Cooperative Node Localization for Mobile Sensor Networks

Hongyang Chen
University of Tokyo, Japan
hongyang@mcl.iis.u-tokyo.ac.jp

Marcelo H. T. Martins
University of Tokyo, Japan
martins@mcl.iis.u-tokyo.ac.jp

Pei Huang
Michigan State University, USA
dolkoff@yahoo.com.cn

Hing Cheung So
City University of Hong Kong
hcso@ee.cityu.edu.hk

Kaoru Sezaki
University of Tokyo, Japan
sezaki@iis.u-tokyo.ac.jp

Abstract

In this paper, we propose a range-free cooperative localization algorithm for mobile sensor networks by combining hop distance measurements and particle filtering. In the hop distance measurement step, a differential error correction scheme is devised to reduce the positioning error accumulated over multiple hops. A backoff-based broadcast mechanism is also introduced in our localization algorithm. It efficiently suppresses redundant broadcasts and reduces message overhead. The proposed localization method has fast converges with small location estimation error. We verify our algorithm in various scenarios and compare it with conventional localization methods. Simulation results show that our proposal is superior to the state-of-the-art localization algorithms for mobile sensor networks.

1. Introduction

Location awareness has become an important feature for many wireless sensor network (WSN) applications. Examples of such applications include position tracking, mapping, location-aided routing, and others. Due to cost and energy constraints, not all nodes may have a reliable source of location information (e.g., GPS receivers). Therefore, localization systems for WSNs usually employ a small set of nodes who are aware of their own coordinates (hereafter called *anchors*) which will distribute this information to regular nodes in the network, helping them estimate their own positions.

Many localization algorithms have been proposed over the past few years. Localization approaches for WSNs can be divided into two main categories. *Range-based* techniques require special hardware for estimating the distance between anchors and regular nodes, which may become prohibitively expensive. *Range-free* techniques, on the other hand, do not impose such demand as an anchor informs other nodes about its own position through message passing. After finishing the distance-from-anchor estimation process, a regular node can determine its own position through a variety of methods, such as multilateration, and triangulation. If necessary, an optional step is performed, in which regular nodes exchange messages among themselves to refine their locations.

We present an algorithm based on hop distance measurement and particle filtering [6] which can effectively achieve high location performance with relatively low communication overhead and computing complexity. Our algorithm is a distributed localization method devised to enable active cooperation between regular nodes and their neighbors in mobile sensor networks. Simulation results show that the performance of the new design is superior to other conventional localization algorithms. When compared to other algorithms, the proposed algorithm is more suitable for mobile WSNs in terms of communication costs and location accuracy.

This paper makes two major contributions to the localization problem in mobile WSNs. First, we present an attractive localization scheme with higher accuracy and lower communication cost than other state-of-the-art localization algorithms for mobile WSNs. Second, we propose a backoff-based broadcast algorithm which suppresses redundant broadcasts and reduces message overhead related to node positioning.

The rest of the paper is organized as follows. Section 2 provides an overview of the related works on localization techniques for mobile WSNs. Section 3 describes the details of the proposed algorithm. In Section 4, simulation results are reported and a comparative study of the localization performance is conducted. Finally, Section 5 presents our concluding remarks.

2 Prior Work

Localization has been a field of intensive research since the uprise of wireless *ad hoc* networks. There are many techniques in the literature proposed for static and mobile sensor networks. In this section, we discuss some of the related works in mobile WSNs and how they relate to and differ from our proposal.

Localization algorithms for mobile WSNs: Several works have been proposed for mobile WSNs. MCL [10] is designed for mobile sensor networks based on the sequential Monte Carlo method. A range-based version of MCL has also been proposed [5], which combines range-based and range-free location information to reduce the estimation error. Baggio *et al.* [1] improve the Monte Carlo localization scheme by reducing the sample prediction area. Their work, called MCB, draws valid samples faster and reduces the number of iterations necessary to fill the sample set. Computation overhead is reduced by this mechanism, but it still depends on specific parameters such as the fixed radio transmission range. Hsieh *et al.* [9] have proposed a localization algorithm which dynamically updates and makes use of reference information for cost-efficiency, and has a feasible solution for nodes receiving insufficient anchor information.

Different from the former work on localization algorithms for mobile WSNs, we study the cooperation between neighbor nodes that achieve high localization performance. Our scheme belongs to the range-free category of localization algorithms. Our approach differs from the above mentioned works in three significant ways: first, different from DV-Hop [12] and Gradient [11], which are based on simple flooding mechanism, we propose a backoff-based flooding to efficiently suppress redundant broadcasts and reduce communication overhead. For the hop distance measurement, a differential error correction scheme is devised to reduce the measurement error accumulated over multiple hops for the average hop distance. Second, our approach draws a more effective particle prediction area, borrowing some ideas from MCB, based on the positions of virtual anchor nodes, created with the active cooperation between regular nodes and its neighbors. Third, the sensor information is used to reduce even more the prediction area, thus reducing the estimation error of the non-anchor nodes. Furthermore, our approach does not require previous knowledge of the radio transmission range for filtering.

3 Algorithm Development

In this section, we provide the details of our proposed hop distance measurement and particle filter-based cooperative localization algorithm for mobile WSNs. For simplification purposes, we present our algorithm for the two-dimensional scenario. Intuitively, our work can be easily extended to the three-dimensional case. Our proposal can be divided into two main steps:

3.1 Hop Distance Estimation and Backoff-based Message Broadcast

In the first step, each anchor node broadcasts a beacon message throughout the network. The beacon message contains the anchor's location and a hop count with an initial value of zero. Each receiving node maintains the minimum hop count value per anchor node from all beacon messages it receives. Beacon messages with a higher hop count value from a given anchor node are ignored and discarded. On the other hand, valid beacon messages are forwarded with an incremented hop count after each hop. In this way, all nodes in the network can find their minimum hop counts to each anchor node. Nonetheless, this simple flooding method may result in excessive message overhead. To deal with this issue, we propose a backoff-based broadcasting mechanism described at the end of this section.

Conventional hop count localization methods require two separate flooding stages: (a) hop count accumulation and (b) average hop distance (*correction*). In comparison, our method combines the correction process with the hop count accumulation stage to reduce message transmissions. When the effective hop distance is calculated in the flooding process, the hop count is broadcasted simultaneously to all the nodes in the network. This approach effectively helps to reduce the number of transmitted messages, consequently reducing the network energy consumption and the time spent on computing a node's position.

Once an anchor node receives the hop count value from another anchor node, it estimates the average distance of one hop, namely *hop distance*, which will be used as a correction factor to be transmitted to the entire network. After receiving the hop distance, regular nodes multiply it by the hop count number to derive their estimated physical distances to the anchor nodes. For instance, the average hop distance between anchor nodes i and j is calculated as:

$$HopDistance_{i,j} = \frac{\sum_{i \neq j} \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2)}}{\sum_{i \neq j} h_{i,j}} \quad (1)$$

where (x_i, y_i) and (x_j, y_j) are the coordinates of anchors i and j, respectively, and $h_{i,j}$ is the number of hops between them. After hop distance estimation, it is straightforward to estimate the distance between two anchor nodes i and j as follows.

$$d_{est}^{i,j} = HopDistance_{i,j} \times h_{i,j}. \quad (2)$$

On the other hand, the actual distance, $d_{true}^{i,j}$, between anchor nodes i and j is given by

$$d_{true}^{i,j} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (3)$$

Following (2) and (3), the difference between the estimated and actual distances, denoted by $e^{i,j}$, can be expressed as

$$e^{i,j} = d_{est}^{i,j} - d_{true}^{i,j}, \quad (4)$$

which corresponds to the estimation error.

Here, we propose the usage of the differential error in (4) as a correction term to the original hop distance estimation presented in (1). The effective average hop distance, $EffHopDistance_{i,j}$, between anchor nodes i and j is defined as [4]:

$$EffHopDistance_{i,j} = HopDistance_{i,j} - \frac{e^{i,j} + e^{i,m}}{h_{i,j} + h_{i,m}}, \quad (5)$$

where $h_{i,m}$ is the number of hops in-between, and m is the second-closest anchor node to the unknown-position node k.

When anchor nodes i and m broadcast their average hop distances from j to regular node k, the related hop count value will also be broadcasted simultaneously to reduce the total number of transmitted messages. After obtaining both messages from i and m, k calculates $e^{i,j}$ and $e^{i,m}$ using eq. (4). Subsequently, the effective average hop distance can be calculated using the received beacon information via eq. (5). Based on eq. (5), k can compute its distance $d_{eff}^{k,j}$ to anchor node j:

$$d_{eff}^{k,j} = EffHopDistance_{k,j} \times h_{k,j} \quad (6)$$

If i is the closest anchor node to k, it is more accurate to estimate the distance between k and j by using $EffHopDistance_{i,j}$. From this principle, a generalization to any regular node is possible. That is, a given regular node can use the effective average hop distance obtained from its closest anchor node to calculate the distances to its neighbor nodes.

As shown in Fig. 1, simple flooding method suffers from high message overhead when used to provide location information to neighbor nodes, thus we propose a backoff-based broadcast mechanism to suppress redundant messages. A node may receive sequential beacon messages about the same anchor and each one of them leads to a smaller hop count. As a result, this node may need to forward it several times. Fig. 1 illustrates an anchor message propagation initiated by anchor node A. Due to the random exponential backoff of the MAC layer, node C may broadcast A's message before node B. Suppose that node D wins the next channel contention, and both nodes E and F set their minimum hop count to 3. It is possible that node B further fails the channel contention dispute with node E, and thus node E broadcasts an anchor message containing the wrong minimum hop count information before it notices the right value. Once the order is altered, the error is accumulated from each broadcast. Node E only needs to broadcast another anchor message after hearing the lower hop count information from node B, but nodes that are farther away from the anchor node may need to forward it several times to correct the accumulated error. We observe that nodes at the border of the transmission range of a sender have the widest coverage of its two-hop neighbors. Therefore, we should give these nodes the highest priority to broadcast. This has been extensively studied to reduce redundant broadcasts and collisions [2, 8, 13]. However, previous methods only alleviate the redundancy problem rather than completely suppressing it.

Giving receivers farther away from a sender higher priority to broadcast could solve the redundancy mentioned above. However, node G still suffers from redundant broadcast. Previous methods would allow node B to broadcast before node C. According to these methods, node E should broadcast before node D and it is likely that node E would also broadcast before node C. If node F wins the following channel contention, node G would get a wrong hop count of 3.

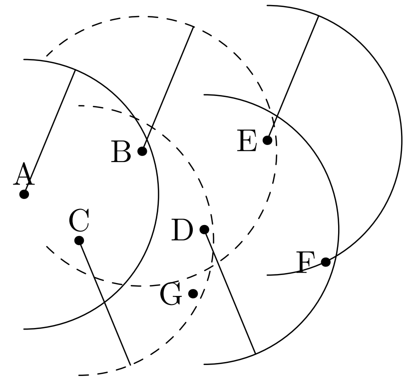

Figure 1: An example of anchor message propagation

Based on these observations, we propose a backoff-based flooding which works as follows. Nodes that are far away from the sender are expected to have a weaker received signal strength. Therefore, we set $ISS = Pr - RXThresh$ for each node, where ISS is the increase in signal strength, Pr denotes the received signal power, and $RXThresh$ denotes the minimum signal power for successfully decoding a packet. Each node defers its broadcast of the anchor message for $ISS * unit\_delay$ time units, where $unit\_delay$ should be large enough to cope with the random backoff of the MAC layer. Otherwise, node C may still broadcast before node B due to its winning of contention in channel access.

Supposing that node B is on the border of the transmission range of node A, it will broadcast immediately after hearing an anchor message from node A. Node E should calculate its sending deferment according to the same rule. In order to cope with the loop problem, node E should wait for an additional delay before initiating its transmission. In general, if a node is not a one-hop neighbor of an anchor node, it should wait for an additional period before forwarding the anchor's message. Suppose all nodes that need to broadcast are located within a ring characteristiced by two circles with radius $\alpha \times R$ and R, where α is determined by the node density obtained by overhearing. As an example, in Figure 1, C may not need to broadcast if there are many nodes located within the ring determined by two circles with radius $0.5R$ and R. The additional delay is then calculated with a Pr value that is expected at the $0.5R$ distance. Simulation results show that in our proposed backoff-based flooding, all nodes broadcast only once.

3.2 Positioning via Particle Filtering

For the positioning step, due to location uncertainty inserted by mobility, a node inserts virtual anchors to aid on constraining the localization error. Let (X_i, Y_i) be the coordinates of a virtual anchor node, where $i = \{1, 2, 3, \cdots, M\}$. Here, we select the midpoint of two one- or two-hop neighbor nodes as the position of virtual anchor node as shown in Fig. 2. Note that a node can estimate its distance to an anchor node and other regular nodes using the effective average hop distance and the number of hops in-between. These distances are used to constrain a small prediction area, from which particles are drawn and filtered.As shown in Fig. 2, the gray rectangle indicates the prediction area from where particles representing possible locations are extracted. The node's current velocity is also used to minimize the prediction area. The speed information can be collected from sensor data. For instance, sensor nodes are able to estimate their velocities using a three-dimensional accelerometer. As shown in Fig. 2, a regular

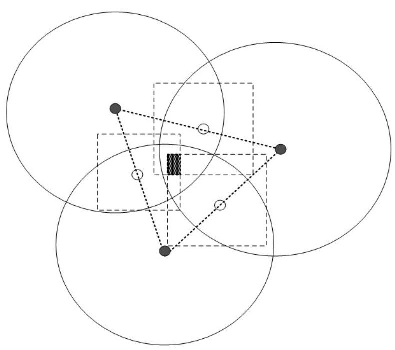

Figure 2: Building the prediction box.

node k located at the center of the intersection area is able to directly communicate with three one-hop neighbors. For each pair of one-hop neighbors, a virtual anchor node i is created and k builds a square of size $2 * d_{eff}^{k,i}$ centered at i, $d_{eff}^{k,i}$ being the estimated distance between k and i. Based on the current velocity of sensor node, the coordinates of the reduced sampling rectangle area are calculated as follows:

$$\begin{aligned} x_{min} &= \max(\max_{i=1}^{n}(X_i - d_{eff}^{k,i}), x_{t-1} - v_t) \\ x_{max} &= \max(\min_{i=1}^{n}(X_i + d_{eff}^{k,i}), x_{t-1} - v_t) \\ y_{min} &= \max(\max_{i=1}^{n}(Y_i - d_{eff}^{k,i}), y_{t-1} - v_t) \\ y_{max} &= \max(\min_{j=1}^{n}(Y_i + d_{eff}^{k,i}), y_{t-1} - v_t), \end{aligned}$$

where (X_{t-1}, Y_{t-1}) are the coordinates of particle l_{t-1}. When we consider two-hop neighbors, we replace $d_{eff}^{k,j}$ with the estimated distance between regular node k and its two-hop neighbor virtual anchor nodes. Based on Fig. 3, let the distance between anchor nodes A and B be d_2, the distance

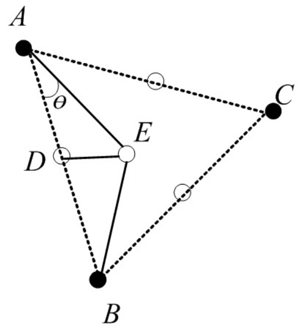

Figure 3: System model

between A and regular node E be d_1 and the distance between regular node E and anchor node B be d_3. The distance between the virtual anchor node D and regular node E is defined as d_0. Finally, θ is the angle between line segments $\overline{AB}$ and $\overline{AE}$, which can be calculated from the cosine rule. After obtaining the values of d_1, d_2 and d_3 from triangle $\triangle ABE$, we have:

$$\cos\theta = \frac{d_2^2 + d_1^2 - d_3^2}{2d_1 d_2}. \tag{7}$$

Similarly, in triangle $\triangle ADE$, we have:

$$\cos\theta = \frac{(d_2^2/4) + d_1^2 - d_0^2}{d_1 d_2}. \tag{8}$$

Thus, we can calculate the value of d_0 based on eqs. (7) and (8) as follows:

$$d_0 = \frac{\sqrt{2d_1^2 + 2d_3^2 - d_2^2}}{2}. \tag{9}$$

Prediction Phase If the anchor's distance and speed constraint are disjointed, for example at initialization time, the node excludes the speed constraint when calculating the prediction area. The prediction area helps to draw valid particles in the prediction phase. The probability of a given current location based on a previous estimate is given by the uniform distribution:

$$p(L_t|L_{t-1}) \\ = \begin{cases} 1 & \text{if } x_{min} \le x_t \le x_{max} \cap y_{min} \le y_u \le y_{max} \\ 0 & \text{otherwise,} \end{cases} \tag{10}$$

where (x_t, y_t) are the coordinates of particle l_t.

Filtering Phase

In the prediction phase, each node generates a set of uniformly distributed random values inside the prediction area. If the drawn particles are located inside the communication range of every neighbor, the node saves this position for the final estimation. Otherwise, the node discards it and repeats the prediction phase. The entire filtering phase in a node can be represented as follows:

$$filter(l_t) = \forall a \in M, d(l_t, a) \leq r + v_t \cap$$
$$\cap \forall b \in M, r - v_t \leq d(l_t, b) \leq 2r + v_t,$$

where l_t is the concerned particle, M and N are the sets of one-hop and two-hop neighbors, respectively, r is the node's radio range, $d(l_t, a)$ is the distance between particle l_t and neighbor a, and v_t is the current velocity of a neighbor node. After obtaining a sufficient number of valid particles, the final location estimate is calculated as the average of the particle set.

4 Performance Evaluation

The general performance of our algorithm was obtained through simulation, using an adapted version of the simulator provided by the authors of MCL [10]. We evaluated the effectiveness of the backoff-based broadcast mechanism and the location accuracy of our proposal in terms of node speed, node density, anchor density, and communication irregularities, and compared it to four other algorithms: Centroid, Gradient, MCL, and MCB.

For the following results, we consider a topology of 320 nodes, placed in a random uniform manner over an area of 500×500 square units. Unless specified otherwise, we consider an anchor density of 1, node density of 10, and radio propagation using the unit disk model. These parameters were also varied and their evaluation can be seen in the next subsections. The radio range was set to 50 units and the maximum speed to 50 units per time interval. Anchor and regular nodes move using a modified version of the random waypoint model, with no pauses between intervals in order to prevent the average speed decay problem [3].

In the case of MCL-based algorithms (MCL, MCB, and our approach), we used a fixed sampling set of 50 units. Just like [10], we have verified that this is the best cost-benefit solution. While maintaining more samples improves accuracy, it also increases computational overhead and memory requirements. For each parameter, a set of 30 simulation runs was performed and results were averaged.

4.1 Evaluation of the Broadcast Mechanism

We investigate the effectiveness of the backoff-based broadcast mechanism which is adopted in step 1 of our proposal. Fig. 4 shows the number of beacon messages an anchor node needs to broadcast using simple flooding and our proposed backoff-based flooding to guarantee node localization. Nodes are ordered in increasing distance to the anchor node located at the left bottom corner. The minimum hop count of the farthest node to the anchor is 9. We can observe that some nodes have to broadcast the same related message up to *five* times in simple flooding. Generally, the farther away a node is from the anchor node, the more messages it needs to broadcast to correct the accumulated error. In contrast, in our proposed backoff-based flooding all nodes obtain their correct minimum hop counts with an overhead of one message per node. Therefore, our design effectively reduces the number of redundant messages. In another experiment, we distribute 200 nodes in a

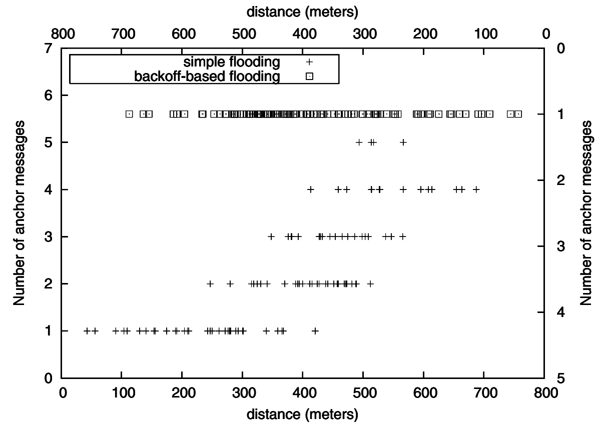

Figure 4: Number of anchor messages transmitted per node

$500m \times 500m$ field and vary the transmission range from 100 to 50 meters. The average total number of anchor messages used by all nodes is presented in Fig. 5. In general, as the transmission range decreases, the accumulated error increases. Distant nodes need to rebroadcast more times. A transmission range less than 60 meters cannot ensure connectivity and thus the overhead decreases suddenly. Fig. 5 shows that our algorithm scales well to network size and substantially suppresses redundant broadcasts.

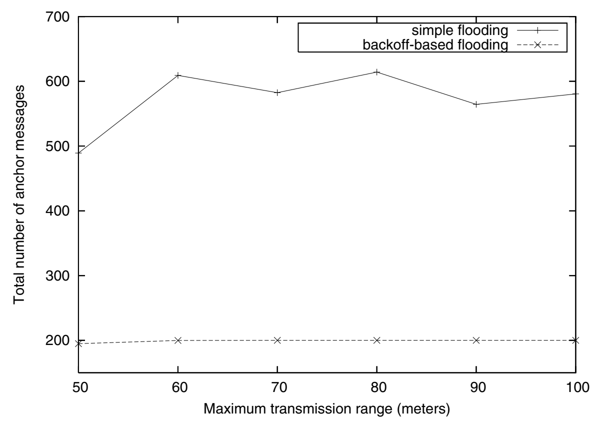

Figure 5: Average number of anchor messages used by all nodes

4.2 Convergence Time

The second simulation analysis concerns the location accuracy convergence. As it can be seen from Fig. 6, the initial sharp decline on the estimation error curves of the MCL-based algorithms is due to new location announcements from anchors being incorporated into the sequential Monte Carlo process. This is followed by a stable phase declaring the balance between updates to the posterior location distribution and uncertainty introduced by mobility. MCB and our proposal give similar performance, though the latter achieves slightly better accuracy. This is due to a smaller sampling area, which effectively speeds up the collection of valid samples, and also due to the optimized filtering phase, which uses actual sensor data for constraining the filtering area, opposed to a theoretical maximum velocity boundary. Centroid and Gradient do not exploit position history information, so their accuracy does not change over time.

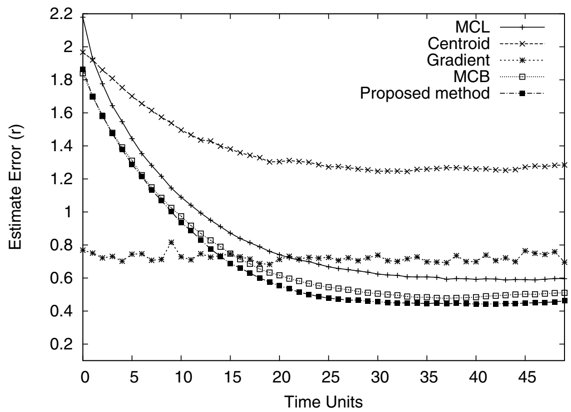

Figure 6: Accuracy comparison

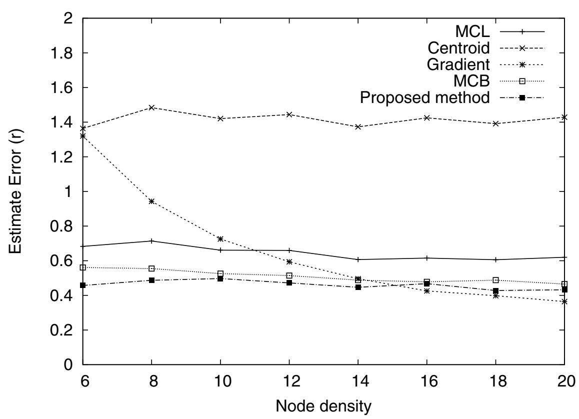

Figure 7: Impact of node density

4.3 Node density

Fig. 7 shows the impact of node density variation on the accuracy of each algorithm while the anchor density is kept constant. Centroid and the MCL-based algorithms are slightly affected by node density. In the case of the MCL-based algorithms, even when there are only two-hop anchor neighbors, enough samples can be filtered to obtain a decent location estimate. For Gradient, a higher density of nodes reduces the estimation error of the hop size leading to a higher accuracy as well. In our simulation experiments, Gradient performs best when the node density is over 16.

4.4 Anchor density

Increasing anchor density facilitates the retrieval of location information by regular nodes. Fig. 8 shows the average estimate error of different localization algorithms when the anchor density varies. We can see that all MCL-based algorithms take advantage of a higher anchor density, as more position announcements will be available for filtering. The same happens to Centroid. In the case of Gradient, the impact over accuracy is negligible as information from the whole network is used independent of the number of anchors available. Our proposal performs better than the other algorithms for reasons explained above (smaller sampling area and better filtering technique).

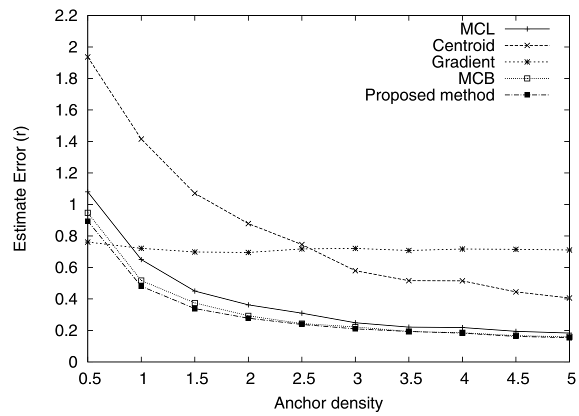

Figure 8: Impact of anchor density

4.5 Node speed

As we can see from Fig. 9, for all the MCL-based algorithms the maximum speed variation does not affect much the location accuracy. At higher speeds, nodes hear new anchors more often which add new information to the filtering phase, keeping the sample set updated. On the other hand, as larger distances are traversed at each time unit, the sampling area becomes larger, decreasing the algorithm's accuracy. In the case of Gradient, it is assumed that all network information is provided to every node at each time unit, a condition that is too optimistic for a real network. Therefore, node speed also does not affect its performance. Still, it can be seen that its estimation error is higher than the MCL-based algorithms. Node speed also does not affect Centroid.

4.6 Communication Irregularities

Although the unit disk model is a simple and easy-to-use communication framework, it does not reflect the reality of wireless communication. The measured signal strength of radios can vary substantially with environmental conditions and antenna irregularities. This also affects the localization accuracy. Although many complex studies on radio propagation models have been proposed, we decided to use the Degree of Irregularity (DoI) [7] to model transmission and

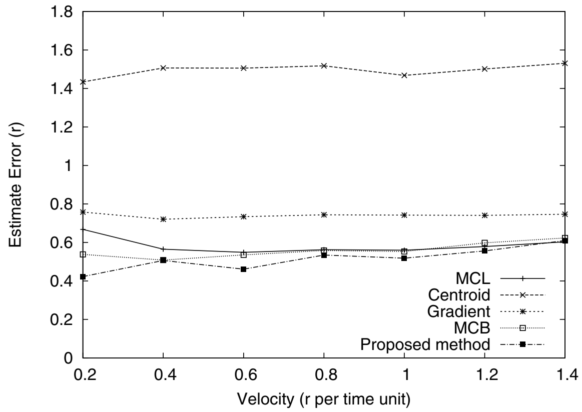

Figure 9: Impact of node speed

reception adversities for comparison purposes. The DoI parameter defines the radio signal strength variation on each direction of radio propagation. As its value increases, multihop communication is affected and the localization error of every algorithm increases, except for Centroid, as shown in Fig. 10. We can see from the results that our proposed algorithm degrades more gracefully. Despite the message exchange hardships, our proposal is still able to filter a larger number of valid samples than the other MCL-based algorithms.

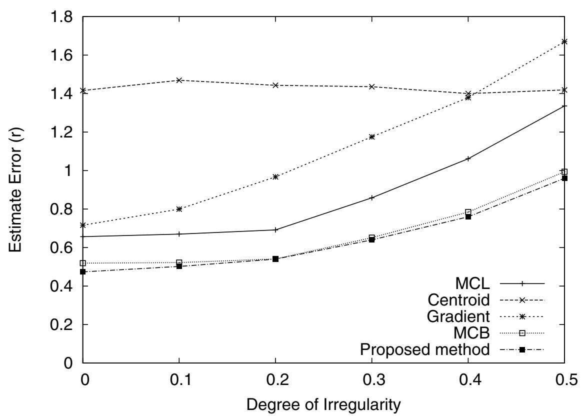

Figure 10: Impact of communication irregularities

5 Conclusion

We present a distributed hop distance measurement and particle filter-based cooperative localization algorithm for mobile WSNs. Our proposal is scalable, robust, and self-adaptive to the dynamics of a mobile sensor network. Our proposed algorithm can reduce the hop distance estimation error accumulated over multiple hops by using a differential error correction scheme. In order to efficiently suppress redundant broadcasts and to reduce communication overhead, a backoff-based broadcast mechanism is proposed. It also improves localization performance by including particle filtering technology. Simulation results show that the proposed algorithm achieves better performance than other state-of-the-art algorithms. Influence of physical obstacles over the communication model remains to be explored as future work.

Acknowldegment

This work is supported by CREST "Advanced Integrated Sensing Technology" project of Japan Science and Technology Agency.

References

[1] A. Baggio and K. Langendoen. Monte Carlo localization for mobile wireless sensor networks. *Ad Hoc Netw.*, 6(5):718–733, 2008.

[2] S. Biswas and R. Morris. ExOR: opportunistic multi-hop routing for wireless networks. *SIGCOMM Comput. Commun. Rev.*, 35(4):133–144, 2005.

[3] T. Camp, J. Boleng, and V. Davies. A survey of mobility models for ad hoc network research. *Wireless Communications & Mobile Computing (WCMC): Special Issue on Mobile Ad Hoc Networking: Research, Trends and Applications*, 2:483–502, 2002.

[4] H. Chen, K. Sezaki, P. Deng, and H. C. So. An improved dv-hop localization algorithm with reduced location error for wsns. *IEICE Transactions on Fundamentals*, E91-A:2232–2236, Aug. 2008.

[5] B. Dil, S. Dulman, and P. Havinga. Range-based localization in mobile sensor networks. *Lecture Notes in Computer Science*, 3868:164–179, 2006.

[6] A. Doucet, N. Defreitas, and N. Gordon. *Sequential Monte Carlo Methods in Practice*. Springer, 2001.

[7] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. Abdelzaher. Range-free localization schemes for large scale sensor networks. In *Proc. of ACM MobiCom*, pages 81–95, New York, NY, USA, 2003.

[8] M. Heissenbüttel, T. Braun, T. Bernoulli, and M. Wälchli. BLR: beacon-less routing algorithm for mobile ad-hoc networks. *Elsevier's Computer Communications Journal (Special Issue)*, 27:1076–1086, 2004.

[9] Y.-L. Hsieh and K. Wang. Efficient localization in mobile wireless sensor networks. In *SUTC '06: Proc. of IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing*, pages 292–297, Washington, DC, USA, 2006.

[10] L. Hu and D. Evans. Localization for mobile sensor networks. In *Proc. of ACM MobiCom*, pages 45–57, New York, NY, USA, 2004.

[11] R. Nagpal, H. Shrobe, and J. Bachrach. Organizing a global coordinate system from local information on an ad hoc sensor network. *Lecture Notes in Computer Science*, 2634:333–348, 2003.

[12] D. Niculescu and B. Nath. Ad hoc positioning system (APS). In *Proc. of IEEE GLOBECOM*, volume 5, pages 2926–2931, Nov. 2001.

[13] F. Ye, A. Chen, S. Lu, and L. Zhang. A scalable solution to minimum cost forwarding in large sensor networks. In *IC-CNC: Proc. of the 10th International Conference on Computer Communications and Networks*, pages 304–309, 2001.