

# Position Estimation for Wireless Sensor Networks

K.-F. Simon Wong   Ivor W. Tsang   Victor Cheung   S.-H. Gary Chan   James T. Kwok

Department of Computer Science

The Hong Kong University of Science and Technology

Clear Water Bay, Kowloon, Hong Kong SAR

**Abstract**—In wireless sensor networks, estimating nodal positions is important for routing efficiency and location-based services. Traditional techniques based on precise measurements are often expensive and power-inefficient, while approaches based on landmarks often require bandwidth-inefficient flooding and hence are not scalable for large networks. In this paper, we propose and investigate a cost-effective and distributed algorithm to accurately estimate nodal positions for wireless sensor networks. In our algorithm, a node only needs to identify and exchange information with a certain number of neighbors (around 30) in its proximity in order to estimate its *relative* nodal position accurately. For location-identification, only a small number of nodes (around 10) are needed to have additional GPS capabilities to accurately estimate the *absolute* position of *every* node in the network. Our algorithm is shown to have fast convergence with low estimation error, even for large networks.

**Index Terms**—position estimation, wireless sensors network, position-based routing, location identification

## I. INTRODUCTION

In recent years, there has been an increasing interest in wireless sensor networks in academic, industrial, and commercial sectors. Unlike ad-hoc mobile networks, such networks are usually assumed to have a high density of nodes with lower computational power. It is often useful to know the relative or absolute nodal positions in order to improve the quality of position-related services provided. Two typical examples are: *Position-based routing* in which relative nodal positions are used for correct and efficient route estimation; *Location-based services* in which absolute nodal positions are used to provide location-specific services

In this paper, we propose and investigate a cost-effective and distributed algorithm to estimate nodal positions in wireless sensor networks. In our algorithm, each node has a certain maximum transmission power. By controlling its power in a *quantized* manner, a node only needs to discover its one-hop neighbors at discrete distances away<sup>1</sup>. Based on this information, each node computes its own position in the network in an autonomous manner, and only needs to exchange the information with a number of its neighbors. Our algorithm starts with a certain number (1 to 10) of “bootstrap” nodes and the position information propagates in the network like ripples until all nodes are able to estimate their own positions.

This work was supported, in part, by the Areas of Excellence (AoE) Scheme on Information Technology funded by the University Grant Council in Hong Kong (AoE/E-01/99), and by the Central Allocation Grant by the University Grant Council in Hong Kong (CA03/04.EG01).

<sup>1</sup>Our scheme differs from the technique of Received Signal Strength (RSS), since wireless nodes do not require to measure the strength of the received RF signal. Thus no measurement device on the signal strength is required.

In our algorithm, we modify a machine learning technique called ISOMAP, which is originally proposed for nonlinear dimensionality reduction [13]. In particular, we employ the multidimensional scaling technique (MDS [3]), which is generally used for data visualization and feature extraction. With some modifications to the techniques, we achieve an algorithm which is robust against measurement errors and yet computationally unintensive and bandwidth efficient.

The contributions of the paper are as follows:

- We propose a distributed algorithm where only local information is exchanged between a node and its neighbors, and the convergence time is quite fast.
- We illustrate the usefulness and accuracy of the algorithm by applying it to position-based routing and location identification. Regarding to position-based routing, our algorithm does not require any GPS devices to estimate relative nodal positions. Regarding to location identification, our algorithm requires only a few (around 10) GPS devices to estimate the global positions of all the nodes. Remarkably, the number of GPS devices needed does not depend on the size of the network. Simulation results confirm the effectiveness of our algorithm.

The paper is organized as follows. In Section II, we briefly discuss the related work. We then present our position estimation algorithm in Section III. We discuss how our algorithm is extended to position-based routing and location-based service in Section IV. Illustrative simulation results are then presented in Section V, followed by the conclusion in Section VI.

## II. RELATED WORK

Many algorithms have been proposed to find nodal positions. In general, these algorithms may be divided into two categories:

- Approaches based on precise measurements  
These approaches include techniques making use of directional antennae or certain measuring devices, examples are RADAR [1], APS [7], VORBA [9]. They are often expensive due to the specialized hardware involved in measuring nodal distances or angle of arrival. A node may also need to consume much power to communicate with distant nodes. Also, in the presence of obstacles or in an enclosed environment, some of these techniques may not work properly. Our idea does not require expensive devices and communication with distant nodes. It can also work in an enclosed environment.

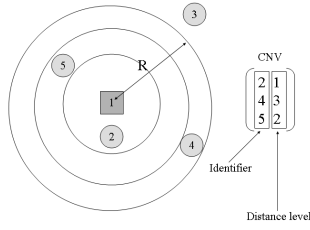


Fig. 1. Quantized Distance and Close-Neighbor Vector (CNV).

- Landmark-based approaches

In such systems, several landmarks are introduced, their position information are transmitted and used to find nodal positions. These landmarks are usually assumed to have GPS and are more powerful in terms of transmission range or computing power. Examples are (APIT)[4] and [8]. In general, all the aforementioned approaches require many powerful landmarks and the number of landmarks required usually increases with the population size. In order to distribute location information to all the nodes, bandwidth-inefficient flooding from landmarks are often required. Our algorithm does not require any landmark or fixed infrastructure in order to estimate the relative nodal positions. To obtain the absolute (global) position of a node, our algorithm requires only a few nodes with GPS capability (typically 5 to 10, and independent of the size of the population).

Using MDS to estimate nodal positions is first studied in [12]. This work studied a centralized algorithm that requires global information obtained by bandwidth inefficient flooding. The system also requires global re-computation at the joining of new nodes or position updating for existing nodes. On the other hand, our distributed algorithm only requires local re-computation in these cases.

Several work has been proposed recently to use MDS in a distributed manner [5], [11]. [5] requires a node (starting anchor) to initialize a flooding to the whole network for the positioning process to begin. After computing the local maps, both of them have to share and merge the maps together to create a global map. In our work, however, an initial flooding to the whole network is not required, and the nodes do not have to share their own local map with each other, they only need to send their own updated coordinates to their neighbors.

### III. POSITION ESTIMATION ALGORITHM

In this section, we present the detailed operations of the position estimation algorithm. First, we discuss how to determine the (quantized) distances between a node and its neighbors. We then discuss how to compute the relative nodal positions given the information of the quantized distances, followed by a summary of the distributed algorithm for position estimation. Finally we provide a complexity analysis for our system.

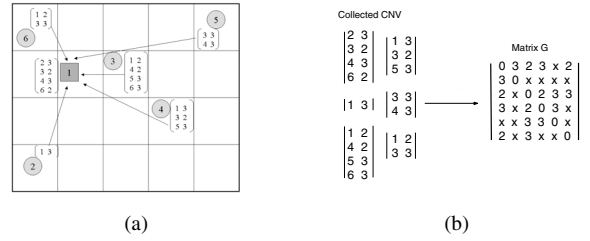


Fig. 2. Compilation of the distance matrix  $G$ . a) Sending of CNV to a central server (Node 1). b) Compilation of the distance matrix  $G$ .

#### A. Quantized Distance Determination

In the system, every node has a unique identifier (ID) and has a maximum transmission range  $R$ . We assume that the node is able to adjust power so that it can communicate with its neighbors at discrete distances away<sup>2</sup>. The idea is shown in Figure 1, where the square node with ID 1 is to identify its one-hop neighbors. First, it sends out beacons with increasing discrete ranges shown as concentric circles. In this example, we have 3 quantization levels. The beacon message contains the level and polls the neighbors for reply. Based on the reply from the neighbors, Node 1 then compiles a close-neighbors vector (CNV), which indicates the distance levels between itself and its neighbors. The CNV for Node 1 is shown on the right of the figure. Since Node 3 is out of the range of Node 1, it cannot be detected and hence is not included. This CNV will be used to compute nodal positions as discussed in the following sections.

#### B. The Estimation Algorithm

In the algorithm, a node first collects a number of CNVs, and compiles a distance matrix  $G$ . Figure 2 illustrates the idea. The entry  $[d_{ij}]$  is the quantized distance between nodes  $i$  and  $j$ . Nodes which are invisible to each other are denoted by a cross “x” as the matrix entry. These entries are then estimated by some shortest path algorithms, such as Dijkstra’s algorithm.

After resolving all the entries in the matrix  $G$ , multidimensional scaling (MDS) is used to find the best node coordinates in the two-dimensional Euclidean space that satisfies all these pair wise distance constraints in  $G$  [3]. Note that the positions obtained from MDS differ from the real positions by an arbitrary translation, rotation and reflection, as all these transformations would not alter the same distance matrix.

#### C. Distributed Algorithm

We assume there is a bootstrap node in the network which triggers the computational processes. The bootstrap node acts as the origin and computes a small set ( $C$ ) of nodal coordinates using the CNVs from  $C$  of its closest nodes. Based on these CNVs, it computes its own coordinates locally. Clearly, the estimation results would be more accurate if  $C$  is large. However, the larger  $C$  is, the larger is the bandwidth consumption.

<sup>2</sup>For simplicity, we assume every node has the same transmission range  $R$  and the same levels of quantized distances.

Hence, there is a tradeoff between accuracy and bandwidth. From our simulations, we find that  $C = 30$  gives very good results. The detailed procedures of our distributed algorithm are summarized as follows:

- **Bootstrap node**  
At the beginning, the bootstrap node requests CNVs from the  $C$  closest neighbors (1 broadcast in max. power range  $R$  of a sensor and  $C$  broadcasts in  $R$  for reply). Then it performs the estimation algorithm with  $C + 1$  nodes (including itself). Lastly, it sends the computed coordinate of its  $C$  neighbors themselves individually ( $C$  broadcasts/multicast in  $R$ ).
- **General node**  
Upon receiving its coordinate from the bootstrap node, the node updates its current coordinates and send the new coordinates to its neighbors (1 broadcast in  $R$ ). Upon receiving a new coordinate from its neighbors, increment the variable “RECEIVED” by 1. If RECEIVED is equal to  $C$ , it performs the estimation algorithm and sends the new computed coordinates of itself only to its neighbors (1 broadcast in  $R$ ), reset RECEIVED. In case there are fewer than  $C$  nodes that are one hop away (within  $R$ ) at this time frame (iteration), it sleeps and increments RECEIVED by 1. If RECEIVED is still less than  $C$  after  $C - L$  iterations, it performs  $N$ -hop broadcasting.  $N$  can be estimated by  $\lceil \sqrt{C/K} \rceil$ , where  $K$  is the number of the node’s one-hop neighbors.

Note that, the  $L$  coordinates from the neighbors generalize the translation, rotation and reflection, and thus the new local coordinate will have the same translation, rotation and reflection as the origin. Although a large  $L$  leads to better estimation results, it takes a longer convergence time. Our results show that  $L = 12$  (for  $C = 30$ ) is a reasonably good choice. The processes are repeated and the nodes in the parameter finally can have their coordinates. Also, in practice,  $N$  is usually not high. And since it is only performed in a sparse network and is limited in scope (not flooding to the boundary of the network), such  $N$ -hop data broadcasting does not consume too much bandwidth.

Our algorithm can be extended to mobile network by periodically performing the triggering process in the bootstrap node (the period can be set according to the mobility) and the triggering process stimulates the re-computation of nodal position.

#### D. Complexity Analysis

Here we analyze the computational complexity of the estimation algorithm, which is the core part of our system.

As mentioned the estimation algorithm starts with a Dijkstra’s algorithm to complete the distance matrix. For a matrix of size  $m \times m$  it takes  $O(em \log m)$  time, where  $e$  is the number of edges covered in the matrix. After that MDS is performed, with a computational complexity of  $O(dm^2)$ , where  $d$  is the number of dimension of embedding.

Since in each node the algorithm is performed with at most  $C$  nodes and the dimension of embedding is at most 3, the

overall computational complexity for each node is therefore  $O(EC \log C + C^2)$ , where  $E$  is the number of edges among the nodes.

## IV. APPLICATIONS

In this section, we discuss how our system can be applied to position-based routing and location-based services.

### A. Position-Based Routing

One possible application of our system is position-based routing [6]. In position-based routing, nodes deliver packets based on the locations of the source and destination. In general, the aim of these systems is to reduce the fraction of failure connections and the number of hop-counts for routing. Here, we employ a simple greedy algorithm as described in [10] to deliver packets and study the performance of our scheme compared with the use of true position information.

Now consider the presence of obstacles in the environment. Two nodes are invisible to each other if their line of sight is blocked by an obstacle. We determine the connectivity (in both directions) for each pair of nodes  $i, j$ . If packets cannot be routed from  $i$  to  $j$  due to obstacles, we say that the connection fails; otherwise, we say that node  $i$  is connected to node  $j$ . Note that, node  $i$  is connected to node  $j$  does not necessary imply node  $j$  is connected to node  $i$ .

### B. Location-based Services

Another possible application of our system is location identification. Our system can be applied to global positions estimation like GPS. However, our system discussed before is for estimating relative positions, and the resulting coordinates may randomly rotate or reflect at the bootstrap node. Landmarks are therefore required to fix the rotation/reflection. They are not required to flood the network or to have high computational power. Also, we assume that there are only a few landmarks, say 5 or 10, in the whole network, and the required number of them is independent of the population size.

Suppose that there are  $n$  landmarks and each of them is a bootstrap node. So for each node it will have  $n$  relative coordinates  $[l_{x1}, l_{x2}, \dots, l_{xn}]$  corresponding to each landmark. The global coordinates  $[g_1, g_2, \dots, g_n]$  of each landmark can be attached in the CNVs where no additional information exchanging is required. Then each node can compute its global coordinates  $g_x$  by minimizing the following:

$$\sum_{i=1}^n (\|g_x - g_i\| - \|l_{xi}\|)^2. \quad (1)$$

## V. ILLUSTRATIVE SIMULATION RESULTS

We present the simulation results of our system in this section. We assume each node has a restricted transmission range ( $R$ ) of 20 units. We randomly generate the node’s coordinates in an enclosed  $100 \times 100$  square area. The *node density* is the average number of peers inside the maximum transmission range  $R$  of a node. We generate 30 independent iterations and take the average for each data point.

TABLE I

DISTRIBUTION OF BROADCASTING LEVELS AT DIFFERENT DENSITIES.

Node Density	20	30	40
1-Hop Broadcasting	0.0%	28.0%	73.7%
2-Hop Broadcasting	85.6%	69.9%	26.3%
3-Hop Broadcasting	14.4%	2.1%	0.0%

We compare our system with the centroid algorithm in [2] in the sense that it is distributed and has no information flooding. We assume that there are 20 landmarks for the centroid algorithm (and no landmark for our system) if we do not specify the number of landmarks.

### A. General Characteristics

In this part, we examine the general characteristics of our system. We use the distance error in terms of range ( $R$ ) as the performance metric. For each pair of nodes, we define the distance error as the difference between the estimated distance and the true distance, normalized to  $R$ . According to the simulation results in [4], the routing performance is highly related to distance error (in terms of  $R$ ). In general, routing performance is significantly decreased when the error is above  $0.4R$ .

We first examine the effect of the number of quantization levels. Figure 3(a) shows the distance error versus the number of quantization levels at a node density of 40. As expected, the distance error in our system decreases with increasing number of quantization levels, as more information on the distances between nodes is available. Generally, the rate of decrease falls with the number of quantization levels. Although for the centroid algorithm the distance error is independent of the number of quantization levels, our system achieves substantially better performance. In the rest of our simulation, we set the number of quantization levels to be 20.

We next examine how fast our system converges. Figure 3(b) shows the cumulative fraction of converged nodes versus the number of iterations, at different node densities. We find that higher node density converges faster, since every node can identify  $L$  of the users faster. In general, the system needs about 30 iterations for 90% of the nodes to converge.

Since each node needs to obtain enough CNVs, different levels of data broadcasting are required to communicate with its  $C$  neighbors. Table I shows the distribution of broadcasting levels at different node densities. In general, most nodes only need to perform 1-hop broadcasting at a density of 40. For sparse networks, most nodes perform 2-hop broadcasting.

### B. Position-Based Routing

In this section, we consider the routing performance of our system. We insert 20 horizontal / vertical obstacles into the graph with the length in terms of transmission range (20 units) in the  $100 \times 100$  square area. We assume a node drops a packet only when it cannot find any neighbor closer to the destination.

Figure 3(c) shows the fraction of connection failures versus the node density. The fraction of connection failures decreases with increasing node densities, since there are more choices for

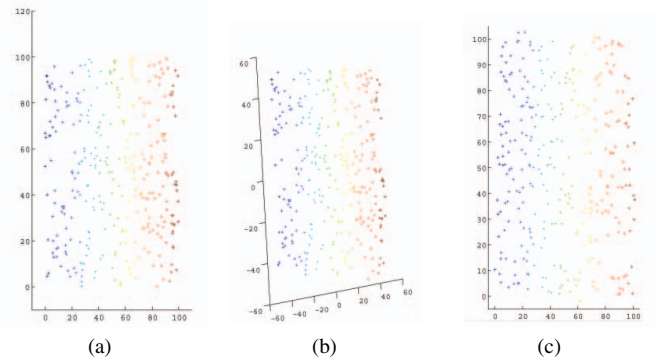


Fig. 4. Position estimation results. a) True position. b) Relative position estimation. c) Global position estimation.

routing when the node density is high and thus lower failure rate. In the simulation, the system with our estimated positions works better than the one using true geometric positions.

Figure 3(d) shows the cumulative distributions of the hop counts for our system and the one with true position information. We only consider the successful connections here, so both curves go to 1 on the right. We find that 90% of the connections have a hop count of 5. Again, the performance of our system is comparable with the one using true position information.

### C. Location-based Services

In this section, we study the performance of our system when being used for location-based services. Figure 4(a) shows the true positions of the nodes. Each cross in the figure represents a wireless node. We randomly generate the nodes in a  $100 \times 100$  square area with node density 40. Figure 4(b) shows the relative positions estimated. We can observe that in this example, the axes are rotated around  $20^\circ$  anti-clockwise and the origin is shifted. Figure 4(c) shows the result after landmark adjustment as discussed in section IV-B.

Figure 3(e) shows the relative distance error defined as:

$$\gamma = \left\| \frac{\text{measured distance} - \text{true distance}}{\text{true distance}} \right\|. \quad (2)$$

We compare our algorithm (with landmark adjustment) with the centroid algorithm. We assume there are 10 landmarks for our system. The relative error first decreases and then increases. Generally, our system gives better estimation for higher density, since more information (in the CNVs) can be exchanged among the  $C$  neighbors. For close pairs, small estimation error can lead to high relative error, and thus results in higher relative errors. We achieve 2% - 17% relative distance error for different densities. For centroid algorithm, the system generally achieves 28% - 30% relative distance error for different densities. Although our system is more sensitive to node density, it is generally better than the centroid algorithm.

Lastly, we examine the effect of varying the number of landmarks. Figure 3(f) shows the relative distance errors versus the number of landmarks. The estimation errors reduces as

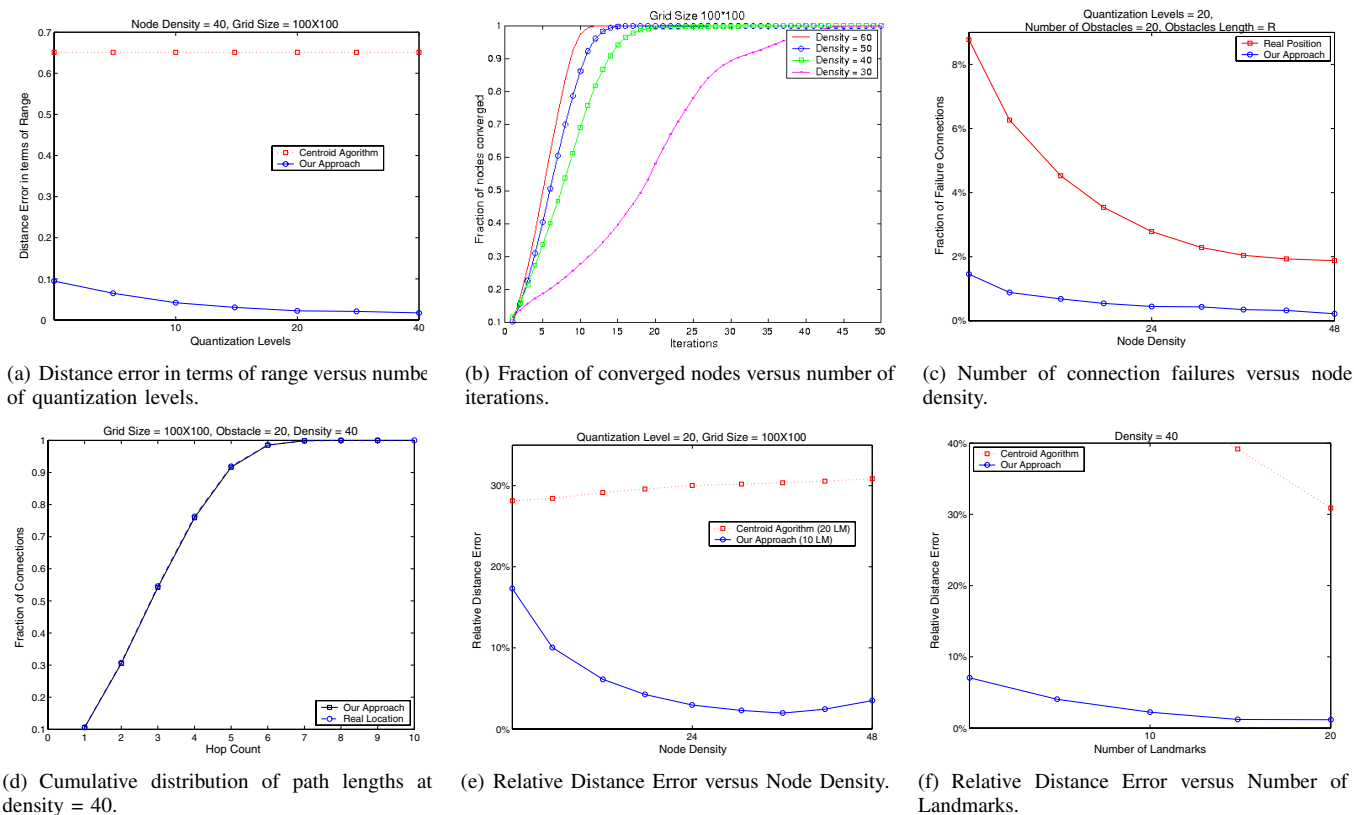


Fig. 3. Simulation Results.

the number of landmark increases for both systems. However, our system is less sensitive to the changes and 10 landmarks already give very accurate estimation.

## VI. CONCLUSION

In this paper, we propose and investigate a position estimation algorithm for estimating nodal positions in wireless sensor networks. We have considered two applications of our system, namely, position-based routing and location identification.

In our scheme, we first obtain the quantized distance information of neighboring nodes by power controlling. After that, Dijkstra's algorithm and MDS are performed at each node in a distributed manner. Since our distributed algorithm depends neither on specific hardware nor bandwidth-consuming data flooding, it is both cost and bandwidth effective.

We have described the details of our system, and showed via simulation that it achieves better results than the system using true position information with a simple geographic routing algorithm. For location identification, we accurately calculated nodal positions with less than 17% relative distance error.

## REFERENCES

- [1] P. Bahl and V. N. Padmanabhan, "Radar: An in-building RF-based user location and tracking system," in *Proceedings of IEEE INFOCOM 2000*, Israel, March 2000, pp. 775–784.
- [2] N. Bulusu, J. Heidemann, and D. Estrin, "Gps-less low cost outdoor localization for very small devices," *IEEE Personal Communications Magazine*, vol. 7, no. 5, pp. 28–34, October 2000. [Online]. Available: <http://lecs.cs.ucla.edu/bulusu/papers/Bulusu00a.html>
- [3] T. Cox and M. Cox, *Multidimensional Scaling*. Chapman & Hall, London, 1994.
- [4] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. Abdelzaher, "Range-free localization schemes for large scale sensor networks," in *Proceedings of the 9th ACM MOBICOM*. New York, NY, USA: ACM Press, 2003, pp. 81–95.
- [5] X. Ji and H. Zha, "Sensor positioning in wireless ad-hoc sensor networks using multidimensional scaling," in *Proceedings of IEEE INFOCOM 2004*, Hong Kong, March 2004, pp. 2652–2661.
- [6] M. Mauve, J. Widmer, and H. Hartenstein, "A survey on position-based routing in mobile ad hoc networks," *IEEE Network Magazine*, vol. 15, no. 6, pp. 30–39, November 2001.
- [7] D. Niculescu and B. Nath, "Ad hoc positioning system (APS) using AoA," in *Proceedings of IEEE INFOCOM 2003*, San Francisco, USA, March 2003, pp. 1734–1743.
- [8] —, "DV based positioning in ad hoc networks," *Telecommunication Systems*, no. 1-4, pp. 267–280, January-April 2003.
- [9] —, "VOR base stations for indoor 802.11 positioning," in *Proceedings of the 10th ACM MOBICOM*. New York, NY, USA: ACM Press, 2004, pp. 58–69.
- [10] A. Rao, C. Papadimitriou, S. Shenker, and I. Stoica, "Geographic routing without location information," in *Proceedings of the 9th ACM MOBICOM*. New York, NY, USA: ACM Press, 2003, pp. 96–108.
- [11] Y. Shang and W. Ruml, "Improved MDS-based localization," in *Proceedings of IEEE INFOCOM 2004*, Hong Kong, March 2004, pp. 2640–2651.
- [12] Y. Shang, W. Ruml, Y. Zhang, and M. P. J. Fromherz, "Localization from mere connectivity," in *Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing*. New York, NY, USA: ACM Press, 2003, pp. 201–212.
- [13] J. Tenenbaum, V. de Silva, and J. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.