Probabilistic Region-based Localization for Wireless Networks [*]

Feng Wang, Lili Qiu, and Simon S. Lam
University of Texas at Austin

Abstract

Determining the physical location of wireless nodes is important to a wide variety of applications. In this paper, we propose a series of probabilistic region-based localization algorithms, including using static grids, segments of grids, and dynamic meshes. These algorithms provide a wide range of trade-off between accuracy and cost, making them suitable for different types of networks, such as sensor networks and mesh networks. Furthermore, we propose several techniques to extract and leverage additional information on location constraints, which is shown to significantly improve the localization accuracy and can be applied to other localization schemes. Finally we develop techniques to enhance robustness of localization, and show that the enhanced scheme can achieve high accuracy even in the presence of significant measurement errors.

I. Introduction

Determining the physical location of wireless nodes is important to a wide variety of applications, ranging from geographic routing [13, 22] to context-aware applications [15, 16], from habitat monitoring [4] to environment surveillance [2, 28].

A global positioning system (GPS) [9] can be used to obtain location information. But it does not work indoors, and it is also costly to equip every wireless node with GPS. The limitation of GPS has motivated researchers to develop algorithms to infer location using cheap hardware by leveraging network connectivity, signal strength, and angle-of-arrival information [29, 3, 19, 17, 11, 27, 26, 12]. Despite extensive research in the area of localization, the following three topics in localization research require further study, which is the subject of this paper.

First, developing accurate localization algorithms based on only connectivity information is an active research topic. A major factor that determines the effectiveness of the algorithms is how the estimated locations are represented. In many previous studies, the location of a node is estimated as a single point. As shown in [7], there are often many coordinate assignments that satisfy the location constraints derived from an underlying network. Therefore assigning the location of a wireless node to a single point may result in significant error. For example, as described in [10], when a node is constrained to be located at four corners of a region, a single point estimation may place the node at the center, which is misleading. In addition, a single point representation is vulnerable to measurement errors – a small perturbation in measurement data may result in a large difference in the estimated location [18]. The novel approaches, proposed by Galstyan et al. and Guha et al. [8, 10], are to represent the estimated location as a region that consists of all points satisfying the location constraints. Such a region-based representation has the potential to yield higher accuracy.

Motivated by [8, 10], we also use a region to represent a node's estimated location. To achieve even higher accuracy, we propose a probabilistic localization approach. In this approach, each node derives a probability distribution over a set of cells that it can possibly reside in. Every cell is associated with a probability about the likelihood that it contains the true position of the node. Furthermore, we propose two techniques to reduce computation cost. The first technique combines cells into segments, which significantly reduces computation cost with a moderate increase in localization error. The second technique is called probabilistic dynamic mesh-based localization (PDM). It uses a mesh generator to partition a region into a mesh, and represents the estimated location of a wireless node as a set of mesh cells. It iteratively refines the estimated location using location constraints extracted from the underlying network. It achieves high accuracy by deriving the probability distribution of a node's position over the region. It achieves reasonable cost by adaptively changing the mesh cell size using DistMesh [5], which is an efficient way to generate an unstructured triangular and tetrahedral mesh to cover a region.

Second, localization accuracy relies heavily on the amount of available information about location constraints. For example, as shown in [6], there is a fundamental limit in localization accuracy using commodity 802.11 hardware. To further improve accuracy, additional information on location constraints is necessary. In this paper, we propose the following ways to obtain and leverage additional information: (i) using network connectivity under different transmission power levels, (ii) using knowledge of whether two nodes can sense each other's carrier, which can be measured empirically as shown in [1], (iii) using layout maps, and (iv) using more powerful anchor nodes (e.g., the anchor nodes can not only extract distance constraints for its neighbors, but also obtain the approximate angles). We also evaluate the benefit of each type of such additional information.

Third, the *robustness* issue in localization has received little attention, even though robustness is essential to the success of any localization scheme since we cannot expect that measurements are always accurate. Erroneous measurement reports may arise from measurement errors, loss of measurement data, and hardware/software problems. Our probabilistic region-based localization provides

[*] Feng Wang and Simon Lam are supported in part by National Science Foundation grants ANI-0319168 and CNS-0434515.

a natural mechanism to handle measurement errors – the probability computation can take into account of the extent to which the location constraints are satisfied. In this way, a mesh cell that is inconsistent with most location constraints is assigned a low probability and pruned out, whereas a mesh cell satisfying most location constraints (but not necessarily all the constraints) will still be retained.

In summary, while localization has been an extensively studied subject, our approach has the following three novel contributions. First, we develop probabilistic region-based localization algorithms, including using static grids, dynamic meshes, and segments of grids. These algorithms provide a wide range of trade-off between accuracy and cost. For example, the segments-based approach yields low cost and high accuracy, and is well suited for networks formed by less powerful nodes, such as sensor networks. In comparison, the PDM achieves a higher accuracy at a higher cost, making it suitable for networks formed by more powerful nodes, such as mesh networks. Second, we propose several techniques to extract and leverage additional information on location constraints. The additional information can be applied to both our and others' localization schemes. Our results show that the additional information can significantly improve localization accuracy. Third, we develop techniques to enhance robustness of localization, and show that the enhanced algorithm can tolerate significant errors from measurement data.

The rest of the paper is organized as follows. In Section II, we overview the related work. We propose the probabilistic region-based localization algorithms in Section III, and evaluate the performance in Section IV. In Section V, we present and evaluate the following two extensions: we further improve accuracy by taking advantage of additional information, and enhance the robustness against erroneous measurements by leveraging our probabilistic-based approach. Finally we conclude in Section VI.

II. Related Work

Localization has been extensively studied due to its great importance. We broadly classify previous work into the following four areas: (i) localization schemes in single-hop wireless networks (e.g., WLAN), (ii) localization schemes in multihop wireless networks, (iii) analysis of the fundamental limitations of localization schemes, and (iv) controlling node placement to ease localization.

Localization in a single hop wireless network: In the area of localization for single-hop wireless networks, a number of interesting approaches have been proposed. For example, Active badge [29] locates users by having them wear infrared badges that transmit unique identifiers. RADAR [3] relies on signal strength measurement gathered at multiple receiver locations to triangulate users' locations. Cricket [21] uses the difference between the arrival time of radio and ultrasound signals to estimate distance. VORBA [19] determines location based on angle of arrival measurements from 802.11 base stations. In [17], Madigan *et al.* develop a Bayesian hierarchical model that simultaneously locates a set of wireless clients (as opposed to localizing one user at a time). Refer to [11] for a nice survey on the location systems for single hop wireless networks.

Localization in a multihop wireless network: Localization in multihop environments is even more challenging, since nodes are often multiple hops away from anchor nodes, thereby increasing the uncertainty in location.

A number of interesting localization algorithms have been proposed for such networks. For example, the authors in [24] develop a distributed localization approach that iterates through a two-phase process: ranging and estimation. During the ranging phase, each node estimates its distance to its neighbors, whereas during the estimation phase, nodes use the ranging information and their neighbors whose positions have been determined to estimate their own location. In [25], the authors enhance the previous approach by formulating the problem as a global non-linear optimization problem. This limits error accumulation in [24]. Shang *et al.* in [27] propose to use multidimensional scaling (MDS) to determine location in a centralized fashion. The localization accuracy is limited partly because it cannot handle violation of triangulation (especially for irregular-shaped networks). Later they develop a distributed version of MDS-based approach in [26]. It is shown to out-perform the original centralized version in irregular-shaped networks by ignoring the distance information among nodes that are far-apart. In [18], the authors present algorithms that use robust quadrilateral for localization. Their approach finds sets of four nodes that are fully connected, and localizes the fourth node based on the positions of the other three nodes. To prevent error accumulation, the four-node set needs to satisfy robust quadrilateral conditions. This improves accuracy at the cost of leaving some nodes unlocalized. In [12], the authors propose a sequential Monte Carlo localization method to enhance the accuracy of localization by exploiting mobility. In particular, the approach leverages mobility history to predict possible locations based on previous location samples and its movement, and uses the new connectivity information to eliminate inconsistent location samples.

Unlike most of the previous approaches, which represent inferred locations using points, Sextant [10] develops a novel approach that denotes inferred locations as regions represented by Bezier curves. Such a representation is shown to significantly improve accuracy. Motivated by their approach, in this paper we also use region-based representation. Different from their work, we use a dynamic mesh to represent a region, and derive the probability for a node to reside in each mesh cell. Such a representation enables us to achieve high accuracy and robustness without significant computation cost.

Analysis of limits on localization accuracy: In addition to developing novel localization algorithms, researchers have also analyzed the fundamental limits on localization algorithms. For example, the authors in [6] compare a series of localization algorithms, and find that using commodity 802.11 technology over a range of algorithms, approaches and environments, it is expected to have a median localization error of 10 feet and 97th percentile error of 30 feet. They conclude that these limitations are fundamental and unlikely to be significantly improved without fundamentally more detailed environmental models or additional localization infrastructure. It points out that leveraging additional information is necessary in order to improve the accuracy.

Node placement: Complementary to developing localization algorithms, researchers have also designed algorithms to place nodes to reduce the ambiguity of estimated location. For example, in [11] Ray *et al.* apply the theory of identifying codes to determine the placement of sensors so that each position is uniquely identified by a set of sensors that it can directly communicate with. The authors further extend their algorithms to tolerate errors (e.g., sensor failures). In [7], Eren *et al.* show that a network has a unique localization if and only if its corresponding grounded graph is generally globally rigid. Applying graph-rigidity literature, they develop approaches to constructing uniquely localizable networks, and study the computation complexity of localization. Node placement algorithms are complementary to localization algorithms. The localization algorithms should be applicable even when we do not have the flexibility to alter the graph to make it uniquely localizable.

III. Probabilistic Dynamic Mesh-Based Localization

As mentioned in the previous section, a significant difference between various localization approaches is how the estimated location is represented. To achieve high accuracy and robustness, we adopt a region-based representation, where an estimated location is represented as a region that consists of all points satisfying the location constraints extracted from the underlying network. We further improve the existing work [8, 10] by deriving a probability distribution over the region to reflect the likelihood of the true position. Such probability distribution, combined with an explicitly represented region, provides much richer location information than a single position, and allows us to achieve higher accuracy in face of insufficient information and measurement errors.

Below we first present a probabilistic region-based localization approach. Then we describe two techniques to improve the efficiency of the approach. The first one combines multiple horizontal (or vertical) cells (in an estimated region) into a single segment, which reduces computation cost at the expense of slightly higher error. The second technique is based on a dynamic mesh, where mesh cells are dynamically adjusted according to the size and shape of the region. It can achieve both efficiency and accuracy.

III.A. Probabilistic Region-Based Localization

The probabilistic region-based localization proceeds as follows. First, every node's location is initialized to be the entire space. Then each node extracts location constraints by measuring the connectivity of the underlying network, and propagates these constraints to nodes within a certain hops away. (We use 3 hops in our evaluation.) If angle and received signal strength index (RSSI) measurements are available, they can be used to extract location constraints and processed in a similar way. Based on the constraints reported by other nodes and its own observation, a node estimates its new location by pruning out the sub-regions that are inconsistent with the constraints. For the sub-regions that are consistent with the constraints, a node further computes a probability distribution over them. The approach is run in a distributed way.

Extracting location constraints: To estimate its location, a node first extracts location constraints from the underlying network. Examples of location constraints include "the distance between node i and node j is at most d" (also called distance constraints), and "the angle between line ij and the direction of North is within $[\theta_1, \theta_2]$" (also called angle constraints). Such location constraints can be obtained by measuring network connectivity and angle-of-arrival. In this section, we only consider distance constraints. We will consider angle constraints in Section V.A.

To handle irregular wireless propagation, each wireless node is associated with two separate radii: R and r ($R \geq r$), where R denotes the maximum transmission range the node can reach, and r denotes the minimum transmission range the node can reach [10]. $R \neq r$ arises when the signal propagation is not the same in all directions. When node i can hear node j, we obtain a constraint: $d_{ij} \leq R_j$. This is a *positive constraint*. When node i cannot hear node j, we obtain a constraint: $d_{ij} > r_j$. This is a *negative constraint*.

Next we introduce some more notations. Let LC_{ji} denote a location constraint for node j using node i as a reference point. Let $POS()$ denote a positive constraint, and $NEG()$ denote a negative constraint. Let S_i and S_j be the estimated region of node i and j, respectively.

If node j can hear node i, we obtain a positive constraint: $d_{ij} \leq R$. Then the estimated region of node j can be expressed as:

$$S_j = POS(S_i, R) = \{p_j | \exists\, p_i \in S_i,\ d(p_i, p_j) \leq R\},$$

where $d(p_i, p_j)$ is the distance between two points p_i and p_j. This region is a union of discs that are centered at each point inside S_i with radius R. Similarly if node j cannot hear node i, we derive a negative constraint, and the region of node j is estimated to be

$$S_j = NEG(S_i, r) = \{p_j | \exists\, p_i \in S_i,\ d(p_i, p_j) > r\}$$

.

If there are multiple constraints derived (e.g., by using multiple reference points), the final output is the intersection of the regions from all these constraints. Note that while we use connectivity information to extract location constraints, our approach can easily incorporate other information, such as angle estimation and layout maps, which will be described in Section V.A.

Computing probability: Next we describe how each node i derives a probability distribution P_i over its region S_i. To do so, we partition the whole space into (small) cells, where each cell is a square with a fixed size. A cell is the smallest unit for which we compute probability. Let s be a cell. $P_i(s)$ is the probability that node i is in s. Each location constraint gives a probability distribution over an estimated region. The final relative probability of each cell is the product of the probabilities derived from all constraints (including both positive and negative constraints). We further derive the absolute probability by normalizing the relative probabilities.

Below we show how to derive a probability distribution from one location constraint. Since the probability computation using positive and negative connectivity information is similar, we illustrate the idea by considering only a positive connectivity constraint.

First we describe how to compute probability $P_i(s)$ using an anchor node, a, whose location is known, as a reference point. Using network connectivity, we obtain a distance constraint from a to i as $d_{ia} \leq k * R$, where k is the number of hops between a and i. Therefore S_i is the disc centered at a with radius $k * R$. Since only connectivity information is available, we assume node i's location is uniformly distributed inside the circle. Therefore, for a cell g,

$$P_i(g) = \begin{cases} 0 & \text{if g is outside the circle,} \\ 1/c_1 & \text{otherwise,} \end{cases}$$

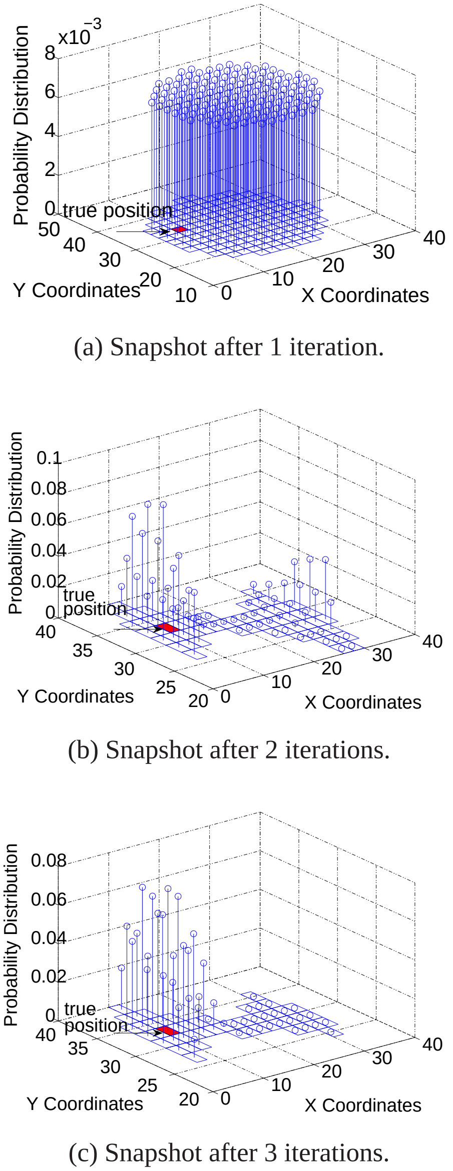
where c_1 is the number of cells inside the circle. (Note that application of negative connectivity information will change the above probability distribution. For example, if a node is 2 hop away from a, the fact that it is not a's immediate neighbor allows us to prune out the area of a circle centered at a with radius r.) To avoid leaving out the true position, a cell is considered "inside" the circle as long as it overlaps with the circle. Consequently, $S_i = \bigcup(g)$ is not exactly the region enclosed by the circle, but the union of all cells considered "inside" the circle. Therefore $1/c_1$ is an approximation since some cells are partially inside the circle. The accuracy of such approximation depends on the cell size. Smaller cell sizes reduce the approximation error at the cost of increasing computation and storage cost.

Next we describe how to compute probability $P_i(s)$ using a non-anchor node (whose location is not known in advance) as a reference point. Consider a node i's neighbor j. For a cell $u_j \subset S_j$, the relative magnitude of its probability is determined by the probability of subregion in S_i that satisfies $d(u_i, u_j) \leq R$. This results in the following:

$$\begin{aligned} P_j(u_j) &= \beta \cdot \frac{\sum_{u_i \subset d(u_i,u_j) \leq R} P_i(u_i)}{\sum_{u_i \subset S_i} P_i(u_i)} && (1) \\ &= \beta \cdot \sum_{u_i \subset d(u_i,u_j) \leq R} P_i(u_i) && (2) \end{aligned}$$

where β is a normalization factor so that $\sum_{u_j} P_j(u_j) = 1$.

Figure 1 shows how a node's estimated location converges. After the first iteration, the region is approximately a circle since this node is a neighbor of an anchor. The probability distribution is uniform over all cells. After the second iteration, the estimated region is refined, with the updated probability distribution and smaller area, by leveraging the constraints from the anchors that are 2 hops away. After the third iteration, the region is reduced further (although the amount of reduction is less than in the second iteration because the constraints from the 3-hop neighbors have less impact on the region than constraints from the 2-hop neighbors). As it shows, the cell containing the true position (marked as the shaded cell) and its surrounding cells have significantly higher probabilities than the remaining region.



(a) Snapshot after 1 iteration.

(b) Snapshot after 2 iterations.

(c) Snapshot after 3 iterations.

Figure 1: Snapshots of a node's estimated location for the first three iterations.

III.B. Enhancing Efficiency

So far we consider using static grids. In this case, the computation cost is determined by the number of cells. If a node's location has high uncertainty due to lack of sufficient location constraints, its estimated region is large, resulting in a large number of cells and hence high computation and storage costs. In this section, we describe two techniques to improve the efficiency of the above localization approach. The first approach reduces the cost by combining horizontally (vertically) contiguous cells into a row (column) segment. The second approach dynamically adapts the cell size so that coarse-grained cells are used when the estimated region is large and fine-grained cells are used when the estimated region is small.

Segment-based localization: One way to reduce the complexity is to combine horizontally (vertically) contiguous cells into a row (column) segment. Since computation using row segments is similar as using column segments, in the following description we focus on using row segments. The width of each segment is fixed, but the length is variable. A row segment is specified by a 3-tuple, (y, x_1, x_2), where (x_1, y) is the left end and (x_2, y) is the right end. Each estimated region is represented as a set of row segments. We want to calculate the probability of each row segment containing the true position. Now the complexity is determined by the number of row segments.

Suppose we obtain node i's estimated region and the probability distribution over the region. We calculate its neighbor j's estimated region and probability distribution as follows. The location constraint LC_{ji} is $d_{ji} \le R$. Hence, $S_j = POS(S_i, R)$. Let u_i denote a row segment of i, and u_j denote a row segment of j. The general formula to derive probability is similar to (1). Since a row segment may be significantly larger than a cell, treating partial overlap as complete overlap may result in high error. Therefore we further calculate the fraction of a row segment that satisfies location constraints.

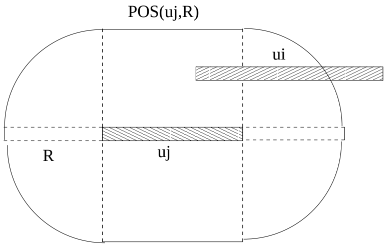


Figure 2: Example of Using Segments

Figure 2 shows an example. u_j is a row segment in S_j. $POS(u_j, R)$ is the region expanded from u_j by R. u_i is a row segment in S_i. u_i is partially in $POS(u_j, R)$. When calculating $P_j(u_j)$, we need to calculate the portion of u_i that is inside $POS(u_j, R)$.

Let $v_i = u_i \cap POS(u_j, R)$. Let $A(S)$ denote region S's size. Assuming uniform distribution within a segment, we have,

$$P_j(u_j) \;=\; \gamma \cdot \sum_{u_i \subset S_i} \frac{A(v_i)}{A(u_i)} \cdot P_i(u_i), \qquad (3)$$

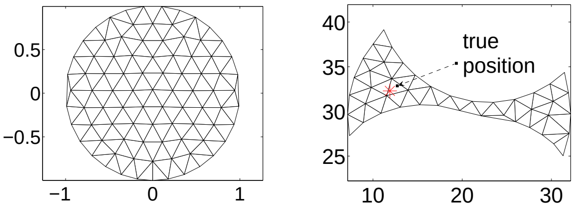
where γ is a normalization factor.

Probabilistic dynamic mesh-based localization (PDM): Combining consecutive cells in one dimension can significantly reduce computation and storage costs. On the other hand, its accuracy depends on how accurately a uniform distribution captures the actual probability distribution over the set of combined cells. When the actual distribution significantly deviates from a uniform distribution, localization accuracy will decrease. To achieve both high accuracy and low cost, we propose an alternative approach that dynamically adjusts the cell size as needed.

At a high level, we use coarse-grained cells when the estimated region is large, and use fine-grained cells when the estimated region is small. To achieve this goal, we leverage mesh generation work developed in the area of computer graphics. We use DistMesh [5, 20] because it can efficiently generate high-quality meshes. DistMesh uses a *signed distance function* $d(x, y)$ to specify a region. The absolute value of $d(x, y)$ is the minimum distance from (x, y) to the boundary of the region, where a negative distance means it is inside the region and a positive distance means it is outside the region. It generates meshes using Delaunay triangulation, and optimizes node locations using a force-based smoothing procedure as described in [5, 20]. It also provides a parameter to control the sizes of triangles.

We apply DistMesh to localize wireless nodes as follows. Each node represents its estimated region using a set of triangular cells. A triangular cell is the smallest unit for which we compute a probability. We control the mesh structure so that each triangle has similar sizes in both dimensions, and the sizes of triangles are adaptive according to the size of the region. It is straightforward to write distance functions for distance constraints and angle constraints. Each node calculates its region based on the measured distance constraints. Given a combined distance function from all location constraints, DistMesh can generate a set of triangular meshes to represent the region that satisfies the location constraints.

Figure 3 illustrates two examples of triangular mesh generated by Distmesh. Figure 3(a) shows the mesh cells for a circle. Figure 3(b) shows the mesh cell that represents the estimated region for the same node as in Figure 1, resulting from subtracting three circles from one circle.



(a) Mesh cells for a circle.

(b) Mesh cells for a node's location.

Figure 3: Triangular mesh generated by Distmesh.

After obtaining its estimated region, a node can derive the probability distribution over the triangles (inside the region) in a similar way as in static grids. Suppose we know the region and probability distribution over the triangles of a given node i. A neighbor j of node i has location constraint $d_{ji} \leq R$, and calculates its region S_j as follows. Let t_i denote a triangle in S_i, and t_j denote a triangle in S_j. We derive the probability associated with t_j by first computing the fraction of t_i satisfying the location constraint, and then weighting the fraction by the probability of node i residing in t_i.

Figure 4 shows an example of deriving probability distribution. t_j is a triangle in S_j. $POS(t_j, R)$ is the region expanded from t_j by R. t_i is a triangle in S_i. t_i is partially in $POS(t_j, R)$. When calculating $P_j(t_j)$, we need to determine what fraction of t_i is inside $POS(t_j, R)$.

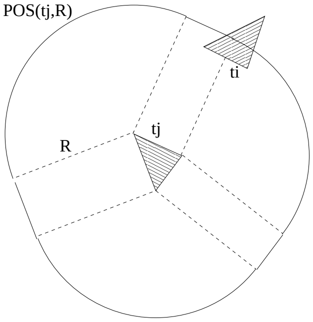


Figure 4: Example of mesh model.

Let $t'_i = t_i \cap POS(t_j, R)$. Assuming uniform distribution within a triangle, we have

$$P_j(t_j) \;=\; \gamma \cdot \sum_{t_i \subset S_i} \frac{A(t'_i)}{A(t_i)} \cdot P_i(t_i), \qquad (4)$$

where γ is a normalization factor.

IV. Performance Evaluation

We evaluate localization schemes using a methodology similar to [27] and [26]. We uniformly place a set of nodes over a 2-dimensional space. We compare different localization schemes while varying the number of nodes (N), the maximum transmission range (R), and the fraction of anchor nodes (A). In this section, our evaluation uses one power level. In section V.C.1, we will further study the effect of power control by varying the number of power levels.

We quantify the localization error using the same method as in [10]. For both Sextant and our approach, we use Monte Carlo sampling to sample 1000 points in a node's estimated region, and pick the one that minimizes the average error to other sampled points inside the region. The localization error is then calculated as the distance from this point to the node's true position.

However, there is a difference in choosing sample points between Sextant and our approach. Sextant uniformly samples points inside a region, whereas in our approach the number of sample points in a cell is proportional to its probability. As we will show, the probabilistic-based approach can significantly improve the localization accuracy.

Effects of the number of nodes Figure 5 shows the cumulative distribution of position errors for $N = 50$, $R = 12.5$, and $A = 10\%$. The size of the space is 50x50.

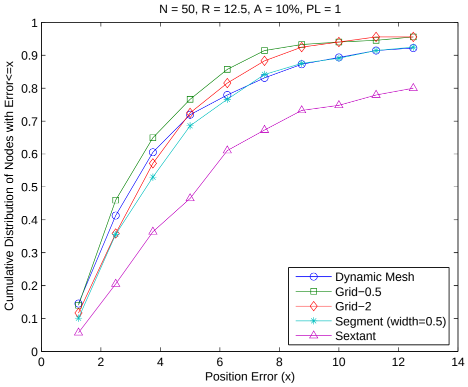


Figure 5: Probability distribution improves localization accuracy (50 nodes)

We make the following observations. First, PDM significantly out-performs Sextant. For example, the percentage of nodes achieving $\leq 30\% * R = 3.75$ errors in dynamic mesh is 60% compared to 36% in Sextant. This is because in Sextant different points inside a region are treated equally, whereas PDM leverages the derived probability distribution over the region. Second, as we would expect, the static grid approach using 0.5x0.5 grids yields smaller errors than using 2x2 grids. Third, the dynamic mesh approach performs better than the static grid approach with 2x2 grids at the lower end of the errors ($\leq 10\%, 20\%, 30\% * R$ errors). Fourth, combining cells into segments with width 0.5 (denoted as "Segment (width=0.5)" in the figure) yields slightly larger errors than using static grids or dynamic meshs, but still out-performs Sextant by a significant amount.

Sextant	Grid-2	Grid-0.5	PDM	Segment
1.98	1.225	56.67	6.82	3.66

Table 1: Average running time in seconds using a 1200 MHz UltraSPARC-III+ processor with 16GB memory.

Table 1 summarizes average running time of different algorithms. As we can see, the running time of static grid approach decreases with increasing grid size. When the grid size is as large as 2x2, the static grid approach takes less time than Sextant. In all other three schemes, the running time is longer than Sextant. (Note that Sextant code is from its original authors and it is implemented in JAVA, while all of our approaches are implemented in MATLAB. We expect the running time of our approaches can be significantly improved by converting the MATLAB code into C or JAVA.)

For the rest of evaluation, we choose PDM as a representative of our probabilistic approaches.

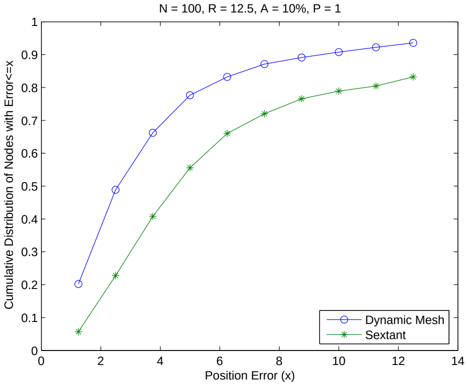


Figure 6: Probability distribution improves localization accuracy (100 nodes)

Figure 6 shows the performance for networks with 100 nodes in a space of size 70x70. Similar to networks with 50 nodes, PDM achieves higher accuracy than Sextant. For example, the percentage of nodes achieving $\leq 30\% * R = 3.75$ errors is 40% in Sextant, and is 67% in PDM. On average, Sextant takes 2.14 seconds per node to compute, and PDM takes 10.23 seconds per node to compute.

Effects of transmission range Transmission range R determines network density. More neighbors mean more location constraints, which usually result in higher localization accuracy. We vary R to obtain different network densities shown in Table 2. For simplicity, we assume the wireless propagation is regular (i.e. $R = r$) in our simulation. It is not difficult to generalize to $R \neq r$ cases.

N	Space	$R = 10$	$R = 12.5$	$R = 15$
50	50x50	6.0612	8.9592	11.28
100	70x70	6.0562	8.58	11.28

Table 2: Average node degrees under different transmission ranges.

As described in [14], 6 is a "magic" average node degree for a wireless network to be connected. So we choose the shortest range to be 10, which gives an average node degree of 6.

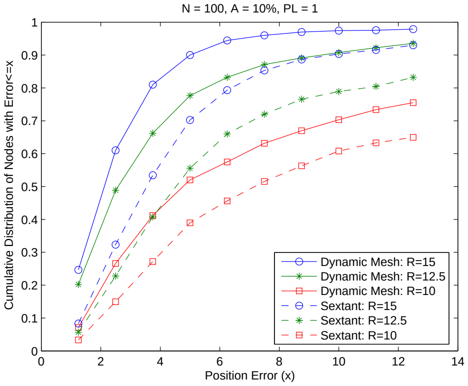


Figure 7: Effects of Transmission Range (100 nodes)

Figure 7 shows the results for different transmission ranges, while fixing $A = 10\%$. The accuracy results of 50-node (not shown) is similar. Again, PDM consistently outperforms Sextant. As we would expect, the accuracy is higher when the transmission range is larger, which results in higher network density. Since the transmission range is determined by transmission power, there is a tradeoff between energy-efficiency and localization accuracy.

Effects of the fraction of anchor nodes Next, we study how the fraction of anchor nodes, A, affects localization accuracy. In our evaluation, $R = 12.5$. Figure 8 shows the localization accuracy of 100-node networks as we vary the anchor fraction from 5% to 20%. (The results of 50-node networks are similar and omitted in the interest of brevity.) As before, PDM yields lower error than Sextant. In addition, we find that the anchor fraction significantly affects localization accuracy. The more anchor nodes, the higher localization accuracy. This is consistent with our expectation, because 1-hop neighbors of anchor nodes can be localized more accurately than nodes multiple hops away from anchor nodes due to smaller uncertainty. As shown in Figure 8, the increase in localization accuracy is significant as the anchor fraction increases from 5% to 10%. A further increase in the anchor fraction leads to more moderate increase in the accuracy. Therefore we use 10% as the anchor fraction for the remaining evaluation.

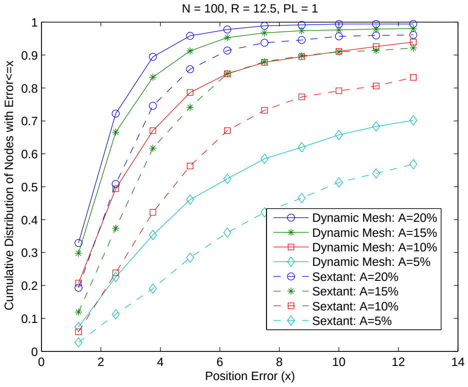


Figure 8: Effects of anchor nodes fraction (100 nodes)

Summary In this section, we compare different localization algorithms. Our results show that probabilistic region-based localization schemes using static grids, dynamic meshes, and segments of grids, achieve higher localization accuracy than Sextant. In addition, PDM provides a reasonable balance between accuracy and computation cost.

V. Two Extensions

In this section, we extend our approach in two directions. First, to further improve the localization performance, we propose to extract and take advantage of additional information by (i) using power control, (ii) using carrier-sense range as another reference distance besides communication range, (iii) incorporating physical layout, and (iv) exploiting more powerful anchor nodes. The additional information is useful to many localization algorithms including ours. Second, we enhance the robustness of our approach against erroneous measurement by tolerating certain degree of inconsistency among location constraints. Finally, we evaluate the effectiveness of these extensions.

V.A. Extract and Leverage Additional Information

The accuracy of a localization system highly depends on the amount of information available. We propose several ways to obtain additional information. They can be used separately or jointly, and can be applied to different localization algorithms. Note that while this is not the first paper that uses additional information besides network connectivity to infer location, several of the techniques presented here are novel. In addition, we evaluate and compare the effects of the additional information.

Using power control: Power control enables wireless nodes to obtain additional information in the following way. Suppose each power level p_k has corresponding maximum and minimum transmission range $R(p_k)$ and $r(p_k)$. By adjusting the transmission power, if a node i finds out that it can communicate with another node j at power level p_k, but cannot communicate at power level p_{k-1}, the distance between i and j should be between $R(p_k)$ and $r(p_{k-1})$. This additional information makes range estimation more accurate, and can be easily incorporated into any localization algorithm. As we would expect, a larger number of power levels provides more information and improves localization accuracy. Power control is an interesting and practical way for obtaining additional information since power control is readily available in commercial wireless cards. In addition, it only requires nodes to obtain network connectivity information, and does not require signal strength measurements or additional hardware.

Using carrier-sense range: Many existing localization algorithms rely on network connectivity information for location estimation. This gives us information as to whether a node is within or outside the communication range of another node. However we do not have further information about the nodes that are outside the communication range.

We make an interesting observation: in addition to communication range, carrier-sense range can also be used as a reference for distance estimation. For example, if two nodes cannot sense each other's carrier, they are outside each other's carrier-sense range. This type of information is not available if we only use network connectivity, since the carrier-sense range is typically larger than the communication range. Let R and $R_{carrier}$ denote communication range and carrier-sense range, respectively. If two nodes are outside communication range but can sense each other's carrier, their distance should be within the range [R, $R_{carrier}$]; if two nodes cannot sense each other's carrier, their distance is larger than $R_{carrier}$.

To determine whether two nodes can sense each other's carrier, we can measure whether these nodes can simultaneously broadcast [1]. More specifically, we measure the broadcast rate from the two senders when they are active simultaneously, and denote it as $T_{together}$. We also measure the broadcast rate when the two senders are active separately, and denote it as $T_{separate}$. If $\frac{T_{together}}{T_{separate}}$ is close to 1, it means that the two nodes do not sense each other's carrier; otherwise they do.

As with power control, we extract more precise distance information using the carrier-sense range, and it can be applied to different localization schemes.

Using physical layout: In some applications, we may have a rough idea of physical layout of wireless nodes. For example, in residential mesh networks [23], we know that wireless nodes are deployed at different houses, and we also have a neighborhood layout map. The map provides additional information for us to narrow down the location. Since a node can only be located at one of the houses, its final estimated location should be the intersection of its estimated region (without considering the physical layout) and the regions occupied by the houses.

Using more powerful anchor nodes: As the previous work shows, angle information is valuable for location estimation. However, obtaining angle information often requires more expensive hardware (e.g., directional antennas or additional transmitters like ultrasound). In order to achieve both high accuracy and low cost, a promising approach is to use a combination of more powerful nodes and less powerful nodes. For example, only the anchor nodes are equipped with powerful devices for more detailed measurement, whereas the remaining nodes use cheap devices as usual. An interesting question is how much benefit such powerful anchor nodes offer. In this paper, we study the following type of powerful anchor nodes: anchor nodes that are equipped with directional antennas for measuring angle information towards its immediate neighbors. We evaluate localization accuracy as we vary the fraction of anchor nodes.

V.B. Enhance Robustness

A node estimates its location by finding regions that satisfy a set of location constraints. Location constraints are usually obtained by measuring distances or angles between nodes. However, such measurements can be erroneous, and in some cases even lead to *inconsistent* location constraints. A set of location constraints are *inconsistent* if there is no point that can satisfy all these constraints.

We propose a technique on top of our probabilistic region-based approach to achieve robustness against inconsistent location constraints. We leverage the fact that majority of location measurements are consistent; and only a few constraints may contain significant errors and result in inconsistency. Therefore a mesh cell belongs to a node's estimated region as long as it satisfies most of the constraints. In our evaluation, we use 80% as a threshold (i.e., a mesh cell is considered to belong to a node's estimated region if it satisfies at least 80% of the constraints for that node). As part of our future work, we plan to choose the threshold adaptively.

Our robust localization proceeds in the following three steps. First, as before, every node propagates location constraints to all nodes within 3 hops away (i.e. TTL=3). Second, each node i calculates its own region based on the location constraints from other nodes. Location constraints from a node j determine a region S_{ij} for i. Unlike in Section III, i does not calculate its region as $S_i = \cap_j S_{ij}$. Instead, S_i is calculated as the set of mesh cells u_i such that u_i satisfies at least 80% of the constraints. Finally, each node calculates the probability distribution over all mesh cells within its estimated region. This step is similar to what we describe in Section III.

V.C. Performance Evaluation of Extensions

In this section, we evaluate the performance benefits of additional information and robustness enhancement.

V.C.1. Evaluation of Leveraging Additional Information

In this section, we study the effects of leveraging additional information. First, we examine the effect of power control by varying the number of power levels PL that a node can use for its transmission. Table 3 lists the transmission power at different levels, where P is the maximum transmission power. Note that $PL = 5$ corresponds to or approximates several commercial wireless cards (e.g., Netgear WAG511 and Cisco Aironet 350 series). Next we examine the effect of carrier-sense range-based constraints by varying $R_{carrier} = 1.5R, 2R, 2.5R, 3R$, where R is communication range. Table 4 summarizes the notation we use. Then we evaluate the performance benefit from incorporating a physical layout map. Finally, we examine the effect of using powerful anchors that have angle information. We consider three levels of angle measurement errors: large errors within $[-20, 20]$ degrees, medium errors within $[-10, 10]$ degrees and small errors within $[-5, 5]$ degrees. These values are consistent with commercial directional antennas.

PL	Fraction of maximum transmission power P
1	100%
2	25%,100%
3	6.25%,25%,100%
5	6.25%,12.5%,25%,50%,100%
10	6.25%,10%,12.5%,20%,25%,35%,50%,65%, 80%,100%

Table 3: Transmission power for different power levels

N	the number of nodes
R	transmission range
A	the fraction of anchor nodes
PL	the number of power levels
$R_{carrier}$	carrier-sense range

Table 4: Notation used in performance evaluation.

Effects of power control When only connectivity information is available, the distance measurement is binary–either $d \leq R$ or $d > R$. By adjusting the transmission power level, a node can extract more accurate distance constraints in the above form. As shown in Figure 9, the accuracy improves with an increasing number of power levels. For example, 20% nodes achieve position error within $10\% * R = 1.25$ when 1 power level is used. In comparison, 32%, 35%, 50%, and 65% nodes achieve similar errors when the number of power levels is 2, 3, 5, and 10, respectively. This demonstrates that power control is effective in improving localization accuracy.

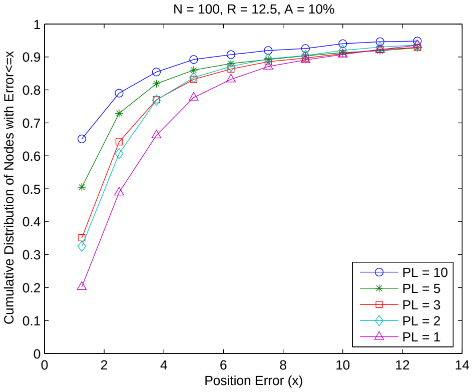


Figure 9: Effects of power control (100 nodes).

Effects of carrier sense constraint Besides power control, carrier-sense range can also help to extract more accurate distance constraints. As shown in Figure 10, compared with the base case without carrier sense information, constraints derived using carrier-sense ranges improve localization accuracy by a considerable amount. As the carrier-sense range increases, the negative constraints (i.e., $d > R_{carrier}$) become tighter, and the positive constraints (i.e., $d < R_{carrier}$) become looser. Interestingly, $R_{carrier} = 2 * R$ yields the highest accuracy among all the carrier-sense ranges considered. This suggests that the positive and negative constraints extracted using $2 * R$ are especially effective under the scenarios we consider.

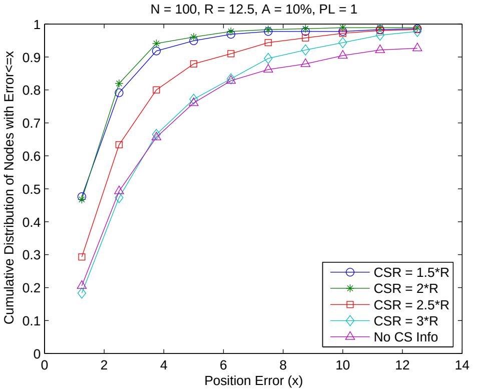


Figure 10: Effects of carrier sense constraints (100 nodes).

Effects of map constraint Next we study the performance gain from a layout map. In our evaluation, we obtain a real neighborhood map, which contains the coordinates of houses. We select 56 houses from the map over a 1400m x 700m space. Since there is no house size information, we generate the regions occupied by the houses as follows. Each house is a square and has the same size. A house is centered at its coordinate, and its size, $hsize$, is determined based on the minimum distance between any pair of houses, d_{min}. In the localization process, each node derives its region and probability distribution based on the constraints imposed by the map (i.e., a node can only be inside a house), as well as the location constraints from other nodes. We use transmission range of 150 meters, which gives an average node degree of 6.39.

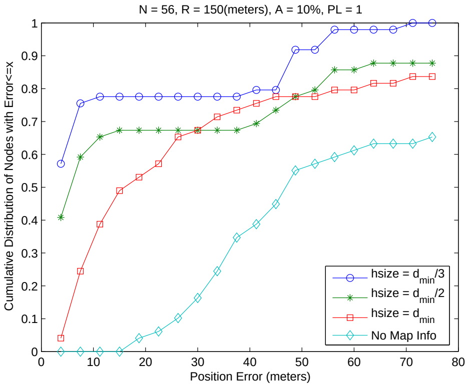


Figure 11: Effects of a physical layout.

As shown in Figure 11, a layout map significantly improves localization accuracy. In addition, the smaller house size, the higher localization accuracy. This is what we would expect. Because a node can only reside in a house, the location constraints imposed by the map is tighter for smaller houses. Nevertheless, even when $hsize = d_{min}$, localization accuracy is still significantly higher than without the layout map.

Effects of powerful anchors Finally, we examine how anchor nodes with angle measurement affect the accuracy of localization. We use three levels of angle measurement errors: $[-20, 20]$ degrees, $[-10, 10]$ degrees and $[-5, 5]$ degrees. An estimated angle is then the true angle plus noise uniformly distributed within the error intervals. Figures 12 summarizes the results.

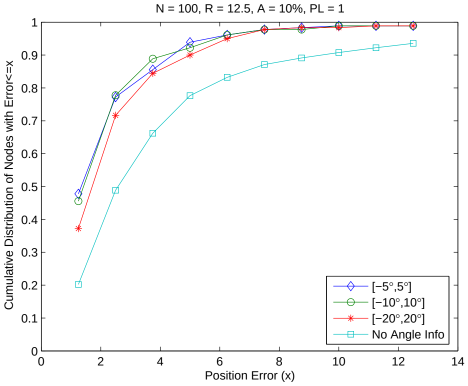


Figure 12: Effects of angle information (100 nodes).

We make the following observations. First, angle information helps to decrease the localization error significantly. Second, even when the angle measurement contains errors of $[-20, 20]$ degrees, localization accuracy is still significantly higher than the accuracy achieved without angle information. Compared with $[-5, 5]$ degrees of angle measurement error, its accuracy is slightly lower at the low end of position errors, and comparable for the remaining position errors.

Summary In this section, we study the effect of additional information, including using power control, carrier-sense range-based constraints, a layout map, and angle measurements from anchor nodes. Our results demonstrate that the additional information is effective in significantly improving localization accuracy.

V.C.2. *Evaluation of Robustness Enhancement*

In this section, we evaluate the robustness of our extended localization algorithms. First we consider the case where the transmission range information is inaccurate. More specifically, each node's true communication range (R) is $R = R_{est} + R_{error}$, where R_{error} is a positive or negative range estimation error, and R_{est} is the communication range that we have estimated. R_{error} arises from the difference in transceivers' properties and environmental effects. While one may try to reduce R_{error} by individually calibrating each node (e.g., obtaining conservative minimum and maximum communication ranges), such calibration is costly. Moreover even with calibration, errors cannot be completely eliminated due to changing environmental effects. As shown in Figure 13, with robustness enhancement, the localization algorithm maintains high accuracy when the communication ranges contain up to $20\% * R$ errors. The accuracy is lower when R_{error} increases up to $40\% * R$, but still all nodes can be localized, with around 60% nodes achieving within $R/2 = 6.25$ position error.

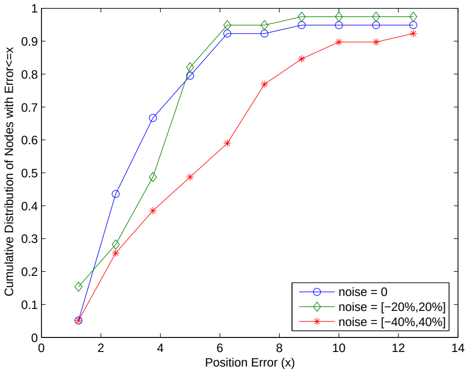


Figure 13: Effects of inaccurate communication range ($N = 50$, $R = 12.5$, $A = 10\%$, $PL = 1$).

Next we consider errors arising from malicious nodes. In our evaluation, we randomly select a few nodes as malicious nodes. Such a node pretends to be at a randomly generated location. It calculates a region of a circle centered at the false location with radius R, and then transmits this region as a false constraint to its neighbors. Figure 14 shows the effects of malicious nodes. There are two sets of curves, corresponding to the results of position errors within $R/2$ and within R. “Grid-2” and “Robust Grid-2” curves represent the results from using fixed 2x2 rectangular cells with and without the additional robustness enhancement, respectively. After introducing such malicious nodes, not all nodes can be localized due to potentially inconsistent constraints. For the nodes that have inconsistent constraints and cannot be localized, their localization error is considered larger than R. As Figure 14 shows, even when the fraction of malicious nodes is only 10%, the percentage of nodes with position errors $\leq R/2 = 6.25$ drops as much as 30% under both Sextant and Grid-2. In comparison, with the additional robustness enhancement, the accuracy reduction under the “Robust Grid-2” is small especially when the fraction of malicious nodes is within 10% (only 10% reduction). Moreover, even when 30% nodes are malicious, majority of nodes can still be localized within errors of R under “Robust Grid-2”. This demonstrates the effectiveness of our robustness enhancement.

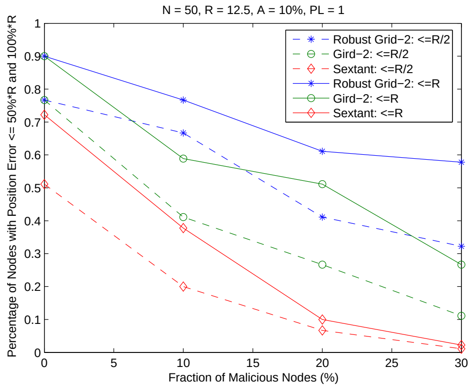


Figure 14: Effects of malicious nodes ($N = 50$, $R = 12.5$, $A = 10\%$, $PL = 1$).

Summary Our evaluation results show that the robustness enhancement is effective. The enhancement helps maintain high localization accuracy even when there are $20\% * R$ range errors or 10% malicious nodes.

VI. Conclusion

In this paper, we present probabilistic region-based localization schemes. We use static grids, dynamic meshes, and segments of grids as units for which probability is computed. Our results show that these probabilistic-based localization schemes achieve high accuracy. We further extend our approach in two ways: we exploit new ways to obtain and leverage additional information, and enhance robustness by tolerating inconsistency in location constraints arising from measurement errors and/or malicious nodes. Our evaluation demonstrates the effectiveness of these extensions.

References

[1] S. Agarwal, J. Padhye, V. N. Padmanabhan, L. Qiu, A. Rao, and B. Zill. Measurement and estimation of link interference in static multi-hop wireless networks. In *Proc. of Internet Measurement Conference*, Oct. 2005.

[2] Alert systems. `http://www.alertsystems.org/`.

[3] P. Bahl and V. N. Padmanabhan. RADAR: an in-building rf-based user location and tracking system. In *Proc. of IEEE INFOCOM 2000*, Mar. 2000.

[4] A. Cerpa, J. Elson, D. Estrin, L. Girod, M. Hamilton, and J. Zhao. Habitat monitoring: application driver for wireless communications technology. In *Proc. of ACM SIGCOMM Workshop on Data Communications*, Aug. 2001.

[5] DistMesh – a simple mesh generator in matlab. `http://www-math.mit.edu/~persson/mesh/`.

[6] E. Elnahrawy, X. Li, and R. P. Martin. The limits of localization using signal strength: A comparative study. In *Proc. of the IEEE Conference on Sensor and Ad Hoc Communication Networks (SECON)*, Oct. 2004. `http://www.cs.rutgers.edu/~rmartin/papers/limits-secon.pdf`.

[7] T. Eren, D. Goldenberg, W. Whitley, Y. R. Yang, A. S. Morse, B. D. Anderson, and P. N. Belhumeur. Rigidity, computation, and randomization of network localization. In *Proc. of IEEE INFOCOM*, Mar. 2005.

[8] A. Galstyan, B. Krishnamachari, K. Lerman, and S. Pattem. Distributed online localization in sensor networks using a moving target. In *Proc. of the International Symposium on Information Processing in Sensor Networks*, Apr. 2004.

[9] Global positioning system standard positioning service specification. *United States Coast Guard Navigation Center*, June 1995.

[10] S. Guha, R. N. Murty, and E. G. Sirer. Sextant: A unified framework for node and event localization in sensor networks. In *Proc. of ACM MOBIHOC*, May 2005.

[11] J. Hightower and G. Borriello. Location systems for ubiquitous computing. *IEEE Computer*, Aug 2001.

[12] L. Hu and D. Evans. Localization for mobile sensor networks. In *Proc. of ACM MOBICOM*, Sept. 2004. `http://www.cs.virginia.edu/~evans/pubs/mobicom2004.pdf`.

[13] B. Karp and H. Kung. Greedy perimeter stateless routing for wireless networks. In *Proc. of ACM MOBICOM*, Aug. 2000. `http://www-2.cs.cmu.edu/~bkarp/gpsr-mobicom2000.ps.gz`.

[14] L. Kleinrock and J. Silvester. Optimum transmission radii for packet radio networks or why six is a magic number. In *NTC '78; National Telecommunications Conference*, Dec. 1978.

[15] M. Korkea-aho. Context-aware applications survey. *Internetworking Seminar*, Apr. 2000. `http://users.tkk.fi/~mkorkeaa/doc/context-aware.html`.

[16] U. Kubach and K. Rothermel. Exploiting location information for infostation-based hoarding. In *Proc. of ACM MOBICOM*, Jul. 2001.

[17] D. Madigan, E. Elnahrawy, and R. P. Martin. Bayesian indoor positioning systems. In *In Proc. of IEEE INFOCOM*, Mar. 2005.

[18] D. Moore, J. Leonard, D. Rus, and S. Teller. Robust distributed network localization with noisy range measurements. In *Proc. of ACM SENSYS*, Nov. 2004.

[19] D. Niculescu and B. Nath. VOR base stations for indoor 802.11 positioning. In *Proc. of International Conference on Mobile Computing and Networking*, Sept. 2004.

[20] P. O. Persson and G. Strang. A simple mesh generator in matlab. *SIAM Review*, Jun. 2004.

[21] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan. The cricket location-support system. In *Proc. of ACM MOBICOM*, Aug 2000.

[22] A. Rao, S. Ratnasamy, C. Papadimitriou, S. Shenker, and I. Stoica. Geographic routing without location information. In *Proc. of ACM MOBICOM*, Sept. 2003. `http://www.cs.ucla.edu/classes/fall03/cs218/paper/p96-rao.pdf`.

[23] MIT Roofnet. `http://www.pdos.lcs.mit.edu/roofnet/`.

[24] A. Savvides, C. Han, and M. B. Strivastava. Dynamic fine-grained localization in ad-hoc networks of sensors. In *Proc. of ACM MOBICOM*, Jul. 2001.

[25] A. Savvides, H. Park, and M. B. Srivastava. The bits and flops of the n-hop multilateration primitive for node localization problems. In *Proc. of WSNA'02*, Sep. 2002.

[26] Y. Shang and W. Ruml. Improved MDS-based localization. In *Proc. of IEEE INFOCOM*, Apr. 2004. `http://www.ieee-infocom.org/2004/Papers/55_1.PDF`.

[27] Y. Shang, W. Ruml, Y. Zhang, and M. Fromherz. Localization from mere connectivity. In *Proc. of ACM MOBIHOC*, Jun. 2003. `http://www.sigmobile.org/mobihoc/2003/papers/p201-shang.pdf`.

[28] D. C. Steere, A. Baptista, D. McNamee, C. Pu, and J. Walpole. Research challenges in environmental observation and forecasting systems. In *Proc. of 6th International Conference on Mobile Computing and Networking*, 2000.

[29] R. Want, A. Hopper, V. Falcao, and J. Gibbons. The active badge location system. *ACM Trans. Information Systems*, Jan. 1992.