Applied Mathematics & Information Sciences
An International Journal

Distributed Network Localization for Wireless Sensor Networks

Seong-Woo Kim , Jong Min Lee and Jung-Hwa Lee
[1] Department of Computer Software Engineering, Dong-Eui University, Busan, Republic of Korea
Email: libero@deu.ac.kr

Received 30 Jun. 2011; Revised 15 Nov. 2011; Accepted 04 Feb. 2012

Abstract: Recent advancements in wireless communication and microelectro-mechanical systems (MEMS) have made possible the deployment of wireless sensor networks for many real world applications. One of most challenging problems with the deployed sensor nodes is to identify their geographic locations given estimates of the distances between them. There have been a large number of localization algorithms, each of which makes a different geometric approximation. Among them, Multidimensional scaling (MDS) based algorithms outperform the others and have the advantage that they are robust for noise and sparse networks, with or without anchor nodes. Its distributed versions compute a local map for each node at first and then merge these maps to a global map. Additional refinement technique can improve the relative maps by forcing them to conform more closely to the distances to nearby neighbors. In this paper, we reformulate the network localization problem as a constrained least square problem mathematically in detail, and then develop a new refinement algorithm which is based on two hop distance constraints and Levenberg-Marquardt optimization technique. Our simulation results demonstrate that the proposed algorithm has good performance in terms of both success rate and node position estimation.

Keywords: Network Localization, Multidimensional Scaling, Levenberg-Marquardt Algorithm

1 Introduction

Wireless sensor networks (WSNs) consist of hundreds of wirelessly connected sensor nodes that collect sensor data and monitor an area of interest. For its prevalent applications like environmental monitoring or target tracking, the ability of such sensor node to determine its geographical location is of fundamental importance. Although global positioning systems (GPS) are getting popular and more accessible, they do not too cheap to embed on all of the nodes.

Recently, various localization systems have been developed for ad hoc wireless sensor networks. Most of the node localization algorithms are based on estimation of distances between neighbor nodes. The estimated can be measured through either received signal strength indicator (RSSI), time of arrival (TOA), time difference of arrival (TDOA), or Angle of Arrival (AOA)[1]. The problem of position computation is to compute a node's position based on the available information of distances and known locations of the reference nodes. Trilateration method is the most intuitive method to compute location by the intersection of three or more circles. Cricket system uses active beacons and passive ultrasonic receivers to measure distance between sensor nodes and beacons[2]. Then, a mobile node can compute its position through multilateration method for an indoor environment.

On the other hand, another important topic is the network localization algorithm, whose objective is to derive the geolocations of all nodes in a sensor network from a set of known locations and range measurements between sensor nodes. Niculescu and Nath proposed APS (Ad hoc Positioning System) that enable sensor nodes to estimate its distance to the beacon nodes in a multihop way[3]. In this system, DV-Distance method propagates distances between a beacon node and a target node hop by hop, instead DV-Hop method propagates the number of hops. Other works focused on the centralized approaches which collect all information of distances among the nodes into a central node and then solve its resultant mathematical optimization problem. Doherty et al. formulated the localization problem as convex optimization problem with connectivity constraints and used semidefinite programming (SDP)[4]. Shang et al. used classical multidimensional scaling (MDS) to derive positions of the nodes given the distance measurements between them[5]. They also developed the distributed version which is called as MDS-MAP(P)[6]. MDS is good at finding the right topology of the network, but not the precise locations of nodes, because MDS uses shortest path distances to approximate the distance between nodes more than 1 hop away and the approximation may not be accurate[5-7].

In this paper we first describe the concept of MDS and its related network localization algorithms in detail. Then we develop a refinement algorithm, MDS-MAP(P, O) which is based on MDS and Levenberg-Marquardt algorithm with barrier constraints. This method refines the relative local maps with avoiding the inaccurate information of shortest path distance and the infeasible solution beyond geometric constraints.

The rest of this paper is organized as follows: Section 2 introduces MDS based localization algorithms, in particular, MDS-MAP and MDS-MAP(P). Then we present a new refined algorithm, MDS-MAP(P, O) for the distributed network localization problem. Section 3 provides detailed comparison of the performance among MDS based localization algorithms for sensor networks. Section 4 concludes this paper.

2 MDS Based Localization Algorithms

2.1 Basic concept

Multidimensional scaling (MDS) is a set of data analysis techniques that display the structure of distance-like data as a geometrical picture. The distance between every pair of objects measures their dissimilarities. MDS starts with matrices representing distances or similarities between objects and find a placement of points in a low-dimensional space, usually two- or three-dimensional. By describing objects as points in a low-dimensional space, their essential information can be preserved while reducing the complexity of original data. Then, MDS is closed related with Principal Component Analysis (PCA) and clustering analysis.

MDS techniques can be classified as to whether the similarities data are qualitative (nonmetric MDS) or quantitative (metric MDS). The number of similarity matrices and the nature of the MDS model can also classify MDS techniques. This classification yields classical MDS (one matrix, unweighted model), replicated MDS (several matrices, unweighted model), and weighted MDS (several matrices, weighted model). Due to the page limit, we will only briefly introduce the classical MDS technique on which the localization is based on.[7]

In classical MDS, the dissimilarities are usually the Euclidean distances between pairs of objects. Let p_{ij} denote the dissimilarity between object i and j, d_{ij} denote the Euclidean distance, and $\{x_i\}_{i=1}^{N}$ denote the coordinates of objects. $\{x_i\}_{i=1}^{N}$ is to be recovered from the dissimilarities $\{p_{ij}\}_{i,j=1}^{N}$ or $\{d_{ij}\}_{i,j=1}^{N}$. Then we have

$$d_{ij}^2 = p_{ij}^2 = \|x_i - x_j\|^2 = (x_i - x_j)^T(x_i - x_j) \qquad (1)$$

The above equation can be rewritten as follows.

$$d_{ij}^2 = x_i^T x_i + x_j^T x_j - 2x_i^T x_j \qquad (2)$$

If we define $X = [x_1, \cdots, x_N]$ as a row vector, then the squared distance matrix, $D = [d_{ij}^2]_{i,j=1}^{N}$ can be expressed as

$$D = \delta(X^T X)e^T + e\delta(X^T X)^T - 2X^T X \qquad (3)$$

where e is the all ones vector, $\delta(\cdot)$ is the diagonal matrix operator, representing $\delta(X^T X) = [x_1^T x_1, \cdots, x_N^T x_N]^T$.

Next, the geometric center of objects in X can be obtained by the following equation.

$$gc(X) = \frac{1}{N} Xe \qquad (4)$$

Then, we can shift the objects amount up to the geometric center, represented as the follows.

$$\begin{aligned} X - gc(X)e^T &= X - \frac{1}{N} Xee^T = XH \\ &= [x_1 - \frac{1}{N}\sum_{i=1}^{N} x_i, \ldots, x_N - \frac{1}{N}\sum_{i=1}^{N} x_i] \end{aligned} \qquad (5)$$

where $H = I - \frac{1}{N} ee^T$ is a geometric centering operator.

Now, if multiplying both sides of D by the centering operator H, we will have

$$HDH = H\delta(X^T X)e^T H + He\delta(X^T X)^T H - 2HX^T XH \qquad (6)$$

Since $H\delta(X^T X)e^T H = He\delta(X^T X)^T H = 0$, we have

$$HXX^TH = -\frac{1}{2}HDH$$
$$= -\frac{1}{2}(D - \frac{1}{N}ee^TD - \frac{1}{N}Dee^T - \frac{1}{N^2}ee^TDee^T) \quad (7)$$

Given $B = HXX^TH$, we can compute matrix X up to linear translation and orthogonal transformation through singular value decomposition (SVD) of B.

$$B = V\Lambda V \quad (8)$$

The coordinate matrix becomes

$$X = V\Lambda^{1/2} \quad (9)$$

Thus, retaining the first n (2 for 2-D or 3 for 3-D) largest eigenvalues and eigenvectors leads to a solution in lower dimension.

2.2 Two basic methods

MDS-MAP is classical method to obtain the coordinates of nodes given an approximation of the Euclidean distances between them. The algorithm has the following steps[6].

1) Compute shortest paths between all pairs of nodes in the region of consideration. The shortest path distances are used to construct the distance matrix for MDS.

2) Apply classical MDS to the distance matrix, retaining the first 2 (or 3) largest eigenvalues and eigenvectors to construct a 2-D (or 3-D) relative map.

3) Given sufficient anchor nodes (3 or more for 2-D networks, 4 or more for 3-D networks), the coordinates of the anchors in the relative map are mapped to their absolute coordinates through a linear transformation. The best linear transformation between the absolute positions of the anchors and their positions in the relative map is computed.

While MDS-MAP is a central shortest path approximation, in the new MDS-MAP(P) each node constructs a local map using its local information and then the local maps are merged to form a global map. The procedure of MDS-MAP(P) is as follows[6]:

1) Set the range for local maps, R. For each node, neighbors within R hops are involved in building its local map. The value of R affects the amount of computation in building the local maps, as well as the quality.

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

Compute shortest paths between all pairs of nodes in its local mapping range R to construct the distance matrix for MDS. Then apply the classical MDS to the distance matrix and retain the first 2 (or 3) largest eigenvalues and eigenvectors to construct a 2-D (or 3-D) local map. The local map can be refined optionally. With use of the node coordinates in the MDS solution as the initial point, a least squares minimization is performed to make the distances between nearby nodes match the measured ones.

3) Merge local maps. Local maps can be merged sequentially or in parallel. There are various ways of merging local maps sequentially, such as randomly or as to certain order best for an application. Then, the global map also can be refined by the optimization techniques.

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

Since the position estimates by MDS are produced based on shortest path distances to approximate the distance between nodes more than 1 hop away, the resulting topology and the approximation may not be accurate. The position estimation accuracy may be enhanced by the optimization technique, which will be described in detail at the following section.

2.3 Constrained Optimization Problem

As mentioned previously, the refinement process can be formulated as a constrained optimization problem. Its exact objective function can measure not only distances between one-hop neighbors, but also distances between some multi-hop neighbors. A refinement range R can be defined in terms of hops to specify what information is considered. For example, R = 1 means only distances between immediate neighbors are considered. On the other hand, R = 2 means distances to all nodes within two hops are considered. Different values of R offer trade-offs between computational cost and solution quality. We consider the case of R = 2.

Let $x_i = (\hat{x}_i, \hat{y}_i)$ for $i = 1, \cdots, N$ represent the 2-D coordinates of the N nodes in a local map. When distance measures between 1-hop neighbors are available, p_{ij}, defined as the proximity between nodes i and j, is the distance measure if they are 1-hop neighbors, or the shortest path distance if i and j are more than 1 hop away.

Considering the noisy environment on measuring distance within the radio range, the refinement problem can be formulated as the following least square problem.

$$\min_x F(x) = \frac{1}{2}\sum_{i,j,i\neq j}^{M}\{w_{ij}(d_{ij}-p_{ij})\}^2$$

for i,j=1,…,N (10)

where w_{ij} is the weight. If $w_{ij}=0$ for all i and j that are more than 2-hop away, then only the 2-hop connectivity or distance measures are used. The refinement improves the map by giving local information between neighbor nodes more weight than that between far away nodes, which may be less accurate.

Now consider the problem objective as follows.

$$F(x) = \frac{1}{2} f(x)^T f(x) = \frac{1}{2}\sum_{i,j=1,i\neq j} f_{ij}^2(x) \quad (11)$$

where $f_{ij}(x) = w_{ij}(d_{ij}-p_{ij})$.

The gradient and Hessian of the objective are as follows.

$$\begin{aligned}\nabla F(x) &= J(x)^T f(x)\\ \nabla^2 F(x) &= J(x)^T J(x) + \sum_{i,j=1,i\neq j}^{M} f_{ij}(x)\nabla^2 f(x)\end{aligned} \quad (12)$$

where $J(x)$ is the Jacobian and $\nabla^2 F(x)$ is the Hessian of $F(x)$. The Jacobian is a $M\times 2N$ matrix containing the first partial derivatives of the function components.

$$J(x) = \left[\frac{\partial f_{ij}}{\partial x_l}\right] = \left[J_{ij,l}\right]_{ij,l}$$ for ij=1,…,M, and l=1,…,2N (13)

For the index ij consisting of the 1 or 2 hop neighbors in the Jacobian, each term of that row is as follows.

$$J_{ij,l}(x) = \begin{cases} w_{ij}\dfrac{\hat{x}_i-\hat{x}_j}{d_{ij}} & for \quad l=i\\ w_{ij}\dfrac{\hat{x}_j-\hat{x}_i}{d_{ij}} & for \quad l=j\\ w_{ij}\dfrac{\hat{y}_i-\hat{y}_j}{d_{ij}} & for \quad l=2*i\\ w_{ij}\dfrac{\hat{y}_j-\hat{y}_i}{d_{ij}} & for \quad l=2*j\\ 0 & for \quad others\end{cases} \quad (14)$$

Now let us consider the difference between distance measure and shortest path length. The distance measure between 1-hop neighbors is same with the shortest path. However, in case of 2-hop neighbors that use the shortest path length instead of effective distance, there is considerable difference between the distance d and the shortest path length p. Thus, the following constraints must be satisfied from their geometric relationship.

$$\begin{aligned} d_{ij} &< p_{ij} + \alpha_+\\ \max(p_{iNi}, p_{jNj}) - \alpha_- &< d_{ij}\end{aligned} \quad (15)$$

where N_i and N_j are one-hop neighbors of node i and j, α_+, α_- are positive constants. Figure 1 is depicted as an example of geometric relationship between neighbor nodes. It implies that the effective distance between two hop neighbors is shorter than the shortest path length and is longer than maximum distance between one hop neighbors.

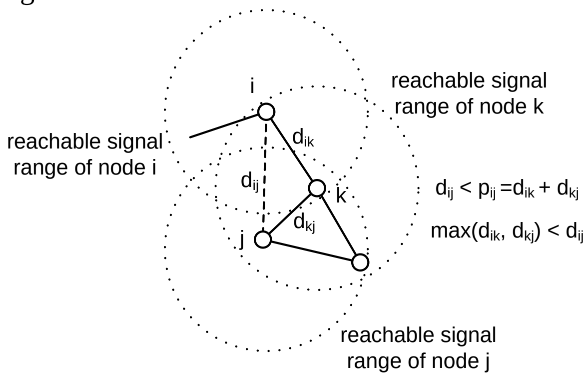


Figure 1: An example of geometric relationship between neighbor nodes. The reachable signal range of each node is denoted by dot.

Considering the above constraints, the above problem can be reformulated by the following equation.

$$\min_x F(x) = \frac{1}{2}\sum_{i,j,i\neq j}\{w_{ij}(d_{ij}-p_{ij})\}^2$$

subject to $c_k(x) < 0$, $k \in N_2$

$$c(x) = \sum_{k=1}^{2N2} c_k(x) \quad (16)$$

where N_2 is the number of two hop neighbor pairs and $c_k(x)$ is $c_k^+(x) = d_k - p_k - \alpha_+$ or $c_k^-(x) = \max(p_{iNi}, p_{jNj}) - d_k - \alpha_-$.

Then this constrained problem can be reformulated as follows.

$$\min_x F_B(x) = \frac{1}{2} f_B(x)^T f_B(x) = \frac{1}{2}\sum_{i,j=1,i\neq j} f_{B,ij}^2(x) \quad (17)$$

where $f_{B,k}(x,\lambda) = w_k(d_k - p_k) + \lambda B_k(x)$

Now introduce logarithmic barrier function in order to include inequality constraints[8].

$$B_k(x) = \begin{cases} -\log(-c_k^+(x)) - \log(-c_k^-(x)) & for\ 2\ hop\ neighbor\\ 0 & for\ others\end{cases} \quad (18)$$

For the index k consisting of the 2 hop neighbors i and j in the Jacobian, the each term of that row is as follows.

$$J_{Bk,l}(x)=\begin{cases}\dfrac{\hat{x}_i-\hat{x}_j}{d_k}*\left(w_k-\dfrac{\lambda}{c_k^+}+\dfrac{\lambda}{c_k^-}\right) & for \quad l=i\\ \dfrac{\hat{x}_j-\hat{x}_i}{d_k}*\left(w_k-\dfrac{\lambda}{c_k^+}+\dfrac{\lambda}{c_k^-}\right) & for \quad l=j\\ \dfrac{\hat{y}_i-\hat{y}_j}{d_k}*\left(w_k-\dfrac{\lambda}{c_k^+}+\dfrac{\lambda}{c_k^-}\right) & for \quad l=2*i\\ \dfrac{\hat{y}_j-\hat{y}_i}{d_k}*\left(w_k-\dfrac{\lambda}{c_k^+}+\dfrac{\lambda}{c_k^-}\right) & for \quad l=2*j\\ 0 & for \quad others\end{cases} \tag{19}$$

The Jacobian can be computed analytically. For a 2-D n-node network, the problem has $2N$ variables and $2N_2$ constraints. Thus this nonlinear least square problem can be solved by optimization techniques such as Levenberg-Marquardt method[9]. Usually only the first few iterations of the optimization algorithm give significant improvement. Thus the maximum number of iteration is set to a small number, e.g., 30. In this paper, we use a variant Levenberg-Marquardt algorithm with the barrier constraints to solve this problem. The algorithm is as follows.

====================================

$k=0$; $\nu=2$; $x=x_0$; $\lambda=\lambda_0$

$A=J_B(x,\lambda)^T J_B(x,\lambda)$; $g=J_B(x,\lambda)^T f_B(x,\lambda)$

found = $(\| g \|_\infty \le \varepsilon_1)$

$\mu=\tau*\max\{a_{ii}\}$

while (not found and $k \prec k_{\max}$)

  $k=k+1$

  solve $(A+\mu I)h_B=-g$

  if $\| h_B \| \le \varepsilon_2(\| x \|+\varepsilon_2)$

    found = true

  else

    $x_{new}=x+h_B$

    $\rho=(F(x)-F(x_{new})/(L(0)-L(h_B))$

    $\lambda=\lambda/2$

    if $\rho \succ 0$

      $x=x_{new}$

      $A=J_B(x,\lambda)^T J_B(x,\lambda)$; $g=J_B(x,\lambda)^T f_B(x,\lambda)$

      found = $(\| g \|_\infty \le \varepsilon_1)$

      $\mu=\mu*\max\{\frac{1}{3},1-(2\rho-1)^3\}$; $\nu=2$

    else

      $\mu=\mu*\nu$; $\nu=2*\nu$

  end

====================================

3 Simulation Results

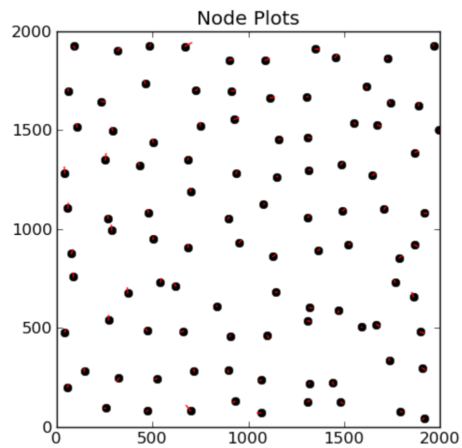
In this section, we evaluate the proposed localization method through simulation. We assume that the nodes in the sensor network are randomly deployed in 20m x 20m square topology with 100 nodes and 4 anchors on each side. The range model is the *noisy disk* model of ranging in which each node obtains a range estimate with Gaussian noise σ to all neighbors within a maximum range 250 centimeters. The simulation program was implemented by using Silhouette, which is a sensor network localization simulator[10].

This simulation does not only measure node position but also average success rate as well as sum of position estimation error. The average success rate denotes as the average ratio of the number of nodes that have been localized to all nodes under 5 different node configurations. The sum of position estimation errors represents the average sum of the square of difference between true and estimated position for all nodes.

Figure 2 shows that the localization results based on the MDS-MAP, MDS-MAP(P), and the proposed algorithm, MDS-MAP(P, O) on a randomly deployed configuration. The dots denote the true node positions under a predefined coordination system. The true and estimated locations of the same node are connected by a solid line.

The longer the line, the larger the error is. The figure shows that the MDS based localization algorithms work well on the uniform topology and MDS-MAP and MDS-MAP(P, O) outperform than MDS-MAP(P) method. MDS-MAP performs well on the regular uniform topology. The distributed version, MDS-MAP(P) can be improved by refining the local maps.

(a)



(b)

(c)

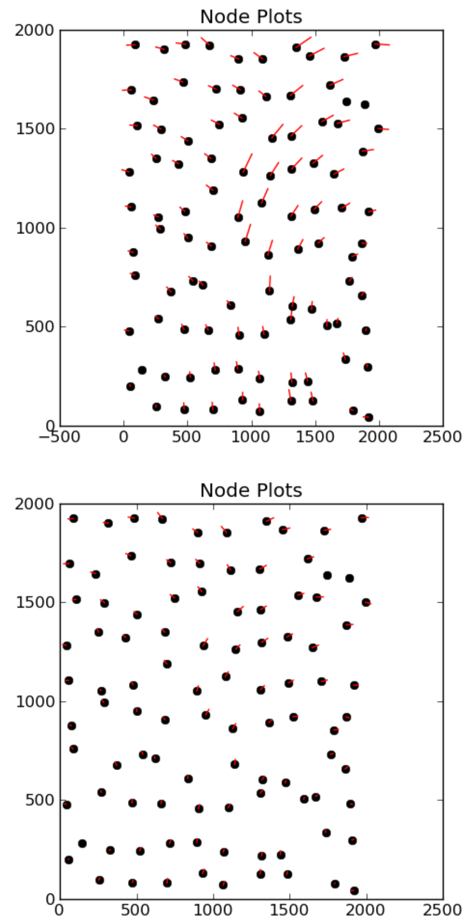


Figure 2: The position estimation results under three algorithms, (a) MDS-MAP, (b) MDS-MAP(P), and (c) MDS-MAP(P, O). The true position is denoted by dot.

Table 1 shows the average success rates and sums of position estimation errors for those algorithms. MDS-MAP obtains better success rate, that is, the more localized nodes. The un-localized nodes with MDS-MAP(P) or MDS-MAP(P, O) represent that the conditions of localizing a node is not satisfied in the local map. The sum of estimation errors is 23743, 380107, and 73080, respectively. MDS-MAP obtains better performance. It means that more localized node produce better position estimation results. Moreover, MDS-MAP(P, O) is also better. It implies that the inaccurate estimation of local nodes can be improved considerably with a refinement algorithm.

Table 1: The average success rates and position estimation errors of three algorithms, MDS-MAP, MDS-MAP(P), and MDS-MAP(P, O) under 5 trials.

	MDS-MAP	MDS-MAP(P)	MDS-MAP(P,O)
Success Rate	99.2 %	97.0 %	97.0 %
Estimation Error	23743 cm^2	180107 cm^2	73080 cm^2

4 Conclusions and future work

In this paper, we proposed a distributed network localization method based on multidimensional scaling and constrained optimization technique to solve the network localization problem. This method based on multidimensional scaling builds a local map for each node. The relative maps are of high quality when connectivity is sufficiently high. Then the method uses the least squares minimization algorithm and two hop distance constraints in order to refine the local maps. Our simulation results show the effectiveness of the proposed method for regular topology.

The further studies include extensive evaluation for more cases with regard to irregular node topology and various noise effects.

Acknowledgements

This work was supported by Dong-eui University Grant(2010AA181). We give thanks to Professor Kamin Whitehouse in the University of Virginia for his open source software, Silhouette. The authors also thank the reviewers for their comments.

References

[1] N. Kosmatov, Azzedine Boukerche, Editor, Algorithms and Protocols for Wireless Sensor Networks, (John Wiley and Sons Inc., (2009).

[2] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan, The cricket location-support system, *Mobile Computing and Networking*, Boston, MA, August (2000), 32-43.

[3] D. Niculescu and B. Nath, Ad hoc positioning system (APS), *IEEE Global Communications Conference (GlobeCom '01)*, San Antonio, TX, (2001), 2926-2931.

[4] L. Doherty, K. S. J. Pister, L. El Ghaoui, Convex position estimation in wireless sensor networks, *IEEE INFOCOM '01*, (2001).

[5] Y. Shang, W. Ruml, Y. Zhang and M. Fromherz, Localization from mere connectivity, *ACM MobiHoc*, Annapolis, MD, June (2003).

[6] Y. Shang and W. Ruml, Improved MDS-based localization, *IEEE INFOCOM '04*, (2004).

[7] C.-H. Wu, W. Sheng, and Y. Zhang, Mobile self-localization using multi-dimensional scaling in robotic sensor networks, *International Journal of Intelligent Control and System*, Vol.11, No.2, (2006), 163-175.

[8] J. Nocedal and S. J. Wright, Numerical Optimization (Springer Series in Operations Research and Financial Engineering), (Springer, 1999).

[9] K. Madsen, H. B. Nielsen and O. Tingleff, Methods for Non-linear Least Squares Problems, 2nd Ed., (Informatics and Mathematical Modelling, Technical University of Denmark, April 2004).

[10] K. Whitehouse and D. Culler, A Robustness analysis of multi-hop ranging-based localization approximations, *IPSN '06*, Nashville, TN, (2006).

Seong-Woo Kim received his B.S., M.S., and Ph.D. degrees in Electrical Engineering from Korea Advanced Institute of Science and Technology (KAIST), in 1991, 1993, and 1999 respectively. He worked at Electronics and Telecommunications Research Institute (ETRI) from 1999 to 2001. Since 2002, he has been a professor at Dong-Eui University. From Aug. 2008 to Aug. 2009, he visited the California State University at Fresno as a visiting professor. His research interests include wireless sensor networks and embedded operating system.

Jong Min Lee received the B.S. degree in computer engineering from Kyungpook National University, Korea, in 1992, and the M.S. and the Ph.D. degrees in Computer Science from Korea Advanced Institute of Science and Technology (KAIST) in 1994 and 2000, respectively. From Sept. 1999 to Feb. 2002, he worked for Samsung Electronics as a senior engineer. Since 2002, he has been a faculty member of the Department of Computer Software Engineering, Dong-Eui University. From Feb. 2005 to Feb. 2006, he visited the University of California, Santa Cruz as a research associate. His research interests include location management in mobile computing, routing in ad hoc networks and sensor networks, and parallel computing.

Jung-Hwa Lee received his B.S. and the Ph.D. degrees at the Department of Computer Science from Pusan National University, Korea, in 1995 and 2001, respectively. He is a professor at the Department of Computer Software Engineering, Dong-Eui University in Korea. His research interests include database and information processing, etc.