RESEARCH CHALLENGES IN INFORMATION ACCESS AND DISSEMINATION IN A MOBILE ENVIRONMENT

Jianliang Xu, Baihua Zheng, Manli Zhu, Dik Lun Lee

Department of Computer Science
Hong Kong University of Science and Technology
Clear Water Bay, Hong Kong
E-mail: {xujl,baihua,cszhuml,dlee}@cs.ust.hk

ABSTRACT

A wireless environment has both benefits and limitations when it is used as a medium for information access and dissemination. In a wireless environment, we can access to not only information anytime and anywhere as if we were connected to the Internet but information tailored to our particular needs depending on where we are located. This leads to location-dependent information services (e.g., querying for local traffic, restaurants, etc.). The capability of the system is also enhanced by the fact that genuine broadcast (as opposed to multicast) can be realized in a wireless environment, making massive broadcast of commonly requested data a simple and inexpensive task. On the other hand, we have to overcome a lot of limitations such as the limited bandwidth, high-cost of data transmission (both in terms of power consumption on the client devices and system bandwidth). In this paper, we give an overview of our research in this area. We then discuss our recent work on query processing in location-dependent information services and outline our future research.

1. INTRODUCTION

Mobile computing is the result of the convergence of high-speed wireless networks and personal mobile devices. In the near future, it is anticipated that a massive number of mobile users, carrying portable devices (e.g. palmtop, laptop, PDA, WAP phone etc.), will be able to access a variety of information from anywhere and at any time. The ability for mobile users to move and access information ubiquitously has opened up new classes of data applications, which promise to make our society more efficient and our lives more alive. For example, alert systems can notify the client anywhere once a registered alert fires (e.g., the stock price grows higher than certain pre-set threshold); mobile users can query location-dependent information (e.g., nearest restaurant) based on their current locations. Without doubt, mobile computing is becoming an important part of our daily life.

A wireless network with mobile clients is essentially a distributed system. However, it is more than just a distributed system. There are some unique properties of a mobile system that make it a fertile research area. These features are summarized as follows [2, 15]:

- **Limited wireless communication.** The radio spectrum used for wireless communication is inherently scarce. For example, GSM operates only between 880 MHz and 960 MHz. The bandwidth for a single wireless channel is limited, varying from 1.2 Kbps for slow paging channel, through 19.2 Kbps for CDPD to about 11 Mbps for wireless LAN. Further, wireless transmission is error-prone. Data might be corrupted or lost due to many factors such as signal interference etc.

- **Scarce power source.** The battery power is limited on mobile clients, which ranges from only a few hours to about half a day under continuous use. Moreover, only a modest improvement in battery capacity can be expected over the next few years [22]. It is also worthwhile noting that it consumes much more power sending than receiving data. For example, a Wavelan card consumes 1.7 W with the receiver "on" and 3.4 W with the transmitter "on".

- **Frequent disconnections.** To save energy or connection cost, mobile clients frequently disconnect themselves from the network and are kept in a weak connection status. Furthermore, due to unreliable wireless communication links, mobile clients are also disconnected often by failure.

- **Asymmetric communication.** Due to resource constraints of mobile clients, the *upstream* communication capacity from clients to servers is much less than the *downstream* communication capacity from servers to clients. Even in the case of an equal communication capacity, the data volume in the downstream direction is estimated to be much greater than that in the upstream direction [1].

- **Unrestricted mobility.** It does not require mobile clients to maintain a fixed position in the network and enables unrestricted mobility of the users. Locations and movements of mobile clients are difficult to predict.

- **Limited client capacities.** Since most portable wireless devices are restricted by weight, size, and ergonomic considerations, they are limited in their CPU, storage, and display capabilities.

In summary, mobile computing is creating new attractive applications. On the other hand, its limitations pose many challenging problems for mobile data applications. Efficient data management and resource management have to be developed before the vision of mobile computing can be fully utilized. In this paper, we give an overview of the research on mobile data access and dissemination, with an emphasis on our on-going project of location-dependent data management.

The rest of the paper is organized as follows. In Section 2, an overview of mobile computing environments is provided, including the mobile computing model and the data access model. Section 3 briefly discusses various data management issues in mobile environments. In Section 4, we describe our recent work on location-dependent data management. Finally, Section 5 concludes the paper with a discussion on future research.

2. SYSTEM MODELS

2.1. Mobile Computing Model

Figure 1 depicts the general mobile computing system model [3]. It consists of two distinct sets of entities: mobile clients (MC) and fixed hosts. Some of the fixed hosts, called mobile support stations (MSS), are augmented with wireless interfaces. An MSS can communicate with the MCs within its radio coverage area called a wireless cell. An MC can communicate with a fixed host/server via an MSS over a wireless channel. The wireless channel is logically separated into two sub-channels: an *uplink channel* and a *downlink channel*. The uplink channel is used by MCs to submit queries to the server via an MSS, while the downlink channel is used by MSSs to disseminate information or to forward the answers from the server to a target client. We assume that an MC contacts only one MSS at the same time. Each cell is associated with an ID (CID) for identification purposes. A CID is periodically broadcast to all the MCs residing in a corresponding cell. An MC can move from one cell to another (called *hand-off*). After hand-off, its wireless connection is switched to the new cell.

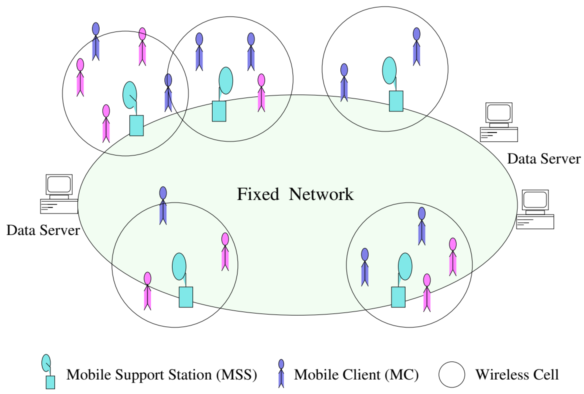


Figure 1: The Mobile Computing Model

2.2. Mobile Data Access Model

To enable mobile data access there are several alternatives, including data caching, on-demand access, and pushed broadcast [11].

- **Data Caching.** Data are cached at the mobile client. When the client issues a query, it first searches the cache. If there is a valid copy in the cache, an answer is returned immediately. Otherwise, the client attempts to obtain the data item from the server.

- **On-Demand Access**: The mobile client submits a request to the server via the uplink channel. The server returns the answer to the mobile client either individually (i.e., point-to-point access) or by multicast (i.e., on-demand broadcast).

- **Pushed Broadcast**: Data are periodically broadcast on the wireless channel. Once a query is issued, the mobile client tunes into the broadcast channel and filters out the desirable data.

The mobile data access model can be a combination of these three techniques. In a typical hierarchical data access model [11], the most frequently accessed data are cached at the client, the commonly requested data subset is broadcast on broadcast channels, and the rest of the data must be pulled from server via explicit client requests.

3. OVERVIEW OF DATA MANAGEMENT ISSUES

This section describes the mobile data management issues that we have been investigating. Other issues include data scheduling, location management, moving object management, and transaction management [2].

- **Client Cache Management.** Client data caching is a commonly used technique for improving access latency and data availability. In the framework of a mobile wireless environment, this is much more desirable due to constraints such as limited bandwidth and frequent disconnections. However, frequent client disconnection and movement between different cells make the design of cache management strategies a challenge. Thus, the issues of cache consistency, cache replacement, and cache prefetching have been investigated [1, 10, 23, 24].

- **Air Indexing.** Since battery power is a constrained resource on the mobile client, one critical issue is to save power consumptions on the client. This motivates the research of energy-efficient air indexing for data broadcast. The idea is that some index information is pre-computed and interleaved with the data on the broadcast channel. Consequently, by looking up the index the client is able to predict the arrival of the desired data. Thus, it can stay in the *power saving* mode and tunes into the broadcast only when the requested data arrives. The research issues in this direction include indexing structures suitable for broadcast and data and index interleaving techniques [3, 12, 13, 14, 19].

- **Bandwidth Allocation.** Because wireless bandwidth is a scarce resource in a cellular mobile network, much work has been done on bandwidth allocation for data dissemination in order to achieve a better access performance [1, 8, 16, 18, 25]. For a single cell, the issues are 1), how much bandwidth should be assigned to on-demand access and pushed broadcast respectively and 2), whether a data item should be served using on-demand access or pushed broadcast. When it goes to a multi-cell environment, the problem becomes more complex since besides the aforementioned two issues we have to decide how much bandwidth should be assigned to each cell.

- **Multi-Cell Data Scheduling.** In a multi-cell environment, since the mobile client may move frequently from one cell to another, it is desirable to satisfy a request before the client leaves the current cell; otherwise the data delivery would only waste the bandwidth. Thus, scheduling algorithms that addressed this problem have been explored [28].

- **Location-Dependent Data Management.** In location dependent services, the answer to a query depends on where the query is issued [21]. Such services exist in the traditional environments (e.g. AltaVista Local[1]). However, it is believed that their benefits lies in mobile computing environments. For example, it is attractive for someone to issue a query like ”find me the nearest hotel with room rate below $1000” from a wireless device in the middle of a journey. As we will see in the next section, location-dependent data brings new issues for data caching and indexing strategies.

In the next section, we focus on the research issue of location-dependent data management that we are currently working on.

4. LOCATION-DEPENDENT DATA MANAGEMENT

Location-dependent information services (LDISs) have a large variety of applications, such as nearest object searching (e.g., finding the nearest restaurant) and local information access (e.g., local traffic, news, and attractions). This section discusses data management issues for LDISs. We first describe location models in the next subsection, followed by a discussion on the issues of location-dependent cache invalida-

[1]Website at http://local.altavista.com/.

tion, location-dependent cache replacement, and location-dependent data indexing.

4.1. Location Model

Location is an important piece of information for representing, storing, and querying location-dependent information. In a location-dependent query, a location is needed to be specified explicitly or implicitly. A location model depends heavily on the underlying location identification technique employed in the system. The available mechanisms for identifying locations can be categorized into two basic approaches:

- **Geometric Model:** A location is specified as a 3-dimensional coordinate, e.g., *GPS* [7] and *Bat* [9]. The main advantage of geometric models is their compatibility across heterogeneous systems. However, because of the considerable cost and complexity involved in providing accurate fine-grained location information, the cost/performance ratio of geometric models might not be promising for a large number of applications such as the "nearby restaurant" example.
- **Symbolic Model:** The location space is divided into disjoint zones, and each zone is identified with a unique name. Examples are Critick [20] and the cellular infrastructure. Being discrete and well-structured, location information based on symbolic models is easier to manage compared to that based on geometric models. For example, location data is much more amenable for database storage and retrieval; they can help analyze location information, such as individual mobility patterns.

These two location models have different overheads and levels of precision in representing a valid scope. The appropriate location model to be adopted depends on the application.

For ease of illustration, we define two notions: *valid scope* and *valid scope distribution*. The *valid scope* of a data instance is defined as the area within which this instance is the only answer.[2] Under a symbolic location model, a valid scope is a set of cell ids. Under a geometric location model, a valid scope often takes the shape of a polygon in a two-dimensional space. Since an item may have different instances at different locations, it is associated with a set of valid scopes, which is called the *scope distribution* of the item. To illustrate, let's consider a four-cell system under a cell-based location model. Suppose that the nearby restaurant for cell 1 and cell 2 is instance A, and the nearby restaurant for cell 3 and cell 4 is instance B. Then, the valid scope of A is $\{1, 2\}$, the valid scope of B is $\{3, 4\}$, and the scope distribution of the nearby restaurant item is $\{\{1, 2\}, \{3, 4\}\}$.

[2] Note that we distinguish *data instance* from *data item*. For example, "nearest restaurant" is an item, and the data instances for this item vary for different locations.

4.2. Location-Dependent Cache Invalidation Schemes

Caching is an important technique to improve system performance in mobile computing environments. For location-dependent information in a mobile environment, cache inconsistency can be caused by location change of a client (hereafter called *location-dependent updates*), which does not exist in the traditional environments. Therefore, to maintain cache consistency, cache invalidation should be performed for location-dependent updates.

An idea for location-dependent cache invalidation is to make use of the *validity information* of the data items [26]. Specifically, the server delivers the valid scope along with a data instance to a mobile client, and the client caches the data as well as its valid scope for later validity checking. This strategy involves two issues, namely validity checking time and validity information organization, that need to be addressed. Since a query result only depends on the location specified with the query, we proposed to do validity checking for a cached data instance until it is queried. Furthermore, with this strategy a data instance that is invalid with respect to the current location is not necessarily removed from the cache immediately as it may be valid again later. For validity information organization, we describe in the following two subsections two schemes, i.e., ISI and CEB, for a symbolic location model and a geometric location model, respectively.

4.2.1. The Implicit Scope Information Scheme

The ISI scheme was proposed for a cell-based symbolic location model [26]. Under this scheme, the server enumerates the scope distributions of all items and numbers them sequentially. The valid scopes within a scope distribution are also numbered sequentially. For any instance of data item i, its valid scope is specified by a 2-tuple (SDN_i, SN_i), where SDN_i is the scope distribution number and SN_i denotes the scope number within this distribution. The 2-tuple is attached to a data instance as its valid scope. For example, suppose there are three different scope distributions (see Figure 2) and data item 4 has distribution 3. If item 4 is cached from cell 6 (i.e., $CID = 6$), then $SDN_4 = 3$ and $SN_4 = 3$. That implies that the cached instance for item 4 is valid in cells 6 and 7 only.

<table>
<tr><td>CID</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td></tr>
<tr><td>(SDN)Scope Distribution #1</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td></tr>
<tr><td>(SDN)Scope Distribution #2</td><td colspan="3">1</td><td colspan="3">2</td><td colspan="3">3</td><td colspan="3">4</td></tr>
<tr><td>(SDN)Scope Distribution #3</td><td colspan="2">1</td><td colspan="3">2</td><td colspan="2">3</td><td colspan="3">4</td><td colspan="2">5</td></tr>
</table>

Figure 2: An Example of Data Items with Different Distributions

At the server-side, a *location-dependent IR* is periodically broadcast. It consists of the ordered valid scope numbers (SN) in a cell for each scope distribution. For example, in cell 8, the server broadcasts $\{8, 3, 4\}$ to mobile clients, where the three numbers are the SN values in cell 8 for scope distributions 1, 2, and 3, respectively (Figure 2).

The validity checking algorithm works as follows. After retrieving a location-dependent *IR*, for item i the client compares the cached SN_i with the SDN_i-th SN in the location-dependent *IR* received. If they are the same, the cached data instance is valid. Otherwise, the data instance is invalid. For example, in cell 8, the client checks for the cached data item 4 whose $SDN_4 = 3$ and $SN_4 = 3$. In the broadcast report, the SDN_4-th (i.e. 3rd) SN equals to 4. Therefore, the client knows that the data instance for item 4 is invalid. The performance analysis of ISI and the comparison of ISI to other cache invalidation schemes were presented in [26].

4.2.2. *The Cache-Efficiency-Based Method*

We now discuss location-dependent cache invalidation strategies for a geometric location model. Under this model, there are two basic schemes for representing valid scopes, i.e., *polygonal endpoints* and *approximate circle* [29]. However, these two schemes perform worse due to either high overhead or imprecision of the invalidation information. To enhance performance, a generic *caching-efficiency-based* (CEB) method for balancing the overhead and the knowledge of valid scopes was proposed in [29].

In the CEB method, a new metric *caching efficiency* was introduced. Suppose that the valid scope of a data instance is v, and v'_i is a sub-region contained in v (see Figure 3). Let D be the data size, $A(v'_i)$ the area of any scope of v'_i, and $O(v'_i)$ the overhead needed to record the scope v'_i. The caching efficiency of the data instance with respect to a scope v'_i is defined as follows:

$$E(v'_i) = \frac{A(v'_i)/A(v)}{(D + O(v'_i))/D} = \frac{A(v'_i)D}{A(v)(D + O(v'_i))} \qquad (1)$$

Having defined the caching efficiency metric, the proposed CEB method works as follows.

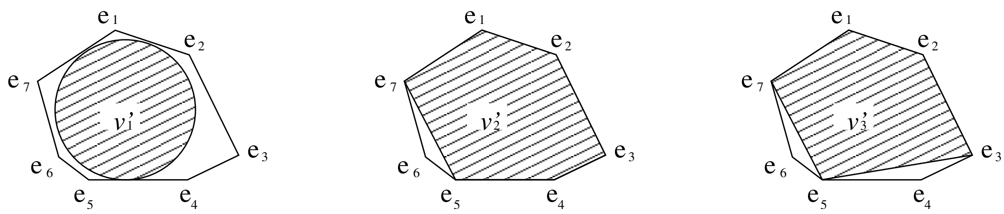
> *For a data instance with a valid scope of v, given a candidate valid scope set $V' = \{v'_1, v'_2, \cdots, v'_k\}$, $v'_i \subseteq v$, $1 \leq i \leq k$, we choose the scope v'_i that maximizes caching efficiency $E(v'_i)$ as the valid scope to be attached to the data.*

Figure 3 illustrates an example where the valid scope of the data instance is $v = p(e_1, e_2, \cdots, e_7)$, and v'_1, v'_2, and v'_3 are three different sub-regions of v, $A(v'_1)/A(v) = 0.788$, $A(v'_2)/A(v) = 0.970$, and $A(v'_3)/A(v) = 0.910$. Assuming that the data size D is 128 bytes, eight bytes are needed to represent an endpoint, and four bytes for the radius of an inscribed circle, we have $O(v) = 56$, $O(v'_1) = 12$, $O(v'_2) = 48$, and $O(v'_3) = 40$. Thus, we obtain $E(v) = 0.696$, $E(v'_1) = 0.721$, $E(v'_2) = 0.706$, and $E(v'_3) = 0.694$. As a result, we will choose v'_1 as the valid scope to be attached to the data. The simulation-based evaluation performed in [29] showed that CEB is very effective and outperforms other cache invalidation methods.

4.3. Location-dependent Cache Replacement Polices

Since a mobile client has only limited cache space, cache replacement is another important issue to be tackled in client cache management. In traditional cache replacement policies, access probability is considered the most important factor that affects cache performance. A probability-based policy is to replace the data with the least access probability. However, in LDISs besides access probability, there are two other factors, namely *data distance* and *valid scope area*, which have to be considered in cache replacement strategies.

We realize that a promising cache replacement policy should choose as its victim data with a low access probability, a small valid scope area, and a long distance if data distance is also an influential factor [29]. Therefore, we



(a) v_1'=inscribed circle of v

(b) $v_2' = p(e_1, e_2, e_3, e_4, e_5, e_7)$

(c) $v_3' = p(e_1, e_2, e_3, e_5, e_7)$

Figure 3: An Example of Possible Candidate Valid Scopes ($v = p(e_1, e_2, \cdots, e_7)$)

have proposed two cost-based cache replacement policies, *PA* and *PAID*, which integrate the factors that are supposed to affect cache performance.

- **Probability Area (PA):** As the name suggests, for this policy, the cost of a data instance is defined as the product of the access probability of the data item and the area of the attached valid scope. That is, the cost function for data instance j of item i is as follows:

  $$c_{i,j} = P_i \cdot A(v'_{i,j}), \qquad (2)$$

  where P_i is the access probability of item i and $A(v'_{i,j})$ is the area of the attached valid scope $v'_{i,j}$ for data instance j. The *PA* policy chooses the data with the least cost as its victim when cache replacement is performed.

- **Probability Area Inverse Distance (PAID):** Compared with *PA*, this scheme further integrates the data distance factor. For the *PAID* policy, the cost function for data instance j of item i is defined as follows:

  $$c_{i,j} = \frac{P_i \cdot A(v'_{i,j})}{D(v'_{i,j})}, \qquad (3)$$

  where P_i and $A(v'_{i,j})$ are defined in the same way as above, and $D(v'_{i,j})$ is the distance between the current location and the valid scope $v'_{i,j}$. Similar to *PA*, *PAID* ejects the data with the least cost during each replacement. Depending on different methods of calculating $D(v'_{i,j})$, we have two variants of *PAID*, i.e., *PAID-U* and *PAID-D*. In *PAID-U*, the data distance is undirectional and is calculated regardless of the current direction of movement of the client. In *PAID-D*, the calculation of the data distance considers the client's current direction of movement: if the client is currently moving away from the valid scope, the distance is multiplied by a very large number δ; otherwise it is calculated normally as in *PAID-U*. This way, *PAID-D* favors keeping the data in the direction of movement of the mobile client.

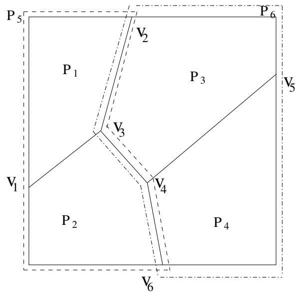
From the performance evaluation [29], we came to the conclusion that *PA* and *PAID* substantially outperform the existing policies such as *LRU*, *P*, and *FAR*. In particular, consideration of the valid scope area can improve performance in all settings, and consideration of the moving direction in calculating data distance is effective only for short query intervals and short moving intervals.

4.4. Location-Dependent Data Indexing

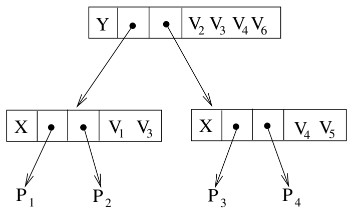
As mentioned, to serve mobile data access, either on-demand access or pushed broadcast can be used. In both methods, a fundamental problem is how to efficiently locate appropriate data with respect to a query point. To this end, we proposed a novel index structure, called *D-tree*, for location-dependent data under a geometric location model [27].

4.4.1. The D-tree Index Structure

The D-tree is a binary height-balanced tree similar to the kd-tree [6]. The D-tree is built based on the divisions between data regions. When constructing the D-tree, we recursively partition a space that consists of a set of data regions, into two *complement* subspaces containing similar number of regions until every subspace has one region only. The partition of two subspaces is represented by a set of divisions between regions, i.e., one or more polylines in a



(a) Divisions in the Example



(b) D-tree for the Example

Figure 4: Index Construction Using the D-tree

two-dimensional space. Figure 4(a) shows the partitions of an example data set containing four data instances with valid scopes P_1, P_2, P_3, and P_4, respectively. The polyline $pl(v_2, v_3, v_4, v_6)$ partitions the original space into P_5 and P_6, and $pl(v_1, v_3)$ and $pl(v_4, v_5)$ further partition the spaces P_5 into P_1, P_2 and P_6 into P_3, P_4, respectively. The detailed partition algorithm was described in [27].

For processing location-dependent queries, the problem becomes that given a query point, how to locate the pointer pointing to the appropriate data. Starting from the root, we recursively follow either the left pointer or the right pointer based on the partition information until a data pointer is reached.

The performance of the proposed D-tree has been thoroughly evaluated using both synthetic and real datasets [27]. The result showed that overall the D-tree performs much better than the typical *object decomposition* and *object approximation* based index structures [4, 5, 6].

5. CONCLUSION AND FUTURE WORK

In recent years, mobile computing has gained much attention from all of the government, industrial and academic sectors. This paper gives an overview of the characteristics of mobile computing environments, the system model, and related data management issues. In particular, we discussed in detail our recent work on client-side caching and query processing in location-dependent information services.

It is natural that we can further our research in pervasive computing environments, where computing is embedded in the environment "invisible" to the user and on a large variety of heterogeneous devices. To achieve more efficient and automatic information dissemination for pervasive computing environments, research has to been done on the representation of user interests, how to match the data/event to the user interests, and more efficient information dissemination models required. The emergence of the pervasive computing paradigm will continue to open up new research opportunities for us.

6. REFERENCES

[1] S. Acharya, R. Alonso, M. Franklin, and S. Zdonik. Broadcast disks: Data management for asymmetric communications environments. In *Proc. SIGMOD'95*, pages 199–210, May 1995.

[2] D. Barbara. Mobile computing and databases – a survey. *IEEE Trans. on Knowledge and Data Engineering*, 11(1):108–117, 1999.

[3] D. Barbara and T. Imielinksi. Sleepers and workaholics: Caching strategies for mobile environments. In *Proc. of SIGMOD'94*, pages 1–12, May 1994.

[4] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf. *Computational Geometry*. New York: Springer-Verlag, 1997.

[5] E. Bertino, B. C. Ooi, R. Sacks-Davis, K. L. Tan, J. Zobel, B. Shilovsky, and B. Catania. *Indexing techniques for advanced database systems*. Boston: Kluwer Academic, 1997.

[6] V. Gaede and O. Günther. Multidimensional access methods. *ACM Computing Surveys*, 30(2):170–231, June 1998.

[7] I. A. Getting. The global positioning system. *IEEE Spectrum*, 12(30):36–38, 43–47, Dec. 1993.

[8] S. Hameed and N. H. Vaidya. Efficient algorithms for scheduling data broadcast. *ACM/Baltzer J. Wireless Networks (WINET)*, 5(3):183–193, 1999.

[9] A. Harter, A. Hopper, P. Steggles, A. Ward, and P. Webster. The anatomy of a context-aware application. In *Proc. MobiCom'99*, pages 59–68, Aug. 1999.

[10] Q. L. Hu and D. L. Lee. Cache algorithms based on adaptive invalidation reports for mobile environments. *Cluster Computing*, 1(1):39–48, Feb. 1998.

[11] Q. L. Hu, D. L. Lee, and W.-C. Lee. Performance evaluation of a wireless hierarchical data dissemination system. In *Proc. MobiCom'99*, pages 163–173, Aug. 1999.

[12] Q. L. Hu, D. L. Lee, and W.-C. Lee. Indexing techniques for power management in multi-attribute data broadcast. *ACM/Kluwer J. Mobile Networks and Applications (MONET)*, 6(2):185–197, 2001.

[13] Q. L. Hu, W.-C. Lee, and D. L. Lee. Power conservative multi-attribute queries on data broadcast. In *Proc. ICDE'2000*, pages 157–166, Feb. 2000.

[14] Q. L. Hu, W.-C. Lee, and D. L. Lee. A hybrid index technique for power efficient data broadcast. *J. Distributed and Parallel Databases (DPDB)*, 9(2):151–177, March 2001.

[15] T. Imielinski and B. R. Badrinath. Wireless mobile computing: Challenges in data management. *Communications of the* ACM, 37(10):18–28, 1994.

[16] T. Imielinski and S. Viswanathan. Adaptive wireless information systems. In *Proc. SIGDBS'94*, pages 19–41, Oct. 1994.

[17] T. Imielinski, S. Viswanathan, and B. R. Badrinath. Data on air - organization and access. *IEEE Trans. on Knowledge and Data Engineering*, 9(3), 1997.

[18] W.-C. Lee, Q. L. Hu, and D. L. Lee. A study on channel allocation methods for data dissemination in mobile computing environments. *ACM/Baltzer J. Mobile Networking and Applications (MONET)*, 4(2):117–129, 1999.

[19] W.-C. Lee and D. L. Lee. Using signature techniques for information filtering in wireless and mobile environments. *J. Distributed and Parallel Databases (DPDB)*, 4(3):205–227, July 1996.

[20] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan. The cricket location-support system. In *Proc. MobiCom'2000*, pages 32–43, Aug. 2000.

[21] Q. Ren and M. H. Dunham. Using semantic caching to manage location dependent data in mobile computing. In *Proc. MobiCom'2000*, pages 210–221, Aug. 2000.

[22] S. Sheng, A. Chandrasekaran, and R. W. Broderson. A portable multimedia terminal. *IEEE Communications Magazine*, 30(12):64–75, Dec. 1992.

[23] J. Xu, Q. L. Hu, D. L. Lee, and W.-C. Lee. SAIU: An efficient cache replacement policy for wireless on-demand broadcasts. In *Proc. CIKM'2000*, pages 46–53, Nov. 2000.

[24] J. Xu, Q. L. Hu, W.-C. Lee, and D. L. Lee. Performance evaluation of an optimal cache replacement policy for wireless data dissemination under cache consistency. *Submitted to IEEE Trans. on Knowledge and Data Engineering*, under revision, May 2001.

[25] J. Xu, D. L. Lee, and B. Li. On bandwidth allocation for data dissemination in cellular mobile networks. *ACM/Kluwer Journal of Wireless Networks (WINET)*, to appear, 2002.

[26] J. Xu, X. Tang, and D. L. Lee. Performance analysis of location-dependent cache invalidation schemes for mobile environments. *IEEE Trans. on Knowledge and Data Engineering (TKDE)*, to appear, 2002.

[27] J. Xu, B. Zheng, W.-C. Lee, and D. L. Lee. Energy efficient index for querying location-dependent data in mobile broadcast environments. *Submitted for possible publication*, Nov. 2001.

[28] B. Zheng and D. L. Lee. Processing location-dependent queries in a multi-cell wireless environment. In *Proc. MobiDE'01*, pages 54–65, May 2001.

[29] B. Zheng, J. Xu, and D. L. Lee. Cache invalidation and replacement strategies for location-dependent data in mobile enviroements. *Submitted for possible publication*, July 2001.