EUCOW: Energy-Efficient Boundary Monitoring for Unsmoothed Continuous Objects in Wireless Sensor Network

Jiehui CHEN, Mitsuji Matsumoto*

GITS, Waseda University, 1-104 Totsukamachi, Shinjuku-ku, Tokyo, 169-8050, JAPAN
chenjiehui0574@fuji.waseda.jp, mmatsumoto@waseda.jp

Abstract

The proliferation of Continuous boundary monitoring in large-scale wireless sensor networks in various applications such as fires and hazardous bio-chemical material diffusion, soil moisture draws tremendous significant attentions. In this paper, we propose an energy-efficient EUCOW algorithm that monitors both interior and exterior boundary of unsmoothed object by selecting only the nodes are within transmission range r distance far from event boundary and a minimum set of representative nodes that actually report to the sink.

The simulation results significantly verify that our algorithm can achieve a thinner accurate boundary with less boundary nodes as well as greatly reduce the report messages to save energy, especially when numerous event objects emerge into less number of objects.

Keywords *wireless sensor networks, continuous object tracking, boundary monitoring*

I. Introduction

Nowadays, numerous researches are focusing on Wireless Sensor Network (WSN) that is well developed to carry out tasks such as bio-chemical diffusion and military surveillance. One of typical research area is object tracking in WSN. The majority of previous works is to detect and track small moving target. However, relatively smaller efforts have been made on continuous object monitoring. Especially in unsmoothed continuous object environment, the research works are really limited. To monitor such complex phenomena in real time, it requires inordinate amount of message exchanges between sensor nodes to collaboratively estimate the object's movement and location information. Therefore, it is judicious to create an efficient algorithm that avoids the data errors so as to minimize the communication cost and prolong the network lifetime.

Literarily, we find various approaches to monitor continuous objects. The simplest and energy inefficient way would be just let all sensor nodes be awake and detect the

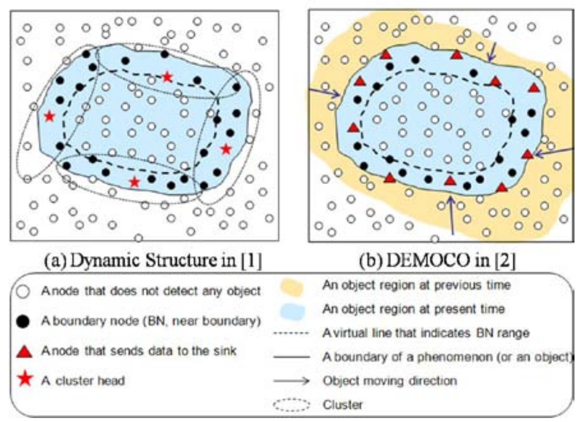


Fig. 1 Comparison of number of representative nodes

phenomena and transmit its readings to the sink. The sink is usually a more powerful node that relays the gathered information directly or indirectly to the base station. However, this approach causes huge energy consumption.

In [1], X.ji et al. proposed a dynamic cluster-based mechanism to track the movement of boundaries and facilitate the fusion and dissemination of boundary information in a sensor network. Fig.1(a) shows that the boundary nodes are grouped into several clusters. (a.k.a. boundary node, hereafter it is acronymized as BN)The solid dot in black indicates the nodes selected nearby the boundary of an object and the solid pentagram in red indicates the cluster head that gathers and relays the data from nodes inside the cluster. It is obvious that this approach performs better than the simplest one since only the nodes near object boundary are associated with transmitting data to the sink.

In [2], J.H.Kim et al. propose an energy efficient DOMOCO algorithm described in Fig.1(b), it seems to be more BNs selected comparing to[1], but the number of nodes that report data to the sink is much less than[1]. We called that kind of nodes as representative nodes (a.k.a. representative node, hereafter it is acronymized as RN) which are triangles in read in Fig.1(b).

This paper presents an energy efficient EUCOW algorithm that monitors unsmoothed object by selecting a minimum set of representative nodes near event boundaries.

* The corresponding author.

We use BN-Array and corresponding BN-Array to store boundary status information in each node's buffer. Ideally, only the nodes with rich boundary neighborhood information are required to report. Moreover, we also know it's more realistic to consider three-dimensional space where we should set more complex spatial data for every sensor node, for simplicity, we just evaluate two-dimensional space in our algorithm.

The remaining part of this paper is organized as follows: Sect.2 shows the related works and their drawbacks. In Sect.3 we explain the definitions as well as required assumptions for our algorithm. Sect.4 presents our proposed algorithm in detail and in Sect.5 we evaluate our algorithm with reliable simulation by using VC++ and Matlab. Finally, we give out the conclusion in Sect.6.

II. Related Works

There are numerous researches on single and multiple targets tracking and localization in wireless sensor network [3][4]. [5] presents a distributed peer-to-peer target tracking framework that requires collaboration between sensor nodes. [6] proposes an efficient target localization algorithm by varying transmission range in wireless sensor network. The idea is innovative but the way of controlling transmission range is ambiguous. In [7] and [8], authors work on detecting some continuous changeable objects which are closely related to our topic. The main difference from our approach is that they don't consider the situation when the object shapes changed and even event objects emerged in real time fashion. Moreover, [7] might lead to huge energy consumption since all BNs report data to sink.

Xiang Ji et al. [1] propose a dynamic cluster based algorithm which can relatively efficiently tracks continuous objects' movements through monitoring their changing boundaries. Once phenomena happen, sensor node immediately broadcasts a query message to all its neighbors to inquire their readings. If the sensor node detects one different detection status from its neighbors, the sensor node becomes a BN. After the BN selection, cluster formation process takes place among the BNs. However, the clustering algorithm is somewhat confused. The main drawback of this paper appears where every BN including every cluster head is directly or indirectly involved in routing data to the sink, which will yield more overhead and traffic.

Z.Cheng et al [9]. COBOM is proposed to monitor boundary of continuous object. Once a sensor node's current reading is different from previous observation, it broadcast its current reading and ID. A node who is going to receive the reading and ID will store them into its array (called BN-array) and until find any different reading in BN-array, the sensor node becomes a BN. Those BNs whose neighborhood information are sufficient will be selected to be RNs. This algorithm is efficient with following two aspects 1) only limited RNs are selected 2) by using BN-array, the report message size will not increase fortunately since each message maintain its neighbors' ID in the form of detection status in bits.

J.H.Kim et al [2] presents DEMOCO that improved COBOM by just considering nodes in "IN" range, ignoring those in "OUT" range which theoretically reduce half of selecting BNs and RNs so as to achieve energy saving as well as prolong network lifetime.

In this paper, we are only focusing on monitoring unsmoothed continuous object. Our paper is a kind of enhanced one to [2]. We use genEvent algorithm to theoretically generate the expected event, if we can truly get more precise prediction on the expected shape of unsmoothed objects, we allege that we can achieve less number of BNs and RNs than [2] especially when events densely happen. And we can also clearly distinguish interior and exterior object boundary which be shown by our simulation results.

III. Preliminaries

In this section, we make general assumptions required to achieve our algorithm and present the definitions that will be used in this paper.

3.1. Assumptions

- All sensor nodes with the same capability and functionality are arbitrarily deployed.
- A sensor node cannot move after deployment and it knows its own location via Global Positioning System (GPS) [10] or other possible techniques such as triangulation [11] or localization [12].
- Each node has the same communication range r and knows the sensor nodes ordering in the network and cyclic ordering round it in communication range.

3.2. Definitions

Network Model: The topology of the network can be presented by a simple graph $G = (V(G), E(G))$ in a 2-Dimensional plane , where an edge e_{ij} ($e_{ij} = e_{ji}$) exists for each pair of nodes u_i and u_j. After being deployed, all the nodes can geographically be remarked with $1,2\ldots n$ (n is the total number of sensor nodes) from the top-left to button-right. We get all sensor nodes ordering $\{u_1, u_2, \ldots, u_n\}$

① **Definition(Neighbors (*Nu*)):** Let *u* and *Nu* represent a node and neighbors of *u* respectively. The neighborhoods *Nu* are those nodes that are within communication range *r* of *u*. e.g. v_1 , $v_2, \ldots, v_n$

② **Definition(Boundary Neighborhood Array (BN-Array)):** BN-Array is an array of sensor nodes' IDs $\{ID_1, \ldots, ID_u, \ldots, ID_n\}$ ($1 \le u \le n$) in consequence, corresponding with sensor nodes ordering.

③ **Definition(Boundary Status (BS)):** If the status of a sensor node is a boundary node, we give this sensor node a value 1 or vice versa. So a sensor node has 1 or 0 to indicate its own boundary status in this way.

④ **Definition(Corresponding BN-Array (CBN-Array)):** CBN-Array is an array of sensor nodes' BS $\{BS_1, \ldots, BS_u, \ldots, BS_n\}$ $(1 \le u \le n)$, corresponding with BN-Array.

⑤ **Definition(ON-message):** an event sensor node will broadcast a message to its ***Nu*** to inform them its changed BS so that ***Nu*** can update their CBN-Array. We call this message as ON-message with 1 byte.

⑥ **Definition(INFO-message):** Once BS of a sensor node changes, it should broadcast its changed BS with 1 byte INFO-message to its ***Nu*** until all sensor nodes get updated.

⑦ **Definition(Representative Nodes (RNs)):** A representative node is a node that actually sends data to the sink. Only a few representative nodes will be selected among BNs to save energy.

IV. BOUNDARY DETECTION AND MONITORING IN WSN

In this section, we introduce our algorithm for boundary monitoring by selecting RNs in unsmoothed continuous object. In section 4.1, we generally illustrate genNet algorithm, genEvent algorithm, BN Selection Algorithm and RN Selection Algorithm presented in detail. In section 4.2, we present our proposed EUCOW algorithm. Moreover, section 4.3 gives statistical performance analysis.

4.1. Algorithms Illustration

A. genNet Algorithm

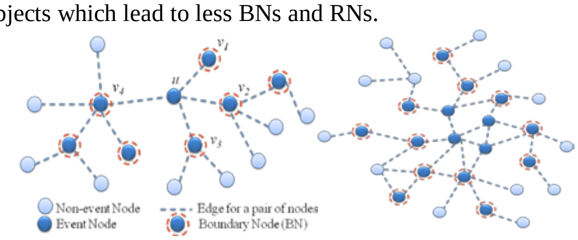
We adapt Voronoi-based network [13], for simplicity, just focusing on 2D deployment of sensor nodes to get accurate and more reliable boundary. We assume that each sensor has enough buffer to store BN-Array and CBN-Array. The BN-Array is decided in priority by the network. After deployment, each sensor node broadcasts a hello message to its ***Nu*** to get neighborhood information so as to get a value "-1" for ***Nu*** and "0" for non-neighborhood nodes in their CBN-Array.

genNet Algorithm

00 Sensor nodes deployment
01 BN-Array assignment
02 Each node send hello message to its N*u*
03 CBN-Array assignment with neighborhood -1 and non-neighborhood 0.
03 End /* the sink announces the end*/

B. genEvent Algorithm

Event can be defined as a single or multi-event. We adapt tree structure concept to describe the nodes' structure as Fig.(a). We called it *treeNet*. In fig. once a phenomenon happens, there should be a start event node *u* and its N*u* $\{v_1, v_2, v_3, v_4\}$. The event signal randomly spreads in neighbors' directions: $u \rightarrow v_1$ or $u \rightarrow v_2$ or $u \rightarrow v_3$ or $u \rightarrow v_4$, and then v_1, v_2, v_3, v_4 spread signals to their N*u* respectively. Once a sensor node receives a signal from another sensor node, it updates the value in the corresponding position of CBN-Array with 1. Theoretically, a final single event object might be like Fig.2(b). In the case of multi-object phenomena, some objects emerge into less number of objects which lead to less BNs and RNs.



(a) (b)

Fig. 2 TreeNet Structure

genEvent Algorithm

00 Events happen
01 Each event signal spreads in randomly chosen N*u* directions/*it's allowed to transmit a signal back*/
02 When objects emerge, back to 01
03 End /*suppose that all event nodes know the end of an event*/

C. BN Selection Algorithm

Afterwards, we deploy an efficient boundary node selection mechanism described as follows: Since each node has already got CBN-Array $\{BS_1, BS_2, \ldots, BS_n\}$ fixed. Regardless of all the values 0, the remaining values should be either -1 or 1.

```
void main()
{
if (BSi =…=BSu =…=BSj=1), u =interior object node
   or
if (BSi =…=BSu=…=BSj= -1), u =exterior object node
  else
    {
     if (BSu=1), u =interior BN
     or
     if (BSu=-1), u =exterior BN
     end
    }
end
}
```

Where $0 \le i \le u \le j \le n$.

D. RN Selection Algorithm

Since BSs of all the BNs are 1 in CBN-Array. Assume that we have a set of BNs' BSs values $\{BS_i, \ldots, BS_i, \ldots,$

BS_k }($1 \le i \le j \le k \le n$). A minimum set of sensor nodes whose CBN-Arrays includes all the values in {BS_i ,…, BS_i,…, BS_k } will be selected to be a set of RNs for a consideration of energy consumption during the transmission to the sink. The less RNs that report to the sink, the more energy be saved. One thing we have to mention is that not only the BNs but other sensor nodes could be RNs. Theoretically, the RNs should have rich BNs information and be closed to the object boundary.

RN Selection Algorithm

00 Comparing CBN-Arrays, to select out the nodes whose CBN-Array has more BNs' BSs. It's allowed to have several top 1s, but we randomly chose one regardless of the previous inefficient one.
01 Among the remaining nodes, each node removes the previous radmonly chosen top 1 node's BSs from its CBN-Array and then use 00 to get a next set of candidate top 1 nodes. And Stop when the selected nodes at present time cooperate with the previous selected nodes can achieve a set of nodes whose CBN-Arrays include all BNs' BSs. Else, move to 01
02 Else, move to 00
03 End

Give a vivid example:

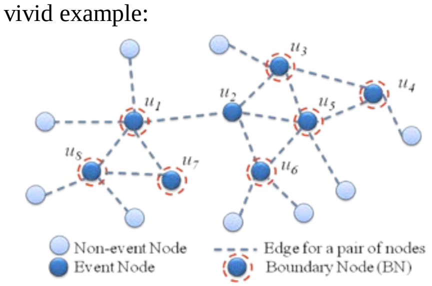


Fig. 3 An example for RN Selection Algorithm

As illustrated in Fig.3, u_1 ,u_3 ,u_4 ,u_5 ,u_6 ,u_7 ,u_8 are BNs with a dotted red circle.

Table 1: CBN-Arrays of {u_1 ,u_3 ,u_4 ,u_5 ,u_6 ,u_7 ,u_8 }

	u_1	u_2	u_3	u_4	u_5	u_6	u_7	u_8
u_1	1	-1	0	0	0	0	1	1
u_2	1	-1	1	0	1	1	0	0
u_3	0	-1	1	1	1	0	0	0
u_4	0	0	1	1	1	0	0	0
u_5	0	-1	1	1	1	1	0	0
u_6	0	-1	0	0	1	1	0	0
u_7	1	0	0	0	0	0	1	1
u_8	1	0	0	0	0	0	1	1

00 get u_2 and u_5 as as top 1s with 4 *BS* values in its CBN-Array.

Possibility 1: if randomly select u_2, a set of next top 1 nodes should be{u_3 ,u_4 ,u_5}or { u_7 ,u_8} because we still need BS_6 BS_7 BS_8. It's obvious that we need more than 3 RNs in this case.

Possibility 2: if randomly select u_5, {u_1 ,u_7 ,u_8} to be a set of next top 1s is the only solution. This case we need 2 RNs.

Finally, we get RNs { u_5 ,u_1}or{ u_5 ,u_7}or{ u_5 ,u_8} as a solution to this example.

4.2 EUCOW Algorithm

In this section, we present our proposed algorithm.

EUCOW Algorithm

/* when the events happen */
00 genNet
01 genEvent, else move to 00
02 BN Selection Algorithm, else move to 01
03 RN Selection Algorithm, else move to 02
04 RNs report to the sink
05 The sink updates the boundary information
06 End

However, we don't deny inevitable collision and unexpected errors existing.

4.3 Performance Analysis

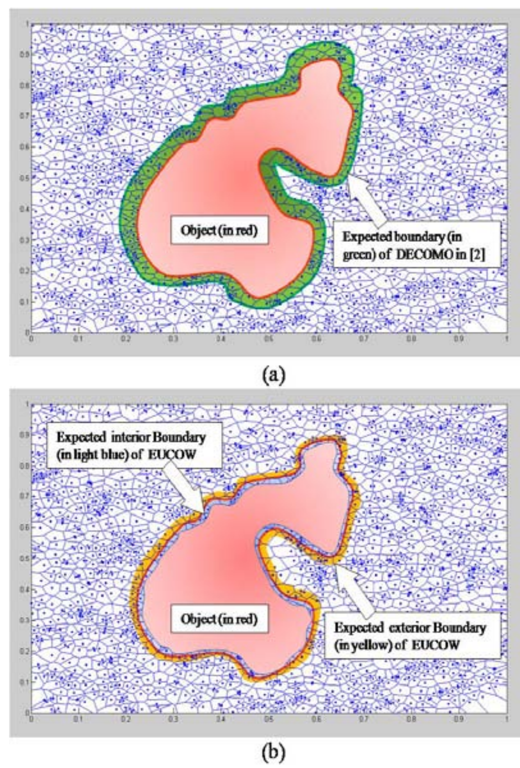


Fig. 4 Comparatively study on expected boundaries based on Voronoi network topology

To measure and comparatively study on the precision of expected boundary, intuitively the more sensor nodes deployed, the better results we can get. We quote a definition of *boundary point* that actually a centroid of RN and *expected boundary line* that connects all boundary points [2].

As shown in fig.4(a), the pink area indicates the object while the thick curved shape in green is the expected boundary of DECOMO. Fig.4(b) shows that our EUCOW can accurately localize both expected interior boundary in light blue and exterior boundary in yellow. Since we only select the nodes within one hop distance to the boundary line, we achieve extremely thinner boundary than DECOMO in the form of both interior and exterior boundary.

To gain some intuitive understanding about how many BNs and RNs will be selected and how to efficiently set the parameters in our voronoi cell based wireless sensor network that our proposed algorithm surpass DECOMO[2] in boundary accuracy and energy efficiency, we statistically analyze the performance. For our analysis, we assume that we have enough sensor nodes, hundreds and thousands nodes deployed in a voronoi cell based network so that we can ideally consider it as a uniform dense area with a density ρ sensors per unit area. And also we suppose that S_g denotes the boundary area in DECOMO, S_b denotes the interior boundary area and S_y denotes the exterior boundary area in our proposed algorithm. we define our boundary area as the sum of S_b and S_y.Therefore the expected number of BNs in DECOMO and EUCOW can be expressed as follows:

$$E_{DECOMO}[BN] = \rho S_g \quad (1)$$

$$E_{EUCOW}[BN] = \rho S_y + \rho S_b = (S_b + S_y)\rho \quad (2)$$

Where $E_{DECOMO}[BN]$ indicates the expected number of BNs in COBOM algorithm and E_{EUCOW} represents the expected number of BNs within range r respectively in DEMOCO algorithm. Since we restrict BNs should be selected within r distance far from boundary line, usually we get $S_b + S_y \ll S_g$ so *(2)* $\ll$ *(1)*. Therefore, we can make a conclusion that our proposed algorithm selects much less BNs than DECOMO without decreasing boundary accuracy when numerous objects emerge except for network is of low density.

V. Simulation and Results

In this section, we develop a simulator using VC++ to evaluate the performance of DECOMO [10] and our proposed algorithm. Unlike DECOMO algorithm, we don't consider the smoothed moving objects, in contrast, we fully focusing on unsmoothed objects. The possible data loss and contention are not concerned. Our simulation results will experientially verify that EUCOW can achieve a thinner accurate boundary of object as well as energy efficiency due to relatively less number of BNs and RNs. we carry out each simulation 100 times. The parameters for the simulations are shown in table 2.

Table 2: Simulation parameters

Parameter	***Value***
Area Size	500*500
Quantity of sensor	10000(dense)
Communication Range	25*m*
Total reporting time	10 times
Total time slots/ simulation	120
sensing and reporting periodicity	3 time slots
Total simulation times	100 (times)
Increase an event time by 3 time slots (initial value:21 time slots)	

In our simulation, sensor nodes are arbitrarily distributed over a 500*500 field. Since we are focusing on the precision of object boundary, in each simulation, we deploy 10000 sensor nodes to simulate a dense setting. The communication range is set to be 25m. All sensor nodes are activated and make local observations every 3 time slots. An *event time* denotes the time period that an event lasts for in a single simulation trial.

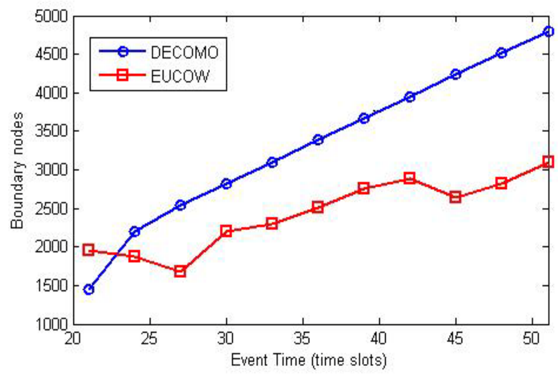


Fig. 5 Comparing the number of selected BNs

As shown in Figure 5, our proposed EUCOW performs better than DECOMO in selecting BNs. At the beginning time, almost all sensor nodes are BNs for temporary we get nearly worse performance. With the increasing of event nodes, objects are shaped and growing. There is a very interesting phenomenon that when we adjust event time to be 30 time slots, the BNs suddenly increased. The reason should that some objects emerged. Moreover, we get a similar phenomenon when event time is at a value of 45 time slots.

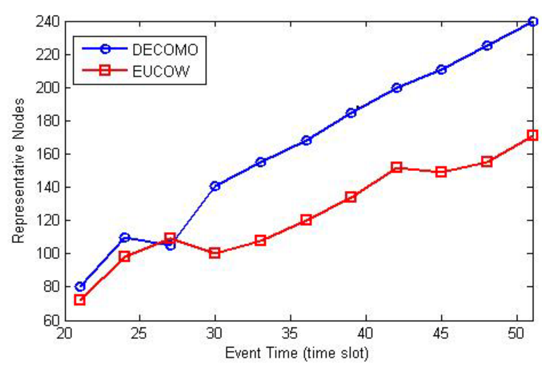


Fig. 6 comparing the number of selected RNs

Figure.6 shows that the difference of the number of selected RNs in DECOMO and EUCOW almost the same when the value of event time is 21, 24 and 27 since the event regions slowly expend, no emerged objects. As the event region grows, the number of RNs in our proposed Algorithm might be of high possibility to surpass that in DECOMO for theoretically we need more RNs to cover all the BNs in a large scale. Fortunately, some objects emerged to greatly reduce the number of BNs as well as RNs. Meanwhile, we get less report messages so as to achieve energy efficiency and prolong the network lifetime.

VI. Conclusions and Future work

In this paper, our proposed EUCOW algorithm achieves two significant things: 1) our genEvent algorithm successfully generates the randomly happened events. 2) we use BN selection algorithm to simply and efficiently select BNs in the form of interior and exterior BNs. Actually, we got some errors and data deviations in evaluating the performance of our proposed algorithm, but it's not serious to boundary accuracy and BNs, RNs selection. One limitation is that some mechanisms of our system should be directly implemented on normal sensor nodes. For our future work, we are going to concentrate on boundary monitoring in spatial wireless sensor network.

Acknowledge

This research was supported by Waseda University Global COE Program International Research and Education Center for Ambient SoC sponsored by MEXT, Japan.

References

[1] X. Ji, H. Zha, J. J. Metzner, and G. Kesidis, "Dynamic Cluster Structure for Object Detection and Tracking in Wireless Ad-Hoc Sensor Networks", Proc. IEEE ICC, 2004.

[2] J.H.Kim, K.B KIM, S.H.CHAUHDARY, "DECOMO:Energy-Efficient Detection and Monitoring for Continuous Objects in Wireless Sensor Networks", IEICE TRANS.COMMUN. VOL.E91–B, NO.11 NOVEMBER 2008

[3] J. Singh, U. Madhow, R. Kumar, and S. Suri, and R. Cagley, "Tracking Multiple Targets Using Binary Proximity Sensors," Proc. IPSN, 2007.

[4] J. Jeong, T. Hwang, T. He, and D. Du, "MCTA: Target Tracking Algorithm based on Minimal Contour in Wireless Sensor Networks", Proc. IEEE INFOCOM, 2007.

[5] Xue.W, Sheng.W,D.W.Bi and J.J.Ma, "Distributed Peer-to-Peer Target Tracking in Wireless Sensor Networks", sensors 2007 by MDPI.

[6] N. Bulusu, J. Heidemann, and D. Estrin, "GPS-less low cost outdoor location for very small devices", Proc. IEEE Pers. Commun. (Special Issue on Smart Space and Environments), vol. 7, pp. 28-34, Oct. 2000.

[7] K. Chintalapudi and R. Govindan, "Localized Edge Detection In Sensor Fields", Proc. Ad-hoc Networks Journal, pp. 273-291, 2003.

[8] J. Liu, P. Cheung, L. Guonidas, and F. Zho, "A dual-space approach to tracking and sensor management in wireless sensor networks", Proc. ACM International Workshop on Wireless Sensor Networks and Applications Workshop, 2002.

[9] C. Zhong and M. Worboys, "Energy-Efficient Continuous Boundary Monitoring in Sensor Networks", Technical Report, 2007. http://www.spatial.maine.edu/~czhong/boundary_monitoring.pdf

[10] US Naval Observatory (USNO) GPS Operations, 2001 Available: http://tycho.usno.navy.mil/gps.html

[11] N. Bulusu, J. Heidemann, and D. Estrin, "GPS-less low cost outdoor location for very small devices", Proc. IEEE Pers. Commun. (Special Issue on Smart Space and Environments), vol. 7, pp. 28-34, Oct. 2000.

[12] P.-K. Liao, M.-K. Chang, and C.-C. J. Kuo, "Distributed Edge Detection with Composite Hypothesis Test in Wireless Sensor Networks", Proc. IEEE, Communication Society Globecom, 2004.

[13] Atsuyuki Okabe,Atsuo Suzuki "Locational optimization problems solved through voronoi diagrams"European Journal of Operational Research 98, 1997