Scientific Journal of Impact Factor (SJIF): 5.71

e-ISSN (O): 2348-4470
p-ISSN (P): 2348-6406

International Journal of Advance Engineering and Research Development

Volume 5, Issue 03, March -2018

MDS- MAP Algorithm For Localization

Lalita Baburao Choudhary[1], V.V.Yerigeri[2]

[1]*PG Student, Dept. of PG, M.B.S.E. College of Engineering, Ambejogai, India,*
[2] *Assistant Professor, Dept. of PG, M.B.S.E. College of Engineering, Ambejogai, India.*

Abstract --- *We propose an approach that uses connectivity information—who is within communications range of whom—to derive the locations of nodes in a network. The approach can take advantage of additional information, such as estimated distances between neighbours or known positions anchor nodes, if it is available. It is based on multidimensional scaling (MDS), an efficient data analysis technique that takes $O(n)^3$ time for a network of n nodes. Unlike previous approaches, MDS takes full advantage of connectivity or distance information between nodes that have yet to be localized. We examined the performance of our algorithm for parameters by conducting simulation.*

Keyword *---- Position estimation, node localization, multilateration, multidimensional scaling, ad-hoc networks, and sensor networks*

I. INTRODUCTION

A. LOCALISATION IN NETWORK

A. Anchor/Beacon nodes

The goal of localization is to determine the physical coordinates of a group of sensor nodes. These coordinates can be global, meaning they are aligned with some externally meaningful system like GPS, or relative, meaning that they are an arbitrary "rigid transformation" (rotation, reflection, translation) away from the global coordinate system. Beacon nodes (also frequently called anchor nodes) are a necessary prerequisite to localizing a network in a global coordinate system. Beacon nodes are simply ordinary sensor nodes that know their global coordinates a priori. This knowledge could be hard coded, or acquired through some additional hardware like a GPS receiver. At a minimum, three non-collinear beacon nodes are required to define a global coordinate system in two dimensions. If three dimensional coordinates are required, then at least four non-coplanar beacons must be present. Beacon nodes can be used in several ways.

Localization plays an important role in wireless sensor networks when the positions of the nodes are not provided in advance. One way to acquire the positions is by equipping all the sensors with a global positioning system (GPS). Clearly, this adds considerable cost to the system. As an alternative, many research efforts have focused on creating algorithms that can derive positions of sensors based on basic local information such as proximity (which nodes are within communication range of each other's) or local distances (pairwise distances between neighboring sensors).The problem is solvable, meaning that it has a unique set of coordinates satisfying the given local information, only if there are enough constraints. Note that based on local information any solution is unique only up to rigid transformations (rotations, reflections and translations).

The problem of nodes localization appears in a variety of wireless sensor network (WSN) applications. The information gathered from the network can often be useless if not matched with the location where it is sensed. Finding the exact physical location is a crucial issue for continual network operation and WSN management. Many different techniques have been proposed for solving this problem; however, since most of them fail to perform well on irregular topologies, this problem remains a challenge. Multidimensional Scaling (MDS) is a set of analytical techniques that have been used for many years in disciplines such techniques for WSN localization can be basically divided into two categories: range-based and range-free methods. The range-based techniques are considered more accurate. And most of the algorithms for localization belong to this category. They use the distance between the nodes in the network. RSSI (receive signal strength indicator) is the most common technique used for distance estimation.

RSSI utilizes small resources without the need for extra hardware. RSSI measures the power of the received radio signal to calculate the distance between two nodes that are in transmission range of each other. Other techniques (time of arrival: ToA, time difference of arrival: TDoA, etc.) for distance measurement translate propagation time into distance. This can be done if signal propagation speed is known in advance. These techniques can be used with acoustic, infrared, and ultrasound signals.

B. Received Signal Strength Indication (RSSI)

In wireless sensor networks, every sensor has a radio. The question is: how can the radio help localize the network? There are two important techniques for using radio information to compute ranges. The other, Received Signal Strength Indication (RSSI), is covered below. In theory, the energy of a radio signal diminishes with the square of the distance from the signal's source. As a result, a node listening to a radio transmission should be able to use the strength of the received signal to calculate its distance from the transmitter. RSSI suggests an elegant solution to the hardware ranging problem: all sensor nodes are likely to have radios –why not use them to compute ranges for localization?

In practice, however, RSSI ranging measurements contain noise on the order of several meters [2]. This noise occurs because radio propagation tends to be highly non-uniform in real environments. For instance, radio propagates differently over asphalt than over grass. Physical obstacles such as walls, furniture, etc. reflect and absorb radio waves. As a result, distance predictions using signal strength have been unable to demonstrate the precision obtained by other ranging methods such as time difference of arrival.

B. CHARACTERISTICS OF LOCALIZATION ALGORITHM

The main objective of a localization algorithm is to determine position of a node. However, there are certain criteria that the algorithm should meet for it to be practicable. The criteria usually depend upon the type of application for which the localization algorithm is designed. General design objectives or desired characteristics of an ideal localization algorithm are:

- It is highly desirable that the localization algorithms are RF –based. The sensor nodes are equipped with a short - range RF transmitter. An efficient localization algorithm exploits this radio capability for localization in addition to its primary role of data communication.
- A wireless sensor network is ad hoc in nature. The localization algorithm should take the ad hoc nature of the network into consideration.
- The nodes should be able to determine their position in as small time as possible so that the localization algorithm has a low response time. This would enable sensor nodes to be deployed quickly.
- The position of the sensor node found by such an algorithm should be Accurate enough for the specific application for which this algorithm is being used.
- The algorithm must be robust so that it may work in adverse conditions.
- The algorithm should be scalable so that if sensor nodes are added or removed, it should still be able to work out the position of the nodes. Furthermore, the algorithm should produce acceptable results for sensor networks comprising of small to large number of nodes.
- The localization algorithm should be energy efficient and preferably energy aware as well because the sensor nodes are autonomous and normally do not have any external source of power.
- The localization algorithm should be adaptive to the change in the number of beacon nodes. If the number of available beacon nodes changes, the algorithm should still be able to provide location estimates.

However, the accuracy of node estimates will change with the change in number of large-scale networks with hundreds and even thousands of very small, battery-powered and wirelessly connected sensor and actuator nodes are becoming a reality [5]. For example, future sensor networks will involve a very large number of densely deployed sensor nodes over physical space. In particular, the nodes are typically highly resource-constrained (processor, memory, and power), have limited communication range, are prone to failure, and are put together in ad-hoc networks.

Imagine a network of sensors sprinkled across a large building or an area such as a forest. Typical tasks for such networks are to send a message to a node at a given location (without knowing which node or nodes are there, or how to get there), to retrieve sensor data (e.g., sound or temperature levels) from nodes in a given region, and to find nodes with sensor data in a given range. Most of these tasks require knowing the positions of the nodes, or at least relative positions among them. 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 them with absolute position, but this is currently a costly solution.

In this paper, we present a method for computing the positions 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, MDS-MAP, has three steps. Starting with the given network connectivity information, we first use an all-pairs shortest-paths algorithm to roughly estimate the distance between each possible pair of nodes. Then we use multidimensional scaling (MDS), a technique from 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 connectivity information to produce a meaningful result. If the distances between neighboring nodes can be estimated, that information can be easily incorporated into the pair-wise shortest-path computation during the first step of the algorithm. MDS yields coordinates that provide the best fit to the estimated pairwise distances, but which lie at an arbitrary rotation and translation. If the coordinates of any nodes are known, they can be used to derive the affine transformation of the MDS coordinates that allows the best match to the known positions. Only three such anchor nodes' are necessary to provide absolute positions for all the nodes in the network.

C. ISSUES IN LOCALIZATION ALGORITHM DESIGN

A. Resource constraints

Sensor networks are typically quite resource-starved. Nodes have rather weak processors, making large computations infeasible. Moreover, sensor nodes are typically battery powered. This means communication, processing, and sensing actions are all expensive, since they actively reduce the lifespan of the node performing them. Not only are those, sensor networks typically envisioned on a large scale, with hundreds or thousands of nodes in a typical deployment. This fact has two important consequences: nodes must be cheap to fabricate, and trivially easy to deploy. Nodes must be cheap, since fifty cents of additional cost per node translates to $500 for a one thousand node network. Deployment must be easy as well: thirty seconds of handling time per node to prepare for localization translates to over eight man-hours of work to deploy a 1000 node network Localization is necessary to many functions of a sensor network; however, it is not the purpose of a sensor network. Localization must cost as little as possible while still producing satisfactory results. That means designers must actively work to minimize the power cost, hardware cost, and deployment cost of their localization algorithms.

B. *Node density*

Many localization algorithms are sensitive to node density. For instance, hop count based schemes generally require high node density so that the hop count approximation for distance is accurate (section 1.2.3). Similarly, algorithms that depend on beacon nodes fail when the beacon density is not high enough in a particular region. Thus, when designing or analyzing an algorithm, it is important to notice the algorithm's implicit density assumptions, since high node density can sometimes be expensive if not totally infeasible.

C. Non-convex topologies

Localization algorithms often have trouble positioning nodes near the edges of a sensor field. This artifact generally occurs because fewer range measurements are available for border nodes, and those few measurements are all taken from the same side of the node. In short, border nodes are a problem because less information is available about them and that information is of lower quality. This problem is exacerbated when a sensor network has a no convex shape: Sensors outside the main convex body of the network can often prove unlocalizable. Even when locations can be found, the results tend to feature disproportionate error.

D. *Environmental obstacles and terrain irregularities*

Environmental obstacles and terrain irregularities can also wreak havoc on localization. Large rocks can occlude line of sight, preventing TDoA ranging, or interfere with radios, introducing error into RSSI ranges and producing affect radios and acoustic ranging systems. Indoors, natural features like walls can impede measurements as well. All of these issues are likely to come up in real deployments, so localization systems should be able to cope.

II. LOCALIZATION USING MDS-MAP

We consider the node localization problem under two different scenarios. In the first, only proximity (or connectivity) information is available. Each node only knows what nodes are nearby, presumably by means of some local communication channel such as radio or sound, but not how far away these neighbors are or in what direction they lie. In the second scenario, the proximity information is enhanced by knowing the distances, perhaps with limited accuracy, between neighboring nodes.

In both cases, the network is represented as an undirected graph with vertices V and edges E. The vertices correspond to the nodes, of which there exist $m \geq 0$ special nodes with known positions, which we will call anchors. For the proximity-only case, the edges in the graph correspond to the connectivity information. For the case with known distances to neighbors, the edges are associated with values corresponding to the estimated distances. We assume that all the nodes being considered in the positioning problem form a connected graph, i.e., there is a path between every pair of nodes There are two possible outputs when solving the localization problem.

One is a relative map and the other is an absolute map. The task of finding a relative map is to find an embedding of the nodes into either two- or three-dimensional space that results in the same neighbor relationships as the underlying network. Such a relative map can provide correct and useful information even though it does not necessarily include accurate absolute coordinates for each node. 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 anchors present to uniquely determine the absolute positions of the nodes.

Furthermore, some applications only require relative positions of nodes, such as in some direction-based routing algorithms. Sometimes, 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 discovering and tracking.

As we will show below, our method can potentially generate both results, depending on the number of anchor nodes. The method first generates a relative map of the network and then transforms it to absolute positions if sufficient anchors are available. Before we describe the details of our method, we first introduce multidimensional scaling (MDS), with a focus on classical MDS, which is used to generate the relative map.

A. MDS-MAP Algorithm

Consider a small cloud of coloured beads suspended in the air. The distances between each pair of beads are measured. Now, if the beads fall down on the ground, the earlier configuration of the beads in the air can be reconstructed by using the distance measurements recorded earlier. Position of each bead in the reconstructed arrangement is chosen such that the distances in the new arrangement match the distances in the original layout. It is this problem of reconstructing the arrangement that is solved by MDS.

It should be noted that the reconstructed arrangement will be an arbitrarily rotated and flipped version of the original layout because the new layout is constructed without using the absolute information. MDS-MAP is a localization method based on multidimensional scaling [3]. The MDS-MAP algorithm consists of three steps:

- Compute the shortest distances between all pairs of nodes in the region. The computed distances are used for building the distance matrix for MDS.
- Apply classical MDS to the distance matrix, retaining the first 2 eigenvalues and eigenvectors to construct a 2D relative map.
- Given sufficient anchor nodes (3 or more), transform the relative map to an absolute map based on the absolute positions of anchors.

In the first step, we 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 algorithm, can be applied. In the second step, classical MDS is applied directly to the distance matrix. The core of classical MDS is singular value decomposition. The result of MDS 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 relative to the true node positions. In the third step, 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 MDS map.

Computing the double centred matrix on the left hand side (call it B) is symmetric and positive semi definite. Performing singular value decomposition on B gives us B = V AV. The coordinate matrix becomes X = V A1/2. Retaining the first r largest eigenvalues and eigenvectors ($r < m$) leads to a solution in lower dimension. This implies that the summation over k in Eq. (1) runs from 1 to r instead of m. This is the best low-rank approximation in the least-squares sense. For example, for a 2-D network, we take the first 2 largest eigenvalues and eigenvectors to construct the best 2-D approximation. For a 3-D network, we take the first 3 largest eigenvalues and eigenvectors to construct the best 3-D approximation.

In nonmetric (ordinal) MDS (first developed by Shepard [14]), the goal is to establish a monotonic relationship between inter-point distances and the desired distances. In- stead of trying to directly match the given distances, one is satisfied if the distances between the points in the solution fall in the same ranked order as the corresponding distances in the input matrix. The advantage of nonmetric MDS is that no assumptions need to be made about the underlying transformation function. The only assumption is that the data is measured at the ordinal level. Just as classical MDS, nonmetric MDS can also be applied to the localization problem. By adopting a more flexible model, the effects of a few highly incorrect measurements might be more easily tolerated.

III. RELATED WORK

Node localization has been a topic of active research in recent years. However, few approaches for locating nodes in an ad-hoc network are described. Most systems use some kind of range or distance information and many of them rely on powerful beacon nodes with extreme capabilities, such as radio or laser ranging devices.

Doherty's convex constraint satisfaction approach formulates the localization problem as a feasibility problem with radial constraints. Nodes which can hear each other are constrained to lie within a certain distance of each other. This convex constraint problem is in turn solved by efficient semi-definite programming (an interior point method) to find a globally optimal solution. For the case with directional communication, the method formulates the localization problem as a linear programming problem, which is solved by an interior point method. The method requires centralized computation. For the technique to work well, it needs anchor nodes to be placed on the outer boundary, preferably at the corners. Only in this configuration are the constraints tight enough to yield a useful configuration. When all anchors are located in the interior of the network, the position estimation of outer nodes can easily collapse toward the center, which leads to large estimation errors. For example, with 10% anchors, the error of unknowns is on the order of the radio range. With 5 anchors in a 200-node random network, the error of unknowns is more than twice the radio range.

Most localization methods for ad-hoc networks require more information than just connectivity and use more powerful beacon nodes. The ad-hoc localization techniques used in mobile robots usually fall into this category [7, 5].

Mo- bile robots use additional odometric measurements for estimating the initial robot positions, which are not available in sensor networks.

Many existing localization techniques for networks use distance or angle measurements from a fixed set of reference points or anchor nodes and apply multilateration or triangulation techniques to find coordinates of unknown nodes [6, 8]. The distance estimates can be obtained from received signal strength (RSSI) or time-of-arrival (ToA) measurements. Due to nonuniform signal propagation environments, RSSI methods are not very reliable and accurate. ToA methods have better accuracy, but may require additional hard- ware at the sensor nodes to receive a signal that has a smaller propagation speed than radio, such as ultrasound [9]. Emphasis has been put on algorithms that can be executed in a distributed fashion on the sensor nodes without centralized computation, communication, or information propagation. The “DV-based” approach by Niculescu and Nath [1] is distributed. The “DV-hop” method achieves an location error of about 45% of the radio range for networks with 100 nodes, 5 anchors, and average connectivity 7.6. It starts with the anchor nodes. The anchors flood their location to all nodes in the network. Each unknown node performs a triangulation to three or more anchors to estimate its own position. The method works well in dense and regular topologies. For sparse and irregular networks, the accuracy degrades to the radio range. The “DV-distance” method uses distance be- tween neighboring nodes and reduces the location error by about half of that of “DV-hop”.

Savarese et al. propose another distributed method [9]. The method consists of two phases: start-up and refinement. For the start-up phase, they use Hop-TERRAIN, an algorithm similar to DV-hop. Hop-TERRAIN is run once at the beginning to generate a rough initial estimate of the nodes’ locations. Again, it needs at least 3 anchor nodes to start. Then the refinement algorithm is run iteratively to improve and refine the position estimates. The algorithm is concerned only with nodes within a one-hop neighborhood and uses a least-squares triangulation method to determine a node’s position based on its neighbors’ positions and distances to them. The approach can deliver localization accuracy within one third of the communication range.

When the number of anchor nodes is high, the collaborative multilateration approach by Savvides et al. can be used [9]. The method estimates node locations by using anchor locations that are several hops away and distance measurements to neighboring nodes. A global nonlinear optimization problem is solved. The method has three main phases: 1) formation of collaborative sub-tree, which only includes nodes that can be uniquely determined, 2) computation of initial estimates with respect to anchor nodes, 3) position refinement by minimizing the residuals between the measured distances between the nodes and the distances computed using the node location estimates. They present both a centralized computation model and a distributed approximation of the centralized model. The method works well when the fraction of anchor nodes is high.

The GPS-less system by Bulusu employs a grid of beacon nodes with known positions. Each unknown node sets its position to the centroid of the beacons near the un- known. The position accuracy is about one-third of the separation distance between beacons, so the method needs a high beacon density to work well.

Almost all the existing methods need some kind of anchor or beacon nodes to start with. Our method does not have this limitation. It builds a relative map of the nodes even without anchor nodes. With three or more anchor nodes, the relative map can be transformed, and absolute coordinates of the nodes are computed. Our method works well in situations with low ratios of anchor nodes and performs even better on regular networks. A limitation of the cur- rent implementation is that it is centralized. There are various ways to apply this method in a decentralized or distributed fashion. For example, the method can be applied to sub-networks to obtained regional relative maps, which are patched together to form an overall map of the network.

IV.EXPERIMENTAL RESULTS

We assumed that a wireless sensor containing 200 sensor nodes distributed over two-dimensional over 10r×10r monitoring area. Sensor nodes maximum radio range. We have assumed that a path is always available between each node pair and nodes in proximity to each other can exchange messages. Also, RSS technique is used for distance estimation between neighboring nodes.

We mainly considered random topology for our work. We have assumed no radio rang error present in the network. We implemented MDS-MAP (Dijkastra’s) technique using Matlab we observe performance over the parameters like connectivity, number of anchor nodes and estimation error. We have simulated the result using 3, 4, 6 and10 anchors for random topology. Thus, we discovered each node location using MDS-MAP (Dijkastra’s) technique. The error estimated through this technique is normalized by using the radio communication range as

$$Error = \frac{\sum_{i=1}^{n} \text{distance}(pos_i^{(estimated)} - pos_i^{(true)})}{(n-N)\cdot R} \cdot 100\%$$

Where
N=number of anchor nodes,
n=number of sensor nodes in WSN.

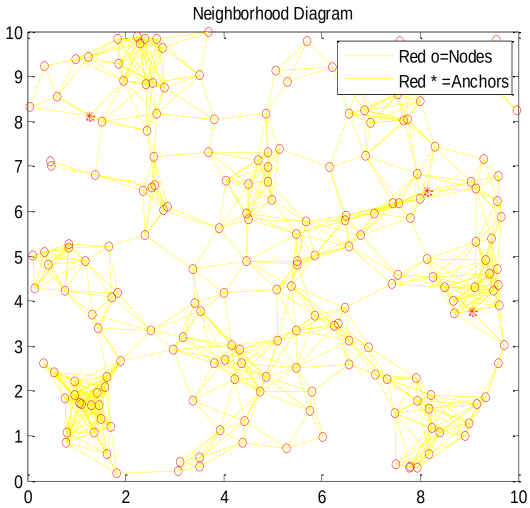

Figure 1a

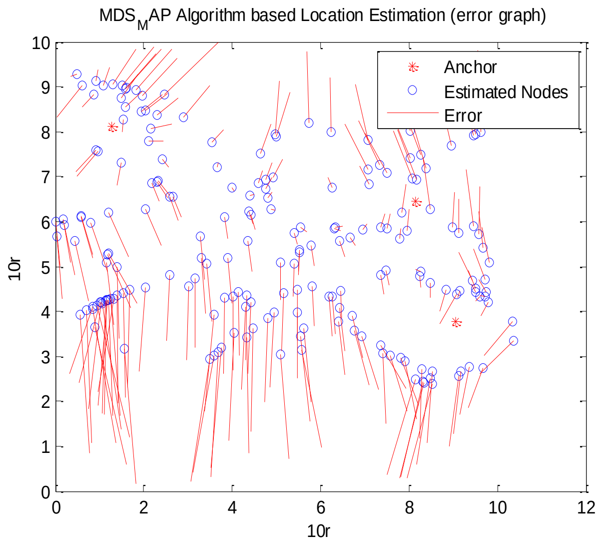

Figure 2a

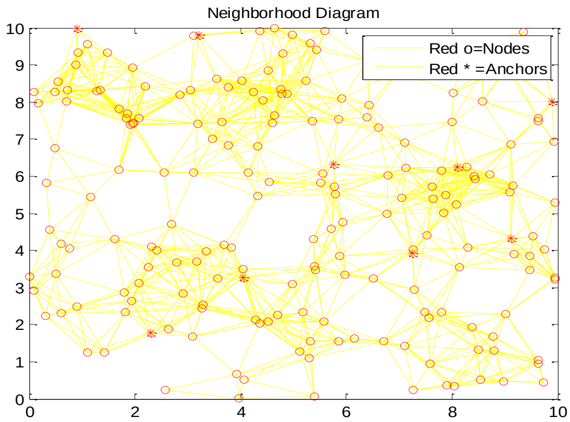

Figure 1b

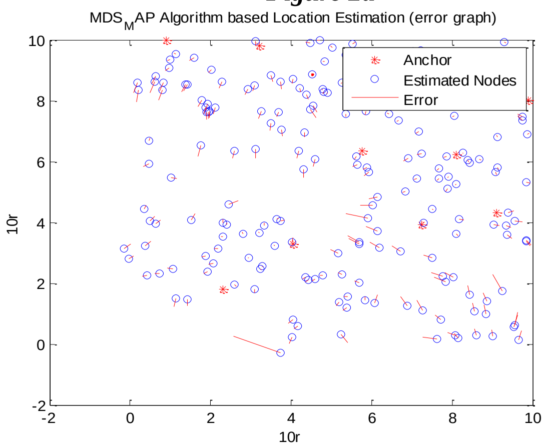

Figure 2b

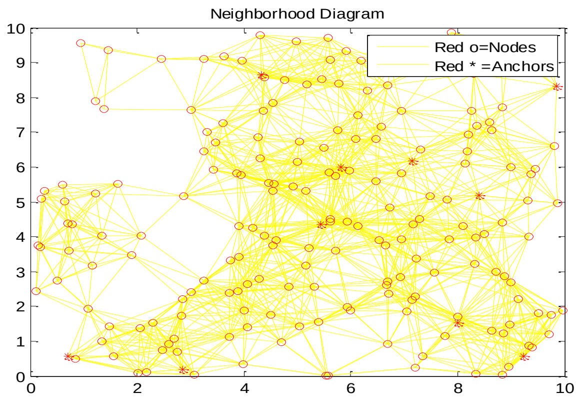

Figure 1c

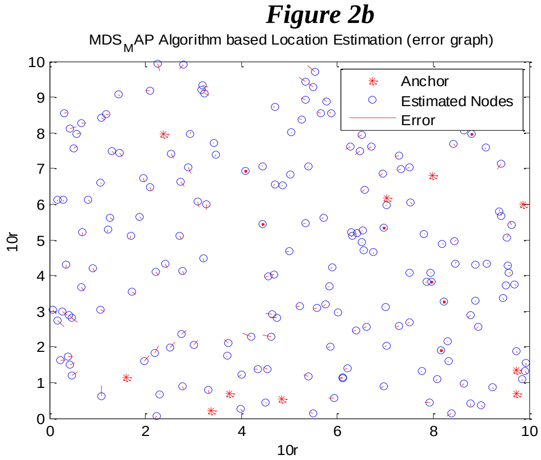

Figure 2c

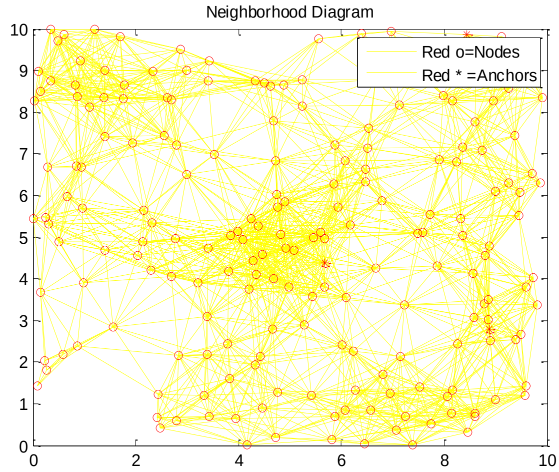

Figure 1d

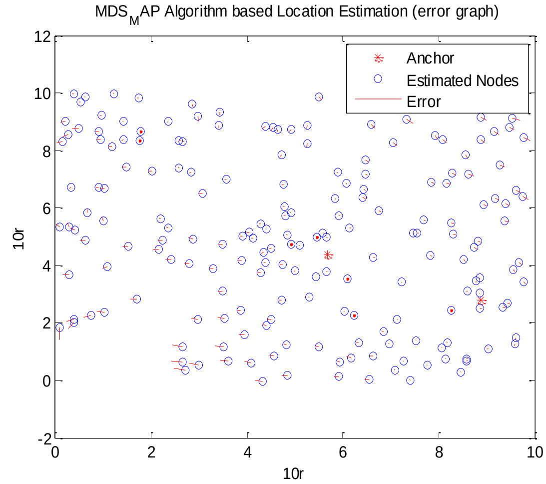

Figure 2d

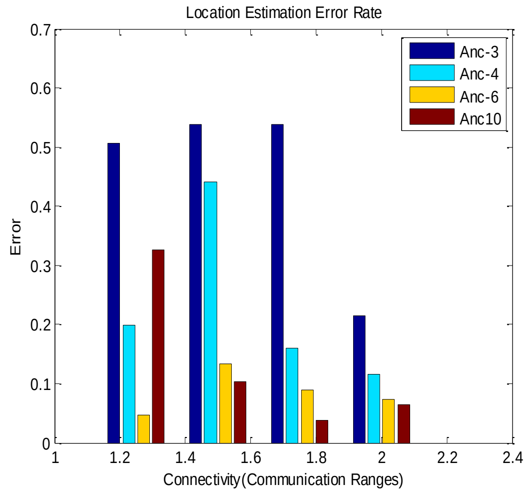

Figure 1e shows the effect of radio ranges on estimated errors

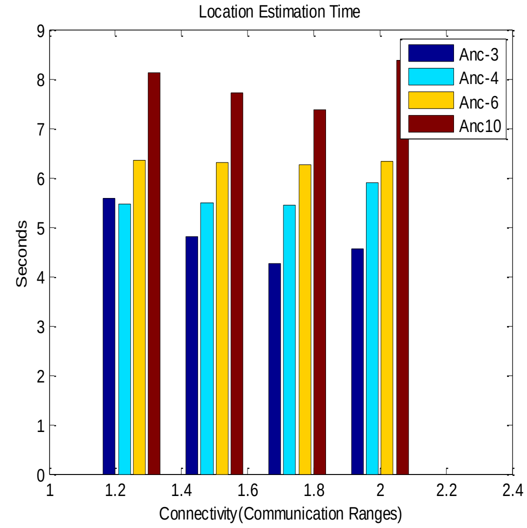

Figure 2e

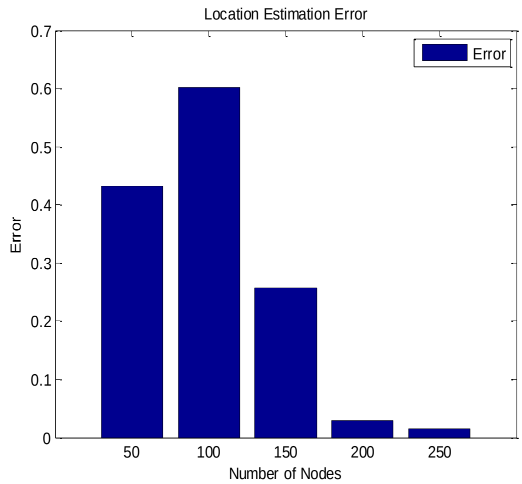

Figure 3e

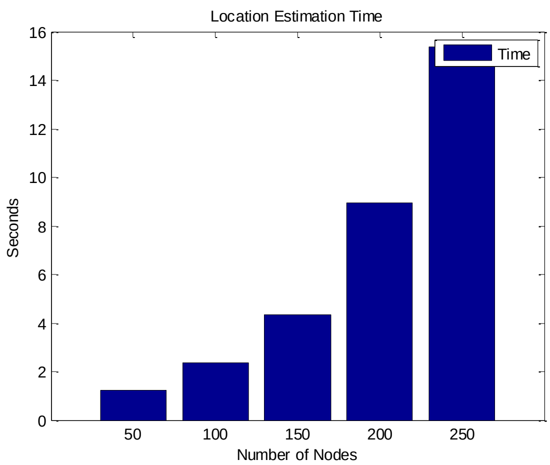

Figure 4d

Figure 1a,b,c,d shows the graphs circles represents sensor nodes and edges represents the connection between the nodes that are within the communication range 1.25r,1.5r,1.75r and 2r. In Figure 2 a, b, c, d shows circles represent the true locations of the nodes and solid lines represent the errors of the estimated position from the true position. The longer line, the longer error. Anchor nodes are represented by asterisks. Figure 1e shows the relationship between the connectivity and localization error. Figure 2e shows time *Vs* connectivity, from this we can see that number of anchors increases time also increase.as expected number of nodes increases time of execution also increases. Figure 3e show the error estimation *Vs* number of nodes. Connectivity can be increased by increasing the number of nodes. From this we can see that after particular connectivity error decreases as connectivity increase. Figure 4d shows time *Vs* number of nodes.

In our experiments, we ran MDS-MAP on various topologies of networks in Matlab. The nodes are placed (a) randomly with a uniform distribution within a square area, (b) on a square grid with some placement errors, or (c) on a hexagonal grid with some placement errors. In a square grid, assuming r is the unit length, n2 nodes are typically placed in an nr by nr square. We model placement errors for the grid layout as Gaussian noises. With a placement error *ep*, a random value drawing from a normal distribution *r* x *ep* x N (0, 1) is added to the node's original grid position. The placement error in a hexagonal grid is defined similarly. The anchor nodes are selected randomly. The data points represent averages over 30 trials in networks containing 100 to 200 nodes.

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. The distance information is modeled as the true distance blurred with Gaussian noise. Assume the true distance is d* and range error is *er*; then the measured distance is a random value drawing from a normal distribution d* (1+N (0; *er*)). The connectivity (average number of neighbors) is controlled by specifying radio range R. To compare with previous results in [4, 8], the errors of position estimates are normalized to R (i.e., 50% position error means half of the range of the radio). We do not consider models of non-uniform radio propagation or widely varying ranging errors. Both modeling these phenomena and simulating their effects are very important directions for future work.

When the distances between neighbors are known, even with limited accuracy, the result of MDS-MAP can be significantly improved. Figure 3 shows the result of MDS-MAP knowing the distance of neighbors with 5% range error. Figure 3(a) shows the map constructed by MDS. Again, it has different scale and orientation than the ones in Figures 1 and 2(a). Figure 3(b) shows the final estimation of MDS-MAP based on the same 4 anchor nodes. It has an average estimation error of 0.24r, much better than the previous result using connectivity only.

Figure 4 shows the average performance of MDS-MAP as a function of connectivity and number of anchors. Figure 4(a) shows results of MDS-MAP based on proximity information only. The radio ranges (R) are 1r, 1.25r, 1.5r, 1.75r, and 2r, respectively, which lead to average connectivity levels 5.9, 8.9, 12.2, 16.2, and 20.7, respectively. 3, 4, 6, and 10 random anchors are used. Position estimates by MDS-MAP have an average error under 100%R in scenarios with just 4 anchor nodes and an average connectivity level of 8.9 or greater. When the connectivity level is 12.2 or greater, the error with just 3 anchors is quite good, close to or better than 50%. On the other hand, when the connectivity is low, e.g., 5.9, the errors can be large. These results are much better than the ones obtained by the convex optimization approach in [4] when the number of anchor nodes is low. For example, with 4 to 10 anchors in a 200-node random network, the convex optimization approach has an average estimation error of more than twice the radio range when the radio range is 1.25R and above. In contrast, our method has an error from about 80% down to 40% as the radio range goes from 1.25R to 2R.

A. POSSIBLE EXTENSIONS

A drawback of the current implementation of MDS-MAP is that it requires global information of the network and centralized computation. One way to address this issue is to divide the network into sub-networks and apply MDS-MAP to each sub-network independently. Since our method does not require anchor nodes in building a relative map of a sub-network, the method can be applied to many sub-networks in parallel. Then adjacent local maps can be combined by aligning with each other. In another words, the complete map of the sensor network consists of many smaller patches. When three or more anchors are present in either a sub-network or the whole network, an absolute map can be computed accordingly. Although this patching approach requires significant computation within each patch, one has considerable flexibility in choosing which nodes perform the computation. Preliminary experiments have been very encouraging and detailed results will be reported in future work.

MDS-MAP can also be extended by applying more advanced MDS techniques. Instead of classical metric MDS, other MDS techniques such as ordinal MDS and MDS with missing data can be applied. This may be useful to handle non-uniform radio propagation and non-uniform ranging errors. We have done some limited experiments with ordinal MDS. Our results show that ordinal MDS is better than classical MDS when the connectivity level of the sensor network is low, and is comparable with classical MDS when the connectivity level is high.

Another drawback of MDS-MAP is that when the number of anchor nodes is large, the performance of MDS-MAP is not as good as previous methods such as the constraint based approach [3], DV-hop" [1], or Hop-TERRAIN [7]. The reason is that the second step of MDS-MAP, the application of classical MDS, is done without using the positioning information of anchor nodes. The information is only used in step 3, when the overall structure and distance ratios between nodes have already been determined. The approach of building a relative map irrespective of the co-ordinations of anchor nodes is double-edged. It works nicely when there are few or no anchor nodes, but not as well when there are more anchor nodes. One solution may be to use a more advanced MDS technique called the anchor point method [2], where coordinates of anchor nodes are explicitly used in determining the scaling.

As we mentioned above, combining MDS-MAP with other methods is another promising avenue. For example, MDS-MAP can be used to get good initial estimates of node positions, which is followed by a refinement phase like the ones in [4] or [8]. Due to the good performance of MDS-MAP comparing to competing methods on the cases of low anchor node densities, one can expect this two-phase approach to generate good results.

V. CONCLUSIONS

We conduct simulation to study the performance of classical MDS-MAP by varying the parameter radio range and number of anchor nodes. We found that more anchor nodes gives slightly smaller estimation error. Number of anchor affects the results when connectivity level is low. For high connectivity level, there is no evident improvement. Estimation error decreases as connectivity increases.

In this paper, we presented a new localization method, MDS-MAP that works well with mere connectivity information. However, it can also incorporate distance information between neighboring nodes when it is available. The strength of MDS-MAP is that it can be used when there are few or no anchor nodes. Previous methods often require well-placed anchors to work well. For example, the constraint-based approach in [4] works well only when the anchors are placed at the outside corners and edges and the constraints are tight. It works poorly when the anchors are inside the network, close to the center. The collaborative multi-lateration approach in [8] also requires anchors throughout the network, as well as a relatively large number of anchors, to work well. Our method does not have this limitation. It builds a relative map of the nodes even without anchor nodes. 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.

In future we plan to work for three dimensional networks and irregular topologies.

REFERENCES

[1] Niculescu, D., & Nath, B. (2003), *DV Based Positioning in Ad Hoc Networks*, Telecommunication Systems, 22(1), 267-280. Kluwer Academic Publishers.
[2] Xiang Ji and Hongyuan Zha, "*Sensor Positioning in Wireless Ad-hoc Sensor Networks using Multidimensional Scaling*," Proceedings of 23rd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2004), March 2004.
[3] Yi Shang and Wheeler Ruml, "*Improved mds-based localization*," in Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM), pp. 2640– 2651, 2004.
[4] T. Cox and M. Cox, Multidimensional Scaling, Chapman & and Hall, London, 1994.
[5] L. Doherty, K. S. J. Pister and L. El Ghaoui, "*Convex position estimation in wireless sensor networks*," IEEE INFOCOM, vol. 3, pp. 1655-1663, April 2001.
[6] D. Niculescu and B. Nath, *Localized positioning in ad hoc networks*, 2003.
[7] B. Stojkoska, D. Davcev, and A. Kulakov, "*Cluster-based MDS algorithm for nodes localization in wireless sensor networks with irregular topologies*," in Proceedings of the 5th International Conference on Soft Computing As Trans disciplinary Science and Technology (CSTST '08), pp. 384–389, Paris, France, October 2008.
[8] A. Savvides, H. Park, and M. Srivastava, *The bits and ops of the n-hop multilateration primitive for node localization problems*. In 1st ACM Int'l Workshop on Wireless Sensor Networks and Applications (WSNA'02), pages 112{121, Atlanta, GA, Sept. 2002.
[9] C. Savarese, J. Rabaey, and K. Langendoen, *Robust positioning algorithm for distributed ad-hoc wireless sensor networks*. In USENIX Technical Annual Conf., Monterey, CA, June 2002.