A NEW DISTRIBUTED WEIGHTED MULTIDIMENSIONAL SCALING ALGORITHM FOR LOCALIZATION IN WIRELESS SENSOR NETWORKS

Fahimeh Doremami[1] , Dr. Hamid Haj Seyyed Javadi[2] and Dr. Ahmad Farahi[3]

[1]Technical & Engineering College, Tehran Payame noor University, Tehran, Iran
fahime_emami@yahoo.co.uk
[2] Basis Science College, Shahed Univesity, Tehran, Iran
h.javadi@shahed.ac.ir
[3] Technical & Engineering College, Payame noor University, Tehran, Iran
afaraahi@pnu.ac.ir

ABSTRACT

There are various applications in wireless sensor networks which require knowing the relative or actual position of the sensor nodes, but adding GPS receivers or other sophisticated sensors to every node can be expensive. Multidimensional scaling (MDS) is a recent localization technique that it uses connectivity information. In this paper, we propose a new iterative distributed localization algorithm based on Multidimensional Scaling in which each sensor updates its position estimate by minimizing the corresponding local cost function, after taking measurements and initial position estimate from its neighboring nodes. The experiments show the proposed algorithm reduces communication cost, complexity, and convergence time.

KEYWORDS

Wireless Sensor Network; WSN; Localization; Distributed Localization Algorithm; Multidimensional Scaling; MDS; Weighted Multidimensional Scaling

1. INTRODUCTION

A wireless sensor network comprises of a large number of wireless sensor nodes. These sensors could be measuring a variety of properties, including temperature, acoustics, light, and pollution. Usually, a wireless sensor network has high density and is randomly deployed. For example, an aircraft can have a large number of wireless sensor nodes and scatter them over a forest. MEMS IC technology has made possible the use of large networks of wireless sensors for a variety of new monitoring and control applications [12].

On the basis of nodes that have sensing and actuation capabilities, many different application scenarios can be constructed. The major application scenarios include; disaster relief applications (i.e. detection of wildfire or forest fire); environment control (i.e. detection of chemical pollution earth movement, habitat monitoring); intelligent buildings (i.e. HVAC systems control for energy consumption optimization, monitoring mechanical stress levels in structures); facility management (tracking people and assets, monitoring leaking chemicals in chemical plants); preventive maintenance of machines, medicine and health care[4].

A micro-sensor is a small sized and low powered electronic device with limited computational and communication capabilities. A Wireless Sensor Network (WSN) is a network containing some tens of millions of such micro-sensors [12][4].

DOI : 10.5121/ijcses.2011.2105

The automatic location detection of sensors, namely localization is an important issue for WSN applications, especially in cases, when the sensors are deployed randomly, or when they move about after deployment. One reason is that a sensor's location must be known for its data to be meaningful. In most cases, the location itself is often the main data that needs to be sensed. Position information is essential to many location-aware sensor network communication protocols such as packet routing and sensing coverage [4][1].

The localization in WSN is a significant, key enabling technology, attracting considerable research interest. With the constrained resources of network sensors, as well as their high failure rate, many challenges exist in the automatic determination of the sensor's location [4].

Multidimensional scaling (MDS) is a recent localization technique. It uses connectivity information—that is within communications range of whom—to derive the locations of the nodes in the network, and can take advantage of additional data, such as estimated distances between neighbors or known positions for certain anchor nodes, if they are available. Recently, there has been different localization algorithms proposed based on Multidimensional Scaling.

In this paper, we explore and analyze distributed weighted multidimensional scaling Algorithm for localization in wireless sensor networks then propose a new distributed weighted multidimensional scaling algorithm. The remaining parts of this paper are organized as follows: In Section 2, reports related work of the proposed Localization Algorithms based on Multidimensional Scaling (MDS), In Section 3, describes Multidimensional Scaling. In Section 4, describes distributed weighted multidimensional scaling (dwMDS) Algorithm. In Section 5, proposes a new distributed weighted multidimensional scaling Algorithm. In Section 6, Simulates and analyses proposed algorithm. Finally, Section 7 concludes mentioned algorithm in this paper.

2. RELATEED WORK

One of the techniques is used for localization, is multidimensional scaling. Recently, there have been different centralized localization algorithms and one distributed algorithm (dwMDS) proposed based on Multidimensional Scaling.

In [11] proposes Multidimensional Scaling Map (MDS-MAP) Algorithm. This Algorithm is based on Multidimensional Scaling. It uses connectivity information within communications range. Fist Compute shortest paths between all pairs of nodes in the region of consideration and construct the distance matrix takes $O(n^3)$ time, where n number of nodes. Second Apply MDS to the distance matrix takes $O(n^3)$ time. Final transform the relative map to an absolute map based on the absolute positions of anchors takes $O(m^3)$ time, where m is the number of anchors. Applying the transformation to the whole relative map takes O(n) time. With three or more anchor nodes, the relative map can be transformed and absolute coordinates of the sensor nodes are computed. Extensive simulations using various network arrangements and different levels of ranging error show that the method is effective, and particularly so for situations with few anchor nodes and relatively uniform node distributions. A drawback of MDS-MAP is that it requires global information of the network and centralized computation and not work well on irregularly-shaped networks and takes $O(n^3)$ time. In addition Sensors and each distance are weighted equally.

In [10] proposes Improved MDS-Based algorithm called MDS-MAP(P). The main idea is to build a local map at each node, consisting of only nearby nodes then merge these patches of relative maps together to form a global solution and takes O(n^3) time. A drawback of MDS-MAP(P) is that it requires complex step of merging local maps together to form the global map and takes O(n^3) time too restricts weights to be either 0 or 1.

In [7] proposes Distributed Weighted-Multidimensional Scaling for Localization (dwMDS). dwMDS is a scalable decentralized algorithm that emphasizes the most accurate range measurements for node localization. This is enabled by a weighted cost function that more heavily weighs measurements believed to be more accurate. The majorization stage has the property that each iteration is guaranteed to improve the cost function and takes O(LN) time. Where L is number of iteration and N is number of node. It has great communication with neighboring nodes because in each iteration transmits its estimated location and local stress to neighboring nodes. It does not have the complex step of merging local maps together to form the global map. A drawback of dwMDS is that complexity, convergence time, high communication cost and initial estimate requirements.

In [6] propose a Hierarchical MDS-based Localization Algorithm (HMDS). First, HMDS partition the network into multiple clusters, each cluster has head. Second, the cluster head can compute distances of all pairs of sensors. Cluster head applying a multidimensional scaling algorithm on a distance matrix computes the relative coordinates of each cluster member and forms a local map. Finally each cluster with ,at least two cluster merge into a unified coordinate system. HMDS localization algorithm is more accurate and can operate in non-convex network environments. Experimental results revealed that the proposed HMDS localization algorithm outperformed the MDS-MAP algorithm in terms of accuracy. A drawback of HMDS is the roughly high error margin. Also It has the complex step of merging local maps together to form the global map and takes O(n^3) time for compute shortest path step.

In [5] propose Alternative Least-Square Scaling Algorithm (ALESSA) for localization. It is a centralized algorithm and uses Least-Square stress function. A drawback of ALESSA that it is centralized algorithm.

New proposed algorithm is a scalable decentralized algorithm also hasn't to estimate distance between out of range sensors. This is enabled by a fuzzy weighted cost function that more heavily weighs measurements believed to be more accurate. If a sensor moves, only the moved sensor will have to recalculate its location locally rather than all sensors are involved. Also the proposed algorithm reduces communication cost, complexity, and convergence time.

3. MULTIDIMENSIONAL SCALING

Multidimensional Scaling (MDS) refers to a set of techniques. One of the first well-known MDS a technique was proposed by Torgerson [3] in 1952. Multidimensional scaling can be considered as involving three basic steps. In the first step, a scale of comparative distances between all pairs of stimuli is obtained. Hence, a comparative distance is not a distance in the usual sense of the term, but is a distance minus an unknown constant. The second step involves estimating this unknown constant. When the unknown constant is obtained, the comparative distances can be converted into absolute distances. In the third step, the dimensionality of the psychological space necessary to account for these absolute distances is determined, and the projections of stimuli on Axes of this space are obtained.

Since then MDS has been used in empirical studies in political science, psychology, sociology, anthropology, economics, and education. Now called Classical MDS, Torgerson's method uses the Law of Cosine to place objects or entities in a Euclidean Space such that the distance between the objects correspond to measured distances, or dissimilarity.

There are two different forms of MDS, depending on how much information we have about the distances. In some of the applications we reviewed above the dissimilarities are known numbers, equal to distances, except perhaps for measurement error. In other cases only the rank order of the dissimilarities is known, or only a subset of them is known.

If all dissimilarities between the points represent true Euclidean distances then metric scaling can be used to find a configuration of these points. The distance between point r and s in an n-dimensional Euclidean space is given by the formula:

$$d_{r,s} = \sqrt{(x_r - x_s)^T (x_r - x_s)} \qquad (1)$$

From equation (1) can be defined to be:

$$d_{r,s}^2 = (x_r - x_s)^T (x_r - x_s) \qquad (2)$$

Now with the squared distances d_{rs}^2 the inner product matrix B is found. Then from B the unknown coordinates:

$$[B]_{rs} = b_{rs} = x_r^T x_s$$

To find B from (2) (Multiplying out):

$$d_{rs}^2 = x_r^T x_r + x_s^T x_s - 2 x_r^T x_s \qquad (3)$$

By placing the centroid of the configuration at the origin and further derivation the following is found:

$$b_{rs} = -\frac{1}{2}\left(d_{rs}^2 - \frac{1}{n}\sum_{r=1}^{n} d_{rs}^2 - \frac{1}{n}\sum_{s=1}^{n} d_{rs}^2 + \frac{1}{n^2}\sum_{r=1}^{n}\sum_{s=1}^{n} d_{rs}^2 \right) \qquad (4)$$

The inner product matrix B can be expressed as$\mathrm{B} = \mathrm{XX}^\mathrm{T}$. B is symmetric, positive semi-definite and of rank dimensionality. Matrix B can now be written in terms of its spectral decomposition or singular value decomposition as $B = V\Lambda \mathrm{V}^\mathrm{T}$. Where $\Lambda = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_\mathrm{n})$the diagonal matrix of eigenvalues of B and V is is the corresponding matrix of eigenvectors. Thus as $\mathrm{B} = \mathrm{XX}^\mathrm{T}$.

$$X = V\Lambda^{1/2} \qquad (5)$$

4. DISTRIBUTED WEIGHTED MULTIDIMENSIONAL SCALING(DWMDS)

Distributed Weighted Multidimensional Scaling or dwMDS presented by Costa, Patwari and Hero in [7][9] is a scalable decentralized algorithm that emphasizes the most accurate range measurements for node localization. This is enabled by a weighted cost function that more heavily weighs measurements believed to be more accurate. The majorization stage has the property that each iteration is guaranteed to improve the cost function. The method builds upon Multidimensional Scaling by minimizing multiple local loss functions. The local non-linear least squares minimization problem is solved using a quadratic surrogate function. Each sensor contributes to a global loss function by additively incorporating the local loss functions via the network.

The problem of selecting neighborhoods is addressed with a two-stage neighbor selection process. This overcomes the problem that selecting neighbors is subject to measurement noise and may favor devices which are reporting shorter than actual distances. However for the Relate system, a complete network is the norm and the whole network can be treated as one neighborhood.

Above, in equations (4) to (5) obtaining the matrix B requires knowledge of all pair wise dissimilarities, or an estimation of them, as [11] uses. In practice however such a scenario is unlikely at a time t, sensor hardware can lose power or malfunction then be replaced. Computing the singular value decomposition of B requires that all dissimilarities are gathered centrally somewhere. Finally as classical scaling minimizes the least squares error between d_{ij}^2 and δ_{ij}^{2} which ignores the underling error distribution model.

4.1. Symbol Definition

The symbols that use in dwMDS mentions as follows:

n: Sensors to be placed.
m: number of Anchor nodes.
x_i: Coordinate vector of sensor X- Coordinate matrix.
$d_{ij}(X)$: Distance between i and j in X, $d_{ij}(X) = \|x_i - x_j\| = \sqrt{(x_i - x_j)^T (x_i - x_j)}$.
$\delta_{ij}^{(t)}$: Range measurement between i and j.
$w_{ij}^{(t)}$: Weight given to range measurement between i and j at time t.
$\overline{\delta_{ij}}$: Weighted average range between i and j, $\overline{\delta_{ij}} = \sum_{t=1}^{K} w_{ij}^{(t)}\, \delta_{ij}^{(t)} / \overline{w_{ij}}$.
$\overline{w_{ij}}$: Weight given to the average measure range between i and j, $\overline{w_{ij}} = \sum_{t=1}^{K} w_{ij}^{(t)}$.

4.2. Cost Function

dwMDS as the Solution to the minimization of The following global STRESS function:

$$S = 2 \sum_{1 \le i \le n} \sum_{i \le j \le n+m} \sum_{1 \le t \le K} w_{ij}^{(t)} (\delta_{ij}^{(t)} - d_{ij}(X))^2 + \sum_{1 \le i \le n} r_i \|x_i - \overline{x_i}\| \qquad (6)$$

This differs from a standard formulation of an MDS stress function in that there is a penalty term to account for prior knowledge. This cost function can automatically adapt to different measurement models and data-dependent weighting schemes. After simple manipulations, *S* can be rewritten as follows:

$$S = \sum_{i=1}^{n} S_i + c \tag{7}$$

Where local cost functions Si are defined for each unknown-location node (ie.1≤i≤n), the local cost function at each unknown-location node is defined:

$$S_i = \sum_{j=1, j\neq i}^{n} \bar{w}_{ij}(\bar{\delta}_{ij} - d_{ij}(X))^2 + \sum_{j=n+1}^{n+m} 2\bar{w}_{ij}(\bar{\delta}_{ij} - d_{ij}(X))^2 + r_i\|x_i - \bar{x}_i\|^2 \tag{8}$$

If $X^{(k)}$ is the matrix whose columns contain the position estimates for all points at iteration k, one can derive an update for the position estimate of node i using equation (9):

$$x_i^{(k)} = a_i\left(r_i\bar{x}_i + X^{(k-1)}b_i^{(k-1)}\right) \tag{9}$$

Where

$$a_i^{-1} = \sum_{j=1, j\neq i}^{n} \bar{w}_{ij} + \sum_{j=n+1}^{n+m} 2\bar{w}_{ij} + r_i \tag{10}$$

And $\mathrm{b}_\mathrm{i}^{(\mathrm{k})} = [\mathrm{b}_1, \mathrm{b}_2, \ldots.., \mathrm{b}_{\mathrm{n+m}}]$ is a vector whose entries are given by:

$$b_j = \bar{w}_{ij}[1 - \bar{\delta}_{ij}/d_{ij}(X^{(k)})] \qquad j \leq n, j \neq i$$

$$b_i = \sum_{j=1, j\neq i}^{n} \bar{w}_{ij}\bar{\delta}_{ij}/\, d_{ij}(X^{(k)}) + \sum_{j=n+1}^{n+m} 2\bar{w}_{ij}\bar{\delta}_{ij}/d_{ij}(X^{(k)})$$

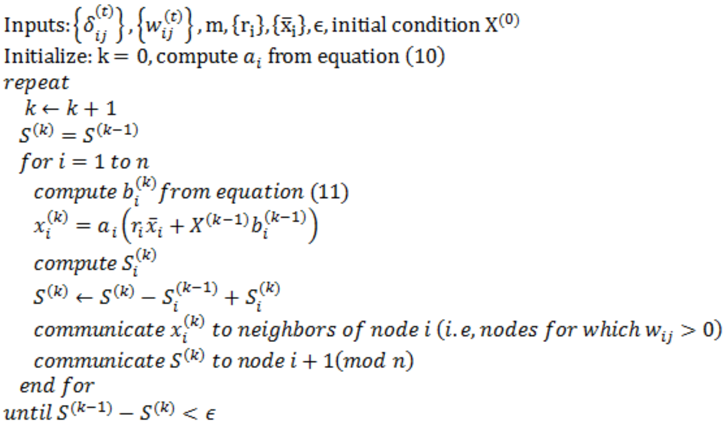
$$b_j = 2\bar{w}_{ij}[1 - \bar{\delta}_{ij}/d_{ij}(X^{(k)})] \qquad j > n \tag{11}$$

The dwMDS algorithm is summarized in Figure1. The choice of weighting function should reflect the accuracy of the range measurements, such that less accurate measurements are down-weighted in the overall cost function. Adopting a model-independent adaptive weighting scheme. The default weighting function for dwMDS is:

$$w_{ij} = \begin{cases} exp\,\{-\delta_{ij}^2/h_{ij}^2\} & , if\ \delta_{ij}\ is\ measured \\ 0 & , otherwise \end{cases} \tag{12}$$

Where

$$h_{ij} = \max\{max_k\ \delta_{ik}, max_k\ \delta_{kj}\}$$



```
Inputs: $\{\delta_{ij}^{(t)}\}, \{w_{ij}^{(t)}\}$, m, $\{r_i\}, \{\bar{x}_i\}, \epsilon$, initial condition $X^{(0)}$
Initialize: $k = 0$, compute $a_i$ from equation (10)
repeat
   $k \leftarrow k + 1$
  $S^{(k)} = S^{(k-1)}$
  for $i = 1$ to $n$
    compute $b_i^{(k)}$ from equation (11)
    $x_i^{(k)} = a_i \left( r_i \bar{x}_i + X^{(k-1)} b_i^{(k-1)} \right)$
    compute $S_i^{(k)}$
    $S^{(k)} \leftarrow S^{(k)} - S_i^{(k-1)} + S_i^{(k)}$
    communicate $x_i^{(k)}$ to neighbors of node i (i.e, nodes for which $w_{ij} > 0$)
    communicate $S^{(k)}$ to node $i + 1 (mod\ n)$
  end for
until $S^{(k-1)} - S^{(k)} < \epsilon$
```

Figure 1. Algorithm for decentralized weighted-multidimensional scaling

5. PROPOSED ALGORITHM

5.1. cost function

In proposed algorithm we define MDS as the solution to the minimization of the following local cost function for each sensor i:

$$S_i = min \sum_{j=1, j \neq i}^{n} w_{ij} \left({d_{ij}}^2 - {\delta_{ij}}^2 \right)^2 \qquad (13)$$

Where d_{ij} denotes the Euclidean distance between sensors i and j, and δ_{ij} is the estimated distance blurred with noise, and n is the number of unknown-location node. The weight function defines follow, for $i = j$, $w_{ij} = 0$ and for $i \neq j$, w_{ij} calculates by the following equation:

$$\begin{cases} 1 - \delta_{ij}/Range & \text{if j is beacon}, \delta_{ij} < Range \\ 1 - \delta_{ij}/radius & if\ j\ \ is\ sensor, \delta_{ij} < radius \end{cases} \qquad (14)$$

The radius of sensor range for select neighbors defines by radius variable. Stress function only depends on the measurements available at node i and the positions of neighboring nodes. i.e., nodes for which $w_{ij} > 0$. It can be viewed as the local cost function at node i. this is the key enabling factor for implementing a distributed localization procedure.

We propose an iterative distributed algorithm in which each sensor updates its position estimate by minimizing the corresponding local cost function, after taking measurements and initial position estimate from its neighboring nodes.

If X^0 is the matrix whose row contain the position estimates for all unknown-location node, stress function of S_i results in the following update equation for the position estimate of node i:

$$x_i^{(k)} = a_i \left(b * X^{(0)} \right) \qquad (15)$$

Where

$$a_i^{-1} = \sum_{j=1, j\neq i}^{n} w_{ij} \qquad (16)$$

And $\mathrm{b} = [\mathrm{b}_1, \mathrm{b}_2, \ldots., \mathrm{b}_\mathrm{n}]$ is a vector whose entries are given by:

$$\mathrm{b_i} = \sum_{\mathrm{j=1}}^{\mathrm{n}} \mathrm{w_{ij}}\, \delta_{\mathrm{ij}}/\mathrm{d_{ij}} + \sum_{\mathrm{j=1}}^{\mathrm{m}} \mathrm{w_{ij}}\, \delta_{\mathrm{ij}}/\mathrm{d_{ij}} \quad \text{where } i \neq j \qquad (17)$$

$$\mathrm{b_j} = \mathrm{w_{ij}} - \mathrm{w_{ij}}\, \delta_{\mathrm{ij}}/\mathrm{d_{ij}}$$

5.2. Algorithm

The proposed algorithm consists of follow steps:

1. Each node has $\delta_{\mathrm{ij}}(\mathrm{j} = 1..\mathrm{m})$ and positions of reference node than estimate initial position by following formula:

$$x_i = \sum_{j=1}^{m} ref_j \times w_{ij} / \sum_{j=1}^{m} w_{ij} \qquad (18)$$

Compute w_{ij} from equation (14).

2. Each node broad cast its estimated position to neighbors which $\delta_{ij} < radius.$

3. Node i calculate local cost function from equation(13) and s_1=s_i.

4. Node i calculate its position from equation (15).

5. Node i calculate local cost function from equation(13) and s_2=s_i.

6. If s_2<s_1 then x_i replace in i_{th} row X^0 and goto 4.

7. End.

6. PERFORMANCE EVALUATION

In our experiments, we ran Proposed Algorithm on various topologies of networks in matlab. We considered two network topologies. The first one is the random uniform topology, and the second is the randomly with a C-shape distribution topology. for both topology the connectivity and average number of neighboring nodes is controlled by setting the communication range radius from 1r to 1.5r where r=1 unit length.

For comparison purposes, the measured distance between neighboring nodes δ_{ij} is modeled with a noise drawn from a normally-distributed random variable with mean 0 and variance of 1 which duplicates the condition in [11].

$$\delta_{ij} = d_{ij} \times (1 + N(0, er)) \qquad (19)$$

Where d_{ij} the true is measured distance and is the er ranging error. The performance measure we employed is the Root Mean Square Error (RMSE) between the true location of sensor nodes and the estimated location of sensor nodes normalized to R.

6.1. Random Placement

In this experiment, 196 nodes were randomly placed in a 10r×10r square area. Four reference nodes are placed in the corners of square area. The communication rang R was set to 1r and 1.5r. Simulations were done with a 5% distance error measurement and 30 trials for each data points. The average of RMS error for the proposed algorithm was for radius 1 and 1.5 are 0.65r and 0.63r respectively whereas the dwMDS has average RMSE 0.85r and 0.77r. Fig. 2 shows the location estimates by the proposed algorithm on the before mentioned data points.

6.2. randomly with a C-shape distribution Placement

In this experiment, 156 nodes were randomly with a C-shape distribution placed in a 10r×10r square area. Four reference nodes are placed in the corners of square area. The communication rang R was set to 1r and 1.5r. Simulations were done with a 5% distance error measurement and 20 trials for each data points. The average of RMS error for the proposed algorithm was for radius 1 and 1.5 are 0.66r and 0.64r respectively whereas the dwMDS has average RMSE 079r and 0.72r. Fig. 3 shows the location estimates by the proposed algorithm on the before mentioned data points.

7. CONCLUSION

Multidimensional scaling uses dissimilarity between all pairs of points because computing the singular value decomposition requires that all dissimilarities are gathered centrally somewhere. This takes $O(n^3)$ time, n number of points.

dwMDS presented is a scalable decentralized algorithm also hasn't to estimate distance between out of range sensors. This is enabled by a weighted cost function that more heavily weighs measurements believed to be more accurate. The weight is zero do not take part in a local sensors processing.

A drawback of dwMDS is that complexity, convergence time and initial estimate requirements. Each sensor must hold k Range measurement and weight between itself and others. in each step every sensor transmit its estimated position and local stress function to the neighbors that makes high communication cost.

New proposed algorithm is a scalable decentralized algorithm also hasn't to estimate distance between out of range sensors. This is enabled by a fuzzy weighted cost function that more heavily weighs measurements believed to be more accurate. If a sensor moves, only the moved sensor will have to recalculate its location locally rather than all sensors are involved. Also the proposed algorithm reduces communication cost, complexity, and convergence time.

A drawback of proposed algorithm that is it requires sufficient anchor nodes (3 or more for 2-D networks, 4 or more for 3-D networks).

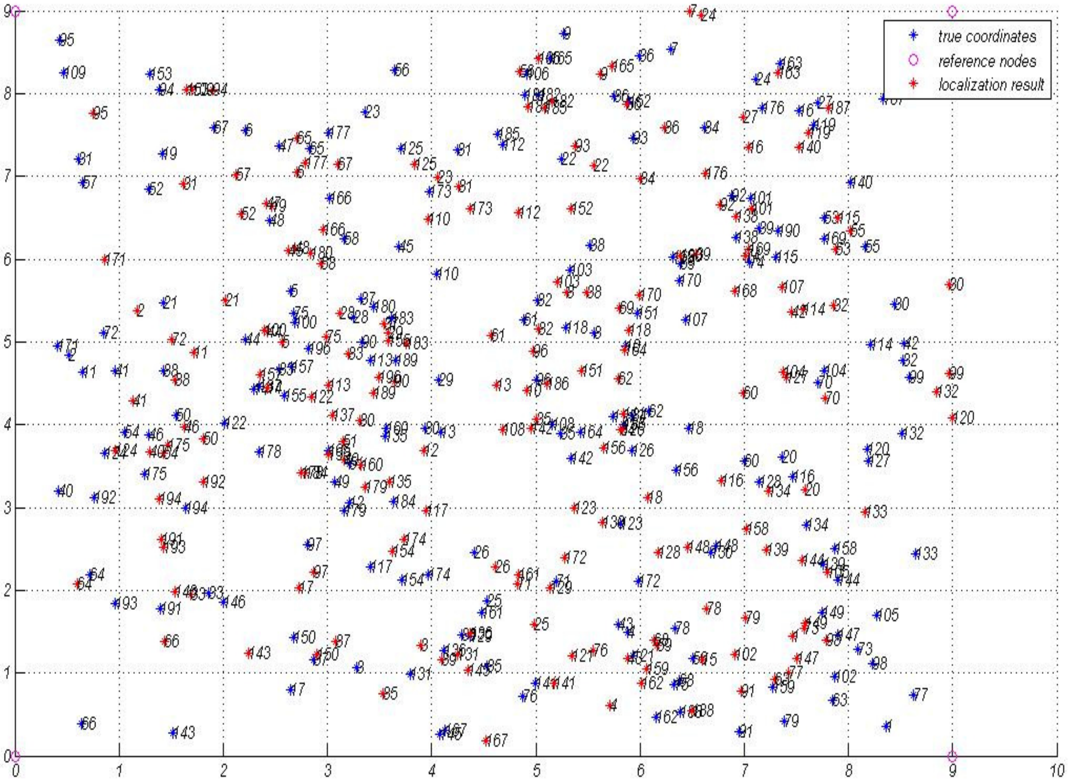


Figure 2. Position estimates for uniform random placement when radius is 1.5r

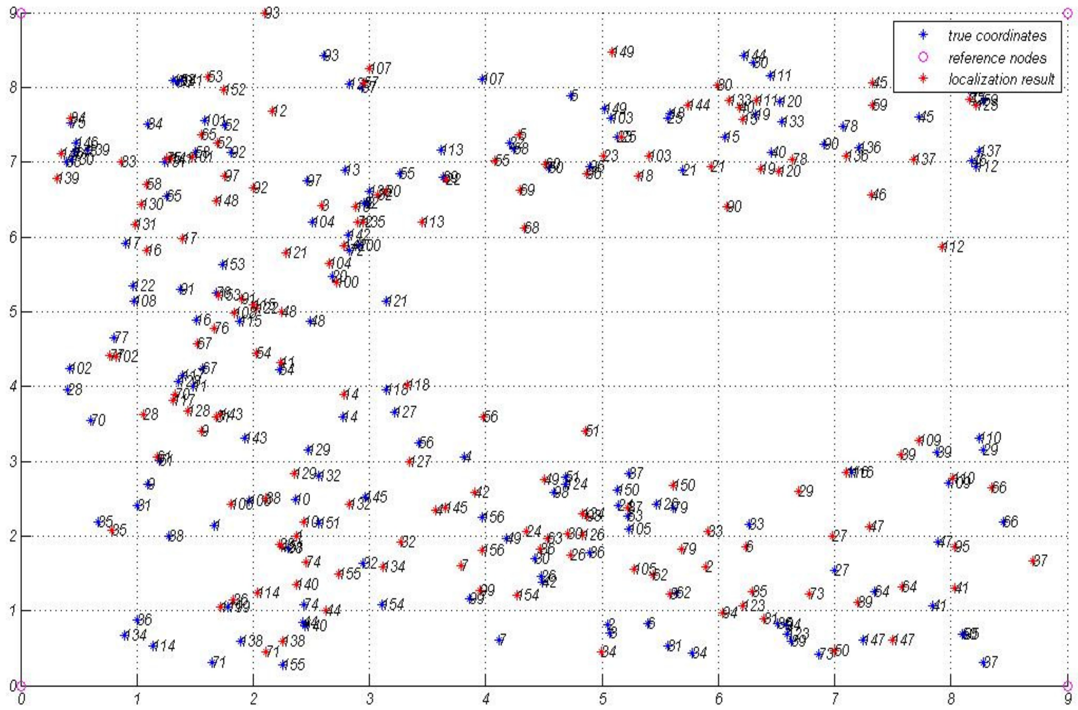


Figure 3. Position estimates for c-shape distribution random placement when radius is 1.5r

REFERENCES

[1] P. Radha, W. Cliff, and R. Sumit, Secure Localization and Time Synchronization for Wireless Sensor an Ad Hoc Networks, Springer, 2007.

[2] C. S. Raghavendra, K. M. Sivalingam , and T. Znati, Wireless Sensor Network, Kluwer Academic Publishers, 2004.

[3] W. S. Torgerson, Theory and Methods of Scaling, John Wiley & Sons, Inc. , New York, 1958.

[4] M. Bal, M. Liu, W. Shen, and H. Ghenniwa, "Localization in Cooperative Wireless Sensor Networks: A Review," Proceedings of the 13th International. Conference on Computer Supported Cooperative Work in Design, 2009.

[5] G. B. E. Carlos, L. G. Sison, "ALESSA: MDS-based Localization Algorithm for Wireless Sensor Networks," IEEE, vol. 2, pp. 856 – 859, 2009.

[6] G. Yu and S. Wang, “A Hierarchical MDS-based Localization Algorithm for Wireless Sensor Networks,” in 22nd IEEE International Conference on Advanced Information Networking and Applications. ICAINA, 2008.

[7] J. A. Costa, N. Patwari, and A. O. Hero, "Distributed Weighted-Multidimensional Scaling for Node Localization in Sensor Networks," ACM Transactions on Sensor Networks, vol. 2, no. 1, pp. 39–64, February 2006.

[8] V. Vivekanandam, and V. Wong, "Ordinal MDS-based Localization for Wireless Sensor Netwoks," in 64th IEEE Vehicular Technology Conference, pp. 1–5, 2006.

[9] J. A. Costa, N. Patwari , and A. O. Hero, "Achieving High-Accuracy Distributed Localization in Sensor Networks," IEEE , 2005.

[10] Y. Shang, W. Ruml, "Improved MDS-Based Localization," in Proc. IEEE Infocom, Hong Kong, China, pp. 2640-2651, March 2004.

[11] Y. Shang, W. Ruml, Y. Zhang and M. Fromherz, "Localization from mere connectivity," in ACM MobiHoc, Annapolis, MD, pp. 201–212, June 2003.

[12] F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," IEEE Communications Magazine, vol. 40, no. 8, pp. 102-114, Aug 2002.

Fahimeh Doremami received the B.S. degree in Software Engineering from Shariaty University in 2007. She is a M.S. student in Software Engineering at Payame Noor University (PNU) now. Her research interests are ad hoc network technologies, sensor network technology, image processing, distributed operating systems, and heuristic Algorithm.

Hamid Haj Seyyed Javadi received the B.S., M.S. and ph.D degrees in Amir kabir University. He has been working as a full-time faculty member and Assistant Professor of Shahed University. His research interests are ad-hoc network technologies, sensor network technology, distributed operating systems, and heuristic Algorithm.

Ahmad Farahi received the pd.D. degree in Computer Sciences from Bradford University, He has been working as a full-time faculty member, Assistant Professor of and Head of Computer and Information Technology Department in the Payame Noor University. His research interests are programming especially that of educational systems.