A Centralized Localization Algorithm Based on Mesh Relaxation in Wireless Sensor Networks

Zhixiong Liang, Xingcheng Liu
Department of Electronic and Communications Engineering
Sun Yat-sen University
Guangzhou, Guangdong, 510275, China
E-mail: isslxc@mail.sysu.edu.cn

***Abstract* - Many applications of Wireless Sensor Networks (WSNs) require sensor nodes know their positions. Equipping GPS modules can solve this problem easily, but it is of high cost. In this paper, we propose a centralized localization algorithm for WSN based on mesh relaxation method. Computer simulations show that with our algorithm the positions of nodes can be estimated in a relative coordinate system, which can be transformed to the global coordinate system with the aid of a few anchor nodes. The proposed algorithm can avoid folds and flips through firstly stretching the mesh of the network by pulling some fringe nodes towards different directions, and then making the mesh to shrink by relaxing it. The position results achieved have higher precision than those with other positioning methods. Our algorithm can be used based on ranging or connectivity.**

Key words - Localization; position estimation; mesh; wireless sensor network

I. Introduction

Due to advances in communications and MEMS [1] technology, wireless sensors are enabled to be deployed in large quantities to form a Wireless Sensor Network (WSN) for wide applications [2]-[4]. The sensor nodes work cooperatively and aggregate large quantity of sensor data to provide continuous monitoring for various kinds of environmental characteristic. WSN has been applied in many areas such as traffic management, medical treatment, ecological studies, disaster fighting, etc. Meanwhile, many applications of WSNs require sensor nodes to know their positions. Many other higher layer services like routing [5], data aggregation, and event reporting are built on a localization service in the network. So it is essential to build a localization service for WSN. However, it is not realistic to manually configure location coordinate for each node in a network containing thousands of nodes. Equipping a GPS module for each node can solve this problem easily, but it is of high cost. Furthermore, GPS does not work in an indoor environment. So a self-localization mechanism for WSN is necessary to be designed. In recent years, many localization algorithms have been proposed. Most of them assume that there are *m* (≥3) anchor nodes, which are equipped with GPS modules and their positions are already known, in the network. Other non-located nodes are located based on the anchor nodes and the estimated distance (e.g. TOA, TDOA, RSSI) or connectivity between nodes.

In this paper, we present a localization algorithm. Given *n* sensor nodes distributed in a plane with *m* (≥3) anchor nodes among them and the connection information between one-hop nodes, the proposed algorithm is able to locate the nodes based on ranging or connectivity.

The proposed algorithm is a centralized one. That is, it demands that all the nodes send the required information for localization to one point (e.g. base station), and this point operates the localization algorithm to estimate the position for each non-located node. Certainly, an isolated node which cannot connect with the network cannot be located. The information required for the localization algorithm contains all nodes' ID and their one-hop neighbors' ID, besides the positions of the anchor nodes. Moreover, the information contains the measured distances between one-hop neighbors if it is based on ranging. The proposed algorithm is based on mesh relaxation which uses the nonlinear optimization to deal with localization problems. But it has been demonstrated that it will lead to local minima if this method was directly operated with a random initial coordinate for each node. These local minima correspond to folded or collapsed layouts of the original network [6]. In the proposed algorithm four fringe nodes are to be found and a centre node of the network is to be determined. After stretching the mesh of the network by pulling fringe nodes towards different directions, the mesh is to be shrunk by relaxing it. We will show that it is possible to avoid folds and flips with this method, especially in a uniform topology. Using the connections between neighbors, the proposed algorithm, Stretch algorithm, can estimates the positions for nodes in a relative coordinate system that can be transformed to a global one through a few anchors. In fact, a distributed algorithm could obtain a good scalability and reduce the computation complexity. However, there are many centralized systems in wide applications. So, it is desirable to design a centralized localization algorithm for them, in which case it generally has a higher accuracy than a distributed one.

The rest of this paper is organized as follow: section II introduces the related work, and section III describes our method. Our simulations and analyses will be presented in section IV while in section V, a conclusion will be drawn.

The project was supported by the National Natural Science Foundation of China (Grant No. 60673086, 60970041). X. Liu is the corresponding author.

CHINACOM 2010, August 25-27, Beijing, China
Copyright © 2011 ICST 973-963-9799-97-4
DOI 10.4108/chinacom.2010.48

II. Related Work

Building accurate self-localization service is highly desirable in WSN for various kinds of applications. Many algorithms have been proposed to solve this problem.

Some range-free distributed algorithms, such as Centroid algorithm [7], APIT [8], CAB [9], are easy to be implemented and of high scalability. The main idea of these algorithms is that each non-located node hears beacon signals from nearby anchors, and even exchanges information with other non-located nodes to estimate the constraint region where it locates. Then, the non-located node calculates the geometric centroid of the constraint region as its estimated position. The localization accuracy of these algorithms relies on the number of anchors.

Centralized algorithms generally require all nodes of the network or subnetwork send the information used for localization to one point (e.g. base station). Some of them use the estimated distance between nodes as constraints and systematically search the coordinate of each node in a relative coordinate system to satisfy those constraints. The node coordinates can be transformed to global coordinate system utilizing some (≥3) anchors. Mesh relaxation is exactly such a method that it uses nonlinear optimization to relax the mesh of network to find the coordinates of the nodes that meet the constraints of estimated distances. But as description in [6], there exists various kinds of folded or collapsed local layouts in the network. If using mesh relaxation to directly release the mesh with random initial node coordinates, it has a high probability to lead to a false minimum. So it is essential to find a special method to search the initial coordinate for each node to solve this problem before relaxing the mesh. In [10], it has been demonstrated that multidimensional scaling (MDS) is good at finding the similar topology of the network compared to the original one. The proposed algorithm, MDS-MAP, uses the shortest path between any two nodes as estimated distance and utilizes the MDS technique to find the coarse coordinate of the nodes. In [11], MDS-MAP(P) and MDS-MAP(P,R) algorithm use the result of the MDS-MAP as initial input and the connections within two-hop neighbors to relax the mesh. The idea that breaks up the whole network into many small local maps, and then locates for each local map, and at last merges the local maps to form an integrated network makes the two algorithms applicable in irregular networks. In [12], KKLoc algorithm calculates the shortest path between any two nodes as estimated distance to create the connections between all nodes, which yields a full connected graph, besides sets different weight values for different hop-count connections, then uses a graph drawing approach to find the similar topology of the network compared to the original one. At last, KKLoc uses the result above as the initial input and the real connections between nodes to relax the mesh. MDS-MAP(P), MDS-MAP(P,R), KKLoc algorithm are able to avoid folds and flips.

In this paper we propose a simple centralized algorithm, Stretch algorithm, to estimate the positions of nodes for WSN through mesh relaxation. We will show that how it can avoid folds and flips.

III. Stretch Algorithm

In this section, Stretch algorithm is present in detail. We assume that after the network is deployed in a plane. All nodes send the information required for localization to a central point (e.g. base station). The information contains IDs of nodes and their one-hop neighbors' IDs along with the estimated distance between them, besides the coordinates of the anchors. If the sensor nodes are equipped with distance-measuring hardware, the estimated distance is the measured distance between one-hop neighbors, and if not, the localization is based on connectivity. Define $G(V, E)$ to be the graph that we want to draw. $G(V, E)$ contains n vertices V, which correspond to n sensor nodes including m anchors, and edges E, which correspond to the connections between nodes. For a connected graph, it must be $m \geq 3$ that the relative coordinate system of G can be transformed to the global one if it is needed. That is our algorithm can used as an anchor-free one or an anchor-based one.

Stretch algorithm is based on mesh relaxation. $G(V, E)$ can be modeled as a mass-spring mesh in the way that the vertices are replaced with masses and the edges are replaced with springs of whom the normal rest lengths equal to the length of the corresponding edges [12][13]. In a real physical situation, the mass-spring system will relax to its lowest energy configuration and the masses will be push to the correct positions. Come back to our problem, we aim to relax the virtual mass-spring system so as to draw the corresponding $G(V, E)$ whose topology satisfies the original network avoiding folds and flips and in whom the length of edges satisfy the estimated distances. Our method mainly contains 4 processes: search process, stretch process, shrink process and refinement process.

Search process: to search for four fringe nodes respectively in four directions in the network and one centre node of the network. If the fringe nodes have been configured manually, skip this process, and if not, we use a similar method described in [6] with a little difference. In the following description, h_{is} is defined as the smallest-hop-count value between node n_i and the *selected* node. There may be several nodes that are eligible for selection. In order to select the unique node, two basic rules are defined as follows:

Rule 1: among the candidates, select the nodes with the smallest one-hop-neighbor number, and then among the new candidates select a node with the smallest ID.

Rule 2: among the candidates, select the nodes with the largest one-hop-neighbor number, and then among the new candidates select a node with the smallest ID.

The way to search the fringe nodes and the centre node is:

1) Select a node n_0 randomly or the one with the smallest ID for simple. Then, select the nodes to maximize h_{0s}. At last, select a node n_1 using rule 1.
2) Select the nodes to maximize h_{1s}. Then, select a node n_2 using rule 1.
3) Select the nodes to minimize $|h_{1s}-h_{2s}|$, and then among the candidates select the nodes to maximize $|h_{1s}+h_{2s}|$. At last, select a node n_3 using rule 1.
4) Select the nodes to minimize $|h_{1s}-h_{2s}|$, and then among the

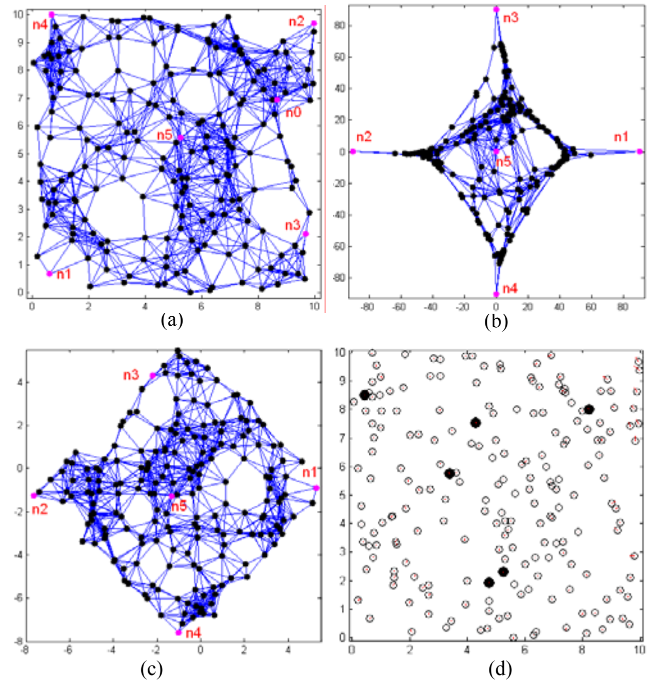


Figure 1. Intermediate outputs of Stretch algorithm – 200 nodes with 6 anchors are randomly distributed in a 10*r*×10*r* square and the connectivity is 12.5. (a) The original network and the result of search process; (b) the result of stretch process; (c) the result of shrink process; (d) the result of localization after refinement and coordinate system transformation. In (d), circles are non-located nodes and points are anchors, besides, the lines are localization errors (Ranging-based, 5% ranging error)

candidates select the nodes to maximize h_{3s}. At last, select a node n_4 using rule 1.

5) Select the nodes to minimize $|h_{1s}$-$h_{2s}|$+$|h_{3s}$-$h_{4s}|$.Then, select a node n_5 as the centre node using rule 2.

For a network with square layout, the four fringe nodes will be the ones in the four respective directions as shown in Fig. 1(a).

Stretch process: Calculate the smallest-hop-count value h_5 from each node to n_5, and $h_{5\max}$ denotes the maximum h_5 value. Let (x_i, y_i) denotes node *i*'s estimated coordinate. Then fix n_5 at (0, 0), n_1 at $(\beta h_{5\max}R, 0)$, n_2 at $(-\beta h_{5\max}R, 0)$, n_3 at $(0, \beta h_{5\max}R)$, n_4 at $(0, -\beta h_{5\max}R)$, respectively, where β denotes stretch factor and is set to be 10 in this paper, and R is the maximum communication range of sensor node. Furthermore, the coefficient of elasticity of the spring between node *i* and *j* is set to be $h_{5i}h_{5j}$, where h_{5i} is the smallest-hop-count value between node *i* and n_5. Then relax the mass-spring mesh with non-fixed nodes starting at random positions or (0, 0) for all for simple. As Fig. 1(b) shows, when the mesh system runs to its lowest energy configuration, the mesh is stretched to be a huge one which is unfolded and the springs are elongated. Such a process is modeled as an energy function E_{stretch} given as

$$E_{\text{stretch}} = h_{5i}h_{5j}\sum_{i\in V',j\in N_i}[\sqrt{(x_i-x_j)^2-(y_i-y_j)^2}-d_{ij}]^2 \qquad (1)$$

where V'=V-$\{n_1, n_2, n_3, n_4, n_5\}$, N_i=$\{j|$ node *i*'s one-hop neighbor$\}$, d_{ij} is the estimated distance between node *i* and *j*.

Shrink process: Using the result of node positions estimated in stretch process as input, release the fixed nodes and other nodes, then the mesh will shrink because of the rest of springs, and the nodes will be pushed to move. Additionally, in this process, we find that the connections of 2-hop neighbors are helpful, so the 2-hop connections are added to the mesh as springs. This process is modeled as another energy function E_{shrink} given as

$$E_{shrink} = \omega_{ij}\sum_{i\in V,j\in N_i^{(2)}}[\sqrt{(x_i-x_j)^2-(y_i-y_j)^2}-d_{ij}]^2 \qquad (2)$$

where $N_i^{(2)}$=$\{j|$ node *i*'s neighbor within two hop$\}$; if *i* and *j* is one-hop away, ω_{ij}=1, d_{ij} is the estimated distance; if *i* and *j* is 2-hop away, ω_{ij}=1/4, d_{ij} is the shortest path. In order to gain the most similar topology of the mesh compared to the original one, when the mesh is relaxing the nodes are controlled to move in the order of their numerical size of h_5 (have been calculated in *stretch process*) from 0 to $h_{5\max}$ (for n_5, h_5=0). When the mesh system runs to the lowest energy configuration, that is formula (2) reach the minimum, each node is possible be pushed to a coarse position and the topology of the mesh is similar to the original one. Fig. 1(c) is the output of this process for a sample network. As showing, the stretch process run with shrink process is good at finding the similar topology of the network and the output of positions of the nodes can be used as initial input for mesh relaxation method.

Refinement process: Utilize the result of node positions estimated in the shrink process as initial input, and use just the one-hop connections between nodes as springs in the mesh, then relax the mesh again. The process of relaxing can be modeled as another energy function E_{refine} given as

$$E_{refine} = \sum_{i\in V,j\in N_i}[\sqrt{(x_i-x_j)^2-(y_i-y_j)^2}-d_{ij}]^2 \qquad (3)$$

When the mesh system once again runs to the lowest energy configuration, that is formula (3) reach the minimum, each node is possible to find its correct position in a relative coordinate system.

When the processes above have been carried through, Stretch algorithm completes the relative localization. Then, with the help of anchors, the relative coordinate system can be transformed to the global one, and the localization is finished. Fig. 1 shows the whole procedure of Stretch algorithm for a sample network.

Otherwise, noticed that formula (1), (2), and (3) have the similar form in math. They can be all considered as the sum of squares of differences of current distances in mesh relaxing and desirable ones for nodes. When the mesh is completely relaxed in a certain process, the corresponding energy function come to its minimum and the condition for this is

$$\frac{\partial E_{cer}}{\partial x_i} = \frac{\partial E_{cer}}{\partial y_i} = 0 \qquad i \in V_{cer}\,, \qquad (4)$$

where E_{cer} is a certain energy function and V_{cer} is the corresponding vertex set. There are many nonlinear optimization techniques to solve formula (1)~(4). Here we adopt a simple solution below: first, calculate the following gradients of each node *i* as

$$\Delta x_i = \frac{\partial E_{cer}}{\partial x_i} \quad \text{and} \quad \Delta y_i = \frac{\partial E_{cer}}{\partial y_i}, \quad i \in V_{cer}. \tag{5}$$

Second, move the nodes through iteration as

$$\begin{cases} x_i^{n+1} = x_i^n - \alpha \Delta x_i \\ y_i^{n+1} = y_i^n - \alpha \Delta y_i \end{cases} \quad i \in V_{cer}, \tag{6}$$

where α is the iteration step, and n is the iteration times. To approximate formula (4), a small threshold T is set, and the condition to stop the iteration is given as

$$\max_{i \in V_{cer}} \sqrt{\Delta x_i^2 + \Delta y_i^2} < T. \tag{7}$$

For formula (5), the gradients are easy to be calculated. For example, for formula (1), the gradients are

$$\begin{cases} \dfrac{\partial E_{stretch}}{\partial x_i} = \displaystyle\sum_{i \in V', j \in N_i} 2h_{5i}h_{5j}(x_i - x_j)\left(1 - \frac{d_{ij}}{\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}}\right) \\ \dfrac{\partial E_{stretch}}{\partial y_i} = \displaystyle\sum_{i \in V', j \in N_i} 2h_{5i}h_{5j}(y_i - y_j)\left(1 - \frac{d_{ij}}{\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}}\right) \end{cases}. \tag{8}$$

For a network containing n nodes and m anchors, the complexity for search process is $O(5n^2)$, and for stretch process is $O(k_1(n\text{-}5)^2)$, where k_1 is the average one-hop neighbor number, and for shrink process is $O(k_2n^2)$, where k_2 is the average neighbor number within two hop, and for refinement process is $O(k_1n^2)$, while for coordinate system transformation is $O(n+m^3)$.

IV. EXPEIMENTAL RESULTS

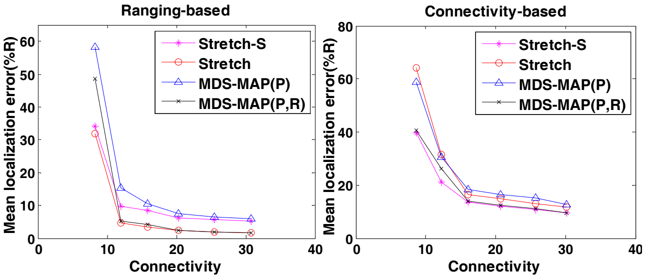
In this section, we show the localization effect of the proposed algorithm through simulations. The program was run in Visual C++ 6.0 on a PC with Intel Core 2.0 GHz processor and 1024MB RAM. In order to verify that our method is good at finding the similar topology of the network that can be used as the initial input for mesh relaxation as shown in Fig. 1, we additionally ran a *simplified version of Stretch algorithm* (*Stretch-S*) without refinement process. Furthermore, the two versions of the Stretch algorithm were compared with MDS-MAP(P) and MDS-MAP(P,R), which are also two versions of MDS-MAP(P). The four algorithms were simulated respectively based on ranging and connectivity under the same situation. If it is based on ranging, the distance measure is modeled as d×(1+N(0, e_r)), where d is the true distance and N(0, e_r) is the normal distribution with zero mean and ranging error e_r as standard deviation. The performance of the algorithms is measured through mean localization error (MLE), which is reported in percentage normalized by the maximum radio range (R) and is given as

$$\text{MLE} = \frac{\sum_{i \in V} \sqrt{(x_i - x_i^r)^2 + (y_i - y_i^r)^2}}{nR} \times 100\% \tag{9}$$

Where, (x_i^r, y_i^r) is the real position of node i. In the experiments, each data point is an average of 20 runs of simulations.

A. *Network Connectivity*

The simulation program was performed on the same type of network as the one shown in Fig. 1(a): 200 nodes with 6 anchors among them are randomly distributed in a 10r×10r square, where r is the placement unit length. The maximum radio range is from 1.25r to 2.5r, with an increment 0.25r, which resulted in average connectivity (average number of neighbors) levels 8.4, 12.1, 15.9, 20.5, 25.6, and 30.5, respectively. Fig. 2(a) shows the result of the four algorithms simulated based on ranging with 5% ranging error. For the connectivity level of 12.1 or above, Stretch algorithm have the MLE lower than 5%R, and the higher is the connectivity, the higher is the localization accuracy. As the result shows, Stretch algorithm is better than MDS-MAP(P,R), especially for the middle and low connectivity level, and they are nearly have the same result for high connectivity level. For example, at the connectivity level 12.1, Stretch has the MEL of 4.6%R compared with 5.28%R for MDS-MAP(P,R). On the other hand, Stretch-S is better than MDS-MAP(P) and has the MLE lower than 10%R when the connectivity level is higher than 12.1. As showing, stretch process running with shrink process, that is Stretch-S algorithm, is good at finding the initial positions of nodes for mesh relaxation and is of high probability to avoid folds and flips.



(a) (b)

Figure 2. Comparison of the Stretch, Stretch-S, MDS-MAP(P) and MDS-MAP(P,R) algorithm – 200 nodes with 6 anchors randomly distributed in a 10r×10r square. (a) Simulation result based on ranging with 5% ranging error. (b) Simulation result based on connectivity

Fig. 2(b) shows the result that the four algorithms simulated based on connectivity under the same types of networks as in Fig. 2(a). According to the result, Stretch-S is better than Stretch and MDS-MAP(P). At the connectivity level 12.1, Stretch-S has the MLE of 21%R compared with 26%R for MDS-MAP(P,R), and they are nearly have the same results at other connectivity levels. This is because based on connectivity, the shortest path is used as the estimated distance which is very coarse and for Stretch-S, 2-hop connections can supply more constraints than one-hop connections used in refinement process for Stretch. On the other hand, based on connectivity Stretch cannot compare with MDS-MAP(P,R), but is better than MDS-MAP(P) at high connectivity level. So it is better to apply Stretch-S than Stretch under connectivity-based condition, also without refinement, Stretch-S is simpler.

Additionally, the four algorithms have been run on grid networks based on ranging with 5% ranging error and connectivity respectively. In grid networks, the nodes are placed with 10%r error, which is a Gaussian noise with zero mean and 10%r standard deviation and is added to each coordinate of a grid point. The networks contain 100 nodes

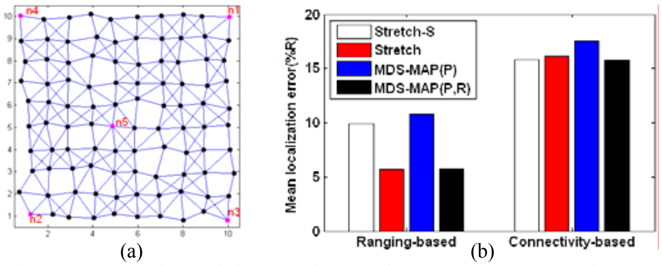


Figure 3. Comparison of the Stretch, Stretch-S, MDS-MAP(P), and MDS-MAP(P,R) – 100 nodes with 4 anchors are distributed on a gird in a 10*r*×10*r* square. The placement error is 10%r. (a) The gird layout of a sample network. (b) Simulation result respectively based on ranging with 5% ranging error and connectivity

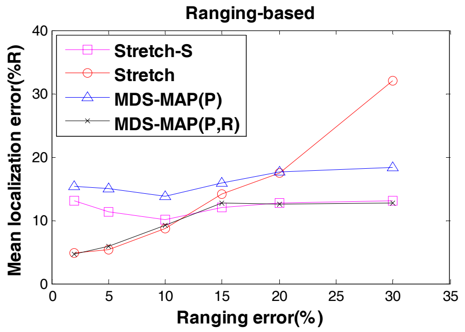


Figure 4. Performance comparison of different localization algorithms under varied ranging errors with 200 nodes in a network of area 10*r*×10*r* and 6 randomly-distributed anchors

with 4 anchors among them in a 10*r*×10*r* square. Fig. 3(a) shows a sample network. *R*=1.5*r*, which lead to average connectivity level of 5.2. Furthermore, we assume that the four fringe nodes have been configured manually as Fig. 3(a) show, while the centre node is still determined through search process. Fig. 3(b) shows the simulation results for the four algorithms. Based on ranging results, the Stretch has the MLE of 5.68%*R* compared with 5.76%*R* for MDS-(P, R), and Stretch-S has the MLE of 9.9%*R* compared with 10.8%*R* for MDS-(P). Based on connectivity, Stretch-S again has a better accuracy than Stretch, and is nearly comparable with MDS-MAP(P, R).

B. Ranging Error

To explore the impact of ranging errors on the localization, we performed simulations under different ranging-error conditions. In this experiment, there are 200 randomly distributed nodes in a 10*r*×10*r* square area, among which 6 anchors are included. Here, *R*=1.5*r*, so the average connectivity level is 12.3. The simulations were run at ranging error of 2%, 5%, 10%, 15%, 20%, and 30%, respectively. The results are shown in Fig. 4 for the four algorithms of interest. After comparing the performance of the four algorithms in Fig. 4, we know that, the algorithm Stretch is better than any onf of the other three algorithms if the ranging error is lower than about 13%, and if not, the performance of Stretch-S and MDS-MAP(P,R) is better, under our experiment conditions. The algorithms Stretch-S and MDS-MAP(P,R) nearly have the same positioning accuracy and behave steadily and excellently even if the ranging error is high.

V. Conclution

In this paper, we presented a centralized localization method for WSNs. Knowing the estimated distance between one-hop neighbors and a few anchors, it is possible with the proposed method to estimate the positions of nodes in the network. The proposed algorithm is based on mesh relaxation. As demonstrated before, for uniform network, the mesh of the network from four different directions stretches to become a huge one in the proposed method, and then, the mesh is made shrunk. The method is applicable to and good at finding the similar topology of the network, compared to the original one, and it is of high possibility to relax the mesh avoiding folds and flips. It is shown that our method can be applied as ranging-based one if distance measure between neighbors is available or connectivity-based one. Furthermore, our method has better localization accuracy than previous methods. The crucial step in our method is searching for the fringe nodes. In the experiments, it is found that for grid networks and irregular shape networks, the search process above may not find the four correct fringe nodes that should respectively locate in the four directions, which is not conductive to the following stretch process. However, this problem can be solved by configuring the fringe nodes manually.

References

[1] A. Chandrakasan, R. Amirtharajah, S. Cho, et al, “Design Considerations for Distributed Microsensor Systems,” in *Proc. IEEE Custom Integrated Circuits Conference (CICC)*, Piscataway, NJ, USA, May 1999, pp. 279-286.

[2] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, et al, “A Survey on Sensor Networks,” *IEEE Communications Magazine*, 40(8): 102-114, Aug. 2002.

[3] P. Bonnet, J. Gehrke, P. Seshadri, “Querying the Physical World,” *IEEE Personal communications*, 7(5):10-15, Oct. 2000.

[4] C. Y. Chong, S. P. Kumar, “Sensor Networks: Evolution, Opportunities, and Challenges,” *Proc. of IEEE*, 91(8): 1247- 1256, Aug. 2003.

[5] X. Liu, X. Bian, H. Cho, “A Novel Cluster-Chain Channel Adaptive Routing Protocol in Wireless Sensor Networks,” in *Proc. of the 5th International ICST Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness*, Hong Kong, Jul. 2008.

[6] N. Priyantha, H. Balakrishnan, E. Demaine, and S. Teller, “Anchor-free distributed localization in sensor networks,” MIT Lab. Comput. Sci., Cambridge, Tech. Rep. 892, Apr. 2003.

[7] N. Bulusu, J. Heidemann, and D. Estrin, “GPS-less low cost outdoor localization for very small devices,” *IEEE Pers. Commun.*, 7(5): 28–34, Oct. 2000.

[8] T. He, C. Huang, B. Lum, J. Stankovic, and T. Adelzaher, “Range-free localization schemes for large scale sensor networks,” in *Proc. ACM MobiCom*, San Diego, CA, Sep. 2003, pp. 81–95.

[9] V. Vivekanandan, V. W. S. Wong, “Concentric anchor beacon localization algorithm for wireless sensor networks”, *IEEE Trans. on Vehicular Communications*, 56(5): 2733 – 2744, Sept. 2007.

[10] Y. Shang, W. Ruml, Y. Zhang, and M. Fromherz, “Localization from mere connectivity,” in *Proc. ACM MobiHoc, Annapolis*, MD, Jun. 2003, pp. 201–212

[11] Y. Shang, W. Ruml, and Y. Zhang, “Improved MDS-based localization,” in *Proc. IEEE Infocom*, Hong Kong, Mar. 2004, pp. 2640–2651.

[12] S. Nawaz, S. Jha, “A Graph Drawing Approach to Sensor Network Localization,” *The Fourth IEEE International Conference on Mobile Ad-hoc and Sensor Systems*, Pisa Italy, Oct. 2007, pp. 1 – 12.

[13] A. Howard, M. J. Mataric, and G. S. Sukhatme, “Relaxation on a mesh: a formalism for generalized localization,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Wailea, Hawaii, Oct 2001, pp. 1055 – 1060