Mobile-Assisted Localization in Wireless Sensor Networks

Nissanka B. Priyantha, Hari Balakrishnan, Erik D. Demaine, Seth Teller
MIT Computer Science and Artificial Intelligence Laboratory
Email: {bodhi,hari,edemaine,teller}@csail.mit.edu
Web: cricket.csail.mit.edu

***Abstract*— The *localization problem* is to determine an assignment of coordinates to nodes in a wireless ad-hoc or sensor network that is consistent with measured pairwise node distances. Most previously proposed solutions to this problem assume that the nodes can obtain pairwise distances to other nearby nodes using some ranging technology. However, for a variety of reasons that include obstructions and lack of reliable omnidirectional ranging, this distance information is hard to obtain in practice. Even when pairwise distances between nearby nodes are known, there may not be enough information to solve the problem uniquely.**

This paper describes *MAL*, a *mobile-assisted localization* method which employs a mobile user to assist in measuring distances between node pairs until these distance constraints form a "globally rigid" structure that guarantees a unique localization. We derive the required constraints on the mobile's movement and the minimum number of measurements it must collect; these constraints depend on the number of nodes visible to the mobile in a given region. We show how to guide the mobile's movement to gather a sufficient number of distance samples for node localization. We use simulations and measurements from an indoor deployment using the Cricket location system to investigate the performance of MAL, finding in real-world experiments that MAL's median pairwise distance error is less than 1.5% of the true node distance.

I. Introduction

The localization problem in sensor networks can be stated as follows: *Given a collection of N nodes, and distance measurements of each node to its neighbors, produce a set of coordinate assignments p_i for each node i, such that the assigned distance between nodes i and j, $\|p_j - p_i\|$, is equal to the measured distance, d_{ij}.* Of course, in the absence of an external coordinate reference, this assignment can be unique only up to an arbitrary rotation, translation, and possible reflection, but its scale is determined by the measured ranges. For example, if three nodes are placed such that the pairwise distances between them are 3, 4, and 5 units, a correct coordinate assignment would be $(0,0)$, $(3,0)$, and $(0,4)$.

The localization problem has received a great deal of recent attention in the literature (*e.g.,* [1]–[7]). Knowledge of location enables nodes in a sensor network to annotate sensed data with location information, making the sensed information more useful to applications. Knowledge of node location can be used to implement efficient message-routing protocols (such as geographic forwarding) in wireless ad-hoc and sensor networks. Localization is also useful in indoor location infrastructures such as Cricket [8], [9] and Bat [10], in which reference nodes at various "known" locations in a building provide location information to mobile devices and sensor nodes. Because manually configuring each reference node with its position is cumbersome and error-prone, such systems benefit from a method to automatically localize the reference nodes.

Previous approaches to solving the localization problem generally rely on each node being able to obtain its distances to the nodes near it (*e.g.,* within radio range). Given these pairwise distances, a distributed or centralized algorithm then computes a coordinate assignment for all nodes that is consistent with the measured pairwise node distances.

When one attempts to implement a localization algorithm in real-world systems, significant problems arise. The first problem is that *physical obstacles* that obstruct line-of-sight connectivity between neighboring nodes prevent pairwise node distances from being obtained. This problem arises inside many buildings, where it is often hard to obtain line-of-sight connectivity between rooms and open spaces. Moreover, physically realizable ranging hardware is often not omni-directional; for example, in an ultrasound-based location infrastructure such as Cricket in which ceiling- and wall-mounted *beacons* broadcast spatial and ranging information to mobile *listeners*, the ultrasonic transmitters point toward the floor, making it less likely that a given pair of beacons will be able to measure their mutual distance.

Another significant problem that arises in practice is that there may be too few distance constraints to obtain a consistent coordinate assignment. Obtaining a coordinate assignment that is unique up to translation, rotation, and reflection requires that the graph formed by available distances be *globally rigid* in a technical sense defined in Section III. An arbitrary deployment of location-infrastructure nodes (either beacons or passive receivers) or sensors will not generally produce a globally rigid structure. Section II describes in more detail these and other barriers to achieving practical localization.

This paper shows that *mobility* can help solve these problems. In *mobile-assisted localization*, a roving human or robot wanders through an area, collecting distance information between the nodes and itself. We describe a simple method that, given distance information between the moving node and the static nodes, formulates an optimization problem whose solution is the pairwise node distances between the static nodes. The challenge is to design movement strategies that produce a globally rigid structure of known distances among the static nodes. Using theoretical results from rigidity theory, we show that it is possible to constrain node movement in a way that achieves our goals. Section III gives one such practical movement algorithm.

The pairwise node distances resulting from our strategy can then be fed into a localization algorithm. We briefly discuss the *Anchor-Free Localization (AFL)* algorithm [6], which does not require any "anchor" nodes that already know their positions. AFL computes an initial coordinate assignment to all the nodes, using the radio connectivity information alone. This initial assignment results in a node layout that resembles a scaled version of the actual node layout, roughly preserving the topological ordering of nodes. AFL then uses an iterative optimization procedure to reduce the sum of squared distance errors between the nodes' true distances and the distances inferred from their current coordinates.

We show using simulations and real-world measurements that mobile-assisted localization is a practical approach that can be used in real-world systems. Although our solution is end-to-end, we believe that our decomposition into the *mobile-assisted topology building phase* and the *localization phase* is valuable. In particular, a variety of solutions to the latter phase can be implemented within our framework, adapting the algorithm to conditions at hand (*e.g.,* node density, range, expected ranging errors, etc.). Section V describes our experimental and simulation results, which show that mobile-assisted localization is both practical and accurate.

II. The Case for Mobile-Assisted Localization

A localization algorithm needs a sufficient number of pairwise node distances to be able to compute node coordinates correctly. However, there are several reasons why it is hard to meet this requirement in practice, especially indoors.

1) *Obstructions* occlude line-of-sight connectivity, making it hard or impossible for nodes to obtain pairwise distances between each other.
2) *Sparse node deployments* make it hard to obtain a *rigid* structure, which is necessary to obtain a unique solution.
3) *Geometric dilution of precision* (GDOP) causes a node that is far from a group of closely spaced nodes to incur large errors in its position estimate.

The rest of this section describes these problems in detail, and explains why mobile-assisted localization helps solve them. We also explain some additional benefits that mobility brings in solving the localization problem.

A. Obstructions

The lack of line-of-sight connectivity may prevent the nodes from obtaining direct node-to-node distances. Most of the ranging technologies used for accurate indoor ranging today, including time-of-flight of ultrasound, laser, and infrared, require line-of-sight between the transmitter and the receiver. Even technologies that do not need line of sight, such as ultrawideband (UWB) radio, have better accuracy when line-of-sight connectivity is available. Based on our experience with deploying Cricket, a system that uses ultrasound ranging, we have found that it is almost impossible to deploy nodes in a typical office or home to achieve sufficient connectivity across all nearby nodes. For example, it is hard to obtain ranging between nodes placed inside and outside a room in a standard building.

The lack of omnidirectional ranging may prevent reference nodes in a location system from obtaining pairwise distances. Because the primary goal of a location system is to help mobile devices obtain distance and location information, a key requirement is to provide maximum coverage to users. As a result, directional ranging transmitters on reference nodes are usually pointed toward where the users are likely to be, rather than toward other reference nodes.[1] Building omnidirectional ranging is usually more expensive (*e.g.,* it requires multiple transceivers) and entails hardware changes; furthermore, it seems wasteful because localization of the reference nodes is done only during deployment and not continually as for mobile localization.

In some cases, the reference nodes may have no ability to *receive* the signals necessary to estimate distances. For example, the reference nodes may emit radio and ultrasonic signals, but not have an ultrasonic receiver; alternatively, the reference node may be a passive tag-like device that mobile units query to obtain distance information. In such cases, mobile-assisted localization will be invaluable to the auto-localization procedure.

B. Sparse Node Deployments

Although a dense node deployment of reference nodes helps achieve good coverage in a location system, economic considerations often force sparse node deployments. In a sensor network, the deployment density may be dictated by cost or application requirements, and may be sparse. Sparse deployments reduce inter-node connectivity and could lead to a structure that is not rigid. For example, a room with four reference nodes where only four distances are known (forming a quadrilateral among the nodes) leads to a non-rigid structure. In such cases, *no* auto-localization algorithm can find the right positions of the nodes, because there are too few constraints and an ambiguous solution space. Yet, there *is* a unique (modulo translation, rotation, and reflection) assignment of position coordinates to nodes.

Mobile-assisted localization is well-suited to collect enough distance samples such that the resulting per-node distances that are inferred form a rigid structure. Because the reference nodes are deployed to cover an area where users move, the moving user or robot can take advantage of the many positions where distance ranges to the sensor nodes are obtained.

[1] For example, due to the radiation pattern of the ultrasonic transducers used, our ultrasonic-based ranging system has a 12 m range when the transmitter and the receiver are facing each other but only a <2 m mutual range when they are on the same horizontal plane facing away from the plane (*e.g.,* downwards from a ceiling).

C. Geometric Dilution of Precision (GDOP)

In practice, the distance measurements used to compute node coordinates almost always have some error. These measurement errors get reflected in the computed node coordinates. The magnitude of the final computed error depends on both the magnitude of the measurement error and the true geometry of the structure induced by the nodes and edges. The contribution due to geometry is called the *geometric dilution of precision (GDOP)* [11][2], and is defined as the ratio between the computed coordinate error and the measurement error. That is, GDOP represents the factor by which the distance measurement error gets multiplied when it is used to compute node coordinates. When distance measurements are used for computing node coordinates by solving for an exactly constrained system of equations, we get GDOP > 1.

It is well-known that using an over-constrained system of equations tends to reduce GDOP errors [12]. Adding additional constraints in conventional approaches leads to increased node density; in contrast, with mobile-assisted localization, a mobile unit can move around a region and usually obtain as many additional constraints as are necessary. Our approach uses these additional constraints only locally to accurately compute the internode distance. Since obtaining a large number of additional constraints is not cumbersome, mobile-assisted localization can greatly improve the accuracy of coordinate position estimation by reducing the adverse effects of GDOP.

D. Other Benefits

Although the use of mobile device positions as "virtual" nodes to add more constraints to a weakly connected graph and running a localization algorithm on the combined set of nodes seems like a simple extension, the flexibility offered by the mobile device acting as a virtual node creates several opportunities that makes our approach quite different from traditional node localization:

1) The dynamic nature of the mobile-assisted scheme enables us to evaluate the currently available distance information "on the fly," and navigate the mobile to obtain any additional distances required.
2) The additional virtual nodes corresponding to mobile positions do not have the associated cost of additional physical nodes. The only criteria that limit the number of such positions are the available computational and storage resources, both of which are usually ample on current hand-held devices.
3) The reference and virtual (mobile) nodes typically occupy different regions in space. In a typical location infrastructure deployment, for example, the reference nodes will be located close to the ceiling of a room and above the space occupied by users. In contrast, the virtual nodes, corresponding to mobile locations, will be located close to the users. This separation helps mobile-assisted localization perform well, as we will see.

[2]GDOP is a well-known problem that arises in all location systems, including GPS.

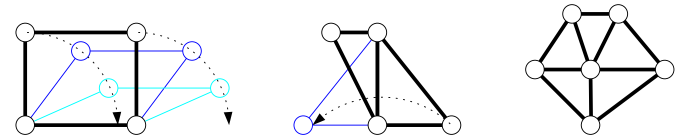

Fig. 1. Examples of graphs that are not rigid (flexible as a bar-and-joint framework), rigid but not globally rigid (multiple embeddings), and globally rigid (one embedding up to rotation, translation, and reflection).

III. Theoretical Framework and Movement Strategies

The standard localization problem is to reconstruct the position assignment of nodes (global geometry) given a graph with edges labeled by measured distances (local geometry). In mobile-assisted localization, only part of the graph (if at all) is given and fixed: the rest we have (limited) control over by moving a mobile node according to a particular strategy.

This section defines MAL's movement strategy for building up such a graph in a way that guarantees a unique solution (and even an easy-to-find solution) of the resulting standard localization problem. More precisely, we show how a mobile can explore a geographic area and incrementally build a localizable graph by adding new virtual nodes (corresponding to the various positions of the mobile) and adding edges between these nodes and stationary nodes. At a high level, the mobile starts by finding a cluster of nearby nodes, and then it explores the visible region for new nodes to which it can measure distance. The number of measurements required by the mobile is linear in the number of stationary nodes, and the total motion required by the mobile can be similarly bounded.

We also show how to reduce the size of the localizable graph, removing the virtual nodes arising from mobile positions, and leaving just the stationary nodes. Indeed, our approach is to find virtual node configurations that allow us to measure one or more distances between two stationary nodes, after which the virtual nodes and their incident edges can be discarded. Then our goal becomes to measure distances between stationary nodes so that the graph on the stationary nodes becomes localizable. We show that this approach is as efficient as possible: no loss is caused by discarding virtual-node information once it is abstracted into the graph of stationary nodes. The benefit is that this reduction in graph size speeds up the final phase of localization using, *e.g.,* the method outlined in Section IV.

A. Connections to Rigidity

We start by describing connections between standard localization, in particular determining whether a problem instance has enough information to have a unique solution, and a branch of mathematics called rigidity theory. These connections provide tools for understanding when we have enough distance information to guarantee localizability.

Given just distance information, at best we can localize the network up to rotation, translation, and reflection. This local coordinate system is enough in many cases, or else can be matched against a global coordinate system if a few nodes have global coordinates obtained through, *e.g.,* GPS. However, for some graphs, the position assignment is not unique even up to rotation, translation, and reflection, as shown in Figure 1. If we treat the graph as a bar-and-joint framework or linkage, the graph should at least be *rigid* in the sense that it cannot be flexed while preserving the distances (as in a rectangle, for example). Even if the graph is rigid, it may be subject to "local flips." For example, if there are just two triangles sharing an edge, one triangle can be reflected through that edge without any distances changing. We call such a graph *locally rigid* but not *globally rigid*. For the localization problem to have a unique solution, we need a *globally rigid* graph that has exactly one embedding with correct edge lengths.

Global rigidity was introduced by Hendrickson [13] as an important variation on the well-studied concept of local rigidity [14]–[16]. Hendrickson showed that, for a graph to be generically[3] globally rigid in d dimensions, it must satisfy two properties: (1) the removal of any d vertices must leave the graph connected ($(d+1)$-connectivity), and (2) the removal of any edge must leave the graph generically locally rigid. Both these properties can be checked in polynomial time. Connelly [17] proved that these two properties are insufficient in 3D: they do not imply generic global rigidity of a 3D graph. However, Hendrickson conjectured that these two properties exactly characterize generic global rigidity in 2D, and this conjecture was recently proved in unpublished work of Jackson and Jordán [18]. Thus, global rigidity is well-understood in 2D, but not in 3D. Nonetheless, we show how to *construct* graph structures that are guaranteed to be globally rigid and therefore localizable in both 2D and 3D.

B. Engineering Rigidity

Although testing global rigidity can be difficult in general, we can build up 3D and 2D structures that are guaranteed to have global rigidity. For the duration of this subsection, we consider what distances should be measured between stationary nodes in order to guarantee a rigid structure among just those nodes. In the following subsections, we show how to measure such distances using a mobile, and ultimately how to organize the motion of the mobile to measure all required distances.

To start a rigid structure, suppose that we can measure all pairwise distances between some four nodes p_1, p_2, p_3, p_4 (which we assume lie at distinct points in space). The resulting structure is a tetrahedron, which is the simplest globally rigid graph in 3D. It is globally rigid no matter where the points p_i are located, but for further building, suppose that they do not lie on a common plane.

[3]The "generic" qualifier here is a technical detail that allows us to consider the qualities of a combinatorial graph, and not the specific edge lengths, that make a structure rigid. With probability 1, the problem is generic.

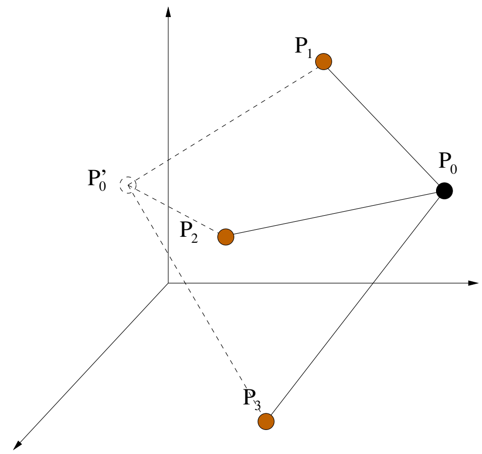

Fig. 2. Connecting a node (p_o) to three already-localized nodes (p_1, p_2, p_3) on a locally rigid graph results in a locally rigid graph.

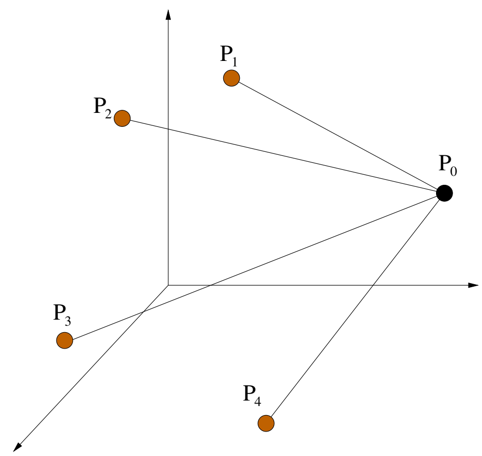

Fig. 3. Connecting a node (p_0) to four non-coplanar points on a globally rigid graph results in a globally rigid graph.

Now suppose that we have already localized three nodes p_1, p_2, p_3 that do not lie on a common line, as in Figure 2, and we want to localize a new node p_0 given the measured distances $d_i = \|p_i - p_0\|$. Point p_0 therefore lies simultaneously on three different spheres, centered at p_i and with radius d_i, for $i = 1, 2, 3$. The intersection of three spheres with non-collinear centers is always at most two points. Thus we can compute in constant time two possible locations for p_0: the true location of p_0 and the reflection of that location through the plane passing through p_1, p_2, p_3. If we know extra information about which side of the plane p_0 lies (from external information, *e.g.,* input by the user), this local rigidity would suffice, but generally it does not.

To attain global rigidity, we need the distances from the new point p_0 to four already localized points p_1, p_2, p_3, p_4 that do not lie on a common plane, as in Figure 3. The extra distance constraint from p_4 defines an additional sphere of radius $d_4 = \|p_4 - p_0\|$. Given that p_4 lies off the plane passing through p_1, p_2, p_3, only one of the two reflection solutions will have the proper distance. Thus we uniquely localize p_0.

This approach for building globally rigid graphs can be summarized as follows:

Theorem 1: A graph is globally rigid if it is formed by starting from a clique of four non-coplanar nodes and repeatedly adding a node connected to at least four non-coplanar existing nodes.

The idea of this incremental construction was first used in 2D by Coullard and Lubiw [19] to prove global rigidity of certain 2D visibility structures. It has since been used in several incremental localization algorithms [1], [20].

Our main novelty is the use of *mobile assistance* to efficiently perform such a construction, as described in the following subsections. In the absence of mobile assistance, incremental localization approaches over the stationary nodes alone often do not yield good results because it is hard to arrange for the stationary nodes to be added to a previously rigid structure while preserving rigidity. Moreover, our approach combats problems due to GDOP and non-rigid node placement, as explained in Section II.

C. MAL: Distance Measurement

Theorem 1 guides our mobile by determining when it has measured "enough" distances: once the graph is constructible according to the theorem, we are guaranteed to have a globally rigid graph. If we have any such strategy for finding a globally rigid graph including both the original stationary nodes and the virtual nodes representing mobile positions, then this rigid structure defines a unique coordinate assignment, so in particular we can measure distances between any pair of nodes. Thus any strategy can be viewed as a method for determining distances (possibly through indirect measurement/deduction) between certain pairs of stationary nodes in such a way that these distances form a globally rigid graph. We take this approach, and by the argued equivalence, this approach does not require us to take any more measurements than inherently necessary.

The rest of this subsection considers the subproblem of how to derive the distance between two stationary nodes given just distances between various positions for the mobile and various stationary nodes, none of which have been localized. We consider several alternatives for solving this problem, depending on what assumptions (if any) can be made of the stationary nodes and mobile locations. The next subsection (Section III-D) describes how the mobile navigates to find a set of distances satisfying Theorem 1.

1) Calculating distance between two nodes: We start with what superficially seems like the most natural problem: compute the distance between two nodes n_0 and n_1 by measuring their distances to various locations of a mobile node m. This problem starts with a single unknown, $\|n_0 - n_1\|$, and no known information. Unfortunately, for every new location of the mobile node, we introduce three new unknowns for the coordinates of that location, and add only two new known quantities. Thus adding more information actually makes the problem less determined. (It is necessary, though not sufficient, for a naïve count of the number of degrees of freedom (unknowns minus knowns) to be at most 0.) Even if we suppose that the mobile node m stays on a plane (say, a fixed height from the floor), the new knowns balance the new unknowns, but we never actually gain any information, so we cannot learn the original unknown.

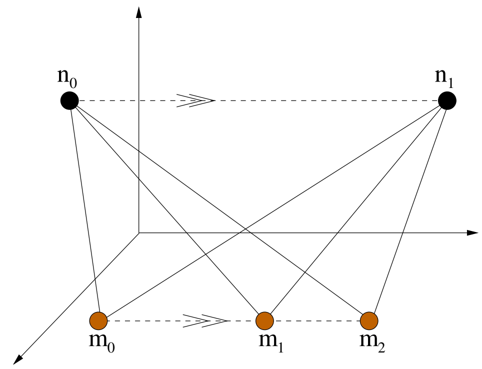

Fig. 4. Computing distance between two nodes by measuring distances from 3 points on a parallel line.

Thus we are forced to turn to a different problem. We start with some restricted forms of motion by the mobile that make distance calculation possible. In most situations, however, the most practical strategy is to involve more than two nodes in distance measurement, as described below.

The first approach is to move the mobile along a line in a plane containing both n_0 and n_1. A practical example of this setting is when the stationary nodes are at a fixed height from the floor, as is the mobile (at a different height), in which case the mobile should move along a projection of the line through n_0 and n_1 (Figure 4). If we now measure distances from three mobile locations, we obtain an extra constraint that these three locations are collinear. This constraint is enough to determine the configuration:

Proposition 2: The geometry of five coplanar points n_0, n_1, m_0, m_1, m_2, where m_0, m_1, m_2 are collinear, is determined by the distances $\|n_i - m_j\|$ for $i = 0, 1$ and $j = 0, 1, 2$.

The solution geometry can be computed in polynomial time using standard techniques in real constraint optimization for numerically solving systems of polynomial equations (the theory of the reals). Although the minimum number of mobile locations required for a solution is three, using a larger number of points would reduce the error caused by GDOP. Note also that it may not be practical to constrain movement in this fashion. A second, simpler approach when both stationary nodes are at a fixed height from the ground is to position the mobile directly under one of the nodes. Now we can use the Pythagorean theorem to compute the distance between two nodes as $\sqrt{d_2^2 - d_1^2}$, where d_1 is the distance to the node directly above and d_2 is the distance to the other node from the mobile [21]. Unfortunately, positioning the mobile directly under a node is error-prone; manual placement could cause an error of several centimeters.

2) Calculating distances among three nodes: Next we consider a more tractable problem: compute the pairwise distances between three nodes n_0, n_1, and n_2 by measuring their distances to various locations of a mobile node m. Now the problem starts with three unknowns, $\|n_0 - n_1\|$, $\|n_1 - n_2\|$, and $\|n_2 - n_0\|$. Without any assumptions, we again run into the problem that each mobile position introduces as many unknowns (the three coordinates of the position) as new constraints (the three distances).

Thus we impose one restriction: that the mobile positions all lie on a common plane. This restriction is easy to achieve in practice by moving the mobile receiver at a fixed height from the ground, assuming that the ground is flat. Now if we have k mobile locations, we obtain $k-3$ additional coplanarity constraints. (The first three mobile locations are automatically coplanar, as are all three points in space.) Therefore, $k = 6$ mobile locations are necessary to reduce the number of degrees of freedom to 0. Moreover, these constraints suffice to uniquely determine the geometry provided we know that the stationary nodes are all above the plane containing the mobile:

Proposition 3: The geometry of three non-collinear points n_0, n_1, n_2 above the plane containing six coplanar points $m_0, m_1, m_2, m_3, m_4, m_5$, no three of which are collinear, is determined by the distances $\|n_i - m_j\|$ for $i = 0, 1, 2$ and $j = 0, 1, 2, 3, 4, 5$.

As above, the solution geometry can be computed in polynomial time using standard techniques in real constraint optimization for numerically solving systems of polynomial equations (the theory of the reals). Although the minimum number of mobile locations required for a solution is six, using a larger number of points would reduce the error caused by GDOP.

3) Calculating distances among four (or more) nodes: Finally we consider a version of the problem that requires no additional assumptions: compute the pairwise distances between $j \geq 4$ nodes n_1, n_2, $\ldots$, n_j by measuring their distances to various locations of a mobile node m. Now each new position of the mobile node adds j more constraints and only 3 unknowns, for a total reduction in the number of degrees of freedom by $j - 3 \geq 1$. Initially there are $3j-5$ unknowns: three coordinates per stationary node, minus 3 degrees of translational motion and 2 degrees of rotational motion. Thus we require at least $\lceil (3j-5)/(j-3) \rceil$ mobile positions for the number of degrees of freedom to reduce to at most 0.

It is impractical to assume that j is too large, both because it requires a large node density to have so many line-of-sight paths (especially indoors) and because solving the resulting system of polynomial equations grows in difficulty (though for any fixed j it is polynomial). Therefore we focus on the simplest form of this case, $j = 4$. Then $\lceil (3j-5)/(j-3) \rceil = 7$. Again, we find that 7 mobile positions suffice to uniquely determine the geometry, as the degree-of-freedom analysis predicts:

Proposition 4: The geometry of eleven points $n_1, n_2, n_3, n_4, m_1, m_2, m_3, m_4, m_5, m_6, m_7$, no four of which are coplanar, is determined by the distances $\|n_i - m_j\|$ for $i = 1, 2, 3, 4$ and $j = 1, 2, 3, 4, 5, 6, 7$.

The non-coplanarity assumption requires no more than three nodes to be a constant distance from the floor. This property is easy to arrange on the ceiling by varying-length mounts, and is easy to arrange for a mobile human by the difference in elevation while walking.

D. MAL: Movement Strategy

Combining the approaches of the previous subsection for deriving distances between stationary nodes with Theorem 1 characterizing which distances will guarantee global localization, we obtain a natural movement strategy for the mobile to collect these distances:

1) Initialize:
    a) Find four stationary nodes that can all be seen from a common mobile location. (In this description, visibility is defined in terms of whether two node locations can directly measure the distance between each other.)
    b) Move the mobile to at least seven nearby locations and measure distances.
    c) Compute the pairwise distances between the four stationary nodes using Proposition 4.
    d) Localize the resulting tetrahedron according to Theorem 1.

    Alternatively, if some information is known about the stationary node positions or the mobile positions, we can use multiple mobile locations that see just some of the four stationary nodes; we do not elaborate for the initialization step.
2) Loop:
    a) Pick a stationary node that has been localized but has not yet been examined by this loop.
    b) Move the mobile around the (perimeter of) the visibility region of that stationary node (i.e., the set of mobile positions that can see the stationary node), searching for positions from which the mobile can hear a not-yet-localized stationary node as well as zero, one, or two additional localized stationary nodes (depending on the assumptions about node positions).
    c) For each such mobile position:
        i) Compute the distances among those two, three, or four stationary nodes using Proposition 2, 3, or 4.
        ii) If the not-yet-localized stationary node now has four known distances to localized stationary nodes, localize it according to Theorem 1.

This algorithm terminates either when every stationary node has been localized (success) or when no more progress can be made according to Theorem 1 (failure). It is easy to see that the algorithm makes as much progress as possible from its starting point. Furthermore, we can show that success is independent of the particular tetrahedron from which we start. As a consequence we obtain the following "correctness" guarantee:

Theorem 5: The mobile movement strategy described above is guaranteed to find a globally rigid graph on the stationary nodes of the type described in Theorem 1 provided that such a graph can be constructed using one mobile.

We can also bound the performance of the algorithm by observing that we stop searching for distances to a stationary node once it has four known distances:

Theorem 6: The number of distance measurements made by the mobile movement strategy described above is linear in the number of stationary nodes.

The total amount of motion required by the strategy depends on the perimeter of node visibility regions (which is normally small) as well as the amount of travel required between measurement points. To minimize the latter travel, we can make Step 2a more specific to follow a depth-first search in the graph of node visibilities, restricted by the constraints required by Step 2a. Because a newly localized node is always adjacent to a previously localized node, the graph of node visibilities is connected, even with the additional constraints placed by Step 2a on adjacencies. Using a standard amortization on the total length of a depth-first search, we obtain the following performance bound:

Theorem 7: The total distance traveled by the depth-first mobile movement strategy described above is proportional to the product of the number of stationary nodes and the perimeter of a stationary node's visibility region.

IV. AFL: ANCHOR-FREE LOCALIZATION

Once we have obtained enough inter-node distances to build a rigid graph of the nodes, we can run any of several localization schemes to compute node coordinates. Some of these localization schemes assume the availability of a fraction of anchor nodes with already known position information for computing node coordinates [22], [23], [2], [1], [24], while other schemes do not use anchor nodes [5], [3], [4], [25], [26], [7].

However, most of the traditional localization algorithms have been designed for well-connected dense networks of nodes deployed in environments with relatively small number of obstacles. For these algorithms, indoor environments become particularly challenging; indoors, node density is often sparse with only 3 to 4 nodes per room, and has poor connectivity across rooms. As a solution, we have developed an anchor-free localization algorithm called *AFL* that is especially well-suited to low connectivity graphs [6]. In this section, we give a brief overview of AFL, which we use to evaluate the performance of MAL.

AFL runs in two phases, it first computes an initial coordinate assignment for nodes, which results in an unfolded and scaled-up version of the actual physical layout of the graph. In this phase, the 2D version of AFL runs multiple instances of a leader election algorithm to elect 5 nodes as shown in Figure 5; here, the lines joining n_1,n_2 and n_3,n_4 are roughly perpendicular to each other, and n_0 is close to the intersection of these two lines. Next, AFL uses the shortest path hop count from these elected nodes to compute the initial coordinates of each node i. For two nodes i, j, let $h_{i,j}$ and $d_{i,j}$ respectively denote the shortest path hop count and the (true) Euclidean distance between i and j, and let R denote the "range" of the nodes in the graph. The range R determines if i and j are neighbors or not, according to whether $d_{i,j} \leq R$ or $d_{i,j} > R$ (this model is only an approximation of reality). If i and j are neighbors, we denote this relationship by $i \leftrightarrow j$. The initial coordinates of node i, computed by AFL, are given by:

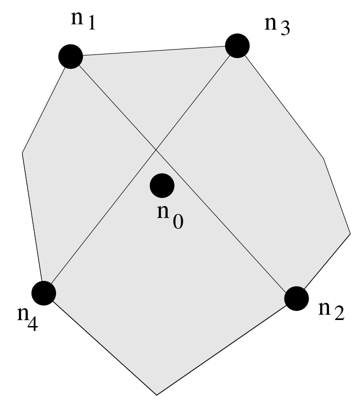

Fig. 5. Nodes elected during the initialization phase of AFL for a well-connected graph.

$$x(i) = Rh_{0,i}\frac{h_{3,i} - h_{4,i}}{\sqrt{(h_{3,i} - h_{4,i})^2 + (h_{1,i} - h_{2,i})^2}}$$

$$y(i) = Rh_{0,i}\frac{h_{1,i} - h_{2,i}}{\sqrt{(h_{3,i} - h_{4,i})^2 + (h_{1,i} - h_{2,i})^2}}$$

AFL's initialization phase uses only node connectivity information, not distance information. This feature makes AFL suitable for indoor environments since pairwise node connectivity (*e.g.,* RF connectivity) is much easier to obtain compared to precise inter-node distances. Although some previous work also compute node coordinates using connectivity information [27], [4], [28], AFL's initialization phase is unique in attempting to compute a coordinate assignment that results in a scaled-up unfolded version of the original graph.

After the initialization phase, AFL uses a non-linear optimization algorithm to minimize the sum-squared energy E of the graph defined by:

$$E = \sum_{i \leftrightarrow j} \|d_m(i,j) - d_c(i,j)\|^2 \quad (1)$$

Here, $d_m(i,j)$ denotes the "measured" distance between the nodes i and j obtained by running MAL; and, $d_c(i,j)$ denotes the "computed" distance between i and j obtained from the current coordinate assignment of the nodes. If $d_m(i,j)$ is equal to the true Euclidean distance between i and j ($d_{i,j}$), then $E = 0$ implies that the current coordinate assignment satisfies the inter-node distances for all $i, j, i \leftrightarrow j$. Because the graph produced by MAL is rigid, $E = 0$ corresponds to a coordinate assignment that is consistent with the true embedded graph. When $d_m(i,j)$ is only approximately equal

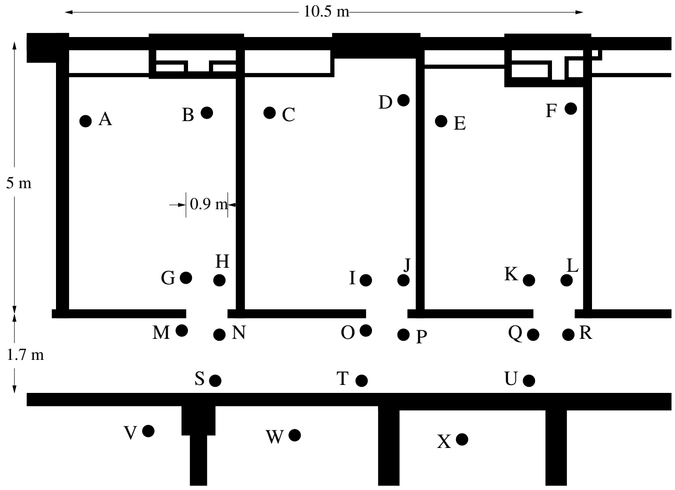

Fig. 6. An indoor deployment of 24 nodes to evaluate the performance of MAL.

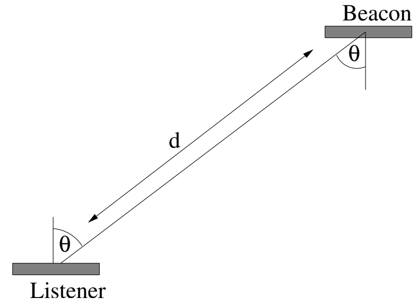

Fig. 7. Cricket ranging accuracy estimation setup.

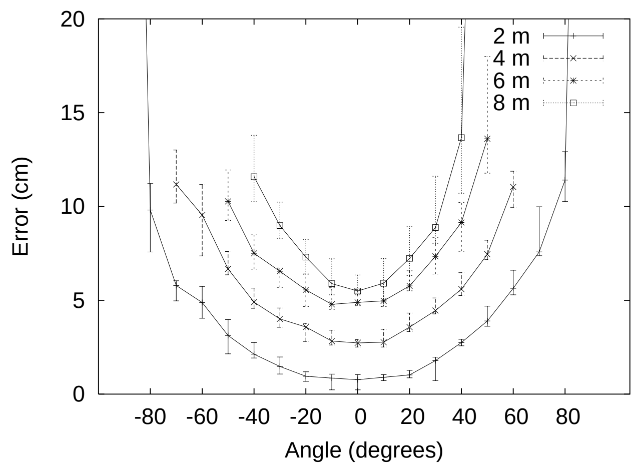

Fig. 8. The Cricket ranging error as a function of the rotation of the beacon and the listener at different beacon to listener distances.

to $d_{i,j}$, the coordinate assignment corresponding to the global minimum of E results in graph that approximates the true embedded graph.

V. Performance Evaluation

In this section we evaluate the performance of MAL, measuring the error characteristics of the pairwise distance estimates it produces and measuring the "end-to-end" localization performance of MAL running in conjunction with AFL. We evaluate performance both using a 24-node real-world Cricket-based testbed and in simulation.

Cricket is an indoor location system that we have developed over the past several years [8]. Cricket consists of two types of nodes: beacons that are attached to the walls and ceiling of a building, and listeners that are attached to various mobile and fixed devices that need to know their location. The beacons periodically transmit location information us an RF signal; at the start of the RF signals, they transmit a narrow ultrasonic signal. The listeners listen to beacon transmissions and compute the distance to nearby beacons using the time-difference-of-arrival of RF and ultrasonic signals. Since the ultrasound signals used in Cricket do not penetrate walls, there should be a line-of-sight path between a beacon and a listener to measure distance. Although Cricket beacons have ultrasound receiving capability, when mounted on the same plane, they cannot measure distance between each other due to the physical properties of the ultrasonic sensors.

A. *Results from deployment*

Our experimental testbed of 24 Cricket nodes deployed indoors is shown in Figure 6. The deployment covers four different rooms, three of which are connected by a common corridor. The only line-of-sight connectivity from one room to the corridor is through the 0.9 m wide door. The rooms have no line of sight connectivity to each other. All the nodes except **O** and **T** were on the ceiling at the same height. Nodes **O** and **T** were on a beam 30 cm below the ceiling.

To compare the distances produced by MAL with the true distances between the nodes, we manually measured the distances between different walls and beacons using a laser range finder; although these distances may contain measurement errors, we will refer to the coordinates obtained from these distances as *true coordinates*, to distinguish them from the coordinates computed using MAL.

First, we examined the distance measurement accuracy of the Cricket system since that determines the overall localization accuracy. We set up a beacon and a listener as shown in Figure 7; this setup mimics a beacon mounted on the ceiling, and a listener held parallel to the ground. Figure 8 shows the error between the measured distance and the true distance for different values of d and θ. Each data point on the graph represents the mean absolute error, calculated over 100 samples; the vertical bars represent the minimum and maximum absolute error within the 100 samples. Because the ultrasonic sensors are not omnidirectional, we could not get distance measurements for (d, θ) combinations that do not have a corresponding data point. We performed the experiment in a controlled environment to prevent outlier distance measurements due to reflected ultrasonic signals. As we observe from this graph, Cricket has $\simeq 0.5\%$ ranging accuracy when the beacon and the listener are 2 m apart and are facing each other; however, the ranging performance degrades as we

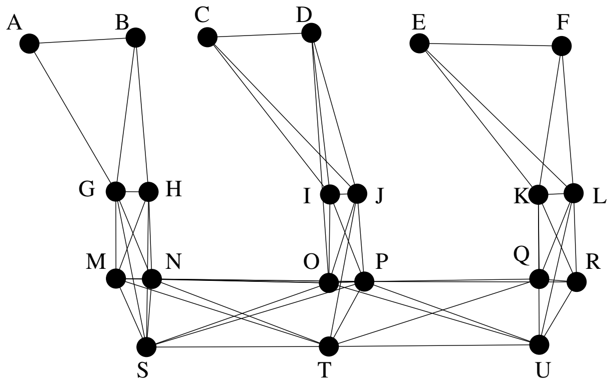

Fig. 9. The node connectivity graph obtained by MAL. Although this graph is only locally rigid, the AFL initialization phase prevents foldings along edges such as G-H during localization.

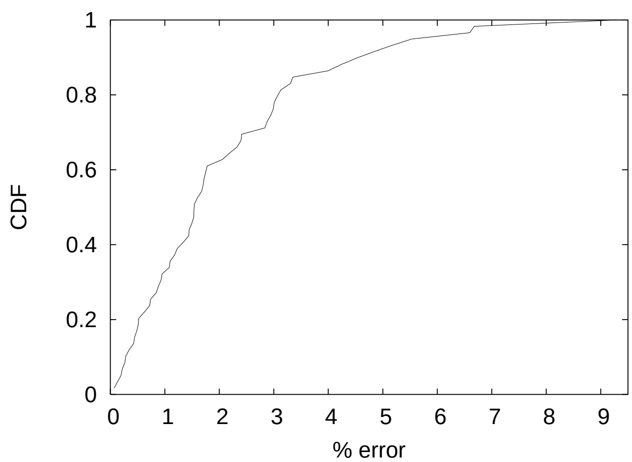

Fig. 10. The CDF of inter-node distance estimate error after filtering and averaging for outlier rejection.

increase the separation and when they do not face each other. However, for the range $(-40^o, 40^o)$, the error is under 5 cm.

1) MAL performance: We collected distance samples using a receiver, mounted on a mobile cart, at 142 cm below the ceiling. We could not collect distance samples from nodes **V**,**W**, and **X**; so we did not attempt to localize these nodes. However, these three nodes were useful for the RF connectivity based initialization phase of AFL.

We collected distance samples by stopping the mobile at 1,592 points. We used the *three nodes at a time* approach described in section III to compute the internode distances. We ran the distance estimation algorithm on 52 different triangles formed by different node combinations. The edges of these triangles represented 59 unique edges connecting the nodes. Figure 9 shows the graph obtained by these edges with nodes at their measured coordinates. We see that MAL enabled us to compute enough edges to build a locally rigid graph from a collection of disconnected nodes. This graph is only locally rigid since sections of the graph can fold along edges such as K-L, B-G while preserving edge lengths. However, as we see later, AFL managed to avoid such folds during localization

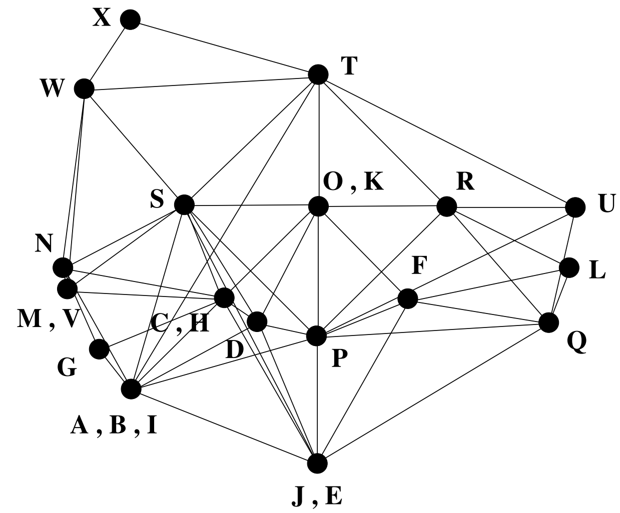

Fig. 11. Graph obtained after running the AFL initialization using the RF connectivity information.

since AFL initialization phase generates an approximately fold-free initial coordinate assignment.

Ultrasonic propagation effects such as bending and reflection off obstacles introduces errors. We used the following easy solution to this problem. We had multiple distance estimates between a given pair of nodes, since an edge is typically shared by several triangles. Since the magnitude of measurement error depends on the position of the mobile, we were able to filter out outliers using a simple binning and majority election algorithm [8]. After filtering, we computed a given edge length by averaging the estimates from different triangles.

Figure 10 shows the CDF of the % edge length error of the distances estimates obtained using MAL, after filtering and averaging to remove outliers. We observe that the distance estimation error is smaller than 1.5% over 50% of the time, and the 90th percentile has $\simeq 5\%$ error. This graph indicates that MAL can provide accurate pairwise node information. We observe that there is a wide range of percentage error values, which we attribute to the differences in the area and the shape of individual triangles, and the restrictions on the coverage area of the listener due to physical obstacles such as furniture. In section V-B, we use simulations to study how these factors affect inter-node distance estimation accuracy.

2) AFL performance: Although indoor RF propagation is highly erratic, RF connectivity and signal strength can be used to obtain coarse granularity location information. Since AFL's initialization phase uses only connectivity information and since RF permeates more thoroughly than ultrasound, RF connectivity is useful for the initialization phase of AFL. In our implementation, we use a simple strategy for measuring RF connectivity. The nodes periodically broadcast RF messages at an average rate of 1 message per second. Each node keeps track of the number of messages it received from other nodes, and it times out these message counts using an expiration timer whose value is inversely proportional to the message arrival rate. This approach filters out far-away nodes, and the nodes

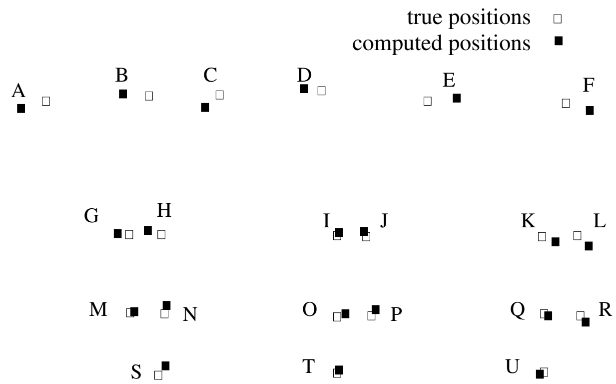

Fig. 12. Coordinates obtained after running the AFL optimization, in comparison with the original node positions.

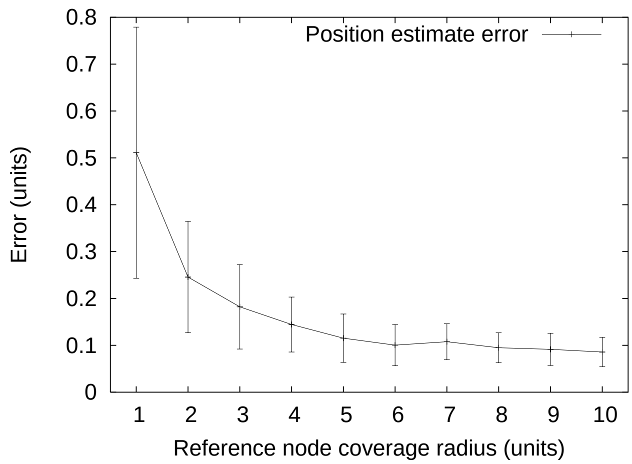

Fig. 13. The position estimate error as a function of the radius of the reference node coverage area.

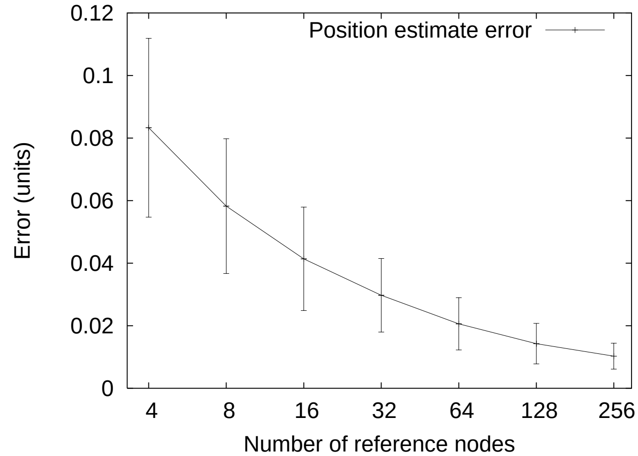

Fig. 14. The position estimate error as a function of the number of reference nodes.

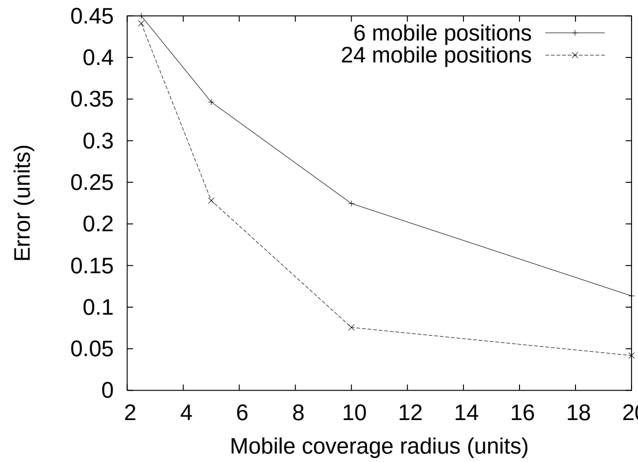

Fig. 15. The average edge length error as a function of the radius of the mobile coverage area.

with a high message arrival rate are considered neighbors.

Figure 11 shows the layout resulting from the initial coordinate assignment obtained by running Phase 1 of AFL. Figure 12 shows the coordinates obtained by running Phase 2 on the previous topology and the true node coordinates obtained by manual measurements. The error between the estimated and true node positions is small, comparable to the errors from the MAL algorithm (Figure 10). These results demonstrate that MAL and AFL can work well in practice.

B. Simulation Results

This section presents the results of running several simulations of the MAL algorithm. Although MAL has theoretical correctness and performance guarantees, it is important to understand how well it performs under errors, scale, and various layout geometries. Due to lack of space, we do not present simulation results of MAL running in conjunction with AFL (we showed these results for the real-world deployment).

1) Impact of GDOP on localization error: We start with some experiments to evaluate the impact of GDOP on location estimation using the following configuration. We have n fixed reference nodes, uniformly spaced, on a circle with radius r. We place a node m, 10 units away from the circle, on the perpendicular passing through the center of the circle. We introduce a uniformly distributed random error in the range $(-0.1, 0.1)$ units on the distances between the reference node and m. We compute the position estimate of m that minimizes the sum-squared-error for different values of r. Figure 13 plots the position estimate error of m, computed by the distance between the estimated and true positions, as a function of r for $n = 4$; each point on the graph represents 100 simulations. We observe that the error decreases with increasing r. Since we have kept the measurement error distribution constant, this graph shows the impact of geometry on the position estimate accuracy. It also shows the importance of reference points that cover a large area for accurate position estimation.

Figure 14 shows how position estimate error changes with n for $r = 10$. The position estimate error decreases with increasing n as positive and negative errors tend to cancel out with large n. This implies that we can improve position estimation accuracy using large number of measurements.

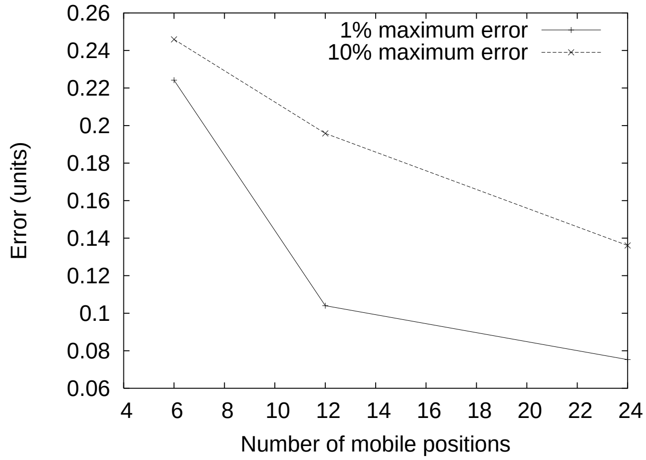

Fig. 16. The average edge length error as a function of the number of mobile positions.

2) MAL performance: Next, we evaluate the performance of MAL as we vary the area covered by the mobile unit and the number of measurements. We selected 3 nodes, representing the fixed nodes, with (x, y) coordinates at randomly selected points on a circle of radius 10 units, with the restriction that the angle incident on the center of the circle by any two points is $> 10^o$. The z coordinates of the points were uniformly distributed between 2.5 and 5.0 units. We selected n mobile node positions uniformly distributed within a concentric circle of radius r, on $z = 0$. To achieve a uniform distribution, we placed a bounding circle of radius r' at each mobile point and iterated over different values of r'.

We examine MAL performance as we vary the mobile node coverage area. Figure 15 shows the average error in computing internode distances among three nodes as we vary r for both $n = 6$ and $n = 24$; We introduced a (1%,-1%) uniformly distributed error on mobile-to-fixed node distance estimates. Each point represents 20 simulations. We observe that a larger mobile coverage area reduces the distance estimate error. This result indicates that MAL performs better when the mobile collects samples within a large coverage area.

Next, we examine the MAL performance as we vary the number of mobile node positions. Figure 16 plots the average distance computation error Vs n for both $\pm 1\%$ and $\pm 10\%$ uniformly distributed mobile-to-fixed node distance error. As expected, the average error decreases with increasing n. This result demonstrates a significant advantage of the MAL approach, where we can obtain a large number of mobile distance estimates at little extra cost on the infrastructure (assuming we can neglect the cost associated with a mobile collecting data).

VI. Related Work

Scott and Hazas examine different approaches to determine fixed node positions using distance estimates [29]. Their experiments include both distances obtained at nodes mounted on a mobile frame and raw distances obtained by placing multiple nodes on the floor or from a mobile carried by users. They report better results using the mobile-frame based approach compared to the raw distance approach (however, the paper does not report the size of the fixed frame used). In the raw distance approach, they used simulated annealing to optimize the positions of all the nodes in parallel, which can degrade performance due to the presence of local minima in the objective function. In contrast, we break the localization problem to two manageable pieces. We use rigidity theory to determine the minimum number of nodes and samples needed per one small group. Our use of groups with small number of nodes reduces the possibility of local minima and also makes the localization algorithm scalable.

Pathirana *et al.* use a mobile robot to localize RF beacons [30]. They assume the availability of precise velocity and acceleration of mobile robot. They obtain distance information between the robot and fixed nodes using RF signal strength. The use of the mobile robot improves the accuracy of RF signal strength based distance measurements, since signal strength variations due to spatial fading of RF signals may be reduced. Corke *et al.* use a flying robot equipped with a GPS receiver to localize stationary nodes [31]. The robot beams down its current GPS coordinates using RF; and the stationary nodes use this information to compute their position. Sichitiu and Ramadurai also use a GPS equipped mobile node to localize fixed receiver nodes; they use the RF signal strength to measure the distance between the mobile node and fixed nodes [32]. These approaches for node localization are similar since they all use a mobile node with known location information to localize a collection of fixed nodes; however they use different mechanism to determine the mobile node position and different algorithms to compute fixed node positions. In contrast to these approaches, we do not assume the availability of location information at the mobile node. However, if accurate distance information between different mobile positions is available—from the robotic odometric system, for instance—we can harness this additional information to improve the MAL performance.

Indoor location systems such as Cricket, Bat, and RADAR use distance or signal strength estimates from fixed reference nodes to determine mobile user positions [8], [10], [33]. These reference nodes needs to be pre-calibrated with their own position; this is currently done by manual measurements. In a building-wide deployment, due to the lack of line of sight among nodes, such manual measurements would require the combination of a map of the building and distance measurements to the walls of the building. However we are interested in an automated approach which does not require a map of the building since we may want use the indoor location system itself to generate the map of the building.

Previous work on anchor-based localization algorithms use inter-node distance estimates and a fraction of nodes with known location information to compute the location information of the rest of the nodes [22], [23], [1], [24], [2]. Anchor free algorithms, such as the AFL algorithm, compute a relative coordinate assignment to nodes based only on internode distance estimates; these algorithms are particularly important for both indoor and ad-hoc deployment since they do not depend on an external location system [5], [4], [25], [3], [26], [7], [28]. All of these algorithms can use the inter-node distance estimate from MAL as an input for node localization.

VII. Conclusion

Most previously proposed approaches to the localization problem assume that the nodes can obtain pairwise distance information using local ranging. Unfortunately, for a variety of reasons that include obstructions and lack of reliable omnidirectional ranging (e.g., using ultrasound), this distance information is hard to obtain in practice. Even when pairwise distances between nearby nodes are known, there may not be enough information to uniquely solve the coordinate assignment problem.

This paper described MAL, a *mobile-assisted localization* method, in which a roving mobile user or robot wanders through the network and collects distance estimates to nodes at various locations. We showed how to constrain this movement such that the roving node can gather sufficient distance samples to solve the localization problem. We gave an algorithm that, given sufficiently many distance samples, produces a consistent coordinate assignment. We evaluated the algorithm's performance using simulations and real-world experiments. Our results show that the median pairwise node distance error in a real-world deployment is less than 1.5% of the distance between the nodes; similar results are confirmed by several simulations as well.

Acknowledgment

This work was funded by NSF under grant number ITR ANI-0205445, by the MIT Project Oxygen partnership, and by a grant from Intel Corporation.

References

[1] C. Savarese, J. Rabaey, and J. Beutel, "Locationing in Distributed Ad-Hoc Wireless Sensor Networks," in *Proc. IEEE ICASSP*, Salt Lake City, UT, May 2001, pp. 2037–2040.

[2] A. Savvides, C. Han, and M. Srivastava, "Dynamic Fine-Grained Localization in Ad-Hoc Networks of Sensors," in *Proc. 7th ACM MOBICOM*, July 2001, pp. 166–179.

[3] R. L. Moses, D. Krishnamurthy, and R. M. Patterson, "A self-localization method for wireless sensor networks," *Eurasip Journal on Applied Signal Processing*, March 2003, special Issue on Sensor Networks.

[4] R. Nagpal, H. Shrobe, and J. Bachrach, "Organizing a Global Coordinate System from Local Information on an Ad Hoc Sensor Network," in *Proc. 2nd IPSN*, Palo Alto, CA, 2003.

[5] Y. Shang, W. Ruml, Y. Zhang, and M. Fromherz, "Localization from mere connectivity," in *Proc. 4th ACM MOBIHOC*, Annapolis, Maryland, 2003, pp. 201–212.

[6] N. Priyantha, H. Balakrishnan, E. Demaine, and S. Teller, "Anchor-Free Distributed Localization in Sensor Networks," MIT Laboratory for Computer Science, Tech. Rep. 892, April 2003.

[7] D. Moore, J. Leonard, D. Rus, and S. Teller, "Robust distributed network localization with noisy range measurements," in *Proc. 2nd ACM SenSys*. Baltimore, MD: ACM Press, November 2004, pp. 50–61.

[8] N. Priyantha, A. Chakraborty, and H. Balakrishnan, "The Cricket Location-Support System," in *Proc. 6th ACM MOBICOM*, Boston, MA, Aug. 2000.

[9] N. Priyantha, A. Miu, H. Balakrishnan, and S. Teller, "The Cricket Compass for Context-Aware Mobile Applications," in *Proc. 7th ACM MOBICOM*, Rome, Italy, July 2001.

[10] A. Harter, A. Hopper, P. Steggles, A. Ward, and P. Webster, "The Anatomy of a Context-Aware Application," in *Proc. 5th ACM MOBICOM*, Seattle, WA, Aug. 1999.

[11] M. A. Spirito, "Accuracy of Hyperbolic Mobile Station Location in Cellular Networks," *Electronics Letters*, vol. 37, no. 11, pp. 708–710, 2001.

[12] A. Savvides, W. Garber, A. Sachin, R. Moses, and M. Srivastava, "On the Error Characteristics of Multihop Node Localization in Ad-Hoc Sensor Networks," in *Proc. 2nd IPSN*, Palo Alto, CA, 2003.

[13] B. Hendrickson, "Conditions for unique graph realizations," *SIAM Journal on Computing*, vol. 21, no. 1, pp. 65–84, 1992. [Online]. Available: ftp://ftp.cs.sandia.gov/pub/papers/bahendr/nec.ps.gz

[14] R. Connelly, "Rigidity," in *Handbook of Convex Geometry*. Amsterdam: North-Holland, 1993, vol. A, pp. 223–271.

[15] J. Graver, *Counting on Frameworks: Mathematics to Aid the Design of Rigid Structures*. Mathematical Association of America, 2001.

[16] J. Graver, B. Servatius, and H. Servatius, *Combinatorial Rigidity*. American Mathematical Society, 1993.

[17] R. Connelly, "On generic global rigidity," in *Applied Geometry and Discrete Mathematics*. American Mathematical Society, 1991, pp. 147–155.

[18] B. Jackson and T. Jordán, "Connected rigidity matroids and unique realizations of graphs," Egerváry Research Group on Combinatorial Optimization, Budapest, Hungary, Tech. Rep. TR-2002-12, March 2003.

[19] C. Coullard and A. Lubiw, "Distance visibility graphs," *International Journal of Computational Geometry and Applications*, vol. 2, no. 4, pp. 349–362, 1992.

[20] D. Niculescu and B. Nath, "Ad Hoc Positioning System (APS) Using AOA," in *Proc. 22nd IEEE INFOCOM*, Salt Lake City, UT, April 2003, pp. 2037–2040.

[21] A. Miu, "Design and Implementation of an Indoor Mobile Navigation System," Master's thesis, Massachusetts Institute of Technology, Jan. 2002.

[22] N. Bulusu, J. Heidemann, and D. Estrin, "GPS-less Low Cost Outdoor Localization For Very Small Devices," Computer Science Department, University of Southern California, Tech. Rep. 00-729, Apr. 2000.

[23] L. Doherty, K. Pister, and L. Ghaoui, "Convex position estimation in wireless sensor networks," in *Proc. 20th IEEE INFOCOM*, April 2001.

[24] C. Savarese, J. Rabaey, and K. Langendoen, "Robust Positioning Algorithms for Distributed Ad-Hoc Wireless Sensor Networks," in *USENIX Annual Technical Conference, General Track*, Monterey, CA, June 2002, pp. 317–327.

[25] S. Capkun, M. Hamdi, and J. Hubaux, "GPS-free positioning in mobile ad-hoc networks," in *Proc. 34th HICSS*, 2001.

[26] A. Howard, M. Mataric, and G. Sukhatme, "Relaxation on a mesh: A formalism for generalized localization," in *Proc. IEEE/RSJ IROS*, Wailea, Hawaii, Oct. 2001.

[27] A. Rao, S. Ratnasamy, C. Papadimitriou, S. Shenker, and I. Stoica, "Geographic routing without location information," in *Proc. 9th ACM MOBICOM*, San Diego, CA, Sept. 2003, pp. 96–108.

[28] T. Moscibroda, R. O'Dell, M. Wattenhofer, and R. Wattenhofer, "Virtual coordinates for ad hoc and sensor networks," in *Proc. 2nd DIALM-POMC*, Philadelphia, PA, Oct. 2004.

[29] J. Scott and M. Hazas, "User-Friendly Surveying Techniques for Location-Aware Systems," in *Proc. 5th UbiComp*. Seattle, WA: Springer-Verlag, October 2003, pp. 44–53.

[30] P. Pathirana, A. Savkin, S. Jha, and N. Bulusu, "Node localization using mobile robots in delay-tolerant sensor networks," *IEEE Trans. Mobile Computing*, 2005, to appear.

[31] P. Corke, R. Peterson, and D. Rus, "Networked robots: Flying robot navigation using a sensor net," in *Proc. 11th ISRR*, Siena, Italy, Nov. 2003.

[32] M. Sichitiu and V. Ramadurai, "Localization of wireless sensor networks with a mobile beacon," in *Proc. 1st IEEE MASS*, Fort Lauderdale, FL, Oct. 2004, pp. 174–183.

[33] P. Bahl and V. Padmanabhan, "RADAR: An In-Building RF-based User Location and Tracking System," in *Proc. 19th IEEE INFOCOM*, Tel-Aviv, Israel, Mar. 2000.