An Access Time Cost Model for Spatial Range Queries On Broadcast Geographical Data Over Air

Jianting Zhang Le Gruenwald

School of Computer Science, the University of Oklahoma
200 Felgar Street
Norman, ok, 73072, USA
{jianting, ggruenwald}@ou.edu

Abstract. Wireless data broadcasting is well known for its excellent scalability. Most geographical data, such as weather and traffic, is public information and has a large number of potential users. Broadcast is a good mechanism that can be used to transmit the data to users at this scale. In this paper, we propose a cost model for access time in processing spatial range queries on broadcast geographical data over air. We also propose heuristics in generating orderings of broadcast sequences and evaluate their performances based on the cost model

1 Introduction

Data broadcasting is well known for its excellent scalability (*Imielinski, 1997*). Most geographical data, such as weather and traffic, is public information and has a large amount of potential users. Thus it is attractive to broadcast geographical data in metropolitan areas to reduce the increasing demands for wireless bandwidth resources. Furthermore, for users that are able to be aware of their locations by using Global Position System (GPS), network infrastructures or their combinations (Konig-Ries, 2002), they can perform Location Dependent Queries (LDQ, Seydim, 2001) to request Location Dependent Services (LDS). It is easy to see that LDQ on broadcast geographical data over air is particular interesting in the context of large-scale resource-efficient data dissemination in mobile computing. Spatial range query (Rigaux, 2002) processing on broadcast geographical data will be one of the most popular techniques to provide LDS.

The performance of a data broadcast system is characterized by two parameters (*Imielinski, 1997*), tune-in time (TT) and access time (AT). TT is defined as the time for a client to download data from a broadcast sequence. During this time the client has to be in active mode and consumes more energy than in doze/sleep mode. AT is defined as the time a client begin to access the broadcast sequence to the time all the requested data items are downloaded. A client may switch to doze mode in between two active downloading where usually less energy is consumed. In Fig. 1, TT is equal to the total of the length of required data items (shaded) while AT is the duration between the first and the last required data items.

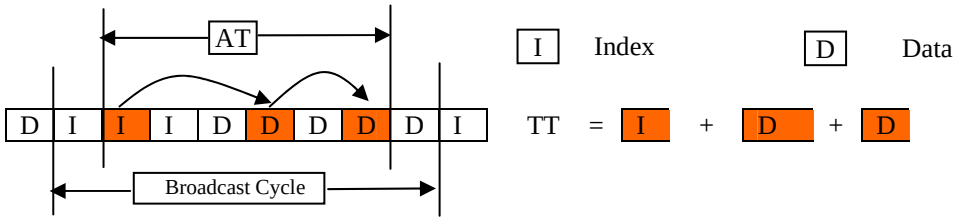

Fig. 1 Illustration of TT and AT

In this paper, we aim at providing a cost model of access time for spatial range queries on broadcast geographical data. We assume a client has already had an ordered set of pointers to data items in the broadcast channel by performing a spatial range query on index segments which are either in the same channel with the data or in a separate index channel. Currently we focus on the data access time only and leave the index access for future work. The scenario we consider is that index and data are broadcast using separate channels where a client may begin to access the data channel randomly (Fig.2).

The rest of this paper is arranged as follows. Section 2 is an overview of related work. We propose our cost model for range queries over broadcast geographical data in Section 3. We present several ordering heuristics and evaluate their performances based on the cost model using a real data set in Section 4. Finally Section 5 concludes the paper with summary and future work directions

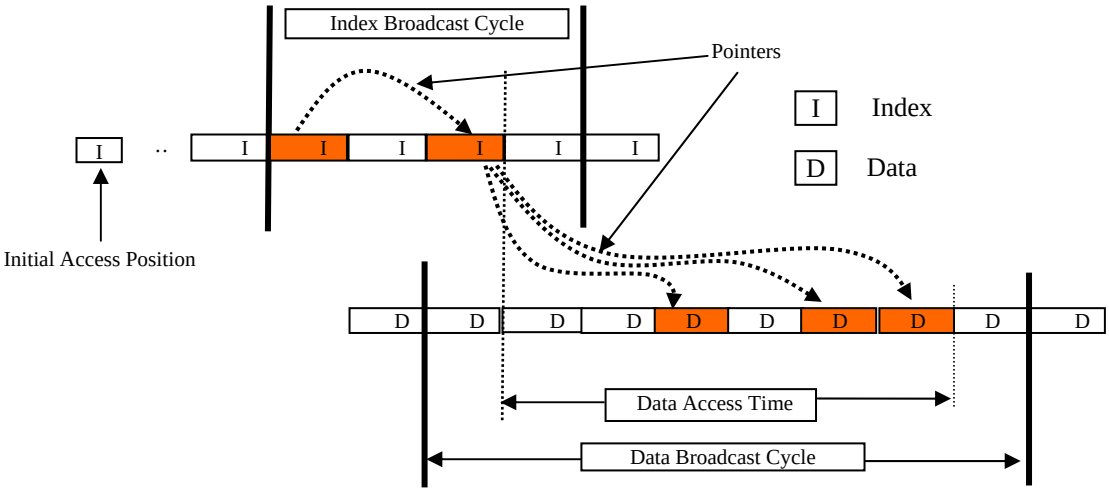

Fig. 2 Index and Data Use Separate Channels

2 Related Work

Range queries are the most frequently used spatial queries and have been extensively studied in disk-resident data management research. Several cost models have been proposed for measuring the performance of spatial indexing on range queries (Pagel, 1993; Theodoridis, 1996; Theodoridis, 2000). The measurement used is the number of disk accesses which is equivalent to tune-in time in broadcasting without considering paging and buffering effects. However, to the best of our knowledge, there is no previous work done on access time for spatial range queries on broadcast data.

There have been several studies on general data broadcast. Many of them focus on indexing techniques to make tradeoffs between TT and AT, such as tree-indexing (Imielinski, 1994a), hashing (Imielinski, 1994b), signature (Lee, 1996) and hybrid (Hu, 2001a). They can support only queries on one-dimensional data and can search only one data item in a query result. Although (Imielinski, 1997) proposed to chain data items that have the same values in different meta-segments in its nonclustering index and multi-index methods, it cannot be applied to data items that have different values but are often in the same query results. Furthermore, in its performance analysis, it assumes that it takes a whole broadcast cycle to retrieve non-clustered data items of a particular value. That is an unnecessary overestimation. The issue of multi-attribute data broadcast and query was first addressed in (Hu, 2001b). However, this work can handle only conjunction/disjunction queries that involve fewer than three attributes. They are not suitable for range queries on geographical data.

Recent works on object-oriented database broadcast (Chehadeh, 1999) and relational database broadcast (Lee, 2002) allow multiple data items to be accessed in a query. However, they assumed the access to data items had predefined orders. They are not suitable for spatial range queries since data items in a query result do not necessarily have a predefined order. The work presented in (Chung, 2001) is essentially similar to our cost model of data access time. However, it excluded the tune in time from access time for the items in the query result set which makes the total access time a summation of multiple quadratic terms. To simplify the result, it used a linear function to approximate the quadratic cost, which makes the model inaccurate. Furthermore, its proof of the approximation is incorrect. We believe that our result in which the access time for a single query is linear with respect to a single quadratic term (see Section 3.2 for details) is more concise and accurate. None of the above cost models are designed for spatial range queries.

The only previous work on geographical data broadcast we know is (Hambrusch, 2001). It studied the execution of spatial queries on broadcast tree-based spatial index structures. Their work assumed the client had very limited memory that the whole R-tree cannot be fit into the client memory and the client has to discard some retrieved R-Tree nodes to hold more useful ones during the query process. Their work focused on reducing extra access time incurred by having to access multiple broadcast cycles due to the discard and replacement. Our cost models assume that a client has already have the pointers to data items in the data channel, either from another separate index channel or from the same channel that combines both data and index. A client can sort the values of the pointers and thus only one scan of data channel is sufficient to retrieve all the data items. We believe that our assumption that a client can hold the entire index segments related to a spatial range query is more realistic for LDQs.

3 The Cost Model

3.1 Preliminaries

Let DS=[x1,x2)× [y1,y2) be the data space that defines all the geographical data items. Suppose the size of a range query window is (q_x,q_y).

We define an Extended Region R_u of data item P_u as the rectangle of (q_x,q_y) centered at P_u. As shown in Fig. 3, the distribution of the centers of query window (q_x,q_y) that contains data item P_u is the extended region of R_u. Furthermore, from Fig. 4 we can see that the distribution of the centers of query window (q_x,q_y) that contains both data items P_u and P_v is the intersection of their extended regions R_u and R_v. This relationship can be extended to higher orders, i.e., up to the intersected region among all *n* extended regions where *n* is the number of points in the data set to broadcast.

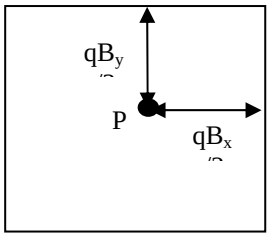

Fig. 3 The Distribution of Centers of Query Window (qB_{xB},qB_{yB}) that Contains PB_{uB}

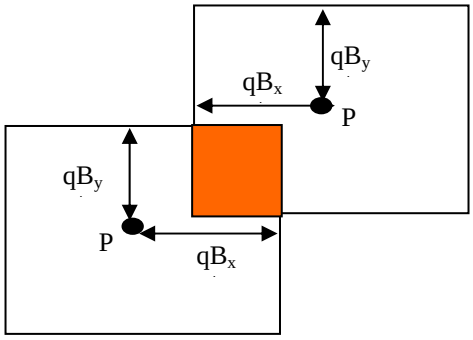

Fig. 4 The Distribution of Centers of Query Window (qB_{xB},qB_{yB}) that Contains both $PB_{u\,B}$ and PB_v ₌(ShadedB₌Area)

Before presenting our cost model, we define the following symbols. Let A_i be the area of R_i, $A_{i,j}$ be the intersection area of R_i and R_j, … $A_{1,2\ldots n}$ be the intersection area of R_1, R_2…R_n. Let $\tilde{A}_i$ be the part of A_i that solely contains one point for all *i* where 0≤i<n, $\tilde{A}_{i,j}$ be the part of $A_{i,j}$ that solely contains two points for all *i* and *j* where 0≤i<j<n, … $\tilde{A}_{1,2\ldots n}$ be the part of the intersection area of R_1, R_2…R_n that contains solely *n* points. Note $\tilde{A}_{1,2\ldots n} = A_{1,2\ldots n}$

Under the assumption that all the locations inside the study region are equally likely to be the locations where users request spatial range queries, the number of expected requests from a region is the multiplication of the area of the region and a constant c, the number of requests per unit area. For the sake of simplicity we omit the constant factor and only use $\tilde{A}_i$, $\tilde{A}_{i,j}$ … $\tilde{A}_{1,2\ldots n}$ as the access frequencies for the corresponding query result sets in the following sections.

3.2 Access Time for Processing A Single Query

Let function π(u) maps point *u* to its position in the broadcast sequence. Suppose the single query result set contains *k* data items n_1, n_2 … n_k. Assume the data broadcast cycle length is L. Let L_2 denote the access time of a query result set with a query window size of (q_x, q_y). Let L_1 and L_3 denote the time before L_2 and after L_2 (Fig. 5). It is easy to see that $L=L_1+L_2+L_3$, $L_1 = \min\{\pi(n_1), \pi(n_2), \ldots \pi(n_k)\}$, and $L_2 = \max\{\pi(n_1), \pi(n_2), \ldots \pi(n_k)\} - \min\{\pi(n_1), \pi(n_2), \ldots \pi(n_k)\}$.

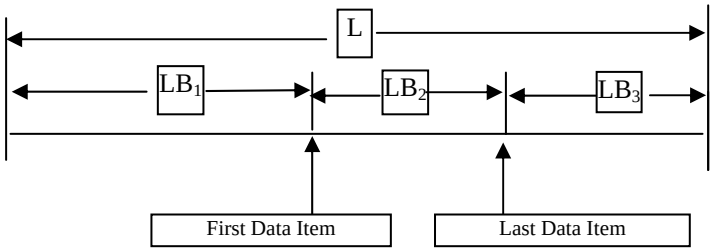

Fig. 5 Illustration of L_1, L_2 and L_3

Since a client might begin to access data channel at any position (between 0 and L), we need to consider the following three cases separately. We first compute the total access time in these three cases and then compute the average.

Case1: Begin access in L_1, the total access time is the sum of the rest of L_1 and the whole L_2: $\sum_{i=0}^{L_1-1}(L_1 - i + L_2) = \frac{L_1(L_1+1)}{2} + L_1 * L_2$

Case 2: Begin access in L_2, the total access time is equivalent to the whole broadcast cycle regardless of the access position:

$$\sum_{i=0}^{L_2-1} L = L * L_2$$

Case 3: Begin access in L_3, the total access time is equal to the rest of L_3 in the current broadcast cycle plus L_1+L_2 in the next broadcast cycle:

$$\sum_{i=0}^{L_3-1}(L_3 - i + L_1 + L_2) = \sum_{i=0}^{L_3-1}(L - i) = L_3 * L - \frac{L_3(L_3-1)}{2}$$

The average access time is :

$$\begin{aligned}
&\frac{1}{L}[\frac{L_1(L_1+1)}{2} + L_1 * L_2 + L * L_2 + L_3 * L - \frac{L_3(L_3-1)}{2}] \\
&= \frac{1}{L}[\frac{(L_1+L_3)(L_1-L_3+1)}{2} + L_1 * L_2 + L * (L_2 + L_3)] \\
&= \frac{1}{L}[\frac{(L-L_2)(L_1-L_3+1)}{2} + L_1 * L_2 + L * (L - L_1)] \\
&= \frac{1}{L}[L^2 - L_1 * (L - L_2) + \frac{(L-L_2)(L_1-L_3+1)}{2}] \\
&= \frac{1}{L}[L^2 - \frac{(L-L_2)(2 * L_1 - L_1 + L_3 - 1)}{2}] \\
&= \frac{1}{L}[L^2 - \frac{(L-L_2)(L_1+L_3-1)}{2}] \\
&= \frac{1}{L}[L^2 - \frac{(L-L_2)(L-L_2-1)}{2}]
\end{aligned}$$

From the result we can see that the average access time to the data channel is determined only by L and L_2. We can rewrite the average data access time as follows:

$$\frac{1}{L}[L^2 - \frac{(L - L_2)^2 - (L - L_2)}{2}]$$
$$= \frac{1}{L}[L^2 - \frac{(L - L_2)^2 - 2 * (\frac{L - L_2}{2}) + (\frac{1}{2})^2 - \frac{1}{4}}{2}]$$
$$= \frac{1}{L}[L^2 + \frac{1}{8} - \frac{(L - L_2 - \frac{1}{2})^2}{2}]$$

Since L_2<L, the average access time of a range query decreases monotonically as L_2 decreases. Since the number of data items in a query is usually much smaller than the number of data items to broadcast, we assume L-L_2>>1. Thus the formula can be simplified as $\frac{L}{2} + L_2 - \frac{L_2^2}{2L}$.

3.3 Access Time for Processing All Queries

Let function g(L_2) be $g(L_2) = \frac{L}{2} + L_2 - \frac{L_2^2}{2L}$. The total data access time for a query window (q_x,q_y) can be written as follows by summarizing the access time over all possible query result sets. Substituting L_2 back with $\max\{\pi(n_1), \pi(n_2), \ldots \pi(n_k)\} - \min\{\pi(n_1), \pi(n_2), \ldots \pi(n_k)\}$, we have

$$Cost^{(q_x,q_y)}$$
$$= \sum_{1 \le i < j \le n} \tilde{A}_{ij}^{(q_x,q_y)} * g(|\pi(i) - \pi(j)|)$$
$$+ \sum_{1 \le i < j \le k \le n} \tilde{A}_{i,j,k}^{(q_x,q_y)} * g(\max(\pi(i),\pi(j),\pi(k)) - \min(\pi(i),\pi(j),\pi(k)))$$
$$+ \ldots$$
$$+ \tilde{A}_{1,2,\ldots n}^{(q_x,q_y)} * g(\max(\pi(1),\pi(2)\ldots\pi(n)) - \min(\pi(1),\pi(2)\ldots\pi(n)))$$

The final total access time will be the summation of $Cost^{(q_x,q_y)}$ over all possible query windows Q, i.e.,

$$Cost = \sum_{(qx,qy) \in Q} Cost^{(q_x,q_y)}$$
$$= \sum_{(qx,qy) \in Q} [\sum_{1 \le i < j \le n} \tilde{A}_{ij}^{(q_x,q_y)} * g(|\pi(i) - \pi(j)|)]$$
$$+ \sum_{(qx,qy) \in Q} [\sum_{1 \le i < j \le k \le n} \tilde{A}_{i,j,k}^{(q_x,q_y)} * g(\max(\pi(i),\pi(j),\pi(k)) - \min(\pi(i),\pi(j),\pi(k)))]$$
$$+ \ldots$$
$$+ \sum_{(qx,qy) \in Q} \tilde{A}_{1,2,\ldots n}^{(q_x,q_y)} * g(\max(\pi(1),\pi(2)\ldots\pi(n)) - \min(\pi(1),\pi(2)\ldots\pi(n)))$$

Let

$$w_{i,j} = \sum_{(qx,qy)\in Q} \tilde{A}_{i,j}^{(q_x,q_y)}$$

$$w_{i,j,k} = \sum_{(qx,qy)\in Q} \tilde{A}_{i,j,k}^{(q_x,q_y)}$$

...

$$w_{1,2,...n} = \sum_{(qx,qy)\in Q} \tilde{A}_{1,2,...n}^{(q_x,q_y)}$$

Then

$$Cost =$$

$$\sum_{1\le i<j\le n} w_{i,j} * g(|\pi(i)-\pi(j)|)$$

$$+ \sum_{1\le i<j\le k\le n} w_{i,j,k} * g(\max(\pi(i),\pi(j),\pi(k)) - \min(\pi(i),\pi(j),\pi(k)))$$

$$+ ...$$

$$+ w_{1,2,...n} * g(\max(\pi(1),\pi(2)...\pi(n)) - \min(\pi(1),\pi(2)...\pi(n)))$$

It is easy to observe that the cost model we have developed is similar to the Minimum Linear Arrangement (MinLA) problem in graph theory defined as follows (Daíz, 2002):

$$la(G) = \sum_{(u,v)\in E} w(u,v) * |\pi(u) - \pi(v)|$$

There are two differences between MinLA and our cost model. First, there are multiple data items in a query result set and a hyper-graph representation is more appropriate for our cost model than a graph representation for MinLA. Second, our cost model is quadratic with respect to the differences in the positions of the beginning and ending nodes in a hyper-edge while it is linear in MinLA. It might be interesting to extend the existing low computation cost approximation methods for MinLA (Bar-Yehuda, 2001; Koren, 2002) to optimise access time based on our cost model which we leave for our future work.

4 Experiments and Results

4.1 Ordering Heuristics

The order of the geographical data items in a broadcast channel determines the total access time of spatial range queries on such data items. Space Filling Curves (SFC, Gade, 1998), such as row-wise enumeration of the cells, Peano curve or Z-

Ordering, Hilbert-Ordering and Gray-Ordering, which transforms multi-dimensional data into one-dimension can be used to generate orderings by comparing the SFC codes. Although spatial index trees such as R-Tree family (Guttman, 1984; Sellis, 1987; Beckmann, 1990) are not originally designed to be aware of the order of data items, traversals of these trees do generate orderings that can be used to sequence the data items. Since spatial indexing methods usually maintain spatial adjacencies, the orderings generated by SFCs and spatial index tree traversals are good candidates with low computation costs. We will evaluate the performances of the two heuristics based on our cost model using a real data set.

4.2 Experiments Setup

We use a data set from the MapInfo census 2000 data samples ([HREF 1]). There are 586 points in the area representing service locations, such as hospitals and parks. The data set is shown in Fig. 6. We choose four query window sizes with $q_x=q_y$ (We thus use q_x to denote the query window size hereafter). The sizes of the query windows are 0.5, 1.0, 2.5 and 5.0 miles respectively. We believe they are meaningful in practice. The C program from ([HREF 2]) was used to generate the Hilbert codes for all the points. The codes are then sorted to generate the Hilbert ordering. We also obtain the code from ([HREF 3]) to generate the R-trees and their traversal orderings. To investigate the effect of the branch factor in generating R-Tree traversal ordering, we vary its value factor from 4 to 19.

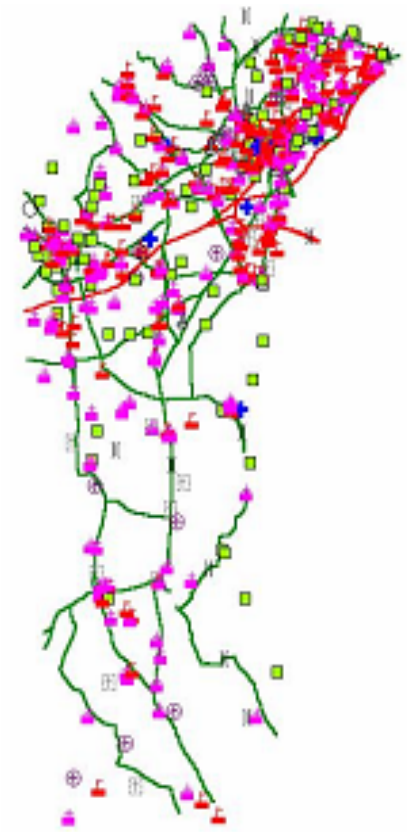

Fig. 6 The Data Set

4.3 Results and Discussion

The results of the total access times versus the branch factors for the four query window sizes are shown in Fig 7. Note that the absolute access time values presented in this section does not reflect the constant factor as discussed in Section 3.1.

From the results we can see that the total access time does not have a perceivable relationship with the R-Tree branch factor. An obvious pattern in the results is that the total access time has a strong relationship with the query window size. Interestingly the total access times reach their maximum for the query window size of 2.5 miles. We suspect that this pattern is rather data-dependent.

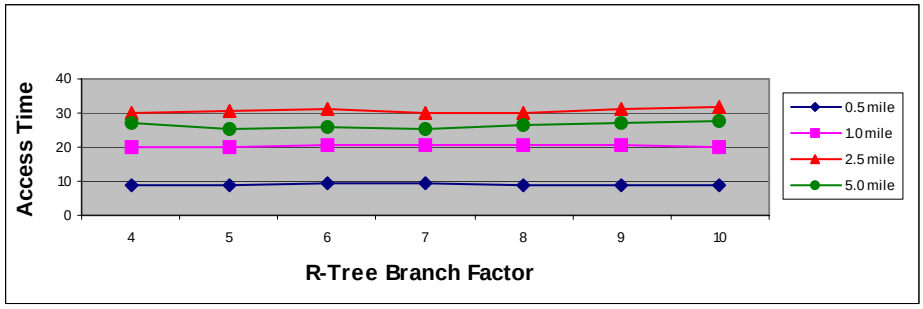

Fig. 7 Access Time vs. R-Tree Branch Factor

By comparing the result of Hilbert ordering and R-Tree traversal orderings we can see that Hilbert ordering is generally better than the best R-tree orderings. In Table 1 the access time of Hilbert ordering is compared with the minimum access times of R-tree orderings for all the four window sizes. The result suggests that the Hilbert ordering is about 9% better than the R-Tree traversal ordering on average.

Table 1. Comparison of Hilbert and Best R-Tree Traversal Orderings

Query Window Size (Miles)	Hilbert (A)	Best R-Tree (B)	(B-A)/A (%)
0.5	8.33124	8.92838	7.17%
1.0	18.03957	20.12991	11.59%
2.5	27.09348	30.08675	11.05%
5	23.51733	25.49852	8.42%

5 Conclusions and Future Work Directions

We believe we are the first to address the problem of spatial range queries over broadcast geographical data. We developed a precise and concise cost model for data access time in processing spatial range queries on broadcast geographical data. We also presented several heuristics in generating orderings and evaluate their performances based on the cost model

For future work, we first would like to extend our cost models to handle the access time both to the data channel and the index channel. Second, we want to optimise the access time by generating better orderings based on our cost model. Finally we would like to do more experiments using real data sets as well as synthetic data sets to evaluate the cost models, ordering heuristics and optimization methods.

References

N.Beckmann, H.-P. Kriegel, R.Schneider, B.Seeger, The R*-tree: An efficient and robust access method for points and rectangles. SIGMOD Conference, 1990:322-331

Y. C. Chehadeh, A. R. Hurson, Mohsen Kavehrad: Object Organization on a Single Broadcast Channel in the Mobile Computing Environment. Multimedia Tools and Applications 9(1): 69-94 (1999)

Josep Daíz and Jordi Petit and María Serna: A Survey on Graph Layout Problems. *ACM* Computing Surveys, 34(3): 313-356 (2002)
Yon Dohn Chung, Myoung-Ho Kim: Effective Data Placement for Wireless Broadcast. Distributed and Parallel Databases 9(2): 133-150 (2001)
V.Gaede, O.Günther: Multidimensional access methods. ACM Computing Survey, 30(2):170-231 (1998)
A.Guttman, R-trees: A dynamic index structure for spatial searching. SIGMOD Conference, 1984:47-54
S. Hambrusch, C.-M. Liu, W. Aref, S. Prabhakar: Query Processing in Broadcasted Spatial Index Trees. SSTD,2001: 502-521
Qinglong Hu, Wang-Chien Lee, Dik Lun Lee: A Hybrid Index Technique for Power Efficient Data Broadcast. Distributed and Parallel Databases, 9(2): 151-177 (2001)
Qinglong Hu, Wang-Chien Lee, Dik Lun Lee: Indexing Techniques for Power Management in Multi-Attribute Data Broadcast. MONET 6(2): 185-197 (2001)
T. Imielinski, S. Viswanathan, B. R. Badrinath: Energy Efficient Indexing On Air. SIGMOD Conference, 1994:25-36
T. Imielinski, S. Viswanathan, B. Badrinath: Power Efficient Filtering of Data on Air. EDBT, 1994: 245-258
T. Imielinski, S. Viswanathan, B. R. Badrinath, Data on Air: Organization and Access. IEEE Transactions on Knowledge and Data Engineering, 9(3): 353-372 (1997)
Birgitta König-Ries, etc.: Report on the NSF Workshop on Building an Infrastructure for Mobile and Wireless Systems. SIGMOD Record 31(2): 73-79 (2002)
Y. Koren, D. Harel: A Multi-Scale Algorithm for the Linear Arrangement Problem, Lecture Notes in Computer Science, Vol. 2573, Springer Verlag, 2002:293-306
Guanling Lee, Shou-Chih Lo, Arbee L. P. Chen: Data Allocation on Wireless Broadcast Channels for Efficient Query Processing. IEEE Transactions on Computers 51(10): 1237-1252 (2002)
Wang-Chien Lee, Dik Lun Lee, Using Signature Techniques for Information Filtering in Wireless and Mobile Environments. Distributed and Parallel Databases, 4(3): 205-227 (1996)
Bernd-Uwe Pagel, Hans-Werner Six, Heinrich Toben, Peter Widmayer: Towards an Analysis of Range Query Performance in Spatial Data Structures. PODS 1993: 214-221
Philippe Rigaux, Michel O. Scholl, Agnes Voisard, Spatial Databases: With Application to GIS, San Diego, CA: Academic Press 2002
T. Sellis, N. Roussopoulos and C. Faloutsos. The R+-Tree: A Dynamic Index for Multi-Dimensional Objects. VLDB Journal, 1987:507-518
Ayse Y. Seydim, Margaret H. Dunham, Vijay Kumar: Location dependent query processing. MobiDE, 2001: 47-53
Yannis Theodoridis, Timos K. Sellis: A Model for the Prediction of R-tree Performance. PODS 1996: 161-171
Yannis Theodoridis, Emmanuel Stefanakis, Timos K. Sellis: Efficient Cost Models for Spatial Queries Using R-Trees. TKDE 12(1): 19-32 (2000)
Reuven Bar-Yehuda, Computing an optimal orientation of a balanced decomposition tree for linear arrangement problems. Journal of Graph Algorithms and Applications, 5(4): 1-27 (2001)
[HREF 1] Http://www.mapinfo.com
[HREF 2] http://www.caam.rice.edu/~dougm/twiddle/Hilbert/
[HREF 3] http://www.cs.ucr.edu/~marioh/rtree/index.html