

A Component-Based Localization Algorithm for Sparse Sensor Networks Combining Angle and Distance Information

Shigeng Zhang, Shuping Yan, Weitao Hu, Jianxin Wang, and Kehua Guo

School of Information Science and Engineering, Central South University

Changsha, 410083, China

[e-mail: sgzhang@csu.edu.cn]

*Corresponding author: Kehua Guo

*Received August 8, 2014; revised November 14, 2014; revised January 4, 2015; accepted January 27, 2015;
published March 31, 2015*

Abstract

Location information of sensor nodes plays a critical role in many wireless sensor network (WSN) applications and protocols. Although many localization algorithms have been proposed in recent years, they usually target at dense networks and perform poorly in sparse networks. In this paper, we propose two component-based localization algorithms that can localize many more nodes in sparse networks than the state-of-the-art solution. We first develop the *Basic Common nodes-based Localization Algorithm*, namely BCLA, which uses both common nodes and measured distances between adjacent components to merge components. BCLA outperforms CALL, the state-of-the-art component-based localization algorithm that uses only distance measurements to merge components. In order to further improve the performance of BCLA, we further exploit the angular information among nodes to merge components, and propose the *Component-based Localization with Angle and Distance information* algorithm, namely CLAD. We prove the merging conditions for BCLA and CLAD, and evaluate their performance through extensive simulations. Simulation results show that, CLAD can locate more than 90 percent of nodes in a sparse network with average node degree 7.5, while CALL can locate only 78 percent of nodes in the same scenario.

Keywords: Wireless sensor networks, component-based localization, range-based localization, sparse sensor networks, component mergence

A preliminary version of this paper appeared in IEEE EUC 2013, November 13-15, Zhangjiajie, China. This version includes an in-depth analysis on the intrinsic reason why the proposed algorithm outperforms state-of-the-art solutions and three sets of new experimental results that demonstrate the superior performance of the proposed algorithms. This research was supported by the National Science Foundation of China under Grant Nos. 61103203, 61202494, 6140060443 and 61202341.

1. Introduction

Wireless sensor networks (WSNs) are composed of a large number of randomly deployed sensor nodes in an ad hoc manner. Because WSNs can be easily deployed in hostile environments without pre-deployed infrastructures, they were widely applied in many fields in recent years, e.g., military surveillance [1], medical care [2], wildlife monitoring applications [3], environment monitoring [4, 5], and rescue after disasters [6]. As a key part of the future Internet of Things that can provide ubiquitous sensing ability, WSNs have attracted more and more research attention in recent years.

Location information of sensor nodes plays a vital role in many WSN applications and protocols. For example, in many WSN applications, the collected data should be labeled with the positions where they are collected. Otherwise, these data would be meaningless because they cannot provide useful information to users. Location information of sensor nodes is also critical in protocols and algorithms designed for WSNs, e.g., coverage optimization [7-9], target tracking [10, 11] and geographical routing [12, 13].

In order to obtain locations of sensor nodes, many localization algorithms have been proposed in recent years [14-17]. In a localization algorithm, there are usually some special nodes called *anchors* that know their positions in advance (e.g., via GPS). The other nodes are called *unknown nodes* which need to calculate their positions with the position information of anchors and distance/angular measurements between adjacent nodes. Generally, if an unknown node can obtain enough distance measurements to anchors (e.g., more than three anchors in 2-dimensional space or more than four anchors in 3-dimensional space), it can uniquely determine its position. A node that can uniquely determine its position is called a *localizable node*. A network in which all the nodes are localizable is called a localizable network. Whether a network is localizable or not is determined by the global topology of the network [18, 19].

It is challenging to localize all the nodes in the network even when the network is localizable, especially when the network is sparse, e.g., the average node degree is smaller than 10 [20-22]. Traditional localization algorithms usually localize individual sensors one by one. We call such algorithms as *sequence localization algorithms*. Sequence localization algorithms can localize only a subset of localizable networks, e.g., wheel networks [20] and trilateration-ordering networks [23]. The main difficulty in localizing sparse networks is that it is difficult to find a proper ordering of nodes such that sequence localization can be performed for one node after another. Moreover, there might be no such an ordering in sparse networks, even when they are localizable. In order to overcome these limitations, component-based localization algorithms were proposed in [21, 22]. In component-based localization algorithms, sensor nodes are grouped into different components. Each component is a sub-network that can be localized with sequence localization algorithms. Every component constructs its local coordinate system. The local coordinate systems of different components are then combined to form the global coordinate system. However, in existing component-based algorithms [21, 22], components are not overlap with each other, and thus they can use only distance measurements between adjacent components to perform component combination, which limits the total number of nodes that can be localized when the network is very sparse.

In this paper, we propose two component-based localization algorithms that can localize many more nodes in sparse networks than the state-of-the-art component-based localization algorithm CALL [21, 22]. We first develop the Basic Common nodes-based Localization Algorithm, namely BCLA, which uses both common nodes and distance measurements to

merge adjacent components. BCLA thus can localize more nodes than CALL, which uses only distance measurements to merge adjacent components. Considering that angular information among nodes can be obtained in many sensor networks [14, 15], we further develop CLAD, the Component-based Localization algorithm with Angle and Distance information. By utilizing angular information among nodes, CLAD can localize more nodes than BCLA and CALL do. Simulation results show that, in sparse networks with average node degree 7.5 and 5 percent anchor ratio, CLAD successfully localizes more than 90 percent of all the nodes in the network, while CALL localizes only 78 percent of nodes in the same scenario.

The rest of this paper is organized as follows. Section 2 reviews related work. In Section 3, the conditions for combining two components with both common nodes and distance measurements are derived, and the detailed design of the BCLA algorithm is presented. Section 4 gives the design and analyses of the CLAD algorithm that exploits angular measurements among nodes when combining components. Simulation results are reported and discussed in Section 5. Finally, conclusion remarks are given in Section 6.

2. Related Work

The localizability theory of sensor networks has attracted a lot of research attention in the past several years [18-21, 26]. A sensor network is uniquely localizable in 2D space if and only if its topology graph is globally rigid and it contains more than three anchors [18]. When the network is not uniquely localizable as a whole, we should consider the conditions under which its sub-networks or separate nodes can be uniquely localizable. In [19, 24] the authors gave conditions for nodes in a sub-network to be uniquely localizable: 1) The topology of the sub-network is redundant rigid; 2) the sub-network is tri-connected; and 3) the sub-network has at least 3 anchors. These conditions are abbreviated as RRT-3B [24]. RRT-3B is a sufficient condition for a node to be uniquely localizable, and it can be used to find the maximum number of uniquely localizable nodes in a network.

It remains challenging to localize all the uniquely localizable nodes in the network even when the network is uniquely localizable, especially when the network is very sparse [20]. The difficulty is that it is difficult to find a trilateration ordering [23] of nodes in which they can be sequentially localized one after another. In order to overcome this problem, in [20] the authors introduced the concept of finitely localizable nodes, which has finite possible positions. They then proposed the Sweeps algorithm, which records all the possible positions for every node and gradually resolves the ambiguity by adding more distance constraints. Sweeps relaxes the requirement on trilateration ordering of nodes, and thus can localize more nodes than sequence trilateration-based algorithms. However, Sweeps requires that nodes can be ordered in bilateration ordering and thus cannot localize all the localizable nodes in many cases [20-22]. Moreover, Sweeps requires that the nodes can form a cycle in order to solve the ambiguity caused by multiple possible positions of a node, and thus has exponentially time complexity in the worst case.

In order to mitigate the effects of node ordering on the localization coverage, in [22] the authors proposed a component-based localization algorithm called CALL. Different from Sweeps and other sequence localization algorithms, CALL groups nodes into components and performs localization on the component level. It first constructs local coordinate system for every component, and then merges local coordinate systems to form a global coordinate system. Because CALL does not require special ordering of nodes, it can localize more nodes than Sweeps does in sparse networks. However, CALL uses only distance measurements to merge adjacent components, which greatly limit the number of nodes it can localize.

Component-based localization algorithms might be sensitive to distance measurement errors, as pointed out in [27]. To guarantee the robustness of component-based localization algorithms and obtain accurate localization results when the distance measurements contain errors, in [27] the authors proposed the Error-Tolerant Component-based algorithm (ETOC). The core idea of ETOC is to realize a component only when the width of the anchors in the component is large than a threshold. This can help mitigate the flip ambiguities when the anchors are nearly collinear. We note that ETOC uses the similar idea as proposed in this paper but for different purposes. In ETOC, distance measurements between components are used to enhance the robustness when realizing components, while in our algorithms the distances between components are used to help merge adjacent components. Moreover, in this paper we also exploit angular information to help component mergence. On the other hand, our algorithms also face the robustness issues mentioned in [27]. We can enhance the robustness of our algorithms by using the technique proposed in ETOC.

Localization for 3D Surface sensor networks has attracted a lot of attention in recent years [28-30]. In [28] the authors discussed the challenges in 3D surface localization and proposed a solution to single value 3D surface networks, i.e., the networks in which any two nodes have different projects on the x-y plane. They also proposed a layered approach to localizing nodes in general 3D surface networks. This approach is extended in [29] to divide a general 3D surface network into a minimal set of single value surface networks. The partitioned single value surface networks are then individually localized and combined to form a global coordinate system. However, the aforementioned approaches both assume that the height information of sensor nodes is known in advance. To mitigate this requirement, in [30] the authors exploited digital terrain model (DTM) of 3D surfaces to help localization in 3D surface networks. These solutions all require dense deployment of sensor nodes.

In [31] the authors proposed a localization protocol based on approximate convex decomposition, namely ACDL, to localize nodes deployed in concave networks. ACDL first divides the concave network into a set of convex sub-networks, and applies multi-dimensional scaling (MDS) technique in each sub-network to obtain a local coordinate system. The local coordinate systems of sub-networks are then combined to form a global coordinate system. Different from traditional convex relaxation approaches, in [32] the authors proposed a non-convex optimization approach to solve the localization problem, which can achieve similar (or even better) localization accuracy as traditional convex optimization based approaches but with lower time complexity. In [33], the authors discussed how to localize a group of autonomous underwater vehicles (AUVs) in order to guarantee that all the AUVs can arrive at their destinations with their locations being traceable during the whole mission. In [34], the authors use the time of charge sequences among different sensors to localize nodes in rechargeable sensor networks. Their solutions are sequence localization algorithms, and they require dense deployment of sensors. Different from them, the algorithms proposed in this paper are component-based localization algorithms, which are more suitable to sparse networks.

3. BCLA: The Basic Common Nodes-Based Localization Algorithm

In this section, we first derive the conditions under which two components can be merged and then give the detailed design of the BCLA algorithm.

3.1 Component Merging Conditions

The key issue in component-based localization algorithms is to merge adjacent components. Generally speaking, a component is a group of nodes that forms a uniquely localizable sub-network. After components are generated, each component constructs its local coordinate system. Then all the local coordinate systems are merged into one global coordinate system. If there are more than three anchors in the network, the global coordinate system can be translated to the absolute coordinate system determined by the anchors. In the traditional coordinate system registration methods [25], two components have to share at least three common nodes in order to register nodes in one coordinate system into another. However, when the network is very sparse, it is rare to have enough common nodes between two adjacent components. In the CALL algorithm [21, 22], the authors proved that two components can be uniquely merged with four distance measurements between nodes in different components. The CALL algorithm can merge components when there is no common node and when there are at least three common nodes between two components. In this paper, we consider how to merge two components when there is exact one and exact two common nodes between two components to further improve the localization coverage.

In the following, we give the conditions of merging two components with both the common nodes and distance measurements between two components. We first prove the conditions on merging two components when they share only one common node, then give the merging conditions when two components share two common nodes.

3.1.1 When Two Components Have One Common Node

When two components have only one common node, we have the following theorem.

Theorem 1: Given two components A and B that are both uniquely localizable and have a common node: 1) B can be uniquely merged into A if there are two edges connecting A and B with distinct nodes; 2) B can be finitely merged into A if there is one edge connecting A and B . Here unique mergence means that nodes in B have unique positions after mergence, and finite mergence means that nodes in B have finite possible positions after mergence.

Proof: We first prove claim 1). As shown in Fig. 1(a), A and B are two uniquely localizable components sharing a common node b . The solid lines ad and ce indicate distance measurements between nodes in A and B . The dashed lines bd , be , and de indicate the implicit distances that can be calculated according to the coordinates of nodes in local coordinate system (note that b , d and e are all in component B). Because node b is a common node, we know its position in A 's coordinate system. If we know the positions of node d and e in A 's coordinate system, the component mergence problem could be reduced to the trivial case when two components share three common nodes. We now give a method to derive the positions of nodes d and e in the coordinate system of component A .

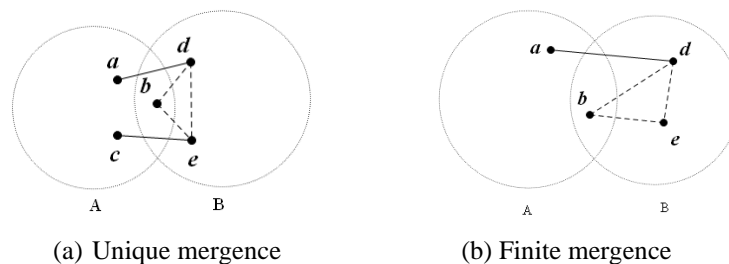


Fig. 1. Merging conditions when two components share one common node

Denote the positions of nodes a , b and c in component A by (x_a, y_a) , (x_b, y_b) and (x_c, y_c) , respectively. Note that these positions are all known because the three nodes are all in component A . We use (x_1, y_1) and (x_2, y_2) to denote the positions of nodes d and e in component A . Note that these positions are unknown and we need to calculate them. As shown in **Fig. 1(a)**, we can obtain five equations as below:

$$\begin{cases} (x_a - x_1)^2 + (y_a - y_1)^2 = d_{ad}^2 \\ (x_b - x_1)^2 + (y_b - y_1)^2 = d_{bd}^2 \\ (x_c - x_2)^2 + (y_c - y_2)^2 = d_{ce}^2 \\ (x_b - x_2)^2 + (y_b - y_2)^2 = d_{be}^2 \\ (x_1 - x_2)^2 + (y_1 - y_2)^2 = d_{de}^2 \end{cases} \quad (1)$$

Note that because nodes b , d and e are in the same component, the distances d_{bd} , d_{be} and d_{de} can be calculated according to their positions in component B . The distances d_{ad} and d_{ce} are measured distances.

There are five equations containing four unknown variables in (1), and thus we can determine the unique values of (x_1, y_1) and (x_2, y_2) . After we obtain the positions of nodes d and e in component A , we can uniquely merge component B into component A because there are three nodes that know their positions in both components. Thus claim 1) holds.

We now prove claim 2). As shown in **Fig. 1(b)**, A and B are two uniquely localizable components. Because node d knows its distance to two nodes in component A (i.e., nodes a and b), it has only two possible coordinates in component A . Because node e knows its distance to node b and d , for each possible position of node d , we can calculate two possible positions of node e . For each combination of the coordinates of node d and node e , we can merge component B into component A . Because there are finite combinations of the coordinates of node d and node e , component B can be finitely merged into component A . Note that although node d have two possible positions and node e have four possible positions, there are only two correct combinations of their coordinates because the distance between node d and node e must be equal to d_{de} .

When components A and B are only finitely localizable, we have the following corollary:

Corollary 1: Given two components A and B that have a common node and are both finitely localizable, they can be finitely merged if there is one edge linking two distinct nodes (both cannot be the common node) in the two components.

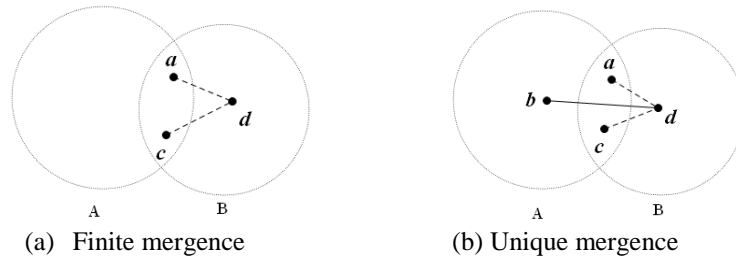


Fig. 2. Merging conditions when two components share two common nodes

3.1.2 When Two Components Have Two Common Nodes

When two components share two common nodes, we have the following theorem:

Theorem 2: Given two components A and B that are both uniquely localizable and have two common nodes: 1) B can be finitely merged into A ; 2) B can be uniquely merged into A if there is one edge connecting A and B .

Proof: We first prove claim 1). As shown in Fig. 2(a), A and B have two common nodes. Thus d has two possible positions in component A . It is easy to see that nodes in component B can be finitely merged into component A by using the finite combinations of the coordinates of a , c and d .

We now prove claim 2). As shown in Fig. 2(b), the solid line bd indicates the measured distance between b and d . The dashed lines ad and cd indicate the implicit distances that can be calculated according to the coordinates of nodes. Because a and c are common nodes, their positions in both components are known. Thus, if we know the position of d in A 's coordinate system, we can uniquely merge component B with component A .

Denote the positions of a , b and c in the coordinate system of component A by (x_a, y_a) , (x_b, y_b) and (x_c, y_c) , respectively. Note that these positions are known because they are all in component A . We use (x_1, y_1) to denote the position of d in component A . As shown in Fig. 2(a), we can obtain three equations as below:

$$\begin{cases} (x_a - x_1)^2 + (y_a - y_1)^2 = d_{ad}^2 \\ (x_b - x_1)^2 + (y_b - y_1)^2 = d_{bd}^2 \\ (x_c - x_1)^2 + (y_c - y_1)^2 = d_{cd}^2 \end{cases} \quad (2)$$

With the three equations in (2), we can uniquely determine the values of x_1 and y_1 , i.e., we can calculate the position of d in component A , and consequently uniquely merge component B into component A . ■

We need to point out that node d cannot be grouped into component A , because it has only one distance measurements to nodes in component A . The distances d_{ad} , d_{ac} are not measured distances; they are implicit distances that are calculated according to the coordinates of nodes. For example, d may be far away from a and c and thus the distance between them cannot be directly measured. However, the distances between them can be calculated according to their coordinates, because they are in the same component. This is different from the traditional trilateration localization method that requires three directly measured distances.

Algorithm 1: Framework of BCLA

Input: The ranging graph G representing the network

Output: Coordinate of nodes of G

Phase 1: Component generation: Group nodes into components and separate nodes (Section 3.2.1)

Phase 2: Local coordinate system construction: Construct local coordinate system for each component (Section 3.2.2)

Phase 3: Component combination: Merge adjacent components until all the components are merged into one component, and establish the global coordinate system (Section 3.2.3)

Phase 4: (Optional) Translate the global coordinate system into absolute coordinate system if there are enough anchors in the network (Section 3.2.4)

Fig. 3. The framework of the BCLA algorithm

3.2 Design of BCLA

BCLA consists of four phases: component generation, local coordinate system construction, component combination, and an optional phase that is used to translate positions of nodes to absolute positions when there are enough anchors in the network. Fig. 3 gives the framework of BCLA. In the following, we describe each phase in detail.

3.2.1 Component Generation

The component generation phase consists of three stages. In the first stage, all the nodes are initialized as *waiting-for-grouping* state. In the second stage, if there are three nodes that can form a triangle (i.e., they are connected to each other) and at least one of them is in the waiting-for-grouping state, then the triangle is used as a seed to form a component. If a node has more than three edges linked to nodes in a component, it is added into that component. The component is extended until no more nodes can be added into it. After a component is formed, we mark all the nodes in that component as *grouped*. The second stage is recursively executed until no further components could be formed. At last, in the third stage, all the nodes that are not grouped into any components are marked as *isolated* nodes.

When finding a triangle to form a component in the second stage, we require that at least one node of the triangle is in the waiting-for-grouping state. With this restriction, the number of generated components can be controlled. Otherwise, the number of generated components may exponentially increase along with the number of nodes in the network. Meanwhile, the components generated with this approach can overlap with each other, and thus provide more opportunities to localize more nodes by using both common nodes and distances in component merging. In contrast, in CALL [21] the components do not overlap with each other, and thus rely on only distances between adjacent components to perform merging, which limits the number of nodes that can be localized.

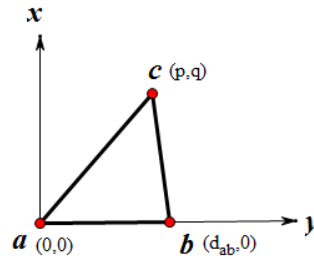


Fig. 4. Local coordinate system construction

3.2.2 Local Coordinate System Construction

After components are generated, we construct local coordinate systems for every component. As same as in CALL [21], we use the three nodes that form the triangle to establish the coordinate system. Denote the three nodes by a , b , c , respectively. As shown in Fig. 4, we put a at the original point $(0, 0)$, put b on the positive x -axis with coordinate $(d_{ab}, 0)$, and put c in the first quadrant of the coordinate system and denote its coordinate as (p, q) . According to the measured distances between the three nodes, we have the following equations:

$$\begin{cases} p^2 + q^2 = d_{ac}^2 \\ (d_{ab} - p)^2 + q^2 = d_{bc}^2 \end{cases} \quad (3)$$

where d_{bc} and d_{ac} denote the distances between b and c and between a and c , respectively. Note that because c is in the first quadrant, q is positive, i.e., $q > 0$. We can calculate the values of p and q according to equations in (3) as:

$$\begin{cases} p = \frac{d_{ab}^2 + d_{ac}^2 - d_{bc}^2}{2 \times d_{ab}} \\ q = \sqrt{d_{ac}^2 - \left(\frac{d_{ab}^2 + d_{ac}^2 - d_{bc}^2}{2 \times d_{ab}}\right)^2} \end{cases} \quad (4)$$

After the local coordinate system is established, the positions of other nodes could be calculated by using the trilateration method.

3.2.3 Component Combination

In this phase, local coordinate systems of individual components are merged into a global coordinate system. We select the component that has the largest number of nodes as the starting component, and merge other components and isolated nodes into this component. There are two cases to be considered:

Combination of components: When two components satisfy the merging conditions given in Theorem 1 and Theorem 2, we combine them into one component. Denote the component containing more nodes as component A , and denote the component containing fewer nodes as component B . We then merge component B into component A . If component B can be uniquely merged into component A , we assign a unique position for every node in B . If the component can only be finitely merged into component A , we record all the possible positions of nodes in component B . We will introduce a position pruning method to reduce the number of possible positions of nodes in B in Section 3.3.

Combination of isolated nodes: Isolated nodes can only be finitely merged into a component. When an isolated node has two edges associated to two distinct nodes in a component, we merge the node into the component, and record all its possible positions. Similarly, we will prune impossible positions of isolated nodes by using the method introduced in Section 3.3.

3.2.4 Translation to Absolute Coordinate System

This is an optional phase. When there are enough anchors in the final merged component, we can register the component in the absolute coordinate system determined by the anchors. In order to do this, we first form a new component that contains only anchor nodes, and then register the merged component into this component.

3.3 Illogical Position Pruning

When components are finitely merged, BCLA needs to record all the possible positions of a node in order to increase the chances of merging more components. However, in the worst case, the number of possible positions of nodes might exponentially increase with the number of nodes in the component. This results in high time and space overhead. In order to reduce the time and space complexity during the merging process, the illogical node positions should be removed as early as possible. In [22], the authors proposed a ranging-model-based estimation (RMBE) method to evaluate the confidence of a merged or realized component, which could be used to cut candidate results whose confidence values are below a threshold. We adopt this method to further prune illogical positions in the results obtained by BCLA.

The basic idea is to infer whether two nodes should obtain distance measurement between them or not, according to their possible position combinations. The combination is marked as illogical if the inferred result contradicts with the actually observed result. Due to the limited distance measuring capability of nodes, two nodes can measure their distance only when the distance is smaller than a threshold. Thus, for a possible position combination of two nodes, if the distance calculated according to the nodes' position is actually larger than the threshold, the position combination is marked as illogical. Otherwise, if the distance calculated according to the position combination is far smaller than the threshold but actually not distance measurement between is obtained, then the position combination is also marked as illogical. For a possible position p of a node, if all the position combinations containing p are illogical, we remove p from the node's possible position set. After each component merging, we execute this position pruning process to remove illogical positions of all the nodes in the component to reduce the time complexity.

Fig. 5 shows an example of illogical position pruning. In this example, we assume that two nodes can measure their distance if the distance is smaller than a threshold r . Node a obtains two distance measurements to nodes b and c , respectively. Thus it has two possible positions, i.e., p and p' . If a locates at p' , it should obtain the distance measurements to d and e , i.e., $d_{a'e}$ and $d_{a'd}$. However, actually these two distance measurements are not obtained. Thus p' is an illogical position of a and should be pruned. With this illogical position pruning method, a can be uniquely localized.

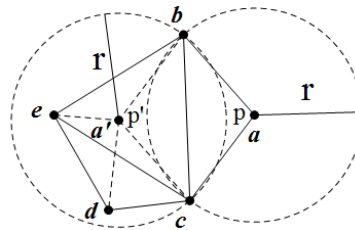


Fig. 5. Illogical position pruning

3.4 Comparison with CALL

In this section, we investigate the intrinsic reason why BCLA outperforms CALL [21,22]. In CALL, components do not overlap with each other, and thus they can use only distance measurements to merge components. Due to this restriction, CALL may neglect some cases in which two components could be merged in BCLA but could not be merged in CALL. For example, even in some cases two components could be merged with common nodes and distance measurements, CALL fails to merge them because the two components cannot be merged with only distance measurements. Thus BCLA localizes more nodes than CALL.

We use **Fig. 5** to illustrate an example. As we have pointed out in the previous section, in BCLA the two components, namely component abc and component cde can be finitely merged because they share a common node and there is an edge between them. After applying the illogical position pruning, the two components can be uniquely merged. However, the two components cannot be uniquely merged in CALL. If we use abc to form a component, then node d and node e will become isolated nodes because they cannot form a component. Similarly, if use cde to form a component, then node a and node b will become isolated nodes. In both cases the two nodes can only be finitely merged into the component.

Fig. 6 plots the rate of such special cases when the average node degree changes from 5 to 10. We can observe that when the network is very sparse, this special case happens more often than when the network is dense. When the average node degree is 6, the ratio of this special case takes around 30 percent. This is the intrinsic reason why BCLA outperforms CALL in very sparse networks.

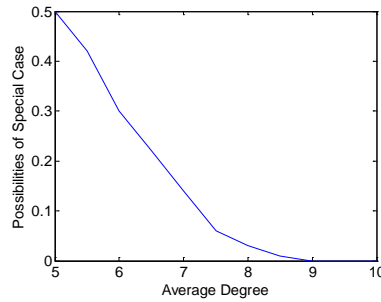


Fig. 6. The rate of special cases when the network degree varies

4. CLAD: Component-Based Localization with Angle and Distance Information

In this section, we discuss how to further improve the performance of BCLA when angular information among nodes can be obtained.

4.1 Inferring Distance between Nodes with Angular Measurement

The angular measurements can be used to infer distance between two nodes that are not neighbors. For example, as shown in **Fig. 7(a)**, originally the distance between d and f is unknown, and thus component A and component B cannot be uniquely merged because they don't satisfy the merging condition given in Theorem 1. However, if the angle α between db and bf is known, we can infer the distance between d and f according to the law of cosines:

$$d_{df}^2 = d_{bd}^2 + d_{bf}^2 - 2 \times d_{bd} \times d_{bf} \times \cos \alpha \quad (5)$$

After d_{df} is calculated, the two components can be uniquely merged. In the following, we give conditions on unique or finite merging of two components when angular information among nodes can be obtained.

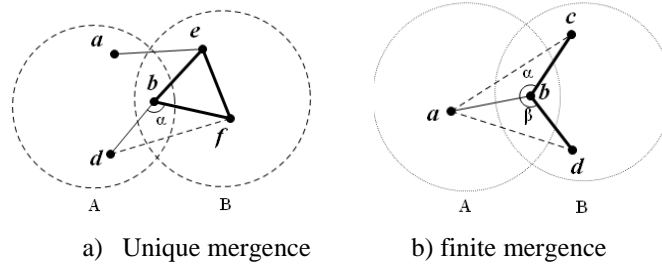


Fig. 7. Merging condition when two components have one common node and angle among nodes can be obtained

4.2 Component Merging Condition with Angular Information

When Two Components Have One Common Node: As shown in **Fig. 7(a)**, there is only one distance measurements d_{ae} between component *A* and component *B*. According to the merge conditions of BCLA, the two components *A* and *B* cannot be uniquely merged. However, if we know the angle α , the distance between *d* and *f* can be inferred, because distance d_{bd} and d_{bf} can be calculated according to their coordinates in component *A* and component *B*, respectively. In this case, *A* and *B* can be uniquely merged.

Similarly, as shown in **Fig. 7(b)**, when either of the two angles α and β is known, the two components *A* and *B* can be finitely merged even when there are no distance measurements between them. This is because d_{ac} (or d_{ad}) can be calculated by using α (β) and d_{ab} and d_{bc} (or d_{bd}), which can all be calculated according to their coordinates in the two components.

When Two Components Have Two Common Nodes: Figure 8 shows the merging condition of two components when they have two common nodes. As there is no edge between component *A* and component *B*, they can be only finitely merged in BCLA. However, if we know α or β , we can calculate the distance d_{ce} or d_{cd} , with which the two components can be uniquely merged.

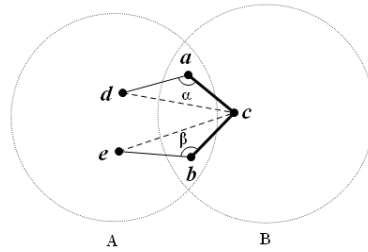


Fig. 8. Merging condition when two components have two common nodes and angle among nodes can be obtained

4.3 Design of CLAD

CLAD differs from BCLA only in the component merge phase. Different from BCLA, when we merge two components in CLAD, we consider both distance measurements and angular measurements. Besides the merging conditions specified for BCLA in Section III, we use the following additional conditions when merging two components:

- If two components have two common nodes and one of the two angles (α and β) can be measured (**Fig. 8**), then the two components can be uniquely merged;
- If the two components have only one common node and there is one edge ae (or df) between them, then when alpha α can be measured (**Fig. 7(a)**), the two components can be uniquely merged;
- If two components have only one common node and there is no edge between them, then when α or β can be measured (**Fig. 7(b)**), the two components can be finitely merged.

5. Simulation Results

5.1 Experiment Setup

We use two metrics to evaluate the performance of the proposed algorithm. The first metric is *localization ratio*, which is defined as the ratio of uniquely localized nodes to all the nodes in the network. The second metric is *localization accuracy*, which is defined as the average localization error over all the localized nodes in the network. A node's localization error is defined as the Euclidean distance between its estimated position and its ground-truth position. When a node is not uniquely localized, we define its localization error as the maximum distance between the node's ground-truth location and its candidate positions. To be consistent with existing researches [21], we measure the localization error as a multiple of the communication radius r .

The main parameters that may impact the performance of different algorithms include the average network degree, the distance measurement error, the angle measurement error, and the anchor ratio. We tune the average network degree by deploying different number of nodes in the network. The size of the deployment region is a $10r \times 10r$ square region, where r is the communication radius of sensor nodes. In the default setting, we deploy 200 nodes in the network. The nodes are uniformly deployed in the network, and this results in an average network degree of approximately 6.5. We change the scale of the network by changing the number of nodes deployed in the network to investigate its impact on the performance of the proposed algorithms. In default, we assume that the distance and angle measurements are accurate and the communication is ideal. To be consistent with previous studies [21,22], we set the default anchor ratio as to 5%. We randomly select 5% nodes and use them as anchors, i.e., every sensor node has a probability of 0.05 to be an anchor node. We investigate the impact of distance/angle measurement error on the localization accuracy in Section 5.5 and 5.6, respectively.

We compare BCLA and CLAD with the state-of-the-art component-based localization algorithm CALL [22]. The reported data here are averaged over 50 randomly generated network topologies.

5.2 Impact of Average Network Degree

We first evaluate how the average network degree affects the localization ratio of different algorithms. Fig. 9 plots the localization ratio of the three algorithms when the anchor ratio is 5%. It can be observed that BCLA and CLAD both localize more nodes than CALL. CLAD and BCLA perform much better than CALL when the network is extremely sparse. For example, when the average network degree is 5.5, CLAD can uniquely localize more than 50% nodes in the network, while CALL can localize only less than 30% nodes. The performance gaps between our algorithms and CALL become insignificant when the network becomes denser. When the average network degree is larger than 10, the localization ratios of the three algorithms are nearly the same. This indicates that our two algorithms are more suitable to sparse networks.

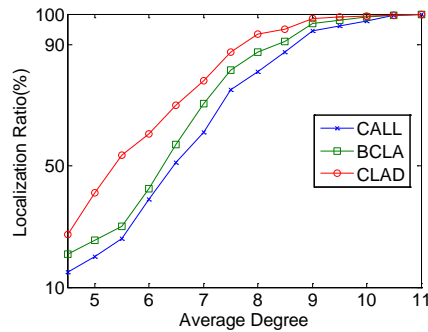


Fig. 9. Impact of average network degree on localization ratio of different algorithm

Fig. 10(a) and Fig. 10(b) plot the ratio of finitely localized nodes and the average number of candidate positions when the node is finitely localized in different algorithms. As shown in Fig. 10(a), the ratio of uniquely localized nodes in BCLA and CLAD is smaller than that in CALL. This is because in our algorithms, some illogical position candidates are pruned by using the distance constraints, as have been discussed in Section 3.3. The utilization of the angular information in CLAD further reduces the ratio of finite localized nodes by converting some finite localized nodes into unique localized nodes furthermore. It can be seen from Fig. 10(b) that the average possible position estimates in BCLA and CLAD is much less than that in CALL.

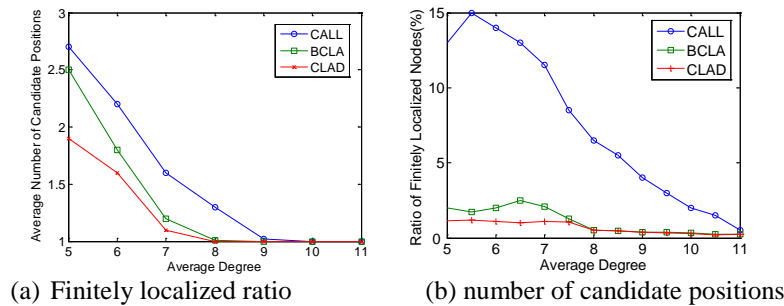


Fig. 10. Impact of average network degree on: (a) Finitely localized node ratio; (b) number of candidate positions

Fig. 11 shows the ratio of nodes in the largest component of CLAD and CALL. Because CLAD utilizes angle information to group nodes, it obviously can merge more nodes than CALL under different average network degree. We observe that the increasing average network degree has impact on the ratio of nodes in the maximum components. When the network reaches middle density (the average network degree is 9), the largest component can cover almost 80% nodes in CLAD and 70% nodes in CALL, respectively. This also shows that angular information could effectively improve the localization ratio.

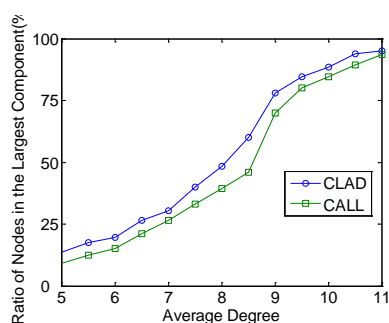


Fig. 11. The ratios of nodes in the largest component in CLAD and CALL

5.3 Impact of Anchor Ratio

The ratio of anchors impact the performance of our algorithm. **Fig. 12** shows the localization ratio of BCLA and CLAD when the anchor ratio changes for two network degree, i.e., when the average network degree is extremely low (6) and middle (9). We observe that when the anchor ratio increases, both algorithms can uniquely locate more nodes. When the average network degree is 9 (which means that the network is middle dense), BCLA and CLAD could uniquely localize almost 100% nodes even when the anchor ratio is as low as 5%. When the network is extremely sparse (the average network degree is 6), CLAD could localize more nodes than BCLA. This is because CLAD uses both distance constraint and angle information to perform component merging. In fact, in our simulations, when the average degree is greater than 9, the improvement of CLAD over BCLA is very insignificant. In such cases, the increase of anchor ratio does not affect the performance of BCLA and CLAD.

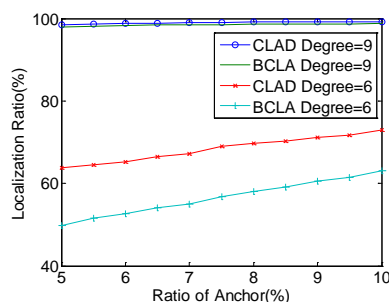


Fig. 12. The localization ratio of BCLA/CLAD when anchor ratio changes

5.4 Impact of Distance Measurement Error on Localization Accuracy

Fig. 13(a) plots localization accuracy of the three algorithms when relative distance measurement error changes from 0.01 to 0.1. For example, assume that the ground-truth distance between two nodes is d , then the measured distance falls into $[0.95d, 1.05d]$ if the relative distance estimation error is 0.05. Recall that in CLAD distances between some non-adjacent nodes are inferred by using the erroneous distance measurement and the angular information among nodes. When these inferred distances are used in localization, they might cause error accumulation problem. Thus the localization accuracy of CLAD is a bit lower than that of BCLA. In contrast, CALL uses only existing edges to reduce the potential positions of nodes, and thus for some nodes that BCLA and CLAD can uniquely localize it can only

finitely localize. This consequently makes CALL’s accuracy lower than that of BCLA and CLAD.

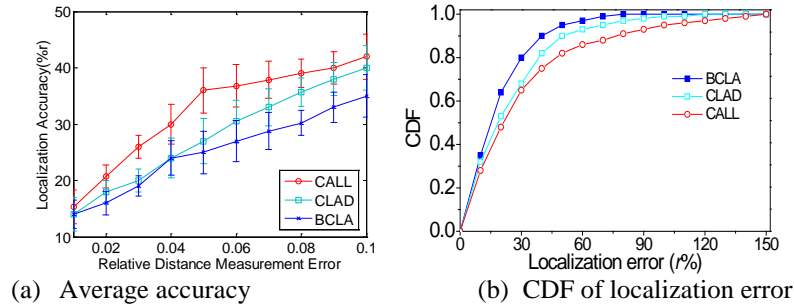


Fig. 13. Localization accuracy of different algorithms versus relative distance measurement error: (a) average accuracy, (b) CDF

We plot the cumulative distribution functions (CDFs) of localization error in different algorithms when the relative distance measurement error is 5% in Fig. 13(b). BCLA and CLAD outperform CALL because they can uniquely localize more nodes. The median localization accuracy in the three algorithms are similar, namely, 0.15r, 0.19r, and 0.21r for BCLA, CLAD, and CALL, respectively. However, the ratios of nodes with large localization error in CLAD and CALL are significantly larger than that in BCLA. For the example, the 90th percentile localization error is about 0.4r in BCLA, while the 90th percentile localization errors are 0.5r and 0.74r in CLAD and CALL. The reason is as follows. In CLAD, some components use angular information that contains error to perform component merging. Although this can help localize more nodes, this also magnifies the localization error. In CALL, some nodes cannot obtain unique location estimations, and thus the localization error is relatively large.

5.5 Impact of Angle Measurement Error on Localization Accuracy

Fig. 14(a) shows the localization accuracy of CLAD when average node degree is 7. We can observe that the accuracy degrades when angle measurement error increases. We can also observe that, by increasing the number of anchors, the localization accuracy can be effectively improved. This is because with more anchors the distance inferred from angle information can be more accurate, and thus CLAD can relieve the accuracy degradation caused by angle measurement noisy. Fig. 14(b) plots the accuracy of CLAD when the angle measurement error increases. We observe that with higher node degree the accuracy degradation is also relieved.

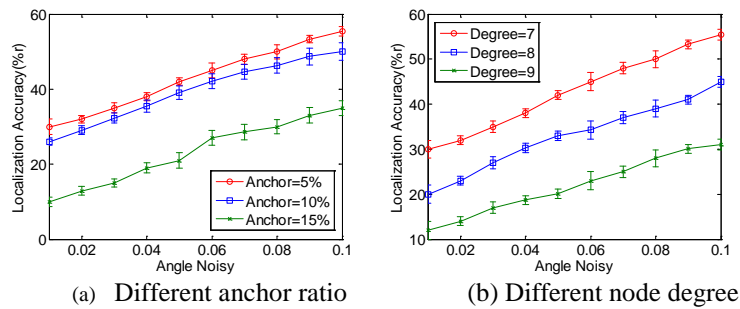


Fig. 14. Localization accuracy of CLAD when angle measurement error varies

Fig. 15 plots the CDF of nodes' localization error in CLAD for different combinations of anchor ratio and node degree, assuming that the relative angular measurement error is 5%. We can observe that when the localization accuracy improves when both anchor ratio and node degree increases. When there are more anchors, components can be merged in early stages and the possible number of candidate positions of nodes can be decreased, and thus the localization accuracy improves. When the node degree increases, components have larger chance to be merged and more nodes can be uniquely localized. From **Fig. 15**, we can also observe that the node degree has larger impact on the localization error than anchor ratio does.

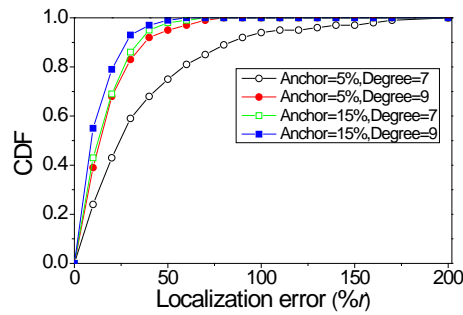


Fig. 15. CDF of nodes' localization error in CLAD for different combinations of anchor ratio and node degree.

5.6 Percent of Nodes Using Angle Information

Fig. 16 plots the ratio of nodes using angle information among all the localized nodes when the average node degree increases. In cases when the network is very sparse, i.e., when the average degree is less than six, it is difficult to infer distance between non-adjacent node pairs because the links between nodes are very sparse. Thus the improvement caused by utilizing angle information is insignificant. Angle information shows its advantage when the network density is middle, i.e., when the average degree is between 7 and 9. In this case, it is easy to infer distance between non-adjacent nodes, which greatly improves the ratio of uniquely localized nodes. However, when the network degree is high, e.g., when network degree is larger than 9, the links between nodes is dense enough to uniquely merge adjacent components and thus the improvement caused by using angle information is also limited. To summarize, the improvement caused by using angle information is most significant when the network is in middle density.

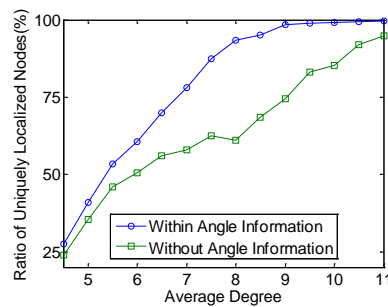


Fig. 16. Percent of nodes using angle information in different degree.

6. Conclusion

In this paper, we proposed two component based localization algorithms, namely BCLA and CLAD, which can achieve high localization coverage (i.e., the ratio of uniquely localized nodes in the network) in sparse WSNs. Different from CALL, the state-of-the-art component-based localization algorithm, BCLA and CLAD group nodes into overlapping components and utilize both common nodes and distance measurements between adjacent components to merge components, and thus greatly improves localization coverage. CLAD further uses angle information to improve the localization coverage. Simulation results show that, CLAD can uniquely localize more the 90 percent nodes in a sparse network with average node degree 7.5, while CALL can localize only 78 percent of all nodes. Besides, the localization accuracy of BCLA and CLAD is also higher than that of CALL. In the future, we will investigate how to further improve the localization coverage and localization accuracy of component-based algorithms by simultaneously merging more than two components.

References

- [1] F.Y. Ren, H.N. Huang and C. Lin, "Wireless sensor networks," *Journal of Software*, vol. 14, no. 1, pp. 1282-1291, 2003.
- [2] C. Wang, L. Xing and V.M. Vokkarane, "Reliability modeling of wireless sensors," in *Proc. of Annual Reliability and Maintainability Symposium*, pp. 1-6, 2013. [Article \(CrossRef Link\)](#).
- [3] M. Baratch, N. Meratnia, PJ Havinga and A. Skidmore, "Sensing Solutions for Collecting Spatio-Temporal Data for Wildlife Monitoring Applications: A Review," *MDPI/Sensors*, vol. 13, no. 5, pp. 6054-6088, 2013. [Article \(CrossRef Link\)](#).
- [4] M. Navarr, TW Davis, Y. Liang and X. Liang, "ASWP: a long-term WSN deployment for environmental monitoring," in *Proc. of the 12th Int. Conf. on Information Processing in Sensor Networks*, pp. 351-352, 2013. [Article \(CrossRef Link\)](#).
- [5] XP Guan, "Real-Time Localization Algorithm for Maritime Search and Rescue Wireless Sensor Network," *International Journal of Distributed Sensor Networks*, pp.2013, 2013. [Article \(CrossRef Link\)](#).
- [6] M. Liu, B. Liu and Y. Wen, "An Efficient Data Evacuation Strategy for Sensor Networks in Postdisaster Applications," *International Journal of Distributed Sensor Networks*, pp.2013, 2013. [Article \(CrossRef Link\)](#).
- [7] ZF Liao, JX Wang, SG Zhang, JN Cao and GY Min, "Minimizing Movement for Target Coverage and Network Connectivity in Mobile Sensor Networks," *Accepted to appear in IEEE Transactions on Parallel and Distributed Systems*, 2014. [Article \(CrossRef Link\)](#).
- [8] WZ Luo, JX Wang, J Guo, and JE Chen, "Parameterized complexity of Max-lifetime Target Coverage in wireless sensor networks," *Theoretical Computer Science*, vol. 518, pp. 32-41, 2014. [Article \(CrossRef Link\)](#).
- [9] ZF Liao, JX Wang, SG Zhang, and X Zhang, "A Deterministic Sensor Placement Scheme for Full Coverage and Connectivity Without Boundary Effect in Wireless Sensor Networks," *Ad hoc & sensor wireless networks*, vol. 19, no. 3-4, pp. 327-351, 2013.
- [10] J Zhen, MZA Bhuiyan, SH Liang, XF Xing, and GJ Wang, "Auction-based adaptive sensor activation algorithm for target tracking in wireless sensor networks," *Future Generation Computer Systems*, vol. 39, no. 10, pp. 88-99, 2014. [Article \(CrossRef Link\)](#).
- [11] GJ Wang, MZA Bhuiyan, JN Cao, J Wu, "Detecting Movements of a Target Using Face Tracking in Wireless Sensor Networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no.4, pp. 939-949, 2014. [Article \(CrossRef Link\)](#).
- [12] JX Wang, ZX Liu, SG Zhang and X Zhang, "Defending collaborative false data injection attacks in

- wireless sensor networks,” *Information Sciences*, vol. 254, pp. 39-53, 2014. [Article \(CrossRef Link\)](#).
- [13] AF Liu, J Ren, X Li, ZG Chen, and S Shen, “Design principles and improvement of cost function based energy aware routing algorithms for wireless sensor networks,” *Computer Networks*, , vol. 56, no. 7, pp. 1951–1967, 2012. [Article \(CrossRef Link\)](#).
- [14] A. Koubaa, M.B. Jamaa MB and A. AlHaqbani, “An empirical analysis of the impact of RSS to distance mapping on localization in WSNs,” in *Proc. of the 3rd International Conference on Communications and Networking*, pp. 1-7, 2012. [Article \(CrossRef Link\)](#).
- [15] K. Anusha, M. Gosul and GD Kumar, “Localization of Mobile Nodes in Wireless Networks with Correlated in Time Measurement Noise,” *International Journal of Computational Science and Engineering*, vol. 4, no. 9, pp. 1587-1593, 2012.
- [16] Y. Yan, H. Wang, and X. Shen, “Efficient convex optimization method for underwater passive source localization based on RSS with WSN”, in *Proc. of IEEE Int. Conf. on Signal Processing, Communication and Computing*, pp. 171-174, 2012. [Article \(CrossRef Link\)](#).
- [17] Y. Gao, WS Zhao and C. Jing, “WSN node localization algorithm based on adaptive particle swarm optimization,” *IEEE Applied Mechanics and Materials*, vol. 143, pp. 302-306, 2012. [Article \(CrossRef Link\)](#).
- [18] J. Aspnes, T. Eren and DK Goldenberg DK, “A theory of network localization,” *IEEE Transactions on Mobile Computing*, vol. 5, no. 12, pp. 1662-1678, 2006. [Article \(CrossRef Link\)](#).
- [19] T. Eren, OK Goldenberg OK and W. Whiteley, “Rigidity, computation, and randomization in network localization,” in *Proc. of the 23rd Annual Joint Conf. of the IEEE Computer and Communications Societies*, pp. 2673-2684, 2004. [Article \(CrossRef Link\)](#).
- [20] J. Fang, M. Cao, and A.S. Morse, “Localization of sensor networks using Sweeps,” in *Proc. of 45th IEEE Conf. on Decision and Control*, pp. 4645-4650, 2006. [Article \(CrossRef Link\)](#).
- [21] X. Wang, J. Luo, and S. Li, “Component Based Localization in Sparse Wireless Ad Hoc and Sensor Networks,” in *Proc. of IEEE Int. Conf. on Network Protocols*, pp. 288-297, 2008. [Article \(CrossRef Link\)](#).
- [22] X. Wang, J. Luo, YH Liu, SS Li and DZ Dong, “Component Based Localization in Sparse Wireless Networks,” *IEEE/ACM Transactions on Networking*, vol. 19, no. 12, pp. 540-548, 2011. [Article \(CrossRef Link\)](#).
- [23] BD Anderson, PN Belhumeur and T. Eren, “Graphical properties of easily localizable sensor networks,” *Wireless Networks*, vo. 15, no. 2, pp. 177-191, 2009. [Article \(CrossRef Link\)](#).
- [24] DK Goldenberg and A. Krishnamurthy, “Network localization in partially localizable networks,” in *Proc. of 24th IEEE Annual Joint Conf. of the IEEE Computer and Communications Societies*, pp. 313-326, 2005. [Article \(CrossRef Link\)](#).
- [25] J. Bachrach and C. Taylor, “Localization in Sensor Networks,” *Handbook of Sensor Networks Algorithms and Architectures*, Chapter 9, pp. 277-310, WILEY & SONS, 2005. [Article \(CrossRef Link\)](#).
- [26] SG Zhang, JN Cao, LJ Chen, and DX Chen, “Accurate and energy-efficient range-free localization for mobile sensor networks,” *IEEE Transaction on Mobile Computing*, vol. 9, no. 6, pp. 897-910, 2010. [Article \(CrossRef Link\)](#).
- [27] XP Wang, YH Liu, Z Yang, K Lu and J Luo, “Robust component-based localization in sparse networks”, *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 5, pp. 1317-1327. [Article \(CrossRef Link\)](#).
- [28] Y Zhao, HY Wu, M Jin, S Xia, “Localization in 3D surface sensor networks: Challenges and solutions,” in *Proc. of the 31st Annual IEEE Int. Conf. on Computer Communications*, pp. 55-63, 2012. [Article \(CrossRef Link\)](#).
- [29] Y Zhao, H Wu, M Jin, Y Yang, H Zhou, and S Xia, “Cut-and-sew: A distributed autonomous localization algorithm for 3d surface wireless sensor networks,” in *Proc. of the 14th ACM Int. Symposium on Mobile Ad Hoc Networking and Computing*, pp. 69-78, 2013. [Article \(CrossRef Link\)](#).
- [30] Y Yang, M Jin, HY Wu, “3D surface localization with terrain model,” in *Proc. of the 33rd Annual IEEE Int. Conf. on Computer Communications*, pp. 46-54, 2014. [Article \(CrossRef Link\)](#).

- [31] WP Liu, D Wang, HB Jiang, WY Liu, CG Wang, "Approximate Convex Decomposition Based Localization in Wireless Sensor Networks," in *Proc. of the 31st Annual IEEE Int. Conf. on Computer Communications*, pp. 1853-1861, 2012. [Article \(CrossRef Link\)](#).
- [32] SS Ji, KF Sze, ZR Zhou, A MC So, YY Ye, "Beyond convex relaxation: A polynomial-time non-convex optimization approach to network localization," in *Proc. of the 32nd Annual IEEE Int. Conf. on Computer Communications*, pp. 2499-2507, 2013. [Article \(CrossRef Link\)](#).
- [33] J Liu, ZH Wang, Z Peng, JH Cui, L Fiondella, "Suave: Swarm underwater autonomous vehicle localization," in *Proc. of the 33rd Annual IEEE Int. Conf. on Computer Communications*, pp. 64-72, 2014. [Article \(CrossRef Link\)](#)
- [34] YC Shu, P Cheng, Y Gu, JM Chen, T He, "TOC: Localizing wireless rechargeable sensors with time of charge," in *Proc. of the 33rd Annual IEEE Int. Conf. on Computer Communications*, pp. 388-396, 2014. [Article \(CrossRef Link\)](#)



Shigeng Zhang received the BSc, MSc, and DEng degrees, all in Computer Science, from Nanjing University, China, in 2004, 2007, and 2010, respectively. He is currently an Assistant Professor in School of Information Science and Engineering at Central South University, China. His research interests include Cloud Computing, Internet of Things, wireless sensor networks, and RFID systems. He has published more than 30 technique papers in international journals and conferences. He is a member of IEEE, a member of ACM, and a member of CCF.



Shuping Yan received the BSc degree in Computer Science from Hunan Normal University, China, in 2014. Currently, she is a master student in the School of Information Science and Engineering at Central South University, China. Her research interests include Internet of Things, indoor localization algorithms and systems.



Weitao Hu received his BSc and MSc degrees in Computer Science from Central South University in 2011 and 2014, respectively. His research interests include localization in wireless sensor networks.



Jianxin Wang received his BSc and MSc degrees in Computer Science from Central South University of Technology, P. R. China, and his PhD degree in Computer Science from Central South University. Currently, he is the vice dean and a professor in the School of Information Science and Engineering, Central South University, Changsha, Hunan, P.R. China. Jianxin Wang is currently serving as executive editor of International Journal of Bioinformatics Research and Applications and serving in the editorial board of International Journal of Data Mining and Bioinformatics. He has also served as a program committee member for many international conferences. He was a program committee co-chair for the 7th and 8th International Symposium on Bioinformatics Research and Applications (ISBRA2011 and ISBRA2012), will be a program committee co-chair for the 8th International Frontiers of Algorithmics Workshop (FAW2014) and 10th International Symposium on Bioinformatics Research and Applications (ISBRA2014). His current research interests include algorithm analysis and optimization, parameterized algorithm, bioinformatics and computer networks. He has published more than 200 papers in various International journals and refereed conferences. He is a senior member of the IEEE.



Kehua Guo received his PhD degree in Computer Science from Nanjing University of Science and Technology in 2008. He was an Associate Professor in the School of Information Science and Engineering at Central South University since 2012. Currently, he is a visiting Scholar in Texas A&M University. His research interests focus on green computing, transparent computing and multimedia computing..