Improving Usability of Location-Based Services with User-Centric Data Querying

Artem Katasonov

Information Technology Research Institute, University of Jyväskylä,
P.O.Box 35 (Agora), FIN-40014, Jyväskylä, Finland
artem.katasonov@titu.jyu.fi

Abstract. In this paper, we describe a computational model that produces a dataset which we consider to be an appropriate response for "give me a description of my neighborhood" type of queries in location-based services. Our attempt is to reflect the abstraction ability of a specific user, and in this way to maximize the usability of the service under restrictions on data volume posed by technical, economical and cognitive factors of mobile environments. We also present the results of instantiating the model for a simple LBS.

1 Introduction

Development of mobile networks and terminals provides basis for advanced mobile services and applications. Voice capabilities of mobile networks have been augmented with data capabilities of increasing speed, mobile phones have already reached considerable complexity and performance, got Java-support and color displays. One of the important new features is support for determining the location of a mobile terminal. The location data can be transmitted to third parties offering various services, including services for personal information and navigation (yellow-pages, route guidance), finding and tracking services (friends, children, property), and physical services (taxi, emergency services). While all of those are considered to be Location-Based Services (LBS), in this paper we restrict consideration to the first group only.

Overviews of LBS enabling technologies and technical issues can be found in books devoted to LBSs, e.g. [4], or numerous journal and conference papers, e.g. [3, 14]. LBSs have been a hot topic for both researchers and practitioners for a decade already, pushed forward by various players including mobile operators expecting LBSs to form an important revenue-generating class of mobile services, mobile phone manufacturers, Geographic Information Systems (GIS) community, and other content providers (e.g. publishers of yellow-pages directories). However, LBSs are still, in 2004, in a very early stage; commercially available services are few and rather simple.

Since the main value of LBSs comes from providing the user with some relevant information at appropriate place and time, data management is a major LBS concern. In addition, peculiarities of the mobile computing environment make communication of data to the users (one of the data management activities) a non-trivial problem. Mobile networks have limited bandwidth and high latency for transmitting data. Mobile devices have limited memory for storing, limited performance for processing and small displays

W. Lindner et al. (Eds.): EDBT 2004 Workshops, LNCS 3268, pp. 365–374, 2004.

for representing data. In addition, mobile users are typically in an unstable environment in varying conditions, where their cognitive capacity is demanded for other tasks as well. The mobile users have less “mental bandwidth” - capacity for absorbing and processing content - than a stationary user in front of a PC since the interaction with the mobile phone often is reduced to a secondary task that must not interfere with their primary task (e.g., driving or walking) [2].

Failing to address those issues may lead to situations when information does not arrive to the user in time, or when the user is unable to access or comprehend it in time, thus making the service undependable. There is also an economic aspect – in practice, a user has to pay for every byte of transmitted data, for connection time in GSM or for data volume in GPRS and above. Therefore, there is an obvious need for limiting the amount of data transmitted and presented to the user. However, doing that in a straightforward way may render the service useless (see example below).

In recent years, researchers from the database field have been actively attacking LBS data management issues. As [7] lists, the range of the studied questions includes data placement (centralization vs. distribution), data replication, indexing, query scheduling, data caching, broadcast (push) strategies, and combination of broadcast with on-demand access. The relevant research also includes indexing of moving objects [12], and multi-dimensional data modeling [5] for LBSs.

All those investigations should collectively contribute to optimization of processing of location-dependent queries, reduce response times and cost of data delivery in LBSs. However, the database researchers seem to have made so far little attempt to extend the notion of “location-dependent query” beyond basic spatial queries, i.e. range or nearest-neighbors search. So, query classification in [7] includes only local vs. nonlocal (current vs. specified location), and simple vs. general queries (equality conditions vs. spatial or nonspatial constraints). Very recently however, the discussion is started also of queries that would take the transportation network into account [6, 11], thus increasing the utility of LBSs.

However, one of the common queries in LBSs is the indefinite “give me a description of my neighborhood”. Such a query is traditionally executed with one or a series of range queries; the user is presented therefore with all the data items (using selective inclusion based on their theme or other attributes) linked to locations within a certain distance from the user. In other words, only the total data extent and maybe granularity is affected. Even if the network distance is used, the result is probably too simplistic. For example, some of the existing LBSs provide the user with a map depicting his neighborhood. Usability of this function is usually quite low, in part because as in the traditional or web cartography only the extent (spatial and thematic) and granularity of the map are controlled. An overview map fitting into a mobile phone display (or small enough to be transmitted in reasonable time) loses most of the detail and is therefore of a little use. But a map detailed enough can show only, say, a few city blocks. Given that accessing other portions of the map and zooming operations take considerable time (new queries have to be made), it is difficult to put that information into use either.

On the other hand, in Mobile Human-Computer Interaction (HCI) and GIS fields, it is common to speak about the need for using context-aware user-adapted data representations [1, 2, 10, 15]. The talk goes beyond controlling the total extent/granularity;

in general, removing any detail that is unnecessary and presenting only relevant is advocated. For example, [15] proposes using "focus maps" in LBSs, where areas that are farther away from the current user location are represented with less detail than those in the close vicinity. [1] discusses "context mediation", in which selection for presentation amongst different map elements and amongst their variants is based on a set of utility functions defined on elements attributes and spatial distances. HCI and GIS studies, however, usually only describe general principles and heuristics for creating such data representations; we did not encounter in literature descriptions of formal algorithms or models that would give means for doing this automatically and efficiently.

We consider it to be a strange situation that those technical and cognitive aspects of the mobile environment (see above) tend to be attacked separately (technical by the database researchers and cognitive by the HCI researchers), even while both have a common denominator – amount of information.

In this work, we make a step towards an integrated solution. We describe a computational model that automatically produces a dataset which we consider to be an appropriate response for "give me a description of my neighborhood" type of queries. The dataset is both focused (as in [15]) and user-centric, it includes only those details that are believed to be relevant for the specific user in the specific context. The objective we pursue is to maximize the usability of the service under some posed restrictions on data volume. We also present an example of implementing the model in simple LBS.

2 Novel Type of Location-Dependent Queries

Our attempt is to mimic the abstraction ability of a human. Humans instinctively know how to filter out unnecessary detail, this is an important survival trait, since we cannot possibly save in our brain all the detail around us. An abstraction can be understood as a selection of a set of objects, attributes or processes out of a larger set of objects, attributes and processes, according to certain criteria determined by the current task [13].

Let us consider what happens, for example, when a human looks at a printed map. A map is an abstraction of reality aimed to serve needs of some group of users. However, it is still a multi-purpose instrument while it is not designed for one specific user in one specific context. Therefore, every user of a map conducts abstraction further, filtering out unnecessary details.

The area of interest for the user is around his current location. If he has a destination then the area of interest includes also all the way from the current location to that point. The area of interest does not have a strict border. The user pays primary attention to objects that are close to his current location and to the way to the destination, objects that are farther away are noticed also, but less attention is paid to details (this leads to the idea of "focus maps", as in [15]). Only some types of objects represented on the map are relevant for the user. For example, he may be interested in restaurants but not in hotels or interested only in motorways but not in footpaths. Different groups of objects have different regions, inside which they are of interest for the user. For example, the user may be interested in hotels that are anywhere in the city, and restaurants only if they are quite close to the current location or the way (this is noticed in [1] proposing that different utility functions for distance could defined for different object types). Different groups of objects have different relative importance. The same might be true even for objects inside one group. Depending on the current goals, the user could be interested in maps at different scales and overall detail levels. If the user is driving at a high speed then a low-detailed map is usually enough, whereas if the user is walking then he could need a detailed one.

These observations are quite general and apply to any kind of location-dependent data. We conclude that LBSs call for a novel type of location-dependent queries, which would be (1) spatio-contextual, i.e. implement range search in a multidimensional space, (2) focused, and (3) motion-adaptive. In the next section, a computational model is described realizing such a query.

3 Computational Model

For every object (data item), our model estimates its relative importance, based on its spatial distance from the user and the context of the user. Given that the total allowed amount of details (data volume) is set, the relative importance of objects define whether they will be represented in the result dataset and also the levels of detail, with which they will be represented.

The model calculates the relative importance of an object based on the distance from the user to this object in some metric "importance" space. This space is multidimensional. Two of its axes are obviously the spatial X and Y coordinates (or longitude and latitude). However, all the other coordinate axes depend on the specific implementation and information that the service has about objects and about the user. Therefore, developers of a specific LBS need to define all these axes and decide about methods for transformation of the available data into values of these coordinates. All the coordinates are to be metric; this means that the distance d between any two values of a coordinate can be calculated.

Both users and objects are modeled as points (or regions if appropriate) in this space. Besides X and Y, the meaning of a coordinate values is different for a user and for an object. For an object, a coordinate value represents the value of an object property, for the user, it represents the preferred value of this property – the closer the value of the property for the object to the preferred value, the more important the object is for the user. Those preferred values could be requested directly from the user. Another approach is to estimate them based on information the service may have about the user, including permanent properties (e.g. gender), relatively permanent properties (e.g. age, preferences), and context parameters (e.g. driver/pedestrian, current needs). While providing obvious advantages, use of this approach may restrict the possible set of coordinates.

For calculating the distance between a user and an object, our choice is use of the Manhattan metrics that defines the overall distance D as a simple sum of distances with respect to individual coordinates $D = d_1 + d_2 + ... + d_N$. The Manhattan metrics has some advantages in comparison with, e.g. the Euclidean metrics. First, it is computationally simpler. Second, it allows treating coordinates in groups. We could define, for example, $D = d_{spatial} + d_{nonspatial}$ and use then another method for calculation of the spatial distance (the Euclidean metrics or take into account the transportation network). Third, it allows considering coordinates in fully separate way, e.g. the overall average distance $\overline{D}$ is equal to the sum of average distances $\overline{d_i}$ with respect to individual coordinates, i.e. $\overline{D} = \overline{d_1} + \overline{d_2} + ... + \overline{d_N}$ (see below).

Since we cannot use the same units for all the coordinates, there is a question about coordinates ratio. Our choice is to define some weights k_i for all of the coordinates and therefore have $D = k_1 d_1 + k_2 d_2 + ... k_N d_N$. Ideally, coefficients k_i should be user-dependent (estimated or requested directly). However, in practical applications, they could be predefined by designers of the system.

The user of an LBS is mobile. The objects could also be mobile (e.g. taxi cars). Additionally, mobility may be also considered with respect to other coordinates, not only the spatial location. Therefore, we use predicted average distance over some period (it could be, e.g., the average time between consequent requests). With respect to one coordinate we define it as

$$\overline{d} = \frac{\int_0^T k(t) d(t) dt}{\int_0^T k(t) dt} \tag{1}$$

T is the period, $d(t)$ is the distance at time t, $k(t)$ is a function of t describing the degree of determinacy of the user and the object motions. If we are sure that the distance between the user and the object will change by the $d(t)$ rule, we can assign $k(t)$ to one. More practically, when $d(t)$ is only a prediction, $k(t)$ could be selected as $k(t) = \frac{T-t}{T}$ defining continuous linear fade from one to zero.

The simplest case (however, the most probable) is the following. For both the user and the object, the values of the coordinate, x_u and x_o, as well as the speeds of change of the coordinate, v_u and v_o, at time zero are only known. Therefore, the prediction for d(t) is $|(x_u + v_u t) - (x_o + v_o t)|$. In such a case, $\overline{d}$ could be easily computed.

As mentioned above, using the Manhattan metrics, average distances with respect to individual coordinates can be combined as

$$\overline{D} = k_1 \overline{d_1} + k_2 \overline{d_2} + ... + k_N \overline{d_N}. \tag{2}$$

Based on $\overline{D}$, the relative importance I of the object is calculated. We can use various equations, for example linear or quadratic

$$I = \frac{1}{\overline{D}} \quad or \quad I = \frac{1}{\overline{D}^2} \tag{3}$$

Based on I, the level of detail LOD is calculated, which the object will have in the result dataset. On designing the system, some $\widehat{I}$ value is to be selected. If an object has I equal to or greater than $\widehat{I}$ then it is represented with all the available details, i.e. its LOD is assigned to one. We assume also that we have the total volume of details V fixed (could come from data volume restrictions). Therefore, for all of the objects included into the result we have $\sum_i LOD_i \leq V$. Formally, this means that there is some lowest importance value I_0, under which objects are not included into the result, i.e. their LOD are assigned to zero. Consequently,

$$LOD = \begin{cases} 1, & \text{if } I \geq \widehat{I}, \\ I/\widehat{I}, & \text{if } I_0 \leq I < \widehat{I}, \\ 0, & \text{if } I < I_0 \end{cases} \tag{4}$$

Algorithmically, we need to select objects from the top of the descending-order list until $\sum LOD$ reaches V. There is no need to calculate I_0 directly.

The level of detail for an object, as we defined above, is a value in the interval between zero to one. If LOD is equal to zero the object is not represented at all; if LOD is equal to one the object is represented with all the available details. With LOD in between, a simplified representation of the object is included into the result dataset. Simplification affects both spatial and nonspatial properties.

The task of simplification of an object representation is similar to what is called generalization in cartography and GIS [9]. Generalization is a complicated process which is difficult to automate. We, however, advocate for simplification of each object separately and use of simple operations only removing some data and not introducing new, i.e. against e.g. data aggregation. The reason is not only to allow automatic generalization, but also to provide for simple join of datasets, if continuous querying or progressive data transmission is used. We advocate also for generalization that would fulfill the requirement that if $LOD_2 < LOD_1$ then the dataset representing an object at LOD_2 is a subset of the dataset at LOD_1. Then, knowing the set of LOD values for all the objects present in the user device memory, we could calculate what exactly data is there, regardless of how many queries and updates have contributed to it.

For nonspatial properties, generalization therefore involves only omitting some of them. We assume that for every property, a threshold value is defined. When LOD of an object is greater than or equal to this value, the property is included into the simplified representation of the object. In selection of thresholds, naturally, the importance of different properties and amount of data needed for encoding them is to be considered.

Generalization of a spatial geometry involves omitting some of its points. One simple approach is the following procedure. We calculate the number of points to be retained $m = round(M*LOD)$, where M is the number of points in the geometry. If geometry is a line and $m < 2$ then we assign m to 2 (end points are retained). If geometry is a polygon and $m < 3$ then we assign m to 1 (collapse to a point). For geometry simplification, an appropriate approach is use of the classical Douglas-Peucker algorithm (see [9]). Among other advantages, this algorithm has the important feature that the produced set of points for m_1 is always a subset of the set for m_2 if $m_1 < m_2$. The sequence of critical points could even be computed beforehand and stored. Then in run time, one needs only to select m first points from this sequence.

4 Example of Implementation

In this section, we present a simple example of implementing the described model. User-centric data querying functionality was added to the MultiMeetMobile LBS pilot system [8, 14]. The system is a mobile service for navigation and location-based information in a city environment, aimed for Java-enabled PDAs and smart-phones.

The system is based on two datasets. The first dataset on the street network of the city Jyväskylä was provided by the National Land Survey of Finland. A basic unit in this dataset is a street segment, which endpoints are either street crossings or dead ends. Segments are characterized with a large set of attributes; however, we use only some of them. Based on the type attribute, we divided the network into three sub-networks:

highways, streets, and footpaths. The other relevant attributes are: name of the street to which the segment belongs, allowed direction of traffic, length of the segment, address range on both sides of the segment, and spatial geometry consisting of at least two endpoints and possibly a number of vertices. The coordinate system used is the Finnish National Map Grid Coordinate System known as KKJ. This system uses rectangular map projection coordinates X and Y with meter as the measurement unit. This simplifies calculation on such data and makes it more human understandable.

The second dataset on points of interest in the city Jyväskylä has been artificially produced. We selected three important groups of points of interest: shops, restaurants, and hotels. We defined the following attributes for these points of interest: name, address, description, and spatial location.

The following metric coordinates were defined. X and Y are the rectangular map projection coordinates with meter as the measurement unit. For a street segment, we calculate the average of all its points and take coordinates of this average point as X and Y of the segment. Z is abstract elevation. All the objects have $Z = 0$. For a user, it can be any non-negative value (in meters). Therefore, the user experiences the area from a vantage-point, as if he were, for example, in an airplane. $iHighways$, $iStreets$, $iFootpaths$, $iShops$, $iRestaurants$, and $iHotels$ are coordinates with real values in the interval 0...1 encoding the relative importance of different data themes for the user. For an object, only one of these coordinates is applicable (that corresponds to its own type) and is equal to one.

X and Y coordinates of the user, as well as the current speeds of their change, are retrieved from the emulator of the location service. Z and the preferences coordinates are requested from the user directly (by the client application) and included as parameters into a query to the server.

For this demonstrative implementation, we did not consider the question of selection of model parameters very carefully. Some of parameters were thoughtfully selected, appropriate values for other were found in series of trials. The ratio coefficients k for X, Y and Z were assigned to one, for the preferences variables to 1000. $\widehat{I}$ was assigned to 0.005, V to 45, and T to 60. Calculation of I applied linear formula $1/\overline{D}$. The following threshold values were selected for the segment attributes: street name - 0.2, direction of traffic - 0.4, length of segment - 0.6, address range - 0.8; for the attributes of points of interest: name - 0.25, address - 0.5, description - 0.75.

Since making calculations for all the objects, for which we have data, is very time consuming, we use pre-selection based on X and Y coordinates, with the size of pre-selection window estimated based on Z. All the data is stored in an Oracle database with Spatial extension. Pre-selection is performed with Spatial native range queries. Consequent processing is performed by in-house software.

A few figures follow demonstrating what kind of output described instance of the model produces. Since a graphical representation may depict only the spatial component of data, we decided not to present screenshots of the client terminal display, and generated instead pictures conveying a little more information. The pictures have white background and the color of objects (street segments and points of interest) ranges from almost white (that means that their LOD is close to zero) to black (LOD is equal to one). Thin lines in the pictures represent footpaths, thicker ones correspond to streets, and even thicker to highways. The larger black circle in the center of each picture shows the position of the user. The area covered by each picture is 700 by 700 meters square.

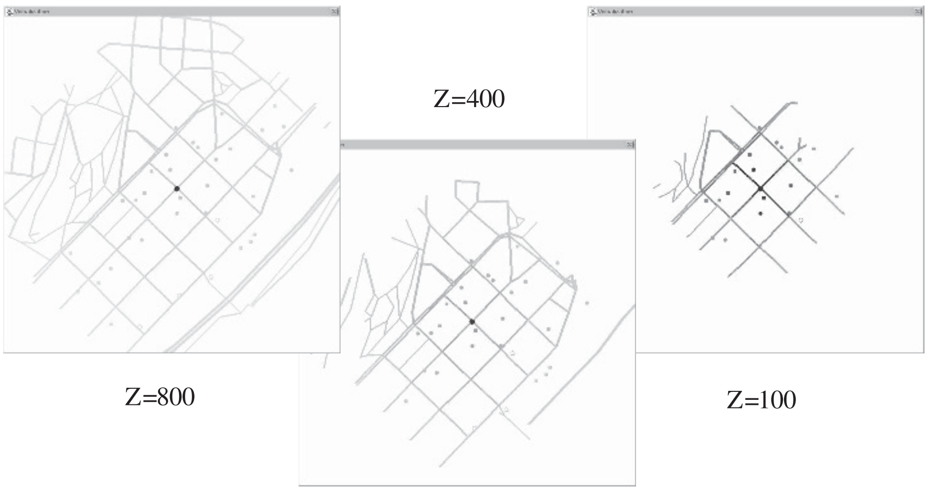


Fig. 1. Different overall detail levels

Figure 1 presents the same city area for different values of Z coordinate (800, 400 and 100 meters correspondingly). As can be seen, with reduction of Z the represented area decreases; however, the levels of detail of all the objects grow (scale change effect). Especially in the picture for Z=100, it can be noticed (by color difference) that the dataset is focused – the objects in the user's close vicinity are represented with more detail than those farther away. Change in segments' vertices number is probably difficult to notice. However, recall that, most importantly, LOD affects also nonspatial data – so, segments

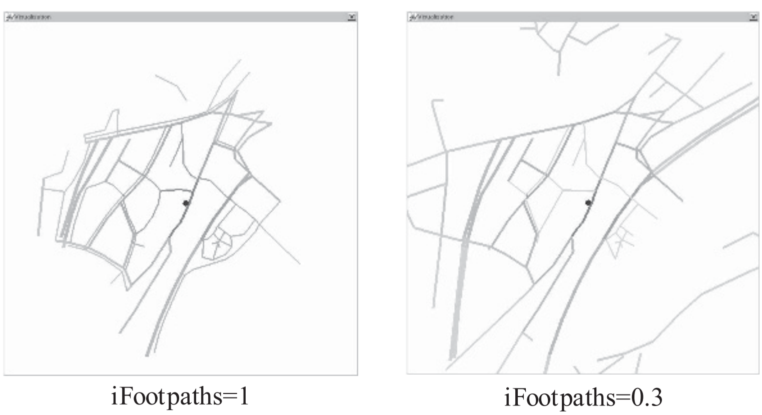
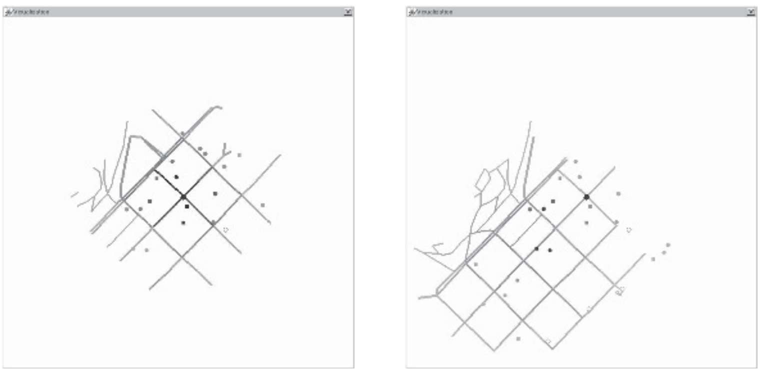


Fig. 2. When streets are more important than footpaths

shown as black include all the properties up to the address range, while those light-gray may have even the street name omitted.

In 2, both pictures depict the same area; however, while the picture on the left is for the case $iFootpaths$ = 1, the picture on the right shows the case when $iFootpaths$ = 0.3 ($iHighways$ = 1 and $iStreets$ = 1 for both pictures). As can be seen, there are many footpaths in the picture on the left, and their extent is equal to that of bigger roads. However, when their relative importance has been reduced, footpaths left only in the close vicinity of the user, and level of detail of those left was greatly reduced. This also enabled representing additional objects of other types (in this case streets and highways).

Figure 3 compares the cases when the user is stationary and when the user is moving in the southwest direction ($v_x = v_y$ = - 8 m/sec). As can be seen, the selection window has shifted to the direction of motion. The most detailed region now is not around the current user location. Additionally, the average level of detail of objects has reduced and the extent of the representation has slightly increased.



$v_x = v_y = 0$ $v_x = v_y = -8$

Fig. 3. When user is moving southwest

5 Conclusions

In this paper, we described a computational model that produces a dataset which we consider to be an appropriate response for “give me a description of my neighborhood” type of queries in location-based services. Our attempt is to reflect the abstraction ability of a specific user, and in this way to maximize the usability of the service under restrictions on data volume posed by technical, economical and cognitive factors of mobile environments. As can be seen from our example of instantiating the model, its simple mathematics allows achieving quite complex effects, such as data focusing, controlling the balance of different data themes, taking mobility into account, and other.

The technical performance of the model is yet to be validated. Anyway, we see our work as rather the first step towards a solution that will actually be applied in practice. Our work can be considered to belong mainly to the Mobile HCI field. However, a formal presentation of HCI ideas in the form of a computational model provides for continuation of the work from the database perspective, towards further formalization and optimization.

The questions related to criteria/algorithms for selecting model parameters are left outside the scope of the present paper and belong to future work. We believe that our approach in general is viable. However, empirical usability evaluation with real users and for different LBS types is another important direction of future work.

Acknowledgements

The greater part of this work was done in MultiMeetMobile project, which was financially supported by the National Technology Agency of Finland (TEKES) under contracts 40599/99, 40779/99, 40640/00, and industrial partners Hewlett-Packard, Nokia Networks and Yomi.

References

1. D. Chalmers, N. Dulay, and M. Sloman. A framework for contextual mediation in mobile and ubiquitous computing applied to the context-aware adaptation of maps. *Personal and Ubiquitous Computing*, 8(1):1–18, 2004.
2. D. Chincholle, M. Goldstein, M. Nyberg, and M. Eriksson. Lost or found? A usability evaluation of a mobile navigation and location-based service. In *Proc. Mobile HCI 2002*, volume 2411 of *LNCS*, pages 211–224, 2002.
3. D. Dao, C. Rizos, and J. Wang. Location-based services: Technical and business issues. *GPS Solutions*, 6(3):169–178, 2002.
4. J. Hjelm. *Creating Location Services for the Wireless Web*. John Wiley & Sons, 2002.
5. C. S. Jensen, A. Kligys, T. B. Pedersen, and I. Timko. Multidimensional data modeling for location-based services. *The VLDB Journal*, 13(1):1–21, 2004.
6. C. S. Jensen, J. Kolar, T. B. Pedersen, and I. Timko. Nearest neighbor queries in road networks. In *Proc. 11th ACM Int. Symposium on Advances in GIS*, pages 1–8, 2003.
7. D. L. Lee, J. Xu, B. Zheng, and W.-C. Lee. Data management in location-dependent information services. *IEEE Pervasive Computing*, 1(3):65–72, 2002.
8. J. Markkula, A. Katasonov, and A. Garmash. Developing MLS location-based service pilot system. In *Proc. IFIP 7th Conf. on Intelligence on Networks*, pages 229–244, 2002.
9. R. B. McMaster and K. S. Shea. *Generalization in digital cartography*. Resource Publication of the Association of American Geographers, Washington, D.C., 1992.
10. A.-M. Nivala and L. T. Sarjakoski. Need for context-aware topographic maps in mobile devices. In *Proc. 9th ScanGIS Conference*, pages 15–29, 2003.
11. D. Papadias, J. Zhang, N. Mamoulis, and Y. Tao. Query processing in spatial network databases. In *Proc. 29th VLDB Conference*, pages 802–813, 2003.
12. S. Saltenis and C. S. Jensen. Indexing of moving objects for location-based services. In *Proc. 18th Int. Conf. on Data Engineering*, pages 463–472, 2002.
13. S. Timpf. Abstraction, levels of detail, and hierarchies in map series. In *Proc. Int. Conf. Spatial Information Theory - Cognitive and Computational Foundations of Geographic Information Science*, volume 1661 of *LNCS*, pages 125–139, 1999.
14. A. Tsalgatidou, J. Veijalainen, J. Markkula, A. Katasonov, and S. Hadjiefthymiades. Mobile E-commerce and location-based services: Technology and requirements. In *Proc. 9th ScanGIS Conference*, pages 1–14, 2003.
15. A. Zipf. User-adaptive maps for location-based services (LBS) for tourism. In *Proc. 9th Int. Conf. for Information and Communication Technologies in Tourism*. Springer, 2002.