2009 29th IEEE International Conference on Distributed Computing Systems

EUL: an Efficient and Universal Localization Method for Wireless Sensor Network

Wei Xi*, Jizhong Zhao*, Xue Liu†, Xiang-Yang Li‡, Yong Qi*
Dept. of Comp. Sci. and Tech., Xi'an Jiaotong Univ., huiwei@live.com
†*School of Computer Science, McGill University, xueliu@cs.mcgill.ca*
‡*Dept. of Computer Science, Illinois Inst. of Tech., xli@cs.iit.edu*

Abstract

Localization is a crucial service for various applications in wireless sensor networks (WSNs). Although most researches assume stationary nodes, sensor mobility can enrich the application scenarios. Existing dynamic localization approaches require high seed density or incur a large communication overhead. In order to address these problems, we propose an efficient rang-free localization algorithm, EUL, which utilizes the relationship between neighboring nodes to estimate their possible location boundaries. Our algorithm not only allows all the nodes to remain static or move freely but also reduces the dependence on seeds, which achieves a uniform energy distribution to address the excessive energy drain around seeds and lengthen the network lifetime. We have evaluated EUL together with other major dynamic localization approaches. Simulation results show that EUL outperforms existing approaches in terms of accuracy under many different mobility conditions.

1. Introduction

The ability to automatically determine sensor locations is fundamental to various applications in wireless sensor networks (WSNs). Location information is often heavily used in these applications, such as environment monitoring [1, 2], event detection [3], context aware [4], target tracking [5], packets delivery using geometric-aware routing [6], etc. In order to get an accurate location, one solution is to equip each sensor with localization-hardware (e.g., GPS [7]). However, this is not viable in many applications due to the restrictions imposed by factors such as cost, energy consumption, and expansibility. A widely-adopted solution is to deploy sensor network with some location-aware sensors (called seeds) which know their locations and can provide this information to other ordinary nodes to assist them in estimating their locations.

Existing researches for sensor localization mainly fall into two categories: range-based approaches and range-free approaches [8, 9]. Range-based approaches measure the distance or direction among the ordinary nodes (or target nodes) and seeds to compute the position of each node. Range-free approaches locate nodes using network connectivity information instead of accurate distance measurements between nodes.

Recently, mobile sensor networks are becoming an attractive design paradigm since it not only extends the range of applications but also improves the network coverage. We can classify the current localization algorithms into static localization and dynamic localization according to whether sensors are mobile or not.

The existing static localization approaches usually suffer from following problems:

- Requirement for special hardware. Range-based approaches need to equip target sensor nodes with costly devices to measure the distance or direction, such as time of arrival (TOA) [7], angle of arrival (AOA) [10], or other ranging information. Moreover, these devices may increase energy consumption when operating.
- Requirement for high sensor density. Some range-based approaches (e.g., ultrasound-based TDOA [11]) need to deploy enough number of nodes due to limited effective range. Range-free approaches (e.g., APIT [8]) depend on high density of seeds to improve location accuracy.

However, sensor mobility brings the new challenges to large-scale sensor networks. Dynamic techniques such as hop-counting positioning techniques [12] need to regularly flood location information to a local area from seeds and update per-node's hop-count at each time instant. The iteration process causes a high communication cost and an excessive energy drain. Other approaches including MCL [13] and MSL [14], based on particle filters combined with probabilistic models of robot perception and motion, outperform previous approaches in terms of both accuracy and computational efficiency. Those advantages, however, come along with an implicit assumption that each node must have a seed within two-hop distance away from itself.

The above limitations call for new localization approaches that are efficient in computation and energy cost as well as universal for both static and dynamic sensor networks. In this paper we propose EUL, an efficient and universal localization (EUL) approach for WSNs. One novel idea

1063-6927/09 $25.00 © 2009 IEEE
DOI 10.1109/ICDCS.2009.78

of EUL is that nodes utilize relative motion to estimate their possible range of locations rather than only using the information provided by seeds. The contributions of EUL are two-fold:

- We use a hop-based algorithm to estimate all the nodes initial locations. This algorithm efficiently determines an approximate location of each node and avoids the situation that a sensor node never knows its location which is far away from seeds.
- We present a collaborative neighbor update algorithm such that a node collaborates with its one-hop neighbors (including both target nodes and seeds) to filter impossible locations which cannot satisfy constraints of its neighbors. This method reduces the dependence on seeds, avoids maintaining a large number of samples, achieves a uniform energy distribution, and increases convergence rate of estimate error.

The rest of the paper is organized as follows: In Section 2, we summarize related work on localization algorithms. In Section 3, we present our new localization algorithm EUL. We analyze its' performances in Section 4. In Section 5, we evaluate the proposed scheme through comprehensive simulation studies and compare it with other localization techniques. We conclude the paper in Section 6.

2. Related Work

Extensive approaches have been proposed to locate sensor nodes in WSNs. In this paper, we classify them into two categories: static localization and dynamic localization. Most of current research focuses on static sensor networks whereas a few on dynamic networks. In this section, we provide a brief discussion of existing localization approaches and contrast them with EUL.

2.1. Static Localization

The localization approaches in static sensor networks can be further divided into two classes: range-based localization and range-free localization.

Range-based localization approaches leverage the information such as Time of Arrival (TOA) [7], Time Difference of Arrival (TDOA) [11] and Angle of Arrival (AOA) [10]. They are based on point-to-point distance or direction measurement to locate each node. These solutions are less cost-effective for large-scale sensor network. Additionally, constraints on signal attenuation (e.g. ultrasound signals) result in limited effective range and high deployment density. Some other rang-based approaches based on information such as Received Signal Strength (RSS) [15, 16] are low-cost, however, they are easily disturbed by background noise.

Range-free localization approaches are cost effective. Shang et al. [17] proposed MDS-MAP which is a centralized technique using MDS (MultiDimensional Scaling). He et al. proposed APIT [8] based on PIT (Point-In-Triangulation). Nodes use neighbor information and exchange it using beaconing to estimate movement in perfect PIT test. Event-driven localization [18] is a special kind of range-free approach which makes use of events generated and spread across the sensor network to locate each node.

2.2. Dynamic Localization

No approaches mentioned above consider the scenario that sensors move freely. Bergamo et al. [19] are the previous research to perform localization in mobile sensor network. They assumed that network includes two fixed seeds and mobile nodes can accurately measure the received power strength. The location is estimated by means of triangulation.

Tilak et al. [20] showed that applications of mobile sensor network are extensive. In addition, they investigate the tradeoff between energy and accuracy: a frequent positioning reduces error but increases energy consumption.

Hu et al. [13] used Monte Carlo Localization (MCL) to estimate node location. Each node maintains samples which are probabilistic distribution of its location. A node estimates new possible locations based on previous locations and maximum velocity, and then removes samples that are inconsistent with new observations. There is a limiting condition that node must be one or two-hop neighbor of certain seeds.

DRL [12] proposed by Hsieh et al. is an improved algorithm of DV-Hop. Each mobile node locates itself by regular triangulation and other reference information. Then seeds update the parameters provided for other nodes, which increases network load. In addition, if nodes move in a straight line or are not in overlay area of seeds, there will be massive calculation.

2.3. Contrast with EUL

There are some drawbacks of existing localization approaches. The first is that most of them are not universal. They usually work well either in static or mobile sensor network. The second is that most of them are not efficient. They overly depend on the location information of seeds. The underlying approaches must be invoked repeatedly to update location information to maintain the positioning accuracy. It is crucial to determine sensor locations accurately and efficiently in terms of a low communication cost as well.

3. The Design of EUL

In this section, we present the details of EUL design. We utilize the relationship between neighboring nodes to update their locations and then filter impossible locations which are out of neighbors' radio range. Our algorithm mainly consists of two phases:

- Initial Location Estimation (ILE): Each node estimates its initial location by triangulation.
- Collaborative Neighbor Update (CNU): Each node updates location boundary according to previous location and neighbors information.

Table 1: Overview of EUL algorithm

Phase	**Description**
ILE	A node estimates its initial location P_0 according to the hop-count to seeds.
CNU	**Prediction**: A node predicts possible locations set S_t based on S_{t-1}. **Correction**: A node collaborates with its neighbors to filter S_t.

Table 1 shows an overview of two phases of EUL algorithm. Let t be the discrete time units, S_t denote the predicted node's possible locations set at time unit t.

In the following, we discuss the design of the two phases in details.

3.1. Initial Location Estimation

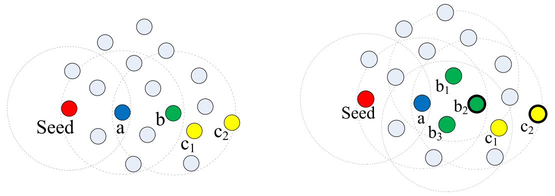
ILE utilizes a hop-counting based technique to make all nodes know their initial locations. This is the foundation of the next phase. We relax assumption in network topologies that nodes are uniform distribution. There are three stages in ILE.

Algorithm 1 Table Update and Forwarding

```
Repeat
  Each node receives and forwards message;
  If (new $h_i$ < stored $h_i$)
    Update $h_i$ to the new value;
    $t_i = 1$;
    $TTL--$;
    $h_i$++;
  Else If (new $h_i$ = stored $h_i$)
    $t_i$ ++;
    $TTL--$;
    $h_i$++;
  Else
    Stop forwarding;
Until $TTL = 0$;
```

First, seeds flood their location information to the whole network. Each node maintains a table $\{X_i, Y_i, h_i, t_i, TTL\}$ and forwards this information except t_i to its neighbors as algorithm 1 shows. Let i be the seed id, (X_i, Y_i) denote the position coordinates of seed i, h_i denote the hop-count to seed i, t_i denote the number of times with the same h_i received by a node, and TTL be Time-To-Live.

Table update algorithm ensures h_i with minimum value and reduces network load by stopping forwarding, but there may be a large estimate error in last hop. Figure 1(a) shows the difference between last hop-distance with the same hop-count. Node c_1 and c_2 are both three hops to seed but c_1 is close to two hops node. We propose a solution that with the same h_i, a node closer to seed has lager t_i as Figure 1(b) shows. Node c_1 receive a message with $h_i = 3$ more than three times, while c_2 receive it just once from b_2. In order to reduce estimate error at last hop, h_i is corrected to $h_i - 1 + 1/t_i$.



(a) The difference of last hop-distance

(b) The relationship between last hop-distance and the number of previous-hop nodes

Figure 1: Hop-count estimate

Second, each seed receives messages from other seeds to estimate an average hop-distance which is then forwarded to neighbors. A seed (X_i, Y_i) computes the average hop-distance as follows:

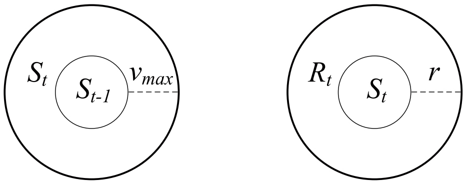
$$d_i = \frac{\sum \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}}{\sum h_i}, \quad i \neq j, \ \text{all seeds } j. \tag{1}$$

From above formula, we can see that different seeds compute different average hop-distance, so which value should a node reserve becomes a crucial problem. Average hop-distance distributed by uncontrolled flooding increases network load, while setting TTL to restrict announcement flooding results in announcement not available for some nodes far away from seed. We propose to let seeds transmit the average hop-distance only to their neighbors. A node reserves and forwards the average hop-distance to its neighbors. Once the node has received the average hop-distance value from more than three different seeds, it stops receiving and forwarding any other packets. By this means, a node can receive the average hop-distance value from its nearest neighbor seeds.

In the third stage, each node has received both hop-count and hop-distance from more than two seeds. A node computes the product of hop-count and hop-distance as the range to a seed. These values are plugged into the triangulation procedure similar with the one used in GPS position calculation [7] to compute nodes' positions in the plane.

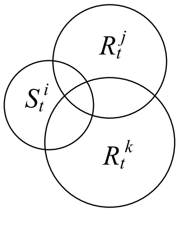
$$\Delta\rho = J\Delta r, \tag{2}$$

where ρ is the respective ranges to seeds, $\Delta\rho$ is approximated linearly using Taylor expansion, J is the unit vector of ρ and



(a) The prediction of a node's location range

(b) The prediction of a node's radio range



(c) The correction of a node's location range

Figure 2: Collaborative neighbors update

Δr is the position correction.

The iteration process applies Δr to the current position estimate and it doesn't stop until Δr is below a chosen threshold.

3.2. Collaborative Neighbors Update

CNU is the core of our localization approach, which achieves efficient localization by making use of mobility. In this phase, all nodes move freely, which causes a constant change in network connectivity.

We assume that time is discrete. Sensors are moving with variable velocity that moving speed and direction are both random variables. With regard to its current moving status, a node knows nothing except the maximum speed v_{max}. For the ease of discussion, in the following, we assume that each sensor has the same ideal radio range, r. We are interested in obtaining the boundary of nodes' possible locations. All nodes need to estimate their locations according to previous locations, movement velocity and information of neighbors in each time unit. There are two main stages in this phase: prediction and correction.

In the prediction stage, each mobile node has to predict boundary of its possible locations next time instance based on current locations set and v_{max}. Initially a node predicts possible locations set S_1 in a circle whose center is its initial location estimated in ILE and radius is v_{max}. In the subsequent time unit, the boundary of S_t and S_{t-1} are concentric circles, and radius of S_t is v_{max} longer than S_{t-1}. Figure 2(a) shows the relation between S_t and S_{t-1}.

For static networks or slowly mobile networks, the radius of S_1 is close to zero and there is no circle to represent boundary of a node's possible locations. In order to address this problem, we use $v_{max} + \alpha$ instead of v_{max}. Experiment results show that selecting α equal to $0.1r$ gives good performance.

The prediction algorithm may not work well with the passage of time because the range of S_t will become bigger, which leads to critical loss of accuracy. Hence, there should be a step to reduce the range of S_t after prediction stage in each time instance.

The correction stage is a process that each node filters impossible locations collaborating with neighbors. A node regularly detects neighbors falling into three categories:

- **Former Neighbor** – Sensor was within range of the node previously, rather than currently.
- **New Neighbor** – Sensor is within range of the node currently, rather than previously.
- **Constant Neighbor** – Sensor keeps up with the node in both previous and current time quanta.

A sensor node only communicates with its current neighbors, so new neighbors and constant neighbors provide more useful location information for the node. We utilize this information to filter impossible locations which are out of neighbors' radio range. If a node has some neighbors, it should be in the overlap of these neighbors' radio range including both new neighbors and constant neighbors. Thus sensors' possible radio range should be estimated before filtering. Figure 2(b) shows relationship between a node's possible radio range R_t and S_t. The node estimates R_t according to its S_t and r similar to location range prediction process.

Each node computes overlap region between its possible locations set S_t^i and neighbors' possible radio range R_t^j as its new locations set S_t^i as Figure 2(c) shows. That is,

$$S_t^i = S_t^i \cap R_t^j, \text{all current neighbors j.} \tag{3}$$

There is a knotty problem that the boundary of a node's new locations set S_t^i is an irregular figure. It is hard to denote when number of neighbors is large, and also increases the computational complexity for next time unit. As time goes on, the complexity will be higher. We use a circle as an approximation of new boundary to address this problem, which will be analyzed in section 4.3 intensively.

4. Analysis

We have described the design of EUL for both static and mobile sensor networks. In this section, we provide the analysis of initialization, error correction, regularization and overhead of EUL.

4.1. Initialization

Initialization in EUL uses a hop-count based technique which avoids the dependence on a large number of seeds.

However, it requires node evenly distributed to insure estimated precision in the network. We consider that artificially deploying each node to a certain position is impossible, but uniformly casting nodes to monitoring area is feasible in deployment phase.

In order to improve efficiency of seeds flooding process, seeds are generally deployed following boundary of monitoring area. This method prolongs the lifetime of wireless sensor networks by reducing the number of seeds participating in the flood process.

4.2. Error Correction

We focus on analysis of problems in CNU caused by estimate error. As estimate error in the ILE phase and CNU phase, there may be two types of problem in filter stage:

- A nodes possible locations range and its neighbors possible radio range have no overlap region.
- A nodes two neighborspossible radio ranges have no overlap region.

Either type of the problem results in the inexistence of the node's correction locations. Both types of problems satisfy the following inequality:

$$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} > r_1 + r_2, \tag{4}$$

where (x, y) and r denote the center and radius of two circles respectively. This kind of error should be corrected as soon as it is discovered.

Algorithm 2 Error Detection and Correction

```
Repeat
  A node detect its neighbors;
  If (node $i$ has a neighbor seed $k$)
    If ($R_t^i \cap R_t^k = \varnothing$)
      Correct location of $R_t^i$;
    For (all neighbors j)
      If ($R_t^j \cap R_t^k = \varnothing$)
        Correct location of $R_t^j$;
Until the node detects all its neighbors;
Compute an intersection as new locations set;
```

Algorithm 2 depicts the computation architecture for error detection and correction. The correction location is computed with a modified formula of define proportion and separated points:

$$\begin{cases} x = x_1 + \frac{(x_2 - x_1)(r_1 + r_2 - \alpha)}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1))^2}}, \\ y = y_1 + \frac{(y_2 - y_1)(r_1 + r_2 - \alpha)}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1))^2}}, \end{cases} \tag{5}$$

where (x_1, y_1) and r_1 are the location coordinates and radio range of seed, (x_2, y_2) and r_2 are the central coordinates and radius of the node's location range respectively. This formula is to translate the false coordinates of the node location into radio range of its neighboring seed. Then each node computes a non-null intersection as new locations set.

4.3. Regularization

As mentioned in section 3.3, the new location boundary of each node needs to be regularized to reduce computational complexity in following time unit. Approximation by maximum inscribed circle (MIC) [21] is an effective method to regularize boundary. The center of MIC is the intersection point of circles which have common centers with S_t or R_t. That is,

$$\begin{cases} (x_1 - a_1)^2 + (x_2 - b_1)^2 - (x_3 - c_1)^2 = 0, \\ (x_1 - a_2)^2 + (x_2 - b_2)^2 - (x_3 - c_2)^2 = 0, \\ \vdots \\ (x_1 - a_n)^2 + (x_2 - b_n)^2 - (x_3 - c_n)^2 = 0, \end{cases} \tag{6}$$

where (a, b) and c denote center and radius of S_t or R_t, (x_1, x_2) and x_3 are center and radius of MIC respectively. When n is greater than three, the equations set is a non-linear overdetermined set of equations which require non-linear optimization methods to approximate. Non-linear least squares [22] is used to fit m observations with a model that is non-linear in n unknown quantities ($m > n$). We use the Gauss-Newton method which is based on a linear approximation of the objective function to approximate an optimum solution, where the Jacobian, J, is a function of constants:

$$J(x_k)^T J(x_k) + \nabla f(x_k) = 0. \tag{7}$$

4.4. Overhead Analysis

We measure communication overhead as the number of message sent in each localization process [13]. EUL uses information from one-hop neighbors including both seeds and nodes, so it has higher communication cost than MCL but has lower cost than MSL*[14].

In EUL, since each sensor communicates with neighbors once per time unit, the communication cost is proportional to $(m + n) * N/2$, where n is the number of nodes, m is the number of seeds, and N is the node density described in section 5.1. In MSL*, the communication cost is proportional to $n * s + m$, where s is the number of samples maintained by each node. Under normal conditions, m is a small value and N is far less than s, so communication cost in EUL is lower than MSL*. In MSL, the cost is proportional to $n+m$. In MCL, communication cost is proportional to m.

As for energy consumption distribution, EUL is more uniform than MCL, MSL, and all the static approaches. In most of the previous localization approaches, communicating with seeds directly or indirectly is the only one way for a node to get enough information and determine its location, which would lead to high energy consumption around the seed. In EUL, each node is on an equal footing, which can efficiently avoid energy hot spots around seeds and lengthen the network lifetime.

5. Performance Evaluation

We carried out extensive simulation studies to evaluate the performance and efficacy of our proposed EUL approach. In this section, we report the results. We implemented EUL based on the simulator that Hu and Evans [13] designed for MCL in Java. We evaluate the location accuracy of EUL in terms of node speed, node density and seed density to compare with some major dynamic localization techniques described in section 2. First, in section 5.1, we define the performance metrics and describe simulation setup.

5.1. Performance Metrics and Simulation Setup

We focus on investigating the accuracy of location estimation which is the key metric for evaluating a localization algorithm.

EUL estimates a node's possible locations range instead of an exact location. We compute the error of location estimation for a node as follows:

$$Error_i = \frac{1}{2}(Error_{max} + Error_{min}), \tag{8}$$

where E_{max} and E_{min} are the maximum and minimum distance between a node' real location L and prediction locations set S. A line across L and center of S intersects boundary of S at two points. E_{max} is the distance between farther point and L. E_{min} is zero if L is in S, otherwise E_{min} is the distance between nearer point and L. The localization error of our approach is computed as follows, where n is the number of nodes:

$$Error = \frac{1}{n}\sum_{i=1}^{n} Error_i. \tag{9}$$

For our experiments, sensor nodes are uniform distributed in a $500m * 500m$ square region at first. After initialization phase, sensor nodes are randomly moved in the region. We set radio range $r = 50m$, $n_d = 10$ and $s_d = 1$ in our experiments. The network and node parameters are similar to those used in [13], described as follows:

- Moving speed (v), the moving distance per time unit. Each sensor's moving speed is randomly chosen from $[0, v_{max}]$.
- Node density (n_d), the average number of one hop neighboring nodes of a sensor.
- Seed density (s_d), the average number of one hop neighboring seeds of a sensor.

We use a modified version of random waypoint mobility model [13, 23] for both nodes and seeds adopted by Hu and Evans. We assume that nodes have no knowledge about their velocity and direction, but know the maximum moving speed v_{max}. In this model, a node randomly varies speed and set pause time during each movement to prevent decay of average speeds in the random waypoint model [23, 24].

Without any distance measuring equipment, a node cannot know the precise distance to the sensor, but can judge whether it is within radio range of any other sensor or not. We compare EUL with DRL [12], MCL [13], and MSL [14] algorithm under various network configurations.

In the following, we report these results.

5.2. Accuracy

Location accuracy is the most important evaluation criterion for a localization algorithm. The estimate error of EUL depends on the moving speed of sensors. Mobility of sensors helps a node receive more location announcements from seeds. Figure 3 shows the error in location estimation of EUL at different moving speeds. We can observe, in each situation the estimate error decrease.

For mobile sensor network, the faster sensors move within certain limits, the lower level the error converges to. That is because mobile nodes have more chances to communicate with seed to obtain precise location information to filter impossible locations. Movement also offers nodes more opportunities to mutually correct possible locations with neighbors, and thus estimate error converges to a low level.

Figure 4 shows the accuracy comparisons of EUL, MSL, DRL and MCL at $v_{max} = 0.2r$. The error increases for DRL with time going on, because stochastic motion results in unevenly nodes distributed which is not conducive to work with hop-based technique. EUL takes advantage of DRL and MCL in the location process. Initial location estimate of EUL is conducive to efficient estimating sensor locations. This phase let each node get approximate initial location information, which reduces the initial error efficiently and provides a foundation for neighbors collaboratively update. CNU makes full use of location information provided by neighbors including both nodes and seeds, which accelerates the convergence of error and avoids high communication overhead caused by multiple-hop transmission.

Figure 5 shows the accuracy comparison for static nodes. Since MCL does not work for static network, we compare the accuracy of EUL with DRL, MSL and MSL*. The error of DRL stays the same, while EUL, MSL and MSL* has a convergent process till it reaches a stable state. The initial error of EUL is the lowest, and convergence of error is the rapidest as well. That is because initialization algorithm uses a hop-based technique which works well for static localization, and the existence of parameter α can aid update process to go on smoothly. After some time units, the error of EUL reaches a stable state. That is because the neighbors of each sensor are constant. When available local position information is used up, the estimation error reaches flat.

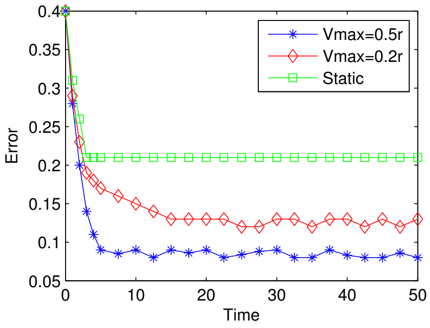


Figure 3: Error convergence of EUL

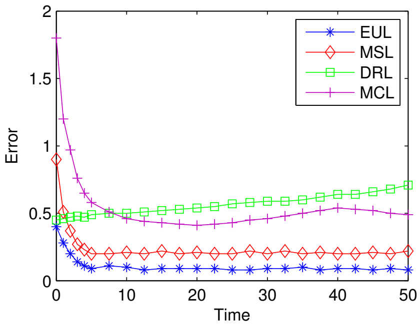


Figure 4: Accuracy for $v_{max} = 0.2r$

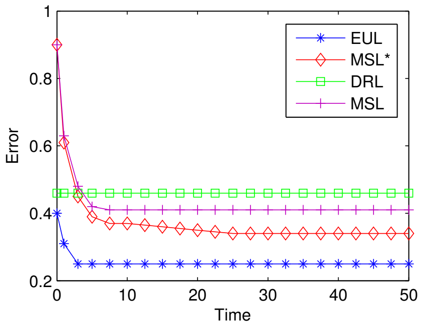


Figure 5: Accuracy for static nodes

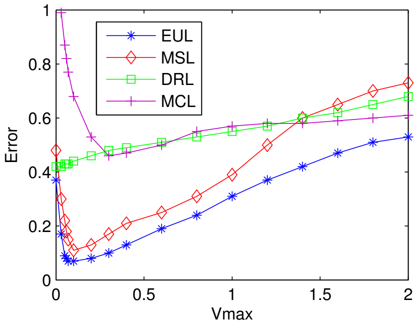


Figure 6: Impact of moving speed

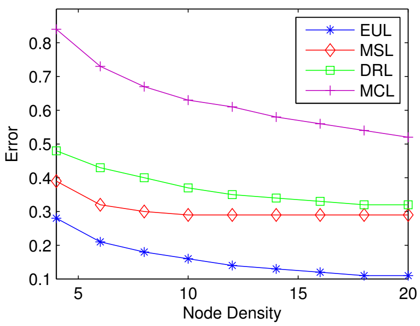


Figure 7: Impact of node density

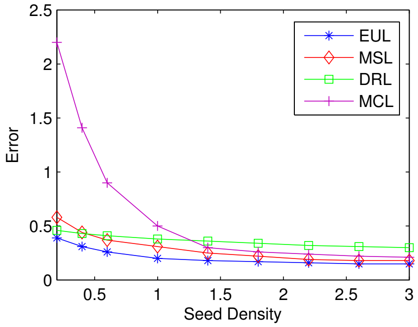


Figure 8: Impact of seed density

5.3. Moving Speed

Moving speed is the major influencing factor of estimate accuracy of EUL. In our motion model each sensor's move speed is random variation instead of a fixed value. We represent the velocity by the moving distance per time unit. The moving speed for all of seeds and nodes is chosen randomly in the range of $[0, v_{max}]$, and we increase v_{max} by 0.2 from 0 to 2 in each iteration.

Figure 6 shows the impact of moving speed on estimate error. Movement does not have too much impact on its performance of DRL. However, DRL is assumed that sensors follow a uniform distribution all the time to insure each hop-distance near the average value. This hypothesis is infeasible for many practical mobile sensor networks.

In EUL, MSL and MCL the location accuracy is seriously influenced by speed variation. The estimate errors drop sharply as moving speeds increase till they reach the minimum value. Then further increasing in the speed causes gradual error growth. There are two main reasons for the result. In the error drop stage, the faster sensors move, the more chance that a node has to exchange information with its new neighbors. Second, when sensors' moving speeds increase to a certain level, high speeds cause previous location information of a node to be no longer useful at all.

The figure shows that most algorithms have good performance around $0.2r$, and EUL has the best performance during the process of speed increase.

5.4. Node Density

Figure 7 compares the effect of node density on estimate error for different localization approaches. In this experiment, we keep the number of seeds constant in the network and vary the number of nodes. We set $s_d = 1$ and $v_{max} = 0.2r$ where most algorithms have good performance. As shown in the figure, higher density of nodes results in higher location accuracy in all algorithms. In DRL, high node density leads to each hop-distance near the average value. In MSL and MCL, high node density provides more first-hop and second-hop neighbors to communication with each node, and therefore location accuracy is improved.

In EUL, as the number of nodes goes up, the estimate error continuously drops. That is because when node density is high, each node has more direct neighbors to collaboratively filter the impossible locations.

5.5. Seed Density

High density of seeds increases network costs. Figure 8 shows estimate error affected by variation of seed density. The accuracy of MCL is considerably influenced by seed density. The estimate error drops fast as seed density increases. In MCL a node filters the impossible locations based on location information from seeds, so increasing the number of seeds provides a node with more location information of them. In DRL, estimate error derives majorly from the difference between each hop-distance value and average value. So its location accuracy is more dependent on node density instead of seed density. In EUL and MSL, a node filters the impossible locations utilizing information including both nodes and seeds. When node density exceeds a threshold, the number of seeds does not influence the location accuracy too much. Moreover, ILE in EUL results in a smaller initial error than MSL.

6. Conclusion

We have presented an efficient and universal dynamic localization approach, called EUL. Compared with other localization approaches, it is efficient in terms of cost, communication overhead, and iteration times. In addition, it is universal since it can work well in both mobile and static large-scale sensor networks. Moreover, EUL achieves a uniform energy consumption to address the issue of energy hot spots around seeds. Our simulation studies reveal that EUL technique has a low estimate error and a rapid convergence rate. Furthermore, EUL provides an accurate localization even when the density of seed nodes is low, and limits of energy and computing power are severe.

Acknowledgment

This work was supported in part by China 863 Program under grant No. 2007AA01Z180, China 973 Program under grant No. 2006CB30300, NSFC under Grant No. 60828003, NSFC under Grant No. 60873262, Shaanxi ISTC under Grant No. 2008KW-02, Canada NSERC Discovery Grant No. 341823-07, a McGill Startup Grant, NSF CNS-0832120, NSF CCF-0515088, RGC under Grant HKBU 2104/06E, and CERG under Grant PolyU-5232/07E.

References

[1] Z. Yang, M. Li, and Y. Liu, “Sea Depth Measurement with Restricted Floating Sensors”, In Proc. of RTSS, 2007.

[2] M. Li and Y. Liu, “Underground Structure Monitoring with Wireless Sensor Networks”, In Proc. of IPSN, 2007.

[3] Q. Wang, R. Zheng, A. Tirumala, X. Liu, and L. Sha, “Lightning: A Hard Real-Time, Fast, and Lightweight Low-End Wireless Sensor Election Protocol for Acoustic Event Localization”, IEEE Transactions on Mobile Computing (TMC), 7(5), 2008.

[4] Z. Yang and Y. Liu, “Quality of Trilateration: Confidence-based Iterative Localization ”, In Proc. of ICDCS, 2008.

[5] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “Wireless sensor networks: a survey”, Computer Networks, 38(4): p. 393-422, 2002.

[6] B. Karp and H. Kung, “GPSR: greedy perimeter stateless routing for wireless networks”, In Proc. of MobiCom, 2000.

[7] B. Hofmann-Wellenhof, H. Lichtenegger, and J. Collins, Global Positioning System: Theory and Practice: Springer-Verlag, 1997.

[8] T. He, C. Huang, B. Blum, J. Stankovic, and T. Abdelzaher, “Range-free localization schemes for large scale sensor networks”, In Proc. of MobiCom, 2003.

[9] M. Li and Y. Liu, “Rendered Path: Range-Free Localization in Anisotropic Sensor Networks with Holes”, In Proc. of MobiCom, 2007.

[10] D. Niculescu and B. Nath,“Ad hoc positioning system (APS) using AOA”, In Proc. of INFOCOM, 2003.

[11] A. Savvides, C. Han, and M. Strivastava,“Dynamic fine-grained localization in Ad-Hoc networks of sensors”, In Proc. of MobiCom, 2001.

[12] Y. Hsieh and K. Wang, “Efficient Localization in Mobile Wireless Sensor Networks”, In Proc. of SUTC, 2006.

[13] L. Hu and D. Evans,“Localization for mobile sensor networks”, In Proc. of MobiCom, 2004.

[14] M. Rudafshani and S. Datta, “Localization in wireless sensor networks”, In Proc. of IPSN, 2007.

[15] P. Bahl and V. Padmanabhan,“RADAR: an in-building RF-based user location and tracking system”, In Proc. of INFOCOM, 2000.

[16] D. Lymberopoulos, Q. Lindsey, and A. Savvides,“An Empirical Characterization of Radio Signal Strength Variability in 3-D IEEE 802.15.4 Networks Using Monopole Antennas,” in Proc. of EWSN, 2006.

[17] Y. Shang, W. Ruml, Y. Zhang, and M. Fromherz, “Localization from mere connectivity”, In Proc. of MobiHoc, 2003.

[18] Z. Zhong and T. He, “MSP: multi-sequence positioning of wireless sensor nodes”, In Proc. of SenSys, 2007.

[19] P. Bergamo and G. Mazzini,“Localization in sensor networks with fading and mobility”, In Proc. of PIMRC, 2002.

[20] S. Tilak, V. Kolar, N. Abu-Ghazaleh, and K. Kang, “Dynamic localization control for mobile sensor networks”, In Proc. of IPCCC, 2005.

[21] W.Y. Jywe, C.H. Liu, and C.K. Chen, “The min-max problem for evaluating the form error of a circle”, Measurement, 26(4): p. 273-282, 1999.

[22] J. Dennis Jr, D. Gay, and R. Walsh,“An Adaptive Nonlinear Least-Squares Algorithm”, ACM Transactions on Mathematical Software (TOMS), 7(3): p. 348-368, 1981.

[23] T. Camp, J. Boleng, and V. Davies, “A survey of mobility models for ad hoc network research”, Wireless Communications and Mobile Computing, 2(5): p. 483-502, 2002.

[24] J. Yoon, M. Liu, and B. Noble, “Sound mobility models”, In Proc. of MobiCom, 2003.