Multi-hop-based Monte Carlo Localization for Mobile Sensor Networks

Jiyoung Yi , Sungwon Yang and Hojung Cha

Department of Computer Science, Yonsei University
Seodaemun-gu, Shinchon-dong 134, Seoul 120-749, Korea
{jyyi, swyang, hjcha}@cs.yonsei.ac.kr

Abstract—**Many low-cost localization techniques have been proposed for wireless sensor networks. However, few consider the mobility of networked sensors. In this paper, we propose an effective and practical localization technique especially designed for mobile sensor networks. Our system is based on the sequential Monte Carlo method, but dissimilar to other conventional localization schemes, our algorithm covers a large sensor field with very few anchor nodes by information flooding. The algorithm works without knowledge of the maximum transmission range, and covers some of the problems caused by the flooding beacons. We discuss factors to implement the algorithm in the real world and present several solutions. Our mechanism is implemented in a real environment, and its feasibility is validated by experiments. The simulation results show that our algorithm outperforms conventional Monte Carlo localization schemes by decreasing estimation errors by up to 50%, and the overhead of the algorithm could be minimized by appropriately adjusting the system parameters.**

Index Terms— **location estimation, mobility, probability, wireless sensor network**

I. INTRODUCTION

In wireless sensor networks, many applications such as vehicle tracking or environment monitoring systems assume that the system knows the location of each sensor node. Since manual deployment of each node is not effective, an automatic localization technique is required. Using GPS (Global Positioning System) [1] is a simple solution, but the method is impractical due to deployment constraints and cost. Hence, developing an effective localization scheme is a fundamental and important issue in wireless sensor networks.

Numerous localization techniques for wireless sensor networks have recently been proposed. Many techniques calculate the position of nodes based on the information of a set of anchor nodes that know the locations. The methods typically assume static network topologies; however, many sensor network applications demand the consideration of mobile sensor nodes. The Princeton ZebraNet project [2] is a good example of a mobile sensor network application that explores wireless protocols and position-aware computation from a power-efficient perspective. One of the project's fundamental requirements is to find the nodes' location on mobile objects. If the project adopts a localization technique for the mobile sensors, they could reduce much of the cost for the system.

Some recent work also discusses localization when dealing with mobile nodes [3], [4]. However, most of the studies suggest that supporting mobility can be achieved by repeating the static localization algorithm. They periodically repeat the algorithms and calculate new positions for the nodes. These methods are feasible in certain circumstances; however, we believe that a new localization algorithm should specifically consider the mobility of sensor nodes.

There are several challenges to designing a localization algorithm for mobile sensor networks. First, the algorithm should work reliably in the deployment condition, where the location of the node changes continuously. Many previous localization schemes for static networks restrict environment conditions such as uniformly distributed anchor nodes or a fixed radio transmission range. In a mobile sensor network, localization schemes should overcome these assumptions. Second, the localization should have minimum overhead. Since the localization is a part of the whole application, the method cannot consume most of the resources such as CPU, battery, and network resource.

In this paper, we are interested in a localization system in which both anchor nodes and sensor nodes have mobility. The mobile sensor networks consist of position-aware anchor nodes and position-unaware general nodes. We propose a practical localization algorithm that works in a large mobile sensor network field with very few anchor nodes using the sequential Monte Carlo method [5]. The localization technique is based on global information gathered through multi-hop propagation. With the multi-hop anchors information, every general node constructs location boundaries where the nodes could possibly exist and estimate the location. Our system is free from additional hardware and knowledge of the maximum radio transmission range.

The rest of this paper is organized as follows. In Section Ⅱ, we describe previous localization schemes. The motivation for our work is explained in Section Ⅲ. In Section Ⅳ, we propose a new localization technique. Section Ⅴ describes implementation issues of the system. In Section Ⅵ, we analyze the performance of our system through both experiments and simulations. Section Ⅶ concludes the paper.

II. Related Work

Various techniques to estimate the location of wireless sensor nodes are found in the literature. The well-known range-based methods typically use received radio signal strength [6], [7], time difference of arrival of different signals [8], [9] or angle of arrival [10] as a ranging measure. However, the ranging measure usually provides unreliable data [11] or requires additional devices to implement.

Several localization schemes have been designed to localize sensor nodes without range information. Many such techniques assume that some nodes already know their global positions, and information from these nodes is used to estimate the unknown node's location. Many schemes focus on improving the accuracy. Centroid [12] by Nirupama et al. is a simple localization scheme without ranging. Each node in the system calculates the center of the locations of all anchor nodes the node hears and regards the center as its position. Location error can be reduced by deploying the anchor nodes in good positions [13], but this assumption is not appropriate for a mobile ad-hoc network. Niculescu et al. [14] proposed an ad-hoc positioning system (APS). APS includes three different propagation methods: DV-hop, DV-distance, and Euclidean. These methods obtain relatively accurate results in multi-hop networks. Among the methods, DV-hop converts hop count to the distance between an unknown node and an anchor node. The system calculates a corrected factor, which means the average distance of a hop, and uses the factor for conversion. The method is implemented in other work and compared with other mechanisms [15]. Lim et al. [16] proposed the Proximity Distance Map (PDM), which calculates the particular transformation matrix to convert hop count to distance. The matrix characterizes anisotropic network topologies and proximities to the anchor nodes in all directions. Shang et al. [17], [18] proposed multi-dimensional-scaling-based localization methods MDS-map and MDS-map(P). MDS-map, the first model of their scheme, gathers the connectivity information of the sensor nodes and builds a relative map of the sensor field by using multi-dimensional scaling. This method is accurate compared to other methods. However, the time complexity of constructing a relative map is $O(n^3)$, where n is the number of nodes. Shang et al. also proposed the distributed version of MDS-map, MDS-map(P), but the methods are not appropriate for a mobile network due to the computational complexity.

MCL [19] is designed for mobile sensor networks based on the sequential Monte Carlo method. Sensor nodes randomly predict their positions based on their previous positions, and filter the prediction by using the transmission range of the seed nodes. When the nodes obtain a sufficient number of position samples, the locations are estimated by calculating the center of the sample positions. Dissimilar to the other localization method, the movement of sensor nodes improves the method's estimation accuracy. A range-based version of MCL has also been proposed [20]. This version provides a sampling and filtering method based on range measurements, and weighs each valid sample to obtain accurate estimation results. Baggio et al. [21], [22] proposed an enhanced Monte Carlo localization scheme, MCB. By reducing the sampling area, MCB draws good samples, and thereby the number of iterations to construct the sample set is reduced. Computation overhead is reduced by the mechanism; however, the algorithm still depends on specific parameters such as fixed radio transmission range.

In our work, we propose a new localization mechanism based on the sequential Monte Carlo method with multi-hop propagation. Our method keeps the strengths of previous Monte Carlo localization algorithms and removes some of their weaknesses.

III. MOTIVATION

Previous range-free localization algorithms designed for mobile sensor networks [19], [21], [22] have two major constraints. First, a sufficient number of anchors are required for the algorithms. The position estimation of MCL depends on local anchor information; hence, the location error could be large when the density of anchor nodes is low. Although MCL is an extension approach that includes information about the neighbors, the problem is not solved completely. According to the simulation results of MCL [19], MCL needs more than one anchor in a one-hop transmission range to obtain reasonable accuracy. This number of anchors is relatively small but not sufficient to apply the algorithm in a real environment. We suggest that flooding anchor information allows the system to work with a few anchors in the sensor network field.

Second, the previous algorithms assume that the fixed radio transmission range is known. In a real environment, however, the radio range varies by residual battery, geometric characteristics, and many other factors. Especially, the maximum transmission range is affected by how far away a node is from the ground. Often, in mobile sensor deployment, the height of a node usually varies due to the feature of the mobile object. Since figuring out the radio transmission range is difficult in real environments, we should remove the assumption.

These constraints are possibly lifted by DV-hop [14]. It floods anchor information to entire sensor field. Every anchor receives information about the other anchors and calculates the average one-hop distance by using the information. The average distance is used to convert the nodes' hop count to actual distance. However, the distance conversion shows negative results such as unstable transmission range in real deployment [23].

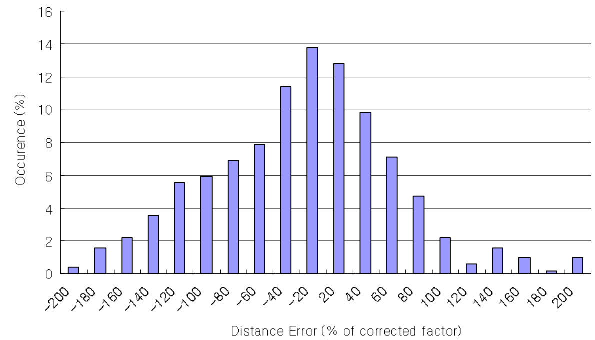


Fig. 1. Distance error occurrence with DV-hop (experiment)

We implemented the DV-hop propagation method and observed the difference between the actual distance and the estimated distance. We deployed 25 sensor motes in various topologies. Several fundamental problems should be considered when deploying sensor network applications in the real world. We discuss the implementation issues in Section Ⅴ. Fig. 1 illustrates the error distribution for the distance estimation experiment. The results show that DV-hop causes both overestimation and underestimation problems. We also simulated the DV-hop to analyze the distance estimation error in large mobile sensor networks. Details of the simulation are described in Section Ⅵ. Fig. 2 illustrates the simulation results obtained with 400 nodes in a 500mx500m of network field with a 50m transmission range.

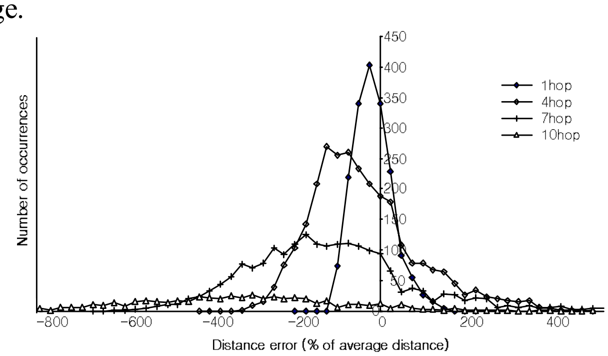
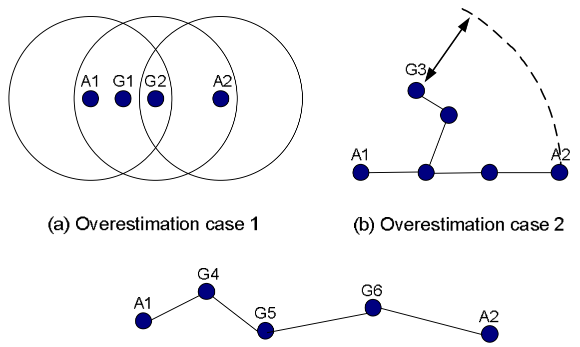


Fig. 2. Distance error occurrence with DV-hop (simulation)

The simulation results show that overestimation occurs more frequently than underestimation. The results also show that DV-hop overestimates the distance badly when the hop count is large. Some of the reasons which occur the error explain these results.



(a) Overestimation case 1 (b) Overestimation case 2

(c) Over- and under- estimation case

Fig. 3. Cases causing distance estimation error

Fig. 3 shows the specific cases in which distance estimation error is occurred. In Fig. 3 (a), overestimation occurs when general node G1 computes the distance from the anchors, A1 and A2. In DV-hop, the hop distance is calculated from the ratio of total absolute distances to the total hop counts between anchors. The calculated hop distance would be similar to the actual radio transmission range; however, G1 is not on the boundary of the radio range. Although G1 is closer to the anchor A1 than G2, DV-hop regards G1 as being apart from A1 as far as G2. Fig. 3 (b) illustrates the situation in which the propagation path does not form a straight line. Although G3 is relatively close to A1, the path from A1 to G3 generates three hop communications; hence, G3 assumes it is on the dashed arc. This situation occurs often when the number of hop counts from an anchor is large. Under a condition such as Fig. 3 (c), both overestimation and underestimation arise. When G5 calculates the distance from the anchors A1 and A2, G5 overestimates the distance from A1 and underestimates the distance from A2. Even in the situation of anisotropic deployment of nodes, DV-hop calculates only the average hop distance between anchors and applies the value to the whole network.

The distance error of DV-hop increases when applied to a mobile sensor network since the network topology changes during the propagation progress. Distance overestimation and underestimation of the DV-hop propagation algorithm occurs for different reasons and with different frequency; hence, proper solutions for each case are required to apply the DV-hop in a real environment. In our work, we adopt the DV-hop propagation method as a part of our localization system to calculate the average hop distance. We modify the estimated distance to apply DV-hop to our system.

IV. Multi-hop-based Monte Carlo Localization

In this section, we describe the proposed localization algorithm for mobile sensor networks. First, we state the problem and describe an overview of previous work on which we conceptually base our algorithm. Our localization algorithm is described subsequently.

A. Background

We consider the localization problem for a mobile sensor network. Sensor nodes are deployed in the sensor field, and all have their own mobility. The network topology can be dynamically changed by mobile nodes. We are particularly interested in the situation when the density of anchor nodes is low. We assume that full network connectivity is guaranteed in spite of node mobility.

A sensor field consists of two types of sensor nodes: general nodes and anchor nodes. General nodes are not aware of their locations, whereas anchor nodes always know their exact positions and send beacon messages containing location information. We assume that at least three anchor nodes exist in the network and that all nodes are equally likely to move in any direction with any speed between 0 and v_{max}. The problem is to estimate the location of general nodes with knowledge about the position of anchor nodes and the hop count between the anchor nodes and general nodes.

The sequential Monte Carlo (SMC) [5] method is a group of simulation-based methods from a sequence of probability distributions. The method is extensively used to solve sequential Bayesian inference problems in econometrics, signal processing, and robotics. SMC methods approximate the sequence of probability distributions of interest using a large set of random samples. As the number of particles goes to infinity, the convergence of these particle approximations toward the sequence of probability distributions can be ensured under weak assumptions.

Localization techniques based on SMC were developed early in robotics [24]. Robotics localization assumes a pre-learned map, and tries to estimate the robot's position based on its motion and perceptions. However, in sensor networks, nodes have limited knowledge of local information and computational power. The sensor network version of the SMC-based localization was introduced by MCL [19]. MCL uses the local anchors' locations and the maximum radio transmission range.

The MCL algorithm is as follows: At the first step, a set of

Propagation

Initial propagation: Initially the anchors have no knowledge of other anchors. flood beacon message without corrected factor

for each anchor node i **do**
calculate corrected factor
flood beacon message
end do

Location Estimation

for each general node j **do**
$L_t = \{\}$
while(size(L_t) < N) **do**
//Construct prediction area with multi-hop constraints
$P_t = \{x_{\min}, x_{\max}, y_{\min}, y_{\max}\}$

//Prediction
$R = \{l_t \mid l_t \text{ is selected witin } P_t \text{ with } p(l_t \mid l_{t-1}) > 0\}$

//Filtering
$R_{filtered} = \{l_t \mid l_t \in R \wedge p(o_t \mid l_t) > 0\}$

$L_t = \text{choose}(L_t \cup R_{filtered}, N)$
end do
end do

Figure 4. Multi-hop-based Monte Carlo Localization

positions, $L_0 = \{l_0^1, l_0^2, ..., l_0^{N-1}\}$ is randomly selected for the initial locations of the nodes. A prediction phase follows in which every node predicts its position based on the previous location and maximum speed. The filtering phase, after prediction, decides if the predicted position is valid. The nodes use one- and two-hop anchors' information to filter the position. One-hop anchors are assumed to be within the radio transmission range r of the sensor node. Two-hop anchor positions, gathered from neighbor nodes, are used for negative information. In other words, the anchors are assumed to be in the range $2r$ but not in a radius r. The nodes repeat the prediction and filtering until they obtain a sufficient number of valid positions. A node considers the center of the valid positions to be its new location. These processes, except for initialization, operate at every time unit.

B. *Multi-hop-based Monte Carlo Localization*

In this section, we describe our localization scheme, Multi-hop-based Monte Carlo Localization (MMCL). In MMCL, anchors flood location information periodically. Each node uses the multi-hop information and constructs location boundaries shaped as rings. The MMCL localization procedure consists of two parts. The first part provides an average hop distance among anchor nodes as DV-hop does. Every sensor node obtains the average hop distance from the anchor nodes and calculates its position. The second part is to run the sequential Monte Carlo process in each sensor node. The MMCL localization system repeats these processes in every time period. Fig. 4 provides the overview of the MMCL localization algorithm.

The hop distance calculation is similar to DV-hop. In the beginning of each localization period, every anchor node floods the sensor field with a beacon message that contains the anchor's exact location, ID, sequences, and hop counts initially set to zero. Each unknown sensor node maintains an anchor table $\{ID_i, x_i, y_i, h_i, s_i\}$ where x_i and y_i represent the position of the anchor, h_i the hop counts, and s_i the sequence number. The node updates the table when the node receives a beacon message from a new anchor, a new sequence number, or smallest hop counts from a anchor. When an anchor node i receives a sufficient number of location messages from other anchor nodes, the corrected factor is calculated as shown in Equation (1) [14]. In traditional DV-hop, the anchor node floods the corrected factor c_i to unknown nodes after the beaconing, which generates large network communication overhead. In our algorithm, we attached the corrected factor to the next beacon message, i.e., the beacon messages have another data field for the corrected factor. Hence, we use only half of the packet transmissions, compared to DV-hop. We stochastically analyzed the characteristics of the mobile network and obtained parameters to modify the distance calculation for considering distance estimation error.

$$c_i = \frac{\sum \sqrt{(X_i - X_j)^2 + (Y_i - Y_j)^2}}{\sum h_i} \tag{1}$$

$i \neq j$, for all anchor nodes j

The second part of MMCL, the position calculation, is operated on each unknown node at fixed periods. During the period, nodes gathers anchor information and relay the data to other nodes and calculate the position at the end of the period. Instead of deciding the distance from an anchor node, each node assumes that it is located in a certain range from the anchor node. Equation (2) represents the distance range that each node assumes. The size of the distance range depends on the value of α and β. The parameters should be decided carefully since the unknown node must exist in every distance range to calculate the position correctly. The definite values of α and β are described in Section Ⅳ-*C*.

$$c_i \times \alpha \le d_i \le c_i \times \beta \qquad (2)$$

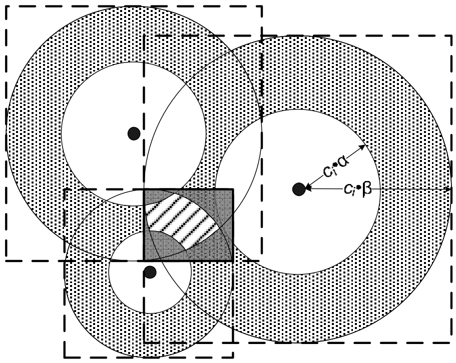


Fig. 5. Multi-hop constraints, prediction area, and valid region

Before predicting the position of an unknown node, the node draws a prediction area, as shown in Fig. 5. The dark rectangle in the middle of Fig. 5 illustrates the prediction area. The maximum speed of the node is also used for prediction area estimation. The coordinates of the rectangle area are represented as follows:

$$x_{\min} = \max(\max_{i=1}^{n}(x_i - c_i \cdot \beta), x_{t-1} - v_{\max}) \qquad (3)$$

$$x_{\max} = \min(\min_{i=1}^{n}(x_i + c_i \cdot \beta), x_{t-1} + v_{\max}) \qquad (4)$$

$$y_{\min} = \max(\max_{i=1}^{n}(y_i - c_i \cdot \beta), y_{t-1} - v_{\max}) \qquad (5)$$

$$y_{\max} = \min(\min_{i=1}^{n}(y_i + c_i \cdot \beta), y_{t-1} + v_{\max}) \qquad (6)$$

If the anchor distances and speed constraint are disjointed, the node assumes that the previous location has a large error, and excludes the speed constraint for prediction area estimation. The prediction area helps to draw good samples during the prediction phase [21].

In the prediction phase, each node generates uniformly distributed random values inside the prediction area. After the prediction, the node goes through the filtering phase. If the prediction is located in a valid range from every anchor, the node saves the position as a sample. Otherwise, the node discards the prediction and finds another position. The entire prediction and filtering phases in a node are represented as follows:

Prediction

$$p(l_t \mid l_{t-1}) = \begin{cases} 1 & \text{if } x_{\min} \le x_t \le x_{\max} \cap y_{\min} \le y_t \le y_{\max} \\ 0 & \text{otherwise} \end{cases}$$

Filtering

$$\mathit{filter}(l_t) = \forall a \in A, \quad c_a \cdot \alpha \le d(l_t, a) \le c_a \cdot \beta$$

Here, A is the set of anchors that an unknown node has recognized, and $d(l_t, a)$ represents the Euclidean distance from anchor a to predicted location l_t. After obtaining a sufficient number of valid samples, the nodes regard their locations as the average of the valid samples. All of these steps are processed at every estimation period.

C. Parameter Decision

This section describes the decision process of the parameters, α and β, in Equation (2). These parameters present appropriate distance ranges from anchor nodes to general nodes. To decide the specific parameter value of MMCL, we simulated the DV-hop propagation method and analyzed the distance estimation errors. Simulation results were obtained with an average of 30 executions with different random number generator seeds. We assumed 75 to 400 sensor nodes were deployed in a 500mx500m area of the sensor field. Each sensor node has a 50m radio transmission range and the same length of maximum speed. Three anchors among the sensor nodes calculate the corrected factor and flood the information about the location and the calculated corrected factor.

$$\alpha = h_i \cdot 0.4 \qquad (7)$$

$$\beta = h_i + 2 \qquad (8)$$

As we noted in Section Ⅲ, it is necessary to consider underestimation and overestimation errors separately to compensate for the distance error. Fig. 2 in Section Ⅲ showed that the distance error is in proportion to the hop count in the case of overestimation. We constructed various boundaries according to the hop counts. Fig. 6 shows the ratio of nodes that exist in the boundaries. When we construct the boundary as 40% of the estimated distance by DV-hop, only a few percentages of nodes are positioned inside the boundary. In other words, most of the distances between an anchor and nodes are greater than $c_i \cdot h_i \cdot 0.4$, where c_i and h_i represent the corrected factor and hop count from anchor *i*, respectively. We choose the boundary as a negative constraint of MMCL; hence, α is set as Equation (7), where h_i is the hop count from the anchor *i*, and every node regards it as located out of the boundary.

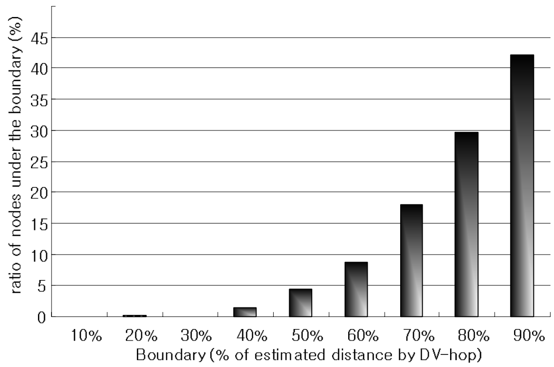


Fig. 6. Domain of overestimation

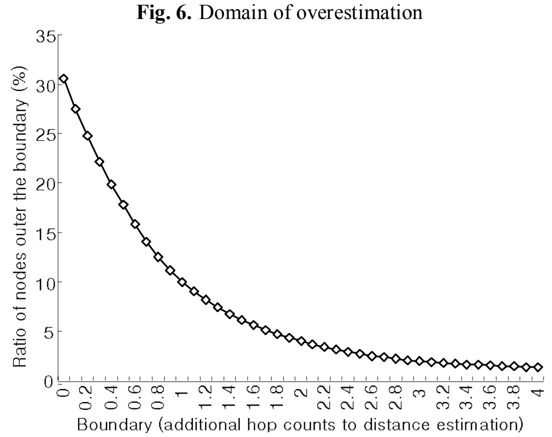


Fig. 7. Domain of underestimation

In the case of underestimation, the distance estimation error of DV-hop occurs in two cases: non-uniform nodes density and the movement of nodes during beacon propagation. Since underestimation is not affected by other elements, we added increasing values to hop counts and constructed the boundary as a multiple of the corrected factor and the increased hop counts. Fig. 7 illustrates the ratio of nodes that are located out of the tested boundary. According to the results, more than 95% of the nodes are covered by the boundary of two additional hop counts. The boundary is chosen as a positive constraint of MMCL, which means that a node exists inside the boundary. Hence, β is represented as Equation (8). With the decision about α and β, Equation (2) covers the experimental result in Fig. 1.

V. EVALUATION

In this section, we evaluated MMCL using experiments and simulations. The real experimental results validate the feasibility of MMCL, and the simulation results show the performance characteristics of our method in mobile sensor networks.

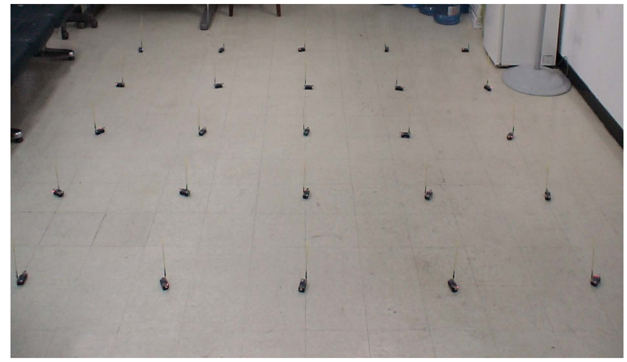


Fig. 8. System deployment

A. Experiment Results

To validate the feasibility of our work, we implemented the algorithm in a real environment and observed the performance of the algorithm. We used Tmote Sky [25] and TinyOS [26] for implementation. The radiation pattern of the internal antenna in Tmote Sky is uneven so that the difference in the measured RSS is up to 20dBm [27]. We modified the mote hardware and attached an external antenna. For the evaluation, we controlled the radio transmission power and deployed the system in a small region. Also we used external antenna for horizontally omnidirectional radio pattern. Although the system was not deployed in large outdoor field, we adjusted the system to work similar to the large field in relatively small area. Twenty-one general nodes and four anchor nodes were deployed in grid topology. The unit length of the grid was 90cm. We set the RSSI constraint to -75dbm, which allows reliable communication within 90cm but rarely transmits to 180cm far. Fig. 8 shows a snapshot of the deployment of the system.

Evaluating the accuracy of localization is difficult while the nodes are moving; hence, we estimated the nodes' locations in a static situation for the evaluation. Fig. 9 shows the location estimation error of each node. The average location error was about 95cm. In the real-world implementation, it is difficult to find out the exact maximum transmission range; hence, we cannot compare the results directly with those of the simulation. However, we know the transmission range is longer than 90cm and less than 180cm. If we apply 90cm as the maximum radio range, the average location error ratio to radio transmission range is 1.06*r*, where *r* represents the radio range.

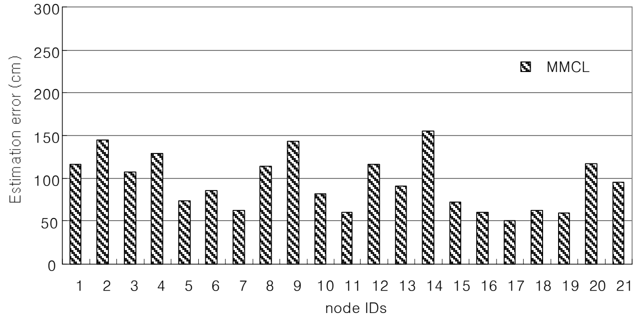
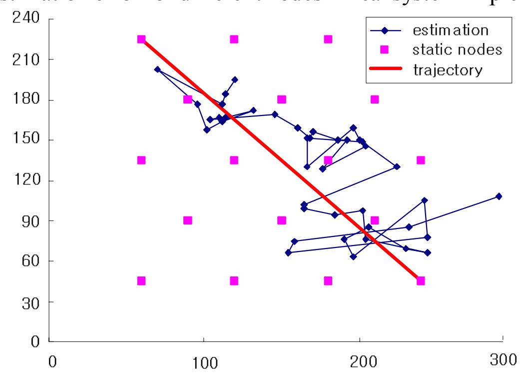
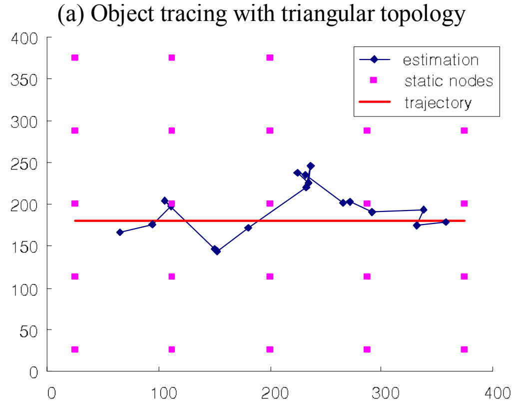


Fig. 9. Estimation error for different nodes in real system implementation



(a) Object tracing with triangular topology



(b) Object tracing with grid topology

Fig. 10. Tracking mobile node with MMCL

Instead of measuring the exact accuracy of nodes in a mobile environment, we obtained a trace of a mobile node in network fields. Fig. 10 shows the results of tracing a mobile node in different topologies. Static nodes guarantee the connectivity of the nodes. The estimated line follows the real trajectory of a node. The experimental results validate that MMCL works reasonably well in a real environment.

B. Simulation Results

The general performance of MMCL was analyzed through simulation. We observed the characteristics of MMCL in various conditions and large networks. In our simulation, we varied the parameters of the sensor networks. For the simulations, 400 nodes were deployed in rectangular 500m x 500m region. We assumed the transmission range was 50m for all nodes. We varied the parameters such as the number of anchors, speed of nodes, TTL (maximum hop count) of a beacon message, and transmission range.

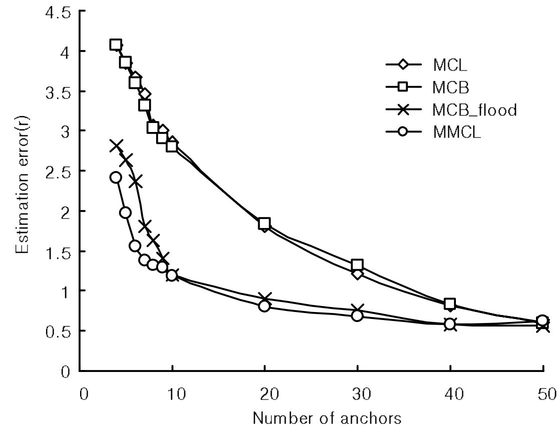


Fig. 11. Estimation error vs. anchors number (speed of nodes: 1*r*, TTL: 15)

The accuracy of localization is directly related to the number of anchors. As mentioned earlier, previous Monte Carlo method-based localization algorithms require relatively large number of anchors. Fig. 11 shows the impact of the number of anchors on the localization error. Although [19] barely mentioned information flooding, we assumed a flooding version of MCB (MCB_flood) for comparison. Estimation errors include un-localized nodes that chose random positions as their location, and the results are represented as the ratio of transmission range. MCL has an average 4*r* of location error ratio with four anchors since most of the nodes failed to obtain anchor information. MCB reduces the computation overhead, but the accuracy is similar to MCL. Both mechanisms work well with 40 anchor nodes. MMCL outperforms other algorithms with 50% less error with few anchors. The accuracy of MMCL is similar to the other mechanism when the number of anchor nodes is large. MCB_flood is as accurate as MMCL; however, the mechanism has a critical assumption, about the knowledge of the fixed radio range, in its mechanism.

As we described in Section Ⅴ, finding maximum radio transmission ranges is difficult in the real world since the transmission range varies due to many other factors such as the remaining power, height of a node, obstacles and so on. To investigate the effects of wrong information about the transmission range, we simulated MCB_flood with the wrong transmission range. We inputted 40 anchors for the simulation. Fig. 12 shows the impact of transmission range prediction error. If the expected radio range is smaller than the actual radio range, MCB_flood generates poor result. With a short transmission range, MCB_flood fails to construct a sampling box and picks random values for localization. When the knowledge of the radio range is longer than the actual range, the estimation error also reduced since the large transmission range constructs too large a sampling box. On the other hand, MMCL does not require pre-knowledge about the transmission range; hence, the result is not influenced by incorrect knowledge about radio range. This result also shows that MMCL is adaptable in real environments.

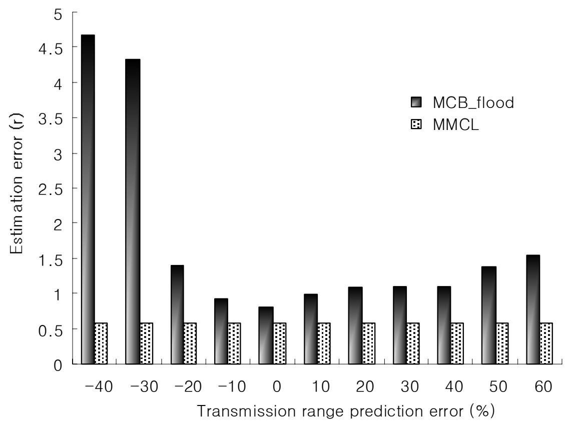


Fig. 12. Estimation error vs. knowledge of transmission range

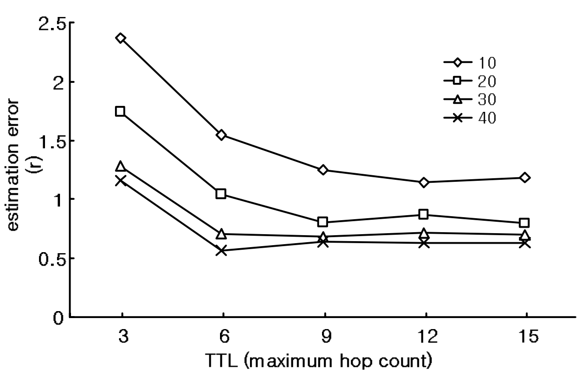


Fig. 13. Estimation error vs. TTL (number of anchors: 30, speed of nodes: $1r$)

Although MMCL is appropriate for a mobile sensor network, the mechanism has weaknesses. One problem is that flooding requires large communication overhead. Since many applications in mobile sensor networks are not only for tracking objects, communication overhead should be adaptively reduced. One easy way to decrease the network overhead in MMCL is controlling flooding. Instead of global flooding, we could set up the maximum hop count (TTL) of a beacon message. To observe the impact of TTL, we varied the TTL for the system and simulated the accuracy of MMCL. Fig. 13 shows the simulation result how TTL influences accuracy. MMCL requires sufficient TTL, but TTL does not affect estimation error after some point. The point could change due to the network size or number of anchors since MMCL requires multi-hop communication between anchors. However, the results reveal that the system could reduce the communication overhead without loss of accuracy.

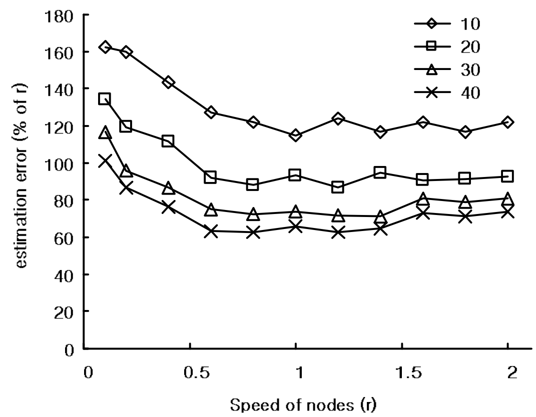


Fig. 14. Estimation error vs. speed of nodes (TTL: 15)

Increasing the speed of the moving nodes is similar to increasing the unit time between location estimations. If the location estimations are less frequent, the relative communication overhead would decrease. We varied the maximum speed of nodes $v_{\max}$. In the simulation, every node moves to a randomly chosen position within $v_{\max}$ from the previous location. Fig. 14 shows the result of estimation error influenced by speed of nodes. The result reveals two facts. First, the left half of the graph shows that the mobility of nodes makes the localization accurate. The second half of the graph shows the accuracy of localization is not influenced by the node's speed after some threshold. The system covering fast mobile nodes means the system supporting a long period for localization. This does not mean a reduction of the absolute communication overhead. However, the application has more time to do other work instead of localization; hence, the relative communication overhead would be diminished.

VI. CONCLUSION

In this paper, we proposed a multi-hop-based Monte

Carlo localization algorithm. By flooding anchor information to the sensor field, the number of sensors that cannot receive anchor information decreases. The simulation results show that, compared to other Monte Carlo-based algorithms, up to 50% of errors are reduced where anchor nodes are sparsely deployed. The results also show the network overhead could be reduced by controlling some of the system parameters. One of our key contributions is to have implemented the proposed algorithm and checked the feasibility of the algorithm with real experiments. We presented implementation issues and showed that our system operates reasonably well in a real environment. Future work includes the validation of our algorithm in a densely deployed environment.

ACKNOWLEDGMENTS

This work was supported by the Korea Science and Engineering Foundation (KOSEF) through the National Research Lab. The program was funded by the Ministry of Science and Technology (No. M10500000059-06J0000-05910), and the ITRC (Information Technology Research Center) programs of the IITA (Institute of Information Technology Advancement) (IITA-2006-C1090-0603-0015).

REFERENCES

[1] B. Hoffman-Wellenhof, H. Lichteneeger, and J. Collins, *Global Positioning System: Theory and Practice* (4th ed.). New York: Springer-Verlag, 1997.

[2] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. S. Peh, and D. Rubenstein, “Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences with ZebraNet,” in *Proc. ASPLOS-X*, San Jose, 2002, pp. 96-107.

[3] L. Lazos, S. Capkun, and R. Poovendran, “ROPE: Robust position estimation in wireless sensor networks,” in *Proc. IPSN*, Los Angeles, 2005, article No.43.

[4] D. Moore, J. Leonard, D. Rus, and S. Teller, “Robust distributed network localization with noisy range measurements,” in *Proc. Sensys*, San Diego, 2004, pp. 50-61.

[5] Fishman, G. S. (ed.), *Monte Carlo Concepts, Algorithms, and Applications*. New York: Springer-Verlag, 1996.

[6] P. Bahl and V. N. Padmanabhan, “RADAR: An in-building RF-based user location and tracking system,” in *Proc. INFOCOM*, Tel Aviv, Israel, 2000, pp. 775–784.

[7] K. Yedavalli, B. Krishnamachari, S. Ravula, and B. Srinivasan, “Ecolocation: A technique for RF-based localization in wireless sensor networks,” in *Proc. IPSN*, Los Angeles, 2005, article No. 38.

[8] N. Pryiantha, A. Chakaborty, and H. Balakrishnan, “The Cricket location-support system,” in *Proc. MobiCom*, Boston, 2000, pp. 32–43.

[9] A. Savvides, C. C. Han, and M. B. Srivastava, “Dynamic fine-grained localization in ad-hoc sensor networks,” in *Proc. Mobicom,* Rome, Italy, 2001, pp. 166–179.

[10] D. Niculescu and B. Nath, “Ad-hoc positioning system (APS) using AOA,” in *Proc. INFOCOM*, San Francisco, 2003, pp. 1734-1743.

[11] E. Elnahrawy, X. Li, and R. Martin, “The limits of localization using signal strength: A comparative study,” in *Proc. SECON*, Santa Clara, 2004, pp. 406-414.

[12] N. Bulusu, J. Heidemann, and D. Estrin, “GPS-less low cost outdoor localization for very small devices,” *IEEE Personal Communications Magazine*, Vol. 7, No. 5, pp. 28-34. October, 2000.

[13] N. Bulusu, J. Heidemann, and D. Estrin, “Adaptive Beacon Placement,” in *Proc. ICDCS*, Phoenix, 2001, pp. 489–498.

[14] D. Niculescu and B. Nath, “Ad hoc positioning system (APS),” in *Proc. Globecom*, San Antonio, 2001, pp. 2926–2931.

[15] R. Stoleru and J. Stankovic, “Probability grid: A location estimation scheme for wireless sensor networks,” in *Proc. SECON*, Santa Clara, 2004, pp. 430-498.

[16] H. Lim and J. C. Hou, “Localization for anisotropic sensor networks,” in *Proc. INFOCOM*, Miami, 2005, pp. 138–149.

[17] Y. Shang, W. Ruml, and Y. Zhang, “Localization from mere connectivity,” in *Proc. MobiHoc*, Annapolis, 2003, pp. 201–212.

[18] Y. Shang and W. Ruml, “Improved MDS-based localization,” in *Proc. INFOCOM*, Hong Kong, 2004, pp. 2640–2651.

[19] L. Hu and D. Evans, “Localization for mobile sensor networks,” in *Proc. Mobicom*, Philadelphia, 2004, pp. 45–57.

[20] B. Dil, S. Dulman, and P. Havinga, “Range-based localization in mobile sensor networks,” in *Proc. EWSN*, Zurich, 2006, Volume 3868.

[21] A. Baggio and K. Langendoen, “Monte-Carlo Localization for mobile wireless sensor networks,” *MSN*, to be published.

[22] A. Baggio. “Monte-Carlo localization for mobile wireless sensor networks,” Delft University of Technology, Delft, Tech. Rep. PDS-2006-004, June 2006.

[23] S. W. Wong, J. G. Lim, S. Rao, and W. K. Seah, “Density-aware hop-count localization (DHL) in wireless sensor networks with variable density,” in *Proc. WCNC*, New Orleans, 2005, pp. 1848–1853.

[24] F. Dellaert, D. Fox, W. Burgard, and S. Thrun, “Monte Carlo localization for mobile robots,” in *Proc. ICRA*, Detroit,1999, pp. 1322-1328.

[25] Tmote Sky. Available: http://www.moteiv.com.

[26] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister, “System architecture directions for network sensors,” in *Proc. ASPLOS*, Cambridge, 2000, pp. 93-104.

[27] S. Yang and H. Cha, “An empirical study of antenna characteristics toward RF-based localization for IEE 802.15.4 sensor nodes,” *EWSN*, to be published.