12-2003

An Egocentric Spatial Data Model for Intelligent Mobile Geographic Information Systems

Christopher E. Frank

Recommended Citation

Frank, Christopher E., "An Egocentric Spatial Data Model for Intelligent Mobile Geographic Information Systems" (2003). *Electronic Theses and Dissertations*. 574.
http://digitalcommons.library.umaine.edu/etd/574

AN EGOCENTRIC SPATIAL DATA MODEL FOR INTELLIGENT MOBILE GEOGRAPHIC INFORMATION SYSTEMS

By

Christopher E. Frank

B.S. University of Maine, 2001

A THESIS

Submitted in Partial Fulfillment of the

Requirements for the Degree of

Master of Science

(in Spatial Information Science and Engineering)

The Graduate School

The University of Maine

December, 2003

Advisory Committee:

Max J. Egenhofer, Professor of Spatial Information Science and Engineering, Advisor

Kathleen Hornsby, Assistant Research Professor, National Center for Geographic Information and Analysis

Silvia Nittel, Assistant Professor of Spatial Information Science and Engineering

AN EGOCENTRIC SPATIAL DATA MODEL FOR INTELLIGENT MOBILE GEOGRAPHIC INFORMATION SYSTEMS

By Christopher E. Frank

Thesis Advisor: Dr. Max J. Egenhofer

An Abstract of the Thesis Presented
in Partial Fulfillment of the Requirements for the
Degree of Master of Science
(in Spatial Information Science and Engineering)
December, 2003

Individuals in unknown locations, such as utility workers in the field, soldiers on a mission, or sightseeing tourists, share the need for an answer to two basic questions: "Where am I?" and "What is in front of me?" Because such information is not readily available in foreign locations, aids in the form of paper maps or mobile GISs, which give individuals an all-inclusive view of the environment, are often used. This panoptic view may hinder the positioning and orienteering process, since people perceive their surroundings perspectively from their current position. In this thesis, I describe a novel framework that resolves this problem by applying sensors that gather the individual's spatial frame of reference. This spatial frame of reference, in combination with an *egocentric spatial data model* enables an injective mapping between the real world and the data's frame of reference, hence, alleviating the individual's cognitive workload. Furthermore, our *egocentric spatial data model* allows *intelligent mobile Geographic Information Systems* to capture the notions of *here* and *there*, and, consequently, provides

insight into the individual's surroundings. Finally, our framework, in conjunction with the context given by the task to be performed, enables *intelligent mobile Geographic Information Systems* to implicitly answer questions with respect to where, what, and how.

ACKNOWLEDGMENTS

This thesis would not have been possible without the support and guidance of many people. I take this opportunity to extend my sincere gratitude to all.

First, I gratefully acknowledge my advisor, Dr. Max Egenhofer, whose enthusiasm and guidance was crucial for the progress of my thesis. His generous cooperation and encouragement helped me accomplish the successful completion of this thesis.

I would like to thank my committee members, Dr. Kathleen Hornsby and Dr. Silvia Nittel, for their willingness to participate in many discussions, which helped me in clarifying many doubts.

Thanks to all my colleagues in the SIE department and NCGIA, especially Isolde Schlaisich, David Caduff, Konstantinos Nedas, Helen Duckham, Markus Wuersch, Matt Duckham and Lars Kulik for their many stimulating discussions. This research would not have been possible without their huge personal and academic support.

I would also like to thank my family and friends, whose encouragement and guidance enabled me to pursue my research at SIE.

The research was partially supported by the National Image and Mapping Agency under grant number NMA201-01-1-2003.

TABLE OF CONTENTS

Chapter

Chapter 1

INTRODUCTION

People navigate for very different reasons, whether they are utility workers, soldiers, or tourists. Many of the basic problems they encounter are similar, because much of the information people want or need cannot be directly interpreted from their surroundings. It is this problem of not being able to gather the right information from their surroundings that can lead people to getting lost. At any given intersection turning right might look just as adequate as turning left (Raubal and Worboys 1999); therefore, people who have never been to a place before and do not have a mental model of the environment use maps, which give them a panoptic view of the area. Maps are also useful as they provide information about a place that is not apparent to a person's senses. For example, looking at a building one might be able to tell that it is a church but not when it was built, or what the overall shape of the church is (Stea *et al.* 1996). It is these problems people have interacting with foreign environments that has led to this research of egocentric mobile geographic information systems (GISs).

To help people deal with the problems that arise from traveling, different types of navigational aids exist. These navigational aids have many different forms and some have existed for centuries. The goal of these aids is to improve people's wayfinding abilities and increase their cultural connection with a foreign space. The objective of this thesis is to develop a query model for spatially-aware egocentric GISs. This egocentric query model allows a location-based information system to accurately process users questions pertaining to their perception of *here*, and *there*.

1.1 Navigational Aids

International travel has become easier and more cost efficient over the last couple of decades (Taylor 2002). The Internet is also creating a more unified global economy, increasing the necessity for effective ways to aid travelers, both navigationally and culturally (Stea *et al.* 1996). There are currently many in-the-field GISs and navigation aids, including:

- **Guidebooks** one of the more comprehensive information providers of the in-the-field GISs. They contain maps, spatial, cultural and historical descriptions (Simonis 2002). Descriptions in written guidebooks are limited, as they typically supply only one or two routes between objects of interest. Guidebooks are useful for people spending a short period of time in an environment. They provide brief and concise descriptions that are stationary and cannot offer users in-depth information.
- **Signage** is one of the best ways to direct people to a location while minimizing their cognitive load. Signage works best if located at every decision point (Raubal *et al.* 1997). Signage also needs to be correct, easy-to-find, and easy-to-read. Not only does signage need to help people make decisions at different intersections, signs also need to encourage people between the decision points indicating that they are going the right way. Signage is usually only one-way so if a traveler gets lost it is difficult to retrace his or her steps. Since signage minimizes the traveler's

cognitive load it does not create mental models of a space, so it is difficult to get back on the right path once they become lost.

- **Paper Maps** supply a graphical panoptic view of the world. Paper maps offer a means to allow travelers to freely decide the best way to navigate (Woodward 1992). This panoptic view helps travelers to create a good mental model of an area quickly. Travelers, however, need a different map every time they require information at different scales, such as a whole city or a building. Travelers also require different maps every time they are interested in different thematic information, such as, attractions versus banks.
- **Immovable Maps** are attached to physical objects, such as subway stations and bus stations. Travelers cannot fold these maps and put them into their pockets, therefore, they need to look at these maps long enough to memorize the necessary information for traveling to their destination. An advantage these kinds of maps have over paper maps is the “you are here” information. These maps are useful where the travel routes are confined, for instance, in a transportation system.
- **Guided Tours** offer good culture information, but they are the most spatially constrained information outlets. This constraint is because guides decide where to go and how to get there. The cultural information a guide supplies is useful for wayfinding, because it helps travelers to remember landmarks.
- **Route Instructions** could be from a book or from a local expert on the street (Franklin 1996). Route instructions are useful, because they provide information that is egocentric, which is easy to understand for travelers. Like signage this

form of spatial information does not help the traveler to create a mental model of the space, so once lost it is difficult to get back on the correct path.

- **Spatial Exploration** allows travelers to create a cognitive map of an area through exploration excursions, which are usually performed in loops. This wayfinding behavior is a good way to create mental model of a space (Golledge 1999). As time passes, travelers will remember more about their local space and their loops will become bigger.

These descriptions of in-the-field GISs and navigation aids show the advantages and problems with current systems. One of the general goals of this work is to develop a GIS that incorporates the advantages of the different systems into one system while minimizing their problems. This classification of current GISs is used in the description of the map's evolution. Maps have evolved to incorporate more of the tasks proved by these systems. In Chapter 4 a classification is developed of the types of queries users perform in the field. It is this query operation classification, which was developed based on the functionality of these systems, which forms the core of egocentric queries.

Whichever category the navigational aid belongs to, in order to provide the appropriate information to the user it is necessary to understand the user's point-of-view. A synonym for a user's point-of-view is egocentric view. The term *egocentric* encompasses the information that is collected about the environment from the users' personal sensors. They see (Sholl 1996), hear, smell, and feel their surroundings. Once navigational aids have an insight into the user's egocentric view they can more easily offer specific information and geographic data, which allows the navigational aids to function as separate information appliances. The idea of information appliances (Norman

1999) is to have easy-to-use, dedicated devices. People have separate devices for blending food and drying hair, instead of one device for all functions in a household. Propagating this idea to the information technology domain, we use *intelligent mobile geographic information systems (GISs)* to refer to all egocentric mobile geospatial information appliances. Such GISs can provide geospatial information that is more *intelligent* (Minsky 1985) than paper maps by providing, for instance, route instructions and additional information about any thematic type in an environment to improve a user's mental model of a place (Timpf *et al.* 1992). For a system to intelligently adapt to a user's specific geospatial information requirements, *intelligent mobile GISs* require an insight into the user's egocentric view.

In order to understand the benefits intelligent mobile GISs have compared to other navigational aids, we will examine the history of one of the most popular and useful navigational aids, the map, focusing on how maps have changed over time. The evolution of the map shows that maps are sometimes difficult to use due to their inflexibility.

1.2 Evolution of the Map

Before we improve the concept of the map, we need to understand what a map is, what it provides, and how people use it. The history of the map reveals how it has changed dramatically throughout time. At the same time, however, it shows how the map has provided the basic functionality of giving people spatial information that is difficult to gather directly with their own senses.

The concept of the map has existed for thousands of years (Harley and Woodward 1991). From the beginning, maps in various forms have helped people remember events that happened at a certain time and place: for instance cave drawings can be considered maps (MacEachren 1995). For a map to be categorized as useful it needs to provide information that is consistent with reality and this information needs to be presented in a way that is easy for people to understand. Examining this map evolution we see that not only have maps become more flexible, they have also incorporated most of the functionality of other navigation aids.

From cave drawings to paper, maps have been both mirrors and catalysts of their times by reminding people what the human perspective of a geographic area used to be like, and providing an idea that there might be more area to discover (Harley and Woodward 1991). Medieval maps were historical, literary representations of Christian space, often centered on Jerusalem, with an introverted view of the classical world, describing unknown areas with words like, "Beyond here be dragons." Though Ptolemy introduced the idea of latitude and longitude in 150BC, this concept was not widely used as a basis for maps until the renaissance (Bergren and Jones 2000). The Ptolemaic grid created a single, global orientation of north-south-east-west and led to a major conceptual shift in ways of orienting and positioning geographic objects. A global scale was provided by this coordinate system, where Euclidian distances could be measured in real space and translated onto map space maintaining general spatial relations. This concept of maintaining a consistent scale throughout the map is one of the constraining parameters of what and how objects, or themes of objects, can be displayed on a map (Richardson 1992). For example, a map of the town of Orono probably does not show what furniture is in my office, whereas a more detailed map of the Target Technology Center building, or of room 214, would. This framework of scale, orientation, and theme had an immeasurable effect on people's perception of geographic space. In the next sections we show how improving these aspects of orientation, scale, and theme enhance maps by making them easier to use and appear more intelligent.

1.2.1 Paper Maps

Paper maps are one of the most popular map mediums and the oldest form of spatial information still in use. They provide bird-eye-views of a space showing large-scale relations. Some of the problems with paper maps are their fixed themes, orientations and scales. This makes it difficult for users to find their location on the map. Since paper maps have these fixed attributes there is a high cognitive load for users to calculate the relationship between the map's orientation and their own orientation (Kuipers 1982). The main difficulty with these framework attributes of the classical paper map is their inflexibility. A paper map has a single scale (Hudson 1992), which cannot be changed unless a new map is created. The user cannot zoom into a different scale to analyze objects in more detail. A paper map also has one orientation, for example, populations in the Northern Hemisphere are used to north being in-line with the top of the map. To navigate effectively, users have to align the map's orientation with the real world's orientation. Paper maps also have a fixed set of thematic information. A map of an area's road network probably will not display what soil types are found in that area. With the constraint of printing a map on paper comes the limitation that if a user wants information about soil types, he or she will need to print a different paper map. A user cannot re-center a paper map, which usually has its center aligned with the center of the mapped physical space (Couclelis 1992). For instance, a map of the State of Maine is centered on Brownville, which may make it difficult for a user in Portland to determine

1.2.3 Mobile GIS

A mobile GIS is a digital mapping system that functions on a mobile computing platform, such as a Pocket PC or Palm device (Chen and Kotz 2000). There is a difference between

the "you are here" location whereas a map centered on Portland showing some of New Hampshire and Massachusetts would facilitate this task. Many of the problems that paper maps are known to have are addressed with the development of digital maps.

1.2.2 Digital Maps

There have been many attempts to alleviate the problems paper maps are known to have. Digital maps have provided many solutions. Since digital maps incorporate more than just a map display (Kuhn 1991), the modern, encompassing term for these maps is GIS. GISs offer the functionality of overlaying different map thematic layers, which allows users to combine such spatial information as county boundaries, road networks, soil types, and vegetation to derive new information. With a GIS a user can also zoom between multiple scales, gathering different information about the same object. A GIS analyst can also easily rotate and translate the map's projection to different orientations and shift the map so that it is centered on different locations. This flexibility of alteration allows the map to seem more malleable. The problems that have arisen from GIS use are due to the system's complexity (Egenhofer and Kuhn 1998). GISs in general have complex human interfaces and their complex processing requires a substantial amount of computing resources (Worboys 1995). This need for powerful computing has made it difficult to bring GISs from the desktop into the field. Currently, GISs are mainly used in the PC desktop and workstation environment, where digital maps form the basis for the production of paper maps, which are printed and brought into the field. The problem with this situation is that a user must know all the information that will be required in the field beforehand.

2001). These systems are virtually desktop GISs that have been ported to handheld devices: the human-computer interaction framework has not been modified for the in-the-field tasks that users are likely to perform (Mackaness 2002).

1.2.4 Intelligent Mobile GIS

Travelers are more confident and relaxed when they know some background information and can understand why an environment is structured in a certain way (Stea *et al.* 1996). The more relaxed and "at home" people feel, the easier it is for them to function and accomplish their intended travel goals, such as attending a conference or enjoying Hawaii. By extending the basic functionality of a mapping system to allow it to gain some insights into a traveler's egocentric point-of-view we advocate *intelligent mobile GISs* that help travelers not only to navigate (Rodden *et al.* 1998), but also to gather background information regarding the context of their current locations (Schilt *et al.* 1994).

Based on familiar, non-digital travel aids discussed earlier in section 1.1, such as a "Lonely Planet Travel Guide," a set of criteria was developed for an enhanced *intelligent mobile GIS*. An *intelligent mobile GIS* needs to be diverse with respect to handling spatial and temporal constraints. As many travelers are uncomfortable exploring foreign spaces without first having some background details on a site, the *intelligent mobile GIS* must provide well-structured and well-presented background information (Rodden *et al.* 1998). This background information includes spatial details as well as historical and cultural information about an area. While bus and walking tours are typically highly structured, allowing little opportunity for additional exploration by the traveler, a criterion for the *intelligent mobile GIS* is that it must be spatially and temporally flexible. This flexibility will assist travelers in exploring a foreign space at their own pace, gathering information that pertains to their specific interest.

Ideally, therefore, an *intelligent mobile GIS* should provide Cicerone-like interaction. A Cicerone is the most user-centric guide. Cicerones are private tour guides that offer very effective user interaction based on face-to-face human contact (Schmidt *et al.* 1999). An *intelligent mobile GIS* will take advantage of the traveler's location, orientation, pointing gestures, as well as indirectly-gathered information about the user's interests (Rodden *et al.* 1998). Being aware of such attributes allows an *intelligent mobile GIS* to more intelligently guide the traveler, both, spatially and temporally.

In the near future maps will seem more intelligent because of a translation between the user's egocentric view of the world and the panoptic view provided by the mapping system (Egenhofer 1999). This translation will allow the mapping system to more effectively support a user's needs. These *intelligent mobile GISs* will have the ability to automatically adapt to the necessary scale based on the user's needs. Such a GIS can be aligned with the user's view of reality as well as inform the user of the global orientation. Such maps will also be user-centric and the scale and themes will adapt to the user's tasks.

Users in the field have a very constrained perception of the environment around them. Their perspective point-of-view provides them with an egocentric understanding of the local geographic space. Current mapping systems' lack of an egocentric translation makes it difficult for users to interact with the information systems. *Intelligent mobile GISs* overcome this limitation by sensing when and how to translate between the user's on-the-ground egocentric view and the bird-eye-view. The core of the translation process is the spatial awareness based on seven degrees of freedom: three location parameters (x, y, z), three orientation parameters (pitch, roll, yaw), and time (Narayanan 2001).

A mapping system that is aware of the user's spatial context (Dix *et al.* 2000) will have the data necessary to decide what kind, amount, and format of information will help the user the most. For example, knowing the user's location and orientation at all times allows the system to derive the user's speed. Linking this information to data about the environment provides an insight into user's mode of travel (Majumdar *et al.* 2003). If the user's location is on train tracks moving for a prolonged period of time at a certain speed, the system can infer that the user is most likely traveling by train. Knowing the time of day and weather conditions could be used to decide on how to provide information to the user (e.g., as text, in graphical form, or by sound). An egocentric mapping system has the benefit of linking the spatial information around the user's local environment to cultural, non-spatial attribute information.

Making mobile maps egocentric alleviates some of the problems inherent in their limited computing functionality. Users will have the perception that they are not directly interacting with the computing device, because the mapping system will be able to derive what the user wants from spatial contextual information and interact with the user the way another human would (Schmidt *et al.* 1999).

1.3 Research Questions

To improve the map and provide people with useful information about their surroundings we investigate ubiquitous and context-aware computing and show how these systems improve mobile GISs. This thesis incorporates the *egocentric spatial data model* into context-aware computing to answer questions users might ask in the field, such as "Where am I?" or "What is that?" To return meaningful results for these user queries we address the following challenging research questions in this thesis:

- How would sensors need to be incorporated into a mobile device in order to become aware of the user's spatial context and what kinds of sensors are necessary?
- How would a database management system use spatial awareness from a client device to process the terms *here* and *there* in order to provide information that is in tune with the user's environmental context?

To process these questions, an information system needs models, and methods that can consider the user's location and orientation (Lukowicz *et al.* 2002), as well as the spatial and temporal aspects of their surroundings. In order to improve interaction with a spatial information system, the system needs to provide information that is intuitive for the user. The concept of intuitive is relative to the user's spatial, temporal, and cultural context. This thesis develops a data model that processes the users' spatial and temporal context, thereby providing information that is more aligned to their needs.

1.4 Goal and Hypothesis

There are currently many limitations within traditional geospatial information tools. The goal of this thesis is to create a seamless integration of the user's sensed position and orientation into a data model for mobile GISs. This goal is accomplished by identifying the role of spatially-aware sensors within mobile information devices, as well as developing a data model that has an insight into the user's spatial information needs. This data model incorporates the egocentric spatial abstract data type, which has methods to process such egocentric spatial concepts as *here* and *there*. Consequently, the hypothesis of this thesis is:

> *Measurements from position and orientation sensors are sufficient to formulate executable spatial queries about "here" and "there."*

1.5 Approach

Egocentric mobile spatial information systems have two components: first, a spatially-aware client device, and second, a data model that can process egocentric insights into the user's perspective view of the world. We investigate spatially-aware mobile client devices showing how spatially-aware sensors can improve the map and better aid travelers, examining the functionality they offer and the form of user-interaction they have (Baus and Kray 2002).

In order for *intelligent mobile GISs* to have a similar usability as paper maps, mobile computing needs to be just as ubiquitous as paper (Abowd and Mynatt 2000). An examination of ubiquitous computing is performed showing spatial-awareness as a core property of many context-aware computing systems (Burnett *et al.* 2001). Context awareness is also shown as a necessity for a system to appear ubiquitous (Jiang and Steenkiste 2002). This idea is also justified with the existence of other spatially-aware information systems that showcase the use of context-awareness to ubiquitously provide information to the user.

An *intelligent mobile GIS* affords many more functional properties than other spatial information systems used in the field. We develop two forms of spatially-aware systems: (1) a map system and (2) a pointing system. The map system is spatially-aware, which allows it to automatically pan, zoom, and rotate the map based on the user's needs. The pointing technology uses spatial-awareness and functions like a computer mouse, but instead of selecting objects on a computer screen to receive information users are selecting objects by pointing at them in their real world surroundings.

Upon the completion of an investigation of these spatially-aware client devices we examine a second aspect of *intelligent mobile GISs*, which is an *egocentric spatial data model*. Such a data model has two components: first, a mechanism to represent the egocentric attributes, such as position and orientation and second, methods of the abstract data type to process the concepts of *here* and *there*. Abstract data types are developed to represent the user's egocentric attributes of position and orientation and show how the standard database query language can be used for egocentric queries. An execution model is then developed for egocentric spatial queries with query re-write rules demonstrating the algorithms behind the queries' functionalities. This term query re-write means the process of transforming user queries into executable statements.

1.6 Scope of Thesis

The setting for this thesis is a system that is aware of the user's position and orientation. The data are sensed at high precision. All user queries are based on a typical travel scenario, so they are constrained to information within a single city. Other scales and granularities are not considered. The *egocentric spatial data model* works well even when it is not constrained, though current sensor accuracy does not allow us to test the model in more general situations at a global level. Another assumption is that the system has knowledge about objects within the environment, not only their location, but also non-spatial attribute information.

There is no human subject testing perform for the prototype. The prototype developed for this thesis should not be considered a fully functional commercial *intelligent mobile GIS* application. We are only interested in providing spatially-aware map displays. Though we describe the functionality and specifications of an egocentric mobile query system, its implementation is considered future work.

Other attributes that are discussed, but are not incorporated into our system at this time, are the use of distance and spatial data representations displayed in both, two-dimensional and three-dimensional space.

1.7 Assumptions

Many assumptions are made upon which the *egocentric spatial data model* is based. These assumptions are because in order to develop a robust data model, certain database management system traits must exist. They include:

- **Object-Relational DBMS:** An object-relational DBMS is a relational database system that has been extended to support object-orientated principles. This assumption is necessary because the egocentric data model is developed as a data object.
- **Open DBMS:** A database is considered open if developers can create their own abstract data types. This abstract data type creation is done to enhance the inherent modeling power of databases beyond built-in types such as integer and character.

- **Spatial Data Support:** The egocentric data type processes spatial data; therefore, in order for a database to incorporate the egocentric data type it needs to support the spatial geometry data type.
- **Object Inheritance:** In object-orientated design a subclass inherits properties from a super class. To ensure that the egocentric data type supports spatial data it should inherit properties from the spatial data type.

1.8 Results

Based on the framework for egocentric query operations some innovative examples of spatially-aware prototypes are described. These descriptions show how current geospatial information systems can be enhanced to more effectively improve user's knowledge and interaction with foreign environments. The geospatial information systems are enhanced by applying knowledge about the user's location and orientation.

An *egocentric spatial data model* is also developed integrating the concepts of *here* and *there* into a database management system. This integration is achieved by obtaining position and orientation data and translating these data into actual database queries. This data model is necessary for an *intelligent mobile GIS* to function as an effective information system providing information that is aligned with the egocentric view of its user.

1.9 Intended Audience

This thesis is intended for researchers and developers interested in the design of future geospatial information appliances, in particular mobile spatial information systems. Such an audience includes researchers concerned with location-based services, ubiquitous computing, and intelligent spatial appliances, as well as computer scientists investigating how mobile computing affects human-computer and human-environment interaction in the spatial sciences. The developed prototype environment provides a test bed that may be of interest to scientists who want to perform experimental human subject tests or experiments on how people navigate and gather information pertaining to a foreign environment.

1.10 Organization of Thesis

The remainder of this thesis is organized as follows:

Chapter 2 examines current spatially-aware systems where spatial-awareness is a subset of context-awareness. I evaluate mobile mapping systems, and augmented virtual reality maps. Next, some background information is provided regarding spatial database systems, showing how the standard query language is extended for spatial data. This background information is provided to show how we expand on the existing spatial data model to obtain the egocentric spatial data model that is developed in the thesis.

Chapter 3 showcases two forms of spatially-aware systems: a map and a pointer. The map improves user interaction, because the system can automatically pan, zoom, and rotate based on the user's needs. The pointer allows users to point at an object in their surrounding environment and receive information about it via sound, text, or video. The pointer performs the same query-by-selection as a computer mouse does within current desktop GISs, but instead of interacting with the screen, users are interacting with the real world.

Chapter 4 develops abstract data types to store the egocentric attributes of the user's spatial reference frame. These abstract data types can be used within standard query languages to process egocentric queries. The Chapter also develops such egocentric queries.

Chapter 5 uses the abstract data type developed in Chapter 4 for re-writing the egocentric spatial queries pertaining to the user's concept of here and there. Such query re-writes are necessary to show functionality behind the egocentric query process.

Chapter 6 concludes this thesis. It discusses the results of the research, as well as its implications considering the use of context to retrieve spatial information. This chapter also provides a description of future research activities that could build on the results of this thesis. The outlook focuses on conceptual enhancements of spatially-aware query systems, possible extensions of the prototype implementation, and research activities for which the current prototype application can serve as a test bed. The thesis closes with a portrayal of integrated *intelligent mobile GISs* that include location and orientation as the fundament for other contextual information.

Chapter 2

MOBILE TRAVEL GUIDE INFORMATION SYSTEMS

Starting in the 1990s (Norman 1990) scientists realized that the way people interact with computers is unnatural and inefficient. The problems people are having with computers are similar to the problems they have with maps. To use paper maps people have to put themselves mentally into the map's frame of reference. Research in the ubiquitous computing domain (Weiser 1993) has shown that one way of making computers easier to use is by providing the computer system with some awareness of the user's context (Schilt *et al.* 1994). Context awareness allows the system to have insights into the user's frame of reference and can provide information that is aligned with this frame (Kuipers 1982); therefore, the user's interaction with the computer system seems more intuitive. This idea of context awareness is useful in the spatial information domain. If a map is aware of its user's spatial context then the map can be centered on and oriented around the user, thereby reducing the spatial cognitive load for people (Schilit and Theimer 1994).

In 1991, Mark Weiser introduced the idea that the arcane aura that surrounds personal computers is not just a "user interface" problem, it is that the idea of a "personal" computer itself is misplaced (Weiser 1991). To fix this problem he suggested that computing become more distributed and ubiquitous. A goal of ubiquitous computing is to make users less aware of computing devices so that the information the computer supplies is easier for users to understand. In order for this to happen the computer devices need to disappear from the user's consciousness. This conscious disappearance is a necessary fundamental consequence of human psychology, not of technology. Whenever people learn something sufficiently well, they cease to be aware of it (Norman 1999).

In order for a computer system to be ubiquitous it needs to dynamically adapt to the environment (Harter *et al.* 1999). A computer can intelligently adapt if it is aware of some contextual information about its usage, information about the user, the user's environment, and time (Rodden *et al.* 1998). The necessity of this contextual information shows how context-aware computing was conceived out of ubiquitous and mobile computing. Spatial awareness is a fundamental component of context-awareness (Dix *et al.* 2000), for example, in order for a personal digital assistant to become aware that its user is making coffee, it would need to know that its user is in front of the coffee machine and maybe that it is morning. This example shows that in order to understand why and how some event is occurring one first needs to know when and where this event is happening.

To understand how a map can become spatially-aware, an investigation into other spatially-aware mapping systems is conducted, showing the tasks they perform, the manner in which they collect and use spatial data, and how users interact with these systems. After an examination of related spatially-aware geographic information projects is performed an investigation of augmented reality system is presented. This augmented reality investigation shows how these systems exploit many similar technologies that spatially-aware mapping systems have, thereby showing that the *egocentric spatial data* *model* developed in this research is useful for augmented systems as well. This chapter concludes with some background information about the standard query language and how database management systems can be extended with abstract data types. A description of current spatial and temporal database system is given, which is necessary because the egocentric datatype inherits properties from these current database extensions. This backdrop information is also useful to gain an understanding of the *egocentric spatial data model* that is developed in Chapters 4 and 5.

2.1 Spatially-Aware GISs

Spatially-aware GISs encompass more than the standard idea of GIS. Travel aids should also be considered GIS because they possess mapping and wayfinding assistance as well as connecting spatial data with attribute data. Digital travel guides have provided some of the most innovative concepts for spatial awareness. To provide spatial information for wayfinding processes and increase users' interactions with the environment, Spatially-aware GISs use sensor-enriched mobile computing devices. These devices do not just provide information about a spatial entity, but also connect that information to other background information, such as the entity's history or its physical properties. These GISs need to collect and process relevant information in extremely variable environments.

The ground-breaking Lancaster GUIDE enables visitors to flexibly explore and learn about a city in their own ways (Cheverst *et al.* 2000). The system is capable of acting as an intelligent tour guide or as a richly featured guidebook, depending on the visitor's needs. GUIDE also allows visitors to control their pace of interacting with the system and the environment. At every popular travel attraction an 802.11 base station is located for wireless data transmission. The spatial information in this situation is very coarse, for example, at the Lancaster Castle it might be possible to tell on which side the user is. Since the sensed positional data, which is based on wireless triangulation, is imprecise, the system's feedback must also be at a coarse granularity. The GUIDE has to provide coarse information because the system cannot assume a precise contextual relationship between the user and geographic space. GUIDE can provide information about a large area, like the north side of the Lancaster Castle.

The GUIDE prototype uses Tablet computers, which allows for varied forms of user interaction. The system acts like a spatially driven web browser where users are presented with text, pictures, sound, and maps. Users are also informed of other people using the system nearby so they could send messages to each other about their experiences.

The Outdoor Cyberguide (Abowd *et al.* 1997) functions as a digital guidebook on a mobile PDA, not an augmented reality system. It provides text and maps for users to interact with, but does not provide pictures or sound. It also allows users to send instant messages to other current tourists. The Cyberguide system is unable to change the underlying maps automatically.

The Deep Map Framework (Zipf and Aras 2002) includes a web-interface for pre-trip planning, language-driven mobile prototypes on a portable computer as well as location-based services for smartphones. Its services are intelligent position determination, the dynamic generation of individualized proposals for sightseeing tours and adaptive maps. Deep Map is not an augmentmented reality environment as the main human interface is a Pocket PC device. Instead, users read 2D maps, background text, as well as look at a 3D model of the environment with augment information. This system is voice-driven and the presentation information is based on spatial proximity.

The Tourism Information Provider (Hinze and Voisard 2003) delivers various types of information to mobile devices based on location, time, profile of end users, and their "history". The TIP notification system is composed of mobile devices and a server. The server is comprised of three thematic databases; profile, scheduled event, and spatial. The TIP system is a nice example for showcasing most of the information necessary for an intelligent mobile GIS. A problem that the TIP has that is addressed by our *egocentric spatial data model* is its data management is dispersed where interoperability problems arise. The *egocentric spatial data model* incorporates these separate databases into a single instance of a database management system. Placing the profile, scheduled, history, and spatial data into a single instance allow the system to be autonomous, consistent, and durable.

Another type of spatial information systems are augmented reality systems. These systems overlay spatial features within a user perspective view of their surroundings with attribute information. Though augmented reality systems supply a different form of user interaction than mobile GISs due many of the functionality is similar. Users of augmented system still need spatially-aware information and an ability of egocentrically query about their surroundings

2.2 Augmented Reality Systems

Augmented reality systems use spatially-aware sensors to help users by augmenting the user's view of reality with geospatial information. The "map-in-the-hat" (Thomas *et al.*

1998) enables users to see waypoints and compass headings for walking directions through a head-mounted display. The focus of the "map-in-the-hat" is to guide users in an orienteering task from one point to another. In the "map-in-the-hat" system, users see attributes, but they are not presented with a map to read.

Battuta (Nusser *et al.* 2003) seeks to enable access and use of digital geospatial information for field data gatherers who do not have extensive training in spatial analysis. The Battuta prototype is a wearable system that includes a digital compass and a GPS receiver. The augmented data is displays through a viewer clipped to a regular pair of glasses. The augmented vision has a see-through map plus a locator for the users to read while performing their data gathering tasks.

The Touring Machine (Feiner *et al.* 1997) is a 3D mobile augmented reality system for exploring the urban environment. The prototype assists users by overlaying information about items of interest in their vicinity. As a user moves about, he or she is tracked through a combination of GPS positional and magnetometer/inclinometer orientation sensors. Information is presented and manipulated on a combination of a see-through, headworn 3D display, and an untracked, opaque, handheld 2D display with stylus and trackpad. This system allows the user to continually see a building in reality, read a label of the building, read a 2D map, and read background text.

The next section examines the standard relational and object-relational database systems, and how database management systems have been extended for spatial and temporal data. This analysis is provided as background information for the development of the *egocentric spatial data model*, which is a database extension that inherits properties from the spatial and temporal data extensions.

2.3 Using and Extending Database Management Systems

Database query languages are tools to facilitate the access to a database and have been investigated in computer science for many decades (McFadden and Hoffer 1985). We use the term *query* for a statement requesting the retrieval of data from a database. For the description of queries the Structured Query Language (SQL) is used (Chamberlin and Boyce 1974). SQL is the standard relational query language and enjoys popularity in traditional database applications.

2.3.1 Structured Query Language

The fundamental structure of SQL is the SELECT-FROM-WHERE block. The SELECT clause determines the attributes to display, the FROM clause describes the data sets needed to solve the query and the optional WHERE clause specifies constraints upon the items to be retrieved (Ramakrishnan and Gehrke 2000). Here is an example of a common SQL query:

```
> SELECT  population
> FROM    town
> WHERE   name = "Orono";
```

2.3.2 Spatial Data Queries

Spatial SQL commands (Egenhofer 1994) for the selection of spatial data improve upon the fundamental SQL block by adding support for spatial predicates, such as topological features and object geometries for points, lines, and regions (Guting 1994). An example of a Spatial SQL query is:

```
> SELECT  lake.name
> FROM    lake, city
> WHERE   lake.geometry north_of city.geometry
> AND     city.name = "Orono";
```

In this example lake is the data type, the term *north_of* is the spatial predicate, and Orono is the reference item. This query uses the abstract data type geometries, which incorporates spatial relations, such as inside, covered by, overlaps, equals, contains, and covers.

2.3.3 Abstract Data Types

In order to serve as support for non-built-in types such as; spatial, temporal, and egocentric data the relational data model has been extended. Such data types and their operations are referred to as Abstract Data Types (ADTs) (Guting 1994). ADTs were introduced as a way to circumvent the lack of modeling power inherent in the basic relational database model (Stonebraker 1986). Within extended relational systems, users manipulate values through the use of reference queries whose types are basic, such as integer or real types, but also abstract data types accessible through the operations defined on them as methods.

The term 'abstract' is applied to these data type because the database system does not need to know how an ADT's data is represented nor how the ADT's methods work. It merely needs to know what methods are available and the input and output types for the methods. Hiding of ADT internals is called encapsulation. In an object-relational system, the simplification due to encapsulation is critical because it hides any substantive distinctions between data types and allows object-relation database management systems to be implemented without anticipating the types and methods that users might want to add.

2.3.4 The Spatial Data Type

The term spatial data is used in a broad sense, covering multidimensional points, lines, rectangles, polygons, cubes, and other geometric objects (Guting 1994). A spatial data object occupies a certain region of space called its spatial extent, which is characterized by its location and boundary. From the point of view of the database management system spatial data is classified as either point or region data. Queries that arise over spatial data are of three main types: spatial range queries, nearest neighbor queries, and spatial join queries.

2.3.5 The Temporal Data Type

Time is an important aspect of all real-world phenomena. Events occur at specific points in time; objects and the relationships among objects exist over time. The ability to model this temporal dimension of the real world and to respond within time constraints to changes in the real world as well as to application-dependent operations is essential to many computer applications (Snodgrass 1995).

Discrete interpretation of time has commonly been adapted by the research community in temporal databases because of the simplicity and relative ease of implementation (Tansel *et al.* 1993). Hence, time will be interpreted as a set of equally spaced and ordered time points and denote it by *T*. $T = \{0,1,2,..now...\}$. Any point beyond new is future time. Any interval or temporal element that includes the special constant *now* expands as the value of *now* advances.

The *egocentric spatial data model* inherits properties from both the spatial and temporal database management systems in order to process the terms *here* and *there.* In order for a database to process *here* and *there* queries it needs to be aware that these queries are geospatial and occurring *now*. For a database to allow queries where a user is cognitively immersed in the spatial data environment the *egocentric spatial data model* is necessary as shown in Chapters 4 and 5.

2.4 Summary

This Chapter examined related projects, comprised of ubiquitous and spatially-aware computing. The descriptions showed that there are systems that use sensors such as GPS as well as orientation sensors. The goal of this examination was to see if a map can be improved by giving it an insight into the user's needs and which form he or she wants the information to be presented. A key aspect missing from the projects described in this chapter is a robust query environment of mobile, distributed, and egocentric information systems. It is because of this lack of a query environment that the second half of this Chapter provides some background about query languages and other common data models. This is necessary before the development of the *egocentric spatial data model* in Chapters 4 and 5, which show how the spatial data query language should be extended for egocentric spatial queries.

Chapter 3

CLIENT DEVICES FOR INTELLIGENT MOBILE GISs

As a starting point in designing *intelligent mobile GISs* we consider two forms of egocentric spatial information systems: a map and a pointer. The first system we investigate is a mobile mapping system, whose advantage over other maps is the ability to automatically adjust panning, zooming, and rotation based on the user's real world movements. The second system is based on the concept of pointing. With this system a user can point at a geographic object and receive some form of feedback about the object. The metaphor for the pointer is to function like a magic wand (Egenhofer and Kuhn 1998). To receive information the users point at an object in their surrounding and receive information in a format of their choice. The description of these systems focuses on the advantages they have over current technologies and describes their functionalities.

3.1 Spatially-Aware Map Technology

Maps still provide the main means for understanding spatial environments, as well as for performing tasks such as wayfinding, trip-planning, and location-tracking. Static traditional maps have several disadvantages, such as:

- **Fixed orientation:** that is the map always faces in one direction (typically north). The map users, however, may be facing any direction at any given moment. Hence, in order to understand the map users need to perform some kind of rotation, either of themselves or of the map to align their frame of reference with the map's frame of reference. This process puts an immense cognitive load on the users, because it is not always intuitive and may present considerable difficulties, especially in cases of complex, uniform or unfamiliar spatial environments.

- **Fixed scale:** that is the map cannot be changed to a different granularity level. This limitation is one of the most restrictive aspects of paper maps. The scale determines the level of zooming into a spatial environment, as well as the level of detail and the type of information that is displayed on a map. Users, however, need to constantly change between different scales, depending on whether they want a detailed view of their immediate surrounding environment or a more extensive and abstract view in order to plan a trip or find a destination. Current solutions to the problem include tourist guides that comprise maps of a specific area at many different scales. Tourist guides, however, are bulky books, difficult to carry around, and search time is considerable as they typically consist of hundreds of pages.
- **Inability to represent a changing world:** that is all spatial environments and the objects that they encompass, whether artificial or natural, are displayed statically although they are actually dynamic and change over time. Artificial spatial objects, such as buildings, may get created, destroyed, or extended, while others, such as land parcels, may merge, shrink, or change character (e.g., when a rural area gets developed). The same holds true for natural entities, for instance, a river may expand or shrink because of a flood. The static 2-dimensional map is

restricted to representing a snapshot in time and the information on it may soon become obsolete.

- **Limited display of thematic information:** that is an inability to show many varying thematic information at the same time. There are many different types of maps, such as morphological, political, technical, tourist-oriented, and contour maps. The thematic content of a static map has to be defined a-*priori* and is usually restricted to one area of interest. Even then, the information displayed is minimal. For example, a tourist map will indicate that a building is a church or a restaurant, but it is highly unlikely that more information will be available, such as the construction date of the church or the menu of the restaurant and the type of cuisine it offers.

An egocentric *intelligent mobile GIS* may be defined as a GIS that uses sensors to gather information about the users reference frame and adapts the visualization of the map data accordingly. In abstract terms, this adaptation is defined as a translation of the map reference frame to the absolute reference frame based on the users personal reference frame. The relationship between these three spatial reference frames is illustrated in Figure 3.1.

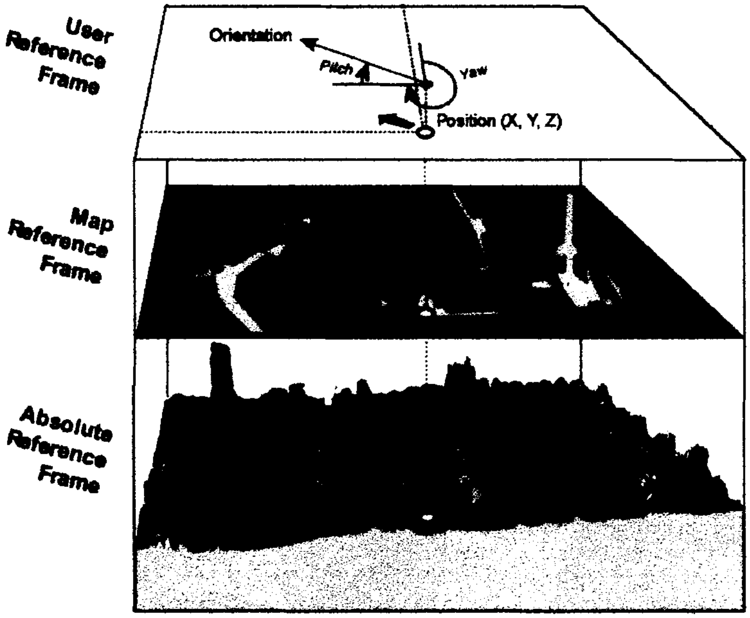

Figure 3.1: Relationship between the real world, map, and user's orientation

Sensing the user's position and orientation allows the map to rotate in accordance with the user's frame of reference. As a result, the system provides what we call *adaptive orientation*. The north-orientation convention can be dropped. Furthermore, the system provides intelligent zooming and panning: The user's location is at all times displayed through a dot on the screen of the mobile computer and the system pans through the map as the user change his or her location by moving in the spatial environments. Zooming is also very easily performed and can be done manually if desired, or automatically, provided that the system has some information about the context of the user (e.g., entities of interest, motion speed, and other parameters).

Since it supports thematic contexts this system is beneficial to any type of user, from map professionals such as surveyors and GIS analysts to technicians and tourists. For instance, when the user is a tourist, the display provides the relevant information and type of map that would serve this type of user best, such as churches and museums. For a sewer technician, however, it would provide a map of the sewer system for the city and the relevant technical information. All the supplied information depends on the underlying geospatial and thematic dataset that is being loaded on the system.

Figure 3.2 shows a prototype interface for such an egocentric mobile map. The sensor output on the right side of the display shows the GPS coordinates and orientation data provided by the sensing hardware. The left side showcases the thematic data types, such as roads or buildings. The map display has a dot highlighting the user's current location and an arrow showing the direction he or she is facing. In this example, the user is interested in buildings and is located on the central mall of the University of Maine facing in the direction of Fogler Library. The bottom panel of the interface shows some attribute information pertaining to the library. In the bottom right panel the user is given the choice of listening to the attribute feedback. This system works with standard geospatial data types comprised of both raster and vector data.

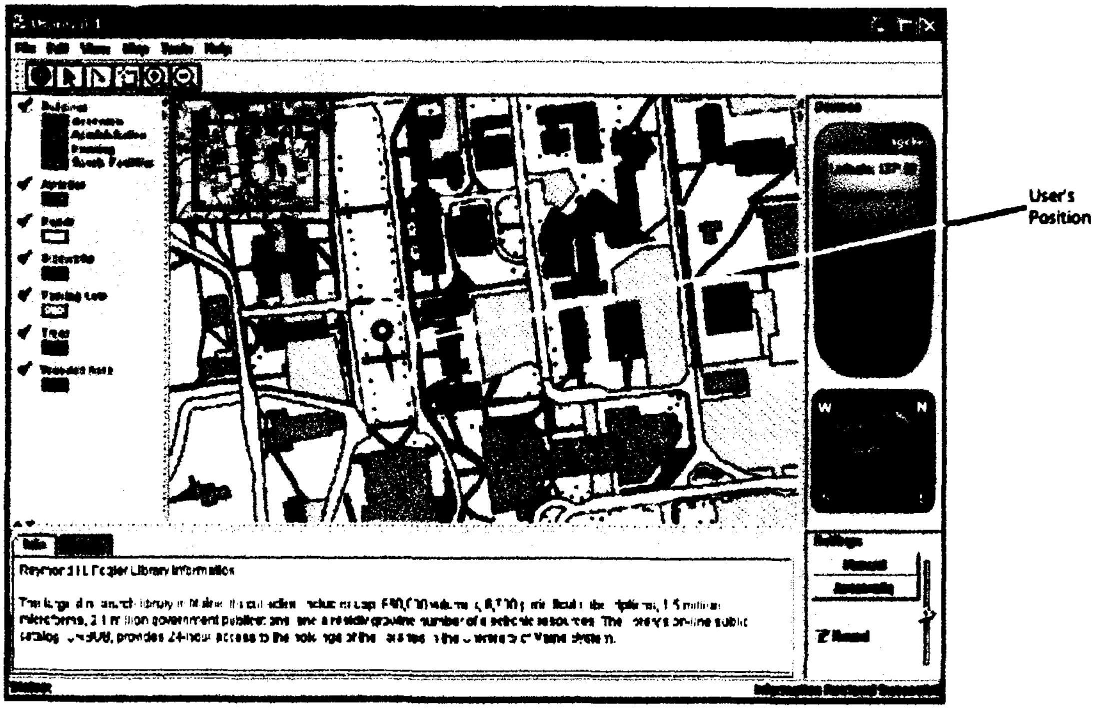

Figure 3.2: An egocentric mobile map running on a Tablet PC.

Such mobile map technology is useful for people who are used to working with maps. Another form of interaction that might be useful for people less experienced in using maps is a pointing device.

3.2 Spatially-Aware Pointing Technology

A spatially-aware pointing technology will allow users to point at an object or location and learn about it (Figure 3.3). The pointer provides information about x, y and z (or latitude, longitude, and elevation) and pitch, roll, and yaw (angle up from the horizon, angle side to side from the horizon, and compass angle). The sensed positional and orientation information is integrated into a database management system. This database management system contains geospatial data. The user's position and orientation are used to create polygons representing the egocentric concept of *here* and *there*. These *here* and *there* polygons are then checked against the geospatial data in the database management system and a list of best-case candidate objects (e.g., building) are identified. The user can then request additional information about the identified objects (e.g., name, date built or history).

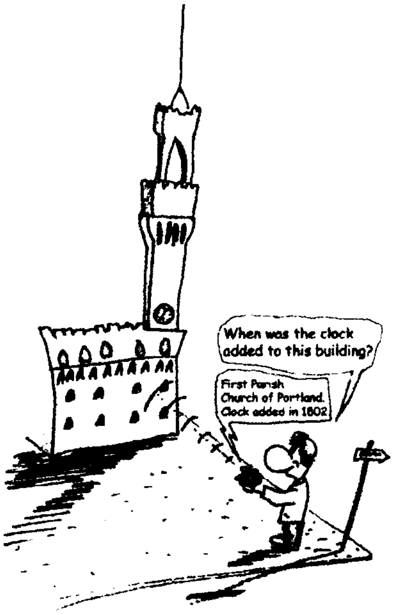

Figure 3.3: Representation of pointer use.

A key challenge for the pointing technology is to optimize the spatial information retrieval. The pointer needs to match the real-time measurements of location and orientation with the best candidate object in a geospatial dataset. Integrating location-orientation data with a digital landscape model and developing a plausible computational model that targets granularity is key to the success of the pointer. The pointing technology will exploit the use of orientation sensors so that geospatial datasets are not only user-centered, but also egocentrically oriented. This aspect is relevant to the pointing technology, since otherwise no distinction could be made between such cognitive aspects as *in front* and *behind.*

The pointing technology consists of two parts: the physical pointer to sense the user's egocentric spatial reference (Figure 3.4a) and a mobile computation platform, such as a PDA, to process the sensed data, calculate within a digital landscape model the element to which the user pointed, and generate an auditory or textual response to the user (Figure 3.4b). The pointer will house an orientation sensor unit that captures *yaw* and *pitch*; a GPS unit to determine location (x, y, and z); a Bluetooth transceiver for wireless transmission of the sensed data to the PDA within a near range; and a battery.

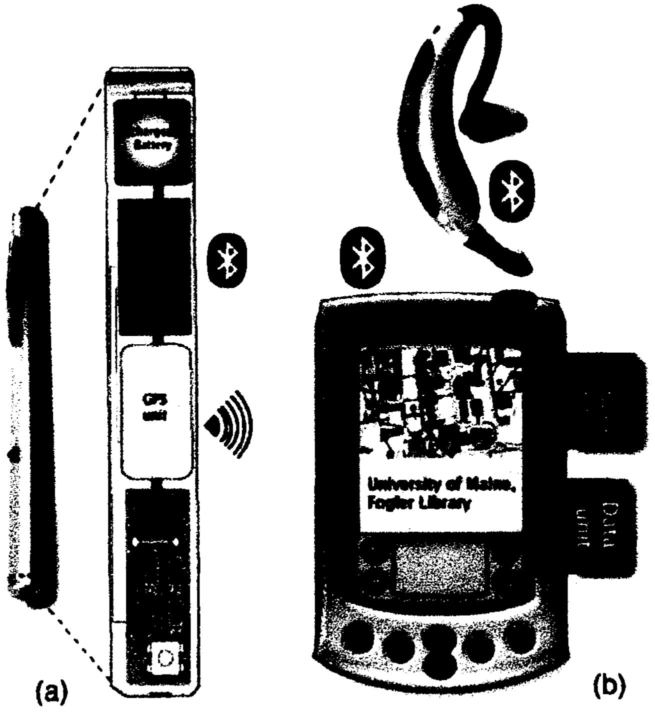

Figure 3.4: (a) Orientation sensor, GPS, Bluetooth, and battery, and (b) PDA with pointer software, data, Bluetooth.

3.3 Summary

Two mobile spatially-aware client technologies are demonstrated in this chapter. The first is a mapping technology that provides automated panning, zooming, and rotating functionality based on the user's real world spatial context. These automated attributes alleviate the user from having to put themselves cognitively into map space. The second technology is a pointer that allows users to point and select objects in their surroundings. Once an object is selected the pointer can provide information about it. The two mobile technologies define the requirements for mobile spatial data processing:

- sensor-driven
- distributed, and
- egocentric

The two spatially-aware mobile clients demonstrated in this Chapter are only the first half of a mobile spatial data processing system. Chapters 4 and 5 develop the *egocentric spatial data model*, which provides a database management system with an insight into the reference frame of the user. This data model extends spatial data management with functionality of incorporating the user's position and orientation to process the concept of *here* and *there* based on sensed data collected from spatially-aware clients like the ones described in this chapter.

Chapter 4

EGOCENTRIC SPATIAL QUEREIS

It is believed that over 80% of all useful data have a spatial component (Maitra and Andersen 2003) that can be linked to a GIS. If data are not inherently spatial, they most likely have a geographic footprint linking the data to space and time (Beard and Sharma 1997). For example, objects referred by most nouns (persons, places, or things) exist at some place at a certain time. Goodchild (1998) defines a *geo-library* as a "library filled with georeferenced information," which is based upon the notion that information can have a geographic footprint. Within such a setting, georeferenced information is broad in scope to include such things as photographs, videos, music, and literature that can be given a locational variable. Most of these spatial data are stored in relational databases that are organized in a panoptic or "birds eye view" format (Raper 2002). This panoptic view format can create difficulties for humans, who have a perspective view that makes their outlook of the world very narrow; therefore, spatial database systems must have a way to translate between the user's egocentric and the data's allocentric reference frame.

With today's technology, users of information systems extract egocentric views of data through queries. Spatial databases provide query results in a panoptic view, which is useful for GISs. In order for the data to be translated to the users' point of view, however, it is necessary for the users to provide their egocentric references. For example, "I am at the corner of 84th Street and 2nd Avenue facing north. Which direction should I turn to get to the Empire State Building?" This chapter develops a spatial data model that implicitly exploits spatial sensors gathering the user's egocentric spatial reference frame, so that the user does not have to explicitly input it. Such a query model is needed for advanced location-based services.

The description of the *egocentric spatial data model* is broken down into three parts: (1) the syntax of the egocentric abstract data type and its use within SQL, (2) the semantics of what the model does and, (3) the execution model of how the system process the egocentric queries.

4.1 Syntax of An Egocentric Spatial Data Model

A data model for egocentric queries is comprised of two components: a data representation structure and data manipulation operations. For example, Codd's (1970) relational model uses tables to represent the data and the data manipulation is based on mathematical foundations expressed by two formal languages; relation calculus and algebra. In a spatial data model for egocentric queries the spatial data objects are the user's time-of-query position and orientation. Before a description can be given of how the user's position and orientation are used to create the egocentric abstract data type, we examine how and why the relational data model has been extended using abstract data types to account for such complex mobile queries by structuring data as objects.

4.1.1 Spatial Data Types For Mobile Queries

In order for a database to process egocentric spatial queries, the database needs to support a data type containing the user's egocentric spatial reference frame. This data type provides the basis for modeling *here* and *there*. Providing a data model with an insight into *here* and *there* is essential for egocentric queries, because in order for the data management system to process the queries the system needs to be able to relate the user's and the data's reference frames. The term egocentric queries is used for location-based mobile services, in which the user's spatial reference frame information adds value to the service as a whole (Frost and Sullivan 2003). This term refers to mobile situations where much of the informational needs of users relate to their surroundings (Golledge 1999). The user's position and orientation are used to create the egocentric ADT, as described in the next section.

4.1.2 The Egocentric Spatial Abstract Data Types

The egocentric spatial ADT is used in the same way as other ADTs. For example, when a town data object is represented in a data base management system this data object could have attributes; name of type variable character, population of type integer, and geo of type geometry as shown in this SQL block:

```
> Create        Table town(
> name          varchar(30),
> pop           int(30),
> year-inc      int(4),
> geo           geometry);
```

The egocentric ADT is very similar to geometry ADT. In fact the egocentric ADT inherits properties from the geometry ADT. This inheritence allows the egocentric ADT to be topologically related to the geometry ADT. For instance, a traveler data object could have attributes; name of type variable character, and address of type variable character. The difference between the town's data object and the traveler's egocentric data object is town has the attribute geo of type geometry traveler has attribute ego of type egocentric. As shown in this SQL block:

```
> Create  Table traveler(
> name    varchar(30),
> address varchar(45),
> DoB     datetime,
> ego     egocentric);
```

There are three relational tables necessary for the egocentric spatial data model: (1), the `egocentric` table represents the user's time-of-query position and orientation, (2) the `userSESSION` table represents data regarding the sensor's and data's accuracy for the querying session, and (3) the `userCONTEXT` table represents data about the user's querying traits such as arm used to point (Figure 4.1). These tables are created before hand by the ADT developers. Most of the data are supplied automatically and implicitly by the sensors and metadata about the geospatial dataset, though some attribute for example, the user's contextual profile are added by the system's users explicitly.

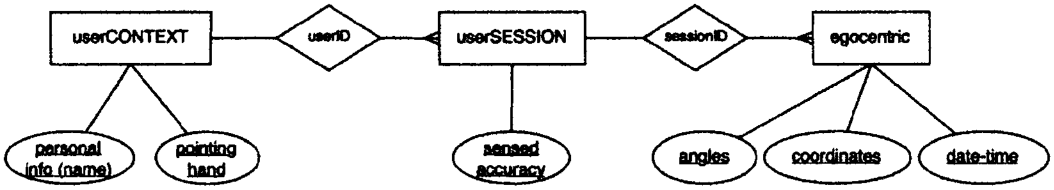

Figure 4.1: ER diagram of the relational tables composing the egocentric ADT.

The `egocentric` table models the user's current geographic position, orientation, and time. In most cases, *here* will be based on stable GPS or cellular triangulation readings. The `egocentric` table also accepts orientation angles as input, which represents the user's current orientation. In many instances the determination of *there* will be based on the system receiving stable angles from magnetometer, gyroscope and accelerometer based sensors. An example of the `egocentric` table is shown in the following SQL block:

```
>  CREATE TABLE egocentric(
>    24, -- ego_id NUMBER PRIMARY KEY,
>    1, -- session_id NUMBER FOREIGN KEY,
>    45.5432156,  -- x-coordinate
>    68.5443433,  -- y-coordinate
>    13.1674934,  -- z-coordinate
>    89.528,  -- yaw angle in degrees
>    21.367,  -- pitch angle in degrees
>    04-27-2003 10:39:52.32, -- date-time
> );
```

The concept of *there* is a method of the `egocentric` ADT to calculate an object the user is selecting at a distance away from *here*. Since a system cannot calculate what a user means by *there* if it does not know what the user means by *here*, the method of *there* uses many of the same aspects of the `egocentric` ADT as the *here* method. Both *here* and *there* utilize the position coordinates x, y, and z. In addition, *there* uses the compass angle (or yaw) and in many cases the vertical angle (or pitch) in order to decide in which direction and what angel up from the horizon the geospatial objects are that the user is interested.

An important issue to consider when determining the best candidate to satisfy the *here* and *there* queries is the context and validity of the attributes that define the egocentric reference frame. For instance, an ontology-based approach (Fonseca *et al.* 2002) may be used to decide the granularity of the query results based on the accuracy and precision of the sensed spatial attributes, as well as the geographic context of the query. This ontology-based system is considered future work for extending the `userSESSION` table, which represents accuracy of the sensors and data. The data of the `userSESSION` table can be used chose the best granularity level of an egocentric query. For example, a hiker in the Swiss Alps might point at a village and ask, "What is that?" If this hiker is using a consumer grade GPS and orientation sensor the hiker should get a different response at a different granularity level than a soldier using military grade GPS and orientation sensors. For example, the consumers might learn that they pointed at the village Degen, whereas the soldier might learn that he or she pointed at the St. Sebastian church in Degen. With consumer grade sensors it might be more difficult to discern what a user is pointing at, so the response must be at a coarser granularity. Though sensed position, orientation, precision, and accuracy should not be considered the only contextual data used to decide the informational needs of the user, these attributes are valuable as they can be gathered implicitly.

The `userSESSION` table is used to provide context for a user's query session. It is also necessary to add validity to the sensed data within the `egocentric` table. The `userSESSION` table represents the stored position, orientation, and temporal accuracy information about the query session. The `userSESSION` table is updated when some querying aspect changes, for example, when the user decides to use information about his or her elevation from one of many different sources. In some cases it is probably better to utilize the user's elevation value from the digital terrain model, or objects on the model like a building. In other cases it might be best to use the elevation sensed by the GPS, for example if the user is not on the ground. One user can have many query sessions, because it is possible for him or her to use different query configurations. The next SQL block shows the kind of data in the `userSESSION` table:

```
> CREATE TABLE userSESSION(
> 1, -- session_id NUMBER PRIMARY KEY,
> 2, -- user_id NUMBER FOREIGN KEY,
> 30, -- x y positional accuracy in feet
> 40, -- z positional accuracy in feet
> 5, -- orientation accuracy in angle degrees
> 5, -- temporal accuracy in seconds
> 1, -- use sensed z Boolean
>);
```

The third table necessary for the *egocentric spatial data model* is the `userCONTEXT` table, which represents the stored data regarding the user's query context, for example, the hand used for query-by-pointing. Knowing which hand the user points with is necessary to compensate between the angle the user sees they are pointing and the angle the pointing device is actually pointing. Whereas the `userSESSION` table represents contextual data about the sensed spatial attributes in the `egocentric` table, the `userCONTEXT` table provides personal contextual information about the user. In the following SQL block we create the attribute `pointing_hand`. This stores personal information about how the user performs the query-by-pointing tasks, which is necessary for accurate results.

The `userCONTEXT` table could be linked to many `userSESSION` tables and each `userSESSION` table could be linked to many `egocentric` tables. Shown is an example of the SQL block to create the `userCONTEXT` table:

```
> CREATE TABLE userCONTEXT (
>  user_id NUMBER PRIMARY KEY,
```

```
>  name VARCHAR2(32),
>  pointing_hand VARCHAR2(10)
> );
```

In the future the *egocentric spatial data model* should be extended with an ontological driven filter. This ontological filter would use the user's preferences to choose the semantic information results. For example, the userCONTEXT table could store information about native language spoken, education level, and other interests.

The egocentric ADT is reference differently than spatial geomentry data objects are. Reference queries in GIS can be classified into three categories (Rigaux *et al.* 2002); (1) queries with alphanumeric criteria, (2) queries with a spatial criterion (i.e., an operation that applies to the spatial part of one or several geographic objects), and (3) interactive queries, which require participation from the user (e.g., to select a particular area by drawing a selection on a display device using a mouse). In the following examples we examine special cases of interactive spatial queries where, instead of users interacting with a computer display, users interact with the environment via their position and orientation. We refer to these types of queries as *egocentric queries*.

To access and manipulate the *egocentric spatial data model* a framework of query operations must be defined. The next section performs an analysis of high-level query operations formalizing queries as either *here* or *there*. This analysis lays the groundwork for the query execution model described as re-write rules in Chapter 5.

4.1.3 Egocentric Query Operations

In order to develop the query operations using the egocentric ADT, we first describe a framework to classify the types of egocentric spatial queries. This framework is created in a top-down design methodology, starting with the types of questions a user in-the-field might ask. An analysis of the high-level query operations develops an egocentric query classification. Each class has two variants, one query without thematic classification type, the other with the thematic type, such as *buildings*, specified. The two query operation classes are *here*, and *there* (Table 4.1).

Spatial Query	**Spatial Operation**
Where am I?	*Here* spatial operations
In what "x" am I?	*Here+* spatial operations
What is that?	*There* spatial operations
What "x" is that?	*There+* spatial operations

Table 4.1: Egocentric spatial query operations.

4.1.3.1 *Here* Operations

The spatial operation *here* functions with the spatial attributes centered on the position of the user's egocentric coordinate system. For an egocentric *here* operation the fundamental SQL block is based on an interactive query-by-position situation. A query-by-position situation would be, "Where am I?" This query uses coordinates from a position sensor and provides information about geographic features relating to the user's position. For example, a useful result would be, "You are in Room 336 of Boardman Hall." This is different from a common desktop GIS, where within the query-by-selection situation the user's primary task is to interact with geospatial information. In the desktop situation the user selects a feature via a computer mouse that generates a query resulting in thematic information about the object. A *here* query occurs in the field, where the user's primary task is to interact with the environment and the interaction with a GIS is a secondary task. The user may not want to directly interact with the computing device via keyboard, mouse, or trackpad. An example of a query-by-position SQL block for the "Where am I?" question is:

```
> SELECT  *.name
> FROM    *
> WHERE   *.geo contains traveler.here;
```

In this example the user might be using a consumer grade GPS with a known accuracy of 30 feet. This means that the average theme object size needs to be greater than 30 feet, for example a building. This query is described in greater detail in the section on *here* queries re-write in Chapter 5. This process of transforming a user query into an executable statement is called query re-writing

4.1.3.2 *Here+* Operations

In *Here+* operations the user specifies the theme type he or she is interested in, for example, a town. This would change a query-by-position question from "Where am I?" to, "What town am I in?" The *here+* query classification re-write is examined in Chapter 5. An example of a *here+* query-by-position SQL block is:

```
> SELECT  town.name
> FROM    town
> WHERE   town.geo overlaps traveler.here;
```

4.1.3.3 *There* Operations

Many egocentric *there* operations are based on an interactive query-by-pointing situation. Similar to the *here* aspect of the `egocentric` ADT, the *there* implements a pointing based spatial query-by-selection, which is an extension of the position based query-by-selection. For example, a basic query-by-selection might be "What object is that?" where *this object* is selected by clicking on it via a computer mouse on a digital map in a desktop-based Graphical User Interface (GUI). In an egocentric query-by-pointing the users are interacting with their surroundings instead of a computer GUI. An example of a query-by-pointing type question would by "What is that?" where *that* is a feature the user is pointing at. An instance of a *there* query-by-pointing SQL block is:

```
> SELECT  *.name
> FROM    *
> WHERE   *.geo overlaps travler.there;
```

Considerations for egocentric *there* operations are the query distance and result presentation format. In most instances it will probably be appropriate to sort the query results based on their distance from the point of query (Silberschatz *et al.* 1999). An example of a sorted question might be, "What is the closest feature this way?" where features are sorted by distance from *here*. An instance of a distance SQL block is:

```
> SELECT      *.name
> Order_BY    distance(*.geo, travler.here)
> FROM        *
> WHERE       *.geo overlaps traveler.there;
```

These query results can also be sorted by other attributes that are of interest to the user, such as size.

Another factor to consider is the extent of the query distance. The way the model is set up now it is theoretically possible for users to ask what buildings are in front of them and receive all the buildings along a geodetic great circle, where the last building listed would be the one right behind them. Though this probably would not happen because a database containing all the buildings on the planet does not exist. What is most likely to happen is for the query distance to be the extent of the viewable dataset. The extent of query distance, however, is based on context (Dey and Abowd 1999), such as the users' position, whether they are on a mountain peak or at the bottom of a valley. Another distance context is based on theme type, that is, whether the user asks about buildings versus countries. Many users will probably not be interested in objects they cannot see. The visibility distance based on the earth's curvature is about 30 miles, which also might be the boundary of the maximum query distance.

For *there* object selection operations the user's positions is used as a starting point. From this point we form a geometric vector based on the yaw or northing angle and the pitch angle of the user's orientation. Error propagation is used to convert this vector into a cone, which is used within an algorithm (terrain intersect model) that utilizes three-dimensional data of the environment. This algorithm determines what objects the geometric cone intersects with (e.g., building geometry). The identified object's data are then augmented with additional information (e.g., name, date built or history). The object's identification and the augmented information are what the *there* query of the `egocentric` provides as output.

4.1.3.4 *There+* Operations

There+ operations, like *here+* operations, are situations when the user specifies the thematic type he or she is interested in, for example, a mountain. This changes a query-by-pointing question from "What is this? to "What mountain is this?" An example of a *there+* query-by-pointing SQL block is:

```
> SELECT  mountain.name
> FROM    mountain
> WHERE   mountain.geo overlaps traveler.there;
```

This section described the syntax of the egocentric spatial ADT. Showing an entity-relationship diagram of the data model and how the ADT is created and used within SQL In the next section a description is given of the semantic of what the egocentric ADT is doing.

4.2 Semantics of An Egocentric Spatial Data Model

The semantics of the egocentric spatial data model concerns the conditions in which the system can be said to be true. Semantics relates to the meanings of the words *here* and *there.*

4.2.1 *Here* Semantics

The question, "Where am I?" is what the egocentric ADT's *here* method is designed to process. Its meaning is, "In what region is the GPS-sensed position?" where the region is the oval polygon and the GPS-sensed position is the X shown in Figure 4.2

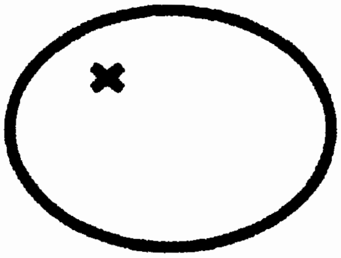

Figure 4.2: *Here* semantics.

4.2.2 *Here+* Semantics

The egocentric ADT *here+* method represents the query "In what Y am I?" Its meaning is, "In what region of type Y is the GPS-sensed position?" where the region is the oval polygon of type Y and the GPS-sensed position is the X as shown in Figure 4.3 The GPS-sensed location X is contained within multiple polygons but only one of type Y, therefore, this scenario can be answered uniquely. Although there are two polygons of type Y, only one contains the GPS-sensed position X.

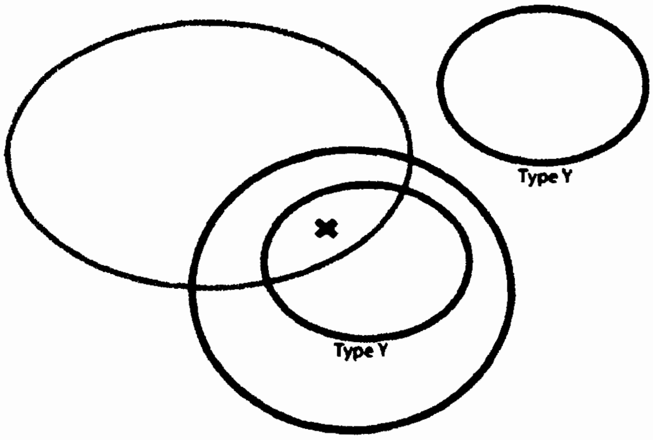

Figure 4.3: *Here+* semantics.

4.2.3 *There* Semantics

The egocentric ADT *there* method represents the query "What is that?" Its meaning is "With what region intersects the ray that originates *here* and points in the sensed direction?", where the region is the oval polygon, the GPS-sensed position is the X, and the sensed direction is the arrow starting from the GPS-sensed position X shown in Figure 4.4

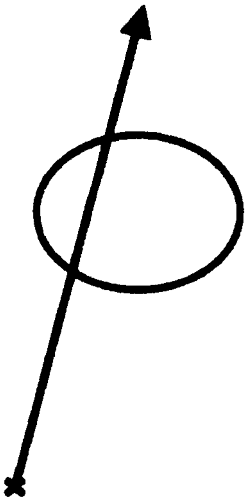

Figure 4.4: *There* semantics.

4.2.4 *There+* Semantics

The egocentric ADT *there+* method represents the query "What Y is that?" Its meaning is "With what region of type Y intersects the ray that originates *here* and points in the sensed direction?", where the region is the oval polygon, the GPS-sensed position is the X, and the sensed direction is the arrow starting from the GPS-sensed position X shown in Figure 4.5.

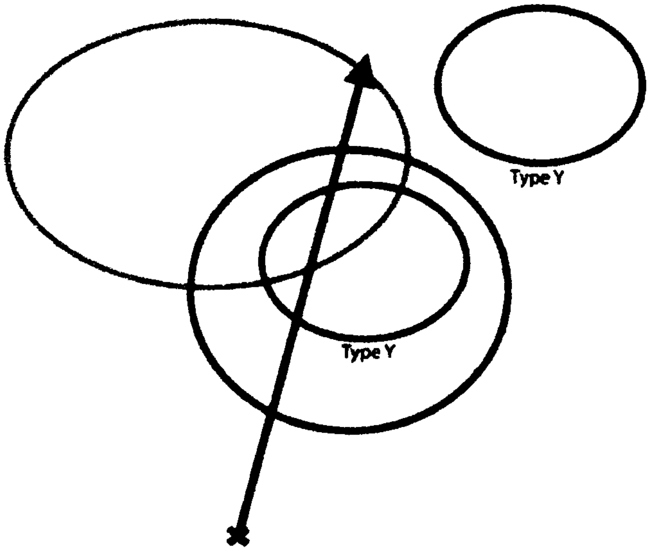

Figure 4.5: *There+* semantic.

The sensed direction intersects multiple polygons, but only one of type Y. Although there are two polygons of type Y, only one polygon intersects the sensed orientation.

These semantic descriptions show unique results of only one candidate for containment or intersection, but what happens if more than one candidate fulfills the geometric constraint? The next section examines the organization of spatial subdivisions. The examination is preformed because the egocentric spatial operations function very differently depending on how the spatial features of interest are organized, because each subdivision structure has its own semantics when more than one candidate fulfills the geometric constraint.

4.3 Egocentric Spatial Subdivisions

In developing these *here*, and *there* spatial operations, a framework for spatial subdivisions is examined (Florence 1997). This examination is necessary because the separate spatial operations function differently in different space configurations when more than one candidate fulfill the geometric constraint. *Intelligent mobile GISs* need to have considerations for the following spatial subdivisions:

- **Partitions:** Partitions are subdivisions of space that consist of cells in the most general case, where any two distinct cells do not have a common interior. When spatial features are organized in a partitioned manner the user should be in only one spatial feature at time. For spatial queries that occur in partitioned space consideration needs to be made for situations where the user's position is located on some of the spatial feature's boundaries. As shown in Figure 4.1, in partitioned space the spatial features do not overlap, instead their boundaries meet, creating at least one shared edge. An example of space that is partitioned is states. As shown by point P, which is on the boundary of three features in Figure 4.1, the difficulty with an egocentric query would be to decide which feature to provide to the user. For example, the user asks, "Where am I?", the system could answer, "You are on the border between Wyoming, Nebraska, and Colorado." In the next Chapter describing the execution model this problem is treated showing an approach to deciding the best candidate feature based on degree of overlap.

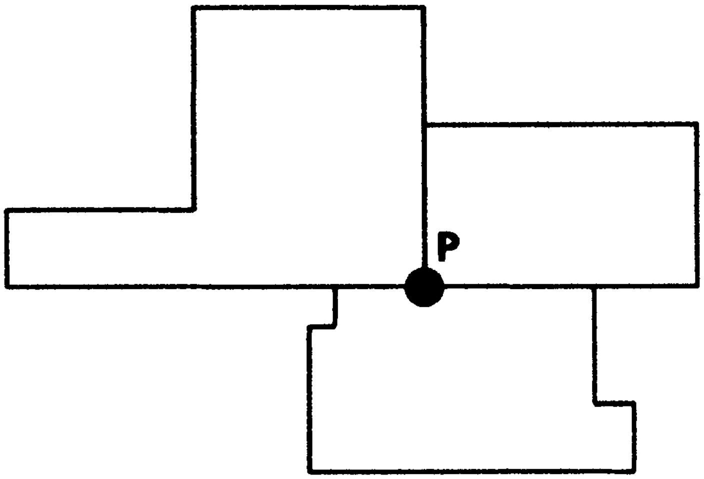

Figure 4.6: Egocentric queries in partitioned space.

- **Overlapping Space:** Spatial features can be organized in an overlapped manner so that the boundaries of the two objects coincide in two points, both boundaries run through the opposite interior, and both interiors share some commonality (Egenhofer 1994). In an egocentric spatial query this means that the user could be in many different spatial features at the same time, as shown by point N in Figure 4.7. An example of overlapped features is wireless communication like radio and mobile phones. One of the difficulties for spatial queries in overlapped space is finding the most likely spatial feature the user is located in. This could be done based on relational attributes, for example, how close the user is to the center of the spatial features.

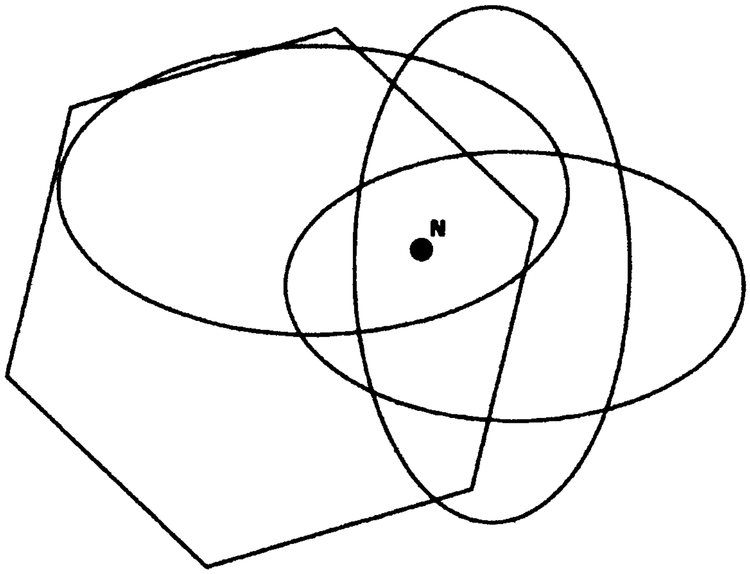

Figure 4.7: Egocentric queries in overlapping space.

- **Hierarchies:** Spatial hierarchies occur when one object category is created through the subdivision of another (Florence 1997). Space that is organized hierarchically means that a super feature fully contains some sub-features, which in turn fully contain other sub-features, for instance, a building that is in a town, in a county, which is in a state. As shown by point H in Figure 4.8, the difficulty with queries in this spatial organization is how to decide on the resulting level of detail. For example, if a user submits query such as, "Where am I?", the system needs to decide which granularity level to provide as a response. Informing users that they are in the State of Maine, when they are interested in knowing in what building they are, would lead to frustration and the feeling of low usability.

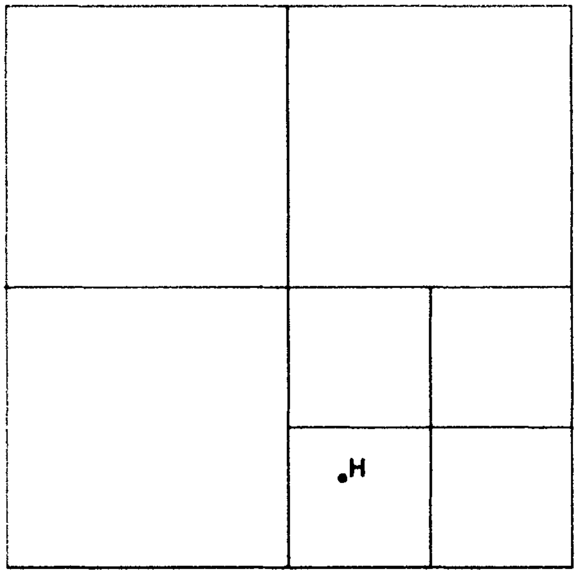

Figure 4.8: Egocentric queries in hierarchical space.

- **Disjoint Space:** Spatial features that do not share any edges are disjoint from each other. As shown by point D in Figure 4.9, users may not be located in any of the disjoint objects. This situation would exist when users specifies a feature classification which they are not located within or have no other relation to, as for example, if the users asks "What building am I in?" and the system discovers that the user in not located in any building. One solution to this problem is a *nearest neighbor* test to check for the closest feature to the user's position. The added value of performing a *nearest neighbor* test is the system is able to respond with suggestive answers, such as, "You are not in a building, but the nearest building is Little Hall." One of the more popular *nearest neighbor* tests was developed by Roussopoulos (1995), who proposed a branch-and-bound algorithm that searches the R-tree in a depth-first manner.

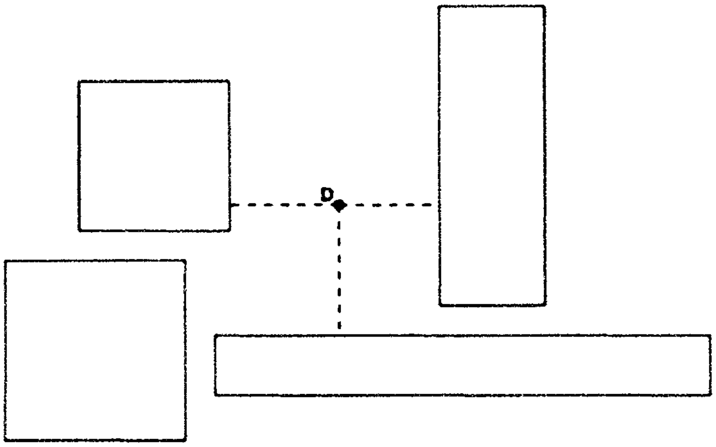

Figure 4.9: Egocentric queries in disjoint space.

Though the description of egocentric query operations does show a seamless integration of the user's perspective into a database query language for mobile GISs, what is missing are rules to formulate executable spatial queries. Chapter 5 describes these query re-write rules which allow spatial SQL commands the execute using the terms "here" and "there"

4.4 Summary

This chapter described the data model for egocentric spatial queries. The first section portrays why and how ADTs are used to extend the standard relational data model. This description of ADTs was followed by the development of the three egocentric spatial ADTs necessary for egocentric spatial queries: `egocentric`, `userSESSION`, and `userCONTEXT`. Next the query operations using these egocentric ADTs are described. There are two types of egocentric spatial queries: *here*, and *there*. Each of these query types has two variations: first, when the user does not specify the thematic data classification and second when the user does provide the thematic classification. The four resulting egocentric query operations are *here*, *here+*, *there*, and *there+*. These query operations provide different results depending on how the spatial features are structured. An analysis of spatial subdivisions is provided to show how egocentric spatial queries operate in the different spatial organization schemes. The query operations are high-level relationships between the egocentric reference and the surrounding's spatial data reference frame. The next chapter provides re-write rules for the query operations. These re-write rules alleviate the burden of users having to understand the process of egocentric spatial queries directly.

CHAPTER 5

EGOCENTRIC SPATIAL QUERE RE-WRITE RULES

The egocentric ADTs that incorporate attributes relating to the user's spatial reference frame were introduced in Chapter 4, which allows for different types of *here* and *there* queries. While the *egocentric ADTs* support *here* and *there* queries at a high conceptual level in an extended SQL syntax, it is necessary to transfer such user queries into executable statements. For example, the keyword "here" needs to trigger a measurement from a position sensor (e.g., a GPS receiver) such that the observed x- and y- coordinates get integrated into the query. Likewise, the sensor's measurement accuracy must be considered in order to generate a reasonable answer. This process of transforming a user query into an executable statement is called query re-writing. Using this approach the queries can be processed against different types of spatial data sets and the query's response is at a reasonable level of granularity.

Query re-writing is necessary in order to take the burden of knowing details about how to process the actual queries away from the user. It avoids having the user explicitly dealing with the sensors accuracies and minimizes otherwise cumbersome dialogues between user and system to determine desired levels of detail. The re-write does not completely eliminate the dialogue between user and system, however. For example, if a user asks the system "Where am I?" and the system responds "on the Earth," the user would become annoyed. The goal of this chapter is to develop a mechanism that allows the system to provide responses at the most detailed level that can be accurately discerned from the sensors.

5.1 Re-Writing *Here* Queries

Here queries are based on the GPS sensor's observed x- and y-coordinates. A simplistic approach would perform a point-in-polygon algorithm to determine all regions that contain that point. This approach would, however, ignore that the coordinates may be included in several—overlapping or hierarchically organized—regions (Section 4.5). Reporting all containing regions would confuse the user, particularly if the regions are coarse and, therefore, obvious to the user. An example of such a coarse response is, "You are within the United States." The approach also does not incorporate the influence of the sensor's inaccuracies, which may lead to blunders when performing the point-in-polygon test when the observed location is close to a region's fringes. The following equations derive better answers, considering that in addition to the observed x- and y-coordinates, the sensor's accuracy is used for choosing an appropriate granularity for the query response.

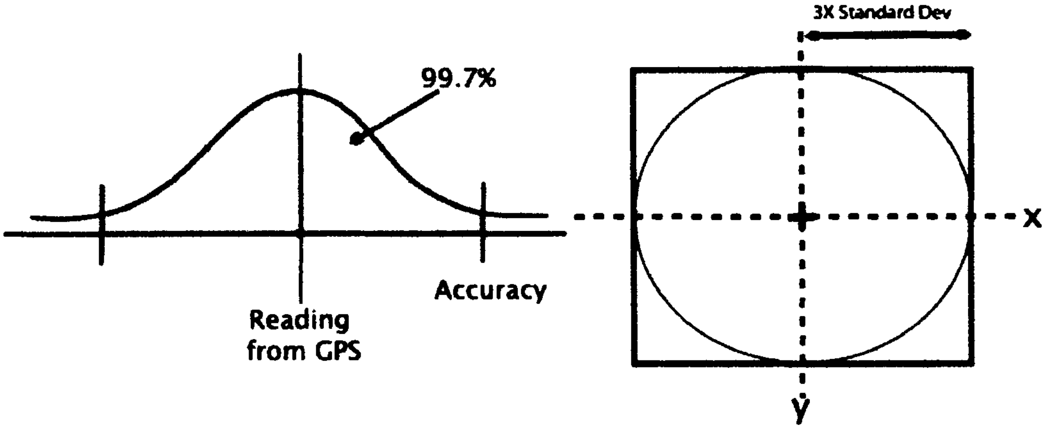

Figure 5.1: Standard deviation of position sensor and the *here* query window.

The GPS sensor's accuracy metadata (typically published by the manufacturer based on a series of calibration measurements) are usually expressed as a standard deviation (*sd*). The Normal Distribution empirical rule in statistics states that 99.7% of all measurements are within a range of three standard deviations (Figure 5.1). In this model for processing *here* queries a square query window (*QW*) is formed, with the observed *x0*-*y0*-coordinate pair at the center and a side length of six times the sensor's accuracy (Equation 5.1). The *QW* side length is six times larger, because the standard deviation can be thought of as a radius around the point. Since most spatial queries are based on window queries, a minimum-bounding rectangle is placed around this circle.

$$QW[x0, y0, sd] := \begin{pmatrix} x\_bottom\_left := x0 - 3 \times sd \\ y\_bottom\_left := y0 - 3 \times sd \\ x\_top\_right := x0 + 3 \times sd \\ y\_top\_right := y0 + 3 \times sd \end{pmatrix} \tag{5.1}$$

All entities that have something in common with this query window are then candidates for the query result, whereas entities that are outside the query window are not candidates. The constraint in the SQL WHERE clause is re-written to:

WHERE QW {inside, coveredBy, overlaps, equal, contains, covers} *.geometry

The second step is to sort the candidates such that the most reasonable response is returned to the user. This sorting is established based on the degree of overlap (*OD*) between the query window and the candidate's geometry in the form of a ratio between the common area and the window's area (Equation 5.2).

$$OD[*.name] := \frac{(area(*.geometry) \cap area(QW))}{area(*.geometry)} \tag{5.2}$$

Depending on the topological relation with the query window, different *OD* ranges will be obtained (Equations 5.3 a-f).

$$QW contain\ *.geometry : OD[*.name] > 1 \tag{5.3a}$$

$$QW\ \mathrm{cov}eredBy\ *.geometry : OD[*.name] > 1 \tag{5.3b}$$

$$QW equal\ *.geometry : OD[*.name] = 1 \tag{5.3c}$$

$$QW inside\ *.geometry : 0 < OD[*.name] < 1 \tag{5.3d}$$

$$QW\ \mathrm{cov}ers\ *.geometry : 0 < OD[*.name] < 1 \tag{5.3e}$$

$$QW overlaps\ *.geometry : 0 < OD[*.name] < 1 \tag{5.3f}$$

The degree of overlap is now a measure for best fit, with 1 being the ideal value. This value is of course a hypothetical target as in most settings the closest candidate regions will have *OD*s that deviate more or less from this value. Two strategies can be pursued to identify the best candidate (and a list of candidates sorted in decreasing order).

- Sort the candidates by the deviation of the overlap degree (*ODD*) from the target value (Equation 5.4).

$$ODD[*.name] := abs(OD[*.name] - 1) \tag{5.4}$$

Since the two ranges of *OD* values ($0 < OD < 1$ and $OD > 1$) differ and since it is likely that very large objects exist that contain the query window, this measure favors candidates in the query window, while penalizing large candidate objects.

- Normalize the two *OD* ranges by the smallest (*ODmin*) and largest (*ODmax*) values and calculate the deviation of the normalized overlap degree (*ODND*) from the target value (Equations 5.5 a and b).

$$if OD[*.name] > 1 then ODND[*.name] := abs(1 - OD[*.name]/OD\mathrm{min}) \tag{5.5a}$$

$$if OD[*.name] < 1 then ODND[*.name] := abs(1 + OD[*.name]/OD\mathrm{min}) \tag{5.5b}$$

The best candidate is then the object with the smallest *ODD* or *ODND* value. Subsequent selections of the next-best response can be made from both lists (e.g., when the user is interested in additional responses). While the *ODD* list offers browsing at coarser or more detailed granularities, the *ODND* list offers integrated navigation (i.e., "next best").

For example, if a GPS has a standard deviation of five meters, then the *QW* area will be $900m^2$. Imagine for reasons of simplicity that the Town of Orono has an area of

90,000m^2 and the State of Maine has an area of 900,000m^2 and that the QW is completely contained in both. Then the degrees of overlap are:

Town of Orono 900/ 90,000 = 0.01

State of Maine 900/ 900,000 = 0.001

Sorting these candidates by the deviation of the overlap degree ODD from the target value gives the State of Maine a value of 0.099 and the Town of Orono 0.999. A normalization of these candidates based on equations 5.5a and b shows that the Town of Orono is the "best fit." The reason we are interested in the region that is closest in size to *here* query window is because it is most likely that the user is interested in the spatial object the is at the granularity level that can be accurately discerned.

5.2 Re-Writing *Here+* Queries

Here+ queries are conducted when the user specifies the thematic classification of interest, such as buildings or roads, which results in queries like, "What *building* am I in?" A *here+* query uses many of the same algorithms as *here* queries, except that with the thematic classification of interest known the system is aware of the granularity level that the user is interested in. Here+ queries must deal with size of object versus observation accuracy. If the object is too small, then some response about incompatible granularity must be generated. For *here+* queries the query window QW is created the same way as for *here* queries. Since *here+* are similar to the *here* queries the point-in-polygon algorithm can also be used. There are many of the same challenges with this approach as with *here* queries, as the QW may be included in several—overlapping or hierarchically organized—regions. Once a sorted descending list of best-fit regions is created, the regions are tested to determine whether they are of the theme type the user is interested in. The QW is tested to see if it has something in common with a building region, for example, in the SQL WHERE clause, re-written as:

WHERE QW\{inside, coveredBy, overlaps, equal, contains, covers\} building.geometry

This WHERE clause for *here+* queries is different from the one for *here* queries because in *here+* building.geometry is used instead of *.geometry. The *.geometry checks for all geometries that have something in common with the QW where as the building.geometry only checks for building geometries. From this point on the scenario is very similar to *here* queries except for the use of thematic classes instead of all geometries.

5.3 Re-Writing *There* Queries

There queries are based on a position sensor's observed x- and y-coordinates, as well as an orientation sensor's observed pitch and yaw angles. As with *here* queries, a simplistic approach would perform a point-in-polygon algorithm, similar to the one described for *here* queries, to determine all regions that intersect in some way with the query window. Before this algorithm can be performed, however, the query window QW needs to be developed. In *here* queries the QW was a buffer around the person's location based on the standard deviation of the position sensor. For *there* queries users are interested in an area away from their current location in a direction they select. An orientation sensor determines the direction to this area in which the user is interested. This sensed direction has a standard deviation specified by the hardware manufacturer. Based on the *here* QW

and sensed orientation direction plus the sensors standard deviation the *there QW* is created. Where the user's position *O* and positional standard deviation (sdP), as well as the direction of interest and orientational standard deviation (sdO), it is possible to create an error propagation area, this area it the *there QW*, shown in the shaded area as part of (Figure 5.2). The *there QW* is broken into smaller polygons based on predefined distances from the user. For example area (a, b, c, d) could be within the first 50 meters from the user.

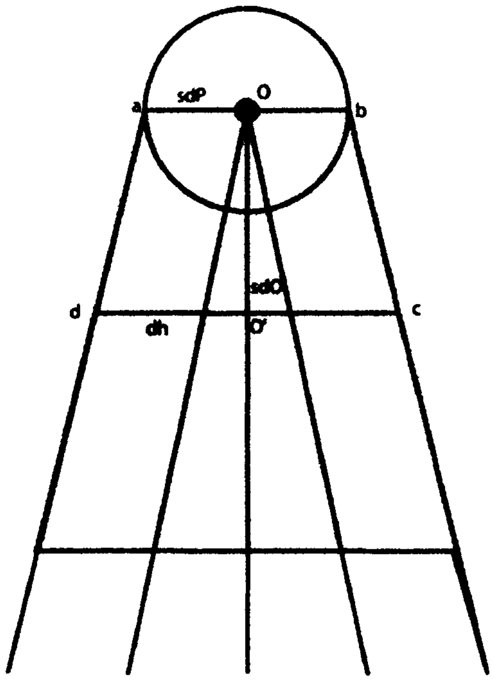

Figure 5.2: The *there* query window.

Equations 5.6-5.11 show how to calculate the coordinates for the corners abcd of the *there QW* polygon, where *dist* is the distance from the user's position *O* to segment *dc*.

$$QW\left[sdP, sdO, x0, y0, yaw0\right] := \begin{pmatrix} a(x1, y1) \\ b(x2, y2) \\ c(x3, y3) \\ d(x4, y4) \end{pmatrix} \tag{5.6}$$

Equations 5.7 a and b calculate the coordinates of points a and b with input x0 and y0, which are the coordinates of the user's position O. The standard deviation of the sensed position, *sdP* is given by the sensor device manufactures. The angle yaw is sensed by the orientation device and is the angle from magnetic north clockwise to the direction the user is interested in.

$$a\left[sdP, x0, y0, yaw0\right] := \begin{pmatrix} x1 = x0 - \left(sdP \times \sin(yaw - 90)\right); \\ y1 = y0 - \left(sdP \times \cos(yaw - 90)\right) \end{pmatrix} \tag{5.7a}$$

$$b\left[sdP, x0, y0, yaw0\right] := \begin{pmatrix} x2 = x0 + \left(sdP \times \sin(yaw - 90)\right); \\ y2 = y0 + \left(sdP \times \cos(yaw - 90)\right) \end{pmatrix} \tag{5.7b}$$

The point O′ is on segment dc, which is a distance *dist* away from the user's position O. The coordinates x′ and y′ are then used to calculate the coordinates for the *QW* corners c and d.

$$O'\left[x0, y0, dist, yaw\right] := \begin{pmatrix} x' = x0 + \left(dist \times \sin(yaw)\right); \\ y' = y0 + \left(dist \times \cos(yaw)\right) \end{pmatrix} \tag{5.8}$$

Once the coordinates for the point O′ are found the next step is to calculate the distance from O′ to the corners c and d of the *QW* polygon. This distance *dh* is calculated in Equation 5.9.

$$dh[dist, sdP, sdO] := sdP + dist \times \tan(sdO) \tag{5.9}$$

Equations 5.10 a and b calculate the coordinates for the corner points c and d. This is done with the coordinates of O′ and the distance *dh* from this point to the *QW* corners c and d.

$$c[x', y', dh, yaw] := \begin{pmatrix} x3 = x' + (dh \times \sin(yaw - 90)); \\ y3 = y' + (dh \times \cos(yaw - 90)) \end{pmatrix} \tag{5.10a}$$

$$d[x', y', dh, yaw] := \begin{pmatrix} x4 = x' - (dh \times \sin(yaw - 90)); \\ y4 = y' - (dh \times \cos(yaw - 90)) \end{pmatrix} \tag{5.10b}$$

After the development of the *there QW* the rest of the query process is the same as for *here* queries. All entities that have a part of their area in common with this query window are then candidates for the query result; entities that are outside the query window are not candidates. The constraint in the SQL WHERE clause is re-written.

WHERE QW *{inside, coveredBy, overlaps, equal, contains, covers}* *.geometry*

5.4 Re-Writing *There+* Queries

There+ queries function with many of the same algorithms as *there* queries, except that for *there+* queries the theme of interest is known, which allows the information system to decide on the granularity level the user should be provided with as a result. In order for the theme type to be known it had to be selected by the user, for example, "What *mountain* is over there?" The *there+* object selection operation can be structured similar to the way queries are structured for non-egocentric window queries. By windowing a theme, one obtains another theme that includes only those objects of the input theme that overlap a given area or window (Rigaux *et al.* 2002). WINDOWING (g, r) is the Boolean operation that consists of testing whether a geometric object g intersects a rectangle r. Testing whether one vertex of g is within the rectangle is insufficient, as a rectangle may intersect a polygon or a polyline without containing any endpoint; therefore, the algorithm consists of scanning the edges of the polygon boundary of object g and testing whether an edge intersects one of the rectangle edges. This algorithm performs two inclusion tests because the geometric object g might be entirely covered by (or might entirely contain) rectangle r.

In an egocentric spatial operation situation where the user and the sensors provide a perspective view, the viewing window is a two-dimensional polygon for two-dimensional map information, instead of a rectangle. The one end of this polygon is the user's current position plus the positional standard deviation. In a three-dimensional query this query polygon is a cone where the vertex is the user's current position plus the accuracy standard deviation and the radius of the cone's cross section is the accuracy propagation of the digital sensors. In both two- and three-dimensional cases the window is skewed a little, based on the fact that there are discrepancies between the direction the sensors are pointed and the direction the user sees the sensors pointed to (Figure 5.3).

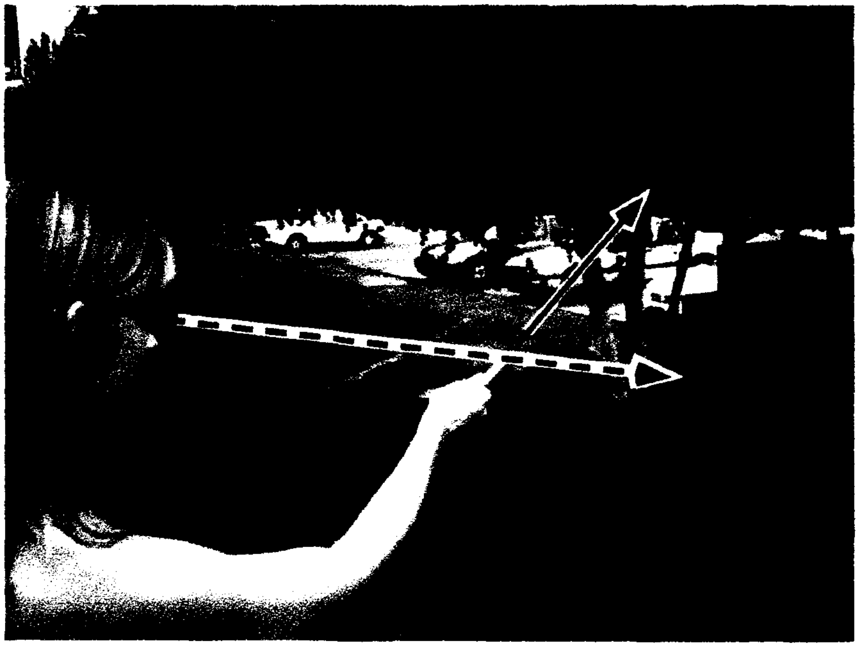

Figure 5.3: Difference between line-of-sight over the tip of the pointer (dashed) and sensed orientation of the pointer (solid).

As a result, there are three possible WHERE clause types:

1. mountain.geometry intersects with a ray
2. mountain.geometry containes a cone
3. mountain.geometry inside, covers, or covered by a cone

For *there+* queries the query window *QW* is created in the same way as for *there* queries and the cone window query is also used. Once a sorted descending list of best-fit regions is created, the *QW* is tested to see if it has something in common with a building region. Such an SQL WHERE clause may be re-written as:

WHERE *QW*{inside, coveredBy, overlaps, equal, contains, covers} building.geometry

This *there+* WHERE clause is different from that for *there* queries because in *there+* building.geometry is used instead of *.geometry. The *.geometry checks for all geometries that have something in common with the *QW* whereas the building.geometry only checks for building geometries.

5.5 Hypothesis Confirmation

The goal of this research is to seamlessly integrate the user's perspective into the query language of a mobile GIS. The hypothesis is supported by because the measurements from position and orientation sensors are sufficient to formulate executable spatial queries about "here" and "there". The queries re-write rules demonstrated in this chapter show how database ADTs, which store data about the user time-of query position and orientation can be used to process SQL queries where the terms "here" and "there" are used. These re-write rules therefore allow a database management system to process the egocentric perspective that users have of their surroundings.

5.6 Summary

The *egocentric spatial data model* incorporates two kinds of egocentric spatial queries, relating to the user's concept of *here* and *there*. In this chapter, the *here* and *there* queries are re-written so that they can incorporate different kinds of spatial data sets. Such query re-writing is necessary in order to relieve the user of the cognitive load of knowing the details regarding how to process the actual query. This query re-write avoids the necessity of a user dealing explicitly with sensor accuracies, and to minimize otherwise cumbersome dialogues between user and system to determine desired levels of detail.

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

Individuals in unknown locations, such as utility workers in the field, soldiers on a mission, or sightseeing tourists, share the need for an answer to two basic questions: "Where am I?" and "What is in front of me?" Because such information is not readily available in foreign locations, aids in the form of paper maps or mobile GISs, which give individuals an all-inclusive view of the environment, are often used. The panoptic view of these maps may impede the user's positioning and orienteering process, since people perceive their surroundings in a perspective way from their current position. This thesis describes a novel framework that resolves the problem of finding the correct reference frame by applying sensors that gather the individual's spatial frame of reference. This spatial frame of reference, in combination with an *egocentric spatial data model*, enables an injective mapping between the real world and the data, hence alleviating the individual's cognitive workload. Furthermore, this *egocentric spatial data model* allows information systems to capture the notions of *here* and *there* and, consequently, provides insight into an individual's surroundings. Finally, this framework, in conjunction with the context given by the task to be performed, enables information systems to implicitly answer questions with respect to where, what, and how things are occurring in the user's surroundings.

6.1 Summary

Paper maps, which provide users with an allocentric view of their surroundings, are often difficult to use, because they are static and inflexible. The inflexibility of paper maps creates three key problems: (1) users have to cognitively place themselves into map space, (2) the user cannot change the thematic information displayed on a map, and (3) users cannot change the granularity or zoom level of the map without creating a new map. Digital maps and GISs have alleviated some of these problems. In a GIS the depicted space is more malleable, the user can pan and zoom over the map. Some of the problems associated with a paper map still exist in the GIS environment. For example, users who want to find information about their local environment still need to cognitively place themselves into the map space. A problem of GISs beyond those associated with paper maps is that GISs are used in desktop computing environments, where interaction with the computing device is the user's primary task. Users in the field, such as tourists or tax assessors, are mainly interested in finding information about their surroundings. A direct interaction with their surroundings means that interacting with the mapping system becomes a secondary focus. This makes standard GIS manipulation in the field complicated and inefficient, since a user in the field has a different focus than a user at a desk. Finding an innovative solution to this problem lead to research on *intelligent mobile GISs,* finally resulting in this thesis.

Intelligent mobile GISs are aware of the user's spatial context and can automatically provide map information in a form the user wants. An *intelligent mobile GIS* senses the user's movements and decides what to center the map on, how to orient the map, and what zoom level or granularity best supports the user's task. The system intelligently decides on these map attributes by sensing the egocentric reference frame of the user. This reference frame is based on the user's location, orientation, speed, path traveled, and queries made. Once the map data is aligned with the user's reference frame it is also possible to process egocentric queries.

An *intelligent mobile GIS* is comprised of a collection of spatially-aware mobile client devices and an *egocentric spatial data model.* This thesis examined two kinds of client devices: (1) a mapping device, which looks like a digital map but has a context driven intelligence, and (2) a pointing device, which acts like a computer mouse for the real world. With the pointing device users can point at objects in their surrounding, click a button, and receive information about the selected object. Users are still communicating with a geospatial information system, but they are not aware of this interaction since it occurs on an application server. Instead of working directly with the map, users of the pointing technology receive attribute information about the object they selected as video, audio, or text. These two mobile computing devices make it easier for users to find information about their surroundings. The second part of the *intelligent mobile GIS* is a data model. The *egocentric spatial data model* translates between the allocentric data in the database management system and the egocentric point of view users have of their surroundings.

An *egocentric spatial data model* has two components: (1) ADTs that store up-to-date data about a user's position and orientation, and (2) procedures to relate the egocentric reference frame created by these ADTs to the absolute reference frame of the spatial dataset. Such procedures provide the information system with an understanding of the notion of *here* and *there*. An information system with an insight into the user's egocentric perspective can use this contextual information to answer questions users might have about their surroundings with a grammar that is easier for the user to understand. Consequently the interaction between user and information system is similar to interaction between humans.

6.2 Findings

In developing the *egocentric spatial data model* several thing were learned:

- Spatial databases can process the term "*here*" and "*there.*"
- This processing can occur with the use of egocentric ADTs that function with up-to-date sensed spatial data.
- Location sensors enable new types of spatial queries.
- New query paradigm automated based on sensors.
- Integration of spatial aware sensors with a query language allows for mobile queries that are transparent to the user.

6.3 Future Work

The egocentric ADT developed by this research allows for possible future research tasks. A variety of issues remain to be resolved. One of them is to study the effect of spatial-awareness on the design of dynamic graphical user interfaces. The use of egocentric spatial-awareness in database management system is an emerging field and many questions are open. The next sections discuss new questions that became apparent through the results of this thesis. They address ADT implementation, sensed data histories, context-aware human-computer interaction, privacy issues, ontology-driven user profiles, multimedia integration, and digital terrain models.

6.3.1 ADT Implementation

This thesis develops the syntax, semantics, and execution model for the *egocentric spatial data model* the next step is to implement the ADT within a database management system. Once this system is implemented it will be possible to examine query optimization procedures. With an implemented system it will also be possible to develop query evaluation models, and investigate use case studies.

6.3.2 Egocentric Histories

The *egocentric spatial data model* allows for a users sensed position and orientation to be incorporated into a database management system. It will be possible represent more than the current positional and orientation data. The system can contain past sensed data thereby allowing the system to track the user. An insight into where the user has been will make it easier to predict where they are going. This will allow the system to pre-buffer data that it deems necessary for the user. Keeping track of the sensed data will allow the system to perform root mean square analysis and improve the data's accuracy. Some of the difficulties of developing a sensed data tracking system would be; to decide how much history to keep track of, how to represent, and index the historical data.

6.3.3 Context-aware Human-Computer Interfaces

The criteria for creating *intelligent mobile GISs*, are based on human-to-human communication, such as between a traveler and a Cicerone. When people talk to each other, they relate their information transmission to the context of the situation. This context can be spatially, temporally, and culturally based. In order for a computer to be able to provide a similar level of communication it needs to be spatially-aware. This spatial awareness will allow the system to modify its Graphical User Interface (GUI) to better serve the user's needs. In this thesis we look at driving a map's centricity, orientation, and zoom level based on the user's spatial context. In the future the information system interface should be driven by context. This context driven interface more than the GUI, the system should be able to use context to decide what, when, and where to use sound, video, and text to present its information to the user. The interface will be media rich. The basic principles developed in this thesis can be used as a platform for research in human-computer interaction.

6.3.4 Ontology-Driven User Profile

Chapter 2 argues that mobile maps, as well as computers, need to become ubiquitous in order for the map to disappear from the user's conscious perception. One way to make a technology ubiquitous is to build the technology based on an insight into the context of the tasks a user is performing. This contextual knowledge allows the technology to adapt to the needs of the user. The focus of this thesis was to collect and use information about the user's spatial context. In the future, information systems will need to collect other attribute information about the user, such as their likes, dislikes, historical experiences, education, and training. A system with this kind of knowledge will be able to provide information with an increased level of usability to a user. An *ontology-driven* user profile query system will be able to semantically align users' informational needs with the information describing their environment. Spatial information is very diverse and collected and stored by different organizations. An ontology profile system will be able to examine the semantics of these diverse data sets and filter the information so that it pertains to the users' interests. This ontology profile system can intelligently decide on the level-of-detail to provide to the user.

6.3.5 Privacy Issues

There are two key privacy issues that have arisen from the research in *intelligent mobile GISs*: the first key issue is related to the privacy of the system's user, and the second one relates to information that can be accessed by the system. The *egocentric spatial data model* holds information about users that they might want to keep private, for example their geographic position and orientation at any point in time. From this stored information people will be able to know where, when, and what the system's user is looking at. This information is necessary to accurately aid the user, but it must be secure and private. As using other context information about the user extends the system, the privacy problem becomes even bigger. The second issue relates to the geospatial information that users of the system could have access to. The pointing technology should allow different levels of access, for example, pointing at someone's house, a police officer or tax assessor should have access to information about the land parcel and house owner, but other people should not. Privacy should be an important consideration when developing the egocentric spatial data model because in order for a mobile GIS to be considered intelligent users must trust the information system. This trust is created though a privacy structure used by the information system.

6.3.6 Multimedia Geo-Footprint Digital Earth

Both the mapping and pointing technologies have the ability to provide non-spatial information. This information can be in many formats, such as audio, video, text, and multimedia. These multimedia datasets need to be linked to the selected features of interest, which could be points, lines, or regions. One problem is that the system might not know what multimedia file to access when a user points at a road or building. Future research needs to examine the development of geo-encapsulation or creating geo-footprints of data. This footprint allows media to link to any geospatial feature. The geo-footprint needs to travel with the media file. Having a separate meta-data file will not work because this information needs to be updated when the file is. Some research has already been conducted in this area of geo-libraries, but it needs to be extended to include all media formats, as well as, being encapsulated in the media themselves.

6.3.7 Digital Terrain Models

An *intelligent mobile GIS* client uses sensed data about its users' egocentric spatial reference frame to calculate the features of interest for them. This selection process is based on a terrain intersection model algorithm (Faisal 2003), where a ray from the user's position intersects a three-dimensional object in a terrain model. These terrain models need to be robust and their file size needs to be small so they can be transferred fast and efficiently. For a terrain model to be useful in such an environment it needs to store information about physical terrain, which is a continuous field data structure as well as data about physical geospatial objects such as buildings and trees. In order for a terrain model to function at many different levels of detail it also need to be database driven.

Today's digital terrain models do not live up to these requirements yet and more research in this area has to be done.

BIBLIOGRAPHY

G. Abowd, C. Atkeson, J. Hong, S. Long, R. Kooper, and M. Pinkerton (1997) Cyberguide: A Mobile Context-aware Tour Guide. *Baltzer / ACM Wireless Networks* 3(5): 421-433.

G. D. Abowd and E. D. Mynatt (2000) Charting Past, Present, and Future Research in Ubiquitous Computing. *ACM Transactions on Computer-Human Interaction* 7(1): 29-58.

J. Baus and C. Kray (2002) Frames of Reference, Positional Information and Navigational Assistance, *Flairs 2002*, Pensacola, FL, pp. 461-466, AAAI Press.

K. Beard and V. Sharma (1997) Multidimensional Ranking for Data in Digital Spatial Libraries. *International Journal on Digital Libraries* 1(2): 153-160.

J. L. Bergren and A. Jones (2000) *Ptolemy's Geography: An Annotated Translation of the Theoretical Chapters*. Princeton, NJ, Princeton University Press.

M. Bertolotto, J. D. Carswell, L. McGeown, and P. Thijs (2002) Geospatial Data Handling for Web-based and Mobile Applications. *Processing of Joint International Symposium on Geospatial Theory*. Ottawa, Canada, Electronic Proceedings.

M. Burnett, P. Prekop, and C. Rainsford (2001) Intimate Location Modeling for Context Aware Computing. in: M. Beigl, P. Gray, and D. Salber (Eds.), *Ubicomp 2001*, Atlanta, GA, Location Modeling for Ubiquitous Computing, pp. 77-82.

D. Chamberlin and R. Boyce (1974) SEQUEL: A Structured English Query Language, *Proceedings of the ACM-SIGMOD Workshop on Data Description, Access, and Control*, pp. 249-264.

G. Chen and D. Kotz (2000) *A Survey of Context-Aware Mobile Computing Research*. Technical Report TR2000-381, Hanover, NH, Dartmouth College.

K. Cheverst, N. Davies, K. Mitchell, A. Friday, and C. Efstratiou (2000) Developing a Context-Aware Electronic Tourist Guide: Some Issues and Experiences, *Proceedings of the CHI 2000 Conference on Human Factors in Computing Systems*, The Hague, Netherlands, pp. 17-24.

E. F. Codd (1970) A Relational Model of Data for Large Shared Data. *Communications of the ACM* 16(6): 377-387.

J. R. Cooperstock, S. S. Fels, W. Buxton, and K. C. Smith (1997) Reactive Environments: Throwing Away Your Keyboard and Mouse. *Communications of the ACM* 40(9): 65-73.

H. Couclelis (1992) Location, Place, Region, and Space. in: R. F. Abler, M. G. Marcus, and J. M. Olson (Eds.) *Geography's Inner Worlds*: 215-233. New Brunswick, NJ, Rutgers University Press.

A. Dey and G. Abowd (1999) The Context Toolkit: Aiding the Development of Context-Aware Applications, *Proceedings of Human Factors in Computing Systems*, Pittsburgh, PA, pp. 434-441, ACM Press.

A. Dix, T. Rodden, N. Davies, J. Trevor, A. Friday, and K. Palfreyman (2000) Exploiting Space and Location as a Design Framework for Interactive Mobile Systems. *ACM Transactions on Computer-Human Interaction* 7(3): 285-321.

M. Egenhofer (1994) Spatial SQL: A Query and Presentation Language. *IEEE Transactions on Knowledge and Data Engineering* 6(1): 86-95.

M. Egenhofer (1999) Spatial Information Appliances: A Next Generation of Geographic Information Systems. in: J. Paiva (Ed.), *Geo-Info 1999 I Brazilian Workshop on GeoInformatics*, Campinas, Brazil, Vol. 1, pp. 4, SIBGRAPI.

M. J. Egenhofer and W. Kuhn (1998) *Beyond Desktop GIS*. www.spatial.maine.edu/~max/BeyondDesktopGIS.pdf

F. Faisal (2003) *Query-By-Pointing: Algorithms and Pointing Error Compensation*. MS. thesis, Orono, University of Maine.

S. Feiner, B. MacIntyre, T. Hoellerer, and A. Webster (1997) A Touring Machine: Prototyping 3D Mobile Augmented Reality Systems for Exploring the Urban

Environment, *Proceedings ISWC '97 Int. Symp. on Wearable Computing*, Cambridge MA, pp. 74-81, IEEE Press.

J. Florence (1997) *Predicting Frequences of Topogical Relations in Geographic Datasets*. MS. thesis, Orono, University of Maine.

F. Fonseca, M. Egenhofer, P. Agouris, and G. Camara (2002) Using Ontologies for Integrated Geographic Information Systems. *Transactions in GIS* 6(3): 231-257.

N. Franklin (1996) From Language to Cognitive Maps. in: J. Portugali (Ed.) *The Construction of Cognitive Maps*. The GeoJournal Library 32: 275-296, Kluwer Academic Publishers.

Frost and Sullivan (2003) *North American Location-Based Services Markets*. Technical Report.

R. G. Golledge (1999) *Wayfinding Behavior: Cognitive Mapping and Other Spatial Processes*. Baltimore, MD, The Johns Hopkins University Press.

M. F. Goodchild (1998) The Geolibrary. in: S. Carver (Ed.), *Fifth National Conference on GIS Research UK*, London, Vol. 2, pp. 59-68, Taylor & Francis.

R. Guting (1994) An Introduction to Spatial Database Systems. in: P. Apers, H.-J. Schek, J. Gray, and S. Su (Eds.) *The International Journal on Very Large Data Bases* 3: 357-400.

B. J. Harley and D. Woodward (1991) An Alternative Route to Mapping History. *Historical Research - Equipment and Supplies Maps*. 43: 6-8.

A. Harter, A. Hopper, P. Steggles, A. Ward, and P. Webster (1999) The Anatomy of a Context-Aware Application, *International Conference on Mobile Computing and Network (Mobicom)*, Vol. 5, pp. 59-68, ACM Press.

A. Hinze and A. Voisard (2003) Location- and Time-Based Information Delivery in Tourism. in: T. Hadzilacos, Y. Manolopoulos, J. F. Roddick, and Y. Theodoridis (Eds.), *Advances in Spatial and Temporal Databases*, Santorini Island, Greece, Lecture Notes in Computer Science 2750, pp. 489-507, Springer.

J. C. Hudson (1992) Scale in Space and Time. in: R. F. Abler, M. G. Marcus, and J. M. Olson (Eds.) *Geography's Inner Worlds*: 280-300. New Brunswick, NJ, Rutgers University Press.

C. Jiang and P. Steenkiste (2002) A Hybrid Location Model with a Computable Location Identifier for Ubiquitous Computing. in: G. Borriello and L. E. Holmquist (Eds.), *UbiComp 2002*, Teborg, Sweden, Ubiquitous Computing, Lecture Notes in Computer Science, Vol. 2498, pp. 246-280, Springer.

W. Kuhn (1991) Are Displays Maps or Views?, *Tenth International Symposium on Computer-Assisted Cartography*, Baltimore, MD, Vol. 6, pp. 261-274, American Congress on Surveying and Mapping and American Society for Photogrammetry and Remote Sensing.

B. Kuipers (1982) The "Map in the Head" Metaphor. *Environment and Behavior* 14(2): 202-220.

P. Lukowicz, H. Junker, M. Stäger, T. von Büren, and G. Tröster (2002) WearNET: A Distributed Multi-Sensor System for Context Aware Wearables. in: G. Borriello and L. E. Holmquist (Eds.), *UbiComp 2002*, Teborg, Sweden, Ubiquitous Computing, Lecture Notes in Computer Science, Vol. 2498, pp. 361-370, Springer.

A. M. MacEachren (1995) *How Maps Work: Representation, Visualization, and Design*. New York, NY, The Guilford Press.

W. Mackaness (2002) Conveying Route Information on Mobile Devices Using Communicative Acts. in: M. Egenhofer and D. Mark (Eds.), *GIScience 2002, Lecture Notes in Computer Science*, Vol. 2418, pp. 101-104, Springer.

J. B. Maitra and N. Andersen (2003) Geospatial Standards. *ISO Technical Committee 211*: 13-35.

R. K. Majumdar, K. Ramamritham, and M. Xiong (2003) Adaptive Location Management in Mobile Environments. in: M.-S. Chen, P. K. Chrysanthis, M. Sloman, and A. Zaslavsky (Eds.), *4th International Conference Mobile Data Management*, Melbourne, Australia, pp. 196-210, Springer.

F. R. McFadden and J. A. Hoffer (1985) Boyce-Codd Normal Form *Data Base Management*: 520. Menlo Park, California, The Benjamin Cummings Publishing Company.

M. Minsky (1985) *The Society of Mind.* New York, NY, Simon and Schuster.

A. K. Narayanan (2001) Realms and States: A Framework for Location Aware Mobile Computing, *Seventh Annual International Conference on Mobile Computing and Networking (MOBICOM)*, Rome, Italy, pp. 48-54.

D. Norman (1990) *The Design of Everyday Things.* New York, Currency and Doubleday.

D. Norman (1999) *The Invisible Computer.* Cambridge, MA, The MIT Press.

S. Nusser, L. Miller, K. Clarke, and M. Goodchild (2003) Geospatial IT for Mobile Field Data Collection. *Communications of the ACM* 46(1): 45-46.

R. Ramakrishnan and J. Gehrke (2000) *Database Management Systems.* Boston, MA, McGraw-Hill.

J. Raper (2002) The Dimensions of GIScience.

http://www.soi.city.ac.uk/~raper/research/GIScience2002-OHs-pub.ppt

M. Raubal, M. J. Egenhofer, D. Pfoser, and N. Tryfona (1997) Structuring Space with Image Schemata: Wayfinding in Airports as a Case Study. in: S. Hirtle and A. Frank (Eds.), *COSIT '97, Lecture Notes in computer Science*, Vol. 1329, pp. 85-102.

M. Raubal and M. Worboys (1999) A Formal Model of the Process of Wayfinding in Built Environments. in: C. Freksa and D. Mark (Eds.), *COSIT'99 International Conference*, Spatial Information Theory, Vol. 1661, pp. 381-401.

B. C. Richardson (1992) Places and Regions. in: R. F. Abler, M. G. Marcus, and J. M. Olson (Eds.) *Geography's Inner Worlds*: 27-49. New Brunswick, NJ, Rutgers University Press.

P. Rigaux, M. Scholl, and A. Voisard (2002) *Spatial Databases with Applications to GIS.* San Francisco, CA, Morgan Kaufmann Publishers.

T. Rodden, K. Chervest, N. Davies, and A. Dix (1998) Exploiting Context in HCI Design for Mobile Systems, *Workshop on Human Computer Interaction with Mobile Devices*, Glasgow, Scotland.

N. Roussopoulos, S. Kelley, and V. Frederic (1995) Nearest Neighbor Queries, *ACM-SIGMOD on Management of Data*, San Jose, CA, pp. 71-79.

B. Schilit and M. Theimer (1994) Dissemination Active Map Information to Mobile Hosts. *IEEE Network* 8(5): 22-32.

B. N. Schilt, N. Adams, and R. Want (1994) Context-Aware Computing Applications, *In Proceedings of the Workshop on Mobile Computing Systems and Applications*, Santa Cruz, CA.

A. Schmidt, M. Beigl, and H.-W. Gellersen (1999) There is More to Context than Location. *Computer & Graphics Journal* 23(6): 893-902.

J. Sholl (1996) From Visual Information to Cognitive Maps. in: J. Portugali (Ed.) *The Construction of Cognitive Maps.* The GeoJournal Library 32: 157-186, Kluwer Academic Publishers.

A. Silberschatz, H. F. Korth, and S. Sudarshan (1999) *Database System Concepts.* Boston, MA, McGraw-Hill.

D. Simonis (2002) *Florence.* Lonely Planet Publications.

R. T. Snodgrass (1995) *The TSQL2 Temporal Query Language.* Boston, Kluwer Academic Publishers.

D. Stea, J. M. Blaut, and J. Stephens (1996) Cognitive Mapping and Culture. in: J. Portugali (Ed.) *The Construction of Cognitive Maps.* The GeoJournal Library 32: 345-360, Kluwer Academic Publishers.

M. Stonebraker (1986) Inclusion of New Types in Relational Database Systems, *International Conference on Data Engineering*, Los Angeles California, Vol. 2, pp. 262-269, IEEE Computer Society.

A. Tansel, J. Clifford, S. Gadia, S. Jajodia, A. Segev, and R. Snodgrass (1993) *Temporal Databases Teeory, Design, and Implementation*. Redwood, CA, The Benjamin/Commings Publishing Company Inc.

M. Taylor (2002) *World Travel Atlas*. London, Pete Kornicky Columbus Publishing.

B. Thomas, V. Demczuk, W. Piekarski, D. Hepworth, and B. Gunther (1998) A Wearable Computer System with Augmented Reality to Support Terrestrial Navigation, *2nd International Symposium on Wearable Computers ISWC1998*, Pittsburgh, PA, USA, pp. 168-172, IEEE Computer Society.

S. Timpf, G. S. Volta, D. W. Pollock, and M. J. Egenhofer (1992) A Conceptual Model of Wayfinding Using Multiple Levels of Abstraction, *Theory and Methods of Spatio-Temporal Reasoning in Geographic Space, Lecture Notes in Computer Science*, Vol. 639, pp. 348-367.

M. Weiser (1991) The Computer for the 21st Century. *Scientific American* 1991(September).

M. Weiser (1993) Some Computer Science Issues in Ubiquitous Computing. *Communications of the ACM* 36(7): 74-83.

D. Woodward (1992) Representations of the World. in: R. F. Abler, M. G. Marcus, and J. M. Olson (Eds.) *Geography's Inner Worlds*: 50-76. New Brunswick, NJ, Rutgers University Press.

M. F. Worboys (1995) *GIS: A Computing Perspective*. London, Taylor & Francis.

A. Zipf and H. Aras (2002) Proactive Exploitation of the Spatial Context in LBS through Interoperable Integration of GIS-Services with a Multi Agent System. *5th AGILE Conference on Geographic Information Science*. Palma, Spain, Electronic Proceedings.

BIOGRAPHY OF THE AUTHOR

Christopher Frank was born in Burlington, Vermont on February 14, 1977 and grew up on the shores of Lake Champlain in Colchester, Vermont. Mr. Frank studied Spatial Information Science and Engineering at the University of Maine in Orono Maine. He holds a Bachelor of Science degree in Spatial Information Engineering with a minor in Computer Science. In addition to extensive information systems design and development experience, Mr. Frank has participated in several Entrepreneurship and Small Business Management classes at the University and The Target Technology Incubator Center. From 2000 to 2003 Mr. Frank worked with Dr. Max Egenhofer researching and developing innovative spatial technologies at the National Center for Geographic Information and Analysis. During this time, Mr. Frank gained a strong knowledge of general spatial principles and developed a sensor-based mobile spatial query system. Since April 2003 Mr. Frank has been President of Intelligent Spatial Technologies (IST), a software company that is commercializing the intelligent mapping and pointing technology showcased in this thesis. The company has a strong background in spatial information research combined with over five years of experience in software design and development. IST's mission is to provide a full-range of integrated navigation, wayfinding, traveling, and information services that give the user control of when, where and what kind of information is provided. Christopher Frank is a candidate for the Master of Science degree in Spatial Information Science and Engineering from The University of Maine in December, 2003.