Sequential Multidimensional Scaling For Realtime Sensor Network Localization

Lan Anh Trinh
Electronics Engineering Department, Post and Telecommunication Institute of Technology-HCM, Vietnam
lananhtrinh@ptithcm.edu.vn

Nguyen Duc Thang*
Biomedical Engineering Department, International University
National Universities-HCM, Vietnam
ndthang@hcmiu.edu.vn

Nguyen Luong Nhat
Electronics Engineering Department, Post and Telecommunication Institute of Technology-HCM, Vietnam
nhatnl@ptithcm.edu.vn

Tran Cong Hung
Science Technology Department, Post and Telecommunication Institute of Technology-HCM, Vietnam
conghung@ptithcm.edu.vn

Hoang-Hai Tran
School of Information and Communication Technology-Hanoi University of Science and Technology
thhai29@yahoo.com

***Abstract*— We investigate on the localization of nodes in a sensor network based on multidimensional scaling (MDS). The distances of node pairs are given and MDS attempts to locate the position of nodes given the distance matrix. However, conventional MDS addresses the mapping problem using eigenvector decomposition which is complicated in computation, mitigating the efficiency of this approach for real-time applications. In this paper, we introduce a sequential MDS for the fast implementations of MDS for a changing sensor network. The fast-fixed point algorithm is used to initialize the locations of nodes. When the distance matrix is varied due to the movements of nodes, sequential MDS aims at reallocating the nodes with a fast and effective update process. We validated sequential MDS with a number of experiments and shown the proposed approach is superior to conventional MDS when it deals with the location problems of moving nodes of the sensor network.

Keywords-Multidimensional Scaling; Localization; Sequential Eigenvector Decomposition.

I. INTRODUCTION

In recent years, the development of radio and embedded systems has enabled the proliferation of wireless sensor networks to be tremendously applied in fields of monitoring tasks such as intruder detection, rescue, disaster relief, target tracking and other tasks within smart environments. Besides, advances in micro-electromechanical technology, computing and communication have motivated the emergence of massively distributed wireless sensor networks which consist of hundreds and thousands of nodes. Each node in network is able to sense surrounding environments, to perform computations and to communicate with each other to accomplish a task. Particularly, localization plays a crucial role in a wireless sensor network when the position of node is not provided in advance. The node localization is needed to report the origin of events in network, to assist group querying of sensors and geographical routing and to address network coverage. Therefore, node localization is one of essential problems in wireless sensor network.

A great deal of efforts over the past few years have investigated on the node localization problem. Existing approaches discussing on localization can be found in [1]. Conventional localization techniques estimate the distances between two nodes using received signal strength indicator (RSSI), time of arrival (TOA), time difference of arrival (TDOA) and so on. However, such methods needed extra setup with direction of arrival (DOA) estimation to fully locate the position of sensor nodes. An innovated and effective method to recover the localization maps of every sensor nodes using only distance information in a network copes with the multidimensional scaling (MDS) algorithm. The MDS algorithm was originated from the 1960s [2] and over the last ten years, MDS algorithms have been proposed for the network localization. For instance, Agrawal et al. [3] took advantage of MDS algorithm for localization in wireless sensor network that consists of a large number of sensor nodes. The authors analyzed their approaches with simulation considering numerous cases and suggested how to address shortcomings caused by anisotropic network topology and complex terrain. The simulated results showed that the algorithm is effective, yet the analysis of the communication

* Corresponding author: ndthang@hcmiu.edu.vn

costs, messaging complexity and power consumption were lack to show how MDS is applicable in practical situations. In [4], MDS-MAP algorithm has been introduced to obtain the sensor location on a unit square within a given radio range. The results reported that the position of node is estimated only when local connectivity information is known. Actually, the original MDS algorithm is not suitable for large scale networks since it requires $O(N^3)$ complexity for the task of eigenvector decomposition.

Recently, researchers have investigated more effective MDS for the localization problems. Changhua et al. [5] proposed a dynamic mobility-assisted MDS localization approach for a mobile sensor network. This method highly depends on the degree of the network; therefore, a number of virtual nodes are added to acquire precise location leading to more data transferred over a network and increasing computational cost. In [6], classical MDS was combined with neural network but it requires additional hardware and huge temporary memory for computations. Hierarchical MDS (HDMS) [7], ordinal MDS [8], universal MDS [9], improved MDS [10], deterministic annealing MDS [11] have been other approaches developed to improve the MDS-MAP algorithm with respect to improving the accuracy of localization computation. Nevertheless, the algorithms are highly computational.

In this paper, we develop a sequential MDS for a real-time localization so that it can be applicable for widely practical applications. Conventional MDS finds an embedding space to preserve the distances of input nodes in a higher dimensional space. Since MDS algorithm running on and *N*-by-*N* pair-wise distance matrix where *N* is the number of nodes, the disadvantage of MDS relates to the eigenvector-decomposition process when all eigenvectors are estimated, leading to the limited ability of MDS for real-time applications.

The aim of this paper is to cope with the real time localization in the wireless sensor networks using MDS. We propose the fast-fixed point algorithm to initialize the first *k*-eigenvectors of the pair-wise distance matrix. The second concern is how to update the location of nodes with regards to small changes in their position. To avoid the process of recalculating the eigenvectors, we propose a procedure with sequential MDS to update the eigenvectors with respect to small changes of the pair-wise distance matrix. We present a number of simulated experiments to illustrate and verify our propose model. The remaining sections are organized as follows. In Section 2 we describe the proposed algorithm. In Section 3 we show experimental results to compare the proposed algorithms with conventional MDS. Finally, conclusions and discussion are presented in Section 4.

II. Algorithm

A. Classical MDS

The objective of MDS [3,12] is to determine a position of nodes in multidimensional space as a best fit to the similarity measured by a linear transformation. The algorithm supposes that there are *N*-sensor nodes in the network. Let d_{ij} be the dissimilarity measure between *i*-th and *j*-th nodes, and $\boldsymbol{D}$ be the $N \times N$ distance matrix. Here, the Euclidean distance between two points $p_i=(p_{i1}, p_{i2}, \dots, p_{in})$ and $p_j=(p_{j1}, p_{j2}, \dots, p_{jn})$ is given by

$$d_{ij} = \|p_i - p_j\| = \sqrt{\sum_{r=1}^{m} (p_{ir} - p_{jr})^2} \tag{1}$$

where *r* is the value of dimension and *n* is the number of dimension. The target of classical MDS is to find an assignment of $\boldsymbol{P} = [p_1, p_2, \dots, p_N]$ in *n*-dimensional space that minimizes a stress function. The stress function measuring the degree of inaccuracy is defined as

$$Stress(\boldsymbol{P}) = \frac{1}{2}\sum_{i}^{N}\sum_{j}^{N}(d_{ij} - \|p_i - p_j\|)^2. \qquad \widehat{\boldsymbol{P}} = \mathrm{argmin}_{\boldsymbol{P}} Stress(P) \tag{2}$$

In classical MDS, the elements of $\widehat{\boldsymbol{P}}$ are calculated by using eigenvalue decomposition of the double centered squared distance matrix. The double centered squared distance matrix is denoted by a matrix $\mathbf{B}$ as $\mathbf{B} = -\frac{1}{2}\mathbf{JDJ}$, where $\boldsymbol{J}$ is a centering operator and computed as $\boldsymbol{J}=\boldsymbol{I}-(1/N)ee^T$ and *e* is vector of all ones. The elements of matrix B is defined as

$$b_{ij} = -\frac{1}{2}\left(d_{ij}^2 - \frac{1}{m}\sum_{r=1}^{m} d_{rj}^2 - \frac{1}{m}\sum_{r=1}^{m} d_{ir}^2 + \frac{1}{m^2}\sum_{r=1}^{m}\sum_{s=1}^{m} d_{rs}^2\right). \tag{3}$$

The matrix $\widehat{\boldsymbol{P}}$ is obtained by using matrixes $\boldsymbol{U}$ and $\boldsymbol{V}$ which are the results of performing the singular decomposition (SVD) to the matrix $\boldsymbol{B}$

$$\boldsymbol{B} = \boldsymbol{U}\boldsymbol{V}\boldsymbol{U}^T = \widehat{\boldsymbol{P}}^T\widehat{\boldsymbol{P}} \qquad \widehat{\boldsymbol{P}} = \boldsymbol{U}\boldsymbol{V}^{1/2} \tag{1}$$

where $\boldsymbol{U}$ and $\boldsymbol{V}$ contains all eigenvectors and eigenvalues of the double centered squared distance matrix $\mathbf{B}$.

B. Our Proposed Sequential MDS (sMDS)

1) Fast-fixed points algorithm to initialize the locations of nodes

In order to estimate the eigenvector of a matrix *B* with the size $N \times N$, solving an exact solution is infeasible with $N \geq 5$ since it requires to find *N* roots of the polynomial of degree *N*. Numerical methods cost $O(N^3)$ computations with QR decomposition [13,14] and therefore recovering the whole eigenvectors for $\boldsymbol{B}$ is time consuming.

TABLE I: INITIALIZE THE NODE POSITIONS USING FAST FIXED POINT ALGORITHM

```
To compute coordinates of N nodes in n-dimensional space from their pair-wise distances:
   1. Construct N×N squared-distance matrix D
   2. Compute inner product matrix B = -1/2 JDJ,
      where J=I-(1/n)ee^T is double-centering matrix
   3. Initialize eigenvector w_p of size n×1 randomly
   4. Update w_p as w_p ← Bw_p
   5. Do the Gram-Schmidt orthogonalization process w_p ← w_p − Σ_{j=1}^{p−1} (w_p^T w_j) w_j
   6. Normalize w_p
   7. If w_p has not converged, go back to step 4
   8. Increment counter p ← p + 1 and go to step 3 until p equals n
```

In this work, we formulate the methods to estimate the largest eigenvector as an optimization problem

$$\begin{aligned} &\underset{w}{\text{minimize}} \quad w^T \boldsymbol{B} w \\ &\text{subject to } \|w\| = 1. \end{aligned} \tag{4}$$

The optimization problem described in Eq. (2) is addressed by the fast-fixed point algorithm with normalization at each iteration of update

$$w \leftarrow \boldsymbol{B} w. \tag{5}$$

Here, the eigenvector is normalized after recalculated $w \leftarrow w/\|w\|$. The p-th eigenvector can be computed in a similar way while the Gram-Schmidt process is applied to make it orthogonal with the previous $1^{st}, 2^{nd}, \ldots, p\text{-}1^{th}$ eigenvectors

$$\begin{aligned} w_p &\leftarrow \boldsymbol{B} w_p \\ w_p &\leftarrow w_p - \sum_{j=1}^{p-1} (w_p^T w_j) w_j. \end{aligned} \tag{6}$$

Let $[b_1^T, b_2^T, \ldots, b_N^T]$ be the rows of the $\boldsymbol{B}$ matrix, and the elements of the eigenvector w_p are $[w_p^1, w_p^2, \ldots, w_p^N]$, then an equation to compute the eigenvalue is expressed by,

$$\lambda_p \leftarrow \frac{\sum_{j=1}^{N} (b_j^T w_p)/w_p^j}{N}. \tag{7}$$

Usually, the locations of nodes is determined within 2-D or 3-D space, the maximum number of eigenvectors is thereby less than or equal three. The incremental updates illustrated in (2) and (3) usually takes 5-10 steps to converge. The fast-fixed point algorithm for MDS is detailed in Table I and requires about $O(N^2)$ computations.

2) Sequential MDS (sMDS)

Sequential MDS deals with the localization of a dynamic sensor network when parts of its sensor nodes are moving. Assume that the whole system is not moving rapidly so that the variation of the pair-wise distance matrix of every nodes is considered small. In this case, it is unnecessary to perform the whole MDS procedure to reallocate the new positions of sensor nodes. The localization is estimated directly from the distance derivation between nodes. Our proposed sequential MDS is detailed as follows.

Let $\delta\boldsymbol{B}_t$ be the small variation of the double centered square distance matrix $\boldsymbol{B}_t$ at the time index t. Assume that n eigenvector of the matrix B_t has been computed. We have

$$\begin{aligned} \boldsymbol{B}_{t+1} &= \boldsymbol{B}_t + \delta\boldsymbol{B}_t \\ \boldsymbol{B}_t w_t^i &= \lambda_t^i w_t^i \end{aligned} \tag{8}$$

where λ_t^i is the eigenvalue and $i=1,2,\ldots,n$. Our aim is to find the eigenvectors $w_{t+1}^i = w_t^i + \delta w_t^i$ and eigenvalues $\lambda_{t+1}^i = \lambda_t^i + \delta\lambda_t^i$ with regards to small changes $\delta\mathbf{B}_t$ of $\mathbf{B}_t$. It is equivalent to solving an equation

$$\boldsymbol{B}_{t+1} w_{t+1}^i = \lambda_{t+1}^i w_{t+1}^i \tag{9}$$

for all $i=1,2,\ldots,n$ or

$$(\boldsymbol{B}_t + \delta\boldsymbol{B}_t)(w_t^i + \delta w_t^i) = (\lambda_t^i + \delta\lambda_t^i)(w_t^i + \delta w_t^i) \tag{10}$$

Remove the high-order terms in (10) and note that $\boldsymbol{B}_t w_t^i = \lambda_t^i w_t^i$, we get

$$\delta\boldsymbol{B}_t w_t^i + \boldsymbol{B}_t \delta w_t^i = \lambda_t^i \delta w_t^i + \delta\lambda_t^i w_t^i \tag{11}$$

Use the set w_t^i as the basic to expand δw_t^i as

$$\delta w_t^i = \sum_{j=1}^{n} \varepsilon_{ij} w_t^j \tag{12}$$

Substitute (12) into (11), this leads to

$$\delta\boldsymbol{B}_t w_t^i + \sum_{j=1}^{n} \varepsilon_{ij} \boldsymbol{B}_t w_t^j = \sum_{j=1}^{n} \varepsilon_{ij} \lambda_t^i w_t^j + \delta\lambda_t^i w_t^i \tag{13}$$

or

$$\delta\boldsymbol{B}_t w_t^i + \sum_{j=1}^{n} \varepsilon_{ij} \lambda_t^j w_t^j = \sum_{j=1}^{n} \varepsilon_{ij} \lambda_t^i w_t^j + \delta\lambda_t^i w_t^i \tag{14}$$

Left-multiply both sides of (14) with ${w_t^i}^T$ and apply the properties of eigenvectors

$${w_t^i}^T w_t^j = \begin{cases} 1 \text{ if } i = j \\ 0 \text{ if } i \neq j \end{cases}, \tag{15}$$

we have

$$\delta\lambda_t^i = {w_t^i}^T \delta\boldsymbol{B}_t w_t^i. \tag{16}$$

By left-multiply ${w_t^j}^T$ with $\neq i$, we obtain

$$\varepsilon_{ij} = \frac{w_t^{j^T} \delta \boldsymbol{B}_t w_t^i}{\lambda_t^i - \lambda_t^j}. \quad (17)$$

Finally, in order to estimate ε_{ii}, note that $\|w_{t+1}^i\| = 1$, the value of ε_{ii} should be zero.

In overall, the proposed sequential MDS use a set of rules to update eigenvectors w_t^i and eigenvalues λ_t^i correspondent to a small variation of the double centered square matrix as follows

$$\lambda_{t+1}^i = \lambda_t^i + w_t^{i^T} \delta \boldsymbol{B}_t w_t^i$$

$$w_{t+1}^i = w_t^i + \sum_{\substack{j=1 \\ j\neq i}}^{n} \frac{w_t^{j^T} \delta \boldsymbol{B}_t w_t^i}{\lambda_t^i - \lambda_t^j} w_t^j. \quad (18)$$

3) Re-initialize the locations of nodes

The set of nodes recovered by MDS (or sMDS) receive a rigid transformation (rotation, reflection, and translation) of the original nodes. In order to complete the localization tasks, some location of nodes should be known in advance, namely beacons. Let $\boldsymbol{X_A}$ and $\boldsymbol{X_B}$ be the coordinates of beacons and their corresponding estimated nodes. The registration process presented in the following session will align the two set of points *A* and *B*. The new coordinate of *B* will be $\widehat{\boldsymbol{X}_B}$.

After an iteration of update, we evaluate again the differences between the beacons and their corresponding nodes given by $\epsilon = \| \widehat{\boldsymbol{X}_B} - \boldsymbol{X}_A \|$. If ϵ is larger than some threshold, we perform the fast-fixed point algorithm to reinitialize the locations of nodes.

C. Registration of Localization Results

In order align the estimated nodes with some know location of beacons, we perform the registration by compute the parameters of rigid transformation including the transformation and the rotation/reflection matrix.

The rotation/reflection matrix **R** is computed to minimize the term

$$J(\boldsymbol{R}) = \|\boldsymbol{X}_A - \boldsymbol{R}\widehat{\boldsymbol{X}_B}\|^2. \quad (19)$$

The solution for this problem is acquired by a singular value decomposition (SVD) of matrix $\boldsymbol{X}_A \widehat{\boldsymbol{X}}_B^{\;T} = \boldsymbol{U}_{AB} \boldsymbol{\Lambda}_{AB} \boldsymbol{V}^T{}_{AB}$,

$$\boldsymbol{R} = \boldsymbol{U}_{AB} \boldsymbol{V}^T{}_{AB}. \quad (20)$$

With regards to transformation, the whole set of nodes $\boldsymbol{Y}_B$ is registered to the whole set $\boldsymbol{Y}_A$

$$\boldsymbol{Y}_B = \boldsymbol{R}(\boldsymbol{Y}_B - \mu_A) + \mu_B \quad (21)$$

where μ_A and μ_B are the centers of the two set *A* and *B* respectively.

III. Experimental Results

We evaluate our proposed approaches with simulated experiments. The nodes are randomly placed in a C-shaped area or randomly uniform in a $10m \times 10m$ square as illustrated in Fig. 1. Four beacons with known location have been used. Because the whole matrix distance between one nodes and other are hard to be estimated precisely, we add zero Gaussian noises with variance $0.1m$ into the pair-wise distance matrix. In real implementation, dependent on signal ranges, each sensor node is able to obtain distance to its neighbors only. In such cases, a connected network is established among all sensor nodes and the pair-wise Euclidean distance matrix is approximated by a pair-wise geodesic distance matrix in which the geodesic distance is the lengths of the shortest path between two nodes in the connected network [4]. This paper copes with the method to fast update embedded space of MDS when the location of nodes is varied sequentially, therefore the pair-wise distance matrix is assumed to be given.

A. Comparision of MDS with Fast-fixed Poin Algorithm to Initialize the Location of Nodes

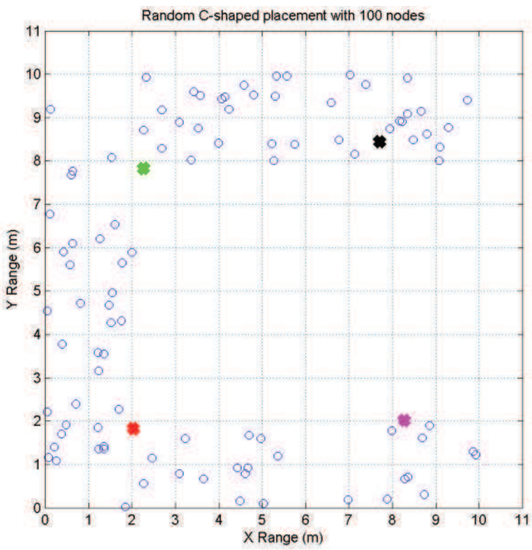
As mentioned in Session II, our proposed sMDS utilizes the fast-fixed point algorithm to initialize the locations of nodes. The fast-fixed point algorithm helps us to compute the largest eigenvectors of the double centered squared distance matrix rather than to apply SVD to recover the whole eigenvector space of the double centered squared distance matrix as in conventional MDS. The computational times and the root mean square errors (RMSE) obtained by conventional MDS and MDS using fast-fixed point algorithm to locate the positions of 100, 200, 500, and 1000 uniform distributed nodes and C-shaped distributed nodes are given in Table II and Table III respectively.

TABLE II. COMPARISONS BETWEEN CONVENTIONAL MDS AND MDS USING FAST-FIXED POINT ALGORITHM ON UNIFORM DISTRIBUTED NODES.

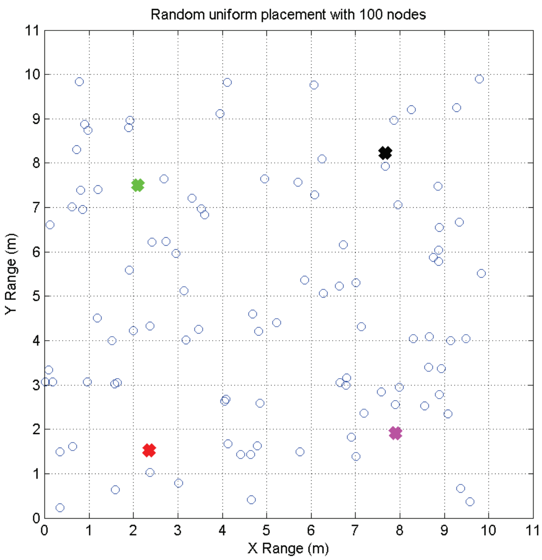
Number of nodes	**Conventional MDS**		**MDS using fast-fixed point algorithm**	
	Computational time(s)	***RMSE (m)***	***Computational time(s)***	***RMSE (m)***
100	0.02	0.03	<0.01	0.03
200	0.09	0.02	<0.01	0.03
500	0.80	0.02	0.01	0.02
1000	11.00	0.01	0.08	0.02

TABLE III. COMPARISONS BETWEEN CONVENTIONAL MDS AND MDS USING FAST-FIXED POINT ALGORITHM ON C-SHAPED DISTRIBUTED NODES.

Number of nodes	**Conventional MDS**		**MDS using fast-fixed point algorithm**	
	Computational time(s)	***RMSE (m)***	***Computational time(s)***	***RMSE (m)***
100	0.03	0.03	<0.01	0.04
200	0.08	0.02	<0.01	0.03
500	1.17	0.02	<0.01	0.03
1000	11.40	0.01	0.03	0.01



(a)



(b)

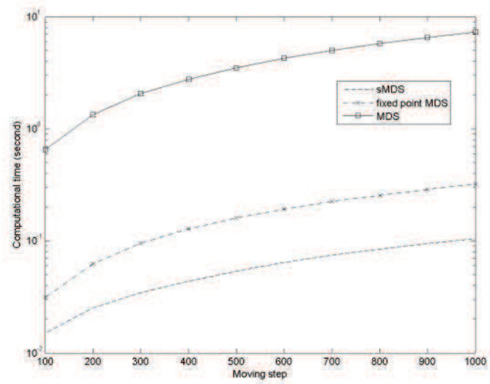
Figure 1. Example of 100 sensor nodes distributed in a $10m \times 10m$ square with four beacons noted by *'X'* markers with different colors. (a) Random C-shaped replacement and (b) Random uniform replacement of nodes.

As we can see in Table II and Table III, the computational time of conventional MDS is rapidly raising with increasing the number of nodes. Meanwhile, the computational time of MDS using fast-fixed point algorithm is less than *0.1s* even with 1000 sensor nodes.

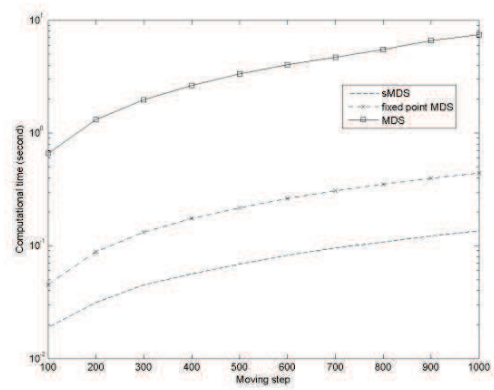
B. Comparisons between sMDS and Conventional MDS on Localizing the Posions of Moving Nodes

This section concerns the performance of sMDS on moving sensors nodes. We randomly locate 100 sensor nodes with uniform and C-shaped distribution within an area of $10m \times 10m$ square. Then, at reach running step, we add an increment (random uniform distribution) within a range *[0cm-5cm]* for both *x*- and *y*-coordinates of nodes. The sMDS aims at updating the location of nodes with regards to the small changes of the sensor network. We compare sMDS, MDS using fast-fixed point algorithm and conventional MDS to address the localization problems on the aforementioned configuration. Here, MDS using fast-fixed point algorithm means that we apply fast-fixed-point algorithm described in Section B.1 for the eigenvector decomposition to decrease the number of estimated eigenvector. Experimental results depicted in Fig. 2 reveals that sMDS is faster than both MDS using fast-fixed point algorithm and conventional MDS.

In Table IV, the differences of RMSE between sMDS, MDS using fast-fixed point and conventional MDS are less than $2cm$, that is considered small when compared with the distributed range of $10m$ of nodes. Meanwhile, after 1000 moving steps, the computational time of sMDS is just about 1/3 of MDS using fast-fixed point algorithm and about 1/70 of conventional MDS.

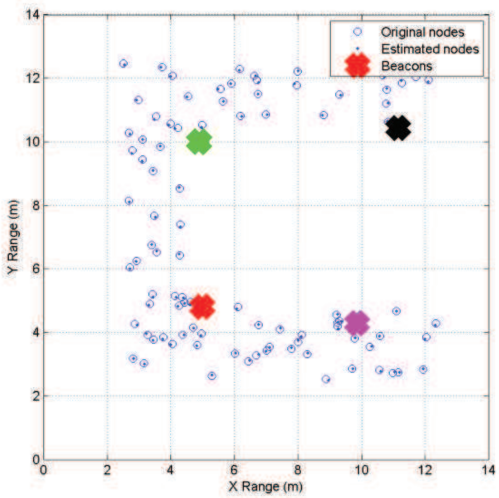


(a) Random C-Shaped placement

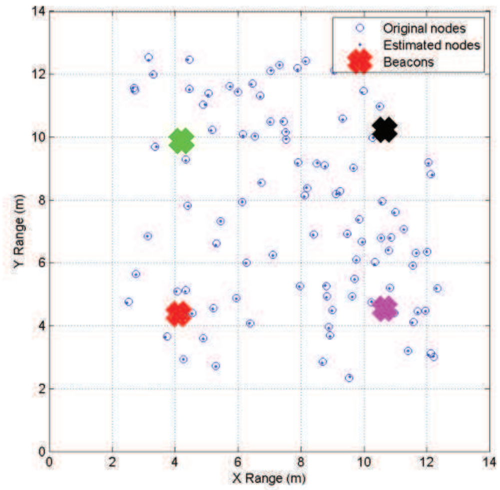


(b) Random uniform placement

Figure 2. Comparisons of sMDS, MDS using fixed point algorithm and conventional to locate the positions of 100 moving nodes.



(a)



(b)

Figure 3. The localization nodes found by sMDS after 100 moving steps. (a) Random C-shaped replacement and (b) Random uniform replacement.

TABLE IV. COMPARISONS OF AVERAGE RSME ACQUIRED BY SMDS, MDS USING FAST-FIXED POINT ALGORITHM AND CONVETIONAL MDS FOR THE LOCALIZATION OF 100 NODES OVER 1000 MOVING STEPS.

	sMDS	Fixed-point MDS	MDS
Random uniform	0.05	0.03	0.03
Random C-shaped	0.05	0.04	0.03

Figure 3 shows the node placements found by sMDS after 100 moving steps in which the nodes are shifted to the right-top corner due to increasing node coordinates along the *x*- and *y*-axes. The estimated nodes are close to the original.

IV. CONCLUSIONS AND DISCUSSION

This paper introduces a sequential MDS for the localization of nodes when the distances among nodes are given. Conventional MDS applies the eigenvector decomposition to find an arrangement of nodes to preserve the pair-wise distance among them. With small derivation of the pair-wise distance when the nodes are moving, our sequential MDS reallocates the nodes using an effective update without utilizing the whole eigenvector decomposition as in conventional MDS. Therefore, our proposed is fast when compared with conventional MDs but still ensures the small estimation errors. In future work, we plan to integrate our proposed sMDS with tracking algorithm to better follow moving nodes of a dynamic sensor network.

REFERENCES

[1] I.Stojmenovic. Localization in Sensor Networks. Handbook of Sensor Networks: Akgorithms and Architectures, Wiley , 2005.

[2] T. Cox, M. Cox. Multidimensional Scaling. Number 59 in Monographs on Statistics and Applied Probability. Chapman and Hall, 1994.

[3] R. Agrawal, B. Patel, “Localization in Wireless Sensor Network Using MDS,” International Journal of Smart and Ad Hoc Networks, vol. 1, pp. 26-31, 2012.

[4] S. Oh, A. Montanari, A. Karbasi, “Sensor Network Localization from Local Connectivity: Performance Analysis for the MDS-MAP Algorithm”, IEEE Information Theory Workshop, 2010.

[5] C. Wu, W. Sheng, W.-Z. Song, “A Dynamic MDS-Based Localization Algorithm for Mobile Sensor Networks”, Proceeding of IEEE International Conference on Robotics and Biomimetics, ROBIO 2006.

[6] U. Singh, M. Kumar Jha, “Localization Using Classical MDS & NN Implementation”, International Journal of Engineering Research & Technology, vol.1, 2012.

[7] G.-J. Yu, S.-C. Wang, “A Hierarchical MDS-based Localization Algorithm for Wireless Sensor Networks”, Journal of Information Technology and Applications, vol. 2, pp. 197-202, 2008.

[8] V. Vivekanandan, V. Wong, “Ordinal MDS-based Localization for Wireless Sensor Networks”, International Journal of Sensor Networks, vol. 1, pp. 169-178, 2006.

[9] A. Agarwal, J. M. Phillips, S. Venkatasubramanian, “Universal Multi-Dimensional Scaling”. Proceeding KDD ’10 proceeding of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2010.

[10] Y. Shang, W. Ruml, “Improved MDS-Based Localization”, Proceeding of INFOCOM 2004, Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies, 2004.

[11] S.-H. Bae, J. Qiu, G. C. Fox, “Multidimensional Scaling by Deterministics Annealing with Iterative Majorization Algorithm”. Proceeding ESCIENCE ’10 Proceedings of the 2010 IEEE Sixth International Conference on e-Science, 2010.

[12] J. C. Platt, “FastMap, MetricMap, and Landmark MDS are all Nyström Algorithm”. In proceeding of Microsoft Research, 2005.

[13] J. G. F. Francis, "The QR Transformation, II (part 2)", The Computer Journal, vol. 4, pp. 332–345, 1962.

[14] J. G. F. Francis, "The QR Transformation, I (part 1)", The Computer Journal, vol. 4, pp. 265–271, 1961.