Representing Geography

Data models and axioms
Road map

Representing the real world
Conceptual models: objects vs fields
Implementation models: vector vs raster
Vector topological model
Surfaces and networks
The contents of a spatial database represent a limited view of reality (mandates):

The world is infinitely complex

The spatial database is *representation* of a *model* of reality

**Ontological** and **epistemological** issues

Representing the real world in a GIS
What kinds of objects exist in the real world?

If I tell you that in the figure on the right there are two cities (Vancouver and Burnaby), what exists that you cannot see?

Representing the real world
Some things you can hold, are clearly delimited. Other things are known through measurements, are infinite in scope.

Representing the real world
Sometimes, the distinction between discrete and continuous is not very clear.
Representing the real world

Computers are good at storing discrete spatial data, but bad at storing continuous spatial data (since everything in a computer is ultimately represented as a binary [0,1] number) (also, fractals)
GIScientists have been developing a *conceptual* model of space, independent of the way the data is *physically* represented in the computer:

**Object** or entity view
“empty space littered with objects” (points, lines or areas)

**Field** view
value is defined for every location

**Objects versus Fields**

Can you transmute the concepts?
Examples of objects

Vancouver / Burnaby Census Tracts

3-D view of Prague
Examples of fields

Elevation (a DEM)

Groundwater nitrate concentration
Implementing, in a computer, valid representations of fields and objects requires a set of rules. These rules will depend on the particular (physical) data model selected.

(aka: the spatial database is representation of a model of reality)

These rules convert (conceptual) geographic models into discrete (physical, computer-based) representations.

Data model implementation
What are the two dominant models?

GIS data models
Divides the study area into square cells (a grid)
Register the corners of the grid to the Earth
Represent discrete objects as collections of one or more cells

Represent fields by assigning attribute values (NOIR) to cells

More commonly used to represent fields than discrete objects

The raster data model
The raster data model

Legend

- Yellow: Mixed conifer
- Orange: Douglas fir
- Teal: Oak savannah
- Gray: Grassland

Raster representation: each color represents a different value of a (e.g.) nominal-scale field denoting land cover class.
Pixel size

The size of the cell or picture element, *defining the level of spatial detail*

All variation within a pixel is lost (mixed pixel problem)

Assignment schemes

The value of a cell may be an average over the cell ($\bar{x}$), a total within the cell ($\Sigma$), the commonest value in the cell (*mode*), based on a priority (ensure that rare things get recorded), or a boolean value (present [1] or not [0])

It may also be the value found at the cell’s centralpoint (centroid). (There are more rules.)

The raster data model
Regardless of the assignment rule used, ambiguity will always be present.

Mixed pixel problem
The raster data model is a member of a larger group of **field models** known as tessellation data models.

Grid or raster
Hexagonal
TIN
Quadtree

All completely cover the space, and therefore can be used to represent fields (but not all are raster-based models)

**Tessellation data models**
Tessellation models

Hexagons
Spatial ecology
Quadtree

Cells
Real world objects are represented as points, lines and areas
Points identify locations (0 D)
Lines connect points (1 D)
Areas (polygons) consist of connected line segments (2 D)

The vector data model
The vector data model
Objects are defined by their $(x,y)$ coordinates in a coordinate system (spherical: lat, long or planar: $x,y$)

Precision of coordinates virtually infinite (only machine-dependent) (resolution)
but: accuracy most often limited!

$E.g.$, Distance $A$ to $B$ is given as $1458.394958397$ m
Data was digitized off of a 1:20,000 scale map. Is that distance meaningful? (Significant Digits)

The vector data model
Resolution: raster & vector

Some thoughts
**Precision**: the smallest difference between adjacent positions that was recorded and stored (e.g., were distances measured to the nearest m or mm?). (Also applies to attributes)

**Accuracy**: the freedom from error (what is the truth?)

**Uncertainty**: now the more commonly-used term, as it represents the overall ambiguity (think of the map scale)

Which demonstrates accuracy? Which demonstrates precision?

**Precision versus Accuracy**

A good review of the terms
Conceptual models

- Objects (e.g. spot hts)
- Fields (e.g. Elevation)

Implementation models

- Vector (e.g. Contours, TIN)
- Raster (e.g. DEM)

Transmutation
Road map
Representing the real world
Conceptual models: objects vs fields
Implementation models: vector vs raster
Vector topological model
Surfaces and networks
Levels of GIS data model abstraction

Increasing Abstraction

Reality

Conceptual Model

Logical Model

Physical Model

Mandates

Data structures

Human-oriented

Computer-oriented

Inset diagram:

Building

Room

Bedroom

part-of

feature-of

is-a
Object representations

Points

Lines

Polygons
What are some of the ways we can store fields in a GIS?

Common field representations
There are three main vector data structures:

Simple (or spaghetti) data structure

*No intelligence, duplication of data (within a layer)* \(\text{(KML)}\)

Point dictionary

*No intelligence, no duplication of data (within a layer)*

Topological structure

*Intelligence, no duplication of data (within a layer)*

Vector data structures
Each point, line, or polygon is stored as a record in a file that consists of that entity’s ID and a list of coordinates that define geometry.

For Points:

<table>
<thead>
<tr>
<th>ID</th>
<th>Coordinates</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2,3</td>
</tr>
<tr>
<td>2</td>
<td>5,5</td>
</tr>
</tbody>
</table>

Spaghetti Vector Data Model
Each point, line, or polygon is stored as a record in a file that consists of that entity’s ID and a list of coordinates that define geometry.

For Lines:

<table>
<thead>
<tr>
<th>ID</th>
<th>Coordinates</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(0,1), (3,4), (5,6)</td>
</tr>
<tr>
<td>2</td>
<td>(3,1), (5,2), (4,3)</td>
</tr>
</tbody>
</table>

Spaghetti Vector Data Model
Each point, line, or polygon is stored as a record in a file that consists of that entity’s ID and a list of coordinates that define geometry.

**For Polygons:**

<table>
<thead>
<tr>
<th>ID</th>
<th>Coordinates</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(2,4), (4,3), (3,6), (2,4)</td>
</tr>
<tr>
<td>2</td>
<td>(3,1), (5,2), (4,3), (3,2), (3,1)</td>
</tr>
</tbody>
</table>

Spaghetti Vector Data Model
• **Advantages**
  
  • simple
  
  • efficient for display and plotting

• **Disadvantages**
  
  • inefficient for most types of spatial analysis, especially for generalization

**Spaghetti Vector Data Model**
Representing the real world

Conceptual models: objects vs fields

Implementation models: vector vs raster

Vector topological model

Surfaces and Networks
Records coordinates (x,y) of spatial features and encodes spatial relations (i.e., which arcs are connected to a node, which polygons lie on either side of an arc, which arcs make up a polygon)

Also called “arc-node” data model

arc = line
node = end-point of a line, or a point where two or more lines connect
Planar Enforcement:

- No two individual features can overlap.
- There are no ‘holes’ or ‘islands’ that are not themselves features.
- Every feature is represented as a record in the attribute table.

4 records – Not topological

6 records - Topological

Topological data model
Spaghetti: can encode as 2 or 3 polygons (and have 2 or 3 records in the attribute table)

Topologic: must be encoded as 3 polygons (and have 3 records in the attribute table)

Topological data model vs. Spaghetti
Topological data model
Can quickly answer these questions:
which roads connect to the central square?
which roads do I take to get from here to the hospital?
what are the fertility rates in the neighboring districts?

Topology: adjacency, connectivity, overlap, intersect
Advantages
- stores spatial relations explicitly
- spatial analysis can be done without accessing coordinates
- Very useful for data validation (if topological rules are established)

Disadvantages
- more complex data structure
- topology needs to be re-established after each update

Preferred system for high-end systems

Topological data model

ESRI’s help files on topology
Road map

Representing the real world
Conceptual models: objects vs fields
Implementation models: vector vs raster
Vector topological model
Surfaces and Networks
Two additional spatial entities should be introduced since they extend the basic properties of points, lines, areas and raster cells.

For some groups / agencies, these are the most important routines / data structures in a GIS.

These are:

- Surfaces
- Networks

Two special spatial entities
As mentioned before, surfaces are examples of field data.

There are both raster and vector approaches to storing / presenting surface data.
Surface presentations

3D?
2.5D

ArcGlobe
Remote sensing
   Radar (grid)
   Lidar (dense points)
Stereo aerial or satellite imagery (grid)
   Photogrammetry (softcopy) (points, lines) (Phodar)
   Orthophotos
Field collection
   Surveying (leveling, total stations) (points)
   GPS (points, lines)
Existing data
   paper maps—contours—digitization (lines, points)

Surfaces — data collection
Surfaces – data storage

Grid data – store as a raster

Points, lines – can store as is, but generally would want to create a more intelligent data structure.

TINs (Triangulated Irregular Networks) were developed to intelligently store vector surface data. TINs are based on a Delaunay triangulation or constrained Delaunay triangulation.
Quality TINs are built using VIPS (very important points, such as peaks, pits) and break lines (cliffs, roadways, rivers, lake shores).

Triangulated Irregular Network (TIN)
A topological model that stores elevations at the nodes, and associates slope with the edges and Δ faces.

Threading a contour
TINs can, most often, represent a complex surface with much less data than a DEM, since in a DEM an elevation must be known for every cell in the DEM.

The ‘information’ associated with every cell in a high resolution DEM can, in fact, be very low if there is little topographic relief. (Think spatial autocorrelation [e.g., a lake]).

However, deriving drainage networks, shortest paths, etc., are much easier to compute using DEMs.

Surfaces – TINs vs DEMs
Converting raster data structures (DEM$s$) to vector data is both easy (creating contours) and complex (creating quality TIN$s$)

Converting vector data structures (TIN$s$) to other vector structures (contours) and to raster structures (DEM$s$) is easy

Surfaces – data conversion
Networks are another special case of a vector topological data structure. Networks can be looped (e.g., bus routes) or radial (e.g., rivers). Special attributes are associated with Arc – Node – Arc relations. Turn tables (can you turn left/right, go straight); Stop, no stop. Arcs Directionality (one way, two way, speed limit).
Networks

Used for route analysis
- Couriers
- Emergency vehicles
- Garbage collection
- Vehicle navigation systems

Used for hydrological analysis
- Flow of water through a river system
Networks

Location-allocation

Shortest path
## Networks

### Planar enforcement?

<table>
<thead>
<tr>
<th>Situation</th>
<th>Representation</th>
<th>Turntable</th>
</tr>
</thead>
<tbody>
<tr>
<td>U-Turn</td>
<td><img src="image" alt="U-Turn Diagram" /></td>
<td>NODE# 6  \rightarrow 20  \rightarrow 7  \rightarrow 9  \rightarrow 8  \rightarrow 6  \rightarrow 20  \rightarrow 7  \rightarrow 9</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FROM 20  \rightarrow 6  \rightarrow 6  \rightarrow 180  \rightarrow 20</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TO 6  \rightarrow 6  \rightarrow 9  \rightarrow 9  \rightarrow 8</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TIME IMPEDANCE (seconds) 20</td>
</tr>
<tr>
<td>Stop sign</td>
<td><img src="image" alt="Stop sign Diagram" /></td>
<td>NODE# 6  \rightarrow 20  \rightarrow 7  \rightarrow 9</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FROM 20  \rightarrow 6  \rightarrow 7  \rightarrow 0  \rightarrow 15</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TO 20  \rightarrow 6  \rightarrow 8  \rightarrow 90  \rightarrow 20</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ANGLE 20  \rightarrow 6  \rightarrow 9  \rightarrow 10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TIME IMPEDANCE (seconds) 20</td>
</tr>
<tr>
<td>No Right Turn</td>
<td><img src="image" alt="No Right Turn Diagram" /></td>
<td>NODE# 6  \rightarrow 20  \rightarrow 7  \rightarrow 9</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FROM 20  \rightarrow 6  \rightarrow 9  \rightarrow -90  \rightarrow -1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TO 20  \rightarrow 6  \rightarrow 7  \rightarrow 0  \rightarrow 5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ANGLE 20  \rightarrow 6  \rightarrow 8  \rightarrow 90  \rightarrow 10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TIME IMPEDANCE (seconds) 20</td>
</tr>
</tbody>
</table>
Networks

An extended topological database model
Network analysis

Vector or raster
How real can we be?
Thinking of space (O/F)
Putting space into a computer (R/V)
Structuring space (Topology)
Adding complexity (Surfaces and Networks)

Looking back