Research Journal of Applied Sciences, Engineering and Technology 4(21): 4252-4257, 2012
ISSN: 2040-7467
© Maxwell Scientific Organization, 2012
Submitted: December 20, 2011 Accepted: April 23, 2012 Published: November 01, 2012

Distributed Localization Algorithm based on Intersection and Exclusion

Mantian Xiang, Chengzhi Long and Guicai Yu
Nanchang University, Nanchang, Jiang Xi 330047 China

Abstract: In this study, we present a novel mobile anchor-assisted node Localization algorithm using Intersection and Exclusion (LIE) for wireless sensor networks. It sets up a discrete network model and then solves a system of geographic constraints based on connectivity information from the underlying communication network. Moreover, after the nearby nodes have completed localization, they become beacons, then cooperate with each other to localize the left unknown nodes in an incremental way and lastly obtains the optimum trajectory of mobile anchors and all the nodes locations. Simulation results demonstrate that the proposed LIE algorithm can improve significantly the localization accuracy compared to other similar positioning technology.

Keywords: Connectivity, LIE, negative, positive, wireless sensor network

INTRODUCTION

In Wireless Sensor Networks (WSN), location awareness is important since many applications depend on a successful localization (Bulusu *et al*., 2001; Bulusu *et al*., 2000; Estrin, 2001). For instance, sensor applications usually require sensor data to be tagged with the physical location of the measurements. Geographic routing protocols rely on node location in order to forward packets with low overhead. And context-aware applications need to determine the locations of network participants in order to customize content for users depending on their location. These and many other location-sensitive applications (Simic and Sastry, 2002; Whitehouse, 2002; Bergamo and Mazzini, 2002), require determining the position of mobile nodes with high accuracy and low cost. In addition, location-based routing protocols can save significant energy by eliminating the need for route discovery (Stupp and Sidi, 2005) and improve caching behavior for applications where requests may be location dependent (Doherty *et al*., 2001; Galstyan *et al*., 2004). Security can also been enhanced by location awareness (Alberto and Estrin, 2002).

However, putting GPS receivers in every node or manually configuring locations is not cost effective for most sensor network applications. One way of deploying a sensor network is to scatter the nodes throughout some region of interest. This makes the network topology random. Since there is no a priori communication protocol, the network is ad hoc. The first task that has to be solved is to localize the nodes, i.e., to compute their positions in some fixed coordinate system. It is of great importance to design scalable localization algorithms and provide error estimates that will enable us to choose optimal network parameters before deployment.

Due to high long range communication costs and low battery power, it is natural to seek decentralized, distributed algorithms for sensor networks. This means that instead of relaying data to a central location which does all the computing, the nodes process information in a collaborative, distributed way. For instance, they can form computational clusters, based on their distance from each other. The outcome of these distributed, local computations is stored in local memory and can then be, when necessary, relayed to a centralized computing unit. Robustness to node failures is another reason to seek distributed rather than centralized algorithms.

There are three fundamentally different techniques for location discovery based in the type of hardware used for inferring location. The first and simplest technique is to statically record the location of each node at deployment time. This approach clearly fails when nodes move. Even in static sensor networks, this approach requires an extra step during deployment that is often costly and sometimes, for instance, in the case of aerial sensor dispersion, simply infeasible. A second approach is to outfit each device with dedicated positioning hardware, such as infrared transmitters, ultrasound transcievers and GPS receivers. These schemes require a substantial pre-deployed hardware infrastructure in the environment. The dedicated hardware on each node is often expensive, takes up volume and consumes a significant portion of the total energy budget. Finally, the third and last approach is to extract geographic information from wireless communication hardware that is already present on wireless nodes.

In this study, we propose a new distributed localization algorithm LIE (Localization based on Intersection and Exclusion) in the discrete network model. Comparing to Bounding Box algorithm, LIE reduces the

Corresponding Author: Mantian Xiang, Nanchang University, Nanchang, JiangXi 330047 China

requirement for the anchor nodes' density inordinately, improves the localization estimation accuracy, especially of sparse nodes localization and coverage ability and also reduce the computational cost, satisfying the requirement of wireless sensor networks' high accuracy localization. The study makes the following contributions:

- it makes a case for the use of both positive and negative information derived from the physical layer, for the transitive dissemination of location information and for explicitly representing location estimates as sets of points. These mechanisms enable LIE to greatly improve the fidelity of its location estimates.
- it Second, it outlines a distributed, efficient and scalable algorithm for estimating node location. This protocol enables nodes without any dedicated hardware to determine their position with high accuracy based on a small number of anchor nodes.
- it describes a simple implementation of the model and presents initial results, shows that the LIE approach is both effective and practical.

This study proposes a distributed LIE localization algorithm for sensor networks and analyzes its accuracy in terms of the expected area of uncertainty for a node's position. It does not rely on special hardware such as beacons to localize a node and lows cost consequently. Moreover, the computational and communication cost of the LIE algorithm is very less, suitable for the scene which has strict limitation with the nodes' computing ability and has certain application value.

LITERATURE REVIEW

In this section, we provide an overview of several localization algorithms. Survey study can be found in Rabacy *et al*. (2000).

Numerous range free localization mechanisms were described in the literature. The first solutions were typically central. For example, the connectivity matrix of the sensors is used in Estrin (2001) to constraint the possible locations of the sensors in the network, achieving a position estimation by centrally solving a convex optimization problem. Many distributed solutions were also suggested.

The research of range based systems has been more extensive, supplying bounds on the achievable positioning accuracy when assuming specific calibration data, noise patterns and location distribution. In particular, lower bounds for localization uncertainty using the Cramer-Rao bound were presented in Simic and Sastry (2002). The behavior of the specific algorithms, however, is almost always investigated by simulation, either by the authors or in subsequent works Whitehouse (2002).

In the centralized convex optimization scheme, the intersection of all neighboring nodes' communication range is considered. A bounding box is constructed and the centroid is taken as the position estimate of the node. Algorithms based on Multidimensional Scaling (MDS) can operate in both range-free and range-based scenarios (Bulusu *et al*., 2000). They are very accurate and only three or four anchor nodes are required. However, they also require considerable amount of overhead for communication and computation.

Schemes based on anchor location propagation throughout the network include (Estrin, 2001; Simic and Sastry, 2002). The anchor's position as well as hop counts or distance from the node are used to bound the location of the node. At least three anchors are needed for nodes to obtain a unique position estimate. The estimation accuracy depends on the accuracy of the distance information used in lateration or bounding.

For the angle-based schemes, the direction between nodes can be determined by using antenna-arrays. In (Whitehouse, 2002), nodes which have at least three bearings to anchors can triangulate their positions. Both range and angle information is used in order to necessitate only one anchor's information for position estimation (Bergamo and Mazzini, 2002).

Several schemes have explored the use of mobile anchors and nodes. In Stupp and Sidi (2005), a single mobile anchor traverses the network and allows stationary sensor nodes to compute their location estimates based on at least three neighboring nodes' locations. Multiple mobile anchors are used and both anchors and sensor nodes are mobile. The Monte Carlo algorithm is used for localization.

The study in Doherty *et al*. (2001) considered the possibility of quantized Received Signal Strength (RSS) and presented mathematical analysis of the accuracy with varying levels of quantization. The performance comparisons between ROCRSSI and APIT schemes are given in Galstyan *et al*. (2004). In the APIT (Approximated Point-In-Triangulation) algorithm (Alberto and Estrin, 2002), each node first determines if it is within a particular triangle formed by a set of anchors within anchor range. The position is estimated to be the center of intersection of all triangles in which the node has identified to be within.

In the Bounding Box methods, it defines bounds on the coordinates of a mobile node in terms of the estimated distance to each anchor node. Such methods are analyzed in Bulusu *et al*. (2000) and used for initial position estimation in Collaborative Multilateration, described below. However, Bounding Box methods fail miserably when all mobile nodes are not within the convex hull defined by the anchor nodes. They also fail under noisy range estimates, a single terribly small estimate can render the bounding box of all nodes in the network inconsistent.

However, Bounding Box methods gracefully degrade under non-convex networks because ranging estimates that are too large do not effect the Bounding Box.

Other approaches (Rabacy *et al*., 2000; Shang *et al*., 2003) assumed node to node communication can be used to flood the location of all the anchors to all the sensors. Using the hop-count as an estimate of the euclidian distance (by computing the average distance between sensors or by analytically deriving it), a sensor can estimate its position via triangulation. The accuracy of all of these solutions, however, was evaluated experimentally, rather than analytically.

LIE localization protocol: In this section, we begin with a discussion of the motivations and assumptions. It is followed by the description of the LIE localization algorithm. We then discuss the advantages and limitations of our proposed scheme.

Motivations and assumptions: Although distributed range-based algorithms have a higher accuracy than the distributed range-free approaches in general, the range-free approaches are more cost effective. In our study, we focus on the design of a distributed range-free localization algorithm that has a high accuracy and does not require communication between neighboring sensor nodes.
LIE achieves high accuracy through three mechanisms:

- It aggressively extracts geographic constraints from the link layer
- It propagates constraints throughout the network
- It explicitly tracks the set of all possible locations for any given node

In Fig. 1, LIE aggressively extracts positive and negative information from the link layer and converts it into geographical constraints.

By positive information, receiving a direct beacon from a nearby node enables such a constraint to be established. In contrast, negative information corresponds to a constraint that can be generated when a node indirectly determines the existence and position of another node *C* but fails to receive direct transmissions.

Using both positive and negative information enables node position to be narrowed down significantly. If node *C* can receive beacons from *B* (positive information), but not from node *A* (negative information), its position is limited to an annulus, shown in dark gray.

LIE translates these constraints into geographic terms and disseminates them transitively throughout the network, creating an interdependent web. Transitively propagating location information enables nodes that are not within the immediate vicinity of anchors to determine their location. It also enables nodes to extract negative

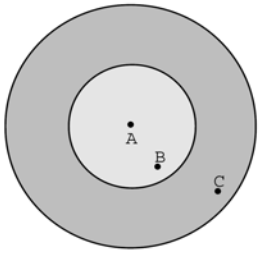

Fig. 1: Positive and negative information

information by discovering the presence and estimated location of other nodes in the network. Transitively combining position estimates enables information from sparsely distributed anchors to be coalesced together to reduce positioning error.

LIE algorithm: We now describe the LIE localization algorithm in details. Each anchor transmits the beacon signals at varying power levels consecutively. Each beacon signal packet includes the anchor's ID, anchor's location and the transmit power level information. Each node listens for beacons and collects the anchor's information. For each beacon heard, the sensor node determines which region of the anchor's concentric transmission circles it lies within.

In a square area Q = [-s/2, s/2]×[-s/2,s/2] with N nodes distributed randomly and K anchor nodes in it. Anchor nodes broadcast their position information to the whole network, radio radius is r. We assume all nodes' position within $[x_i^{min}, x_i^{max}]\times[y_i^{min}, y_i^{max}]$, a rectangular range. Making the circus communication range approximately as its inner square bounding box (Bulusu *et al*., 2000), it will simple the two constraints of the node position as linear constraints and it will narrow the coverage area range and reduce the uncertainty of node position.

In order to facilitate the study, we will discretize the model above, as shown in Fig. 1. Assuming n is a even number, Q is decomposed into $(n+1)^2$ square cells and Q = [-n/2, n/2]×[-n/2, n/2]. For the cell (i,j), its node communication range is:

$$B^{\rho}_{(i,j)} = [i - \rho-, i+\rho]\times[j- \rho, j+\rho]$$

where, $\rho = \left\lceil \frac{m}{\sqrt{2}s} \right\rceil$ and we set $0 \le \rho \le n/8$. S is a unknown node, $Q\rho = B_{(0,0)}^{n/2\rho}$ represents the rectangular area whose distance to Q boundary at least ρ. We don't discuss critical effect in this study, so except anchor nodes, the rest nodes' random distribution range is Q_ρ. When we discuss the mathematical expectation of variable, we just study the nodes in the range of $Q_{3\rho}$.

Supposing unknown node S has m neighbor anchor nodes $S_{K_1}, ..., S_{K_m}$, for which their coordinates, respectively are (x_1, y_1), …, (x_m, y_m), the rest anchor nodes' coordinates are (x_{m+1},y_{m+1}), ..., (x_K, y_K). A_S represents all possible position's area of S.

When *m* is not 0, A_S is the communication range's intersection of these *m* neighbor anchor nodes; if *m* is 0, A_S is the complement set of all *K* anchor nodes' communication range's union:

$$A_S = \begin{cases} \left(\bigcap_{i=1}^{m} B^{\rho}_{x_i y_i}\right) - \left(\bigcup_{j=m+1}^{K} B^{\rho}_{x_j y_i}\right) & m > 0 \\ Q_{\rho} \bigcup_{i=1}^{K} B^{\rho}_{x_i y_i} & m = 0 \end{cases} \quad (1)$$

DISCUSSION

There are three distinct advantages of LIE localization algorithm. First of all, LIE is distributed and is simple to implement. For the anchors, their only task is to transmit beacons signals with different power levels. For each sensor node, the determination of the intersection points from three chosen anchors as well as the position estimate by averaging are not computationally intensive. Second, no information exchange between neighboring sensors is necessary. This reduces the energy requirement for localization. In addition, LIE has a higher accuracy than some other range-free localization algorithms. Simulation comparisons will be presented in the next section.

Algorithm performance analysis: Next, In this section, we present the performance evaluation of LIE algorithm. Let random variable $X = |A_S|$ to represent the cell amount in A_S, $1 \le X \le (n+1-2\rho)^2$. We assume *S* locate within (0, 0). Set $p = (2\rho+1)^2/(n+1)^2$, $q = 1-(2\rho+1)^2/(n+1)^2$.

The expectation E(X) of localization area size: Definition:

$$\lambda_{ij} = \begin{cases} 1 & (i,j) \in A_S \\ 2 & (i,j) \notin A_S \end{cases} \quad (2)$$

Then,

$$E(X) = \sum_{i=-n/2+\rho}^{n/2-\rho} \sum_{j=-n/2+\rho}^{n/2-\rho} E(\lambda_{ij})$$

$$= \sum_{i=-n/2+\rho}^{n/2-\rho} \sum_{j=-n/2+\rho}^{n/2-\rho} \Pr(\lambda_{ij} = 1) \quad (3)$$

To LIE algorithm:

$$\because \Pr(\lambda_{ij} = 1, (i,j) \in Q\rho\, B^{2\rho}{}_S) = (1-{}^{2p}{}_S)^K$$

and,

$$\Pr(\lambda_{ij} = 1, (i,j) \in B^{2\rho}{}_S)$$

$$= \left[1 - 2p + \frac{(2\rho+1-i)(2\rho+1-i)}{(n+1)^2}\right]^K + \sum_{m=1}^{K} \left\{ C_K^m \left[1 - \frac{2(2\rho+1)2}{(n+1)^2} + \frac{(2\rho+1-i)(2\rho+1-i)}{(n+1)^2}\right]^{K-m} \cdot \left[\frac{(2\rho+1-i)(2\rho+1-j)}{(n+1)^2}\right]^m \right\}$$

$$= \left[1 - \frac{2(2\rho+1)^2}{(n+1)^2} + \frac{2(2\rho+1-i)(2\rho+1-i)}{(n+1)^2}\right]^K + \sum_{m=0}^{K} \left\{ C_K^m \left[1 - \frac{2(2\rho+1)2}{(n+1)^2} + \frac{(2\rho+1-i)(2\rho+1-i)}{(n+1)^2}\right]^{K-m} \cdot \left[\frac{(2\rho+1-i)(2\rho+1-j)}{(n+1)^2}\right]^m \right\}$$

$$\therefore E(X) = (1-2p)K\left[(n+1-2\rho)^2 - (4\rho+1)^2\right] + 1 + 4\sum_{i=0}^{2\rho}\sum_{j=1}^{2\rho}\left[1 - \frac{2(2\rho+1)^2}{(n+1)^2} + \frac{2(2\rho+1-i)(2\rho+1-j)}{(n+1)^2}\right]^K \quad (4)$$

The required amount K of anchor nodes: When n and ρ is invariant, $\lim_{K\to\infty} E(X) = 1$, it indicates that when the amount of anchor nodes is infinite large, localization estimation is approaching to the most ideal situation. Set e>0, we study the required optimization value of anchor nodes amount K when |E(X)-1|<e.

To LIE algorithm, obviously when $i = 0$, $j = 1$, $E(X)$ gets the maximum value.
Since:

$$E(X) - 1 \le \left[1 - \frac{2(2\rho+1)}{(n+1)^2}\right]^K \cdot \left[(n+1-2\rho)^2 - (4\rho+1)^2 + 8\rho \times (2\rho+1)\right] = \left[1 - \frac{2(2\rho+1)}{(n+1)}\right]^K \left[n^2 + 4\rho^2 - 4\rho n - 4\rho + 2n\right] \quad (5)$$

Therefore, to LIE algorithm, anchor nodes' density must satisfy:

$$\frac{K}{(n+1)^2} > \frac{\log(n^2 + 4\rho^2 - 4\rho n - 4\rho + 2n) - \log}{2(2\rho+1)} = K_{\min} \quad (6)$$

The influence of cell density: Set $n' = \delta n$ and δ>1, on the basis of $\rho = \left[\frac{rn}{\sqrt{2s}}\right]$, we get $\rho' \approx \delta\rho$. Because the length of side of continuous area and the actual value of node communication radius don't change and:

$$p' = \frac{(2\rho'+1)^2}{(n'+1)^2} = \frac{(2\delta\rho+1)^2}{(\delta n+1)^2} \approx p = \frac{(2\rho+1)^2}{(n+1)^2},\ q' \approx q$$

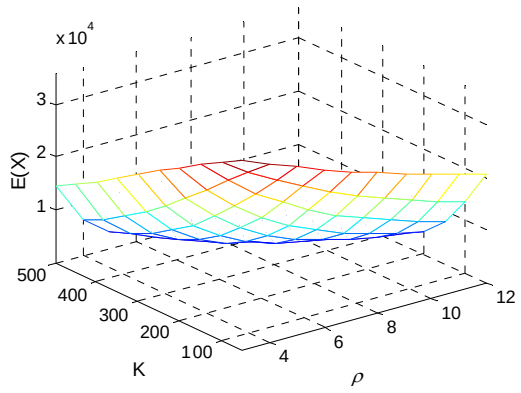

Fig. 2: *E(X)* of bounding box with different ρ and *K* (*n* = 200)

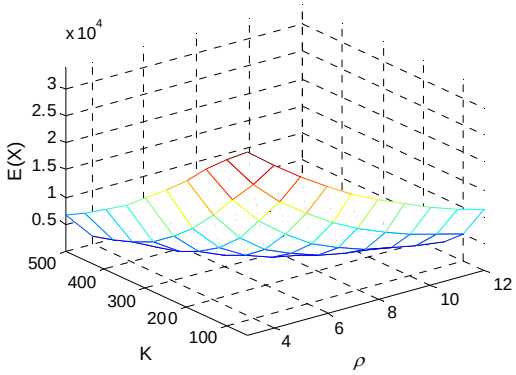

Fig. 3: *E(X)* of LIE with different ρ and *K* (*n* = 200)

So, according to (4) we get:

$$
\begin{aligned}
E(X') &= \sum_{i=-\delta n/2+\delta\rho}^{\delta N/2-\delta\rho} \sum_{j=-\delta n/2+\delta\rho}^{\delta n/2-\delta\rho} \Pr(\lambda_{ij} = 1) \\
&= (1-2p')^K \left[(\delta n + 1 - 2\delta\rho)^2 - (4\delta\rho + 1)^2 \right] + 1 \\
&\quad + 4 \sum_{i=0}^{2\delta\rho} \sum_{j=1}^{2\delta\rho} \left[1 - \frac{2(2\delta\rho+1)^2}{(\delta n+1)^2} + \frac{2(2\delta\rho+1-i)(2\delta\rho+1-j)}{(\delta n+1)^2} \right]^K \\
&\approx \delta^2 E(X)
\end{aligned} \tag{7}
$$

From (7), we can find that the last full size of localization estimation area is not relative to the intensive degree of discrete area cell, which is also the advantage of our LIE algorithm.

SIMULATION RESULTS AND ANALYSIS

In this section, we present the performance evaluation of LIE as well as the comparisons with Bounding Box algorithm. Both the two algorithms are simulated in MATLAB 7.0 on a processor Intel Core2 Duo 2.0 GHz with 2048 MB RAM. All the nodes are randomly placed according to a uniform distribution and the anchor nodes are randomly generated. To model the errors in grid placements, we add Gaussian noises to the coordinate of nodes. To allow for easy comparison between different scenarios, position estimate error is normalized to ρ, denoted by ρ %. As a example, for a 10% ρ placement error, a random variable from N (0.10% ρ) is added to each coordinate of a grid point. The data points represent averages over 100 independent and distinct runs.

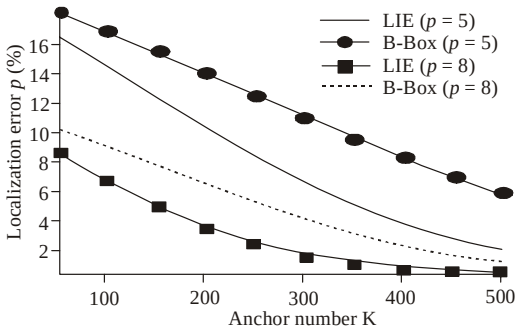

Fig. 4: The comparison of two algorithms' position error with different amount of anchor nodes (*n* = 200)

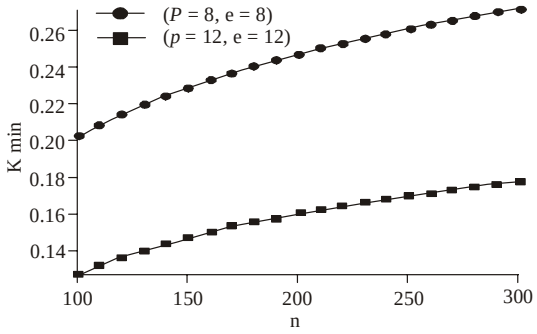

Fig. 5: *K*min with different network size *n*

Localization accuracy of LIE and bounding box: In this section, we compare typical Bounding Box (abbreviated as B-Box) with LIE algorithms' positioning accuracy. For *n* = 200, ρ∈[3, 12], K∈[50, 550] the average size of the position estimate by Bounding Box and LIE is shown in Fig. 2 and 3, respectively. Observe that the average error of the algorithm steadily decreases as *n* increases. Since ρ is fixed, this corresponds to increasing the size of the region of operations. Compared with Bounding Box, the positioning area estimated by LIE contains the least cells, it shows the positioning result of LIE algorithm is better and more stable and the positioning area of Bounding Box has the higher uncertainty.

The position error of these two algorithms for ρ = 5 and 8, respectively is shown in Fig. 4. Both the localization error of two algorithms decreases with the anchors increase. It is obvious to find that Bounding Box algorithm has a poor position accuracy when anchor nodes' density is not high, for example as ρ is 5 and *K* is

100, the density of anchor node is about 1%, the estimation error of Bounding Box is about 17%, but LIE algorithm reduces to below 14.2%; when the number of anchor nodes reach to more than 220, the position error of LIE algorithm will reduce to less than 10% quickly, while Bounding Box maintains upon 13%. LIE algorithms' estimation error is less than Bounding Box algorithm. It shows that comparing to Bounding Box algorithm, LIE algorithm improves the position accuracy greatly.

The average uncertainty in the nodes position can be larger than the sensing range ρ. The poor performance of Bounding Box can be explained by the relatively weak constraints imposed by it. This is more pronounced at higher fractions of unknown nodes. However, we can significantly improve node localization results by using constraints imposed by LIE, which achieves better localization for smaller value of ρ.

Dependence of the localization error on the anchor nodes' density: Figure 5 shows the minimum anchor nodes' density of the Bounding Box (averaged over all unknown nodes) versus different numbers of the fraction of nodes and error threshold *e*. In the results shown, we used ρ = 5, e = 8 and ρ = 12, e = 12. There is a slow increase in K as the the number of cells increases. Although we don't show results here, we also found that the localization error decreases with increasing density; this is because at higher node densities, there are more nodes (including known ones) in the vicinity of a anchor, which helps to better localize the nodes and hence, impose stricter constraints on the unknown nodes that sense it.

In the scene ρ = 8, *e* = 8, n∈[100, 300], checking the LIE algorithms' requirement to anchor nodes' density. The minimum anchor nodes density is about 0.27. The LIE approach seems to achieve better localization for smaller values of ρ. We believe this is because the localization error is largely determined by the connectivity-imposed constraints, which become less effective as ρ increases.

CONCLUSION

The main contributions of this study are a distributed LIE localization algorithm for sensor networks and analyze its accuracy in terms of the expected area of uncertainty for a node's position. It does not rely on special hardware such as beacons to localize a node and lows cost consequently. There are many possible improvements of the LIE algorithm. For example, representing possible node locations as sets throughout the LIE algorithm, instead of collapsing them to a single, best-estimate point, enables LIE to keep track of locations accurately. The computational and communication cost of the LIE algorithm is very less, suitable for the scene which has strict limitation with the nodes' computing ability and has certain application value. We plan to address these issues in future study.

ACKNOWLEDGMENT

This study was supported by the Significant Scientific Research Programs Foundation of the Education Department of Jiangxi Province (No. GJJ11062); The Main Technology Pathfinder Programs Foundation of Jiangxi Province (070002); The Pillar Programs Foundation of the Science and Technology Department of Jiangxi Province (2007ZD03700).

REFERENCES

Alberto C. and D. Estrin, 2002. Ascent: Adaptive self-configuring sensor network topologies. ACM SIGCOMM Comput. Commun. Rev., 32(1): 62-69.

Bergamo, P. and G. Mazzini, 2002. Localization in sensor networks with fading and mobility. IEEE PIMRC 2002, Lisbon, Portugal, pp: 71-83.

Bulusu, N., J. Heidemann and D. Estrin, 2000. GPS-less low cost outdoor localization for very small devices. IEEE Pers. Commun. Magaz., 7(5): 28-34.

Bulusu, N., J. Heidemann and D. Estrin, 2001. Adaptive beacon placement. Proceedings of the 21st International Conference on Distributed Computing Systems (ICDCS-21), Phoenix, Arizona, April, pp: 86-88.

Doherty, L.K., S.J. Pister and L. El Ghaoui, 2001. Convex position estimation in wireless sensor networks. INFOCOM 2001, 3(7): 1655-1663.

Estrin, D., 2001. Embedded, everywhere: A research agenda for networked systems of embedded Computers. National Research Council Report, pp: 141-153.

Galstyan, A., B. Krishnamachari and K. Lerman, 2004. Distributed online localization in sensor networks using a moving target. IPSN 2004, pp: 61-70.

Rabacy, J.J., M.J. Ammer, J.J.L daSilva, D. Patel and S. Roundy, 2000. Picorodio supports ad hoc ultra-low power wireless networking. Computer, 33(7): 42-48.

Simic, S. and S. Sastry, 2002. Distributed localization in wireless ad hoc networks. Technical Report UCB/ERL M02/26, UC Berkeley.

Shang, Y., W. Ruml, Y. Zhang and M.P.J Fromherz, 2003. Localization from mere connectivity. Proceeding of the 4th ACM Int'l Symp. on Mobile Ad Hoc Networking and Computing. ACM Press, Annapolis, pp: 201-212.

Stupp, G. and M. Sidi, 2005. The expected uncertainty of range free localization protocols in sensor networks. First Theoret. Comput. Sci. (TCS) J., pp: 86-99.

Whitehouse, C., 2002. The design of calamari: an ad-hoc localization system for sensor networks. MA. Thesis, University of California at Berkeley.