Research Article

An Eliminating Method to Improve Localization Accuracy

Liangyin Chen, Bingshu Yan, Junjun He, Jingyu Zhang, Wen Chen, Yushi Jiang, Yan Liu, Qian Luo, and Baoqiu Wang

1 College of Computer Science, Sichuan University, Chengdu 610064, China
2 College of Electrical Information, Sichuan University, Chengdu 610064, China
3 School of Economic and Management, Southwest Jiaotong University, Chengdu 610044, China
4 School of Software and Microelectronics, Peking University, Beijing 100871, China
5 The Second Research Institute of Civil Aviation Administration of China Logistics Technology Branch, Chengdu 610041, China
6 Department of Computer Science and Engineering, Jinjiang College, Sichuan University, Pengshan 620860, China

Correspondence should be addressed to Liangyin Chen; chenliangyin@gmail.com

Received 24 January 2014; Revised 15 May 2014; Accepted 18 May 2014; Published 12 June 2014

Academic Editor: Ki-II Kim

Node localization is essential in wireless sensor networks (WSNs). However, due to limited node energy and overhead network, node localization faces many challenges in WSNs. In addition, it is difficult to meet the requirement of localization accuracy and energy consumption at the same time in existing algorithms. In this paper, we proposed a novel eliminating algorithm called NPLA which achieves high localization accuracy and reduces energy consumption effectively. It makes full use of multiple-hop neighbor nodes of unknown node to eliminate the possible location area (PLA) of the unknown node. Simulations show that the localization error of algorithm NPLA can be controlled within 20%.

1. Introduction

Wireless sensor networks (WSNs) have been widely applied in target tracking, intrusion detection, and some related fields. However, most of such applications are based on the location information of the target. Therefore, node localization is essential and fundamental in the research of WSNs.

So far, there are many localization algorithms exiting in the world. Based on whether they measure the distance or not, those localization algorithms can be grouped into two categories: the range-based method and the range-free method [1, 2]. The range-based localization method estimates the node position by measuring the physical distance or the angle among sensor nodes. After the distance or angle has been determined, the location can be estimated. There are many rang-based localization methods which had been proposed during the past decades such as time of arrival (TOA) [3], time difference of arrival (TDOA) [4], angle of arrival (AOA) [5], and received signal strength indicator (RSSI) [6]. And many of them had been applied in practical scene, for example, TOA based indoor human tracking system [7] and the real-time health monitoring system [8]. In spite of the good localization accuracy of the range-based method, the study of node localization is recently focused on the range-free method. That is because the range-free methods such as DV-Hop [9], Amorphous [10], Bounding Box [11], APIT [12], DRLS [13], and SOM [14] do not need additional hardware and can be deployed easily. However, there are some common problems of the present range-free methods: (1) the localization accuracy is low and (2) the computation is complex. These problems cause more energy consumption to the overhead network.

In order to improve the localization accuracy and reduce the computation, based on the possible location area (PLA) of the unknown node, we propose an eliminating algorithm named NPLA. It eliminates the possible location area (PLA) constantly, where the unknown node was located in. In our algorithm, in addition to utilizing the communication range and hop-count message of all nodes, we also take full advantage of the possible location circle (PLC) of
the unknown node to improve the accuracy of localization. Given the fact that the unknown node in WSNs is majority, NPLA can still obtain better localization accuracy. Analysis of simulation shows that the proposed NPLA algorithm can improve the localization accuracy significantly.

The rest of the paper is organized as follows. Section 2 lists the related work. Section 3 depicts the definitions and assumptions. Section 4 describes the algorithm. Section 5 describes the algorithm implementation. Section 6 introduces the simulation settings and analyzes the simulation performance. Finally, Section 7 concludes the paper.

### 2. Related Work

Over the past ten years, node localization algorithm mainly concentrates on the range-free field in wireless sensor network, and many range-free localization algorithms have been proposed, such as DV-Hop [9], Amorphous [10], Bounding Box [11], APIT [12], DRLS [13], and SOM [14].

DV-Hop algorithm proposed by Niculescu and Nath is the earliest range-free localization algorithm [9]. This algorithm first gets hop count between nodes by flooding information. Second, it calculates average hop distance using the hop count between anchor nodes. Third, it calculates the distance between unknown nodes and anchor nodes utilizing average hop distance and hop count. At last, it computes node coordinates using trilateration. DV-Hop algorithm achieves better effects in a homogeneous distribution network. However, if the nodes in the network are unevenly distributed, average hop distance calculation will have larger deviations; for this reason, the localization accuracy will be affected seriously.

Based on the DV-Hop, Nagpal et al. proposed algorithm Amorphous [10] which uses node communication radius as the average distance per hop. By this algorithm, although the network communication overhead is reduced and the calculation is simple, the positioning error is large. And subsequent algorithms modified the average hop distance to improve the positioning accuracy.

Simic and Sastry proposed Bounding Box [11] at the same period. The algorithm reduces the possible location area of unknown node by using the one-hop anchor neighbors of the node. And the implement of this algorithm is simpler. However, in this algorithm, it considers that the communication radius of all nodes is the same which is inconsistent with the real world scenario. And it also needs the higher density of anchor nodes.

He et al. introduced the point-in-triangulation test (PIT) to the wireless sensor networks and proposed APIT [12] localization algorithm. In APIT [12], an unknown node uses the PIT method to test whether it locates in the triangles constituted by its three different one-hop anchor neighbors or not. If the unknown node locates within several triangles, the possible position of it is the overlapping region of those triangles. However, those unknown nodes whose one-hop anchor neighbor number is less than three cannot be located. In addition, the implement of APIT [12] is simple, but the communication cost is relatively large for PIT testing.

To improve accuracy, Sheu et al. proposed a distributed range-free localization scheme (DRLS) [13] for static WSNs. In this scheme, each node gathers the nearby anchors’ locations and then estimates its own location. The paper improved the grid-scan algorithm and derived a vector-based refinement scheme.

Giorgetti et al. proposed the SOM (self-organizing maps) algorithm [14], which is based on a neural network formalism known as self-organizing map and generates virtual coordinates that describe the relative positions of nodes [10]. The SOM [14] algorithm estimates the distance between every possible pairs of nodes roughly by using the shortest path algorithm and deduces the pairs of nodes distance by utilizing multidimensional scaling method and locates unknown nodes by using the position of anchor nodes.

To improve the accuracy of existing range-free localization algorithm, we propose a scheme called NPLA. We contrast the localization performance between DV-Hop [9], APIT [12], SOM [14], and NPLA in Section 6. According to the results of simulation, NPLA performs best in most scenarios.
3. Definition and Assumptions

3.1. Definition. To describe our method more conveniently, we list those definitions and symbols used in our paper in the Abbreviations Section. Before describing our method, we explain some definitions as follows.

3.2. Assumptions

Assumption 1. Every anchor node knows its own standard communication radius: the maximum communication radius and the minimum communication radius.

Assumption 2. No auxiliary devices in the wireless sensor network.

Assumption 3. Every node can find its neighbors by utilizing the neighbor discovery algorithm [15, 16].

4. Algorithm Description

In NPLA, every unknown node synthesizes the position information and hop-count of all neighbors to itself, which includes not only anchor neighbors but also unknown neighbors, to determine its estimated position. In detail, every unknown node utilizes its neighbors’ virtual communication circles relative to itself and the inevitable communication circles of its neighbors to narrow its own PLA. After traversing all anchor neighbors in the network, every unknown node also traverses all its unknown neighbors to help it to locate itself. When the value of the unknown node’s PLA will not change, then the unknown node computes the gravity center of its present PLA as its estimated position.

The concrete method includes the following steps.

Step 1. Unknown node \( U_j \) puts the network area \( \Omega \) as the initial PLA.

Step 2. Unknown node \( U_j \) finds all its anchor neighbors in the network by utilizing neighbor discovery algorithm. And, it collects the position information of its anchor neighbors as well as hop-count information between its anchor neighbors and itself.

Step 3. Unknown node \( U_j \) traverses all its anchor neighbors in the ascending order of hop-count which is from its anchor neighbor to itself. Above all, unknown node \( U_j \) calculates \( V_{VCC}(U_j, A_i) \) (the virtual communication circle of anchor neighbor \( A_i \) relative to node \( U_j \)). If the hop-count from a anchor neighbor to unknown node \( U_j \) is 1, the virtual communication circle of this anchor neighbor relative to node \( U_j \) is the possible communication circle of this anchor neighbor. There are two situations of position between circle \( V_{VCC}(U_j, A_i) \) and PLA of node \( U_j \): the circle \( V_{VCC}(U_j, A_i) \) passes through PLA of node \( U_j \); the circle \( V_{VCC}(U_j, A_i) \) contains PLA of node \( U_j \). The descriptions of these two situations are shown as follows.
(1) The circle $V_{VCC}(U_j, A_s)$ passes through the PLA of node $U_j$: firstly, the algorithm finds all the vertexes of $U_j$’s present PLA which locates outside the circle $V_{VCC}(U_j, A_s)$ and then links every found vertex to the GPS positional coordinates of anchor neighbor $A_s$ to get a line, respectively. There is a point of intersection between every line and the circle $V_{VCC}(U_j, A_s)$. Secondly, draw a tangent of the circle $V_{VCC}(U_j, A_s)$ through every intersection point, respectively. Every tangent cuts node $U_j$’s present PLA to two parts. Lastly, put every tangent as a dividing line to eliminate the region of $U_j$’s present PLA, which is away from the circle $V_{VCC}(U_j, A_s)$, respectively. The remaining part is the new PLA of node $U_j$.

(2) The circle $V_{VCC}(U_j, A_s)$ contains the present PLA of node $U_j$: $U_j$ keeps its present PLA unchanged.

Step 4. Unknown node $U_j$ traverses all its anchor neighbors in the ascending order of hop-count which is from its anchor neighbor to itself. Here, hop-count is bigger than 1. Above all, unknown node $U_j$ determines its present PLC and $V_{ICC}(A_k)$ (the inevitable communication circle of anchor neighbor $A_k$). There are two situations of position between node $U_j$’s present PLC and circle $V_{ICC}(A_k)$: there is an intersection between node $U_j$’s present PLC and circle $V_{ICC}(A_k)$; there is no intersection between node $U_j$’s present PLC and circle $V_{ICC}(A_k)$. The descriptions of these two situations are shown as follows.

(1) There is an intersection between node $U_j$’s present PLC and circle $V_{ICC}(A_k)$: in this case, there must be two intersection points between node $U_j$’s present PLC and the circle $V_{ICC}(A_k)$. Firstly, the algorithm links these two points of intersection to get a line, and the line cuts node $U_j$’s present PLA into two parts. Lastly, put the line as a dividing line to eliminate the region of $U_j$’s present PLA close to the circle $V_{ICC}(A_k)$. The remaining part is the new PLA of node $U_j$.

(2) There is no intersection between node $U_j$’s present PLC and circle $V_{ICC}(A_k)$: $U_j$ keeps its present PLA unchanged.

Step 5. Every unknown node executes Step 2 to Step 4 to get a new PLA. Unknown node $U_j$ finds all its unknown neighbors in the network utilizing neighbor discovery algorithm and gets the position as well as hop-count information of its unknown neighbors to itself.

Step 6. Unknown node $U_j$ traverses all its unknown neighbors in the ascending order of hop-count which is from its anchor neighbor to itself. Above all, unknown node $U_j$ determines the present $V_{VCC}(U_j, U_i)$ (the virtual communication circle of unknown neighbor $U_i$ relative to node $U_j$; if the hop-count which is from an unknown neighbor to unknown node $U_j$ is 1, the virtual communication circle of this unknown neighbor relative to node $U_j$ is the possible communication circle of this unknown neighbor). There are two situations of position between circle $V_{VCC}(U_j, U_i)$ and unknown node $U_j$’s present PLA: the circle $V_{VCC}(U_j, U_i)$ passes through unknown node $U_j$’s present PLA; the circle $V_{VCC}(U_j, U_i)$ contains unknown node $U_j$’s present PLA. The descriptions of these two situations are shown as follows.

(1) The circle $V_{VCC}(U_j, U_i)$ passes through unknown node $U_j$’s present PLA: firstly, the algorithm finds all the vertexes of node $U_i$’s present PLA outside the circle $V_{VCC}(U_j, U_i)$ and then links every found vertex to the center of the unknown neighbor $U_j$’s present PLC to get a line, respectively. There is an intersection point between every line and the circle $V_{VCC}(U_j, U_i)$. Secondly, draw a tangent of the circle $V_{VCC}(U_j, U_i)$ through every intersection point, respectively. Every tangent cuts node $U_j$’s present PLA into two parts. Lastly, put every tangent as a dividing line to eliminate the region of $U_j$’s present PLA, which is away from the circle $V_{VCC}(U_j, U_i)$, respectively. The remaining part is the new PLA of node $U_j$.

(2) The circle $V_{VCC}(U_j, U_i)$ contains present PLA of node $U_j$: $U_j$ keeps its present PLA unchanged.

Step 7. Unknown node $U_j$ traverses all its unknown neighbors in the ascending order of hop-count which is from its unknown neighbors to itself. Here, hop-count is bigger than 1. Above all, unknown node $U_j$ determines its present PLC and $V_{ICC}(U_p)$ (the inevitable communication circle of unknown neighbor $U_p$). There are two situations of position between node $U_j$’s present PLC and circle $V_{ICC}(U_p)$: there is an intersection between node $U_j$’s present PLC and circle $V_{ICC}(U_p)$; there is no intersection between node $U_j$’s present PLC and circle $V_{ICC}(U_p)$. The descriptions of these two situations are shown as follows.

(1) There is an intersection between node $U_j$’s present PLC and circle $V_{ICC}(U_p)$: in this case, there must be two intersection points between node $U_j$’s present PLC and the circle $V_{ICC}(U_p)$. Firstly, the algorithm links these two intersection points to get a line, and the line cuts node $U_j$’s present PLA into two parts. Lastly, put this line as a dividing line to eliminate the region of $U_j$’s present PLA close to the circle $V_{ICC}(U_p)$. The remaining part is the new PLA of node $U_j$.

(2) There is no intersection between node $U_j$’s present PLC and circle $V_{ICC}(U_p)$: $U_j$ keeps its present PLA unchanged.

Step 8. Unknown node $U_j$ executes Step 2 to Step 7 iteratively until its present PLA is not changed. Then the unknown node $U_j$ gets a final PLA. $U_j$ computes the gravity center of its final PLA to obtain its estimated position. Similarly, every unknown node in the network executes the same operation as node $U_j$ does to obtain an estimated position.

Pseudocode of algorithm NPLA is shown in Pseudocode 1.
Pseudocode 1: Continued.

(1) for (j = m + 1; j ≤ n; j++)
(2) do 𝑉_{PLA}(𝑈_j) ← Ω (Ω: the area of whole network)
(3) The maximum hop found is \(h_m\)
(4) Phase 1: \(U_j\) eliminates PLA by utilizing \(V_{VCC}(U_j, A_j)\)
(5) Phase 2: \(U_j\) eliminates PLA by utilizing \(V_{ICC}(A_j)\)
(7) do Find each \(U_j\) who is an \(I\)-hop unknown neighbor
(8) node of \(U_j\)
(9) The maximum hop found is \(l_m\)
(10) Phase 3: \(U_j\) eliminates PLA by utilizing \(V_{VCC}(U_j, U_i)\)
(11) Phase 4: \(U_j\) eliminates PLA by utilizing \(V_{ICC}(U_i)\)

Phase 1: \(U_j\) eliminates PLA by utilizing \(V_{VCC}(U_j, A_j)\)
(1) for (ℎ = 1; ℎ ≤ ℎ_m; ℎ++)
(2) |
(3) for (𝑠 = 1; 𝑠 ≤ ℓ; 𝑠++)
(4) |
(5) Draw \(V_{VCC}(U_j, A_j)\)
(6) if \(V_{VCC}(U_j, A_j)\) passes through \(V_{PLA}(U_j)\)
(7) then \(U_j\) eliminates the part of present \(V_{PLA}(U_j)\),
(8) which is away from \(V_{VCC}(U_j, A_j)\)
(9) if \(V_{VCC}(U_j, A_j)\) contains \(V_{PLA}(U_j)\)
(10) then \(U_j\) keeps present \(V_{PLA}(U_j)\)
(11)}
(12)}

Phase 2: \(U_j\) eliminates PLA by utilizing \(V_{ICC}(A_j)\)
(1) for (ℎ = 2; ℎ ≤ ℎ_m; ℎ++)
(2) |
(3) for (𝑠 = 1; 𝑠 ≤ ℓ; 𝑠++)
(4) |
(5) Draw \(V_{ICC}(A_j)\)
(6) if there is an intersection between \(V_{PLC}(U_j)\) and \(V_{ICC}(A_j)\)
(7) then \(U_j\) eliminates the part of present \(V_{PLA}(U_j)\),
(8) which is close to \(V_{ICC}(A_j)\)
(9) if \(V_{ICC}(A_j)\) contains \(V_{PLA}(U_j)\)
(10) if there is not an intersection between \(V_{PLC}(U_j)\) and \(V_{ICC}(A_j)\)
(11)}
(12)}

Phase 3: \(U_j\) eliminates PLA by utilizing \(V_{VCC}(U_j, U_i)\)
(1) for (𝑖 = 1; 𝑖 ≤ 𝑖_m; 𝑖++)
(2) |
(3) for (𝑡 = 𝑚 + 1; 𝑡 ≤ ℓ; 𝑡++)
(4) |
(5) Draw \(V_{VCC}(U_j, U_i)\)
(6) if \(V_{VCC}(U_j, U_i)\) passes through \(V_{PLA}(U_j)\)
(7) then \(U_j\) eliminates the part of present \(V_{PLA}(U_j)\),
(8) which is away from \(V_{VCC}(U_j, U_i)\)
(9) if \(V_{VCC}(U_j, U_i)\) contains \(V_{PLA}(U_j)\)
(10) then \(U_j\) keeps present \(V_{PLA}(U_j)\)
(11)}
(12)}

Phase 4: \(U_j\) eliminates PLA by utilizing \(V_{ICC}(U_i)\)
(1) for (𝑖 = 2; 𝑖 ≤ 𝑖_m; 𝑖++)
(2) |
(3) for (𝑡 = ℓ + 1; 𝑡 ≤ ℓ; 𝑡++)
(4) |

Pseudocode 1: Continued.
5. Algorithm Implementation

Now we show the specific algorithm steps for unknown node $U_{m+1}$ to explain how the algorithm NPLA is implemented.

**Step 1.** Unknown node $U_{m+1}$ puts the network area $\Omega$ as its initial PLA.

**Step 2.** Unknown node $U_{m+1}$ finds all its anchor neighbors in the network utilizing neighbor discovery algorithm and gets the position information and hop-count (hop-count is less than 3) to itself. Three anchor neighbors are found: $A_1$, $A_2$, and $A_3$; $A_1$ is the 1 hop anchor neighbor of unknown node $U_{m+1}$; $A_2$ and $A_3$ are 2 hop anchor neighbors of unknown node $U_{m+1}$.

**Step 3.** Unknown node $U_{m+1}$ traverses all its anchor neighbors in the ascending order of hop-count which is from its anchor neighbor to itself.

**Step 3.1.** Unknown node $U_{m+1}$ uses the $V_{PCC}(A_1)$ of anchor node $A_1$ ($V_{VCC}(U_{m+1}, A_1)$), as shown in Figure 4(a).

1. Draw the circle $V_{PCC}(A_1)$ of anchor node $A_1$.
2. Find all the vertexes of $U_{m+1}$’s present PLA outside the circle $V_{PCC}(A_1)$ and obtain four vertexes: $V1, V2, V3,$
and V4. Then, link anchor node A1’s GPS coordinate position to every vertex of PLAs, respectively. Then there are four points of intersection between these four lines and circle V_PCC(A1): P1, P2, P3, and P4.

3. Draw the tangent of circle V_PCC(A1) through P1, P2, P3, and P4, respectively.

4. Every tangent in (3) cuts U_m+1’s present PLA into two parts. Then put every tangent as a dividing line to eliminate the region of U_m+1’s present PLA away from the circle V_PCC(A1), respectively. The remaining part (convex polygon ABCD) is the new PLA of U_m+1.

Step 3.2. Unknown node U_m+1 uses the V_VCC(U_m+1, A2) of anchor node A2, as shown in Figure 4(b).

1. Draw the circle V_VCC(U_m+1, A2).
2. Find all the vertexes of U_m+1’s present PLA outside the circle V_VCC(U_m+1, A2) and obtain three vertexes: B, A, and D. Then link anchor node A2’s GPS coordinate position to every vertex to get a line, respectively. Then there are three points of intersection between these three lines and circle V_VCC(U_m+1, A2): Q1, Q2, and Q3.

3. Draw the tangent of circle V_VCC(U_m+1, A2) through Q1, Q2, and Q3, respectively.
4. Every tangent in (3) cuts U_m+1’s present PLA into two parts. Then put every tangent as a dividing line to eliminate the region of U_m+1’s present PLA which is away from the circle V_VCC(U_m+1, A2), respectively. The remaining part (convex polygon CEFGH) is the new PLA of U_m+1.

Step 3.3. Unknown node U_m+1 uses the V_VCC(U_m+1, A3) of anchor node A3, as shown in Figure 4(c).

1. Draw the circle V_VCC(U_m+1, A3).
2. Find all the vertexes of U_m+1’s present PLA outside the circle V_VCC(U_m+1, A3) and obtain two vertexes: H and G. Then link anchor node A3’s GPS coordinate position to every vertex to get a line, respectively. Then there are two points of intersection between these two lines and circle V_VCC(U_m+1, A3).
3. Draw the tangent of circle V_VCC(U_m+1, A3) through the two points of intersection in (2), respectively.
4. Every tangent in (3) cuts U_m+1’s present PLA into two parts. Then put every tangent as a dividing line to eliminate the region of U_m+1’s present PLA which is away from the circle V_VCC(U_m+1, A3), respectively. The remaining part (convex polygon CEFIJ) is the new PLA of U_m+1.

Step 4. Unknown node U_m+1 traverses all its anchor neighbors in the ascending order of hop-count which is from its anchor neighbor to itself. Here, hop-count is bigger than 1.

Step 4.1. Unknown node U_m+1 uses the V_ICC(A2) of anchor node A2, as shown in Figure 5.

Step 5. Every unknown node U_j (m + 1 ≤ j ≤ n) executes Step 2 to Step 4 to get a new PLA. Unknown node U_m+1 finds
all its unknown neighbors in the network utilizing neighbor discovery algorithm and gets the position information and hop-count (hop-count is less than 3) to itself. Three unknown neighbors are found: $U_{m+2}$, $U_{m+3}$, and $U_{m+4}$; $U_{m+2}$ is the 1 hop unknown neighbor of unknown node $U_{m+1}$ and $U_{m+3}$ and $U_{m+4}$ are 2 hop unknown neighbors of unknown node $U_{m+1}$.

Step 6. Unknown node $U_{m+1}$ traverses all its unknown neighbors in the ascending order of hop-count which is from its unknown neighbor to itself.

Step 6.1. Unknown node $U_{m+1}$ uses the $V_{PCC}(U_{m+2})$ of unknown node $U_{m+2}$ ($V_{VCC}(U_{m+1}, U_{m+2})$), as shown in Figure 7.

1. Draw the circle $V_{PCC}(U_{m+2})$.
2. Find all the vertexes of $U_{m+1}$'s present PLA outside the circle $V_{PCC}(U_{m+2})$ and obtain one vertex: $K$. Then link $O_{m+2}$ to vertex $K$ to get a line. Then there is one intersection point between the line and circle $V_{PCC}(U_{m+2})$.
3. Draw the tangent of circle $V_{PCC}(U_{m+2})$ through the point of intersection in (2).
4. The tangent in (3) cuts $U_{m+1}$'s present PLA into two parts. Then put this tangent as a dividing line to eliminate the region of $U_{m+1}$'s present PLA which is away from the circle $V_{PCC}(U_{m+2})$. The remaining part (convex polygon FIOPMN) is the new PLA of $U_{m+1}$.

Step 6.2. Unknown node $U_{m+1}$ uses the $V_{VCC}(U_{m+1}, U_{m+3})$ of unknown node $U_{m+3}$, as shown in Figure 8.

1. Draw the circle $V_{VCC}(U_{m+1}, U_{m+3})$.
2. Find all the vertexes of $U_{m+1}$'s present PLA outside the circle $V_{VCC}(U_{m+1}, U_{m+3})$ and obtain one vertex: $N$. Then link $O_{m+3}$ to vertex $N$ to get a line. Then there is one point of intersection between the line and circle $V_{VCC}(U_{m+1}, U_{m+3})$.
3. Draw the tangent of circle $V_{VCC}(U_{m+1}, U_{m+3})$ through the point of intersection in (2).
4. The tangent in (3) cuts $U_{m+1}$'s present PLA into two parts. Then put this tangent as a dividing line to eliminate the region of $U_{m+1}$'s present PLA which is away from the circle $V_{VCC}(U_{m+1}, U_{m+3})$. The remaining part (convex polygon FIOPMQR) is the new PLA of $U_{m+1}$.

Step 6.3. Unknown node $U_{m+1}$ uses the $V_{VCC}(U_{m+1}, U_{m+4})$ of unknown node $U_{m+4}$, as shown in Figure 9.

1. Draw the circle $V_{VCC}(U_{m+1}, U_{m+4})$.
2. Find all the vertexes of $U_{m+1}$'s present PLA outside the circle $V_{VCC}(U_{m+1}, U_{m+4})$ and obtain two vertexes: $M$ and $P$. Then link $O_{m+4}$ to every vertex to get a line, respectively. Then there are two points of intersection between these two lines and circle $V_{VCC}(U_{m+1}, U_{m+4})$.
3. Draw the tangent of circle $V_{VCC}(U_{m+1}, U_{m+4})$ through two points of intersection in (2), respectively.
4. The tangent in (3) cuts $U_{m+1}$'s present PLA into two parts, respectively. Then put every tangent as a dividing line to eliminate the region of $U_{m+1}$'s present PLA which is away from the circle $V_{VCC}(U_{m+1}, U_{m+4})$. The remaining part (convex polygon FIOPMQR) is the new PLA of $U_{m+1}$.

Step 7. Unknown node $U_{m+1}$ traverses all its unknown neighbors in the ascending order of hop-count which is from
its unknown neighbors to itself. Here, hop-count is bigger than 1.

**Step 7.1.** Unknown node $U_{m+1}$ uses the $V_{ICC}(U_{m+3})$ of unknown node $U_{m+3}$, as shown in Figure 10.

1. Draw $U_{m+1}$’s present PLC and the circle $V_{ICC}(U_{m+3})$.
2. There are two points of intersection between $U_{m+1}$’s present PLC and the circle $V_{ICC}(U_{m+3})$. Link these two points to get a line.
3. The line in (2) cuts $U_{m+1}$’s present PLA into two parts. Then put the line as a dividing line to eliminate the region of $U_{m+1}$’s present PLA closing to the circle $V_{ICC}(U_{m+3})$. The remaining part (convex polygon $FVWTUQR$) is the new PLA of $U_{m+1}$.

**Step 7.2.** Unknown node $U_{m+1}$ uses the $V_{ICC}(U_{m+4})$ of unknown node $U_{m+4}$, as shown in Figure 11.

1. Draw $U_{m+1}$’s present PLC and the circle $V_{ICC}(U_{m+4})$.
2. There are two points of intersection between $U_{m+1}$’s present PLC and the circle $V_{ICC}(U_{m+4})$. Link these two points to get a line.
3. The line in (2) cuts $U_{m+1}$’s present PLA into two parts. Then put the line as a dividing line to eliminate the region of $U_{m+1}$’s present PLA closing to the circle $V_{ICC}(U_{m+4})$. The remaining part (convex polygon $WTUQRXY$) is the new PLA of $U_{m+1}$.

**Step 8.** Unknown node $U_{m+1}$ executes Step 2 to Step 7 iteratively until its present PLA does not change. Unknown node $U_{m+1}$ can get a final PLA. Then $U_{m+1}$ computes the gravity center of its final PLA as its estimated position.

### 6. Simulation and Evaluation

This section describes the simulation parameters and model we use in our evaluation and then provides a detailed quantitative analysis by comparing the performance of our method with other range-free localization algorithms like DV-HOP [9], APIT [12], and SOM [14]. When evaluating localization schemes, the most important parameters of comparison are localization error, $\epsilon$, and $P$. We use several influence factors to simulate the experiments: (1) anchor node degree, (2) unknown node degree, (3) degree of irregularity (DOI), and (4) GPS error.

#### 6.1. Simulation Placement

Anchor nodes and unknown nodes are distributed in a $1000 \times 1000$ m$^2$ square in our simulations. There are 50 anchor nodes and 620 unknown nodes in the default setting. In order to simplify the computation, average radio radius of each node is assumed to be 100 m. Each simulation experiment tests 20 times to improve computational accuracy. Also, in order to simulate the realistic scenario, we take the degree of irregularity (DOI) and GPS error into consideration.

#### 6.2. Simulation Parameters

**6.2.1. Basic Parameters**

1. Anchor node degree (AD) is the average number of anchor nodes per unit circle.
2. Unknown node degree (UD) is the average number of unknown nodes per unit circle.
3. Degree of irregularity (DOI) is the maximum radio range variation per unit degree change in the direction of radio propagation. There are upper bound and lower bound of the communication radius. Consider
   \[
   DOI = \frac{R_{\text{max}} - R_{\text{min}}}{R_{\text{max}}} \times 100\%.
   \]

4. GPS error is the position error of the anchor node localized by GPS.

**6.2.2. Performance Parameters**

1. Localization error ($L_E$) is as follows:
   \[
   L_E = \frac{1}{n} \sum_{i=1}^{n} \frac{|Y_i - X_i|}{R} \times 100\%.
   \]
where $Y_i$ is the estimated coordinate of unknown node $(i)$ and $X_i$ is real coordinate of $i$. The number of unknown nodes is $n$. $R_i$ is $i$’s radio radius. In our simulation test, we not only calculate localization error of localized unknown nodes but also calculate average error of unlocated nodes, which is why the performance of APIT in our paper is different from paper [8].

(2) $r_{\text{MAX}}$ is the radius of possible location circle of unknown node.

(3) $P$ is the product of $r_{\text{MAX}}$ and the area of the final PLA of the unknown node. For some algorithms, even though the location area of unknown node is small, the localization error may be bigger. So, it is difficult to test the true performance of those algorithms by other parameters. But the performance of algorithm can be evaluated accurately by considering $r_{\text{MAX}}$ and the area of the final PLA.

All distances including error estimation are normalized to units of node radio range ($R$) to ensure the generally applicable results in our evaluation. When evaluating localization schemes, the most important measurements for comparison are localization error, $r_{\text{MAX}}$, and $P$. In our simulation, we take several influence factors into consideration: (1) anchor node...
6.3. Localization Error, $r_{\text{MAX}}$, and $P$ versus AD. We analyze the effect of varying the average numbers of anchor node per unit circle in experiment.

It is found that the localization error decreases as anchor nodes degree increase in Figure 12(a). The reason is that when AD increases, the numbers of anchor nodes increase in the wireless network, so the localization accuracy could be improved inevitably. APIT utilizes the anchor nodes to constitute the triangle. If the density of the anchor node is too low, the number of triangles is less, so the localization result is not so satisfied. NPLA algorithm performs better than APIT on this occasion, which is because NPLA not only uses the communication range of anchor nodes but also utilizes the PLA of the unknown node. As shown in Figure 12(b), the
6.4. Localization Error, $r_{\text{MAX}}$, and $P$ versus UD. We research the impact of UD on the precision of localization estimation and other performance in this experiment. The localization error of both APIT and NPLA decreases slightly as UD increases as shown in Figure 13(a), but the localization error of APIT is larger. Because APIT did not utilize unknown nodes to assist in reducing the PLA of unknown node in its design, both $r_{\text{MAX}}$ and $P$ of APIT are not be affected as shown in Figures 13(b) and 13(c). However, when UD increases, NPLA has more information of unknown neighbors to utilize. So the localization error of NPLA is smaller than others. The $r_{\text{MAX}}$ and $P$ of NPLA decrease as UD increases.

6.5. Localization Error, $r_{\text{MAX}}$, and $P$ versus GPS Error. In real environment, the error located by GPS of anchor nodes cannot be ignored; we take the GPS error into consideration in our simulation. The first step of NPLA algorithm is

$$r_{\text{MAX}}$$ of NPLA is lower, and it changes slightly when AD increases. The final localization area is a prolate polygon due to the algorithm feature in APIT, so $r_{\text{MAX}}$ and $P$ of APIT are bigger.
6.6 Localization Error, $r_{\text{MAX}}$, and $P$ versus DOI.

We also consider the effect of degree of irregularity (DOI) to the performance of the algorithm.

According to the definition of DOI, when DOI increases, the real communication range of node is more uncertain. So the localization error of algorithms which use communication range to locate unknown nodes such as DV-Hop and NPLA is affected obviously. The localization error, $r_{\text{MAX}}$, of NPLA increases as shown in Figure 14(b). But the area of final PLA in NPLA is still small, so the $P$ of NPLA keeps low. In APIT, when GPS error increases, the inaccuracy of nodes’ coordinate position will increase and then lead to the deviation of triangle in APIT. So the localization error, $r_{\text{MAX}}$, and $P$ are the biggest in all algorithms.

Figure 15: Varying DOI: anchor number = 50, unknown number = 0.1, and GPS error = 0.05.
and P in all algorithms increase as DOI increases as we can see in Figures 15(a), 15(b), and 15(c). NPLA utilizes hop-count information only to eliminate unknown node’s PLA rather than measure the distance. And it depends on not only hop-count information but also other information to locate unknown nodes, so the performance in NPLA is better than other algorithms.

6.7. Evaluation Discussion. From the comparison above, NPLA performs better than others. To reduce the computational complexity, NPLA, which is a distributed algorithm, just uses unknown node’s neighbors within three-hop. And NPLA keeps the PLA of an unknown node as a convex polygon in every step to further reduce the computation overhead. In general, the energy consumption of NPLA is less than most of other algorithms.

7. Conclusion

In this paper, we proposed a novel eliminating method NPLA for unknown nodes, which narrows the possible location area (PLA) of unknown node by utilizing the VCCs and ICCs of its anchor neighbors and unknown nodes. Even though there are only few anchor nodes in the wireless network, NPLA can still get better localization accuracy. Taking GPS error and DOI into account, the localization error is still acceptable.

Abbreviations

\( \Omega \): The whole network area
\( A_i \): The anchor node, \( 1 \leq i \leq m \), represents node whose position is known already
\( U_j \): The unknown node, \( m + 1 \leq j \leq n \), represents node whose position is unknown
\( R \): The standard communication radius of the node
\( e_g \): The relative error of anchor’s coordinate position measured by GPS
\( Re_g \): The absolute error of anchor’s coordinate position measured by GPS
\( R_{\text{max}} \): The maximum communication radius of the node \( k \) \( 1 \leq k \leq n \)
\( R_{\text{min}} \): The minimal communication radius of the node \( k \) \( 1 \leq k \leq n \)
\( O_i \): The coordinate position of anchor node measured by GPS \( k \) \( 1 \leq i \leq m \)
\( V_{\text{PLA}}(A_k) \): The possible location area of anchor node: center is \( O_i \) and radius is \( Re_g \)
\( V_{\text{PLA}}(U_k) \): The possible location area of unknown node: a convex polygon
\( V_{\text{PLC}}(U_j) \): The possible location circle of unknown node: the minimum circle that contains all vertexes of \( V_{\text{PLA}}(U_k) \) of the unknown node (the thin circle)
\( O_j \): The center of \( V_{\text{PLC}}(U_j) k \) \( m + 1 \leq j \leq n \)
\( U_j \): The radius of possible location circle of unknown node
\( R_{\text{MAX}} \): The radius of possible communication circle of anchor node:
\( R_{\text{A}k} \): The radius of possible communication circle of unknown node:
\( R_{\text{VCC}} \): The radius of possible communication circle of unknown node:
\( R_{\text{ICC}} \): The radius of inevitable communication circle of anchor node:
\( R_{\text{ICC}} \): The radius of inevitable communication circle of unknown node:
\( V_{\text{PCC}}(A_k) \): The possible communication circle of anchor node: the center is \( O_i \) and the radius is \( R_{\text{PCC}} \)
\( V_{\text{PCC}}(U_k) \): The possible communication circle of unknown node: the center is \( O_i \) and the radius is \( R_{\text{PCC}} \)
\( V_{\text{VCC}}(A_k) \): The inevitable communication circle of anchor node: the center is \( O_i \) and the radius is \( R_{\text{VCC}} \)
\( V_{\text{VCC}}(U_k) \): The inevitable communication circle of unknown node: the center is \( O_i \) and the radius is \( R_{\text{VCC}} \)
\( V_{\text{ICC}}(d,s) \): The virtual communication circle of node \( s \) relative to node \( d \) with the least hops, except node \( s \) and node \( d \). Here, \( d \) is unknown node and \( s \) is anchor or unknown node
\( V_{\text{ICC}}(d,s) \): The radius of the virtual communication circle of node \( s \) relative to node \( d \). Consider \( R_{\text{ICC}}^{(d,s)} = R_{\text{PCC}} + R_{\text{ICC}}^{(m+1\leq k \leq n)} \)
\( V_{\text{VCC}}(d,s) \): The virtual communication circle of node \( s \) relative to node \( d \); the radius is \( R_{\text{VCC}}^{(m+1\leq k \leq n)} \)

Conflict of Interests

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

Acknowledgments

This research is supported partly by China NSF 61373091, 61202478, 6093301, 11102124, and 71102113 and the US NSF Grant CNS-0917097.

References

[1] X. Cheng, H. Shu, Q. Liang, and D. H.-C. Du, “Silent positioning in underwater acoustic sensor networks,” IEEE Transactions on Vehicular Technology, vol. 57, no. 3, pp. 1756–1766, 2008.
[2] X. Cheng, A. Thaeler, G. Xue, and D. Chen, “TPS: a time-based positioning scheme for outdoor wireless sensor networks,” in Proceedings of the 23rd Annual Joint Conference of the IEEE Computer and Communications Societies, pp. 2685–2696, Hong Kong, March 2004.

[3] G. J. Pottie and W. J. Kaiser, “Wireless integrated network sensors,” Communications of the ACM, vol. 43, no. 5, pp. 51–58, 2000.

[4] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan, “The cricket location-support system,” in Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (MobiCom ‘00), pp. 32–43, August 2000.

[5] A. Savvides, C.-C. Han, and M. B. Strivastava, “Dynamic fine-grained localization in ad-hoc networks of sensors,” in Proceedings of the 7th Annual International Conference on Mobile Computing and Networking (MobiCom ’01), pp. 166–179, July 2001.

[6] J. Hightower and G. Borriello, “A survey and taxonomy of location systems for ubiquitous computing,” IEEE Computer, vol. 34, no. 8, pp. 57–66, 2001.

[7] Y. Geng, J. He, H. Deng, and K. Pahlavan, “Modeling the effect of human body on TOA ranging for indoor human tracking with wrist mounted sensor,” in Proceedings of the 16th IEEE International Symposium on Wireless Personal Multimedia Communications (WPMC ’13), pp. 1–6, Atlantic City, NJ, USA, June 2013.

[8] Y. Geng, J. Chen, and K. Pahlavan, “Motion detection using RF signals for the first responder in emergency operations: a PHASER project,” in Proceedings of the IEEE 24th International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC ’13), pp. 358–364, London, UK, September 2013.

[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] R. Nagpal, H. Shrobe, and J. Bachrach, “Organizing a global coordinate system from local information on an ad hoc sensor network,” in Information Processing in Sensor Networks: Proceedings of the 2nd International Workshop, IPSN 2003, Palo Alto, CA, USA, April 22-23, 2003, vol. 2634 of Lecture Notes in Computer Science, pp. 333–348, Springer, Berlin, Germany, 2003.

[11] S. N. Simic and S. Sastry, “Distributed localization in wireless ad hoc networks,” Tech. Rep., 2001.

[12] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. Abdelzaher, “Range-free localization schemes for large scale sensor networks,” in Proceedings of the 9th Annual International Conference on Mobile Computing and Networking (MobiCom ’03), pp. 81–95, San Diego, Calif, USA, September 2003.

[13] J.-P. Sheu, P.-C. Chen, and C.-S. Hsu, “A distributed localization scheme for wireless sensor networks with improved grid-scan and vector-based refinement,” IEEE Transactions on Mobile Computing, vol. 7, no. 9, pp. 110–123, 2008.

[14] G. Giorgetti, S. K. S. Gupta, and G. Manes, “Wireless localization using self-organizing maps,” in Proceedings of the 6th International Conference on Information Processing in Sensor Networks (IPSN ‘07), pp. 293–302, Cambridge, Mass, USA, April 2007.

[15] L. Chen, Y. Gu, S. Guo et al., “Group-based discovery in low-duty-cycle mobile sensor networks,” in Proceedings of the 9th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON ’12), pp. 542–550, Seoul, Republic of Korea, June 2012.
Submit your manuscripts at
http://www.hindawi.com