Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2015, Article ID 805061, 16 pages
http://dx.doi.org/10.1155/2015/805061

Research Article

Decomposition Based Localization for Anisotropic Sensor Networks

Baojian Gao,[1] Xiaoning Zhao,[2] Jun Wang,[1] and Xiaojiang Chen[1]

[1]*School of Information Science and Technology, Northwest University, Xi'an 710127, China*
[2]*AVIC Xi'an Aircraft Industry (Group) Company, Xi'an 710089, China*

Correspondence should be addressed to Baojian Gao; esu7031@sina.com

Received 19 June 2015; Accepted 17 September 2015

Academic Editor: Jianping He

Range-free localization algorithms have caused widespread attention due to their low cost and low power consumption. However, such schemes heavily depend on the assumption that the hop count distance between two nodes correlates well with their Euclidean distance, which will be satisfied only in isotropic networks. When the network is anisotropic, holes or obstacles will lead to the estimated distance between nodes deviating from their Euclidean distance, causing a serious decline in localization accuracy. This paper develops HCD-DV-Hop for node localization in anisotropic sensor networks. HCD-DV-Hop consists of two steps. Firstly, an anisotropic network is decomposed into several different isotropic subnetworks, by using the proposed Hop Count Based Decomposition (HCD) scheme. Secondly, DV-Hop algorithm is carried out in each subnetwork for node localization. HCD first uses concave/convex node recognition algorithm and cleansing criterion to obtain the optimal concave and convex nodes based on boundary recognition, followed by segmentation of the network's boundary. Finally, the neighboring boundary nodes of the optimal concave nodes flood the network with decomposition messages; thus, an anisotropic network is decomposed. Extensive simulations demonstrated that, compared with range-free DV-Hop algorithm, HCD-DV-Hop can effectively reduce localization error in anisotropic networks without increasing the complexity of the algorithm.

1. Introduction

Node localization is one of the most fundamental technologies in wireless sensor networks (WSNs). For most applications of WSNs [1–3], only when nodes know their locations can they tell the system "what is happening in what position." For instance, in a fire monitoring system, a sensor node detects that a fire breaks out. The system is able to notify people where the fire is only when it knows the node's location. This allows fireman to reach the correct location and take effective measures immediately in order to prevent the spread of the fire. In addition, accurate localization is the foundation of many other WSNs technologies, such as position-aware data processing [4–6] and geographic routing [7, 8].

Among existing localization algorithms, range-free schemes [9–12] have been widely used due to their low cost and low power consumption. However, these schemes assume that the hop count distance between two nodes correlates well with their Euclidean distance, which will be satisfied only in isotropic networks [9, 13] (the definition of isotropic and anisotropic networks can be found in [14]). In practical applications, such as large-scale heritage protection [15] and military reconnaissance [16], sensor nodes are often randomly deployed in complex environments, which makes their network topologies highly irregular, with the possibility of holes or obstacles. Such network is called anisotropic network.

For example, Figure 1 shows an application of WSNs. In this scenario, sensors are distributed in the region to be monitored, and Figure 2 shows the corresponding node deployment. It is obvious in Figure 2 that the hop count distances between some pairs of nodes heavily deviate from their Euclidean distances (Take A and B in Figure 2, e.g., the black solid line represents the hop count distance between A and B, and the orange dotted line represents the Euclidean distance between them). Using the deviated shortest path distance

Figure 1: Application example of WSNs.

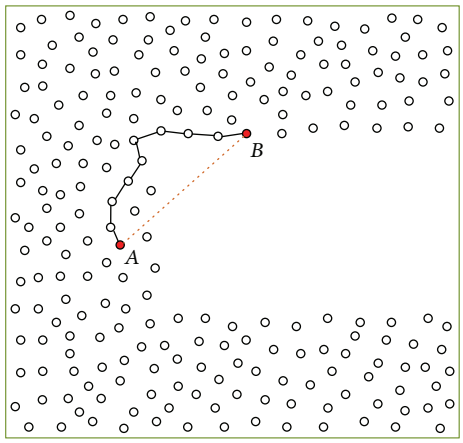


Figure 2: Node deployment.

instead of Euclidean distance to take part in localization will cause a serious decline in localization accuracy.

(1) Related Work. In order to improve the node localization accuracy in anisotropic networks, scholars have put forward many schemes in recent years, which can be divided into three categories.

The first scheme is mainly dedicated to amending the estimated distance between nodes. Li and Liu [17] proposed REP (Rendered Path) for localization in anisotropic sensor networks. REP assumes that the boundary nodes around obstacles have been obtained by a boundary recognition algorithm. The shortest path between nodes is divided into many different subsections according to the location of the obstacles. The angle between two subsections is calculated by constructing a unit circle in the intersection point of the shortest path and network holes; then, the Euclidean distance can be obtained by using the cosine law. REP is able to amend the estimated distance that is affected by obstacles. The main drawback is that REP suffers heavy computation and communication overhead, requiring high density and computation power of nodes. PDM (Proximity-Distance Map) [18] first uses a matrix to record the estimated distance and the Euclidean distance between anchor nodes. Then, it calculates the linear transformation matrix between the estimated distance and the Euclidean distance using the least square method. By using the linear transformation matrix, the estimated distance can be transformed into an amended distance, which is used to participate in localization. The disadvantage is that PDM requires the uniform deployment of anchor nodes, which is difficult in real applications.

The second solution aims at reducing the localization error by avoiding serious bent path participating in localization. Reference [19] proposed an improvement scheme for DV-Hop. It uses the cubic polynomial interpolation to obtain paths that are not heavily bent. By using these "good" paths in the calculations, the performance of DV-Hop in an anisotropic network can be improved. Reference [20] estimates the shortest paths affected by obstacles or network boundaries between unknown nodes and anchor nodes. By grading the estimating result, [20] chooses the anchor nodes corresponding to those less affected paths to participate in calculation, thus reducing the localization error. Reference [21] chooses some suitable anchor nodes in the networks and calculates the locations of unknown nodes through iterative multilateration. The localization error is controlled by avoiding the use of serious bent paths. Since iterative multilateration is employed, there is cumulative effect in localization error. Reference [22] proposed an improved multihop localization algorithm which is called i-multihop. It can automatically identify and eliminate severely miscalculated distances. In the localization process, these "bad" distances will not be used in calculation. Thus, the localization error can be controlled.

The third method achieves localization of unknown nodes by means of building landmark network. Reference [23] first locates a landmark network, which is composed of nodes that are uniformly sampled from the original network. The density of the landmark network is preset by the system. Each nonlandmark node uses trilateration to calculate its own location according to the distances to its closest three landmarks. References [24, 25] first select some landmark nodes using particular rules. Then, they calculate the Voronoi diagram and its dual Delaunay diagram according to the landmark nodes. The location of an unknown node is computed in each subdiagram. Finally, the original network can be restored through all the Delaunay diagrams. The difference between [24] and [25] is the method of selecting landmarks. Since both of the algorithms require Delaunay division for anisotropic networks, they are much more complicated to implement, which limits the application of the algorithms.

As can be seen from the above analysis, the similarity among the methods in [17–22] is that they all use various mechanisms to avoid or amend serious bent paths, which does not fundamentally solve the impact of anisotropic networks characteristics on localization. Although the schemes in [24, 25] proposed new ideas, their implementation processes are complex, which is not suitable for large-scale sensor networks.

(2) The HCD-DV-Hop Scheme. In this paper, a new method for localization is proposed based on decomposition. Our

scheme is composed of two steps. First, it decomposes an anisotropic network into several isotropic subnetworks by using the proposed Hop Count Based Decomposition (HCD) algorithm. Second, it uses the typical range-free DV-Hop algorithm to achieve node localization in each subnetwork. For simplicity, we call the proposed method HCD-DV-Hop algorithm. The contributions of this paper are summarized as follows:

(i) We analyze the reason why range-free DV-Hop algorithm is suitable for localization in isotropic networks but causes serious localization error in anisotropic networks. Then, we give an improved solution.

(ii) We introduce a distributed localization algorithm for anisotropic networks, which is called HCD-DV-Hop algorithm. HCD-DV-Hop first decomposes an anisotropic network into several different isotropic networks, so that the influence of holes or obstacles on the shortest communication path between far-away nodes is avoided. Since there is no longer serious bent path in each isotropic subnetwork, DV-Hop can be carried out for node localization in such subnetwork.

(iii) A Hop Count Based Decomposition (HCD) algorithm for anisotropic networks is proposed. HCD includes four steps: boundary recognition, concave/convex node recognition and cleansing, boundary segmentation, and region decomposition. All of these steps are distributed and only require connectivity information.

(iv) Simulation results demonstrate the superior performance of HCD-DV-Hop under different ratios of anchor node and communication radius in anisotropic networks.

The rest of this paper is organized as follows. In Section 2, we present the motivation of our scheme. Section 3 is an overview of HCD-DV-Hop. Section 4 is devoted to a description of the proposed HCD algorithm for anisotropic networks. The performance of HCD-DV-Hop is evaluated in Section 5. Finally, Section 6 concludes this paper.

2. Motivation

2.1. DV-Hop Algorithm. DV-Hop [9, 26] is one of the most widely used range-free schemes and it consists of three steps.

Step 1 (obtain the minimum hop count). Each anchor node broadcasts its location information and hop count value with a packet (x_i, y_i, h_i), where (x_i, y_i) is the coordinate of anchor node i and h_i is the minimum hop count from the anchor node i. By this means, all nodes (including anchor and unknown nodes) can obtain locations of anchor nodes and minimum hop count to each anchor.

Step 2 (calculate the estimated distance). Once an anchor node gets distances to other anchors, it estimates an average size for one hop, that is, C_i, which is then deployed as a correction to the entire network:

$$C_i = \frac{\sum_{j\neq i} \sqrt{\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2}}{\sum_{j\neq i} h_i}, \tag{1}$$

where (x_i, y_i) and (x_j, y_j) are the coordinates of anchor node i and anchor node j and h_i is the minimum hop count between anchor nodes i and j. Each unknown node receives the correction from its nearest anchor node and multiplies it by the minimum hop count, so that the estimated distance to each anchor node is obtained.

Step 3 (each unknown node estimates its location by using the maximum likelihood estimation). Let the coordinate of unknown node M be (x, y), and the distance between anchor node i and unknown node M is d_i. Assuming there are n anchor nodes in the network, (2) can be obtained:

$$\begin{aligned} \left(x - x_1\right)^2 + \left(y - y_1\right)^2 &= d_1^2, \\ \left(x - x_2\right)^2 + \left(y - y_2\right)^2 &= d_2^2, \\ &\vdots \\ \left(x - x_n\right)^2 + \left(y - y_n\right)^2 &= d_n^2. \end{aligned} \tag{2}$$

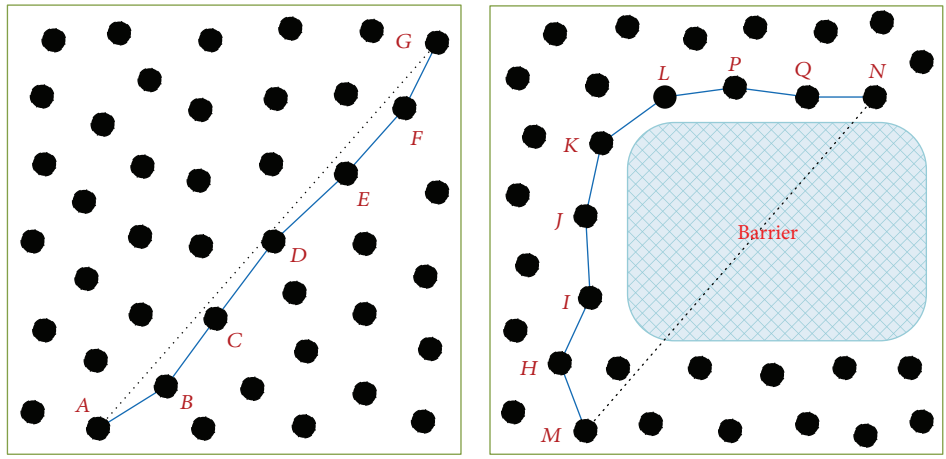
By using the maximum likelihood estimation, we get the location of unknown node M by

$$X = \left(A'A\right)^{-1} A'B, \tag{3}$$

where

$$\begin{aligned} A &= \begin{bmatrix} 2\left(x_1 - x_n\right) & 2\left(y_1 - y_n\right) \\ 2\left(x_2 - x_n\right) & 2\left(y_2 - y_n\right) \\ \vdots & \vdots \\ 2\left(x_{n-1} - x_n\right) & 2\left(y_{n-1} - y_n\right) \end{bmatrix}, \\ B &= \begin{bmatrix} x_1^2 + y_1^2 - x_n^2 - y_n^2 + d_n^2 - d_1^2 \\ x_2^2 + y_2^2 - x_n^2 - y_n^2 + d_n^2 - d_2^2 \\ \vdots \\ x_{n-1}^2 + y_{n-1}^2 - x_n^2 - y_n^2 + d_n^2 - d_{n-1}^2 \end{bmatrix}, \\ X &= \begin{bmatrix} x \\ y \end{bmatrix}. \end{aligned} \tag{4}$$

2.2. The Limitation of DV-Hop in Anisotropic Networks. As can be seen from DV-Hop, the estimated location accuracy of an unknown node depends on the accuracy of its estimated distances to the anchor nodes. That is, if the estimated distance between two nodes is closer to their Euclidean distance, the localization result will be more accurate. If



(a) Isotropic network (b) Anisotropic network

Figure 3: The shortest communication path between nodes in different networks.

the estimated distance heavily deviates from the Euclidean distance, the localization accuracy will be seriously declined.

In Figure 3, we use $\|AG\|$ and $\|MN\|$ to, respectively, represent the Euclidean distance between nodes A and G and M and N (the dotted line in Figures 3(a) and 3(b)). $|AG|$ and $|MN|$ are the estimated distances between A and G and M and N, respectively. In DV-Hop, the estimated distance between nodes i and j is calculated by $h_{ij} \times HS$, where h_{ij} is the minimum hop count between i and j (the hop count represented by the solid line in Figure 3). HS is the average size for one hop. The estimated distance between A and G in Figure 3(a) is

$$\begin{aligned} |AG| &= h_{AG} \times HS \\ &\approx \|AB\| + \|BC\| + \|CD\| + \|DE\| + \|EF\| + \|FG\|. \end{aligned} \tag{5}$$

The estimated distance between M and N in Figure 3(b) is

$$\begin{aligned} |MN| &= h_{MN} \times HS \\ &\approx \|MH\| + \|HI\| + \|IJ\| + \|JK\| + \cdots + \|KL\| \\ &\quad + \|LP\| + \|PQ\| + \|QN\|, \end{aligned} \tag{6}$$

where $\|\cdot\|$ represents the Euclidean distance between two nodes. From Figure 3(a), we have

$$\begin{aligned} &\|AB\| + \|BC\| + \|CD\| + \|DE\| + \|EF\| + \|FG\| \\ &\quad \approx \|AG\|. \end{aligned} \tag{7}$$

Then,

$$|AG| \approx \|AG\|. \tag{8}$$

That is to say, the estimated distance between A and G is approximately equal to their Euclidean distance. Therefore, in such isotropic networks, when the estimated distance takes the place of the Euclidean distance in localization, it will not cause serious error. That is why DV-Hop is suitable for localization in isotropic networks.

However, in an anisotropic network like Figure 3(b), it is obvious that

$$\begin{aligned} &\|MH\| + \|HI\| + \|IJ\| + \|JK\| + \|KL\| + \|LP\| + \cdots \\ &\quad + \|PQ\| + \|QN\| \gg \|MN\|. \end{aligned} \tag{9}$$

Taking formula (6) into consideration, then we have

$$|MN| \gg \|MN\|. \tag{10}$$

Formula (10) means that the estimated distance between M and N is much larger than their Euclidean distance. When we use the estimated distance to replace the Euclidean distance to participate in localization, it will cause serious localization error.

From the above analysis, we observe that the shortest communication path between nodes is crucial to localization result. Only when the shortest estimated distance between nodes is approximately equal to their Euclidean distance can the estimated location of node obtained by localization algorithm be approximate to its real location. In an anisotropic network, because of the barrier and irregularity of the network topology, the shortest estimated distance between far-away nodes will heavily deviate from the Euclidean distance, which will cause serious localization error.

A straightforward way is decomposing an anisotropic network into many different isotropic subnetworks and then carrying out DV-Hop for node localization in each subnetwork. Since there is no longer serious bent path between nodes in each subnetwork, the estimated distance can be approximately equal to the Euclidean distance. Therefore, in theory, this scheme can reduce localization error to a certain extent in anisotropic networks.

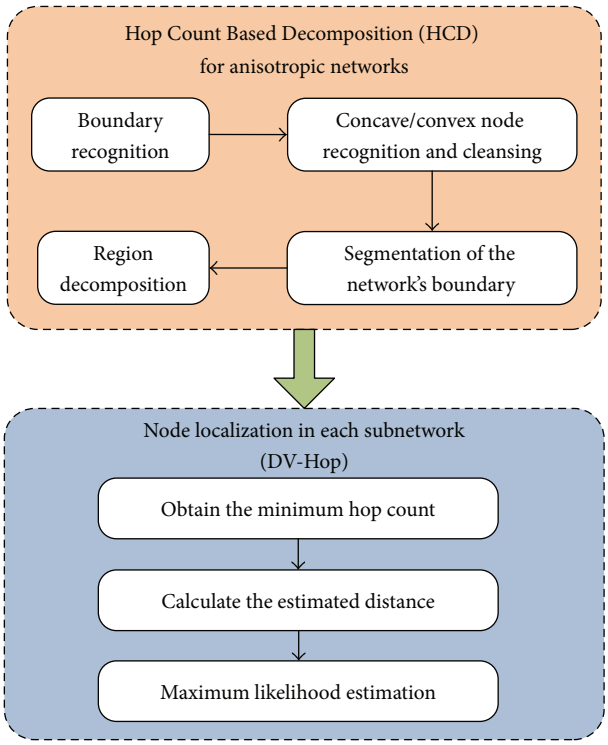


Figure 4: HCD-DV-Hop overview.

3. HCD-DV-Hop Overview

In this paper, we assume that the network is fully connected. Our proposed decomposition based localization scheme, that is, HCD-DV-Hop, mainly consists of the following two steps, as is shown in Figure 4.

(i) The anisotropic network is decomposed into several different isotropic subnetworks by using the Hop Count Based Decomposition (HCD) algorithm. HCD goes through four steps: boundary recognition, concave/convex node recognition and cleansing, boundary segmentation, and region decomposition, which will be shown in Section 4.

(ii) For each isotropic subnetwork, the typical range-free DV-Hop algorithm will be carried out for node localization.

4. Hop Count Based Decomposition (HCD) Algorithm for Anisotropic Networks

In large-scale wireless sensor networks, there may be thousands of sensor nodes, which makes it almost impossible to decompose the network by artificial means. So this paper will design an algorithm to achieve decomposition by using connectivity information of the network. The proposed Hop Count Based Decomposition (HCD) algorithm consists of four steps.

Step 1 (boundary recognition). This step aims at recognizing the boundary nodes in the network.

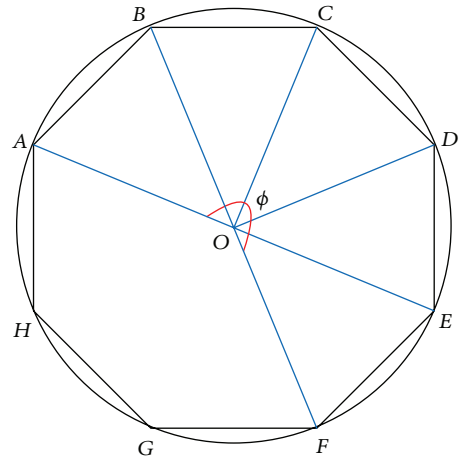


Figure 5: Schematic diagram of the cyclotomy.

Step 2 (concave/convex node recognition and cleansing). Based on boundary recognition, the concave/convex node recognition algorithm and cleansing criterion is proposed to obtain the optimal concave and convex nodes.

Step 3 (boundary segmentation). The optimal concave and convex nodes are used to flood the boundary nodes to segment the boundary into different branches.

Step 4 (region decomposition). The neighboring boundary nodes of optimal concave nodes flood the network to decompose an anisotropic network into different subnetworks.

4.1. Boundary Recognition. The existing boundary recognition algorithms can be divided into two categories: statistical methods [27] and geometrical methods [28]. After years of researches and developments, scholars have proposed many boundary recognition algorithms with good performance, such as [29–31]. Since this paper mainly focuses on region decomposition and node localization, we directly use [31] in our simulations for boundary recognition.

4.2. Concave/Convex Node Recognition and Cleansing

4.2.1. Definition of Concave/Convex Node Based on Cyclotomy. The cyclotomy [32] refers to increasing the number of sides of the regular polygon inside a circle to estimate the circumference by summing them (see Figure 5), based on which the concavity and convexity function of O point is given as

$$\varphi = \frac{\phi}{180^\circ} \approx \frac{n \times d}{\pi \times R}, \tag{11}$$

where d represents side length of the polygon, R is the radius of the circle, and n is the number of sides corresponding to ϕ. It means in (11) that the summation of n sides approximately equals the arc length of ϕ. When the vertexes of the polygon in Figure 5 are regarded as sensor nodes in the network, d can be considered as the maximum communication distance and n is the hop count between nodes.

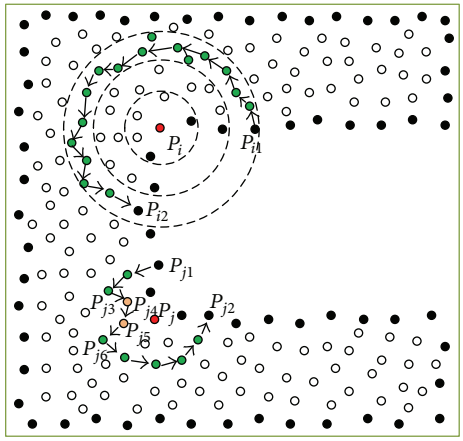


FIGURE 6: Boundary nodes are filled with black. (a) For P_i (filled with red), the nodes in $\partial N_3(P_i)$ are filled with green. p_{i1} and p_{i2} are both boundary nodes and 3-hop nodes of P_i. Intuitively, $\delta = 16$. The minimum hop count from p_{i1} to p_{i2} is $n = 16-1 = 15$, as is indicated by the arrows. (b) For P_j (filled with red), the nodes in $\partial N_2(P_j)$ are filled with green. p_{j1} and p_{j2} are both boundary nodes and 2-hop nodes of P_j. The auxiliary nodes (p_{j4} and p_{j5}) to make $\partial N_2(P_j)$ a connected component are filled with orange, $\lambda = 1$. The minimum hop count from p_{j1} to p_{j2} is $n = (11-1)-1 = 9$, as is indicated by the arrows.

In practical application, since the sensor network is discrete, there may be deviations in the calculation results due to the boundary noise. Considering this factor, the definitions of the concave nodes and the convex nodes based on the boundaries of $\phi = 157.5^\circ$ and $\phi = 202.5^\circ$ are given by

$$\begin{aligned} &0 < \varphi \le \frac{7}{8}, \quad O \text{ is a convex node,} \\ &\frac{7}{8} < \varphi < \frac{9}{8}, \quad O \text{ is a nonconcave/convex node,} \\ &\frac{9}{8} \le \varphi < 2, \quad O \text{ is a concave node.} \end{aligned} \tag{12}$$

Therefore, the concavity/convexity of point O can be determined by calculating the value of φ.

4.2.2. Concave/Convex Node Recognition Algorithm Based on Minimum Hop Count. Given an anisotropic network (see C-shape network in Figure 6 for an example), the boundary nodes can be recognized through [31]. Define ω as the set of boundary nodes (the black nodes in Figure 6). For an arbitrary node $P_i \in \omega$, we define $\partial N_k(P_i)$ as k-hop neighborhood of P_i, which means the set of nodes exactly k hops away from P_i. Intuitively, $\partial N_k(P_i)$ can be considered as a (or a part of) circle centered at P_i (see P_i in Figure 6). Given two nodes $p_{i1}, p_{i2} \in \partial N_k(P_i)$, define δ as the set of nodes that are on the shortest path from p_{i1} to p_{i2} (including p_{i1} and p_{i2}) through the nodes in $\partial N_k(P_i)$. Then, we define the minimum hop count between p_{i1} and p_{i2}, denoted by n, as δ minus one.

For each boundary node P_i, as shown in Figure 6, it first floods the network for k hops to obtain its k-hop neighborhood $\partial N_k(P_i)$. Two nodes p_{i1} and p_{i2} belonging to both $\partial N_k(P_i)$ and boundary nodes identify themselves. Note that p_{i1} and p_{i2} are on different sides of P_i. It is possible that there may be several boundary nodes belonging to $\partial N_k(P_i)$ at each side of P_i, and these boundary nodes naturally form a connected component. For this case, we randomly choose one boundary node at each side of P_i. p_{i1} and p_{i2} then flood $\partial N_k(P_i)$ to obtain the minimum hop count between them.

Because of the discreteness of the network, $\partial N_k(P_j)$ might be disconnected, as P_j in Figure 6 for an example. For this case, we use the following method to solve the problem. p_{j1}, p_{j2}, p_{j3}, p_{j6}, and the nodes filled with green belong to $\partial N_2(P_j)$, and p_{j4} and p_{j5} belong to $\partial N_1(P_j)$. Since node p_{j3} cannot send the message received from p_{j1} to p_{j6} directly, p_{j3} then sends this message to its neighbor p_{j4}, called auxiliary node, which is $k-1$ hops from P_j. As p_{j4} has no neighbor belonging to $\partial N_2(P_j)$, it sends the message to p_{j5}. p_{j5} then sends the message to p_{j6}, which belongs to $\partial N_2(P_j)$. With the help of auxiliary nodes, $\partial N_2(P_j)$ is considered connected. Thus, the minimum hop count from p_{j1} to p_{j2} is obtained. Obviously, the message from p_{j2} to p_{j1} can travel in the same way. The minimum hop count is estimated by $(n-\lambda)$, where λ is the difference between k and the hop count ($k-1$ in Figure 6) of the closest auxiliary node to the boundary node P_j. Another case that needs to be considered is when $k-1$ hop auxiliary nodes cannot make $\partial N_k(P_j)$ connected, it will turn to $k+1$ hop auxiliary nodes for help. The process is the same as the case of $k-1$. The worst case, which is almost impossible as the network is fully connected, is that neither $k-1$ hop nor $k+1$ hop auxiliary nodes can make $\partial N_k(P_j)$ connected. For this case, we discard the concavity/convexity recognition of P_j.

Next, we give the detail derivation process of concave/convex node recognition criterion. Take $k = 2$ for example. For an arbitrary boundary node P_i, with the above definitions of $\partial N_k(P_i)$, k-hop neighboring boundary nodes p_{i1} and p_{i2} for P_i (i.e., $p_{i1}, p_{i2} \in \omega$ and $p_{i1}, p_{i2} \in \partial N_2(P_i)$), and the minimum hop count (denoted by n) from p_{i1} to p_{i2}, we assume each hop count distance is S_i, $i = 1, 2, \ldots, n$. According to (11), the concavity and convexity function of P_i can be described by

$$\varphi = \frac{S_1 + S_2 + \cdots + S_n}{\pi \times R'}. \tag{13}$$

Taking account of the deployment efficiency of sensor nodes, it is reasonable to assume that S_i is uniformly distributed in $[R/2, R]$, where R is the maximum communication distance. Let $S_i = S_i' \cdot R$; then, S_i' is uniformly distributed in $[1/2, 1]$. As shown in Figure 6, it can be seen that $R < R' \le 2R$. Because of the randomness distribution of the nodes, we take $R' = 1.5R$. Similarly, we take $R' = 2.5R$ when $k = 3$, $R' = 3.5R$ when $k = 4$, and $R' = 4.5R$ when $k = 5$. Then, (13) can be rewritten as

$$\varphi = \frac{S_1'R + S_2'R + \cdots + S_n'R}{\pi \times 1.5R} = \sum_{i=1}^{n} \frac{S_i'}{1.5\pi}. \tag{14}$$

Table 1: Initial concave/convex node recognition criterion in different values of k.

k	Convex node	Nonconcave/ convex node	Concave node
2	$n \le 5$	$6 \le n \le 7$	$n \ge 8$
3	$n \le 9$	$10 \le n \le 11$	$n \ge 12$
4	$n \le 12$	$13 \le n \le 16$	$n \ge 17$
5	$n \le 16$	$17 \le n \le 21$	$n \ge 22$

Note that each $S_i'/1.5\pi$ is a small random variable which is uniformly distributed in $[1/3\pi, 1/1.5\pi]$. According to the central limit theorem, when n is large enough, the distribution of φ is close to normal distribution. As $E[S_i'/1.5\pi] = 1/2\pi$ and $D[S_i'/1.5\pi] = 1/108\pi^2$, we have $\mu = E[\varphi] = n/2\pi$ and $\sigma^2 = D[\varphi] = n/108\pi^2$. Then,

$$\varphi = \sum_{i=1}^{n} \frac{S_i'}{1.5\pi} \approx \frac{1}{2\pi} \times n = \frac{n}{2\pi}. \tag{15}$$

Let n_1, n_2, and n_3 be the minimum hop count between p_{i1} and p_{i2} when P_i, respectively, is a convex node, a nonconcave/convex node, and a concave node. Then, $\varphi_1 = n_1/2\pi$, $\varphi_2 = n_2/2\pi$, and $\varphi_3 = n_3/2\pi$. According to (12),

$$\begin{aligned} 0 < \frac{n_1}{2\pi} &\le \frac{7}{8}, \\ \frac{7}{8} < \frac{n_2}{2\pi} &< \frac{9}{8}, \\ \frac{9}{8} \le \frac{n_3}{2\pi} &< 2. \end{aligned} \tag{16}$$

Thus, $n_1 \le 5.5$, $5.5 < n_2 < 7.1$, and $7.1 \le n_3 < 12.6$. Since n is a positive integer, we have $1 \le n_1 \le 5$, $n_2 = 6, 7$, and $8 \le n_3 \le 12$. Calculating the different values of n through the above processes, we can get the initial concave/convex node recognition criterion in different values of k based on the minimum hop count, as is shown in Table 1.

In order to make the results more accurate, the values of n_1, n_2, and n_3 will be corrected using the following method. We also take $k = 2$, for example, and choose $n_1 = 3$, $n_2 = 7$, and $n_3 = 10$ during the calculation.

As shown in Figure 7, α and β, respectively, are the intersection points of φ_1 and φ_2 and φ_2 and φ_3. The probability theory shows that when α and β are taken as optimal threshold, the misclassification probability can be minimized.

The normal probability density function is

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right), \tag{17}$$

Table 2: Final concave/convex node recognition criterion in different values of k.

k	Convex node	Nonconcave/ convex node	Concave node
2	$n \le 4$	$5 \le n \le 8$	$n \ge 9$
3	$n \le 7$	$8 \le n \le 13$	$n \ge 14$
4	$n \le 9$	$10 \le n \le 18$	$n \ge 19$
5	$n \le 12$	$13 \le n \le 23$	$n \ge 24$

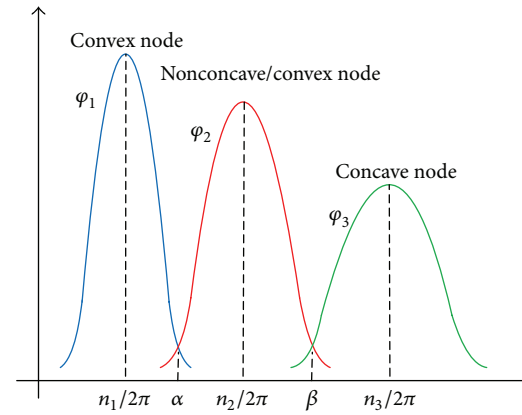


Figure 7: Distribution curve when k is 2.

where $\mu_1 = n_1/2\pi$, $\sigma_1^2 = n_1/108\pi^2$, $\mu_2 = n_2/2\pi$, and $\sigma_2^2 = n_2/108\pi^2$. Then,

$$\begin{aligned} &\frac{1}{\sqrt{2\pi}\sigma_1} \exp\left(-\frac{(\alpha-\mu_1)^2}{2\sigma_1^2}\right) \\ &= \frac{1}{\sqrt{2\pi}\sigma_2} \exp\left(-\frac{(\alpha-\mu_2)^2}{2\sigma_2^2}\right). \end{aligned} \tag{18}$$

Thus,

$$\alpha \approx 0.729. \tag{19}$$

The corresponding minimum hop count of α is $n_1^* = \alpha \cdot 2\pi \approx 4.6$. The value of β can be calculated in the same way. Thus, $\beta \approx 1.332$, and its corresponding minimum hop count is $n_2^* = \beta \cdot 2\pi \approx 8.4$. Since the hop count value is a positive integer, it is easy to know that, in the case of $k = 2$, when the minimum hop count $n \le 4$, P_i is identified as a convex node, when $5 \le n \le 8$, P_i is a nonconcave/convex node, and when $n \ge 9$, P_i is a concave node.

Let k, respectively, be 3, 4, and 5, and calculate the corresponding values of α and β. Then, the corrected value of the minimum hop count n is determined and the concavity or convexity of P_i can be identified, as is shown in Table 2.

Assume that there are M boundary nodes in the network, represented by P_i, $i = 1, 2, \ldots, M$. For an arbitrary node $P_i \in \omega$, it can use its k-hop neighboring boundary nodes p_{i1} and p_{i2} to identify the concavity or convexity of itself, where $k \in$

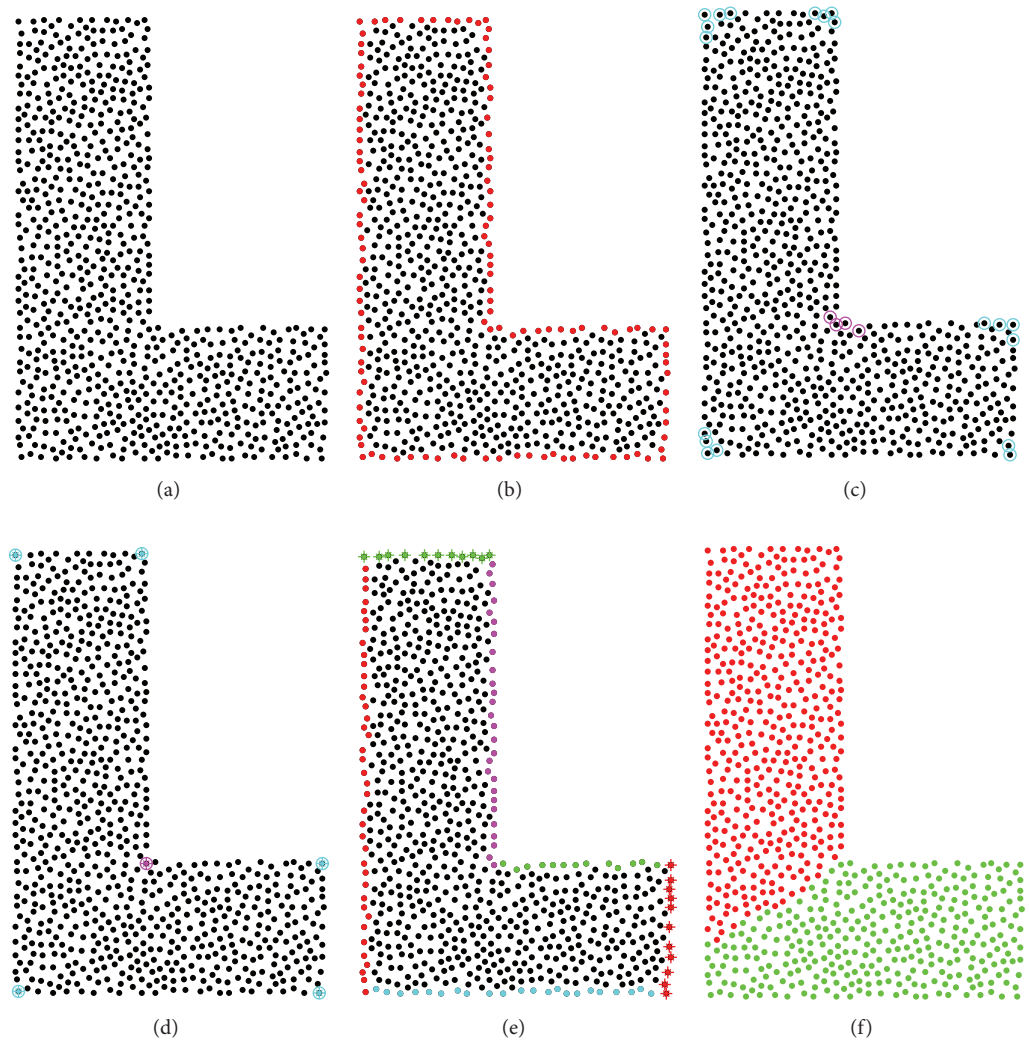


Figure 8: Region decomposition of L-shape network. 763 nodes are uniformly distributed, and average degree of the network is 9.13. (a) Original map. (b) Boundary recognition result. The red nodes are boundary nodes. (c) Concave/convex node recognition result, $k = 3$. The nodes with peach circles are concave nodes, and nodes with cyan circles are convex nodes. (d) Optimal concave/convex nodes. Optimal concave nodes are marked with peach, and optimal convex nodes are marked with cyan. (e) Boundary segmentation result. Different color and markers represent different boundary branches. (f) Region decomposition result. The same color nodes belong to one subregion.

$\{2, 3, 4, 5\}$. According to the above-mentioned mathematical analysis and criterion, we give the concave/convex node recognition algorithm based on the minimum hop count as follows, which includes four steps.

Step 1. For each boundary node P_i, P_i obtains its k-hop neighboring boundary nodes p_{i1} and p_{i2}; that is, $p_{i1}, p_{i2} \in \omega$ and $p_{i1}, p_{i2} \in \partial N_2(P_i)$, where $k \in \{2, 3, 4, 5\}$, and its value is preset by the system.

Step 2. If there are p_{i1} and p_{i2} that meet the condition in Step 1, then turn to Step 3. Otherwise, P_i gives up the concavity/convexity identification of itself.

Step 3. P_i derives the minimum hop count n between p_{i1} and p_{i2}.

Step 4. P_i identifies its concavity or convexity by using the criterion in Table 2.

4.2.3. Concave/Convex Node Cleansing. After concave/convex node recognition algorithm, the concavity or convexity of each boundary node has been identified. Since the sensor network is discrete, it is possible that a nonconcave node which is in the nonconcave region is identified as a concave node by mistake. These mistaken concave nodes often appear in an isolate way. In contrast, there always are several concave nodes in a concave area; that is, real concave nodes always gather together (seen in Figures 8(c), 9(c), and 10(c), e.g.; the black nodes are ordinary nodes, and the red nodes are boundary nodes. The nodes with peach circles are concave nodes, and those with cyan circles are convex nodes. Other

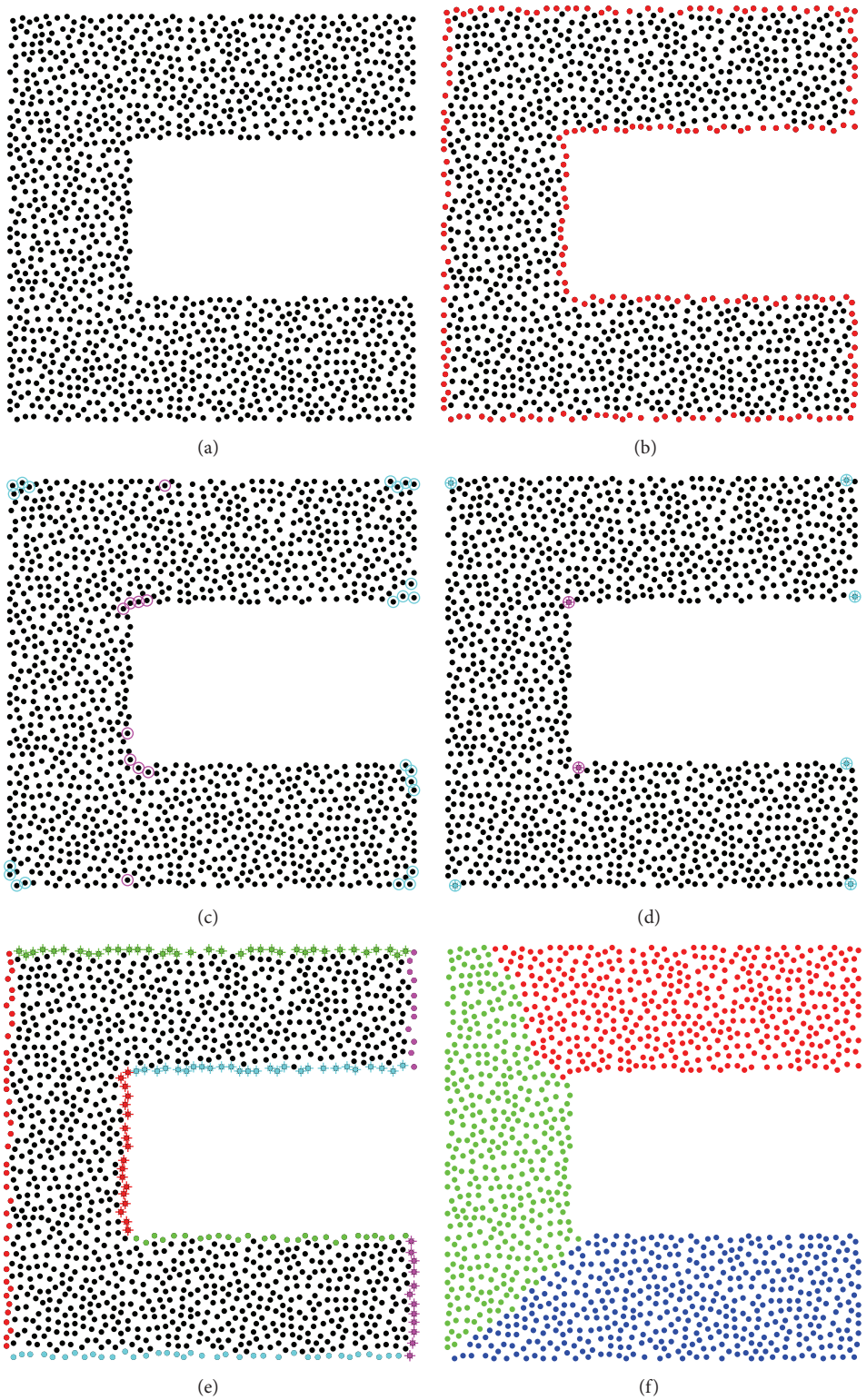


Figure 9: Region decomposition of C-shape network. 1295 nodes are uniformly distributed, and average degree of the network is 9.04. (a) Original map. (b) Boundary recognition result. The red nodes are boundary nodes. (c) Concave/convex node recognition result, $k = 3$. The nodes with peach circles are concave nodes, and the nodes with cyan circles are convex nodes. (d) Optimal concave/convex nodes. Optimal concave nodes are marked with peach, and optimal convex nodes are marked with cyan. (e) Boundary segmentation result. Different color and markers represent different boundary branches. (f) Region decomposition result. The same color nodes belong to one subregion.

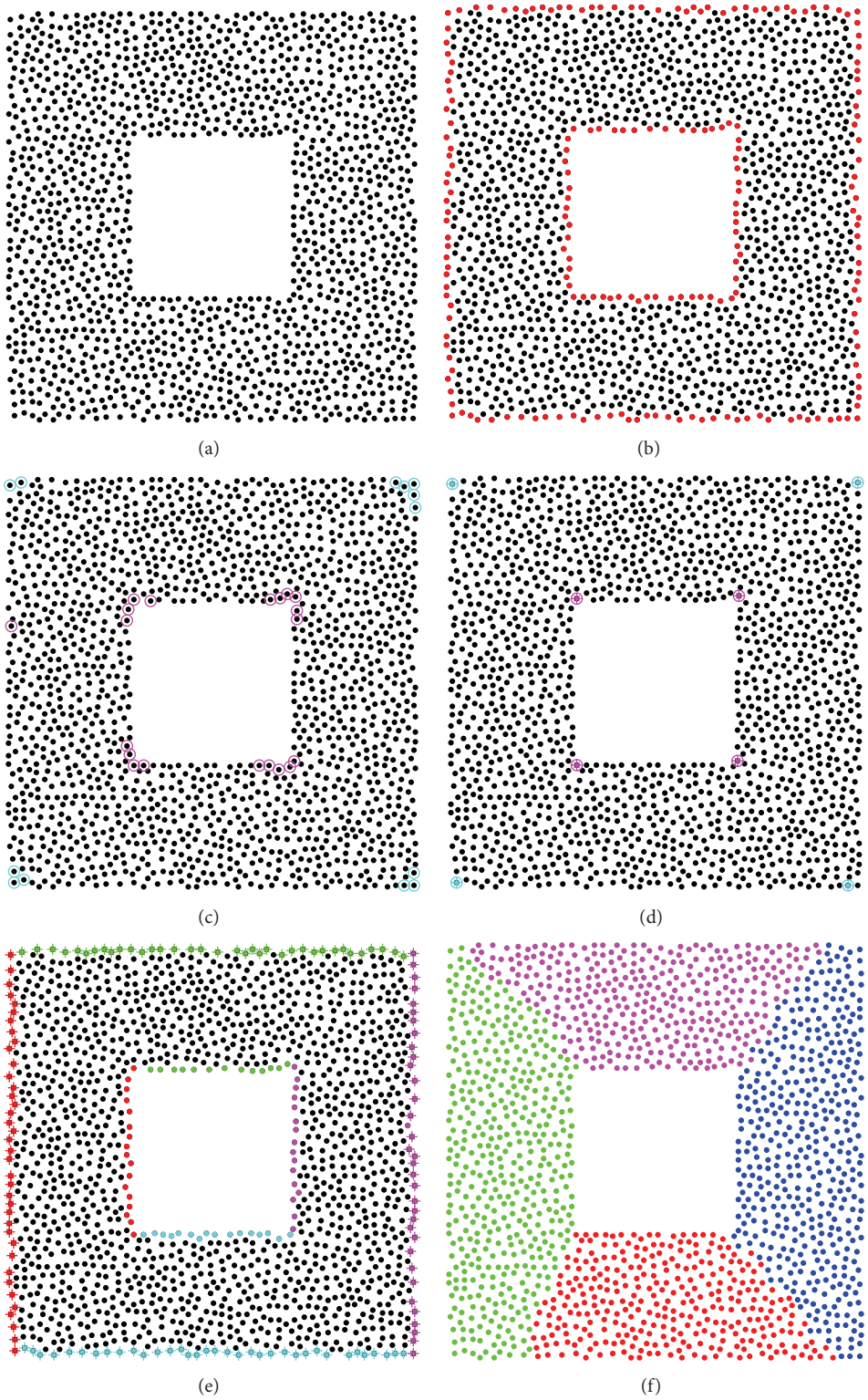


FIGURE 10: Region decomposition of single-window-shape network. 1505 nodes are uniformly distributed, and average degree of the network is 9.01. (a) Original map. (b) Boundary recognition result. The red nodes are boundary nodes. (c) Concave/convex node recognition result, $k = 3$. The nodes with peach circles are concave nodes, and the nodes with cyan circles are convex nodes. (d) Optimal concave/convex nodes. Optimal concave nodes are marked with peach, and optimal convex nodes are marked with cyan. (e) Boundary segmentation result. Different color and markers represent different boundary branches. (f) Region decomposition result. The same color nodes belong to one subregion.

red nodes without any circles are nonconcave/convex nodes). Therefore, the isolate concave node can be directly eliminated from the recognition result (to simplify our algorithm, we define isolate concave node as the concave node which has no concave node within its two hops; similarly, isolate convex node is the convex node which has no convex node within its two hops), as well as the isolate convex node.

For the concave nodes that gather together in one concave area, since larger concavity of the concave node will lead to larger deviation between hop count distance and the Euclidean distance, we choose the concave node with the maximum concavity as the optimal concave node in this area. In our algorithm, the node with maximum concavity corresponds to the node which is with the biggest minimum hop count between its two k-hop neighboring boundary nodes. It is possible that in some concave area there may be more than one concave node with the same biggest minimum hop count. For this case, we randomly choose one of them as the optimal concave node. The above process is called cleansing process.

Similarly, we use the cleansing criterion to cleanse the convex node in the network. The difference is that the convex node with the smallest minimum hop count should be taken as the optimal convex node. If there is more than one convex nodes with the same smallest minimum hop count in a convex area, we randomly choose one of them as the optimal convex node.

Through the cleansing criterion, the optimal concave/convex node is obtained. For example, the cleansing results of Figures 8(c), 9(c), and 10(c), respectively, are Figures 8(d), 9(d), and 10(d), where the optimal concave nodes are marked with peach and the optimal convex nodes are marked with cyan.

4.3. Boundary Segmentation. Assume that the boundary of the network is fully connected. For simplicity, the optimal concave and convex nodes are called the optimal nodes in this section. In order to segment the boundary, we use the optimal nodes to flood Boundary Segmentation PacKeT (BS-PKT) among the boundary nodes. Each BS-PKT concludes the ID number of the node which generates the BS-PKT. When a boundary node A receives a BS-PKT, A deals the current BS-PKT with the following ways:

(1) If A is not a boundary node, A discards BS-PKT.

(2) If A is a boundary node but not an optimal node, A records the ID number in BS-PKT and sends BD-PKT to its neighbor nodes.

(3) If A is both a boundary node and an optimal node, but A has not recorded any ID number of other optimal nodes, then A records the ID number in BS-PKT and does not send BS-PKT anymore.

(4) If A is both a boundary node and an optimal node, and A has recorded an ID number of another optimal node, A discards the BS-PKT.

After flooding, each boundary node will record two different ID numbers of the optimal nodes. Assume the ID numbers, respectively, are I and J. Let $\text{Bound}_{\text{ID}} = [I, J]$. Only when two boundary nodes have the same I and J is their Bound_{ID} defined to be equal. In such situation, the boundary nodes which have the same Bound_{ID} belong to the same boundary branch, and Bound_{ID} is the boundary number of this branch. Thus, the boundary of the network is segmented (e.g., Figures 8(e), 9(e), and 10(e) are the boundary segmentation results of the networks, where different color and markers represent different boundary branches).

4.4. Region Decomposition. For a fully connected anisotropic network, define U as the set of the optimal concave nodes. For an arbitrary optimal concave node $q_i \in U$, its two k-hop neighboring boundary nodes have been obtained in the concave/convex node recognition process. For simplicity, the k-hop neighboring boundary nodes are also called neighboring boundary nodes.

To decompose the network, all neighboring boundary nodes of the optimal concave nodes flood Region Decomposition PacKeT (RD-PKT) in the network. Each RD-PKT concludes the ID number of the node which generates this RD-PKT and the boundary number Bound_{ID} recorded by the node. The format of RD-PKT is (ID, Bound_{ID}). When an arbitrary node B receives a RD-PKT for the first time, B records (ID, Bound_{ID}) in RD-PKT and sends RD-PKT to its neighbor nodes; if B receives a RD-PKT not for the first time, B discards it directly. This policy ensures that most nodes will receive only one RD-PKT from the closest neighboring boundary node.

By this means, each node in the network will record only one RD-PKT, in which the ID and Bound_{ID} number of the corresponding neighboring boundary node are included. The nodes which have the same ID number belong to one subregion; then, the original region decomposition result is obtained. To avoid the network from decomposing to too many subregions, the nodes with the same Bound_{ID} can be merged to one subregion. Finally, an anisotropic network is decomposed into many different subnetworks. See Figures 8(f), 9(f), and 10(f), for example; different colors represent different subnetworks. It is obvious that there is no longer serious bent path between far-away nodes in each subnetwork. Thus, each subnetwork can be considered as an isotropic network.

In summary, after boundary recognition, concave/convex node recognition and cleansing, boundary segmentation, and region decomposition, an anisotropic network finally is decomposed into several different isotropic subnetworks. During the process of HCD, we only need to use the connectivity information of the network, without any additional hardware on sensor nodes. As a range-free localization algorithm, the complexity of communication cost of DV-Hop is $O(N \cdot m)$. Here, N and m are the numbers of sensor nodes and anchor nodes, respectively. For HCD (Hop Count Based Decomposition) scheme, the complexity of communication cost is $O(N)$. Since our proposed HCD-DV-Hop scheme is composed of HCD and DV-Hop,

the complexity of communication cost of HCD-DV-Hop is the same as DV-Hop.

5. Performance Evaluation

To evaluate the proposed algorithm, we simulate on three typical anisotropic network topologies: L-shape, C-shape, and single-window-shape network topologies. In simulations, nodes are uniformly distributed. All nodes have the same communication range. Two nodes are connected if and only if the Euclidean distance between them is smaller than a given communication radius R.

We compare our algorithm with DV-Hop. For comparison, two metrics are used in this section: LE (localization error) and ALE (average localization error). The LE of an unknown node i is calculated by

$$\mathrm{LE}_i = \frac{\sqrt{(x_i - x_i')^2 + (y_i - y_i')^2}}{R} \times 100\%, \tag{20}$$

where (x_i, y_i) is the real location of node i and (x_i', y_i') is the estimated location which is obtained by the localization algorithm. ALE is the average localization error of all unknown nodes in the network, it can be computed by

$$\mathrm{ALE} = \frac{1}{N}\sum_{i=1}^{N}\mathrm{LE}_i, \tag{21}$$

where N is the number of unknown nodes. It is obvious that the smaller the LE and ALE, the better the localization algorithm.

5.1. Region Decomposition. Given three typical anisotropic networks, L-shape network, C-shape network, and single-window-shape network, as shown in Figures 8(a), 9(a), and 10(a), respectively, nodes are uniformly distributed in each network. The communication radius is 40 m. First we use HCD (Hop Count Based Decomposition algorithm) to decompose the networks. Simulation results are shown in Figures 8–10.

5.2. Performance Evaluation of Algorithms under Certain Communication Radius and Ratio of Anchor Node. Here the communication radius is 40 m and the ratio of anchor node is 10%. Anchors are randomly distributed in the networks.

For each anisotropic network, after decomposing, DV-Hop is carried out in each subnetwork. Thus, the LE (localization error) of each unknown node is obtained through HCD-DV-Hop algorithm, as shown in Figures 11(a), 11(c), and 11(e). The LE obtained by DV-Hop is shown in Figures 11(b), 11(d), and 11(f), respectively. Table 3 gives the corresponding ALE (average localization error) of each network by HCD-DV-Hop and DV-Hop.

As shown in Figure 11 and Table 3, we found that whatever the network topology is, HCD-DV-Hop can effectively reduce the LE of most unknown nodes compared with DV-Hop, and the ALE obtained by HCD-DV-Hop is much smaller than that by DV-Hop. Obviously, the reason is that HCD-DV-Hop uses decomposition scheme to avoid serious bent path between far-away nodes participating in localization.

Table 3: ALE of each network by HCD-DV-Hop and DV-Hop algorithm.

Network topology	HCD-DV-Hop	DV-Hop
L-shape	57.8%	172.4%
C-shape	65.8%	595.3%
Single-window-shape	65.7%	161.3%

5.3. Performance Evaluation of Algorithms under Different Ratios of Anchor Node. Given a certain communication radius (take 40 m as the communication radius in the simulations), we examine the performance of HCD-DV-Hop and DV-Hop under different ratios of anchor node. In simulations, we randomly generated 200 networks and computed the ALE of each network. Finally, we take the average value of the 200 ALE as the average error under each ratio of anchor node. Simulation results of L-shape, C-shape, and single-window-shape network are shown in Figures 12(a), 12(b), and 12(c), respectively.

Figure 12 shows that, for different communication radius in Lshape, C-shape, and Single-window-shape networks, compared with DV-Hop, the average error of HCD-DV-Hop is significantly decreased. That is, HCD-DV-Hop performs much better in localization than DV-Hop.

5.4. Performance Evaluation of Algorithms under Different Communication Radius. Given a certain ratio of anchor node (take 10% as the ratio of anchor node in the simulations), we examine the performance of HCD-DV-Hop and DV-Hop under different communication radius. In simulations, we randomly generated 200 networks and computed the ALE of each network. Finally, we take the average value of the 200 ALE as the average error under each communication radius. Simulation results of L-shape, C-shape, and single-window-shape network are shown in Figures 13(a), 13(b), and 13(c), respectively.

From Figure 13, we observe that the average error of our proposed HCD-DV-Hop is much smaller than that of DV-Hop, which obviously proves the localization advantage of HCD-DV-Hop in anisotropic networks.

6. Conclusion and Future Work

In this paper, we have proposed a novel decomposition based localization (HCD-DV-Hop) scheme for anisotropic sensor networks. The main idea of HCD-DV-Hop is first decomposing an anisotropic network into several different isotropic subnetworks in order to avoid the influence of holes or obstacles on the shortest communication path between far-away nodes; then, the typical range-free DV-Hop algorithm

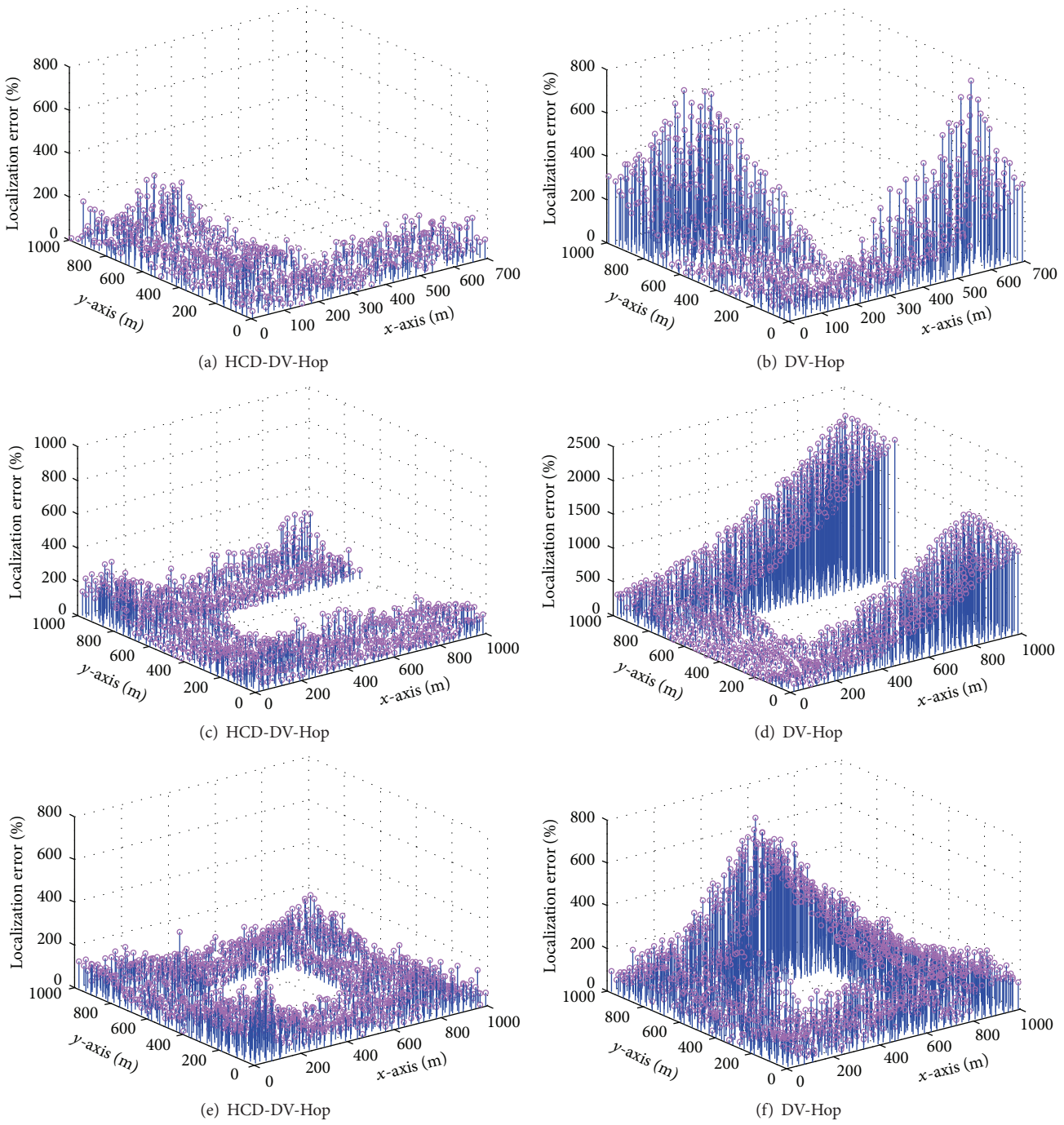
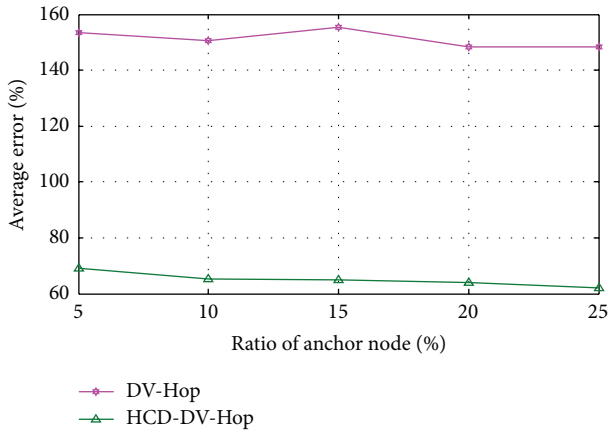


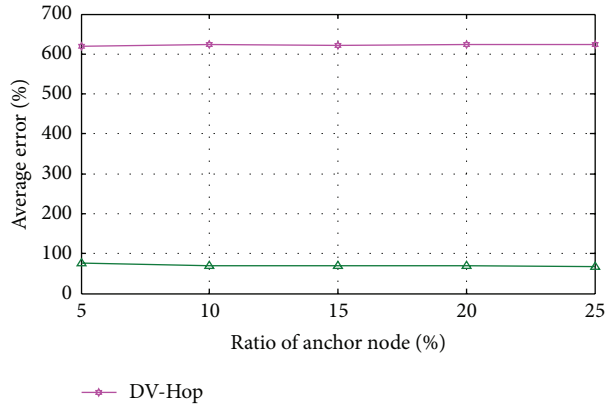
Figure 11: LE (localization error) of each network by different algorithms. (a) LE of L-shape network by HCD-DV-Hop. (b) LE of L-shape network by DV-Hop. (c) LE of C-shape network by HCD-DV-Hop. (d) LE of C-shape network by DV-Hop. (e) LE of single-window-shape network by HCD-DV-Hop. (f) LE of single-window-shape network by DV-Hop.

is used for node localization in each subnetwork. The HCD algorithm is the core of HCD-DV-Hop. It includes boundary recognition, concave/convex node recognition and cleansing, boundary segmentation, and region decomposition. All of these steps are distributed and only require network connectivity information. Simulation results show that no matter how node communication radius or the ratio of anchor node changes, compared with DV-Hop algorithm, the proposed HCD-DV-Hop scheme can effectively reduce localization error in anisotropic networks without increasing the complexity of the algorithm.

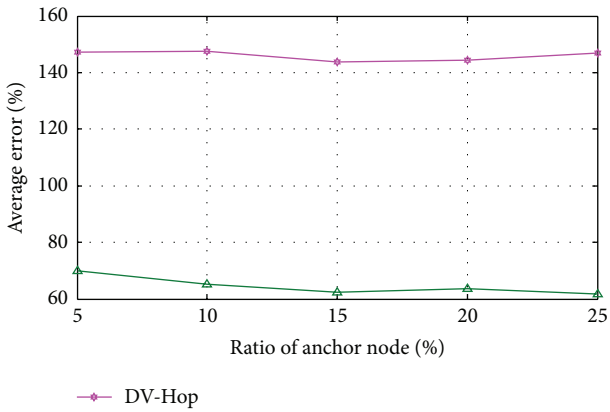
In HCD-DV-Hop, we use the typical range-free DV-Hop algorithm to achieve node localization in each isotropic subnetwork. As HCD scheme is able to decompose



(a) L-shape network

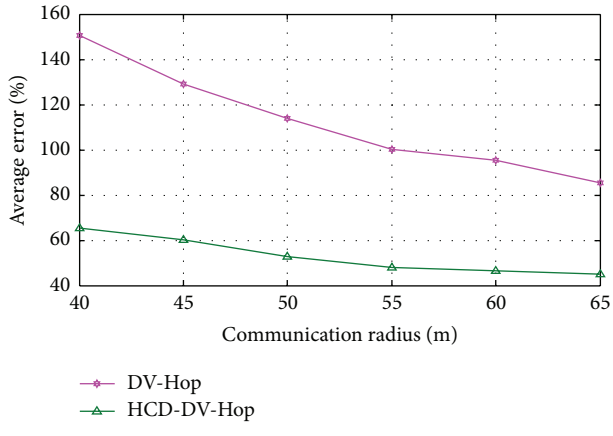


(b) C-shape network

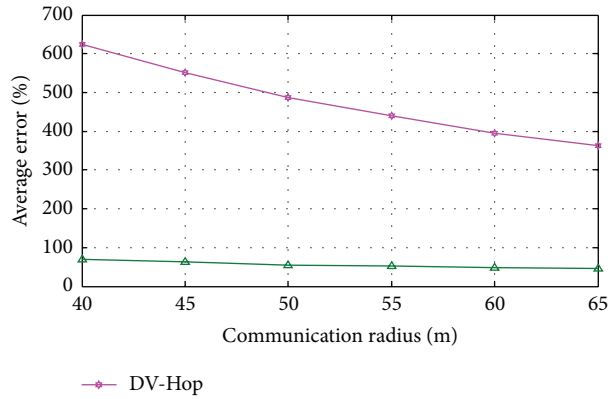


(c) Single-window-shape network

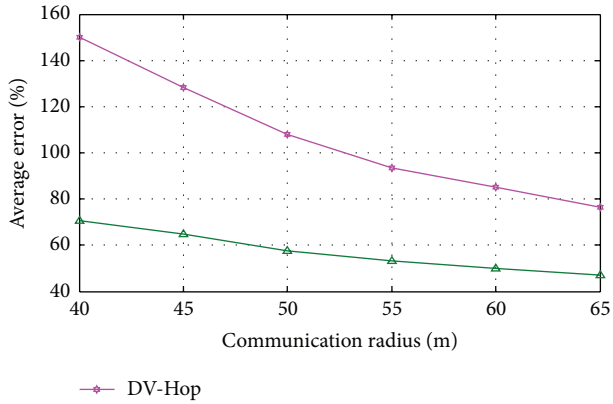
Figure 12: Average error under different ratios of anchor node.



(a) L-shape network



(b) C-shape network



(c) Single-window-shape network

Figure 13: Average error under different communication radius.

an anisotropic network into different isotropic subnetworks, it can be used to investigate the localization performance on the combination of HCD with other range-free algorithms. Additionally, the proposed HCD-DV-Hop scheme is designed for 2D networks. In our future research, we will explore the possible improvements on HCD-DV-Hop in order to adapt to 3D networks.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was supported by the Project NSFC (61170218, 61272461, 61373177, 61070176, 61202198, and 61202393), by the National Key Technology R&D Program (2013BAK01B02), and by the Key Project of Chinese Ministry of Education 211181.

References

[1] S. He, X. Gong, J. Zhang, J. Chen, and Y. Sun, "Curve-based deployment for barrier coverage in wireless sensor networks," *IEEE Transactions on Wireless Communications*, vol. 13, no. 2, pp. 724–735, 2014.

[2] Z. B. Wang, J. Liao, Q. Cao, H. Qi, and Z. Weng, "Achieving k-barrier coverage in hybrid directional sensor networks," *IEEE Transactions on Mobile Computing*, vol. 13, no. 7, pp. 1443–1455, 2014.

[3] H. Ma, X. Zhang, and A. Ming, "A coverage-enhancing method for 3D directional sensor networks," in *Proceedings of the IEEE INFOCOM*, pp. 2791–2795, Rio de Janeiro, Brazil, April 2009.

[4] H. Jiang, J. Cheng, D. Wang, C. Wang, and G. Tan, "Continuous multi-dimensional top-k query processing in sensor networks," in *Proceedings of the IEEE INFOCOM*, pp. 793–801, IEEE, Shanghai, China, April 2011.

[5] H. Jiang, S. Jin, and C. Wang, "Parameter-based data aggregation for statistical information extraction in wireless sensor networks," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 8, pp. 3992–4001, 2010.

[6] H. Jiang, S. Jin, and C. Wang, "Prediction or not? An energy-efficient framework for clustering-based data collection in wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 6, pp. 1064–1071, 2011.

[7] H. Zhang and H. Shen, "Energy-efficient beaconless geographic routing in wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 21, no. 6, pp. 881–896, 2010.

[8] X. Li, J. Yang, A. Nayak, and I. Stojmenovic, "Localized geographic routing to a mobile sink with guaranteed delivery in sensor networks," *IEEE Journal on Selected Areas in Communications*, vol. 30, no. 9, pp. 1719–1729, 2012.

[9] D. Niculescu and B. Nath, "DV based positioning in ad hoc networks," *Telecommunication Systems*, vol. 22, no. 1–4, pp. 267–280, 2003.

[10] Y. Shang and W. Ruml, "Improved MDS-based localization," in *Proceedings of the 23rd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '04)*, pp. 2640–2651, IEEE, Hong Kong, March 2004.

[11] S. Zhang, J. Cao, C. Li-Jun, and D. X. Chen, "Accurate and energy-efficient range-free localization for mobile sensor networks," *IEEE Transactions on Mobile Computing*, vol. 9, no. 6, pp. 897–910, 2010.

[12] G. Wu, S. Wang, B. Wang, Y. Dong, and S. Yan, "A novel range-free localization based on regulated neighborhood distance for wireless ad hoc and sensor networks," *Computer Networks*, vol. 56, no. 16, pp. 3581–3593, 2012.

[13] Y. Zhang, S. Xiang, W. Fu, and D. Wei, "Improved normalized collinearity DV-hop algorithm for node localization in wireless sensor network," *International Journal of Distributed Sensor Networks*, vol. 2014, Article ID 436891, 14 pages, 2014.

[14] H. Lim and J. C. Hou, "Localization for anisotropic sensor networks," in *Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '05)*, vol. 1, pp. 138–149, IEEE, March 2005.

[15] F. D'Amato, P. Gamba, and E. Goldoni, "Monitoring heritage buildings and artworks with wireless sensor networks," in *Proceedings of the 3rd IEEE Workshop on Environmental, Energy, and Structural Monitoring Systems (EESMS '12)*, pp. 1–6, Perugia, Italy, September 2012.

[16] I. Bekmezci and F. Alagöz, "Energy efficient, delay sensitive, fault tolerant wireless sensor network for military monitoring," *International Journal of Distributed Sensor Networks*, vol. 5, no. 6, pp. 729–747, 2009.

[17] M. Li and Y. Liu, "Rendered path: range-free localization in anisotropic sensor networks with holes," in *Proceedings of the 13th Annual ACM International Conference on Mobile Computing and Networking (MobiCom '07)*, pp. 51–62, Montreal, Canada, September 2007.

[18] H. Lim and J. C. Hou, "Distributed localization for anisotropic sensor networks," *ACM Transactions on Sensor Networks*, vol. 5, no. 2, article 11, 26 pages, 2009.

[19] Z. Fan, Y. Chen, L. Wang, L. Shu, and T. Hara, "Removing heavily curved path: improved DV-hop localization in anisotropic sensor networks," in *Proceedings of the 7th International Conference on Mobile Ad-hoc and Sensor Networks (MSN '11)*, pp. 75–82, Beijing, China, December 2011.

[20] H. Huang, G. Chen, Y.-E. Sun, X. Li, and L. Huang, "A range-free localization algorithm in concave areas," in *Proceedings of the 4th IEEE International Conference on Wireless Communications, Networking and Mobile Computing*, pp. 1–5, IEEE, Dalian, China, October 2008.

[21] G. Tan, H. Jiang, S. Zhang, Z. Yin, and A.-M. Kermarrec, "Connectivity-based and anchor-free localization in large-scale 2D/3D sensor networks," *ACM Transactions on Sensor Networks*, vol. 10, no. 1, article 6, 2013.

[22] C. Wang and L. Xiao, "Locating sensors in concave areas," in *Proceedings of the 25th IEEE International Conference on Computer Communications (INFOCOM '06)*, pp. 1–12, Barcelona, Spain, April 2006.

[23] M. Jin, S. Xia, H. Wu, and X. Gu, "Scalable and fully distributed localization with mere connectivity," in *Proceedings of the IEEE INFOCOM*, pp. 3164–3172, IEEE, Shanghai, China, April 2011.

[24] Y. Wang, S. Lederer, and J. Gao, "Connectivity-based sensor network localization with incremental delaunay refinement method," in *Proceedings of the 28th IEEE International Conference on Computer Communications (INFOCOM '09)*, pp. 2401–2409, IEEE, Rio de Janeiro, Brazil, April 2009.

[25] S. Lederer, Y. Wang, and J. Gao, "Connectivity-based localization of large-scale sensor networks with complex shape," *ACM Transactions on Sensor Networks*, vol. 5, no. 4, article 31, 2009.

[26] H. Chen, W. Lou, Z. Wang, J. Wu, Z. B. Wang, and A. Xi, "Securing DV-Hop localization against wormhole attacks in wireless sensor networks," *Pervasive and Mobile Computing*, vol. 16, part A, pp. 22–35, 2015.

[27] Q. Fang, J. Gao, and L. J. Guibas, "Locating and bypassing holes in sensor networks," *Journal of Mobile Networks and Applications*, vol. 11, no. 2, pp. 187–200, 2006.

[28] S. P. Fekete, A. Kröller, D. Pfisterer, S. Fischer, and C. Buschmann, "Neighborhood-based topology recognition in sensor networks," in *Algorithmic Aspects of Wireless Sensor Networks: First International Workshop, ALGOSENSORS 2004, Turku, Finland, July 16, 2004. Proceedings*, vol. 3121 of *Lecture Notes in Computer Science*, pp. 123–136, Springer, Berlin, Germany, 2004.

[29] D. Dong, Y. Liu, and X. Liao, "Fine-grained boundary recognition in wireless ad hoc and sensor networks by topological methods," in *Proceedings of the 10th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '09)*, pp. 135–144, ACM, New Orleans, La, USA, May 2009.

[30] O. Saukh, R. Sauter, M. Gauger, and P. J. Marrón, "On boundary recognition without location information in wireless sensor networks," *ACM Transactions on Sensor Networks*, vol. 6, no. 3, article 20, 2010.

[31] Y. Wang, J. Gao, and J. S. B. Mitchell, "Boundary recognition in sensor networks by topological methods," in *Proceedings of the 12th Annual International Conference on Mobile Computing and Networking (MobiCom '06)*, pp. 122–133, ACM, Los Angeles, Calif, USA, September 2006.

[32] L. D. Baumert, W. H. Mills, and R. L. Ward, "Uniform cyclotomy," *Journal of Number Theory*, vol. 14, no. 1, pp. 67–82, 1982.