2007 IEEE International Conference on
Robotics and Automation
Roma, Italy, 10-14 April 2007

Mobile Sensor Networks Self Localization based on Multi-dimensional Scaling

Chang-Hua Wu, Weihua Sheng and Ying Zhang

Abstract— **In this paper, we define a mobile self-localization (MSL) problem for sparse mobile sensor networks, and propose an algorithm named Mobility Assisted MDS-MAP(P), based on Multi-dimensional Scaling (MDS) for solving the problem. For sparse sensor networks, all the existing localization algorithms fail to work properly due to the lack of distance or connectivity data to uniquely calculate the geo-locations. In MSL, we use mobile sensors to add extra distance constraints to a sparse network, by moving the mobile sensors in the area of deployment and recording distances to neighbors at some intermediate locations. MSL can also be used for localizing and tracking mobile objects in a robotic or body sensor network. Experiments and evaluations of the new algorithm are provided.**

Index Terms— **Sensor networks, MDS-MAP, Mobile self-localization**

I. Introduction

Recent advancements in wireless communication and micro-electro-mechanical systems (MEMS) have made possible the deployment of wireless sensor networks for many real world applications, such as environmental monitoring, search and rescue, military surveillance, and intelligent transportation, etc [1], [2], [3]. The ability of a sensor node to determine its geographical location is of fundamental importance in sensor networks.

Most of the localization algorithms are developed for stationary sensor networks where the sensor nodes do not move once they are deployed. Recent years have seen the growing interest in mobile sensor networks [4] where all or partial of the sensor nodes have motion capability endowed by robotic platforms. Mobile *actuated* sensor networks have more flexibility, adaptivity and even intelligence compared with stationary sensor networks. Tracking and self-localizing various types of moving objects become an important research topic. Various work has been done on solving localization, tracking and mapping problems for mobile robots in robotics, which heavily relies on the sophisticated sensors such as sonar, laser ranger finder, or camera onboard the mobile platforms [5]. However, most of these mobile sensors have very stringent constraints on the cost and complexity. To the best of our knowledge, only very limited work has been done on mobile sensor network self-localization. Tilak et al. [6] developed dynamic localization protocols for mobile sensor networks. However, their main interest is on how often the localization should be carried out in a mobile sensor network and not on the localization method itself. Recently, Hu and Evans [7] proposed sequential Monte Carlo (SMC) localization method to solve the localization problem and they found that the mobility of the sensors can be exploited to improve the accuracy and precision of the localization.

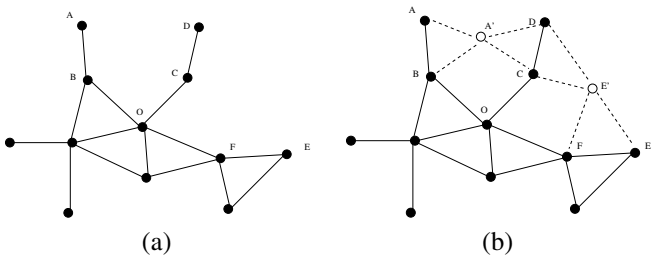
Using mobile nodes to assist self-localization of a sparse sensor network is a new research direction. Sparse sensor networks are deployed due to environment constraints, e.g., range measurements are missing due to obstructions in a room, or to reduce cost by minimizing the number of sensors. In an extreme case, the static sensors do not have range measurements between themselves since they are transmitters or receivers only. For sparse sensor networks, all the existing localization algorithms fail to work properly due to the lack of distance or connectivity data to uniquely calculate the geo-locations. *Mobile-assisted localization* [8] is to use one or more mobile sensors to add extra distance constraints to a sparse network, by moving the mobile sensors in the area of deployment and recording distances to neighbors at these intermediate locations. As long as the number of distance measurements is greater than the degree of freedom of the location coordinates, extra constraints are added to solve the localization problem. Pathirana et. al [9] developed a method based on Robust Extended Kalman Filter for a mobile node in a disconnected sensor network to estimate locations for sensor nodes it passes. For this purpose, one may use more than one mobile nodes to add extra range measurements. For example, Virtual Ruler [10] uses two nodes attached to a mobile vehicle to achieve better localization results in an indoor environment.

In this paper we develop a mobility assisted self localization algorithm, MA-MDS-MAP(P), for the sensor networks based on a distributed multidimensional scaling approach [11], [12]. Due to page limit, we omitted some details. For a detailed version of this paper, please refer to [13]. The rest of this paper is organized as follows: In section II we define the mobile self-localization problem and discuss its properties. Section III introduces MDS based localization algorithms, in particular, MDS-MAP(P). Section IV presents the MA-MDS-MAP(P) for the mobile self-localization problem. Section V provides detailed comparison of the performance between MDS-MAP(P) for static networks and the proposed MA-MDS-MAP(P) algorithm for mobile networks using four different topologies. The influence of noise and the effect of the number of virtual nodes and the number of mobile nodes on the accuracy of localization are also discussed in this section. Section VI concludes this paper.

Dr. Chang-Hua Wu is with the Science and Mathematics Department, Kettering University, Flint, MI 48504 cwu@kettering.edu

Dr. Weihua Sheng is with the Department of Electrical and Computer Engineering, Oklahoma State University, Stillwater, OK, 74078 weihua.sheng@okstate.edu

Dr. Ying Zhang is with Palo Alto Research Center, 3333 Coyote Hill Road, Palo Alto, CA 94304 yzhang@parc.com



(a) (b)

Fig. 1. Improvement on the distance estimate with node movement. (a) a sparse network. (b) If we can insert a node at A' and a node at E', then the graph becomes more rigid.

II. Mobile Self-Localization (MSL) Problem

The *self-localization* problem for sensor networks can be formalized as follows. Given a distance graph $G = \langle X, D, A\rangle$ where X is the set of locations of N nodes in a s-dimensional spaces (s is 2 or 3), D is a set of distance values between the nodes: $d_{ij} \in R$, and A is a set of anchor nodes, i.e., a subset of X, with cardinality $m \ll N$, whose elements have known locations, find $n = N - m$ unknown locations, $X = [x_1 x_2 ... x_n]$, where $x_i \in R^s$ denotes the location of node i, such that $|x_i - x_j| = d_{ij}$. If A is empty or too few, only relative geolocations among nodes can be recovered.

Since the range data may be noisy, this problem can be formalized as a least squares problem, i.e., minimizing $\sum w_{ij}(|x_i - x_j| - d_{ij})^2$ where w_{ij} are weights related to the noise levels of d_{ij}. The problem is difficult even for centralized solutions, since there is a large degree of freedom ($s \times n$ variables) when n is large. Furthermore, the solution may not be unique if the distance graph is not rigid [14]. When the connectivity is sparse, none of the existing self-localization algorithms would work well.

The mobility of the sensor nodes allows us to increase the density of the network through virtual nodes. Here a virtual node represents an instant location of a mobile node during its movement. A mobile sensor node takes distance measurements to a set of nearby sensor nodes at its intermediate points along its trajectory. Each such point adds a virtual node as well as a set of distance measurements. If the number of measurements added by a virtual node is greater than the degree of freedom of the location coordinate, more constraints are added to the distance graph to make the graph more rigid and have better localization results. In general, mobile nodes can be used to localize the whole network or turn a nonrigid network into a rigid network.

Therefore, the *mobile self-localization* (MSL) problem can be defined as a self-localization problem in which $X = X_r \cup X_v$ where X_r is a set of real node locations and X_v is a set of *virtual node* locations. Following are two scenarios that require mobile self localization:

- a partial mobile sensor network, where a subset of the nodes are mobile nodes.
- a full mobile sensor network, where all nodes are mobile.

In both scenarios, we may assume that a mobile sensor can measure the distance between two consecutive points in its trajectory, e.g., with an extra inertial sensor on board. Without that assumption, one has one less distance measurement added for each virtual node (e.g., AA' in Fig. 1-(b)). Although application-wise these are two different problems, technically they can be solved by the same type of algorithms.

In this paper, we will evaluate the second scenario, full mobile self-localization problem. The algorithms developed can be directly applied to partial mobile sensor networks.

III. MDS based Algorithms for Self Localization

Various self-localization algorithms have been developed in the last few years, including using semi-definite programming [15] and using multi-dimensional scaling [11], [16], [17]. The difference between [11] and [16], [17] is that the former uses shortest path distances to approximate Euclidean distances between missing distance pairs and then applies classical MDS to solve the problem on the complete distance graph, while the later uses iterative methods on the original graph. Although all localization algorithms can be applied to solve mobile self-localization problems, MDS-based methods have the advantage that it is robust for noisy and sparse networks, with or without anchor nodes.

Our new algorithm is based on the localization algorithm MDS-MAP (P) developed by Shang and Wheeler [11]. The steps of MDS-MAP(P) are as follows [11]:

1) Set the range for local maps, R_{lm}. For each node, neighbors within R_{lm} hops are involved in building its local map.

2) Compute local maps. Each node does the following:

- Compute shortest paths between all pairs of nodes in range R_{lm}. The shortest paths are used to construct the distance matrix for MDS.
- Apply the classical MDS to the distance matrix and retain the first 2 (or 3) largest eigenvalues and eigenvectors to construct a 2-D (or 3-D) local map.
- Refine the local map. Using the node coordinates in the MDS solution as the initial point, a least squares minimization is performed to make the distances between nearby nodes match the measured ones.

3) Merge local maps. Local maps can be merged either sequentially or in parallel.

4) Given sufficient anchor nodes (3 or more for 2-D networks, 4 or more for 3-D networks), transform the global map to an absolute map based on the absolute positions of the anchors.

One assumption of the MDS-based methods is that the shortest path between two nodes is approximately proportional to their Euclidean distance. While this may be true if the network is dense and uniform. In the situation where a dense network is not possible due to limited resources or the network topology is not uniform, this assumption is not necessarily true. For example, in Fig. 1-(a), while the shortest path between some nodes, such as $B \rightsquigarrow F$, $B \rightsquigarrow E$, are approximately proportional to the Euclidean distances, the shortest paths between A and D, and D and E are, however, significantly larger than their Euclidean distances.

In this situation, the assumption of the MDS methods is not valid anymore, which may lead to inaccurate localization. Moreover, in this graph, node A actually can be anywhere in the circle centered at B with radius equal to $|AB|$. This is due to the fact that this graph is not rigid, which means the existing constraints in the network are not enough to unambiguously determine the position of the nodes in it. In this situation, the traditional MDS-based method can't guarantee correct results.

In some type of networks, such as mobile sensor networks, some or all nodes can move around within a certain range. Therefore we can utilize the mobile capability of the nodes to get additional information about the networks, thus improving the accuracy of localization. An example is shown in Fig. 1. With the improved shortest distance between nodes, we expect the accuracy of the MDS-based localization will get better. Moreover, under certain deployment, mobile nodes can be used to localize the whole network or turn a nonrigid network into rigid network. In this paper, we propose a mobility assisted MDS-MAP(P) (MA-MDS-MAP(P)) algorithm for mobile sensor networks based on multi-dimensional scaling. During the movement of mobile nodes, virtual nodes are added into the network for additional constraints. The detailed and formal description of the proposed scheme is discussed in the following section.

IV. MA-MDS-MAP(P) FOR MSL

A. The addition of virtual nodes

Following the notation in section II, let v_i denote the node whose identification number is i. We also use v to denote a general node in the network. We assume the identification number ranges from 1 to n, where n is the number of nodes in the network, and each node has a unique identification number (ID). In the network, each node v_i keeps a local adjacency table $\langle i, j, d_{ij}\rangle$, where i is the identification number of v_i, j is the identification number of v_i's neighbor, and d_{ij} is the distance between v and its neighbor.

We assume that distance between sensors within communication range can be measured reliably and all nodes in the network are mobile. During the localization process, a node can be in two status: *Moving* or *Rest*. Before the localization starts, all nodes are in *Rest* status. During the movement, a node may send messages to its neighbor at several positions, including the initial position, to add virtual nodes when certain condition is met. Each step of the proposed scheme is discussed as follows.

- A node in the network broadcasts *Start-Localization* message to start the localization process. This node can be any node in the network which discovers the necessity to start a localization process.
- Upon receiving the *Start-Localization* message, each node, denoted by v_i, in the network starts moving in the following way.
  - At the initial position or any intermediate position during the movement, v_i broadcasts a message *AddVirtualNodes*(vid) to all of its neighbors, where vid is identification number of the virtual node to be added at this position.
  - When a neighbor, denoted by v_j, receives a *AddVirtualNodes*(vid), it measures the distance d_{ij} between v_j and v_i and sends an message *ACK*(j, d_{ij}) back to v_i.
  - When v_i receives more than three *ACK* messages from its neighbors, it broadcasts a *ConfirmVirtualNodes*(vid) to all the neighbors and for each *ACK*(j, d_{ij}) received, it adds an entry $\langle vid, j, d_{ij}\rangle$ in the local adjacency table to record the distance between the virtual node and the neighbor. Otherwise, it sends an *AbortVirtualNodes*(vid) to its neighbors. Upon receiving a *ConfirmVirtualNodes*, the neighbor, v_j, that has sent an *ACK* message to v_i adds a new entry $\langle j, vid, d_{ij}\rangle$ into its adjacency tables. If receiving an *AbortVirtualNodes*(vid), the neighbors simply delete all message records relevant to the potential virtual node vid.
  - v_i continues moving and repeats the above steps until it finishes the movement. Then v_i sends a message *MoveStoped*(i) to its neighbors. Upon receiving this message, its neighbors update the distances to v_i in their adjacency tables. v_i also updates the distances to its neighbors after sending the *MoveStoped*(i) message. Finally, it changes its status to *Rest*.
- Proceed the localization of the network in a revised MDS-MAP(P) method, which will be discussed below.

If each node adds virtual nodes at s positions, then the total messages sent during the movement is $O(2 + d)ns$, where d is the average number of neighbors of a mobile node. At the final stop, each node sends one *MoveStoped* message to its neighbors. The number of message in this step is $O(nd)$. Since only a limited number of virtual nodes can be added during the movement of a node, the total message complexity is $O(nd)$.

The above protocol addresses a general case, where every node in the network is mobile. However, it can also be applied to those networks where only a subset of nodes are mobile. In these type of networks, the nodes which can not move are always in *Rest* status. Let m be the number of mobile nodes in these type of networks, the message complexity is $O(md)$. The anchor nodes can also participated in the movement if they know their absolute positions during the movement.

To ensure the uniqueness of the identification numbers, the IDs of the virtual nodes added during the movement of a node v_i can be easily set as $n * s + i$, where s indicates index of the positions where v_i adds virtual nodes. For the first position, s is equal to 1; for the second position, s is equal to 2, and so on.

B. The merging of local maps

After the movement of all nodes, there are two types of nodes in the network: one type is real nodes, the other type is virtual nodes added during the movement of nodes. The

virtual nodes only exist in the adjacency tables of the real nodes. There is no communication between a virtual node and any other nodes. However, the distances kept in the adjacency tables provide additional information of the network. Based on these information, more precise localization can be obtained through a revised MDS-MAP(P) method. The details are discussed as follows:

- Step 1: Each node v sends requests to the nodes in its two-hop neighborhood for the adjacency tables and combines the adjacency tables of its neighbors into a local table T. After removing the duplicated entries (entries describing the same edge), node v constructs a local graph by identifying its two-hop neighbors, shown in Fig. 2-(a). Let the two hop neighbors be denoted by S_v. v then constructs the distance matrix for nodes in S_v using only edges between nodes in S_v. The elements in the distance matrix are the shortest path between every two nodes in S_v. Obviously, S_v contains all the real nodes within two hop distance of v and some virtual nodes.
- Step 2: v builds its local relative map using the classical MDS method for nodes in S_v
- Step 3: v refines the location of nodes, including the virtual nodes, in its local relative map via least squares minimization using the distance information in T and the coordinates from the step 2 as the initial point. Let $\langle i, j, d_{ij} \rangle$ denote the entries in T and p_{ij} denote the Euclidean distance between v_i and v_j based on their coordinates. The formulation of the minimization is

$$\min \sum_{i,j \in S_v} w_{ij}(d_{ij} - p_{ij})^2 \quad \text{for all } \langle i, j, d_{ij} \rangle \text{ in } T \quad (1)$$

  where w_{ij} is the weight for the distance between v_i and v_j. To compensate for the potential noise introduced in the movement of nodes, higher weight can be assigned to the distances between real nodes.
- Step 4: Merge the local maps until all the real nodes are included in a core map, which is grown by merging it with the local maps of neighboring real nodes. The local map with the maximum number of common nodes, including virtual nodes, with the core map is chosen to be merged with the core map. The merging can be done sequentially [11] or in a distributed way [12].
- Step 5: Transform the core map into an absolute map based on the absolute positions of anchors. Obviously, in this step, we only have to transform the coordinates of the real nodes.

Let k be the average number of nodes, including virtual nodes, in a local map. The overall complexity for computing each local map is $O(k^3)$. The total complexity for step 2 and 3 is $O(k^3 n)$. Similar to the discussion in [11], the complexity of step 4 is $O(k^3 n)$. For r anchors, the complexity of step 5 is $O(r^3 + n)$. So the total complexity of this revised MDS-MAP(P) method is also $O(n)$. The total number of messages in the localization using centralized merging, where each node sends its local map to a center node, is $O(nd) + O(n\log n) = O(n\log n)$, where $O(n\log n)$ is the number of messages sent in the merging process since the local map information of each real node has to travel a path of $\log n$ hops to the center.

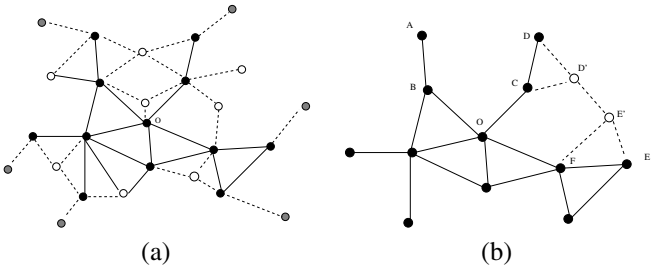


Fig. 2. (a) Only the real nodes and the virtual nodes, shown as non-filled circles, within two hops from the center node O are kept in the local map of O. The virtual nodes shown in gray spheres are not included because they are more than two hops away from O. (b) Node D and E move to D' and E' respectively and get into communication range of each other. The distance between them is kept in the adjacency table of both D and E.

V. Simulation results

A. Performance comparison with MDS-MAP(P) method

Simulation experiments have been carried out to evaluate the localization improvement after introducing virtual nodes under various network settings. As we know, the accuracy of the MDS method is highly dependent on the density/degree of the network. The higher the degree of each node, the more accurate the localization is. Therefore, we will focus on evaluating the improvement of the proposed scheme in the localization of sparse networks, whose degrees are under 5. We compared the performance of the MA-MDS-MAP(P) with the MDS-MAP(P) method in network of both uniform topology and irregular topology, see Figure 3. In this experiment, only one virtual node is added per movement.

For the four networks in this experiment, the field size is 10×10 and the communication range of each node is 1.0. In the random uniform network, 200 nodes are randomly positioned within a 10×10 field. To prevent the nodes from getting too close to each other, the minimum distance between any two nodes is larger than 0.5. The average degree is 4.72. In the regular uniform network, 144 nodes are regularly positioned in a 12×12 grid. To model the error in the grid placement, the positions of the nodes are adjusted by a uniformly distributed noise. The mean of the noise is 0, the range is between -0.09 and 0.09. The average degree is 4.39. The average degrees of the network in regular C shape topology and the network in random C shape topology are 3.96 and 4.12 respectively. The regular C shape network has 112 nodes and the random C-shape network contains 150 nodes. The error comparison between the MDS-MP(P) and the MA-MDS-MAP(P) is shown in Table I. As we can see, more than 90% decrease in the localization error can be achieved by incorporating virtual nodes.

B. Analysis of the fraction of mobile nodes

Up to now, we assume that every node in a network is mobile. For many networks, only a subset of nodes can move.

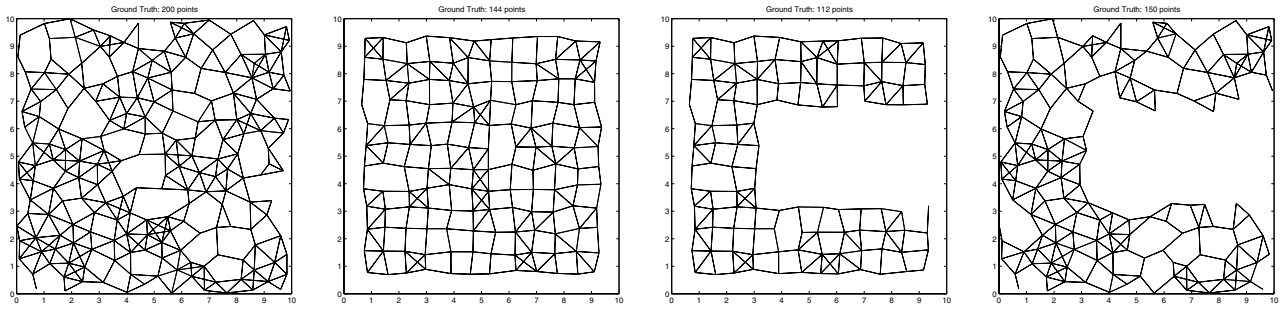


Fig. 3. The networks of four types of topologies

TABLE I

THE ERROR COMPARISON ON FOUR TYPES OF NETWORKS: FROM LEFT TO RIGHT, RANDOM UNIFORM, REGULAR UNIFORM, REGULAR C SHAPE, AND RANDOM C SHAPE.

mean	MDS-MAP(P)	0.182	0.162	0.695	0.286
	MA-MDS-MAP(P)	0.010	0.001	0.008	0.070
deviation	MDS-MAP(P)	0.086	0.105	0.509	0.271
	MA-MDS-MAP(P)	0.012	0.001	0.007	0.035

Even for the network where all nodes are mobile, we may not want to let all nodes participate in the movement so as to save the energy cost and reduce the localization error caused by noise in tracking the movement trajectory. In this experiment, we will evaluate for a network of certain degree, how many mobile nodes are sufficient for accuracy localization. We randomly created 10 networks for each type of topology and on each randomly generated network, we change the communication range, between 1.0 and 1.4, of the nodes to simulate networks with various degree. Each movement node creates three virtual nodes in a circular movement. In this experiment, each node has a probability p of participating in the movement. By evaluating the accuracy of localization at various levels of p, we can see that with a network of certain average degree, how many nodes should participate in the movement to get the desired accuracy so as to avoid unnecessary overhead and error due to moving more nodes than needed. Figure 4 shows the results of this experiment. As we can see from the figure, the higher the network degree, the less mobile nodes are needed. For network of degree higher than 6, the localization is good even with very small number of mobile nodes. For network with degree less than 5, 40% of mobile nodes are sufficient.

C. Analysis of reliability to noise

Up to now, we have not considered the influence of noise on the accuracy of the localization. In real applications. there are two types of noise. One is the noise in measuring the distance between real nodes. Since this noise is not caused by the movement of nodes, we are not going to analyze it here. The other noise is in tracking the trajectory of mobile nodes. The noise influences the distance estimates between virtual nodes and also the distances between virtual nodes and the final position of mobile nodes. To evaluate how well the proposed algorithm adapts to the noise inherent in the tracking of the trajectories of mobile nodes, we carried out a set of experiments with various levels of Gaussian noise. In this experiment, we will analyze the influence of noise on networks with 50% mobile nodes and each mobile node moves in a circular way and adds three virtual nodes during the movement. The mean of the noise is 0 and the deviation of the noise is from 0 to 0.15. Fig. 5 shows the results of the experiments. We can see that for uniform network, under 6 percent of additive noise, both the mean and standard deviation of the error is less than 0.05, which is 5 percent of the communication range. For C-shape networks, we see more vibrations. However, the mean and the standard deviation are still quite small if the deviation of the noise is less than 0.03, which is 3 percent of the communication range.

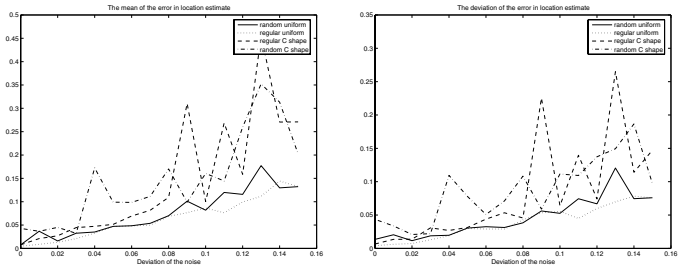


Fig. 5. Evaluation of the effect of the noise in sensor movement. The horizon axis is the deviation of the noise. The vertical axis is the estimate error.

VI. CONCLUSION

In this paper, we propose a mobility assisted self localization method (MA-MDS-MAP(P)) based on multi-dimensional scaling for sparse sensor networks. The MDS-MAP(P) highly depends on the degree of the network, which leads to poor performance in sparse network. Based on the fact that sensors in a mobile network have limited mobile capability, in the proposed approach, more information is obtained by moving the sensors and adding virtual nodes during the movement. The distances between the virtual nodes and the real nodes are kept in adjacency tables. The

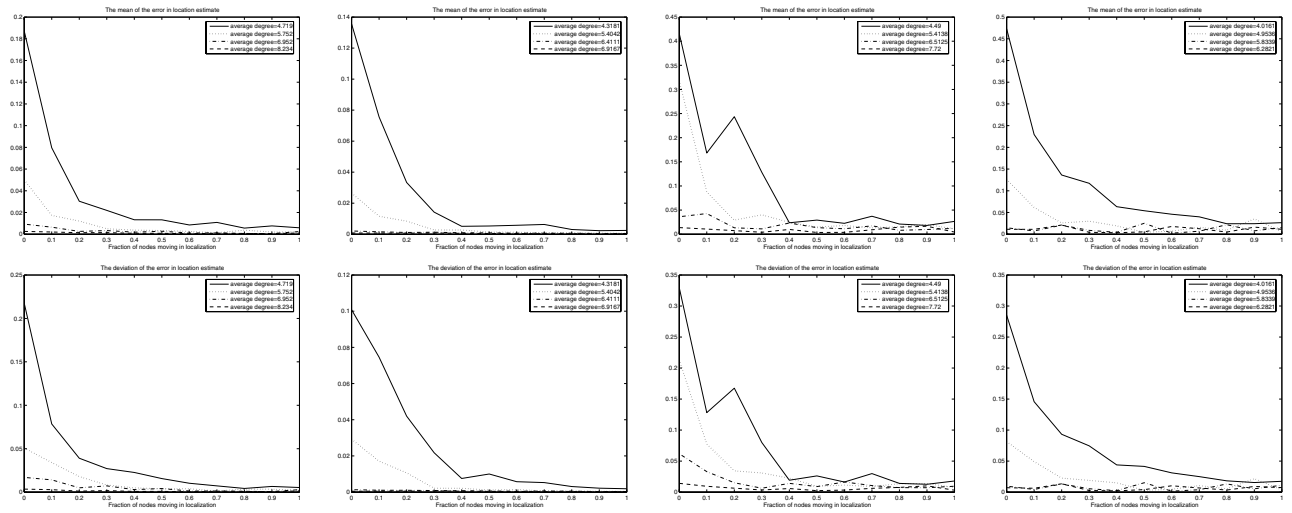


Fig. 4. The localization error with respect to the fraction of mobile nodes on networks of four topologies: random uniform, regular uniform, random C shape and regular C shape. The first row shows the mean of the error and second row shows the standard deviation of the error.

virtual nodes are incorporated in building and merging the local maps. Evaluation of the performance of the proposed approach is carried out on four types of networks: random uniform, random C-shape, regular uniform, and regular C-shape. The results have shown significant improvement over the MDS-MAP(P) approach on sparse networks. The MA-MDS-MAP(P) algorithm can be used in both partial mobile sensor networks and full mobile sensor networks, which include robotic networks and body sensor networks.

VII. Acknowledgment

We give special thanks to Professor Yi Shang in the University of Missouri-Columbia for kindly sharing with us the MATLAB simulation code.

References

[1] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. A survey on sensor networks. *IEEE Commun. Mag.*, 40:102–114, 2002.

[2] A. Mainwaring, J. Polastre, R. Szewczyh, D. Culler, and J. Anderson. Wireless sensor networks for habitat monitoring. In *Proceedings of Wireless Sensor Network and Applications*, 2002.

[3] S. N. Simic and S. Sastry. Distributed environmental monitoring using random sensor networks. In *Proceedings of the 2nd International Workshop on Information Processing in Sensor Networks*, pages 582–592, 2003.

[4] J. Cortes, S. Martinez, T. Karatas, and F. Bullo. Coverage control for mobile sensing networks. In *IEEE Trasactions on Robotics and Automation*, pages 243–255, 2004.

[5] J. R. Spletzer. *Sensor Fusion Techniques for Cooperative Localization in Robot Teams*. PhD thesis, University of Pennsylvania, 2003.

[6] S. Tilak, V. Kolar, N. B. Abu-Ghazaleh, and K. D. Kang. "dynamic localization control for mobile sensor networks". In *Proceedings of IEEE International Workshop on Strategies for Energy Efficiency in Ad Hoc and Sensor Networks*, April 7-9 2005.

[7] L. Hu and D. Evans. Localization for mobile sensor networks. In *Proceedings of MobiCom'04*, pages 45–57, 2004.

[8] N. B. Priyantha, H. Balakrishnan, E. D. Demaine, and S. Teller. Mobile-assisted localization in wireless sensor networks. In *IEEE Conference on Computer Communications (InfoCom05)*, 2005.

[9] P. N. Pathirana, N. Bulusu, A. Savkin, and S. Jha. Node localization using mobile robots in delay-tolerant sensor networks. *IEEE Transactions on Mobile Computing*, 4(4), July/August 2005.

[10] C. Wang, Y. Ding, and L. Xiao. Virtual ruler: Mobile beacon based distance measurements for indoor sensor localization. In *The Third International Conference on Mobile Ad-hoc and Sensor Systems (MASS06)*, 2006.

[11] Y. Shang and W. Ruml. Improved mds-based localization. In *Proceedings of IEEE InfoCom'04*, 2004.

[12] Y. Shang, W. Ruml, and M. Fromherz. Positioning using local maps. *"Ad Hoc Networks" Journal of Elsevier*, 4:240–253, 2006.

[13] Chang-Hua Wu, Weihua Sheng, and Ying Zhang. Mobile self-localization using multi-dimensional scaling in robotic sensor networks. *The International Journal of Intelligent Control and Systems*, Dec, 2006.

[14] A. So and Y. Ye. Theory of semidefinite programming for sensor network localization. In *SODA05*, 2005.

[15] P. Biswas and Y. Ye. Semidefinite programming for ad hoc wireless sensor network localization. In *ACM/IEEE Conference Information Processing in Sensor Networks*, 2004.

[16] X. Ji and H. Zha. Sensor positioning in wireless ad-hoc sensor networks using multidimentional scaling. In *IEEE Conference on Computer Communications (InfoCom04)*, 2004.

[17] J.A. Costa, N. Patwari, and A. Hero. Distributed weighted-multidimentional scaling for node localization in sensor networks. *ACM Transactions on Sensor Networks*, 2(1), February 2006.