f = u
99-HE
6106

REGULAR PAPERS

CORRESPONDENCE

Dr. Naphtali Rishe
Florida International University
School of Computer Science
Southwest 8th St. and 107th Avenue
University Park
Miami, FL 33199

1

Φ
IEEE
COMPUTER SOCIETY

INTERNATIONAL CONFERENCE ON IMAGING SCIENCE, SYSTEMS, AND TECHNOLOGY

Editor:
Hamid R. Arabnia
The University of Georgia

Associate Editors:
F. Ahmed, D. Bouchaffra, L. A. Costa,
J. T. Favata, R. M. Haralick, Q. Ji,
Y. Mun, M. Sarfraz

CISST'99

June 28 - July 1, 1999
Las Vegas, Nevada, USA

1999 CSREA

World Scientific

Engineering Society
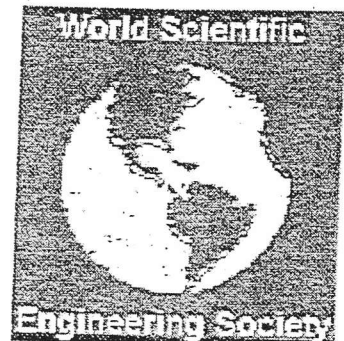
This volume contains papers presented at the International Conference on Imaging Science, Systems, and Technology (CISST'99). Their inclusion in this publication does not necessarily constitute endorsements by the editors or by the publisher.

Printed in the U.S.A.

Spatial Overlay of Vector Data on Raster Data in a Semantic Object-Oriented Database Environment[*]

Nagarajan Prabhakaran. Venkat Maddineni, and Naphtali Rishe
School of Computer Science
Florida International University
University Park, Miami, FL 33199

Abstract

One of the major contributions of Geographical Information Systems (GIS) is their ability to efficiently manage geographical information. Geographical information can be stored and represented in a variety of ways, each with its advantages. Thus a challenging problem facing GIS is their ability to effectively store and utilize the types of geographic data. Even though we have GIS software such as ENVI and Arc/Info, for storing spatial data, a high performance database system is essential to store and utilize all types of spatial and attribute data to meet the emerging needs of GIS.

The overlay of vector data over raster data is vital for the end user so that both types of data can be correlated. Although vector data can be stored in either vector or raster format, the vector format is preferred because it is easier for the computer to "understand" the vector data and manipulate them more efficiently. By efficiently storing the external attribute data associated with vector data, users can easily identify the locations on the resulting superimposed map.

A database schema and an indexing technique are introduced to store vector data in a high performance semantic database environment. This vector data is accessed on demand, at runtime from the database through dynamic queries and the vector data is displayed over raster data. The spatial raster database was already built at Florida International University under TerraFly project. This paper describes the storage of spatial vector data in a Semantic Object-Oriented Database (Sem-ODB)

and the integration of vector data and raster data in a semantic database environment.

Keywords
Vector data, Raster data, Overlay, Tiling

1. Introduction

The traditional data types used in common database applications are integer, float, character, monetary-unit and date. This limited set of data types makes the real world spatial applications extremely difficult [1, 5]. There also exist spatial databases without the capability of storing attribute data in the same physical database. Storing spatial and attribute data in different physical databases suffers from processing overhead and synchronization problems. The other major challenge concerning spatial database is the complexity involved in the handling and display of spatial data. For a commercial database, geo-spatial data can be classified as two types: vector data and raster data. Vector data is a coordinate based data structure commonly used to represent linear map features. Each linear feature is represented as a list of ordered x, y coordinates and is associated with its attributes. Raster data structure is composed of rows and columns where each feature is represented by a group of cells (pixels). The value of each cell represents the value of feature. Unlike in vector data, attributes are associated with the cell grid.

There has been a remarkable surge of environmental, agricultural, scientific, and academic interest in Geographic Information Systems since their graphical nature allows planners to easily visualize the data, which aids in decision making. A GIS is a sophisticated

computer mapping and information retrieval system, consisting of three primary components: a powerful computer graphics program, a set of analysis tools, and one or more databases which serve as the data repository. All these components must be tightly integrated in order to establish an efficient system. In this paper we shall describe how all these components integrate in a semantic database environment.

2. Spatial Vector Data Set

The following data types are supported by vector data:

- *Point*: A single x, y coordinate that represents a geographic feature too small to be displayed as a line or area. Example: the location of a mountain peak or a building location on a small scale map
- *Multipoint*: Represents a set of points in a particular area of space. Example: All the parks in a county
- *Arc*: A string of x, y coordinate pairs that begin at one location and end at another. Connecting the arc's vertices creates a line. Each Arc starts and ends at a node and can define areas to the left and right of their direction of travel. Geographic features like roads, streams can be represented as arcs in vector data.
- *Polygon:* A multisided figure that represents an area on a map. Its aerial feature defined by the arcs that make up its boundary. Polygons generally represent the areas in space. Island, county, city etc. are some features that can be represented as polygons

Spatial vector data is modeled in layers, all defined by the same object space and coordinate system [2]. For example, the spatial representation of a city might include separate layers for outlines of political districts or socio-economic neighborhoods, business and domestic locations, and the maze of water, gas, sewer, and electric lines. Because all these layers share a common object space, they can be related through their spatial locations.

3. Image Storage in a Semantic Database

Modern databases can store more than textual data. Many commercial databases can store Binary Large Objects (BLOBs). The implication is that raster images, which are often used as back drop of GIS vector data, can also be stored in the database. Current usage does not usually exploit this feature of the database, as many see the raster image just as a file instead of a collection of records that are realm of the database domain. However, the fact is, image can be stored in database and there could be several advantages to this approach. First the image will be stored in a consistent centralized location. Second, many descriptive data about the image that are usually not stored (or stored sporadically as files where users cannot access them easily) can now be maintained by the database. For instance, if a variety of satellite and/or aerial images were used to complement the vector data of a GIS project, many descriptive data about the image would be useful in cataloging them. Information such as date of data acquisition, coordinates of the images, data provider, description of the image, plus others information can become very handy in allowing the user to judge the usefulness of an image for a particular task at hand.

It is in binary storage where Sem-ODB has a real advantage over traditional relational databases. The object-oriented framework employed in Sem-ODB allows a very efficient data storage and manipulation[3]. For instance, automatic compression can be applied to the image at the database level. This will reduce the storage space requirement, and the compression can be performed independent of the image format. This is advantageous because compressed files means reduced hard disk access. This, plus the optimized query from Sem-ODB, allows large images to be retrieved faster than many commercial databases. Preliminary tests showed very promising results.

To facilitate the needs of GIS functionality, we have implemented the following features in Sem-ODB:

- Data types are unlimited—strings can be of any length and we have developed techniques to represent numbers of unlimited length and precision.
- The semantic database is highly parallel. An efficient load balancing algorithm has been designed that will allow arbitrary chunks of data to be stored on different servers, optimizing the server performance. This has been successfully implemented on a parallel machine cluster Beowolf.
- Techniques for the efficient storage of spatial data to allow random access to the data, along with associated textual data, have been investigated. A simple

lossless algorithm has been adopted for testing purposes. Simple GIS systems illustrating random access to spatial data at high speed are being investigated and are in development. Those illustrating high speed access via workstations and ATM communications provide animation of the images with an excellent approximation to real time interactively generated movies. Those illustrating remote access at conventional speeds provide random access to spatial data via the World Wide Web. The display of complex spatial data using 3-D visualization and VRML facilities has been investigated and forms a part of our most complex GIS, which is still under development.

- We have developed compression algorithms for spatial data sets; storage on immediate access devices of substantial volumes of such data; bulk storage or "data warehousing" of huge volumes of spatial data; rapid random access and presentation of the data; and algorithmic manipulation of the data from the *raw* stored state to a visual presentation state ready for distribution.

- For storage of spatial data. Sem-ODB is able to store files as field data without any logically change to the data files, allowing the programs that are currently used to generate, process, and access these files to remain in operation. Towards this goal, we have enabled our database engine to store raw values of attributes of arbitrary length. e.g. a datum 2GB long. transparently fragmented and load-balanced among many disks comprising the database, with highly efficient access to any offset within this attribute value datum.

4. Representation of Spatial Vector Data in a Semantic Database

The source for spatial vector data used in this project is shapefile(ESRI-ArcView standard vector file format)[2]. To create a vector based map, the database needs to obtain the coordinates of a large number of points. Furthermore, the database needs to know how these nodes are connected to form line segments (arcs), and how these arcs in turn form polygons. In this section. we will outline a Sem-ODB schema for the storage of this spatial data set. Figure

1 shows the schema necessary for storing spatial vector data in Sem-ODB. Every feature in a layer has its own geometric bounding box. Categories are shown as rectangles, while relations between the categories are shown as solid arrows. The diagram makes it easy to understand the Sem-ODB schema and how the data is related. The diagram shows us that Points are grouped to form Lines, which in turn form Arcs, and Polygons.

For the efficient storage of spatial vector data we need spatial indexing. In spatial indexing each layer is subjected to quad-tree decomposition, creating a mosaic of covering tiles for the geometrically represented data. A binary representation of the tile is then stored in the spatial index. The information from several layers for a particular geo-location can be accessed efficiently from the covering tiles [6].

As spatial vector data is added to a database, the spatial index subdivides the tiles, preserving the spatial organization of the data. Database designers can specify the number of times a geometry should be subdivided to optimize the coverage with smaller and smaller uniformly sized tiles. Designers can also specify the maximum number of subdivisions for adaptive quad tree decomposition. Using either smaller fixed tiles or more variable sized tiles enables designers to choose a suitable representation of their spatial data.

5. Implementation

The basis for this project is TerraFly [4] (spatial raster data visualization tool), developed at Florida International University. For the integration of vector data with TerraFly. we reviewed several existing vector data formats (ArcInfo coverage. AutoCAD DXF, etc.) and selected the shapefile (ESRI-ArcView vector data format file) format. We decompose vector data of shapefiles spatially and store in a semantic database. As vector data also has external text attribute data associated with the spatial features. the corresponding attribute data are also stored in the same database for a complete vector database. An Sem-ODB schema shown in Figure 1 depicts the database design for spatial vector and attribute data.

The prototype for spatial vector data storage and overlay has been implemented in Java. At runtime. retrieval operations of vector data utilize the indexing technique addressed in

section 4. The retrieved vector data are displayed over raster data image (accessed from spatial raster database, TerraFly). The overall implementation is shown in Figure-2. A snap shot of the software is shown in Figure-3.

6. Conclusion

Previously, systems have been proposed for managing geographic data which add explicit support for spatial data types into the relational model, but such systems retain the widely recognized weaknesses of relational systems for handling the aspatial aspects of geographic applications, there by diluting the gains achieved as a result of incorporating spatial data types and operations. The solution to this dilemma can be found in a semantic database, which has the features to satisfy the spatial data storage and management via object-oriented techniques. Database performance has largely been a factor of database size and index efficiency. But with the modern indexing techniques, performance is a factor of the amount of data actually retrieved. The performance is optimized through the use of advanced quad tree mechanism for storage and retrieval of spatial data.

References

[1] Aref, W. G and Samet, H. Extending a DBMS with Spatial Operations. In Gunther, O. and Sheck, H. J., editors, *Advances in spatial databases*, SSD'91, LNCS 525, pp. 299-318, Springer -verlag, 1991.

[2] ESRI-ArcView, shapefile technical description,pp.1-10,1997
http://www.geocities.com/~vmushinskiy/fformats/files/esrishp.txt

[3] Naphtali, Rishe, A semantic approach to database design, chapters 1-3 pp. 1-147, 1996.

[4] FIU-HPDRC, TerraFly project documentation
http://hpdrc.cs.fiu.edu/_terrafly/terrafly.html

[5] Shashi, S., Sanjay C., Spatial database book project, University of Minnesota, pp.1-20, 1998.

[6] Petry, Fred, A multiple criteria based approach to performing conflation in Geographical Information Systems, pp. 5-30, Ph.D. dissertation presented in Computer Science, Tulane University, 1998.
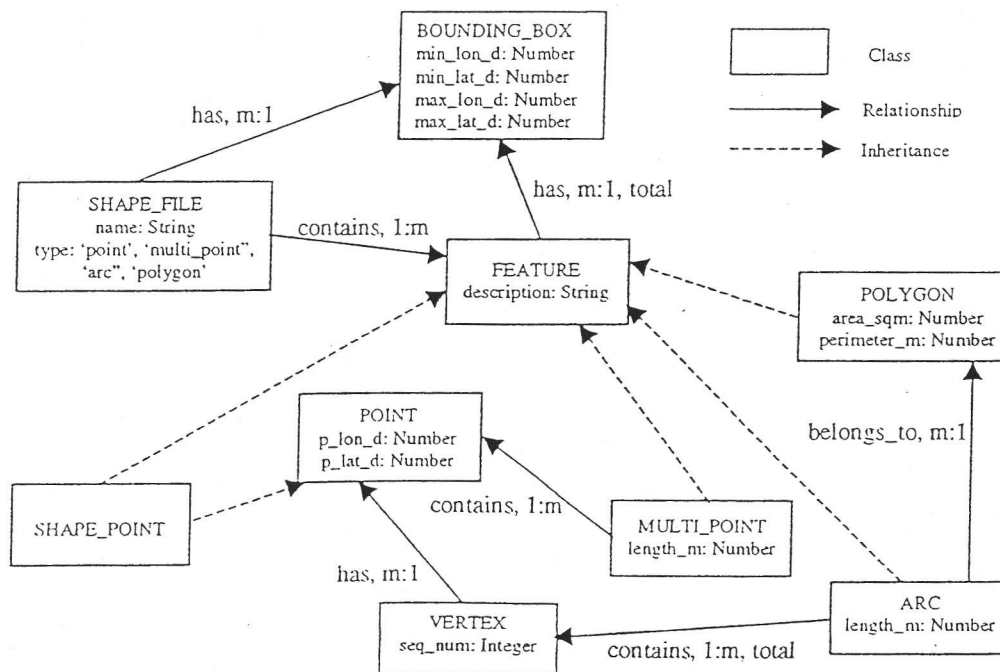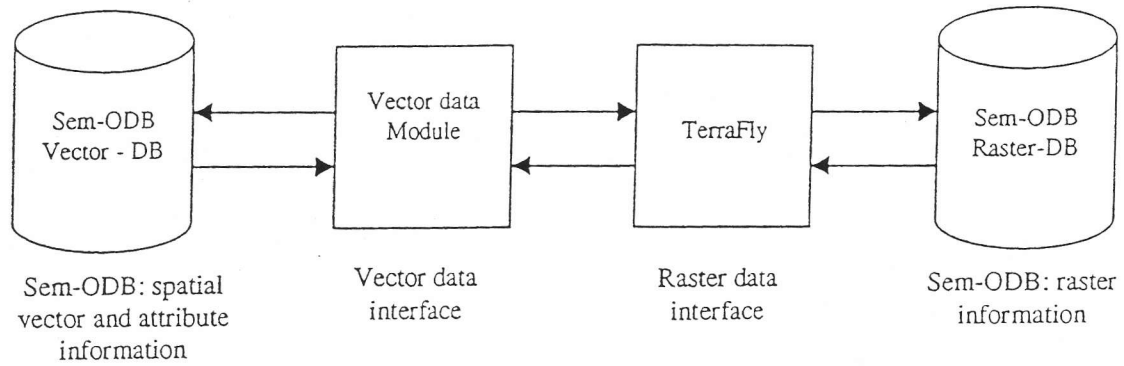
Figure-1: Semantic schema of vector data
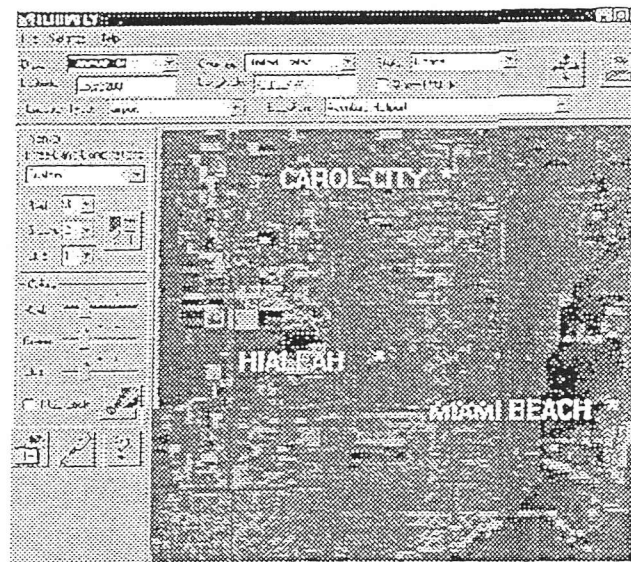
Figure-2: Integration of vector and raster data

Figure-3: Snapshot of TerraFly with raster and vector data