URC97019

Managing Data in a GIS Environment

Maria Beltran and Haris Yiasemis
Department of Computer Science
The University of Texas at El Paso
El Paso, TX 79968
{ mbeltran, haris }@cs.utep.edu

1 Introduction

A Geographic Information System(GIS) is a computer-based system that enables capture, modeling, manipulation, retrieval, analysis and presentation of geographically referenced data. A GIS operates in a dynamic environment of spatial and temporal information. This information is held in a database like any other information system, but performance is more of an issue for a geographic database than a tradition] database due to the nature of the data.

What distinguishes a GIS from other information systems is the spatial and temporal dimensions of the data [5] and the volume of data (several gigabytes). Most traditional information systems are usually based around tables ant] textual reports, whereas GIS requires the use of cartographic forms and other visualization techniques. Much of the data can be represented using computer graphics, but a GIS is not a graphics database. A graphical system is concerned with the manipulation and presentation of graphical objects whereas a GIS handles geographic objects that have not only spatial dimensions but non-visual, i e., attribute and components. Furthermore, the nature of the data on which a GIS operates makes the traditional relational database approach inadequate for retrieving data and answering queries that reference spatial data [.5].

The purpose of this paper is to describe the efficiency issues behind storage and retrieval of data within a GIS database. Section 2 gives a general background on GIS, and describes the issues involved in custom vs. commercial and hybrid vs. integrated geographic information systems. Section 3 describes the efficiency issues concerning the management of data within a GIS environment. The paper ends with a summary of the main concerns of this paper.

2 Geographic Information Systems (GIS)

2.1 Data

GIS data is handled in a single database or a collection of databases [4]. Like any other database, it must be secure, reliable, and consistent. Some special characteristics of GIS databases include:

- Spatial Data. Spatial Data is traditionally divided into two classes, raster and vector [1]. Raster data is structured as a two-dimensional array of cells or pixels. Each cell in a raster is addressed by its position in the array. A point may be represented by a single cell, and a connected area by a collection of contiguous cells. When a raster is stored with no compression, it can be extremely inefficient in terms of storage. The other type of data is vector data. A vector is a finite straight line segment defined by its end-points, and the locations of the end-points are given with respect to some coordinates of the plane.

- Data capture. Data capture involves two requirements. The first requirement is to provide the physical devices for capturing data external to the system and inputting it to the database. The second is to provide software for converting data to structures that are compatible with the data model of the database and checking for integrity of data before entry into the system. Geographic databases have a wider variety of sources and types of data than traditional databases. The main problem here is to get the data into a format that is acceptable by a particular GIS. Primary input devices used by GIS in addition to keyboard and voice recognition systems are as follows:

Remote sensing captures data by means of sensors on a satellite that provide measurements of reflectance or images of portions of the earth. The data is usually raster in structure.

– *Global* Positioning *Systems (GPS)* allow the capture of terrestrial position and vehicle tracking, using a network of navigation satellites. Data is captured as a set of point position readings and are in vector format.

Secondary data capture usual] y is from paper-based maps, The following devices are used:

Scanners convert an analog data source (e. g., a map) into a digital dataset in raster format.

– *Digitizers* convert an analog data source into a digital dataset in vector format.

Currently there is interest in new methods for raster-to-vector and vector-to-raster conversion.

- Data **retrieval**. Most interactions with a database are attempts to retrieve data [5], [4], [3]. A GIS allows real spatial processing to take place. Examples of spatial queries are:

  What is at a particular location? This may be clone by clicking the mouse at a particular location on the screen or by giving coordinates.

  – What locations satisfy these requirements? For example, find the names of all land areas that satisfy the following requirements : 1) less than average price for lan d, and 2) within 15 minutes drive of 110.

  Performance is a bigger problem for a geographic database than a general-purpose database because of the volume of data. Also, the nature of the data is often hierarchical (e.g., a point is a part of an arc and an arc is a part of a polygon) and this creates difficulties for traditional database approaches. Special storage structures and access methods are required.

- Data **presentation.** Tradition! databases provide output in the form of text usually in tabular form. They also may generate reports with charts and other graphical displays. A GIS requires a more sophisticated presentation of results, which might be multi d dimensional [5],[2]. output might be in the form of maps and other sophisticated forms. Graphics and visualization tools are a key component of a GIS and include tools for the creation, storage, and manipulation of models and images of objects. Graphic images are essential to a GIS, but these images also need a huge amount of space for storage. Multimedia computing presents new opportunities for GIS, but also creates new problems because of space issues. Storage and compression of such d ata is a major research area [5].

- Data **distribution**. The trend in recent technology is to move from centralization towards a distributed computer system [3] in which machines communicate through a network. As a consequence, data and database management systems (DBMS) are distributed. through the network. In this way, the reliability of the system is achieved because failure at one site will not mean failure for the whole system. Furthermore, distributed data may be natural and appropriate for a GIS because particular data may be associated with a particular site, e.g., details of local weather conditions may be better held at a local site where local control and integrity checks [3] may be maintained.

2,2 Analytical **processing**

Analytical processing is one of the major requirements for GIS[5],[6]. Some of the requirements include the following:

- Geometric/topological analysis: Most geographically- referenced objects have geometric or topological properties. Topological operations include adjacency and connectivity relationships. Geometric analysis would involve locating a spot in a region considering the distance from a place of reference to the spot.

- **Terrain** and **field** analysis: Terrain analysis is usually based upon datasets giving topographical elevations of point locations, i.e., degree and direction of slope. This would include analyzing the visibility between locations. Spatial fields arc variations of attributes over a region, or the topographical elevation over an area. Fields may be scalar (variations of a scalar quantity represented as a surface) or vector (variations of a vector quantity like wind-velocity). Field operations include slope analysis, view-shed analysis and path finding.

- Network **analysis**: A network is a configuration of connections between nodes. Application of network analysis in GIS may be found in many areas from transportation networks to utilities. Network operations include connectivity analysis, path finding and flow analysis, An example would be to provide a route in order to visit all attractions of a tourist region minimizing the time.

2.3 Custom vs. Commercial GIS

Many commercial GIS packages exist today. Some of the commercial packages, such as ARC/INFO, provide a programming language and interfaces that allow the user, for instance, to access an external relational database package giving the user some flexibility in organizing the system. An organization may decide to develop and program a GIS from scratch to meet a special need (called a custom *GIS* in this paper) or to create application programs that interact with commercial packages.

Due to the large amount of data, and to the nature of spatial data, a custom GIS sometimes is preferred when fast access of data is required. The data in such cases is stored in an application-dependent way to provide this capability. The disadvantage in this approach is that it has only limited query capabilities because there is no connection with an external relational database package. Also, most of the query operations are hard-coded, sacrificing generality and flexibility for better query performance.

2.4 Hybrid vs. Integrated GIS

GIS can be categorized into two groups according to their general architecture: hybrid and integrated [5],[2]. Suppose that we want to keep spatial and non-spatial data about. a particular piece of land. Usually spatial data describes location in two or more dimensions, For example, the geometry of the land with topological relationships might be kept in a map. The name, address, and owner information constitute the non-spatial data. These non-spatial data might he kept in a relational database. On the other hand, the spatially referenced data are not immediately compatible with a relational database and must be stored in a proprietary database. This is because in order to store spatial data in a relational database, each dimension must occupy a separate column, making spatial queries very time consuming.

The primary reason for having a hybrid architecture is the distinction between spatial and non-spatial data. Usually in hybrid systems (e.g. ARC/INFO) spatial data is stored in a set of system files, and non-spatial(attribute) data is stored in a relational database. In such systems. one part forms the graphics and spatial data engine, and the other part, handles the non-spatial data in tile database. The advantage of this approach is that the search performance problem is minimized, but of course with the loss of generality. This approach has the disadvantage that, the spatial data are handled outside the database and cannot take advantage of the capabilities of relational database technology such as integrity, security and reliability. Furthermore, due to the proprietary database in this approach, the exchange of data between different databases is complicated, if not impossible.

The idea behind an integrated architecture is to manage the data in a single database. The spatial data is handled in the same way as non-spatial. The problem with such a solution is that performance on retrieval of spatial data is poor due to the large number of relational operations that, are required to reconstruct the spatial objects. Keeping everything in a traditional relational database solves some problems but creates others. This might explain why most GIS systems are based on the hybrid architecture.

3 Data Management in GIS

'the large volume of GIS data makes the traditional relational database approach inadequate for retrieving and managing data. Spatial and temporal data with possibly two or more dimensions must also be referenced making data management more complicated than a traditional information system.

3.1 Data Retrieval and Storage

Data retrieval and storage within the traditional relational model is based on indexing, but the following subsection shows through an example why it is not appropriate for spatial data. Another structure, called a quadtree, is introduced as a solution to this problem. An example is given using Oracle7 with the spatial option. See [4] for a complete reference on relational databases.

3.1.1 Indexing in Traditional Relational Databases

Traditional relational database approaches are inefficient for retrieving spatial data and answering queries that include spatial conditions because they do not take advantage of the ordering of data in two or more dimensions [5] ,[2]. For example, consider a segment of a database that contains information about various places of interest in a particular city. A segment, of the table might look as follows :

Id	Site	East	North
1	City Museum	15	60
2	Special Events	30	67
3	Sun Bow] Stadium	45	20
4	Civic Center	34	**20**

Now, consider the following point (example 1) and range (example 2) queries:

1. Retrieve any site at location (30,67).

2. Retrieve any site in the rectangular area defined by (10,10) and (35,70)

Using the traditional approach, it is reasonable to have two indices for the two spatial coordinate fields of our table. The indices would look as follows :

East	Site
15	City Museum
30	Special Events
34	Civic Center
45	Sun Bowl Stadium

North	Site
20	Civic Center
20	Sun Bowl Stadium
60	City Museum
67	Special Events Center

To answer the first query, we would probably do a binary search of the *East* index to locate records whose first coordinates have value 30. We then must, go to the original table to check if the second coordinates have value 67 and retrieve the records for which we have a match. For the second query, we would have to do a range search on [10,3.5] on the *East* index, where 10 refers to lower bound of the *East* index and 35 refers to the upper bound of the *East* index, giving us a list of pointers to the original table. For each pointer in the list, the specific record must be accessed and the *North* value checked in order to see if it falls in the range [10,70] for the *North* index. If it falls in that range, we retrieve the record.

The problem with this approach is that only one of the indices is used in these retrievals. This kind of indexing would be very inefficient for a large database consisting of several gigabytes of data, which is usually the case for a GIS. An indexing scheme is needed that takes advantage of the ordering in two or more dimensions.

3.1.2 Quadtrees in Spatial Databases

A structure that can be employed to overcome the problems of indexing multidimensional data is the quadtree [5],[2]. This structure is a leveled tree where all non-leaf nodes have exactly four descendants, For example, a two-dimensional region is decomposed by recursively subdividing its regions into four equal-sized quadrants. The decomposition is applied to each quadrant until the desired degree of resolution is achieved. Quadtrees arc stored in a leveled-tree data structure with the root at the top level. For each non-leaf node, its four constituent quadrants are represented by its four descendant nodes. A quadrant where no further subdivisions is required is stored as a leaf node.

Oracle7 with the spatial option uses the idea of quadtrees to handle multidimensional data within the relational data model taking advantage of the customary relational query capabilities without using the traditional relational DBMS indices. To do that, Oracle7 uses an encoding technique that maintains the

dimensional organization of data. Records that reference information that are geographically near to each other arc logically stored near each other. This encoding technique makes use of a new data type called the Helical Hyperspatial Code (HHCODE)[2] that allows the encoding of multiple dimensions into a unique value that is stored in a single column of a table,

The figure below shows a two-dimensional decomposition of North America. The map is divided into four equal-sized quadrants, and then each one is divided recursively into four according to what precision we want for a specific location. If a greater level of resolution is needed, the quadrants will keep subdividing. Only those quadrants in which needed data exists will keep subdividing. The four quadrants at each subdivision are given one value from 0 to 3.

The HHCODE contains a string of values from O to 3 describing the specific object that is represented according to its position and level. By level we mean the degree of decomposition, i.e., level O is the whole non-decomposed region, level 1 is the first level of decomposition, and so on. When a subdivision occurs, the number of the top-level quadrant is appended to the number of the new quadrants created by the subdivision.

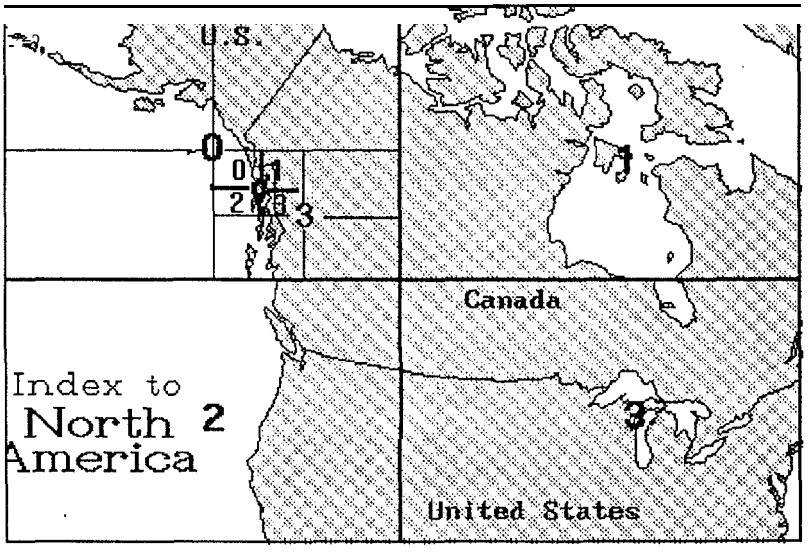

Figure 1: A decomposition of a map.

In the above figure, the data would be encoded as follows:

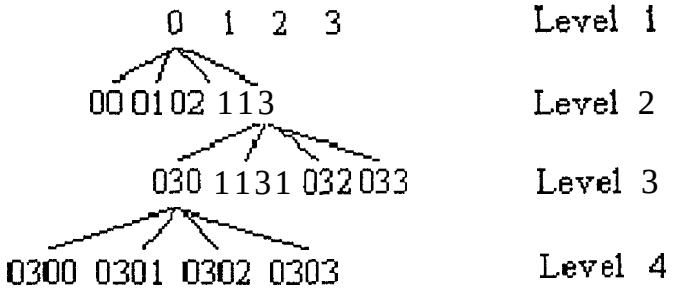

Figure 2: The quadtree representation of the decomposed map.

Note from the generated HHCODE in the example that the records containing data that are near each other geographically have common substrings. As we described in the previous section, traditional relational databases with indices on spatial data do not maintain the dimensional organization of the data. This problem is overcome by using the HHCODE, and fast retrieval of records for spatial queries is achieved.

3.2 Security

Security [5], [4],[3] is a primary concern for any database. Unauthorized access to the data should be prevented and different levels of authorized access should be allowed. For a GIS specifically, our concerns may be increased because many GIS due to their nature have distributed databases. Unauthorized access from the Internet should be prevented.

3.3 Integrity

The integrity [5],[4],[3] of data must be enforced. Data in the system should be correct and consistent with each other, e.g., data in one tile referencing an object, and data in another file referencing the same object must be consistent. Integrity over distributed data in different locations is more difficult to be enforced. Also this is true for a hybrid GIS, where data is not kept in a single database, but spatial and non- spatial data are handled separately. Precautions must be taken by the designers of the system so the integrity of the system in any case is preserved.

3.4 Data Formats

A GIS database, as we explained in section 2, must handle a wide variety of sources and types of data. The main problem here is to get the data into a format [1],[5] that. is recognizable by a particular GIS. GIS designers should be aware of the source that is **used** for capturing the data in order to avoid incompatibilities with particular data formats. Because graphic data is used extensively in a GIS, an important issue here is the space needed to store this information. Usually data in raster format occupies more space than data in vector format. Various compression techniques exist and can be employed to reduce the space required for storage of some data formats. For more on this, a good reference is [1],

4 Summary

In this paper, we gave a general background on geographical information systems and the principles on which they are based. We looked into the issues relating to the managing of data within a GIS environment, and we explained the problems with the traditional relational database approach using indexing on spatial multidimensional data. Using an example, we explained why performance with this approach is poor, We also described bow this problem can be solved using a different structure, the quadtree. An example on how this can be used was given using the Oracle7 HHCODE data structure. Also, the general principles of data security, integrity, and the handling of data formats were discussed.

Acknowledgments. This work was supported by NASA under contract NCCW-0089.

References

[1] B. Fortner., *The Data Handbook.* Santa Clara, CA: Telos, 1995.

[2] G. Gwendolyn, C. Kristian, M, Bradley, J. Rawlings., *Oracle7 Multidimension, Advances* in *Relational* Database Technology *for* Spatial *Data Management.* Redwood Shores, CA: Oracle Corporation, 1995.

[3] H. I?. Korth and A. Silberschatz, *Database System Concepts. New* York: McGraw-Hill, 1991.

[4] J. Ullman, *Principles of Database and Knowledge-Base Systems,* Volume 1. Rockville, MD: Computer Science Press, 1988.

[5] M. F. Worboys, *GIS: A Computing Perspective.* Bristol, PA: Taylor and Francis, 1995.

[6] L. Worral, *Spatial Analysis and Spatial Policy Using Geographic Information Systems.* London: Belhaven Press, 1991.