Localization from Mere Connectivity

Yi Shang
University of Missouri–Columbia
Columbia, MO 65211
shangy@missouri.edu

Wheeler Ruml, Ying Zhang,
Markus P. J. Fromherz
Palo Alto Research Center
Palo Alto, CA 94304
{ruml, yzhang, fromherz}@parc.com

ABSTRACT

It is often useful to know the geographic positions of nodes in a communications network, but adding GPS receivers or other sophisticated sensors to every node can be expensive. We present an algorithm that uses connectivity information—who is within communications range of whom—to derive the locations of the nodes in the network. The method can take advantage of additional information, such as estimated distances between neighbors or known positions for certain anchor nodes, if it is available. The algorithm is based on multidimensional scaling, a data analysis technique that takes $O(n^3)$ time for a network of n nodes. Through simulation studies, we demonstrate that the algorithm is more robust to measurement error than previous proposals, especially when nodes are positioned relatively uniformly throughout the plane. Furthermore, it can achieve comparable results using many fewer anchor nodes than previous methods, and even yields relative coordinates when no anchor nodes are available.

Categories and Subject Descriptors

C.2.3 [**Computer Systems Organization**]: Network Operations

General Terms

Algorithms, Performance

Keywords

Position estimation, node localization, multilateration, multidimensional scaling, ad-hoc networks, sensor networks

1. INTRODUCTION

Large-scale networks with hundreds and even thousands of very small, battery-powered and wirelessly connected sensor and actuator nodes are becoming a reality [5]. For example, future sensor networks will involve a very large number of densely deployed sensor nodes over physical space. In particular, the nodes are typically highly resource-constrained (processor, memory, and power), have limited communication range, are prone to failure, and are put together in ad-hoc networks.

Imagine a network of sensors sprinkled across a large building or an area such as a forest. Typical tasks for such networks are to send a message to a node at a given location (without knowing which node or nodes are there, or how to get there), to retrieve sensor data (e.g., sound or temperature levels) from nodes in a given region, and to find nodes with sensor data in a given range. Most of these tasks require knowing the positions of the nodes, or at least relative positions among them. With a network of thousands of nodes, it is unlikely that the position of each node has been pre-determined. Nodes could be equipped with a global positioning system (GPS) to provide them with absolute position, but this is currently a costly solution.

In this paper, we present a method for computing the positions of nodes given only basic information that is likely to be already available, namely, which nodes are within communications range of which others. The method, MDS-MAP, has three steps. Starting with the given network connectivity information, we first use an all-pairs shortest-paths algorithm to roughly estimate the distance between each possible pair of nodes. Then we use multidimensional scaling (MDS), a technique from mathematical psychology, to derive node locations that fit those estimated distances. Finally, we normalize the resulting coordinates to take into account any nodes whose positions are known.

As we will demonstrate, this simple technique often outperforms existing methods. Furthermore, it requires only connectivity information to produce a meaningful result. If the distances between neighboring nodes can be estimated, that information can be easily incorporated into the pairwise shortest-path computation during the first step of the algorithm. MDS yields coordinates that provide the best fit to the estimated pairwise distances, but which lie at an arbitrary rotation and translation. If the coordinates of any nodes are known, they can be used to derive the affine transformation of the MDS coordinates that allows the best match to the known positions. Only three such 'anchor nodes' are necessary to provide absolute positions for all the nodes in the network.

The next section of the paper describes MDS-MAP in more detail. We will then provide an overview of previous proposals before presenting our empirical evaluation. Our presentation focuses on a centralized version of the algorithm, al-

MobiHoc'03, June 1–3, 2003, Annapolis, Maryland, USA.

though we will briefly mention how the computation can be distributed. We will examine the performance of MDS-MAP on networks of 100 to 200 nodes, with node locations either chosen randomly or according to a rough grid or hexagon layout. We consider a variety of node densities (nodes per communications radius) and a variety of ranging errors when using the estimated distances between neighbors. We will see that MDS-MAP recovers more accurate maps of node locations while using much less information.

2. LOCALIZATION USING MDS-MAP

We consider the node localization problem under two different scenarios. In the first, only proximity (or connectivity) information is available. Each node only knows what nodes are nearby, presumably by means of some local communication channel such as radio or sound, but not how far away these neighbors are or in what direction they lie. In the second scenario, the proximity information is enhanced by knowing the distances, perhaps with limited accuracy, between neighboring nodes.

In both cases, the network is represented as an undirected graph with vertices V and edges E. The vertices correspond to the nodes, of which there exist $m \geq 0$ special nodes with known positions, which we will call anchors. For the proximity-only case, the edges in the graph correspond to the connectivity information. For the case with known distances to neighbors, the edges are associated with values corresponding to the estimated distances. We assume that all the nodes being considered in the positioning problem form a connected graph, i.e., there is a path between every pair of nodes.

There are two possible outputs when solving the localization problem. One is a relative map and the other is an absolute map. The task of finding a relative map is to find an embedding of the nodes into either two- or three-dimensional space that results in the same neighbor relationships as the underlying network. Such a relative map can provide correct and useful information even though it does not necessarily include accurate absolute coordinates for each node. Relative information may be all that is obtainable in situations in which powerful sensors or expensive infrastructure cannot be installed, or when there are not enough anchors present to uniquely determine the absolute positions of the nodes. Furthermore, some applications only require relative positions of nodes, such as in some direction-based routing algorithms. Sometimes, however, an absolute map is required. The task of finding an absolute map is to determine the absolute geographic coordinates of all the nodes. This is needed in applications such as geographic routing and target discovering and tracking.

As we will show below, our method can potentially generate both results, depending on the number of anchor nodes. The method first generates a relative map of the network and then transforms it to absolute positions if sufficient anchors are available. Before we describe the details of our method, we first introduce multidimensional scaling (MDS), with a focus on classical MDS, which is used to generate the relative map.

2.1 Multidimensional Scaling (MDS)

Imagine a small cloud of colored beads suspended in mid-air. To characterize the arrangement, one could measure the straight-line distance between each pair of beads. If the cloud were shattered and the beads fell to the floor, one could imagine trying to recreate the arrangement based on the recorded interpoint distances. One would try to determine a location for each bead such that the distances in the new arrangement matched the desired distances. This recreation process is exactly the problem that multidimensional scaling (MDS) solves. Intuitively, it is clear that while the $O(n^2)$ distances will be more than enough to determine $O(n)$ coordinates, the result of MDS will be an arbitrarily rotated and flipped version of the true original layout because the interpoint distances make no reference to any absolute coordinates.

MDS has its origins in psychometrics and psychophysics. It can be seen as a set of data analysis techniques that display the structure of distance-like data as a geometrical picture [1]. MDS starts with one or more distance matrices (or similarity matrices) that are presumed to have been derived from points in a multidimensional space. It is usually used to find a placement of the points in a low-dimensional space, usually two- or three-dimensional, where the distances between points resemble the original similarities. MDS is often used as part of exploratory data analysis or information visualization. By visualizing objects as points in a low-dimensional space, the complexity in the original data matrix can often be reduced while preserving the essential information. MDS is closely related to principal component analysis, and is also related to factor analysis and cluster analysis.

There are many types of MDS techniques. They can be classified according to whether the similarity data is qualitative (nonmetric MDS) or quantitative (metric MDS). They can also be classified according to the number of similarity matrices and the nature of the MDS model. Classical MDS uses one matrix. Replicated MDS uses several matrices, representing distances measurements taken from several subjects or under different conditions. Weighted MDS uses a distance model which assigns a different weight to each dimension. Finally, there is a distinction between deterministic and probabilistic MDS. In deterministic MDS, each object is represented as a single point in a multidimensional space, whereas in probabilistic MDS each object is represented as a probability distribution over the entire space.

We focus on classical metric MDS in this paper. Classical metric MDS is the simplest case of MDS: the data is quantitative and the proximities of objects are treated as distances in a Euclidean space [15]. The goal of metric MDS is to find a configuration of points in a multidimensional space such that the inter-point distances are related to the provided proximities by some transformation (e.g., a linear transformation). If the proximity data were measured without error in a Euclidean space, then classical metric MDS would exactly recreate the configuration of points. In practice, the technique tolerates error gracefully, due to the overdetermined nature of the solution. Because classical metric MDS has a closed-form solution, it can be performed efficiently on large matrices.

Let p_{ij} refer to the proximity measure between objects i and j. The Euclidean distance between two points $X_i = (x_{i1}, x_{i2}, \cdots, x_{im})$ and $X_j = (x_{j1}, x_{j2}, \cdots, x_{jm})$ in an m-

dimensional space is

$$d_{ij} = \sqrt{\sum_{k=1}^{m} (x_{ik} - x_{jk})^2}.$$

When the geometrical model fits the proximity data perfectly, the Euclidean distances are related to the proximities by a transformation $d_{ij} = f(p_{ij})$. In classical metric MDS, a linear transformation model is assumed, i.e., $d_{ij} = a + bp_{ij}$.

The distances D are determined so that they are as close to the proximities P as possible. There are a variety of ways to define "close". A common one is a least-squares definition, which is used by classical metric MDS. In this case, we define

$$I(P) = D + E$$

where $I(P)$ is a linear transformation of the proximities, and E is a matrix of errors (residuals). Since D is a function of the coordinates X, the goal of classical metric MDS is to calculate the X such that the sum of squares of E is minimized, subject to suitable normalization of X.

In classical metric MDS, P is shifted to the center and coordinates X can be computed from the double centered P through singular value decomposition (SVD). For an $n \times n$ P matrix for n points and m dimensions of each point, it can be shown that

$$\begin{aligned} & -\frac{1}{2}(p_{ij}^2 - \frac{1}{n}\sum_{j=1}^{n} p_{ij}^2 - \frac{1}{n}\sum_{i=1}^{n} p_{ij}^2 + \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n} p_{ij}^2) \\ = & \sum_{k=1}^{m} x_{ik}x_{jk} \qquad (1) \end{aligned}$$

The double centered matrix on the left hand side (call it B) is symmetric and positive semidefinite. Performing singular value decomposition on B gives us $B = VAV$. The coordinate matrix becomes $X = VA^{1/2}$.

Retaining the first r largest eigenvalues and eigenvectors ($r < m$) leads to a solution in lower dimension. This implies that the summation over k in Eq. (1) runs from 1 to r instead of m. This is the best low-rank approximation in the least-squares sense. For example, for a 2-D network, we take the first 2 largest eigenvalues and eigenvectors to construct the best 2-D approximation. For a 3-D network, we take the first 3 largest eigenvalues and eigenvectors to construct the best 3-D approximation.

In nonmetric (ordinal) MDS (first developed by Shepard [14]), the goal is to establish a monotonic relationship between inter-point distances and the desired distances. Instead of trying to directly match the given distances, one is satisfied if the distances between the points in the solution fall in the same ranked order as the corresponding distances in the input matrix. The advantage of nonmetric MDS is that no assumptions need to be made about the underlying transformation function. The only assumption is that the data is measured at the ordinal level. Just as classical MDS, nonmetric MDS can also be applied to the localization problem. By adopting a more flexible model, the effects of a few highly incorrect measurements might be more easily tolerated.

2.2 The MDSMAP Algorithm

Based on classical MDS, our method, called MDS-MAP, consists of three steps:

1. Compute shortest paths between all pairs of nodes in the region of consideration. The shortest path distances are used to construct the distance matrix for MDS.
2. Apply classical MDS to the distance matrix, retaining the first 2 (or 3) largest eigenvalues and eigenvectors to construct a 2-D (or 3-D) relative map.
3. Given sufficient anchor nodes (3 or more for 2-D, 4 or more for 3-D), transform the relative map to an absolute map based on the absolute positions of anchors.

In step 1, we first assign distances to the edges in the connectivity graph. When the distance of a pair of neighbor nodes is known, the value of the corresponding edge is the measured distance. When we only have connectivity information, a simple approximation is to assign value 1 to all edges. Then a classical all-pairs shortest-path algorithm, such as Dijkstra's or Floyd's algorithm, can be applied. The time complexity is $O(n^3)$, where n is the number of nodes.

In step 2, classical MDS is applied directly to the distance matrix. The core of classical MDS is singular value decomposition, which has complexity of $O(n^3)$. The result of MDS is a relative map that gives a location for each node. Although these locations may be accurate relative to one another, the entire map will be arbitrarily rotated and flipped relative to the true node positions.

In step 3, the relative map is transformed through linear transformations, which include scaling, rotation, and reflection. The goal is to minimize the sum of squares of the errors between the true positions of the anchors and their transformed positions in the MDS map. Computing the transformation parameters takes $O(m^2)$ time, where m is the number of anchors. Applying the transformation to the whole relative map takes $O(n)$ time.

3. RELATED WORK

Node localization has been a topic of active research in recent years. A detailed survey of the area is provided by Hightower and Borriello [6]. However, few approaches for locating nodes in an ad-hoc network are described. Most systems use some kind of range or distance information and many of them rely on powerful beacon nodes with extreme capabilities, such as radio or laser ranging devices.

Doherty's [4] convex constraint satisfaction approach formulates the localization problem as a feasibility problem with radial constraints. Nodes which can hear each other are constrained to lie within a certain distance of each other. This convex constraint problem is in turn solved by efficient semi-definite programming (an interior point method) to find a globally optimal solution. For the case with directional communication, the method formulates the localization problem as a linear programming problem, which is solved by an interior point method. The method requires centralized computation. For the technique to work well, it needs anchor nodes to be placed on the outer boundary, preferably at the corners. Only in this configuration are the constraints tight enough to yield a useful configuration. When all anchors are located in the interior of the network, the position estimation of outer nodes can easily collapse toward the center, which leads to large estimation errors. For example, with 10% anchors, the error of unknowns is on the order of the radio range. With 5 anchors in a 200-node random network, the error of unknowns is more than twice the radio range.

Most localization methods for ad-hoc networks require more information than just connectivity and use more powerful beacon nodes. The ad-hoc localization techniques used in mobile robots usually fall into this category [7, 10]. Mobile robots use additional odometric measurements for estimating the initial robot positions, which are not available in sensor networks.

Many existing localization techniques for networks use distance or angle measurements from a fixed set of reference points or anchor nodes and apply multilateration or triangulation techniques to find coordinates of unknown nodes [8, 11]. The distance estimates can be obtained from received signal strength (RSSI) or time-of-arrival (ToA) measurements. Due to nonuniform signal propagation environments, RSSI methods are not very reliable and accurate. ToA methods have better accuracy, but may require additional hardware at the sensor nodes to receive a signal that has a smaller propagation speed than radio, such as ultrasound [12]. Emphasis has been put on algorithms that can be executed in a distributed fashion on the sensor nodes without centralized computation, communication, or information propagation. The "DV-based" approach by Niculescu and Nath [9] is distributed. The "DV-hop" method achieves an location error of about 45% of the radio range for networks with 100 nodes, 5 anchors, and average connectivity 7.6. It starts with the anchor nodes. The anchors flood their location to all nodes in the network. Each unknown node performs a triangulation to three or more anchors to estimate its own position. The method works well in dense and regular topologies. For sparse and irregular networks, the accuracy degrades to the radio range. The "DV-distance" method uses distance between neighboring nodes and reduces the location error by about half of that of "DV-hop".

Savarese et al. propose another distributed method [11]. The method consists of two phases: start-up and refinement. For the start-up phase, they use Hop-TERRAIN, an algorithm similar to DV-hop. Hop-TERRAIN is run once at the beginning to generate a rough initial estimate of the nodes' locations. Again, it needs at least 3 anchor nodes to start. Then the refinement algorithm is run iteratively to improve and refine the position estimates. The algorithm is concerned only with nodes within a one-hop neighborhood and uses a least-squares triangulation method to determine a node's position based on its neighbors' positions and distances to them. The approach can deliver localization accuracy within one third of the communication range.

When the number of anchor nodes is high, the collaborative multilateration approach by Savvides et al. can be used [13]. The method estimates node locations by using anchor locations that are several hops away and distance measurements to neighboring nodes. A global nonlinear optimization problem is solved. The method has three main phases: 1) formation of collaborative subtree, which only includes nodes that can be uniquely determined, 2) computation of initial estimates with respect to anchor nodes, 3) position refinement by minimizing the residuals between the measured distances between the nodes and the distances computed using the node location estimates. They present both a centralized computation model and a distributed approximation of the centralized model. The method works well when the fraction of anchor nodes is high.

The GPS-less system by Bulusu et al. [3] employs a grid of beacon nodes with known positions. Each unknown node sets its position to the centroid of the beacons near the unknown. The position accuracy is about one-third of the separation distance between beacons, so the method needs a high beacon density to work well.

Almost all the existing methods need some kind of anchor or beacon nodes to start with. Our method does not have this limitation. It builds a relative map of the nodes even without anchor nodes. With three or more anchor nodes, the relative map can be transformed and absolute coordinates of the nodes are computed. Our method works well in situations with low ratios of anchor nodes and performs even better on regular networks. A limitation of the current implementation is that it is centralized. There are various ways to apply this method in a decentralized or distributed fashion. For example, the method can be applied to sub-networks to obtained regional relative maps, which are patched together to form an overall map of the network.

4. EXPERIMENTAL RESULTS

In our experiments, we ran MDS-MAP on various topologies of networks in Matlab. The nodes are placed (a) randomly with a uniform distribution within a square area, (b) on a square grid with some placement errors, or (c) on a hexagonal grid with some placement errors. In a square grid, assuming r is the unit length, n^2 nodes are typically placed in an nr by nr square. We model placement errors for the grid layout as Gaussian noises. With a placement error e_p, a random value drawing from a normal distribution $r \times e_p \times N(0,1)$ is added to the node's original grid position. The placement error in a hexagonal grid is defined similarly. The anchor nodes are selected randomly. The data points represent averages over 30 trials in networks containing 100 to 200 nodes.

In the connectivity-only cases, each node only knows the identities of nodes in its neighborhood but not the distance to them. In the known-distance cases, each node knows the distances to its neighbor nodes. The distance information is modeled as the true distance blurred with Gaussian noise. Assume the true distance is d^* and range error is e_r; then the measured distance is a random value drawing from a normal distribution $d^*(1+N(0,e_r))$. The connectivity (average number of neighbors) is controlled by specifying radio range R. To compare with previous results in [4, 11], the errors of position estimates are normalized to R (i.e., 50% position error means half of the range of the radio). We do not consider models of non-uniform radio propagation or widely varying ranging errors. Both modeling these phenomena and simulating their effects are very important directions for future work.

4.1 Random Placement

In this set of experiments, 200 nodes are placed randomly in a $10r \times 10r$ square. Figure 1 shows an example using a radio range of $1.5r$, which leads to an average connectivity of 12.1. In the graph, points represent nodes and edges represent the connections between neighbors who can hear each other. Figure 2 shows the result of MDS-MAP based on the connectivity information of the network. Figure 2(a) shows the intermediate result of classical MDS on the connectivity matrix. It can be seen that the center of the map is the origin (0,0), and it has a different orientation than the original

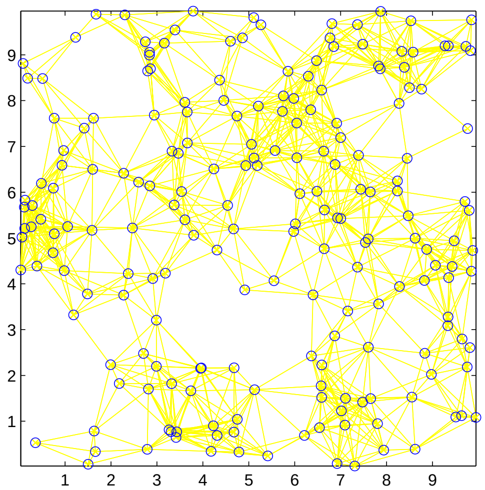
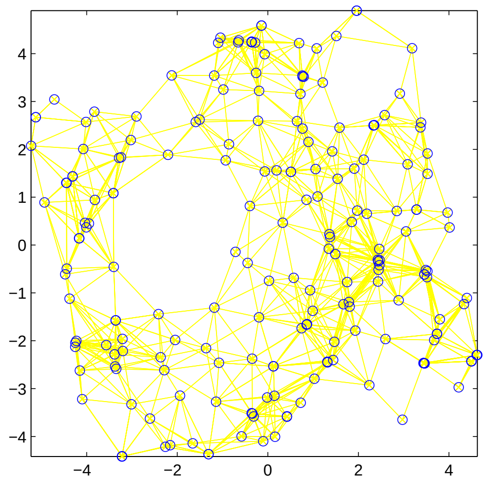


Figure 1: 200 nodes randomly placed in a $10r \times 10r$ square. The radio range is $1.5r$, yielding an average connectivity of 12.1.

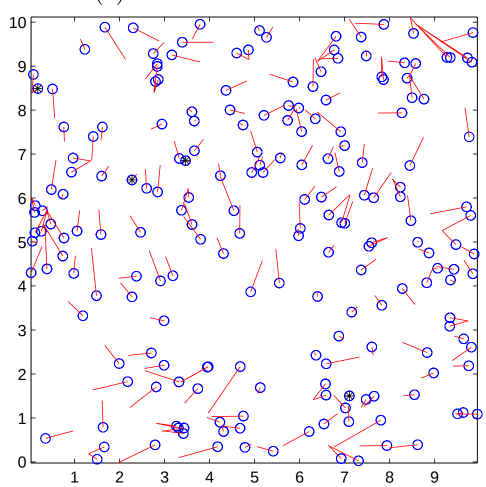
network in Figure 1. Figure 2(b) shows the final solution of MDS-MAP where the MDS result is transformed based on 4 random anchor nodes, denoted by the stars (*) in the network. The circles represent the true locations of the nodes, and the solid lines represent the errors of the estimated position from the true position. The longer the line, the larger the error. The average estimation error in this example is about $0.46r$.

When the distances between neighbors are known, even with limited accuracy, the result of MDS-MAP can be significantly improved. Figure 3 shows the result of MDS-MAP knowing the distance of neighbors with 5% range error. Figure 3(a) shows the map constructed by MDS. Again, it has different scale and orientation than the ones in Figures 1 and 2(a). Figure 3(b) shows the final estimation of MDS-MAP based on the same 4 anchor nodes. It has an average estimation error of $0.24r$, much better than the previous result using connectivity only.

Figure 4 shows the average performance of MDS-MAP as a function of connectivity and number of anchors. Figure 4(a) shows results of MDS-MAP based on proximity information only. The radio ranges (R) are $1r$, $1.25r$, $1.5r$, $1.75r$, and $2r$, respectively, which lead to average connectivity levels 5.9, 8.9, 12.2, 16.2, and 20.7, respectively. 3, 4, 6, and 10 random anchors are used. Position estimates by MDS-MAP have an average error under $100\%R$ in scenarios with just 4 anchor nodes and an average connectivity level of 8.9 or greater. When the connectivity level is 12.2 or greater, the errors with just 3 anchors is quite good, close to or better than 50%. On the other hand, when the connectivity is low, e.g., 5.9, the errors can be large. These results are much better than the ones obtained by the convex optimization approach in [4] when the number of anchor nodes is low. For example, with 4 to 10 anchors in a 200-node random network, the convex optimization approach has an average estimation error of more than twice the radio range when the radio range is $1.25R$ and above. In contrast, our method has an error from about 80% down to 40% as the radio range goes from $1.25R$ to $2R$.



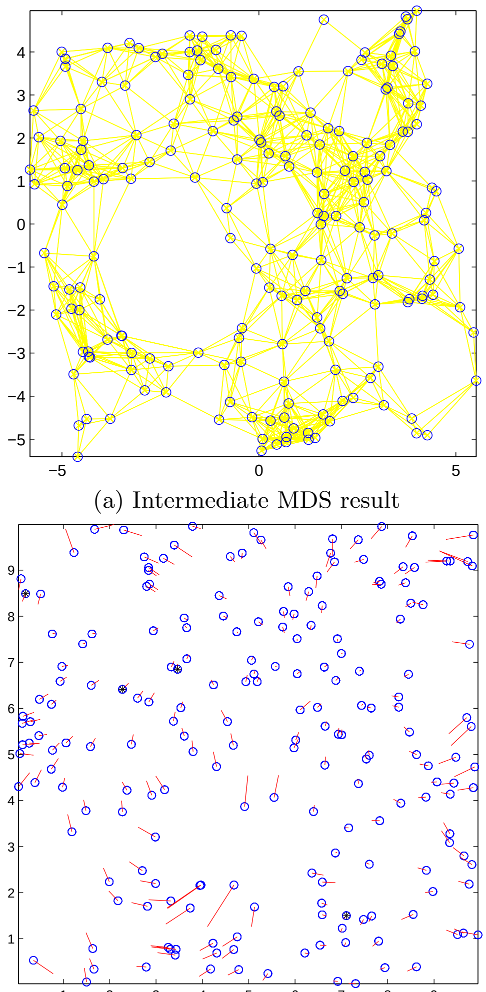
(a) Intermediate MDS result



(b) Final position estimation by MDS-MAP

Figure 2: Location estimation for the random network using connectivity information only.

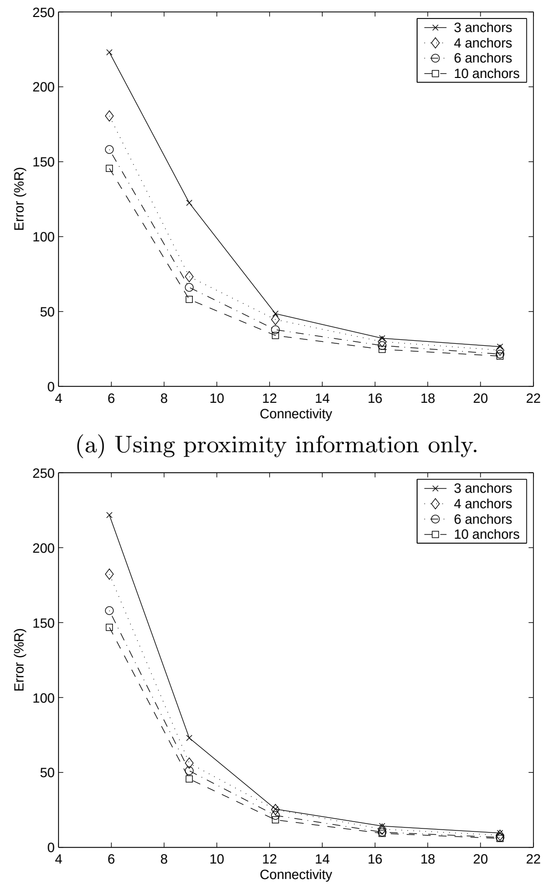
Figure 4(b) shows results of MDS-MAP using distance measurement between neighbors with 5% range error. Knowing the distances between neighbors leads to much better solutions when the connectivity in the network is high. When the connectivity level is 12.2 or greater, the errors are about half of those by MDS-MAP using proximity information only. On the other hand, when the connectivity is low, e.g., 5.9, knowing the local distance does not help much and the errors are still large. These results improve on the results of Hop-TERRAIN [11], especially when the number of anchors is small. For example, with 2% anchors and a connectivity level about 12, MDS-MAP has an average error of about $50\%R$, whereas Hop-TERRAIN has an error of about $90\%R$. After the refinement phase in [11], the error is reduced to about $18\%R$. MDS-MAP should be compared to Hop-TERRAIN since it can be followed by a refinement phase like the ones in [11] and [13]. By starting from the better initial estimates generated by MDS-MAP, a refinement phase should find even better results. Our preliminary experiments along this avenue have been encouraging.



(a) Intermediate MDS result

(b) Final position estimation by MDS-MAP

Figure 3: Estimation for the random network using distances between neighbors with 5% range error.



(a) Using proximity information only.

(b) Using distances between neighbors (5% range error).

Figure 4: Average position error of MDS-MAP on random networks of 200 nodes.

Figure 5 shows the number of nodes participating in the location estimation in MDS-MAP. Recall that the largest connected subnetwork is extracted for processing. When the connectivity level is low, such as 5.9, about 7% of the nodes are not connected to the main subnetwork, and hence their positions are not estimated. Among the nodes that are part of the main subnetwork, many of them have only 1 or 2 connections to their neighbors. The lack of sufficient information to determine the position of a node causes large errors in MDS-MAP solutions. When the average connectivity exceeds 12.2, the network tends to be fully connected and all nodes can be localized.

Sensitivity to range errors has been a major concern for localization algorithms. Figure 6 shows the effects of increasing range errors on the estimation errors. The radio ranges are $1.5r$ and $2r$, which lead to connectivity levels 12.2 and 20.7. 4 random anchors are used. The range errors vary from 0 to 50%. Estimation error increases steadily as the range error increases. The errors with a larger connectivity (20.7) are more than $10\%R$ lower than those with a smaller connectivity (12.2) in most cases. The estimation error goes up faster after the range error is over 30%.

4.2 Grid Placement

In this set of experiments, we assume that the sensor nodes are deployed according to some regular structures such as a square or a hexagonal grid. Actually, nodes are placed in the neighborhood of the vertices due to random placement error. 100 nodes are placed on a $10r \times 10r$ grid, with a unit edge distance r. Our results show that MDS-MAP obtains much better results on the grid layout than on the random layout for the same connectivity level.

Figure 7 shows an example of the regular grid with 10% placement error. The radio range is $R = 1.4r$, which leads to connectivity 5.06. In the graph, points again represent sensor nodes and edges represent the connections between neighbors. Figure 8 shows the result of MDS-MAP based on the connectivity of the network. Figure 8(a) shows the intermediate result of classical MDS on the connectivity matrix. It can be seen that the center of the map is the origin (0,0), and it has a different orientation than the original network in Figure 7. Figure 8(b) shows the final solution of MDS-MAP where the MDS result is transformed based on 4 random anchor nodes (* nodes). The circles represent the

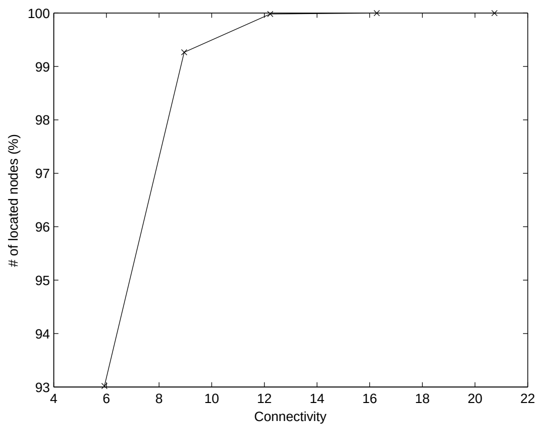


Figure 5: Fraction of nodes participating in the localization process. This fraction is independent of the number or position of anchor nodes.

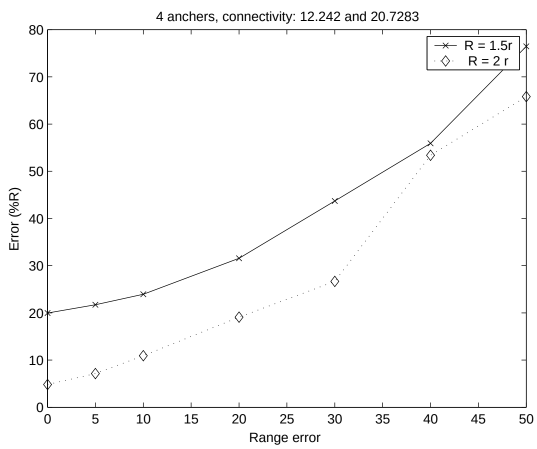


Figure 6: The effect of range error on the estimation error.

true locations of sensor nodes, and the solid lines represent the errors between the estimated and true positions. The longer the line, the larger the error.

When the distance measurement between neighbors is available, the result of MDS-MAP is significantly improved. Figure 9 shows the result of MDS-MAP knowing the distance of neighbors with 5% range error. Figure 9(a) shows the map constructed by MDS. Again, it has different scale and orientation than the ones in Figures 7 and 8(a). Figure 9(b) shows the final estimation of MDS-MAP based on the same 4 anchor nodes. The estimation errors are now very small.

Figure 10 shows the average performance of MDS-MAP using proximity information as a function of connectivity and placement errors, given 3 or 5 random anchors respectively. The radio ranges (R) are $1.5r$, $2r$, $2.5r$, and $3r$, respectively. For different placement errors, the same radio range leads to different connectivity levels. With 3 anchors, position estimates by MDS-MAP have an average error under $50\%R$ for the placement errors up to 50% in scenarios with an average connectivity level of 8 or greater. With 5 anchors, position

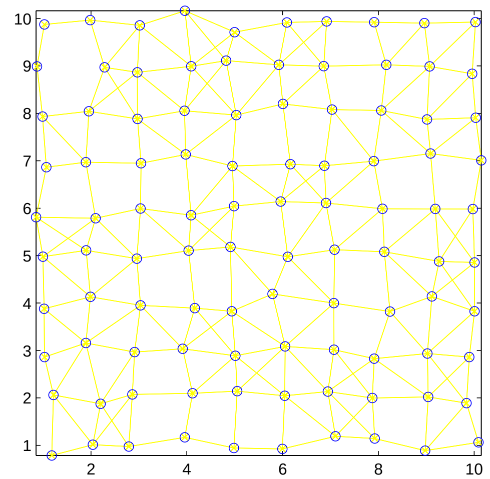


Figure 7: 100 nodes placed on a $10r \times 10r$ grid with 10% random placement error.

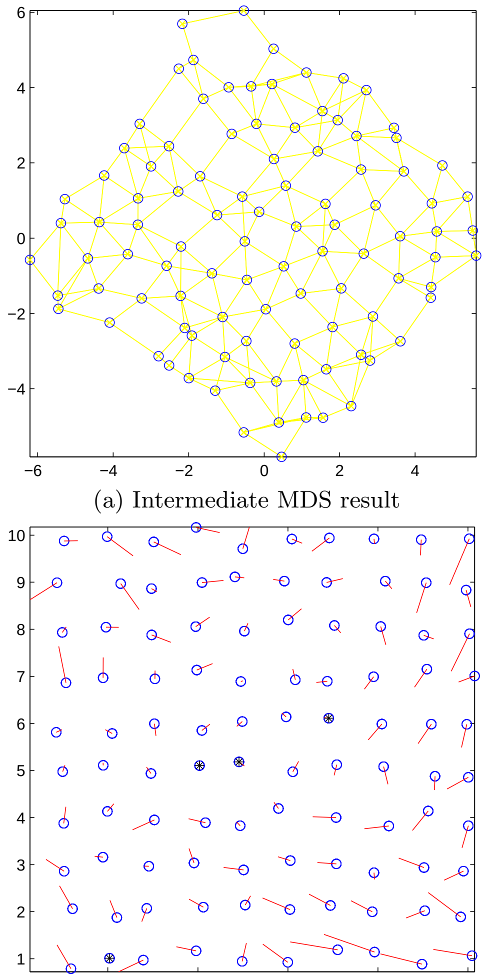
errors by MDS-MAP is reduced to 35% and below for the same cases.

Figure 11 shows results of MDS-MAP using distance measurement between neighbors with range errors from 5% to 50%. Knowing the distances between neighbors leads to much better solutions. When the range error is below 20%, the estimation errors are below 25%. Having more anchors (5 vs. 3) improves performance, especially for the case with large range errors.

In addition to the experiments with $10r \times 10r$ grids, we also tried similar experiments with grids of other sizes, such as $8r \times 8r$ grids (64 nodes) and $20r \times 20r$ grids (400 nodes). The average position errors obtained by MDS-MAP on the different size grids are very similar given the same number of anchor nodes.

Similar results are obtained for the hexagonal grid layout. MDS-MAP achieves slightly better performance than for the square grid layout due to the increased regularity of the distances between neighboring nodes. Figure 12 shows an example of the regular grid with 10% placement error. The radio range is $R = 1.4r$, which leads to connectivity 5.32. Figure 13 shows the result of MDS-MAP based on the connectivity of the network. Figure 13(a) shows the intermediate result of classical MDS on the connectivity matrix. Figure 13(b) shows the final solution of MDS-MAP where the MDS result is transformed based on 4 random anchor nodes (* nodes). When using the distance measurement between neighbors, the result of MDS-MAP is again much better. Figure 14 shows the result of MDS-MAP knowing the distance of neighbors with 5% range error. Figure 14 shows the map constructed by MDS and the final estimation of MDS-MAP based on the same 4 anchor nodes. The estimation errors are now very small.

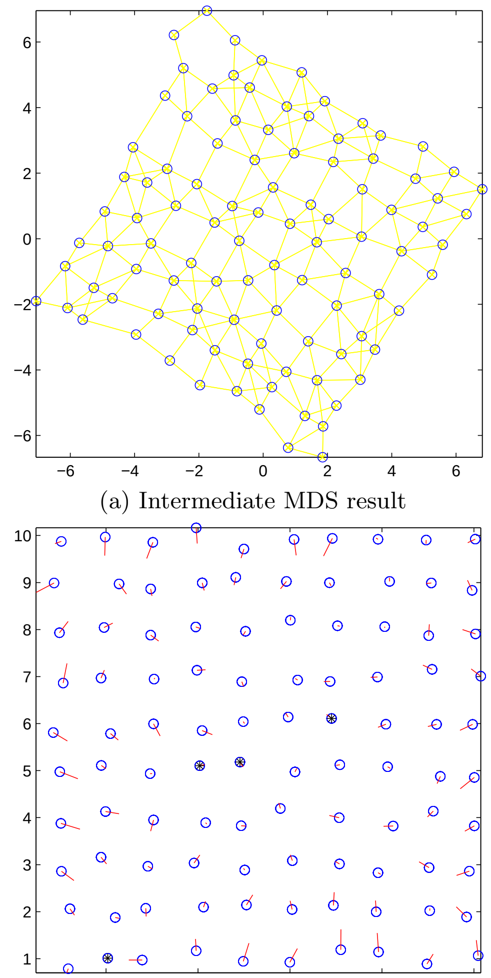
Figure 15 shows the average performance of MDS-MAP using proximity information as a function of connectivity and placement errors, given 3 or 5 random anchors respectively. The radio ranges (R) are $1.5r$, $2.5r$, and $3.5r$, respectively. For different placement errors, the same radio range leads to different connectivity levels. With 3 anchors, position estimates by MDS-MAP have an average error under $30\%R$



(a) Intermediate MDS result

(b) Final position estimation by MDS-MAP

Figure 8: Estimation on the grid network using connectivity information only.



(a) Intermediate MDS result

(b) Final position estimation by MDS-MAP

Figure 9: Estimation on the grid network using neighbor distances with 5% error.

for the placement errors up to 50% when the connectivity level is 14 or greater. With 5 anchors, position errors by MDS-MAP are reduced slightly.

Figure 16 shows results of MDS-MAP using distance measurement between neighbors with range errors from 5% to 50%. Again, knowing the distances between neighbors leads to much better solutions. When the range error is below 20%, the estimation errors for low connectivity level (5.2) are below 30%. With 3 anchors, position estimates by MDS-MAP have an average error under 15%R with an average connectivity level of 14 or greater. Having more anchors (5 vs. 3) improves its performance, especially for the case with large range errors.

In summary, MDS-MAP performs well when the level of connectivity is over 9 for the grid placements and over 12 for the random placement. The number of anchor nodes needed by MDS-MAP is very small. When there is sufficient connectivity, 3 anchors for the grid placements and 4 anchors for the random placement are usually enough for MDS-MAP to find solutions with position errors less than half of the range of radio. MDS-MAP works well when the placement errors are less than a quarter of the radio range and when the range errors are less than 20% of the radio range. For both the random and grid placements, the position error of MDS-MAP increases proportional to the range error.

5. POSSIBLE EXTENSIONS

A drawback of the current implementation of MDS-MAP is that it requires global information of the network and centralized computation. One way to address this issue is to divide the network into sub-networks and apply MDS-MAP to each sub-network independently. Since our method does not require anchor nodes in building a relative map of a sub-network, the method can be applied to many sub-networks in parallel. Then adjacent local maps can be combined by aligning with each other. In another words, the complete map of the sensor network consists of many smaller patches. When three or more anchors are present in either a sub-network or the whole network, an absolute map can be computed accordingly. Although this patching ap-

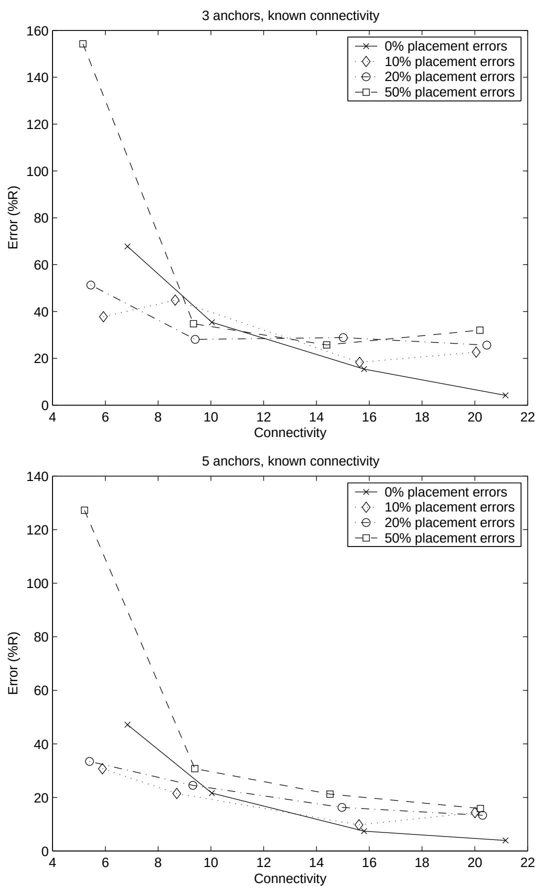


Figure 10: Average position error of MDS-MAP on grid networks of 100 nodes when using only connectivity information.

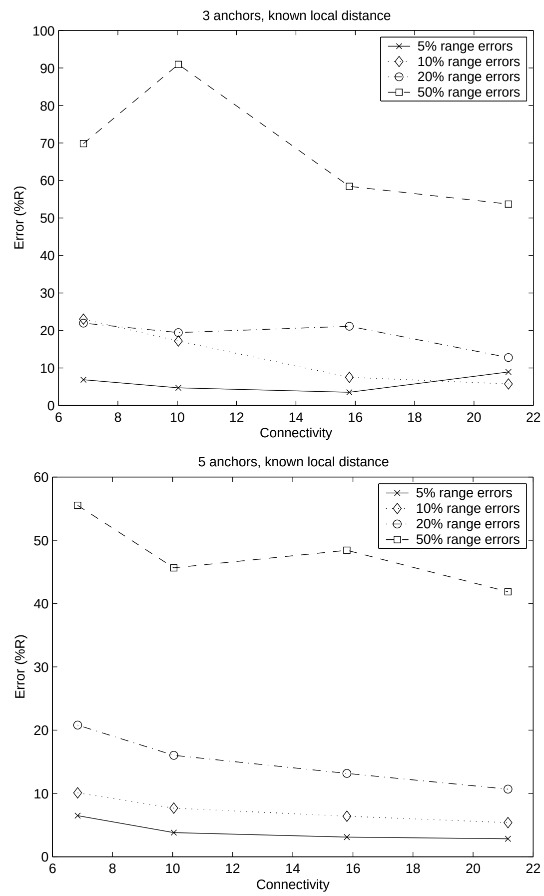


Figure 11: Average error of MDS-MAP on grid networks of 100 nodes when using distance information.

proach requires significant computation within each patch, one has considerable flexibility in choosing which nodes perform the computation. Preliminary experiments have been very encouraging and detailed results will be reported in future work.

MDS-MAP can also be extended by applying more advanced MDS techniques. Instead of classical metric MDS, other MDS techniques such as ordinal MDS and MDS with missing data can be applied. This may be useful to handle non-uniform radio propagation and non-uniform ranging errors. We have done some limited experiments with ordinal MDS. Our results show that ordinal MDS is better than classical MDS when the connectivity level of the sensor network is low, and is comparable with classical MDS when the connectivity level is high.

Another drawback of MDS-MAP is that when the number of anchor nodes is large, the performance of MDS-MAP is not as good as previous methods such as the constraint-based approach [4], "DV-hop" [9], or Hop-TERRAIN [11]. The reason is that the second step of MDS-MAP, the application of classical MDS, is done without using the positioning information of anchor nodes. The information is only used in step 3, when the overall structure and distance ratios between nodes have already been determined. The approach of building a relative map irrespective of the coordinations of anchor nodes is double-edged. It works nicely when there are few or no anchor nodes, but not as well when there are more anchor nodes. One solution may be to use a more advanced MDS technique called the anchor point method [2], where coordinates of anchor nodes are explicitly used in determining the scaling.

As we mentioned above, combining MDS-MAP with other methods is another promising avenue. For example, MDS-MAP can be used to get a good initial estimates of node positions, which is followed by a refinement phase like the ones in [11] or [13]. Due to the good performance of MDS-MAP comparing to competing methods on the cases of low anchor node densities, one can expect this two-phase approach to generate good results.

6. CONCLUSIONS

In this paper, we presented a new localization method, MDS-MAP, that works well with mere connectivity information. However, it can also incorporate distance information between neighboring nodes when it is available. The strength of MDS-MAP is that it can be used when there are few or no anchor nodes. Previous methods often require well-placed anchors to work well. For example, the constraint-based approach in [4] works well only when the

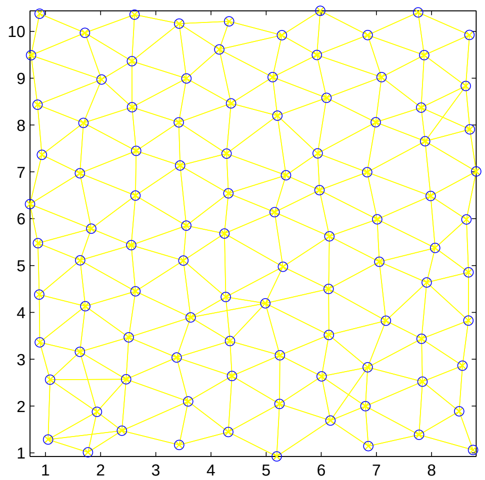


Figure 12: 100 nodes randomly placed on a 10×10 hexagonal grid with 10% placement error.

anchors are placed at the outside corners and edges and the constraints are tight. It works poorly when the anchors are inside the network, close to the center. The collaborative multilateration approach in [13] also requires anchors throughout the network, as well as a relatively large number of anchors, to work well. Our method does not have this limitation. It builds a relative map of the nodes even without anchor nodes. With three or more anchor nodes, the relative map can be transformed and absolute coordinates of the sensor nodes are computed. Extensive simulations using various network arrangements and different levels of ranging error show that the method is effective, and particularly so for situations with few anchor nodes and relatively uniform node distributions.

7. ACKNOWLEDGEMENTS

This work was supported in part by DARPA under contract F33615-01-C-1904.

8. REFERENCES

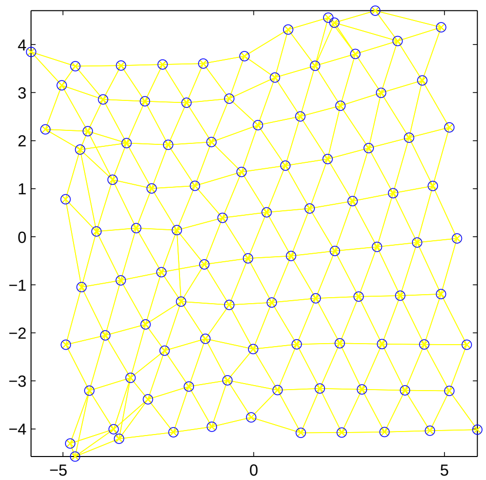
[1] I. Borg and P. Groenen. *Modern Multidimensional Scaling, Theory and Applications*. Springer-Verlag, New York, 1997.

[2] A. Buja, D. F. Swayne, M. Littman, N. Dean, and H. Hofmann. XGvis: Interactive data visualization with multidimensional scaling. *Journal of Computational and Graphical Statistics*, page (to appear), 2001.

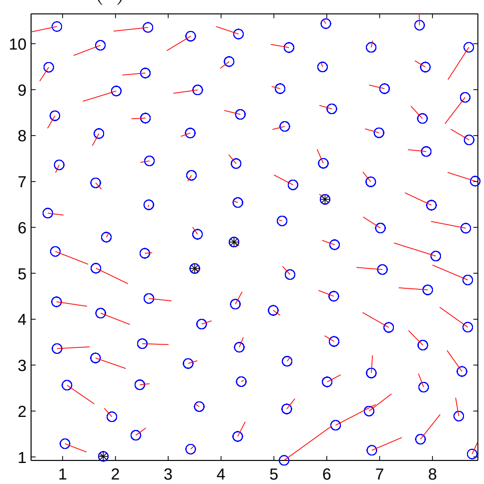
[3] N. Bulusu, J. Heidemann, and D. Estrin. GPS-less low-cost outdoor localization for very small devices. *IEEE Personal Communications*, 7(5):28–34, Oct. 2000.

[4] L. Doherty, L. E. Ghaoui, and K. Pister. Convex position estimation in wireless sensor networks. In *Proc. Infocom 2001*, Anchorage, AK, April 2001.

[5] D. Ganesan, B. Krishnamachari, A. Woo, D. Culler, D. Estrin, and S. Wicker. An empirical study of epidemic algorithms in large scale multihop wireless networks. Technical report UCLA/CSD-TR-02-0013, UCLA Computer Science Department, 2002.



(a) Intermediate MDS result



(b) Final position estimation by MDS-MAP

Figure 13: Estimation on the hexagonal grid network using connectivity information only.

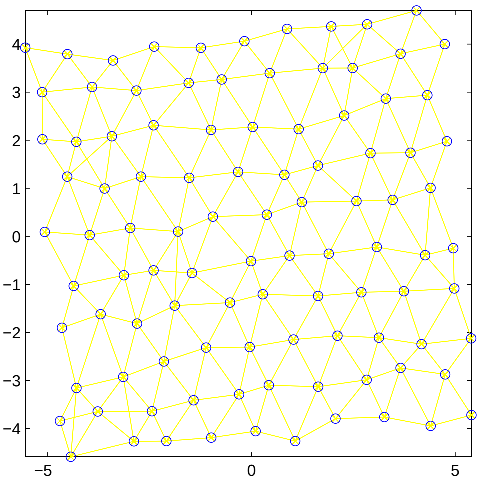
[6] J. Hightower and G. Boriello. Location systems for ubiquitous computing. *IEEE Computer*, 34(8):57–66, Aug. 2001.

[7] A. Howard, M. J. Mataric, and G. S. Sukhatme. Relaxation on a mesh: a formalism for generalized localization. In *Proc. IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems (IROS01)*, pages 1055–1060, 2001.

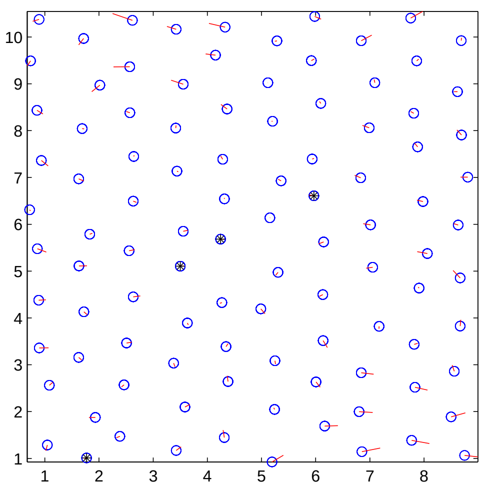
[8] A. Nasipuri and K. Li. A directionality based location discovery scheme for wireless sensor networks. In *1st ACM Int'l Workshop on Wireless Sensor Networks and Applications (WSNA'02)*, pages 105–111, Atlanta, GA, Sept. 2002.

[9] D. Niculescu and B. Nath. Ad-hoc positioning system. In *IEEE GlobeCom*, Nov. 2001.

[10] S. I. Roumeliotis and G. A. Bekey. Synergetic localization for groups of mobile robots. In *Proc. 39th IEEE Conf. on Decision and Control*, Sydney,



(a) Intermediate MDS result



(b) Final position estimation by MDS-MAP

Figure 14: Estimation on the hexagonal grid network using distances with 5% range error.

Australia, Dec. 2000.

[11] C. Savarese, J. Rabaey, and K. Langendoen. Robust positioning algorithm for distributed ad-hoc wireless sensor networks. In *USENIX Technical Annual Conf.*, Monterey, CA, June 2002.

[12] A. Savvides, C. C. Han, and M. Srivastava. Dynamic fine-grained localization in ad hoc networks of sensors. In *ACM/IEEE Int'l Conf. on Mobile Computing and Networking (MOBICON)*, July 2001.

[13] A. Savvides, H. Park, and M. Srivastava. The bits and flops of the n-hop multilateration primitive for node localization problems. In *1st ACM Int'l Workshop on Wireless Sensor Networks and Applications (WSNA'02)*, pages 112–121, Atlanta, GA, Sept. 2002.

[14] R. N. Shepard. Analysis of proximities: Multidimensional scaling with an unknown distance function I & II. *Psychometrika*, 27:125–140, 219–246, 1962.

[15] W. S. Torgeson. Multidimensional scaling of similarity. *Psychometrika*, 30:379–393, 1965.

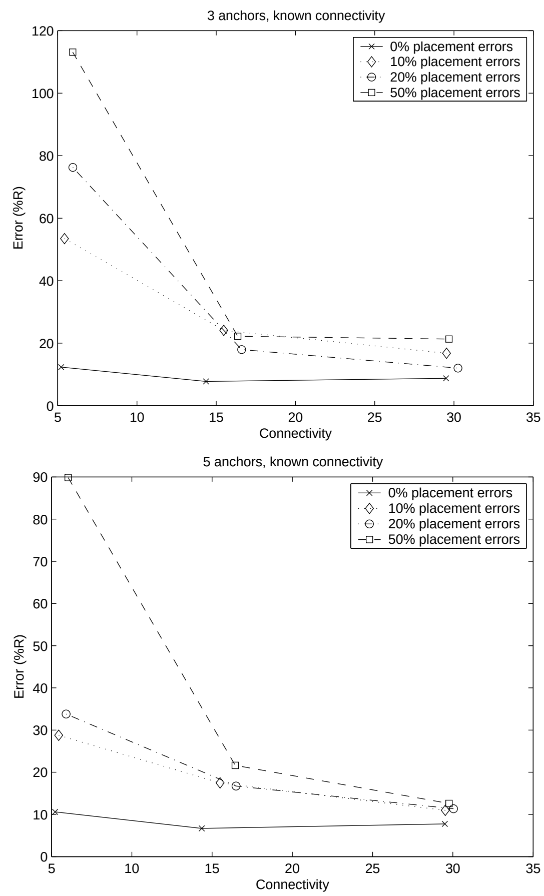


Figure 15: Average error on hexagonal grid networks when using only connectivity information.

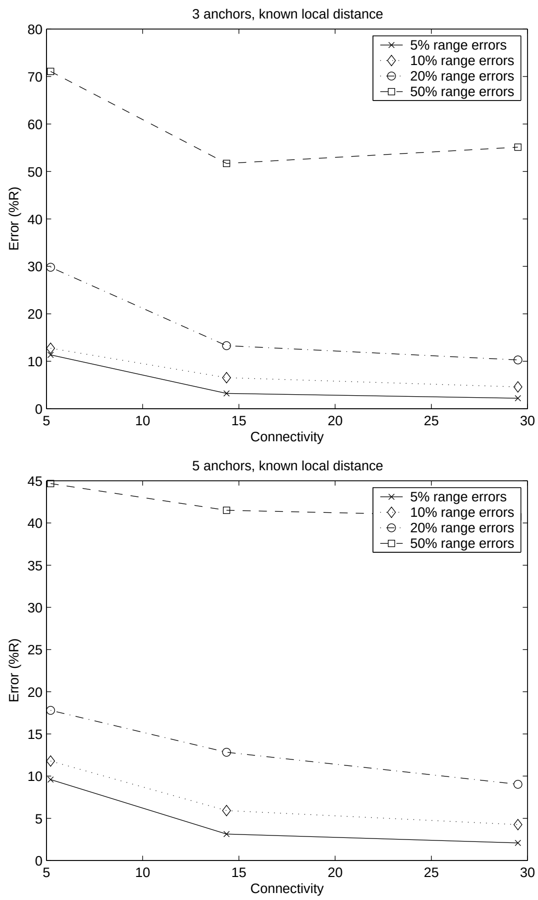


Figure 16: Average error on hexagonal grid networks when using distances between neighboring nodes.