# NN Search based on Non-uniform Space Partition on Air

**SeokjinIm*[1]**

[*1] Computer Engineering, Sungkyul University, Anyang, Gyeonggi-Do, Korea imseokjin@sungkyul.ac.kr[1]

**Abstract:** It is one of most important challenges in information services based on location to support a huge number of clients to process the NN query for a given query point. A scheme of NN query processing based on wireless data broadcasting is an effective way to overcome the challenge. In this paper, we propose an indexing scheme NSPI (Non-uniform Space Partition Index) for quick NN search over wireless data broadcasting. For implementing the aim, we use a non-uniform spatial partition and provide an index based on the partition for equidistance between indexes on the wireless channel. The index scheme enables the clients to process NN quickly by lessening the time for the clients to meet the index on the channel. To show the effectiveness of the proposed scheme, we compare the access time and tuning time of the clients with existing indexing schemes by simulation studies. The proposed NSPI shows shorter access time than the other existing scheme. Also, NSPI outperforms in the aspect of tuning time than the others.

**Keywords:** NN(Nearest Neighbor), Non-uniform Space Partition, Wireless Data Broadcast

## 1. Introduction

The advanced information technologies blast the interest and needs to various information services and explode the number of clients requesting information through high speed networks. The high computing-powered and micro-sized devices allows the clients to obtain their desired data items at any place and any time. Also, the devices connected to the networks make IoT (Internet of Things)into reality [1, 4, 5].

In the IoT environments, the scalability is critical challenge because of a great number of clients try to access the network and receive information services [2, 3]. Wireless data broadcasting system is the effective way to treat the challenge because it supports the clients regardless of the number of them [7, 11, 12].

The broadcast system is organized with a broadcast server, a wireless channel, and huge number of clients [8, 9, 10]. With the broadcast system, the information service is provides as follows: the broadcast server places the data items on the wireless channel periodically. Then, the clients tune into the channel, search their desired data items on the channel and download them from the channel. Thus, each client accesses information independently in the system.

The broadcast system can adopt spatial data items like hotels for location-based information services [12]. For example, with the system broadcasting data on hotels, the clients can process various types of queries, like window query and nearest neighbor (NN) query. A window query is the type of query searching data items contained within a given window, called query window. NN query is the type of query searching the nearest data item to a given location point, called query point. With the system broadcasting data on hotels in a city, a client can search NN hotel from its current location.

In the broadcast system, the clients have to listen to the channel in order to search its desired data items until they appear on the channel [14]. That way makes the clients equipped with batteries having restricted life time consume much energy while processing given queries. For efficient data search in the broadcast system, the server adopts an index carrying time information about when each data item appears on the channel. The server broadcasts data items with the index on the wireless channel. The client accesses the index on the channel firstly and obtains the broadcasting time of the desired item. Then, the client switches to doze mode (energy saving mode) till the broadcasting time. The client switches to active mode (energy consuming mode) at the time and listen to the channel and download the data item. Thus, an index on the wireless channel helps the clients process their own queries because they can know the time pointers to the target data items before downloading them [15, 16, 17].

We categorize the indexing scheme into $(1,m)$ index and distributed index [7]. With $(1, m)$ indexing scheme, the broadcast server places the index for whole data items to be broadcast m times on the wireless channel. This scheme provides the clients with time pointers to data items as well as the view of the distribution of the data items to be broadcast. The distribution view by the index is advantageous when the clients process queries. However, the index scheme makesthe access time of the clients lengthened because of the long broadcast cycle by the indexrepeated $m$ time.

In distributed indexing scheme, the broadcast server partitions the whole data items to be broadcast into $n$ parts [8, 11]. The serve generates partial index for each part the data items and interleaves each part of data items and partial index for it on the wireless channel. The partial indexes are connected with link information on the channel. As a result, the clients obtain the view of the distribution of the whole data after the access to all partial indexes on the channel. However, the distributed indexing scheme takes advantage of the access time of the clients because it does not repeat the index for whole data items like $(1, m)$ indexing scheme. In order to improve the access time of

the clients in the distributed indexing scheme, it is critical that the partial indexes keepsequi-distant on the channel. The equi-distant indexes reduce the average access time of the clients by shortening the average index waiting time.

In this paper, we propose Non-uniform Space Partition Index (NSPI) in the distributed scheme, targeting to support the clients to search NN to a given query point in the wireless spatial data broadcast environment. NSPI partitions the data space of the whole data items into non-uniform cells. All cells in the partition have the same number of data items. In the proposed scheme, the server generates the index for each cell and broadcast the index with the cell. With this scheme, the indexes for cells on the wireless channel can maintain equi-distant because all cells have the same number of data items. The proposed NSPI provide the clients with the compact view of the distribution of the whole data items to be broadcast in the level of cells. That view allows the client to process NN query efficiently.

We organize this paper as follows. Section 2 provides related works with this paper. Section 3 presents the proposed indexing scheme NSPI in detail. In Section 4, we evaluate the performances of NSPI by comparing NSPI with other existing indexing schemes. Then, we conclude this paper in Section 5.

## 2. Related Works

In order to support for the clients to process efficiently NN queries, a variety of indexing schemes are proposed. HCI (Hilbert Curve Index) uses Hilbert Curve for presenting the locations of the spatial data items with integers. HCI in B-tree provides the distribution of the entire data items in (1, m) index scheme. HCI lets the client determine quickly the search space for a given NN query by the entire view of the distribution. Here, the search space means the clients have to access data items to determine NN to the query point. HCI lengthensthe access time of the clients for NN query because the search space by HCI is very large. Also, (1, m) indexing scheme of HCI lengthens the access time of the clients. With HCI, the clients prune out the candidates for NN after downloading the data items because HCI does not provide real location of the data items. It causes the clients to consume much large energy during processing given NN queries.

DSI (Distributed Spatial Index) has proposed to support the clients to process NN queries. DSI uses Hilbert curve to present locations of spatial data items [6,18]. In DSI, a data item has time pointers to data items separate exponentially in Hilbert curve value. This scheme causes the broadcast cycle to lengthen by repeating the Hilbert curve value of an item several times. The lengthened broadcast cycle results in the long access time of the clients. Also, with DSI, the clients prune out the candidates for NN after downloading them because DSI does not provide real locations of the data items. That results in the large energy consumption of the clients during processing NN queries. Thus, HCI and DSI using Hilbert curve value for locations of data items cannot avoid to prune candidates for NN after downloading them.

CHI (Cell-based Hybrid Index) has proposed to overcome the shortcomings of the HCI and DSI. CHI provides real locations in order to enable the clients to filter out the candidates for NN to the given query point before downloading them unlike HCI and DSI. Thus, Using CHI, the clients enables to process NN queries energy-efficiently, consuming less energy while finding NN.

Although CHI is in distributed indexing scheme, it provides the view of the distribution of the data items to be broadcast in units of cells [13,19]. The view allows the clients to determine quickly the search space for NN queries. However, CHI partitions the data space of data items to be broadcast into a regular grid. That causes the average index waiting time long when the data items are distributed in the skewed manner because the indexes for cells on the channel does not keep equi-distant. That results in the long access time of the clients.

In this paper, we propose NSPI to overcome the shortcoming of the CHI for improved access time of the clients.

## 3. NN Search with Non-uniform Space Partition

For the proposedNN search scheme for spatial data items, a broadcast server pushes data items with a distributed index for them over a wireless broadcast channel. A client tunes into the channel and decides the search space for NN that guarantees containing the NN using the distributed index on the wireless broadcast channel. Then, the client downloads the data items in the search space and determines the NN from the downloaded items. To support for clients to search NN efficiently, we propose Non-uniform Space Partition Index (NSPI).

### 3.1 Non-uniform Space Partition

In order to reduce the access time of the client in the wireless spatial data broadcast environment using the distributed index, it is critical to keep equi-distances on the broadcasting channel between the indexing tables holding the information on the broadcasting time of each data item. That results from that the equi-distant indexing tables reduce the average time for the client to meet the table after tuning into the wireless channel. We adopt the non-uniform space partition to allocate equi-distantly the index tables in the distributed manner using a cell-based allocation scheme. For the space partition, we follow the steps as below:

1) Row Partition

In the proposed scheme, we partition irregularly the data space, $S_{data}$, into $n$x$n$ cells, constrained to containing the same number of data items. For the $n$x$n$cell partition, the data space containing $N_{data}$data items for broadcasting is partitioned into $n$ rows in rectangle. Here each partitioned row has the same number of data items, therefore, each row has different size irregularly.

$$S_R = \{\ R_i \mid R_i \text{ is the } i\text{-th row }\}, \text{for} 0 \leq i < n \quad (1)$$

Here, $S_R$is the set of rows with $|S_R| = n$. $R_i$ is the $i$-th row from the bottom of the data space $S_{data}$ in the partition and has $N_R$ data items, $N_R = N_{data}$ /$n$.

2) Cell Partition
Each row is partitioned into n cells in irregular rectangle like row partition. Also all cells in $R_i$ has the same number of data items as each row has the same number of data items.

$$S_C = \{\ C_k \mid C_k \text{ is the } j\text{-th cell in } R_i, k = i \cdot n + j\ \},$$
$$\text{for} 0 \leq j < n \quad (2)$$

Here, $S_C$ is the set of cells with $|S_C| = n$. The cell $C_k$ is the $j$-th cell from the left of $R_i$ and has $N_C$ data items, $N_C = N_R$ /$n$.

For example, Figure 1 shows non-uniform space partition for a data space with 24 spatial data items. In the partition, each row has 6 data items and each cell in a row has 2 data items. Using the cells in the partition, the broadcast server disseminates data items in the unit of cells and places the indexing tables with the cells. That results in the equi-distance allocation of the indexing table and reduction of the access time of the clients in the broadcast environment.
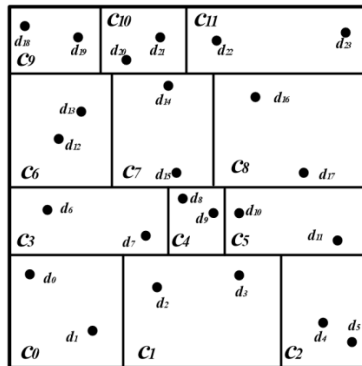


Figure 1. An example of non-uniform space partition for NN Search

3.2 Distributed Index for NN Search
For the efficient NN search in wireless data broadcast environment, the client needs two kinds of information. The one is the information about the distribution of all data items over data space $S_{data}$. This information enables the client to determine the search space that guarantees the NN. Also, to help the effective access of the client, the information about the distribution, it is critical that the size of the information keeps small. The other is the information about the data items to be broadcast. This information supports the client to prune the candidates of NN and download NN from the wireless channel.

In the proposed scheme using the non-uniform space partition, the broadcast server transmits the index information with two kinds of indexing tables, $IDX_{cell}$ and$IDX_{data}$, that provide the distribution of data items over the data space $S_{data}$ and data broadcast time, individually.

1) Data Distribution Information using $IDX_{cell}$
$IDX_{cell}$ provides the information on the data distribution in the unit of cells of the non-uniform partition. $IDX_{cell}$ carries the information about cells using the two corners of cells, the number of data items in cells and the broadcasting time of cells on the wireless broadcast channel.$IDX_{cell}$is configured as follows:

$$IDX_{cell} = \{I_{Ci} \mid I_{Ci} = <LL,\ UR,\ N_C,\ T_{Ci}>,\ 0 \leq i < n^2\} (3)$$

In the equation (3), the table$IDX_{cell}$ is the set of $I_{Ci}$.Here, $I_{Ci}$is a tuple that means cell information for cell $C_i$. $LL$ and $UR$ mean the lower-left point and upper-right point of the cell $C_i$, respectively. $N_C$ is the number of data items in cell $C_i$ and $T_{Ci}$ is the broadcasting time of the index for data items in $C_i$.

Figure 2 shows the table $IDX_{cell}$ for the cells in Figure 1. Through $IDX_{cell}$, the client is aware of the distribution of data items in the units of cells.

| $LL_{C0}$ | $UL_{C0}$ | $N_{C0}$ | $T_{C0}$ |
|---|---|---|---|
| $LL_{C1}$ | $UL_{C1}$ | $N_{C1}$ | $T_{C1}$ |
| … | … | … | … |
| $LL_{C11}$ | $UL_{C11}$ | $N_{C11}$ | $T_{C11}$ |

Figure 2. $IDX_{Cell}$ for cells in Figure 1

2)    Data Information using $IDX_{data}$

$IDX_{data}$ offers the information of the locations of the data items in cell $C_i$ and the broadcasting time of each data item in $C_i$.

$$IDX_{data}(i)= \{I_{dk} \mid I_{dk} = \; <(d_{k\_x}, d_{k\_y}), T_{dk}>, d_k \in C_i\}$$

(2)

In equation (2), the table $IDX_{data}(i)$is a set of $I_{dk}$. $I_{dk}$is a tuple that means the information on data $d_k$ in $Ci$. $(d_{k\_x}, d_{k\_y})$ is the coordinates of $d_k$and $T_{dk}$ is the broadcasting time of $d_k$.

For example, Figure 3 shows $IDX_{data}(4)$ for cell $C_4$ in Figure 1. As shown in Figure 3, the $IDX_{data}$ provides the locations of each data items for the client to prune the candidates of NN and deter mine NN among the data in cells of search space for NN.

| $( d_{8\_x}, d_{8\_y})$ | $T_{d8}$ |
|---|---|
| $( d_{9\_x}, d_{9\_y})$ | $T_{d9}$ |

Figure 3. $IDX_{data}(4)$ for cell $C_4$ in Figure 1

3.**3 Processing an NN Search**

In the proposed scheme for searching NN from a given query point using the distributed indexing tables using the non-uniform space partition, the client follows the steps as below:

- Determine the search space guaranteeing NN using $IDX_{cell}$.
- Pruning the candidates of NN using $IDX_{data}$.
- Determine NN and download NN from the channel

1)    Determine the search space

For deciding the search space that guarantees containing the NN, the client accesses the table$IDX_{cell}$ after tuning into the wireless broadcast channel. The client calculates the maximum distance $MaxDIST_i$ from the given query point to cell $C_i$ using $LL$ and $UL$ of $C_i$. Then the client determines the minimum of $MaxDIST_i$for all cells as the $NNDist$ for the given query point. Here, the $NNDist$ is a radius of the minimum circle with the query point as the center, guaranteeing the NN.

For example, Figure 4 shows the $MaxDIST$ of $C_1$ and $C_4$. For the given query point QP, the $NNDist$ is determined with the $MaxDIST_4$, the minimum $MaxDIST$s to all cells.
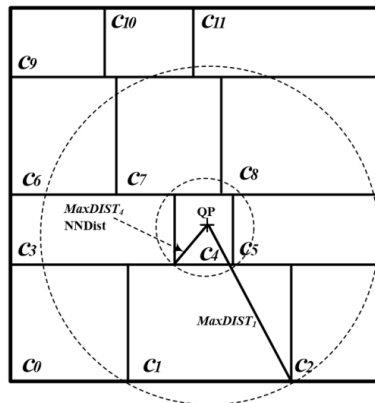


Figure 4. $MaxDIST$and$NNDist$

2)  Pruning the candidates of NN using $IDX_{data}$

The client accesses all cells overlapped with the search space, i.e., the circle with *NNDist* as the radius in order to prune the candidates and determine the NN using $IDX_{data}$ of each cell.

For example, in order to determine the NN from the given query point in Figure 4, the client has to access cell $C_1$, $C_3$, $C_4$, $C_5$, $C_7$ and $C_8$. They are overlapped with the circle of NNDist. Accessing each cell, the client calculates the distance from the query point and to each data in the accessed cell, then finds the minimum of the calculated distances, and finally determines NN as the data item with the minimum distance.

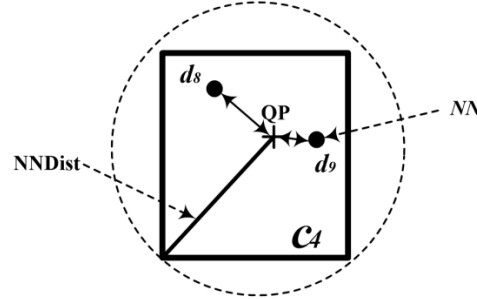Figure 5 shows the procedure for pruning the candidates of NN and determining NN in cell $C_4$.



Figure 5. Pruning candidates of NN and determining NN in $C_4$

3)  Download  NN using $IDX_{data}$

After determining the NN for the given query point, the client downloads the NN from the wireless broadcast channel using the broadcasting time information $T_{dk}$ in $IDX_{data}$.

Algorithm 1 shows the procedure for searching NN using the distributed index based on the non-uniform space partition.

| NN Search |
|---|
| **Input**: QP, query point |
| **Output**: NN, the nearest data item from QP |
|  |
| 1: Access the first $IDX_{cell}$ on the broadcast channel |
| 2: Determine the *NNDist* using $I_{ci}$. |
| 3: Decide the search space by *NNDist* |
| 4: Determine $Q_{cell}$, the priority queue of broadcasting times of cells overlapped with the search space |
|  |
| 5: *NNDist*← ∞ |
| 6: While $Q_{cell} \neq \Phi$ do |
| 7: $t_{access}$← dequeue from $Q_{cell}$ |
| 8:    Access $IDX_{data}(i)$ for the cell disseminated at $t_{access}$ |
| 9:    Calculate the distance, dist, from QP to $d_k$ |
| 10: if *dist<NNDist* |
| 11:      then *NNDist*← *dist* |
| 12:      download the data at $T_{dk}$ from the channel |
| 13:   Replace NN with the data item |

Algorithm 1. The procedure for NN search.

Line 1 and 2 in the algorithm depicts the procedure of determining *NNDist*. In Line 3 and 4, the client determines search space $Q_{cell}$, that is the priority queue of broadcasting times of cells overlapped with the circle with *NNDist* as its radius.

With $Q_{cell}$, the client accesses all cells overlapped with the search space. At line 5 in the algorithm, the client resets *NNDist* into the infinity in order to find NN because the cells to be accessed are determined. The client accesses all the cells at the broadcasting times in $Q_{cell}$ in line 6 and 7. With $IDX_{data}(i)$ of accessed cell $C_i$, the client calculates *dist,* the distance from the query point to the data item, and updates *NNDist* with *dist* when *dist* is smaller than *NNDist*. Thus, the client renews *NNDist* in the search procedure and determines NN from the query point.

Algorithm 2 shows that the broadcast server partitions the data space, organizes indexing table $IDX_{cell}$ and $IDX_{data}$, and schedules the indexing tables and data items.

| Data Indexing & Scheduling |
|---|
| **Input**: D, a set of $N_{data}$ spatial data items |
| **Output**: $IDX_{cell}$ and $IDX_{data}$, indexing Table |
| $Q_{bcast}$, the queue for scheduling of the broadcasting |
|  |
| 1: $S_R \leftarrow$ Split $S_{data}$ into $n$ rows with $N_R$ ($= N_{data} / n$) items |
| 2: foreach $n$ rows do |
| 3: $\quad S_C \leftarrow$ Split $S_R$ into $n$ cells with $N_C$ ($= N_R / n$) items |
|  |
| 4: Organize $IDX_{cell}$ with cells in $S_C$ |
| 5: foreach cell $C_i$ in $S_C$ do |
| 6: $\quad$ Organize $IDX_{data}$ of $C_i$ in $S_C$ |
|  |
| 7: for i from 0 to n-1 do |
| 8: $\quad$ enqueue $IDX_{cell}$ to $Q_{bcast}$ |
| 9: $\quad$ for j from 0 to n-1 do: |
| 10: $\quad\quad$ $k \leftarrow i \cdot n + j$ |
| 11: $\quad\quad$ enqueue $IDX_{data}(k)$ to $Q_{bcast}$ |
| 12: $\quad\quad$ enqueue all data items in $C_k$ |

Algorithm *2*. The procedure for data indexing and scheduling.

The server splits the data space $S_{data}$ into $S_R$, $n$ rows, and $S_C$, $n^2$ cells, for organizing the indexing tables from line 1 to line 3 of Algorithm 2. Using the partitioned rows and cells, $S_R$ and $S_C$, the server organizes the two indexing tables, $IDX_{cell}$ and $IDX_{data}$ in line 4 to line 6.

Then, the server schedules the two indexing tables and data items using $Q_{bcast}$ to broadcast in the units of rows. For the i-th row, the server enqueues $IDX_{cell}$ to $Q_{bcast}$, $IDX_{data}$ for cell $C_k$ in the row and data items in $C_k$. Line 7 to line 12 shows the scheduling procedure.

## 4. Performance Evaluation

We show the effectiveness of the proposed NSPI by comparing the performance of NSPI with other indexing schemes supporting NN search.

For evaluating the performances of NSPI by simulation studies, we implemented a simulation testbed for the wireless data broadcasting. In the testbed, we implemented a broadcast server, a client and a wireless channel. We use 10432 American historical sites as skewed spatial data items. Table 1 shows the parameters for simulation and their values.

Table 1. The parameter values for simulation

| Parameter | Value |
|---|---|
| Cell ID Number | 8 bytes |
| Time Pointer | 8 bytes |
| Coordinate | 8 bytes |
| Data Item | 1024 bytes |
| Bucket Size | 64 bytes |

The bucket in Table 1 is the unit of data delivery like packet in a network. We set the size of the bucket for the simulation with 64 bytes.

We compare the proposed NSPI with DSI, HCI, and CHI that are existing indexing schemes supporting a client to search NN. For the comparison with the proposed scheme, we also implemented DSI, HCI and CHI as the existing indexing schemes [4, 5].

While simulating the testbed, the client measures the access time and the tuning time every conduction of a NN query. The client runs 100,000 NN queries during simulation and we measure and use the average access time and tuning time for the comparison.

Comparison of the access time

The access time shows how long time the client takes till downloading NN from the wireless channel since tuning into the channel for processing a given NN query. Thus, reducing the access time is the critical performance issue.

Figure 6 depicts the access time of NSPI and other 3 existing indexing schemes in the number of buckets. NSPI shows shorter access time than that of the other indexing schemes. That results from the two reasons. The first reason is that the proposed NSPI keeps the distributed indexing table equi-distant on the wireless channel by adopting the non-uniform space partition. That results to the reduction of the index waiting time of the clients and to shorter access time. The second reason is that the proposed NSPI makes the search space for the given query point smaller than the other indexing scheme. That makes the average access time shorter and improves the performance of the client.
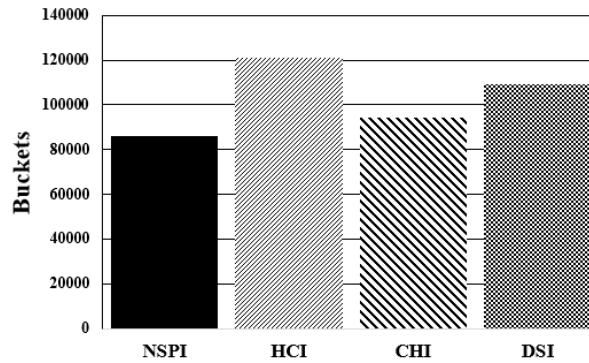


Figure 6. The comparison of the access time

Comparison of the tuning time

As a performance metric, the tuning time shows how much energy the client consumes while processing a given NN query. The client accesses the indexing tables in the energy-consuming active mode on the wireless channel in order to decide the search space. Then the client switches to the doze mode until the data items in the search space appears on the channel. The client switches to the active mode in order to filter and download the NN item.

The tuning time is the total time for the client to stay in the active mode during processing a given NN query. Thus, the tuning time shows the amount of energy consumption of the client in the unit of time. Figure 7 depicts the tuning time of the NSPI and other indexing schemes.
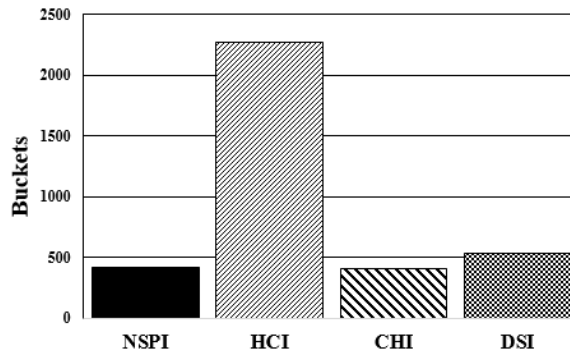


Figure 7. The comparison of the tuning time

The tuning time of NSPI is smaller than HCI and DSI and similar to CHI. That results from the search space determined by indexing scheme. HCI shows much longer tuning time because HCI lets the client decide the search space by the B-tree based Hilbert curve index. DSI shows smaller tuning time than HCI but longer than CHI and NSPI. This is because DSI uses the exponential indexing scheme of Hibert curve value of data items, not real locations of them, for determining the search space. Because NSPI and CHI carry real coordinates of data items, the client filters out candidates of NN and determines NN efficiently. That affects the reduction of the tuning time.

## 5. Conclusion

In this paper, we proposed the indexing scheme with a non-uniform space partition and the algorithm to process NN search on the wireless data broadcast channel. The indexing scheme provides the clients with the information on the distribution of the data items in the level of cells. Also, the scheme supplies the information on data items in each cell. Using the indexing scheme, the clients determine the search space for NN and then download the NN from the channel. The proposed scheme based on the non-uniform space partition enables the indexes to keep equidistance on the wireless channel. That results in the quick access time of the clients because the reduced time until the clients meet the index on the channel. We show the effectiveness of the proposed scheme by simulation studies. The access time of the proposed scheme is less than that of existing indexing schemes.

## 6. Acknowledgment

## References

1. Dong-Hwan Gong. (2019). IoT Device Testing for Efficient IoT Device Framework. *International Journal of Internet, Broadcasting and Communication*, 12(2), 77 –82.
2. Sung GeunYoo, Sangil Park and Won-Young Lee. (2020). A Study of Time Synchronization Methods for IoT Network Nodes, *International Journal of Advanced Smart Convergence*, 9(1), 109 – 112.
3. Yong-Kwang Kwonand Sun-Young Kim. (2020). Trend Analysis of IoT Technology Using Open Source, *The Journal of the Institute of Internet, Broadcsting and Communication*, 20(3), 65 –72.
4. Soowook Lee and Manyong Han. (2019). Utilization and Analysis of Big-data, *International Journal of Advanced Culture Technology*7(4), 92 – 99.
5. Dongyoung Shin and JaeWan Park. (2020). Modularization for Personal Social Service Robots, *The Journal of the Convergence on Culture Technology*, 6(2), 349 – 355.
6. B. Zheng, W. C. Lee, K.C.K Lee, D. L. Lee, and M. Shao. (2009). A distributed spatial index for error-prone wireless data broadcast. - VLDB J., 18(4), 959-986.
7. B. Zheng, W. C. Lee, and D. L. Lee. (2004). Spatial Queries in Wireless Broadcast systems. - Wireless Network, 10(6), 723-736.
8. S. Im, H. Youn, J. Choi, and J. Ouyang. (2011). A novel air indexing scheme for window query in non-flat wireless spatial data broadcast. - Journal of Comm. and Networks, 13(4), 400-407
9. K. Tao, Y. Kou, X. Lu, and R.Song. (2015). Design of data distribution protocols for next generation broadcast wireless systems. - 2015 10th International Conference on Communication and Networking in China, Shanghai.
10. A. B. Waluyo, F. Zhu, D. Taniar, and B. Srinivasan. (2014). Design and Implementation of a Mobile Broadcast System - 2014 IEEE 28th International Conference on Advanced Information Networking and Applications, Vitoria, Canada
11. J. Kuang, and S. Yu. (2017). Broadcast-based Content Delivery in Information-Centric Hybrid Multihop Wireless Networks. - IEEE Comm. Letters, vol. 21, no. 4, pp. 889-892.
12. A. B. Waluyo, F. Zhu, D. Taniar, and B. Srinivasan. (2014). Design and Implementation of a Mobile Broadcast System - 2014 IEEE 28th International Conference on Advanced Information Networking and Applications, Vitoria, Canada.
13. Seokjin Im and Hee Yong Youn. (2008), A Cell-based Hybrid Indexing Scheme for Energy Conserving k Nearest Neighbor Search on Air, *IEICE Transactioins on Communications,* E91-B(11), 3799 –3802.
14. Seokjin Im and HeeJoung Hwang. (2017), Quick Window Query Processing using a Non-uniform Cell-based Index in Wireless Data Broadcast Environment, *IEICE Transactioins on Fundamentals of Electronics, Communications and Computer Sciences,* E100-A(4), 1092 – 1096.
15. Taranjot Kaur Bajwa, Sourav Garg and Kumar Saurabh. (2016), GAIT analysis for identification by using SVM with k-NN and NN techniques, *2016 4th International Conference on Parallel, Distributed and Grid Computing*
16. Dawei Li and Mooi Choo Chuah. (2015), A Novel Unsupervised 2-Stage k-NN Re-Ranking Algorithm for Image Retrieval, *2015 IEEE International Symposium on Multimedia*
17. Guobin Li and Jine Tang. (2010), A new K-NN query algorithm based on the dynamic movement of spatial objects. 2010 International Conference on Educational and Information Technology

18. Adle, A. A., &Akdemir, Ö. (2019). Achieving Competitive Advantage In Technology Based Industry: How Developing Intellectual Capital Matters. International Journal OfEbusiness And Egovernment Studies, 11(2), 89-103.

19. Auriacombe, C. J., & Vyas-Doorgapersad, S. (2019). Critical Considerations For The Role Of Governments In The Interface Between Good Governance And Sustainable Development In Developing Countries.