

MDS and Trilateration Based Localization in Wireless Sensor Network

Shailaja Patil, Mukesh Zaveri

Department of Computer Engineering, Sardar Vallabhbhai National Institute of Technology, Surat, India E-mail: {p.shailaja, mazaveri}@coed.svnit.ac.in Received March 31, 2011; revised April 27, 2011; accepted May 10, 2011

Abstract

Localization of sensor nodes is crucial in Wireless Sensor Network because of applications like surveillance, tracking, navigation etc. Various optimization techniques for localization have been proposed in literature by different researchers. In this paper, we propose a two phase hybrid approach for localization using Multidimensional Scaling and trilateration, namely, MDS with refinement using trilateration. Trilateration refines the estimated locations obtained by the MDS algorithm and hence acts as a post optimizer which improves the accuracy of the estimated positions of sensor nodes. Through extensive simulations, we have shown that the proposed algorithm is more robust to noise than previous approaches and provides higher accuracy for estimating the positions of sensor nodes.

Keywords: Wireless Sensor Network, Localization, Multidimensional Scaling, Trilateration

1. Introduction

The field of Wireless Sensor Network (WSN) is a multidisciplinary area of research offering a wide variety of applications ranging from home to industry, medical to military. Sensors integrated to structures [1], spread across a battlefield [2], and embedded in forest [3]; deliver the sensed information efficiently, to take further action for a particular application. Generally, these applications require large scale networks with hundreds of very small, battery powered and wirelessly connected nodes.

Intrinsically nodes have limitations of battery power, constrained communication computations and storage capability. When a large number of nodes are to be used in the applications, these sensor nodes are required to be cheaper, smaller in size, and robust to sustain environmental changes. Considering all these limitations, *i.e.*, scalability, storage, power; developing accurate and efficient localization algorithm is a challenging task.

Localization algorithm should possess characteristics of energy efficiency, distributed computation, scalability, and high precision. Radio signal propagation and message exchange consume considerable amount of energy of nodes, so algorithms need to run with least communication between nodes due to energy scarcity. The lifetime of network increases if there is balanced energy consumption of every node, and this requires an algorithm should preferably be distributed in nature. Also, it should be robust enough to noisy input, immune to node failure. Another important factor is that, it should be scalable, and even if the number of nodes is increased, algorithm's efficiency should not be hampered. Similarly, the localization algorithm needs to have high precision. The precision is measured by a ratio of position error and communication radius. Also, the algorithm should work with minimum density of reference nodes.

The reference nodes are known as anchor nodes. Usually, anchor nodes use GPS or are disposed manually. Manual disposition of large number of anchors is impossible for large scale WSNs. Also, as GPS nodes are expensive, there is limitation on use of more number of anchors, and so the algorithm should provide accurate estimation with low density of anchor nodes.

Considering all these factors, determining precise position of each node is a daunting task for a network with large number of nodes. It is unlikely that localization of each node can be done accurately. Hence a lot of research is being carried out for finding out accurate or near to accurate positions without using GPS support. Few techniques were developed to localize the nodes with GPS free positioning [4]. These algorithms yield a relative map of the nodes spread across the physical area. Such technique is useful in applications like direction based routing [5,6]. Few applications like geographical routing [7], target tracking [8], and localization [9], need exact positions of nodes. Multidimensional Scaling (MDS) is one of the techniques which yield two types of output maps called relative and absolute map [10]. The initial output of MDS is relative map and is converted to absolute map with sufficient known location of nodes.

MDS has its origin in psychometrics and psychophysics [11,12]. In literature, a number of localization techniques have been reported which use Multidimensional scaling. It is a set of data analysis techniques which display distance like data as geometric structure. This method is used for visualizing dissimilarity data. It is often used as a part of data exploratory technique or information visualization technique. MDS based algorithms are energy efficient as communication among different nodes is required only initially for obtaining the inter-node distances of the network. Once all distances are obtained, further computation for finding positions does not need communication among nodes. In this paper, we present an energy efficient hybrid algorithm for localization of nodes using MDS and trilateration. MDS is not only energy efficient but also provide good initial or starting points for any optimization technique [13].

In our proposed algorithm, initial locations are obtained using MDS and later these locations are refined using optimization method of trilateration by adjustment. Trilateration is basically a surveying technique which involves the determination of absolute or relative positions of points by measurement of distances, using the geometry of spheres or triangles [14]. Advantage of using trilateration by adjustment is that, it estimates locations accurately given good initial points. We have shown that, our proposed approach provides higher accuracy in estimating the sensor node positions as compared to previous approaches reported in the literature.

The next section presents a brief overview of related work in this area. In Section 3, the proposed algorithm is presented. Section 4 deals with the simulations results for isotropic topology. Conclusion is described in Section 5.

2. Related Works

Location awareness is of great importance for several wireless sensor network applications. Precise and quick self localization capability is highly desirable in wireless sensor network. Localization algorithms have been developed with various approaches. A detailed survey of localization techniques is provided in [15,16]. Localization techniques can be classified as *range free* or *range based*, depending on whether the range measurement methods are used or connectivity information is used. Range based methods require range measurement infor-

mation, such as Received Signal Strength Indicator (RSSI) [17], Angle of Arrival (AOA) [18], Time of Arrival (TOA) [18] and Time Difference of Arrival (TDOA) [18] etc. However, the measurement accuracy of these methods can be affected by the environmental interference [15]. Though, range free methods [19] cannot provide accurate location estimation, they are cost effective and robust to noise since range measurements are not involved in it. The range-based methods have connectivity or proximity information between neighbor nodes who can communicate with each other directly.

A triangulation based method is presented in "Adhoc Positioning System" (APS) by Niculescue and Nath [20]. Three methods are proposed by authors namely, DV-Hop, DV-Distance and Euclidean distance. In DV-Hop method only connectivity information is used, whereas in DV-Distance, the distance measurements between neighboring nodes are used, and Euclidean uses the local geometry of the nodes. Initially anchors flood their location to all the nodes in the network and every unknown receiver node performs triangulation to three other anchor nodes to estimate the position. These methods do not perform well with anisotropic or irregular network topology. Savarese [21] improved Niculescu's algorithm by introducing refinement phase. In this phase, distance measurements between neighboring nodes are used to improve localization accuracy. Though accuracy is improved significantly, it only works for well connected nodes. In another triangulation based approach [22], a technique of iterative multiplication is used. It provides good results if the number of anchor nodes is high. Nodes connected to 3 or more anchors compute their position by triangulation and upgrade their location. This position information is used by the other unknown nodes for their position estimation in the next iteration. Niculescue and Nath's algorithm of APS [20] is refined using trilateration by adjustment by Feng Tian and Wei Gao, called as LATN [23]. In this method with DV-Hop or DV-Distance the positions are estimated and trilateration is performed on unknown nodes for reducing the localization error.

Savvides [24] used least squares estimation with Kalman filter to locate the positions of sensor nodes to reduce error accumulation in the same algorithm. This method needs more anchors to work well than other methods. Doherty [25] used approach of convex optimization using semidefinite programming (SDP). The connectivity of the network has been represented as a set of convex localization constraints for the problem of optimization. This method works well, if anchor nodes are placed on the outer boundary, preferably at the corners. When all anchors are placed in the interior of the network, a large estimation error is obtained due to the shift of position estimate of outer nodes towards the center. Biswas [26] has extended the technique of Doherty's algorithm [25] by taking the non- convex inequality constraints. Basically, this technique converts the non- convex quadratic distance constraints into linear constraints with introduction of relaxation to remove the quadratic term of the equation. The distance measurements among nodes are modeled as convex constraints, and semidefinite programming (SDP) methods were adopted to estimate the location of nodes. Biswas's method was further improved by Tzu-Chen Liang [27], using a gradient search technique. The main disadvantage of semidefinite programming is amount of computation, which is $O(n^3 + c^3)$, where *n* is the number of rows (or columns) of the matrix and *c* is the number of constraints [28].

Shang *et al.* [13] presented a centralized algorithm based on MDS, namely, MDS-MAP(C). Initially, using the connectivity or distance information, a rough estimate of relative node distances is obtained. Then, MDS is used to obtain a relative map of the node positions and finally an absolute map is obtained with the help of anchor nodes. The initial location estimation is refined using least square technique in MDS-MAP(C,R) [13]. Both techniques work well with few anchors and reasonably high connectivity.

3. Proposed MDS-RT Algorithm

The refinement step in MDS-MAP(C,R) is slower than MDS itself. To further improve the accuracy and complexity of localization, we are proposing a hybrid algorithm here, namely, MDS with refinement using trilateration by adjustment (MDS-RT).

As discussed earlier in this paper, MDS is energy efficient localization method, so it has been used in the proposed algorithm. The accuracy of estimates obtained using MDS depends upon the accuracy of sensor observations. The poor accuracy of sensor observations results into poor estimates of locations. To overcome this problem, the method of trilateration by adjustment is used as an optimization tool. The estimates obtained using MDS are refined using trilateration method which increases the precision of estimates. The MDS-RT algorithm is described in the following section.

First, MDS technique is reviewed and introduced in brief. There are many types of MDS techniques and usually classified according to the way similarity/ dissimilarity data or matrix is formed. One way of classification is whether the similarities data are qualitative or quantitative. Qualitative and quantitative MDS are also known as Nonmetric, and Metric MDS respectively [10]. The Non-metric MDS (NMDS), also called as ordinal MDS, is developed by Shepard [12]. In NMDS, a monotonic relationship between inter-point distance and desired distance is established and assumed that data is measured at ordinal level.

According to the number of similarity matrices and the nature of the MDS model, MDS is classified as classical MDS (CMDS), replicated MDS (RMDS), and weighted MDS (WMDS). In CMDS single matrix is used whereas, RMDS and WMDS require several matrices. The RMDS uses unweighted matrices, whereas WMDS uses weighted matrices [10]. The distance model of weighted MDS uses different weight to each dimension. Another way of classification of MDS is deterministic or probabilistic [11]. In deterministic MDS each object is represented as single point whereas probabilistic MDS uses probability distribution in the MDS space.

In our proposed algorithm, we have used CMDS, where data is quantitative and object proximities are distances in Euclidean space. This algorithm consists of two phases; first phase is localization using MDS and second refinement using trilateration by adjustment.

3.1. Localization Using MDS

Nodes are randomly scattered in the region of consideration. Let p_{ij} refer to the proximity measure between objects *i* and *j*. The simplest form of proximity can be represented using Euclidean distance between two objects. This distance in *m* dimensional space is given by (1)

$$d_{ij} = \sqrt{\sum_{k=1}^{m} \left(X_{ak} - X_{bk} \right)^2}$$
(1)

Where, $X_a(x_{a1}, x_{a2}, \dots, x_{am})$ and $X_b(x_{b1}, x_{b2}, \dots, x_{bm})$ are distance vectors.

The steps of localization using MDS are illustrated as follows:

STEP 1: Obtain the distance between all pair of nodes using any ranging technique to construct distance matrix for MDS. Run the shortest path algorithm such as Floyd's or Dijktra's algorithm to get all distances and complete the distance matrix.

STEP 2: Compute the matrix of squared distances of proximities namely D^2 .

$$\boldsymbol{D}^{2} = \begin{vmatrix} 0 & d_{12}^{2} & \dots & \dots & d_{1n}^{2} \\ d_{21}^{2} & 0 & d_{23}^{2} & \dots & d_{2n}^{2} \\ \dots & \dots & 0 & \dots & \dots \\ \dots & \dots & \dots & 0 & \dots \\ d_{n1}^{2} & d_{n2}^{2} & d_{n3}^{2} & \dots & 0 \end{vmatrix}$$
(2)

STEP 3: Apply double centering to matrix of D^2 .

In CMDS, proximity matrix P *i.e.* D^2 is shifted to the center. Centering is placing the centroid of the configu-

ration of points at origin. It is required to overcome the indeterminacy of the solution due to arbitrary translation.

Double centering is done by multiplying both sides of (2) by centering matrix,

$$\boldsymbol{J} = \boldsymbol{I} \frac{1}{n} \boldsymbol{1} \boldsymbol{1}^{\mathrm{T}}$$
(3)

Here, I is Identity matrix of $n \times n$ size, where n is number of nodes and 1 is a vector of ones.

Double centering of squared distance matrix leads to equation (4)

$$\boldsymbol{B}_{dc} = \boldsymbol{J}\boldsymbol{D}^2\boldsymbol{J} \tag{4}$$

STEP 4: Compute singular value decomposition (SVD) of double centered matrix B_{dc} . Perform SVD on **B**-

$$\boldsymbol{B} = \boldsymbol{Q} \boldsymbol{A} \boldsymbol{Q}^{\mathrm{T}}$$
 (5)

Where $A = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n)$; is the diagonal matrix of eigen values and Q is the matrix of corresponding eigenvectors.

STEP 5: Modify SVD output of matrix *B* according to dimensions.

To get solution in lower dimension we have to retain the first m largest eigenvalues and eigenvectors. This is the best low rank approximation in the least-squares sense. For example, for a 2D network, consider the first 2 largest eigenvalues and eigenvectors to construct the best 2D approximation.

The modified B_{dc} now becomes **B**

$$\boldsymbol{B} = \boldsymbol{Q}_1 \boldsymbol{A}_1 \boldsymbol{Q}_1^{\mathrm{T}} \tag{6}$$

STEP 6: Obtain coordinates from matrix **B**.

The coordinate matrix can be estimated from B using Q_1 and A_1 as shown in following equation

$$\boldsymbol{X} = \boldsymbol{Q}_1 \boldsymbol{A}_1^{1/2} \tag{7}$$

This coordinate matrix gives relative map.

STEP 7: Transform relative map to absolute map using anchor nodes.

The recovered matrix X is rotated, translated, shifted as it has different orientation than original.

3.2. Refinement Using Trilateration by Adjustment

Once the estimation of distance matrix using RSSI and consequently the location estimation by MDS are performed, the next step is to refine the locations. For this task the technique of trilateration by adjustment is used as post optimization tool. The trilateration refines the estimates using successive adjustments. Here, every link is assigned a weight according to the signal quality.

First, trilateration by adjustment is reviewed in brief and then steps of refinement are illustrated.

Consider Figure 1, consisting of two points *P* and *Q*.

$$P(X_p, Y_p)$$
 o $Q(X_q, Y_q)$

Figure 1. Link PQ.

Let the estimated coordinates of P and Q be $P(X'_p, Y'_p)$, $Q(X'_q, Y'_q)$ and true coordinates be $P(X_p, Y_p)$, $Q(X_q, Y_q)$ respectively. The distance between them is link value of PQ expressed as L'. The true distance *i.e.* Euclidean distance can be expressed as in (8)-

$$L_{i} = \sqrt{\left(X_{p} - X_{q}\right)^{2} + \left(Y_{p} - Y_{q}\right)^{2}}$$
(8)

The relation between X_p and X'_p and remaining coordinates is given by following set of equations.

$$X_{p} = X'_{p} + x_{p}, Y_{p} = Y'_{p} + y_{p}$$

$$X_{q} = X'_{q} + x_{q}, Y_{q} = Y'_{q} + y_{q}$$
(9)

In (9), variables x_p, y_p, x_q, y_q are the corrections in estimated coordinates, which give us adjusted values in link *L*.

$$L_i = L'_i + v_i \tag{10}$$

Where v_i is correction in L'_i . According to Taylor's expansion, above equation can be written as:

$$L'_{i} = Z'_{pq} + \frac{X'_{q} - X'_{p}}{Z'_{pq}} \left(x_{q} - x_{p} \right) + \frac{Y'_{q} - Y'_{p}}{Z'_{pq}} \left(y_{q} - y_{p} \right)$$
(11)

Where

$$Z'_{pq} = \sqrt{\left(X'_{p} - X'_{q}\right)^{2} + \left(Y'_{p} - Y'_{q}\right)^{2}}$$
(12)

Let the correction in link PQ be l, and relation between l and v can be obtained using (10) and (11):

$$v'_{i} = \left(-l'_{i}\right) + \frac{X'_{q} - X'_{p}}{Z'_{pq}} \left(x_{q} - x_{p}\right) + \frac{Y'_{q} - Y'_{p}}{Z'_{pq}} \left(y_{q} - y_{p}\right) \quad (13)$$

Where
$$l'_i = (L'_i - Z'_{pq})$$

Consider **Figure 2**, consisting of known nodes $A(X_{1, Y_1})$, $B(X_{2, Y_2})$, and $C(X_{3, Y_3})$ and unknown node $D(X_{4, Y_4})$. The links between nodes are ad, bd, cd, ab, ac, and bc.

As seen before, the link value obtained by ranging device may not be equal to Euclidean distance due to presence of noise. Let variances of the link *ad*, *bd*, *cd* be σ_{ad} , $\sigma_{bd} \sigma_{cd}$, and σ_0 be the variance of unit weight which is selected arbitrarily.

From these values link weight can be computed as -

$$W_i = \frac{\sigma_0^2}{\sigma_i^2} \quad (i = ad, bd, cd) \quad (14)$$

STEP 1: Form a cluster of four nodes as shown in **Figure 2**, Obtain the weight matrix *W*.



Figure 2. Trilateration.

The Euclidean distance between known and unknown nodes of **Figure 2** can be expressed as-

$$d_{i} = \sqrt{\left(X_{4} - X_{i}\right)^{2} + \left(Y_{4} - Y_{i}\right)^{2}}$$
(15)

Where *i* vary from 1 to 3.

The objective function can be formed from (15).

$$f_i = d_i - \sqrt{\left(X_4 - X_i\right)^2 + \left(Y_4 - Y_i\right)^2} = 0 \qquad (16)$$

STEP 2: The first estimate of positions is obtained with first phase. These values are used as initial values of non-anchor node (X_4 , Y_4).

STEP 3: Using link value between all the four nodes and objective function as given in (16), obtain Jacobian matrix J and f vector of objective functions.

$$\boldsymbol{B} = \begin{bmatrix} \partial F_1 / \partial X & \partial F_1 / \partial Y \\ \partial F_2 / \partial X & \partial F_2 / \partial Y \\ \partial F_3 / \partial X & \partial F_3 / \partial Y \end{bmatrix}$$
(17)

Functions F1, F2, F3 are evaluated at X_4 and Y_4 . The vector matrix f can be obtained by:

$$\boldsymbol{f} = \begin{bmatrix} F_1(X_4, Y_4) \\ F_2(X_4, Y_4) \\ F_3(X_4, Y_4) \end{bmatrix}$$
(18)

Now, we will compute corrections in coordinates (Δ) and in link value (v).

STEP 4: Obtain correction vector $\mathbf{\Delta}$, of the coordinates (*X*₄, *Y*₄). Solving (16) for finding corrections $\mathbf{\Delta}$ in (*X*₄, *Y*₄), we get

$$\boldsymbol{\Delta} = \begin{bmatrix} X_{4c}, Y_{4c} \end{bmatrix} \tag{19}$$

Where Δ is computed by following equation

Copyright © 2011 SciRes.

$$\boldsymbol{\Delta} = \left(\boldsymbol{B}^{\mathrm{T}} \boldsymbol{W} \boldsymbol{B}\right)^{-1} \left(\boldsymbol{B}^{\mathrm{T}} \boldsymbol{W} \boldsymbol{f}\right)$$
(20)

The final estimated coordinates are-

$$\left(X_4, Y_4\right)_{new} = \left(X_4, Y_4\right)_{old} + \Delta \tag{21}$$

STEP 5: Obtain correction vector \boldsymbol{v} to the link

The corresponding correction in distance v can be obtained from above equation as-

$$\boldsymbol{v} = \boldsymbol{f} - \boldsymbol{B}^* \boldsymbol{\Delta} \tag{22}$$

So the estimated values of distance for correction in (X_4, Y_4) is

$$\hat{d}_i = d_i + v_i \tag{23}$$

Here v can also be obtained by following equation

$$v_{i} = \sqrt{\left(X_{4new} - X_{4old}\right)^{2} + \left(Y_{4new} - Y_{4old}\right)^{2}}$$
(24)

STEP 6: Estimate new link value for (X_4, Y_4)

$$[X_{4new}, Y_{4new}] = [X_4, Y_4] + [X_{4c}, Y_{4c}]$$
(25)

STEP 7: Keep iterating for new values of (X_4, Y_4) till Δ vector values *i.e.* correction in coordinates does not reach the predetermined precision value or given number of iterations are not completed.

The refined estimate of position of a non-anchor node obtained using the steps described above allow us to use this node as an anchor node for refining the positions of other non-anchor nodes.

STEP 8: Repeat steps 1 to 7 for remaining non-anchor nodes.

Thus, correction in position is determined by link measurement error and number of iterations is determined by initial position error of nodes. If this error is less, number of iterations required is less. Also, if link measurement precision is high, the final position error will be low. The trilateration method itself suffers from poor accuracy due to errors in ranging measurements. Our proposed algorithm overcomes this problem as initial locations are estimated by MDS algorithm and this serve as good initial estimates for trilateration method.

The advantages of our proposed algorithm are:

1) The position accuracy is high.

2) The algorithm relies on no explicit communication other than that between immediate neighbors, which avoids excessive communication.

3) It is robust even when the measured distance between the neighboring nodes is degraded with noise.

3.3. Complexity Analysis

First Phase (MDS-MAP(C)): In this algorithm, to complete the distance matrix, shortest path (Floyd's) algorithm is applied. Its time complexity is $O(N^3)$, where N is the number of nodes. MDS uses singular value decomposition (SVD) of matrix **B**. The complexity of SVD is $O(N^3)$. The result of MDS is a relative map that gives location of nodes, relative to each other. The relative map is transformed to absolute map through a linear transformation, which may include scaling, rotation and reflection. Computing the transformation parameters takes $O(k^3)$ time, where k is the number of anchors. Applying the transformation to the whole relative map to obtain absolute map takes O(N) time.

Refinement Phase (MDS-MAP(C,R)): Refinement phase use Levenberg-Marquardt algorithm. Its complexity is $O(N^3)$, where N is number of nodes.

Refinement Phase (MDS-RT): The basic trilateration algorithm does j iterations (here 10) in worst case. For Nnumber of nodes j(N-3) times the algorithm will run. If the radio range of anchor is less and all nodes are not in range of three anchors, then nearest (at a distance of single hop) localized node will start acting as pseudo anchor node. Thus in worst case the complexity will be $O(jN^2)$ as the algorithm will run for j(N-3)(N-3) times and in best case complexity will be O(jN).

4. Experimental Results

We have performed exhaustive simulations with varying number of nodes. Typically, the number of nodes is varied from 50 to 200. These nodes are placed randomly with a uniform distribution in a square area of $10l \times 10l$ of *l* unit length. The performance of MDS-RT has been examined for various radio ranges with different range errors, anchors and node densities. Radio range is blurred with noise to introduce error in radio link. Presently the *distance-only* case of MDS-MAP is considered, where each node knows distance to each neighbor node. An uniform radio propagation is considered. The performance measure used for evaluation is root mean square error (RMSE), as shown in (26).

$$RMSE = \sqrt{\sum \left[\left(X_i - X_j \right)^2 + \left(Y_i - Y_j \right)^2 \right]} / \sqrt{N} \quad (26)$$

Where (X_i, Y_i) are estimated locations, (X_j, Y_j) are corresponding true locations, and N is number of nodes.

As radio range increases, connectivity increases, leading to connecting more number of nodes of the network. For observing effect of increase in range error and connectivity on RMS error, the range error is increased from 5% to 50% of communication range and connectivity is increased from 21.5 to 45 for 200 nodes. This allows evaluating our proposed method with worst scenarios. The simulations are performed with varying number of anchor nodes from 4 to 10. Following section describes the effect of various factors like different node density, range error, connectivity and anchor nodes on RMSE. We performed Monte-Carlo simulations and the number of simulations for each experiment is set to 50. Errors are normalized with radio range. We have simulated and compared performance of our proposed algorithm MDS-RT with standard algorithms MDS-MAP(C) and MDS-MAP(C,R) respectively.

Figure 3 shows 50 and 100 nodes randomly placed in $10l \times 10l$ square area. Lines between points show connectivity of nodes at connectivity level of 29 and 26.1.

Above figure shows connectivity of 50 and 100 nodes with 20% error in radio range. Here, the connectivity or connectivity level represents the average number of nodes connected with each other in the network, *i.e.*, the average number of nodes within the radio range of a particular node.

Figure 4 shows the connectivity information for 200 nodes with connectivity level of 21.5 and 10% range error. Figure 3 & 4 represent the scenarios with low node density to high node density. In these figures, line joining nodes are connectivity links.

Figure 5(a) & (b) show scatter-plots of 200 nodes



Figure 3. Network connectivity with (a) 50 and (b) 100 nodes.



Figure 4. 200 nodes placed randomly at connectivity of 21.5 and 10% error.



Figure 5. Scatter plot of 200 localized nodes with (a) MDS-MAP(C) and (b) MDS-RT.

with connectivity of 21.6 and range error of 20% for MDS-MAP(C) and MDS-RT respectively. Original positions are shown by symbol "O", estimated positions by "*" (asterisk) and anchor positions by solid diamonds. Black lines show error. Longer the line, larger is the error. Same nomenclature is applicable to all the remaining scatter-plots. The RMSE obtained in Figure 5(a) is 18.5% with MDS-MAP(C) and in Figure 5(b), it is 0.04% with MDS-RT.

With increase in radio range, connectivity between nodes is increased and more number of nodes participates in network formation. **Figure 6** shows connectivity Vs RMSE with 4 anchor nodes and 5% error in radio range for all the three algorithms. RMS Error decreases from 11.4% to 2.7% for MDS-MAP(C) when connectivity level is increased from 21.63 to 43.19. Similarly, reduction in RMSE can be observed for MDS-MAP(C,R) and MDS-RT as well. The RMSE obtained for MDS-MAP(C,R) and MDS-RT at connectivity of 26.2 are 1.25% and 0.27% respectively. With connectivity level of 43.19 these values are 0.83% and 0.033% respectively.

As the number of anchor nodes is increased the total RMSE decreases. **Figure 7** shows a comparative plot of RMSE vs. connectivity with 5% noise and 6 anchors. The average RMSE obtained at connectivity of 39.06 for MDS-MAP(C,R) is 1.11% and for MDS-RT it is 0.05% respectively which is a very large reduction in RMSE using our proposed algorithm. This result shows that our proposed MDS-RT outperforms MDS-MAP(C,R) algorithm.

A significant amount of reduction is observed in RMSE of MDS-RT as the anchor nodes are increased from 6 to 10 which is depicted in **Figure 8**. At connectivity of 26.2 and 43.19 the amount of RMS Error is [1.67%, 0.68%] for MDS-MAP(C,R) and [0.129%,



Figure 6. RMSE vs. connectivity with 5% range error and 4 anchors.

204



Figure 7. RMSE vs. connectivity with 5% error and 6 anchors.



Figure 8. RMSE vs. connectivity level with 5% range error and 10 anchors.

0.0002%] for MDS-RT respectively. Thus the accuracy increases by about 45% for connectivity of 43.19.

The efficiency and excellent performance of the proposed algorithm is very much evident from these graphs.

Specifically when number of anchor nodes is 10, RMS error obtained is negligible and it is shown as zero here. When simulations were performed for 10% range error, RMSE obtained at connectivity of 21.6 is 4.2e-04, and as the radio range goes on increasing, the error goes on decreasing. At connectivity level of 45, error is reduced to 2.1e-05.

Next, we will analyze performance of algorithm in presence of various levels of noise. It is obvious that with increase in noise level, RMSE will also increase. The comparative plot of effect of increase in range error on RMSE using all three methods is shown in **Figure 9**. It is obtained at connectivity level of 29.1 with 6 anchors.



Figure 9. Effect of increase in rage error on RMSE.

When the range error is increased from 5% to 50% (worst scenario), the RMSE increases from 3.9% to 8.1% for MDS-MAP(C), 0.87% to 4.45% for MDS-MAP(C,R) and 0.067% to 2.35% for MDS-RT respectively.

With increase in the number of anchors, the location estimate is more accurate and reduction in RMSE is observed.

When range error is increased to 10% the effect on RMSE is seen as shown in **Figure 11**. Comparing **Figures 10** and **11**, it can be observed that, the RMSE of MDS-MAP(C) increases to 9.6% from 9.2%, when error increases from 5% to 10% with 4 anchors. With MDS-MAP(C,R), RMSE increases from 1.3% to 1.5%, and with MDS-RT it increases from 0.4% to 0.7% respectively. With 10 anchors, RMSE of MDS-RT increases from 0.028% to 0.038% whereas MDS-MAP(C,R) increases from 0.5% to 0.9% respectively.

Table 1 shows the performance of all three algorithms with 4, 6, and 10 anchors. The data shows the effect on RMSE with 5% and 10% range error at connectivity of



Figure 10. Effect of increase in number of anchors on RMSE with 5% range error.



Figure 11. Effect of increase in number of anchors on RMSE with 10% range error.

Table 1	Perf	formance	comnar	ison at	conn	ectivi	tv o	f 76	\$ 2
I abit I		ormance	compai	150ff at	conn	ccuvi	ty U	1 20	

	Range Error (%)	RMSE(%)						
MDS MAP(C)								
1 Anahora	5	0.2						
4 Anchors	10	9.2						
(A als a sec	10	9.34 5.2						
6 Anchors	5	5.5						
	10	/.3						
10 Anchors	5	3.5						
	10	3.7						
MDS-MAP(C,R)								
4 Anchors	5	1.25						
	10	1.8						
6 Anchors	5	0.9						
	10	1.35						
10 Anchors	5	0.6						
	10	0.83						
MDS RT								
4 Anchors	5	0.26						
	10	1.01						
6 Anchors	5	0.21						
0 Anchors	10	0.21						
10 4 1	10	0.85						
10 Anchors	5	0.023						
	10	0.031						

26.2. The RMSE is normalized to radio range and percentage error is computed.

As it is clear from above figures and **Table 1**, our proposed method performs better than MDS-MAP(C,R). As can be seen from **Figure 11**, the error approaches nearly to zero even in the case of 10% range error. The remarkable difference is seen when number of anchor nodes are high *i.e.* > 5 and connectivity > 26.1.

We have also performed the simulation of placing anchor nodes colinearly for 50, 100 and 200 nodes. Placement of anchors plays a very important role in any localization algorithm. This is very important to note that the trilateration method fails, if anchor nodes are collinearly located. The proposed algorithm MDS-RT overcomes this problem. As MDS provides good initial estimate of locations, even though the anchor nodes are collinearly located, trilateration refines the estimated locations only. If anchors are not in radio range of each other or non-anchor nodes are not in the range of anchor nodes then algorithm may give erroneous results. However, this problem can be overcome with dense network topology.

Figure 12 shows the localization using MDS-MAP(C) and MDS-RT for 50 nodes when nodes are placed colinearly. At connectivity of 21.6, with 20% range error and 4 number of anchor nodes, the RMS Error is 50% for MDS-MAP(C) and 15% for MDS-RT. As the density of nodes increases, with less radio range also nodes start communicating as they get enough connectivity.

Figure 13 and **14** show simulation for 100 and 200 nodes with 20% range error. At connectivity of 26.1 with range error of 20%, RMSE obtained is 37% and 0.2% for MDS-MAP(C) and MDS-RT respectively for 200 nodes.

From all the graphs and scatter-plots, it is clear that our proposed algorithm MDS-RT outperforms MDS-MAP(C,R) algorithm. If accuracy is not to be compro-



Figure 12. (a)(b): Scatter plot of 50 nodes for MDS-MAP(C), and MDS-RT.



Figure 13. (a)(b): Scatter plot of 50 nodes with MDS-MAP(C), MDS-RT.



LOCALIZATION WITH MDS-C WITHOUT REFINEMENT



Figure 14. (a)(b): Scatter plot of 200 nodes localized with MDS-MAP(C) and MDS-RT.

mised, then at the cost of increased radio range or more anchors, the algorithm performs best.

5. Conclusions

Localization using Multidimensional Scaling is one of the robust algorithms in the literature of localization in sensor network. MDS-MAP(C) has proved its efficiency as compared to other algorithms like SDP, APS etc. Error in basic MDS-MAP(C) has been improved in MDS-MAP(C,R) algorithm. However, MDS-MAP(C,R) is computationally intensive. We have shown that, if refinement step is performed using trilateration by adjustment (MDS-RT), the algorithm not only becomes computationally light weight but also significant error reduction is observed and more accurate results are obtained. The current work assumes that the network is isotropic. The proposed method may be extended to a network with irregular topology as a future work.

6. References

- M. Li and L. Yunghao, "Underground Structure Monitoring with Wireless Sensor Networks," *Proceedings of International Symposium on Information Processing in Sensor Networks*, Cambridge, 25-27 April 2007, pp.69-78. doi:10.1109/IPSN.2007.4379666
- [2] N. Alsharabi, L. R. Fa, F. Zing and M. Ghurab, "Wireless Sensor Networks of Battlefields Hotspot Challenges and Solutions," *Proceedings of the 6th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks and Workshops*, Berlin, 1-3 April 2008, pp.192-196. doi:10.1109/WIOPT.2008.4586064
- [3] M. Hefeeda and M. Bagheri, "Wireless Sensor Networks for Early Detection of Forest Fires," *Proceedings of In*ternational Conference on Mobile Ad Hoc and Sensor

Systems, Pisa, 8-11 October 2007, pp. 1-6. doi:10.1109/MOBHOC.2007.4428702

- [4] N. Bulusu, J. Heidemann and D. Estrin, "GPS-Less Low Cost Outdoor Localization for Very Small Devices," *IEEE Transactions on Personal Communication*, Vol. 7, No. 5, 2002, pp. 28-34.
- [5] M. Chu, H. Haussecker and F. Zhao, "Scalable Information-Driven Sensor Querying and Routing for Ad Hoc Heterogeneous Sensor Networks," *International Journal* of High Performance Computing Applications, Vol. 16, No. 3, 2002, pp. 1-22. doi:10.1177/10943420020160030901
- [6] B. Karp and H. T. Kung, "GPSR: Greedy Perimeter Stateless Routing for Wireless Networks," *Proceedings of* the 6th International Conference on Mobile Computing and Networks (ACM Mobicom), 2000, pp.1-10.
- [7] Y. Yu, R. Govindan and D. Estrin, "Geographical and Energy Aware Routing: A Recursive Data Dissemination Protocol for Wireless Sensor Networks," *University of California, Los Angeles Computer Science Department Technical Report UCLA/CSD-TR-*01-0023, May 2001. http://citeseerx.ist.psu.edu
- [8] Z. Guo and M. C. Zhou, "Optimal Tracking Interval for Predictive Tracking in Wireless Sensor Network," *IEEE Communication Letters*, Vol. 9, No. 9, 2005, pp. 805-807. doi:10.1109/LCOMM.2005.1506709
- [9] Y. Shang, W. Ruml, Y. Zhang and M. Fromherz, "Localization from Mere Connectivity," *The 4th ACM International Symposium on Mobile and Ad-Hoc Networking & Computing Symposium on Mobile and Ad-Hoc Networking & Computing*, 2003, pp. 201-212.
- [10] I. Borg and P. Groenen, "Modern Multidimensional Scaling, Theory and Applications," Springer, New York, 1997.
- [11] W. S. Torgeson, "Multidimensional Scaling of Similarity," *Psychometrika*, Vol. 30, No. 4, 1965, pp. 379-393. doi:10.1007/BF02289530
- [12] R. N. Shepard, "The Analysis of Proximities: Multidimensional Scaling with an Unknown Distance Function," *Psychometrika*, Vol. 27, No. 2, 1962, pp. 125-140. <u>doi:10.1007/BF02289630</u>
- [13] Y. Shang, W Ruml, Y. Zhang and M. Fromherz, "Localization from Connectivity in Sensor Networks," *IEEE Transactions on Parallel and Distributed Systems*, Vol. 15, No. 11, 2004, pp. 961-974. doi:10.1109/TPDS.2004.67
- [14] T. Hornoch, "Notes on the Adjustment of Trilateration," Survey Review, Vol. 18, No. 135, 1965, pp. 14-18.
- [15] G. Q. Mao, B. Fidan and B. D. O. Anderson, "Wireless sensor Network Localization Techniques," *The International Journal of Computer and Telecommunications Networking Computer Networks*, Vol. 51, No. 10, 2007, pp. 2529-2553.
- [16] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci, "A Survey on Sensor Networks," *Computer Networks*, Vol. 40, No. 8, 2002, pp. 393-422. doi:10.1016/S1389-1286(01)00302-4

- [17] X. Li, H. Shi and Y. Shang, "A Sorted RSSI Quantization Based Algorithm for Sensor Network Localization," *Proceedings of 11th International Conference on Parallel* and Distributed Systems, 20-22 July 2005, pp. 557-563.
- [18] P. Xing, H. Yu and Y. Zhang, "An Assisting Localization Method for Wireless Sensor Networks," *Proceedings of* the Second International Conference on Mobile Technology, Applications and Systems, Guangzhou, 15-17 November 2005, pp. 1-6.
- [19] T. He, C. Huang, B. Blum, J. Stankovic and T. Abdelzaher, "Range-Free Localization Schemes for Large Scale Sensor Networks," *Proceedings of the Ninth Annual International Conference on Mobile Computing and Networking (ACM Mobicom)*, San Diego, September 2003, pp. 81-95.
- [20] D. Niculescu and B. Nath, "Ad Hoc Positioning System," Proceedings of the Global Telecommunications Conference, San Antonio, 25-29 November 2001, pp. 2926-2931.
- [21] C. Savarese, J. Rabay and K. Langendoen, "Robust Positioning Algorithms for Distributed Ad-Hoc Wireless Sensor Networks," USENIX Technical Annual Conference, June 2002, pp. 1-10.
- [22] A. Savvides, C. Han and M. B. Srivastava, "Dynamic Fine-Grained Localization in Ad-Hoc Networks of Sensors," *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking (ACM Mobicom)*, 2001, pp. 166-179.
- [23] F. Tian, W. Guo, C. Wang and Q. Gao, "Robust Localization Based on Adjustment of Trilateration Network for Wireless Sensor Networks," *Proceedings of 4th International Conference on Wireless Communications, Networking and Mobile Computing*, 12-14 October 2008, pp. 1-4.
- [24] A. Savvides, H. Park and M. B. Srivastava, "The Bits and Flops of the N-Hop Multilateration Primitive for Node Localization Problems," *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*, Atlanta, 8 September 2002, pp. 112-121.
- [25] L. Doherty, K. pister and L. El. Ghaoui, "Convex Position Estimation in Wireless Sensor Networks," *Proceedings of the Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies*, Anchorage, 22-26 April 2001, pp. 1655-1663.
- [26] P. Biswas and Y. Ye, "Semidefinite Programming for Ad Hoc Wireless Sensor Network Localization," *Proceedings of the Third International Symposium on Information Processing in Sensor Network*, new York, 26-27 April 2004, pp. 46-54. doi:10.1145/984622.984630
- [27] T. C. Liang, T. C. Wang and Y. Ye, "A Gradient Search Method to Round the Semidefinite Programming Relaxation Solution for Ad Hoc Wireless Sensor Network Localization," Stanford University, Formal Report 5, 2004. http://www.stanford.edu/-yyye/ formal-report5.pdf
- [28] B. Borchers, "CSDP-A C Library for Semidefinite Programming," Optimization Methods and Software, Vol. 11, No. 1, 1999, pp. 613-623. doi:10.1080/10556789908805765