuous time. And then O obtains only one anchor coordinate, such as $i-1$. As shown in Fig. 1 (b), the MN slows down after entering into the communication range of O, and then O can obtain two anchor coordinates. Obviously, this strategy could change the location of the MN in order that unknown nodes obtain anchor coordinates as much as possible.

2) Without any acknowledgement signal received, the MN will speed up. This ensures that the MN can fastly move to other parts of the surveillance region. Similarly, we modify $\overline{s}$ in (6) to $\overline{S}_{\max}$, where $\overline{S}_{\max}$ is the maximum mean value of the velocity. And then adding a polynomial $(1-\alpha)s_{n-1}$ and subtracting a polynomial $(1-\alpha)s_{n-1}$ in (6), we obtain new equation:

$$s_n = s_{n-1} + (1-\alpha)(\overline{S}_{\max} - s_{n-1}) + \sqrt{(1-\alpha^2)}s_{x_{n-1}} \quad (13)$$

Similar to analysis of (12), we adjust the value of $\overline{S}_{\max}$ to obtain suitable value of s_n. Further discussion for selecting the optimal value of $\overline{S}_{\max}$ is given in Subsection 3.3. As shown in Fig. 1 (c), $i-3$, $i-2$, $i-1$, and i are the trajectories which the MN speeds up; $i'-3$, $i'-2$, $i'-1$, and i' are the trajectories which the MN moves based on the Gauss-Markov mobility model. Obviously, the larger velocity could enhance mobile efficiency of the MN in non-target region.



(a) Scenario 1 (b) Scenario 2

(c) Comparison of different velocities

Fig. 1 Impact of $\overline{S}_{\min}$ and $\overline{S}_{\max}$ on the trajectory

2.4 Perpendicular bisector strategy

In (7), the change in direction caused by the random variable $d_{x_{n-1}}$ is irregular and unpredictable. Due to the randomness of the Gauss-Markov mobility model, the MN is likely to move out of the communication range of an unknown node before this unknown node obtains enough anchor coordinates. And then that unknown node just continues waiting for the arrival of the MN again. In order to ensure unknown nodes could receive enough non-collinear anchor coordinates as quickly as possible, we present the perpendicular bisector strategy, which takes partial trajectory adjustment for the MN and ensures the MN moves within the communication range of an unknown node until the number of anchor coordinates received by unknown node exceeds a predefined anchor threshold ξ. Once obtaining enough anchor coordinates, that unknown node will run acoustic energy measurement localization mechanism and shut down channel to switch to sleep state (or switch to another channel) in order to conserve energy. We will illustrate this strategy with the combination of Fig. 2, where the solid line denotes the MN has passed the movement path; the thin dashed line is the perpendicular bisector line of the connecting line of two anchors; and the thick dashed line is the movement path of the MN in future time.

We consider two different situations as running the perpendicular bisector strategy:

1) Receiving one acknowledgement from one node

When entering into the communication range of an unknown node, the MN receives only one acknowledgement as shown in Fig. 2 (a):

Step 1. Once the MN receives two acknowledgements sent by the same unknown node at continuous time, such as

$i-1$ and i, we take two points a and b in the perpendicular bisector line of the connecting line of these two anchors. The distances of a and b to two anchors are calculated by (14). θ, where $0.57 \leq \theta \leq 1.92$[1], is a tuning parameter used to adjust the coordinates of a and b; L is the Euclidean distance between two coordinate points.

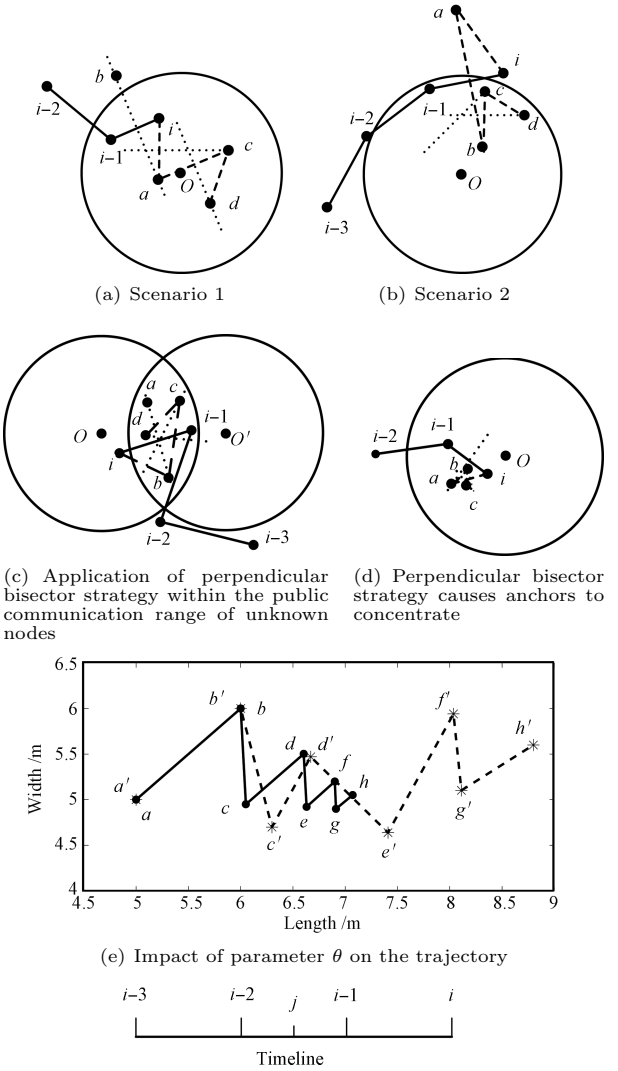
$$L_{i,j} = \theta \times L_{i,i-1}, \quad j = a, b, \cdots \tag{14}$$

Step 2. We select a point appeared earlier between two anchors $i-1$ and i which formed connecting line, such as $i-1$, and then take its last point as the reference point, such as $i-2$. The MN will move to the point a or b which is relatively farther from $i-2$ at the $(n+1)$-th time interval[2], such as a, and then sends a packet while examines whether to receive an acknowledgement sent by the same unknown node O. If no acknowledgement received, the MN moves to the other one, such as b, at the $(n+2)$-th time interval, and then check whether to receive an acknowledgement sent by O again. If the MN receives an acknowledgement, go to Step 3; if no acknowledgement, the MN moves according to the velocity adjustment strategy.

Step 3. If the number of anchor coordinates which O received are less than predefined anchor threshold ξ, we continue selecting the latest two anchors to establish a perpendicular bisector line, such as i and a. According to analysis in Step 2, we choose the point c as the next coordinate. The rest may be deduced by analogy. If this unknown node meets the localization requirement, it will attach its coordinate in the acknowledgement message. The MN receives this acknowledgement and moves based on virtual repulsive force strategy at next time interval which will be introduced in Subsection 2.5. The MN will pass $i-2$, $i-1$, i, a, c, and d successively in Fig. 2 (a).

As shown in Fig. 2 (b), the MN received only one acknowledgement and go out of the communication range of O. We still may run the perpendicular bisector strategy in this case. Likewise, the MN will pass $i-3$, $i-2$, $i-1$, i, a, b, c, d successively.

When we choose two points in a perpendicular bisector line based on (14) every time, parameter θ must be different from the last time. If θ remains fixed, anchors received by the unknown node will be aligned. When anchors are much aligned (but not totally), a little distance estimation error could cause great position estimation error due to the intersection points between circumferences could be very far from the real sensor position[23]. As shown in Fig. 2 (e), the solid line denotes the movement trajectory of the MN when θ remains fixed. The MN passes a, b, c, d, e, f, g, and h successively, but a, c, e, g and b, d, f, h are aligned, respectively. However, using the acoustic energy measurement localization mechanism, at least four anchors not locating on the same straight line are required to locate a node in a 2D sensor field; and at least five sensors not all locating on the same plane are required to locate a node in a 3D sensor field[5]. So, we must randomly choose θ among its value range every time for the purpose of avoiding anchors locating on the same straight line. The dashed line denotes the movement trajectory of the MN using different value of θ every time. Obviously, a, c, e, g and b, d, f, h have not been aligned.



(a) Scenario 1 (b) Scenario 2

(c) Application of perpendicular bisector strategy within the public communication range of unknown nodes

(d) Perpendicular bisector strategy causes anchors to concentrate

(e) Impact of parameter θ on the trajectory

(f) Sending and listening time interval

Fig. 2 Impact of perpendicular bisector strategy on the trajectory

2) Receiving more acknowledgements from multiple nodes

When coming into the public communication range of several unknown nodes, the MN will receive more acknowledgements as shown in Fig. 2 (c). $i-1$ denotes the MN is within the public communication range of O and O', i denotes the MN is within the communication range of O. If the MN could move within the public communication range as much as possible, several unknown nodes might receive anchor coordinates simultaneously, which will increase efficiency of this strategy.

Step 1. When this situation mentioned above occurs, we construct a perpendicular bisector line between i and $i-1$ as shown in Fig. 2 (c). Similarly, we take two points a and b in this perpendicular bisector line.

Step 2. Similar to Step 2 of the first situation, the only

[1] As shown in Fig. 2 (e), θ cannot be too small, otherwise, a, b, and c could be aligned approximately, which leads to lower localization accuracy. So, we define $\angle abc \geq 30^\circ$, i.e., $\theta \geq 0.57$; similarly, θ cannot be also too large, otherwise, $\angle abc \to 90^\circ$, $|bc| \to \infty$. We define $\angle abc \leq 75^\circ$, i.e., $\theta \leq 1.92$. In addition, if the value of θ is too large, point c could be out of the communication range of an unknown node, which leads to lower efficiency for the perpendicular bisector strategy.

[2] Moving to a point which is relatively farther from the reference point will make anchors scatter as shown in Fig. 2 (e); moving to a point which is relatively nearer to the reference point will cause anchors to concentrate in a relatively small region as shown in Fig. 2 (d).

difference is that we choose $0.57 \leq \theta \leq 1$ when taking the perpendicular bisector for the first time, and the MN moves to a point which is relatively nearer to reference point; we choose $1 < \theta \leq 1.92$ when taking the perpendicular bisector for the second time, and the MN moves to a point which is relatively nearer to reference point[3]. The rest may be deduced by analogy.

Step 3. Similar to Step 3 of the first situation. The MN will pass $i-3$, $i-2$, $i-1$, i, b, c, and d successively.

On the one hand, the perpendicular bisector strategy causes the movement trajectory of the MN to tend to be zigzag or concentrative, so the distribution of anchor coordinates may concentrate in a relatively small portion of the simulation region. On the other hand, as running the perpendicular bisector strategy, the time which the MN spends within the communication range of an unknown node is 2ξ s[4] at worst. The more the time which is spend within the communication range of an unknown node is, the more the energy consumption of other unknown nodes is. Hence, when the MN and that unknown node receive a packet from the opposite side, we add once channel listening for that unknown node and once packet sending for the MN at next time interval, but all other unknown nodes keep normal listening periodicity; if no packets received, the MN and that unknown node will keep the normal sending and listening periodicity. This improvement may reduce total listening time for all unknown nodes and save their energy. As shown in Fig. 2 (f), we assume that the MN and an unknown node receive an acknowledgement sent by opposite side at $i-2$, and then the MN will add once packet sending and unknown node will add once channel listening at the $(i-2)$-th time interval, such as j. If no acknowledgement received at j, the MN and unknown node will keep normal periodicity at the $(i-1)$-th time interval.

2.5 Virtual repulsive force strategy

The relationship between the virtual repulsive force and the velocity is as follows[24]:

$$\boldsymbol{F}_{\text{repulsive}} = m\boldsymbol{a} \tag{15}$$

$$\boldsymbol{V} = \boldsymbol{a}\Delta t \tag{16}$$

where $\boldsymbol{F}_{\text{repulsive}}$ denotes the virtual repulsive force; m is the mass of the MN; $\boldsymbol{a}$ is the acceleration of the MN; Δt is the sampling time interval; and $\boldsymbol{V}$ is the velocity of the MN. We can obtain another expression of force by substituting (16) for $\boldsymbol{a}$ in (15) as follows:

$$\boldsymbol{F}_{\text{repulsive}} = m\frac{\boldsymbol{V}}{\Delta t} \tag{17}$$

For convenience, let $k = m/\Delta t$, whose unit is kg/s. With this notation, (17) can be simplified to

$$\boldsymbol{F}_{\text{repulsive}} = k\boldsymbol{V} \tag{18}$$

As shown in (18), let $k = 1$ kg/s, the value of $\boldsymbol{V}$ is equal to the value of $\boldsymbol{F}_{\text{repulsive}}$. And then, we build the transforming relationship between the virtual repulsive force and the velocity.

The virtual repulsive force strategy will be divided into two stages:

1) Within the surveillance region

In this section, the virtual repulsive force strategy described by

$$\boldsymbol{F} = \begin{cases} \boldsymbol{0}, & d'_{MN-i} > R \\ \boldsymbol{F}_{\text{repulsive}}, & d'_{MN-i} \leq R \end{cases} \tag{19}$$

where $\boldsymbol{F}$ denotes the virtual force; d'_{MN-i} denotes the distance between the MN and the location-aware node i; and R is the communication radius of nodes.

In this strategy, once the MN enters into the communication range of a location-aware node, the MN will be applied a virtual repulsive force, which ensures the MN can go away from this location-aware node as soon as possible.

Fig. 3 (a) illustrates a virtual repulsive force scenario. O is a location-aware node; $i-3$, $i-2$, $i-1$, and i are the movement trajectories of the MN at continuous time. f denotes the virtual repulsive force, the direction of the arrow is the direction of the virtual repulsive force. When the MN moves to $i-1$, the virtual repulsive force impels MN to be away from O at the next time interval. Equation (18) has established a relationship between the virtual force and the velocity, the MN can move to i due to being applied a virtual repulsive force.

Fig. 3 (b) shows the MN moves based on the Gauss-Markov mobility model within the communication range of location-aware node O, it takes six steps in departing from the communication range of O, such as $i-6$, $i-5$, $i-4$, $i-3$, $i-2$, and $i-1$. This process consumes some unnecessary time and leads to less efficiency for VB-ERL algorithm. Compared with the Gauss-Markov mobility model, the virtual repulsive force strategy is better as it only spends 1 step staying within the communication range of a location-aware node.

Although the virtual repulsive force may force the MN to leave the communication range of location-aware nodes rapidly, a series of location-aware nodes could form a virtual wall which does not allow the MN to pass, and then the MN only moves within a limited region until simulation time ends. Fig. 3 (c) illustrates an example. A, B, C, D, E denote location-aware nodes, the circle denotes their communication range, the dashed line denotes the boundary of the surveillance region. The communication ranges of A, B, C, D, E divide the whole surveillance region into two parts, as due to the existence of the virtual repulsive force, the MN always moves within the upper left corner, such as from $MN1$ to $MN2$. $a1$, $a2$, $a3$, $b1$, $c1$, $d1$, $e1$ denote coordinates of the MN within the communication range of A, B, C, D, E. $a2$ denotes that it is the second time that the MN comes into the communication radius of A, the rest may be deduced by analogy. In order to avoid this disadvantage, we predefine a count threshold τ. When the count that the MN enters into the communication range of the same location-aware node exceeds τ, the virtual repulsive force of this location-aware node will disappear. Then, the MN can pass this vanished virtual wall. Fig. 3 (d) illustrates this process. We predefine τ as 3 and $a4$ denotes that it is the fourth time the MN enters into the communication range of A. When the virtual repulsive force vanishes, the MN will speed up to pass this virtual wall region. Finally, the MN arrives at $MN2$. Further discussion for selecting an optimal value of $|\boldsymbol{F}_{\text{repulsive}}|$ within the surveillance region is given in Subsection 3.3.

2) Outside of the surveillance region

The Gauss-Markov mobility model modifies the mean direction variable $\overline{d}$ in (7) to ensure that the MN does not remain near an edge of surveillance region for a long period of time[20], but due to the existence of random variable

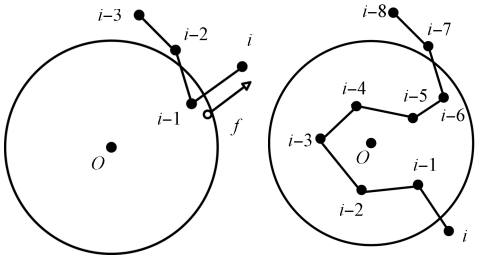
[3]Since the public communication range is smaller, we should ensure anchors distribute in this range as much as possible. If we continue to choose $0.57 \leq \theta \leq 1$ at the next time interval, anchors will concentrate excessively as shown in Fig. 2 (d). Then, the smaller received signal error will cause a larger position estimation error. In order to avoid anchor coordinates over-concentrated, we choose the value of θ alternately in $[0.57, 1]$ and $(1, 1.92]$. In Fig. 2 (c), $\angle(i-1)ib \in [0.57,1]$ and $\angle ibc \in (1, 1.92]$.

[4]We assume the normal broadcasting interval was 1 s.

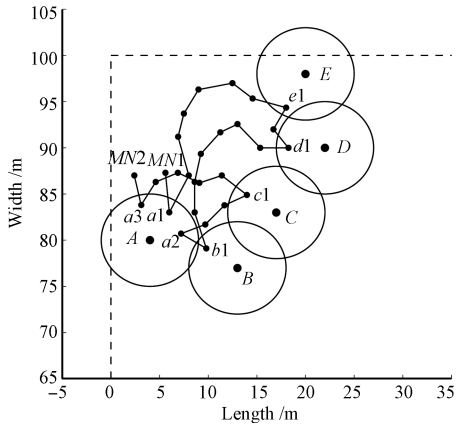
$d_{x_{n-1}}$, this process is not very effective. After being out of the boundary, the MN would still take a long time to return to the surveillance region. In this strategy, once leaving the surveillance region, the MN will be applied a virtual repulsive force to return to the surveillance region rapidly by the region outside of the boundaries. This virtual repulsive force will change the velocity and direction of the MN. Returning velocity is related to the velocity of the MN at the last time interval, i.e., if the MN goes out of the surveillance region at higher velocity, it will be exerted a larger virtual repulsive force; otherwise, it will be exerted a smaller virtual repulsive force. The returning velocity at the next time interval will be calculated by

$$s_{\text{returning}} = s_{n-1} \tag{20}$$

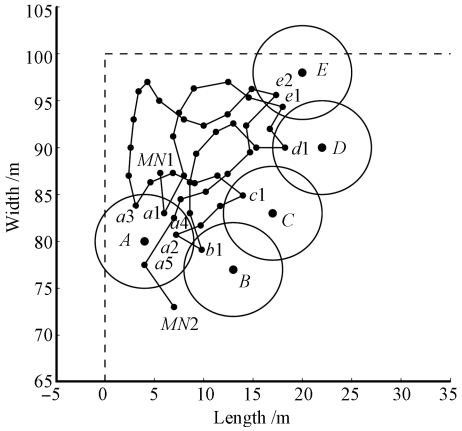
where $s_{\text{returning}}$ denotes the returning velocity; s_{n-1} denotes the velocity of the MN at the $(n-1)$-th time interval.



(a) Scenario 1 (b) Scenario 2



(c) Formation of a virtual wall



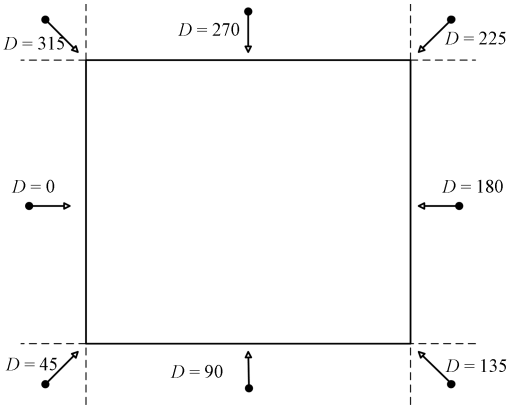
(d) Disappearance of virtual repulsive force

Fig. 3 Impact of the virtual repulsive force on the trajectory within the surveillance region

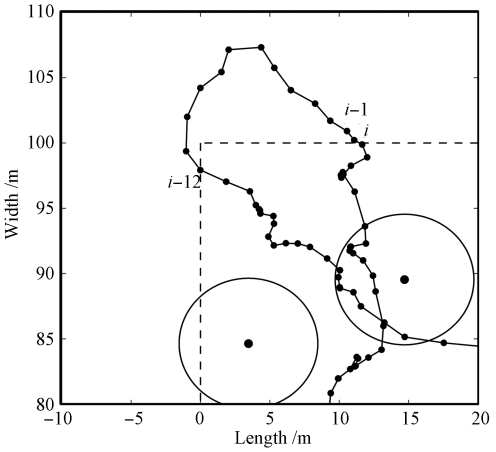
The new direction at the next time interval can be defined as follows:

$$d_n = D_{\text{virtual}} \tag{21}$$

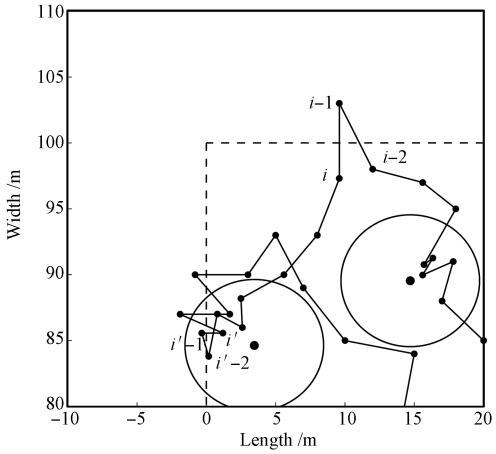
where D_{virtual} denotes the new direction which is changed by the virtual repulsive force of the region outside the boundary. The values of the direction for different locations outside of the surveillance region are shown in Fig. 4 (a).



(a) Direction



(b) Movement trajectory of the MN based on the Gauss-Markov mobility model outside of the surveillance region



(c) Movement trajectory of the MN based on virtual repulsive force strategy outside of the surveillance region

Fig. 4 Impact of the virtual repulsive force on the trajectory outside of the surveillance region

Fig. 4 (b) illustrates a movement trajectory of the MN based on the Gauss-Markov mobility model outside of the surveillance region. The dashed lines denote the boundaries of the surveillance region; the anchor threshold ξ is 5. Then, the MN spends 13 steps in moving outside of the surveillance region, such as from $i-12$ to i. Compared with Fig. 4 (b), the MN spends only 1 step in returning to the surveillance region using the proposed virtual repulsive force strategy in Fig. 4 (c). The MN moves to $i-1$ from $i-2$ at a higher velocity, where the virtual repulsive force produced by the region outside of upper boundary will change the direction of the MN to 270° at the next time interval, its velocity value is equal to one at the last time interval. Similarly, the MN is also exerted a virtual repulsive force at $i'-1$, since the MN moves to $i'-1$ from $i'-2$ at a lower velocity, the corresponding virtual repulsive force is also smaller, so the Euclidean distance between $i'-1$ and i' is shorter than that between $i-1$ and i.

2.6 Algorithm efficiency analysis

In this section, we evaluate the performance of the proposed algorithm through mathematical analysis. Due to the randomness of the Gauss-Markov mobility model, we just make an approximate time consumption evaluation for two algorithms. Similar to the method in [18], we consider a squared sensing field in which N unknown sensor nodes are randomly distributed. Assume that the moving distances which the MN covers in the whole surveillance region are equal for the proposed algorithm and VB-ERL, and we evenly divide this distance into n parts (L_n), each part corresponds to a time interval. Firstly, we compute the whole running time using the velocity adjustment strategy and VB-ERL, respectively, during the time when the MN moves in the whole surveillance region. We had already deduced s_n explicitly in terms of the initial velocity s_0 in (10). For the simplicity of analysis, we assume the values of the random variables are equal at corresponding time interval for two algorithms; the movement directions of the MN are also the same at corresponding time interval. We use H instead of the random variables polynomial $\sqrt{(1-\alpha^2)}\sum_{i=0}^{n-1}\alpha^{n-i-1}s_{x_i}$. According to analysis in Subsection 2.3, (10) can be simplified to

$$s_n = (1-\alpha^n)\overline{S}_{\max} + H \tag{22}$$

$$s_{n\_\text{VB-ERL}} = (1-\alpha^n)\overline{s} + H \tag{23}$$

where s_n and $s_{n\_\text{VB-ERL}}$ are the new velocities of the MN at time interval n.

$$T = \sum_{i=1}^{n} \frac{L_n}{(1-\alpha^n)\overline{S}_{\max} + H} \tag{24}$$

$$T_{\text{VB-ERL}} = \sum_{i=1}^{n} \frac{L_n}{(1-\alpha^n)\overline{s} + H} \tag{25}$$

where T denotes the total movement time of the MN when just running the velocity adjustment strategy, $T_{\text{VB}-\text{ERL}}$ denote the total movement time of the MN for VB-ERL during the time when the MN moves in the whole surveillance region. Comparing the running time of two algorithms, we can get

$$\frac{T}{T_{\text{VB-ERL}}} = \frac{\sum\limits_{i=1}^{n} \frac{L_n}{(1-\alpha^n)\overline{S}_{\max}+H}}{\sum\limits_{i=1}^{n} \frac{L_n}{(1-\alpha^n)\overline{s}+H}} = \frac{\sum\limits_{i=1}^{n} \frac{1}{(1-\alpha^n)\overline{S}_{\max}+H}}{\sum\limits_{i=1}^{n} \frac{1}{(1-\alpha^n)\overline{s}+H}} \tag{26}$$

Since $1-\alpha^n \approx 1$ ($n > 10$), while putting this result into (26), we have

$$\frac{T}{T_{\text{VB-ERL}}} \approx \frac{\sum\limits_{i=1}^{n} \frac{1}{\overline{S}_{\max}+H}}{\sum\limits_{i=1}^{n} \frac{1}{\overline{s}+H}} = \frac{\frac{1}{\overline{S}_{\max}+H} \times n}{\frac{1}{\overline{s}+H} \times n} = \frac{\overline{s}+H}{\overline{S}_{\max}+H} \tag{27}$$

Since $\overline{s} < \overline{S}_{\max}$, $\overline{s}+H < \overline{S}_{\max}+H$, and $0 < \frac{\overline{s}}{\overline{S}_{\max}} < 1$, i.e., $T < T_{\text{VB-ERL}}$. And then, if $H > 0$, $\frac{\overline{s}+H}{\overline{S}_{\max}+H} - \frac{\overline{s}}{\overline{S}_{\max}} = \frac{H(\overline{S}_{\max}-\overline{s})}{(\overline{S}_{\max}+H)\overline{S}_{\max}} > 0$. Consequently, $0 < \frac{\overline{s}}{\overline{S}_{\max}} < \frac{T}{T_{\text{VB-ERL}}} < 1$. If $H < 0$, $\frac{\overline{s}+H}{\overline{S}_{\max}+H} - \frac{\overline{s}}{\overline{S}_{\max}} = \frac{H(\overline{S}_{\max}-\overline{s})}{(\overline{S}_{\max}+H)\overline{S}_{\max}} < 0$, accordingly, $0 < \frac{T}{T_{\text{VB-ERL}}} < \frac{\overline{s}}{\overline{S}_{\max}} < 1$, i.e., when just running the velocity adjustment strategy, the proposed algorithm takes less time than VB-ERL and the value of $\frac{T}{T_{\text{VB-ERL}}}$ converges to the value of $\frac{\overline{s}}{\overline{S}_{\max}}$.

After the MN enters the communication range of unknown nodes, the proposed algorithm spends ξ s in the best case and 2ξ s in the worst case while running the perpendicular bisector strategy. Taking general case into account, we take the middle value (T_{p_mid}), i.e., $T_{p\_\text{mid}} = (\xi + 2\xi)/2 = 1.5\xi$. Similarly, the velocity adjustment strategy is the same as Gauss-Markov mobility model, which have randomness. So the velocity adjustment strategy spends ξ s in the best case, but it could spend longer time ($\varphi \gg 2\xi$) in the worst case. We take the middle value ($T_{VA\_\text{mid}\_1}$), i.e., $T_{VA\_\text{mid}\_1} = (\xi + \varphi)/2$. Obviously, $T_{p\_\text{mid}} < T_{VA\_\text{mid}\_1}$. Let $a_1 = T_{VA\_\text{mid}\_1} - T_{p\_\text{mid}}$ and the whole running time of the proposed algorithm becomes T':

$$T' = T - a_1 N \tag{28}$$

where N denotes the number of unknown sensor nodes. According to analysis above, $T' < T_{\text{VB-ERL}}$, the larger N is, the smaller T' is.

Similarly, Once being out of boundary, the MN only spends 1 s in returning to the surveillance region using the virtual repulsive force strategy in the best case or in the worst case, i.e., the middle value $T_{VR\_\text{mid}} = (1+1)/2 = 1$. The velocity adjustment strategy also spends 1 s in returning to the surveillance region in the best case, but it could spend longer time ($\delta \gg 1$) returning to the surveillance region in the worst case, i.e., the middle value $T_{VA\_\text{mid}\_2} = (1+\delta)/2$. Obviously, $T_{VR\_\text{mid}} < T_{VA\_\text{mid}\_2}$. Let $a_2 = T_{VA\_\text{mid}\_2} - T_{VR\_\text{mid}}$ and the whole running time of the proposed algorithm becomes T'':

$$T'' = T' - a_2 K = T - a_1 N - a_2 K \tag{29}$$

where K denotes the number of times which the MN leaves the surveillance region.

Obviously, $T'' < T_{\text{VB-ERL}}$. The larger K is, the smaller T'' is. Moreover, $\frac{T''}{T_{\text{VB-ERL}}} < \frac{T}{T_{\text{VB-ERL}}}$, i.e., the value of $\frac{T''}{T_{\text{VB-ERL}}}$ also converges to the value of $\frac{\overline{s}}{\overline{S}_{\max}}$.

2.7 Localization algorithm

According to three strategies offered above, the whole localization algorithm can be expressed in Algorithm 1.

Algorithm 1

```
if (the MN enters into the communication range of node)
  {
    if (node is the location-aware)
      {
        the MN moves using the virtual repulsive force
        strategy at the next time interval;
      }
    else
      {
        if (anchor threshold value ≥ξ)
          {
            unknown node runs acoustic energy
            measurement localization mechanism;
            the MN moves using the virtual repulsive force
            strategy at the next time interval;
          }
        else
          {
            the MN moves using the minimum mean value
            of the velocity S̄min or perpendicular bisector
            strategy at the next time interval;
          }
        end
      }
    end
  }
elseif (the MN does not enter into the communication
      range of node)
  {
    if (the MN goes out of the surveillance region)
      {
        the MN moves using the virtual repulsive force
        strategy at the next time interval;
      }
    elseif (unknown node which the MN passed the last
          second has not located)
      {
        the MN moves based on the perpendicular bisector
        strategy at the next time interval;
      }
    else
      {
        the MN moves using the maximum mean value of
        the velocity S̄max at the next time interval;
      }
  }
end
```

3 Simulation and analysis

This section will verify the effectiveness of the proposed algorithm by simulating several scenarios and study the impact of different parameters on the proposed algorithm.

3.1 Simulation scenario and settings

We set simulation scenario and some key parameters as follows:

The network consisted of $30 \sim 50$ unknown nodes deployed randomly and a mobile anchor node in a region of $100 \times 100\,\mathrm{m}^2$. The initial value of $\overline{s}$ was $1\,\mathrm{m/s}$, the value of $\overline{d}$ was initially 90^{o} but changed over time according to the different edge proximity, $s_{x_{n-1}}$ and $d_{x_{n-1}}$ were random variables from a Gauss distribution, $\alpha = 0.75$, the communication radius of all nodes R was $5\,\mathrm{m}$, $\overline{S}_{\max} = 3\,\mathrm{m/s}$, $\overline{S}_{\min} = 0.5\,\mathrm{m/s}$, $|\boldsymbol{F}_{\mathrm{repulsive}}| = 3$. The normal broadcasting interval was $1\,\mathrm{s}$. The MN began its movement in the center of the simulation region $(50, 50)$ and movement time is $1\,000\,\mathrm{s}$. And the background noise level was set at $\sigma = 1$ for all nodes in the field. The count threshold τ was 2 and the anchor threshold ξ was 5.

3.2 Evaluation metrics

To analyze the simulation results, we use the following four metrics to evaluate the performance of the proposed algorithm.

1) Average localization error is:

$$\overline{E} = \frac{\sum_{i=1}^{N_{\text{location-aware}}} e^i}{N_{\text{location-aware}}} \tag{30}$$

where $\overline{E}$ is the average localization error (unit is meter); $N_{\text{location-aware}}$ denotes the number of all location-aware nodes; and e^i represents the localization error of node i, whose unit is also meter.

2) Coverage area ratio is:

$$C = \frac{A_{\text{located}}}{A_{\text{total}}} \times 100\,\% \tag{32}$$

where C denotes the coverage area ratio; A_{located} denotes the areas which have been covered; and A_{total} denotes the area of the whole surveillance region.

3) Residue unknown nodes: We find out how parameters, including τ and ξ, impact on the residue unknown nodes. The less the residue unknown nodes are, the better the performance of this approach is.

4) Computational complexity: It is easy to judge that the computational complexity of the proposed algorithm is $\mathrm{O}(n)$, which is in line with the VB-ERL.

3.3 Results analysis

Our simulations consisted of five parts: 1) finding out the difference of the movement trajectories between the proposed algorithm and VB-ERL; 2) investigating how the key parameters, including the virtual repulsive forces, $\overline{S}_{\max}$ and $\overline{S}_{\min}$, impact on the efficiency of the proposed algorithm; 3) comparing the coverage ratio of two algorithms; 4) researching how three strategies and their combinations impact on the localization accuracy and the time; 5) through adjusting the value of parameters, such as ξ and τ, estimating the performance of the proposed algorithm and VB-ERL.

Case 1. In this section, a movement trajectory corresponding with the proposed algorithm or VB-ERL is given, respectively. Simulation scenario and parameter settings are in accordance with Subsection 3.1. A movement trajectory of the MN using the proposed algorithm is shown in Fig. 5. The crisscross denotes unknown node; the black dot denotes location-aware node; the pentagram denotes the original coordinate of the MN; and the asterisk is the ending coordinate of the MN. As shown in Fig. 5, when the moving time is $1\,000\,\mathrm{s}$, the trajectory of the MN can cover most of the surveillance region and the residue unknown nodes are two. However, Fig. 6 shows that the movement trajectory of the MN based on the Gauss-Markov mobility model only cover fewer fields and there are nineteen residual unknown nodes at $1\,000\,\mathrm{s}$. For the proposed algorithm, the MN can automatically speed up or speed down in accordance with environmental change based on the velocity adjustment strategy, and rapidly leaves location-aware nodes and returns to the surveillance region based on the virtual repulsive strategy. In addition, the perpendicular bisector strategy can ensure that unknown nodes gain enough noncollinear anchor coordinates in the shortest time. Hence, the proposed algorithm can save more time, the MN, in turn, has more time to move to other portions of the surveillance region. Accordingly, when all things are equal, the

MN can move longer distances using the proposed algorithm than one using VB-ERL, so the former can cover more surveillance region than the latter.

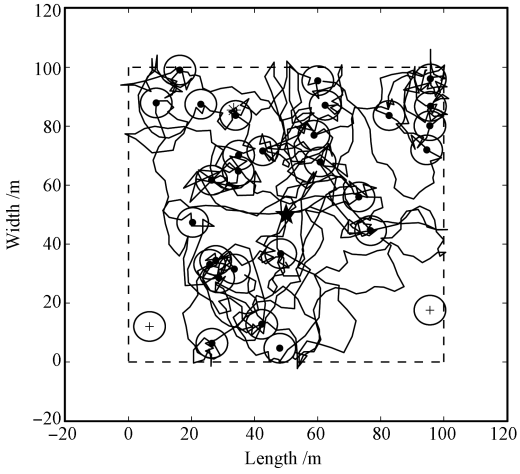


Fig. 5 Movement trajectory of the MN based on the proposed algorithm

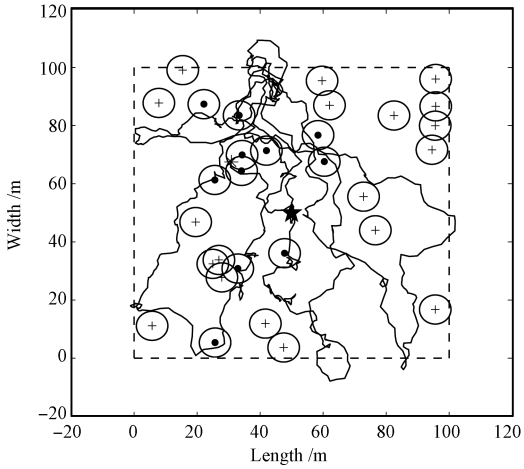


Fig. 6 Movement trajectory of the MN based on the Gauss-Markov mobility model

Case 2. We simulated with different virtual repulsive forces $\overline{S}_{\max}$ and $\overline{S}_{\min}$ to evaluate the performance of the proposed algorithm.

1) Virtual repulsive force

In this section, there are 30 unknown nodes in the surveillance region which are randomly deployed 40 times. We adjusted the value of the repulsive force from 0.1 to 5 in interval 0.1 and simulation time from 500 s to 1 000 s in time interval 100 s, and other parameters were invariant, i.e., $R = 5$ m; $\overline{S}_{\max} = 3$ m/s; $\overline{S}_{\min} = 0.5$ m/s. Afterward, we took the average number of residue unknown nodes at corresponding time and established a relationship between residue unknown node, time, and $|\boldsymbol{F}_{\text{repulsive}}|$ as shown in Fig. 7. With the increase of $|\boldsymbol{F}_{\text{repulsive}}|$, firstly, the number of residue unknown nodes declines smoothly, and then it increases. As the value of $|\boldsymbol{F}_{\text{repulsive}}|$ is between 2 to 3, the number of residue unknown nodes reaches minimum. If the value of $|\boldsymbol{F}_{\text{repulsive}}|$ is maintained fixed, the number of residue unknown nodes will decrease gradually with the advance of time. The minimum number of residue unknown nodes is close to 3 at 1 000 s.

Fig. 7 shows that a too large or too small virtual repulsive force cannot make the number of residue unknown node achieve minimum. Since the virtual repulsive force is too large, the MN would cross unknown nodes directly which are in the vicinity of location-aware nodes; while the virtual repulsive force is too small to ensure efficiency of this strategy.

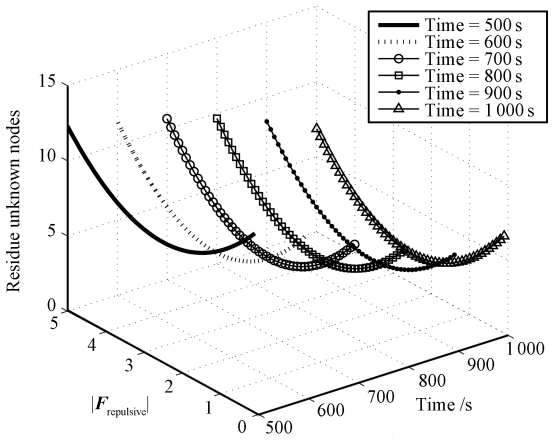


Fig. 7 Relationship between $|\boldsymbol{F}_{\text{repulsive}}|$ and residue unknown node

2) $\overline{S}_{\max}$

This section focuses on investigating the influence of the changes in the maximum mean velocity for localization. We deployed 30, 40, and 50 unknown nodes, respectively, and adjusted $\overline{S}_{\max}$ from 0.2 m/s to 6 m/s in interval 0.2. $|\boldsymbol{F}_{\text{repulsive}}| = 3$; $\overline{S}_{\min} = 0.5$ m/s; time = 1 000 s. Similarly, other parameters are maintained fixed.

As shown in Fig. 8, when $\overline{S}_{\max} = 0.2$ m/s, the number of residue unknown nodes is two-third of all unknown nodes, such as 18, 26, and 35, respectively. The number of residue unknown nodes can achieve minimum as $\overline{S}_{\max}$ ranges from 3 m/s to 4 m/s, i.e., 2, 3, and 2, respectively.

Fig. 8 shows that the maximum mean velocity should not be too large, otherwise, the MN would also cross unknown nodes directly, which leads to lower efficiency of this algorithm.

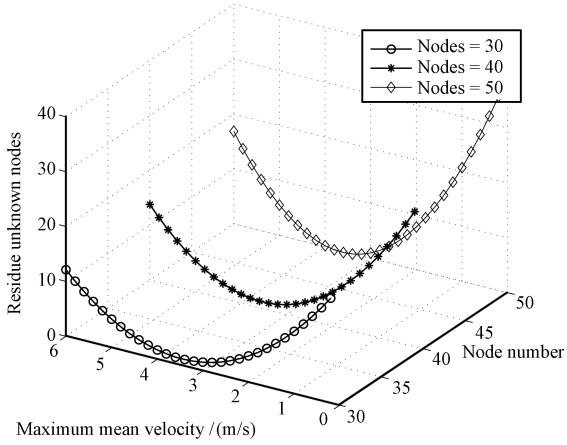


Fig. 8 Relationship between $\overline{S}_{\max}$ and residue unknown node

3) $\overline{S}_{\min}$

Similar to the simulation above, we deployed 30, 40, and 50 unknown nodes, respectively, and varied the value of

$\overline{S}_{\min}$ from 0 to 1 m/s in interval of 0.1. $|\boldsymbol{F}_{\text{repulsive}}| = 3$; $\overline{S}_{\max} = 3.5$ m/s; time = 1 000 s. Other parameters are maintained fixed.

In Fig. 9, with the increase of $\overline{S}_{\min}$, the number of residue unknown nodes gradually increases. When $\overline{S}_{\min} = 0$, the number of residue unknown nodes realizes minimum, i.e., 2, 3, and 5 respectively. When $\overline{S}_{\min} = 1$ m/s, the number of residue unknown nodes achieves maximum, i.e., 11, 14, and 15 respectively. The simulation results in Fig. 9 testify that the analysis mentioned in Subsection 2.3 is correct.

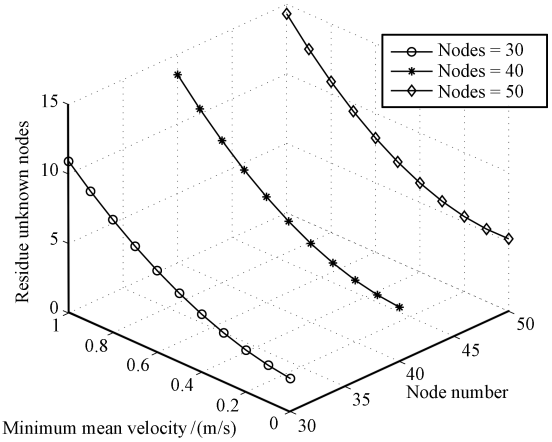


Fig. 9 Relationship between $\overline{S}_{\min}$ and residue unknown node

Case 3. We measured the coverage ratio of the proposed algorithm and VB-ERL at different times. $|\boldsymbol{F}_{\text{repulsive}}| = 3$; $\overline{S}_{\max} = 3.5$ m/s; $\overline{S}_{\min} = 0$. Other parameter settings were similar to the experiment above. As shown in Fig. 10, firstly, the coverage ratio of the proposed algorithm increases rapidly, which is close to 80 % at 650 s. But the coverage ratio of VB-ERL is only close to 36 % at 650 s. Afterward, the coverage ratio of the proposed algorithm increases slowly and reaches 100 % as time is about 1 300 s; however, the coverage ratio of VB-ERL is only 62 % at this time. When the coverage ratio is equal to 20 %, 30 %, 40 %, 50 %, 60 %, 70 %, 80 %, 90 %, the proposed algorithm approximately spends 100 s, 150 s, 200 s, 350 s, 450 s, 550 s, 650 s, and 950 s, respectively; the VB-ERL spends about 400 s, 550 s, 700 s, 950 s, 1 050 s, 1 550 s, 2 150 s, and 3 000 s, respectively. Accordingly, $T''/T_{\text{VB-ERL}}$ is equal to 0.25, 0.27, 0.29, 0.36, 0.4, 0.35, 0.3, and 0.3, respectively. $\overline{S}/\overline{S}_{\max} = 1/3.5 \approx 0.29$. The experimental results show that the mathematical analysis is correct in Subsection 2.7, i.e., the values of $\frac{T''}{T_{\text{VB-ERL}}}$ is in the neighboring range of 0.29. So, the proposed algorithm can rapidly cover the total surveillance region as compared to VB-ERL.

Case 4. This section focuses on investigating how three strategies and their combinations impact on the localization accuracy and the time. All parameters were the same as in Case 3.

As shown in Table 1, the VB-ERL spends 2 150 s achieving 80 % coverage ratio, while the algorithm using the velocity adjustment strategy spends only 875 s covering 80 % surveillance region. Since the velocity adjustment strategy can adaptively increase the velocity of the MN in non-target area, it can shorten the localization process and save the localization time; therefore, when moving distances are equal, it costs less time than VB-ERL. However, the randomness of the Gauss-Markov mobility model cannot ensure unknown nodes obtain non-collinear anchor coordinates, which brings lower localization accuracy. We can approximately compute the whole running time based on the velocity adjustment strategy and VB-ERL according to (24) $\sim$ (27). The perpendicular bisector strategy can also save the localization time (1 950 s), but the time-saving is not remarkable compared to the velocity adjustment strategy; its main role is to improve the localization accuracy. Similarly, the virtual repulsive force strategy ensures the MN spends only 1 s leaving the communication range of location-aware nodes or returning to the surveillance region, which can also shorten the localization process (1 800 s). Likewise, compared to velocity adjustment strategy, the time-saving of the virtual repulsive force strategy is not remarkable and its localization accuracy is lower as well. Since the times which the MN leaves the surveillance region are random, the more the time is, the more the sowing time is. Combining the velocity adjustment strategy and the perpendicular bisector strategy cannot only reduce the running time in correspondence to 80 % coverage ratio to 800 s, but also improve the localization accuracy. Although combining the velocity adjustment strategy and the virtual repulsive force strategy can make the running time in correspondence to 80 % coverage ratio reduce to 750 s, the localization accuracy is still lower. Combining the perpendicular bisector strategy and the virtual repulsive force strategy can save more time than running these two strategies separately; however, this combination approach still cannot achieve the effect of the velocity adjustment strategy. Finally, if three strategies, including the velocity adjustment strategy, the perpendicular bisector strategy, and the virtual repulsive force strategy, combined together, the time is reduced to 650 s and the localization accuracy is higher. Hence, these experimental results validate that (29) is correct.

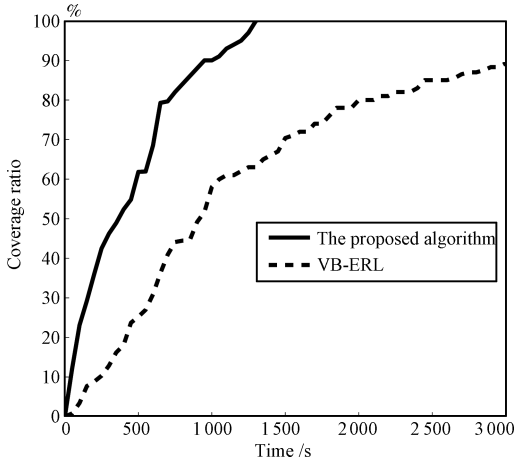


Fig. 10 Coverage ratios of the proposed algorithm and VB-ERL

Case 5. This section estimates the performance of the proposed algorithm and VB-ERL. We adjusted τ from 1 to 3, and defined anchor threshold ξ as 5, 10, 15. The simulation time was 1 000 s. Other parameters were the same as in the experiment above.

As listed in Table 2, $\tau = 1$, ξ is set to 5, 10, and 15, $\overline{E} = 0.16$ m, 0.12 m, 0.06 m for the proposed algorithm respectively, $\overline{E} = 0.28$ m, 0.18 m, 0.12 m for VB-ERL, respectively. Obviously, the accuracy of the proposed algorithm is higher than VB-ERL, this is because the proposed algorithm considers how unknown nodes obtain more non-collinear anchor coordinates. Similarly, when $\xi = 5$, $\tau = 1$, and the number of all unknown nodes is equal to 30, 40, and 50, respectively, the number of residue unknown nodes for

the proposed algorithm is 3, 4, and 4, respectively; the corresponding number of residue unknown nodes for VB-ERL is 14, 10, and 7. If we continue increasing the value of τ, the number of residue unknown nodes for the proposed algorithm will continue decreasing. Consequently, compared with VB-ERL, the proposed algorithm can provide more location-aware nodes and higher localization accuracy.

Table 1 Comparison of the running times and the localization accuracies for three strategies and their combinations

Model	Time for 80% coverage ratio	Localization accuracy
Gauss-Markov mobility model (VB-ERL)	2 150 s	Low
Velocity adjustment strategy	875 s	Low
Perpendicular bisector strategy	1 950 s	High
Virtual repulsive force strategy	1 800 s	Low
Velocity adjustment strategy & Perpendicular bisector strategy	800 s	High
Velocity adjustment strategy & Virtual repulsive force strategy	750 s	Low
Perpendicular bisector strategy & Virtual repulsive force strategy	1700 s	High
Velocity adjustment strategy & Perpendicular bisector strategy & Virtual repulsive force strategy	650 s	High

Table 2 Performances of the proposed algorithm and VB-ERL

Approach	Parameter	Statistical item	Nodes = 30	Nodes = 40	Nodes = 50
The proposed algorithm	$\tau = 1$	Residue unknown nodes	3	4	4
	$\xi = 5$	$\overline{E}$	0.16	0.15	0.13
	$\tau = 2$	Residue unknown nodes	3	4	3
	$\xi = 5$	$\overline{E}$	0.17	0.15	0.14
	$\tau = 3$	Residue unknown nodes	2	3	2
	$\xi = 5$	$\overline{E}$	0.19	0.17	0.16
	$\tau = 1$	Residue unknown nodes	5	7	6
	$\xi = 10$	$\overline{E}$	0.12	0.11	0.1
	$\tau = 2$	Residue unknown nodes	5	6	5
	$\xi = 10$	$\overline{E}$	0.13	0.12	0.11
	$\tau = 3$	Residue unknown nodes	4	4	4
	$\xi = 10$	$\overline{E}$	0.13	0.13	0.12
	$\tau = 1$	Residue unknown nodes	6	8	7
	$\xi = 15$	$\overline{E}$	0.06	0.09	0.09
	$\tau = 2$	Residue unknown nodes	5	6	6
	$\xi = 15$	$\overline{E}$	0.07	0.1	0.1
	$\tau = 3$	Residue unknown nodes	4	5	5
	$\xi = 15$	$\overline{E}$	0.09	0.1	0.11
VB-ERL	$\xi = 5$	Residue unknown nodes	14	10	7
		$\overline{E}$	0.28	0.25	0.29
	$\xi = 10$	Residue unknown nodes	17	13	11
		$\overline{E}$	0.18	0.20	0.19
	$\xi = 15$	Residue unknown nodes	21	16	14
		$\overline{E}$	0.12	0.15	0.14

4 Conclusion

Localization is one of the substantial issues in wireless sensor networks. This paper addresses the issue of the localization based on a mobile anchor node. Using a mobile anchor node instead of all stationary anchor nodes has several notable advantages, including cost reduction, enhancing the localization accuracy, and so on.

In this paper, we present an adaptive localization approach for wireless sensor networks based on Gauss-Markov mobility model which includes three strategies: 1) The perpendicular bisector strategy ensures that unknown nodes obtain enough non-collinear anchor coordinates as soon as possible; 2) The virtual repulsive force strategy impels that the MN rapidly leaves the communication range of location-aware nodes or returns to the surveillance region after the MN was out of the boundary; 3) The velocity adjustment strategy ensures that the MN automatically adjusts its velocity in accordance with environmental change within the surveillance region. We use four metrics to evaluate the performance of the proposed algorithm, i.e., average localization error, residue unknown node ratio, coverage area ratio, and computational complexity. Simulation results and analysis, comparing the proposed algorithm with VB-ERL, show the computational complexity of two algorithm are equivalent and the proposed algorithm is more efficient, including less running time, lower energy consumption, and higher localization accuracy.

References

1 Liao W H, Shih K P, Lee Y C. A localization protocol with adaptive power control in wireless sensor networks. *Computer Communications*, 2008, **31**(10): 2496−2504

2 Xu Y, Heidemann J, Estrin D. Geography-informed energy conservation for Ad Hoc routing. In: Proceedings of the 7th Annual International Conference on Mobile Computing and Networking. Rome, Italy: ACM, 2001. 70−84

3 Akkaya K, Younis M. A survey on routing protocols for wireless sensor networks. *Ad Hoc Networks*, 2005, **3**(3): 325−349

4 Savvides A, Han C C, Strivastava M B. Dynamic fine-grained localization in Ad-Hoc networks of sensors. In: Proceedings of the 7th Annual International Conference on Mobile Computing and Networking. Rome, Italy: ACM, 2001. 166−179

5 Li D, Hu Y H. Energy-based collaborative source localization using acoustic microsensor array. *EURASIP Journal on Applied Signal Processing*, 2003, **2003**(4): 321−337

6 Bahl P, Padmanabhan V N. RADAR: an in-building RF-based user location and tracking system. In: Proceedings of the 19th Annual Joint Conference of the IEEE Computer and Communications Societies. Tel Aviv, Israel: IEEE, 2000. 775−784

7 Girod L, Estrin D. Robust range estimation using acoustic and multimodal sensing. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems. Hawaii, USA: IEEE, 2001. 1312−1320

8 Priyantha N B, Chakraborty A, Balakrishnan H. Cricket location-support system. In: Proceedings of the 6th Annual International Conference on Mobile Computing and Networking. Boston, USA: ACM, 2000. 32−43

9 Baronti P, Pillai P, Chook V W C, Chessa S, Gotta A, Hu Y F. Wireless sensor networks: a survey on the state of the art and the 802.15.4 and Zigbee standards. *Computer Communications*, 2007, **30**(7): 1655−1695

10 Luo Hai-Yong, Li Jin-Tao, Zhao Fang, Lin Quan, Zhu Zhen-Min, Yuan Wu. Robust node localization based on distributed weighted-multidimensional scaling in wireless sensor networks. *Acta Automatica Sinica*, 2008, **34**(3): 288−297 (in Chinese)

11 Baggio A, Langendoen K. Monte Carlo localization for mobile wireless sensor networks. *Ad Hoc Networks*, 2008, **6**(5): 718−733

12 Mao G Q, Fidan B, Anderson B D O. Wireless sensor network localization techniques. *Computer Networks*, 2007, **51**(10): 2529−2553

13 Niculescu D, Nath B. DV based positioning in Ad Hoc networks. *Telecommunication Systems*, 2003, **22**(1-4): 267−280

14 He T, Huang C D, Blum B M, Stankovic J A, Abdelzaher T. Range-free localization schemes for large scale sensor networks. In: Proceedings of the 9th Annual International Conference on Mobile Computing and Networking. San Diego, USA: ACM, 2003. 81−95

15 Du W L, Fang L, Peng N. LAD: localization anomaly detection for wireless sensor networks. *Journal of Parallel and Distributed Computing*, 2006, **66**(7): 874−886

16 Zhang R B, Zhang L L, Feng Y B. Very low energy consumption wireless sensor localization for danger environments with single mobile anchor node. *Wireless Personal Communications*, 2008, **47**(4): 497−521

17 Koutsonikolas D, Das S M, Hu Y C. Path planning of mobile landmarks for localization in wireless sensor networks. *Computer Communications*, 2007, **30**(13): 2577−2592

18 Zhang Zheng-Yong, Sun Zhi, Wang Gang, Yu Rong, Mei Sun-Liang. Localization in wireless sensor networks with mobile anchor nodes. *Journal of Tsinghua University (Science and Technology)*, 2007, **47**(4): 534−537 (in Chinese)

19 Kuang Xing-Hong, Shao Hui-He, Feng Rui. A new distributed localization scheme for wireless sensor networks. *Acta Automatica Sinica*, 2008, **34**(3): 344−348

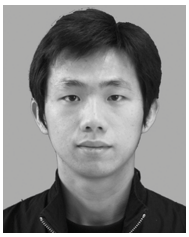
20 Liang B, Haas Z J. Predictive distance-based mobility management for PCS networks. In: Proceedings of the 18th Annual Joint Conference of the IEEE Computer and Communications Societies. New York, USA: IEEE, 1999. 1377−1384

21 Liang B, Haas Z J. Predictive distance-based mobility management for multidimensional PCS networks. *IEEE/ACM Transactions on Networking*, 2003, **11**(5): 718−732

22 Camp T, Boleng J, Davies V. A survey of mobility models for Ad Hoc network research. *Wireless Communications and Mobile Computing*, 2002, **2**(5): 483−502

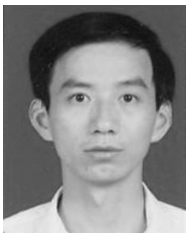
23 Poqqi C, Mazzini G. Collinearity for sensor network localization. In: Proceedings of the 58th IEEE Vehicular Technology Conference. Orlando, USA: IEEE, 2003. 3040−3044

24 Naik P V. *Principles of Physics*. New Delhi: Prentice-Hall, 2004. 25−27



ZHONG Zhi Ph. D. candidate at the School of Information Science and Engineering, Central South University. His research interest covers localization and routing of wireless sensor networks.
E-mail: zhongzhi717@163.com

LUO Da-Yong Professor at the School of Information Science and Engineering, Central South University. His research interest covers wireless sensor networks, information fusion, and image processing.
E-mail: dyluo@mail.csu.edu.cn



LIU Shao-Qiang Ph. D., associate professor at the School of Information Science and Engineering, Central South University. His research interest covers wireless sensor networks, sensor technology, measurement technology, and automatic equipment.
E-mail: liussqq@mail.csu.edu.cn

FAN Xiao-Ping Ph. D., professor at the School of Information Science and Engineering, Central South University. His research interest covers wireless sensor networks, robotics, and intelligent transportation system. Corresponding author of this paper. E-mail: xpfan@mail.csu.edu.cn

QU Zhi-Hua Ph. D., professor at the School of Information Science and Engineering, Central South University, P. R. China, professor in the Department of Electrical Engineering and Computer Science, University of Central Florida, USA. His research interest covers systems theory, control theory, and robotics.
E-mail: qu@mail.ucf.edu