

<http://ansinet.com/itj>

ITJ

ISSN 1812-5638

# INFORMATION TECHNOLOGY JOURNAL

**ANSI***net*

Asian Network for Scientific Information  
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

## Distributed Localization Based on Hessian Local Linear Embedding Algorithm in Wireless Sensor Networks

Shancang Li and Deyun Zhang

School of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an 710049, China

---

**Abstract:** This study shows a distributed and accurate algorithm for location based on manifold learning algorithms: Hessian Local Linear Embedded (HLLE) algorithm and a localization method framework based on manifold learning is proposed. The method takes advantage of additional information such as estimated distances between neighbors, or reference nodes' positions. The algorithm based on Hessian local linear embedded to obtain the relative map of nodes. Through simulation studies, we demonstrate that the algorithm is more robust to measurement error than previous proposals, especially when nodes are positioned relatively uniformly throughout the plane and it can achieve comparable results using fewer reference nodes than previous methods, if there are no reference nodes, relative coordinates are available.

**Key words:** Wireless sensor network, localization, manifold algorithm, Hessian local linear embedding

---

### INTRODUCTION

Node localization is an essential tool for the development of low-cost sensor networks with many small, battery-powered, wirelessly connected cost nodes for using in location-aware applications and ubiquitous networking (Ganesan *et al.*, 2002). Typical applications for wireless sensor networks (WSN) are to send a message to a node at given location (without knowing which node or nodes are there, or how to get there), to retrieve sensor data from nodes in a given region and to find nodes with sensor data in a given range. With a network of thousands of nodes, it is unlikely that the position of each node has been pre-determined. Nodes could be equipped with a global positioning system (GPS) to provide themselves with absolute positions. However, this is currently a costly solution for localization in large scale sensor networks. By eliminating the need for additional hardware for sensor localization, such as for GPS, ultrasound, or high accuracy RF time-of-arrival (TOA), we can widen the WSN application space (Patwari and Hero, 2004). Distributed computation and robustness in presence of measurement noise are key ingredients for a practical localization algorithm that will give reliable results over a large scale network.

As described Patwari and Hero (2004), connectivity measurement based localization, has been found considerable applications in sensor and ad hoc networks. Yi *et al.* (2003) presented an algorithm that used the connectivity information to derive the locations of the

nodes in the wireless sensor network. In the method, multi-dimensional scaling (MDS) was used to obtain the location map in low dimensions according to high dimension data samplings. Patwari and Hero (2004) and Yi *et al.* (2003) introduced a scalable, distributed weighted multidimensional scaling (dwMDS) algorithm that adaptively emphasized the most accurate range measurements and naturally accounted to communication constraints within the sensor network. Each node adaptively choose a neighborhood of nodes to update its coordinate estimate by minimizing a local cost function and then passed this update to neighboring sensors.

In this study, we present a method based on manifold learning for computing the location of nodes given only basic information that is likely to be already available, namely, which nodes are within communications range of which others. The method, HLLEMAP, has three steps. At first, starting with the known connectivity information, we use an all-pairs shortest-paths algorithm to roughly estimate the distance between each possible pair of nodes. Then we use Hessian local linear embedding techniques (HLLE) (Donoho, 2003), a technique for mathematical psychology, to derive node locations that fit those estimated distances. Finally, we normalize the resulting coordinates to take into account any nodes whose positions are known.

This method outperforms existing methods as we will demonstrate and it requires only connectivity information to produce localization information. If the distances between neighboring nodes can be estimated, that

information can be easily incorporated into the pairwise shortest-path computation during the first step of the algorithm and if the distances between neighboring nodes cannot be obtained, we use the connectivity information instead of the distances. HLLC yields coordinates that provide the best fit to the estimated pairwise distance and if there are more than three reference nodes in WSN, they can be used to derive the affine transformation of the HLLC coordinates that allows the best match to the known positions. Only three such reference nodes are necessary to provide absolute positions for all the nodes in the wireless sensor networks.

### LOCALIZATION BASED ON DHLLE

**Problem state:** We consider the node localization problem under two scenarios, in the first, only connectivity information is available, each node only knows which nodes are nearby, presumably by means of some local communication channel but not how far away these neighbors are or in what direction they lie. In the second scenario, the distances between node and neighbors are known. In both cases, we use an undirected graph with vertices  $V$  and edges  $E$  to represent wireless sensor networks. The vertices correspond to the nodes, of which zero or more may be special nodes, which are called reference nodes whose positions are already known. For the case with known distances to neighbors, the edges are associated with values corresponding to the estimated distances (Niculescu, 2004). We assume that all the nodes being considered in a connected graph.

There are two possible outputs when solving the localization problem, one is a relative map and the other is an absolute map. 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 reference nodes present to uniquely determine the absolute positions of the nodes. Furthermore, some applications only require relative position of nodes, such as in some direction-based routing algorithms. Sometime, 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 discovery and target tracking.

**Connectivity measurement model:** Connectivity is a binary measurement of whether or not two nodes are in communication range of each other (Patwari *et al.*, 2004; Savarese *et al.*, 2001). Typical digital receivers have a minimum received power threshold below which it is unlikely that a packet will be correctly modulated. Connectivity can be considered as a binary quantization

of Received Signal Strength (RSS). As a good approximation, when the RSS is below a power threshold, two nodes will not be connected; when the RSS is above the threshold, two devices will be connected. As a binary quantization of RSS, it is clear that connectivity is less informative than RSS and will result in higher localization variance bounds (Costa *et al.*, 2004). Research in connectivity based localization is often called range-free localization. There is an assumption that connectivity does not suffer from the same fading phenomena as RSS and instead that radio coverage is a perfect circle around the transmitter, can be a valuable simplification during algorithm development. However, this assumption cannot be used to accurately test the performance of such algorithms. Range-free localization algorithms can be simulated by generating measurements of RSS between each pair of sensors using the log-normal model and then considering each pair with an RSS measurement above a threshold power to be connected. By sampling over time  $t = 1, \dots, T$ ,  $n$  sensors generate a data point in a high dimensional space. Two dimensional sensor localization estimations can be seen as the reduction of the dimensionality of the data from  $n$  to 2. In this study, we consider only 2D location, however, the analysis can be extended to 3D easily. Many dimension reduction techniques have been developed in the statistical learning community, such as principal components analysis (PCA) and multi-dimension scaling. In (Savarese *et al.*, 2001), node location estimation using MDS was introduced and demonstrating that centralized, global sensor localization could be achieved without resorting to iterative optimization algorithms that do not always converge to the global coverage.

In the existing range-based localization methods (Niculescu, 2004; Savarese *et al.*, 2001), as well as RSS, when range errors are i.i.d. and the Gaussian distribution, however, in the wireless communication environments the range errors are often non-Gaussian distribution and most range estimation methods degrade with increasing distance. This effect is particularly severe in the DC localization problems. The error is a function of range, so it can be used to estimate accurately the distance. Here, we model the connectivity measurement between sensor  $I$  and sensor  $j$  with a binary quantization of RSS: there are a received power-threshold at sensor  $I$  transmitted by sensor  $j$ , if the power under than the threshold, the value of the variable is 0 it represents under the power received packets can not be demodulated and the two sensors are not connected, else, the value is 1, it represent that under the received power the two sensor can communicate normal. The connectivity measurement model of sensor  $I$ ,  $j$  as follow:

$$Q_{ij} = \begin{cases} 1, & P_{ij} \geq P_1 \\ 0, & P_{ij} < P_1 \end{cases} \quad (1)$$

Here,  $P_{ij}$  is the sensor  $I$  received power (dBm) transmitted by sensor  $j$  and  $P_1$  is the receiver threshold (dBm) which is related to the range threshold  $R$ . In (Ganesan *et al.*, 2002), we can obtain the calculation and measurement of  $P_{ij}$ . In this study, we used TPKNN (Costa *et al.*, 2004) as the neighbor nodes selection algorithm that based on connectivity model described above we can get the neighbors list of each sensors.

**DHLLLE algorithm:** DHLLLE is based on a well-established technique know as Distributed Hessian Locally Linear embedding (DHLLLE), DHLLLE is a method for recovering the underlying parameterization of scattered data ( $m_i$ ) lying on a manifold  $M$  embedded in high-dimensional Euclidean space. HLLLE may be viewed as a modification of locally linear embedded and its theoretical framework as a modification of the Laplacian Eigenmaps framework, where HLLLE substitute a quadratic form based on the Hessian in place of one based on the Laplacian. DHLLLE starts with one or more matrices representing distances or similarities between objects and finds a placement of points in a low-dimensional space, usually two-dimensional or three-dimensional, where the distances between the points resemble the original similarities. DHLLLE is often used as part of exploratory data analysis or information visualization. By visualizing objects as points in a low-dimensional space, the complexity in the original data matrix can often be reduced while preserving the essential information.

DHLLLE algorithm is applied to reduce the dimension of the sensor data  $\{s_i\}$ , DHLLLE output a relative map possibly a non-physical scale.

We have node sampled data ( $m_i$ ) and we would like to recover the underlying parameterization  $\psi$  and parameter setting  $\theta$ , DHLLLE is a novel algorithm based on LLE algorithm.

**DHLLLE algorithm**

**Input:** ( $m_i, I = 1, \dots, N$ ), a collection of  $N$  data points in  $R^n$ ;

**Parameters:**  $d$ , the dimension of the parameter space;  
 $k$ , the size of the neighborhoods for fitting;

**Constraints:**  $\min(k, n) > d$

**Output:** ( $w_i, I = 1, \dots, N$ ) a collection of  $N$  node data points in  $R^d$ , the recovered parameterization.

**Step 1: Identify neighbors:** For each data point  $m_i, I = 1, \dots, N$ , identify the indices corresponding to the  $k$ -nearest neighbors in Euclidean distance, let  $N_i$  denote the collection of those neighbors. For each neighborhood  $N_i$ , form a  $k \times n$  matrix  $M^i$  with rows that consist of the reentered points  $m_j - \bar{m}_i, j \in N_i$ , where  $\bar{m}_i = \text{Ave}\{m_j : j \in N_i\}$  (Donoho, 2003).

**Step 2: Obtain tangent coordinates:** Perform a singular value decomposition of  $M^i$  and getting matrices  $U, D$  and  $V$ ;  $U$  is  $k$  by  $\min(k, n)$ , the first  $d$  column of  $U$  give the tangent coordinates of points in  $N_i$ .

**Step 3: Develop Hessian estimator:** Develop the infrastructure for least-squares estimation of the Hessian. Matrix  $H^i$  with the property that if  $f$  is a smooth function  $f: M \rightarrow R$  and  $f_i = (f(m_i))$ , then the vector  $v^i$  with entries sponding to points in the neighborhood  $N_i$ ; then, the matrix vector produce  $N_i, v^i$  gives a  $d(d+1)/2$  vector with entries that approximate the entries of the Hessian matrix

$$\frac{\partial^2 f}{\partial U_i \partial U_j}$$

**Step 4: Develop quadratic form:** Build a symmetric matrix  $H_{ij}$  having, in coordinate pair  $ij$ , the entry

$$H_{ij} = \sum_1 \sum_r ((H^1)_{r,i} (H^1)_{r,j})$$

Here, by  $H^1$  we mean, again, the  $d(d+1)/2 \times k$  matrix associated with estimating the Hessian over neighbourhoood  $N_i$ , where rows  $r$  correspond to specific entries in the Hessian matrix and column  $I$  correspond to specific points in the neighbourhoood.

**Step 5: Find approximate null space:** Perform an eigenanalysis of  $H$  and identify the  $(d+1)$ -dimension subspace corresponding to the  $d+1$  smallest eigenvalues. There will be an eigenvalues  $0$  associated with the subspace of constant functions; and the next  $d$  eigenvalues will correspond to eigenvector spanning a  $d$ -dimensional space  $\hat{V}_d$  in which our embedding coordinates are to be found.

**Step 6: Find basis for null space:** Select a basis for  $\hat{V}_d$ , which has the  $N_0$  provides an orthonormal basis. The given basis has basic vector  $w^1, \dots, w^d$ , these are the embedding coordinates.

The algorithm is a straightforward implementation of the idea of estimating tangent coordinates, the tangent Hessian and the empirical version of the operator  $H$ .

DHLLLE algorithm output the local map of nodes in sensor networks., after merging local maps we can redefine the global map (optimal), using the node coordinates in the global map as the initial solution, we apply the least squares minimization to make the distances between neighboring nodes match the measurement ones. Given sufficient reference nodes (3 or more for 2D networks, 4 or more for 3D networks), transform the global map to an absolute map based on the absolute positions of reference nodes.

**The DHLLLE method:** Here, we describe how Distributed HLLLE (DHLLLE) is applied to estimate sensor localization. In the DHLLLE algorithm, given a network the values of edges are assigned to a constant, such as 1, when only connectivity information is available. Otherwise, initially, each sensor  $I$  records data  $s_i(t_k)$  at time  $t_k$  and after time  $T$ , each node sends its data to its immediate neighbors. Here, we define  $k_i$  as the number of nodes with which node  $I$  can communicate and  $K$  is the number of neighbors defined by sensor nodes selection algorithm. If  $k_i < K$ , node  $I$  queries nodes that are one or more hops away from itself and if  $k_i = K$ , node  $I$  only receives data from those  $k_i$  nodes. Node  $I$  calculates Euclidean distance in  $R^T$  between its own data and its neighboring nodes' data. Here, we define  $N_i$  to be the set of  $K$  nodes which have data closest to node  $I$ 's data.

Based on DHLLLE, our method consists of four steps:

- Neighbor selection use the pairwise distance  $\delta_{ij}$  to determine which nodes could be considered as neighbors by each node. In our method, all pairs  $(i,j)$  with  $\delta_{ij} < R$  for some constant radius  $R$ , are considered to be neighbors. Since  $\delta_{ij} = \delta_{ji}$ , the  $R$ -radius neighbor relation is symmetric. However, if the radius  $R$  is set too low, the neighborhood graph may end up sparse or disconnected, while setting it too high would result in large neighborhoods, thus variants of the  $K$ -nearest neighbor (KNN) method are presented: sensor  $j$  is a neighbor of sensor  $I$ , i.e.,  $j \in N_i$  if and only if  $\delta_{ij}$  is one of the  $K$  smallest of the set  $\{\delta_{ik}\}_{k \neq i}$ . Clear,  $|N_i| = K$ , note KNN is not a symmetric relation. In dwMDS algorithm AKNN (Patwari *et al.*, 2003) was used to select neighbors and in ISOMAP algorithm, SKNN (Patwari *et al.*, 2003) was used to select neighbors.
- Compute the shortest paths between all pairs of nodes in the region of sensing which are used to construct the distance matrix for DHLLLE.
- Apply DHLLLE algorithm to the distance matrix to construct a 2-D (or 3-D) relative map.

- Matching reference 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 reference nodes.

In step 1, Apply KNN algorithm to find the neighbors of nodes, the time complexity is  $O(n)$ .

In step 2, we first assign distances to the edges in the connectivity graph. When the distance of a pair of neighbor nodes is known, the value of the corresponding edge is the measured distance. When we only have connectivity information, a simple approximation is to assign value 1 to all edges. Then a classical all-pairs shortest-path algorithm, such as Dijkstra's or Floyd's algorithm, can be applied. The time complexity is  $O(n^3)$ , where  $n$  is the number of nodes.

In step 3, 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 DHLLLE is a relative map that gives a location for each node. Although these locations may be accurate relative to one another, the entire map will be arbitrarily rotated and flipped relative to the true node positions.

In step 4, the relative map is transformed through linear transformations, which include scaling, rotation and reflection. The goal is to minimize the sum of squares of the errors between the true positions of the anchors and their transformed positions in the DHLLLE map. Computing the transformation parameters takes  $O(m^2)$  time, where  $m$  is the number of reference nodes. Applying the transformation to the whole relative map takes  $O(n)$  time.

DHLLLE requires the distance between every pair of nodes. The short path distance between two remote nodes provides an estimate of the true Euclidean distance. This estimate is fine when the networks are dense or uniform, but is not good for those irregular ones. When the estimation is off, the result of DHLLLE is not good.

## SIMULATION EXAMPLES

The above algorithms are implemented on NS2. In our experiments, 300 nodes are placed randomly in a  $300 \times 300$  m square and all simulations are run using this square. In the connectivity-only cases, each node only knows the identities of nodes in its neighborhood but not the distance to them. In the known-distance cases, each node knows the distances to its neighbor nodes by using Received Signal Strength (RSS). The distance is modeled as the true distance blurred with Gaussian noise. Assume the true distance is  $d_i$  and range error is  $e_i$ ; then the measured distance is a random value drawing from a normal distribution  $d_i(1+N(0,e_i))$ . The connectivity is

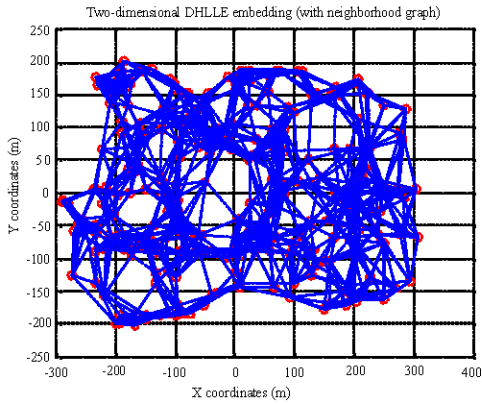


Fig. 1: Connectivity graph based on DHLLE algorithm

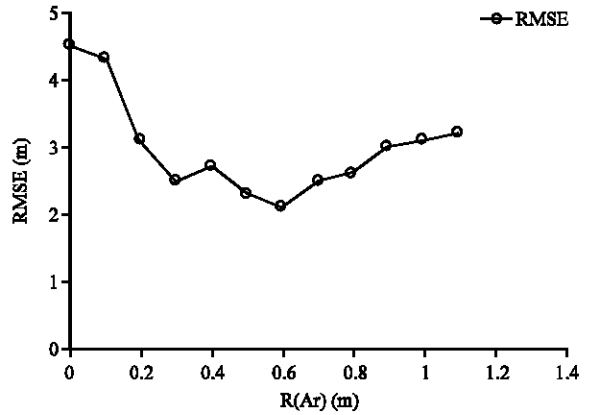


Fig. 3: RMSE versus threshold distance for the random uniform example using DHLLE

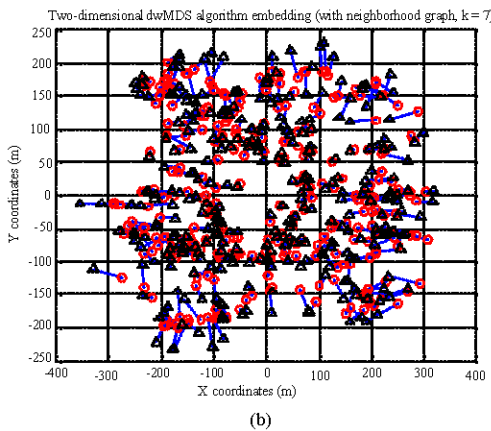
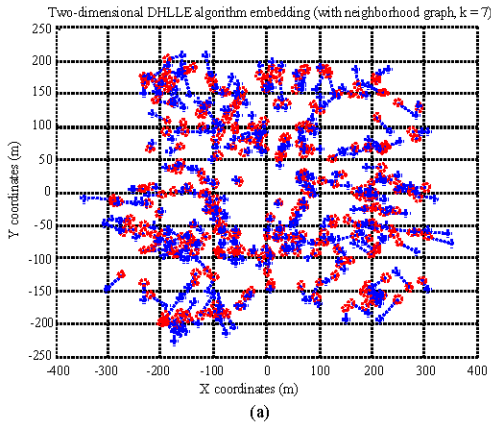


Fig. 2: Results of DHLLE and dwMDS on the example of random placement. a): Results of DHLLE and b): Results of dwMDS

controlled by specifying radio range  $R$ . Figure 1 shows the final map constructed by DHLLE using radio range of 15 m, which leads to an average connectivity of 11. In

the graph,  $o$  represent nodes and edges represent the connections between neighbors who can directly communicate and the graph illustrate the relative map of all nodes in WSN which is obtained by using the DHLLE algorithm. We find for DHLLE, when  $k = 7$ , it seem to work best. For each test described below, we fix a particular geometry of devices and then run 100 trials, from which we calculate the means node position estimates. Since the dwMDS localization method has the best estimation performance, we only compared our method with dwMDS method in this study.

First, we investigate the performance when nodes are placed random, i.e., the case when the  $x$  coordinates and  $y$  coordinates of nodes are uniformly distributed in  $[-300, 300]$  for all nodes. In our experiments the distances between neighbors are known, even with limited accuracy, the result of DHLLE can be significantly improved. Figure 2 shows the result of DHLLE knowing the distance of neighbors with 5% range error. Figure 2a and b show the estimations of the DHLLE and dwMDS algorithms for a particular realization of the uniformly random node deployment. Here, we ran 100 Monte Carlo simulation trails to determine the estimation of position and root-mean-square error (RMSE). The results are displayed in Fig. 2 and 3. In Fig. 2 we plot the estimated position and compare it to the actual location. Figure 2a shows the final estimated position of DHLLE is denoted by  $+$ . The circles represent the actual position of nodes and the solid lines represent the errors of the estimated position from the actual position, the longer the line, the larger the error. From the Fig. 2a and 2b we can seen that the estimation error of DHLLE is smaller than that of dwMDS.

When the connectivity level is 11 or grater, the RMSE of positions estimates by DHLLE have an average under 2.0, with the same scenario, the RMSE of dwMDS

is about 3.5, it is obvious that the estimation by DHLLE is much better than that of dwMDS which has the best estimation performance in existing approaches. In contrast, our method reduces the error from about 80% down to 40% as the radio range goes from 1.25R to 2R.

We also studied the influence of the threshold distance on the RMSE performance of the DHLLE algorithm. Figure 3 shows a plot of the RMSE vs. threshold distance, for the 300×300 m random uniform example using DHLLE. It can be seen that there is an optimal threshold distance,  $d_r = 15$  m, beyond which, no performance increase occurs. By the RSS measurement model, the accuracy of range measurements degrades quickly with distance, thus adding these far way nodes will not bring any gain to the estimation algorithm.

### CONCLUSION

This study proposes a distributed HLLLE method specially suited for node localization in wireless sensor network. First, the method reflects the distributed nature of the problem, incorporating network communication constraints in its design. In this way, the need to transmit all range measurements to a central unit is eliminated, resulting in energy savings for a dense sensor network. Second, DHLLE works well with mere connectivity information. However, it can also incorporate distance information between neighboring nodes when it is available. The strength of DHLLE is that it can work well when there are few or no reference nodes. Previous approaches often require well-placed reference nodes to work well. DHLLE builds a relative map of the nodes even without reference nodes. When there are three or more reference nodes, the relative map can be transformed and absolute coordinates of the nodes are computed.

Simulations show that the method is effective and particularly for situations with few reference nodes and relatively uniform node distributions.

### REFERENCES

- Costa, J.A., N. Patwari and A.O. Hero III, 2004. Distributed multidimensional scaling with adaptive weighting for node localization in sensor networks [J], submitted to ACM Transactions On Sensor Networks (TOSN), 2: 39-64.
- Donoho, D.L., C. Grimes and H. Eigenmaps, 2003. Locally linear embedding techniques for high-dimensional data. *Science*, 100: 5591-5596.
- Ganesan, D., B. Krishnamachari and A. Woo *et al.*, 2002. An empirical study of epidemic algorithms in large scale multi-hop wireless network. Technical report UCLA/CSD-TR-02-0013, UCLA Computer Science Department.
- Niculescu, D., 2004. Positioning in ad hoc sensor networks. *IEEE Network*, 18: 24-29.
- Patwari, N., A.O. Hero III, M. Perkins, N. Correal and R.J.O. Dea, 2003. Relative location estimation in wireless sensor networks [J]. *IEEE Trans. Sig. Proc.*, 51: 2137-2148.
- Patwari, N. and A.O. Hero, 2004. Manifold learning algorithms for localization in wireless sensor networks. *Proc. of Acoustics, Speech and Signal 2004 (ICASSP '04)*, 3: 857-860.
- Savarese, C., J.M. Rabaey and J. Beutel, 2001. Locationing in distributed ad-hoc wireless sensor networks. *ICASSP.*, pp: 2037-2040.
- Yi, S., W. Ruml and Z. Ying, 2003. Localization from Mere Connectivity, *MobiHoc03*, 2003, Maryland, USA., pp: 201-212.