Chapter 3: Maps as Numbers

- 3.1 Representing Maps as Numbers
- 3.2 Structuring Attributes
- 3.3 Structuring Maps
- 3.4 Why Topology Matters
- 3.5 Formats for GIS Data
- 3.6 Exchanging Data

Maps as Numbers

- GIS requires that both data and maps be **represented as numbers**.
- The GIS places data into the computer's memory in a **physical data structure** (i.e. files and directories).
- Files can be written in **binary** or as **ASCII text**.
- Binary is **faster to read and smaller**, ASCII can be **read by humans and edited** but uses more space.

Binary Notation

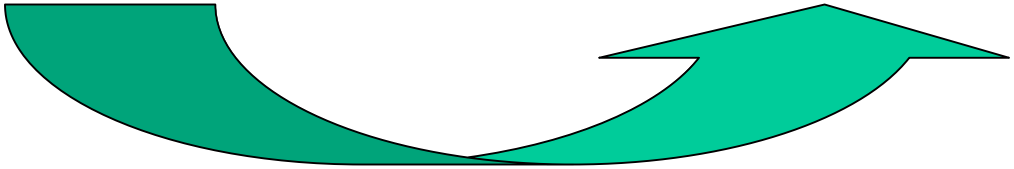
•Everything is represented as 0s and 1s in a computer. These two-state forms correspond to yes/no, on/off, open/closed

	Binary
1 digit	0, 1
2 digits	00, 01
	10, 11
3 digits	000, 001
	010, 011
	100, 101
	110, 111

	Decimal
1 bit	0,1,2,…9
2 bits	00, 01,…
	97, 99
3 bits	000, 001,
	002, 003,
	…
	998, 999

One to one correspondence

Decimal	Binary
0	0
1	1
2	10
3	11
4	100
5	101
6	?



Binary Notation

Decimal:

$72{,}479 = 70{,}000 = 7\times10^4$

$2{,}000 = 2\times10^3$

$400 = 4\times10^2$

$70 = 7\times10^1$

$9 = 9\times10^0$

10^4	10^3	10^2	10^1	10^0

Binary:

2^4	2^3	2^2	2^1	2^0
1	0	1	0	0

$1\times 2^4 + 0\times 2^3 + 1\times 2^2 + 0\times 2^1 + 0\times 2^0$

$= 16 + 0 + 4 + 0 + 0$

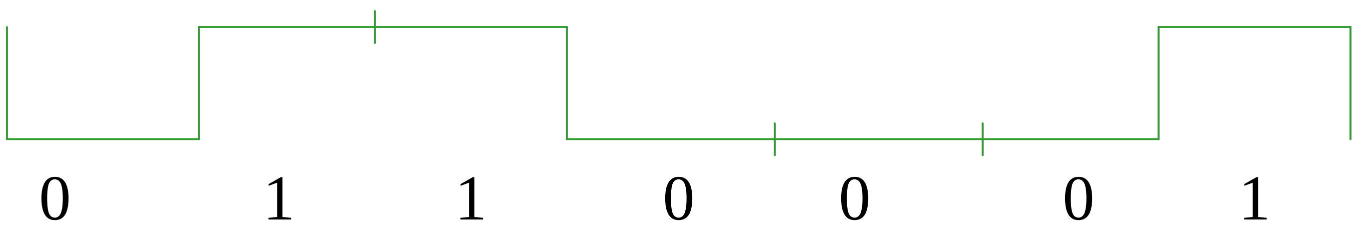
$= 20$

Note: In binary

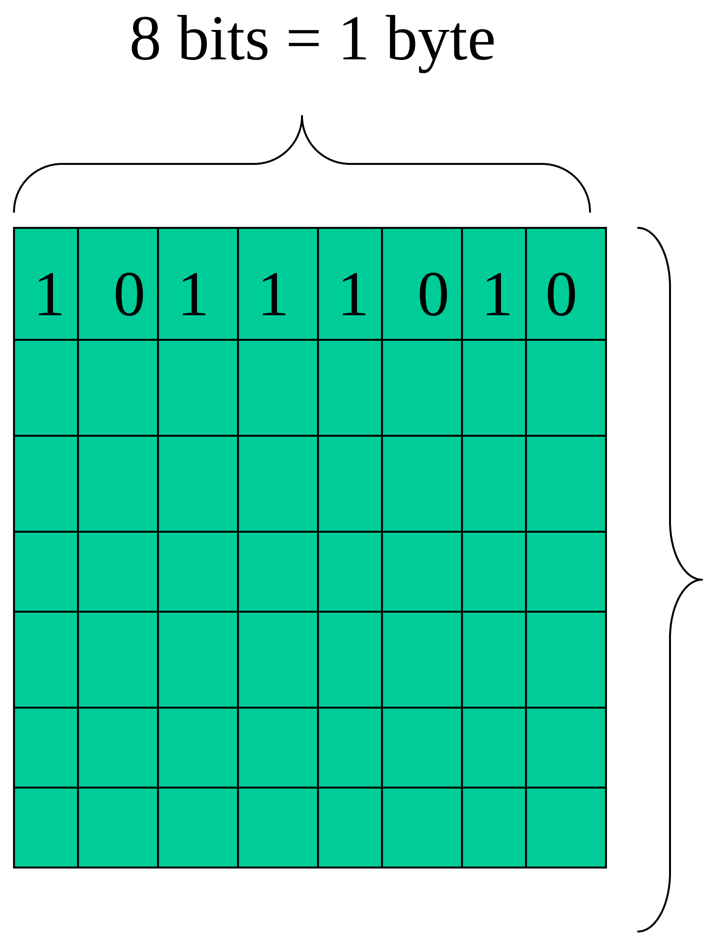
$$\begin{array}{r} 1010 \\ +\quad 110 \\ \hline 10000 \end{array}$$

Electronic Representation of Bits

- Binary information can be represented by electrical pulses passing along wires with two different voltage levels as shown in the following sequence of pulses



Bits and Bytes



1 bit = 1 binary digit
1 byte = 8 bits

1024 bytes = 1 Kb
1024 Kb = 1 Mb
1024 Mb = 1 Gb
1024 Gb = 1 Tb
1024 Tb = 1 Pb

ASCII Encoding

•If computers store everything using 0s and 1s, then how are **characters** represented?

•The **ASCII** (American Standard Code for Information Interchange) code assigns the numbers 0 through 127 to 128 characters, including upper and lower case alphabets plus various special characters, such as white space etc.

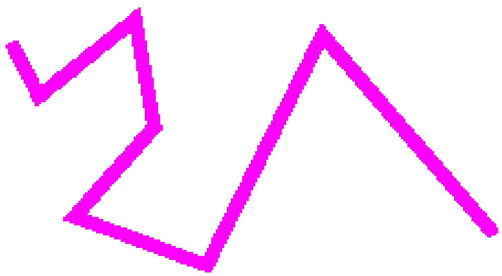
•e.g. decimal 85 is assigned to represent upper case U. In binary, 01010101 = 85. Thus the computer represents U using 01010101.

•Files which contain information encoded in ASCII are **easily transferred** and processed by different computers and programs. These are called “ASCII” or “text” files.

GIS Data Models

- A GIS map is a **scaled-down** digital representation of **point, line, area, and volume** features.
- A **logical data model** is how data are organized for use by the GIS.
- Traditionally there are **two GIS data models** used:
  - Raster
  - Vector

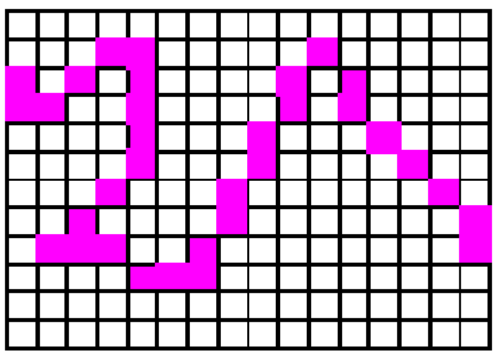
Rasters and vectors can be flat files … if they are simple



Vector-based line

```
4753456 623412
4753436 623424
4753462 623478
4753432 623482
4753405 623429
4753401 623508
4753462 623555
4753398 623634
```

Flat file



Raster-based line

```
0000000000000000
0001100000100000
1010100001010000
1100100001010000
0000100010001000
0000100010000100
0001000100000010
0010000100000001
0111001000000001
0000111000000000
0000000000000000
0000000000000
```

Attribute data

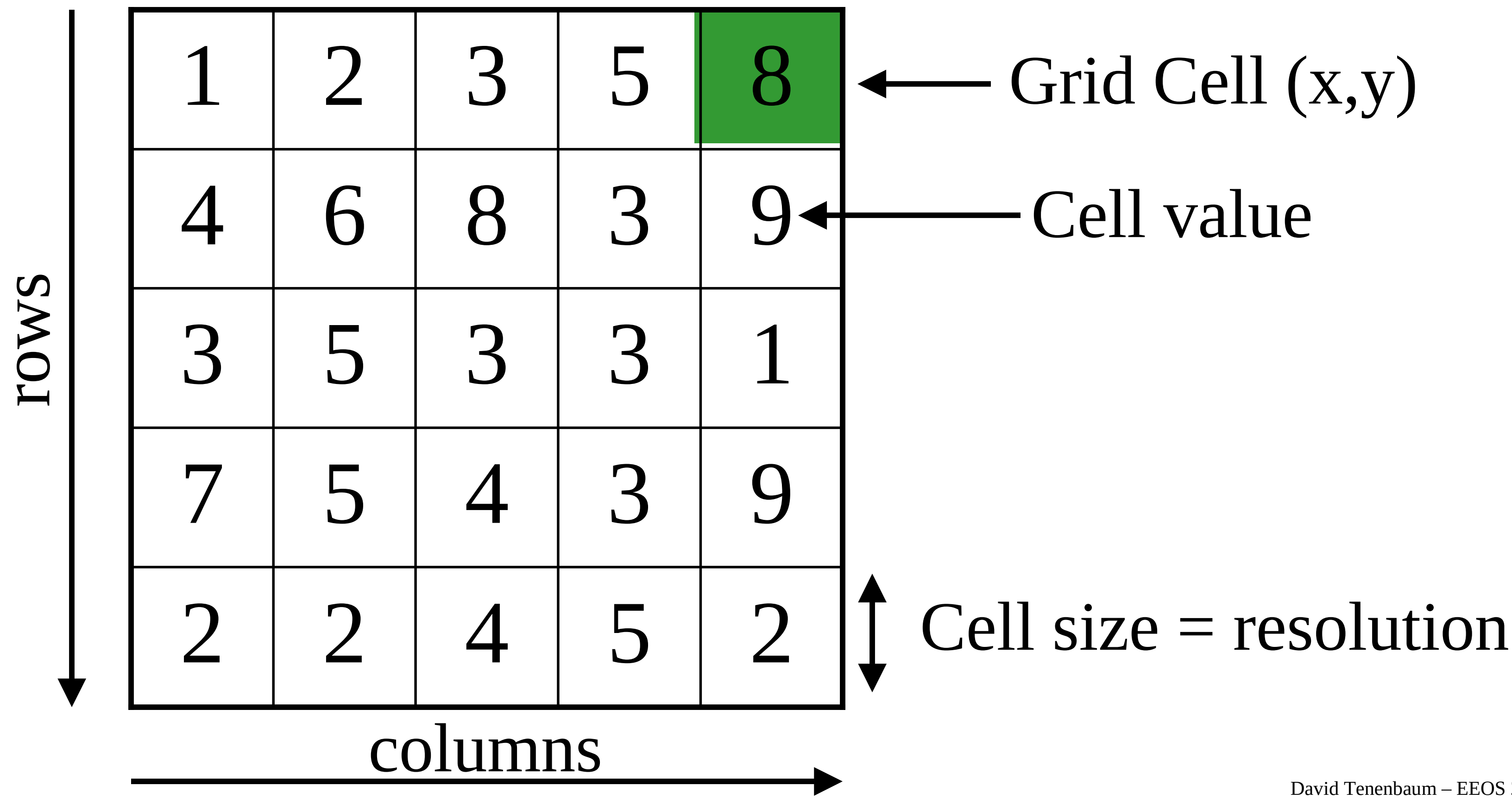
- Attribute data are stored **logically** in flat files.
- A **flat file** is a matrix of numbers and values stored in rows and columns, like a spreadsheet or a table.
- Both **logical and physical** data models have **evolved** over time.
- **DBMSs** use many different methods to **store and manage** flat files in **physical files**.
- **RDBMS** - Relational database management system. A type of database in which the **data is organized across several tables**. Tables are associated with each other through common fields. Data items can be recombined from different files.

Raster Data Model

- A representation of the world as a surface divided into a **regular grid of cells.** Raster models are useful for storing data that **varies continuously** such as in an aerial photograph, a satellite image, a surface of chemical concentrations, or an elevation surface.
- **One grid cell** is **one unit** or holds **one attribute**.
- **Every** cell has a **value**, even if it is **"missing."**
- A cell can hold **a number (absolute values) or an index value (coded values)** standing for an attribute.
- A cell has a **resolution**, given as the cell size in ground units.

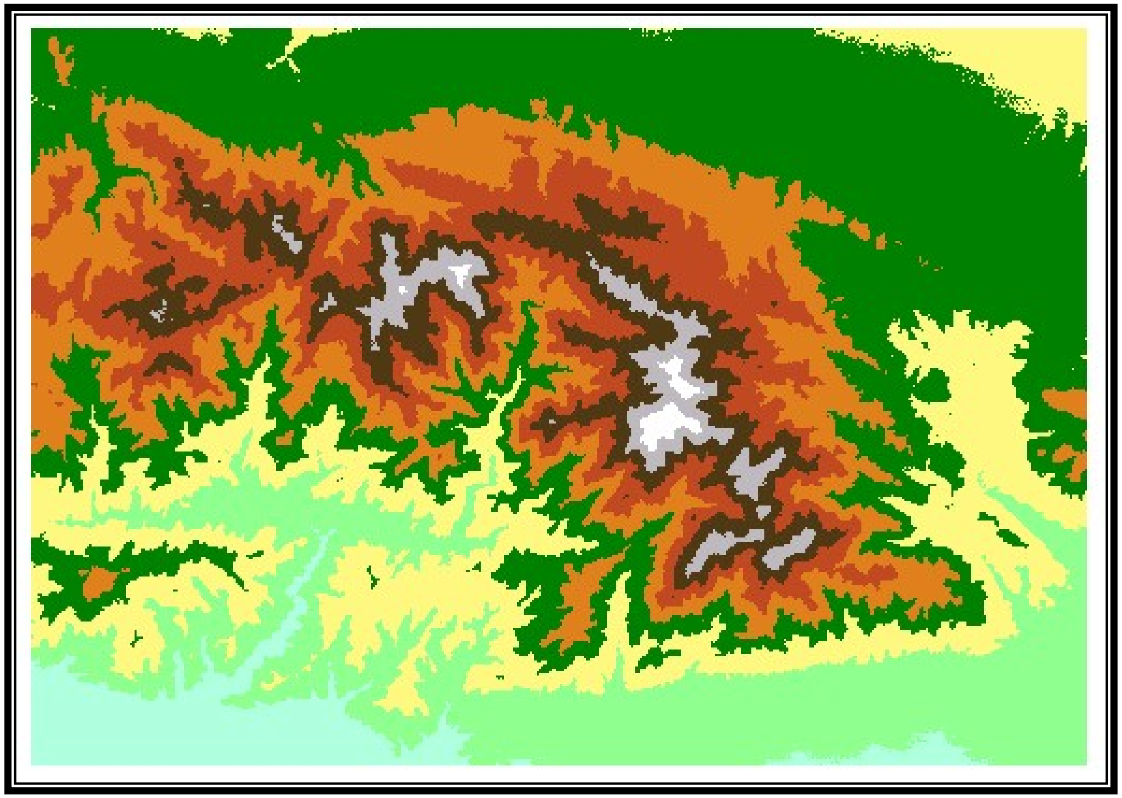
Generic structure for a grid

•The raster data model represents the Earth's surface as an **array** of two-dimensional grid cells, with each cell having an associated value:



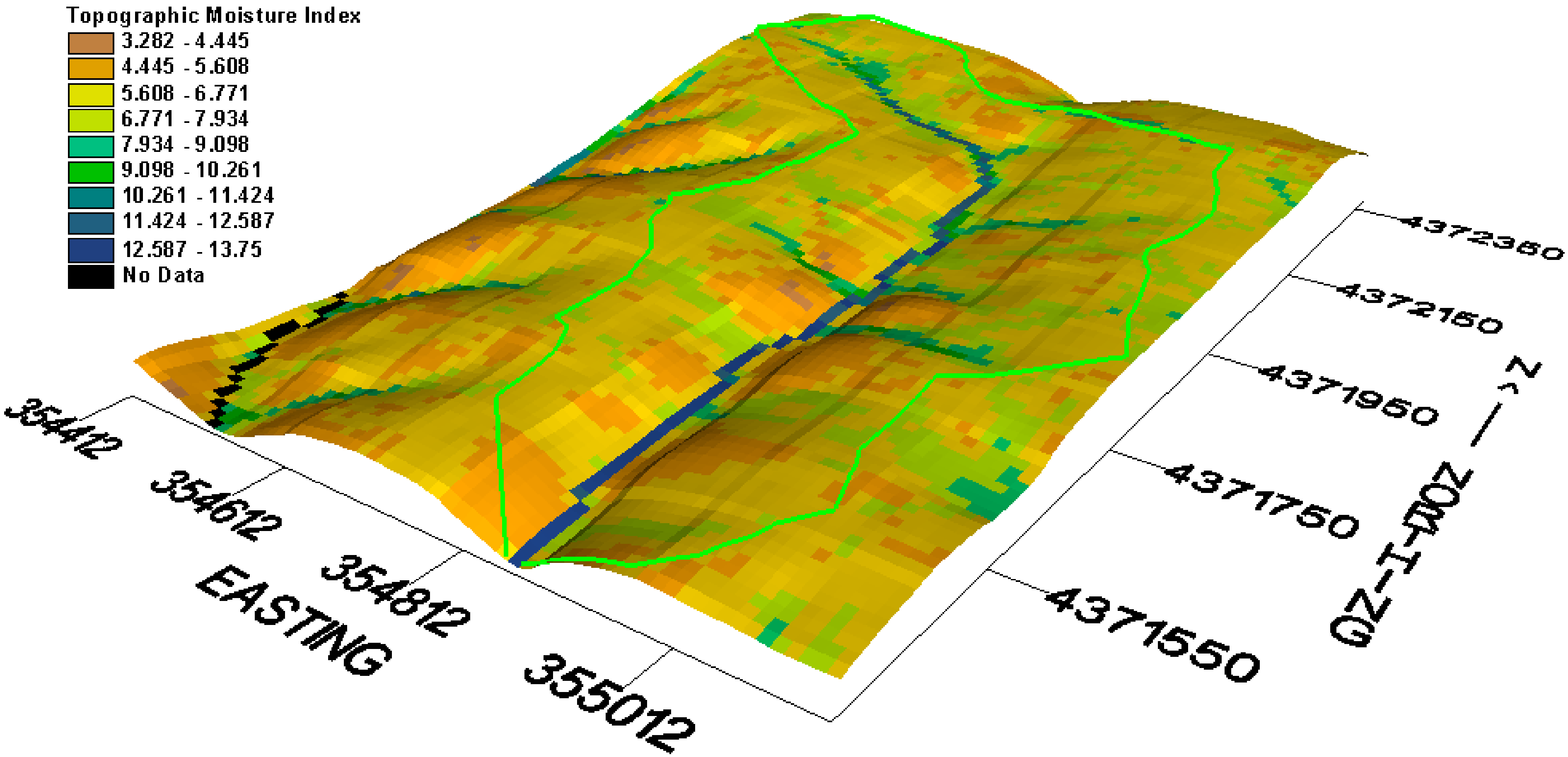
Cells - Absolute Values

•In this instance, the **value** of the cell is actually the value of the phenomenon of interest, e.g. elevation data (whether floating point or integer):



Pond Branch Catchment – Control

Topographic Index Example

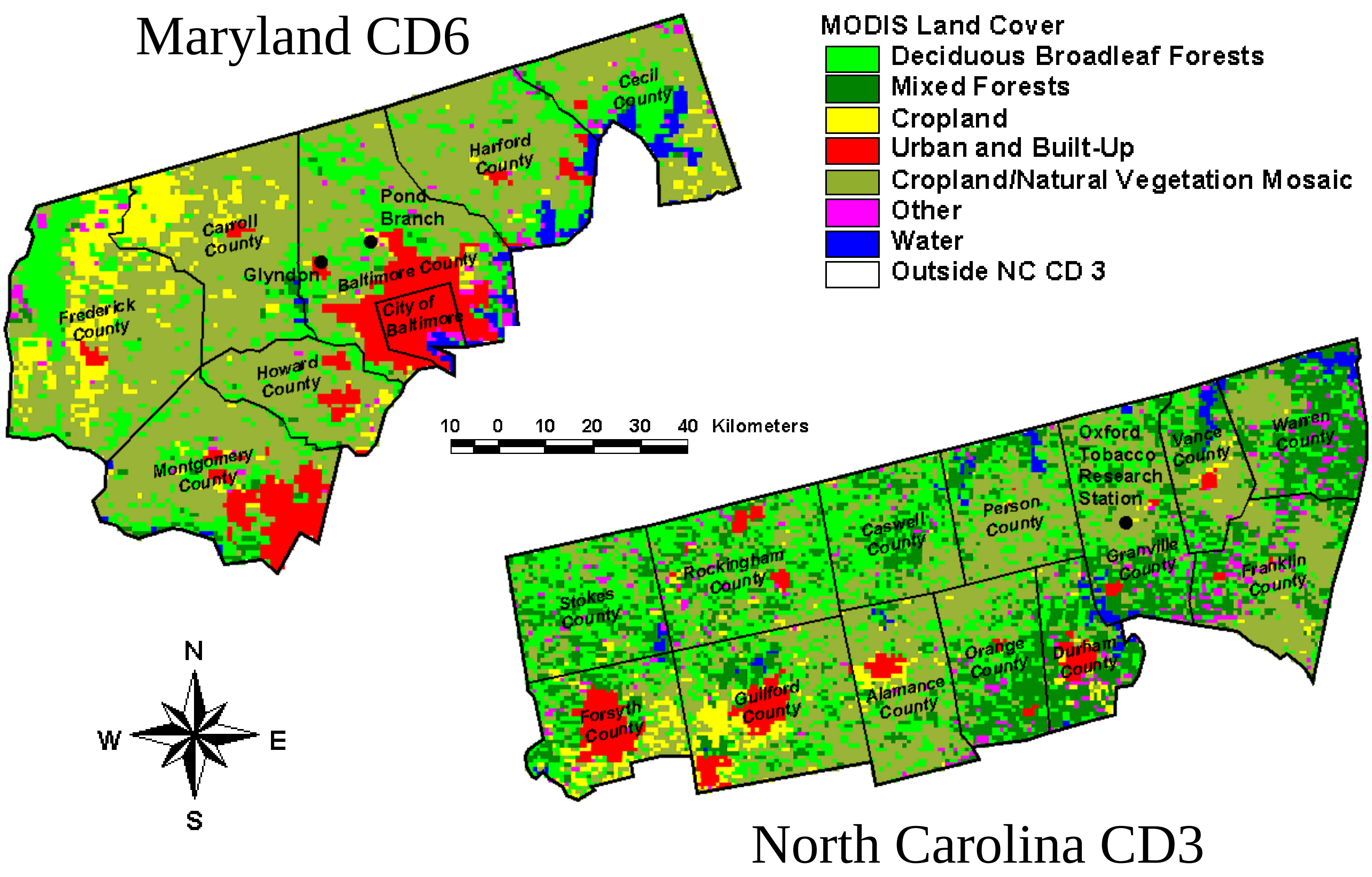


Cells - Coded Values

•Here, the values stored in each cell are used as **substitutes** for some **nominal** or **categorical** data, e.g. land cover classes:

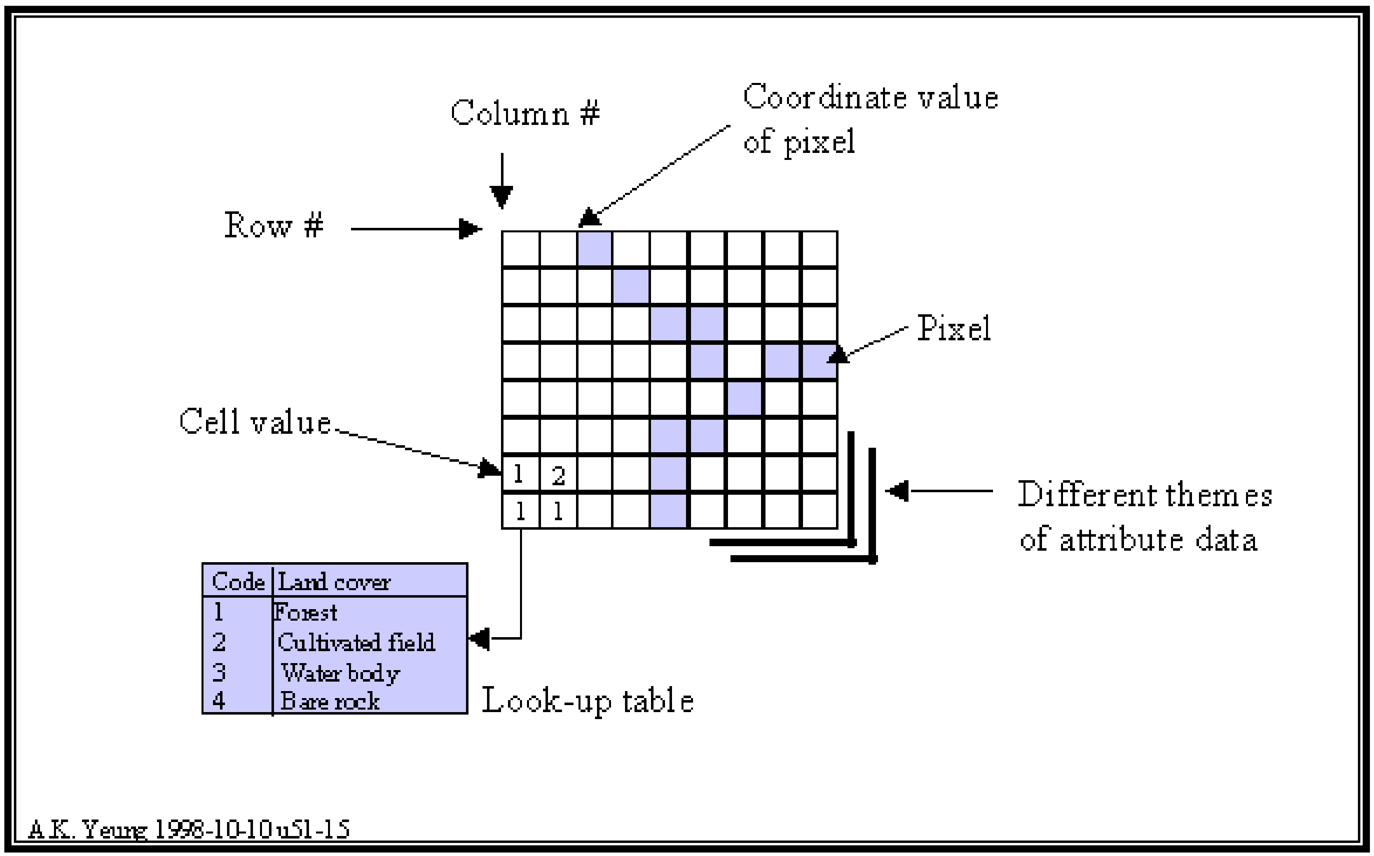
ID	Land Cover Type	Ownership
1	Grass	Smith
…		
8	Water	Smith
13	Sand	Smith

MODIS LULC In Climate Divisions



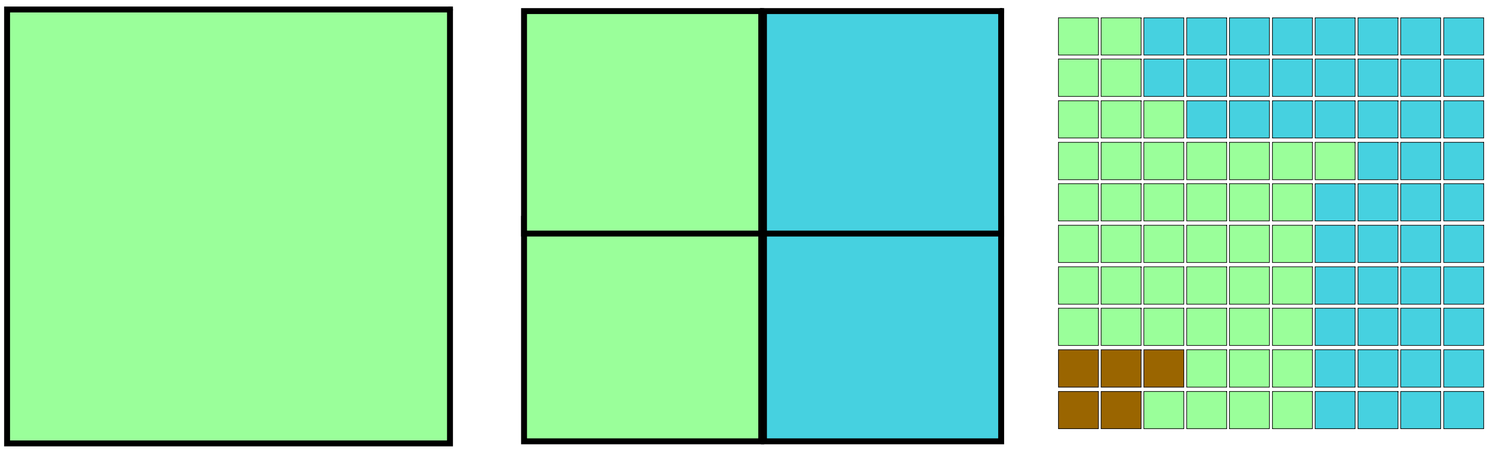
Cells – Coded Values

•The coded values can then **link** to one (or more) attribute tables that associate the cell values with various themes or attributes:



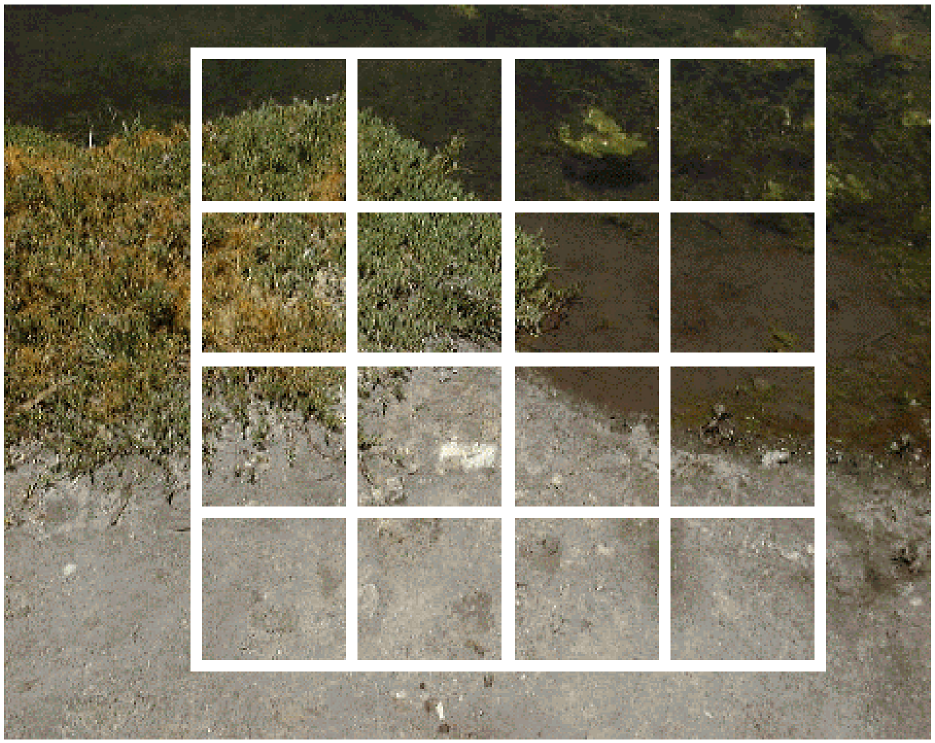
Cell Size & Resolution

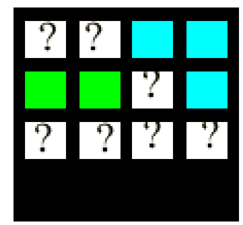
- The **size** of the **cells** in the raster data model determines the **resolution** at which features can be represented
- The selected **resolution** can have an **effect** on how features are represented:



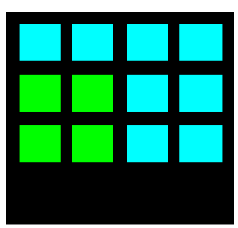
10 m Resolution   5 m Resolution   1 m Resolution

Rules for assigning values / The mixed pixel problem

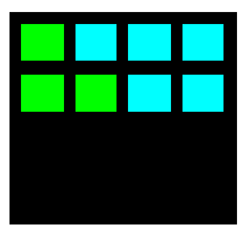




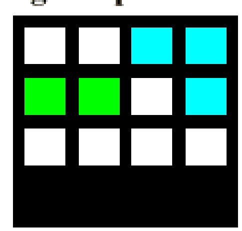
Water/Veg dominates



Winner takes all



Edges separate

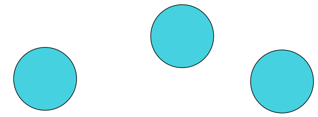


Raster Data Model - Objects

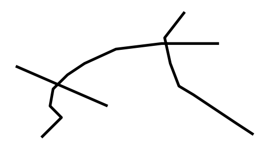
The raster data model still represents spatial objects, but does so differently from the vector model:

<u>Geographic Primitives</u>

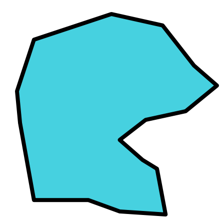
- Points
  - –0 dimensional



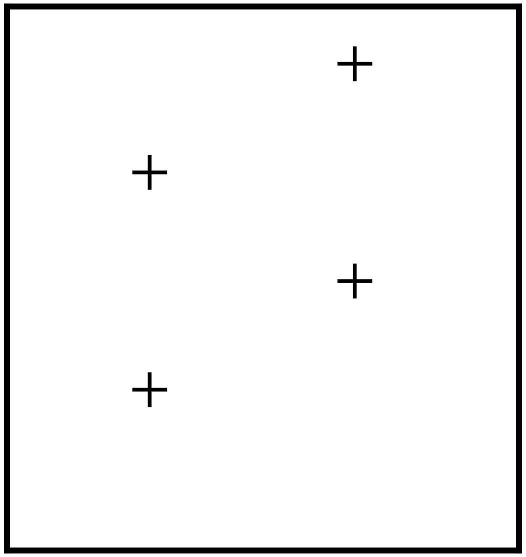
- Lines
  - –1 dimensional

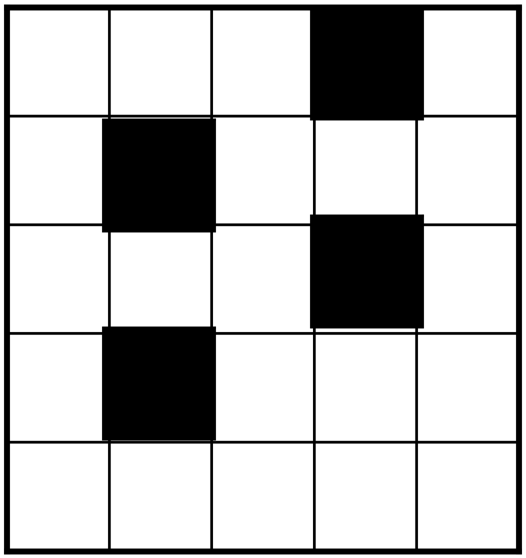


- Polygons
  - –2 dimensional



Raster Data Model - Points

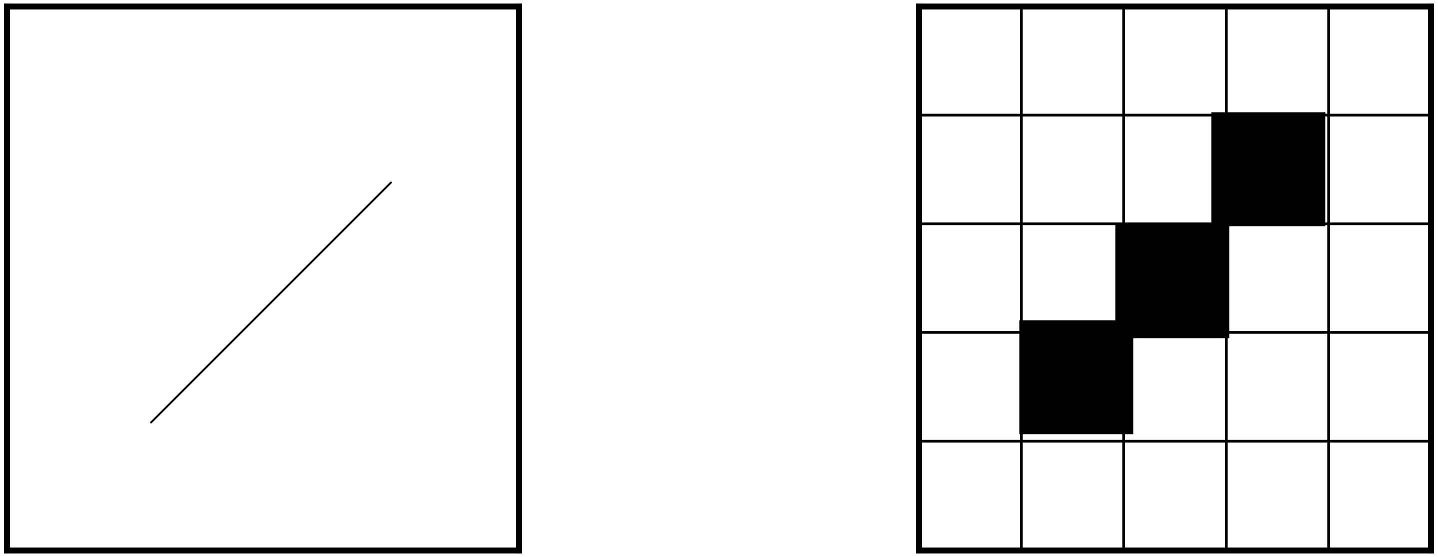




1 point = 1 cell

What problem do we have here? How can we solve it?

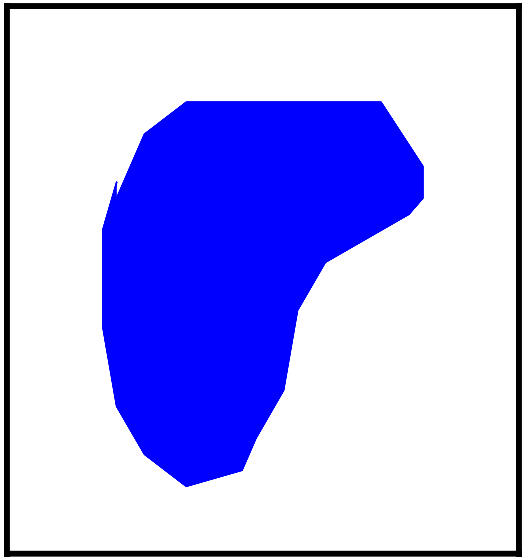
Raster Data Model - Lines

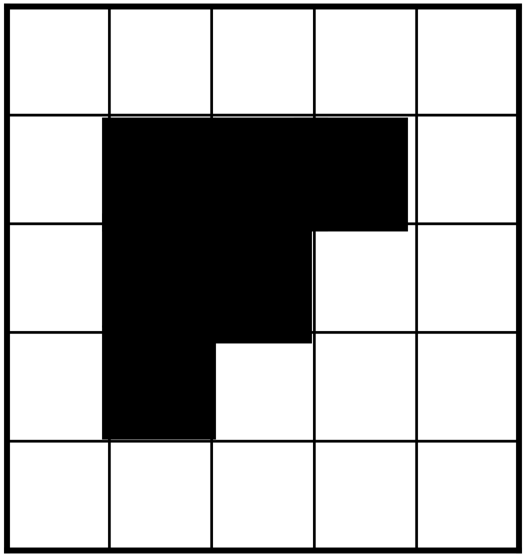


A line = a series of connected cells that portray length

Is there a problem with this representation?

Raster Data Model - Areas

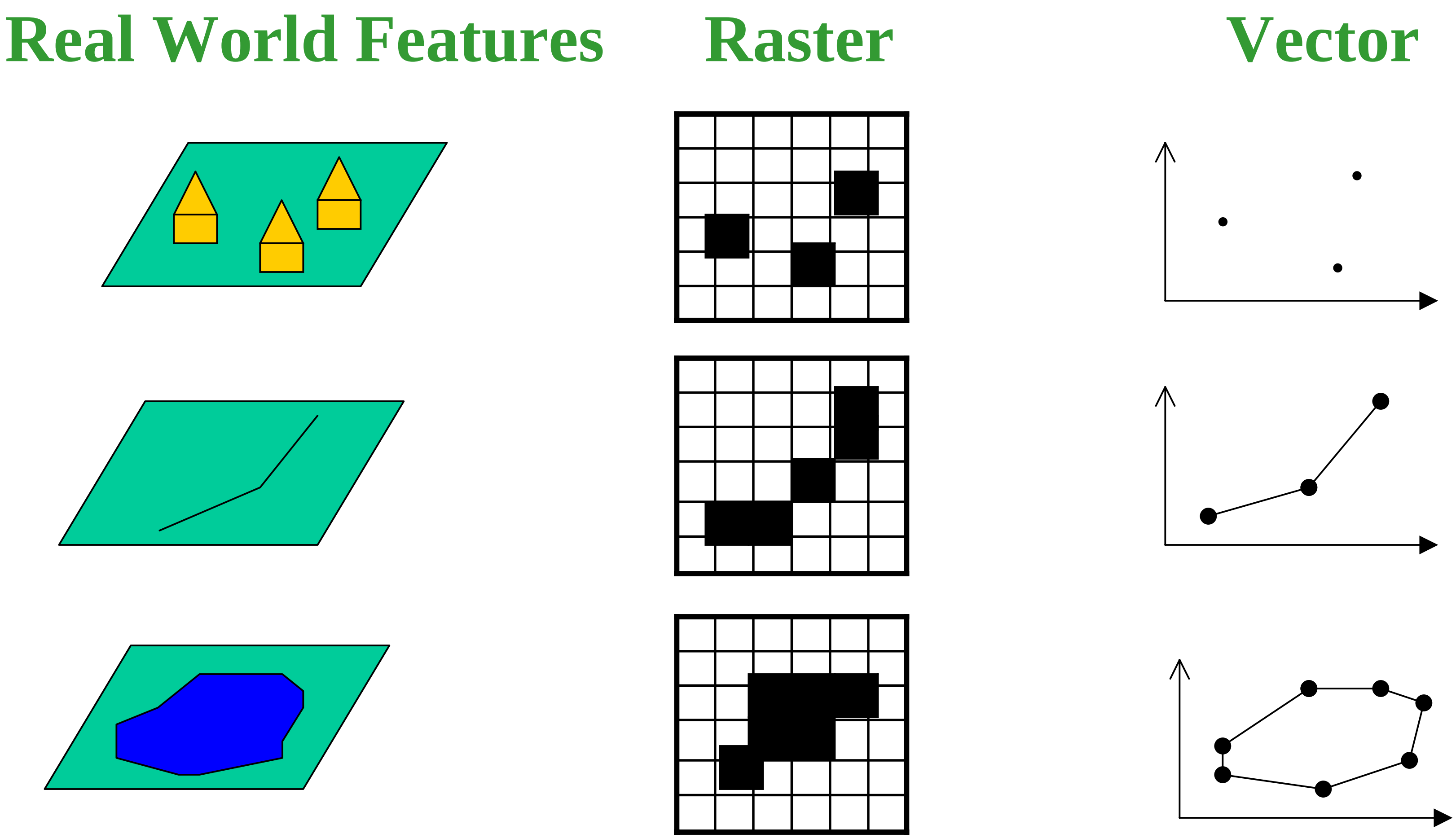




Area = a group of connected cells that portray a shape

What problems could we have with this representation?

Raster and Vector Data Model Comparison



"A raster model tells what occurs everywhere, while a vector model tells where every thing occurs"

Grids and missing data

Rasters are faster...

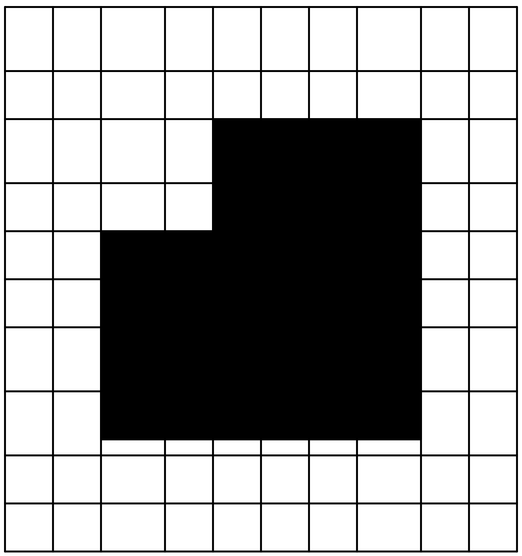
- Points and lines in raster format have to move to a **cell center**.
- Lines can become **fat**. Areas may need **separately coded edges**.
- Each cell can be owned by **only one feature**.
- As data, all cells must be able to hold the **maximum cell value**.
- Rasters are **easy** to understand, easy to read and write, and easy to draw on the screen.

RASTER

- A grid or raster maps directly onto a programming computer memory structure called **an array**.
- Grids are **poor** at representing **points, lines and areas**, but **good at surfaces**.
- Grids are a natural for **scanned or remotely sensed data**.
- Grids suffer from the **mixed pixel problem**.
- Grids must often include **redundant or missing data**.
- Grid compression techniques used in GIS are **run-length encoding and quad trees**.

Raster Data Storage – No Compaction

This approach represents each cell **individually** in the file:



0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	1	1	1	0	0
0	0	0	0	1	1	1	1	0	0
0	0	1	1	1	1	1	1	0	0
0	0	1	1	1	1	1	1	0	0
0	0	1	1	1	1	1	1	0	0
0	0	1	1	1	1	1	1	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

max. cell value

rows

columns

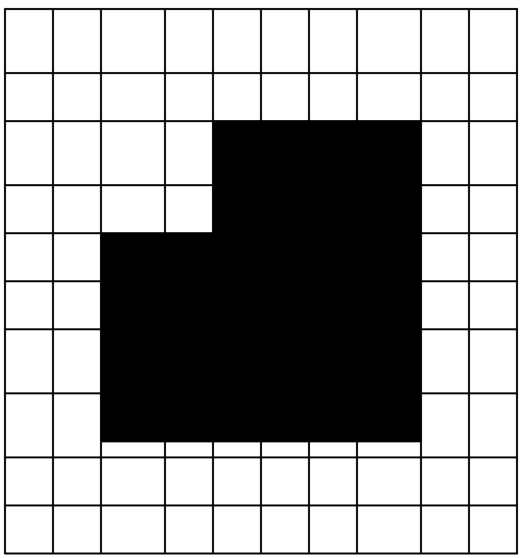
10, 10, 1
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 1 1 1 1 0 0
0 0 0 0 1 1 1 1 0 0
0 0 1 1 1 1 1 1 0 0
0 0 1 1 1 1 1 1 0 0
0 0 1 1 1 1 1 1 0 0
0 0 1 1 1 1 1 1 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0

103 values

Problem: too much **redundancy**

Raster Data Storage – Run Length Encoding

This approach takes advantage of **patterns** in the data, taking advantage of the **repetition** of values in a row:



0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	1	1	1	0	0
0	0	0	0	1	1	1	1	0	0
0	0	1	1	1	1	1	1	0	0
0	0	1	1	1	1	1	1	0	0
0	0	1	1	1	1	1	1	0	0
0	0	1	1	1	1	1	1	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

header → 10,10,1

row by row:

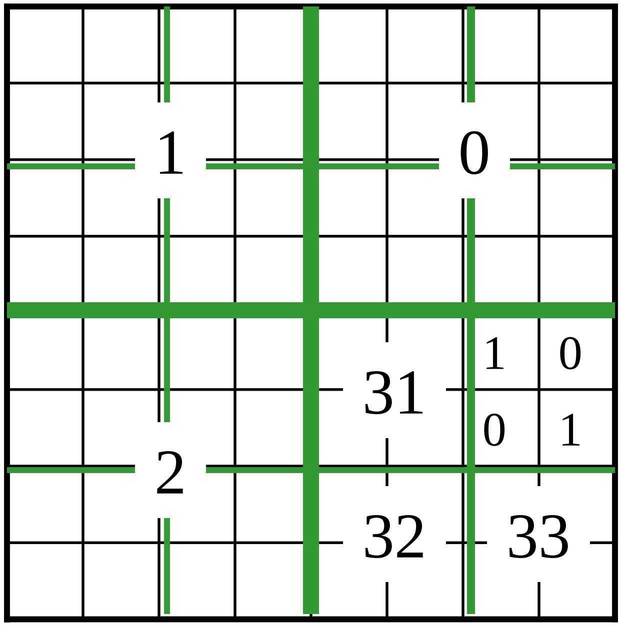
0, 10
0, 10
0, 4, 1, 4, 0,2
0, 4, 1, 4, 0,2
0, 2, 1, 6, 0,2
0, 2, 1, 6, 0,2
0, 2, 1, 6, 0,2
0, 2, 1, 6, 0,2
0, 10
0, 10

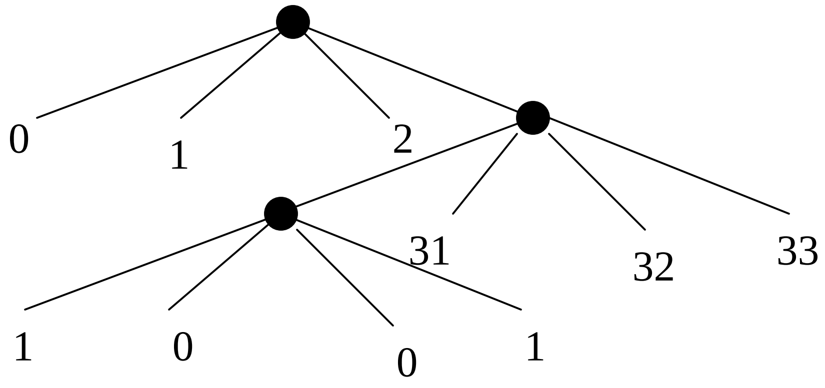
45 values

There is a tendency towards **spatial autocorrelation**; for **nearby cells** to have **similar values** - values often occur in runs across several cells

Raster Data Storage - Quadtrees

The quadtree method **recursively subdivides** the cells of a raster grid into quads (quarters) until **each quad** can be represented by a **unique cell value**:



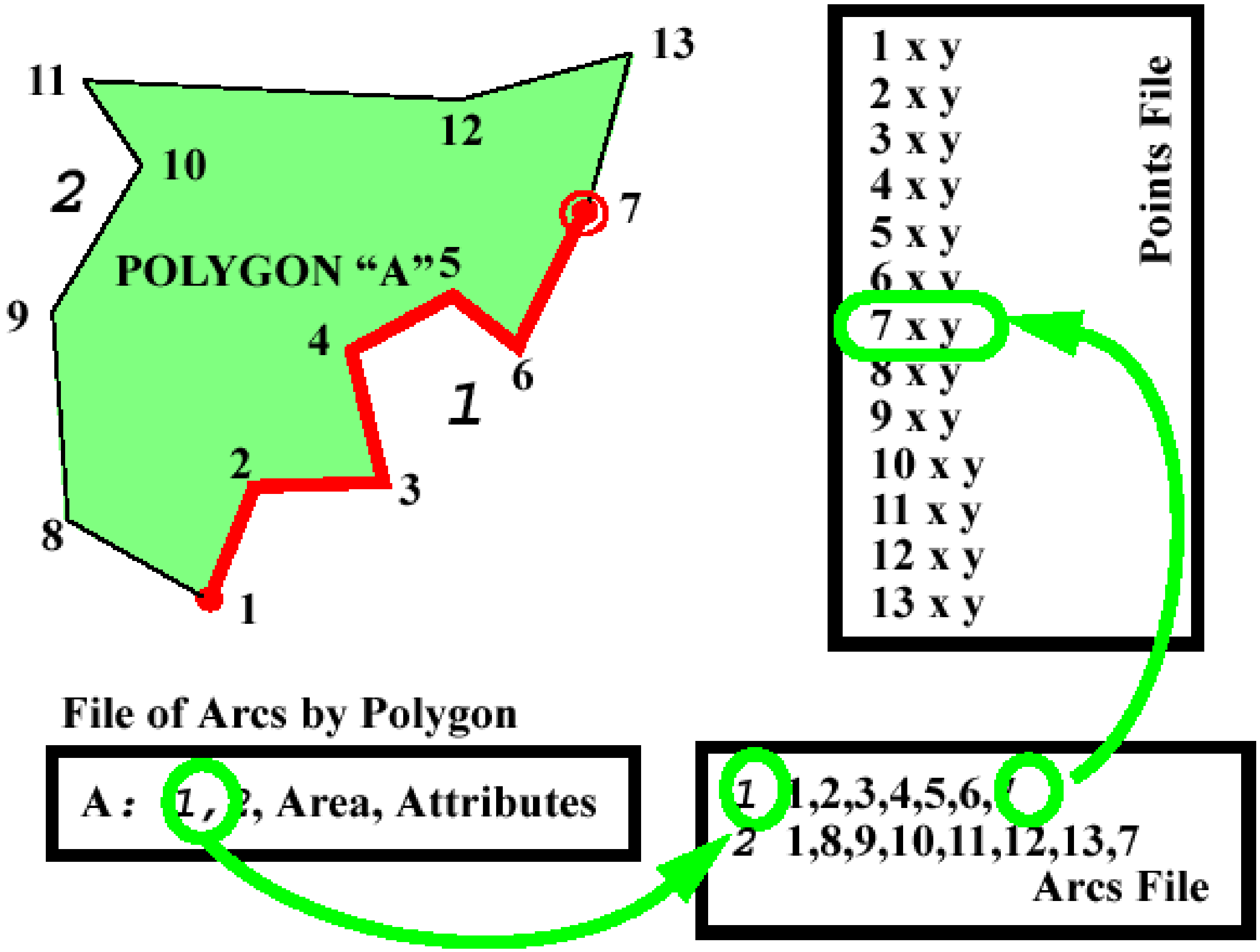


The number of subdivisions depends on the **complexity** of features and stores **more detail** in areas of greater complexity

The Vector Model

- A vector data model uses **points** stored by their **real (earth) coordinates**.
- **Lines and areas** are built from **sequences of points in order**.
- Lines have a **direction** to the **ordering** of the points.
- **Polygons** can be built from **points or lines**.
- Vectors can store information about **topology**.

Arc/node map data structure with files



VECTOR

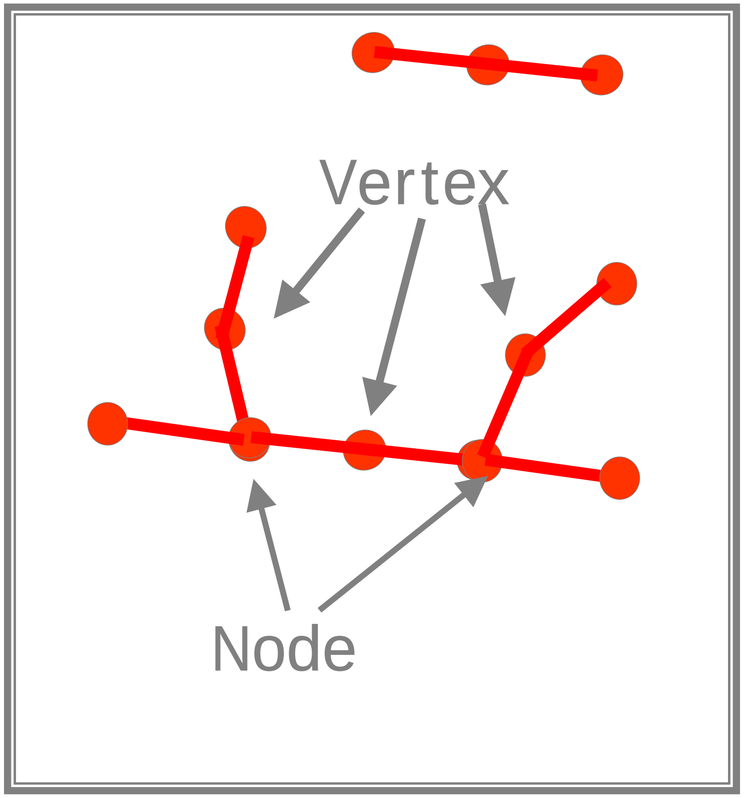
- At first, GISs used vector data and **cartographic spaghetti structures**.
- Vector data evolved the **arc/node model** in the 1960s.
- In the arc/node model, an **area consist of lines** and a **line consists of points**.
- Points, lines, and areas can each be **stored in their own files**, with **links between** them.
- The **topological vector model** uses the **line (arc)** as a basic unit. **Areas (polygons)** are **built up** from arcs.
- The **endpoint** of a line (arc) is called a **node**. Arc junctions are only at nodes.
- Stored with the arc is the **topology** (i.e. the connecting arcs and left and right polygons).

… but vectors are correcter (sp)

- Vector can represent point, line, and area features **very accurately**.
- Vectors work well with **pen and light-plotting devices** and **tablet digitizers**.
- Vectors are **not good at continuous coverages** or plotters that fill areas.

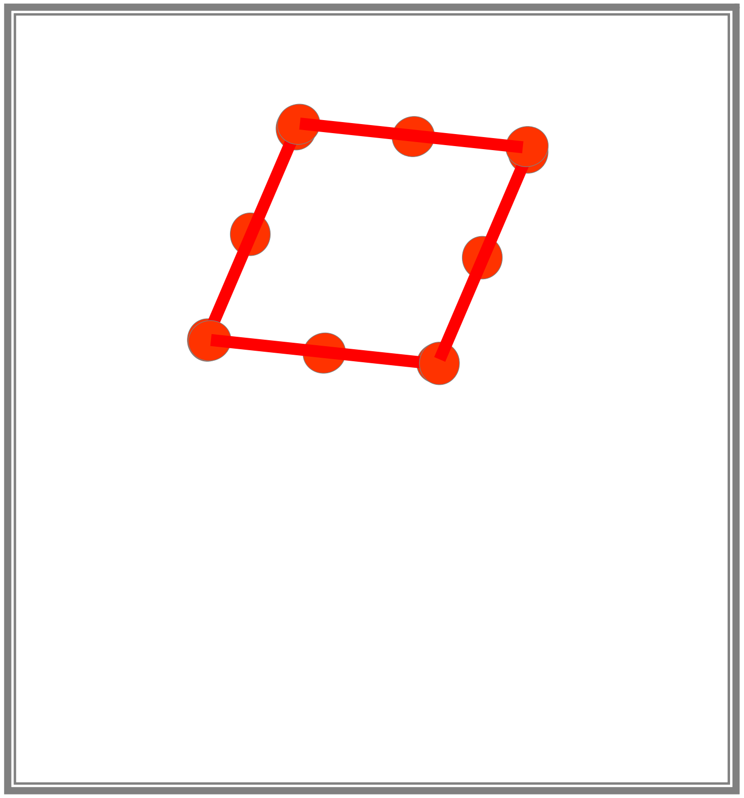
Vector Data Model - Objects

- Lines/Arcs
  - these are formed by **joining** multiple points
  - points at the junctions of lines are called **nodes**



Vector Data Model - Objects

- Polygons
  - these are composed of **multiple lines or arcs**
  - They are required to have the property of **closure**, meaning that the multiple lines/arcs must form a **closed shape** for it to be a polygon



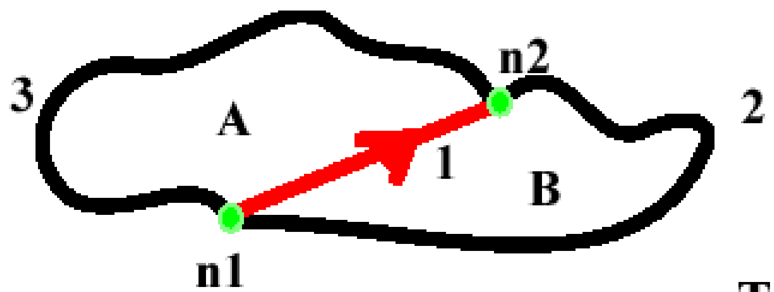
Vector Data Model - Objects

- Each spatial data object is assigned a unique object ID (**unique identifier**)
- **Spatial coordinates** are stored in tables for each data object:
  - **Lines**
    - Coordinates for endpoints
    - Vertices along line (optional)
  - **Polygons**
    - Multiple (x,y) coords defining polygon
    - List of lines that define the polygon boundary

TOPOLOGY

- Topological data structures **dominate** GIS software.
- Topology allows **automated error detection and elimination**.
- Rarely are maps **topologically clean** when digitized or imported.
- A GIS has to be able to **build topology** from unconnected arcs.
- Nodes that are close together are **snapped**.
- **Slivers** due to double digitizing and overlay are **eliminated**.

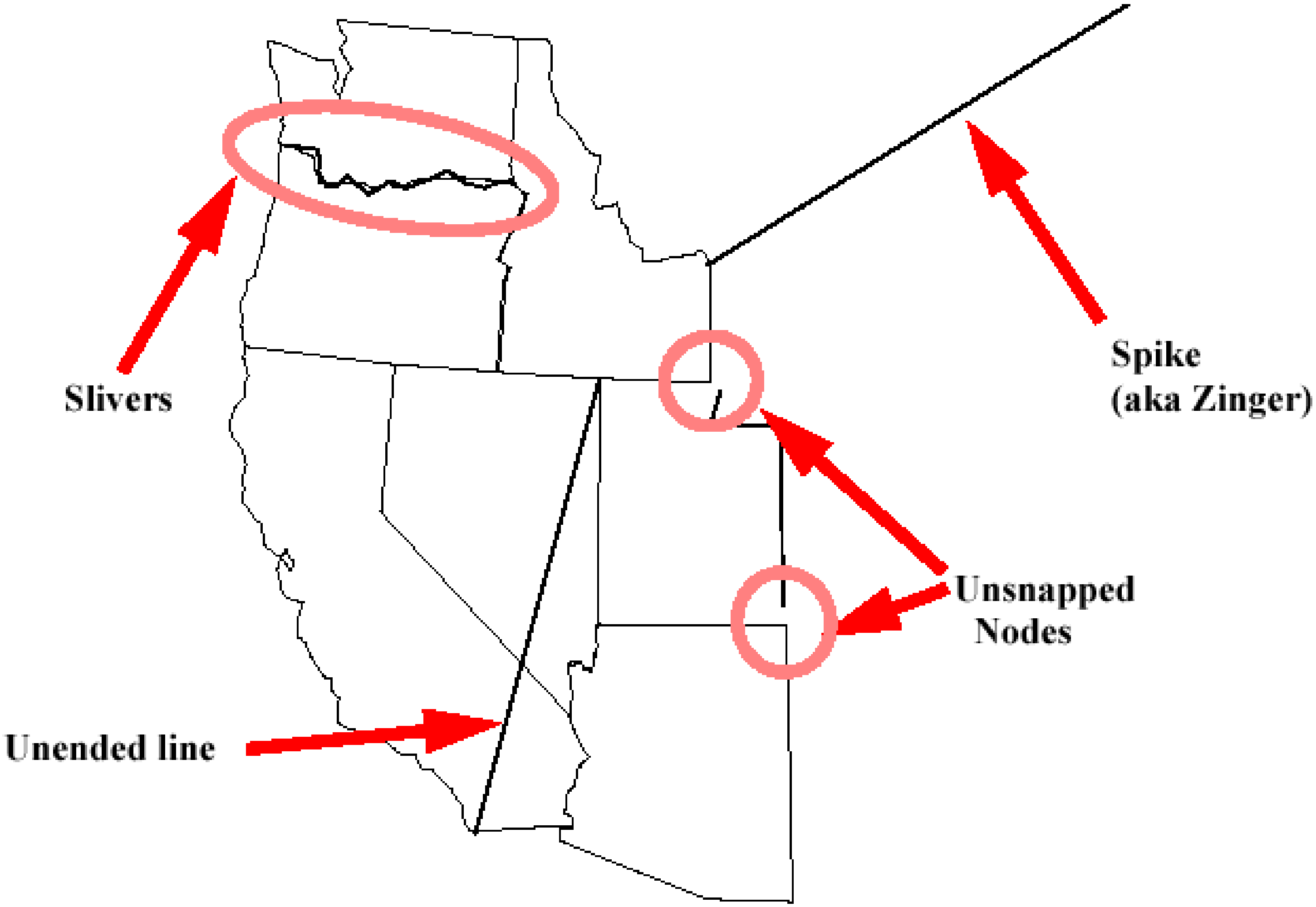
Basic arc topology



Topological Arcs File

Arc	*From*	*To*	*PL*	*PR*	*n1x*	*n1y*	*n2x*	*n2y*
1	n1	n2	A	B	x	y	x	y

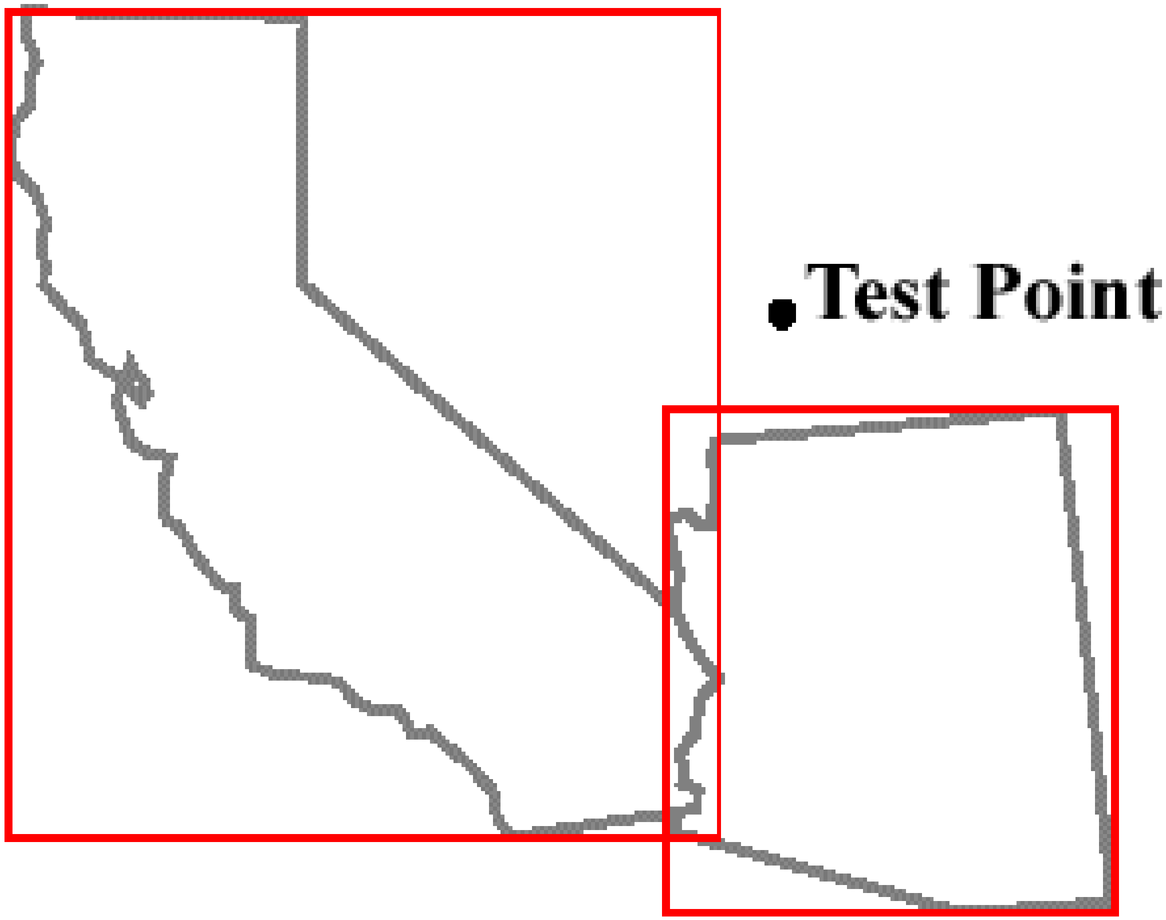
Topological errors



Topology Matters

- The **tolerances** controlling snapping, elimination, and merging must be **considered carefully**, because they can move features.
- Complete topology makes **map overlay feasible**.
- Topology allows many GIS operations to be done **without accessing the point files**.

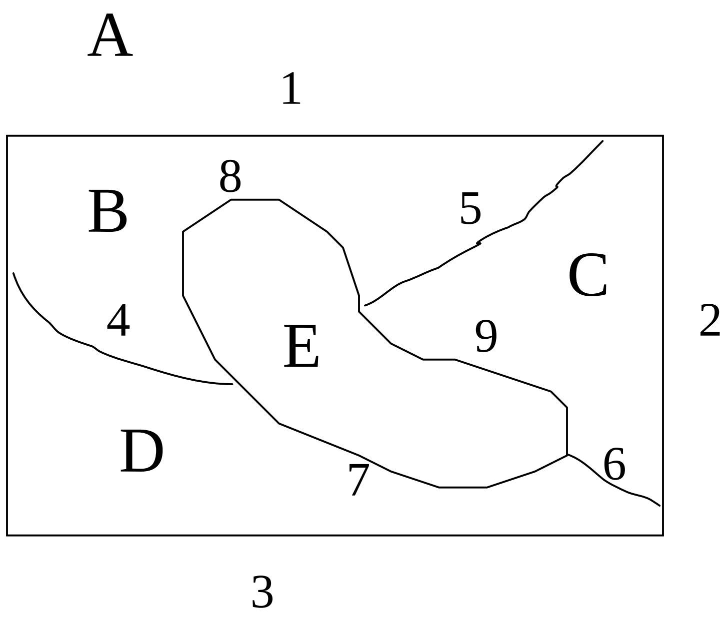
The bounding rectangle



Vector Data Model - Topology

- Topology defines spatial relationships. The arc-node data structure supports the following topological concepts:
- **Area definition:** Arcs connect to surround an area, defining a polygon
- **Containment:** Nodes (or arcs) can be found within a polygon
- **Connectivity:** Arcs connect to each other at shared nodes
- **Contiguity:** Arcs have a defined direction, and left and right sides

Topology – Area Definition



Polygon-Arc Topology

Polygon	Arc List
B	1,5,8,4
C	2,6,9,5
D	6,3,4,7
E	9,7,8

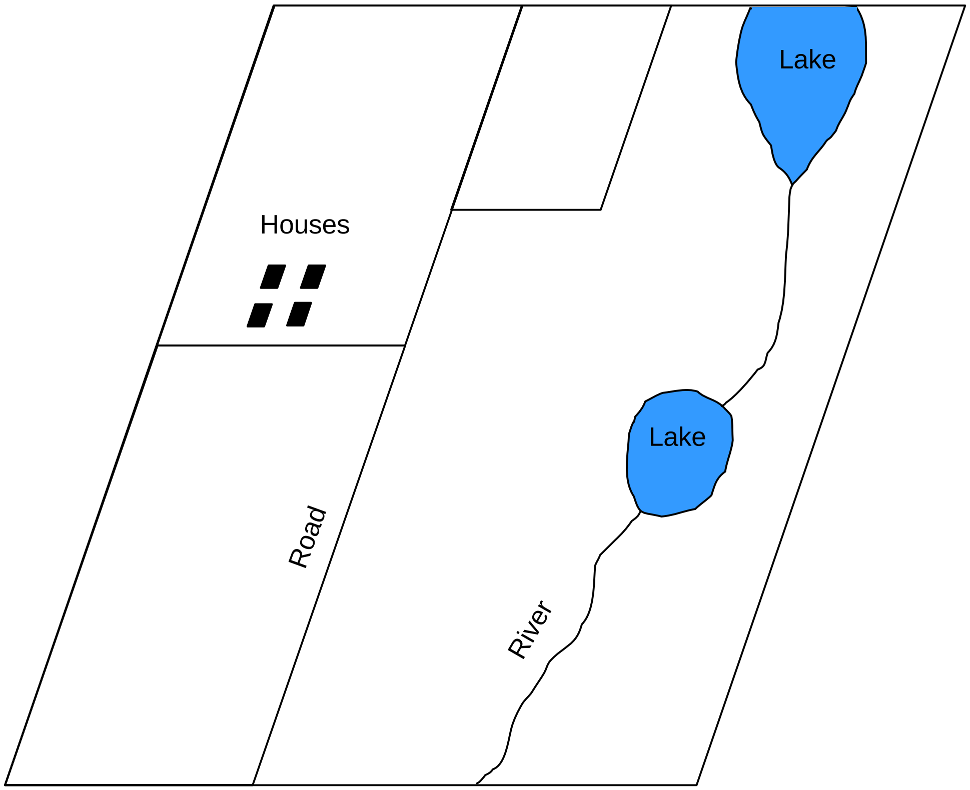
Polygons are stored as the list of arcs defining their boundaries, and arc coordinates are stored only once, reducing the amount of data and ensuring that the boundaries of adjacent polygons don't overlap (because they are formed from the same arcs)

Topology - Points

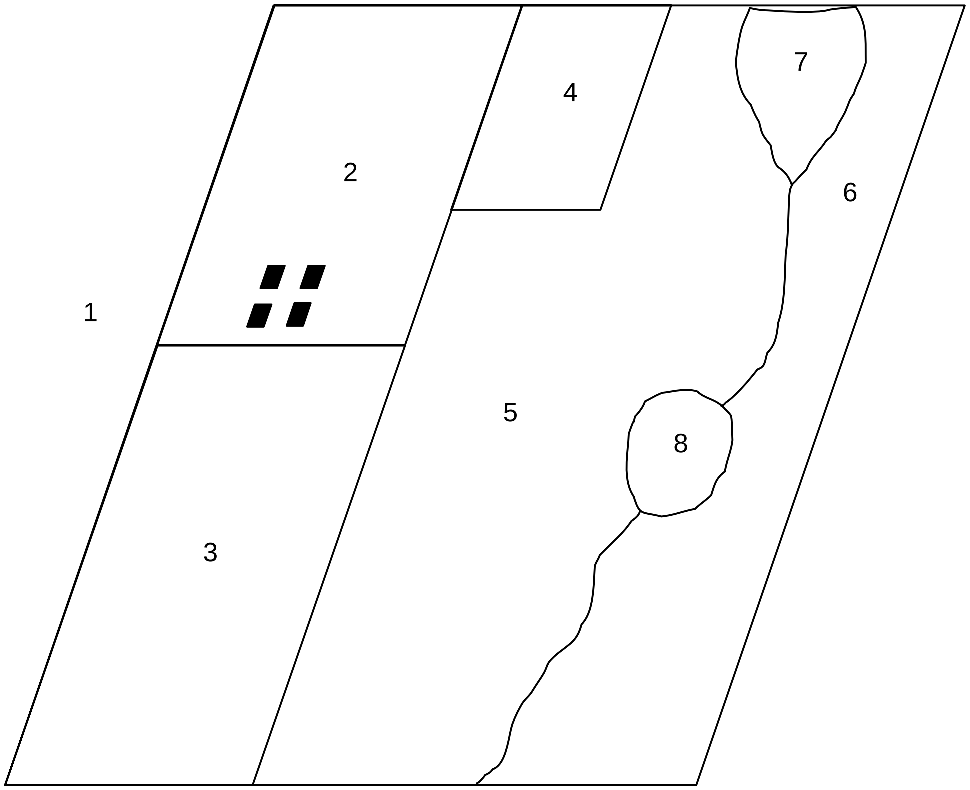
Points

- Containment (requires Area Definition)
  - Point-in-Polygon analysis
  - This can be extended to Arc- and Polygon-in-Polygon analysis, since in the Arc-Node model, arcs and polygons are simply built from points, thus this approach is operated on all the points in an arc or polygon to determine if it is contained

Topology - Containment



Topology - Containment



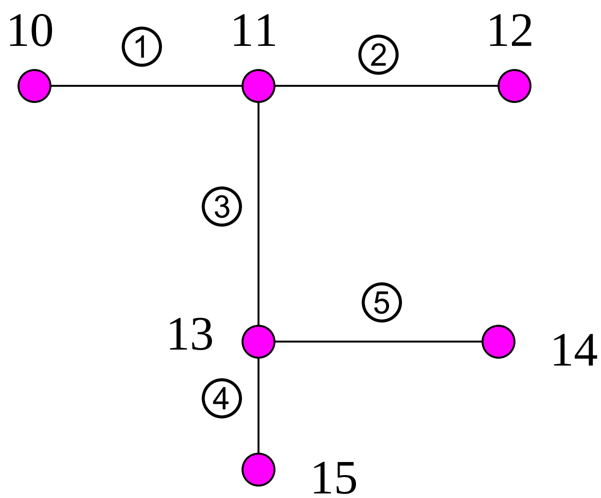
Topology - Lines

Lines

- Containment (requires Area Definition)
  - Line-in-Polygon analysis
- Connectivity
  - lines that share a node are connected

Topology - Connectivity

Connected arcs are determined by searching through the list for common node numbers.



Arc-node list

Arc	From-Node	To-Node
1	10	11
2	11	12
3	11	13
4	13	15
5	13	14

Arcs 1, 2, and 3 all intersect, because they share Node 11. Thus, the computer can determine that it is possible to travel along arc 1 and turn onto arc 3. But it is not possible to travel from Arc 1 to Arc 5

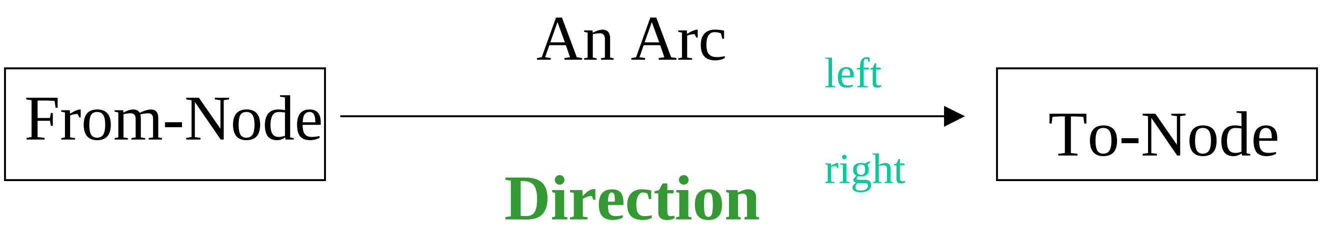
Topology - Polygons

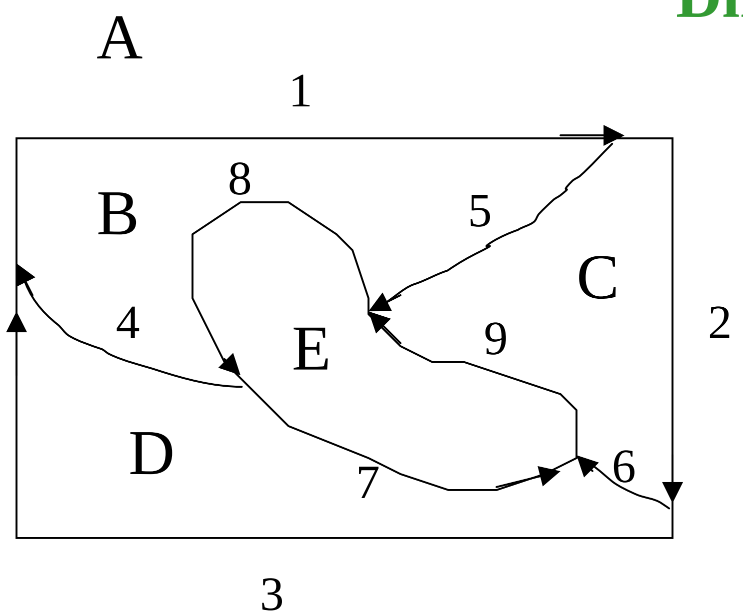
Polygons

- Containment
  - Polygon-in-Polygon analysis
- Adjacency (via contiguity)
  - Polygons that share an arc are adjacent

Topology - Contiguity

Two geographic features which share a boundary are called adjacent. Contiguity is the topological concept which allows the vector data model to determine adjacency.

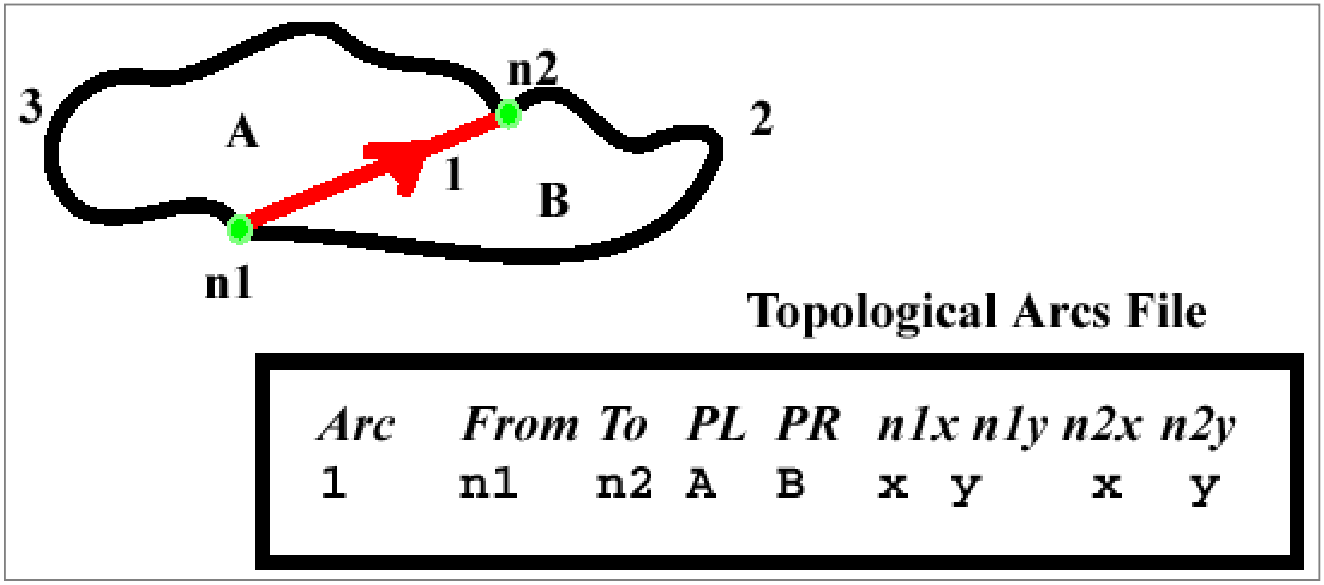




Arc	Left Polygon	Right Polygon
5	B	C
9	E	C
6	?	?
1	?	?

Topology

- A **spatial data structure** used primarily to ensure that the associated data forms a **consistent and clean topological fabric**. For instance, the **arc-node topology**: Typically the arc is stored as the base unit, storing with it the polygon left and right, the forward and reverse arc linkages and the arc end nodes.



Topology (ESRI)

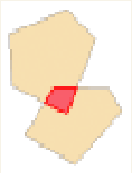
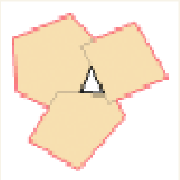
- With advances in object-oriented GIS development, an **alternative view** of topology has evolved.
- The **geodatabase** supports an approach to modeling geography that **integrates the behavior** of different feature types and supports different types of **key relationships**.
- In this context, topology is **a collection of rules and relationships** that, coupled with a set of editing tools and techniques, enables the geodatabase to **more accurately model geometric relationships** found in the world.

Topology (ESRI)

- Topology, implemented as **feature behavior and rules**, allows a **more flexible** set of geometric relationships to be modeled than topology **implemented as a data structure**. It also allows topological relationships to exist **between more discrete types of features** within a feature dataset.
- In this alternative view, topology may still be employed to ensure that the data forms a clean and consistent topological fabric, but also more broadly, it is used to **ensure that the features obey the key geometric rules defined for their role in the database**.

Topology Rules (ESRI)

Polygon rules

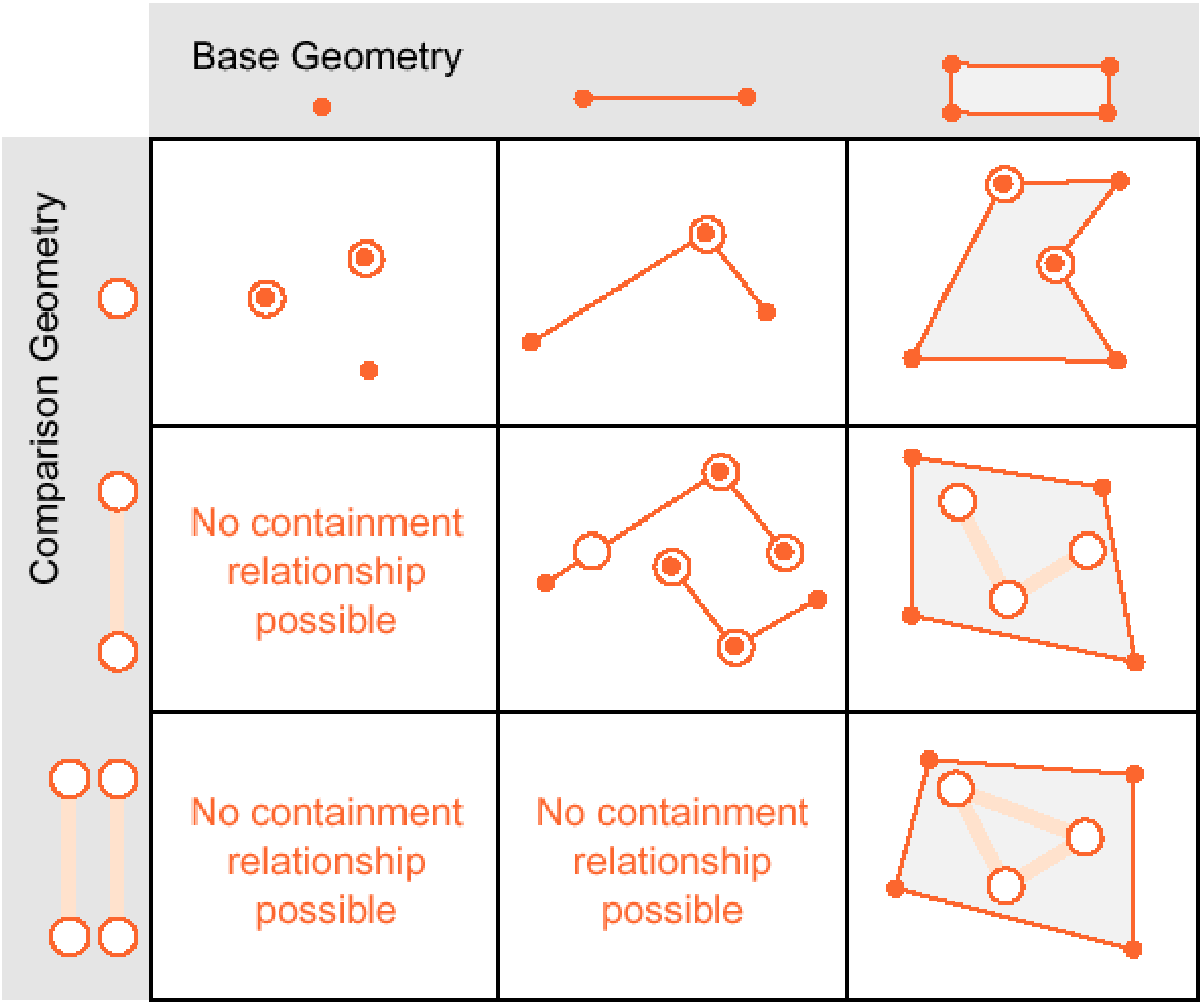
Topology rule	Rule description	Potential fixes	Examples
Must Not Overlap	Requires that the interior of polygons in the feature class not overlap. The polygons can share edges or vertices. This rule is used when an area cannot belong to two or more polygons. It is useful for modeling administrative boundaries, such as ZIP Codes or voting districts, and mutually exclusive area classifications, such as land cover or landform type.	Subtract, Merge, Create Feature	
Must Not Have Gaps	This rule requires that there are no voids within a single polygon or between adjacent polygons. All polygons must form a continuous surface. An error will always exist on the perimeter of the surface. You can either ignore this error or mark it as an exception. Use this rule on data that must completely cover an area. For example, soil polygons cannot include gaps or form voids—they must cover an entire area.	Create Feature	

Spatial Relations

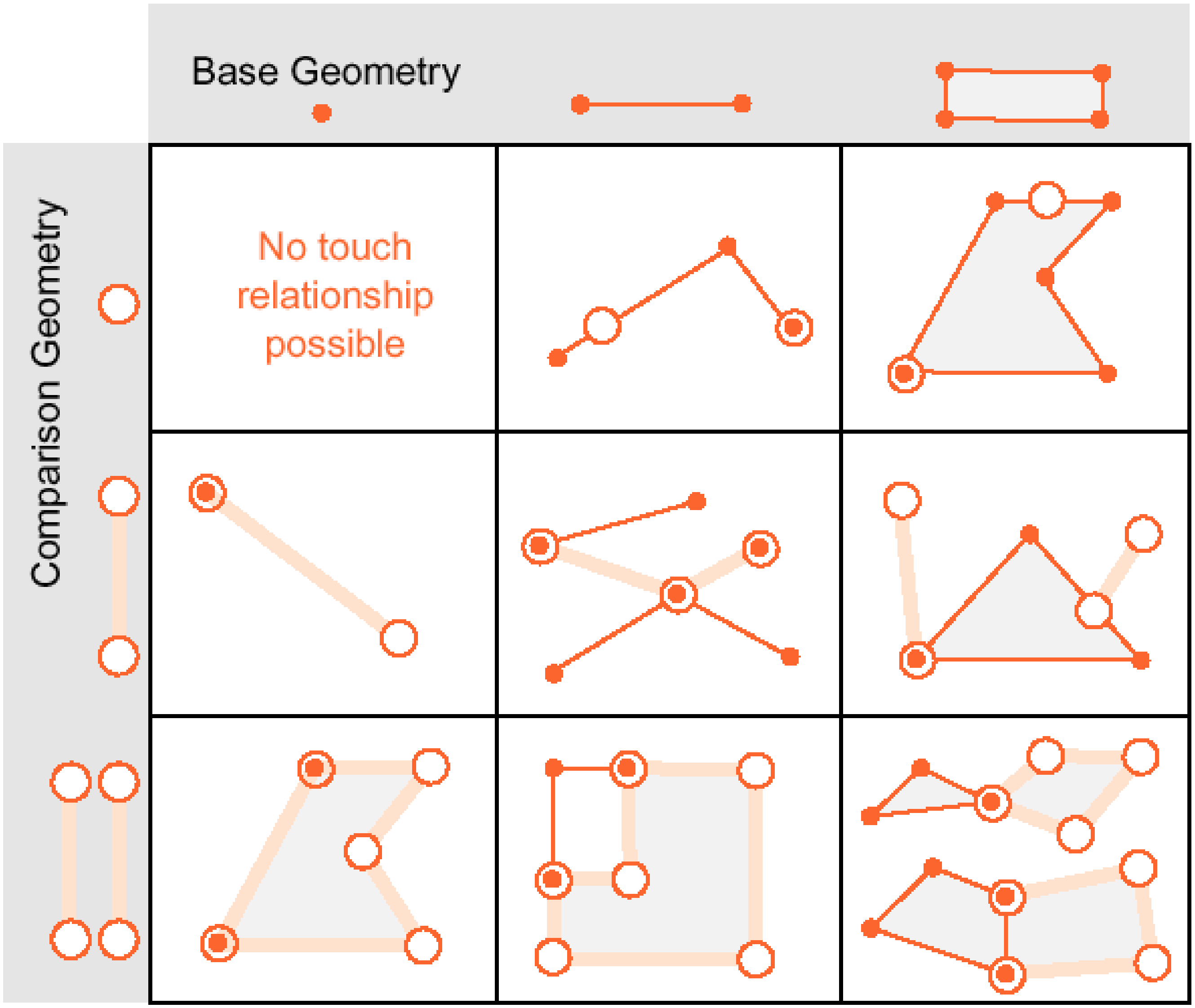
•In addition to relations that join tables based on an **identical common key**, we can evaluate relations between the **spatial characteristics of features**:

- Equals – same geometries
- Disjoint – geometries share common point
- Intersects – geometries intersect
- Touches – geometries intersect at common boundary
- Crosses – geometries overlap
- Within– geometry within
- Contains – geometry completely contains
- Overlaps – geometries of same dimension overlap
- Relate – intersection between interior, boundary or exterior

Contains Relation



Touches Relation



Why Does Topology Matter?

- Topology is used most fundamentally **to ensure data quality** and allow your GIS database to more **realistically represent** geographic features.
- **Automatic** error detection
- Enables **efficient spatial analysis** without continuously dealing with the original x,y data.

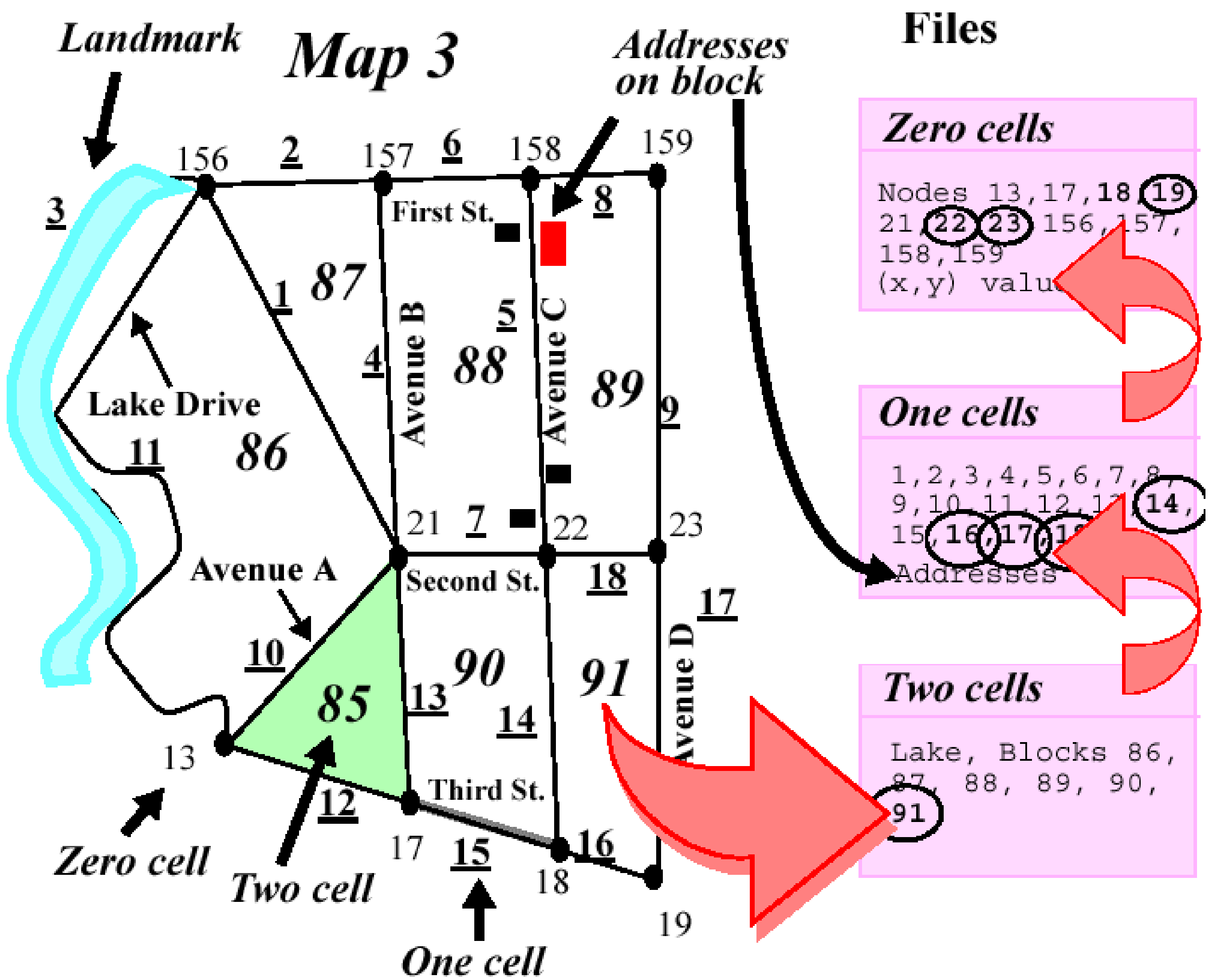
FORMATS

- Most GIS systems can **import different data formats**, or use utility programs to **convert** them.
- **Data formats** can be industry standard, commonly accepted or standard.

Vector Data Formats

- Vector formats are either **page definition languages** or **preserve ground coordinates**.
- **Page languages** are HPGL, PostScript, and Autocad DXF.
- **True vector GIS data formats** are DLG and TIGER, which has **topology**.

The TIGER data structure



Vector Data Formats

- **DXF** - Contour elevation plots in AutoCAD DXF format
- **SHP** - ESRI's vector data format using SHP, SHX and DBF files
- **Simple Features** - Open Geospatial Consortium specification for vector data
- **TAB** - MapInfo's vector data format using TAB, DAT, ID and MAP files
- **NTF** - National Transfer Format (mostly used by the UK Ordnance Survey)
- **TIGER** - Topologically Integrated Geographic Encoding and Referencing
- **XYZ** - Simple point cloud
- **VPF** - NIMA's Vector Product Format - the format of vectored data for large geographic databases (NIMA – National Geospatial Intelligence Agency)

Raster Data Formats

- Most raster formats are **digital image formats**.
- Most GISs accept TIF, GIF, JPEG or encapsulated PostScript, which **are not georeferenced**.
- **DEMs** are **true** raster data formats.

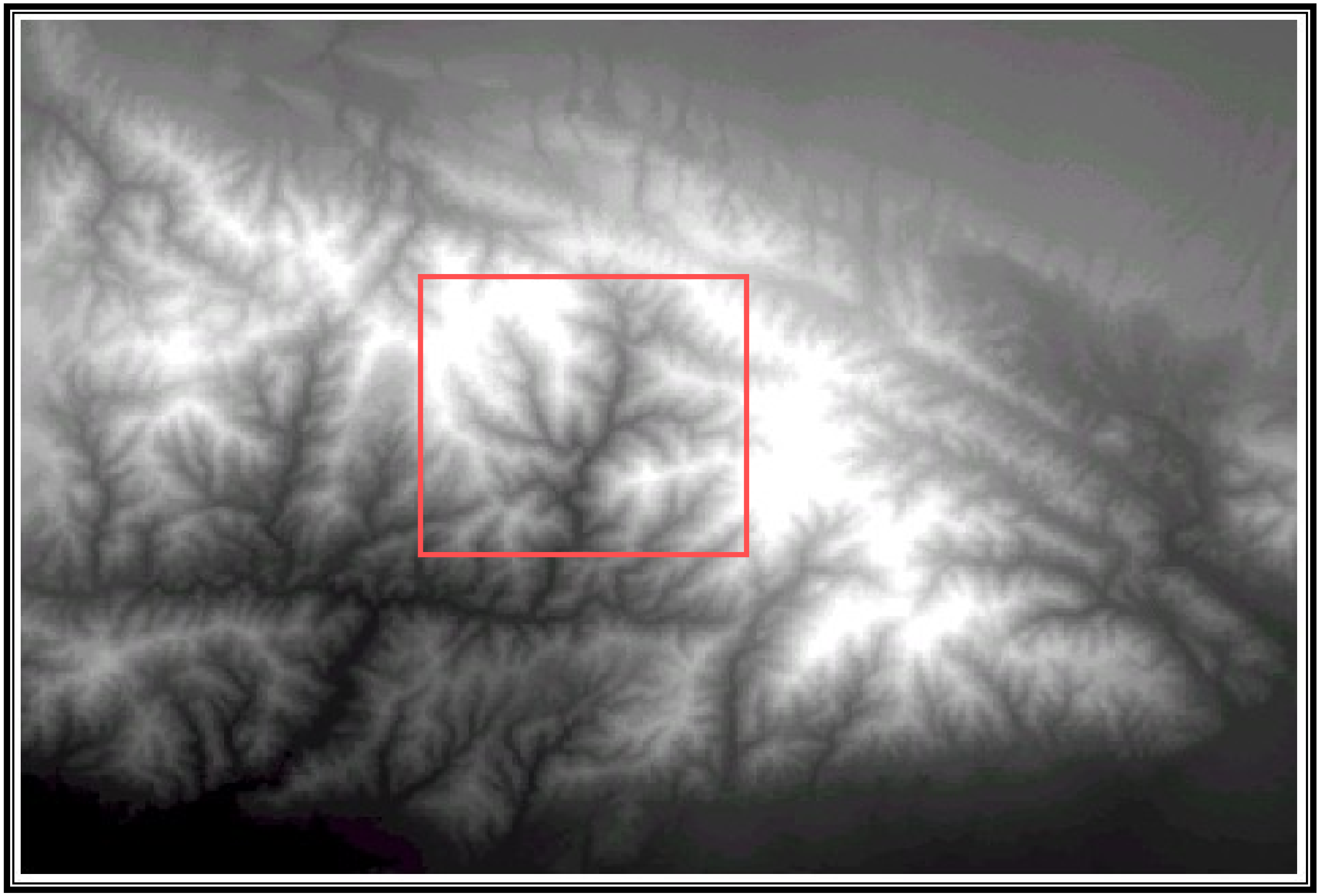
Common Used Raster Data Formats

- **ADRG** - NIMA's ARC Digitized Raster Graphics
- **BIL** - Band Interleaved by Line (image format linked with satellite derived imagery)
- **CADRG** - NIMA's Compressed ARC Digitised Raster Graphics (nominal compression of 55:1 over ADRG)
- **CIB** - NIMA's Controlled Image Base (type of Raster Product Format)
- **DRG** - digital scan of a paper USGS topographic map
- **ECW** - Enhanced Compressed Wavelet (from ERMapper). A compressed wavelet format.
- **GeoTIFF** - TIFF variant enriched with GIS relevant metadata
- **GRID** – ESRI's native raster format
- **IMG** - ERDAS IMAGINE image file format
- **MrSID** - Multi-Resolution Seamless Image Database (by Lizardtech). A compressed wavelet format.

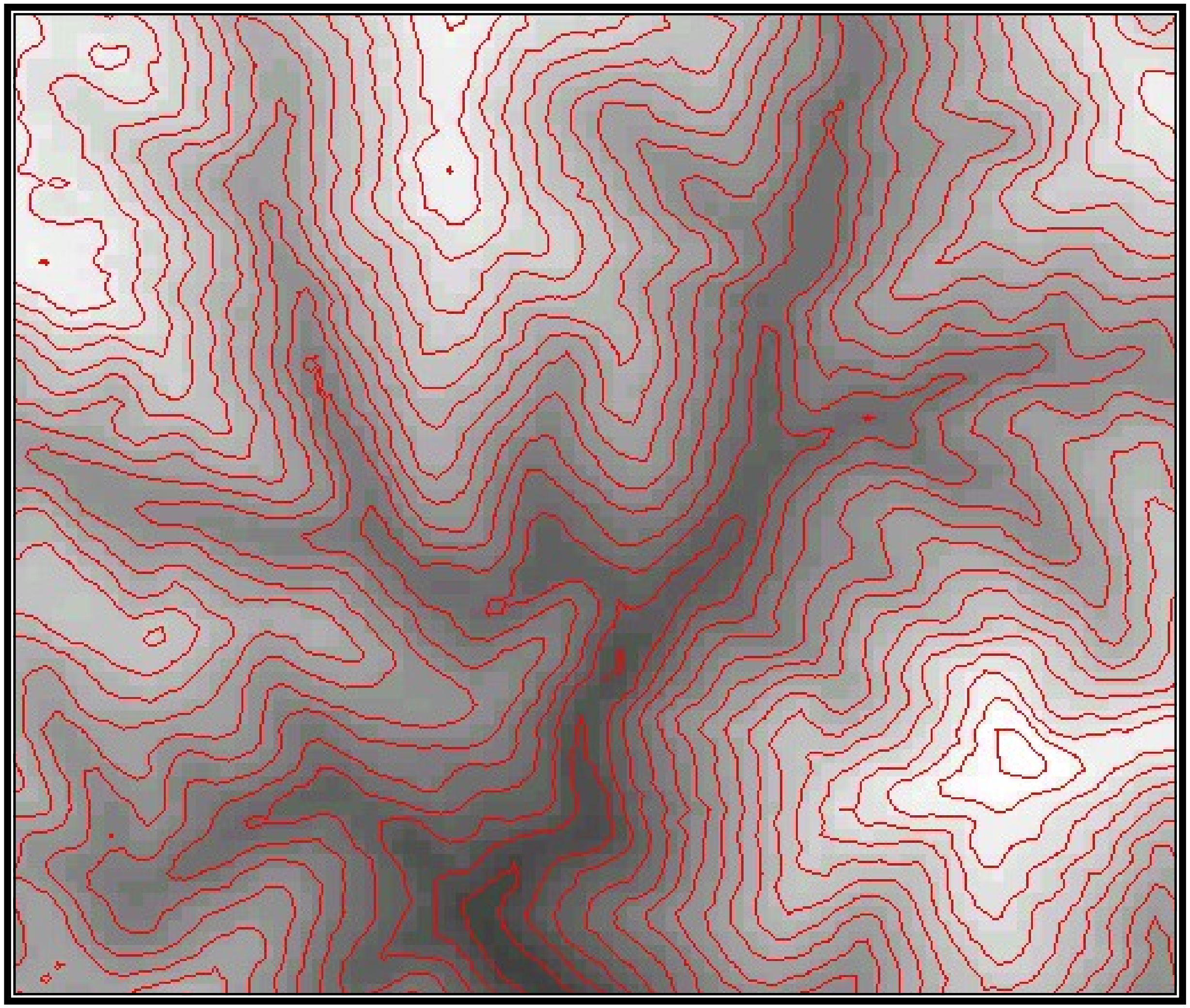
Digital Elevation Model

- A **digital elevation model** (DEM) is a representation of the topography of the Earth or another surface in digital raster format, that is, by coordinates and numerical descriptions of altitude.
- Digital elevation models may be prepared in a number of ways, but they are **frequently obtained by remote sensing** rather than direct survey
- USGS DEM/USGS STDS DEM (http://en.wikipedia.org/wiki/USGS_DEM)

Digital Elevation Model



Digital Elevation Model

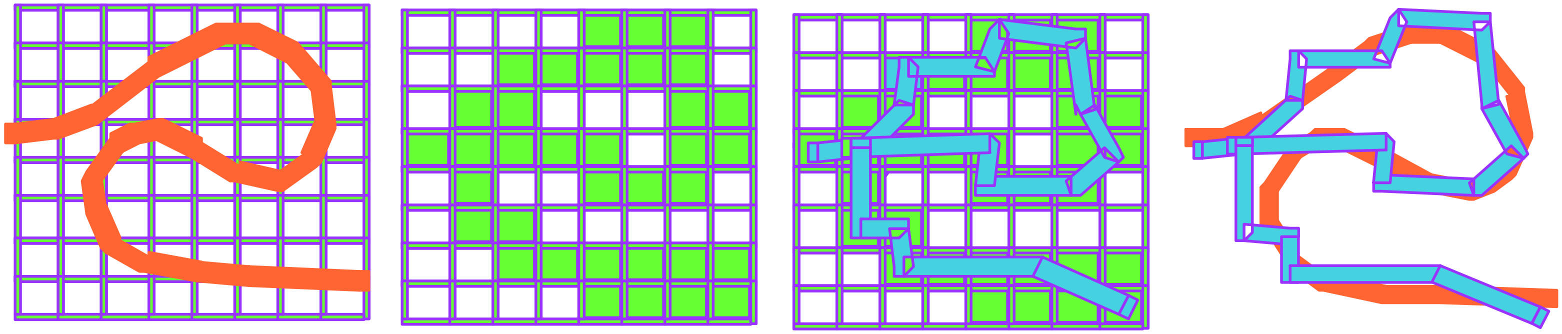


EXCHANGE

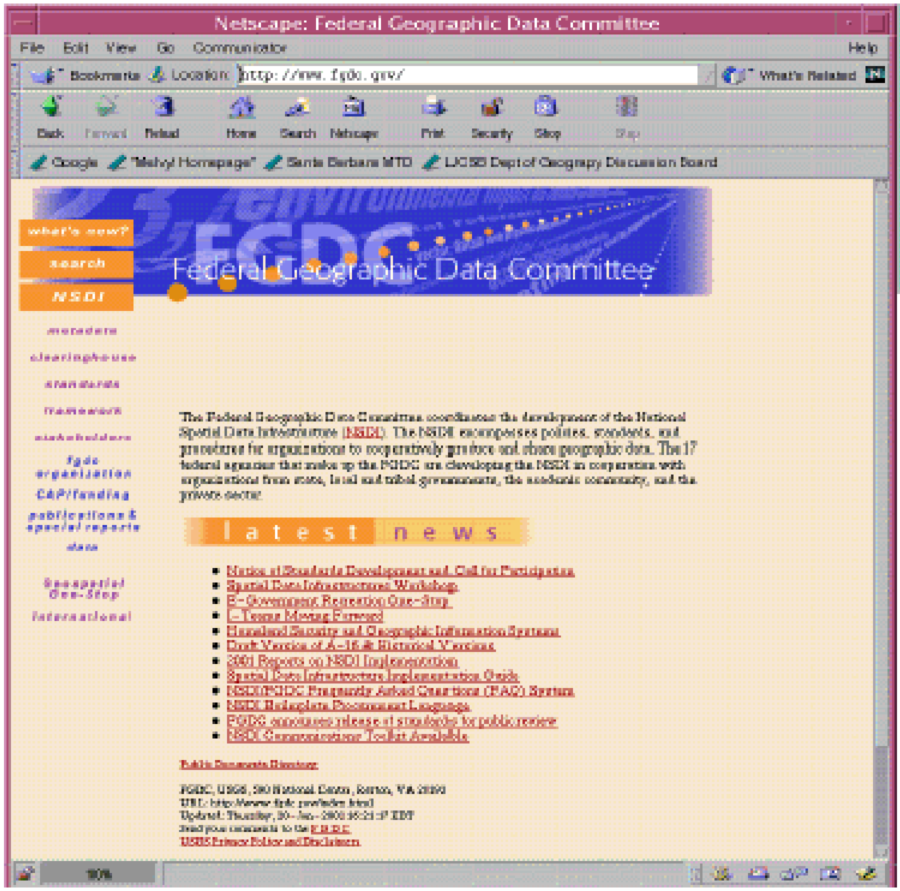
- Most GISs use **many formats** and **one data structure**.
- If a GIS supports many data structures, changing structures becomes the **user's responsibility**.
- Changing **vector to raster is easy**; **raster to vector is hard**.
- Data also are often exchanged or transferred between **different GIS packages and computer systems**.
- The history of GIS data exchange is **chaotic and has been wasteful**.
- **Standards** have been developed now by FGDC and other organizations to **improve interoperability**.

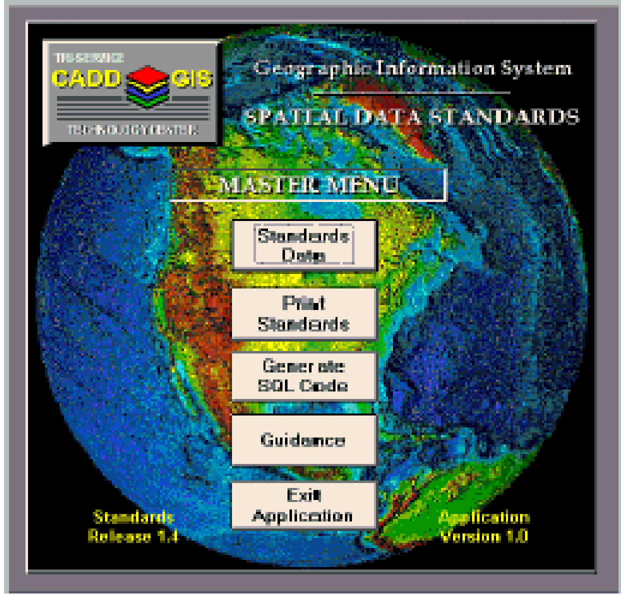
Vector to Raster Transformations

- Quite often, data in the **vector and raster** models need to be **used together**, and data from one model is **transformed** to be represented in the other model
- Any such transformations can cause **distortions** – consider this line transformed from vector to raster, and then back to vector:



Spatial Data Transfer Standards





http://www.fgdc.gov

GIS Data Exchange

- Data exchange by translation (export and import) can lead to **significant errors** in attributes and in geometry.
- In the United States, the **SDTS** was evolved to facilitate data transfer.
- SDTS became a **federal standard** (FIPS 173) in 1992.
- SDTS **contains** a terminology, a set of references, a list of features, a transfer mechanism, and an accuracy standard.
- Both **DLG and TIGER data** are available in SDTS format.
- **Other standards efforts** are DIGEST, DX-90, the Tri-Service Spatial Data Standards, and many other international standards.
- Efficient data exchange is **important** for the future of GIS.

Chapter 3: Maps as Numbers

- 3.1 Representing Maps as Numbers
- 3.2 Structuring Attributes
- 3.3 Structuring Maps
- 3.4 Why Topology Matters
- 3.5 Formats for GIS Data
- 3.6 Exchanging Data

Next Topic:

Getting the Map into the Computer