Vector GIS

presented by:

Tim Haithcoat

University of Missouri

Columbia

With materials from:

Holly Dickinson, State University of New York at Buffalo

Introduction to Vector Data Model

- Based on vectors (as opposed to space-occupancy raster structures)
- Fundamental primitive is a point
- Objects are created by connecting points with straight lines
  - Some systems allow points to be connected using arcs of circles

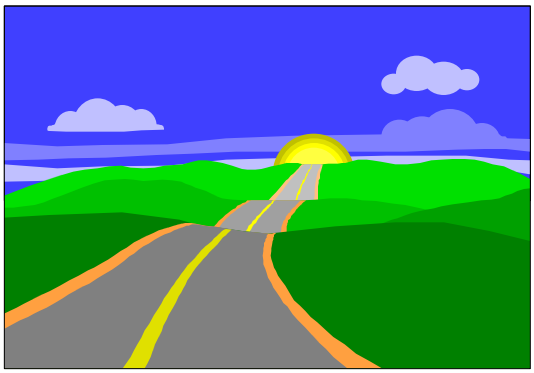
Introduction Continued

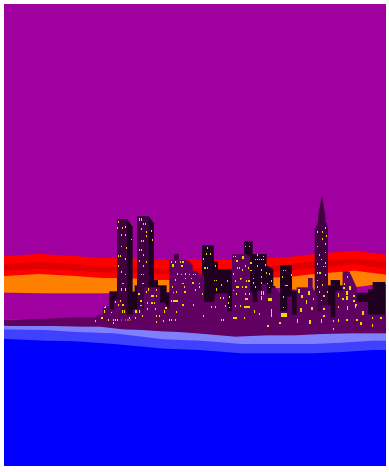
- Areas are defined by sets of lines
  - The term polygon is synonymous with area in vector databases because of the use of straight line connections between points
- Very large vector databases have been built for different purposes
  - Vector tends to dominate in transportation, utility, marketing applications
  - Raster & vector both used in resource management applications

Example: Vector GIS Data

SCALE ↓

point	line	area
city wells	highway political boundary streams	ag. land urban land city highway airport





Example Attributes:

city:	population, name
wells:	depth
highway:	number
political boundary:	type
streams:	name
ag. land:	growth potential, acreage
urban land:	urban landuse type, acreage
airport:	name

"ARCS"

- When planar enforcement is used, area objects in one class or layer cannot overlap and must exhaust the space of a layer
- Every piece of boundary line is a common boundary between two areas
- The stretch of common boundary between two junctions (nodes) has various names
  - **Edge:** favored by graph theorists, "vertex" for the junctions
  - **Chain:** word officially sanctioned by the US National Standard
  - **Arc:** used by several systems

"ARCS" Continued

- Arcs have attributes which identify the polygons on either side
  - these are referred to as "left" and "right" by reference to the sequence in which the arc coded
- Arcs (chains/edges) are fundamental in vector GIS

Two Storing Areas

POLYGON STORAGE

- Every polygon is stored as a sequence of coordinates
- Although most boundaries are shared between two adjacent areas, all are input and coded twice, once for each adjacent polygon
- The two different versions of each internal boundary line may not coincide
- Difficult to do certain operations (i.e., dissolve boundaries between neighboring areas and merge them
- Used in some current GISs, many automated mapping packages

Two Storing Areas

ARC STORAGE

- Every arc is stored as a sequence of coordinates
- Areas are built by linking arcs
- Only one version of each internal shared boundary is input and stored
- Used in most current vector-based GISs

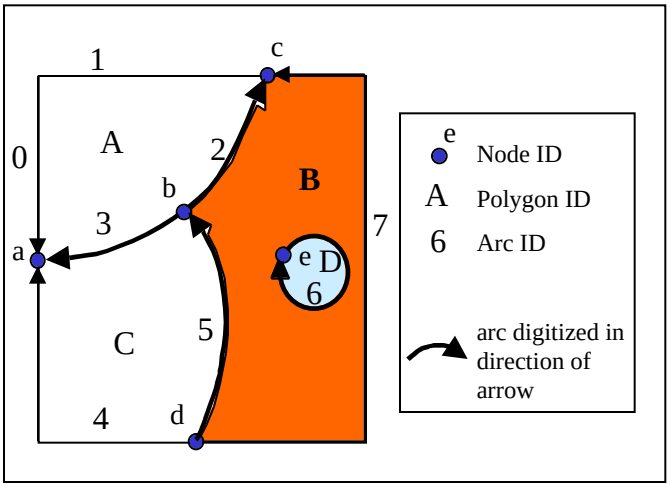
Database Creation

- Involves several stages:
  - Input of the spatial data
  - Input of attribute data
  - Linking spatial and attribute data
- Spatial data is entered via digitized points and lines, scanned and vectorized lines or directly from other digital sources
  - Once the spatial data has been entered, much work is still needed before it can be used

Building Topography

- Once points are entered and geometric lines are created, topology must by “built”
  - This involves calculating and encoding relationships between the points, lines and areas
  - This info may be automatically coded into tables of information in the database
  - Let’s look at an example

Example of “Built” Topology (ARC/INFO)



Arc ID	Left Poly	Rt Poly	From Node	To Node
1	A	0	c	a
2	A	B	b	c
3	C	A	b	a
4	0	C	d	a
5	C	B	d	b
6	B	D	e	e
7	B	0	d	c

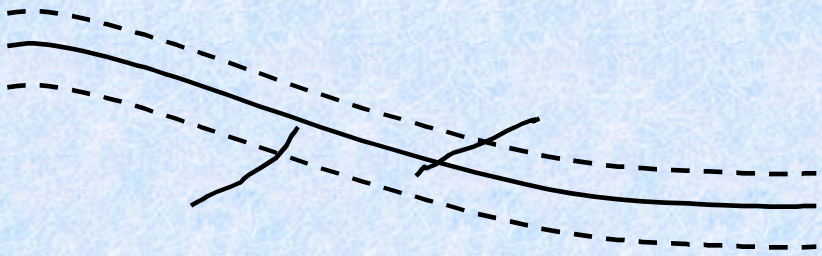
Polygon ID	No. of Arcs	List of Arcs
A	3	-1,-2,3
B	4	2,-7,5,0,-6
C	3	-3,-5,4
D	1	6

Editing

- During this topology generation process, problems such as overshoots, undershoots and spikes are either flagged for editing by the user or corrected automatically
  - Automatic editing involves the use of a tolerance value which defines the width of a buffer zone around objects within which adjacent objects should be joined

Editing - continued

- Tolerance value is related to the precision with which locations can be digitized



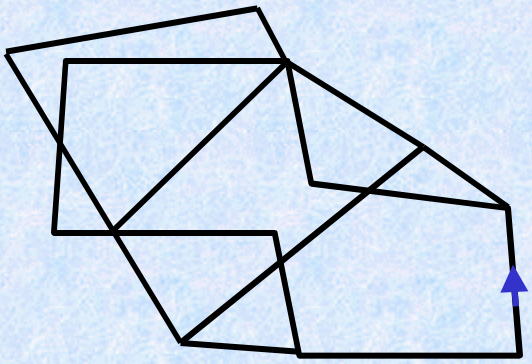
- These edit procedures include such functions as snap, move, delete, split, join, etc.

Relationship between Digitizing & Editing

- Digitizing and editing are complementary activities
  - Poor digitizing leads to much need for editing
  - Good digitizing can avoid most need for editing
  - Both can be very labor-intensive
- The process used to digitize area objects can affect the need for later editing

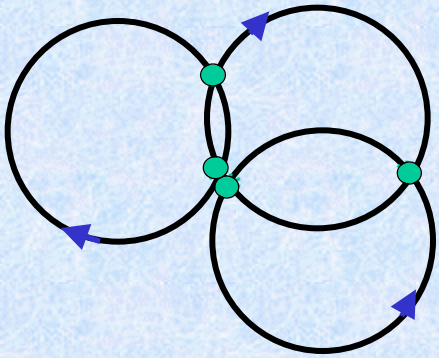
Relationship between Digitizing & Editing
(continued)

- In “blind” digitizing all linework is digitized once as “noodles” in any order
  - It is unlikely that the building and cleaning operations will be able to automatically sort out area objects unambiguously from the resulting jumble



Relationship between Digitizing & Editing
(continued)

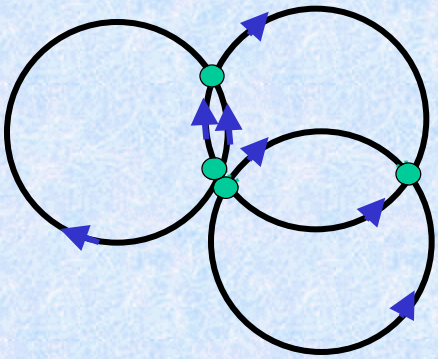
- Some systems require the user to identify junctions between digitized “noodles” explicitly
  - Usually by touching a special button on the cursor



  - Mistakes in building topology are less likely

Relationship between Digitizing & Editing
(continued)

- Some systems require the user to digitize each individual arc/chain separately



  - Much easier to sort our polygons - less need for editing

Relationship between Digitizing & Editing
(continued)

- Some systems support the building of topology "on the fly"
  - The system searches constantly for complete area objects as digitizing proceeds
  - The users is informed by a sound or by blinking as soon as the object is detected

Edgematching

- Compares and adjusts features along the edges of adjacent map sheets
- Some edgematches merely move objects into alignment
- Others “join” the pieces together logically - conceptually they become one object
  - The user “sees” no interruption
- An edgematched database is “seamless” - the sheet edges have disappeared as far as the user is concerned

Adding Attributes

- Once the objects have been formed by building topology, attributes can be keyed in or imported from other digital databases
- Once added to the database, attributes must be linked to the different objects
  - Attributes can be linked by pointing to the appropriate object on the screen and coding its corresponding object ID into the attribute table
- Unlike many raster GIS systems, attribute data is stored and manipulated in entirely separate ways from the locational data

Example Analysis using Vector GIS

- OBJECTIVE:Identify areas suitable for logging
- An area is suitable if it satisfies the following criteria:
  - is Jackpine (Black Spruce are not valuable)
  - is well drained (poorly drained and waterlogged terrain cannot support equipment, logging causes unacceptable environmental damage)
  - is not within 500 m of a lake or watercourse (erosion hazard)

Analysis Steps

- Buffer hydrography out to 500 m
- Merge buffer and lake
- Extract Jack pine polygons (species = Jack pine)
- Extract drained soil polygons
  (drainage = 2, therefore soil = A)
- Overlay buffer, Jack pine and soil polygons
- Build topology
- Extract polygons not in the buffer but in others
  (buffer = n, Jack pine = y, drainage = y)
- RESULT: loggable area shown in final map

Vector GIS Capabilities

- Analysis functions with vector GIS are not quite the same as with raster GIS
  - More operations deal with objects
  - Measures such as area have to be calculated from coordinates of objects, instead of counting cells
- Some operations are more accurate
  - Estimates of area based on polygons more accurate than counts of pixels
  - Estimates of perimeter of polygon more accurate than counting pixel boundaries on the edge of a zone

Vector GIS Capabilities

- Some operations are slower
  - Examples: Overlaying layers, finding buffers

- Some operations are faster
  - Example: Finding patch through road network

Simple Display

- Using points and “arcs” can display the locations of all objects stored
- Attributes and entity types can be displayed by varying colors, line patterns and point symbols
- May only want to display a subset of the data
  - Example: want to display areas of urban land use with some base map data
    - Select call political boundaries and highways, but only areas that had urban land uses

Simple Display (continued)

- How would the user do this?
  - E.g. One of the layers in a database is a “map” of land use, called USE
  - Area objects on this layer have several attributes
  - One attribute, called CLASS, identifies the area’s land use
  - For urban land use, it has the value “U”
  - Need to extract boundaries for all areas that have CLASS = “U”

Standard Query Language (SQL)

- Different systems use different ways of formulating queries
- SQL is used by many systems
- SQL phrase structure:
  - SELECT, attribute name(s)>FROM<table> WHERE<condition statement>
    - Ex: SELECT FROM USE WHERE CLASS="U"
    - This selects only the objects for display - no attributes are retrieved by the query

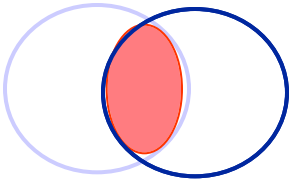
SQL (continued)

- SQL phrase structure (continued)
  - SQL examples using a list of student names
    - SELECT name FROM list (selects all names)
    - SELECT name FROM list WHERE grade = “A” (selects names of students receiving an “A”)
    - SELECT name FROM list WHERE cumgrade > 3.0 (selects names of students with a cumulative gpa greater than 3.0)
  - SQL operators:
    - Relational: <, >, =, <= ,>=
    - Arithmetic: =, -, *, / (only on numeric fields
    - Boolean: and, or, not

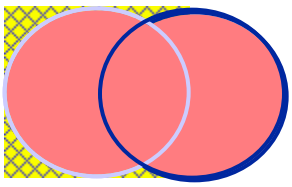
Boolean Operators

- Used to combine conditions
  - Ex: WHERE cumgrade > 3.0 AND grade = “A” (selects students satisfying both conditions only)
- Can have spatial meaning in GIS as well
  - Ex: when two maps are overlayed, areas (polygons) that are superimposed have the “and” condition
- A spatial representation is used to illustrate Boolean operators in the study of logic, through the use of diagrams called Venn diagrams
  - Thus GIS area overlay is a geographical instance of a Venn Diagram

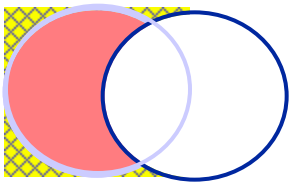
Boolean Operators



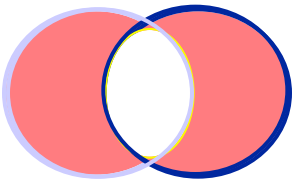
A AND B



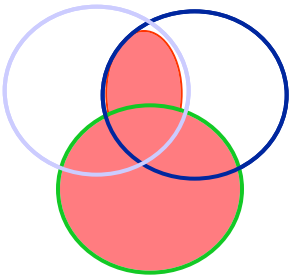
A OR B



A NOT B



A XOR B



(A AND B) OR C

SQL Extensions for Spatial Queries

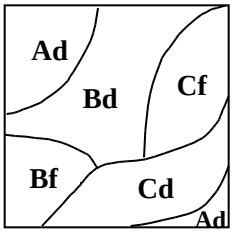
- Some systems allow specifically spatial queries to be handled under SQL
  - Ex: WITHIN operator
    - SELECT <objects> WITHIN <specific area>
- The criteria for these spatial searches may include searching within the radius of a point, within a bounding rectangle, or within an irregular polygon

Reclassify, Dissolve, & Merge

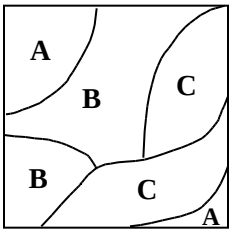
- The operations are used frequently in working with area objects
  - These are used to aggregate areas based on attributes
- Consider a soils map:
  - We wish to produce map of major soil types from a layer that has polygons based on much more finely defined classification scheme

Steps:

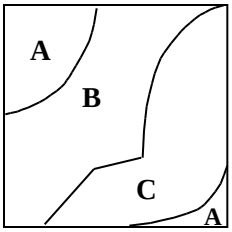
1. **Reclassify** soil areas by soil type only.
2. **Dissolve** boundaries between areas of same soil type.
3. **Merge** polygons into large objects.



Soil Types A, B and C with growth potentials d and f



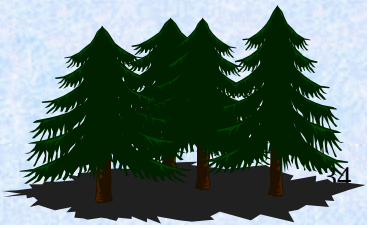
Soil Types A, B and C



Soil Types A, B, and C

Reclassify, Dissolve & Merge: Forestry Example (1 of 2)

- Consider a forestry GIS where the forest is divided into “stands”, average size 10 ha:
  - Each stand carries a list of attributes, including tree species and average tree age
  - Attributes apply homogeneously to area of each stand
  - Boundary occurs between stands whenever at least one attribute changes

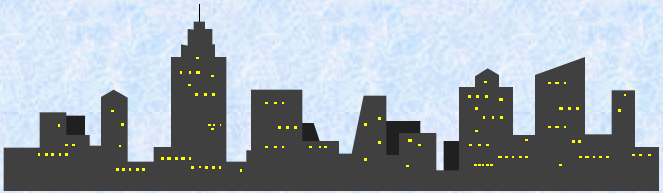


Reclassify, Dissolve & Merge: Forestry Example (2 of 2)

- Problem: identify all cuttable areas of white spruce
  - Assign new attribute “cuttable” to each stand
    - Value = “y” if white spruce AND age > 50 years
    - Value = “n” otherwise
  - After assigning new attribute, all others can be dropped
- Now you wish to identify cuttable areas, each may be merger of several individual stands
  - Dissolve boundaries between polygons with same value of “cuttable” attribute
  - Merge polygons into larger objects

Reclassify, Dissolve & Merge:
City Zoning Example

- Need to know how many individual landuse zones have been created in the city and how these are distribute geographically
- Each land parcel in the city has a zoning attribute attached to it
- Dissolve boundaries between parcels if the zoning is the same
- Result can be a map showing larger areas of similar zoning classes



Topological Overlay

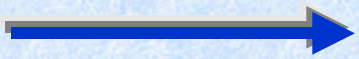
- Suppose individual layers have planar enforcement (required in many systems, not all)
- When two layers are combined ("overlayed", "superimposed") the result must have planar enforcement as well
  - New intersection must be calculated and created wherever two lines cross
  - A line across an area object creates two new area objects
- Topological overlay is the general name for overlay followed by planar enforcement

Topological Overlay (continued)

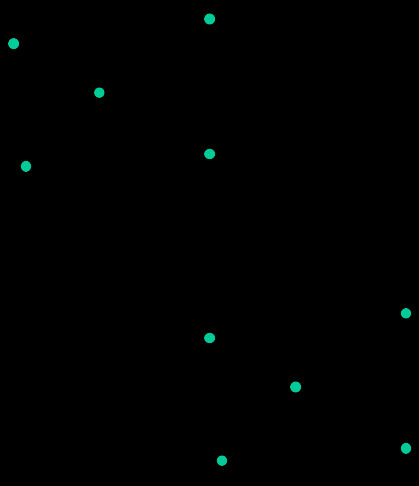
- Relationships are updated for the new, combined map
- Result may be information about relationships (new attributes) for the old (input) maps rather than the creation of new objects
  - Ex: overlay map of school districts on census tracts
    - Result is map showing every school district/census tract combination
    - For each combo, the database contains an area objects
    - However, concern may be with obtaining the number of overlapping census tracts as a new attribute of each school district rather than with new objects themselves

Point in Polygon

- Overlay point objects on areas, compute “is contained in” relationship
- Result is a new attribute for each point
  - Example: combine wells and planning districts, find district containing each well



Point in Polygon: Example

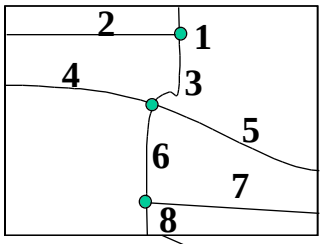


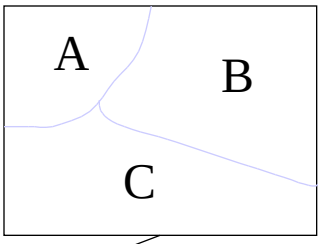
Line on Polygon

- Overlay line objects on area objects, compute "is contained in" relationships
- Lines are broken at each area object boundary
  - Number of output lines is greater than number of input lines
- Containing area is new attribute of each output line
  - Example: combine roads and counties, find county containing each road segment

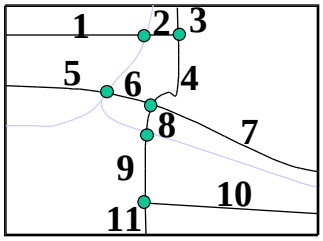
Line on Polygon: Example

ID	Highway
1	35
2	22
3	35
4	60
5	60
6	35
7	82
8	35





ID	County
A	Black
B	Cole
C	Fall

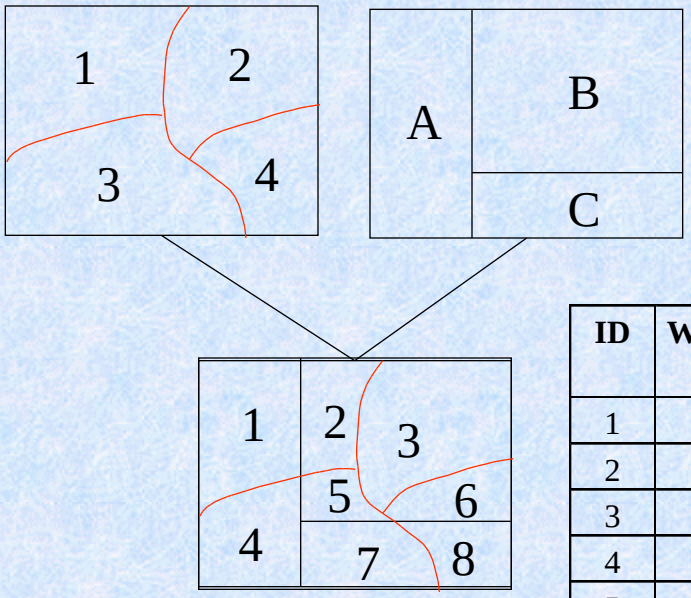


ID	Original	Hwy	County
1	2	22	Black
2	2	22	Cole
3	1	35	Cole
4	3	35	Cole
5	4	60	Black
6	4	60	Cole
7	5	60	Cole
8	6	35	Cole
9	6	35	Fall
10	7	82	Fall
11	8	35	Fall

Polygon on Polygon
("Polygon Overlay")

- Overlay two layers of area objects
- Boundaries are broken at each intersection
- Number of output areas likely greater than the total number of input areas
  - Example: input watershed boundaries, county boundaries, output map of watershed/county combinations
  - After overlay we can recreate either of the input layers by dissolving and merging based on the attributes contributed by the input layer

Polygon on Polygon: Example



ID	Watershed ID	County ID
1	1	A
2	1	B
3	3	B
4	2	A
5	2	B
6	4	B
7	2	C
8	4	C

Example: Topological Overlay

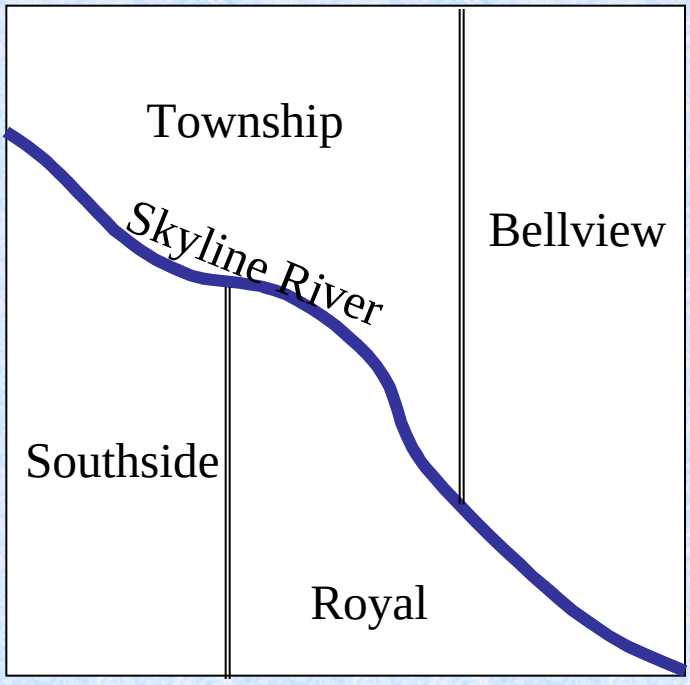
- Wish to use those areas that are the best land for timber harvesting
- After overlay, each original layer contributes attributes to the combined layer
- We get the final map by electing the desired attributes of the combined layer
  - SELECT FROM OVERLAY WHERE Species = “Jackpine” AND Soil = “C”

Spurious Polygons

- During polygon overlay, many new and smaller polygons are created, some of which may not represent true spatial variations
- See example
- The small, invalid polygons are called spurious or sliver polygons and can be a major problem in polygon overlay

Sliver or Spurious Polygons

(page 1 of 3)

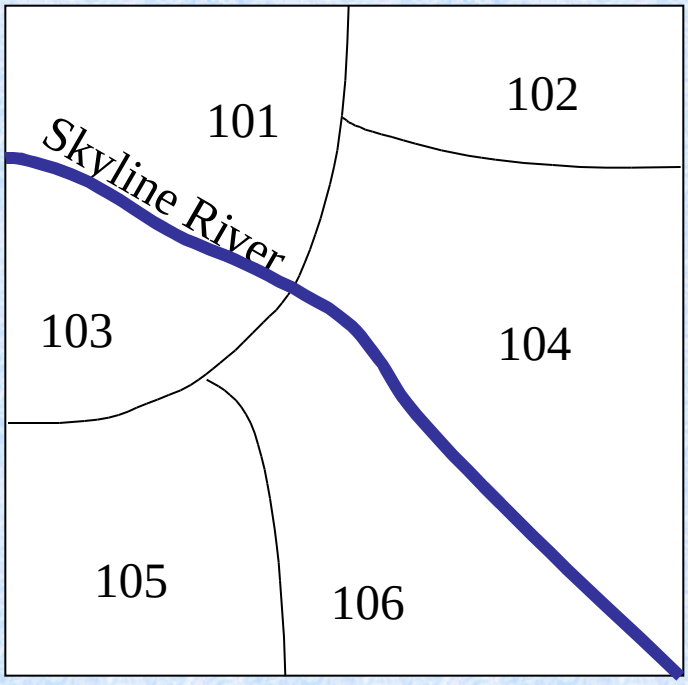


Skyline River

School District Boundaries

Sliver or Spurious Polygons

(page 2 of 3)

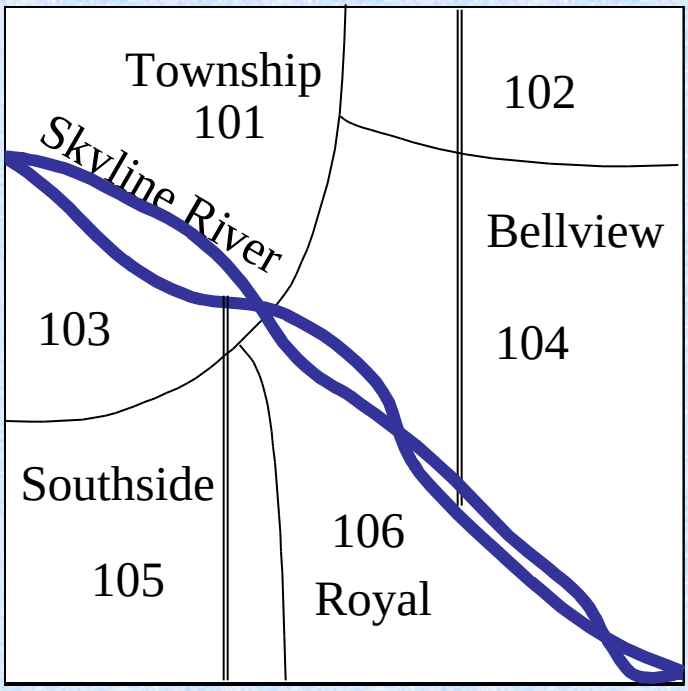


Skyline River

Census Boundaries

Sliver or Spurious Polygons

(page 3 of 3)



Skyline River

Census Boundary

School District Boundaries

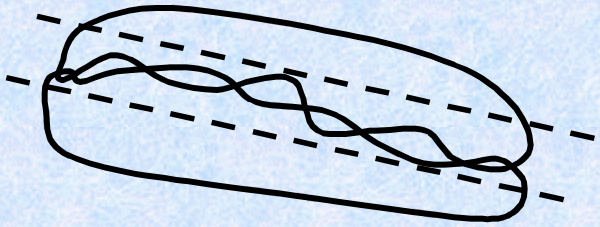
Spurious Polygons

Spurious Polygons (continued)

- Spurious polygons arise when two lines are overlaid which are actually slightly different versions of the same line
  - If the same line occurs on two input maps, the digitized versions may be slightly different
  - In many cases the lines on the source maps have been compiled from different sources, but are nevertheless the same line on the ground
  - Example: a road may be part of a county boundary, also the boundary between two fields or two soil types or two vegetation types
- The problem cannot be removed by more careful digitizing - more points simply lead to more slivers

Spurious Polygons (continued)

- Some GISs allow the user to set a tolerance value for deleting spurious polygons during overlay operations

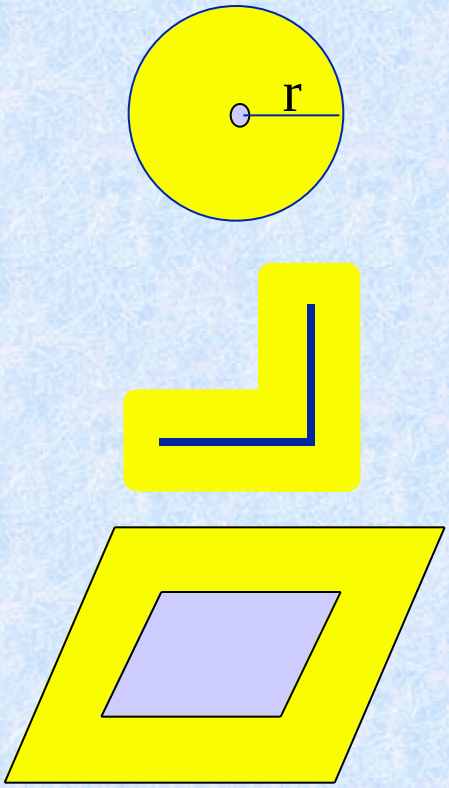


  - If tolerance is set too high, some legitimate polygons may be deleted
  - If set too low, some erroneous polygons will remain
  - Deletion rules might also be based on shape, as spurious polygons tend to be long and think

Buffering

- A buffer can be constructed around a point, line or area
- Buffering creates a new area, enclosing the buffered object
- Applications in transportation forestry, resource management
  - Protected zones around lakes and streams
  - Zone of noise pollution around highways
  - Service zone around bus route (ex: 300 m walking distance)
  - Groundwater pollution zone around waste site

Buffering



Buffering a Point

example: All area within one mile of the city.

Buffering a Line

example: All areas within 1000 meters of a road.

Buffering an Area

example: All areas within 500 meters of a wetlands area.

Buffering (continued)

- Options available for raster, such as a “friction” layer, do not exist for vector
- Sometimes, width of the buffer can be determined by an attribute of the object
  - Example: buffering residential buildings away from a street network:
    - Three types of street (1,2,3 or major, secondary, tertiary) with the setbacks being 600 feet from a major street, 200 feet from a secondary street, and only 100 feet from a tertiary street

Buffering (continued)

- Buffering is much more difficult in vector from the point of view of the programmer
- Problems with buffer operations may occur when buffering very convoluted lines or areas: