Efficient Processing of KNN Queries in Wireless Sensor Networks

Jeong-Joon Kim[1], Jeong-Jin Kang[2], Ki-Young Lee[3*], Gyoo-Seok Choi[3],
Yong-Soon Im[5] and Eun-Young Kang[6]

*[1]Department of Computer Science and Information Engineering,
KonKuk University, 1, Hwayang-dong, Gwangjin-gu, Seoul, 143-701, Korea
[2]Department of Information and Communication, Dong Seoul University, 76,
Bokjeong-ro, Sujeong-gu, Seongnam-si, Gyeonggi-do, 461-714, Korea
[3]Department of Medical IT and Marketing, Eulji University, 553, Sanseong-daero,
Sujeong-gu, Seongnam-si, Gyeonggi-do, 461-713, Korea
[4]Department of Computer Science, ChungWoon University
San29, Namjang-ri, Hongseong-eup, Hongseong-gun, Chungnam, 350-701, Korea
[5]Department of Broadcasting Production, Kookje University
56, Janganut-gil, Pyeongtaek-si, Gyeonggi-do, 459-070, Korea
[6]Department of Information and Communication, Dongyang Mirae University
62-160, Gocheok-dong, Guro-gu, Seoul, 152-714, Korea*

*[1]jjkim9@db.konkuk.ac.kr, [2]jjkang@du.ac.kr, [3]kylee@eulji.ac.kr,
[4]lionel@chungwoon.ac.kr, [5]ysim@kookje.ac.kr, [6]eykang@dongyang.ac.kr*

**Corresponding Author: kylee@eulji.ac.kr*

Abstract

Recently, research on techniques of processing of KNN (K Nearest Neighbor) queries in order to find the nearest neighbor in wireless sensor networks is actively in progress. The existing representative techniques of processing of KNN queries suggest the structure-based routing technique and the non-structure-based routing technique. However, the existing representative techniques of processing of KNN queries have problems of the consumption of high energy by sensor nodes or much time spending in query processing. This paper comes up with QKNN (Quad-tree based KNN) in order to solve such problems with existing KNN query processing techniques and more efficiently process KNN queries. QKNN searches for the sensor node nearest to the query by the means of GPSR, and then composes an R-tree in order to set up the KNN Boundary with the searched node as the reference. Then, based on the R-tree, it performs parallel queries by structuring the query area in a Quad-tree in accordance with the distribution of nodes, and in each cell within the Quad-tree it processes queries by means of the non-structure based itinerary routing technique. Lastly, via various experiments using sensor data, this paper proves the excellence of the proposed technique of processing of the nearest neighbor queries.

Keywords: *Sensor Networks, KNN, Quad-Tree, Itinerary Routing*

1. Introduction

In accordance with the rapid development of wireless communication technology and various data sensing sensor technology including temperature sensors, moisture sensors, and pressure sensors, recently research for utilizing wireless sensor network related technology is actively in progress in various fields including military affairs, medical service, weather, environment, traffic, home, and companies [3, 5, 6].

Especially, processing of KNN (K Nearest Neighbor) for bringing desired data from the sensor node nearest to a particular location is recognized as an important field of research.

The reduction of energy consumption by sensor nodes is regarded as a crucial issue as sensor nodes are operated by little batteries in wireless sensor networks [9, 10]. Since the greatest energy consumption by sensor nodes is due to communication costs and the cost of one time communication is ten to several hundred times as expensive as that of one time computation, it is very important to reduce the amount of sent data and the times of sending when processing KNN queries that are frequently used in sensor networks. However, the existing techniques of processing KNN queries are problematic in that they consume much energy or take long time.

Representative KNN query processing techniques include PT (Peer-Tree), KPT (KNN Perimeter-Tree), and DIKNN (Density-aware Itinerary KNN) [2, 4, 7]. PT is problematic in that particular sensor nodes (the root and internal nodes) can consume higher energy than other sensors and have shortened availability, or unnecessary consumption of energy by sensor nodes can occur in the course of message delivery among sensor nodes. KPT has the problem of large-scale energy consumption by sensor nodes due to the fact that KNN Boundary increases as the value of K does. On the other hand, DIKNN is inefficient in the sensor network environment with non-uniform distribution. That is to say, because the establishment of the optimal KNN Boundary is difficult, DIKNN has the problem of high energy consumption by sensor nodes and increase in query latency.

This paper proposes QKNN (Quad-tree based KNN) in order to solve such problems with the existing techniques of KNN query processing and carry out more efficient processing of KNN in wireless sensor networks. QKNN searches for the H-node nearest to QP and uses the GPSR technique to send queries from S-node to H-node. As for the establishment of the KNN Boundary, it begins constructing R-tree with H-node as the reference, and finishes the construction of R-tree in case that K pieces of sensor nodes are included in the course of R-tree construction. Then, as for the division of the KNN Boundary, it constructs KQ-tree from the KNN Boundary in accordance with the distribution of sensor nodes, and uses the information of the number of sensor nodes within the R-tree node and the locations of sensor nodes. Lastly, it carries out the sending of queries and the collection of their results by using the divided KNN Boundary as KQ-tree and uses the itinerary query processing technique for search within the KNN Boundary.

2. Related Works

2.1. PT (Peer-Tree)

PT, as a technique for KNN query processing in order to reduce communication costs by constructing a hierarchical routing, is a structure-based routing technique by which every sensor node in a sensor network is constructed as a hierarchical routing tree [6]. Figure 1 shows the routing structure of PT.

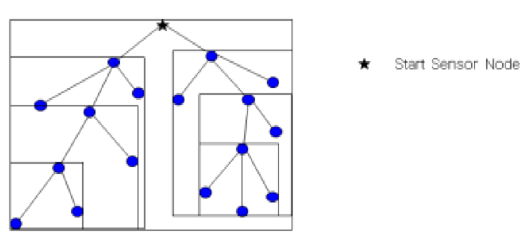

Figure 1. Routing Structure of PT

As shown from Figure 1, PT expresses node locations as MBR's (Minimum Bounding Rectangles) and selects the sensor node nearest to the query starting sensor node as the parent sensor node. In addition, the parent node knows the ID's and locations of all the child nodes included the MBR's, and vice versa. Thus, PT performs search for KNN by using a hierarchical routing (tree). However, PT is problematic in that particular sensor nodes (root and internal nodes) consume higher energy than others and the time of availability can be shortened or unnecessary energy consumption by sensor nodes occur in the course of message delivery among sensor nodes.

2.2. KPT(KNN Perimeter-Tree)

KPT is a technique of KNN query processing in order to reduce communication costs by constructing a hierarchical routing [5]. It is a structure-based routing technique by which sensor nodes around the query location rather than all sensor nodes are constructed as a hierarchical routing (minimum spanning tree). Figure 2 show the routing structure of KPT.

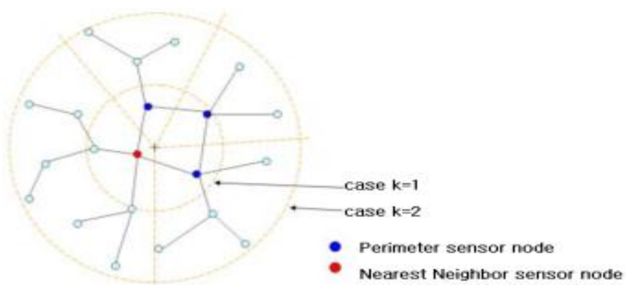

Figure 2. Routing Structure of KPT

As shown in Figure 2, KPT finds out the sensor node nearest to the query location, and sets up the maximum KNN Boundary. In addition, it uses the GPSR (greedy perimeter stateless routing) to find out the nearest sensor node, and searches for K pieces of sensor nodes by constructing a static routing with the nodes within KNN Boundary. It constructs a minimum spanning tree within the KNN Boundary, gives queries to K pieces of sensor nodes, and after receiving results, returns the final results to the querying sensor node. KPT uses the MHD-2 (Maximum Hop Distance-2) technique for setting up the KNN Boundary. By the MHD-2 technique, one can set up as the KNN Boundary the value resulting from the distance from the starting location of the query to the nearest node divided by the number of sending, that is to say the value of MHD-2 * k. However, KPT has the problem of the very high energy consumption by sensor nodes due to the increase of the KNN Boundary with the K value increased.

2.3. DIKNN(Density-aware Itinerary KNN)

DIKNN is a KNN query processing technique by which one can carry out search by dividing the KNN Boundary into multiple cone-shape areas in order to reduce communication costs [4]. Also, while PT and KPT are structure-based routing techniques, DIKNN is a non-structure-based routing technique. Figure 3 shows the routing structure of DIKNN

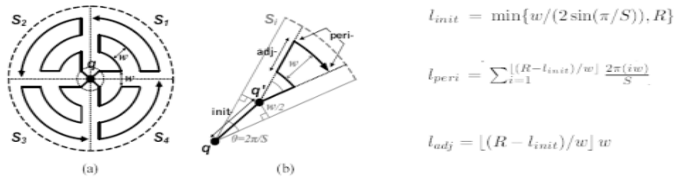

Figure 3. Routing Structure of DIKNN

As shown in Figure 3, DIKNN finds out the node (H-node) nearest to the query location Q from the query starting node (S-node) by means of the GPSR at the stage of search. Then, it sets up the KNN Boundary with H-node as the reference at the stage of KNN Boundary setup, and sets up routing structure on the basis of the density of sensor nodes calculated at the stage of search for the nearest node. Lastly, it delivers the query message to all the sensor nodes within the KNN Boundary at the stage of query sending, and then returns final results to S-node. On the other hand, Q-node sends the query to D-nodes within the range of communication and then D-node sends query results to Q-node. Q-node sends collected query results to next Q-node. As for next Q-node, DIKNN selects as next Q-node the sensor node among other nodes in the range of Q-node communication that can perform the farthest query sending in accordance with the ideal itinerary. However, DIKNN is inefficient in the environment of the sensor network with non-uniform distribution. That is to say, DIKNN is problematic in that the energy consumption by sensor nodes is high and Query Latency increases because it is difficult to set up the optimal KNN Boundary setup.

3. QKNN (Quad-tree based KNN)

3.1. KNN Boundary Setup

QKNN is a Quad-tree based KNN processing technique for reducing communication costs in sensor networks. It is efficient in the sensor network with non-uniform distribution as it is a non-structure-based routing technique and a KNN search based on Quad-tree. Also, the optimal KNN Boundary setup is possible with it, and query latency is small due to parallel processing of the KNN Boundary. Table 1 shows the terms used in this paper.

Table 1. Terminology

S-node	Query Start Sensor Node
H-node	Nearest Sensor node in Query Point
Q-node	Query Send Sensor Node
D-node	Data Send Sensor Node
KNNB	KNN Boundary
KQ-Tree	Quad-Tree in KNN Boundary
QP	Query Point
KM	Maximum Sensor Node in KQ-Tree
SD	Send Data
R	Sensor Node Communication Range
W	$\frac{\sqrt{3}}{2}R$

Table 1 defines S-node as a query start sensor node, H-node as the nearest sensor node in query point, Q-node as a query sending sensor node, D-node as a data sending sensor node, KNNB as the KNN Boundary, KQ-tree as Quad-tree in the KNN Boundary, QP as the query point, KM as the maximum sensor node in KQ-tree, SD as sent data, R as the sensor node communication range, and was the routing interval.

QKNN uses the GPSR technique for searching for the nearest sensor node [1]. Figure 4 shows the way to search for H-node, the sensor node nearest to the query node QP.

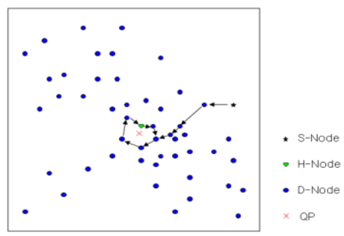

Figure 4. H-node search

As shown in Figure 4, the GPSR technique is used to send queries from S-node to H-node. The GPSR technique uses greedy forwarding, perimeter forwarding, and RN-perimeter. In greedy forwarding, the basic method of the GPSR, S-node delivers a message to the node nearest to the destination among other neighbors in order to reach the destination in the minimal number of deliveries. Perimeter forwarding is the way to find a next delivery node in case that the message cannot reach the destination in the way of greedy forwarding (in case there is an area without any sensor). Lastly, by means of RN-perimeter, one can find out the nearest of the sensors located around the query point.

QKNN constructs a hierarchical tree routing with H-node as the reference in order to set up the KNN Boundary. Figure 5 shows the structure of hierarchical tree routing.

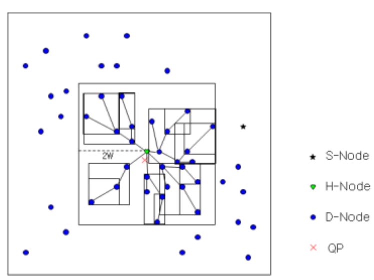

Figure 5. Structure of Hierarchical Tree Routing

As shown in Figure 5, QKNN terminates the construction of routing in case that K pieces of sensor nodes are included in the course of the construction of routing. That is to say, it constructs R-tree extending KNNB by W multiples until the number of sensor nodes within the KNN Boundary is equal to or bigger than K. Figure 6 shows the node structure of R-tree.

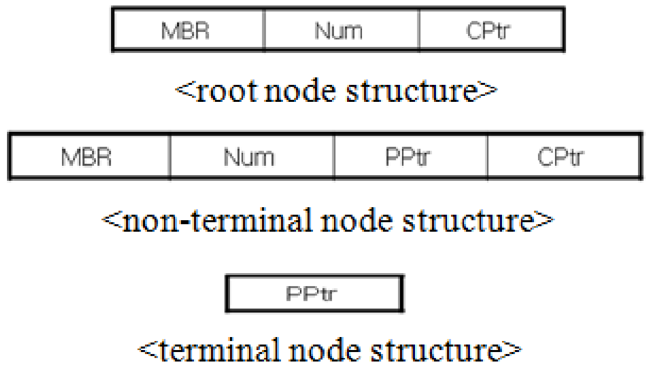

Figure 6. Node Structure of R-tree

As shown from Figure 6, the root node structure has MBR (areal coordinates of the whole sensor network), Num (number of the sensor nodes within the MBR area), and CPtr (pointer to child sensor nodes), while the non-terminal node structure has MBR (areal coordinates of the whole sensor network), Num (number of the sensor nodes within the MBR area), and PPtr (pointer to parent sensor nodes). Lastly, the terminal node structure has PPtr (pointer to parent sensor nodes).

3.2. Query sending and query result collection

QKNN generates KQ-tress by dividing the KNN Boundary in accordance with the distribution of sensor nodes. Figure 7 shows the structure of KQ-tree.

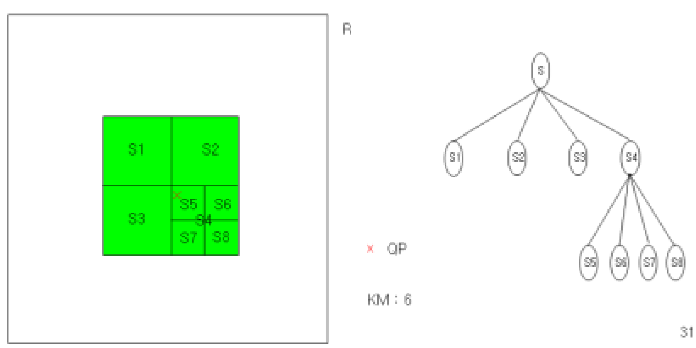

Figure 7. Structure of KQ-tree

As seen from Figure 7, by means of R-tree, KQ-tree divides ???until the number of the sensor nodes within the tree node is smaller than KM. Figure 8 shows the node structure of KQ-tree.

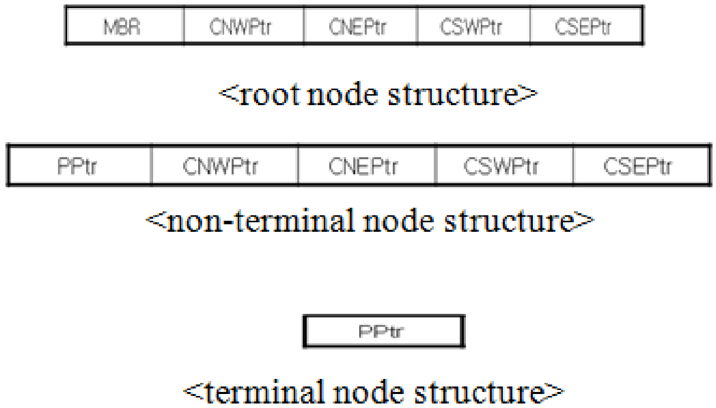

Figure 8. Node Structure of KQ-tree

As shown from Figure 8, the root node structure has MBR (areal coordinates of the whole sensor network), CNWPtr, CNEPtr, CSWPtr, and CSEPtr (pointer to child sensor nodes), while the non-terminal node structure has PPtr (pointer to parent sensor nodes), CNWPtr, CNEPtr, CSWPtr, and CSEPtr (pointer to child sensor nodes). Lastly, the terminal node structure has PPtr (pointer to parent sensor nodes).

QKNN uses the Itinerary query processing technique to send queries and collect query results [8]. Figure 9 shows the Itinerary structure.

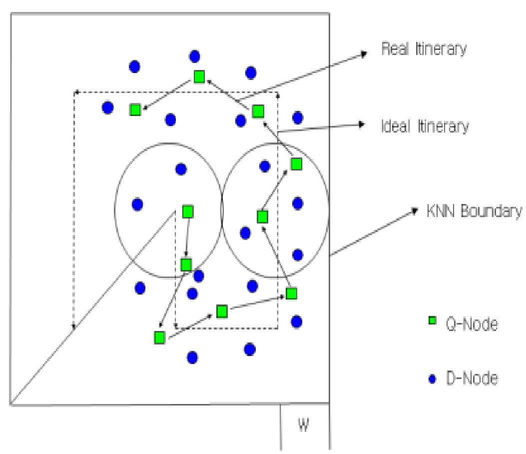

Figure 9. Itinerary Structure

As seen from Figure 9, Q-node sends queries to D-node, which in turn query results to the former. Then, Q-node sends the collected query results for next Q-node. As for next Q-node, QKNN selects as next Q-node the nearest sensor node among other nodes to an arbitrary point in the range of Q-node communication.

Moreover, QKNN processes the divided KNN Boundary in parallel. Figure 10 shows the parallel processing structure.

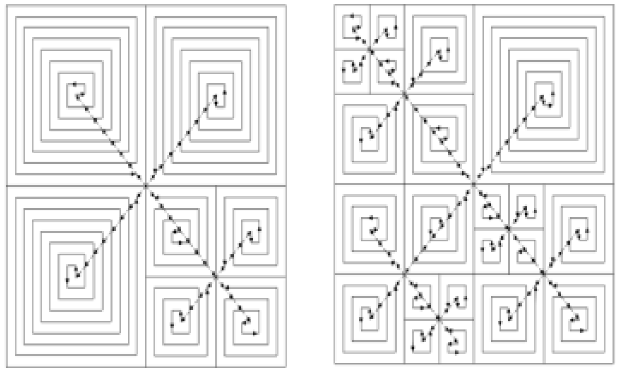

Figure 10. Parallel Processing Structure

As shown in Figure 10, after construct the divided KNN Boundary into Quad-tree, QKNN processes it in parallel by performing itinerary setup in terms of the coordinate of each node in KQ-tree.

4. Performance Evaluation

The hardware specification of the system used for performance evaluation includes Intel Core 3.2GHz CPU, 4GB RAM, 500GB HDD, and Windows XP as the operating system. Moreover, MFC (Microsoft Foundation Class Library) has been used for simulation.

As for performance evaluation, in terms of the number of sensor nodes and the K pieces of them, we have evaluated the energy consumption and the query processing time of PT, KPT, DIKNN, and QKNN, which is proposed in this paper.

Figure 11 shows the results of the performance evaluation of the energy consumption and the query processing time based on the number of sensor nodes.

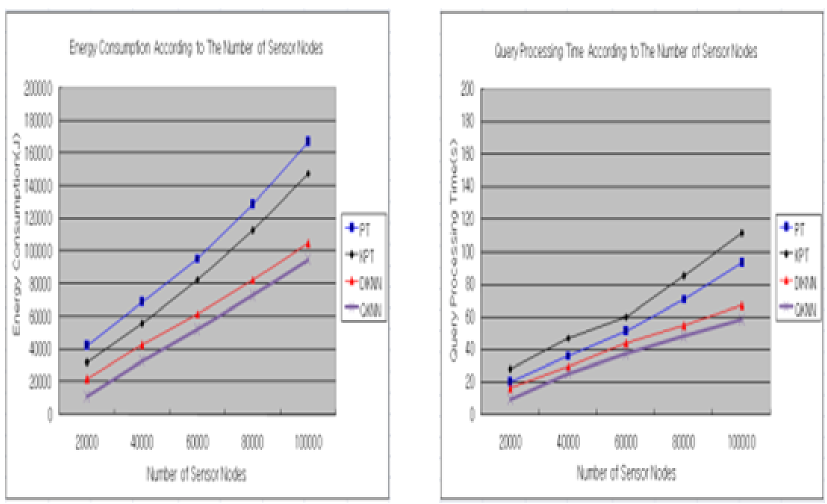

Figure 11. Performance Evaluation based on the Number of Sensor Nodes

As can be seen from Figure 11, for the energy consumption, QKNN is more enhanced than PT by an average of 59 %, than KPT by an average of 37 %, than DIKNN by an average of 15 %. As for the query processing time, it is more enhanced than PT by an average of 49 %, than KPT by an average of 25 %, and than DIKNN by an average of 13 %.

Figure 12 shows the performance evaluation of energy consumption and query processing time in terms of K pieces.

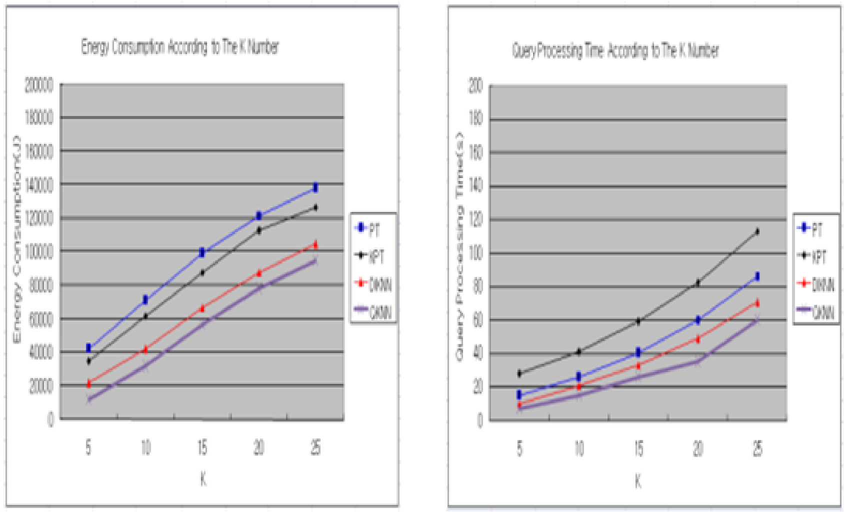

Figure 12. Performance Evaluation in Terms of K Pieces

As can be seen from Figure 12, or the energy consumption, QKNN is more enhanced than PT by an average of 66 %, than KPT by an average of 42 %, than DIKNN by an average of 35 %. As for the query processing time, it is more enhanced than PT by an average of 56 %, than KPT by an average of 26 %, and than DIKNN by an average of 17 %.

5. Conclusions

Recently, research on techniques of processing of KNN (K Nearest Neighbor) queries in order to find the nearest neighbor in wireless sensor networks is actively in progress. However, the existing representative techniques of processing of KNN queries have problems of the consumption of high energy by sensor nodes or much time spending in query processing.

Thus, this paper comes up with QKNN (Quad-tree based KNN) in order to solve such problems with existing KNN query processing techniques and more efficiently process KNN queries. QKNN searches for the sensor node nearest to the query by the means of GPSR, and then composes an R-tree in order to set up the KNN Boundary with the searched node as the reference. Then, based on the R-tree, it performs parallel queries by structuring the query area in a Quad-tree in accordance with the distribution of nodes, and in each cell within the Quad-tree it processes queries by means of the non-structure based itinerary routing technique.

References

[1] B. Karp and H. T. Kung, Eds., "Greedy Perimeter Stateless Routing for Wireless Networks", Proceedings of the International Conference on Mobile Computing and Networking, **(2000)**, pp. 243-254.

[2] B. Wu, K. T. Chuang, C. M. Chen and M. S. Chen, Editors, "An Itinerary-based KNN Query Processing Algorithm for Mobile Sensor Networks", Proceedings of the IEEE International Conference on Data Engineering, **(2007)**, pp. 456-465.

[3] D. Culler and D. Estrin, "Overview of Sensor Networks", Journal of the IEEE Computer Magazine, vol. 37, no. 8, **(2004)**, pp. 41-49.
[4] J. Winter and W. Lee, Editors, "A Dynamic KNN Query Processing Algorithm for Location-aware Sensor Networks", Proceedings of the ACM International Workshop on Data Management for Sensor Network, **(2004)**, pp. 119-124.
[5] K. Y. Lee, M. J. Lim, J. J. Kim, K. H. Kim and J. L. Kim, "Design and Implementation of a Data Management System for Mobile Spatio-Temporal Query", Journal of the IWIT, vol. 11, no. 1, **(2011)**, pp. 109-113.
[6] K. Y. Lee, M. J. Lim, K. H. Kim and J. J. Kim, "Spatio-temporal Query Processing Systems for Ubiquitous Environments", Journal of the IWIT, vol. 10, no. 3, **(2010)**, pp. 145-152.
[7] M. Demirbas and H. Ferhatosmanoglu, Eds., "Peer-to-peer Spatial Queries in Sensor Networks", Proceedings of the IEEE International Conference on Peer-to-Peer Computing, **(2003)**, pp. 32-39.
[8] Y. Xu, W. Lee, J. Xu and G. Mitchell, Eds., "Processing Window Queries in Wireless Sensor Networks", Proceedings of the ICDE International Conference, **(2006)**, pp.70-70.
[9] A. A. Aaqif, and K. Asif, "An Intelligent Neural-Wireless Sensor Network Based Schema for Energy Resources Forecast", Journal of the IJAST, vol. 33, **(2011)**, pp. 121-130.
[10] S. Razieh, J. Sam and K. Z. Ahmad, "Comparison of Energy Efficient Clustering Protocols in Heterogeneous Wireless Sensor Networks", Journal of the IJAST, vol. 36, **(2011)**, pp. 27-40.

Authors

Jeong-Joon Kim received his B.S. and M.S. degrees in Computer Science at Konkuk University in 2003 and 2005, respectively. In 2010, he received his Ph.D. degree in Databases at Konkuk University. He is currently a professor at the department of Computer Science at Konkuk University. His research interests include database systems, geographic information systems(GIS), location-based systems(LBS), main memory databases, embedded databases, and wireless sensor networks (WSN) *etc*.

Jeong-Jin Kang is currently the faculty of the Department of Information and Communication at Dong Seoul University in SeongNam, Korea since 1991, and currently the President of the Institute of Webcasting, Internet and Telecommunication(IWIT). During 3 years from Feb. 2007 to Feb. 2010, he worked as a Visiting Professor at the Department of Electrical and Computer Engineering, The Michigan State University. He was a lecturer of the Department of Electronic Engineering at (Under) Graduate School(1991-2005), The Konkuk University. Dr. Kang is a member of the IEEE Antennas and Propagation Society (IEEE AP-S), the IEEE Microwave Theory and Techniques Society(IEEE MTT-S), and a member of the Institute of Webcasting, Internet and Telecommunication(IWIT), Korea. His research interests involve Smart Mobile Electronics, RF Mobile Communication, Smart Convergence of Science and Technology, RFID/USN, u-Healthcare and ultrafast microwave photonics, as well as GIS, LBS, moving objects databases, and telematics etc.

Ki-Young Lee received his B.S. degree in Computer Science at Soongsil University in 1984. In 1988 and 2005, he received M.S. and Ph.D. degrees in Databases at Konkuk University, respectively. From 1984 until 1991, he worked for Korea Institute of Ocean Science & Technology(KIOST) as a researcher in Data Information & Processing department. He is currently a professor at the department of Medical IT and Marketing at Eulji University. He was also the head of department of S/W development in Bio-Meditech Regional Innovation Center at Eulji University. He is the director of the Korea Webcasting, Internet & Telecommunication(IWIT). His research interests include spatial databases, geographic information systems(GIS), location-based services(LBS), u-Healthcare, ubiquitous sensor network(USN), moving objects databases, and telematics *etc*.

Gyoo-Seok Choi received the B.S., M.S., and Ph.D. degrees in electrical engineering from the Yonsei University, Seoul Korea, in 1982, 1987, and 1997, respectively. He worked at the laboratory of DACOM Company as a researcher from 1987 to 1990. He also worked at the laboratory of SK Telecom Company as a senior researcher from 1991 to 1996. He is currently a professor at the Dept. of Computer Science in Chungwoon University. He is a vice-president of the Korea Webcasting, Internet & Telecommunication (IWIT). His current research interests include Artificial Intelligence, Telematics, Mobile Computing, etc.

Yong-Soon Im received his B.S. degree in Electronic Engineering at Sungkyunkwan University in 1988. In 1993 and 1999, he received M.S. and Ph.D. degrees at Sungkyunkwan University, respectively. From 1988 until 1990, he worked for LG Electronics as a researcher in Video Camcorder. He is currently a professor at the department of Broadcasting Production at Kookje University. He is the Vice President of the Korea Webcasting, Internet & Telecommunication(IWIT). His research interests include Image processing, Image communication and mobile contents *etc*.

Eun-Young Kang received her B.S. degree in Computer Science at Sookmyeong Women's University in 1987. In 1999 and 2009, she received M.S. and Ph.D. degrees at Sungkyunkwan University, respectively. From 1987 until 2002, she worked for Kyobo as a researcher in Computer Information department. She is currently a professor at the department of Information & Communication at Dongyang Mirae University. She is the director of the Korea Webcasting, Internet & Telecommunication(IWIT). Her research interests include u-Healthcare , mobile computing etc.