Continuous Spatial Queries via Wireless Data Broadcast

KwangJin Park
Department of Computer Science and Engineering
Korea University
Anam-Dong, Seoul, Korea
kjpark@disys.korea.ac.kr

MoonBae Song
Department of Computer Science and Engineering
Korea University
Anam-Dong, Seoul, Korea
mbsong@gmail.com

Chong-Sun Hwang
Department of Computer Science and Engineering
Korea University
Anam-Dong, Seoul, Korea
hwang@disys.korea.ac.kr

ABSTRACT

This paper proposes a generic framework for continuous range queries via wireless data broadcast. This framework distinguishes itself from existing work by being the first to address the continuous spatial query issue, without an index, making effective use of the low wireless bandwidth, and therefore being ideal for achieving maximal scalability with the fastest access time. The task of the query processor is to selectively monitor the wireless broadcast channel, as the data items are disseminated according to their location by the server.

Keywords
location-based services, spatial query, air index

1. INTRODUCTION

The recent advance in wireless networks and computer downsizing techniques have led to the development of mobile computing. An important step toward the realization of this mobile environment is to be able to disseminate timely and relevant information to the user anytime, anywhere. Examples of these applications include location-aware services, traffic monitoring, and emergency services. Location-aware applications have distinguishing characteristics, such as a large number of mobile and stationary objects, and consequently a large number of mobile and stationary queries [1]. An important class of queries in location-aware services is Continuous Range Query (CRQ), which continuously finds objects located in a query range, while the client moves to the destination.

Definition 1.1(Continuous Range Query): Given a destination from s to e, continuously determine the set $S_{CRQ} \subseteq S$ that contains objects within the search range.

Unlike snapshot queries that are only evaluated once, continuous queries require continuous evaluation as the query result becomes invalid with the change of information [2]. For example, one such query can be posed by a user as: “give me the positions and names of the gas stations within the 3 miles along the route from current location to the central station”. The spatial range query is derivative of a complex query. The query response time is greatly affected by the order in which data items are broadcast.

There are two basic approaches to disseminating data to mobile clients, the push-based and pull-based approaches [3]. In the push-based approach, data items are broadcast periodically on the downlink channel. Its major advantage is that it can be accessed concurrently by any number of clients without resulting in any performance degradation. In the pull-based approach, the client requests a piece of data items through the uplink channel and the server responds by transmitting the data to the client on the downlink channel.

In the broadcast-based model, the broadcasting of data together with an index structure is an effective way of disseminating data in a wireless mobile environment [1, 4]. Indexing(directory of the file) can be used to guide the client in the listening process. The client is able to predict the arrival time of the desired data items, by first accessing the broadcast index. Indexing techniques can be evaluated in terms of the following factors: Access Time(AT) and Tuning Time(TT). AT is the average time elapsed from the moment a client issues a query to the moment the required data item is received by the client and TT is the amount of time spent by a client listening to the channel. Then, AT consists of two separate components, namely: Probe Wait: The average duration for getting to the next index segment. If we assume that the distance between two consecutive index segments is $L$, then the probe wait is $L/2$. Bcast Wait: The average duration from the moment the index segment is encountered to the moment when the required data item is downloaded. The Access Latency is the sum of the Probe Wait and Bcast Wait, it is important to note that these two factors work against each other [4].

In this paper, the data broadcast model is focused on, because of its strength in terms of scalability. An objective of our work is to minimize two parameters: the tuning time and the latency they count the number of nodes broadcast by the server during the execution of the query.

2. RELATED WORK

Spatial databases have been studied extensively during
the last two decades, from this research, several spatial access methods have been proposed. In this section, we provide some background on the continuous spatial query processing techniques and air index schemes, which we adapt in this study.

In [5], authors present a supporting Continuous Nearest Neighbor (CNN) search in wireless data broadcast services. In this paper, the authors address the search algorithms for CNN in wireless data broadcast, which require revision in order to fit the linear streaming property. Then, two air indexing techniques are proposed, namely R-tree air index and Hilbert Curve air index, for processing CNN search on air. The Hilbert Curve air index achieves superior performance on uniformly distributed data, while the R-tree air index provides excellent performance on a skewed data distribution. In [6], authors present a Query-Index approach for in-memory evaluation of continuous range queries on moving objects. In this paper, no constraints are imposed on the speed or path of moving objects. In [2, 7], authors present the continuous query processor of the Pervasive Location-Aware Computing Environments (PLACE) server. To efficiently handle large number of continuous queries, authors employ an incremental evaluation paradigm.

Index method allows mobile clients requesting data to tune into a continuous broadcast channel only when spatial data of interest and relevance is available on the channel, thus minimizing power consumption. (1, m) index [4] is one of the most famous air index techniques. In this method, the index is broadcast m times during a single broadcast cycle. The broadcast index is broadcast every fraction (1, m) of the broadcast cycle. In [8], authors propose a novel adaptive technique for evaluating wireless data access methods. They present an analytical model for each index scheme and conduct exhaustive simulations to gauge the behavior of each indexing technique.

3. CRQ PROCESSING ON THE AIR

The wireless broadcast environment has the linear property, since the data items are sequentially delivered on the broadcast channel. In this paper, it is assumed that the trajectory of the query movement is known a priori. Then, by using computational geometry for stationary objects, the objects that will be within range of the query trajectory can be identified.

3.1 Data Organizing for CRQ Processing

Indexing for CRQ have been well studied, however, to the best of our knowledge, without an index of broadcast systems introduces new challenges. In a recent paper [10], the concept of data sorting for broadcasting called the Broadcast-based Location Dependent Data Delivery Scheme (BBS), has been proposed. In the BBS method, the server periodically broadcasts the IDs and coordinates data objects, without an index segment, to the clients, these broadcasted data objects are sorted sequentially, according to the location of the data objects, before being transmitted to the clients. In this method, since the data objects broadcasted by the server are sequentially ordered, based on their location, it is not necessary for the client to wait for the index segment, if the desired data object is able to be identified before the associated index segment arrives. In this method, the structure of the broadcast affects the distribution of the data object. The BBS is simple and provides the fastest access time, since no index is broadcasted along with the data and thus, the size of the entire broadcast cycle is minimized.

Figure 1: Broadcast sequence of (a) HB and (b) VB

A simple sequential broadcast can be generated by linearizing the two dimensional coordinates in two different ways: i.e. Horizontal Broadcasting (HB) or Vertical Broadcasting (VB). In HB, the server broadcasts the Location Dependent Data (LDD) in horizontal order, that is, from the leftmost coordinate to the rightmost coordinate. Conversely, in VB, the server broadcasts the LDD in vertical order, that is, from the bottom coordinate to the top coordinate (e.g., Figure 1).

3.2 CRQ Algorithms with BBS

With the BBS scheme [10], the client can significantly reduce the average access time, since it eliminates the probe wait time for the clients as pointed out in Section 3.1. However, the tuning time may increase, since the client is required to tune the broadcast channel until the desire data item arrives (See Fig. 3(b)). In the previous index schemes [4], in order to make all of the data items self-identifying, each data item contains the following header information:

- bucket id: the offset of the data item from the beginning of the bcast
- pointer index id: the offset to the beginning of the next bcast index pointer
- data type id: the offset to the beginning of the next index segment data type

In our scheme, all data items contain pointers that contain IDs, locations and arrival times of the data items to be broadcast afterward. In this section, the CRQ technique is presented under the BBS environment namely, SEquential Broadcast for Continuous Range Queries (SERIES). SERIES minimizes the access time and provides the clients with selective tuning capabilities.

The server broadcasts the data items without an index. Thus, in order to allow selective tuning, each data item has the following information:

- Object ID and its location
- Bcast_Pointer(BP): the offset to the beginning of the next bcast
- Next_Pointer(NP): IDs locations and arrival times of the data items that will be broadcast at Ti. The maximal number of NP from the each data item is k − 1,
where $k=\lfloor \log_e N \rfloor$, $c$ be the exponent value and $N$ be the number of data items that will be broadcasted (e.g., if $c = 2$ and $N = 32$, the first broadcasted data item $O_1$ has the following NPs: the data items located in $T_2$, $T_3$, $T_4$, $T_5$ and $T_6$ (see Figure 2))

\begin{align*}
\end{align*}

\begin{itemize}
  \item $T_N$: nearest boundary line on the left-side of the $s$, e.g., $T_4'$ in Figure 2. If x-coordinate of $T_i + 1 \geq x$-coordinate of $s$, then $T_i = T_N$.
  \item $T_{Ns}$: candidate of nearest boundary line on the left-side of the $s$, e.g., $T_5$ and $T_4'$ in Figure 2.
  \item $S_{range}$: set of data objects within the search range
  \item $S_{range}^c$: set of data objects for search
\end{itemize}

Observation 3.2.1: While the server broadcasts the data objects using HB, if $x$-coordinate of $O_i < T_N$, then $O_i$ is out of $S_{range}$.

Observation 3.2.2: While the server broadcasts the data objects using HB, if $x$-coordinate of $O_i > x$-coordinate of $T_n$, then $O_i$ and the rest of the broadcast data objects are located outside of the $S_{range}$, where $x$-coordinate of $T_n > x$-coordinate of $e$.

\begin{align*}
\end{align*}

Theorem 3.2.1 (Set of data for search): The result of Observation 3.2.1 and Observation 3.2.2 lead to the conclusion that if $x$-coordinate of $T_N \leq x$-coordinate of $O_i \leq x$-coordinate of $T_n$, where $x$-coordinate of $T_n > x$-coordinate of $e$, then $O_i \in S_{range}^c$.

3.3 Reducing Search Cost

The search cost and the tuning time can both be improved by adding specific information, such as the UL (Uppermost Line) and LL (Lowermost Line) to NP. This method is based on the observation that there is no requirement to search the entire data items between $s$ and $e$. Let $UL$ denote the uppermost line between $T_n$ and $T_{n+1}$ and $LL$ denote the lowermost line between $T_n$ and $T_{n+1}$ as presented in Figure 3. Let $R_n$ be a rectangle which has the following edges: lines of $T_n$ and $T_{n+1}$ and $UL$ and $LL$, which are lines between $T_n$ and $T_{n+1}$ (e.g., $R_4$ of Figure 3 has the following edges: $T_4$, $T_5$, $UL(O_{10})$ and $LL(O_{12})$). Suppose that there are 19 data objects and a client desires to find all data objects located within the search range between $s$ and $e$ (see Figure 3). Let $R_{s,e}$ denote the search range between $s$ and $e$ and $R_{T_n-T_{n+1}}$ denote the search range between $T_n$ and $T_{n+1}$. As shown in the figure, $R_{s,e}$ passes through $R_2$ and $R_3$, while $R_{s,e}$ does not pass through $R_1$ and $R_4$.

\begin{itemize}
  \item $T_i$: boundary lines of the $O_i$, e.g., $T_1$, $T_2$, $T_3$, $T_4$, $T_5$, $T_6$ in Figure 2.
  \item $T_i$: set of $T_i$
\end{itemize}

Notations:

- $S$: Server data set
- $O_i$: broadcast data object, where $O_i \in S$
- $O_i$: the client’s first tuned data item in the broadcast channel
- Data first: the server’s first broadcast data item in the current broadcast cycle
- $T_i$: boundary lines of the $O_i$, e.g., $T_1$, $T_2$, $T_3$, $T_4$, $T_5$, $T_6$ in Figure 2
- $T_i$: set of $T_i$

Figure 2: Data broadcast of the SERIES method

Let us assume that $(x$-coordinate of $s) \geq (x$-coordinate of $e)$. The client obtains the ID, location information and NP of the first tuned data item. Then, it switches to doze mode until the desired data item appears on the broadcast channel. The client repeatedly switches between the doze and wake up modes until the required data items are obtained within the range. Let us consider the example in Figure 2.

As presented in the figure, the result objects of the query are $O_{25}$, $O_{27}$, $O_{28}$ and $O_{30}$. First, the client tunes into the broadcast channel at $T_1$ and obtains following pointers $NP=\{2, 4, 8, 16, 32\}$ from data item $O_1$. Then, it switches to sleep mode until $O_{16}$ (at $T_5$) appears on the broadcast channel, since $T_5$ is the nearest boundary line on the left-hand side of the query point $q$ up to the present time. Then, the client wakes up at $T_5$, and obtains $NP=\{17, 19, 23, 31\}$ from the data item $O_{16}$. The client again switches to sleep mode until $O_{23}$ (at $T_4'$) appears on the broadcast channel, since $T_4'$ is the nearest boundary line on the left-hand side of the query point $q$ up to the present time. Finally, the client wakes up at $T_4'$, follows algorithm 2 and returns the data items $O_{25}$, $O_{27}$, $O_{28}$ and $O_{30}$ as the result.

Figure 3: Motivation of reducing the clients’ search cost and tuning time with SERIES method

Observation 3.3.1: If $R_{T_{n-1}-T_n}$ belongs or intersects with $R_n$, where $R_n$ located between $T_n$ and $T_{n+1}$, then
Lemma 3.3.1: If \( R_{Tn-Tn+n} \) does not belongs or intersects with \( R_n \), where \( R_n \) located between \( Tn \) and \( Tn+n \), then the client is not required to tune the broadcast channel between \( Tn \) and \( Tn+n \).

**Proof:** Let \( R_{Tn-Tn+n} = R_{T4-T5} \) (see Figure 4(b)). As shown in the figure, the client desires to find the data items within the search range beginning from \( q_s \) to \( q_e \). Since the search range between \( q_s \) and \( q_e \) does not pass through \( R4 \), the client is not required to tune the broadcast channel from \( T4 \) to \( T5 \). Then, the result for the query of \( R_{T4-T5} = \emptyset \).

Theorem 3.3.1: The result of Observation 3.3.1 and Lemma 3.3.1 leads to the conclusion that if \( R_{Tn-Tn+n} \) are located outside of \( R_n \), then data objects between \( Tn \) and \( Tn+n \) are out of \( S_{range} \) (set of data objects within the search range), where \( R_n \) is located between \( Tn \) and \( Tn+1 \).

![Figure 4: Selective tuning for continuous range query in wireless data broadcast](image)

Algorithm 2. Client algorithm

1: do
2:  read \( O_i \)
3:  if (x-coordinate of \( O_j \) > x-coordinate of \( s \) AND \( O_j \neq Data \) first) then switch to sleep mode until Data first comes
4:  \( O_i = Data \) first
5:  else
6:  read \( NP \) from \( O_i \)
7:  check x-coordinate of \( T_i \) and x-coordinate of \( s \)
8:  if \( T_i-TN \) // x-coordinate of \( T_i+1 \) ≥ x-coordinate of \( s \) then check UL and LL and follow the Observation 3.3.1 and Lemma 3.3.1
9:  selective tuning until satisfy the Observation 3.2.2
10:  else
11:  find \( TNc \), on the left hand side of \( s \) (e.g., \( T5 \) in Figure 2)
12:  then switch to sleep mode until the data object of \( TNc \) appears on the channel
13:  \( O_i = \) object of \( TNc \)
14:  } while(find out final result of CRQ)

4. PERFORMANCE ANALYSIS

In this section, the performance of the proposed SERIES scheme is compared with that of the famous \((1, m)\) index scheme. First the Access Latency between the SERIES and \((1, m)\) index schemes is compared. Then, the Tuning Time between these two schemes is compared. It is assumed that during each broadcast cycle, the server broadcasts the same data items in the same order, and that these data items contain the IDs and coordinates of the data items and the index segments. Let \( k' \) be the index search cost for the single data item and \( S_r \) be the set of data items of the search range. The following presents a comparison of the Probe Wait and the Bcast Wait between SERIES and previous index method. Let \( m \) denote the number of times broadcast indices, \( \text{AAT}_{\text{SERIES}} \) represent the average access time for SERIES and \( \text{AAT}_{(1, m)} \) represent the average access time for the previous index technique:

The average duration for getting to the next index segment. If it is assumed that the distance between two consecutive index segments is \( L \), the probe wait is \( L/2 \). In addition, as the number of \( m \) increases, Probe Wait time decreases. In the SERIES method, since the data objects broadcast by the server are sequentially ordered based on their location, it is not necessary for the client to wait for an index segment. Therefore, Probe Wait of SERIES=0.

**Bcast Wait** is the average duration from the moment the index segment is encountered to the moment when the required data item is downloaded. Thus:

\[
\text{SERIES} = \frac{(N - S_r)}{2} + S_r
\]

Since the AAT is the sum of the Probe Wait and the Bcast Wait, AAT for:

\[
\text{AAT}_{(1, m)} = 1 + (k' \times S_r) + S_r
\]

Now, tuning time for the proposed schemes with \((1, m)\) index is evaluated. The probability distribution of the initial probe of clients is assumed to be uniform within a broadcast and data items of the same size. Let \( ATT \) be the average tuning time, \( e \) be the exponent value. The minimum number of steps is 1 and the maximum number of steps is \( k-1 \), where \( k = \lceil \log_e N \rceil \). For example, if \( N=1024 \) and \( e=2 \), then in the best case, the client obtains the desired data item within a single step while, in the worst case, the client obtains the desired data item within 9 steps. The frequency of the worst case for \( N = 1 \), while the frequency of the best case for \( N = k \).

\[
\text{ATT}_{(1, m)} = 1 + (k' \times S_r) + S_r
\]

\[
\frac{k \times 2^{k-1} \times \sum_{i=0}^{\lceil \log_e N \rceil - 1} i + S_r}{N} \approx S_r
\]

\[
\frac{k \times e(e-1)}{4} + S_r
\]
Table 1: Access latency as the moving distance increases

<table>
<thead>
<tr>
<th>Distance (km)</th>
<th>10</th>
<th>20</th>
<th>30</th>
<th>40</th>
<th>50</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hilbert (ms)</td>
<td>77.3</td>
<td>181.2</td>
<td>249.6</td>
<td>321.7</td>
<td>415.6</td>
</tr>
<tr>
<td>R-tree (ms)</td>
<td>92.7</td>
<td>217.4</td>
<td>299.5</td>
<td>386.0</td>
<td>498.8</td>
</tr>
<tr>
<td>SERIES (ms)</td>
<td>27.8</td>
<td>66.0</td>
<td>97.6</td>
<td>138.2</td>
<td>209.1</td>
</tr>
</tbody>
</table>

Table 2: Energy consumption as the moving distance increases

<table>
<thead>
<tr>
<th>Distance (km)</th>
<th>10</th>
<th>20</th>
<th>30</th>
<th>40</th>
<th>50</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hilbert (mW)</td>
<td>0.23</td>
<td>0.69</td>
<td>1.16</td>
<td>1.62</td>
<td>2.09</td>
</tr>
<tr>
<td>BBS (mW)</td>
<td>0.56</td>
<td>1.70</td>
<td>2.84</td>
<td>3.98</td>
<td>5.11</td>
</tr>
<tr>
<td>SERIES (mW)</td>
<td>0.19</td>
<td>0.59</td>
<td>0.98</td>
<td>1.37</td>
<td>1.77</td>
</tr>
</tbody>
</table>

5. NUMERICAL RESULTS

This section evaluates the performance of the proposed SERIES scheme by comparing it to that of traditional and state-of-the-art spatial indexing methods for wireless data broadcast.

First, the effect of the moving distance (i.e., from s to c) on the access time is studied. The moving distance is increased from 5(km) to 50(km). As shown in the table 1, Proposed scheme outperforms other schemes. This is due to the previous index technique of broadcasting data without considering the properties of the locality of the data, resulting in increased client search cost and access time.

Second, the effect of the moving distance on the energy consumption is studied. Table 2 presents the energy consumption, as the moving distance increases from 5 to 50. As shown in the table, the SERIES outperforms the BBS and 1,m Hilbert, since (1,m) method minimizes the tuning time but not the access time, while the proposed schemes reduce the TT and AT at the same time. In other words, when the energy consumption is estimated, it is necessary to consider not only the active time, but also the doze time, even if the doze time is considerably smaller than the active time, the client also consumes battery power when it stays in power save mode.

As shown in this table, proposed scheme demonstrates best performance as the moving distance increases, since the (1,m) with R-tree and Hilbert significantly increases access latency as the moving distance increases.

Figure 5: Client’s energy consumption as the size of data item increase

Figure 5 present the energy consumption as the size of data items increases from 128bytes to 8192bytes. As shown in the figures, SERIES demonstrates superior performance as the size of data increases, since the (1,m) with Hilbert significantly increases access latency, as the size of the data items increases. That is, with the proposed scheme, the clients can significantly reduce their access time, since this scheme eliminates the probe wait time for the clients. Moreover, the proposed scheme offers the ability for clients to selectively tune into relevant data without an index segment. Therefore, the proposed scheme minimizes not only clients’ access time but also energy consumption.

6. CONCLUSION

In this paper, a pioneering study on Continuous Range Queries(CRQ) in wireless data broadcast environments was conducted. Two techniques are presented, broadcast-based sequential data dissemination and selective tuning, and algorithms are developed based on these two techniques, to process CRQ on the air. The task of the query processor is to selectively monitor the wireless broadcast channel, as the data items are disseminated according to their location on the server. Theoretical analysis of SERIES in terms of the execution costs is provided. Comprehensive experiments illustrate that SERIES is highly scalable and is more efficient than previous techniques in terms of both access latency and energy consumption.

7. REFERENCES