International Journal of Advanced Smart Convergence Vol.12 No.1 1-8 (2023)
http://dx.doi.org/10.7236/IJASC.2023.12.1.1

IJASC 23-1-1

A Hybrid Index of Voronoi and Grid Partition for NN Search

Seokjin Im†

Professor, Department of Computer Engineering, Sungkyul University, Korea
imseokjin@gmail.com

Abstract

Smart IoT over high speed network and high performance smart devices explodes the ubiquitous services and applications. Nearest Neighbor(NN) query is one of the important type of queries that have to be supported for ubiquitous information services. In order to process efficiently NN queries in the wireless broadcast environment, it is important that the clients determine quickly the search space and filter out NN from the candidates containing the search space. In this paper, we propose a hybrid index of Voronoi and grid partition to provide quick search space decision and rapid filtering out NN from the candidates. Grid partition plays the role of helping quick search space decision and Voronoi partition providing the rapid filtering. We show the effectiveness of the proposed index by comparing the existing indexing schemes in the access time and tuning time. The evaluation shows the proposed index scheme makes the two performance parameters improved than the existing schemes.

Key words: *Voronoi Diagram, Grid Partition, Nearest Neighbor, Wireless Data Broadcasting.*

1. INTRODUCTION

Mobile devices with performances comparable to that of PCs and high speed networks trigger the needs for information services based on the location of the clients [1]. Minimized devices like smart watches connected to the internet activate the needs for the ubiquitous services without the limitation of time and place. Also, smart IoT(Internet of Things) explodes the ubiquitous services and applications as well as the number of clients. These features cause the importance of maintaining the qualities of the services regardless of increasing rapidly the number of the clients [2, 3].

In order to support the ubiquitous information services based on the locations, several types of queries play an important role for enhancing the efficiency of the services. Nearest Neighbor(NN) query is one of the important type of queries that have to be processed. NN query is the query that finds the nearest data item to a query point.

The wireless data broadcasting system is an efficient method for the ubiquitous information services, not affected the number of clients [4]. The system enables seamless information services for any number of clients at any time and any place. In the system, the server broadcasts data items on the wireless channel and the clients obtain the data items from the channel by downloading them. To elevate the efficiency for the clients to access data items on the wireless channel, the server provides an additional information on the channel, i.e., the

Manuscript Received: December. 4, 2022 / Revised: December. 7, 2022 / Accepted: December. 8, 2022
Corresponding Author: imseokjin@gmail.com
Tel: +82-31-467-8168, Fax: +82-31-467-8053
Professor, Department of Computer Engineering, Sungkyul University, Korea

index that holds the time information when the data items are appeared on the channel. With the index broadcast, the clients access the wireless channel and check out the broadcasting time for their queried data items. Then the clients download the data items at the broadcasting time.

The index for the broadcast system can be provided in two manners, a centralized scheme and a distributed scheme. The index in the centralized scheme holds the time information on all data items to be broadcast. To reduce the waiting time for the index, the index are disseminated several times in a broadcast cycle [5]. This scheme has advantage that the clients figure out the distribution of all data items and lets the clients determines easily the search space for query processing. However, the scheme makes the length of a broadcast cycle long so that the access time waiting for the data items is lengthened. On the other side, the distributed scheme disperses indexes for partial data items [6]. The scheme has advantage to reduce the length of the broadcast cycle because it repeats the information of the whole data items differently from the centralized scheme. However, the scheme causes the clients to spend long time for deciding the search space for the given queries. The index scheme can affect the performances of query processing in the wireless broadcast system . Especially NN query processing is strongly affected by the index scheme, because the query requires the data distribution information of all data items to determine the search space for NN. The centralized indexing scheme can bestow the entire distribution information but can degrade the data waiting time. On the other hand, the distributed indexing scheme can reduce the data waiting time for given queries but can increase the time for determining the search space because the scheme provides only partial information about data distribution.

In order to process efficiently NN queries in the wireless broadcast environment, it is important that the clients determine quickly the search space and filter out NN from the candidates containing the search space. In this paper, we propose a hybrid index of Voronoi and grid partition to provide quick search space decision and rapid filtering out NN from the candidates. Grid partition plays the role of helping quick search space decision and Voronoi partition providing the rapid filtering out.

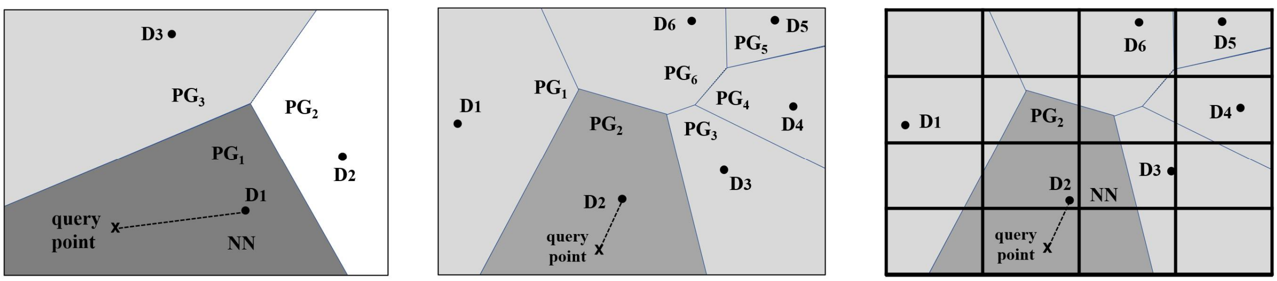
The rest of this paper is organized as follows. Section 2 offers related works about Voronoi partition and existing indexing schemes for supporting NN query processing. Section 3 provides the proposed hybrid index of Voronoi and grid partition and the procedure processing a NN query. In Section 4, we compare the proposed hybrid with the existing indexes to evaluate the performance and show how efficient the hybrid index is than the existing. Then, we make a conclusion of this paper.

2. Related Works

2.1 Voronoi Partition

Voronoi partition is a method of dividing a plane with N data items, $\{D_1, D_2, \dots D_N\}$ into N polygons $\{PG_1, PG_2, \dots PG_N\}$, while satisfying that Polygon PG_k is a set of points with the closest distance to data item D_k [9]. Due to the characteristic of Voronoi partition, the NN to a given query point contained in PG_k is D_k. For example, Figure 1(a) shows Voronoi partition for 3 data items. The NN to the given query point is D_1 because the query point is contained in PG_1 and the closest data item for any point in PG_1 is D_1.

With Voronoi partition, the NN to a query point is determined by examining the containment of the query point to a pologon, instead of calculating the distance from the query point to each data item. The complexity of NN query processing with Voronoi partition for N data items is O(N) because the clients have to examine N polygons for the containment. NN search using the contaiment test with Voronoi partition provides a clue for making NN query processing more quickly. The clue is reducing the number of polygons to be examined for determining NN. We can associate Voronoi partition with regular grid partition in order to reduce the number of plolygons to be examined for the containment test. For example, Figure1(b) shows Voronoi partition for 6 data item and a query point for NN search. With only Voronoi partition, the clients have to examime 6 polygons for NN. However, in case of using the association Voronoi partition with a regular grid partition, the polygons the clients have to examine are those overlapped with the grid cell containing the query point. In Figure 1(c), the clients only examine PG_2 overlapped with the grid cell containing the query point. Thus, the association of Voronoi partition with the grid partition makes the number of polygons to be examined for NN. The association of Voronoi partition for N data items with n*n grid partition reduces the complexity of NN query processing $O(N/n^2)$ when data items are distributed uniformly over a plane.



(a) For 3 items (b) For 6 items (c) Association of Voronoi Partition with Grid Partition

Figure 1. Voronoi Partition

2.2 Indexes for the NN search

Various indexes have been proposed to let the clients efficiently search for NN in wireless broadcasting environments. They can be categorized in two groups, the one using the Hilbert curve like Hilbert Curve Index (HCI) and Distributed Spatial Index (DSI) and the other using grid space partition like Non-uniform Space Partition (NSPI).

HCI adopts the Hilbert curve for representing the positions of the spatial data in integer [6]. In HCI each data item is assigned to its own integer that is related to its location. HCI is organized in B-tree with the integers of data items and is broadcast m times on the wireless channel interleaving with data. Using HCI, the client can decide quickly search space for NN for a given query point because HCI holds the distribution information on all data items. Then, the clients decide NN for the given query point, downloading the candidates, i.e., all the data containing within the search space. HCI, however, makes the access time of the clients long because the search space by HCI is large.

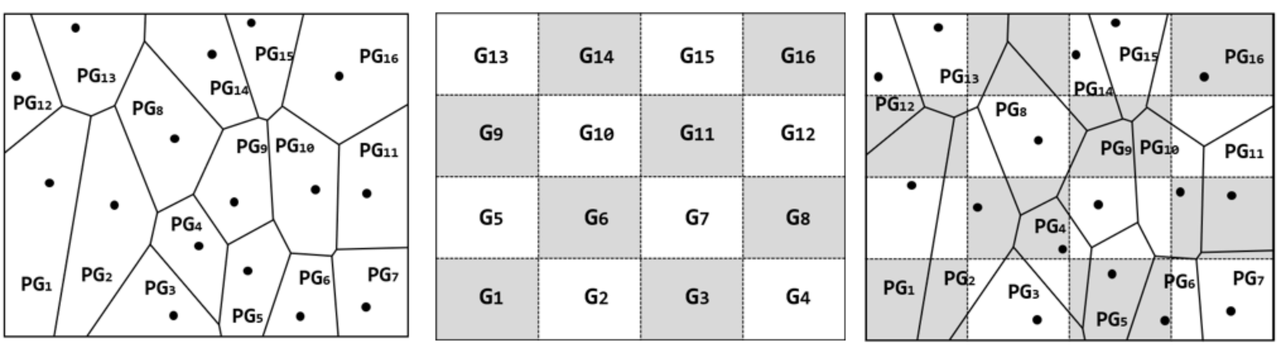
DSI also adopts the Hilbert curve for the same reason with HCI. DSI is organized in the distributed scheme, differing from HCI using the centralized scheme that keeps the information on all data items [7]. DSI uses a chunk that has one data item and a timetable that keeping the broadcast times of items with Hilbert curve numbers separate exponentially in integers. DSI makes long the length of the broadcast cycle because the broadcasting times of data items are repeated frequently. The clients must determine the search space for NN with the distributed index on the channel. That makes the access time of the clients very long. Also, pruning the candidates within the search space takes long time and consumes a lot of energy. HCI and DSI using the Hilbert curve for representing the locations of the data items make the clients spend more time to obtain NN for the given query points because they must download more data items from the channel, containing within the large search space determined by Hilbert curve not the real locations.

NSPI is proposed to avoid the inefficiency using the Hilbert curve and to enhance the efficiency of NN query processing by adopting the real locations of the data items [8]. NSPI partitions the data space in grid cells and organizes the index for each cell in a table using a distributed scheme. NSPI provides two kinds of index information: the one for the distribution of whole data items and the other for data broadcasting times of data items in a grid cell. Using NSPI the clients determine the search space for NN to a given query point and then they prune the data items within the space for deciding the NN. NSPI enables the clients to process efficiently NN queries more than HCI and DSI because NSPI uses the real coordinates of data items instead of the Hilbert Curve as HCI and DSI.

Although NSPI shows the improved performances to HCI and DSI, NSPI still remain restricted in the way to search for NN, i.e., determining the search space then deciding NN to a given query point by pruning the candidates withing the search space. In this paper, we propose the indexing scheme determining the data space and pruning the candidates in more simple and easy way. For the aim, the scheme adopts the Voronoi and grid partition of the data space.

3. HYBRID INDEX OF VORONOI AND GRID PARTITION

For the proposed HIVG (Hybrid Index of Voronoi and Grid Partition), we partition the data space D containing N data items in two levels, the one for NN search and the other for organizing the index. First, we partition the space into Voronoi partition that has N polygons with only one in itself. For example, Figure 2(a) shows the Voronoi diagram for data items at N=16 depicted as dot. The Voronoi diagram makes the clients determine NN to a query point by finding the subregion containing the query point. Next, we partition the data space into n*n grid that is for organizing the index. Figure 2(b) shows a n*n grid partition of the data space at n=4. Figure 2(c) shows the Voronoi polygons overlapped with the grid partitions.



(a) (b) (c)

Figure 2. Data Space Partition with Voronoi and Grid

3.1 Hybrid Index Organization and Data Scheduling

The proposed HIVG is organized with two indexes, GI(Grid Index) and VI(Voronoi Index). GI is organized in a table holding the information on grid cells and their broadcasting time as follows:

$$GI = \{ <G_k, t_k> \mid 0<k\leq n^2\} \qquad (1)$$

Here, G_k means the grid cell number and t_k means the broadcasting time of G_k on the channel.

VI_k is organized in a table holding the information on the Voronoi partition overlapped with grid cell G_k and data items in the grid cell G_k as follows:

$$VI_k = \{< PG_i, t_i> \mid 0<i\leq N \text{ and } (PG_i \cap G_k \neq \Phi) \}, \text{ for } G_k \qquad (2)$$

Here, PG_i is the polygon of a subregioin of Voronoi partition overlapped with grid cell G_k and t_i is the broadcasting times of the data items contained in polygon PG_i. In VI_k, N means the number of polygons of the Voronoi partition overlapped with grid cell G_k.

GI helps the clients to decide a subset of the Voronoi partition for NN search to the given query point.

Also, GI supports the clients to access the subset for the query point. VI_k lets the client decide NN by searching the subregion of Voronoi containing the query point. Also, the clients access efficiently the NN data item with the broadcasting time in DT_i of VI_k. That is because HIVG lets the clients avoid pruning the candidates for NN after deciding the data search space for NN.

3.2 Index Generation and Data Scheduling

The broadcast server organizes GI and VI after partitioning the data space D and composes Q_{BC} that is the scheduling queue storing GI, VI and data items in D in the order of broadcasting. Algorithm 1 shows the procedure of organizing the indexes and scheduling the data items in D. The line 1 to 3 shows that the server organizes GI after the n*n grid partitioning and generates Voronoi polygons. The for-block in line 4 depicts organizing VI by assigning Voronoi polygons overlapped with each grid cell to it. The server schedules GI, VI and data items in the broadcasting order by queuing them into Q_{BC} in line 7 to 12. By line 8 and 9, GI is repeated n times every n grid cells, i.e., every first grid cell of each row in the grid partition. The condition (k%n ==1) means the first grid cell number of each row in the n*n grid partition. Then, VI_k is enqueued to Q_{BC} (line 10) and data items contained within PG_i in VI_k are enqueued to Q_{BC} (Line 11 and 12).

```
Algorithm 1: Space Partition and Scheduling
Input: D, a data set of N spatial data items
Output: GI ,VI and Q_BC (a queue for scheduling indexes and data items)

1: Split D to Grid SG with n^2 grid cells
2: Organize GI with the partition D.
3: Generate N Voronoi polygons for D

4: for k from 1 to n^2
5:      Assign Voronoi polygons overlapped with G_k
6:      Organize VI_k with the assigned polygons

7: for k from 1 to n^2
8:      if(k % n ==1)
9:           Enqueue GI to Q_BC
10:     Enqueue VI_k to Q_BC
11:     for PG_i in VI_k
12:          Enqueue the data item d_i in PG_i
```

3.3 Nearest Neighbor Search

Using HIVG, the client searches and downloads NN to the query point from wireless broadcast channel in three steps.

- **First Step**: the client tunes in to the wireless channel and accesses GI that is encountered firstly on the channel. Then, the client determines the grid cell G_k containing the given query point P_q using GI and the broadcasting time t_k of VI_k for the determined grid cell.
- **Second Step**: the client accesses VI_k at t_k. Then the client finds the polygon containing the query point Pq from polygons in VI_k and determines the broadcasting time t_{data} for NN to P_q as follows:

$$\left\{ \begin{array}{l} \text{for } \langle PG_i, t_i \rangle \text{ in } G_k \\ \quad \text{if } (P_q \text{ in } PG_i) \text{ then } t_{data} \leftarrow t_i \end{array} \right. \quad (3)$$

Third Step: the client waits for the NN in energy-saving mode until t_{data}. Then the client tunes into the wireless channel and downloads the NN from the channel.

Algorithm 2 shows the procedure for searching NN with the proposed HIVG. The first step corresponds to line 1 and 2. The second step corresponds to line 3 to 5 and the third step to line 6 and 7.

Algorithm 2: NN Search
Input: P_q, a query point Output: D_{NN}, the Nearest Neighbor Item to P_q 1: Access the first GI after tuning in to the broadcast channel 2: Determine G_k containing P_q and the broadcasting time t_k for G_k 3: Access VI_k at t_k 4: Search the polygon PG_q containing P_q with $<PG_i, t_i>$ in VI_k 5: Decide the broadcasting time t_{data} for the data item in the polygon PG_q with $<PG_i, t_i>$ 6: Switch into the energy-saving mode and tune in the broadcasting channel at t_{data} 7: Download D_{NN} to P_q , the data item contained within PG_q

When the query point P_q for NN search, for example, is given as in Figure 3, the client accesses index VI_1 for grid cell G_1 and finds PG_2 as the Voronoi polygon containing P_q among 3 Voronoi polygons overlapped with G1. Then, the client downloads the data item d_2 contained within PG_2 as the NN to P_q.

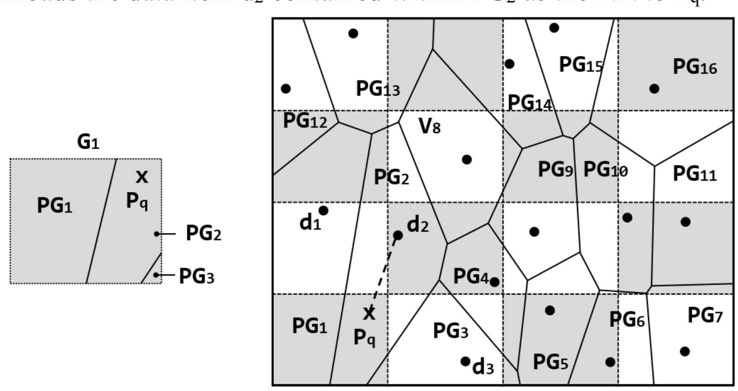


Figure 3. An example of searching for NN to the given P_q

4. Performance Evaluation

For evaluating how effective the proposed HIVG is, we compare it with other existing indexes, NSPI, HCI and DSI, for NN search. We implemented a testbed for simulations with SimJava, a discrete time simulation package in JAVA. The implemented testbed is organized with a broadcasting server, a wireless broadcasting channel and a client. In broadcasting environments, each client accesses to the channel and downloads data items independently so we implemented one client.

For the simulation we use 7023 American shopping sites as a real distribution of spatial data items. As the parameters for the simulations, we set grid cell id number to 8 bytes, each point for Voronoi polygons to 8 bytes, the size of a data item 2048 bytes, and bucket size to 128 bytes. The bucket is the logical unit for delivering data as packets in real networks. The client runs 100,000 NN queries in a simulation condition. In each NN query processing we measured the access time and tuning time as performance parameters. The access time depicts that how long time the client takes for the NN to the given query. The tuning time shows that the amount of energy for the client to use in the processing because the time means the sum of times staying in active mode, i.e., energy consuming mode. We compare the access time and tuning time of proposed HIGV with those of the existing indexes HCI, DSI and NSPI that support the NN search.

4.1 The access time comparison

The access time for NN search in wireless data broadcast system is affected by the two main factors. The first is the how quickly the clients determine the search space to the given query point and the second is how short the length of the broadcast cycle is. The proposed HIVG can reduces the time for determining the search space to the given query point because the grid cell containing the query point is the search space for NN. This is very simple and effective to reduce the access time. Also, HIVG reduces the length of a broadcast cycle by adopting the distributed indexing scheme. The two factors of HIVG let HIVG outperform the existing indexing schemes, HCI, DSI and NSPI. Figure 4 shows how effective HIVG is in the aspect of the access time. HIVG enables the clients to decide the answer NN to the given query points and access it rapidly. HCI degrades the access time because of the lengthened broadcast cycle by m time replication of the index in a cycle in spite of providing the data distribution information of all items. DSI and NSPI lengthen the access time because they make the clients take long time to decide the search space by partial distribution information of data items in spite of the short broadcasting cycle.

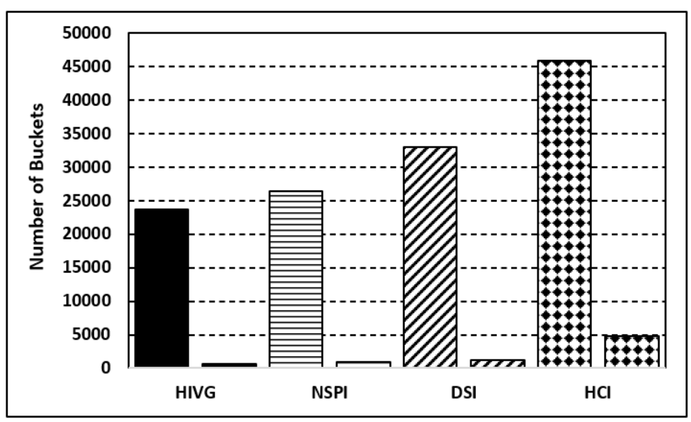


Figure 4. Access Time in the number of Buckets

4.2 The tuning time comparison

The tuning time depicts how efficiently the clients obtain the NN from the channel in the aspect of energy consumption. Figure 5 depicts the tuning time of HIVG as well as HCI, DSI, and NSPI. The figure shows that HIVG outperforms the three existing index schemes. This means that HIVG makes it enable for the clients to process more queries than the other indexes with the same battery capacity.

HCI with the longest tuning time makes the clients stay long in energy-saving mode. In spite that the energy consumption in the energy saving mode is low, staying long in the mode is not negligible in the aspect of entire procedure of query processing. DSI and NSPI in the distributed scheme make the clients download candidates in the active mode for filtering out NN from them.

HIVG shows the shortest tuning time because HIVG does not make the clients download many candidates to NN. HIVG lets the client enable to determine the NN with the Voronoi partition after accessing the grid cell on the wireless broadcast channel.

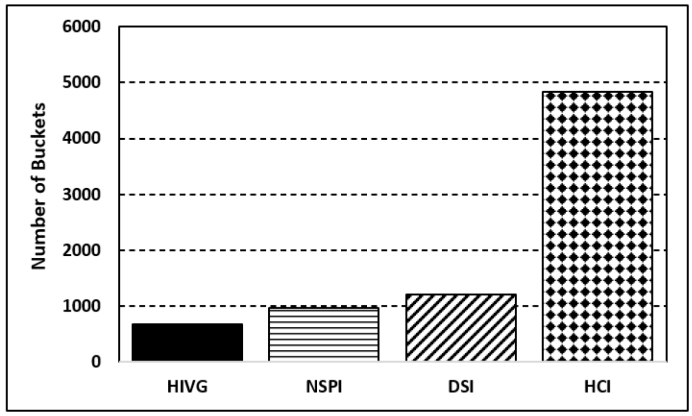


Figure 5. Tuning Time in the number of Buckets

5. CONCLUSION

This paper has proposed a hybrid index using Voronoi and grid partition to provide quick search space decision and rapid filtering out NN from the candidates. The index enables the clients to decide the search space simple and easily with grid partition and to filter out NN with Voronoi partition containing the query point. Through the performance evaluation, we showed that the proposed HIVG outperformed the existing index schemes, HCI, DSI, and NSP. Two factors of HIVG, quick search space decision and rapid filtering out NN, let HIVG outperform the existing indexing schemes, HCI, DSI and NSPI in the access time. Also, in the tuning time HIVG shows the shortest tuning time. That results from that HIVG does not make the clients download many candidates for NN by letting the client determine NN with the Voronoi partition after accessing the grid cell on the wireless broadcast channel.

REFERENCES

[1] S.M. Jung, "Data Hiding Technique using the Characteristics of Neighbor Pixels and Encryption Techniques," International Journal of Advanced Smart Convergence(IJASC), Vol. 11, No. 4, pp. 163-169, December 2022.

[2] S.J. Jang, "Design of a Heart Rate Measurement System using a Web Camera," International Journal of Advanced Smart Convergence (IJASC), Vol. 11, No. 4, pp. 179-186, September 2022.

[3] D.J. Shin, S.Y. Hwang, and J.J. Kim, "Development of System for Drunk Driving Prevention using Dig Data in IoT Environment," The Journal of The Institute of Internet, Broadcasting and Communication (IIBC), Vol. 22, No. 6, pp. 69-74, December 2022.

[4] A.B. Waluyo, F. Zhu, D. Taniar, and B. Srinivasan, "Design and Implementation of a Mobile Broadcast System," IEEE International Conference on Advanced Information Networking and Applications, Victoria, Canada, 2014.

[5] S. Im, H. Youn, J. Choi, and J. Ouyang, "A Novel Air Indexing Scheme for Window Query in Non-flat Wireless Spatial Data Broadcast,", Journal of Communication and Networks, Vol. 13, No. 4, pp. 400-407

[6] B. Zheng, W. C. Lee, and D. L. Lee, "Spatial Queries in Wireless Broadcast Systems," Wireless Network, Vol. 10, No. 6, 723-736, pp. 723-236, 2004.

[7] B. Zheng, W.C. Lee, K.C.K, D.L. Lee, and M. Shao, "A Distributed Spatial Index for Error-Prone Wireless Data Broadcast," VLDB Journal, Vol. 18, No. 4, pp. 959-986, 2009.

[8] Seokjin Im, "NN Search based on Non-uniform Space Partition on Air," Turkish Journal of Computer and Mathematics Education, Vol. 12, No. 5, pp. 317-326, 2021.

[9] P. Bhattacharya, "Voronoi Diagram in Optimal Path Planning," International Symposium on Voronoi Diagrams in Science and Engineering, Wales, UK, 2007.