Sensor Localization in Concave Environments

CHEN WANG and LI XIAO
Michigan State University

In sensor network localization, multihop based approaches have been proposed to approximate the shortest paths to Euclidean distances between pairwise sensors. A good approximation can be achieved when sensors are densely deployed in a convex area, where the shortest paths are close to straight lines connecting pairwise sensors. However, in a concave network, the shortest paths may deviate far away from straight lines, which leads to erroneous distance estimation and inaccurate localization results. To solve this problem, we propose an improved multihop algorithm that can recognize and filter out the erroneous distance estimation, and therefore achieve accurate localization results even in a concave network.

1. INTRODUCTION

It is important to determine sensors' locations because it provides fundamental support for many location-aware protocols [Karp and Kung 2000; Ko and Vaidya 2000; Navas and Imielinski 1997] and applications [Simon et al. 2004]. A simple approach is to infer a sensor's position through GPS, which measures distances from a sensor to multiple reference points in Satellites and calculates the sensor's position through triangulation computation. However, due to the

design constraint of low cost, it is prohibitive to equip GPS receivers in all sensors. A compromised solution is to deploy GPS receivers to a few sensors that are defined as beacons. The rest of sensors infer their positions based on their relative distances to those beacons. Based on this model, the sensor localization problem can be formalized as follows. Given a network graph $G = (V_m \bigcup V_n, E)$, the vertex set V_m defines the beacons set, the vertex set V_n defines the sensor set whose coordinates are unknown, and the edge set E defines all the measurable distances between pairs of vertex (i, j) where $i, j \in V_m \bigcup V_n$. Sensor localization is to recover coordinates of the vertex in the set V_n under the constraints of edge set E and beacon set V_m.

Despite its simple description in mathematics, sensor localization is a challenging task in engineering that imposes tight design constraints on sensor nodes such as low costs, power saving, and small dimensions. Under such constraints, current available distance measurement techniques based on ultrasound can only achieve accurate results over short distances, such that beacons are not globally accessible to all sensors, especially when beacons are sparsely distributed. Consequently, the simple triangulation algorithm cannot be directly applied to locate all sensors because some sensors may not have sufficient beacons available as their immediate neighbors.

To overcome the limitation of short-range distance measurements, multihop based approaches were proposed to infer distances between any pair of sensors (including beacons) by approximating the lengths of the shortest paths to the Euclidean distances. The localization accuracy of multihop based approaches is built on the basis that the Euclidean distances between pairwise sensors can be well approximated by the lengths of the shortest paths. Such an approximation is achievable only when the shortest paths are close to straight lines, which requires that sensor nodes are uniformly and densely distributed in a convex area. Although the uniform and dense distribution can be achieved through controlled deployment, it cannot be guaranteed that sensors are deployed in a convex area. A typical example is in habitat monitoring, sensors are deployed to complex areas such as valleys or rivers (Figure 1), which may have concave shapes. The other scenario is that sensors are deployed in streets of urban areas where sensors may be separated from each other by buildings, which results in concave network topologies. In such cases, the lengths of the shortest paths may not reflect the Euclidean distances correctly, because the shortest paths between some pairwise sensors have to detour along the concave areas (for instance a C shape or S shape shown in Figure 2 and Figure 3) and cannot be close to a straight line no matter how densely sensors are deployed. Since concave scenarios in the real world such as rivers are the combinations of C shapes and S shapes, we choose the C shape and the S shape as the representatives to be discussed in this paper.

Previous work [Shang and Ruml 2004; Lim and Hou 2005] has shown that the localization results of Multidimensional Scaling (MDS) and triangulation are severely corrupted by the large errors of incorrect distance measurements that are distorted by concave shapes. While an intuitive solution is that those distorted distance measurements should be filtered out and not involved in the localization, it is impossible to recognize whether a single distance measurement

Fig. 1.   Sensors are deployed along a river.
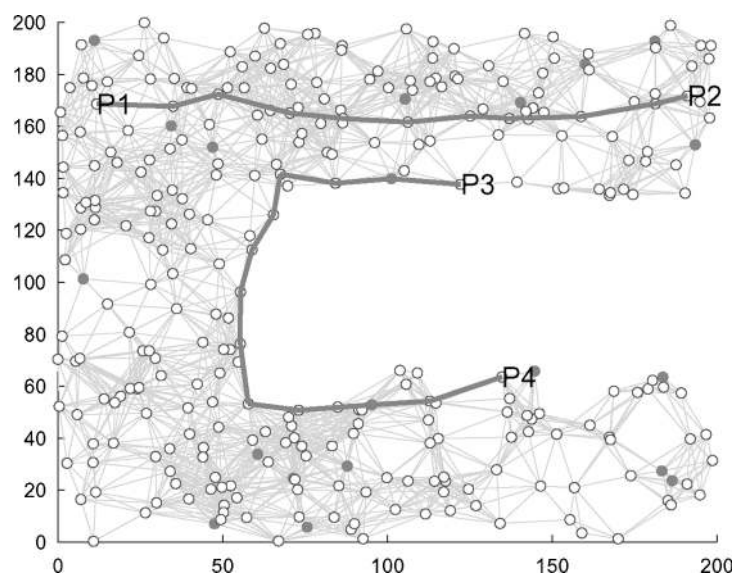
Fig. 2.   Sensors are deployed in a C shape area.

is affected by concave shapes or not due to the lack of a global geometrical view of a network. However, our work shows that it is possible to filter out the impact of distorted distance measurements when multiple distances from beacons to a sensor are available. Our approach is different from previous work in that the distorted distance measurements are directly filtered out and not involved in the localization any more, and therefore their impact on the final localization results are minimized.
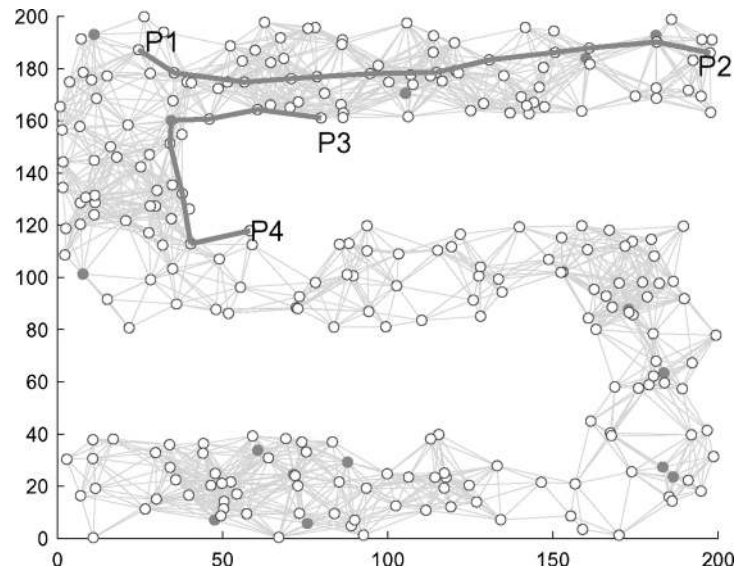
Fig. 3. Sensors are deployed in an S shape area.

The remainder of the article is organized as follows. Section 2 introduces previous work on sensor localization. Section 3 describes the Multihop approaches. Section 4 describes how the distorted distance measurements are filtered out in our improved Multihop algorithm. Section 5 evaluates the performance of our approach through a comparison with previous work. We extend the improved Multihop algorithm to the iterative approach in section 6. We conclude in Section 7.

2. STATE OF THE ART

As described in Section 1, the goal of sensor localization is to recover sensors' coordinates under the edge length constraints. Two steps are involved in this process: 1) the edge lengths are obtained through in-network distance measurements between neighboring sensors; 2) coordinates of sensors are calculated by localization algorithms based on measured distances.

Due to the cost, dimension, and energy design constraints of sensors, currently available in-network distance measurements based on radio received signal strength (RSS) or ultrasound are either error-prone [Savvides et al. 2001] or short-range, which raises challenges in designing an effective localization algorithm. Many algorithms have been proposed to locate sensors by overcoming the limitation of incapable distance measurements. According to whether actual distances are involved in the localization, we divide those algorithms to two categories: area-based algorithms and distance-based algorithms.

2.1 Area-Based Algorithms

In area-based algorithms, actual distance values are not involved in the localization algorithms, and the position of a sensor is estimated by picking up a

point within an area. A typical example is the Centroid approach [Niculescu and Nath 2001], which estimates a sensor's position as the centroid of the polygon formed by beacons that are within the radio transmission range of the sensor.

The APIT [He et al. 2003] approach further improves the localization accuracy by utilizing the redundancy of available beacons, from which 3 beacons are selected out to form a triangle. The APIT approach can determine if a sensor is within the triangle by comparing the beacons' RSS with immediate neighbors. It is possible to obtain multiple triangles from different beacon combinations, and a set of triangles containing the sensor can be selected out. The sensor's position can be pinpointed to the intersection of all the containing triangles, which can be a small area when multiple triangles are involved in the intersection. As a result, the APIT approach can achieve better localization results than the basic Centroid approach.

2.2 Distance-Based Algorithms

Distance-based algorithms estimate sensors' positions from pairwise node distances, which are obtained through the media of radio signals or ultrasound with different measurement accuracy. All the distance-based algorithms recover coordinates of sensor nodes from distance constraints, such that pairwise node distances calculated from recovered coordinates are consistent with correspondent measured distances.

The basic distance-based algorithm is the multilateration, which is applicable when distances from a sensor to multiple beacons are available. To tolerate distance measurement errors, least squares fitting is proposed to minimize the difference between calculated distances and measured distances from the sensor to all available beacons:

$$\widehat{\mathbf{p}} = \arg\min_{\mathbf{p}} \sum (|\mathbf{p} - \mathbf{p}_i| - \widehat{d}_i)^2, \tag{1}$$

where $\mathbf{p}$ is the sensor's position to be estimated, $\mathbf{p}_i$ are beacons' positions, $|\mathbf{p}-\mathbf{p}_i|$ are calculated distances, and $\widehat{d}_i$ are measured distances.

As we mentioned before, in-network distance measurements between pairwise sensors are often short-range. The basic multilateration algorithm may fail for some sensors due to an insufficient number of beacons available as their immediate neighbors. Two possible solutions to overcome the limitation of short-range measurement are recursive approaches and mutihop based approaches.

Recursive approaches [Albowicz et al. 2001; Savvides et al. 2001; Moore et al. 2004] repeatedly apply the basic multilateration algorithm by converting sensors to beacons after their positions are determined. In recursive approaches, "converted" beacons can be propagated from an area that is close to the "starting" beacons to an area where the "starting" beacons are inaccessible. This makes it possible for sensors faraway from "starting" beacons to locate themselves with the aid of "converted" beacons. One problem of recursive approaches is that the localization error may accumulate and the final result may be severely distorted. To minimize the jeopardy of accumulated errors, recursive approaches are usually built on the basis of accurate distance measurements such as time of flight (ToF) of ultrasound. As pointed out in Moore et al. [2004], it

is necessary to avoid large errors such as flip over when beacons are distributed closely to a straight line. An error control mechanism has also been proposed in Liu et al. [2006] to prevent errors from accumulating and propagating in the iterative process.

By approximating the length of the shortest path to the Euclidean distance between pairwise nodes, multihop based approaches [Niculescu and Nath 2001; Nagpal et al. 2003; Savvides et al. 2002, 2003] can infer distances between any pair of nodes in a connected network, hence all beacons are accessible to each sensor. Consequently, each sensor can locate itself through the basic multilateration algorithm. Multihop based approaches can only provide coarse localization results due to their approximate distance estimation. However, multihop based approaches are low-cost solutions because they suggest reusing the communication radio signals to infer pairwise node distances. Their low-cost character makes multihop based approaches ideal candidates for applications that have tight restriction on cost and dimension while less demand on localization accuracy.

When a global view of all pairwise node distances is available, coordinate assignment can be achieved through centralized localization algorithms such as Multidimensional Scaling (MDS) [Shang et al. 2003, 2004; Ji and Zha 2004] and Semidefinate Programming (SDP) [Doherty et al. 2001; Biswas and Ye 2004]. Both approaches use optimization algorithms to search for coordinate assignment such that the distance constraints can be best fit. For instance, the MDS is used to solve the following optimization problem, which minimizes the difference between all calculated distances and measured distances:

$$\widehat{P} = \arg\min_{P} \sum_{i,j \in V} (|\mathbf{p}_i - \mathbf{p}_j| - \widehat{d}_{ij})^2.$$

Here P is the position matrix that contains all the sensors' positions that need to be estimated. $|\mathbf{p}_i - \mathbf{p}_j|$ is the calculated distance, $\widehat{d}_{ij}$ is the measured distance between sensor i and j, and V is the vertex set representing all sensors. It is notable that the multihop approaches are also suggested in MDS [Shang et al. 2003] to infer distances between any pair of sensors since the classic MDS requires distance knowledge of all pairs of nodes. The MDS approach is further improved by the MDS-MAP(P) [Shang and Ruml 2004], which uses the MDS algorithm to locate sensors within small areas (less than two hops), and then patches the small maps together to get sensors' full positions.

Both the multilateration and MDS are built on the basis of least squares fitting, which fits the calculated distances to measured distances. The least squares fitting algorithm is based on the belief that all the distance measurements are close to their true values and have equal error distributions. However, when sensors are deployed in a concave environment, some of the distances estimated by the multihop based approach may deviate far away from their true values because the shortest paths have to detour along the concave shapes and deviate from straight lines. If we still use the least squares fitting algorithm to equally fit all the measurements, the final positioning results will be deteriorated by those "bad" distance measurements. Two approaches have been proposed to address this issue in multihop algorithms.

One approach suggested in Shang and Shi [2004], is to use the four nearest beacons instead of using all of them when performing multilateration localization. This is based on the observation that the distances from a sensor to the nearest beacons will be less affected by the concave shapes. However, it is still possible that the shortest path to the nearest beacon is affected by concave shapes. We will compare this approach with our approach in detail in the coming sections.

Proximity-distance map (PDM) is proposed in Lim and Hou [2005] to approximate lengths of the shortest paths to Euclidean distances correctly in anisotropic networks. In PDM, each sensor is assigned a coordinate in M-dimensional embedding space defined by the lengths of the shortest paths from sensors to all M beacons:

$$\mathbf{p}_i = [p_{i1}, \ldots p_{iM}]^T,$$

where p_{ij} is the length of the shortest path from the ith sensor to the jth beacon. The objective of PDM is to find out sensors' coordinates in M-dimensional embedding space defined by Euclidean distances from sensors to all M beacons:

$$\mathbf{l}_i = [l_{i1}, \ldots l_{iM}]^T,$$

where l_{ij} is the Euclidean distance from the ith sensor to the jth beacon. When the Euclidean distances from a sensor to all the beacons are available, the multilateration such as least squares fitting can be used to calculate a sensor's location based on those Euclidean distances. PDM assumes there exists a linear transform between $\mathbf{p}_i$ and $\mathbf{l}_i$ such that $\mathbf{l}_i = T\mathbf{p}_i$. The linear transform T can be learned from beacons, where beacons' coordinates of both $\mathbf{p}_i$ and $\mathbf{l}_i$ are known.

The intuition behind PDM is that the topology character of the entire network can be well represented by the beacons, since they are uniformly distributed in the network. Therefore, the linear transform T learned from beacons can be also applied to other sensors to transform their coordinates $\mathbf{p}_i$ to coordinates $\mathbf{l}_i$ defined by Euclidean distances. In this paper, we are trying to improve the least squares fitting algorithm in the multihop scenario. It is possible to apply our improved Multihop algorithm based on the Euclidean distances obtained by the PDM.

3. EXISTING MULTIHOP APPROACHES

In this section we will introduce existing multihop approaches and explain why they fail to accurately locate sensors in concave areas. To facilitate our discussion, we define symbols in the following table.

Symbol	Definition
$\mathbf{p}$	position of the sensor to be estimated
$\|\mathbf{p} - \mathbf{p}_i\|$	Euclidean distance calculated from a sensor's position $\mathbf{p}$ to a beacon's position $\mathbf{p}_i$
$\widehat{d}_i$	measured distance between a senor and the ith beacon
d_i	true Euclidean distance between a sensor and the ith beacon

3.1 Multihop Algorithm: Distance Fitting Approach

The key idea of multihop approaches is to discover a sensor network's geometry structure from its communication network topology. In multihop approaches, a sensor network is viewed as a connected graph $G = (V, E)$, where V is the vertex set representing sensors and E is the edge set representing links between a pair of sensors that are within radio transmission range. Multihop approaches infer the distance between a pair of sensors by approximating the length of the shortest path to the Euclidean distance. The length of the shortest path between vertex m and n is calculated as $L_{mn} = \sum l_i$, where l_i are the lengths of intermediate edges included in the shortest path. The value of l_i can be inferred from RSS, which attenuates exponentially when the transmission distance is increased.

In the cases where RSS value is not available, multihop approaches infer the length of the shortest path from the average length per hop, which can be sampled by beacons as follows.

(1) Distances D_{ij} between any pair of beacons can be evaluated from their known coordinates.
(2) The number of hops H_{ij} of the shortest path between pairs of beacons can be inferred from the Dijkstra or Distance Vector algorithm.
(3) The average length per hop to the ith beacon can be calculated as

$$h_i = \frac{\sum_{j \in V_m} D_{ij}}{\sum_{j \in V_m} H_{ij}},$$

where V_m is the beacon set.

When the average length per hop is available, the length of the shortest path from a sensor to the ith beacon can be calculated as

$$L_i = h_i \times H_i,$$

where H_i is the number of hops of the shortest path from the sensor to the ith beacon.

The accuracy of multihop approaches is built on the assumption that the shortest path between a pair of sensors is close to a straight line, which is possible as long as the following assumptions hold:

(1) Sensors and beacons are densely and uniformly distributed.
(2) The network is maintained as a connected graph.
(3) The deployed area has a convex shape.

The uniform and dense distribution of sensors can be achieved through a carefully controlled deployment. How to maintain *k-connectivity* of a network by selecting a proper transmission power is well studied in Li and Hou [2004], and Li et al. [2003]. Therefore, it is not difficult to hold the first two assumptions. However, the third assumption cannot be guaranteed when sensors are deployed in areas full of obstructions.

As we pointed out before, the concave shape has severe impact on distance estimation of multihop approaches. Although a good approximation between the
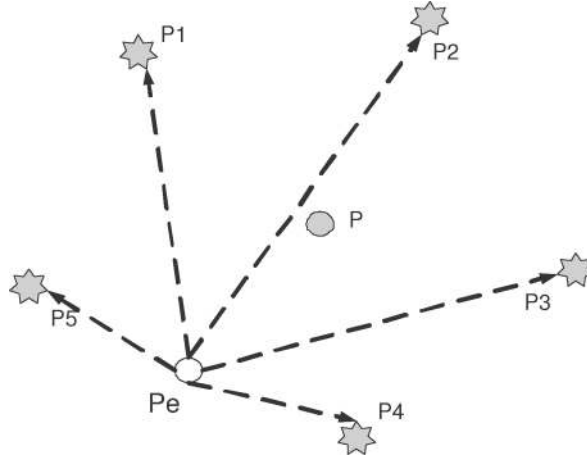
Fig. 4. The estimated position Pe is push away from the true position P by the *incorrect* distance measurement between P and P_2.

length of the shortest path and the Euclidean distance can still be achieved in some scenarios (for instance P_1P_2 shown in Figure 2), the lengths of the shortest paths may differ significantly from the Euclidean distances between pairwise nodes. This is because the shortest paths may be distorted by the concave area and cannot be close to a straight line. Such an example is shown as P_3P_4 in Figure 2. To distinguish the distorted distance estimation from the rest, we divide the distances estimated by the multihop approach into two categories: one is *incorrect* distance measurements, which are distorted by the concave shape, the other is *correct* distance measurements, which are not affected by the concave shape. Previous work [Shang and Ruml 2004; Lim and Hou 2005] has shown that the localization results of MDS and multilateration are severely corrupted by the large error of *incorrect* distance measurements.

As we discussed before, to offset the inaccuracy of distance measurements, the least squares fitting algorithm is suggested in multilateration to estimate sensors' positions by minimizing the difference between the calculated distances and measured distances as Equation (1). Based on the belief that all the distance measurements $\widehat{d}_i$ have the same error distribution and are close to their true values d_i ($\widehat{d}_i \approx d_i$), the least squares fitting tries to fit each measured distance equally. Thus, the final estimated position is the averaged result of all measurements and the impact of an individual measurement error is reduced. However, the assumption $\widehat{d}_i \approx d_i$ does not hold in a concave area, since the *incorrect* distance measurement has a much larger error than the rest. The final localization result is corrupted by fitting to the *incorrect* distance measurements. An example is shown in Figure 4.

3.2 *n*-Multihop Algorithm: The Four Nearest Beacons

To eliminate the impact of concave shapes, a solution is proposed in Shang and Shi [2004] that uses the 4 nearest beacons instead of all of them. The intuition is that the shortest path from a sensor to the nearest beacon may

be less affected by concave shapes. To facilitate our presentation, we denote this algorithm as the n-Multihop algorithm. The n-Multihop algorithm has two potential limitations.

(1) The shortest path to the nearest beacon does not necessarily mean it is not affected by the concave shape. An example is shown in Figure 3, where the path P_1P_2 is longer than P_3P_4, while the former is less affected by the concave shape and closer to its Euclidean distance.
(2) By only using 4 beacons, some of the good distance measurements are eliminated from the localization, and therefore the redundancy of available beacons is sacrificed.

4. THE i-MULTIHOP ALGORITHM

Now we detail our i-Multihop algorithm below.

4.1 i-Multihop: Upper Bound Approach

To facilitate the discussion, we first assume the in-network distance measurements between immediate neighbors are accurate, thus the mismatch between the shortest paths and straight lines connecting pairwise sensors is the only source of the distance measurement error. We will relax this assumption in later discussion. Based on this assumption, we can have the following observation:

Observation 1.   All the measured distances $\widehat{d}_i$ are no less than their true value d_i ($\widehat{d}_i \geq d_i$), because the length of the shortest path is always longer than the Euclidean distance of the straight line connecting pairwise nodes. Especially, the *incorrect* distance $\widehat{d}'_i$ distorted by the concave shape is much larger than its true value d'_i ($\widehat{d}'_i \gg d'_i$), because the shortest path deviates significantly from the straight line.

Based on the observation above, we model sensor localization in concave areas as described below.

Model 1.   Given a network graph $G = (V_m \cup V_n, E_s \cup E_t)$, the vertex set V_m defines the beacons set; the vertex set V_n defines the sensor set whose coordinates are unknown; the edge set E_s defines all the *correct* distance measurements $\widehat{d}_i \geq d_i$; and E_t defines all the *incorrect* distance measurements $\widehat{d}'_i \gg d'_i$. The objective of sensor localization is to filter out the *incorrect* distance measurements E_t and recover coordinates of the vertex set V_n under the constraints of *correct* measurements E_s and beacon set V_m.

The challenge is how to recognize the *incorrect* distance measurements in the model where *incorrect* distance measurements are mixed together with *correct* distance measurements, which is impossible to achieve by observing an individual distance measurement alone. However, we show that it is possible to eliminate the impact of *incorrect* distance measurements from the final localization result when multiple distance measurements are available. Instead of fitting distance measurements, we use upper bound constraints to locate

sensors as described in the following.

$$\widehat{\mathbf{p}} = \arg\min_{\mathbf{p}} \sum |\mathbf{p} - \mathbf{p}_i|^2 \tag{2}$$

$$\text{subject to } |\mathbf{p} - \mathbf{p}_i| \leq \widehat{d}_i$$

Remark 1. The algorithm described in Equation (2) can filter out incorrect distance measurements and achieve accurate localization results if and only if the number of correct distances is no less than 3.

Based on our assumption $\widehat{d}_i \geq d_i$ we have distance measurement error $\delta_i = \widehat{d}_i - d_i \geq 0$. Without losing generality, for all the distances measurements between the senor and beacons, we assume $\delta_1 \leq \delta_2 \leq \ldots \leq \delta_m \ll \delta_{m+1} \leq \delta_{m+2} \ldots \leq \delta_n$. Here, δ_i ($1 \leq i \leq m$) are small errors of *correct* distance measurements, and δ_i ($m+1 \leq i \leq n$) are large errors of *incorrect* distance measurements.

Since $\widehat{d}_i \geq d_i$, we have $\mathbf{p} \in C_i$, where C_i is the circular region with origin $\mathbf{p}_i$ and radius $\widehat{d}_i$. We define the area of C_i as S_i, which represents the uncertainty of the estimated position $\widehat{p}$. Based on the knowledge of $\widehat{d}_i \geq d_i$, we have $|\mathbf{p} - \mathbf{p}_i| \leq \widehat{d}_i$. The probability of estimated position $\mathbf{p}$ follows the uniform distribution:

$$p(\mathbf{p}) = \begin{cases} 1/S_i, & \text{if } |\mathbf{p} - \mathbf{p}_i| \leq \widehat{d}_i; \\ 0, & \text{otherwise.} \end{cases}$$

When S_i becomes smaller, the probability density of $p(\mathbf{p})$ is increased. Thus, the uncertainty of the estimated position $\widehat{\mathbf{p}}$ is decreased. When multiple distance measurements are available, the true position of node $\mathbf{p}$ should be in the intersection of all circular regions C_i, i.e., $\mathbf{p} \in I = \bigcap_{1 \leq i \leq n} C_i$, and the probability of $\mathbf{p}$ follows:

$$p(\mathbf{p}) = \begin{cases} 1/S(I), & \text{if } \mathbf{p} \in I; \\ 0, & \text{otherwise.} \end{cases}$$

Here, the area $S(I)$ of intersection region I represents the uncertainty of the final estimation result. If $\delta_i \to 0$, $S(I) \to 0$, the estimated position $\widehat{\mathbf{p}}$ can be accurately pinpointed to its true position $\mathbf{p}$.

Let $S_t = \bigcap_{1 \leq i \leq m} C_i$. We have $S(I) \leq S_t$, which means the uncertainty will not be increased when *incorrect* distance constraints are added in the localization. Therefore, the *incorrect* distance measurements will not deteriorate the final localization result. The fundamental reason why the *i*-Multihop: upper bound approach can tolerate the incorrect distance measurement is that its assumption $\widehat{d}_i \geq d_i$ is consistent with the observation $\widehat{d}_i \gg d_i$, while the assumption $\widehat{d}_i \approx d_i$ of the original distance fitting algorithm is inconsistent with the observation $\widehat{d}_i \gg d_i$.

An example of *i*-Multihop: upper bound approach is shown in Figure 5, where the distance measurement between P and P_4 is much longer than its true value, which results in the large circle constraint. However, the *incorrect* distance measurement between P and P_4 has no impact on the final estimation result, since sensor P is tightly constrained by the constrained circular regions of P_1, P_2 and P_3.
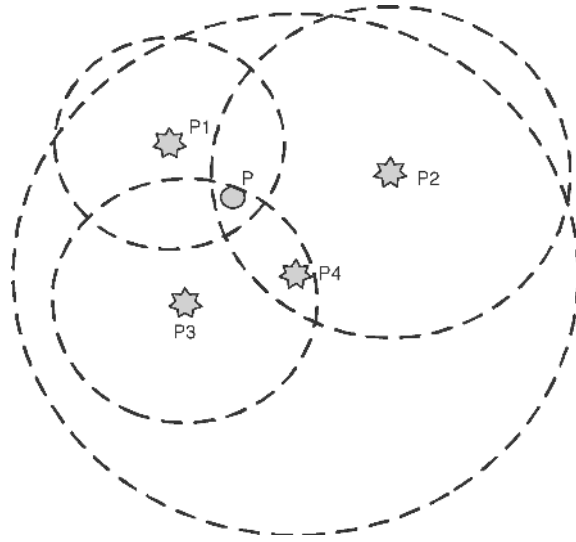
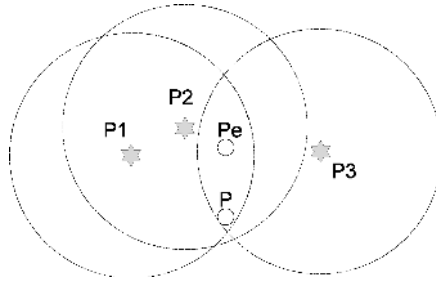Fig. 5.   Sensor P is constrained in the intersection of the circular regions P_1, P_2 and P_3.

Fig. 6.   Collapsed result of the i-Multihop: upper bound approach.

The previous work [Doherty et al. 2001; Wang et al. 2005] also suggests to use the upper bound constraints to locate sensors. Instead of using the local optimization where only distances to immediate neighboring beacons are used, Doherty et al. [2001] use the global optimization of semidefinite programming. However, all approaches suffer the same problem, which requires that beacons are placed on the outside boundary of a deployed area. Otherwise, the estimated positions will collapse toward the center. Such a phenomena has been observed in previous work [Biswas and Ye 2004; Shang et al. 2004]. Below is a formalized description of the problem.

Problem 1.   Given beacons $\mathbf{p}_i$, there exists a polygon region P with all the vertices of $\mathbf{p}_i$. if sensor $\mathbf{p}$ is not within the polygon P ($\mathbf{p} \notin P$), the sensor's position $\widehat{\mathbf{p}}$ estimated by i-Multihop: upper bound approach will collapse toward the P.

An example is shown in Figure 6, where $\mathbf{P} \notin \triangle\mathbf{P}_1\mathbf{P}_2\mathbf{P}_3$. The sensor is constrained by a large intersection area, and therefore has high uncertainty.

Position Pe estimated by the upper bound algorithm of Equation (2) is attracted towards the $\triangle\mathbf{P}_1\mathbf{P}_2\mathbf{P}_3$ and deviates from its true position. This is in contrast to the previous example in Figure 5, where sensor $\mathbf{P}$ is tightly constrained in a small intersection because beacons are distributed around the sensor.

We propose our solution in the following to solve the collapse problem.

4.2 *i*-Multihop: Hybrid Approach

In this section, we assume that sensors are densely distributed, thus the correct distance measurements, which are not affected by the concave shape, are close to their true values. Based on this assumption, we have the observation below.

Observation 2.   All the measured distances $\widehat{d}_i$ are no less than its true value d_i ($\widehat{d}_i \geq d_i$). For all the *correct* distance measurements $\widehat{d}_i$, we have $\widehat{d}_i \approx d_i$. For *incorrect* distance measurements, we have ($\widehat{d}'_i \gg d'_i$).

Based on Observation 2, we model the sensor localization below.

Model 2.   Given a network graph $G = (V_m \bigcup V_n, E_s \bigcup E_t)$, the vertex set V_m defines the beacons set; the vertex set V_n defines the position unknown sensor set; the edge set E_s defines all the correct distance measurements $\widehat{d}_i \geq d_i$ & $\widehat{d}_i \approx d_i$; and E_t defines all the incorrect distance measurements $\widehat{d}'_i \gg d'_i$. The objective of sensor localization is to recover coordinates of vertex set V_n by filtering out the *incorrect* distance measurements E_t and fitting the *correct* distance measurements.

Based on Model 2, we propose the following localization algorithm.

$$\widehat{\mathbf{p}} = \arg\min_{\mathbf{p}} \sum (\widehat{d}_i^2 - |\mathbf{p} - \mathbf{p}_i|^2) \tag{3}$$
$$\text{subject to } |\mathbf{p} - \mathbf{p}_i| \leq \widehat{d}_i.$$

Remark 2.   The algorithm described in Equation (3) can filter out the incorrect distance measurements, and the final localization results will not collapse even when the sensor is not contained in the polygon formed by beacons.

The hybrid algorithm described in Equation (3) combines the advantage of upper bound constraints and distance fitting. First, it uses the upper bound constraints to filter out the impact of *incorrect* distance measurements and pinpoint the estimated position to the intersection constrained by correct distance measurements. Second, it uses the distance fitting to fit *correct* distance measurements, which pushes the estimated position $\widehat{\mathbf{p}}$ toward its true position $\mathbf{p}$ and the final estimated position is not affected by the layout of beacons.

Since $\widehat{d}_i$ are constants, we can rewrite the objective function in Equation (3) as:

$$\widehat{\mathbf{p}} = \arg\min_{\mathbf{p}} -\sum (|\mathbf{p} - \mathbf{p}_i|)^2$$
$$\text{subject to } |\mathbf{p} - \mathbf{p}_i| \leq \widehat{d}_i,$$

which is equivalent to the objective function below.

$$\widehat{\mathbf{p}} = \arg\max_{\mathbf{p}} \sum (|\mathbf{p} - \mathbf{p}_i|)^2 \qquad (4)$$

$$\text{subject to } |\mathbf{p} - \mathbf{p}_i| \le \widehat{d}_i.$$

The intuition of the algorithm above is that the estimated position is pushed far away from beacons, thus towards the outside constrained boundaries. The final optimization result is that the estimated position is limited in the intersection area of circular constrained regions (upper bound constraints) and pushed towards the constrained boundaries (distance fitting), which is the intent of the i-Multihop: hybrid approach.

4.3 Apply i-Multihop Algorithm with Inaccurate Range Measurements

In this section, we will relax the assumption that the in-network distance measurements between immediate neighbors are accurate. Without this assumption, distances estimated from the lengths of the shortest paths will have two error sources: one is the measurement error between immediate neighbors, the other is incurred when the shortest path is not close to a straight line. Under this circumstance, it is possible that some of the distance measurements are less than their true values. This happens in *correct* distance measurements where the shortest path is very close to a straight line and the measured distance of each segment is less than its true values, such that the length of the shortest path is less than the Euclidean distance due to the error accumulation of each segments. Under this circumstance, we need to slightly modify Observation 2 as follows. All the *correct* distance measurements $\widehat{d}_i$ are close to (either larger or less than) its true value d_i ($\widehat{d}_i \approx d_i$) instead of always larger than its true value ($\widehat{d}_i \ge d_i$). In this case, the i-Multihop: hybrid approach with the upper bound constraints will not work correctly, because *incorrect* distance measurements may be less than their true values such that the constrained circular regions cannot intersect with each other. Therefore, no feasible position exists to satisfy all the upper bound constraints and the objective function will not find out a suitable solution.

To solve the problem above, we add slack variables ε_i to the i-Multihop: hybrid approach to get our final version of the i-Multihop algorithm.

$$\widehat{\mathbf{p}} = \arg\min_{\mathbf{p}} \sum (\widehat{d}_i + \varepsilon_i - |\mathbf{p} - \mathbf{p}_i|)^2 + k \sum \varepsilon_i \qquad (5)$$

$$\text{subject to } |\mathbf{p} - \mathbf{p}_i| \le \widehat{d}_i + \varepsilon_i,$$

where k is the weight coefficient which is set to a large value (10^6 in our computation). Due to the large value of the weight coefficient k, the second part $k \sum \varepsilon_i$ has much higher priority to be minimized than the first part, which means the slack variable ε_i has higher priority to be minimized than the difference between calculated distance $|\mathbf{p} - \mathbf{p}_i|$ and the measured distance $\widehat{d}_i$.

The intuition behind the objective function Equation (5) can be described as follows. For those distance measurements that are less than their true values, the slack variables ε_i can increase the upper bound to a minimum extent such that the summary of the measured distance $\widehat{d}_i$ and the slack variable

ε_i is greater than the true value d_i. The consequence is that all the circular region constraints intersect into an nonempty set that contains a feasible solution for the objective function. Through this way, the problem is transformed to Model 2 where all the upper bound constraints are larger than the true distances. Therefore, we can use an approach similar to Model 2 to locate sensors by filtering out *incorrect* distance measurements and fitting *correct* distance measurements.

The algorithm above is based on the assumption that distance measurements between neighboring sensors have small errors and the concave shape contributes the major error source of the distance estimated from the shortest path. Therefore, when the *correct* distance measurement $\widehat{d}_i$ is less than its true value d_i, it is still close to its true value ($\widehat{d}_i \approx d_i$). By selecting a large value k, the *correct* distance measurements $\widehat{d}_i$ will be increased by a minimal value ε_i to form the upper bound, which can be utilized to find the accurate locations of sensors. The performance evaluation in Section 6.5 shows that the i-Multihop algorithm performs well when distance measurements between neighboring sensors have small errors, and its performance can be degraded by large measurement errors between neighboring sensors.

Again, we simplify the objective function (5) to facilitate our optimization computation as follows.

$$\widehat{\mathbf{p}} = \arg\min_{\mathbf{p}} - \sum (|\mathbf{p} - \mathbf{p}_i|)^2 + k \sum \varepsilon_i \qquad (6)$$
$$\text{subject to } |\mathbf{p} - \mathbf{p}_i| \leq \widehat{d}_i + \varepsilon_i.$$

4.4 Average Length per Hop

In the case where distance measurements between immediate neighbors are unavailable, the multihop algorithm estimates distances by multiplying the number of hops of the shortest path with the average length per hop. Here, the average length per hop to the ith beacon can be sampled as follows.

$$h_i = \frac{\sum_{j \in V_m} D_{ij}}{\sum_{j \in V_m} H_{ij}},$$

where D_{ij} is the Euclidean distance from the jth beacon to the ith beacon, and H_{ij} is the number of hops from the jth beacon to the ith beacon.

However, the average length per hop as already calculated can be severely affected by the concave shapes. Because the shortest path between beacons may also be distorted by the concave shape and deviate far away from a straight line, the actual length L_{ij} of the shortest path will be much longer than the Euclidean distance D_{ij} calculated from the coordinates of the beacons. Therefore, the true value of the average length per hop L_{ij}/H_{ij} is much larger than the value estimated from D_{ij}/H_{ij}. This will make the average value estimated by $\sum_{j \in V_m} D_{ij} / \sum_{j \in V_m} H_{ij}$ less than the actual one $\sum_{j \in V_m} L_{ij} / \sum_{j \in V_m} H_{ij}$ because some of the distances D_{ij} are much less than the lengths of L_{ij}.

To solve this problem, we need to filter out the pairwise beacon distances that are distorted by concave shapes, which is achievable by reusing the i-Multihop

algorithm above. Since the calculation of the average length per hop is to infer distances from coordinates, it is a reversed process of the localization algorithm that infers coordinates from distances. Therefore, we can use an idea similar to the i-Multihop algorithm to filter out distorted pairwise beacon distances and calculate correct average length per hop:

$$\widehat{l} = \arg\min_{l} \sum (lh_i + \varepsilon_i - |\mathbf{p}_i - \mathbf{p}|) + k \sum \varepsilon_i$$
$$\text{subject to } |\mathbf{p}_i - \mathbf{p}| \leq lh_i + \varepsilon_i.$$

where l is the average length per hop, h_i is the number of hops of the shortest path between $\mathbf{p}$ and $\mathbf{p}_i$, ε_i is the slack variable, and k is the weight coefficient. The intuitive explanation of the objective function is to find the optimal value of the per hop's average length l which can minimize the difference between the calculated distance $|\mathbf{p}_i - \mathbf{p}|$ and the summary of the measured distance lh_i and the slack variable ε_i under the upper bound constraints. Similar to the i-Multihop algorithm, with the help of the upper bound constraints, only measured distances that are close to their true Euclidean distances will be involved in the optimization, and the *incorrect* distance measurements that are much larger than the true values are filtered out. To facilitate the optimization computation, we simplify the objective function above to the following linear optimization.

$$\widehat{l} = \arg\min_{l} \left(\sum h_i \right) l + k \sum \varepsilon_i \qquad (7)$$
$$\text{subject to } |\mathbf{p}_i - \mathbf{p}| \leq lh_i + \varepsilon_i.$$

5. PERFORMANCE EVALUATION

We compare the i-Multihop algorithm with the original Multihop algorithms and n-Multihop algorithm in this section. Since all these three algorithms use the same beacon message flooding or Distance Vector algorithm to compute the number of hops along the shortest paths, the communication cost of the three is the same. Therefore, we can ignore the details of message communication and focus on the character of their geometry calculation. Such an abstraction can help us to evaluate their performance in the Matlab simulation, where a sensor network is described as a network graph with vertices representing sensor nodes and edges representing the measurable distances between immediate neighbors.

To investigate the impact of concave shapes on the performance of the Multihop algorithms, we used three basic configurations. In the first configuration, 400 nodes were randomly deployed in a $200 \times 200m^2$ square area that has a convex shape. In the second configuration, a portion of sensors in the square area were moved out and the square shape of the network topology was transformed to the C shape as shown in Figure 2. In the third configuration, we transformed the network topology to the S shape as shown in Figure 3.

The following metrics are used in our evaluation.

—*Transmission range R*: the maximum measurement range between neighboring sensors.

—*Estimation error* μ_i: the distance between the estimated position and true position of sensor node i, and $\mu_i = |\widehat{\mathbf{p}}_i - \mathbf{p}_i|$.

—*Average estimation error* $\widehat{\mu}$: the average value of the estimation error μ_i and $\widehat{\mu} = \sum \mu_i / N$, where N is the total number of sensors.

—*Median estimation error*: when estimation error μ_i are ordered from the smallest to largest, the Median estimation error is defined as the one in the middle of the list.

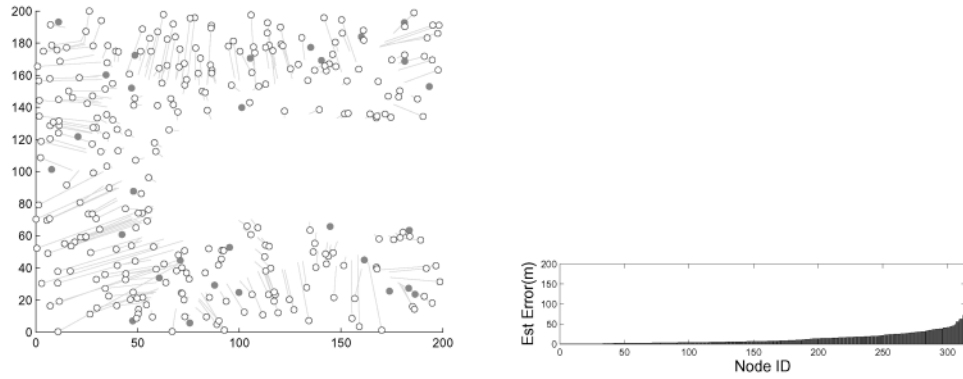## 5.1 Comparison of Multihop, *i*-Multihop: Upper Bound, and *i*-Multihop: Hybrid Approaches

We compared the performance of Multihop, *i*-Multihop: upper bound, and *i*-Multihop: hybrid approaches in the C shape configuration. In this comparison, we assumed distance measurements between immediate neighbors are accurate enough, thus the deviation of the shortest path from the straight line is the only error source of distance measurement. The comparison result is shown in Figure 7, where circles represent the true positions of sensors (solid circles for beacons and empty circles for sensors), and the lines represent the estimation error μ_i. The distribution of estimation error μ_i is also described by the bar graph on the right side, where the sensors are ordered by their estimation error μ_i. The comparison shows that the Multihop approach (Figure 7(a)) has the worst performance, because it tries to fit the distances to all the beacons while some of them are severely distorted by the C shape. Figure 7(b) shows that the performance is improved significantly by the *i*-Multihop: upper bound approach, which uses the distance upper bound to filter out the impact of distorted distance measurements. The performance is further improved by the *i*-Multihop: hybrid approach (Figure 7(c)), which solves the problem that the *i*-Multihop: upper bound approach may have large estimation errors when all the beacons are located on one side of a sensor. This comparison shows that the *i*-Multihop: hybrid approach can achieve the most accurate results among the three approaches. In the following discussion, we only evaluate the *i*-Multihop: hybrid approach, and will not evaluate the *i*-Multihop: upper bound approach any more. For simplicity, we use the short name *i*-Multihop approach to refer to the *i*-Multihop: hybrid approach in the discussion that follows.
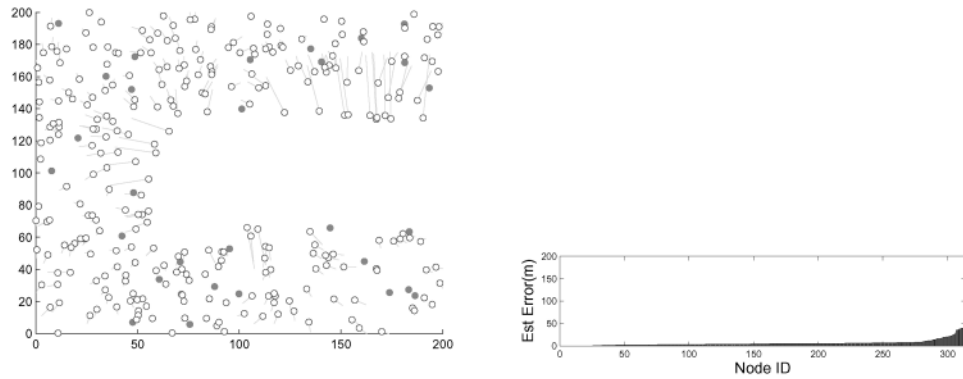
5.2 Average Length per Hop

When distances between immediate neighbors are not available, we can sample the average distance per hop from beacons. However, as we have discussed, the average length per hop calculated in the original multihop algorithm will also be affected by concave shapes because the shortest paths between beacons may deviate far away from straight lines. To eliminate the effect of concave shapes, we used the upper bound constraints in the *i*-Multihop algorithm to filter out the shortest paths, which are distorted severely by concave shapes. We evaluated the estimation accuracy of the average length per hop as follows. First, the average length per hop was estimated in the square shape configuration using the original Multihop algorithm. The experiment was repeated multiple times
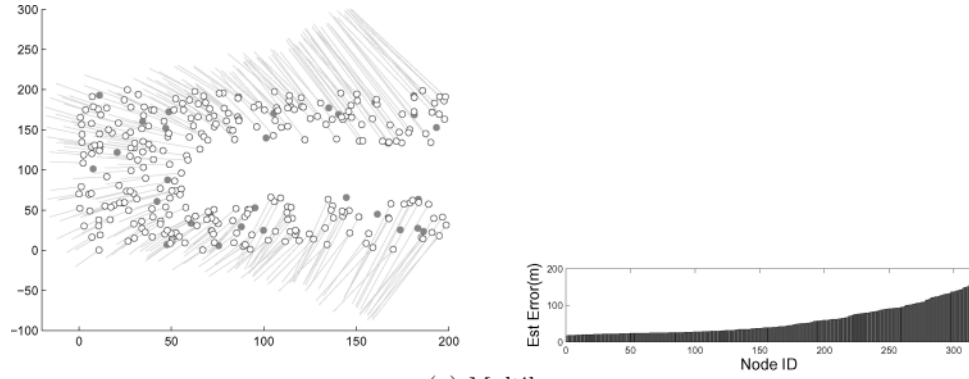
(a) Multihop

(b) *i*-Multihop: upper bound

(c) *i*-Multihop: hybrid

Fig. 7. Comparison of Multihop, *i*-Multihop: upper bound and *i*-Multihop: hybrid approaches in C shape configuration.
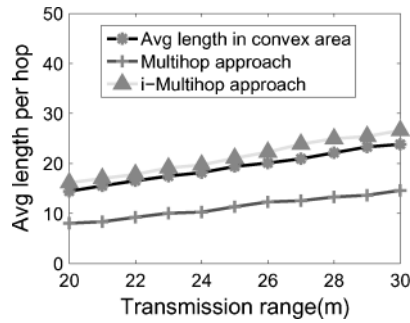
Fig. 8.   Average length per hop.

with different radio transmission ranges. For each radio transmission range, the square shape configuration was transformed to the C shape configuration by removing some sensor nodes, and the average length per hop was estimated again by the i-Multihop algorithm and the Multihop algorithm respectively. Since the C shape is the subset of the square shape, they both should share similar topology properties including the average length per hop. Figure 8 shows that the average length per hop calculated by the i-Multihop algorithm is closer to the average length per hop of the square shape configuration than the one calculated by the Multihop algorithm. This demonstrates that the i-Mulithop algorithm can recover the average length per hop correctly even in concave shapes.

A notable thing on the average length per hop of the shortest path is that it is different from the average distance between immediate neighbors. The reason is explained as follows. To minimize the total number of hops between two sensors, each hop of the shortest path is stretched to its maximum value. The consequence is that the average length per hop is increased with the radio's maximum transmission range, as shown in Figure 8. In the experiment, we repeated the test by varying the radio transmission range while keeping all other network configuration the same as each other, thus the distances between immediate neighbors were the same in each test. However, the experiment shows that the average length per hop is increased when the maximum radio transmission range becomes longer.

5.3 Impact of Concave Shapes

In this section, we focus the performance comparison on the connectivity-based multihop algorithms, where the distances between immediate neighbors are not available and the average length per hop is sampled from beacons. To investigate the impact of concave shapes on the performance of Multihop algorithms, we compared the Multihop algorithm, n-Multihop algorithm, and i-Multihop algorithm in square shape, C shape and S shape configurations.

Figure 9 shows the performance comparison of the three algorithms in the square shape configuration. The Multihop and the i-Multihop algorithms have similar performance, while the performance of the n-Multihop algorithm is much worse than the other two algorithms. The n-Multihop algorithm has the
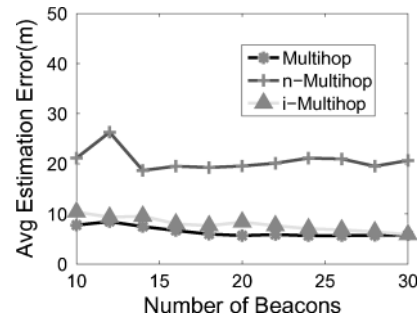
Fig. 9.   Square shape configuration.

worst performance because in connectivity-based multihop algorithms, the distance estimated from the nearest beacon does not guarantee it is the best estimation that is closest to the true Euclidean distance. The performance of the n-Multihop algorithm becomes worse when the maximum transmission range becomes much longer than the average distance between immediate neighbors. As we discussed above, the average length per hop h is different from the average distance d between immediate neighbors, and the former is strongly related to the maximum radio transmission range R. When the density of deployed sensors is fixed, choosing a large transmission range R will improve the network connectivity, which is helpful to sensor localization since the network becomes more rigid with higher connectivity. However, the large transmission range R will lead to the consequence that the average length per hop h is much larger than the average distance d between immediate neighbors. If we estimate distances from the nearest beacon, it is possible that the beacon is within the range of one or two hops. If it is within the one hop range, the true distance to the beacon is close to the average distance d between immediate neighbors. The consequence is that the true distance is less than a single average length per hop h, which is the smallest measurable unit in a connectivity-based Multihop algorithm. This will cause relatively large errors in distances estimated from the sensor to nearby beacons. Such a distance estimation error imposes a limitation on the positioning accuracy of the n-Multihop algorithm. On the other hand, instead of fitting distances to the nearest beacons, the i-Multihop algorithm tries to fit the distance measurements that are closest to their true Euclidean distances, such that the final positioning result of a sensor is closer to its true location.

Figure 10 shows the performance comparison of the three algorithms in the C shape configuration. The Mulihop algorithm performs worst, while the i-Multihop algorithm is the best of the three. The localization accuracy of the i-Multihop algorithm is improved significantly when the number of beacons is increased from 10 to 16, and it eventually converges to a fixed value when the number of beacons is continuously increased. There exists a critical point, because in C shape configuration, when the number of beacons exceeds certain value, most of the sensors can have three beacons that are connected by the shortest paths that are close to straight lines. The Mulithop algorithm has the
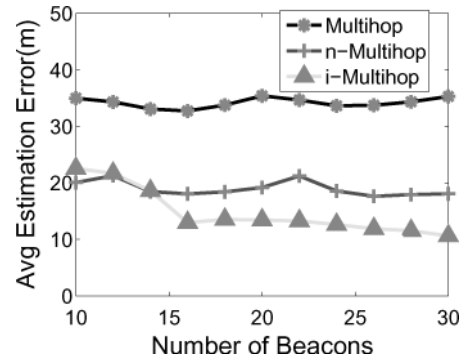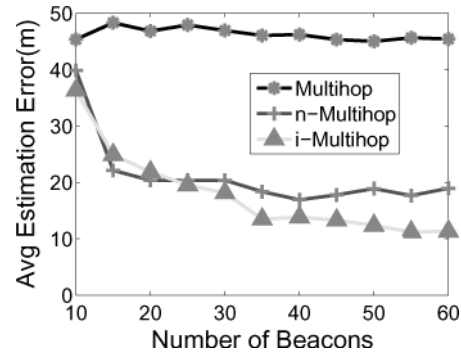
Fig. 10.   C shape configuration.

Fig. 11.   S shape configuration.

worst performance because some of the distance estimation are distorted by the C shape. The *n*-Multihop algorithm does not perform as well as the *i*-Multihop algorithm because its localization accuracy is upper bounded by the granularity of the transmission range, as we have discussed.

Figure 11 shows the performance comparison of the three algorithms in the S shape configuration, which is more concave than the C shape, and some of the distances estimated by the shortest paths deviate further from their true Euclidean distances. The comparison shows that the Multihop algorithm performs much worse than the *n*-Multihop algorithm and *i*-Multihop algorithm, and the *i*-Multihop algorithm has the best performance. In the S shape configuration, the performance of the *i*-Multihop algorithm is increased significantly when the number of beacons is increased to 30, and after that, it eventually converges to a fixed value. The critical point of the number of beacons is larger than the C shape because more beacons are necessary for all sensors to have at least three beacons connected by the close-to-straight-line shortest paths.

From the given comparison we can conclude that the *i*-Multihop algorithm performs best out of the three algorithms, and it can locate sensors in concave environments with positioning accuracy comparable to that of convex environments if the number of beacons reaches the threshold.

6. ITERATIVE APPROACHES

These simulations show that a certain minimum number of beacons is required for the i-Multihop algorithm to achieve sufficient localization accuracy in concave areas. More beacons are demanded when deployed areas become more complicated. This is because a sensor can accurately locate itself only when it is connected to at least three beacons by the close-to-straight-line shortest paths. If only a few beacons are deployed in a concave area such as a C shape or a S shape, it is possible that some of sensors are connected to less than three beacons by the close-to-straight-line shortest paths, which results in large estimation errors. In this section, we show that high accuracy can be achieved with less beacons by iteratively applying the i-Multihop algorithm.

In the iterative approach, a few beacons are deployed as initial beacons. Due to the small number of initial beacons, they may be "visible" to only a small portion of sensors through the close-to-straight-line shortest paths. That small portion of sensors can accurately locate themselves by referring to the initial beacons. After that, those sensors with accurately determined positions "convert" themselves into new beacons by advertising beacon signals. It is possible that the newly added beacons are connected through the close-to-straight-line paths to some sensors that did not have sufficient initial beacons before. By utilizing the beacon signals sent from newly added beacons, those sensors with previously inaccurate estimation results can refine their positions and achieve accurate positioning results. The whole process is recursively repeated until all sensors are accurately located or no more beacons are added.

The challenging of applying the i-Multihop algorithm into the iterative process is how to identify "good" candidates which can accurately locate themselves. In other words, we need to estimate how accurately a sensor can locate itself before we can iteratively apply the i-Multihop algorithm. Due to the absence of a global view of the entire network, a sensor cannot judge if it is connected to at least three beacons through the close-to-straight-line paths. Thus, we cannot identify "good" candidates by simply counting the distances estimated from close-to-straight-line paths. In the following discussion, we propose the upper bound approach to estimate sensors' positioning accuracy.

6.1 Estimate Positioning Accuracy

As we discussed in Section 4.1, suppose a sensor $\mathbf{p}$ is within circular region constraints C_i with origin $\mathbf{p}_i$ and radius $\widehat{d}_i$. Here $\mathbf{p}_i$ are beacons' positions and $\widehat{d}_i$ are estimated distances from $\mathbf{p}$ to $\mathbf{p}_i$. Let $S_t = \bigcap C_i$, which represents the intersection area of all constrained regions C_i. We notice that the position of the sensor can be pinpointed more accurately when the area of S_t becomes smaller. On the other hand, if fewer than three distance measurements are *correct* distance estimations, the intersection area S_t tends to be large. This observation shows that we can identify good candidates for new beacons by checking the size of the intersection area S_t.

However, it is difficult to accurately calculate the area of the intersection S_t due to its irregular shape. Instead, we use the radius of the intersection S_t to estimate the positioning accuracy. As shown in Figure 12, the intersection's
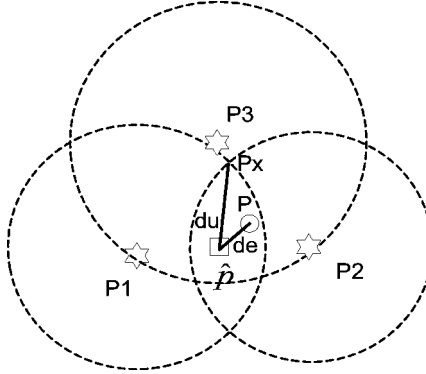
Fig. 12. By its definition, the radius d_u is larger than the estimation error d_e.

radius d_u is defined as the maximum distance between the estimated position $\widehat{\mathbf{p}}$ to any other point $\mathbf{p}_x$ within the intersection area and can be calculated as below.

$$d_u = \max_{\mathbf{p}} |\widehat{\mathbf{p}} - \mathbf{p}_x| \tag{8}$$

$$\text{subject to } |\mathbf{p}_x - \mathbf{p}_i| \leq \widehat{d}_i$$

Here, $\widehat{\mathbf{p}}$ is the position estimated by our i-Multihop algorithm, $\mathbf{p}_i$ are beacons' positions and $\widehat{d}_i$ are measured distances. By the definition of d_u, we have $d_u \geq d_e$, where d_e is the estimation error between estimated position $\widehat{\mathbf{p}}$ and the true position $\mathbf{p}$. In other words, d_u is the upper bound of the estimation error d_e and can be used to estimate the positioning accuracy. Sensors with small estimation errors are identified if they have the small estimation radius d_u.

6.2 Apply i-Multihop Algorithm Iteratively

Based on the radius d_u, we can determine sensors' locations with high positioning accuracy, which makes it possible to apply the i-Multihop algorithm iteratively in concave areas. In the iterative approach, positions of beacon nodes are broadcasted through beacon signals. Beacon signals have counters that are increased by the lengths of hops when they are forwarded between neighboring sensors. The length of the shortest path from a sensor to a beacon can be determined from the minimum counter value among all the received beacon signals sent out by that beacon. Therefore, each sensor can learn beacons' positions and the lengths of the shortest paths to those beacons. Sensors keep listening to beacon signals and refine their positions each time when new beacon signals are received. At the same time, sensors' position accuracy is estimated by their radiuses d_u. When the radius of a sensor is less than the threshold, it will advertise itself as a new beacon, whose beacon signals can be utilized by other sensors to refine their estimated positions. The positioning processes are repeated until all sensors are accurately located or no more beacons are added. Note the iterative process can be achieved by sensors localized computation, which consists of beacon signals listening and position refining. Therefore, it can be implemented in a fully distributed fashion.

While *true* **do**
    if *the sensor is a beacon* **then**
        send out beacon packets which contain the sensor's coordinate;
        break;
    end
    keep listening to beacon packets for a time period;
    if *beacon packets received* **then**
        Update the sensor's position with the newly received beacon packets;
    else
        break;
    end
    Estimate the upper bound of localization error du;
    if $du \leq beacon\_threshold$ **then**
        set the sensor as a beacon;
    end
end

Algorithm 1.　Iterative i-Multihop algorithm.

6.3 Implement Iterative *i*-Multihop Approach in a Distributed Fashion

The pseudo-algorithm of the iterative i-Multihop approach is described in Algorithm 1. In the iterative algorithm, if a sensor is a beacon, it exits the localization algorithm after it sends out beacon packets. If the sensor is not a beacon, the sensor keeps listening to beacon packets. After it receives new beacon packets, the sensor recalculates its location together with the estimation error. If the estimation error falls below the threshold that determines it will become a beacon, the sensor will announce itself as a new beacon and send out beacon packets. The entire algorithm terminates when no new beacons are added into the system. This algorithm is fully distributed because it can be finished by individual sensors without global coordination. The localized operations involve three simple steps: 1) keep listening to beacon packets; 2) update the estimated location and estimation accuracy; and 3) announce itself as a new beacon if the estimation accuracy falls below the threshold.

6.4 Performance of the Iterative *i*-Multihop Algorithm

We evaluate the performance of the iterative i-Multihop algorithm as follows. In the evaluation, 314 nodes were deployed in a C shape area with only 4 initial beacons deployed at the four corners. We assume that distances between neighboring sensors are measurable; thus the mismatch between the shortest path and the straight line is the main source of the distance measurement error. Figure 13 shows both the average estimation error and the median estimation error are decreased along the iterative process when more and more beacons are involved in the localization process. We also note that the median estimation error is always smaller than the the average estimation error. This is because a small portion of sensors have much larger errors than the rest of the sensors. To further illustrate the iterative i-Multihop algorithm, an
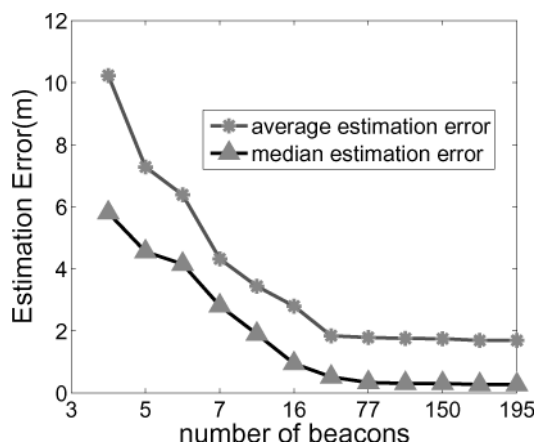
Fig. 13. Positioning accuracy is improved along the iterative process.

example is shown in Figure 14. At the beginning (Figure 14(a)) when only the four initial beacons are used, a number of sensors have large errors because they do not have three beacons visible through the close-to-straight-line shortest paths. Figure 14(b) shows the intermediate status of the iterative process, where some sensors improve their positioning accuracy by referring to newly added beacons. Figure 14(c) shows the final stage of the iterative process, where the majority of sensors can locate themselves accurately. We notice that there are a few sensors that can not locate themselves accurately in the final stage. This is because those sensors do not have good access to beacons even with the help of the iterative approach. Those sensors with large estimation errors can be identified by their radius d_u, thus we can notify upper layer location-aided applications when sensors have large estimation errors and the positioning results are unreliable.

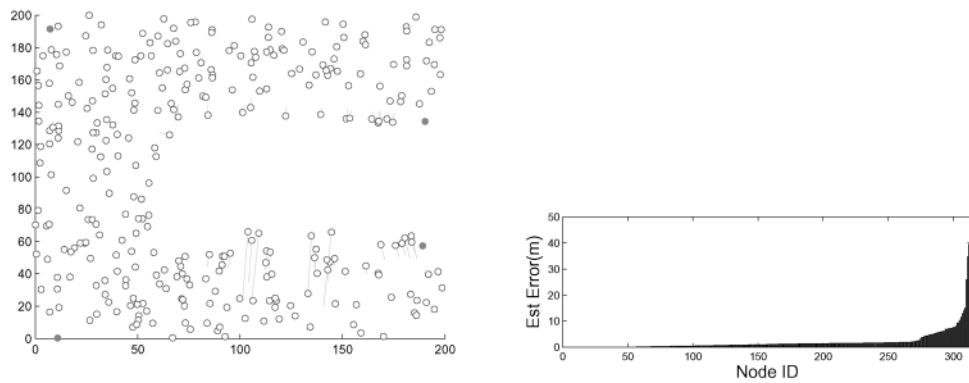The iterative i-Multihop algorithm differs from previously proposed iterative approaches in the following aspects:

—It does not require that initial beacons are adjacent to each other, which is an implicit assumption for other iterative approaches to initiate the iterative process.

—For beacons to be propagated to entire areas, previous iterative approaches usually require dense and uniform sensor distribution. Otherwise, the newly added beacons cannot approach some sensors and the whole iterative process is interrupted. On the contrary, in the iterative i-Multihop algorithm, all sensors can be located as long as they form a connected network. This is because sensors' positions are first estimated from initial beacons, and then refined by newly joined beacons. Here, the iterative strategy is mainly used to improve the positioning accuracy.

—If the positioning accuracy cannot be improved by the iterative process due to awkward beacon layout, the positioning error can be estimated by the radius d_u, which can be sent to upper layer applications together with the positioning
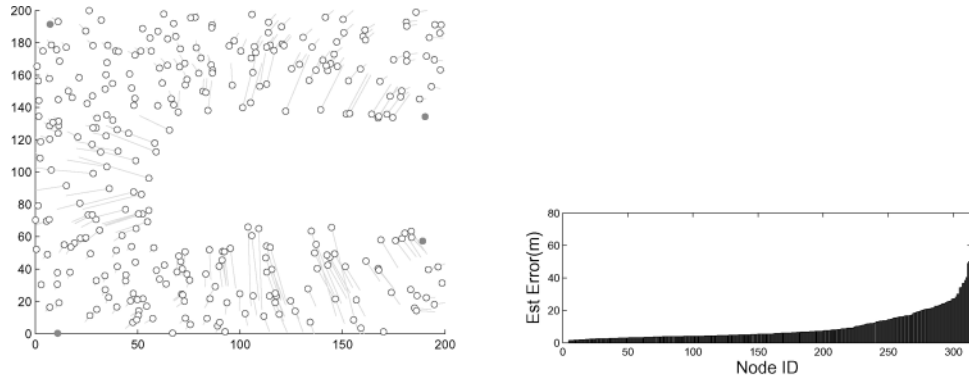
(a) Initial stage
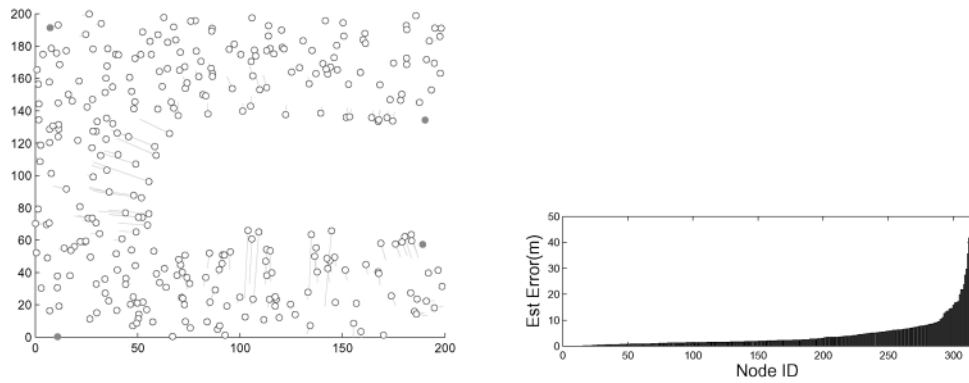
(b) Intermediate stage

(c) Final stage

Fig. 14.   Demo of iterative i-Multihop algorithm.

data. With the notification of positioning accuracy, the upper layer location-aided applications can utilize the location information more intelligently by prudently dealing with the sensors with large estimation errors.

—A potential problem of the previous iterative approach is that the positioning errors may accumulate along the iterative process. This is because the newly joined beacons are not as accurate as initial beacons and may have large estimation errors, especially the flip-over errors discussed in Moore et al. [2004]. The estimation errors of newly joined beacons may accumulate in the following localization process and the final results are severely corrupted. The accumulative errors are minimized in the iterative i-Multihop algorithm because of two reasons: (1) We use the radius d_u to estimate positioning accuracy and beacons are only converted from sensors that can accurately locate themselves. (2) In the iterative i-Multihop algorithm, the positioning accuracy is consistently increased because sensors' estimated positions are updated by new beacon signals only when their estimation accuracy is improved, that is, smaller radius d_u can be achieved.

6.5 Comparison between Iterative i-Multihop Algorithm and MDS-MAP

We further evaluate the performance of the iterative i-Multihop algorithm by comparing it with the MDS-MAP(P) algorithm [Shang and Ruml 2004]. The basic idea of the MDS-MAP(P) algorithm is to first locate sensors within a small area (usually less than two hops range), which produces multiple small pieces of maps comprising relative coordinates of sensors. Since adjacent maps share the same edges, they can be patched together. This process can be iteratively continued until all the maps are patched together to form a single localization map comprising the relative coordinates of all sensors. With the knowledge of a few beacons, the map can be rotated and scaled such that sensors' relative coordinates can be converted to their physical coordinates. In the above process, since MDS-MAP(P) only uses distances within two hops, distance measurements inferred from the shortest path are less likely to be affected by the concave shape. As a result, the MDS-MAP (P) can also achieve good localization results when sensors are deployed in a concave environment. We have conducted intensive simulations to compare these two algorithms and summarize their advantages and disadvantages as follows.

We first compared their localization accuracy under different network connectivity. We deployed 241 sensors in a C shape area with the longest edge of 200m. By varying the measurement range from 25m to 45m, we got different network connectivity from 12 to 36. We introduced random measurement error between neighboring sensors, which follows the Gaussian distribution with mean of 0 and standard deviation of 5%R, where R is the maximum measurement range between neighboring sensors. 10 beacons are used for both algorithms. The comparison results are presented in Figure 15, which shows that the two algorithms have similar performance. Both have large localization errors (around 15%R) when the network connectivity is low, and have small localization errors (close to 5%R) when the network connectivity is high. The high network connectivity improves the localization accuracy of the i-Multihop
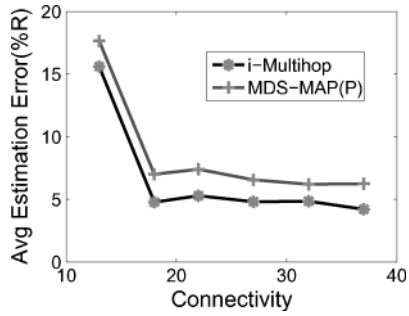
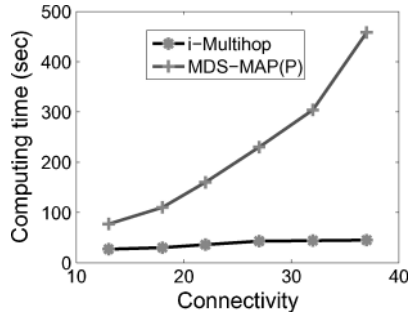Fig. 15. Localization error of i-multiple and MDS(P) under different connectivity.

Fig. 16. Localization time of i-multiple and MDS(P) under different connectivity.

algorithm because the shortest path can be close to a straight line in a network consisting of densely distributed sensors. The high network connectivity also improves the localization accuracy of the MDS-MAP(P) because adjacent small maps share more edges and therefore can be patched more precisely.

We further evaluated the localization cost with the metric of computing time. We used the same configuration as above. The two algorithms were run in Matlab 7.0 on a Dell DIMENSION 8250 PC with a 2.4GHz Pentium 4 CPU and 1G RAM. We measured the computation time under different connectivity. The comparison results are shown in Figure 16, which illustrates that the two algorithms have close computing time when the network connectivity is low. However, the i-Multihop algorithm performs faster when the network connectivity is high. Particularly, the i-Multihop algorithm has constant computing time, while the computing time of the MDS-MAP(P) is proportional to the network connectivity. This is because when the network connectivity becomes higher, adjacent maps share more edges, which incurs more intensive computation to patch maps.

We also compared how these two algorithms performs with different number of beacons. The comparison results in Figure 17 shows that a certain number of beacons is required for the i-Multihop algorithm to achieve high localization results, while only a small number of beacons is necessary for the MDS-MAP(P) to obtain good results. The MDS-MAP(P) has less reliance on the number of
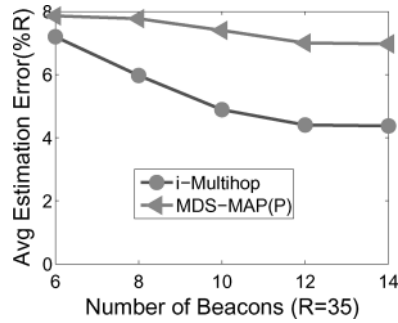
Fig. 17. Localization error of i-multiple and MDS(P) under different beacons.
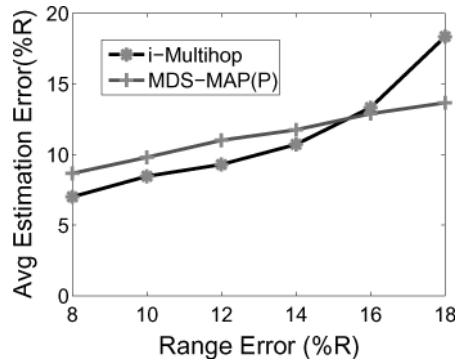

Fig. 18. Localization error of i-multiple and MDS(P) under different measurement errors.

beacons because it computes the relative coordinate first and only uses beacons to convert relative coordinates to physical coordinates.

We evaluated how the measurement error affects localization accuracy by varying the standard deviation from 8%R to 18%R. The comparison results in Figure 18 show that when the standard deviation is less than 16%R, the *i*-Multihop has better localization accuracy than the MDS-MAP(P). However, when the standard deviation become larger, the MDS-MAP(P) performs better. The MDS-MAP(P) is more resilient to measure error because it only uses distances within two hops. In contrast, the *i*-Multihop algorithm uses distances along multihop, and the measurement errors can accumulate to a large value in this case. Based on this comparison, we suggest that distance measurement with ultrasound is more suitable for the *i*-Multihop algorithm that has short measurement range, but high measurement accuracy (usually less than 5%R).

7. CONCLUSION

In this article, we investigate the Multihop algorithms in concave environments, where some of the distances estimated from the shorted path significantly deviate from the true Euclidean distances. While previous approaches cannot perform well due to distorted distance estimation, our proposed *i*-Multihop can

filter out the impact of distance measurements with large errors and achieve sufficient localization accuracy. Therefore, we extend the original Multihop algorithm to concave environments, which is necessary in many cases when sensors are deployed in either urban areas or wild habitats. We also show that the i-Multihop algorithm can be iteratively applied to achieve accurate localization results with less beacons. Moreover, the i-Multihop algorithm can estimate the positioning accuracy, which provides a good basis for upper layer application to intelligently utilize the location data.

REFERENCES

ALBOWICZ, J., CHEN, A., AND ZHANG, L. 2001. Recursive position estimation in sensor networks. In *Proceedings of the International Conference on Network Protocols*.

BISWAS, P. AND YE, Y. 2004. Semidefinite programming for ad hoc wireless sensor network localization. In *Proceedings of the International Symposium on Information Processing in Sensor Networks*.

DOHERTY, L., PISTER, K. S. J., AND GHAOUI, L. E. 2001. Convex position estimation in wireless sensor networks. In *Proceedings of the Annual Joint Conference of the IEEE Computer and Communications Societies*.

HE, T., HUANG, C., BLUM, B. M., STANKOVIC, J. A., AND ABDELZAHER, T. F. 2003. Range-free localization schemes in large scale sensor networks. In *Proceedings of the Annual International Conference on Mobile Computing and Networking*.

JI, X. AND ZHA, H. 2004. Sensor positioning in wireless ad-hoc sensor networks with multidimensional scaling. In *Proceedings of the Annual Joint Conference of the IEEE Computer and Communications Societies*.

KARP, B. AND KUNG, H. T. 2000. GPSR: Greedy perimeter stateless routing for wireless networks. In *Proceedings of the Annual Internationl Conference on Mobile Computing and Networking*.

KO, Y. AND VAIDYA, N. H. 2000. Location-aided routing (LAR) in mobile ad hoc networks. In *Proceedings of the Annual Internationl Conference on Mobile Computing and Networking*.

LI, N. AND HOU, J. C. 2004. Flss: A fault-tolerant topology control algorithm for wireless networks. In *Proceedings of the Annual Internationl Conference on Mobile Computing and Networking*.

LI, X.-Y., WAN, P.-J., WANG, Y., AND YI, C.-W. 2003. Fault tolerant deployment and topology control for wireless ad hoc networks. In *Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing*.

LIM, H. AND HOU, J. C. 2005. Localization for anisotropic sensor networks. In *Proceedings of the Annual Joint Conference of the IEEE Computer and Communications Societies*.

LIU, J., ZHANG, Y., AND ZHAO, F. 2006. Robust distributed node localization with error management. In *Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing*.

MOORE, D., LEONARD, J., RUS, D., AND TELLER, S. 2004. Robust distributed network localization with noisy range measurements. In *Proceedings of the Conference on Embedded Networked Sensor System*.

NAGPAL, R., SHROBE, H., AND BACHRACH, J. 2003. Organizing a global coordinate system from local information on an ad hoc sensor network. In *Proceedings of the International Symposium on Information Processing in Sensor Networks*.

NAVAS, J. C. AND IMIELINSKI, T. 1997. Geographic addressing and routing. In *Proceedings of the Annual Internationl Conference on Mobile Computing and Networking*.

NICULESCU, D. AND NATH, B. 2001. Ad hoc positioning system (APS). In *Proceedings of the Global Telecommunications Conference*.

SAVVIDES, A., HAN, C., AND STRIVASTAVA, M. B. 2001. Dynamic fine-grained localization in ad-hoc networks of sensors. In *Proceedings of the Annual Internationl Conference on Mobile Computing and Networking*.

SAVVIDES, A., PARK, H., AND SRIVASTAVA, M. B. 2002. The bits and flops of the n-hop multilateration primitive for node localization problems. In *Proceedings of the ACM International Workshop on Wireless Sensor Networks and Applications*.

SAVVIDES, A., PARK, H., AND SRIVASTAVA, M. B. 2003. The n-hop multilateration primitive for node localization problems. *Mobile Netw. and Appl. 8*, 4, 443–451.

SHANG, Y. AND RUML, W. 2004. Improved MDS-Based Localization. In *Proceedings of the Annual Joint Conference of the IEEE Computer and Communications Societies*.

SHANG, Y., RUML, W., ZHANG, Y., AND FROMHERZ, M. P. J. 2003. Localization from mere connectivity. In *Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing*.

SHANG, Y., RUML, W., ZHANG, Y., AND FROMHERZ, M. P. J. 2004. Localization from connectivity in sensor networks. *IEEE Trans. Parall. Distrib. Syst. 15*, 11, 961–974.

SHANG, Y., SHI, H., AND AHMED, A. 2004. Performance tradeoffs of localization methods in ad-hoc sensor networks. In *Proceedings of the IEEE International Conference on Mobile Ad Hoc and Sensor Systems*.

SIMON, G., MARUTI, M., LEDECZI, A., BALOGH, G., KUSY, B., NDAS, A., PAP, G., SALLAI, J., AND FRAMPTON, K. 2004. Sensor network-based countersniper system. In *Proceedings of the Conference on Embedded Networked Sensor System*.

WANG, C., XIAO, L., AND RONG, J. 2005. Sensor localization in an obstructed environment. In *Proceedings of the International Conference on Distributed Computing in Sensor Systems*.