

A Skewed Spatial Index for Continuous Window Queries in the Wireless Broadcast Environments

Jun-Hong Shen, Ching-Ta Lu, Ming-Shen Jian and Tien-Chi Huang

Abstract Location-based services (LBSs) via wireless data broadcast can provide a huge number of mobile clients for simultaneously accessing spatial data according to their current locations. A continuous window query is one of the important spatial queries for LBSs. It retrieves spatial objects in a fixed window region of every point on a line segment and indicates the valid segments of them. In this paper, we propose a skewed spatial index for continuous window queries considering skewed access patterns in the wireless broadcast environments.

Keywords Continuous window queries · Skewed access patterns · Spatial index · Wireless data broadcast

1 Introduction

Due to its high scalability, wireless data broadcast enables a huge number of mobile clients simultaneously access data at anytime and anywhere. With the advances in positioning systems, e.g., GPS, location-based services (LBSs) via wireless data broadcast provide the mobile clients with applications according to their current locations. A continuous window query is one of important spatial queries for such applications. It retrieves spatial objects in a fixed window region

J.-H. Shen (✉) · C.-T. Lu
Department of Information Communication, Asia University, Taichung, Taiwan
e-mail: shenj@asia.edu.tw

M.-S. Jian
Department of Computer Science and Information Engineering,
National Formosa University, Yunlin County, Taiwan

T.-C. Huang
Department of Information Management, National Taichung University
of Science and Technology, Taichung, Taiwan

of every point on a line segment and indicates the valid segments of them. For example, a tourist issues a query asking attractions within one mile according to his location when driving. Since the tourist is moving, the answered result should contain attractions within the query region and indicate their valid segments.

Due to the limited battery life of mobile devices, energy conservation is a crucial issue for lengthening the recharging period of the devices. Spatial indexes are interleaved with spatial objects to save energy consumption. In this case, the mobile devices can skip unwanted data by slipping into the doze mode, and stay in the active mode to listen the wireless channel only when the desirable data is arrived [5, 7].

In real life applications, a few hot attractions are frequently visited than the others, i.e., skewed access patterns. Under such the scenario, more popular data should appear more times in a broadcast cycle than less popular ones, resulting in the decrease of the average client waiting time, i.e., a nonuniform broadcast [6, 8]. In this paper, we focus on the design of the spatial index for supporting continuous window queries in the nonuniform broadcast.

In the literature, many spatial indexes are proposed to support the efficient query processing of spatial queries in the wireless broadcast systems. Distributed spatial index [9] and the neighbor-index method [7] were proposed to support continuous window queries over wireless data broadcast. Considering skewed access patterns, a grid-based distributed index (GDIN) [2] was developed for window queries in the nonuniform broadcast by using a regular space partition. Multi-leveled air index scheme (MLAIN) [3] was developed to improve the disadvantage occurred in GDIN by using a multi-level space partition. These methods were designed for broadcasting data on the one wireless channel. A grid-based indexing scheme on multiple channels (GRIM) [4] was proposed to support window queries considering skewed access patterns on the multiple-channel broadcast.

To the best of our knowledge, there is no existing work on spatial indexes for supporting the continuous window queries with skewed access patterns. In this paper, we propose a skewed spatial index for efficiently processing continuous window queries according to the clients' skewed access patterns in the wireless broadcast systems. To keep a good spatial locality, our proposed method partitions a data space into cells and allocates them in the one-dimensional space according to the Hilbert curve. Then, our proposed method applies Acharya et al. [1] Broadcast Disks to allocate popular spatial objects contained in the cells more times on the wireless channel than regular ones, resulting in quick access for the popular objects. The cell indexes considering the nonuniform broadcast are interleaved with the spatial objects on the wireless channel to support efficient access.

The rest of this paper is organized as follows. In Sect. 2, we give a brief description of the related techniques applied in our proposed method. In Sect. 3, we present our proposed skewed spatial index. Finally, a conclusion is presented in Sect. 4.

2 Background

In this paper, we apply Acharya et al.'s Broadcast Disks [1] to organize spatial objects into a popularity hierarchy which results in a skewed transmission of objects and quicker access to more popular objects. To preserve the spatial locality of objects, we allocate spatial objects into the wireless channel in the order of the Hilbert curve. Broadcast Disks and the Hilbert curve are briefly described in this section.

2.1 Broadcast Disks

Acharya et al. [1] have proposed the use of a periodic dissemination architecture in the context of wireless mobile systems, called Broadcast Disks (*BD*). The algorithm has the following steps:

1. Order data items ($=N$) from the hottest (most popular) to the coldest.
2. Partition the list of the data items ($=N$) into multiple disks ($=S$ disks), where each disk D_i , $1 \leq i \leq S$, contains pages ($=K_i$) with similar access probabilities. That is, $N = \sum_{i=1}^S K_i$.
3. Choose the relative frequency λ_i of broadcast for each disk D_i , $1 \leq i \leq S$.
4. Split each disk into a number of smaller units, called chunks C_{ij} , where C_{ij} denotes the j th chunk in disk D_i . First, calculate L as the LCM (Least Common Multiple) of the relative frequencies. Then, split each disk D_i into $NC_i = L/\lambda_i$ chunks, $1 \leq i \leq S$, where NC_i denotes the number of chunks in disk D_i .
5. Create the broadcast program by interleaving the chunks of each disk in the following manner:

```

for  $i := 1$  to  $L$  do
begin
  for  $j := 1$  to  $S$  do
  begin
     $k := ((i - 1) \bmod NC_j) + 1$ ;
    Broadcast chunk  $CK_{j,k}$ ;
  end;
end.

```

2.2 Hilbert Curve

The Hilbert curve is a continuous path which passes through every point in a multi-dimensional space once to form a one-one correspondence between the coordinates of the points and the one-dimensional sequence numbers of the points on the

curve [7]. It can preserve the spatial locality of points. The spatial locality means that points that are close to each other in a multi-dimensional space are remained to close to each other in a one-dimensional space. The Hilbert curve of order n recursively divides the space into four equal-sized cells and gives each cell a sequence number from 0 to $(2^{n*2} - 1)$. Figure 1a shows the Hilbert curve of order 1. The curve can keep growing recursively by following the same rotation and reflection pattern at each cell of the basic curve. Figure 1b shows the Hilbert curve of order 2.

3 The Proposed Skewed Spatial Index

In our proposed method, a data space containing N spatial objects is recursively partitioned to $2^n \times 2^n$ cells according to the Hilbert curve of order n ; that is, the data space is processed by an n -level partition. Moreover, each cell contains η spatial objects. The cell containing popular objects is called the hot cell. We apply Acharya et al.'s BD [1] to allocate the spatial objects on the wireless channel. The spatial objects are sorted in the descending order of their access probabilities, and partitioned into S disks, D_1-D_s . In each disk, the sequence of the spatial objects is following the sequence of the Hilbert curve.

Take spatial objects shown in Fig. 2 for example. In the figure, object $O8$ is the most popular object, objects $O4$, $O10$ and $O11$ are popular objects, and the rest are regular objects. The data space is processed by a two-level partition and partitioned into $2^2 \times 2^2$ cells according to the Hilbert curve of order 2. Each cell contains one spatial object, i.e., $\eta = 1$. A cell of the Hilbert curve of order i is denoted by $C(i,j)$, where i is the order of the Hilbert curve, and j is the sequence number of the Hilbert curve of order i , $0 \leq j \leq 2^{i*2} - 1$. The parent cell of the Hilbert curve of order $(i-1)$ of $C(i,j)$ can be easily derived, i.e., $C(i-1, \lfloor j/4 \rfloor)$. For example, object $O1$ is contained in cell $C(2,0)$ of the Hilbert curve of order 2. The parent cell of the Hilbert curve of order 1 of $C(2,0)$ is $C(2-1, \lfloor 0/4 \rfloor) = C(1,0)$.

Assume that the spatial objects are partitioned to 3 disks according to their access probabilities, i.e., $S = 3$, as shown in Fig. 3a. In each disk, the spatial

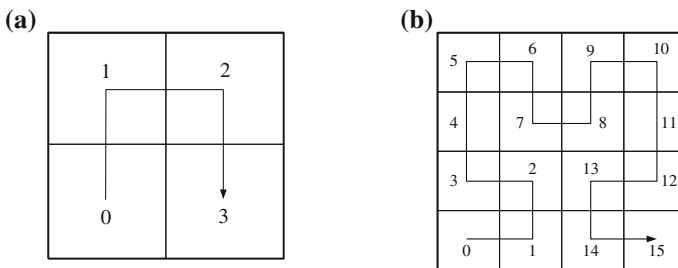


Fig. 1 The Hilbert curve: a order 1; b order 2

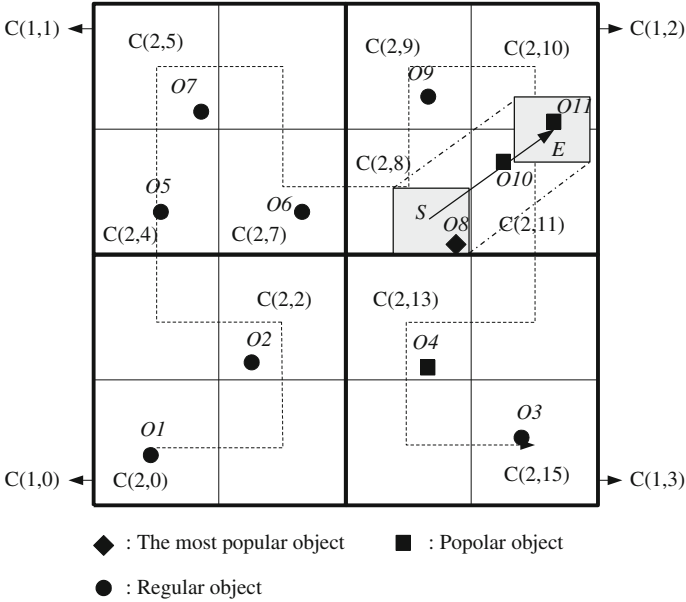


Fig. 2 Spatial objects allocated in the order of the Hilbert curve

objects are allocated in the sequence of the Hilbert curve of order 2. The relative frequencies of disks D_1 , D_2 , and D_3 are set to $\lambda_1 = 4$, $\lambda_2 = 2$, and $\lambda_3 = 1$, respectively. In this case, the least common multiple of the relative frequencies, L , is 4. Then, D_1 has $NC_1 = L/\lambda_1 = 4/4 = 1$ chunk, D_2 has $NC_2 = L/\lambda_2 = 4/2 = 2$ chunks, and D_3 has $NC_3 = L/\lambda_3 = 4/1 = 4$ chunks, as shown in Fig. 3b. For example, D_2 has 3 objects partitioned to 2 chunks, $CK_{2,1}$ and $CK_{2,2}$. The chunks are interleaved to the wireless channel according to Step 5 of Broadcast Disks mentioned in Sect. 2.1. Figure 3c shows the interleaved result consisting of 4 minor cycles, m_1 - m_4 , which contain one chunk from each disk. Each minor cycle has a hot cell group containing hot cells. This broadcast result produces a three-level memory hierarchy in which D_1 is the smallest and fastest level and D_3 is the largest and slowest level.

3.1 Index Structure

Similar to MLAIN, our proposed skewed spatial index has two kinds of indexes: the hot cell index and the cell index. The hot cell index is interleaved before each hot cell group. The cell indexes of the Hilbert curve from order 1 to order $(n-1)$ are interleaved before the first element of the Hilbert curve of the next order. Moreover, the cell index of the Hilbert curve of order n is interleaved before each cell of the Hilbert curve of order n . Figure 4 shows the index allocation of the broadcast

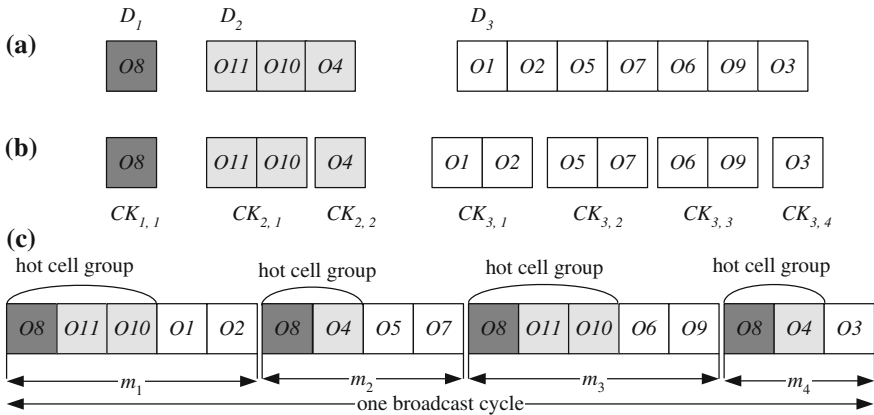


Fig. 3 The allocation of spatial objects: **a** three disks with the corresponding objects; **b** chunks for each disk; **c** the allocation of the spatial objects

spatial objects shown in Fig. 3c. Take object *O1* for example. It is the first element contained in $C(1,0)$ via the first-level partition. Moreover, it is contained in $C(2,0)$ via the second partition. Therefore, the sequence of the index allocation for *O1* is the cell index of $C(1,0)$ followed by the cell index of $C(2,0)$. In Fig. 4, a hot cell index is interleaved before each hot cell group.

The hot cell index contains the arrival time of the nearest cell index of the Hilbert curve of order 1 that will be broadcast on the channel (*NR*), and the hot cell information (*HCI*). *HCI* contains the arrival time of the cell indexes for all hot cells. The entries in *HCI* are of form $\langle C(i,j), t \rangle$, where $C(i,j)$ is the cell identifier and t is the arrival time of the cell index for that cell. Take the first hot cell index shown in Fig. 4 for example. *NR* stores the arrival time, t_4 , of the nearest cell index for cell $C(1,0)$, which is of the Hilbert curve of order 1. *HCI* stores the information about the hot cells, $C(2,8)$, $C(2,10)$, $C(2,11)$ and $C(2,13)$.

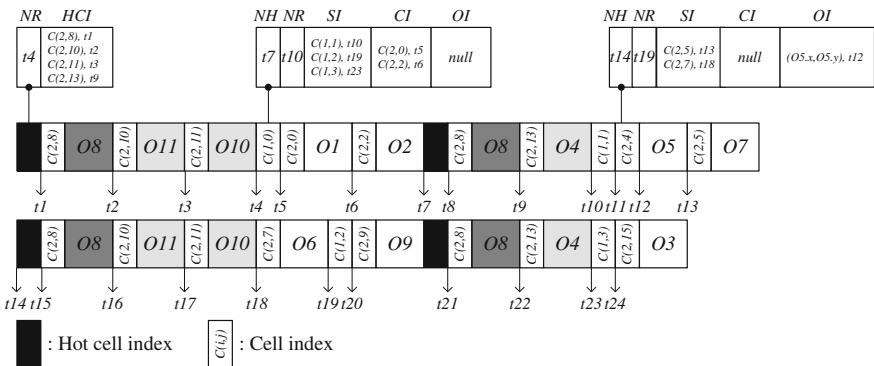


Fig. 4 The index information interleaved with the spatial objects

The cell index contains the arrival time of the nearest hot cell index that will be broadcast on the channel (NH), NR , sibling information (SI), child information (CI) and object information (OI). OI stores the coordinates of the objects in that cell, and their corresponding arrival time. This enables that the query process can check if the objects are within the query region before accessing them. Take the cell index for cell $C(1,0)$ for example. NH stores the arrival time, $t7$, of the nearest hot cell. NR stores the arrival time, $t10$, of the cell index for the cell of the Hilbert curve of order 1, $C(1,1)$. SI stores sibling information about the cells of the Hilbert curve of the same order, $C(1,1)$, $C(1,2)$ and $C(1,3)$. CI stores child information about the cells of the Hilbert curve of the next order, $C(2,0)$ and $C(2,2)$. Since cell $C(1,0)$ does not contain object information, OI is null in this case. Take the cell index for cell $C(2,4)$ for another example. NH stores the arrival time, $t14$, of the nearest hot cell. NR stores the arrival time, $t10$, of the cell index for the cell of the Hilbert curve of order 1, $C(1,2)$. SI stores sibling information about the cells of the Hilbert curve of the same order, $C(2,5)$ and $C(2,7)$. Since cell $C(2,4)$ is not further partitioned, it does not contain any child, i.e., $CI = \text{null}$. OI stores object information in cell $C(2,4)$, i.e., the coordinates of object $O5$ and its arrival time, $t12$.

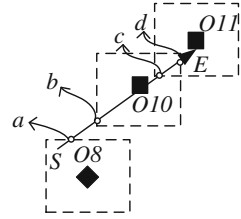
3.2 Continuous Window Queries

For a continuous window query, the cells of the Hilbert curve of order n overlapped with the query region of the continuous window query should be examined. Those cells are recorded in candidate cell set CCS , and their corresponding cell indexes are further examined to check if the spatial objects are within the query region via their coordinates stored in OI before accessing them. The access protocol is as follows.

1. Tune into the wireless channel to get the arrival time of the following nearest hot cell index, and proceed to it.
2. Examine entries in the hot cell index to get the arrival time of the cell indexes overlapped with the query region, and put the arrival time into queue CQ in the ascending order. Note that CQ is a queue storing the arrival time of the next visited index or object in the ascending order.
3. If all information about the cells in CCS can be found out in the hot cell index, follow the arrival time in CQ to retrieve the cell indexes to get the corresponding objects.
4. Otherwise, put the arrival time in NR into CQ in the ascending order. Find out the rest cells in CCS and put their arrival time into CQ by using SI and CI stored in the visited cell indexes. Follow the arrival time in CQ to retrieve the cell indexes to get the corresponding objects.

Take the continuous window query for segment \overline{SE} shown in Fig. 1 for example. In Fig. 1, all the objects covered by the polygon region should be retrieved. Therefore, cells $C(2,8)$, $C(2,10)$ and $C(2,11)$ overlapped with the query region

Fig. 5 Minkowski regions



should be examined, i.e., $CCS = \{C(2,8), C(2,10), C(2,11)\}$. Assume that the client tunes into the broadcast channel at the beginning of the first hot cell index shown in Fig. 4. After examining HCI in this hot cell index, the client has the arrival time of the cell indexes of $C(2,8)$, $C(2,10)$ and $C(2,11)$, i.e., $CQ = [t1, t2, t3]$. In this case, since all cell indexes for the cells in CCS are found out via the hot cell index, the client does not need to visit other cell indexes to process the query process. Following the sequence in CQ , the client checks OI in these cell indexes to determine if the spatial objects are within the query region and then retrieves them from the wireless channel. The answered spatial objects for this query are $O8$, $O10$ and $O11$.

For a continuous window query, the answered objects are only valid along part of the query line segment, i.e., validity segments [9]. Therefore, the answered objects should be assigned with their corresponding validity segment. The validity segment of each answered object can be determined by its Minkowski region [9]. The Minkowski region (Fig. 5) has the same size as the query window and centers at the answered object. The validity segment of the answered object is the intersection of the query line segment with its Minkowski region. In Fig. 1, the answered objects for query line segment \overline{SE} are objects $O8$, $O10$ and $O11$. Their corresponding Minkowski regions are shown in Fig. 4. The validity segments of objects $O8$, $O10$ and $O11$ are \overline{Sa} , \overline{ba} and \overline{ca} , respectively.

4 Conclusion

In this paper, we have addressed the problem of processing continuous window queries for skewed access patterns in the wireless data broadcast environments. We have proposed a skewed spatial index for efficient processing such queries. In our proposed method, popular spatial objects are broadcast more often than regular ones, resulting in quick access of the popular objects.

Acknowledgments This research was supported in part by the National Science Council of Republic of China under Grant No. NSC 99-2221-E-468-019 and Grant No. NSC 100-2511-S-025-002-MY2.

References

1. Acharya S, Franklin M, Zdonik S, Alongso R (1995) Broadcast disks: data management for asymmetric communications environments. In: Proceedings of the 1995 ACM SIGMOD international conference on management of data, pp 199–210
2. Im S, Youn HY, Choi J, Ouyang J (2011) A novel air indexing scheme for window query in non-flat wireless spatial data broadcast. *J Commun Netw* 13(4):400–407
3. Im S, Choi J (2012) MLAIN: multi-leveled air indexing scheme in non-flat wireless data broadcast for efficient window query processing. *Comput Math Appl* 64:1242–1251
4. Im, S Choi J (2012) Quick data access on multiple channels in non-flat wireless spatial data broadcasting. *IEICE Trans Commun E95-B(9)*:3042–3046
5. Shen JH, Chang YI (2007) A skewed distributed indexing for skewed access patterns on the wireless broadcast. *J Syst Softw* 80(5):711–723
6. Shen JH, Chang YI (2008) An efficient nonuniform index in the wireless broadcast environments. *J Syst Softw* 81(11):2091–2103
7. Shen JH, Lu CT, Jian MS (2013) Neighbor-index method for continuous window queries over wireless data broadcast. *Appl Mech Mater* 284–287:3295–3299
8. Yee WG, Navathe SB, Omiecinski E (2002) Efficient data allocation over multiple channels at broadcast servers. *IEEE Trans Comput* 51(10):1231–1236
9. Zheng B, Lee WC, Lee CK, Lee DL, Shao M (2009) A distributed spatial index for error-prone wireless data broadcast. *VLDB J* 18(4):959–986