Robot assisted localization of sensor networks guided by rigidity

Saroja Kanchi
Department of Computer Science
Kettering University
Flint, MI 48504
skanchi@kettering.edu

Changhua Wu
Department of Computer Science
Kettering University
Flint, MI 48504
cwu@kettering.edu

ABSTRACT

In this paper, we propose a robot assisted localization guided by rigidity checking. For a network to be localizable, it must be rigid, which means given the existing edge constraints, there is only one location for each node in it. Many networks, however, are not rigid. To localize these nonrigid network, a robot can be used to add additional distance measure between nodes so that the network becomes rigid and localizable. To facilitate the localization and reduce the travel cost of the robot, the rigid subregions of the network are first identified. The identification of rigid subregions provides a guide for the movement of robots. A brief introduction to rigidity theory is presented in this paper. Experiments that evaluate the performance of the proposed approach are presented. The experimental results prove that the proposed approach is sound and gives excellent results compared to previous approaches for localization.

General Terms

localization, sensor networks, robotics

1. INTRODUCTION

The recent advancements in wireless communication and sensing technology have resulted in wide deployment of sensors in applications like environmental monitoring, search and rescue, military surveillance, and intelligent transportation, etc [1, 8, 12]. In these types of applications, the knowledge of the location of each sensors is important. Due to constraints of these application, however it is often difficult to preset the locations of sensors before they are deployed. Therefore, the capability of obtaining the positions of sensors after the deployment is fundamental to the success of the mission of sensor networks.

Most of the node localization algorithms are based on range measurements, through either time of arrival (TOA) [14], time difference of arrival (TDOA) [11], or received signal strength (RSS) [2, 9]. The problem of *localization* is to derive the geolocation of a node given a set of known locations and range measurements to these locations. The problem of *self-localization* is to derive geolocations of all nodes in a sensor network given range measurements between these nodes. Given the available range measures, there is only one position for the nodes in the network, then the network is localizable. Similarly, if a node has only one position that satisfies all the range measures relevant to it, it is localizable. There are four basic ways to obtain the location of sensors: 1) GPS or other global positioning techniques. If each sensor has GPS receiver and the sensors are deployed in some open space, then the locations of sensors can be easily known. With the appropriate equipment, GPS can provide a resolution of up to 5cm. However, the high cost of GPS receiver makes it impractical for each sensor to have one. 2) Beacon-based trilateration. If there are enough beacons, the localization can be done via trilateration in which a sensor can caliberate its own position if it can measure the distance to three nodes whose locations are known; This approach requires in-sight view of beacons or nodes with known positions. For some sparse network, this may not be true. 3) Mobile self-localization. When all or part of the sensors in a network are mobile, the sensors can move around to add additional distance measures [13, 5] so that the network becomes localizable. This approach requires a large number of mobile nodes. It can be used to localize every sparse network quickly since the mobile nodes can move in parallel, but the localization accuracy is subject to the odometry error of the mobile nodes. 4) Assisted localization [3, 10]. In this method, a robot travels in depth first search (DFS) pattern and adds additional distances between unlocalized nodes to localized ones so that unlocalized nodes become localized.

In this paper, we propose a rigidity-guided and robot assisted localization approach for sparse sensor networks using multiple dimensional scaling (MDS). In this method, the nodes of the sensor networks do not move. Intial distance measures between nodes is provided depending on what each node can see from its fixed location. In order to make the network localizable, additional distance measures may be needed. In our proposed method, a robot is used to travel in the network to add additional distance measures to turn the network into a localizable one. The difference between the proposed approach and the approach in [10] is that the travel of the robot is guided by rigidity checking and the localization is done via multiple dimensional scaling. By checking the rigidity of the network, the rigid subregion of the network can be identified, therefore the robot does not

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
WICON 2008 Maui,Hawaii, USA, Nov 17- Nov 19
Copyright 2008 ACM ICST 978-963-9799-36-3 ...$5.00.

Digital Object Identifier: 10.4108/ICST.WICON2008.4817
http://dx.doi.org/10.4108/ICST.WICON2008.4817

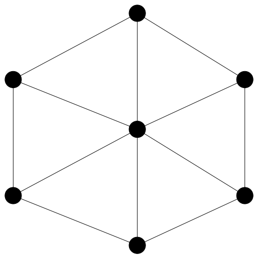


Figure 1: This network is rigid but cannot localized by trilateration if there is only three beacons. However, it can easily be localized by MDS.

have to move to most of the nodes that are inside rigid subregions. Moreover, the proposed method has less stringent requirement on the positions of beacons than the method in [10] due to use of multiple dimensional scaling, see figure 1.

This paper is organized as follows. An introduction to the rigidity theory is presented in Section 2. Section 3 discusses the proposed approach in details and the evaluation and experiments are presented in Section 4. Conclusion and future work are given in Section 5.

2. NETWORK LOCALIZABILITY AND RIGIDITY

In this section, we are going to introduce the theory in network localizability and rigidity. A detailed description can be found in [6, 3, 4].

Let a framework $p(G)$ be a graph $G = (V, E)$ along with a mapping $p : V \rightarrow \mathcal{R}^2$ which assigns each vertex to a point in the plane. A *finite flexing* of a framework $p(G)$ is a family of realizations of G, parameterized by t so that the location, r_i , of each vertex i, is a differentiable function of t and $|r_i(t) - r_j(t)|^2$ is constant for every $(i, j) \in E$. Thinking of t as time, and differentiating the edge length constraints, we have

$$(u_i - u_j)(r_i - r_j) = 0 \text{ for every } (i, j) \in E \qquad (1)$$

An assignment of velocities that satisfies Eq. 1 for a particular framework is an infinitesimal motion of that framework. Every framework has three trivial infinitesimal motion: two translations, and a rotation. If a framework has a nontrivial infinitesimal motion it is infinitesimally flexible. Otherwise it is infinitesimally rigid. The checking for whether a particular framework is rigid or not can be determined from the property of the graph.

Let $G = \{V, E\}$ denote a network of vertices $V = \{1, 2, ..., n\}$ and for any edge $(i, j) \in E$, the distance between V_i and V_j is precisely known. The network localization problem is to determine the unique position of each node in the network given the positions of available beacons and the distance between each pair $(i, j) \in E$. If under the given constraints, there is only one position for each node, then the network is localizable. The network localization problem is closely related to the Euclidean graph realization problem, in which coordinates are assigned to vertices of a weighted graph such that the distance between coordinates assigned to nodes joined by an edge is equal to the weight of the edge.

For a two dimensional graph with n vertices, the positions of its vertices have $2n$ degrees of freedom, of which three are the rigid body motions. Therefore graph is rigid if there are $2n - 3$ constraints. If each edge adds an independent constraint, then $2n-3$ edges should be required to eliminate all nonrigid motions of the graph. Clearly, if any induced subgraph with n vertices has more than $2n-3$ edges then these edges cannot be independent, which leads to the following Laman theorem [7]:

THEOREM 1. *The edges of a graph $G = \{V, E\}$ are independent in two dimensions if and only if no subgraph $G' = \{V', E'\}$ has more than $2n'-3$ edges, where n' is the number of nodes in G'.*

Corollary 1. A graph with $2n-3$ edges is generically rigid in two dimensions if and only if no subgraph G' has more than $2n' - 3$ edges.

Laman's theorem characterizes generic rigidity. However, a direct implementation of it leads to a poor exponential algorithm. An efficient approach to check for rigidity is proposed in [6] based on a pebble game. Jacob *et. al* proposed Jacob's approach uses the following formulation of Laman algorithm:

THEOREM 2. *[6] For a graph $G = \{V, E\}$ having m edges and n vertices, the following are equivalent.*

- *The edges of G are independent in two dimensions.*
- *For each edge (a, b) in G, the graph formed by adding three additional edges identical to (a, b) has no induced subgraph G' in which $m' > 2n'$.*

The basic idea behind Jacob's algorithm is to grow a maximal set S of independent edges one at a time. Initially, S is empty. Let's denote these basis edges by E.

A new edge is added to S if it is discovered to be independent of the edges existing in S. To check whether an edge e is independent of edges in S, each vertex is assigned two pebbles initially and a temporary set S' is created. S' contains all the edges in S plus four copies of e. The pebbles can only travel via the edges in S'. If all edges in S' can be covered by the pebbles, then we know that e is independent of all edges in S and e is added into S. This process is repeated until no more edges can be added into S. Then S is a maximal set of independent edges. If S contains $2n - 3$ edges, then the graph is generically rigid.

Having $2n-3$ independent edges ensures the generic rigidity of a graph. However, it does not guarantee the unique realization of the network. A discontinuous change to the positions of nodes may lead to another realization which satisfies all the constraints of the network, as shown in Figure 2. The following theorem states the condition for a network to be uniquely realizable.

THEOREM 3. *[4] A graph G with $n \geq 4$ vertices is uniquely realizable in two dimensions if and only if it is redundantly rigid and tri-connected.*

Redundant rigidity means after removing any single edge, the remaining graph is still generically rigid. A tri-connected graph is a connected graph such that deleting any two vertices (and incident edges) results in a graph that is still connected. When a network satisfies the condition in Theorem 3, it can be uniquely localized given at least three nonlinear beacons in a two dimensional space.

Digital Object Identifier: 10.4108/ICST.WICON2008.4817
http://dx.doi.org/10.4108/ICST.WICON2008.4817

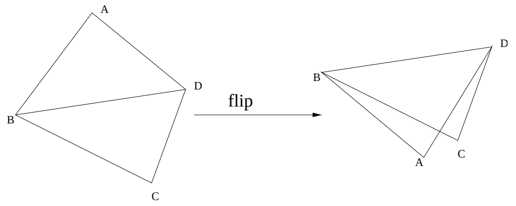


Figure 2: A generically rigid graph subject to flapping transformation. The two realizations are not continuous in two dimension space in that the second one is obtained by a flapping of the first one.

3. THE RIGIDITY-GUIDED LOCALIZATION

As we have discussed above, a network has to be rigid to be localizable. For those networks that are not rigid, we can turn it into a rigid one if we have a mobile robot that can measure the distances between itself and the nodes in the network up to a certain range. In this section, we are going to discuss in detail the proposed approach.

Without lack of generality, we assume that the network is connected. The proposed approach contains two major steps. In the first step, the robot collects the information about the network such as the nodes and the edges in it. With these information, the robot can then identify the rigid subregions of the network. In the second step, the robot travels in the network from one rigid region to the next in order to add additional distance measures so that in the end the whole network becomes one rigid region and is thus localizable. The details of the each step are discussed in the following two subsections.

3.1 Identification of rigid subregions

The reason for searching the rigid subregions is to reduce the travel of the robot in the later phase of the localization. If a subregion is already rigid, the robot does not have to deal with most of the nodes inside the region. To find the rigid subregions, the robot will check the 1-hop neighborhood of every node in the network and put those 1-hop neighborhoods that are tri-connected and redundantly rigid into an initial list of rigid subregions. Recall that a graph is triconnected if there are three paths between every two nodes of the network. Then the robot will try to merge the initial rigid regions in the list into bigger regions. The condition for two rigid subregion to be mergeable is that they must share at least three common nodes. This is implemented as follows: Two regions are merged if they
Have 3 vertices in common
Two vertices in common and one other vertex in one of the areas that has 3 edges to the other area
One vertex in common and two other vertices in one of the areas that have 3 edges each to the other area
No vertices in common but there are 3 vertices in one area that have 3 edges each to the other area
This process is iteratively repeated until there is no more regions to be mergeable. In this way, the maximal rigid subregions of the network can be identified.

3.2 Addition of distance measures

In [10], a method is proposed to measure the distance between two nodes based on distance measures between the robot and the nodes. Let n_1, n_2 be two nodes between which we want to measure the distance. If the robot moves straight in a plane that contains both n_1 and n_2, then the distance between n_1 and n_2 can be computed, see figure 3, according to the following proposition [10]:

Proposition 1. The geometry of five co-planar points n_0, n_1, m_0, m_1, m_2, where m_0, m_1, m_2 are collinear, is determined by the distances $||n_i - m_i||$ for $i = 0, 1$ and $j = 0, 1, 2$.

This proposition is very easy to prove. The coordinates of the five points have 10 degrees of freedom, which include 2 degrees of freedom for translation and 1 degree of freedom for rotation. Therefore, the actual degrees of freedom for the geometry of the five points are 7. Given the six distance measures plus the constraint that m_1, m_2, and m_3 are collinear, the geometry of the five points can be uniquely determined.

For some very sparse network, there might be no rigid subregions at all. In this case, we can still obtain a small rigid subregion. For example, let n_1, n_2, n_3 are three nodes in the visible neighborhood of the robot. The robot can move in straight lines and compute the distance between every pair of the three nodes, thus getting a triangle which is always rigid. After identifying the rigid subregions, each node will be assigned a label by the robot, indicating the ID of the rigid subregions it belongs to. For those nodes that are not in any rigid subregion, their label is -1. The mission of the robot is then to travel in the network and compute additional distances between unconnected nodes, so that when it stops movement, all nodes in the network have the same label. We assume that robot can see and measure distance to nodes up to 3 hops away. For a node n, if the robot can add distance measures from n to three nodes in a rigid subregion, then n can be merged into the subregion and be assign a label equal to the ID of the subregion it merges to. For two rigid subregions R_1 and r_2, if three nodes in R_1 are assigned label equal to the ID of r_2, or vice verse, region R_1 and r_2 can then be merged into a single rigid subregion and all nodes in R_1, r_2 will be assigned a new label. Starting from a rigid subregion, the mobile robot moves as follows:

1. Start from a rigid subregion
2. Move around its visible area, looking for a node that is labeled differently
3. For each such node, compute the distance from it to three nodes in the current rigid subregion using Proposition 1.
4. Change the label of the node and look for the next node labeled differently.

The Robot contains the areaList (list of vertices in each rigid subregion) and the adjList (adjacency list of the graph) of the graph. The Robot has vision that is at 3 hop distance in the graph. What it means is that assuming that maximum length of an edge is 5 units then the Robot can see nodes that are within radius of 15 units.

First the robot adds the area containing the starting vertex to the localized graph by figuring the coordinates of itself. Then it travels to a next adjacent vertex. If the next

Digital Object Identifier: 10.4108/ICST.WICON2008.4817
http://dx.doi.org/10.4108/ICST.WICON2008.4817

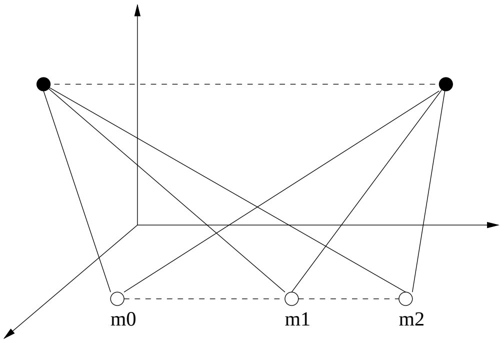


Figure 3: Computing distance by measuring the distances from three points on a straight line.

adjacent vertex has all its edges in the same region, then it travels to another vertex, until it reaches a vertex that has an edge to another region. It then tries to annex the new region by ensuring that there are 3 vertices in that new region that have 3 edges each to the existing localized graph. It may need to add new edges that are in 3 hop distance.

The robotic movement in detail is

1. Robot is at v_1 of the localized graph. Assume that v_5 and v_6 in the localized graph are adjacent to v_1

2. Robot moves v_2, using edge (v_1,v_2). Assume that v_2 is in a new region R_1 Edges to be added while at v_2 are (v_5, v_2) and (v_6,v_2) if they do not exist. Now v_2 has 3 edges to the localized graph, so v_2 is localized

3. Robot moves to v_3 using the edge (v_2,v_3) and assume that v_3 is in R_1 Edges to be added while at v_3 include (v_5,v_3), and (v_6,v_3), if they do not exist. Now v_3 has three edges to the localized graph so it is localized

4. Robot moves to v_4 using the edge (v_3,v_4). Assume that v_4 is in R_1 Edges to be added while at v_4 are (v_4, v_2), (v_4, v_1) if it does not exist. Now v_4 has three edges to the localized graph, so it is also localized

5. There are 3 vertices v_2, v_3, v_4 in R_1 that have 3 edges each to a localized part of the graph, so the entire region R_1 can now be localized. The rest of the vertices in region R_1 need not traversed by the robot. The Robot now continues to travel to a vertex in a region that is not localized yet.

As you can see, the robot moves in a way similar to depth-first search. The above moving strategy will guarantee that the network will eventually become a single rigid region. The robot also minimizes the traversal and the number of edges added.

4. EVALUATION

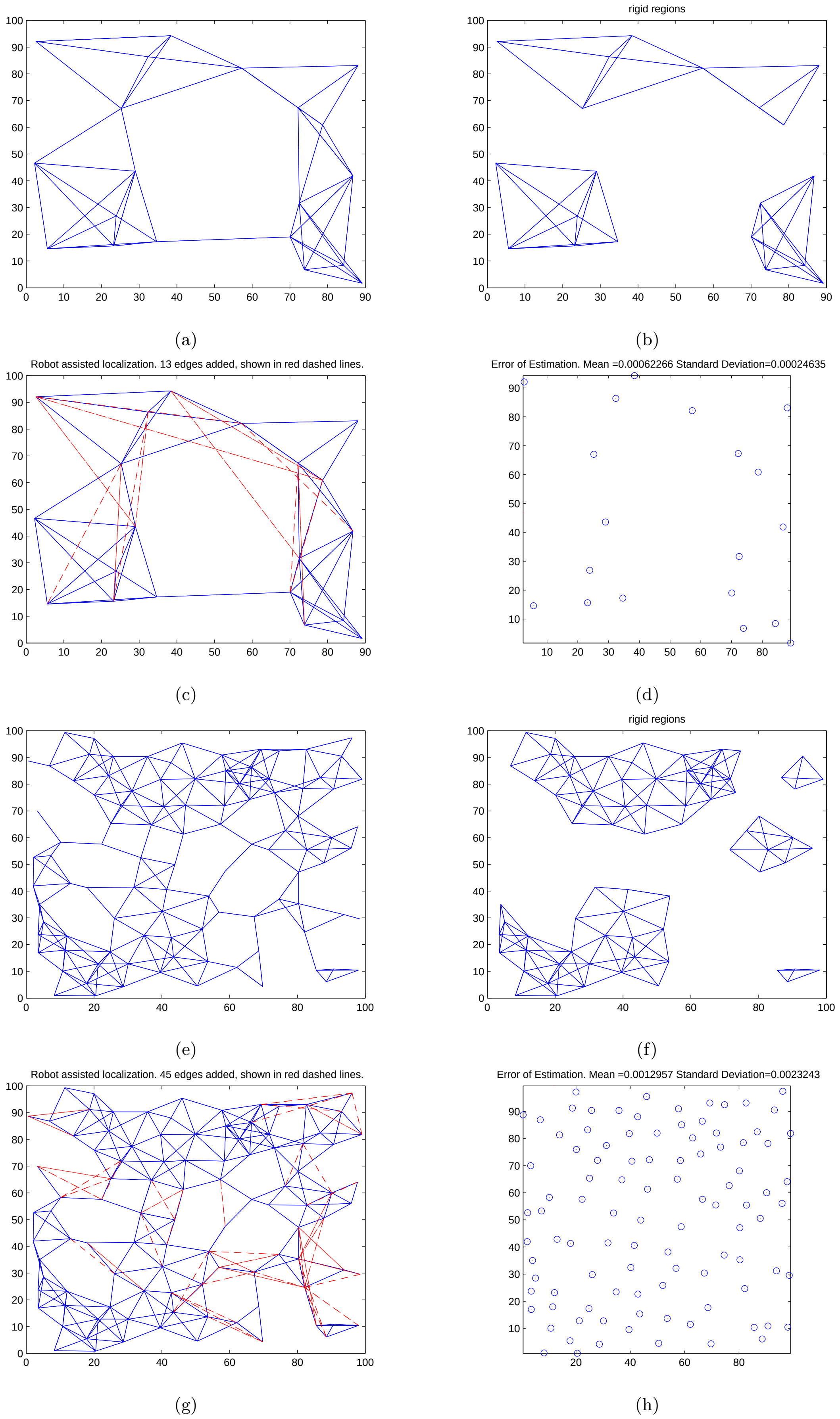
In this section, we are going to evaluate the proposed approach in localizing various networks. We would like to study the proposed approach from the following aspects: the number of edges added by the robot in localizing networks of various degrees, the reliability and accuracy of the localization, and the effects of noise in the distance measures obtained by the robots on the localization accuracy.

Figure 4 shows the results of the localization on two networks. The top four figures show the test on the first network and the bottom four figures show the test on the second network. In each test case, the four figures from left to right and top to bottom are the original network, the rigid subregions, the original network with added edges shown as red dashed lines, and the localization result respectively. As we can see, there are five rigid subregions in both networks. By detecting the rigid subregions, the robot only have to add enough distances between nodes of different labels in order to merge the subregions. Therefore, the travel cost of the robot can be reduced. Figure 4-(d) and figure 4-(h) show the localization results, in which a circle is drawn around the estimated positions to make it more observable. The localization error is indicated by a line segment starting from the estimated position to the known position. The longer the line segment, the larger the localization error is. As we can see, on both cases, the localization error is very small. There are however, some cases that big localization errors can happen due to the fact that in the MDS localization algorithm, the distance between any pair of nodes is approximated by the shortest path between them. In some irregular networks, this approximation causes problems. This problem can be solved by adding more edges to the networks or use a local patch-based MDS method. For a detailed discussion, please refer to our previous work [13].

Figure 5 shows the result of evaluating the relationship between the number of edges added by the robot and the average degree of nodes in a network. The evaluation is done on four cases with different number of nodes. By changing the communication range of nodes, the average degrees of nodes can be changed. As we can see from this figure, the number of edges, shown in percentage of the original edges in the network, decreases sharply when the average degree increases, which matches our expectation. When the average degree is less than 4, the number of edges to be added is more than 50% of the number of the existing edges. When the average degree increases to 6, only about 20% edges have to be added by the robot. When the average degree increases to 8, less than 10% new edges have to be added.

We also carried out an experiment in evaluating how the error in the distance measure done by the robot affects the accuracy of the localization. The evaluation is done on randomly generated networks of 20 nodes but different average degrees. The error in distance measure by robot is modeled as an multiplicative Gaussion noise with zero mean. The standard deviation of the error varies from 0 to 0.15. Figure 6 shows the results. The x-axis shows the deviation of the noise and the y-axis shows the mean localization error. The curves show the mean localization error and the vertical bar shows the deviation. As we can see, the higher the average degree, the smaller effect the error has on the localization. The communication range of the networks in the four figures are 23, 29, 35, and 41. The field size is 100*100. When the average degree is less than 6, we see large localization error which can be explained by the inaccuracy in the approximating the distance between nodes by shortest path. When the average degree increases up to 7, the localization error is

Digital Object Identifier: 10.4108/ICST.WICON2008.4817
http://dx.doi.org/10.4108/ICST.WICON2008.4817



(a)

(b)

(c)

(d)

(e)

(f)

(g)

(h)

Figure 4: Two cases of robot assisted localization. The first four figures show a network of 20 nodes. The last four figures show a network of 100 nodes. For each case, the original network, the rigid subregions, the edges added by the robot, and the error of localization are shown in left to right and top to bottom order.

Digital Object Identifier: 10.4108/ICST.WICON2008.4817
http://dx.doi.org/10.4108/ICST.WICON2008.4817

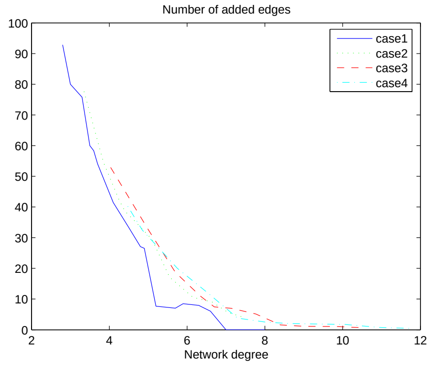


Figure 5: The relation between the number of added edges and the order of the network. The vertical axis shows the number of new edges in percentage of the existing edges in the network.

about 10% of the communication range. When the error in distance measure increases, the localization error also slowly increases until the distance error becomes larger than 12% which triggers a large increase in the localization error.

5. CONCLUSION

Localization is an important problem in the application of sensor networks. In this paper, we proposed a localization approach for sensor networks using a mobile robot. The robot travels in the network and add additional distances between nodes so that the whole network becomes localizable. Based on rigidity theory, the rigid subregions of the network can be identified. The rigid subregions provide a guide for the robot in that the robot does not have to deal with a large portion of the nodes in the rigid subregions. Simulation experiments show the performance of the proposed approach. The future work will be the coordination of multiple robots in localization, the study on the reliability of MDS for localization, and the field test of the proposed approach.

6. REFERENCES

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

[2] N. Bulusu, J. Heidemann, and D. Estrin. GPS-less low cost outdoor localization for very small devices. Technical Report 00-729, Computer science department,University of Southern California, Los Angles, CA, 2000.

[3] T. Eren, D. Goldenberg, W. Whitley, Y. Yang, A. Morse, B. Anderson, and P. Belheumer. Rigidity, computation, and randomization of network localization, 2004.

[4] D. Goldenberg, A. Krishnamurthy, W. Maness, R. Yang, A. Young, and A. Savvides. Network localization in partially localizable networks. In *Proceedings of INFOCOM 2005*.

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

[6] D. J. Jacobs and B. Hendrickson. An algorithm for two-dimensional rigidity percolation: the pebble game. *Journal of Computation Physics*, 137:346–365, 1997.

[7] G. Laman. On graphs and rigidity of plane skeletal structures. *Journal of Engineering Mathematics*, pages 331–340, 4 1970.

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

[9] X. Nguyen, M. Jordan, and B. Sinopli. A kernel-based learning approach to ad hoc sensor network localization. *ACM Transactions on Sensor Networks*, 1(1):134–152, 2005.

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

[11] C. Savarese, J. Rabaey, and J. Beutel. Locationing in distributed ad hoc wireless sensor networks. In *Proc. 2001 Int'l Conf. Acoustics, Speech, and Signal Processing (ICASSP 2001)*, volume 4, pages 2037–2040, May 2001.

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

[13] C. Wu, W. Sheng, and Y. Zhang. Mobile self-localization using multi-dimensional scaling in robotic sensor networks. *The International Journal of Intelligent Control and Systems*, 11(3), September 2006.

[14] F. Zhao and L. Guibas. *Wireless Sensor Networks: An Information Processing Approach*. Elsevier and Morgan Kaufmann Publishers, 2004.

Digital Object Identifier: 10.4108/ICST.WICON2008.4817
http://dx.doi.org/10.4108/ICST.WICON2008.4817

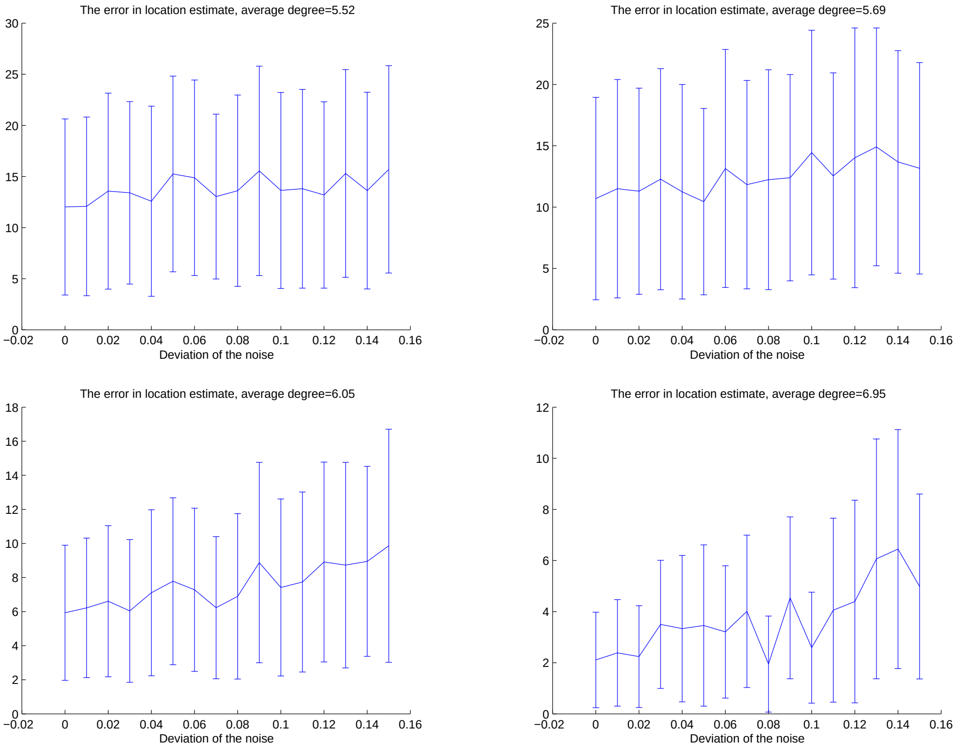


Figure 6: Effect of robot introduced error on localization

Digital Object Identifier: 10.4108/ICST.WICON2008.4817
http://dx.doi.org/10.4108/ICST.WICON2008.4817