A Comparative Review of Connectivity-Based Wireless Sensor Localization Techniques*

Charles J. Zinsmeyer and Turgay Korkmaz†
The University of Texas at San Antonio
San Antonio, Texas, U.S.A.
{czinsmey, korkmaz}@cs.utsa.edu

Abstract

One of the key problems in Wireless Sensor Networks (WSNs) is how to identify the location of each node. The research community has done considerable work on this localization problem and developed various techniques ranging from hardware-based ones requiring extra units (e.g., GPS, directional antennas) to software-based ones requiring only the connectivity information. In this paper, we present an overview of the key localization techniques and then focus on the range-free or connectivity based approaches. After the overview, we implement and compare the performance of connectivity-based techniques using simulations. We show these techniques vary considerably in their ability to localize a wireless sensor network.

Key Words: Wireless Sensor Network, Localization, Multidimensional Scaling, Force Directed, Combinatorial Delaunay Complex

1 Introduction

Intelligent wireless sensing devices are becoming ubiquitous and are being applied in many applications ranging from environmental monitoring to animal tracking and numerous other applications. The devices themselves are usually small and inexpensive. They typically have limited computing resources and limited wireless range. However, to collect and process various types of sensory data over large areas, these devices are often assembled into vast networks called Wireless Sensor Networks (WSNs), where the devices (nodes) communicate with a base station through other nodes.

To efficiently perform various tasks (e.g., routing, clustering, data dissemination etc.) in WSNs, the nodes and/or the base station often need to know the exact or relative locations of the other. To solve the localization problem in WSNs, the research community has done significant amount of work under various assumptions [9]. Some of the early techniques that are discussed in [4] includes the following three techniques: (i) Triangulation, (ii) Proximity, and (iii) Scene analysis. Triangulation uses measures between an unknown point and at least three known points to determine the location of the unknown point. Proximity attempts to measure the nearness to the known set of points. Scene analysis examines a view from a particular vantage point.

One of the key issues in all of the above techniques is how to measure the distance between the nodes. Most of the early work on localization was focused on using physical measurements such as time of flight (TOF) or angle of arrival (AOA) to triangulate the position of a node. These techniques require extra hardware units (e.g., GPS, directional antennas) for physical range measurement. Therefore, we classify such techniques as hardware-based (or range-based) ones. Hardware-based approaches are not always the best solution for locating simple and power limited wireless sensing devices. Design solutions

Journal of Internet Services and Information Security (JISIS), volume: 2, number: 1/2, pp. 59-72
*This work is supported by DoD Infrastructure Support Program for HBCU/MI, Grant: 54477-CI-ISP (UNCLASSIFIED).
†Corresponding author: Dept. of Computer Science, The University of Texas at San Antonio, One UTSA Circle, San Antonio, TX 78249, USA, Tel: +1-210-458-7346

often try to keep node costs, size, and power consumption to a minimum. Also, the range measurements such as RSSI are often not accurate and are significantly affected by environmental factors. GPS is becoming ubiquitous, but would not function for the nodes placed indoors. Clearly, there is a need for new localization techniques that are not limited by these constraints.

In response to this, the research community has been investigating and developing new techniques that utilize connectivity information. We classify such techniques as connectivity-based or range-free due to the fact that all of the work is done in software by processing the connectivity information without requiring physical range measurements. These software-based techniques determine the logical position of nodes relative to the other nodes. The calculated position resulting from these techniques is not in a physical coordinate system such as latitude/longitude as in GPS, but a logical coordinate system perhaps an $x - y$ coordinate system based on hop counts between nodes. The calculated logical positions will often be a rotation, translation and/or flip from a physical position. These positions may be mapped to a physical location when anchor nodes are provided. In many cases though (e.g., routing, clustering), it may not be necessary to determine the exact physical position of a node. In such cases, only computing relative positions is necessary.

In this paper, we first present a review of the existing localization techniques with a special focus on software-based (connectivity-based) techniques. We then select and implement three of the proposed connectivity-based localization techniques and compare their performances using simulations. Our main goal here is to better understand and compare the accuracy and complexity of the proposed techniques. In general, we observed that these techniques only offer a rough localization of the nodes with position errors varying from 7 percent to 78 percent from the original positions. In all cases the estimated positions were requiring at least a translation and rotation to get into a final position. To facilitate comparison, the positions were normalized to be in the range [0, 1].

The rest of this paper is organized as follows. In Section 2, we review the localization techniques. In Section 3, we present our simulation results. In Section 4 we analyze our results and compare the connectivity-based localization techniques. Finally, we conclude this paper and provide some directions for future research in Section 5.

2 WSN Localization Techniques

In essence, we divide existing localization techniques into two classes: hardware-based (range-based) ones requiring extra units (e.g., GPS, directional antennas) to measure ranges, and software-based (connectivity-based, range-free) ones processing only the connectivity information obtained from the readily available radios without any other physical range measurement. We now review the existing techniques under these two classes with the special emphasis on connectivity-based ones.

2.1 Hardware-based Techniques

In this subsection, we review four localization techniques utilizing various hardware units to measure the range: Time of Flight based approach, Angle Based Approach, Global Positioning System, and Signal Strength Based Approach.

2.1.1 Time of Flight Based Approach

Time of Flight (TOF) is used to measure the distance between two nodes by measuring the time it takes for a radio or acoustical signal to move between two nodes. By measuring the time, one can compute the distances. By measuring the distance between three or more nodes and using the known position of at least three nodes, one can use lateration to calculate the position of a fourth node in two dimensional space [5]. Same ideas can be generalized to three dimensions and used for localization with the addition of another node, as described by [5].

2.1.2 Angle Based Approach

Similarly, one can use directional antennas to measure the Angle of Arrival (AOA) of a signal from a known node through angulation. Given three or more angles, the position of a fourth node can be calculated [11]. While lateration uses distance, angulation uses angle measurements.

2.1.3 Global Positioning System

The Global Positioning System or GPS is mature, relatively inexpensive and ubiquitous. It is perhaps the most accurate mechanism for locating a device. GPS uses satellites orbiting the earth and precise measurement of timing signals sent from the satellites to determine position. Time of Arrival (TOA) methods require explicit synchronization within the locating system and uses a signal stamped with an absolute time to measure the time the signal traveled from the transmitter to the receiver. Knowing the propagation time of a signal, the distance traveled can be computed.

Time Difference of Arrival (TDOA), which is used by GPS, uses the time difference of arrival by measuring the time of arrival from three or more synchronized transmitters as received by a receiving station. For GPS to function, it must have clear access to the sky to receive the GPS signals and therefore does not function within locations such as buildings. Like the other approaches discussed so far, GPS requires additional hardware such as an antenna and a GPS chip. This may increase the cost, size and power requirements of a simple wireless sensing device.

2.1.4 Signal Strength Based Approach

Received Signal Strength Indicator (RSSI) has been proposed as alternative to TOF or AOA for making distance measurements. The power of a signal decreases at $1/d^n, n \geq 2$. Therefore, in theory, one could use the received signal strength to estimate the distance between nodes and use lateration to calculate the position of the node. But, it is well known that RSSI is not an accurate indicator of distance. Environmental factors, such as the presences of a wall and other obstructions affect the received signal strength. Transmissions from other devices also interfere with RSSI. On the positive side, RSSI is incorporated into all wireless transceivers and is readily available with no additional hardware.

2.2 Software-based Techniques

In this subsection, we focus on three localization techniques utilizing connectivity information: Multidimensional Scaling, Force Directed, and Combinatorial Delaunay Complex. These three algorithms can be implemented using centralized and distributed approaches.

A centralized approach involves sending connectivity information from the nodes through the network to a central base station where the localization algorithms are executed. If the nodes require the resulting location information, it must be forwarded back to each of the nodes in the network. Centralized algorithms suffer from high traffic cost and reliability (single point of failure).

In a distributed approach, each node is responsible for calculating its position within a network. Messages are often flooded throughout the underlying network so that each node can get connectivity information about the other nodes and calculate its own position.

In the rest of this section, we describe the centralized versions to keep the algorithmic descriptions simple. Some work has already been done to convert them to distributed algorithms [11].

2.2.1 Multidimensional Scaling Algorithm

This algorithm, like others in this section, utilizes connectivity information to derive the location of nodes in a network. Multidimensional Scaling (MDS) is a data analysis technique often used in data visualization to uncover the similarities or dissimilarities within a data set and is taken from work done in psychometrics and psychophysics. In our case the data set comprises the connectivity between neighbors in the network. If a measure of distance is available, then it can be incorporated as a weighting on the connectivity.

The authors in [10] presented an algorithm called MDS-MAP which they described as a classical metric approach based on the work in [12]. This is the simplest case of MDS and takes a matrix containing dissimilarities between pairs of items, in our case the connectivity between nodes, and computes a coordinate matrix that minimizes a loss function called strain. The result is the coordinates of the nodes in a Euclidian space. The MDS-MAP technique is summarized in the following pseudo code:

```
Compute MDS-MAP
   Begin
      Using an all pairs shortest path algorithm, estimate
         the distance between each pairs of the possible nodes
         producing a distance matrix
      Using these distances, apply classical MDS and keep the two
         largest eigenvalues and eigenvectors to build the
         relative 2D map
   End Compute MDS-MAP
```

For our work, we assumed a distance of one for a neighboring node within radio distance and did not weight the distance. Our implementation does not utilize anchor nodes. Had we utilized anchor nodes, we would have performed a third step which would use three anchor nodes to transform the relative map into an absolute map.

2.2.2 Force Directed Algorithm

This approach views the nodes of a network as physical elements, such as weights and springs. The nodes are modeled as forces pulling or pushing each other. The basic idea behind force directed localization is defined as follows [1].

> To embed a graph we replace the vertices by steel rings and replace each edge with a spring to form a mechanical system... The vertices are placed in some initial layout and let go so that spring forces on the rings move the system to a minimal energy state.

Force directed algorithms can be some of the most flexible algorithms for simple undirected graphs. They define a method for using an objective function for mapping each graph layout into a number that represents the energy of the layout. The objective function is defined such that graph layouts in which the adjacent nodes are in some pre-defined distance have lower energies.

In [3] the goal of this technique was to layout an aesthetically pleasing graph. This was later investigated in [2] as an approach to network localization. A total of five different force directed approaches were investigated in [2]: Fruchterman and Reingold (FR) Kamada-Kawai, Fruchterman-Reingold Range Algorithm, Kamada-Kawai Range Algorithm, multi-scale Kamada-Kawai Range Algorithm, and Multi-scale Dead Reckoning.

In essence, the FR algorithm [3] defines two functions: an attractive force function and repulsive force function. The attractive force function is used for adjacent nodes and the repulsive force function is used for non-adjacent nodes. In the algorithm, vertices in the graph are moved repeatedly until a low energy state is achieved. An attractive and repulsive force is computed using Equation (1).

$$\begin{aligned} f_a(d) &= d^2 d^2/k \\ f_r(d) &= -k^2/d \end{aligned} \tag{1}$$

where d is the distance between vertices and k is the empty area around a vertex. The displacement of each vertex is limited to maximum value with the maximum value decreasing with each iteration and in each iteration refinement becomes finer and finer until "low energy state" of the graph is achieved. The technique is summarized in the following pseudo code:

```
ForceDirected
   Begin
   While Refinement > low energy state
      For each Vertex v1
         For each Vertex v2
            if v1 and v2 are not the same vertex
               d = distance between v1 and v2
               Compute Vertex Displacement using fr(d)
            Endif
         End Loop
      End Loop
      For each Edge E
         d = edge distance
         Compute Edge Displacement using fa(d)
      End Loop
      For each Vertex v
         Compute New v Position Using Displacement and Refinement
      End Loop
      Reduce Refinement
   End Loop
   End ForceDirected
```

2.2.3 Combinatorial Delaunay Complex Algorithm

One of the problems faced by many of the algorithms is the ability to handle large networks with complex structures and holes. In [6] the authors proposed an approach that dealt with these problems. Their algorithm utilizes graph rigidity theory and higher order topological extraction to determine the positions of the devices. The steps involved in this algorithm are:

```
Identify Landmarks
Begin
   Compute Boundaries
   Determine Medial Axis
   Identify Landmarks
End Identify Landmarks
```

```
Compute Voronoi Diagram
Begin
   Flood for Voronoi Cells
   Build Voronoi Cells
   Identify Voronoi Cells for Landmarks
End Compute Voronoi Diagram

Compute Delaunay Complex
Begin
   Identify Witnesses
   Collect Delaunay Edges
   Construct Simplices
   Embed landmarks
End Compute Delaunay Complex

Begin Main
   Identify Landmarks
   Compute Voronoi Diagram
   Compute Delaunay Complex
   Trilaterate Remaining Nodes
End Main
```

3 Results

3.1 Input Graphs

To compare the connectivity-based techniques, we utilize different input graphs of varying structure and size. The input graphs were generated using a Hammersley sequence to produce a uniform random distribution. Several of the graphs were then modified to introduce holes in the graph to evaluate how well the approaches handle holes within a graph. The CDC algorithm was designed with this in mind. The size of the graphs was also varied from a hundred nodes to 990 nodes. The randomly generated input graphs are presented in Figure 1, Figure 2, Figure 3, Figure 4 and Figure 5. Because of the way the random locations are generated, Node 0 is always located near $(0,0)$. Table 1 presents the average degree in each input graph.

Table 1: Average Degree of Input Graphs

Graph	Average Degree
100 Node Graph	5.4
95 Node Graph, With Hole	4.9
181 Node Graph, With Hole	5.6
490 Node Graph, With Hole	9.4
990 Node Graph, With Hole	12.8

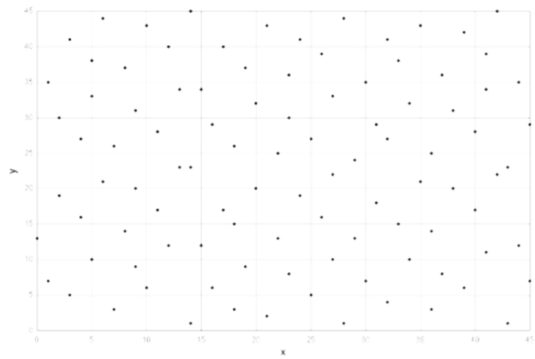

Figure 1: Input Graph 1, 100 Nodes, No Hole

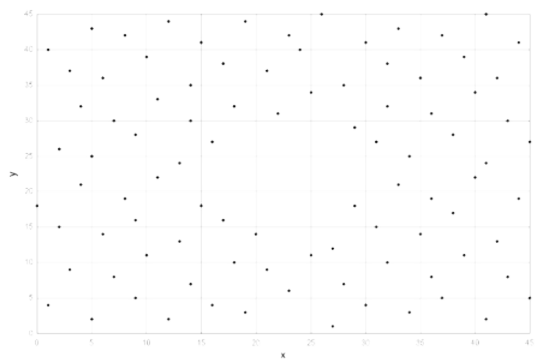

Figure 2: Input Graph 2, 95 nodes, 1 hole

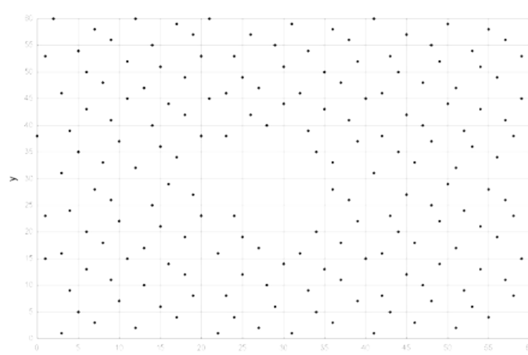

Figure 3: Input Graph 3, 181 Nodes, 1 Hole

3.2 Output Graphs

To facilitate the comparison of the resulting localized graphs with the input graphs, the node locations were normalized to be in the range [0 1] and the output graphs were flipped, rotated and translated so that the nodes roughly match the input graph. Node 0, which was located near $(0,0)$, was used to determine the rotation and translation offsets for the graph. So, Node 0 on the output graph was lined up with Node 0 on the input graph and the resulting offset was applied to the remaining nodes.

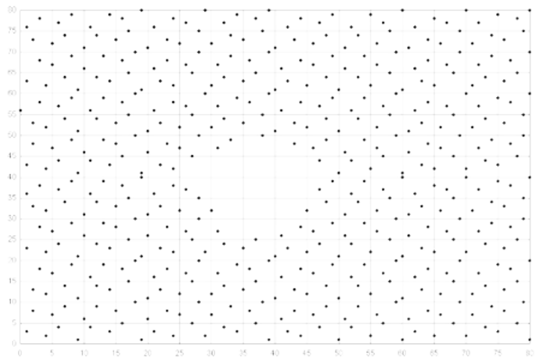

Figure 4: Input Graph 4, 490 Nodes, 1 Hole

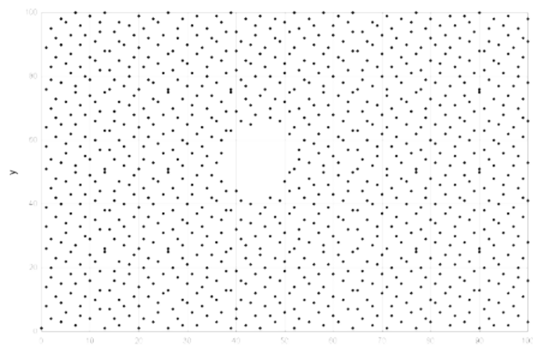

Figure 5: Input Graph 5, 990 Nodes, 1 Hole

3.3 Performance Metric

To measure how well a localization technique works, we compare the given node positions in the input graphs with the estimated node positions in the corresponding output graph, after having flipped, rotated and/or translated the output graph. To quantify the degree of error in each output graph, we use the distance between the original and computed position of every node, as shown in Equation (2).

$$Error = \sqrt{(X_{original} - X_{computed})^2 + (Y_{original} - Y_{computed})^2} \tag{2}$$

We present the minimum, maximum, and average errors under each graph.

3.4 Results for Multidimensional Scaling

The output of the MDS approach was a flip and translate from the original layout. Figures 6, 7, 8, 9 and 10 are the output from graph 1, graph 2, graph 3, graph 4 and graph 5, respectively. Table 2 presents the error for each graph size. We should note that the hole present in graphs 2, 3, 4 and 5 were resolved in the output graphs.

3.5 Results for Force Directed

The output of the force directed approach was a flip and translate from the original graph. Node 0 was located on the left side of the input graph and located on the right side of the output graph. Figures 11,

Table 2: MDS Localization Error

Graph	Average	Min	Max
100 No Hole	0.067	0.005	0.17
95, Hole	0.55	0.06	1.07
200, Hole	0.72	0.06	1.07
490, Hole	0.60	0.02	1.4
990, Hole	0.08	0.002	0.18

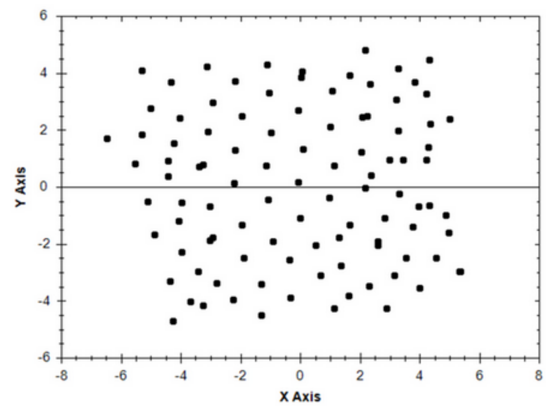

Figure 6: MDS Raw Output, 100 Nodes, No Hole

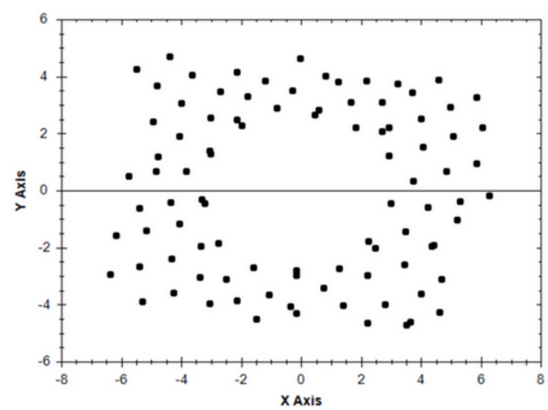

Figure 7: MDS Raw Output, 95 Nodes, With Hole

12, 13, 14 and 15 are the output graphs from the force directed approach. Table 3 presents the error for each graph size.

We note the considerable variability in the general shape of the output graphs. Input graph 3, as seen in 13, resulted in an output that resembles an hour glass and not the square layout of the original graph. Likewise, input graphs 4 and 5, as seen in Figures 14 and 15, resulted in outputs that visually vary considerably from the input graphs.

3.6 Results for Combinatorial Delaunay Complex

The output of the CDC algorithm was only a translation from the original graph. The CDC algorithm was run on the 990 node data set only. It was noted during this work the CDC algorithm was sensitive to

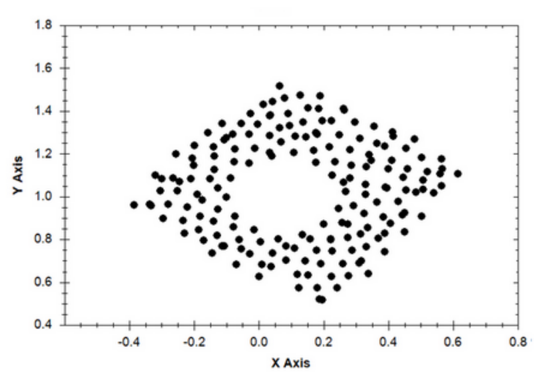

Figure 8: MDS Raw Output, 181 Nodes, With Hole

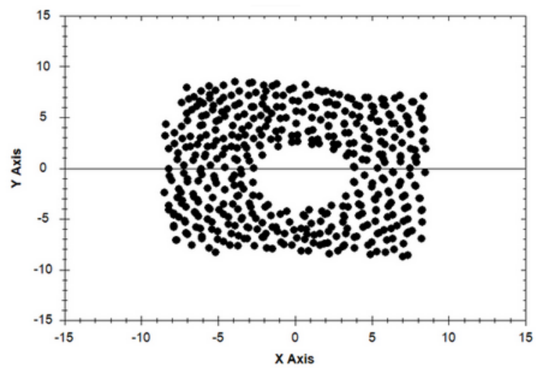

Figure 9: MDS Raw Output 490 Nodes

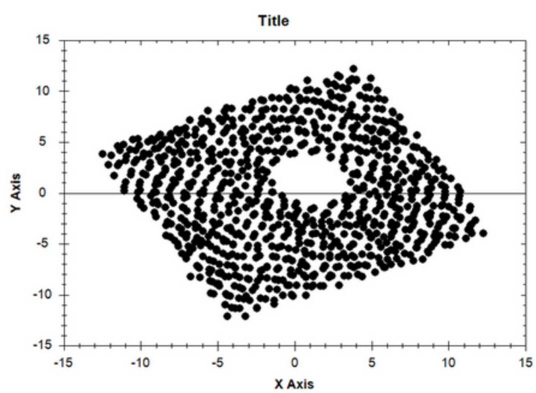

Figure 10: MDS Raw Output, 990 Nodes

the initial embedding for the landmarks, including the selection of the first landmarks that are embedded. The more hops included in the initial embedding, the greater the error that was observed. This is due to the fact that hop counts do not represent distantdistance. Figure 16 presents the output of the CDC algorithm. Table 4 presents the error for the CDC approach.

Table 3: Force Directed Localization Error

Graph	Average	Min	Max
100, No Hole	0.49	0.05	1.19
95, Hole	0.06	0.004	0.16
200, Hole	0.79	0.02	1.33
490, Hole	0.48	0.01	1.08
990, Hole	0.23	0.01	0.74

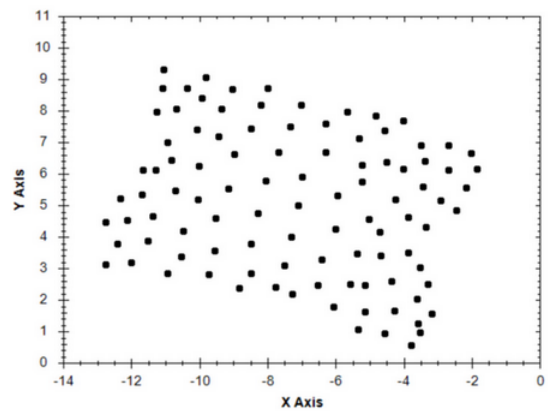

Figure 11: Force Directed Raw Output, 100 Nodes, No Hole

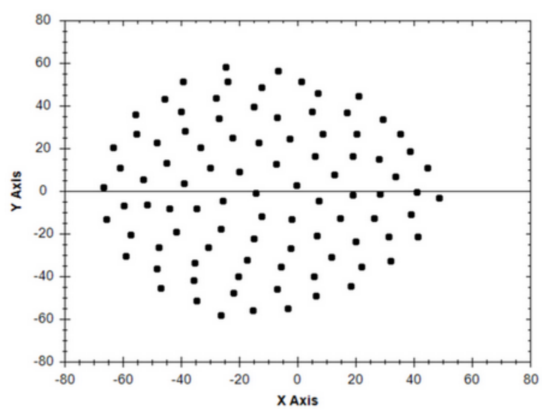

Figure 12: Force Directed Raw Output, 95 Nodes, No Hole

4 Analysis

All of the localization techniques focused on in this paper are based on connectivity. As such, these techniques do not resolve to an embedding within physical dimensions. Also, unless anchors are provided, the resulted embedding may be offset by a rotation, translation or mirror from the actual layout. Nevertheless, these techniques can be useful when it is impractical to include the necessary hardware required by other approaches.

We compare the average errors of the three localization techniques in Table 5. The MDS approach performed the best for the 100 nodes no hole and 990 nodes with hole. It performed well for the 990-node graph, even resolving the hole as seen in Figure 10. Force directed performed better than MDS with 95 nodes with a hole and slightly better than 490 nodes with a hole, though the error was significant. The

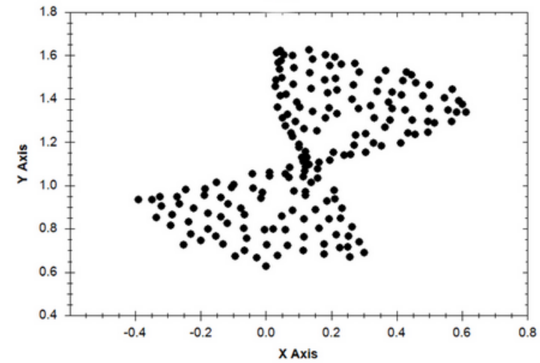

Figure 13: Force Directed Raw Output, 181 Nodes, Hole

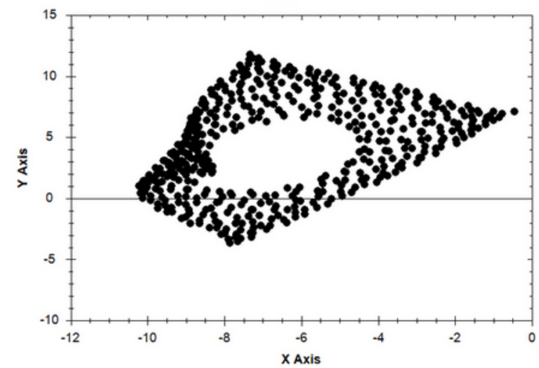

Figure 14: Force Directed Raw Output, 490 Nodes, Hole

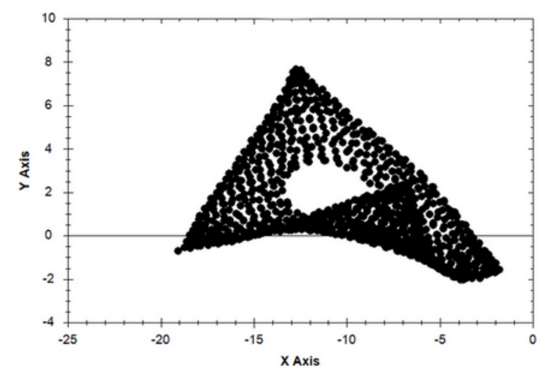

Figure 15: Force Directed Raw Output, 990 Nodes, Hole

CDC approach performed poorer than the MDS approach and roughly the same as the Force directed. It did not resolve the hole well. It should be noted that CDC works best on larger graphs [6].

These algorithms were implemented as centralized approach. So we assume that each node would transmit a single packet of information containing the list of neighboring nodes back to a central computer that would localize the network and then transmit back the result to each node. Traffic across the network would only occur when a change in the network is detected. This would work well for relatively static networks, but for dynamic networks, this approach would incur considerable network overhead. In such dynamic cases, the distributed approaches may work better. Some work has been done in that direction and we plan to evaluate their performance in our future work.

Table 4: CDC Localization Error

Graph	Average	Min	Max
990, Hole	0.25	0.004	0.5

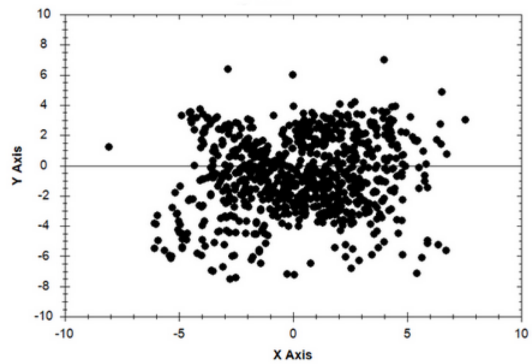

Figure 16: CDC Raw Output, 990 Nodes, Hole

5 Conclusions and Future Work

In this paper we examined localization techniques in wireless sensor networks and compared the performance of three connectivity-based algorithms: multi-dimensional scaling, force directed and combinatorial Delaunay complex, for localizing network nodes using only connectivity information. We examined network sizes up to 990 nodes with a single hole in the graph and compared the performance of the algorithms with respect to their ability to localize the nodes. We have shown the algorithms generated an embedding that was flipped, rotated or translated relative to the input graph. We have also shown these algorithms vary considerably in their ability to localize a graph. Performance ranged from as good as a 7 percent error to as poor as a 78 percent error.

Clearly, connectivity-based solutions are not mature enough to provide low error rate for critical applications. Further research is necessary to improve the performance of connectivity-based approaches while keeping their protocol overheads to a minimum. Specifically the graphs with holes create great challenges to the connectivity-based algorithms. Actually, CDC has been proposed to deal with the problem of holes within the graph. However, it requires larger graph and its performance was not at the desired level yet. In a follow on paper [7], the authors proposed a refinement to original work that utilizes an incremental Delaunay refinment method which allows for a more robust algorithm that is less sensitive to the noise results of their boundary detection. They showed the refinement performed well with networks of low average degree with complex shapes. Most recently the authors in [8] presented an approach they call Approximate Convex Decomposition Localization (ACDL), which decomposes the network into regularly graphs and then uses MDS approach.

Table 5: Average Localization Errors of Different Techniques

Graph	Multi-Dimensional Scaling	Force Directed	CDC
100, No Hole	0.067	0.49	—
95, Hole	0.55	0.06	—
200, Hole	0.72	0.79	—
490, Hole	0.60	0.48	—
990, Hole	0.08	0.23	0.25

In the future, we plan to work on new connectivity based algorithms. We also plan to expand our study to the distributed version of these algorithms to determine their strengths and weakness so that engineers and wireless network practitioners can select the best algorithms for deployment in their systems.

References

[1] P. Eades. A heuristic for graph drawing. *Congressus Nutnervantiunt*, 42:149–160, 1984.

[2] A. Efrat, D. Forrester, A. Iyer, S. Koborouv, C. Erten, and O. Kilic. Force-directed approaches to sensor localization. *ACM Transactions on Sensor Networks*, 7(3), September 2010.

[3] T. M. Fruchterman and E. M. Reingold. Graph drawing by force-directed placement. *Software - Practice and Experience*, 21(11):1129–1160, November 1991.

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

[5] H. Karl and A. Willig. *Protocols and Architectures for Wireless Sensor Networks*. John Wiley and Sons Ltd, April 2005.

[6] S. Lederer, W. Yue, and G. Jie. Connectivity-based localization of large scale sensor networks with complex shape. *ACM Transactions on Sensor Networks*, 5(4), November 2009.

[7] S. Lederer, W. Yue, and G. Jie. Connectivity-based localization with incremental delaunay refinement method. In *Proc. of the 28th Annual IEEE International Conference on Computer Communications (INFOCOM'09), Rio de Janeiro, Brazil*, pages 2401–2409. IEEE, April 2009.

[8] W. Liu, D. Wang, H. Jian, W. Liu, and W. Chonggang. Aproximate convex decomposition based loclization in wireless sensor networks. In *Proc. of the 31st Annual IEEE International Conference on Computer Communications (INFOCOM'12), Orlando, Florida, USA*, pages 1853–1861. IEEE, March 2012.

[9] G. Mao and B. Fidan. *Localization Algorithms and Strategies for Wireless Sensor Networks: Monitoring and Surveillance Techniques for Target Tracking*. IGI Global, May 2009.

[10] Y. Shang, W. Ruml, Y. Zhang, and M. P. J. Fromherz. Localization from mere connectivity. In *Proc. of the 4th ACM International Symposium on Mobile Ad Hoc Networking & Computing (MobiHoc'03), Annapolis, Maryland, USA*, pages 201–212. ACM Press, June 2003.

[11] I. Stojmenovic. *Handbook of Sensor Networks: Algorithms and Architetures*. Wiley, 2005.

[12] W. S. Torgeson. Multidimensional scaling of similarity. *Psychometrika*, 30(4):379–393, 1965.

Charles Zinsmeyer Received his B.Sc in computer science from St. Mary's University, San Antonio TX in 1986 graduating Cum Laude. He received a M.Sc in Computer Information Systems from St. Mary's University, San Antonio TX in 1996. He is currently a graduate student at the University of Texas at San Antonio working with Dr. Korkmaz on sensor localization. He is currently employed at Kinetic Concepts, Inc. as a Staff Engineer responsible for developing medical device software. He also holds an adjunct position at St. Mary's University teaching software engineering.

Turgay Korkmaz received the B.Sc. degree with the first ranking from Computer Science and Engineering at Hacettepe University, Ankara, Turkey, in 1994, and two M.Sc. degrees from Computer Engineering at Bilkent University, Ankara, and Computer and Information Science at Syracuse University, Syracuse, NY, in 1996 and 1997, respectively. In Dec 2001, Dr. Korkmaz received his PhD degree from Elec. and Computer Eng. at University of Arizona, under the supervision of Dr. Marwan Krunz. In January 2002, he joined the University of Texas at San Antonio as an Assistant Professor of Computer Science Department. Dr. Korkmaz received his tenure in September 2008, and he is currently an Associate Professor of Computer Science Department. Dr. Korkmaz works in the area of computer networks and network security.