

# Spatial Query Processing Based on Minimum Bounding in Wireless Sensor Networks

Sun Ok Yang\* and SungSuk Kim\*\*

**Abstract:** Sensors are deployed to gather physical, environmental data in sensor networks. Depending on scenarios, it is often assumed that it is difficult for batteries to be recharged or exchanged in sensors. Thus, sensors should be able to process users' queries in an energy-efficient manner. This paper proposes a spatial query processing scheme- Minimum Bounding Area Based Scheme. This scheme has a purpose to decrease the number of outgoing messages during query processing. To do that, each sensor has to maintain some partial information locally about the locations of descendent nodes.

In the initial setup phase, the routing path is established. Each child node delivers to its parent node the location information including itself and all of its descendent nodes. A parent node has to maintain several minimum bounding boxes per child node. This scheme can reduce unnecessary message propagations for query processing. Finally, the experimental results show the effectiveness of the proposed scheme.

**Keywords:** Notification Message, Parent Selection Message, Spatial Query Process, Minimum Bounding Area

## 1. Introduction

Sensors are deployed to gather physical data for a variety of purposes such as environmental monitoring, surveillance or agriculture in sensor networks [1]. Each sensor can reach its neighbors via wireless channel and do simple computation locally. In the environments, there are many kinds of applications. One of them is to gather useful environmental-related information within user interesting area; that is, "get the highest temperature in the region within (x1, y1, x2, y2)".

Depending on the applications, it operates by use of a limited, small-size battery. It is often assumed that it is difficult for batteries to be recharged or exchanged in sensors [2, 3]. Our main concerns are related with energy-efficient usage. A message transmission ( $P_{send}$ ) consumes much more energy than local computation or a message reception ( $P_{receive}$ ). Energy-usage quantity in message transmission is inversely proportional to distance ( $r$ ) [4] as follows:

$$P_{receive} \propto \frac{P_{send}}{r^\alpha} \quad (1)$$

Therefore, the algorithm to decrease the number of message propagations is one of the viable solutions to use energy more efficiently in sensor networks. There have been several research efforts to reduce unnecessary messages by storing some information in local memory.

Spatial query gets some properties of the target nodes within the submitted query region. Spatial databases maintain location information about nodes. Several kinds of index structures have been developed to deal with location information, and R-tree is one of them. Each node maintains a Minimum Bounding Box (MBB) including all its descendant nodes, which is hierarchically structured [4]. In this way, R-tree supports several types of spatial queries. However, a root node has to maintain all the information (centralized server). Inaccurate information in MBBs incurs unnecessary message propagation during query processing. It is bad to adapt the index to wireless sensor networks. Thus, another algorithm is needed in the environments to improve the accuracy of local information and maintain it at low costs comparatively.

This paper proposes a scheme; *Minimum Bounding Area based Scheme (MBAS)* for energy-efficient spatial query processing in wireless sensor networks. In the initial setup, some useful information is exchanged between the parent and child node. Each sensor node can maintain partial location information for all its descendant sensors. *MBAS* is similar to *SPatial IndeX (SPIX)* structure [5] except that in *MBAS* a parent node maintains MBBs per child node to

Manuscript received August 28, 2009; accepted September 14, 2009.

**Corresponding Author: SungSuk Kim**

\* Dept. of Computer Engineering, Sejong University, Seoul, Korea (soyang@sejong.ac.kr)

\*\* Corresponding Author, Dept. of Computer Engineering, SeoKyeong University, Seoul, Korea (sskim03@skuniv.ac.kr)

increase the information accuracy. During the query process in the scheme, a few sensors based on local information propagate a query message to their child nodes. In this way, the proposed scheme can decrease the number of outgoing messages and thus improve energy efficiency.

The paper is organized as follows. The related work and system model are introduced in Section 2. In Section 3, the routing path setting method and corresponding algorithms are presented. In Section 4, performance analysis and the experimental results are explained. Finally, conclusions are discussed in Section 5.

## 2. Related Works

Lots of works have dealt with the energy issue in query processing. As the interests on sensor networks have increased, many kinds of index methods have been proposed [5-7]. *Minimum Bounded Rectangle (MBR)* based tree has been designed to minimize the number of sensor nodes that have joined the spatial query. *Semantic Routing Tree (SRT)* has been constructed by using the X coordinate value of sensor location and a parent node has stored x coordinate range about its child nodes. *SPIX* has used an index structure based R-tree over the sensor nodes. A parent node has stored one MBR containing all of its child nodes. *SPIX* has been more efficient than *SRT* in terms of query processing. In *SPIX*, query processing algorithm is simple and the exchanged data volume is small. However, those methods generate unnecessary message propagations due to untrustworthy local information about their child nodes.

To reduce communication cost significantly in sensor networks, thus, it has been performed in-network aggregation (e.g., AVG and MIN) [6, 8] or data reduction via wavelets or distributed regression [9-10]. These techniques have not provided the fine data granularity desired by many users. For example, the biologists in the Sonoma Redwoods project have wanted to receive as much detailed data from a sensor network as possible, so that they have tried various physical models and tested various hypotheses over the data.

Approximate query is an effective means to solve problems about prediction and aggregation and so can be another way of that purpose [11-12]. The measurements from real world are often noisy and at a loss. It means that the query of the sensor network is really an approximate query. These methods basically try to increase energy efficiency at the cost of precisions of acquired data.

Finally, in-network aggregation has been related with the algorithm development to query process efficiently a specific event within the sensor network [7, 13]. It is effective

to reduce data transmission. But it needs detailed historical information for applications.

## 3. Minimum Bounding based Scheme (MBAS)

Each sensor maintains partial location information for all of its descendant child sensors in order to reduce the number of outgoing messages in query processing. To do so, a routing path setting is done during initial setup through two phases; Notification Phase and Parent Selection Phase [14]. Before explain each phase, the basic system architecture is first considered as in Fig. 1.

The figure shows that a network is composed of sensors which are scattered in a field. The Sink Node (or Base Node) has enough energy and much higher capabilities unlike the other nodes.

For clarify the model, we assume that sensor networks are as follows:

- Queries delivered from users are propagated via sink node to sensors within the network. And then, it receives the final results which are accomplished by the sensors, and passes it to the users.
- Every sensor has a unique identifier and can know its location.
- Every sensor has the same computation capabilities, communication range, and energy quantity.

In Notification Phase, the routing path setting process is started as a sink node generates and propagates the Notification Message (NM) to neighbor nodes. When a neighbor node receives the NM, it executes Algorithm 1. The neighbor node delivers the message to its neighbor nodes until all nodes receive the NM.

And then, leaf nodes start the Parent Selection Phase after some time; they select their parent node by using NM. They gather location information about themselves and

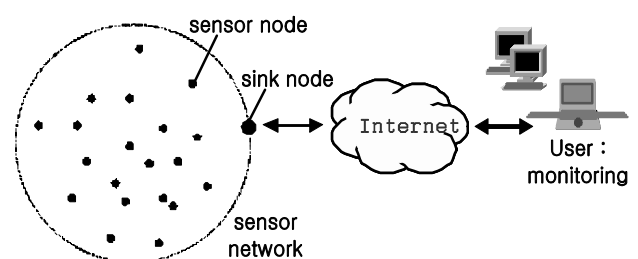


Fig. 1. Architecture of wireless sensor network

**Algorithm 1.** When a sensor  $k$  gets  $NM_i$ 


---

```

//NMj : previous NM if any
//NMi : current NM
//Hops(NMi) : Hops value in NMi
//dist(i, j) : distance between sensor i and j
IF (sensor k already received another NMj) {
  -Compare NMi and NMi-1 hops
  IF (Hops(NMi) > Hops(NMj))
  -Throw away NMj and EXIT
  EndIF
ELSE IF (Hops(NMi) == Hops(NMk)) {
  IF (dist(i, k) ≥ dist(j, k))
  -Throw away NMj and EXIT }
}
// when new parent is selected as sensor i
-New_Hops ← Hops(NMi) + 1
-propagate NMk containing New_Hops, ID, and Loc

```

---

child nodes<sup>‡</sup>. They deliver it to their parent node. This process is repeated until the sink node gets messages from all of its neighbor nodes.

### 3.1 Notification Phase

During the Notification Phase, a node can get one more NMs, so  $NM_i$  is used.  $NM_i$  means that sensor node  $i$  generates NM. NM contains *Hops*, the identifier (*ID*), and the location information (*Loc*) of a sender node. *Hops* means the hop value from a sink node to itself by way of NM delivery path (see Table 1).

If a sensor node  $k$  receives  $NM_i$  from its neighbor sensor  $i$ , it first checks whether it already got another  $NM_j$  ( $i \neq j$ ) or not as shown in Algorithm 1.

When current  $NM_i$  is the first one, sensor  $k$  just determines that nodes  $i$  is its parent node. Sensor  $k$  makes  $NM_k$  containing the information about itself and propagates the message to its neighbors; if sensor  $k$  already got one more NMs, it has to determine which node can be its parent node according to *Hops* and distance between the sender and itself. A sender node with lower *Hops* can be a parent node since it needs fewer messages to communicate with the sink node via the parent node. If two NMs have the same *Hops*, node  $k$  selects a closer node by using *Loc*. After this examination, if new node  $i$  becomes the parent node, node  $k$  makes new  $NM_k$  and propagates it.

Every sensor node can know when to start the Parent Selection Phase by using *Hops*; that is, if  $\max_{hop}$ , maximum

**Table 1.** Notification Message (NM)

symbol	mean	initial value
Hops	the number of hops from sink node	0
ID	identifier of NM sender	sink node's identifier
Loc	location information of NM sender	sink node's location

hops between the sink node and a leaf node, is a constant and it is notified to all nodes, each sensor can calculate the maximum waiting time ( $w_t$ ) as follows:

$$w_t \leftarrow (\max_{hop} - Hops_i) \times t \times 2 \quad (2)$$

Here,  $Hops_i$  is the lowest *Hops* which sensor  $i$  made during Notification Phase. Constant  $t$  is the time which takes to process the NM at each sensor. After the sink node must wait for  $\max_{hop} \times t \times 2$ , it will finish receiving the Parent Selection Message from all its child nodes during Parent Selection Phase; every node has to send the message to its parent node (see the next section).

If a sensor cannot receive a parent selection message from other sensors after  $w_t$ , it determines itself as a leaf node and starts the Parent Selection Phase.

### 3.2 Parent Selection Phase

In the Parent Selection Phase, each sensor notifies its *ID*, *Loc* and relevant information to a parent sensor by using the parent node's *ID*. Each sensor maintains *Minimum Bounding Area (MBA)* which includes all children and descendant nodes. Thus, it sends its parent sensor *Parent Selection Message (PM)*. Table 2 shows the contents of *PM*.

**Table 2.** Parent Selection Message (PM)

symbol	mean
ID <sub>parent</sub>	identifier of parent node
ID	identifier of itself
Loc	location information of itself
MBA	Minimum Bounding Area which includes itself and all its descendant nodes
Hop <sub>max</sub>	Maximum value among all Hops which children maintain

<sup>‡</sup> Of course, leaf node has no child node and thus it is set to null

In the case of leaf nodes, *MBA* is NULL since it has no child and  $Hop_{max}$  is just the smallest *Hops* value in NMs, which it receives from its parent sensors. When a parent node ( $S_{parent}$ ) gets a PM from a neighbor sensor node ( $S_{child}$ ),  $S_{parent}$  determines  $S_{child}$  as its child node and stores *ID*, *Loc*, *MBA*, and  $Hop_{max}$  for  $S_{child}$ . If  $S_{parent}$  gets another PMs from other child nodes before  $S_{parent}$  sends PM to its parent, it does the same job; it means that  $S_{parent}$  stores related information separately per child. When another node gets a PM from  $S_{parent}$ , it does the same process as done in  $S_{parent}$ . PM delivery is repeated one by one until sink processes it.

Fig. 2 shows an example of the Parent Selection Phase in *MBAS*. In Fig.2-(a), leaf nodes send PMs to their parent nodes after  $w_t$ . In Fig.2-(b), after  $s_1$ ,  $s_2$ , and  $s_3$  get one or more PMs from their child nodes, they send their parent sensor  $s_4$  new PMs, which has computed new *MBA* (see the rectangles in Fig.2-(b)), *Loc* of its children, and  $Hop_{max}$  (the highest value among child nodes). And then,  $s_4$  stores each PM locally. It calculates one *MBA*, which includes itself and all of its descendants, and adds the *MBA* to its PM. Finally,  $s_4$  sends the PM to its parent sensor. Like this, since a node sends *MBA*, its parent sensor will be able to grasp region information more accurately about descendent nodes.

From here, nodes can estimate the maximum waiting time ( $w_t$ ) shown in equation (2), since each child maintains  $Hop_{max}$ . Therefore partial aggregation is done naturally.

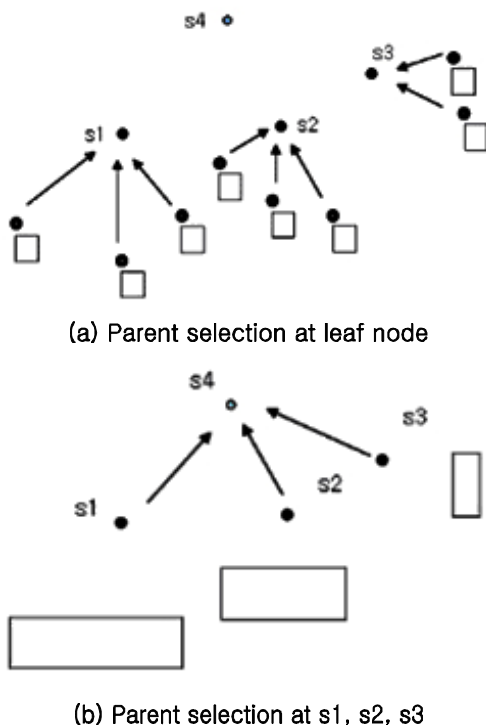


Fig. 2. Examples of parent selection phase

### 3.3 Spatial Query Processing

Before  $S_{parent}$  propagates a spatial query with a region to  $S_{child}$ ,  $S_{parent}$  decides whether the query region is overlapped with any *MBA*. If so, it has to propagate the query and wait for the result during  $w_t$ ; otherwise, it knows that no descendant node can be contained in the query region and thus, it will not propagate the message. In this way, simple aggregation for query results can be applied along the routing path according to query types.

In *MBAS*, Each sensor stores locally more information about its child nodes, but exchanged data size during the initial setup is a little more than that in *SPIX*<sup>§</sup>.

## 4. Performance Analysis

Our main concerns in this paper are related with spatial query processing and energy efficiency in sensor networks. To evaluate the performance metric, *SPIX* [5] is selected as a counterpart algorithm. The reason is that the architecture is almost the same as that in ours and it deals with spatial query processing. In *SPIX*, however, every sensor locally maintains only one *MBA* including all descendants. It worsens the accuracy of the information, especially when the sensor is located closer to sink node. This is because the size of *MBA* is inversely proportional to the distance to sink node. If the local information is inaccurate, unnecessary sensors have to join the query processing and thus energy efficiency gets worse. Of course, if the local information size is too heavy, it increases overall maintenance cost and worsens message-related performance. Thus, it is reasonable to uphold the balance of energy efficiency and local data size.

### 4.1 Simulation Model

To show the effects mentioned above, extensive experiments have been done. We make a simulator by using Java on a Linux machine (Pentium Dual Core 3.0 GHz, 2Gbyte memory, RedHat Linux 9.0)(see Fig. 3).

In the experiments, total network area is set to  $512 \times 512$  and there are  $N$  sensors operating. It is assumed that the communication range of each sensor is 5. The Sink node is located in the center of the network. In the initial setup phase, sensors locations are generated according to a skewed or random (i.e. uniform) distribution pattern. In the case of skewed distribution, a dense area is 1/3 of the whole area and the number of sensors in that area is three

<sup>§</sup> this issue will be discussed in detail at Section 5

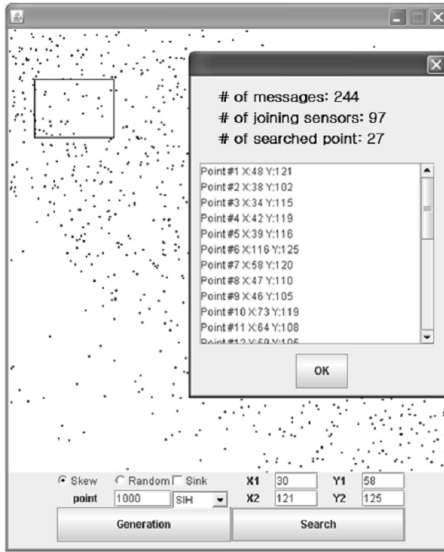


Fig. 3. Simulator for experiment

times more than that in the sparse area. Users forward queries to the network via sink node as the format of  $(x_1, y_1, x_2, y_2)$ .

At first, the maintenance cost for the initial setup phase is evaluated to calculate the local information size. And then, the effects on various parameters -the number of sensors, sensor locations, and query size- are compared with two schemes.

#### 4.2 Initial Setup Cost

In this experiment, the main factor is the number of sensor nodes ( $N_{\text{sensor}}$ ). The maintenance cost is calculated by considering  $N_{\text{sensor}}$  and the delivered data size during the initial setup phase. The proposed scheme tries to enhance query processing based on a higher volume of local information for descendent nodes than *SPIX*. Thus, as there are more sensors in the network, the local information size increases.

In *SPIX*, each node delivers a MBA and the parent node also maintains a MBA. Thus, local information size ( $\text{Size}(SPIX)$ ) is as follows:

$$\text{Size}(SPIX) \leftarrow S_{MBA} \quad (3)$$

Here,  $S_{MBA}$  means the size of a MBA. In *MBAS*, each sensor has to propagate ID, Loc, MBA, and  $Hop_{max}$  to its parent. The parent maintains the information per child node. If a node has many child nodes, local information size is proportional to the number. Local information size is as follows:

$$\text{Size}(MBAS) \leftarrow S_{\text{sensor}} \times (S_{MBA} + S_{ID}) + \alpha \quad (4)$$

Here,  $S_{ID}$  is the size of a sensor identifier, and  $\alpha$  means the remaining data size and can be negligible.

#### 4.3 The Results for Query Processing

The proposed algorithm and *SPIX* can process spatial queries correctly; that is, if spatial query is delivered, all sensors in the region can be found and responded to. Therefore, the number of propagated messages is measured during query processing. The results are shown separately according to the uniform and skewed distribution. In skewed distribution, two kinds of experiments are done whether the query region is mainly located in a dense or sparse area.

**Effects on the distance between sink node and query region:** As the distance between the sink node and query region varied, the number of messages were measured in uniform and skewed distribution. Figure 4 shows the results when the number of total sensors (= 2000) and size of

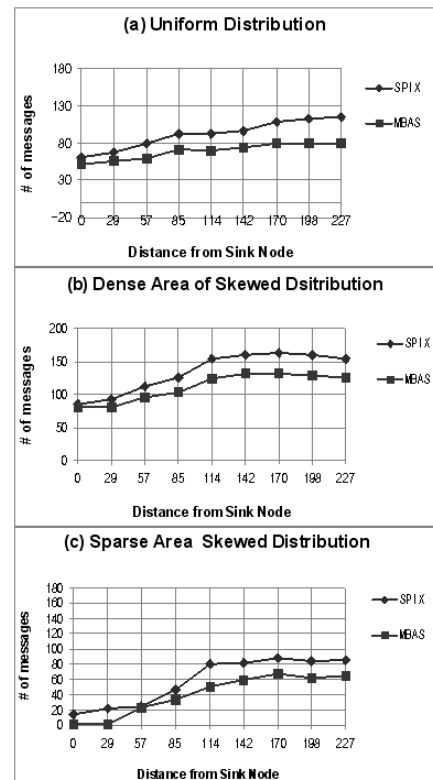


Fig. 4. The case that distance between sink node and query region varied (number of total sensors =2000, query region=2.5%)

the query region (= 2.5% of overall area) were fixed.

In uniform distribution (Fig. 4-(a)), *SPIX* and *MBAS* show similar performance patterns, since the local information and query processing mechanism is almost same. However, *MBAS* maintains a little more information locally, which can reduce unnecessary message propagations. On average, 25% more of messages are reduced in *MBAS* than that in *SPIX*.

Figures 4-(b) and 4-(c) show results executed in skewed distribution but the query region location differs. When users want to process a query in a dense area, the overall pattern is similar to Fig. 4-(a). In the case of a sparse area, however, the number of messages is relatively similar in the two schemes since there are originally only a small number of sensors in the sparse area and so the accuracy of local information is not high.

**Effects on the number of total sensors:** In the second experiment, the number of total sensors varied from 400 to 2800, while the query region was set to 80 away from the sink node and the query size was set to 2.5% of all area.

If many sensors were located in the network area, it could increase the probability of inaccurate local informa-

tion since the sensors are located close to the sink node. This is because of MBA will be larger due to many descendant nodes.

In Fig. 5, the overall patterns are similar to Fig. 4. As more sensors are considered, much more sensors should be participated in processing a query. In Fig.5-(c), however, there are fewer sensors within the sparse area than those in the dense area, so the effects are correspondingly dissimilar.

**Effects on the size of query region:** In the spatial query process, users submit a query region as the format of  $(x_1, y_1, x_2, y_2)$ . When the query region is bigger, it generally needs a large number of sensors to be joined. Several sensors have to deliver unnecessary messages as the size of the query region increases. In Fig.6-(a) and 6-(b), both *MBAS* and *SPIX* using MBA show a considerable increase in energy usage. This is because only two sensors at each corner may generate a big MBA irrespective of the number of sensors within the box. This indicates that a broader MBA may depreciate the accuracy. Since the size is also bigger, the size of intersected area between the MBA and query region is much bigger Therefore many unnecessary sensors have to join the query processing. For example, when a query region is 8% of the total area in uniform distribution,

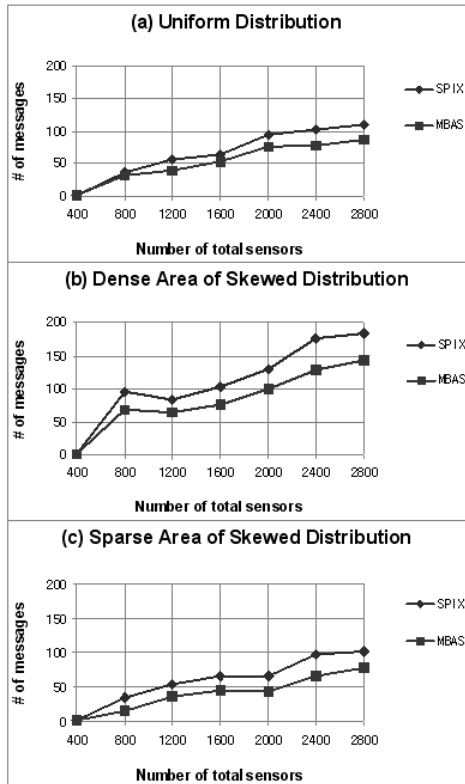


Fig. 5. The case that number of total sensors is varied (query region=2.5%, query region and the distance from sink node is fixed)

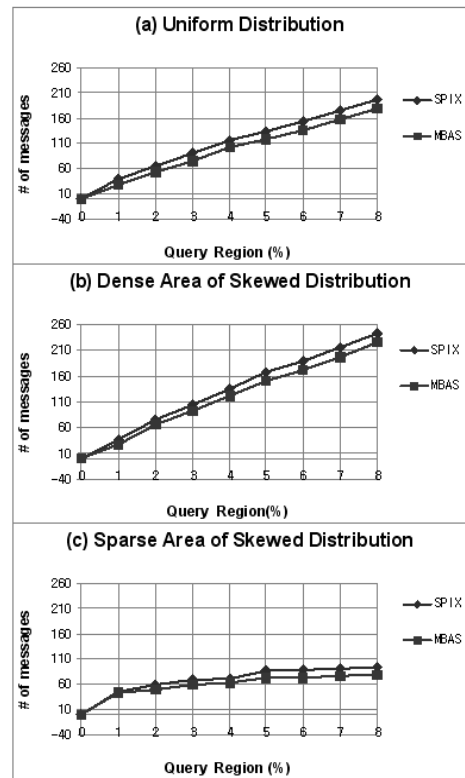


Fig. 6. The case that query region varied (number of total sensors=2000, query region is fixed)

*SPIX* produces 123 more messages than that in *SBS*.

## 5. Conclusion

In this paper, a spatial query processing algorithm (*MBAS*) is proposed to improve the energy efficiency of the sensor. This approach leads to a reduction in the number of unnecessary message propagations by using partial information about the descendant sensors. Of course, if the volume of local information is too large, the cost to exchange or maintain it is not negligible.

Thus, we try to uphold the balance of both conditions during algorithm development. Through extensive experiments, many sensors can avoid disseminating query messages when they and their descendants never lie within the query region.

## References

- [1] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson, "Wireless sensor networks for habitat monitoring", ACM Int'l Workshop on Wireless Sensor Networks and Applications, 2002, pp.88-97.
- [2] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A Survey on Sensor Networks", IEEE Communications Magazine, pp.102-114, 2002
- [3] D.H. Chae, K.H. Han, K.S. An, and S.S. An, "Trend and Technology of Sensor Network", Journal of Korea Information Science, Vol.22, No.12, pp.5-12, 2004
- [4] A. Gutmann, "R-Trees - A dynamic index structure for Spatial Searching", ACM SIGMOD, 1984, pp.47-57.
- [5] A. Soheile, V. Kalogeraki, and D. Gunopulos, "Spatial Queries in Sensor Networks", ACM Symp. on Advances in Geographic Information Systems (GIS), 2005, pp.61-70.
- [6] S. Madden, M.J. Franklin, J.M. Hellerstein, and W. Hong, "Tag: A Tiny Aggregation Service for ad-hoc Sensor Networks", In Proc. Of OSDI, 2002, pp.131-146.
- [7] S. Madden, M.J. Franklin and J.M. Hellerstein, "TinyDB: An Acquisitional Query Processing System for Sensor Network", Journal of ACM Tran. on Database Systems, Vol.30, No.1, 2005, pp.122-173.
- [8] C. Guestrin, P. Bodik, P. Mark, and S. Madden. "Distributed regression: an efficient framework for modeling sensor network data", International Workshop on Information Processing in Sensor Networks (IPSN), 2004, pp.415-418.
- [9] S. Nath, P.B. Gibbons, S. Seshan, and Z. R. Anderson. "Synopsis diffusion for robust aggregation in sensor networks", international conference on Embedded networked sensor systems (SenSys '04), 2004, pp.313-320.
- [10] J. Hellerstein and W. Wang, "Optimization of in-network data reduction", international workshop on Data management for sensor networks: in conjunction with VLDB 2004 (DMSN '04), 2002, pp.321-332.
- [11] D. Chu, A. Deshpande, J.M. Hellerstein and W. Hong, "Approximate Data Collection in Sensor Networks using Probabilistic Models", International Conference on Data Engineering(ICDE), 2006, pp.48.
- [12] H. Jin, Q. Ren, J. Li and Y. Shi, "An Approximate Query Processing Method based on Data Correlation in Sensor Networks", International Conference on Embedded Software and Systems(ICESS), 2008, pp. 13-18.
- [13] A. Deligiannakis, Y. Kotidis, and N. Roussopoulos, "Hierarchical in Network Data Aggregation with Quality Guarantees," In Proc. Of EDBT, 2004, pp.658-675.
- [14] X. Yang, H.B. Lim, M.T. su and K.L. Tan, "In-Network Execution of Monitoring Queries in Sensor Networks," ACM SIGMOD, 2007, pp.61-70.



**Sun Ok Yang**

She received her MS and Ph.D. degrees in Computer Science and Engineering from Korea University, Seoul, South Korea in 2002 and 2006, respectively. From 2006 to 2007 she was a research professor at the Institute of Information and Communications, Korea University, Seoul, Korea. In 2008 she was a research professor at the School of Electrical & Electronic Engineering, Yonsei University, Seoul, Korea. She is currently a visiting professor at the college of Electronics & Information Engineering, Sejong University, Seoul, Korea. Her research interests include mobility management and QoS provision issues in the mobile/sensor networks.

**SungSuk Kim**

He received his MS and Ph.D. degrees in Computer Science and Engineering from Korea University, Seoul, South Korea, in 1999 and 2003, respectively. Currently, he is an assistant professor in the department of Computer Science at Seokyeong University. His primary research interests lie in the areas of mobile/pervasive computing, data management and spatial databases. He is a member of the ACM and IEEE computer society.