A System for Continuous, Real-Time Search and Retrieval of Georeferenced Objects*

Debra Lee Davis-Chu, Nagarajan Prabakar and Naphtali Rishe

High Performance Database Research Center (HPDRC), School of Computer Science

Florida International University, Miami, FL 33199

Tel: (305) 348-6239 E-mail: davisd@cs.fiu.edu Fax: (305) 348-1705

Abstract

As remotely sensed data has become more readily available, so has the number of applications for which this data is used. Dealing effectively with spatial data often involves the integration of different types of data. This can be a difficult and time consuming process for the average end user unless the software package they are using is both user friendly and efficient. This paper describes the technology behind the implementation of an automated system that continuously searches, retrieves and displays information regarding georeferenced objects as the user 'flies' over associated spatial data. These georeferenced objects are stored as digital data, such as GNIS and US Census Tiger/Line data, in the form of point, line and polygon data. The technical issues and solutions involved in its implementation and optimization are presented.

Keywords: remotely-sensed data, georeference, semantic object-oriented database

1. Introduction

The importance of spatial data has greatly increased in recent years. The information that this data provides and the applications for which it can be used are vast. An inherent problem with spatial data, however, is the complexity involved in extracting and understanding the needed information. This has been particularly problematic for end users who are not GIS experts, yet for whom this data is very

* This research was supported in part by NASA (under grants NAGW-4080, NAG5-5095, NAS5-97222, and NAG5-6830) and NSF (CDA-9711582, IRI-9409661, HRD-9707076, and ANI-9876409).

valuable. Part of the problem faced by users is that spatial data sets are inherently very large and come in many different formats. Yet, for the data to be useful for their applications, different types of spatial data must be accessed and combined, such as integrating remotely-sensed data (e.g., aerial photography, satellite imagery, etc.) with digital data sets (e.g., textual based point, line and polygon data). For example, one particularly useful capability for the average user includes combining GNIS (Geographic Names Information System) and US Census Tiger/Line data (street and area information) with aerial photography. This gives the user a greater understanding of information such as positional accuracy. These types of manipulations, however, are often beyond the capabilities of the typical user.

Software systems, such as our Web-enabled TerraFly system, have begun to address the needs of the average end user by simplifying the commands needed to access and combine various types of data [4]. The TerraFly system is a prototype interactive vehicle with which users can seamlessly 'fly' over remotely-sensed data. It provides a user-friendly GUI and easy-to-use data manipulation capabilities. The proposed technique provides the technology to extend the TerraFly system where, as users are 'flying' over remotely sensed data, information regarding associated georeferenced objects, such as the nearest place, place type, street intersection and/or populated-area/county-subdivision, is continuously changing and simultaneously displayed. To accomplish this efficiently, in real time, is a complex matter. This involves constant client/server interaction, database retrieval, data sorting, search procedures and calculation functions. Internet related issues such as limited bandwidth and browser limitations further complicate matters. This paper will discuss the relevant technical issues involved in implementing this system along with corresponding solutions.

2. Background

This proposed system employs the Semantic Object-Oriented Database System (Sem-ODB) [7, 9] and is based upon HPDRC's Web-enabled TerraFly technology [4, 1]. Some details of our Sem-ODB and Web TerraFly research projects are presented below.

The Semantic Object-Oriented Database System (Sem-ODB)

Due to the inherently large size of spatial data, the storage and retrieval method used in this system is central to its success. The Sem-ODB technology under development at HPDRC has been designed to be efficient at dealing with spatial data and related products [7, 9]. Sem-ODB is a general-purpose database management system (DBMS) designed to store varied types of data in an efficient and logical manner, and it easily deals with non-conventional data such as spatial data, as well as with different types of data in the same database [13, 4]. Some of Sem-ODB's key advantages over current database technology are:

- Gives control to the user via an intuitive structure of information.
- The end-user is empowered to pose complex ad hoc queries.
- A conceptual data model of the enterprise is directly supported.
- Queries can be up to ten times shorter (and so easier to pose) than in relational databases.
- User programs for a semantic view are substantially shorter than for a relational view, achieving major improvements in the application software development cycle, maintenance and reliability.
- Data types are unlimited, strings can be of any length and numbers can be of unlimited length and precision.
- We have developed algorithms to provide very efficient full indexing, allowing fast access to every single fact.
- There is no need for NULL attributes. Sparse tables in relational databases waste space and processing time.
- There is no need for tables and indices, reducing the space allocation required.

Sem-ODB can easily handle Terabytes of data. To further improve performance and flexibility, Sem-ODB can be used as a distributed database, as it is in this system. In this way, data could be stored on multiple servers and in multiple locations, and retrieved simultaneously from the various locations [10].

TerraFly

This system is an extension to our Web-enabled TerraFly technology, a prototype interactive vehicle for 'flying' over remotely-sensed data via any standard Web browser. The TerraFly system applies Sem-ODB technology for storage and retrieval of in-house data used by the system and allows its users to 'fly' over and manipulate the retrieved data. TerraFly database currently contains text, remotely-sensed raster data [6] and graphical data (graphical maps). A friendly graphical user interface is provided for ease of use.

Some of the features currently available in TerraFly are:

- Multiple Data Types & Multiple Flight Windows
- Fine Flight Direction Control
- Varied Flight Speed and Refresh Rate
- Informational Textboxes
- Go-To Coordinate, Place and Address Functions
- Sensor Band Controls for multi-spectral data types
- RGB Intensity Control
- Data Dispensing Capability

3. System Overview

The main goal of this system is to improve the ease and usefulness of the capability to combine and understand different types of spatial data via the Internet. This system provides the technology which, when merged with TerraFly, allows users to 'fly' over remotely-sensed data such as Aerial Photography or Landsat imagery while digital information regarding associated georeferenced objects is continuously retrieved and displayed in real time. The information provided includes the name of the closest place, place type, street intersection, populated-area/county-subdivision and distance from the center point of the image to these objects. This system uses a semantic database schema, semantic R-tree [3] and graph data structures, and associated algorithms, which can be used to efficiently store, retrieve, manipulate and

combine georeferenced digital data with remotely-sensed data. Further, this approach can be used in conjunction with remotely-sensed data at varying resolutions.

Georeferenced Object Data

There are many types of georeferenced object data (e.g., textual point, line and polygon data) currently available. Two types of data, GNIS (Geographic Names Information System) [12] and US Census Tiger/Line files [11], are used by this system. GNIS data primarily consists of names and types of places along with associated coordinate point information. US Census Tiger/Line data consist of georeferenced point, line and polygon data. The Tiger/Line data currently used by this system includes information regarding US highways, major roads, streets and populated area/county subdivision shape coordinate points data. The US highways, major roads and street/address data is line data, with each line segment consisting of a series of ordered georeferenced points with separate beginning and ending points. The populated area/county subdivision data is polygon data consisting of a series of ordered georeferenced points that form a polygon shape.

Relevant Implementation Issues

At first, it may seem that the main goal of this work is straightforward and easy to accomplish. When one looks at the details, however, one discovers that it involves a number of rather complex issues that must be addressed. In short, the main issues are:

- Constant searching involving a large amount of georeferenced data
- Internet bandwidth and browser limitations
- CPU and memory intensive functions

Although these issues are outlined separately above, they are, in fact, strongly interrelated. Because we are dealing with constant retrieval of a large amount of georeferenced object data in conjunction with associated remotely-sensed data, efficient storage and retrieval capabilities are extremely important if we want to maintain continual, smooth flight while constantly retrieving the required data. Further, remotely

sensed data is inherently large, and memory and CPU intensive. Every new feature and each piece of additional data can potentially slow down the system dramatically or, if taxed beyond the browser's capabilities, can crash the client side of the system. We must find a way of effectively dealing with sending large amounts of data over the limited bandwidth currently available over the Internet, as well as dealing with the memory and CPU limitations of current browser technology. Both imagery and digital data are constantly updated and information is constantly recalculated as users 'fly' over the data. This is highly CPU and memory intensive and must be dealt with appropriately to maintain stable yet efficient data flow.

4. System Description

This system is designed to automatically and continually provide users with current information regarding georeferenced objects associated with geographic areas and remotely sensed data of interest. From the user's perspective, only a few simple GUI-based commands are required to accomplish this. From the system's perspective, it is not this simple. A major technical issue in this work centers on the continuous searches of large amounts of data that are required, and the client/server architecture and environment in which the system is implemented. The technology behind this system involves constant client/server interaction, database retrieval, data sorting, search procedures and calculation functions, as well as attention to Internet related issues such as limited bandwidth and browser limitations.

In sum, the process to accomplish this is as follows. When users choose to initiate this capability, the client side of the system sends a request for the desired data to the server. This request includes information such as the type of data needed, resolution and geographic coordinates. When the server side receives the request, it retrieves the needed information from the database using a proprietary algorithm designed specifically for use with Sem-ODB and which retrieves only the data needed at that time. This is coupled with a Semantic database schema designed to precisely take advantage of Sem-ODB's enhancements over other types of databases. Once the data is retrieved, the server inserts the data in the

appropriate data structure (R-tree for point/line data or graph for polygon data). The data structure is then sent to the client.

When the client receives the requested data, it performs either a Nearest Neighbor search [2, 5] through the R-tree data structure [3] (in the case of point/line data) or determines which populated area covers the current geographic location (in the case of polygon data). The information of interest is extracted and displayed to the user (e.g., the closest place street intersection, etc.). As users 'fly' over the data, the client is continuously recalculating and searching through the data for the appropriate information to display. This continues until the flight path crosses a predetermined geographic 'boundary'. Upon crossing this "boundary", the client sends a request to the server for the data using the coordinate where the flight path and "boundary" intersect. The server retrieves and organizes the data as described above and sends it to the client. Once the client receives this new data, the old data structure is deleted from memory and the new data is used. Thus, only data that is actually needed is transferred to the client.

Data Handling

To deal effectively with the complex issues involved in the implementation of this system, we employed a number of solutions that, in combination, can satisfactorily resolve these issues. Our solutions are as follows:

- Employ client/server technology where data processing is done on server side, and limiting the work on the client side to searches of the preprocessed data structures.

  We have found that by using efficient data retrieval techniques on the server side, we can reduce the response time of the system. The greatest lag time, however, is associated with data transmission over the Internet. Because of this, we cannot continually send Nearest Neighbor requests for points continuously during flight and expect to receive data in a timely manner. Thus, packets of data must be sent over the Internet with the client side performing these continuous searches. Because spatial data is inherently large and CPU intensive, and because of the limited memory capabilities of browser

technology, we keep the client side as thin as possible. Thus, the server performs as much data processing as possible before sending the data to the client.

- Incorporate the use of Sem-ODB technology, and design a logical and efficient semantic database schema for faster retrieval.

  Our database schema has been designed to store remotely-sensed and georeferenced object data in the same database. By using Sem-ODB, we are able to use its advantages to quickly and efficiently retrieve the data of interest. One way we are accomplishing this is by preprocessing the georeferenced object data as it is inserted into the database. In short, an additional attribute is created specifically to enable faster searches of the data. This attribute is a specialized character string created from the latitude and longitude coordinates provided with the point, line and polygon data. By using this character string, we can perform range queries significantly faster than with conventional methods.

- Employ the use of a highly efficient yet simple data structures that can be quickly searched on-the-fly.

  As users 'fly', the center geographical coordinate which they are 'flying' over constantly changes. Because of this, the places, street intersections and areas closest to that center coordinate will also be constantly changing. In order to provide accurate information, we chose to employ an R-tree data structure on which the client can then perform Nearest Neighbor searches. This data structure has been found to be very efficient with this type of data [3]. In addition, we have employed a Nearest Neighbor algorithm designed for use with specific data structures that are stored in main memory [2]. For polygon data, a graph structure is used instead. Within the graph, each node will contain an area and its information. Each node will have an edge leading to any nodes that contain information on areas, which are adjacent to each other. Thus, when the user 'flies' out of one area, the client can then limit its search to areas immediately adjacent to the previous area.

- Employ a proprietary algorithm that retrieves the data from the database only as needed.

  Previous research as found that fast and efficient searches of GNIS data can be performed by inserting relevant information regarding spatial objects into an R-tree when the TerraFly system is

initially booted [3]. When limiting the data of interest to GNIS data, we feel that this is the preferable course of action. However, when dealing with much larger amounts of data, such as US Census/Tiger files, placing all of the data in the database into a data structure in the server's main memory is not always feasible. As was mentioned above, we have chosen to employ a proprietary algorithm that retrieves data as needed and inserts the retrieved data into an R-tree, which can then be sent to the client. This then provides the client with the advantages of using an R-tree data structure for the continual searches it must perform.

- On the server side, simplify the processing of line data (e.g., streets) to that of points before sending the data to the client.

  An efficient way of finding the nearest line segment to a point in a plane is through the use of the Nearest Vertical Neighbors algorithm [5]. However, because the line data dealt with in this system is street segments, it makes more sense to provide the user with information regarding the closest street intersection to any given point.

- Preprocess the polygon data to provide minimal bounding boxes that can then be used for more efficient data retrieval and calculations.

  Creating minimal bounding boxes for the polygon data increases speed and efficiency on both the client and server sides. One the server side, database searches can be done using the bounding boxes instead of the actual polygon coordinates. Since this can involve searching through significantly fewer coordinate points, our search time is dramatically reduced. On the client side, the use of bounding boxes can reduce the number of calculations in most cases. To determine whether a particular coordinate point is within a specific polygon, it is first determined whether the point is within the polygon's minimal bounding box. If it is not, it can immediately be eliminated without any further calculations.

5. Conclusion

This paper has presented a real-time search technique to allow users to 'fly' over remotely-sensed data such as Aerial Photography or Landsat imagery while information regarding associated georeferenced objects is continuously retrieved and displayed in real time. This technique provides the technological basis for the creation of an extension of our Web-enabled TerraFly system. The major technical issues involved in this work include efficiently performing continuous, real-time searches of large amounts of georeferenced data, and effectively dealing with the limitations imposed by the client/server architecture and environment in which the system operates. As an automated system, the complexity involved in this technique is transparent to the user. For the system, however, it is accomplished through a combination of approaches including recurrent client/server interaction, appropriate work distribution between the client and server, a proper semantic database schema, and use of efficient data structures, search procedures and calculation functions. We feel that the approach presented here helps reduce the time, expense and difficulty users often encounter when dealing with spatial data and moves users towards a better understanding of their data of interest.

References

1. Alvarez, E. and Rishe, N. "Multimedia Spatial Databases." *Proceedings of the First Workshop on Next Generation Database Design and Applications*, Miami, FL, pp. 1-8, April 30 - May 1, 1998.

2. Arya, S., Mount, D.M., Netanyahu, N. S., Silverman, R. and Wu, A.Y. "An Optimal Algorithm for Approximate Nearest Neighbor Searching in Fixed Dimensions." *J. ACM* 45(6):891-923, Nov 1998.

3. Chen, S., Rishe, N., Wang, X. and Weiss, M.A. "A User-Friendly Multimeida System for Querying and Visualizing of Geographic Data." Unpublished Manuscript, Florida International University, 2000.

4. Davis-Chu, D.L., Alvarez, E.L. and Rishe, N. "The Creation of a System for 3D Satellite and Terrain Imagery" In *Proceedings of the 13th International Conference in Applied Geologic Remote Sensing*, Vancouver, British Columbia, Canada, pp. 329-336, March 1-3, 1999.

5. EppStein, D. and Erickson, J. "Iterated Nearest Neighbors and Finding Minimal Polytopes" *Discrete & Computational Geometry* 11(3):321-350, Apr 1994.

6. Muffin, G. "Raster versus Vector Data Encoding and Handling: A Commentary", *Photogrammetric Engineering and Remote Sensing*, Vol. 53, No. 10, pp.1397-1398, 1987.

7. Rishe, N. "A Database Design: The Semantic Modeling Approach", McGraw-Hill, 1992.

8. Rishe, N. and Li, Q. "Storage of Spatial Data in Semantic Databases." In *Proceedings of the 1994 ASME International Computer in Engineering Conference*, Minneapolis, MN, pp. 793-800, Sept 11-14, 1994.

9. Rishe, N., Barton, D., Chekmasov, M., Madhyanapu, K., Graham, S. and Chekmasova, M. "Everglades Data Integration using a Semantic Database System." *International Conference Geospatial Information in Agriculture and Forestry*, Lake Buena Vista, FL, pp. 567-573, June 1-3, 1998a.

10. Rishe, N., Barton, D., Urban, F., Chekmasov, M., Martinez, M., Alvarez, E., Gutierrez, M. and Pardo, P. "High Performance Database Management for Earth Sciences." *NASA University Research Center Technical Conference*, Huntsville, AL., pp. 539-544, Feb 22-25, 1998b.

11. Tiger Overview, URL: http://www.census.gov/geo/www/tiger/overview.html

12. USGS mapping Information: Geographic Names Information System (GNIS), URL: http://mapping.usgs.gov/www/gnis/

13. Waugh, T.C. and Healey, R.G. “The GEOVIEW Design: A Relational Database Approach to Geographical Data Handling”, International Journal of Geographical Informal Systems, Vol. 1, No. 2, pp. 101-118, 1987.