Semijoin-Based Spatial Join Processing in Multiple Sensor Networks

Min Soo Kim, Ju Wan Kim, and Myoung Ho Kim

ABSTRACT—This paper presents an energy-efficient spatial join algorithm for multiple sensor networks employing a spatial semijoin strategy. For optimization of the algorithm, we propose a GR-tree index and a grid-ID-based spatial approximation method, which are unique to sensor networks. The GR-tree is a distributed spatial index over the sensor nodes, which efficiently prunes away the nodes that will not participate in a spatial join result. The grid-ID-based approximation provides great reduction in communication cost by approximating many spatial objects in simpler forms. Our experiments demonstrate that the algorithm outperforms existing methods in reducing energy consumption at the nodes.

Keywords—Sensor network, spatial join, spatial index.

I. Introduction

Recently, researchers have been interested in spatial queries in which a sensor network is queried by location of the sensor nodes [1]. Complex spatial queries can be used to answer a request such as "From chemical and environmental sensor networks, find pairs of chemical and environmental sensor nodes where the distance between two nodes of each network is within 10 m, the CO density of the chemical network is larger than 10 ppm, and humidity of the environmental network is larger than 60%," which includes distributed spatial join processing that materializes spatial relationships between two networks. While such spatial join processing has been extensively studied in conventional spatial databases, it is no longer appropriate to adapt existing spatial join algorithms [2] to sensor networks, due to battery and communication restrictions of sensor nodes. Therefore, there have been recent works proposing a distributed spatial index (SPIX) to process spatial queries in a distributed fashion [1], and energy-efficient spatial join algorithms for an in-network evaluation of a spatial join query having both spatial and selection predicates [3]. However, none consider complex spatial join queries that run in multiple sensor networks.

In this paper, we propose an energy-efficient spatial join algorithm for multiple sensor networks that employs a spatial semijoin strategy. The algorithm may run using a distributed spatial index and a spatial approximation method as a possible optimization technique. Therefore, we additionally propose an energy-efficient grid-based rectangle tree (GR-tree) index and a grid-ID-based spatial approximation method. Experimental results show that our algorithm is more effective in reducing energy consumption than others.

Manuscript received July 11, 2008; revised Oct. 5, 2008; accepted Oct. 22, 2008.

This work was supported by the grant (07KLSGC05) from Cutting-edge Urban Development – Korean Land Spatialization Research Project funded by MLTM.

Min Soo Kim (phone: + 82 42 860 5566, email: minsoo@etri.re.kr) and Ju Wan Kim (email: juwan@etri.re.kr) are with the IT Convergence Technology Research Laboratory, ETRI, Daejeon, Rep. of Korea.

Myoung Ho Kim (email: mhkim@dbserver.kaist.ac.kr) is with the Division of Computer Science, KAIST, Daejeon, Rep. of Korea.

II. 2-SN Spatial Join Algorithm

In this paper, a distributed spatial join query for two sensor networks is defined as follows.

Definition 1: 2-SN spatial join query (*q*). Let SN_1 and SN_2 be two sensor networks. Each sensor node $n \in SN_i$ has its spatial location $n.l$ and a set $n.a$ of sensed attributes (such as, temperature, humidity, CO, and so on). The q is defined by $q=\{<n, m> \mid n \in SN_1, m \in SN_2$, where $f_1(n.a), f_2(m.a)$, and $g(n.l, m.l)$ are satisfied$\}$. Here, f_1 and f_2 are selection predicates on attributes $n.a$ and $m.a$, respectively, and g is a spatial join predicate that determines whether the spatial relationship between two sensor nodes is true. The previous example query can be expressed by $q_e=\{<n.location, n.CO, m.location, m.humidity> \mid n \in SN_1, m \in SN_2,$

where $n.CO>10$, $m.humidity>60$, and $distance(n.location, m.location) < 10\}$.

A straightforward approach to performing q is to transmit all the data of SN_1 and SN_2 to a server, where the spatial join is performed using a nested-loop join algorithm. This approach, though simple, incurs a high transmission cost in sensor networks. Therefore, we propose a new spatial join algorithm based on a spatial semijoin [2] to reduce the transmission cost.

Using the spatial semijoin, q can be performed in five steps. First, $f_1(n.a)$ is evaluated at SN_1 and the resultant set r_1 ($=\sigma_{f1(n.a)}(SN_1)$) is transmitted to a server. Second, r_1 is projected on the spatial join attribute $n.l$, and the resultant spatial point set p_1 ($=\Pi_{n.l}(r_1)$) is transformed into a two-dimensional spatial object set (t_1), which includes the spatial join predicate $g(n.l, m.l)$. For the $distance$ ($n.location$, $m.location$) < 10, the resultant points are transformed into circle objects that have a radius of 10 from the points. Third, t_1 is spatially approximated into a simpler form such as a grid ID. Then, the approximated set (sa_1) and $f_2(m.a)$ are transmitted to SN_2. Fourth, the spatial semijoin between $m.l$ and sa_1 and the $f_2(m.a)$ is performed to produce r_2, which is transmitted to the server. Finally, the refinement phase of the spatial predicate using r_1 and r_2 is performed at the server to produce the final result.

However, the semijoin approach cannot be readily applied to the 2-SN spatial join, due to the following considerations that are unique to sensor networks. First, when transmitting sa_1 in a packet form to sensor nodes, the packet size has a great effect on the performance of the spatial join processing. If the packet size is large enough to hold the entire sa_1, its error rate may be exponentially increased. For example, the error rate of a packet in sensor networking may be $1-(1-p)^n$, where p represents the error rate of a bit, and n represents the number of bits composing the packet. If the packet size is small, it may cause a repetitive transmission of small packets, which incurs a high transmission cost. Second, we have to consider minimization of the hop count because the transmission hop count is the most important aspect of energy efficiency at the nodes.

1. Grid-ID-Based Spatial Approximation Method

Based on the first consideration, we present an effective spatial approximation method to form a minimized sa_1. Figure 1(a) shows the most general minimum bounding rectangle (MBR)-based approximation (MA) method, and Fig. 1(b) shows our grid-ID-based approximation (GA) method using grid IDs for the same query q_e. MA maps each object of t_1 to its MBR, while GA maps each object of t_1 to its grid ID. For example, Fig. 1(a) shows eighteen rectangles, and Fig. 1(b) shows four grid IDs. The transmission of simple grid IDs is definitely less costly than transmission of rectangles in terms of packet size and count because MA needs 16 bytes (4 bytes × 4) for a MBR while GA need 4 bytes for a grid ID.

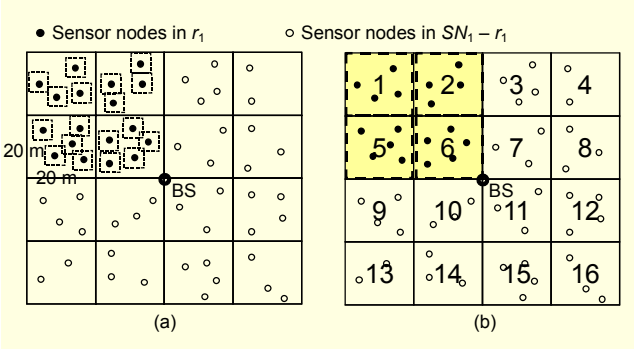

Fig. 1. Spatial approximation set sa_1 for the query q_e: (a) MBR-based approximation and (b) grid ID-based approximation = {1, 2, 5, 6}.

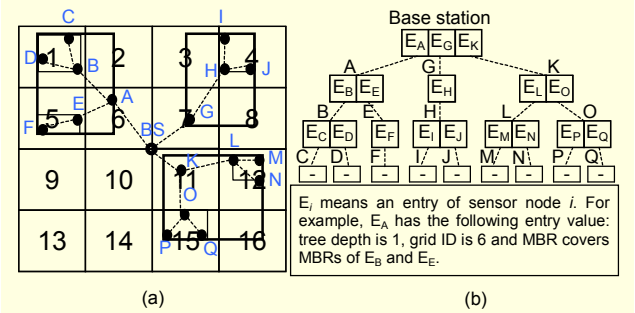

Fig. 2. Example of a GR-tree: (a) distribution of sensor nodes and their grid IDs and (b) the GR-tree index structure.

2. Grid-Based Rectangle Tree (GR-tree) Index

Based on the second consideration, we present the GR-tree for a spatial semijoin. The GR-tree allows each node to efficiently determine whether it needs to participate in a given spatial query. Each sensor node in the GR-tree maintains an entry set for the child nodes below it, which consists of tree depth, grid ID, and MBR. The MBR covers the current node itself and the children. When a node receives a query composed of sa_1 and $f_2(m.a)$, it first intersects the grid areas of sa_1 to its location $m.l$, and if the intersection is not empty, it applies $f_2(m.a)$. Second, it intersects the grid areas to the MBRs of its entry set, and if an intersection exists, it forwards the subset of sa_1 and $f_2(m.a)$ to its children. Figure 2 shows an example GR-tree which was built using an advertisement and a parent selection phase. In this paper, we assume that each node is location-aware and has the same communication range d.

The advertisement phase begins at the base station, and each node waits to receive an advertisement message before it advertises itself to other nodes. The message includes the location of the base station and the advertiser. Each node

calculates its grid ID and depth, and maintains all advertisement messages. In order to avoid a circular network, mutual advertisements between two nodes are disallowed. Finally, each node may have candidate parents and children.

The parent selection starts from a node that has no children. If a node has candidate children, it waits until they select their parent. The GR-tree adopts two optimization techniques for the parent selection criterion. First, a node selects a parent that is less than its tree depth and has a minimum depth within the same grid. If there is no candidate in the same grid, it selects a parent in adjacent extended grids. Second, if there are two or more candidates with the same minimum depth, the node may select a candidate as a parent, where the distance between the candidates and the base station is minimal. These techniques can make the tree depth lower and the MBR smaller.

The implementation complexity of this method is almost equal to that of the SPIX, because this method only has to perform several comparison operators instead of calculating and comparing the MBR area or perimeter as in the SPIX.

III. Evaluation and Conclusion

We evaluated the performance of the GR-tree in comparison with the SPIX and the performance of the GA compared with the MA using the evaluation parameters shown in Table 1.

Figure 3 shows that the GR-tree is always more efficient than the SPIX, irrespective of the number of nodes and the distribution because the GR-tree provides more efficient spatial filtering performance than the SPIX.

Figure 4(a) shows that MA is more efficient than GA under random distribution of query objects because most sensor nodes are more likely to be included in the approximated grids. However, as shown in Fig. 4(b), GA is more efficient than MA under biased distribution if the packet size is small (<512 bytes), which is more realistic.

Table 1. Evaluation parameters.

Parameters	Evaluation enviroments
Sensor nodes	1,000-10,000 nodes with 100 m communication range are randomly distributed in an area of 2,000 m × 2,000 m
Communication cost	Hop count consumed to process sptial queries among sensor nodes
No. of query objects for a spatial semijoin	100 spatial objects 50 m in radius are in random or biased distribution in a sensor network
Packet size	32 to 2,048 bytes. Default packet size is set to 128 bytes in consideration of the IEEE 802.15.4 spec.

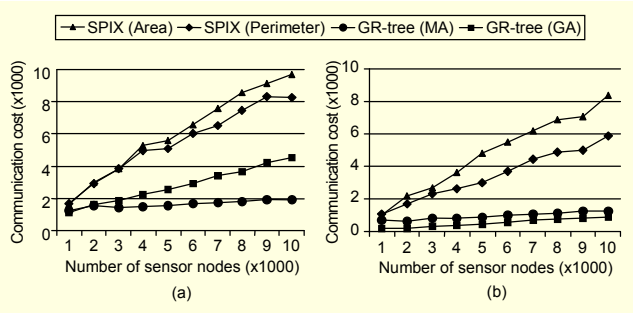

Fig. 3. Comparison of the GR-tree and the SPIX with various numbers of sensor nodes: (a) random distribution and (b) biased distribution of query objects.

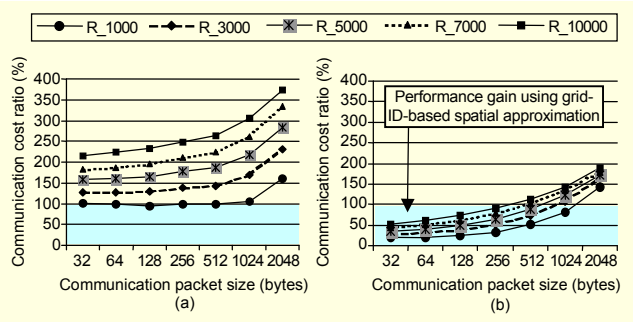

Fig. 4. Communication cost ratios of GA over MA with various packet sizes, where R_1000 to R_10000 mean the number of sensor nodes: (a) random distribution and (b) biased distribution of query objects.

In this paper, we presented a 2-SN spatial join algorithm based on a spatial semijoin strategy. For its optimization, we additionally proposed a GR-tree index and a grid-ID-based spatial approximation method, which greatly reduce the communication cost. We expect our algorithm to be promising in practical use where query objects are especially in biased distribution. We plan to improve our algorithm to process periodic spatial queries for mobile sensor nodes [4].

References

[1] A. Soheili, V. Kalogeraki, and D. Gunopulos, "Spatial Queries in Sensor Networks," *Proc. 13th ACM GIS*, 2005, pp. 61-70.

[2] K.L. Tan and B.C. Ooi, "Exploiting Spatial Indexes for Semijoin-Based Join Processing in Distributed Spatial Databases," *IEEE Trans. Knowl. and Data Engin.*, vol. 12, no. 6, 2000, pp. 920-937.

[3] M.L. Yiu, N. Mamoulis, and S. Bakiras, "Retrieval of Spatial Join Pattern Instances from Sensor Networks," *19th Int'l Conf. on Scientific and Statistical Database Management*, 2007, pp. 25-34.

[4] S.M. Jang, S.I. Song, and J.S. Yoo, "An Efficient PAB-Based Query Indexing for Processing Continuous Queries on Moving Objects," *ETRI Journal*, vol. 29, no. 5, 2007, pp.691-693.