

# Chapter VIII

## MDS-Based Localization

**Ahmed A. Ahmed**

*Texas State University–San Marcos, USA*

**Xiaoli Li**

*University of Missouri–Columbia, USA*

**Yi Shang**

*University of Missouri–Columbia, USA*

**Hongchi Shi**

*Texas State University–San Marcos, USA*

### ABSTRACT

The authors present several network node localization methods that are based on multidimensional scaling (MDS) techniques. Four algorithms are introduced: MDS-MAP(C), MDS-MAP(P), MDS-Hybrid, and RangeQ-MDS. MDS-MAP(C) is a centralized algorithm that simply applies MDS to estimate node positions. In MDS-MAP(P), a local map is built at each node of the immediate vicinity, then these maps are merged together to form a global map. MDS-Hybrid uses MDS-MAP(C) to relatively localize  $N_r$  reference nodes. Then, an absolute localization method uses these  $N_r$  nodes as anchors to localize the rest of the network. Finally, RangeQ-MDS assumes the absence of an RSSI-distance mapping function. It uses a quantized RSSI-based distance estimation technique (called RangeQ) to achieve more precise hop distances than other range-free approaches do. While MDS-MAP(C), MDS-MAP(P), and MDS-Hybrid can be range-aware or range-free, RangeQ-MDS is partially range-aware.

### INTRODUCTION

The multidimensional scaling (MDS) technique has its origins in psychometrics and psychophysics. It is often used as part of exploratory data analysis or information visualization. It is related to principal

component analysis, factor analysis, and cluster analysis. MDS has been applied in many fields, such as machine learning (Tenenbaum *et al.*, 2000) and computational chemistry (Glunt *et al.*, 1993). When used for localization, MDS takes full advantage of connectivity or distance information between nodes that have yet to be localized, unlike previous approaches.

In this chapter, we present four MDS-based localization methods for wireless sensor networks. The first method, called MDS-MAP(C) (Shang *et al.*, 2004), is a simple centralized approach that builds a global map using classical MDS. Like many existing methods, MDS-MAP(C) works well on networks with relatively uniform node density, but less well on more irregular networks, where the shortest-path distance between two nodes does not correspond well to their Euclidean distance.

To tackle this difficult problem, we present MDS-MAP(P) (Shang *et al.*, 2004). It is more complicated than MDS-MAP(C) because it builds for each node a local map of the small sub-network in the node's vicinity and then merges (patches) the local maps together to form a global map. This method avoids using shortest-path distances between far away nodes, and thus the smaller local maps constructed using local information are more accurate. Another advantage of the method is that it can be easily performed in a distributed fashion, which makes it appropriate for large-scale networks. A refinement step that uses minimum-squares minimization can be added to either MDS-MAP(C) or MDS-MAP(P) to improve the solution computed by MDS. We call the resulting methods MDS-MAP(C,R) and MDS-MAP(P,R), respectively.

In relative localization, nodes use the distance measurements to estimate their positions relative to some coordinate system. In absolute localization, a few nodes, called *anchors* (Savarese *et al.*, 2001), need to know their absolute positions, and all the other nodes are absolutely localized in the anchors' coordinate system. Due to the relatively high computational cost of MDS-MAP(C) and MDS-MAP(P), MDS-Hybrid is a relative localization algorithm that looks for a higher degree of granularity in terms of the performance-cost tradeoff than other localization algorithms. MDS-Hybrid, which is based on SHARP proposed by Ahmed *et al.* (2005), selects a number  $N_r$  of *reference nodes*. Then, MDS-MAP(C) relatively localizes these reference nodes using the distance information between them. Finally, an absolute localization method ( $M$ ) localizes the rest of the network relative to the coordinate system of the reference nodes. Choosing  $N_r$  and  $M$  depends on both the application and the interest, either good performance, low cost, or somewhere in between.

We consider the node localization problem under two different scenarios. In the first, only connectivity (or proximity) information is available. Each node only knows what nodes are nearby, but not how far away these neighbors are or in what direction they lie. We call this scenario *range-free* localization. In the second scenario, the proximity information is enhanced by knowing the distances, perhaps with limited accuracy, between neighboring nodes. This is called *range-aware* localization. While MDS-MAP(C), MDS-MAP(P), and MDS-Hybrid can be range-free or range-aware, the RangeQ-MDS algorithm uses a sorted RSSI Partial Range Information (*PRI*)-based quantization scheme, called RangeQ (Li *et al.*, 2006), to improve the range estimation accuracy when distance information is not available.

Through simulation studies on regular as well as irregular networks, we show the improvement in localization performance the four methods presented can achieve.

## PROBLEM FORMULATION

The network is represented as a connected undirected graph  $G = (V, E)$ , where  $V$  is the set of sensor nodes, of which there exist  $A \subset V$  special nodes (called anchors) with known positions, and  $E$  is the set of

edges connecting neighboring nodes. For the range-free case, the edges in the graph correspond to the connectivity information. For the range-aware case, the edges are associated with values corresponding to the estimated distances.

Given a network graph of  $n$  nodes and estimated distances  $\mathbf{P}$  between some pairs of nodes (let  $p_{ij}$  represent the estimated distance between nodes  $i$  and  $j$ ), the localization problem is to find the coordinates  $\mathbf{X} = (\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_n)$  of the nodes such that the Euclidean distances between the estimated positions of the nodes equal  $\mathbf{P}$ , i.e.,  $d_{ij} = p_{ij}$  for available  $p_{ij}$ , where  $d_{ij} = \|\mathbf{X}_i - \mathbf{X}_j\|_2$ . When the estimates  $p_{ij}$  are just the connectivity or inaccurate local distance measurements, usually there is no exact solution to the over-determined system of equations. Thus the localization problem is often formulated as an optimization problem that minimizes the sum of squared errors. This optimization problem is generally non-convex with many local minima. Traditional local optimization techniques, such as the Levenberg-Marquardt method, require good initial points in order to produce good solutions. Global search methods such as simulated annealing or genetic algorithms are generally too slow.

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 some direction-based routing algorithms (Royer and Toh, 1999; Yu *et al.*, 2001). 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 (Chu *et al.*, 2002; Intanagonwiwat *et al.*, 2000; Johnson and Maltz, 1996; Karp and Kung, 2000).

Before we describe the details of the methods presented, we first introduce MDS, and then describe the simulation setup for our experiments.

## MULTIDIMENSIONAL SCALING (MDS)

Multidimensional scaling (MDS) is a method for visualizing dissimilarity data. For example, instead of knowing the latitude and longitude of a set of cities, we may only know their inter-city distances. The typical goal of MDS is to create a configuration of points in one, two, or three dimensions, whose inter-point distances are “close” to the original dissimilarities. The different variants of MDS use different criteria to define “close”. These points represent the set of objects, and so a plot of the points can be used as a visual representation of their dissimilarities. Recently, MDS has been successfully applied to the problem of node localization in wireless sensor networks.

### Basics of MDS Models

MDS models are defined by specifying how the given similarity data  $p_{ij}$  between two objects  $i$  and  $j$  are mapped into distances  $d_{ij}$  of an  $m$ -dimensional MDS configuration  $\mathbf{X}$  consisting of all objects. The mapping is specified by a representation function,  $f: p_{ij} \rightarrow d_{ij}(\mathbf{X})$ , which specifies how the similarity

data should be related to the distances. In practice, one usually does not attempt to strictly satisfy  $f$ . Rather, what is sought is a configuration whose distances satisfy  $f$  as closely as possible. The condition “as closely as” is quantified by a badness-of-fit measure or loss function. The loss function is a mathematical expression that aggregates the representation errors,  $e_{ij} = f(p_{ij}) - d_{ij}(\mathbf{X})$ , over all pairs  $(i, j)$ . A normalized sum-of-squares of these errors define stress, the most common loss function in MDS.

Assume that measures of similarity, for which we use the general term proximity,  $p_{ij}$ , are given for each pair  $(i, j)$  of  $n$  objects. MDS attempts to represent proximities by distances among the points (representing the objects) of an  $m$ -dimensional configuration  $\mathbf{X}$ , the MDS space. Given a Cartesian space, one can compute the distance between any two points  $i$  and  $j$ . The Euclidean distance between points  $i$  and  $j$  in a two-dimensional configuration  $\mathbf{X}$  is computed by the following formula:

$$d_{ij}(\mathbf{X}) = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2},$$

which can be written as

$$d_{ij}(\mathbf{X}) = \left[ \sum_{a=1}^2 (x_{ia} - x_{ja})^2 \right]^{1/2}.$$

MDS maps proximities  $p_{ij}$  into the corresponding distances  $d_{ij}(\mathbf{X})$  of an MDS space  $\mathbf{X}$ . That is,  $f: p_{ij} \rightarrow d_{ij}(\mathbf{X})$ . The distances  $d_{ij}(\mathbf{X})$  are unknowns, and MDS finds a configuration  $\mathbf{X}$  of a predetermined dimensionality  $m$  on which the distances are computed. The function  $f$ , on the other hand, can be either completely specified or restricted to come from a particular class of functions.

Empirical proximities always contain noise due to measurement imprecision. Hence, one should not insist, in practice, that  $f(p_{ij}) = d_{ij}(\mathbf{X})$ , but rather that  $f(p_{ij}) \approx d_{ij}(\mathbf{X})$ , where “ $\approx$ ” can be read as “as equal as possible”. Computerized procedures for finding an MDS representation usually start with some initial configuration and improve it by moving around its points in small steps (iteratively) to approximate the ideal model relation  $f(p_{ij}) = d_{ij}(\mathbf{X})$  more and more closely. A squared error of representation is defined by

$$e_{ij}^2 = [f(p_{ij}) - d_{ij}(\mathbf{X})]^2.$$

Summing  $e_{ij}^2$  over all pairs  $(i, j)$  yields a badness-of-fit measure for the entire MDS representation, *raw stress*  $\sigma_r$ ,

$$\sigma_r(\mathbf{X}) = \sum_{i,j} [f(p_{ij}) - d_{ij}(\mathbf{X})]^2.$$

To avoid scale dependency,  $\sigma_r$  can be normalized as follows,

$$\sigma_1^2(\mathbf{X}) = \frac{\sigma_r(\mathbf{X})}{\sum d_{ij}^2(\mathbf{X})} = \frac{\sum [f(p_{ij}) - d_{ij}(\mathbf{X})]^2}{\sum d_{ij}^2(\mathbf{X})}.$$

Taking the square root yields a value known as *Stress-1*. Thus

$$Stress-1 = \sigma_1(\mathbf{X}) = \sqrt{\frac{\sum [f(p_{ij}) - d_{ij}(\mathbf{X})]^2}{\sum d_{ij}^2(\mathbf{X})}}.$$

In weighted MDS, positive weights  $w_{ij}$  are added into the stress function as follows:

$$\sigma_r(\mathbf{X}) = \sum_{i>j} w_{ij} [f(p_{ij}) - d_{ij}(\mathbf{X})]^2.$$

When working with missing data,  $w_{ij}$  is set to 1 if  $p_{ij}$  is known and 0 if  $p_{ij}$  is missing.

## Classical MDS

The basic idea of classical MDS (Gower, 1966 and Torgerson, 1952) is to assume that the dissimilarities are distances and then find coordinates that explain them. Moreover, a linear transformation model is assumed, i.e.,  $d_{ij} = a + bp_{ij}$ . The distances  $\mathbf{D}$  are determined so that they are as close to the proximities  $\mathbf{P}$  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  $\mathbf{I}(\mathbf{P}) = \mathbf{D} + \mathbf{E}$ , where  $\mathbf{I}(\mathbf{P})$  is a linear transformation of the proximities, and  $\mathbf{E}$  is a matrix of errors (residuals). Since  $\mathbf{D}$  is a function of the coordinates  $\mathbf{X}$ , the goal of classical MDS is to calculate  $\mathbf{X}$  such that the sum of squares of  $\mathbf{E}$  is minimized. In classical MDS, the coordinates  $\mathbf{X}$  can be computed from  $\mathbf{P}$  through singular value decomposition (SVD) on the double centered squared  $\mathbf{P}$ . Double centering a matrix is subtracting the row and column means of the matrix from its elements, adding the grand mean and multiplying by  $-1/2$ .

Let  $\mathbf{X}_{n \times m}$  be the matrix of coordinates of the points. Each row  $i$  of  $\mathbf{X}$  gives the coordinates of point  $i$  on  $m$  dimensions, i.e.,  $x_{i1}, x_{i2}, \dots, x_{im}$ . The squared Euclidean distance is defined by

$$d_{ij}^2(\mathbf{X}) = d_{ij}^2 = \sum_{a=1}^m (x_{ia} - x_{ja})^2 = \sum_{a=1}^m (x_{ia}^2 + x_{ja}^2 - 2x_{ia}x_{ja}).$$

Let  $\mathbf{D}^{(2)}(\mathbf{X})$  denote the matrix of squared distances. For example, when  $\mathbf{X}$  contains the coordinates of three points in two dimensions,  $\mathbf{D}^{(2)}(\mathbf{X})$  can be represented as

$$\begin{aligned} \mathbf{D}^{(2)}(\mathbf{X}) &= \begin{bmatrix} 0 & d_{12}^2 & d_{13}^2 \\ d_{12}^2 & 0 & d_{23}^2 \\ d_{13}^2 & d_{23}^2 & 0 \end{bmatrix} \\ &= \sum_{a=1}^2 \begin{bmatrix} x_{1a}^2 & x_{1a}^2 & x_{1a}^2 \\ x_{2a}^2 & x_{2a}^2 & x_{2a}^2 \\ x_{3a}^2 & x_{3a}^2 & x_{3a}^2 \end{bmatrix} + \sum_{a=1}^2 \begin{bmatrix} x_{1a}^2 & x_{2a}^2 & x_{3a}^2 \\ x_{1a}^2 & x_{2a}^2 & x_{3a}^2 \\ x_{1a}^2 & x_{2a}^2 & x_{3a}^2 \end{bmatrix} - 2 \sum_{a=1}^2 \begin{bmatrix} x_{1a}x_{1a} & x_{1a}x_{2a} & x_{1a}x_{3a} \\ x_{2a}x_{1a} & x_{2a}x_{2a} & x_{2a}x_{3a} \\ x_{3a}x_{1a} & x_{3a}x_{2a} & x_{3a}x_{3a} \end{bmatrix} \\ &= \mathbf{c}\mathbf{1}' + \mathbf{1}\mathbf{c}' - 2 \sum_{a=1}^2 \mathbf{x}_a \mathbf{x}_a' \end{aligned}$$

In general, we have

$$\mathbf{D}^{(2)}(\mathbf{X}) = \mathbf{c}\mathbf{1}' + \mathbf{1}\mathbf{c}' - 2 \sum_{a=1}^2 \mathbf{x}_a \mathbf{x}_a' = \mathbf{c}\mathbf{1}' + \mathbf{1}\mathbf{c}' - 2\mathbf{X}\mathbf{X}' \quad (1)$$

where  $\mathbf{x}_a$  is column  $a$  of matrix  $\mathbf{X}$  and  $\mathbf{c}$  is a vector that has elements  $\sum_{a=1}^m x_{ia}^2$ , the diagonal elements of  $\mathbf{XX}'$ . The matrix  $\mathbf{B} = \mathbf{XX}'$  is called a scalar product matrix.

The problem here is to arrive at a scalar product matrix  $\mathbf{B}$  given a matrix of squared distances  $\mathbf{D}^{(2)}$ . Since distances do not change under translations, we assume that  $\mathbf{X}$  has column means equal to 0. Multiplying the left and the right sides of Eq. (1) by the centering matrix  $\mathbf{J} = \mathbf{I} - n^{-1} \mathbf{1}\mathbf{1}'$  and by the factor  $-\frac{1}{2}$  gives

$$\begin{aligned} -\frac{1}{2} \mathbf{JD}^{(2)}\mathbf{J} &= -\frac{1}{2} \mathbf{J}(\mathbf{c}\mathbf{1}' + \mathbf{1}\mathbf{c}' - 2\mathbf{XX}')\mathbf{J} = -\frac{1}{2} \mathbf{J}\mathbf{c}\mathbf{1}'\mathbf{J} - \frac{1}{2} \mathbf{J}\mathbf{1}\mathbf{c}'\mathbf{J} + \frac{1}{2} \mathbf{J}(2\mathbf{B})\mathbf{J} \\ &= -\frac{1}{2} \mathbf{J}\mathbf{c}\mathbf{0}' - \frac{1}{2} \mathbf{0}\mathbf{c}'\mathbf{J} + \mathbf{J}\mathbf{B}\mathbf{J} = \mathbf{B} \end{aligned} \quad (2)$$

The first two terms are zero because centering a vector of ones yields a vector of zeros ( $\mathbf{1}'\mathbf{J} = 0$ ). The centering around  $\mathbf{B}$  can be removed because  $\mathbf{X}$  is column centered, and hence so is  $\mathbf{B}$ . The operation in Eq. (2) is called double centering. To find the MDS coordinates from  $\mathbf{B}$ , we factor

$\mathbf{B}$  by eigendecomposition,  $\mathbf{Q}\Lambda\mathbf{Q}' = (\mathbf{Q}\Lambda^{1/2})(\mathbf{Q}\Lambda^{1/2})' = \mathbf{XX}'$ . In classical scaling, the  $\mathbf{D}^{(2)}$  matrix is replaced by the squared dissimilarities  $\Delta^{(2)}$ .

The procedure for classical scaling is summarized as follows:

- Compute the matrix of squared dissimilarities  $\Delta^{(2)}$ .
- Apply double centering to this matrix:

$$\mathbf{B}_\Delta = -\frac{1}{2} \mathbf{J}\Delta^{(2)}\mathbf{J}.$$

- Compute the eigendecomposition of  $\mathbf{B}_\Delta = \mathbf{Q}\Lambda\mathbf{Q}'$
- Let  $m$  be the dimensionality of the solution,  $\Lambda_+$  the matrix of the first  $m$  eigenvalues greater than zero, and  $\mathbf{Q}_+$  the first  $m$  columns of  $\mathbf{Q}$ . Then, the coordinate matrix of classical scaling is given by  $\mathbf{X} = \mathbf{Q}_+ \Lambda_+^{1/2}$ .

Classical scaling minimizes the loss function (strain),

$$L(\mathbf{X}) = \left\| -\frac{1}{2} \mathbf{J}[\mathbf{D}^{(2)}(\mathbf{X}) - \Delta^{(2)}]\mathbf{J} \right\|^2 = \left\| \mathbf{XX}' + \frac{1}{2} \mathbf{J}\Delta^{(2)}\mathbf{J} \right\|^2 = \left\| \mathbf{XX}' - \mathbf{B}_\Delta \right\|^2.$$

## Localization Using 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 inter-point 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 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 inter-point distances make no reference to any absolute coordinates.

MDS can be seen as a set of data analysis techniques that display the structure of distance-like data as a geometrical picture (Borg and Groenen, 1997). As shown formally in the previous subsection, MDS starts with one or more distance (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. 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.

There are many types of MDS techniques. They can be classified according to whether the similarity data is qualitative (non-metric 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 distance 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 chapter. 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 (Torgerson, 1965). 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. This is very helpful when we apply it to localization, as our distance estimates can be very rough indeed. Because classical metric MDS has an analytical solution, it can be performed efficiently on large matrices.

In non-metric (also called ordinal) MDS (Shepard, 1962), 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 non-metric 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, non-metric 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.

A thorough analysis of the localization error bounds has been done (Shang *et al.*, 2004). The localization problem has been treated as an estimation problem. The Cramér-Rao error bounds have been derived when the distances between all nodes were used, which was the case for MDS. However, a detailed analysis of error bounds is beyond the scope of this chapter.

## SIMULATION SETUP

In the experiments reported, we assess the average-case performance of the localization methods presented by simulation on Matlab 7.0 on 2-dimensional networks of at least 100 nodes deployed inside a  $10r \times 10r$  square field, where  $r$  is the placement unit length. Two example scenarios are shown in Figure 1: (a) regular networks – 200 nodes are randomly placed in a  $10r \times 10r$  square and (b) irregular network

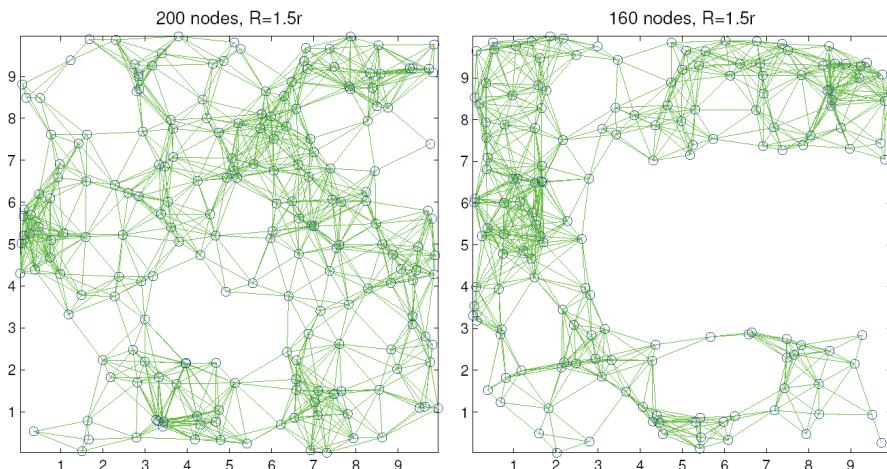
– 160 nodes are randomly placed in an area of C shape within a  $10r \times 10r$  square. Circles represent sensor nodes and lines represent connections between nodes that are within communication range of each other. The radio range is  $1.5r$ , where  $r$  is the placement unit length. The average node degrees of the two problems are 12.1 and 11.5, respectively. For each type of networks, the algorithms are run on at least 30 randomly-generated network instances.

All nodes are assumed to have the same radio range  $R$ , which is modeled as a circle with a predefined radius. We do not consider models of non-uniform radio propagation or widely varying ranging errors. Hence, all communications are assumed to be bidirectional, i.e., if node  $i$  can communicate with node  $j$ , then node  $j$  can communicate with node  $i$ . The average node degree (average number of neighbors) is controlled by specifying  $R$ . The errors of position estimates are normalized to  $R$  (i.e., 50% position error means half of the range of the radio). The resulting average node degree is a function of both  $R$ , network type (topology), and number of nodes. For example, a 200-node regular network with  $R$  equal to  $1.5r$  has an average node degree of 12.1. A 160-node irregular network with  $R$  equal to  $1.5r$  has an average node degree of 11.5. The distance measure is modeled as the true distance blurred with Gaussian noise. Assume the true distance is  $d$  and the standard deviation of the range error is  $e_r$ , then the measured distance  $\hat{d}$  is a random value drawn from the normal distribution  $d(1+N(0,e_r))$ . For simplicity, we will refer to  $e_r$  by just “range error” throughout this chapter.

### **MDS-MAP(C) AND MDS-MAP(C,R)**

The simplest MDS-based localization method is MDS-MAP(C). It builds a global map using a single application of classical MDS. The parameter “C” refers to “centralized”, as the connectivity information of the network is sent to a central location where the computation is carried out. The method with additional refinement to MDS-MAP(C) is called MDS-MAP(C,R), where the parameter “R” is for “refinement”.

*Figure 1. Two example problems similar to those in simulation*



## Algorithm

MDS-MAP(C) consists of three steps as follows.

- Compute the shortest paths between all pairs of nodes in the region of consideration. The shortest path distances are used to construct the distance matrix  $\mathbf{D}$  for MDS.
- Apply MDS to  $\mathbf{D}$ , retaining the first 2 (or 3) largest eigenvalues and eigenvectors to construct a 2-D (or 3-D) relative map.
- 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 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 to all edges the value 1 multiplied by the transmission range. Then, a shortest-path algorithm, such as Dijkstra's or Floyd's, can be applied to find the shortest path between all pairs of nodes. 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 to each node. Although these locations may be accurate relative to one another, the entire map will be arbitrarily rotated, translated, and flipped relative to the true node positions.

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

In MDS-MAP(C,R), the following refinement step is added between steps 2 and 3 of MDS-MAP(C) to improve the relative map.

- Using the position estimates of nodes in the MDS solution as an initial solution, apply least-squares minimization to improve the match between the measured distances between neighboring nodes and their distances in the solution.

Our formulation of the refinement is more general than previous methods (Savarese *et al.*, 2002; Savvides *et al.*, 2002) in two ways: (1) In addition to the information between 1-hop neighbors, information between multihop neighbors is also used, but with different weights. (2) Instead of refining the coordinates of one node at a time while all other nodes remain fixed, the coordinates of all nodes in the relative map are variables in a single optimization. We use a refinement range  $R_{ref}$ , defined based on hops, to specify how much information is considered.  $R_{ref} = 1$  means only information between 1-hop neighbors are used,  $R_{ref} = 2$  means information of nodes within two hops is used, and so on. Different values of  $R_{ref}$  offer a trade-off between computational cost and solution quality.

An important advantage of our refinement approach is that MDS can provide better starting points for the least-squares minimization than other triangulation-based or heuristic methods (Savvides *et*

al., 2002). The least-squares minimization problem is high-dimensional and has lots of local minima. Random starting points usually lead to very bad solutions. MDS is good at finding the right general topology of a network, which corresponds to a starting point in the basin of attraction of an optimal or near-optimal solution.

More formally, let  $(x_i, y_i)$ ,  $i = 1, \dots, N$ , represent the coordinates of the  $N$  nodes in a 2-D local map,  $d_{ij}$  the Euclidean distance between two nodes  $i$  and  $j$  in a candidate solution, and  $p_{ij}$  the measured proximity between nodes  $i$  and  $j$ . When only proximity information is available,  $p_{ij} = 1$  if  $i$  and  $j$  are 1-hop neighbors. When distance measurements between 1-hop neighbors are available,  $p_{ij}$  is the distance between  $i$  and  $j$  if they are 1-hop neighbors or the shortest path distance if  $i$  and  $j$  are further apart. The objective of the refinement step is

$$\min_{x_k, y_k} \sum_{i,j} w_{ij} (d_{ij} - p_{ij})^2, \text{ for } k = 1, \dots, N \quad (3)$$

where  $w_{ij}$  are the weights.

For a 2-D  $n$ -node network, the problem has  $2n$  variables and no constraints. The Jacobian can be computed analytically. In our experiments, we use the Levenberg-Marquardt method (“lsqnonlin” in Matlab’s optimization toolbox) to solve the problem. Usually only the first few iterations of “lsqnonlin” give significant improvement. Thus, the maximum number of iterations is set to a small number, such as 20. Although this local optimization algorithm is fast, it is considerably slower than classical MDS. For 100-node networks, it is more than an order of magnitude slower. For larger networks, the time difference becomes larger.

## Illustrative Examples

Figure 2 shows the results of the range-free versions of MDS-MAP(C) and MDS-MAP(C,R) on a regular network example. Four random anchor nodes, denoted by asterisks, are used to estimate the transformation to absolute coordinates. The circles represent the true locations of the nodes and the solid lines represent the errors of the estimated positions from the true positions. The longer the line, the larger the error is. The average errors of MDS-MAP(C) and MDS-MAP(C,R) are  $0.67r$  and  $0.35r$ , respectively, where the field in which the nodes are placed measures  $10r$  by  $10r$ .

When distances between one-hop neighbors are known, the result of MDS-MAP(C) can be improved. Figure 3 shows results on the same network, but when distances between one-hop neighbors are known with 5% range error. The estimates of MDS-MAP(C) based on the same 4 anchor nodes have an average error of  $0.25r$ , much better than the result when using connectivity only ( $0.67r$ ). The result after refinement in MDS-MAP(C,R) is excellent. The average error is reduced to  $0.06r$ , where  $r$  is the placement unit length and is set to 1 in the experiments.

Irregular topologies are much harder than uniform topologies. Figure 4 shows sample results on the irregular network example. Again, there are four random anchor nodes. The result of MDS-MAP(C) is poor. Although the result of MDS-MAP(C,R) is better than that of MDS-MAP(C), it is much worse than its result on the uniform example. MDS-MAP(C) does not work well because the shortest-path distance between two nodes in different wings of the network is much larger than their actual Euclidean distance. The error of MDS-MAP(C) using connectivity information is very large,  $2.4r$ . The refinement in MDS-MAP(C,R) is useful and reduces the error to  $0.55r$ .

Figure 2. Results of range-free MDS-MAP(C) (left) and range-free MDS-MAP(C,R) (right) on a regular network example. Anchors are '\*'s. Average errors are  $0.67r$  and  $0.35r$ , respectively.

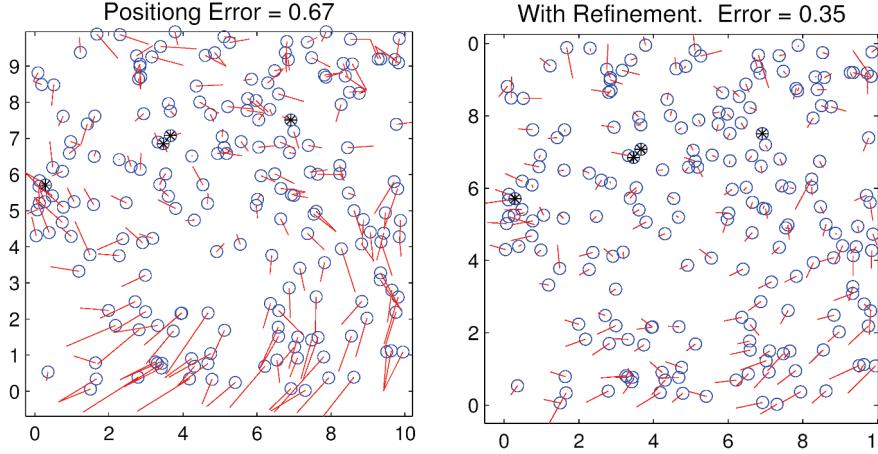
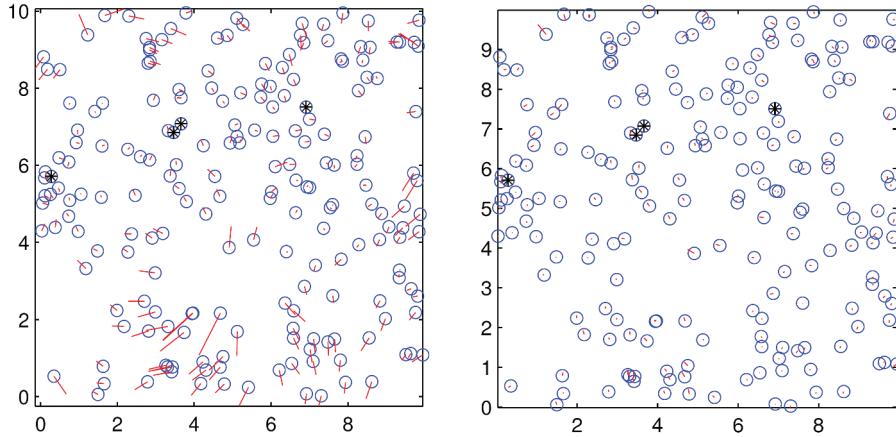


Figure 3. Results of range-aware MDS-MAP(C) (left) and range-aware MDS-MAP(C,R) (right) on a regular network example. Anchors are '\*'s. Average errors are  $0.25r$  and  $0.06r$ , respectively.

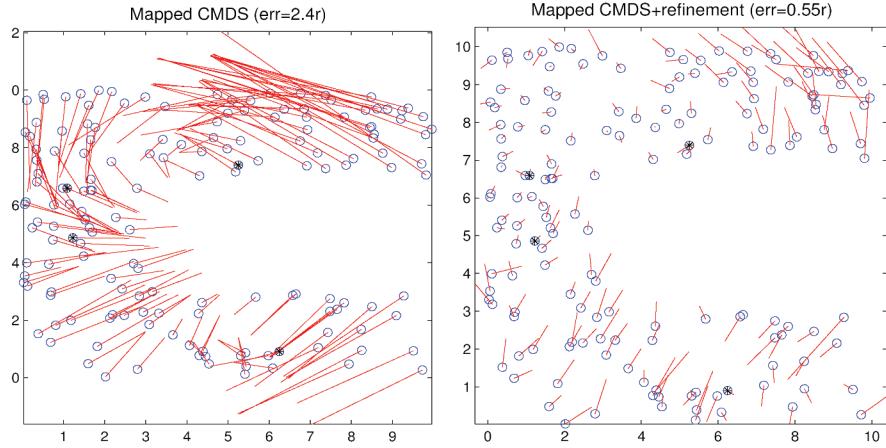


## Experimental Results

In the experiments reported here, 200-node regular networks and 160-node irregular networks are used. Moreover, we set the refinement range  $R_{ref}$  to 2 and the weights  $w_{ij}$  to 1 when  $i$  and  $j$  are one hop apart (see Eq. (5)). When the two nodes  $i$  and  $j$  are two hops apart, we tried several values for the weights  $w_{ij}$ , and the value 1/4 worked the best.

For regular networks, Figure 5 shows the performance of MDS-MAP(C) as a function of average node degree and number of anchors, using 4, 6, or 10 random anchors. Position estimates by MDS-MAP(C) have an average error under  $100\%R$  in scenarios with just 4 anchor nodes and an average node degree level of 8.9 or greater. On the other hand, when the average node degree is low, e.g., 5.9, the errors can be large. Having good estimates of the distances between neighbors leads to much better solutions when the average node degree is high. When the average node degree is 12.2 or greater, the errors are about half of those by the range-free version. On the other hand, when the average node degree is low, e.g.,

Figure 4. Results of range-free MDS-MAP(C) (left) and range-free MDS-MAP(C,R) (right) on an irregular network example. Average errors are  $2.4r$  and  $0.55r$ , respectively.



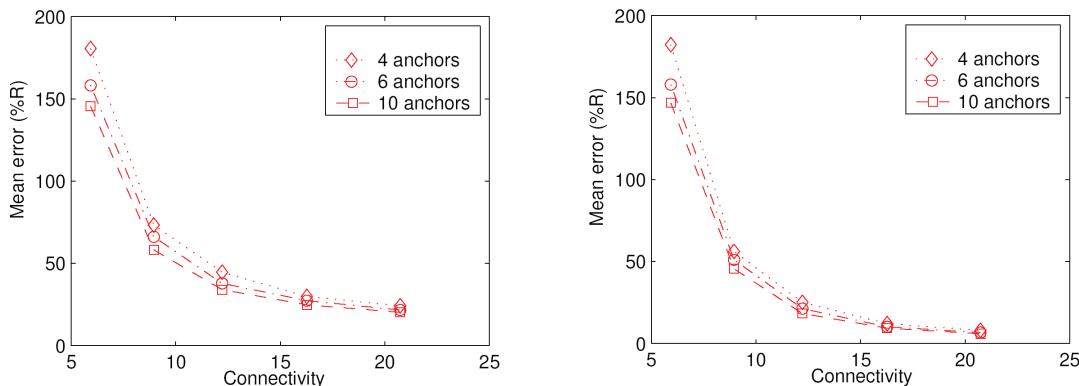
5.9, knowing the local distance does not help much. More experimental results will be shown compared to the performance of MDS-MAP(P) in the next section.

The refinement in MDS-MAP(C,R) improves the solutions significantly. The result can be misleading because it seems that the refinement is the most important and does all the work. This is not the case. From a random starting point, the refinement usually doesn't do much and just returns a bad solution, because there are many local minima. Thanks to MDS, the relative map often has the right topology, which corresponds to a good starting point in the same basin as the optimal or near-optimal solution. This is why the refinement performs so well.

### MDS-MAP(P) AND MDS-MAP(P,R)

MDS-MAP(C) and MDS-MAP(C,R) do not work well on irregular networks because they rely on shortest-path distance estimation, which can have large errors for remote nodes. Another problem with

Figure 5. Results of MDS-MAP(C): range-free (left) and range-aware (right) on 200-node regular networks.



these centralized methods is that they can not be applied easily to large networks, for which reading out the connectivity and distance information is potentially prohibitive. In such cases, in-network computation of coordinates would be much more attractive. MDS-MAP(P) addresses both of these problems.

MDS-MAP(P) is more complicated than MDS-MAP(C). It builds many local maps and then patches them together to form a global map. This method relies on local information and avoids using the distance estimation between remote nodes. As we will show, it achieves better results on irregular networks. Another benefit of MDS-MAP(P) is that it can be easily executed in a distributed fashion. When we add refinement to improve the global map, we call the method MDS-MAP(P,R).

## Algorithm

In MDS-MAP(P), individual nodes simultaneously compute their own local maps using their local information. Then these maps can be incrementally merged to form a global map. The steps of MDS-MAP(P) are as follows.

- Set the range for local maps,  $R_{lm}$ . For each node, neighbors within  $R_{lm}$  hops are involved in building its local map. We use  $R_{lm} = 2$  in our experiments.
- For each node, apply MDS-MAP(C,R) to the nodes within range  $R_{lm}$  to generate its local map.
- Merge local maps. Local maps can be merged in various ways. We use a simple strategy: first randomly pick a node and start with its local map; then merge in the maps of neighboring nodes one by one. Each time, we choose the neighbor to merge whose local map shares the most nodes with the current map. Thus, the initial local map grows by incorporating other local maps and can eventually cover the entire network.
- 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.

Two maps are merged together based on the coordinates of their common nodes. The best linear transformation (minimizing discrepancy errors) is computed to transform the coordinates of the common nodes in one map to those in the other map. Given the coordinates of common nodes in maps  $A$  and  $B$  as matrices  $X_A$  and  $X_B$ , a linear transformation (translation, reflection, orthogonal rotation, and scaling) of  $X_B$  to best conform to  $X_A$  is determined. The “goodness-of-fit” criterion is the sum of squared errors, i.e.,  $\min_T \|T(X_B) - X_A\|_2$ , where  $T(\cdot)$  is the linear transformation.

This method allows for parallel and distributed implementations in several ways. First, the computation of local maps can be done locally at each node in parallel with the others. Second, the local maps can be merged in parallel in different parts of the network. Because the method does not require anchor nodes in order to build a relative map of a sub-network, it can be applied to many sub-networks in parallel. Third, the computation of absolute maps from anchor nodes could be applied to relative local maps and thus also be distributed in the network. For example, as soon as three or more anchors are present in a sub-network, an absolute map could be computed. Furthermore, all local maps bordering on this absolute map could be absorbed in parallel into that map using the merger step. For large networks and a sufficient number of anchor nodes, it should never be necessary to compute a single global map anywhere. Distributed map merging has a number of benefits, including more balanced computation and communication among the nodes, faster construction of the global map, and distribution of map information in the network at multiple levels of granularity, giving the opportunity for better flexibility and robustness.

The amount of error generated when two maps are merged depends on several factors, including the accuracy of the two maps and the number of common nodes. The error will propagate when a linear sequence of maps are merged. In dense networks, the adjacent local maps usually have many common nodes, and thus the error introduced in merging is small. In MDS-MAP(P,R), a refinement step is added between steps 3 and 4 of to improve the global relative map.

## Illustrative Examples

Using the two example problems from Figure 2, we illustrate the performance of MDS-MAP(P) and MDS-MAP(P,R). Figures 6 and 7 show the results on a regular network example for the range-free and range-aware scenarios. Using connectivity information only (range-free), the average error of MDS-MAP(P) is  $0.40r$ , about 60% of the error of MDS-MAP(C) in Figure 2, and slightly worse than MDS-MAP(C,R). After refinement, the error of MDS-MAP(P,R) is  $0.31r$ ; better than that of MDSMAP(C,R). Using local distances, MDS-MAP(P) and MDS-MAP(P,R) obtain much better results. The error of MDS-MAP(P) is  $0.16r$ , better than the  $0.25r$  error of MDS-MAP(C) in Figure 3. After refinement, the error of MDS-MAP(P,R) is  $0.06r$ , at the level of the distance estimation errors.

Figure 8 shows sample results on the irregular placement example for the range-aware case. The solution of MDS-MAP(P) (error  $0.72r$ ) is quite reasonable. The solution of MDS-MAP(P,R) is even better (error  $0.29r$ ).

## Experimental Results

Similar to the experiments done with MDS-MAP(C), 200-node regular networks and 160-node irregular networks are used. Figures 9 and 10 show the performance of MDS-MAP(P) and MDS-MAP(P,R) compared to MDS-MAP(C) and MDS-MAP(C,R) for regular networks. The errors are plotted against the average node degree. The radio range ( $R$ ) goes from  $1.25r$  to  $2.5r$ , in increments of  $0.25r$ , which leads to average node degrees of 8.9, 12.2, 16.4, 20.9, 25.9, and 31.1. Three or ten random anchors are used.

*Figure 6. Results of range-free MDS-MAP(P) (left) and range-free MDS-MAP(P,R) (right) on a regular network example. Average errors are  $0.40r$  and  $0.31r$ , respectively*

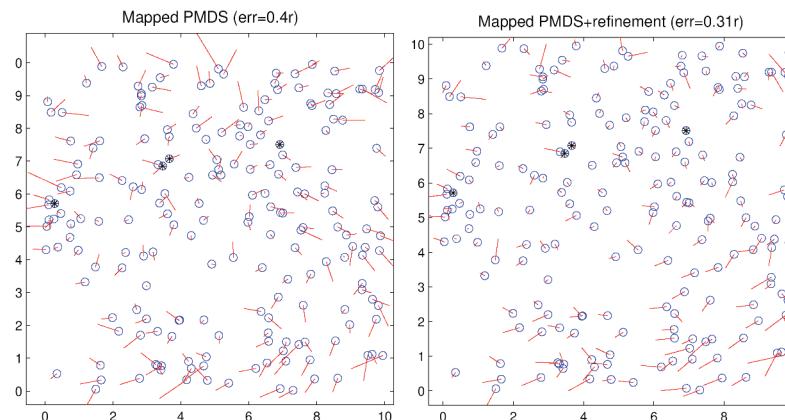
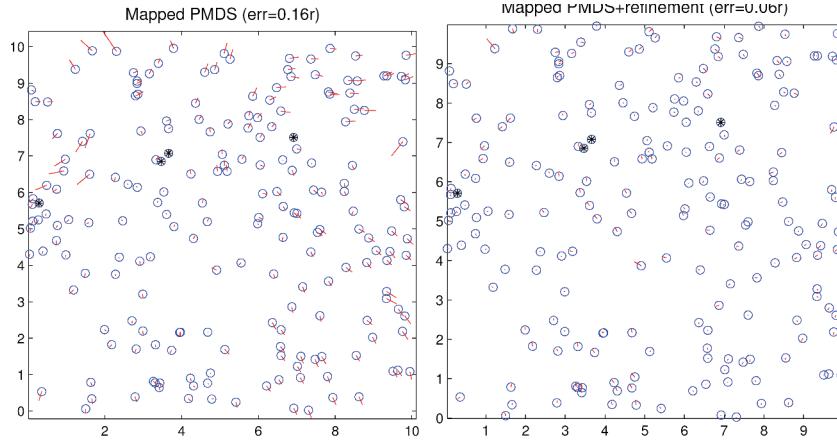


Figure 7. Results of range-aware MDS-MAP(P) (left) and range-aware MDS-MAP(P,R) (right) on a regular network example. Average errors are  $0.16r$  and  $0.06r$ , respectively



When using only connectivity information, MDS-MAP(P) is consistently better than MDS-MAP(C), more than 10%R better when the average node degree is low. MDS-MAP(C,R) and MDS-MAP(P,R) have comparable results and are better than MDS-MAP(P). Although more anchors lead to better results, the improvement with more than 6 anchors is small. For the range-free scenario, MDS-MAP algorithms are much better than the convex optimization approach in (Doherty *et al.*, 2001) 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 average node degree is 8.9 and above. The results are also better than Hop-TERRAIN (Savarese *et al.*, 2002), especially when the number of anchors is small. For example, with 4 anchors (2% of the network) and an average node degree 12.2, MDS-MAP(P) using connectivity information only has an average error of about 27%R, whereas Hop-TERRAIN has an average error of about 90%R.

Figure 8. Results of range-aware MDS-MAP(P) (left) and range-aware MDS-MAP(P,R) (right) on an irregular network example. Average errors are  $0.72r$  and  $0.29r$ , respectively

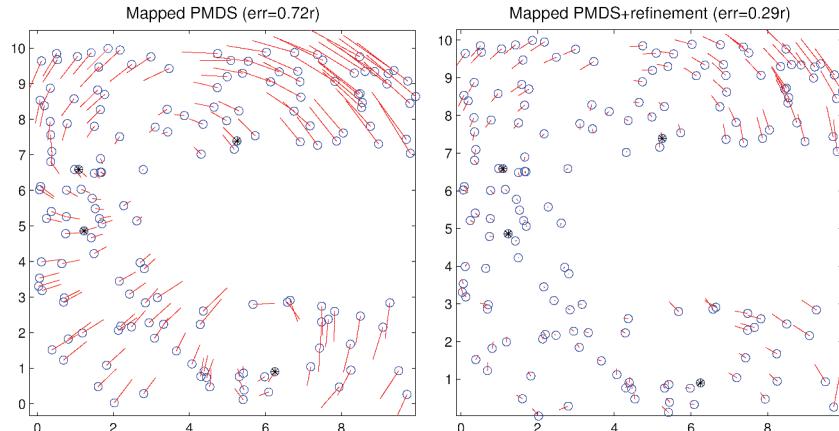


Figure 9. Performance of range-free MDS-MAP methods for regular networks with 3 (left) and 10 (right) anchors for different values of average node degree (horizontal axis)

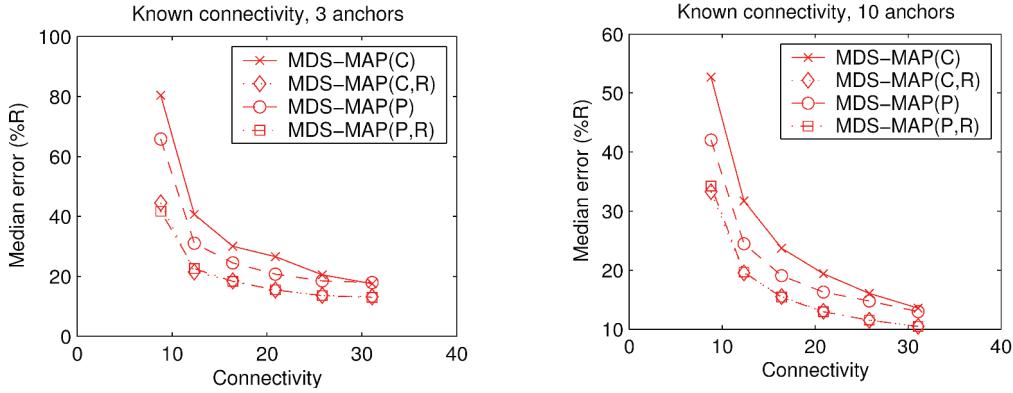
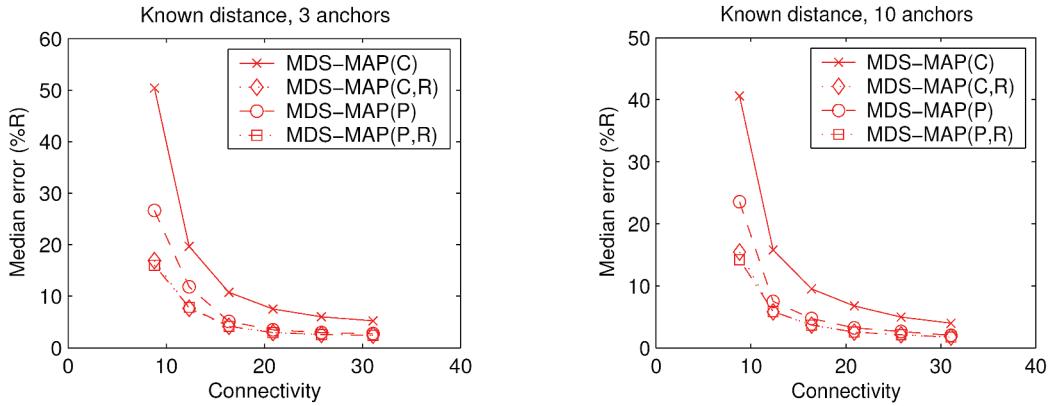


Figure 10. Performance of range-aware MDS-MAP methods for regular networks with 3 (left) and 10 (right) anchors for different values of average node degree (horizontal axis)



Using local distances with 5% error improves the performance of the MDS-MAP algorithms. Their errors are about half of those obtained using only proximity information. MDS-MAP(P) is comparable to MDS-MAP(C,R) and MDS-MAP(P,R) when the average node degree is 12.2 and above. MDS-MAP succeeds in localizing a higher fraction of the nodes in a network than most previous methods. MDS-MAP localizes all nodes in a connected network. In our experiments, when the average node degree is 12.2 or more, the network is usually a connected graph and all nodes are located.

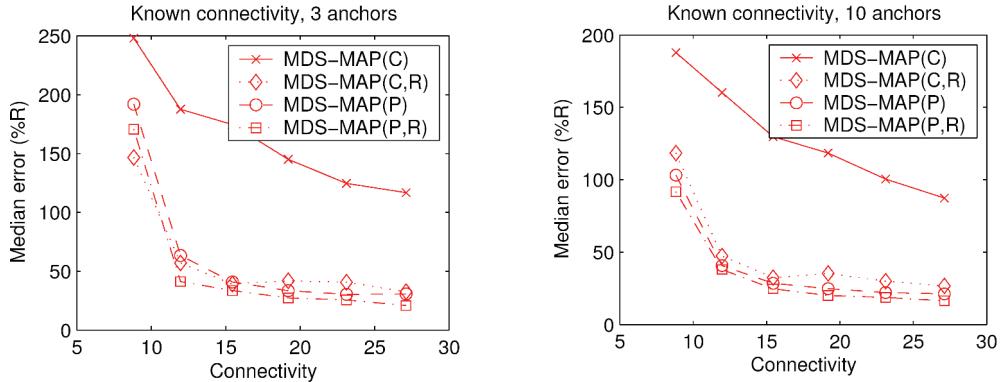
Irregular topologies are much harder than uniform topologies and previous methods reported very poor results on them (Niculescu and Nath, 2001). Figures 11 and 12 show the performance of the MDS-MAP algorithms on the C-shaped irregular networks. The radio ranges ( $R$ ) are from  $1.25r$  to  $2.5r$ , in increments of  $0.25r$ , which leads to average node degrees of 8.8, 12.0, 15.4, 19.2, 23.1, and 27.1. MDS-MAP(P) performs very well on these irregular networks, especially when the average node degree is 12.0 or more, finding solutions just slightly worse than those returned by MDS-MAP(C,R) and MDS-MAP(P,R). The results of MDS-MAP(P,R) are slightly better than those of MDS-MAP(C,R).

On networks with similar average node degrees, the results of MDS-MAP(C) on the irregular networks are worse than those on the uniform networks. In contrast, MDS-MAP(C,R), MDS-MAP(P), and MDS-MAP(P,R) perform well when the average node degree is relatively high. Having accurate estimates of local distances does not improve the performance of MDS-MAP(C), but helps MDS-MAP(P) and MDS-MAP(P,R) tremendously. As pointed out earlier, this is because the shortest-path distance between two nodes does not correspond well to their Euclidean distance. This has a greater impact on the centralized solution than on the distributed (patched) one. The results of MDS-MAP(P) and MDS-MAP(P,R) are very close, indicating that the refinement step in MDS-MAP(P,R) does not do much.

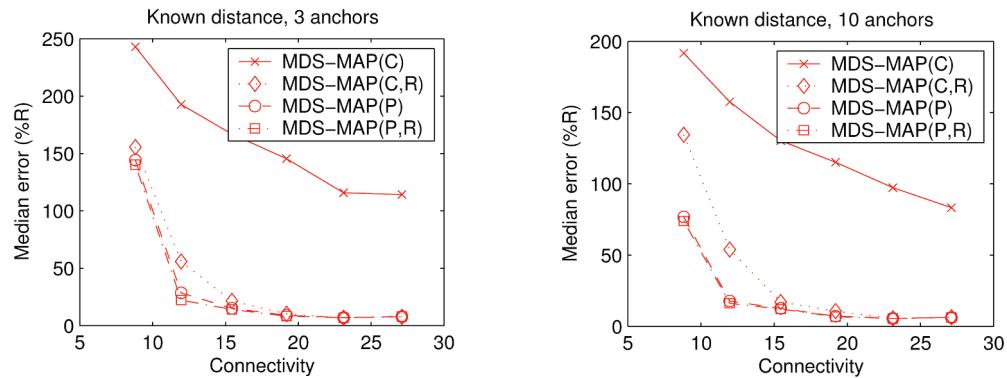
## MDS-HYBRID

Different relative and absolute localization methods have different performance and cost. The performance may be expressed by the localization error, which is either the average distance between the estimated and the true positions of a node in case of absolute localization or the average difference between the

*Figure 11. Performance of range-free MDS-MAP methods for irregular networks with 3 (left) and 10 (right) anchors for different values of average node degree (horizontal axis)*



*Figure 12. Performance of range-aware MDS-MAP methods for irregular networks with 3 (left) and 10 (right) anchors for different values of average node degree (horizontal axis)*



true and the estimated distances between nodes in relative localization. The cost may be measured by the localization delay or by the energy consumption. It is important when part of, or all, the nodes are moving. In this case, the localization process needs to be performed periodically or, at least, more than once. This makes the cost of localization, along with the performance, an important factor in choosing which method to use.

## Performance-Cost Metric

MDS-Hybrid uses the weighted sum of localization accuracy and cost as a metric to evaluate localization. This metric is called *Performance-Cost Metric (PCM)*. For absolute localization, the localization error may be computed as the average distance between the estimated and the true locations of all nodes. For relative localization, the resulting network is subject to translation, rotation, and reflection. Therefore, instead of using the same definition, the average estimated distances between all pairs of nodes are computed and compared to the corresponding true distances. Priyantha *et al.* (2003) proposed a performance metric that they refer to as the *Global Energy Ratio (GER)*. *GER* is defined as:

$$GER = \frac{\sqrt{\sum_{i,j:i < j} \hat{e}_{ij}^2}}{N(N-1)/2}$$

where  $e_{ij}$  is the difference between the true distance  $d_{ij}$  and the distance in the algorithm's result  $\hat{d}_{ij}$ ,  $\hat{e}_{ij} = (\hat{d}_{ij} - d_{ij})/d_{ij}$ , and  $N$  is the number of nodes. We believe that taking the square root of the whole fraction represents the root-mean-square error in a better way. In addition, we normalize this error by the radio range ( $R$ ) of a node. Therefore, to measure the localization error, we introduce the *Global Distance Error (GDE)* defined as:

$$GDE = \frac{1}{R} \sqrt{\frac{\sum_{i,j:i < j} \hat{e}_{ij}^2}{N(N-1)/2}}$$

Thus, the Performance-Cost Metric (*PCM*) may be defined as:

$$PCM = \alpha \cdot GDE + (1 - \alpha) C \quad (4)$$

where  $0 \leq \alpha \leq 1$  represents the degree of interest in the localization error as an evaluation criterion, and  $C$  is the localization cost defined as follows.

$$C = N_r c_1 + (N - N_r) c_2 \quad (5)$$

where  $N_r$  is the number of reference nodes localized with MDS-MAP( $C$ ), and  $c_1$  and  $c_2$  are the average costs of localizing one node using MDS-MAP( $C$ ) and method  $M$ , respectively, normalized by the initial energy at every node. Assuming that  $c_1 = kc_2$  for some real  $k > 0$ , Eq. (5) yields

$$C = c_2 [N + (k - 1)N_r] \quad (6)$$

Substituting (6) in (4) gives

$$PCM = \alpha \cdot GDE + c_2(1 - \alpha)[N + (k - 1)N_r] \quad (7)$$

## Algorithm

MDS-Hybrid consists of three phases: selecting reference nodes, localizing reference nodes, and localizing non-reference nodes. Three design parameters are associated with this method. First, how many reference nodes to use. Second, how to select them. Finally, whether to use as anchors all reference nodes or only the nearest  $n$  reference nodes. Determining these parameters to reach the required point of operation depends on the performance and the cost of both MDS-MAP(C) and  $M$ .

### Phase 1: Selecting Reference Nodes

Since the performance of some absolute localization algorithms depends on the placement of the anchors (Shang *e. al.*, 2004), we consider two approaches of selecting reference nodes: random and outer (along the outer perimeter of the network). The random selection can be done in a distributed way using nodes' *IDs*. For example,  $N_r$  nodes with the smallest *IDs* can be selected using distributed algorithms. For the outer selection, we extend the algorithm used by Priyantha *et al.* (2003), keeping it distributed. First, four nodes 1, 2, 3, and 4 are selected roughly at the corners of the network using only the distance measurements. Then, the algorithm proceeds iteratively, doubling the number of selected nodes with every iteration, till all the  $N_r$  nodes are selected. The algorithm is described below. For a simple description of the algorithm, we assume that  $N_r = 2^m$ , for some integer  $2 \leq m \lfloor \lg N \rfloor$ . Assume that  $d_{ij}$  is the shortest-path distance between nodes  $i$  and  $j$ .

1. Initialize two vectors  $S$  and  $S'$  of size  $N_r$  each to be empty.
2. Select a random node 0. This can be achieved by selecting the node with the smallest *ID* due to the random deployment.
3. Select reference node 1 such that  $d_{01}$  is maximized.  $S[1] \leftarrow 1$ .
4. Select reference node 2 such that  $d_{12}$  is maximized.  $S[3] \leftarrow 2$ .
5. Select reference node 3 such that  $(d_{13} + d_{23})$  is maximized.  $S[2] \leftarrow 3$ .
6. Select reference node 4 such that  $d_{34}$  is maximized.  $S[4] \leftarrow 4$ .
7.  $l \leftarrow 4$ .
8. Repeat until size of  $S = N_r$ .
  - Repeat for  $i=1$  to  $(l-1)$ 
    - Select reference node  $k \notin S'$  such that  $(d_{ki} + d_{k(i+1)})$  is maximized.
    - $S'[2i-1] \leftarrow S[i]$ ,  $S'[2i] \leftarrow k$
  - End
  - Select reference node  $k \notin S'$  such that  $(d_{kl} + d_{kl})$  is maximized.
  - $S'[2l-1] \leftarrow S[l]$ ,  $S'[2l] \leftarrow k$ .
  - $S \leftarrow S'$ ,  $l \leftarrow 2l$
  - End
9. Return  $S$

## Phase 2: Localizing Reference Nodes

MDS-MAP(C) is used to relatively localize the reference nodes selected in phase 1. Although it can give good results, it suffers from the high cost of computation to achieve a good solution. A relative map of the reference nodes is constructed. The work done by Rao *et al.* (2003) shows how the distance information can be exchanged between nodes in the case of outer reference nodes. The performance of MDS-MAP(C) is presented in section 4 of this chapter. Figure 13 illustrates the performance of MDS-MAP(C), expressed by  $GDE$ , under different conditions of ranging errors, average node degrees, and network topologies. In general, the localization error decreases with increasing average node degree. This is the case until the range error reaches some point beyond which the error may decrease if the average node degree is increased. The reason for that, as explained by Langendoen and Reijers (2003), is that a node will have more neighbors from which it can select the next hop on the shortest path. For large ranging errors, a node will prefer the shortest *measured* distance; this will underestimate the Euclidean distance, resulting in large localization errors for large average node degrees when the ranging error is beyond some value. For the isotropic networks, the localization error increases with larger ranging error, which is intuitive. On the other hand, it is hard to correlate the localization error and the ranging error in case of anisotropic networks, where MDS performs poorly. This is simply because the shortest-path distance becomes a bad approximation to the Euclidean distance.

## Phase 3: Localizing Non-Reference Nodes

The result of phases 1 and 2 is a set of nodes with known coordinates in some coordinate system. In this phase, an absolute localization method  $M$  is used to localize the rest of the network using the reference nodes as anchors.

In our simulation of this phase, we used the DV-distance version of the APS method developed by Niculescu and Nath (2001). Each node uses the shortest-path distance information to estimate its distances to anchors. Then, it performs multilateration to estimate its position. APS has the advantages of simplicity and low cost. However, MDS-MAP(C) outperforms APS for most of the network conditions (Shang *et al.*, 2004). The APS method has the following steps:

1. Reference nodes broadcast their positions throughout the network to all nodes.
2. Each reference node  $k$  receives the positions  $(a_j, b_j), j=1, \dots, N_r$ , of all reference nodes and also computes the shortest-path distance  $p_{kj}$  to each reference node.
3. Each reference node  $k$  computes its distance correction value,  $c_k$ .

$$c_k = \frac{\sum_{j=1}^{N_r-1} d_{kj}}{\sum_{j=1}^{N_r-1} p_{kj}},$$

where  $d_{kj} = \sqrt{(a_k - a_j)^2 + (b_k - b_j)^2}$  is the Euclidean distance between reference nodes  $k$  and  $j$ .

4. For each unknown node  $i$ , compute the shortest-path distance  $p_{ij}, j=1, \dots, N_r$ , to all reference nodes. To estimate the position of node  $i$ , perform multilateration based on all reference nodes as follows.

Figure 13. Performance of MDS-MAP(C) expressed by GDE for regular networks (left) and irregular networks (right) for different values of average node degree (horizontal axis)

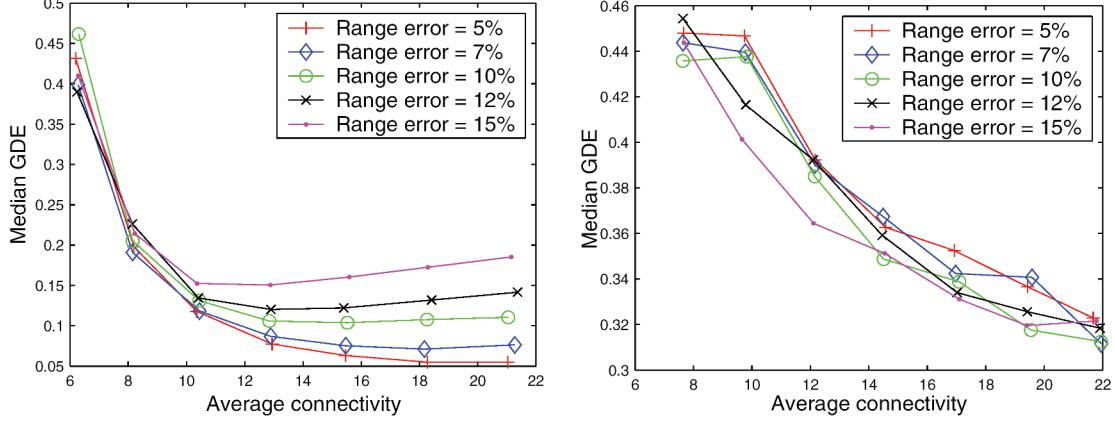
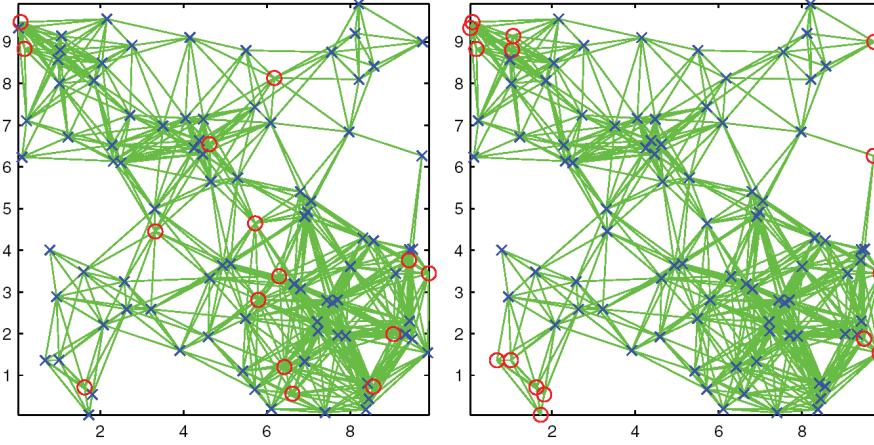


Figure 14. MDS-Hybrid phase 1: selecting reference nodes. This could be randomly (left) or along the outer perimeter of the network (right). Circles represent reference nodes.



- a. A system of quadratic equations of two variables is formed.

$$(x_i - a_j)^2 + (y_i - b_j)^2 = (c_s p_{ij})^2 \quad (8)$$

where  $s$  is the closest reference node to node  $i$ , i.e.,  $p_{is} \leq p_{ij}$  for  $j = 1, \dots, N_r$ .

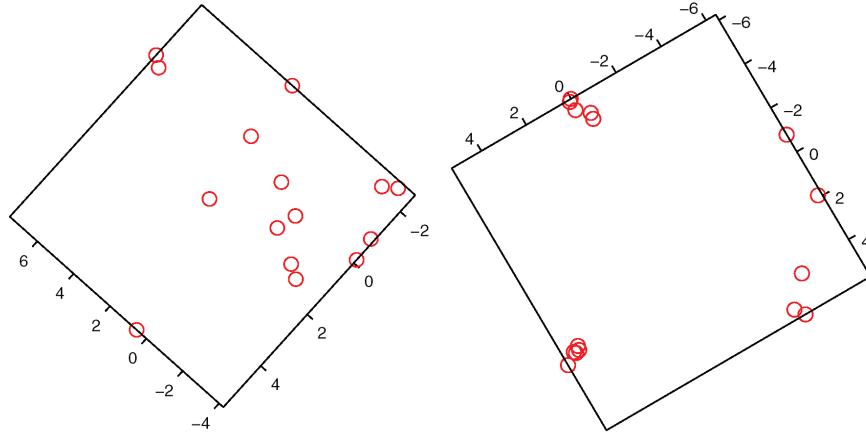
- b. The system of equations (8) is linearized by subtracting one equation, e.g., the first one, from the rest.

$$2(a_1 - x_j)x_i + 2(b_1 - b_j)y_i + a_j^2 + b_j^2 - a_1^2 - b_1^2 = (c_s p_{ij})^2 - (c_s p_{i1})^2 \quad (9)$$

for  $j = 2, \dots, N_r$ .

- c. The linear system (9) is solved. Then, using the solution as the initial point, the nonlinear system in (8) is solved using least-squares minimization.

Figure 15. Result of MDS-Hybrid, phase 2. GDE is 0.0944 for random selection (left) and 0.2154 for outer selection (right).



## Illustrative Examples

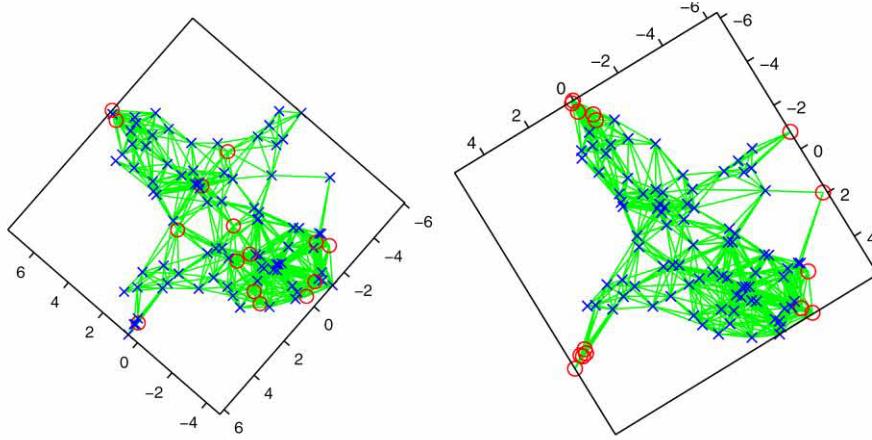
We illustrate the operation of MDS-Hybrid by two concrete phase-by-phase examples for regular networks: one for random selection of reference nodes and one for outer selection. Figure 14 shows the result of phase 1 using 15 reference nodes selected randomly (left) and along the outer perimeter of the network (right). Circles represent reference nodes selected, and  $\times$ 's represent regular nodes. In phase 2, MDS-MAP(C) is applied to estimate the positions of reference nodes. Figure 15 shows the result of this phase, during which a coordinate system is created. The estimated networks were reflected and/or rotated to be easily compared to the true networks in Figure 14. Range error = 5%.

In the final phase, APS is used to localize the rest of the networks using the reference nodes as anchors. Figure 16 shows the result of this phase. Again, the estimated networks were reflected and/or rotated to be easily compared to the true networks in Figure 14. APS uses all reference nodes. Range error = 5%,  $\alpha = 0.8$ ,  $c_2 = 0.005$ , and  $k = 1.5$ , as in Eq. (7).

## Experimental Results

Experiments have been done based on the simulation setup explained earlier in this chapter with 100-node networks, and the results reported are the median of 100 runs. Figure 17 gives sample results for regular networks where only the nearest 4 reference nodes are used and for both random and outer selection of reference nodes. It is for a 5% range error,  $k=1.5$ ,  $c_2 = 0.005$ , and  $\alpha = 0.8$  (see Eq. (7)). The point of the minimum  $PCM$  is the required point of operation if we are 80% interested in a good accuracy and 20% in a low cost ( $\alpha = 0.8$ ). Table 1 summarizes the values of the three design parameters associated with MDS-Hybrid that give the minimum  $PCM$  using APS for method  $M$ . An entry  $(F,S,H)$  is an ordered triple that represents  $F$  reference nodes,  $S$  of them were used by APS as anchors ( $A$  for all and  $T$  for nearest 4), and they were selected using the  $H$  method ( $L$  for random and  $O$  for outer). We note the following:

Figure 16. Result of MDS-Hybrid, phase 3. For random reference nodes (left), overall GDE = 0.1439. For outer reference nodes (right), overall GDE = 0.1634.



1. There is no entry that has the value 100 for the number of reference nodes. Thus, MDS-Hybrid is better than MDS-MAP(C).
2. For irregular networks, the results are the best for the smallest number of reference nodes (5 and 10). This is because MDS-MAP(C) performs poorly for irregular networks, as shown in Figure 18.
3. For regular networks, using all reference nodes is preferred to using only the nearest 4. This is due to the nature of the topology and the fact that using more anchors by APS gives more accurate position estimates.
4. For irregular networks with high average node degrees, using the nearest 4 reference nodes is better than using all of them because of the shortest-path problem discussed earlier. For low average node degrees, both approaches are close, and using all reference nodes might be better in order to tolerate the poor shortest-path distances.

Figure 17. Performance of MDS-Hybrid for regular networks with random (left) and outer (right) selections of reference nodes. Only the nearest 4 reference nodes are used. Range error = 5%,  $\alpha = 0.8$ ,  $c_2 = 0.005$ , and  $k = 1.5$ , as in Eq. (5).

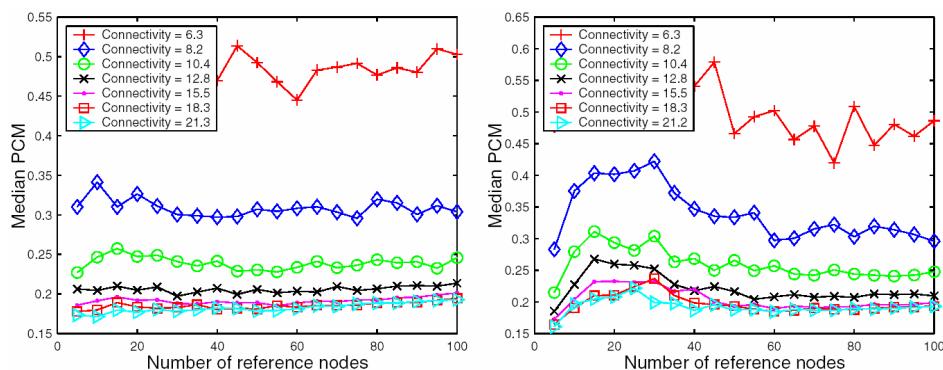
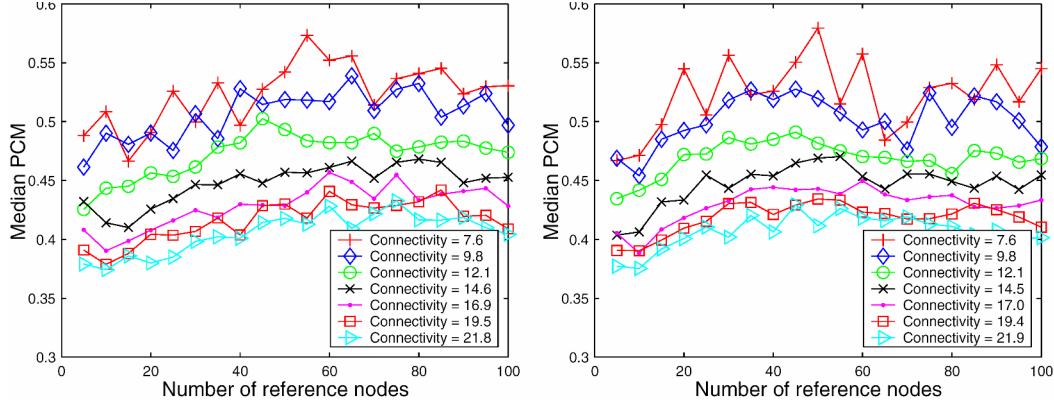


Figure 18. Performance of MDS-Hybrid for irregular networks with random (left) and outer (right) selections of reference nodes. All reference nodes are used. Range error = 5%,  $\alpha = 0.8$ ,  $c_2 = 0.005$ , and  $k = 1.5$ , as in Eq. (5).



5. APS performs slightly better for outer placement of reference nodes. However, this performance might be degraded by the poorly estimated long distances to reference nodes. Therefore, we see both the random and the outer selection of reference nodes in case of regular networks. For the irregular networks, the outer selection is usually better because of the small number of reference nodes that would give a lower *PCM*.

## RANGEQ-MDS

MDS-MAP(C), MDS-MAP(P), and MDS-Hybrid use the shortest-path distances for range-aware localization and the shortest-hop count multiplied by the range for range-free localization. RangeQ-MDS still uses the shortest-hop count, but each hop distance is obtained by the range quantization process. RangeQ-MDS is a *partially* range-aware localization algorithm that is based on Partial Range Information (*PRI*) presented by Li *et al.* (2004). *PRI* is defined as any type of measurement which is

Table 1. Design parameters for MDS-Hybrid's minimum *PCM* for both regular and irregular networks

	Average Node Degree	Range Error (%)		
		5	10	15
Regular	6.3	(15,A,O)	(15,A,O)	(15,A,O)
	10.4	(20,A,L)	(10,A,L)	(15,A,L)
	15.5	(15,A,L)	(5,A,O)	(5,A,L)
	21.3	(20,A,L)	(10,A,O)	(5,T,O)
Irregular	7.6	(5,A,O)	(10,A,L)	(5,A,O)
	12.1	(5,T,O)	(5,T,L)	(5,A,O)
	17.0	(5,T,O)	(5,T,O)	(5,T,O)
	21.8	(5,T,O)	(5,T,O)	(5,T,O)

monotonically increasing or decreasing and has an unknown or environment-dependent one-to-one relationship with the range measurement. It is called partial range information because these types of measurement can not be easily converted to accurate distance measurement due to the unknown exact mapping, yet the *PRI* values can correspond to the distance values based on their monotonic one-to-one relationship. It can be utilized in any range-free localization algorithms to improve their performance. One of the *PRI* examples is Received Signal Strength Indication (RSSI), which is a measure of the RF energy received.

Although wireless sensor systems usually have available RSSI readings, this information has not been effectively used for localization purposes. The RangeQ-MDS algorithm uses a sorted RSSI quantization scheme, called RangeQ, to improve the range estimation accuracy when distance information is not available. The output of this scheme is the distance matrix  $D$ , using which MDS is applied to obtain position estimates, resulting in a partially-range-aware method (Li *et al.*, 2004). The performance of RangeQ-MDS for various sensor networks is shown with experimental results from our extensive simulation with a realistic radio model.

## Sorted RSSI Quantization

Received signal strength indication (RSSI) is a measure of the RF energy received and is closely related to the range. RSSI is supported by sensor node hardware, such as the Berkeley motes (Whitehouse and Culler, 2002). For localization purposes, the information provided by RSSI or similar types of measurements can be used to improve the accuracy of range-free localization algorithms.

The concept of sorted RSSI quantization is similar to that of image quantization in image processing, except that the quantization is not from continuous RSSI to discrete RSSI. The process of sorted RSSI quantization starts with sorting RSSI readings to obtain a sorted range list. It then applies a quantizer on the list to generate a range estimation. In the quantization process, range level represents the number of measurable range units in a hop, which is similar to the gray level representing the intensity of a pixel in a gray-level image. The number of range levels in range quantization is referred to as range-level resolution. For example, the range-level resolution in range-free localization algorithms is  $s = 1$  since each 1-hop connection has one range level with the same range-level value. After obtaining the RSSI values, the sorted RSSI quantization algorithm follows two steps to assign a range value for each 1-hop connection as follows.

### Step 1: Sorting RSSI Values

Let  $N_i$  be the set of all 1-hop neighboring nodes to node  $i$  in a randomly-deployed sensor network and  $p_{ij}$  the RSSI value between node  $i$  and node  $j \in N_i$ . In this step, all  $p_{ij}$  values are sorted in an ascending order by their values, and all nodes in  $N_i$  are rearranged accordingly. The result is an ordered node list  $L_i$  of all neighboring nodes to node  $i$ .

### Step 2: Quantization

The goal of this step is to estimate the distance of each 1-hop connection. The range-free versions of MDS-MAP(C) and MDS-MAP(P) set all the 1-hop distances to the same value, i.e., the radio range  $R$  or a distance correction obtained by dividing the summation of the true shortest-path distance by the

summation of the shortest-hop count of a number of anchors. In the sorted RSSI quantization scheme, and for node  $i$ , a hop is divided into  $s_i$  sub-unit hops of size  $u_i$  each, where  $u_i = R/s_i$ . The number of sub-unit hops  $s_i$  is called *range-level resolution*. Each 1-hop connection to node  $i$  is assigned a *range-level value* of  $u_i, 2u_i, \dots, s_i u_i$ . If the assignment is correct, it obtains a range-level value closer to the true distance for each 1-hop connection than  $R$  as in MDS-MAP(C) and MDS-MAP(P). RangeQ-MDS assumes there is no available mapping function between the received RSSI and the corresponding distance to a neighboring node. Therefore, the problem left unsolved is to find an effective distance distribution model so that the number of nodes falling into each range level  $R_{ij}$  of node  $i$ , ( $j = 1, 2, \dots, s_i$ ) can be estimated with the help of the range order obtained from Step 1. For each node  $i$ , let the size of its neighboring node ordered list  $L_i$  be  $n_i$ . The quantization process involves dividing the maximum 1-hop range ( $R$ ) into  $s_i$  smaller quantities of size  $u_i$  each and then assigning an appropriate quantized range-level value,  $R_{ij}$ , to each neighbor in  $L_i$ . With range-level resolution  $s_i$ ,  $L_i$  is divided into  $s_i$  clusters with  $m_{ij}$  nodes in the  $j$ -th cluster  $C_{ij}$  for node  $i$ , where  $\sum_{j=1}^{s_i} m_{ij} = n_i$ . The range-level value for each node  $k \in C_{ij}$  is set to  $R_{ij} = ju_i$ .

We use an area-proportional model shown in Figure 19 to assign a certain number of nodes from the sorted list into each range level. The area-proportional model estimates the distribution of nodes in the neighborhood. Assuming that the nodes are randomly distributed, we know that the nodes are equally likely to fall into any spot in a circle with radius  $R$ . For node  $i$ , we cut the whole circle into  $s_i$  annuli of equal width. The expected number of nodes falling into the  $j$ -th annulus is

$$m_{ij} = \frac{\pi (2j-1)n_i u_i^2}{\pi (s_i u_i)^2} = \frac{(2j-1)n_i}{s_i^2}$$

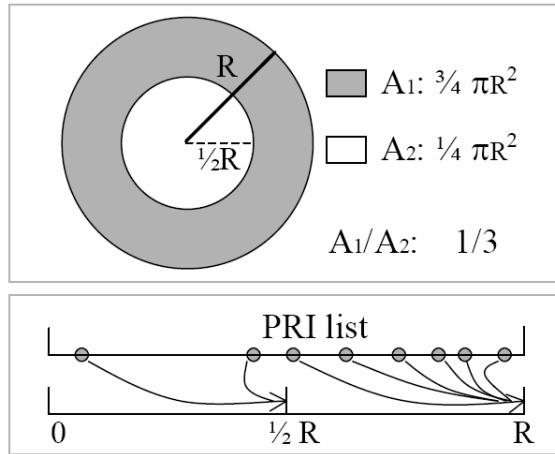
Therefore, the expected numbers of nodes falling into the  $s_i$ -th's annuli are

$$\frac{n_i}{s_i^2}, \frac{3n_i}{s_i^2}, \dots, \frac{(2j-1)n_i}{s_i^2}, \dots, \frac{(2s_i-1)n_i}{s_i^2}$$

## Algorithm and Experimental Results

The pair-wise distance obtained by RangeQ is more like an estimated shortest-path distance except that the 1-hop range estimation itself is not provided by the hardware but provided by RangeQ instead. Simulation results show that the performance of RangeQ-MDS is between those of range-free and range-aware algorithms. Compared to MDS-MAP(C) and MDS-MAP(P), RangeQ-MDS is better than the range-free version of MDS when the range error is less than 35% of the radio range  $R$ , and better than the range-aware version of MDS when the error is more than about 16% of the radio range, as shown in Figure 20, where  $R$  is set to  $1.75r$ . RangeQ range estimation is more accurate than both range-free and range-aware when the range error is between 15% and 35% of radius. Figure 21 shows that the accuracy performance ranking of all the 6 listed algorithms from best to worst follows MDS-Range, MDS-RamgeQ, APS-Range, APS-RangeQ, APS-Hop and MDS-Hop.

Figure 19. Area-proportional model for the RangeQ-MDS algorithm



## Error Analysis

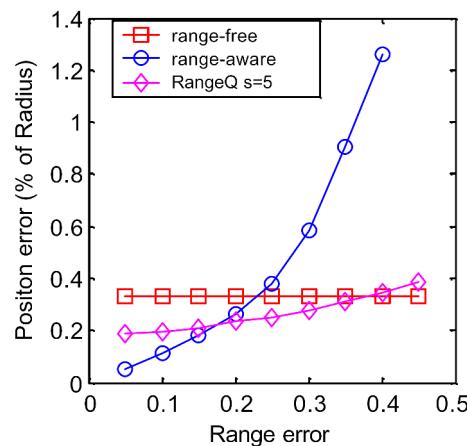
In addition to the localization accuracy comparison, we also use the Cramér-Rao Lower Bound (CRLB) model to formulate the error of RangeQ range estimation technique (Shi *et al.*, 2005). Let  $d_{ij}$  be the true distance between sensor nodes  $i$  and  $j$ . That is,

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

The power  $P_j$  (dBm) transmitted by device  $i$  and received at device  $j$  can be formulated as

$$P_i = P_\circ - 10n_p \log_{10} \frac{d_{ij}}{\Delta_\circ}$$

Figure 20. Performance of RangeQ-MDS



where  $P_{\circ}$  (dBm) is the received power at the reference distance  $\Delta_{\circ}$ . Typically,  $\Delta_{\circ} = 1$  meter, and  $P_{\circ}$  is calculated from the free space loss formula (Rappaport, 2002). The path-loss exponent  $n_p$  depends on the environment, with typical value between 2 and 4.

Patwari and Hero III (2003) claim that the power loss has a log-normal distribution with if  $n_p$  is a fixed constant. The experimental results supporting their claim are 946 pair-wise RSSI measurements with a DS-SS transmitter and receiver in a network of 44 device locations. In the quantile-quantile plot of comparing the distribution of

$$P_i - (P_{\circ} - 10n_p \log_{10} \frac{d_{ij}}{\Delta_{\circ}})$$

(i.e., the attenuation of the channel) to the Gaussian distribution, they match well in the middle part. At both ends, however, they do not match well. A more realistic assumption is that both the environment variance and the measurement errors affect the received power. We assume that  $n_p$  is a Gaussian random variable, i.e.,  $n_p \approx N(\alpha, \delta_{n_p}^2)$  and the measurement error is Gaussian noise,  $N(0, \delta_{n_p}^2)$ . The assumption that  $n_p$  can be modeled as a Gaussian random variable is supported by a few researchers' work. Based on the data obtained by Seidel and Rappaport (1992), the mean of  $n_p$  is 3. Ghassemzadeh *et al.* (2002) find that the path loss exponent  $n_p$  follows normal distribution based on 300,000 frequency response profiles measured in 23 homes. For an outdoor environment, the work of Walden and Rowsell (2005) also shows that the distribution of  $n_p$  over the range 100 m to 2 km appears to be Gaussian in shape.

Figure 22 is a comparison of localization variance of six localization algorithms: range-free MDS-MAP(C) (MDS-HOP in the figure's legend), range-aware MDS-MAP(C) (MDS in the figure's legend), range-free APS (APS-HOP), range-aware APS, RangeQ-MDS, and RangeQ-APS. The last one uses APS for localization along with the RangeQ scheme. The Cramer-Rao Lower Bound of the localization variance is also plotted. The figure shows that range-aware MDS-MAP(C) gives the best localization precision among the six algorithms, and RangeQ-MDS performs better than the range-free versions of APS and MDS-MAP(C). MDS-MAP(C) is also very close to CRB.

## CONCLUSION

We presented four MDS-based approaches for localization in wireless sensor networks, namely: MD-MAP(C), MDS-MAP(P), MDS-Hybrid, and RangeQ-MDS. We considered the localization problem under two difference scenarios: range-free and range-aware. In the first one, only proximity information is available to a sensor node, i.e., neighboring nodes. In the second scenario, distance measurements is assumed to be available between sensor nodes. MDS-MAP(C) and MDS-MAP(P) work well with mere connectivity information. It can also incorporate distance information when it is available. The strength of the MDS-MAP methods is that they can be used when there are few or no anchor nodes. Previous methods often require well-placed anchors to work well. Extensive simulations using various network topologies and different levels of ranging error show that the MDS-MAP methods are effective and surpasses previous methods.

Because MDS-MAP methods are expensive in terms of computational cost, we proposed an approach to relative localization referred to as MDS-Hybrid. This approach tries to combine the advantages of absolute and relative localization methods. It starts by selecting a number of reference nodes in the network based on some criterion. Then, MDS-MAP(C) is used to relatively localize the reference nodes.

Figure 21. Performance comparison of RangeQ-MDS, MDS, and APS

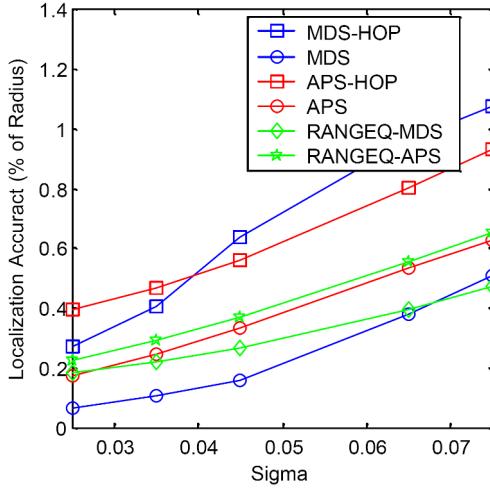
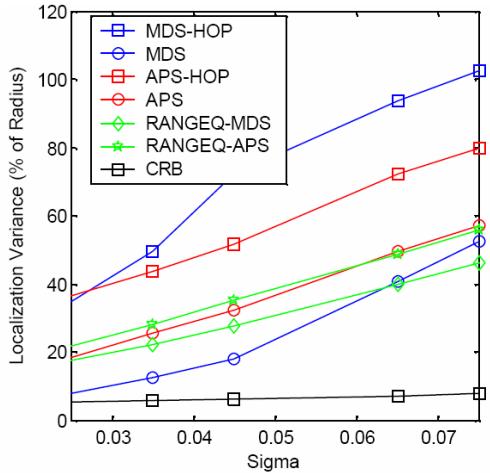


Figure 22. Comparison between 6 localization algorithms along with the Cramer-Rao lower bound on the localization variance



These nodes are used as anchors for an absolute localization algorithm to localize the rest of the network. We select the APS method for the absolute localization part. Simulation results show that MDS-Hybrid helps in choosing the required point of accuracy and cost. Moreover, it achieves a better performance than using MDS-MAP(C) in order to localize the whole network.

We presented the RangeQ-MDS localization algorithm. It is based on the RangeQ method used for RSSI quantization. While requiring only local PRI measurements, the partially range-aware RangeQ-MDS localization algorithm is found to be effective. In addition to being distributed, it can improve localization accuracy of previous range-free methods by up to 50%. It performs better than both the previous range-free and range-aware algorithms when the range error is between 15% and 35% of the radio range. Simulation results show the effectiveness of RangeQ-MDS on range estimation and localization. We analyzed the variance of localization errors using Cramer-Rao Lower Bound and have

studied the effects of network parameters on localization accuracy with considering errors caused by the variance of environment factor  $n_p$ .

## REFERENCES

- Ahmed, A., Shi, H., & Shang, Y. (2005). SHARP: A new approach to relative localization in wireless sensor networks. In IEEE, *Proc. of the Distributed Computing Systems Workshops* (pp. 892-898). Columbus, OH.
- Borg, I., & Groenen, P. (1997). *Modern Multidimensional Scaling, Theory and Applications*. New York, NY: Springer-Verlag.
- Chu, M., Haussecker, H., & Zhao, F. (2002). Scalable information-driven sensor querying and routing for ad hoc heterogeneous sensor networks. *Int. Journal on High Performance Computing Applications*, 16(3), 90-110.
- Doherty, L., El Ghaoui, L., & Pister, K. (2001). Convex position estimation in wireless sensor networks. In IEEE, *Proc. of INFOCOM* (pp. 1655-1663). Anchorage, AK.
- Ghassemzadeh, S.S., Jana, R., Rice, C.W., Turin, W., & Tarokh, V. (2002). A statistical path loss model for in-home UWB channels. In IEEE, *Proc. of the Conference on Ultra Wideband Systems and Technologies* (pp. 59–64).
- Glunt, W., Hayden, T. L., & Raydan, M. (1993). Molecular conformation from distance matrices. *J. Computational Chemistry*, 14, 114-120.
- Gower, J. (1966). Some distance properties of latent root and vector methods used in multivariate analysis. *Biometrika*, 53(3,4), 325-338.
- Intanagonwiwat, C., Govindan, R., & Estrin, D. (2000). Directed diffusion: A scalable and robust communication paradigm for sensor networks. In ACM, *Proc. of the 6th Int'l Conf. on Mobile Computing and Networks (MobiCom)* (pp. 56-67). Boston, MA.
- Johnson, D. B. & Maltz, D. B. (1996). Dynamic source routing in ad hoc wireless networks. In T. Imielinski and H. Korth (Ed.), *Mobile Computing* (pp. 153-181). The Netherlands: Kluwer Academic Publishers.
- Karp, B. & Kung, H. T. (2000). GPSR: Greedy perimeter stateless routing for wireless networks. In ACM, *Proc. of the 6th Int'l Conf. on Mobile Computing and Networks (MobiCom)* (pp. 243-254). Boston, MA.
- Langendoen, K., & Reijers, N. (2003). Distributed localization in wireless sensor networks: a quantitative comparison. *Networks: The International Journal of Computer and Telecommunications Networking*, 43(4), 499-518.
- Li, X., Shi, H., & Shang, Y. (2004). A Partial-Range-Aware Localization Algorithm for Ad-hoc Wireless Sensor Networks. In IEEE, *Proc. of the 29th Int. Conf. on Local Computer Networks (LCN)* (pp. 77-83). Tampa, FL.

- Li, X., Shi, H., & Shang, Y. (2006). Sensor network localisation based on sorted RSSI quantisation. *Int. Journal of Ad Hoc and Ubiquitous Computing (IJAHUC)*, 1(4), 222-229.
- Niculescu, D. & Nath, B. (2001). Ad-hoc positioning system. In IEEE, *Proc. of Global Telecommunications Conference (Globecom)* (pp. 2926-2931). San Antonio, TX.
- Patwari, N., & Hero III, A.O. (2003). Using proximity and quantized RSS for sensor localization in wireless networks. In IEEE/ACM, *Proc. of the 2nd International Workshop on Wireless Sensor Networks and Applications (WSNA)* (pp. 20-29). San Diego, CA.
- Priyantha, N. B., Balakrishnan, H., Demaine, E., & Teller, S. (2003). *Anchor-free distributed localization in sensor networks* (Tech. rep. No. 892). MIT Laboratory for Computer Science.
- Rao, A., Ratnasamy, S., Papadimitriou, C., Shenker, S., & Stoica, I. (2003). Geographical routing without location information. In ACM, *Proc. of the 9<sup>th</sup> Int. Conf. on Mobil Computing and Networking (MobiCom)* (pp. 96-108). San Diego, CA.
- Rappaport, T. S. (2002). *Wireless Communications: Principles and Practice*. NJ: Prentice Hall.
- Royer, E. & Toh, C. (1999). A review of current routing protocols for ad hoc mobile wireless networks. *IEEE Personal Communications*, 6(2), 46-55.
- Savarese, C., Rabaey, J., & Beutel, J. (2001). Locationing in distributed ad-hoc wireless sensor networks. In IEEE, *Proceedings of ICASSP* (pp. 2037-2040). Salt Lake city, UT.
- Savarese, C., Rabaey, J., & Langendoen, K. (2002). Robust positioning algorithm for distributed ad-hoc wireless sensor networks. In USENIX Technical Annual Conf. (pp. 317-328). Monterey, CA.
- Savvides, A., Park, H., & Srivastava, M. (2002). The bits and flops of the n-hop multilateration primitive for node localization problems. In ACM, *1st Int'l Workshop on Wireless Sensor Networks and Applications (WSNA'02)* (pp. 112-121). Atlanta, GA.
- Seidel, S.Y., & Rappaport, T. S. (1992). 914 MHz path loss prediction models for indoor wireless communications in multifloored buildings , *IEEE Transactions on Antennas and Propagation*, 40(2), 207-217.
- Shang, Y., Rumli, W., Zhang, Y., & Fromherz, M. (2004). Localization from connectivity in sensor networks. *IEEE Transactions on Parallel and Distributed Systems*, 15(11), 961-974.
- Shang, Y., Shi, H., & Ahmed, A. (2004). Performance study of localization methods for ad-hoc sensor networks. In IEEE, *Proceedings of the 1st IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS'04)* (pp. 184-193). Fort Lauderdale, FL.
- Shepard, R. N. (1962). Analysis of proximities: Multidimensional scaling with an unknown distance function i & ii. *Psychometrika*, 27, 125-140, 219-246.
- Shi, H., Li, X., Shang, Y., & Ma, D. (2005). Cramer-Rao Bound Analysis of Quantized RSSI Based Localization in Wireless Sensor Networks. In IEEE, *Proc. of the 11th Int. Conf. on Parallel and Distributed Systems - Workshops (ICPADS'05)* (pp. 32-36). Fukuoka, Japan.
- Tenenbaum, J., de Silva, V., & Langford, J. (2000). A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500), 2319-2323.

- Torgerson, W. S. (1965). Multidimensional scaling of similarity. *Psychometrika*, 30, 379–393.
- Torgerson, W. S. (1952). Multidimensional scaling: I. Theory and method. *Psychometrika*, 17, 401–419.
- Walden, M. C., & Rowsell, F. J. (2005). Urban propagation measurements and statistical path loss model at 3.5 GHz. In IEEE, *Proc. of the Antennas and Propagation Society International Symposium* (pp. 363-366).
- Whitehouse, K., & Culler, D. (2002). Calibration as parameter estimation in sensor networks. In ACM, *Proc. of the Int'l Workshop on Wireless Sensor Networks and Applications* (pp. 59-67). Atlanta, GA.
- Yu, Y., Govindan, R., & Estrin, D. (2001). *Geographical and energy aware routing: a recursive data dissemination protocol for wireless sensor networks* (Tech. Rep. No. ucla/csd-tr-01-0023). Los Angeles, CA: University of California-Los Angeles.