Data Representation and Indexing in Location-Enabled M-Services

Christian S. Jensen   Simonas Šaltenis
Department of Computer Science, Aalborg University

Abstract

Rapid, sustained advances in key computing technologies combine to enable a new class of computing services that aim to meet needs of mobile users. These ubiquitous and intelligent services adapt to each user's particular preferences and current circumstances—they are personalized. The services exploit data available from multiple sources, including data on past interactions with the users, data accessible via the Internet, and data obtained from sensors. The user's geographical location is particularly central to these services. We outline some of the research challenges that aim to meet the data representation and indexing needs of such services.

1. Introduction

Several trends in hardware technologies combine to provide the enabling foundation for m-services. These trends include continued advances in *miniaturization* of electronics technologies, *display devices*, and *wireless communications*. Other trends are the improved *performance* of general computing technologies and the general improvement in the *performance/price ratio* of electronics hardware. Perhaps most importantly, *positioning technologies* are becoming increasingly available and accurate.

It is expected that the coming years will witness very large quantities of wirelessly Internet-worked objects that are location-enabled and capable of movement to varying degrees. Some predict that each of us will soon have approximately 100 on-line objects. Example objects include consumers using WAP-enabled mobile-phone terminals and personal digital assistants, tourists carrying on-line and position-aware "cameras" and "wrist watches," vehicles with computing and navigation equipment, etc.

These developments pave the way to a range of qualitatively new types of Internet-based services. These types of services—which either make little sense or are of limited interest in the traditional context of fixed-location, PC- or workstation-based computing—include the following: traffic coordination, management, and way-finding, location-aware advertising, integrated information services, e.g., tourist services, safety-related services, and location-based games that merge virtual and physical spaces.

The area of location-based games offers good examples of services where there is a need to track the positions of the mobile users. In the recently released BotFighters game, by the Swedish company It's Alive, players get points for finding and "shooting" other players via their mobile phones (using SMS messages or WAP). Only players close by can be shot. To enable the game, players can request the positions of other nearby players. In such mixed-reality games, the real physical world becomes the backdrop of the game, instead of the world created on the limited displays of wireless devices. These games are expected to generate very large revenues in the years to come.

A single, *generic scenario* may be envisioned for location-enabled m-services: Moving service users disclose their positional information (position, speed, velocity, etc.) to services, which in turn use this and other information to provide specific functionality.

Each service maintains a log of the requests received and integrates this with other customer data. This data is used for analyzing the user interaction with the service and for mass-customization of the service, so that each user receives a service customized to the user's specific preferences and needs and current situation. In addition, the accumulated data is used for delayed modification of the services provided, and for longer-term strategic decision making.

The integration of location information into this scenario has received scant attention and offers fundamental challenges. One common theme is that of extending techniques that work well for static data to support dynamic, continuously evolving data. Another common theme is to extend techniques, which have previously been subjected only to "simple" data, to support more complex types of data and queries. We proceed to describe some challenges that relate to data representation and indexing. These are but a few of the many challenges posed by the developments outlined above (see also, e.g., [8]).

2. Data Representation

Integrated Representations In order to reuse data across m-services, an integrated representation, or data model, of all relevant geo-referenced content must be developed. Such a data model will promote reuse of content and lower-level services when new services are developed. This may lead to more rapid development of services as well as to higher-quality services, both of which are important to a provider of location-enabled m-services who wishes to maintain a competitive advantage. For example, certain types of services, such as games, have a relatively short life time; new services must be made available quite frequently to retain the customer base.

To better appreciate the complexity of this challenge, it is instructive to learn that The Danish Road Directorate maintains more than 1000 attribute values for each position on each road in their database, which covers all of Denmark. A substantial fraction of these need to be reflected in an integrated data model.

These attributes are quite specialized in that they are closely related to the roads themselves. In addition, the data model must capture the "real content," which is much more voluminous and open-ended. For example, such content includes information about stores, e.g., their opening hours, available inventory, and current sales, and about cultural events, e.g., the artists, attendance prices, and seat availability.

While most geo-referenced content is stationary or changes location only at discrete times, some content changes continuously. The locations of service users are prominent examples of the latter. Such content must also be captured in the data model.

The envisioned data model must be conceptually simple and must also permit the provision of efficient services.

Euclidean Spaces, Obstacles, and Transportation Networks Much initial work on query processing in relation to moving objects has assumed that the geo-referenced objects that queries concern live in a 1-, 2-, or 3-dimensional Euclidean space (e.g., [1, 5, 6]). Indeed, research based on this assumption may lead to results that are very useful for some application contexts.

In other contexts, different assumptions may be more appropriate. Specifically, it is sometimes appropriate to expect some objects to constrain the locations of other objects. For example, the movement of joggers in a forest is constrained by fenced, private properties, etc. As another example, the movement of a ship is constrained by shallow water and land. Little research in query processing has taken into account such constraints on the movement and locations of objects.

These blocking objects do not reduce the dimensionality of the space in which the objects are embedded [10]. In other application contexts, it is appropriate to constrain the locations of objects to a transportation network embedded in, typically, 2-dimensional Euclidean space [3]. This in some sense reduces the dimensionality of the space available for movement—the term 1.5-dimensional space has been used. Examples abound. Cars typically move in transportation networks, and the destinations such as private residences, hotels, and shops may be given locations in transportation networks.

While some proposals for query processing techniques have taken into account the presence of transportation networks, many more opportunities exist for developing query processing techniques that exploit the presence of such networks. Put simply, better performing techniques may be developed when it is possible to constrain the movement of objects to a network, than when objects move more freely.

It should also be noted that Euclidean distances are often not of interest in this setting. Rather, techniques that apply to settings with obstacles or network constrained objects will have to contend with other notions of distance. For some such notions, the distance between two stationary objects varies over time.

The kinds of queries one may expect also depend on the network more generally. Folklore has it that 80-90% of all automobile drivers move towards a destination. This suggests that drivers typically follow network paths that are known ahead of time. This statement may also lead one to assume that drivers have a particular interest in different properties pertaining to the path from their current location to their destination. They may want to know the remaining distance and the anticipated remaining travel time. In this context, typical queries may not be simple range or nearest neighbor queries.

Uncertainty Location uncertainty is inherent to all or most geo-references of content and must be taken into account both in the representation and querying of the content. Object locations are sampled according to some specific protocol [9]. As examples, the users may disclose their locations when they desire a service, they may supply their locations at regular intervals, or they may keep track of where the service thinks they are and then issue location samples to the service exactly when necessary in order for the service's record to be within a certain threshold of the actual location. In this latter example, the threshold is dependent on the specific service desired. Because location data are obtained via sampling, complete traces of the objects' movements are unavailable; rather, the service only knows the locations of the objects at discrete times.

Additionally, the samples themselves are imprecise. The sample imprecision is dependent on the technology used and the circumstances under which a specific technology is used. For example, the cellular infrastructure itself offers quite different precisions than do GPS and server-assisted GPS. The robustness also varies. In other words, the accuracy of the positioning is highly dependent on the user's location.

If a precise (but incorrect) trace is maintained for each user, queries may return suboptimal results. On the other hand, if an overly imprecise record of the positions is kept, query results will also be suboptimal. Maintaining a very accurate record of each user's trace will yield the best query results, but may lead to poor query performance. The envisioned data model for geo-referenced content must maintain a representation that is adequately precise and that does not adversely affect query performance.

3. Indexing

The aim of indexing is to make it possible for multiple users to concurrently and efficiently retrieve desired data in very large databases. Indexing techniques are becoming increasingly important because the improvement in the rate of transfer of data between disk and main memory cannot keep pace with the growth in storage capacities and processor speeds.

Hyper-Dynamic Data In location-enabled m-services that involve large amounts of content, that rely on access to up-to-date information, and where continuous variables (e.g., the locations of users) are monitored, updates occur very frequently.

Traditional indexing techniques do not work well for hyper-dynamic data. Stated briefly, the indices have to be explicitly updated when changes occur to the data. For large, continuously changing datasets, e.g., those capturing the positions of moving objects, the constant updating would tend to render the use of indices either impractical or totally impossible. One problem is that the large volumes of updates and the mechanisms that regulate the concurrent use of the indices would combine to block the querying of these structures.

The fact that existing solutions can accommodate only relatively few updates presents a serious problem for the kind of services considered here (as well as for other services that rely on the monitoring of continuous variables via some forms of sensors).

Accommodating Continuous Change The challenge of how to obtain the normal benefits of indexing when the data being indexed change continuously is an important one. Two types of techniques may be envisioned: those that reduce the numbers of updates and those that make the processing of updates more efficient. We consider each type in turn.

One approach is to model the positions of objects as functions of time. This alleviates the need for frequent updates, as updates are only needed when the changes in function parameters and the time since the last update combine to yield a difference between the real location of the object and the position believed by the database that exceeds the threshold dictated by the services being supported.

Several researchers have explored the use of linear functions for indexing the current and future positions of moving objects in 1-, 2-, and 3-dimensional spaces [3, 6, 7]. The use of more general functions has yet to be explored, as does the combined indexing of the past, current, and future positions of moving objects.

Next, if the locations of moving objects are constrained to a transportation network, the network may also be exploited in various ways to reduce the number of updates. Many open issues remain.

Complementary techniques of accommodating frequent updates should also be studied. For example, buffer techniques may be applied [2]. Briefly, these remedy the inefficiencies of transferring blocks with little data between main memory and disk by buffering updates. The goal of using advanced buffering techniques is to guarantee that the I/O operations needed to maintain an index under updates completely use all the space available in disk pages for useful data. This goal should be reached without adverse effects on query performance.

As other examples, distributed techniques and large main-memory resident structures may be used aggressively to avoid the I/O bottleneck.

Another promising direction is to exploit approximation techniques to enable monotonically improving, as-good-as-possible answers to queries within specified soft or hard deadlines. These may enable querying of almost up-to-date data, which in turn may reduce the need for prompt updates.

References

[1] P. K. Agarwal, L. Arge, and J. Erickson. Indexing Moving Points. *Proc. of the PODS Conf.*, pp. 175–186 (2000).

[2] L. Arge. External-Memory Algorithms with Applications in GIS. In *Algorithmic Foundations of Geographic Information Systems*, LNCS 1340, pp. 213–254 (1997).

[3] G. Kollios, D. Gunopulos, and V. J. Tsotras. On Indexing Mobile Objects. *Proc. of the PODS Conf.*, pp. 261–272 (1999).

[4] D. Pfoser and C. S. Jensen. Capturing the Uncertainty of Moving-Object Representations. In *the Proc. of the SSDBM Conf.*, pp. 111–132 (1999).

[5] D. Pfoser, Y. Theodoridis, and C. S. Jensen. Novel Approaches in Query Processing for Moving Object Trajectories. *Proc. of the VLDB Conf.*, pp. 395–406 (2000).

[6] S. Šaltenis, C. S. Jensen, S. T. Leutenegger, and M. A. Lopez. Indexing the Positions of Continuously Moving Objects. *Proc. of the ACM SIGMOD Conf.*, pp. 331–342 (2000).

[7] S. Šaltenis and C. S. Jensen. Indexing of Moving Objects for Location-Based Services. *Proc. of the International Conf. on Data Engineering*, to appear (2002).

[8] O. Wolfson, B. Xu, S. Chamberlain, and L. Jiang. Moving Objects Databases: Issues and Solutions. *Proc. of the SSDBM Conf.*, pp. 111–122 (1998).

[9] O. Wolfson, A. P. Sistla, S. Chamberlain, and Y. Yesha. Updating and Querying Databases that Track Mobile Units. *Distributed and Parallel Databases* 7(3): 257–387 (1999).

[10] O. Wolfson, S. Chamberlain, S. Dao, L. Jiang, and G. Mendez. Cost and Imprecision in Modeling the Position of Moving Objects. In *Proc. of the International Conf. on Data Engineering*, pp. 687–688 (2000).