

Topic Series 15
GIS Data Structures, Topology, Relational Databases, Accuracy Issues

I. GIS Data

Basics of Computer File / Database Structures

A goal of GIS is to represent and store the graphic entities of mapped information along with relevant attributes in such a way to make all the data easily retrievable and manipulatable. This is done by taking advantages of the ways computers handle data in logical fashion through file and database structures. A brief overview of the ways computers can handle data is offered here.

(Overhead)

Simple List - The most basic file structure is a **simple list**. In this file structure, there really is no absolute ordering of the data. The data occur in the file in essentially the same way in which they were entered. Simple lists may start out in logical fashion but whenever modifications are made, they rapidly get out of order because new data are appended to the end of the list.

Ordered sequential files can be thought of as a rolodex in which we keep everything in alpha-numeric order. As new data are added, the file is restructured (sorted) to maintain that order.

Indexed file structures provide pointers to more efficiently search data. The most efficient system is to develop an index that is based on a commonly searched attribute in the database.

A collection of files that are used for complex information organization is called a **database**. The software used for management and manipulation of databases is called a **database management system (DBMS)** (this is what GIS is). There are three types of DBMS's relevant to GIS: **hierarchical, network** and **relational**; the latter is most often the form of system adopted for complex GIS operations. Lets examine GIS database structures in general format and functionality.

The GIS Database

A **GIS database** is composed of all of the **geographic/spatial information** (maps, imagery) and associated **attribute information (tables, reports)** that are linked in such a way that we can extract either the spatial or attribute information by requests based on the location or characteristics of the data features either singly or as related to other features.

(two overheads)

There are two types of data in a GIS: **spatial and attribute** (tabular).

Attribute Data

These are usually **data tables** that contain information about the spatial components of the GIS themes. These can be **numeric** and/or **character** data such as **timber type, timber volume, road size, well depth**, etc. The attributes are related back to the spatial features by use of unique identifiers that are stored both with the attribute tables and the features in each spatial data layer.

Attributes can be either **qualitative (low med. high income)** or **quantitative (actual measurements)**.

The database allows us to **manipulate information** in many ways: from **simple listing** of attributes, **sorting features** by some attributes, **grouping by attributes**, or **selecting** and singling out groups by attributes.

The database software that we are using in lab is a **relational database** structure. This means that we **cross reference** feature attributes to their spatial definitions based on some **common attribute** stored in the data table for the attributes and graphics. We can select one or more graphic features by use of a **query** of some characteristic of interest in the feature attribute table of the database. And, since the reference from graphics to attributes works both ways, we can **select one or more spatial graphic** features on our computer screen and have the software give us the **associated attributes**. The relational qualities of the database go even further than these simple examples. We frequently attach other **external tables** to our original data sets and relate them to the existing data by common attributes.

Benefits of Relational Database Approach

(overhead)

1. Convenience - easy access
2. Reduce redundancy - cross referencing reduces repeated data entries
3. Shareable - cross organization sharing, multiple uses of data
4. Flexible analysis options - multiple analysis options, expansion of analysis capability
5. Standardization - controlled formats for data entry / manipulation reduce inconsistency and errors.

Spatial data

Graphic Data Structures in GIS

(overhead)

This discussion will focus on spatial data representation on computer systems, the pros and cons of different data structures, and some general comments on use of the data. First consider the following definitions:

Map: is a paper representation of features that can be depicted based on their spatial relationships. Here we will be dealing with maps of the earth's land surface.

Coverage: is the digital form of all or part of a map. In our work, a coverage is usually depicts one major **theme** (ArcView term) such as landuse, roads, streams. The more specific we make the coverage, the more flexibility we build into the analysis capabilities of the GIS.

Data Structure: is the form (format) of the data as stored and manipulated on the computer. In order to get the data into the computer, it must be **scanned** or **digitized**, **edited**, and **converted** to the final form for the GIS (**raster** or **vector**).

Raster vs. Vector

Raster data - cell based, in simple terms this is a data set of **rows and columns** of numbers where the position (column, row; x, y) is the location and the number (or character) is a code representing some attribute of the data.

Figure 15-1. A simple raster data set with three attributes

1	1	1	2	2	2	2	3	3	3	3	3
1	1	1	1	2	2	2	2	3	3	3	3
1	1	1	1	2	2	2	3	3	3	3	3
1	1	1	2	2	2	3	3	3	3	2	2
1	1	1	2	2	3	3	3	3	2	2	2
1	1	2	2	2	3	3	2	2	2	2	2
1	1	2	2	2	3	3	2	2	2	2	2
1	1	2	2	2	2	3	2	2	2	2	2
1	1	1	2	2	2	2	2	2	2	2	2
1	1	1	1	2	2	2	2	2	2	2	2

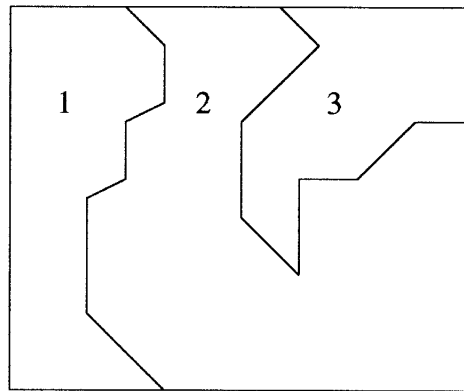
The **size of each cell** is generally determined by some type of **header record** for the file that describes the coordinate of the origin (row 1, column 1) of the file and the x,y dimensions of the cells in the file.

Vector data - objects represented in vector data structures are determined by an **x, y location** in coordinate space. Vector data sets are composed of single **points, lines, polylines or arcs** (connected string of points), and **polygons** (series of coordinates that define an enclosed region).

arc = series of line segments bounded by nodes at end points and vertices.

topology = the way a vector GIS uses points, lines, and polygons to represent map features.

Figure 15-2. A vector data layer with 3 polygons.



Vector Data Characteristics

The most common vector data structure utilizes arc/node topology.

Topology can be defined as the organization of spatial relationships between features in a GIS. In layman's terms, topology is the way a GIS “knows”:

- 1) **where a feature is** in relation to other features,
- 2) **what parts of different features are shared** (points, lines, nodes), and
- 3) **how features share connectivity** (gives us ability to move between features in network applications).

Topologic data structures help insure that information is not unnecessarily repeated. The database stores one line only in order to represent a boundary (as opposed to two lines, one for each polygon). The database tells us that the line is the “left side” of one polygon and the “right side” of the adjacent polygon.

The three aspects of topology that are important in representing spatial relationships are:

- 1) **adjacency** - shared boundary
- 2) **connectivity** - shared node in arc-node topology
- 3) **containment** - accounts for polygons within polygons “islands”

The software in our GIS creates a database that keeps track of the relationships as lists of shared features. A simple map may be composed of land cover polygons. The polygons are composed of “**chains**” (**we'll call them arcs** to be consistent with arc/info). Some of the arcs are shared by polygons, some are not. The database structure is designed to keep a list of all arcs and how they relate to the formation of each polygon.

Vector Database Links

The connectivity documented by topology in a database lets us do operations that define bridges or paths from feature to feature. Relational query capabilities combined with topologic data structures give us

the ability to sort through complex logic to identify sites of interest. Lets derive an example from **multi-use management** of forest land.

We have timber that we are interested in harvesting in an area that is fragmented forest and rangeland. We want to make sure that we provide for a mix of conditions for wildlife. Therefore, we use GIS to query the database for those stands that meet our harvesting specification (we're after sawtimber only), that have: 1) immediately adjacent stands of the **same type**, 2) younger stands with **thick cover**, 3) and **rangelands**. This approach may not be optimal for maximizing timber production but may meet the objective of the particular landowner.

A big application in this arena is **least-cost path analysis**. Consider the problem of the **911 system**. You can build a GIS that has the locations of emergency vehicles, all roads, and addresses along the roads. Getting from the emergency center to someone in need is not just a matter of connecting the roads. We also build into the database road surface (dictates speed), dangerous intersections, traffic congestion by peak hours. These attributes of the roads would be used with total length to determine the route to take that requires the shortest time (not always the shortest distance).

Raster Data Characteristics

Raster data forces all features to be represented in grid cells with specific dimensions. Some areas are fairly well represented by raster data structure, provided the raster structure aligns with the feature orientation and the size of the features.

Irregular features are not well represented as to their true size and shape.

Linear features are represented by rasters being turned on and take on a jagged appearance. This is both annoying and inaccurate.

Coding: another problem in raster data is deciding how to code each cell. You can code by **majority rule** or **cell center**.

Resolution: we can increase the accuracy of spatial representation of features by increasing the resolution of the raster data (make the cells smaller). This is done however, at expense of increased storage space and processing costs on the computer.

Raster Storage Structures

(overhead)

There are numerous ways raster data (**pixels**) are stored. Two of the most common are as **run-length encoded** (a form of compression) and **quadtree** (hierarchical).

Run-length encoded: lets say the first row of a raster data set looked like this:

22222333334455222233

One form of run-length encoding would store the data as data pairs with the first number being the

repetition factor and the second the data value. The above line would code as: 526324254223. Some simple image compression algorithms operate like this.

Quadtree data structures: are used to represent data in layers that are build successively from pixels of the previous layer. A pixel in each higher layer is derived from the average or majority of 4 pixels from the lower layer. This is not a data reduction technique but provides a way to store data for rapid retrieval on display devices (the smallest layer for the display size is used for the display).

Raster / Vector Conversion

(overhead)

Modern GIS software provides for conversion between raster and vector data structures.

1. digitize data in vector format then convert to raster for ease of modeling
2. convert vector to raster for **printing, plotting**
3. convert raster to vector to utilize **topologic data structures** (stream modeling)
4. convert to be **compatible** with other data types in system

Raster Advantages

(overhead)

- A. **simple data structure** for storage (rows and columns)
- B. easy to analyze and **compare to other rasters** (differencing operations)
- C. form of imagery
- D. modeling applications much easier to program and implement

Raster Disadvantages

1. spatial inaccuracy
2. resolution
3. large **storage requirements** for simple data (**note that complex landscapes can actually take more storage with vector data than raster**).
4. general perception that maps are based on vector data (points, lines polygons) because this is how we tend to visualize features on the earth.

Vector Advantages

(overhead)

- A. More **what we expect** as map data (points, lines, polygons)
- B. **better resolutions than raster** generally on detailed landscapes
- C. generally better **spatial accuracy**
- D. advantage of **topology** (can represent connectivity and interrelationships)

Vector Disadvantages

1. **difficult to manage** on a computer
2. **slow to process** complex data sets on low-end computers
3. **more costly to use given the previous**

The combination of the spatial and attribute data along with the creation, editing, data retrieval, and output capabilities is what makes up the sum total of a GIS.