Beyond Triangle Inequality: Sifting Noisy and Outlier Distance Measurements for Localization

Lirong Jian, Zheng Yang and Yunhao Liu
Hong Kong University of Science and Technology
{jlrphx, yangzh, liu}@cse.ust.hk

***Abstract*—Knowing accurate positions of nodes in wireless ad-hoc and sensor networks is essential for a wide range of pervasive and mobile applications. However, errors are inevitable in distance measurements and we observe that a small number of outliers can degrade localization accuracy drastically. To deal with noisy and outlier ranging results, triangle inequality is often employed in existing approaches. Our study shows that triangle inequality has a lot of limitations which make it far from accurate and reliable. In this study, we formally define the outlier detection problem for network localization and build a theoretical foundation to identify outliers based on graph embeddability and rigidity theory. Our analysis shows that the redundancy of distance measurements plays an important role. We then design a bilateration generic cycles based outlier detection algorithm, and examine its effectiveness and efficiency through a network prototype implementation of MicaZ motes as well as extensive simulations. The results shows that our design significantly improves the localization accuracy by wisely rejecting outliers.**

1. Introduction

Growing convergence among mobile computing devices and embedded technology sparks the development and deployment of a wide range of location-aware applications, including Smart Space, Intrusion Detection, Inventory Management, Mobile Peer-to-Peer Computing, Wireless Sensor Networks (WSNs) Surveillance, etc. A number of localization approaches have been proposed in the literature and used in practice to locate wireless devices.

In recent years, novel schemes have been proposed for in-network localization, in which some special nodes (called beacons or anchors) know their global locations and the rest determine their locations by measuring the Euclidean distances to their neighbors. Several distance ranging methods, such as Radio Signal Strength (RSS) [28] and Time Difference of Arrival (TDoA) [26], are adopted in practical systems. Based on these techniques, the ground truth of a wireless ad-hoc network can be modeled as a weighted graph $G = \langle V, E, W \rangle$, where V is the set of wireless nodes, E is the set of links, and $W(u, v)$ is the distance measurement between a pair of nodes u and v. The problem of localization is to figure out the locations of other nodes based on inter-node distance measurements and global locations of those beacons.

In practice, however, noises and outliers are inevitable in distance ranging. The outlier ranging results generally come from the following three sources.

Hardware malfunction or failure. Distance measurements will be meaningless when encountering ranging hardware malfunction. Besides, incorrect hardware calibration and configuration also deteriorate ranging accuracy, which is not much emphasized by previous studies. For example, RSS suffers from transmitter, receiver, and antenna variability, and the inaccuracy of clock synchronization results in ranging errors for TDoA.

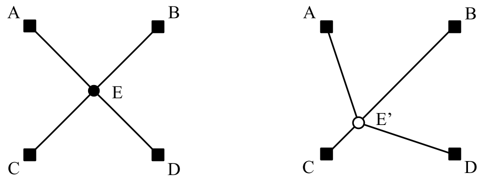
Environmental factors. RSS is sensitive to channel noise, interference, and reflection, all of which have significant impacts on signal amplitude. The irregularity of signal attenuation remarkably increases, especially in complex indoor environments. In addition, for the propagation time based ranging measurements, the signal propagation speed often exhibits variability as a function of temperature and humidity, so we cannot assume the propagation speed is constant across a large field.

Adversary attacks. As location-based services become more and more prevalent, the localization infrastructure is becoming the target of adversary attacks. By reporting fake location or ranging results, an attacker, e.g., a compromised (malicious) node, can completely distort the coordinate system. Different from the previous cases, the outliers here are intentionally generated by adversaries.

Ignoring the existence of outliers is not a choice to deal with them. Fig. 1 shows an example that how outliers destroy the localization accuracy. As shown in Fig. 1, Node A, B, C and D (the black boxes) are beacons at known positions and Node E (the white circle) is to be located. Apparently, the calculated location of E is just the same as its real location when distance measurements are correct. Now suppose an outlier ranging result occurs: the distance between E and B is wrongly measured as 2 ($|\text{BE}| = 2$). In this case, if all ranging results are indiscriminately used to locate E, the estimated location E' (the white circle in Fig. 1(b)) is away from the real location E by using multilateration [26]. However, if we lay the outlier ranging out, a better estimated location can be achieved, which is the same as the real location of E.

To detect outliers, a straight-forward solution is to judge graph embeddability based on triangle inequality. A graph violating triangle inequality is impossible to be embeddable. However, triangle inequality has its drawbacks in the following two aspects:

Coarse granularity. In the case of Fig. 1(b), the outlier ranging $|\text{BE}|$ cannot be detected by triangle inequality since the triangle $\triangle$ABE (the distance between A and B is $\sqrt{2}$, implied by their locations) is still embeddable under the



(a) Ground truth. (b) Estimated location.

Fig. 1. Multilateration with an outlier measurement, where A, B, C and D are at corners of a square of $\sqrt{2}$, and E is at the center of that square.

incorrect rangings. Actually in this case all triangles in Fig. 1(b), including △ABE, ACE, ADE, BCE, BDE and CDE, are embeddable and triangle inequality fails to detect outliers.

Identification. Triangle inequality just indicates the existence of outliers, but cannot identify them. Formally, for a triangle violating the inequality, we are sure at least one distance measurements are incorrect but have no idea which are them.

Such limitations motivate us to design a novel approach to identify outlier distance measurements beyond triangle inequality. We notice that a rigid graph has discrete and finite realizations. A graph is called *rigid* if one cannot continuously deform the graph embedded in the plan while preserving distance constraints [15]. Thus for a graph G and a pair of neighboring vertices u and v, if $G-(u,v)$ is rigid, the distance between them ($W(u,v)$) is accordingly discrete and finite under different realizations of $G-(u,v)$. Although we artificially exclude and ignore (u,v), we still have some ideas to partially figure $W(u,v)$ out. Such redundancy in inter-node distances is an excellent starting point to detect noises and outliers in distance ranging.

The main contributions of this work are as follows. We formally define the problem of outlier detection for localization and analyze the limitations of existing methods that are based on triangle inequality. Based on graph embeddability and rigidity, we establish the theoretical foundations for detecting outliers. An algorithm based on bilateration and generic cycles is accordingly designed to eliminate outliers during the localization process. To validate this design, a network prototype consisting of 25 MicaZ motes is constructed and extensive simulations are conducted to examine the effectiveness and efficiency of our solution.

The rest of this paper is organized as follows. In Section 2, we build the network model and formulate the problem of outlier detection. Section 3 presents the theoretical foundation for identifying outliers and an algorithm based on graph embeddability and rigidity is designed in Section 4. Section 5 analyzes some practical issues of the algorithm. Section 6 and Section 7 present the evaluation results from extensive simulations and field experiments. We summarize the related work in Section 8, and conclude the work in Section 9.

2. Problem Formulation

In this section, we present the network model and formulate outlier detection for localization in wireless ad-hoc and sensor networks.

2.1 Network Model

We assume that each node is located at a distinct physical location in some region of a plane and associated with a specific set of "neighboring" nodes. Let $\mathbb{N}$ be a network of n nodes labeled $v_1, v_2, ..., v_n$. Let $\pi(v_i)$ denote the ground truth position of v_i. And we suppose a small portion of nodes, called *beacons*, are at known locations.

We generate a *weighted grounded graph* [27], [5] $G = \langle V, E, W\rangle$ to represent the geometric constraints of Network $\mathbb{N}$, where each vertex denotes a node in the network and each edge e indicates the neighborhood of its two endpoint nodes. The distance between two neighboring nodes (or any pair of beacons) is available and defined by $W(e)$.

2.2 Error Model of Distance Ranging

Given a weighted grounded graph $G = \langle V, E, W\rangle$, W is determined by the observed information, such as measured inter-node distances and positions of beacons. We call W *measured distance* in the rest of this paper. Because of the presence of noises, those measurements would be corrupted. Formally, for some edge $e = (v_i, v_j) \in E$, $W(e)$ is not necessarily equal to the *ground truth distance* between nodes v_i and v_j, i.e., $\|\pi(v_i) - \pi(v_j)\|$.

In general, there are two kinds of ranging errors in a localization system: *normal error* and *outlier error*. Coming from limitations of hardware and computation precision, normal errors are moderate and predictable, and have attracted a lot of research efforts [23], [21]. By contrast, outlier errors are much severer and unpredictable, caused by hardware malfunction or failure, adversary attacks and etc. We formulate such two kinds of errors as normal edges and outlier edges, respectively, as follows:

Definition 1. Given a weighted grounded graph $G = \langle V, E, W\rangle$, if an edge $e = (v_i, v_j) \in E$ is a *normal edge*, then $W(e) = \|\pi(v_i) - \pi(v_j)\|$; otherwise, e is an *outlier edge*, and $W(e) = \|\pi(v_i) - \pi(v_j)\| + \epsilon$, where ϵ is an arbitrary continuous random variable.

In the error model, we assume normal edges contain no ranging noises, which is a good starting point to address the outlier detection problem. This assumption and abstraction is also accepted in [14], [3], [23]. And in Section 5, by introducing an error threshold, the proposed algorithm can handle a more practical error model, where normal edges are with moderate ranging errors.

2.3 Outlier Detection Problem

Since outlier errors have seriously negative impacts, sifting outliers is essential for highly accurate localization. We formulate the problem of outlier detection as follows:

Problem 1 (Outlier Detection)**.** Given a weighted grounded graph $G = \langle V, E, W\rangle$ consisting of normal edges and outlier edges, identify those outlier edges in G.

Every edge $e \in E$ is associated with a measured distance or a pair of beacons at known positions. For the former, an outlier edge indicates the corresponding measured distance is with a large error, while for the latter, an outlier edge implies the global positions of two corresponding beacons are corrupted.

All normal edges are compatible, e.g., satisfying triangle inequality, since they are telling the truth, while it is quite probable that an outlier edge is inconsistent with other normal edges and outlier edges, e.g., violating triangle inequality, because it is just a random variable independent with the ground truth. This observation motivates us to detect outlier edges through exposing geometric inconsistency.

3. Theoretical Foundation

In this section, we analyze the relationship between outlier edges and graph embeddability, and apply rigidity theory to build a theoretical foundation for outlier detection.

3.1 Embeddability of Weighted Graphs

In [27], the problem of embeddability of weighted graphs is formally defined and its computational complexity has also been discussed.

Definition 2. Given a weighted graph, $G = \langle V, E, W \rangle$, a *realization* (*embedding*) of G in the k-dimensional Euclidean space, $\mathbb{R}^k$, is a function r, mapping V into $\mathbb{R}^k$, such that for each edge $e = (v_i, v_j) \in E$, $W(e) = \|r(v_i) - r(v_j)\|$.

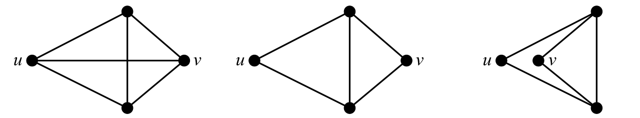
Problem 2 (Embeddability)**.** Given a weighted graph, $G = \langle V, E, W \rangle$, determine whether G is *k-embeddable*, which means whether there is a realization (embedding) r of G in $\mathbb{R}^k$.

Since we focus on networks in a 2D plane, the term "embeddable" is used instead of "2-embeddable" in the following discussion for simplicity. The following theorem reveals a close relationship between outlier existence and graph embeddability.

Theorem 1. *Given a weighted grounded graph $G = \langle V, E, W \rangle$, if G is unembeddable, then E contains at least one outlier edge.*

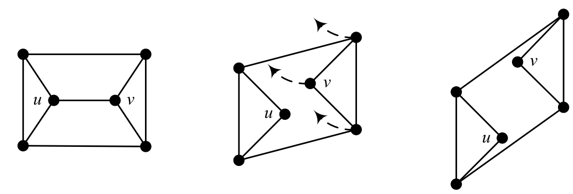
Proof: Assume that E contains no outlier edge, i.e., all the edges of G are normal edges. According to the definition, for any normal edge $e = (v_i, v_j) \in E$, we have $W(e) = \|\pi(v_i) - \pi(v_j)\|$. Let $r(v_k) = \pi(v_k)$ for every vertex $v_k \in V, k = 1, 2, ..., n$, then $W(e) = \|r(v_i) - r(v_j)\|$. Thus, G is embeddable, contradicting that G is unembeddable. ■

Theorem 1 has overcome the issue of coarse granularity associated with triangle inequality. Recall Fig. 1(b), where triangle inequality fails to detect outliers. But Theorem 1 succeeds, since the underlying graph is unembeddable. However, similar to triangle inequality, it is also impossible for Theorem 1 to tell which edges are outliers or which are not outliers. Generally, under an arbitrary error model, for any given weighted grounded graph, the inverse of Theorem 1 doesn't hold, namely, even G contains outliers, it can still be embeddable. One extreme example is illustrated as follows.



(a) Ground truth (b) First realization (c) Second realization

Fig. 2. Remaining rigid after removing (u, v).



(a) Ground truth (b) Flexing 1 (c) Flexing 2

Fig. 3. Flexing After removing (u, v).

Let $G = \langle V, E, W \rangle$, and for every edge $e = (v_i, v_j) \in E$, let $W(e) = \frac{1}{2}\|\pi(v_i) - \pi(v_j)\|$. Thus, every edge in E is an outlier, but apparently, G is embeddable. However, compared with an arbitrary error model, we are more interested in under the normal edge and outlier edge error model, whether there is some kind of weighted graphs whose being embeddable would imply no outlier.

Suppose the graph G shown in Fig. 2(a) is embeddable and we are interested in whether the edge $e = (u, v)$ is an outlier. After removing e, while preserving the remaining distance constraints, there are only two feasible options for the distance between u and v, corresponding to the two realizations illustrated in Fig. 2(b) and Fig. 2(c), respectively. Thus, the embeddability of G suggests that $W(u, v)$ is one of the two options, and what's more, $e = (u, v)$ is a normal edge which is telling the truth, as it is with probability of 0 that the measured distance of an outlier edge is in a set consisting of two discrete values. This conclusion makes the quadrilateral structure play an important role in robust localization [20], [19] and phantom nodes filtering [11].

As aforementioned, the inverse of Theorem 1 doesn't hold for all network structures. Suppose the graph G shown in Fig. 3(a) is embeddable. After removing edge $e = (u, v)$, while preserving the remaining distance constraints, the graph structure can continuously flex, as illustrated in Fig. 3(b) and Fig. 3(c), which means the distance between u and v can continuously vary in an interval having measure non-zero. Similar to Fig. 2, the embeddability of G implies that the measure distance of e falling into the feasible interval. However, since that interval has measure non-zero, it is with probability greater than 0 that e is an outlier edge and the measured distance of e falling into that interval. Thus, we cannot claim that e is not an outlier even knowing G is embeddable.

3.2 Redundant Rigidity

Previous studies have shown that the network localizability problem [5], [6] is closely related to graph rigidity. In this paper, we will show that the outlier detection problem also has a close relationship with graph rigidity. A graph is *generically rigid* (or just called rigid) if it has no continuous deformation other than global rotation, translation, and reflection while preserving distance constraints [15]; otherwise, it is *flexible*. Here the word "generically" means the distances are algebraically independent, i.e., no degeneracy. A graph is called *generically redundantly rigid* (or just called redundantly rigid) if it remains rigid after removing any single edge. Fig. 3(a) shows a rigid but not redundant rigid graph, since after removing $e = (u, v)$, this graph becomes flexible, as illustrated in Fig. 3(b) and Fig. 3(c). And it is also easy to check that after removing any edge of the graph shown in Fig. 2(a), i.e., $e = (u, v)$, the graph still has no continuous deformation while preserving the remaining distance constraints. Thus, it is redundantly rigid.

Before presenting the relationship between the outlier detection problem and graph rigidity, we first give out the definition of *outlier disprovable* as follows.

Definition 3. Given a weighted graph $G = \langle V, E, W \rangle$, G is *outlier disprovable* if and only if the embeddability of G implies that it contains no outlier edge.

Lemma 1. *Given a redundant rigid weighted graph G, if it is embeddable, then with probability of 1 G contains no outlier edge.*

Proof: Let $e = (u, v)$ be an outlier edge in G, and G is embeddable. Because of the redundant rigidity of G, after removing e, it is still rigid, which means there are finite discontinuous realizations up to congruence for $\tilde{G} = G - \{e\}$. In each realization of $\tilde{G}$, the distance between node u and node v is fixed. So, totally there are finite discrete values, denoted by set S, for the distance between u and v. As G is embeddable and $e \in E$, the measured distance of e is in S. However, as an outlier edge, the measured distance of e is a continuous random variable, which means that with probability of 0, $W(e) \in S$, since S has measure zero. So, with probability of 1 there is no outlier edge in G. ■

To analyze the necessary condition for outlier disprovability, we present a proposition first proved by Hendrickson [10].

Proposition 1. *If a graph G is flexible, then for almost all realizations r of G, the flexing space of r contains a submanifold that is diffeomorphic to the circle.*

Lemma 2. *Given a weighted graph G, if G is outlier disprovable, then G is redundantly rigid.*

Proof: Assume the only interesting case is that G is rigid but not redundantly rigid. Thus, there is some edge $e = (u, v)$ whose removal generates a flexible graph $\tilde{G}$. By Proposition 1, for almost all realizations r of $\tilde{G}$, the flexing space of r contains a submanifold that is diffeomorphic to the circle. The distance between nodes u and v will be a multivalued function for almost every point on this circle. Hence, the set of distances between u and v corresponding to r, denoted by $S_{d(u,v)}(r)$, has measure non-zero. So, even e is an outlier edge, with probability greater than 0, $W(e) \in S_{d(u,v)}(r)$, resulting in G is embeddable. Therefore, G is not outlier disprovable. ■

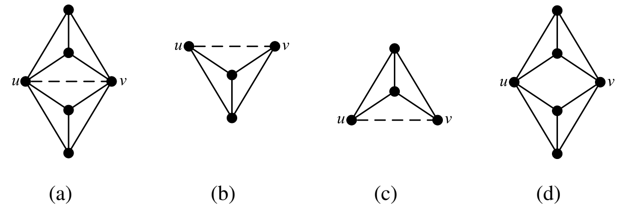


Fig. 4. Redundantly rigid components in a graph with outlier edge (u, v).

Combining Lemma 1 and Lemma 2, we obtain the following theorem.

Theorem 2. *Given a weighted graph G, G is outlier disprovable if and only if G is redundantly rigid.*

By Theorem 1 and Theorem 2, we present a framework for outlier detection , which is outlined by Algorithm 1. We explain the framework through an example shown in Fig. 4, where solid lines denote normal edges and the outlier is denoted by the dashed line. Fig. 4(a) is the ground truth, say it G. Besides Fig. 4(a), there are other 3 redundantly rigid components, shown in Fig. 4(b), Fig. 4(c) and Fig. 4(d), respectively. Intuitively, Fig. 4(a), Fig. 4(b) and Fig. 4(c) are unembeddable, since they all contain $e = (u, v)$, the outlier edge. Following our algorithm, all the edges of G are marked outlier edges before checking the embeddability of Fig. 4(d). As the outlier edge is not contained in Fig. 4(d), it is embeddable. All the edges in Fig. 4(d) are marked normal edges. Finally, we mark all the edges of G except $e = (u, v)$ normal edges, leaving e marked an outlier edge.

Algorithm 1 Component-based Outlier Detection Algorithm.

```
for all redundantly rigid component H in G do
    if H is embeddable then
        mark every edge e ∈ H a normal edge.
    else
        for all edge e ∈ H not marked a normal edge do
            mark e an outlier edge.
        end for
    end if
end for
```

Algorithm 1 addresses the outlier detection problem from the perspective of components: if a redundantly rigid component is embeddable, then all its edge are normal edges. Now, let us look at the problem from another perspective: a single edge, say it e. Intuitively, by Theorem 2, through checking all redundantly rigid components containing e, if there is one embeddable, then we know that e is a normal edge. The question is that is it necessary to check all the redundantly rigid components containing e, especially when e is actually an outlier edge?

Fig. 5(a) illustrates an example and suppose we want to verify edge $e = (u, v)$. There are two redundantly rigid components containing e: the whole graph G and the component H highlighted by the dashed circle. If H is embeddable, then e is definitely marked a normal edge; otherwise, G is also unembeddable as it contains H, and e is marked an outlier finally. Thus, whether we can verify e is only determined by the embeddability of H, and has nothing to do with G. Actually, for any edge in G, checking the embeddability of G is meaningless. This example suggests that for a single edge e, there are some "minimally" redundantly rigid components containing e such that the "identity" of e is only determined by the embeddabilities of these components.

3.3 Generic Cycle

Definition 4 ([12], [2]). A graph $G = \langle V, E \rangle$ with $|V| \geq 4$ is called a *generic cycle* if $|E| = 2|V| - 2$ and G satisfies

$$i(X) \leq 2|X| - 3 \text{ for all X} \subset \text{V with } 2 \leq |\text{X}| \leq |\text{V}| - 1 \quad (1)$$

where $i(X)$ denotes the number of edges induced by X in G.

Two instances of generic cycles are illustrated in Fig. 5(b) and Fig. 5(c). By the Laman condition [15] and the definition of redundant rigidity, a generic cycle is minimally redundantly rigid. Here the word "minimally" means containing no redundantly rigid proper subgraph.

The following lemma shows the relationship between generic cycles and redundantly rigid graphs.

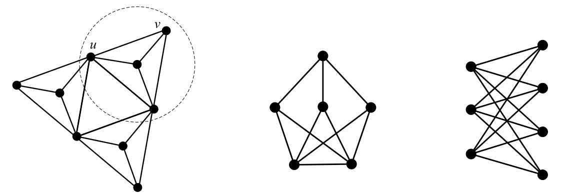
Lemma 3 ([12]). *A graph G is redundantly rigid if and only if G is rigid and each edge of G belongs to a generic cycle in G.*

Based on Lemma 3, from the perspective of a single edge, we propose another framework for outlier detection as Algorithm 2. Recall Fig. 5(a). For every edge $e \in G$, through employing Algorithm 2, only the embeddability of the complete graph K_4 containing e, which is the smallest generic cycle, need to be checked. And the embeddability of G would never be considered. By the redundant rigidity of generic cycles and Lemma 3, we obtain the following theorem, which shows that Algorithm 1 and Algorithm 2 are equivalent.

Theorem 3. *Given a weighted graph $G = \langle V, E, W \rangle$, for any edge $e \in E$, e is marked a normal edge by Algorithm 1 if and only if e is also marked a normal edge by Algorithm 2.*

Algorithm 2 Edge-based Outlier Detection Algorithm.

```
for all e not marked in G do
    if there is a generic cycle H containing e is embeddable
    then
        mark any edge ∈ H, including e, a normal edge.
    else
        mark e an outlier edge.
    end if
end for
```



(a) (b) (c)

Fig. 5. (a) Edge (u,v) contained in two redundantly rigid components. (b) (c) Examples of generic cycles.

4. Algorithm Design and Implementation

Two key challenges exist to implement Algorithm 2: how to find those generic cycles containing edge e and check their embeddabilities. Saxe [27] proved that deciding embeddability of a given weighted graph with exact Euclidean norm edge lengths is strongly NP-hard in k dimensions, for any $k \geq 1$. In particular, even with the Unit Disk Graph (UDG) model, this problem is still weakly NP-hard [1]. A basic approach to decide embeddability is to find a realization, namely, localization. Apparently, localization is not easier than deciding embeddability. However, some classes of graphs are computationally efficient to localize: quadrilateration [23], trilateration [5] and bilateration [6] graphs. Compared to quadrilateration and trilateration, bilateration has the following advantages. (1) None of trilateration graphs is a generic cycle except K_4, but many of them are bilateration graphs. (2) Trilateration requires the network to be dense, while bilateration can be carried out in sparse networks [6]. (3) Trilateration is a LS-based statistics method, which would hide outliers [8]; by contrast, bilateration would expose geometric inconsistency effectively. Based on the above analysis, now our target becomes bilateration generic cycles. We define the problem as follows.

Problem 3 (Bilateration Generic Cycles Search). Given a bilateration ordering l of a graph G, find out those subgraphs H of G such that H is a generic cycle with a bilateration ordering $\tilde{l}$ which is a subsequence of l.

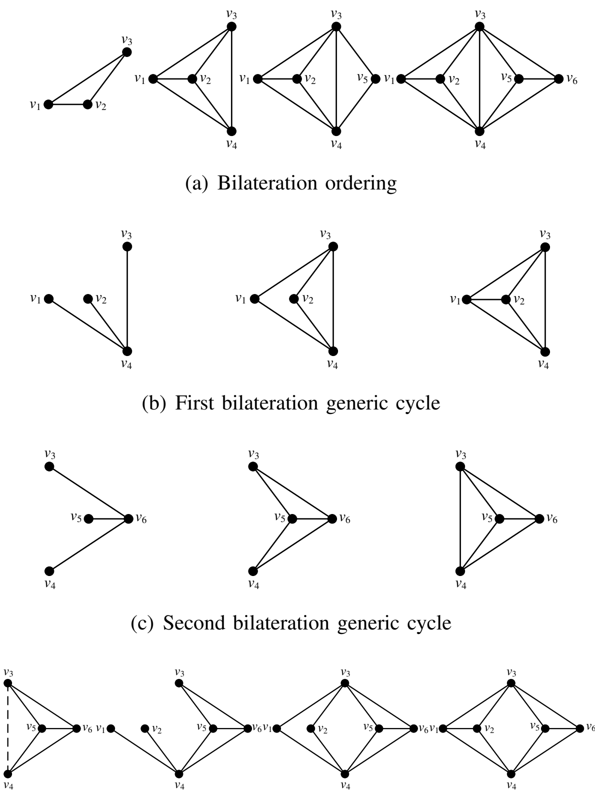
Before presenting our algorithm to find bilateration generic cycles, we first prove their properties.

Lemma 4. *A bilateration graph $G = \langle V, E \rangle$ is a generic cycle if and only if there is a bilateration ordering $l = \langle v_1, v_2, v_3, ..., v_n \rangle$ of V, such that for all $1 \leq i \leq n$, the degree of v_i $d(v_i) \geq 3$, and for all $3 \leq i \leq (n-1)$, v_i is adjacent to exact two distinct vertices v_j with $j < i$, and v_n is adjacent to exact three distinct vertices v_j with $j < n$.*

Proof: Sufficiency. The number of edges of G is $2(n-3)+3+1 = 2n-2$. Assume there are some subsets S of V such that $|S| \geq 4$ and $i_G(S) \geq 2|S| - 2$. Let X denote one of such S with the minimal cardinality, and $k = |X|$. Let $X = \langle v_{i_1}, v_{i_2}, .., v_{i_k} \rangle$, where $1 \leq i_1 < i_2 < ... < i_k \leq n$. The only interesting case is that $k < n$. Let p denote the maximal i such that $v_i \notin X$. If $p = n$, according to the property of l, the degree of v_{i_k} in the subgraph induced by X in G, $d_{G(X)}(v_{i_k}) \leq 2$. Since $i_G(X) \geq 2|X| - 2$, $i_G(X - \{v_{i_k}\}) \geq 2|X - \{v_{i_k}\}| - 2$, which contradicts the minimal cardinality of X. Thus, $i_k = n$. We rewrite X in another way, $X = \langle v_{i_1}, v_{i_2}, ..., v_{i_{k-(n-p)}}, v_{p+1}, v_{p+2}, ..., v_n \rangle$. We show that $X' = \langle v_{i_1}, v_{i_2}, ..., v_{i_{k-(n-p)}} \rangle$ is satisfying $i_G(X') \geq 2|X'| - 2$, which also contradicts the minimal cardinality of X. Since $d(v_p) \geq 3$ in G, and v_p is adjacent to at almost two distinct vertices v_j with $j < p$ (since $p \neq n$), there must be some $j > p$ and v_j is adjacent to v_p. Thus, through removing $v_{p+1}, v_{p+2}, ..., v_n$ from X to get X', we have $i_G(X) - i_G(X') \leq 2(n-p-1) + 3 - 1 = 2(n-p)$. Since $i_G(X) \geq 2|X| - 2$, $i_G(X') \geq i_G(X) - 2(n-p) \geq 2|X| - 2 - 2(n-p) = 2|X'| - 2$.

Necessity. Let G be a generic cycle with a bilateration ordering $l = \langle v_1, v_2, v_3, ..., v_n \rangle$. Thus, for all $1 \leq i \leq n$, $d(v_i) \geq 3$, and for all $3 \leq i \leq n$, v_i is adjacent to at least two distinct vertices v_j with $j < i$. Besides, there must be some v_i adjacent to at least three distinct vertices v_j with $j < i$. Let k denote the minimal of such i. Apparently, $k \geq 4$. Let $X = \langle v_1, v_2, v_3, ..., v_k \rangle$. Assume the only interesting case is that v_k is adjacent to exact three distinct vertices v_j with $j < k$ and for all $1 \leq i \leq k$, $d_{G(X)}(v_i) \geq 3$. By the first part of this proof, the subgraph $G(X)$ induced by X in G is a generic cycle. So, $G(X) = G$ and $X = V$. ■

Based on Lemma 4, an algorithm for bilateration generic cycles search is given in Algorithm 3. Before proving the correctness of Algorithm 3, we give an example to illustrate it, shown in Fig. 6, where Fig. 6(a) shows the bilateration ordering $l = \langle v_1, v_2, v_3, v_4, v_5, v_6 \rangle$. By traversing vertices of G following l, Algorithm 3 figures out all three bilateration generic cycles in G, illustrated by Fig. 6(b), Fig. 6(c) and Fig. 6(d), respectively.



(a) Bilateration ordering

(b) First bilateration generic cycle

(c) Second bilateration generic cycle

(d) Third bilateration generic cycle

Fig. 6. Bilateration generic cycles search.

Theorem 4. *Given a graph G with a bilateration ordering l, Algorithm 3 identifies all bilateration generic cycles in G with respect to l.*

Proof: Correctness. Let $C = \langle V, E \rangle$ denote a cycle identified by Algorithm 3. Apparently, $m = |V| \geq 4$. Reversing the tracing back order in Algorithm 3, we get an ordering of vertices of C, $l' = \langle v_{i_1}, v_{i_2}, ..., v_{i_m} \rangle$, where $1 \leq i_1 < i_2 < ... < i_m \leq n$ such that v_{i_1} and v_{i_2} are neighbors and for all $3 \leq k \leq (m-1)$, v_{i_k} has exact two neighbors earlier in l', and v_{i_m} has exact three neighbors earlier in l'. In particular, for all $1 \leq k \leq (m-1)$, there must be some j with $k < j \leq m$ and v_{i_j} and v_{i_k} are neighbors; otherwise, v_{i_k} would never be inserted into C. Thus, for all $3 \leq k \leq m$, $d_C(v_{i_k}) \geq 3$. Similarly, we can prove both $d_C(v_{i_1})$ and $d_C(v_{i_2}) \geq 3$. By Lemma 4, C is a bilateration generic cycle.

Optimality. By Lemma 4, every bilateration generic cycle C in G with respect to l has a bilateraion ordering $l' = \langle v_{i_1}, v_{i_2}, ..., v_{i_m} \rangle$, which is a subsequence of l, such that $m \geq 4$, and for all $1 \leq k \leq m$, $d_C(v_{i_k}) \geq 3$, and for all $3 \leq k \leq (m-1)$, v_{i_k} has exact 2 neighbors earlier in l', and v_{i_m} has exact 3 neighbors earlier in l'. From the follow of Algorithm 3 and the first part of this proof, we can see that Algorithm 3 outputs all such bilateration orderings. ■

Algorithm 3 solves the first issue, how to find generic cycles, in Algorithm 2. And through employing Sweeps [6], we can check the embeddabilities of generic cycles based on localization, which addresses the second issue.

5. DISCUSSION

In this section, we discuss some practical issues of the proposed algorithm and solutions to address them.

5.1 Distributed Implementation

Since Algorithm 2 is a framework for outlier detection from the point of view of a single edge, it is not difficult to implement it in a distributed fashion. From every single edge, after exploring a bilateration ordering l, Algorithm 3 can identify all bilateration generic cycles with respect to l. In the exploring procedure, l uses the start edge as its ID, and every involved node remembers its order in l. Once a node finds that it has more than 2 neighbors earlier in l, a bilateration generic cycles search procedure is initiated. After identifying a bilateration generic cycle, a distributed solution Sweeps [6] can be implemented to test the embeddability of the generic cycle.

5.2 Practical Error Model

We assume normal edges are without any ranging errors. This abstraction is impractical in real world networks, where

Algorithm 3 Bilateration Generic Cycles Search

```
BGCS(List order)
for all vertex x with more than 2 neighbors earlier in order do
    List ancestors = (neighbors of x earlier in order)
    for all all triple (u,v,w) in ancestors do
        SortedList visited = new SortedList
        visited.add(u,v,w)
        Graph cycle = new Graph
        cycle.add(u,(u,x),v,(v,x),w,(w,x))
        Traceback(cycle, visited, order)
    end for
end for
```

```
Traceback(Graph cycle, SortedList visited, List order)
Node x = visited.pop()
List ancestors = (neighbors of x earlier in order)
if visited.length()==1 and x and visited[0] are neighbors then
    cycle.add((x,visited[0]))
    output cycle as a generic cycle
    ancestors.remove(visited[0])
end if
for all pair(u,v) in ancestors do
    visited.add(u,v)
    cycle.add(u,(u,x),v,(v,x))
    Traceback(cycle,visited,order)
end for
```

ranging errors always exists. Our basic algorithm fails in this situation, since it is almost impossible for those generic cycles identified by Algorithm 3 to be embeddable. We introduce an error threshold δ to handle this problem. Given a bilateration generic cycle $C = \langle V, E, W \rangle$, Sweeps computes all realizations r of C. Define *additive distortion* [1] as

$$\epsilon = min_r\{max_{(u,v)}\{|W((u,v)) - \|r(u) - r(v)\||\}\} \quad (2)$$

for every $(u, v) \in E$. If $\epsilon \leq \delta$, then we say C is embeddable; otherwise, it is unembeddable. The value of δ has direct impacts on the performance of the proposed algorithm, and should be determined according to the ranging accuracy. The introduction of δ, however, raises another issue: a generic cycle with outlier edges may be claimed embeddable; namely, there are false negative results. But fortunately, the measured distances of outlier edges are usually away from their ground truth. Thus, the false negative rate is often small. If false negative results are unacceptable in some scenarios, such as security applications, besides choosing a tighter δ, we can require a normal edge must be contained in more than one embeddable generic cycles.

6. Evaluation

We explore the negative impacts of outlier rangings and how outlier detection can improve location accuracy. We generate networks of 200 nodes randomly distributed in a square area with 10% beacons. The communication radius of each node is 10 meters and the average degree of network topology is 16. We choose these network parameters to ensure every node can be localized by trilateration. The ranging errors of outlier edges is 10 times of that of normal edges. We integrate the results from 100 network instances.

The proposed outlier detection algorithm doesn't serve as a localization algorithm, but an intermediate component between measuring inter-node distances and localizing wireless devices based on such ranging information. We utilize AHLoS [26] as the basic localization method, and compare the localization performances of three strategies towards outliers: sifting, ignoring (the basic AHLoS approach), and eliminating completely (the idealized case). Note that our outlier detection algorithm works well with arbitrary localization methods. We choose AHLoS since it delegates a major class of widely used multilateration (trilateration) based localization techniques.

Fig. 7 reports the result in case of $\sigma = 5\%$ outlier edges. For all strategies, the average location error linearly increases along with increasing ranging error. In the idealized case, labeled as Without Outlier, the location errors are much less than the ignoring case labeled as No Sifting, while employing Algorithm 2 to sift outliers achieves a median location accuracy, which is close to the idealized case and also much better than the basic AHLoS approach.

We also study the typical localization errors for every single node, illustrated in Fig. 8. For all 19 nodes, the location accuracy has been improved by sifting outliers. Fig. 9 plots the characteristics of error propagation. As shown, nearly 100% of nodes have at most 2m error and 65% have at most 1m error by sifting outliers, while for the basic AHLoS algorithm, only 80% of nodes have less than 2m error and 40% of nodes have less than 1m error.

7. Field Experiment and Results

We conduct a localization experiment using *MicaZ* motes in an indoor environment. In the experiment, a network prototype consisting of 25 MicaZ motes is constructed in a 5 $\times$ 5 grid on the floor, with the grid spacing of 2 feet. Each MicaZ mote is augmented with a ranging sensor board that detects distance between two nodes by measuring the time of flight of specified acoustic pulses. Due to the reflection of walls and ceilings, which gives rise to null regions, the network connectivity is unexpectedly irregular and unsymmetrical, illustrated in Fig. 10(a).

7.1 Pairwise Ranging Measurements

Each ranging sensor board has a sounder, which can generate 4kHz fixed frequency acoustic signals, and a microphone that detects the tone produced by the sounder. In our experiment, even after employing robust statistics methods to filter some outliers caused by unexpected events, we find that qualities of pairwise ranging measurements are still various. Most are stable with variances less than 4 cm, while some

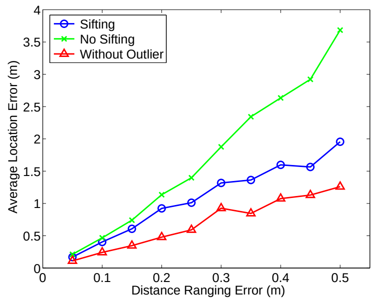


Fig. 7. The impact of ranging error on accuracy.

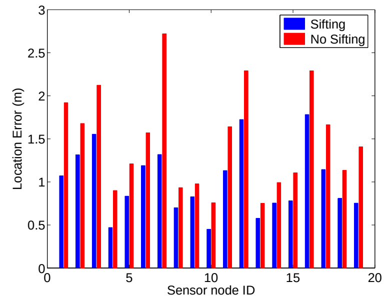


Fig. 8. Localization errors of sensor nodes.

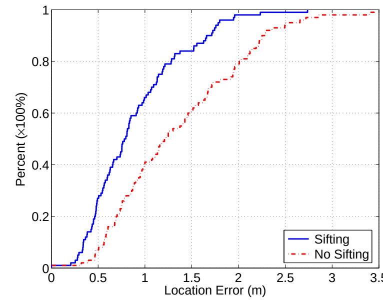


Fig. 9. Empirical cumulative function of location errors.

TABLE I
LOCALIZATION RESULTS. THE REPORTED DATA IS IN CENTIMETERS.

Data Sets	1	2	3	4	5	6
NSNM	15.73	12.43	10.64	10.80	9.78	11.42
SNM	12.52	11.95	8.72	9.54	9.74	12.57
NSM	18.95	26.24	20.79	18.29	14.03	14.70
SM	10.94	16.00	11.18	10.20	13.34	12.21

are highly uncertain with variances as large as 25 cm, two of which are showed in Fig. 10(a) colored red.

7.2 Localization Experiment Results

We have done six sets of experiments using the 5×5 setting with the connectivity graph as Fig. 10(a). For each data set, we have performed the following four types of AHLoS [26] localization:

- **NSNM**: no outlier sifting, no malicious beacon.
- **SNM**: with outlier sifting, no malicious beacon.
- **NSM**: no outlier sifting, with a malicious beacon.
- **SM**: with outlier sifting, with a malicious beacon.

For NSNM and SNM, there are three honest beacons, which broadcast their true locations, and for NSM and SM, we have four beacons, one of which is malicious and fakes its location. As dicussed in Section 2, the proposed method is capable of detecting malicious beacons. Table 1 presents the average location errors of these scenarios. No matter whether there is a malicious beacon, the proposed algorithm largely improves the location accuracy by rejecting outliers during the localization process. We further provide two examples to demonstrate the effectiveness of the proposed outlier detection method. Fig. 10(b) shows the location results of NSNM and SNM, and Fig. 10(c) gives those of NSM and SM, where the node at the top right corner is a malicious beacon.

8. RELATED WORK

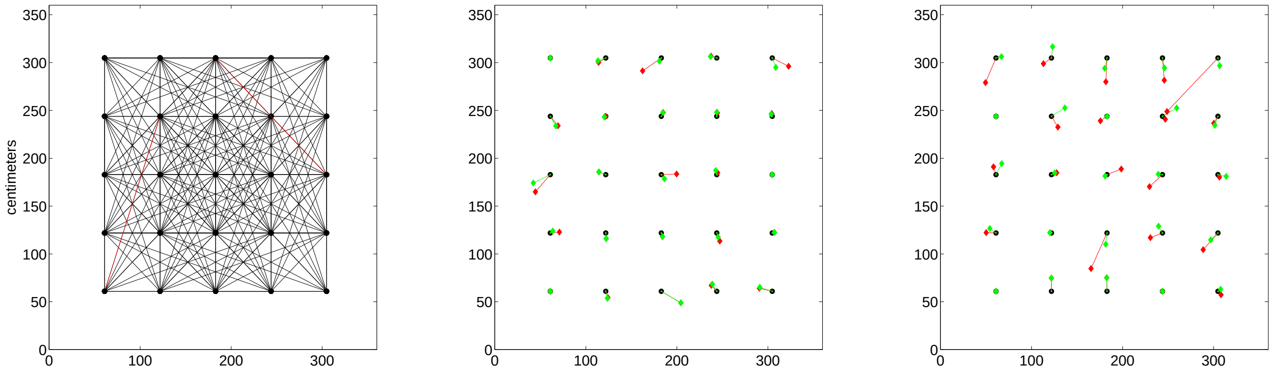
Location information is essential for a wide range of pervasive and mobile applications [29], [18], [22]. In the literature of wireless ad-hoc and sensor networks, many localization methods have been proposed and used in real world positioning systems. In general, we can classify these methods into two major categories: range-based and range-free. Range-based methods usually assume sensor nodes have the capability to measure distances, based on Radio Signal Strength (RSS) [28] and Time Difference of Arrival (TDOA) [26], [25], and/or relative orientations of neighbor nodes, i.e., Angle of Arrival (AOA) [24], [4]. By contrast, range-free methods only use node connectivity and hop-count [9], [16], [17]. To build a full theoretical foundation, some works [5], [7] show that network localizability has a close relationship with graph rigidity theory. Recent work [30] employs the wheel structure to study node localizability.

Because of the inevitability of noise in practice, how to control error accumulation in localization also attracts a lot of research efforts. Error management [21] uses error registries to select nodes that participate in the localization procedure based on their relative contribution to the localization accuracy, and [23] requires "robust quadrilaterals" to prevent large location errors introduced by flip ambiguities. From the perspective of security, outlier-resistant localization has also been studied. Minimum Mean Square Estimation (MMSE) is used to identify and remove malicious nodes in [20], and Least Median of Squares (LMS), an estimator with high breakdown point, is adopted in [19]. A speculative filtering algorithm is designed in [11] to detect phantom nodes. Deriving inspiration from robust statistics, the recent work SISR [14] uses a residual shaping influence function to de-emphasize the "bad nodes" and "bad links" during the localization procedure. Besides requiring dense links, SISR is a centralized algorithm, which may prevent it from applying to large-scale deployments.

Different from those methods that require dense networks, the proposed outlier detection algorithm pays more attention to exploring and utilizing the topological structure, and thus, works properly in the networks with moderate connectivity, which, to the best of our knowledge, can not be realized by existing approaches.

9. CONCLUSION

We have shown that the noisy and outlier distance measurements greatly degrade the location accuracy of existing localization approaches. Hence, outlier detection serves as an essential and prior component for all range-based localization approaches. Based on graph embeddability and rigidity theory, we build the theoretical foundation for identifying outliers and accordingly design a bilateration generic cycles based



(a) Connectivity graph.

(b) NSNM vs. SNM.

(c) NSM vs. SM.

Fig. 10. The experiment network prototype and results, where Black circles are true locations, and red and green diamonds represent the estimates of NSNM (NSM) and SNM (SM), respectively.

algorithm. The results of a network prototype and extensive simulations show that the proposed algorithm largely improves the location accuracy by modestly and wisely rejecting outliers during the localization process.

Acknowledgment

This work is supported in part by NSFC/RGC Joint Research Scheme N_HKUST602/08, National High Technology Research and Development Program of China (863 Program) under Grant No.2007AA01Z180, National Basic Research Program of China (973 Program) under Grant No.2006CB303000, and NSF China Key Project 60736016.

References

[1] M. Badoiu, E. D. Demaine, M. Hajiaghayi, and P. Indyk, "Low-Dimensional Embedding with Extra Information," in Proceedings of ACM SCG, 2004.

[2] A. Berg and T. Jordan, "A Proof of Connelly's Conjecture on 3-Connected Generic Cycles," *Journal of Combinatorial Theory Series B*, 2002.

[3] B. Berge, J. Kleinberg and T. Leighton, "Reconstructing a Three-Dimensional Model with Arbitrary Errors," in Proceedings of ACM STOC, 1996.

[4] H. Chang, J. Tian, T. Lai, H. Chu, and P. Huang, "Spinning Beacons for Precise Indoor Localization," in Proceedings of ACM SenSys, 2008.

[5] T. Eren, D. K. Goldenberg, W. Whiteley, Y. R. Yang, A. S. Morse, B. D. O. Anderson, and P. N. Belhumeur, "Rigidity, Computation, and Randomization in Network Localization," in Proceedings of IEEE INFOCOM, 2004.

[6] D. Goldenberg, P. Bihler, M. Cao, J. Fang, B. Anderson, A. S. Morse, and Y. R. Yang, "Localization in Sparse Networks Using Sweeps," in Proceedings of ACM MobiCom, 2006.

[7] D. Goldenberg, A. Krishnamurthy, W. Maness, Y. R. Yang, A. Young, A. S. Morse, A. Savvides, and B. Anderson, "Network Localization in Partially Localizable Networks," in Proceedings of IEEE INFOCOM, 2005.

[8] F. R. Hampel, E. M. Ronchetti, P. J. Rousseeuw and W. A. Stahel, "Robust Statistics: The Approach Based on Influence Functions," Wiley New York, 1986.

[9] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. F. Abdelzaher, "Range-Free Localization Schemes in Large Scale Sensor Networks," in Proceedings of ACM MobiCom, 2003.

[10] B. Hendrickson, "Conditions for Unique Graph Realization," *SIAM Journal of Computing*, 1992.

[11] J. Hwang, T. He, and Y. Kim, "Detecting Phantom Nodes in Wireless Sensor Networks," in Proceedings of IEEE INFOCOM, 2007.

[12] B. Jackson and T. Jordan, "Connected Rigidity Matroids and Unique Realizations of Graphs," *Journal of Combinatorial Theory Series B*, 2005.

[13] D. Jacobs and B. Hendrickson, "An Algorithm for Two Dimensional Rigidity Percolation: The Pebble Game," *Journal of Computational Physics*, 1997.

[14] H. T. Kung, C. K. Lin, T. H. Lin and D. Vlah, "Localization with Snap-Inducing Shaped Residuals (SISR): Coping with Errors in Measurement," in Proceedings of ACM MobiCom, 2009.

[15] G. Laman, "On Graphs and Rigidity of Plane Skeletal Structures," *Journal of Engineering Mathematics*, 1970.

[16] S. Lederer, Y. Wang, and J. Gao, "Connectivity-based Localization of Large Scale Sensor Networks with Complex Shape," in Proceedings of IEEE INFOCOM, 2008.

[17] M. Li and Y. Liu, "Rendered Path: Range-Free Localization in Anisotropic Sensor Networks with Holes," in Proceedings of ACM MobiCom, 2007.

[18] M. Li and Y. Liu, "Underground Coal Mine Monitoring with Wireless Sensor Networks," *ACM Transations on Sensor Networks*, Volume 5, Issue 2, March 2009.

[19] Z. Li, W. Trappe, Y. Zhang, and B. Nath, "Robust Statistical Methods for Securing Wireless Localization in Sensor Networks," in Proceedings of ACM/IEEE IPSN, 2005.

[20] D. Liu, P. Ning, and W. K. Du, "Attack-Resistant Location Estimation in Sensor Networks," in Proceedings of ACM/IEEE IPSN, 2005.

[21] J. Liu, Y. Zhang, and F. Zhao, "Robust Distributed Node Localization with Error Management," in Proceedings of ACM MobiHoc, 2006.

[22] L. Mo, Y. He, Y. Liu, J. Zhao, S. Tang, X. Li and G. Dai, "Canopy Closure Estimates with GreenOrbs: Sustainable Sensing in the Forest," in Proceedings of ACM SenSys, 2009.

[23] D. Moore, J. Leonard, D. Rus, and S. Teller, "Robust Distributed Network Localization with Noisy Range Measurements," in Proceedings of ACM SenSys, 2004.

[24] D. Niculescu and B. Nath, "Ad Hoc Positioning System (APS) Using AOA," in Proceedings of IEEE INFOCOM, 2003.

[25] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan, "The Cricket Location-Support System," in Proceedings of ACM MobiCom, 2000.

[26] A. Savvides, C. Han, and M. B. Strivastava, "Dynamic Fine-grained Localization in Ad-hoc Networks of Sensors," in Proceedings of ACM MobiCom, 2001.

[27] J. B. Saxe, "Embeddability of Weighted Graphs in k-Space is Strongly NP-hard," in Proceedings of the 17th Allerton Conference on Communications, Control and Computing, 1979.

[28] S. Y. Seidel and T. S. Rappaport, "914 MHz Path Loss Rrediction Models for Indoor Wireless Communications in Multifloored Buildings," *IEEE Transactions on Antennas and Propagation*, 1992

[29] Z. Yang, M. Li and Y. Liu, "Sea Depth Measurement with Restricted Floating Sensors," in Proceedings of IEEE RTSS, 2007.

[30] Z. Yang and Y. Liu, "Beyond Trilateration: On the Localizability of Wireless Ad-hoc Networks," in Proceedings of IEEE INFOCOM, 2009.