Geo-Mobile Queries: Sketch-Based Queries in Mobile GIS-Environments

David Caduff[1] and Max J. Egenhofer[2]

[1] Department of Geography, University of Zurich – Irchel,
Winterthurerstr. 190, CH-8057 Zurich, Switzerland
caduff@geo.unizh.ch

[2] Department of Spatial Information Science and Engineering,
5711 Boardman Hall, University of Maine, Orono, Maine
max@spatial.maine.edu

Abstract. Traditional GIS tools are well suited for desktop workstations, but need to be adapted in order to satisfy the requirements of mobility. We propose a concept for sketch-based spatial querying in mobile GIS environments. This concept combines newest techniques for spatial querying and mobile technologies. Such a combination is beneficial because it allows formulating queries by drawing the desired spatial configuration on touch-sensitive screens, thus avoiding typing complex statements in some SQL-like query language. Client-server architectures in mobile environments are characterized by low and fluctuating bandwidth, and by frequent disconnections. We discuss client-server strategies in mobile environments and suggest an adaptive client-server architecture for geo-mobile querying. It is shown that adaptation to the mobile environment is necessary in order to ensure efficiency of geo-mobile queries.

Keywords: mobile, wireless GIS, geo-mobile, sketch-based queries.

1 Introduction

The combination of mobile appliances and wireless technology allows transferring certain portions of GIS technology from the desktop into the users' hands. One of the key operations in GISs is the retrieval of spatial information. Conventional query languages, such as the SQL, use text-based statements. While text-based query statements work well within data domains where data can easily be stored in tables, they lack expressiveness and flexibility within more complex domains, such as images, maps, or other spatially related, multi-dimensional data [1-3]. Images and especially maps, however, are an integral part of most GISs and, therefore, query methods for GISs need to be sufficiently expressive and efficient.

The use of handheld devices, such as cellphones, PDAs, or sub-notebooks as tools for querying spatial data in mobile environments, is becoming increasingly popular [4]. For spatial queries in mobile environments, particular attention needs to be paid to query languages that are responsive to bandwidth fluctuations, frequent disconnections, and various constraints of the mobile device, such as limited input bandwidth.

K.-J. Li and C. Vangenot (Eds.): W2GIS 2005, LNCS 3833, pp. 143 – 154, 2005.

Recent research activities in visual information retrieval systems investigated novel techniques to query spatial data more efficiently [5, 6]. Visual information retrieval systems stress the use of visual tools to formulate a query. Unlike the SQL-based approach, these systems focus more directly on the end result, since an example of a user's query can be used as a formulation of a query statement. This paper studies visual information retrieval techniques, specifically sketch-based queries systems [5, 7, 8] in the context of mobile GISs. We will term this type of queries *geo-mobile queries*. As a framework and foundation, we use a sketch-based user interface for information retrieval systems [5, 6, 9], which allows users to formulate a query in form of a sketch that represents the spatial scene users want to find in a spatial database.

GIS applications on PDAs are unlikely to have all relevant data sources readily available in the device; hence, another critical aspect of information retrieval in mobile GIS environments is the response time. In addition, the use of handheld devices as spatial-query-by-sketch interfaces underlies the many restrictions of mobile appliances and wireless technology, such as interface restrictions, limited bandwidth, and power supply. One goal of this paper is to investigate the interplay between mobile clients, wireless networks, and static servers, and prove the practicability of Spatial-Query-by-Sketch [5] in dynamic use configurations. The main goal of this paper, however, is to extend the theoretical foundation of sketch-based querying [5, 6] from static to mobile environments.

The result of this investigation is a client-server architecture for geo-mobile querying that implements application adaptation in order to overcome the limitation of mobile environments. The investigation of the workflow and dataflow for sketch-based information retrieval systems in client-server architectures demonstrates the suitability of handheld devices as appropriate tools for performing geo-mobile queries. It is shown that adaptation to the mobile environment is necessary in order to ensure efficiency of geo-mobile queries.

2 Approach

Client-server architectures in mobile environments differ from the classical, wired architectures in many aspects [10]. The first part of this paper is concerned with identifying constraints of mobility that affect both, system and user behavior, so that properties of geo-mobile query systems can be defined. In the second part, these properties are applied to designing an adaptive client-server architecture that allows efficient geo-mobile querying of spatial data under varying conditions of the mobile environment.

Adaptation in a mobile client-server environment consists of three main steps: (1) resource monitoring, (2) an adaptation strategy, and (3) the adaptation process [11, 12]. Resource monitoring is concerned with identifying vital resource parameters for the application, while the adaptation strategy defines how these parameters influence adaptation for a specific system. Finally, the adaptation process controls the functionality of the application. Adaptation includes both the client and the server; therefore, we use a *mobile sketch*, which is derived from the sketched spatial scene and contains additional information about the client for guiding the adaptation process. The mobile sketch propagates the level of adaptation from the client to the server and contains a symbolic representation of the sketched scene, which is used for completion of the query process. We focus on the query formulation, because the resulting concepts and findings are generic and, therefore, valid for a wide range of applications. The presentation and analysis of the results, on the other hand, are application-specific and, therefore, should be investigated separately.

3 The Geo-Mobile Query-by-Sketch Architecture

The architecture of the geo-mobile query-by-sketch application is based on application-aware adaptation [12, 13] and the extended client-server model [10, 11, 14]. Since client and server share the responsibility of executing spatial queries, the adaptation logic resides on both, client and server. The idea of this client-server adaptation strategy is that the application on the mobile client is able to react to changes of the resources in the mobile environment. The request for adaptation is then propagated to the server in order to adjust the server's functionality.

3.1 The Mobile Client

The mobile client is the central part of the query system. It provides the interface that users use to draw the spatial configurations. The sketch is either drawn using a pen on a touch-sensitive screen or some other input device that allows freehand drawing of sketches. In addition to the user interface, the mobile client also hosts the adaptation logic. The architecture of the mobile client consists of the operating system running on top of the hardware, a middleware layer that acts as a mediator between system resources and the applications, and the geo-mobile querying application.

The role of the operating system and the middleware layer is to monitor scarce resources, and to respond to external events. The resource monitor keeps track of the resources, allocates the available resources among competing applications, and notifies the applications of changes to these resources. Complementary, the role of the application is to adapt to changing conditions by using the information and resources provided by the resource monitor. The application reacts to the changes by switching to a different level of functionality that guarantees best performance.

The change in resources affects both the user interface and the generation of the query statement. The user interface adapts to the change by enabling or disabling a specific set of functions for the current level of adaptation, while the query statement is issued in form of a mobile sketch. The mobile sketch is a digital representation of the user input, that is, the sketched scene, and reflects the effects of adaptation to available resources in terms of informative content and metadata of the sketched scene. The data transfer between client and server and the request for adaptation on the server is based on a transfer mechanism that utilizes the mobile sketch as control protocol.

3.2 The Server

The server is the core of the retrieval mechanism and is responsible for processing the query against the database. The role of the server in the adaptation process is passive because the client monitors the mobile environment and decides the extent of adaptation. For every query statement (i.e., mobile sketch) transmitted, the server identifies the level of representation and executes the appropriate tasks.

The query process is partitioned in such a way that the steps on the client and the steps on the server are complementary and result in a digital sketch that can be processed against a database. For instance, a mobile client with poor resources parses the user input, creates the mobile sketch, and transmits it instantly to the server. The server resumes the querying process and generates the objects, creates the digital sketch, processes the query against the database, and finally prepares the result for presentation to the user.

Sketch-based queries typically generate a set of results, which the server prepares for presentation to the user. The presentation of the result is based on a set of parameters of the mobile environment in order to guarantee efficient result browsing. Such parameters include screen size of the mobile client, color depth, etc. These parameters are captured in form of a user profile on the client and transmitted to the server.

3.3 Functional Partitioning

Functional partitioning is the process of decomposing an application's functionality into non-divisible pieces, called functional objects, and to allocate the objects to system components [13]. For our purpose, we desist of the system-allocation step of the functional partitioning process and replace it with a data flow analysis, that is, we merely use functional partitioning to define the modules (i.e., functional objects) of the system at a high level. This approach supports allocation of functionality to system components based on interaction and communication, which is an important aspect of mobile client-server applications [15].

The purpose of the data flow analysis is to show movement of information at a high level between a system and its environment, as well as data movement at a lower level between the individual modules of the system. The main objective of the data flow analysis of Spatial-Query-by-Sketch is to document how information flows within the system and to define the boundaries of the single modules of the system in order to design a plan on how to partition the client and the server. The tasks of the Spatial-Query-by-Sketch system are specified in Egenhofer [5] and Blaser [6]. The granular objects we investigate are sketch parsing, object processing, digital sketch generation, and query processing (Fig. 1).

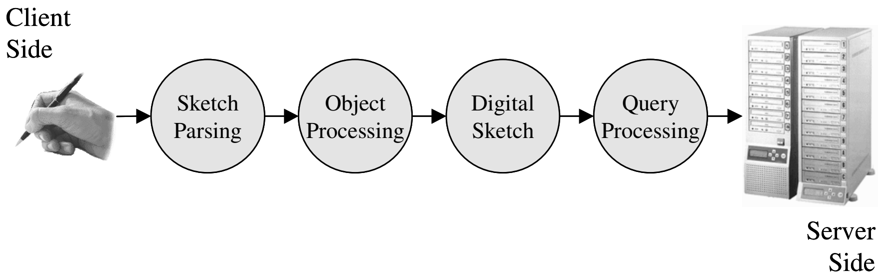

Fig. 1. The functional objects of Spatial-Query-by-Sketch

The main purpose of the first functional object, sketch parsing, is to generate a simplified stroke from the user input. This operation reduces the amount of data, and consequently the costs of transmission. Therefore, it needs to be located on the client and a data flow analysis becomes needless. Similarly, the purpose of the query-processing object is to assess the similarity between the digital sketch and a set of sketches in the database. This comparison is based on geometric -, topologic -, metric -, and direction similarity [16]. As a result, the query processing is a CPU- and memory-intense task that necessitates according infrastructure.

These considerations leave two functional objects left for the data flow analysis, that is, the object processing and the generation of the digital sketch. The simplified strokes from the sketch parser are the input for the second functional object, the object processing. The object processing analyses the stroke and yields a command that is instantly executed, or a data object (i.e., ASCII text, symbol, line, or region) that is added to the digital sketch. In addition, the process extracts the kernel and centerline of the objects and adds them to the digital sketch. Kernel and centerline are vital components of the digital sketch and are needed during the evaluation of the spatial relations (i.e., topology, direction, and metric), as well as during the query processing.

The second functional object that is relevant for the data flow analysis is the generation of the digital sketch. The first step in this process generates the association graph that defines the set of binary spatial relations among the sketched objects. The association graph generation is a computing intense task and the resulting binary relations together with the contained information (i.e., topology, direction, and metric) attribute considerably to the digital sketch in terms of memory consumption. A large digital sketch results in increased traffic between client and server, and thus contradicts the general principle of reducing data flow between the two parties. This observation advocates placing the generation process for the digital sketch on the server, if the available resources on the mobile client are insufficient.

The data flow analysis produces a coarse plan on how to map the functional objects onto system components. The first important discovery of the analysis is that two functional objects, sketch parsing and query processing, need to be assigned to the client and to the server, respectively. The second discovery is that the two other functional objects, (i.e., the object processing and the generation of the digital sketch) are responsible for data exchange between client and server and need to be carefully partitioned (Fig. 2).

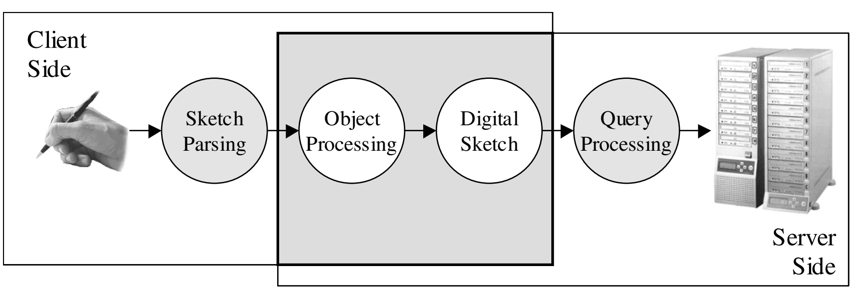

Fig. 2. Allocation of functional objects to system entities

In order to glue the functional objects together in a distributed system, an efficient data exchange mechanism is required. This mechanism and the refinement of the function allocation are investigated in the next section.

4 The Mobile Sketch

The result of the first three steps of the Spatial-Query-by-Sketch is a digital representation of a spatial scene consisting of spatial objects (points, lines, and regions) and the binary spatial relations among these objects. This digital representation of the sketched scene is called the *digital sketch* and is used to perform queries against the spatial database.

The main component of our approach is the mobile sketch. The mobile sketch is similar to the digital sketch as it is essentially a digital representation of the user's mental map of a spatial scene. Additionally, we extend and enable the mobile sketch to coordinate the workflow between client and server. The mobile sketch reflects important characteristics that are required to derive the digital sketch on the server, which is the final, meaningful representation of a sketch used for querying spatial databases [6].

4.1 Components of the Mobile Sketch

The mobile sketch is an abstraction of the digital sketch since its main goal is to efficiently capture the vital properties of the sketch at different levels of object representation. The main objective of this approach is to reduce the amount of data transmitted between client and server. The structure of the mobile sketch consists basically of four distinct sections. The following is a high level specification of the different sections.

Mobile Sketch Signature: A mobile sketch begins with a signature containing general information such as creation time, document size, and history. The most important information conveyed by the mobile sketch signature, however, is the level of representation of the mobile sketch. This information enables the server to perform the remaining steps that are necessary to create the digital sketch.

Hardware Profile: This profile contains system-specific information describing the properties of the mobile client. For instance, mobile clients may differ in terms of screen size, color depth, and resolution. This information is crucial for effective presentation of the query results on the mobile client.

User Profile: The user profile contains preferences that are set by the user on the mobile client. Such information includes the selected association graph model for the digital sketch, weights for binary spatial relations, and thresholds for result presentation.

Data Section: The data section reflects the sketch that the user draws on the touch-sensitive screen. Unlike the other three sections, which remain essentially the same for all adaptation levels, the content of this section may change depending on the chosen level of representation of the mobile sketch.

The purpose of the four sections of the mobile sketch is to enable the server to complete the generation of the digital sketch, support the query process, and supply parameters for a valuable presentation of the results. The next section discusses the generation scheme used to create the mobile sketch.

4.2 Mobile Sketch Generation Scheme

The scheme used to generate the mobile sketch may be described as a lossy compression technique applied on the digital sketch. Lossy compression techniques involve a compression such that if expanded less information may be available then what was in the original [17]. In return, such techniques generally obtain much higher compression ratios than is possible with lossless compression. Unlike most compression techniques, however, no decomposition and no compression algorithms are required to create multiple representations and reduce the amount of data of the mobile sketch. The degree of complexity (i.e., the actual object representation) of the mobile sketch is directly dependent on the number of steps involved in the creation of the sketch. This approach reduces not only the amount of data, but also the required infrastructure to create the mobile sketch. Therefore, it is the ideal approach for an adaptive application on a mobile client.

The sketch generation scheme is based on the functional objects of the query process: sketch parsing, object processing, and digital sketch generation. The general idea is that each of the three functional objects generates a mobile sketch at a different computational level, as depicted in Fig. 3. The mobile client selects an appropriate level of adaptation based on the available resources in the mobile environment, that is, the application running on the handheld device determines what steps are ideally performed on the mobile client given the set of parameters for the actual mobile environment. Accordingly, the selected level of adaptation defines the level of representation of the mobile sketch, since only the selected steps of the query process are executed. After the mobile sketch is generated, it is sent to the server where the generation of the mobile sketch is completed. Subsequently, the mobile sketch is converted into a digital sketch that can be used for the query against the database.

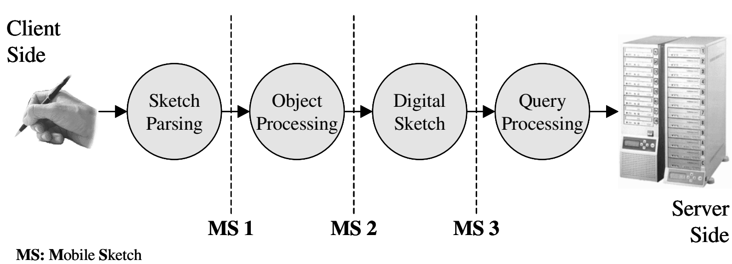

Fig. 3. Generation scheme for the multiple representations of the mobile sketch

4.3 Levels of Representation

The mobile sketch produced by the generation scheme may reflect any of the three different levels of representation. The term multiple representations in GISs refers to changes in geometric and topological structure of a digital object that may occur with the changing resolution at which that object is encoded for computer storage, analysis and depiction [18]. Accordingly, level of representation in our context does not refer to spatial details, since the sketched scene is the same for both, mobile sketch and digital sketch. Instead, it refers to qualitative and quantitative aspects (i.e., objects type, binary relations) of the mobile sketch compared to the digital sketch. The following description of the representation levels explains these differences.

MS 1-Simplified Line Strokes: Simplified line strokes are the lowest representation possible in a mobile sketch. Simplified line strokes basically consist of connected, time-stamped points (i.e., x- and y-coordinate, and creation time) as drawn by the user on the touch-sensitive user interface. Consequently, this level of representation requires the least powerful infrastructure and produces the smallest amount of data.

MS 2-Geographic Objects: This representation reflects the user input in terms of geographic objects (i.e., lines and regions), ASCII text, or symbols. The geographic objects consist of interconnected line segments and a set of properties. However, this level of representation contains no information about spatial relations among the sketched objects. The significant difference compared to the lowest representation level is the higher degree of computing resources required to process user input and to generate the objects and its properties.

MS 3-Digital Sketch: This level of representation corresponds to the digital sketch, as used for the query processing. It consists of a set of distinguishable sketched object and the corresponding spatial relations between them. The attributes and properties of the single object are the same as in the previous representation. The generation of the association graph of binary spatial relations and the assessment of topological, metrical, and directional attributes requires appropriate CPU and memory resources, and is therefore executed only if the resources are available. Furthermore, the amount of data increases drastically with the addition of the binary relations.

The mobile sketch facilitates an adaptive geo-mobile system architecture that guarantees both, an appropriate level of workload on the client and an amount of data that corresponds to the available network bandwidth. In order to achieve such adaptation to the mobile environment, we need to define a strategy that guides the adaptation.

5 Mobile-Aware Adaptation Strategy

Mobile-aware adaptation involves dynamic partitioning of the functionality between mobile host and server. By varying the partition of duties, however, we also vary the functionality of the user interface and above all, the quality of data produced on the mobile host. Consequently, adaptation involves the trading of data quality and user experience for resource consumption. The proposed architecture captures this notion of data degradation through three different levels of representation of the data produced on the mobile client (i.e., the mobile sketch).

5.1 Complementary Distribution of the Query Process

The representation levels define the degree to which data delivered to the server requires further processing. The lowest representation level results when resources on the mobile client are scarce and thus, full server support is required. The second level of representation applies when processing power, memory, and energy supply are abundant, but wireless communication with the server is not reliable or not existent. Finally, the application on the mobile client produces the highest level of representation if both, the handheld device and the wireless network provide sufficient resources. Fig. 4 illustrates the distributed query process and the three possible scenarios.

The choice of an appropriate adaptation level is based on predefined policies. Adaptation policies capture different system behavior in a flexible and customizable manner. The policies govern a discrete adaptation algorithm, which allows applications to move up along step or staircase shaped utility functions, rounding off the assigned value to the lower discrete adaptation level. The adaptation algorithm is discussed in the next section.

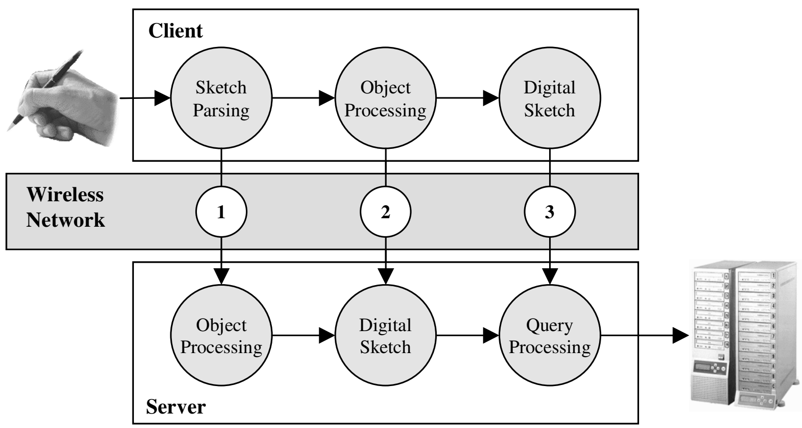

Fig. 4. Mobile-aware adaptation strategy

5.2 Adaptation Policy

In order to adapt to a changing environment, a system must evaluate its present situation and try to change the situation to another configuration that guarantees acceptable performance. Therefore, the role of the adaptation policy is central to capturing the application specific responses to resource availability. Due to the difficulty in obtaining an analytic expression that takes into account all possible parameters (e.g., CPU cycles, memory I/O operations, and jitter); profile-based modeling is used to approximate the mapping of the available resources to the application functionality.

We propose a profile-based, discrete adaptation policy, which allows applications to move up along step or staircase-shaped utility curves, rounding of the assigned parameters to the lower discrete parameter value (Fig. 5). Discrete adaptation requires complete increments of single steps to support multiple representations. It considers the portion of a step to determine a discrete representation for the mobile sketch.

The user profile defines the relation between the single steps of the utility curve, that is, it defines the thresholds that guide the transition from one step of the utility curve to the next. We utilize a compression factor (i.e., low, middle, high) to divide the resource in three sections, each representing a representation level for the mobile sketch. The compression factor divides the resource in a 1:2:3 (i.e., low: middle: high) ratio. The application adapts the thresholds for the steps every time the user changes the compression factor.

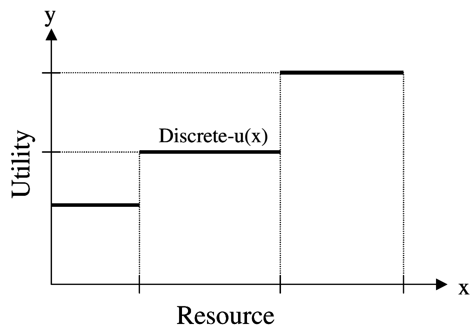

Fig. 5. Example of a discrete utility curve

We define two spaces that reflect the available resources with respect to the multiple representations; a Resource Space *R*, and a Performance Space *P*. *R* is dimensioned by resource characteristics in the mobile environment, which define the operational spectrum of the application. We utilize a two-dimensional resource space that

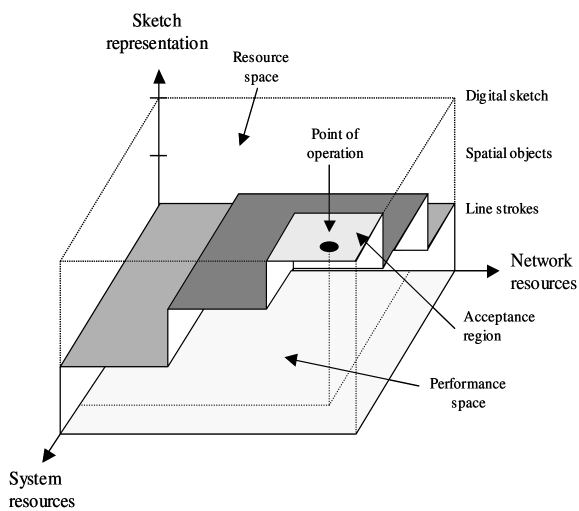

Fig. 6. Discrete adaptation model

includes system characteristics (i.e., CPU power and memory) and network properties (i.e., bandwidth and latency). Each axis reflects the resources assigned to the application by the middleware.

P is dimensioned along user-oriented parameters and includes three acceptance regions (i.e., representations of the mobile sketch). The Acceptance Region for a specific representation *AR*, of *P* is defined as the region in which the application is considered to be working properly with the current parameters. The adaptation model is illustrated in Fig. 6.

The application adapts its functionality every time it detects changes in the parameters received from the resource monitor. In addition, three protocols guide adaptation in case of abrupt changes of the wireless connection (i.e., a Disconnected Protocol, a Weak Connection Protocol, and a Connected Protocol). The disconnected protocol redirects the user input into a file that is stored locally. As soon as the connection is reestablished, the Connection Protocol or Weak Connection Protocol checks for such files and prepares them for transmission to the server.

6 Summary and Future Work

This paper investigates the implications of mobility on sketch-based information retrieval systems. Important issues in this context include challenges related to mobile technology, wireless communication, and the architectural structure of information retrieval systems in such environments. The main objective of this paper was to identify the major factors that influence the extension of sketch-based query techniques from static to dynamic mobile environments. The paper follows a top-down approach in that it first identifies characteristics of mobile computing and the challenges involved therein and creates an architectural framework that permits efficient sketch-based querying in mobile GIS environments.

Our work introduces a new adaptive approach for geo-mobile querying about which many questions remain unanswered and many variations and extensions remain to be explored. One important aspect of geo-mobile querying is context. Three types of context that need to be addressed by future research are spatial context dependency, context dependency of application adaptation, and task-based context dependency.

References

1. Egenhofer, M., Why not SQL! International Journal of Geographical Information Systems, 1992. **6**(2): p. 71-85.
2. Egenhofer, M., Deficiencies of SQL as a GIS Query Language, in Cognitive and Linguistic Aspects of Geographic Space: An Introduction, D.M. Mark and A.U. Frank, Editors. 1991, Kluwer Academic Publishers: Dordrecht, Boston and London. p. 477-492.
3. Egenhofer, M. and A.U. Frank, Query Languages for Geographic Information Systems. 1991, National Center for Geographic Information and Analysis: Orono, ME.
4. Artem, G., Management of Geographic Information in Mobile Environments, in Department of Computer Science and Information Systems. 2000, University of Jyväskylä: Jyväskylä, FIN. p. 107.

5. Egenhofer, M. Spatial-Query-by-Sketch. in VL '96: IEEE Symposium on Visual Languages. 1996. Boulder, CO: IEEE Society.
6. Blaser, A.D., Sketching Spatial Queries, in Spatial Information Science and Engineering. 2000, University of Maine: Orono, ME. p. 199.
7. Saund, E. and T.P. Moran. A perceptually-supported Sketch Editor. in Proceedings of the ACM symposium on User interface software and technology. 1994. Marina del Rey, CA USA.
8. Haarslev, V. and M. Wessel. Querying GIS with Animated Spatial Sketches. in IEEE Symposium on Visual Languages. 1997. Capri, Italy: IEEE Computer Society Press.
9. Blaser, A.D., Prototype Application Sketcho! 1999, University of Maine, Department of Spatial Information Science and Engineering and National Center for Geographic Information and Analysis: Orono, ME. p. 25.
10. Satyanarayanan, M. Fundamental Challenges in Mobile Computing. in Symposium on Principles of Distributed Computing. 1996.
11. Jing, J., A. Helal, and A. Elmagarmid, Client-Server Computing in Mobile Environments. ACM Computing Surveys, 1999. **31**(2): p. 117-157.
12. Katz, R.H., Adaptation and Mobility in Wireless Information Systems. IEEE Personal Communications, 1994. **1**(1): p. 6-17.
13. Satyanarayanan, M., et al., Application-Aware Adaptation for Mobile Computing. 1995, School of Computer Science, Carnegie Mellon University: Pittsburgh, PA. p. 7.
14. Rao, S.K., An Architecture for Adaptable Wireless Networks, in Department of Electrical Engineering and Computer Science. 2000, Massachusetts Institute of Technology. p. 65.
15. Imielinski, T. and B.R. Badrinath, Mobile Wireless Computing: Challenges in Data Management. Communications of the ACM, 1994. **37**(10): p. 18-28.
16. Blaser, A.D. and M.J. Egenhofer. A Visual Tool for Querying Geographic Databases. in Advanced Visual Interfaces - AVI2000. 2000. Palermo, Italy.
17. Sayood, K., Introduction to Data Compression. 2 ed. 2000, San Diego, CA: Academic Press, A Harcourt Science and Technology Company.
18. Buttenfield, B.P., NCGIA Research Initiative 3 - Multiple Representations. 1993, National Center for Geographic Information and Analysis: Buffalo. p. 27.