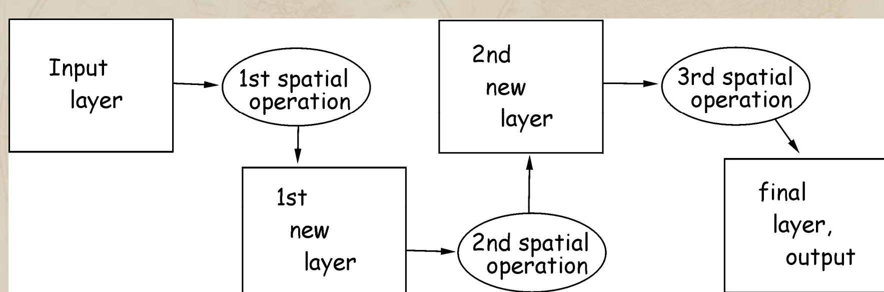
COA 690/790 GIS in Marine Science

Lecture 7
Basic Spatial Analysis

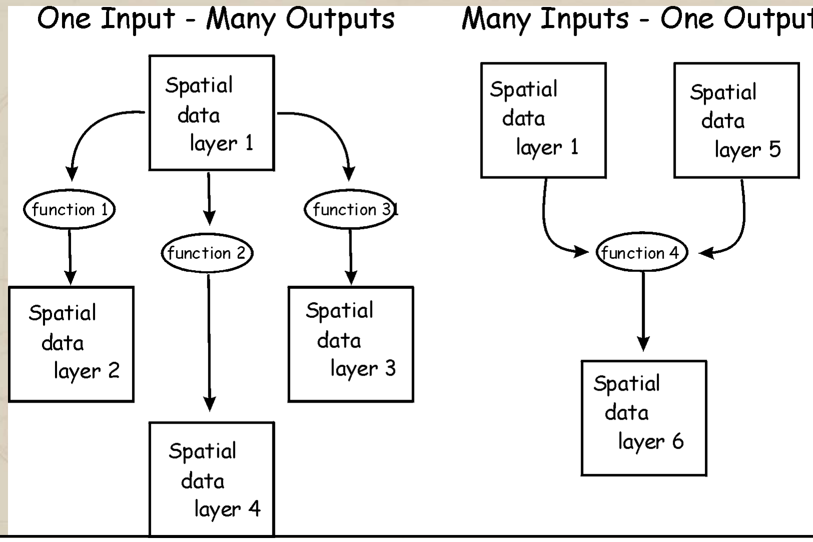
Wei Wu
March 19, 2019

Spatial data analysis

Input -> *spatial operation* -> output



Spatial data analysis
Input -> *spatial operation* -> output

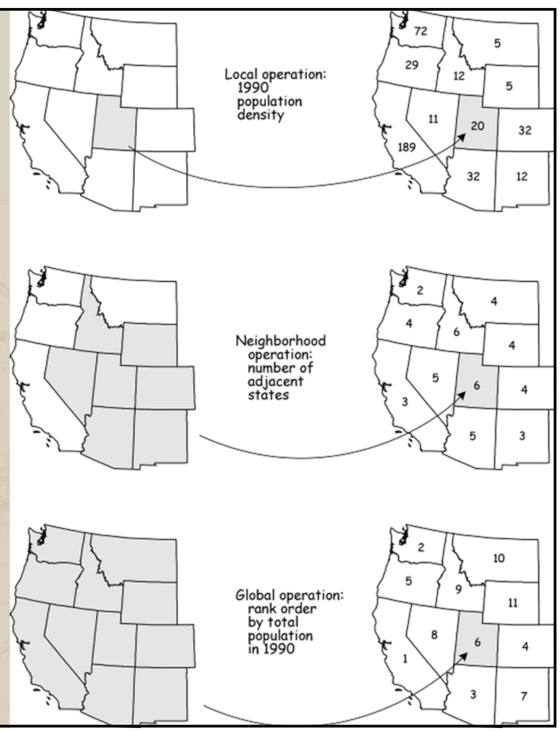


Input Scope

Local – "point" to "point"

Neighborhood – adjacent regions have input

Global – the entire input data layer may influence output



Spatial data analysis

Usually involves manipulations or calculation of coordinates or attribute variables with a various operators (tools), such as:

Selection
Reclassification
Dissolving
Buffering
Overlay
Cartographic Modeling (a combination of the above)

Spatial Selection

Identifying features based on spatial criteria

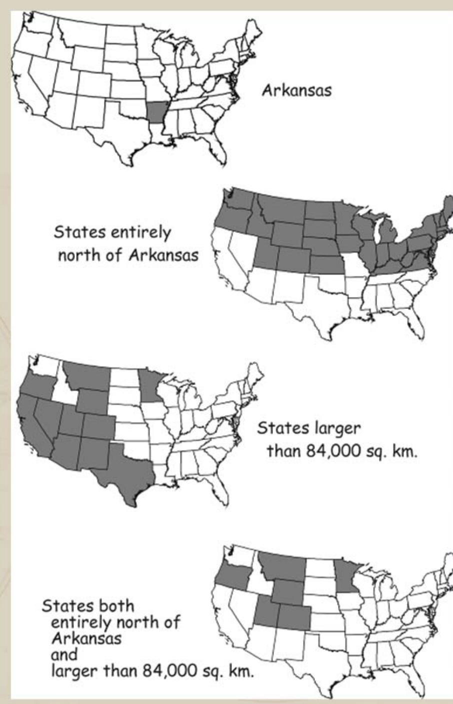
Adjacency, connectivity, containment, arrangement

Spatial Selection

Identifying features based on spatial criteria

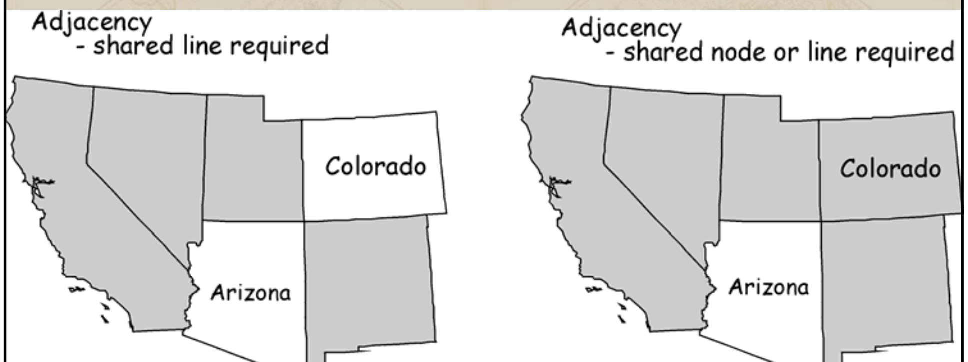
Adjacency, connectivity, containment, arrangement

Selection based on spatial and non-spatial attributes



Adjacency depends on the algorithm used

(the same is true for all spatial operations)



Spatial data analysis

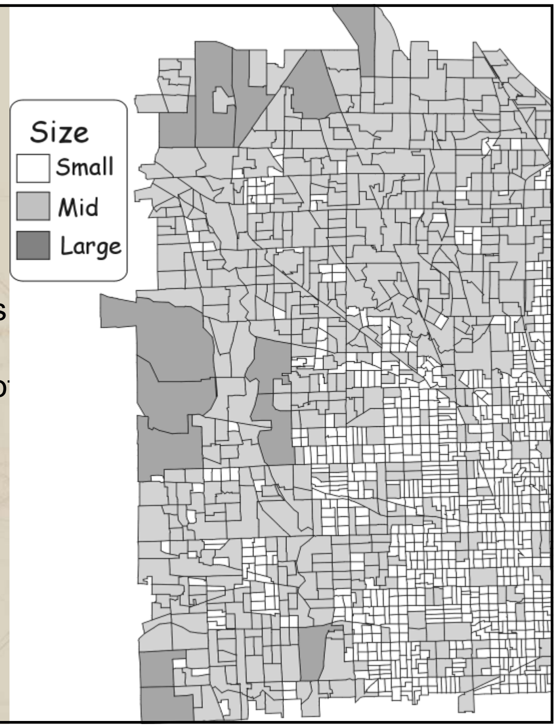
Usually involves manipulations or calculation of coordinates or attribute variables with a various operators (tools), such as:

Selection
Reclassification
Dissolving
Buffering
Overlay
Cartographic Modeling (a combination of the above)

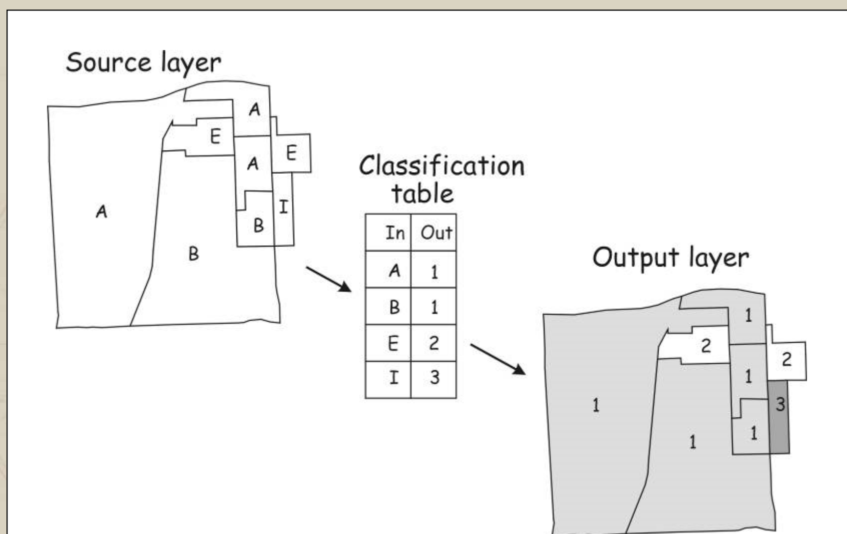
Spatial data analysis: Reclassification

An assignment of a class or value based on the attributes or geography of an object

Example:
Parcels
Reclassified
By size

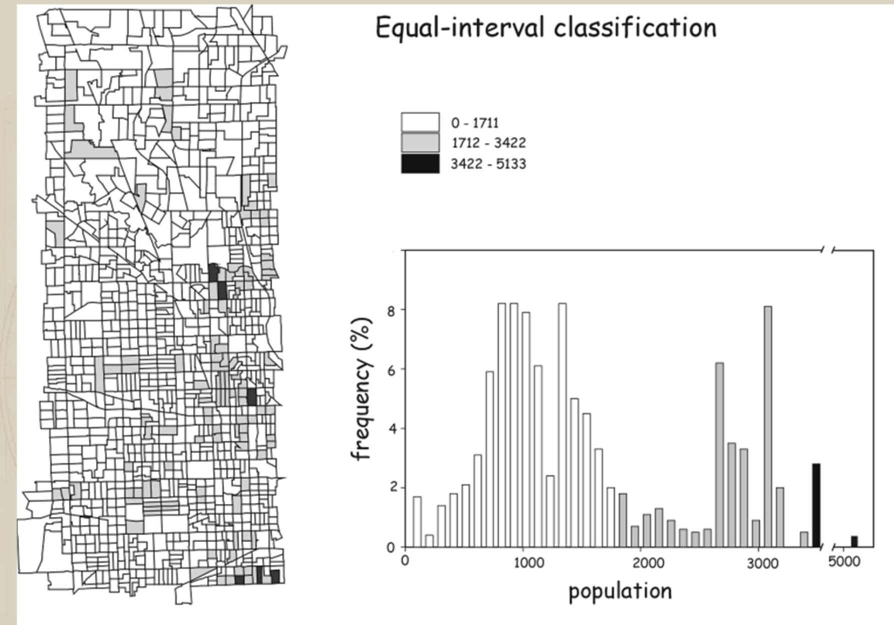


Spatial data analysis: Reclassification

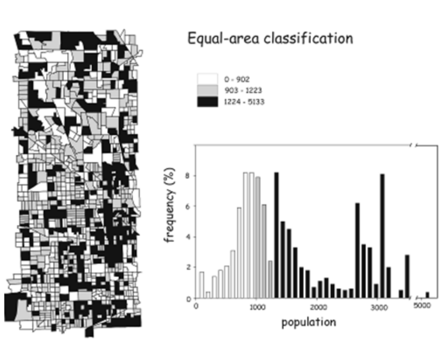


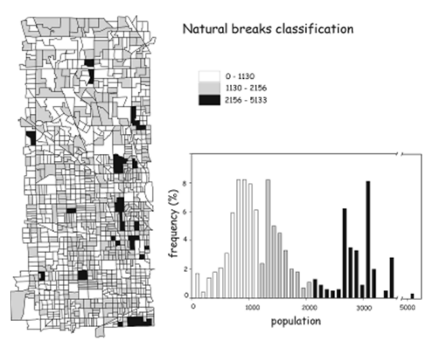
In	Out
A	1
B	1
E	2
I	3

Reclassification: defining categories



Spatial data analysis: reclassification defining categories





Spatial data analysis

Usually involves manipulations or calculation of coordinates or attribute variables with a various operators (tools), such as:

Selection
Reclassification
Dissolving
Buffering
Overlay
Cartographic Modeling (a combination of the above)

Spatial data analysis :dissolve

A function whose primary purpose is to combine like features within a data layer.

Adjacent polygons may have identical values. Dissolve removes or “dissolves away” the common boundary.

Used prior to applying area-based selection in spatial analysis

Figure 10-18: An example of a dissolve operation. Note the removal of lines separating polygons of the same size class. This greatly reduces the number of polygons.

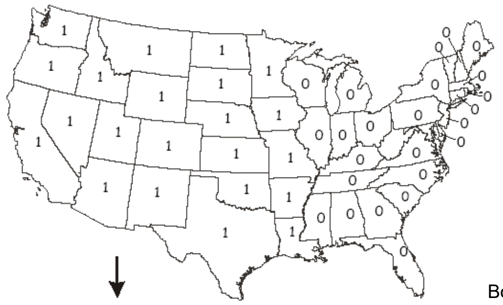
Dissolve: often used after reclassification

States west of the main branch of the Mississippi River assigned 1, east of the River assigned 0

Classification table

state name	is_west
Alabama	0
Arizona	1
Arkansas	1
Colorado	1
Connecticut	0
....	...
Wyoming	1

Dissolve: often used after reclassification



Dissolve operation

Dissolve Table

state name	is_west	dissolve value
Alabama	0	E
Arizona	1	W
Arkansas	1	W
Colorado	1	W
Connecticut	0	E
....	...	...
Wyoming	1	W

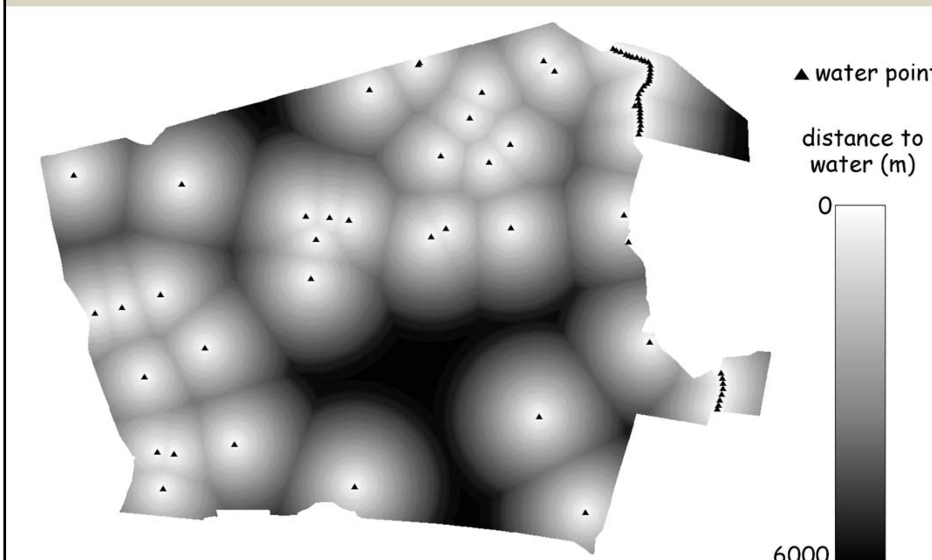
Boundaries are removed when they separate states with the same value.
The dissolve attribute is_west.

Spatial data analysis

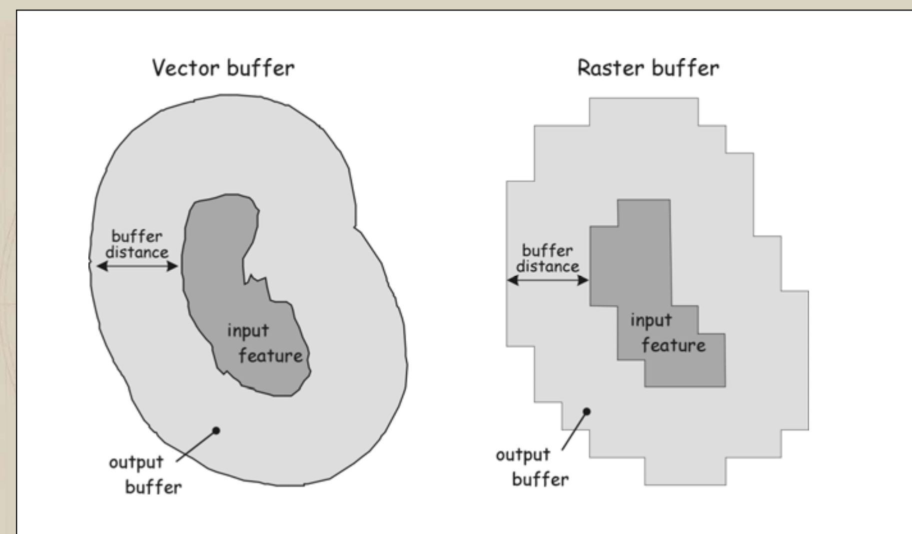
Usually involves manipulations or calculation of coordinates or attribute variables with a various operators (tools), such as:

Selection
Reclassification
Dissolving
Buffering
Overlay
Cartographic Modeling (a combination of the above)

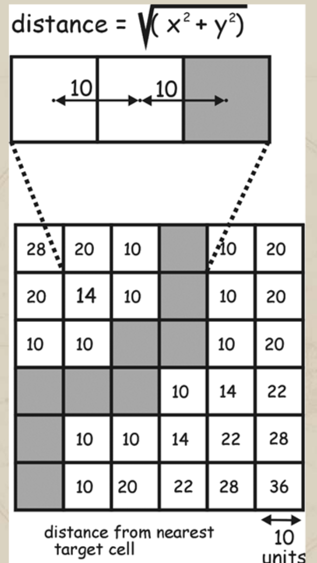
Buffering and other Proximity Functions

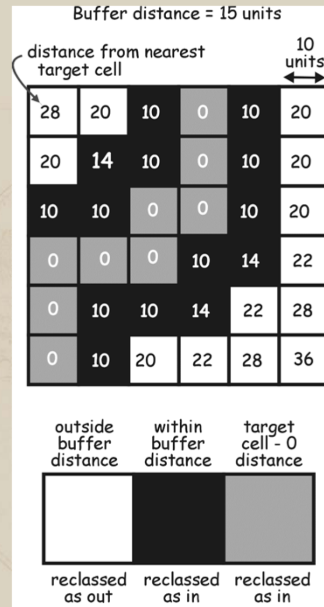


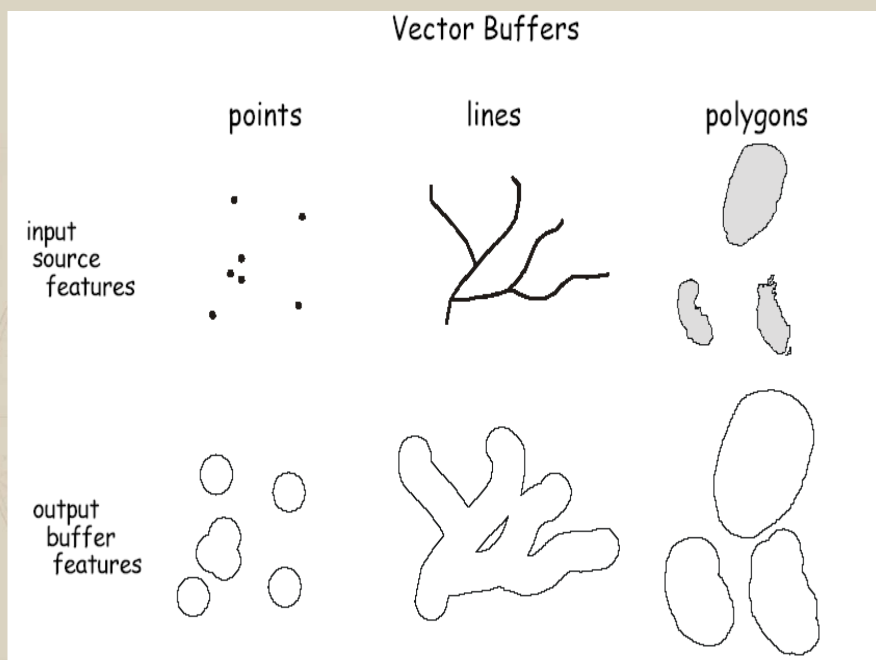
Buffering and other Proximity Functions



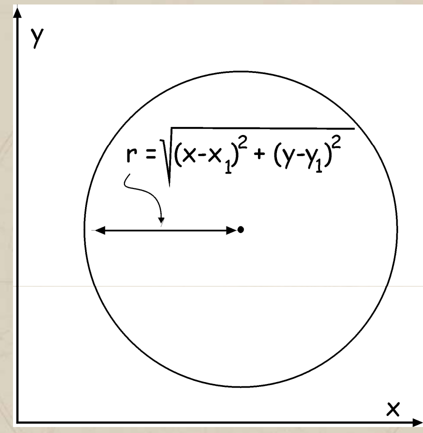
Raster buffer is an array of distances

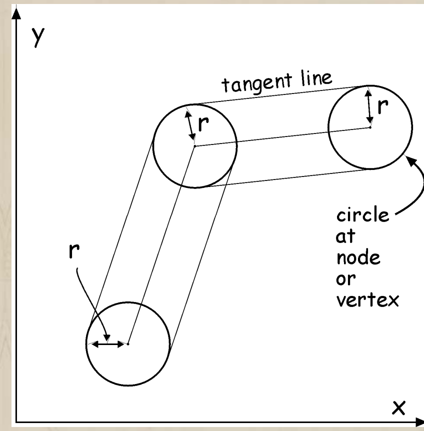




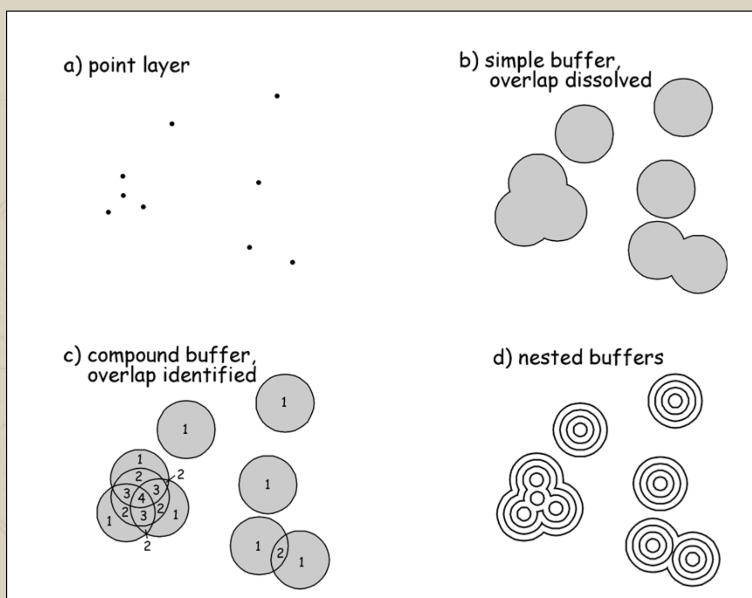


Mechanics of Point and Line Buffering





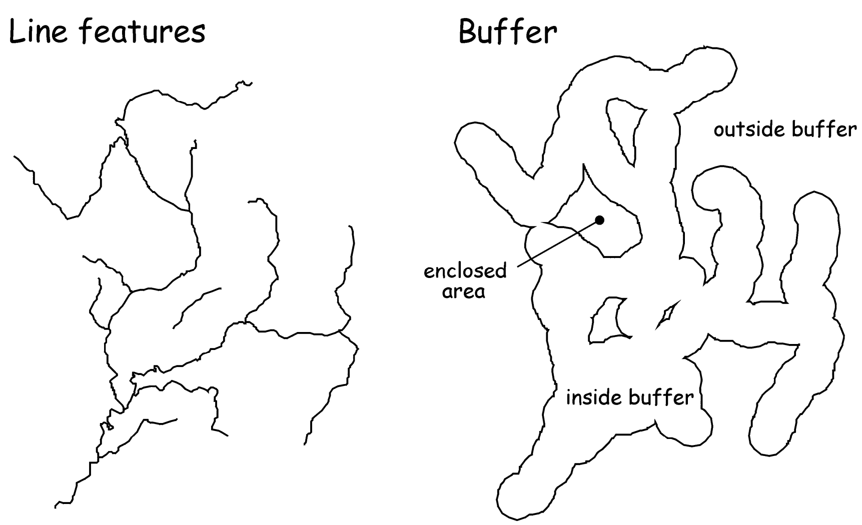
Buffering Variants: point buffer examples



Variable-distance buffer:
a line buffer is shown with a variable buffer distance, 100 km from main stem of the Mississippi River, 75 km from larger tributaries, and 50 km from remaining tributaries.

river_identifier	buffdist
mississippi	100
missouri	50
arkansas	50
ohio	75
tennessee	75

Regions in Buffering – inside, outside, enclosed



Spatial data analysis

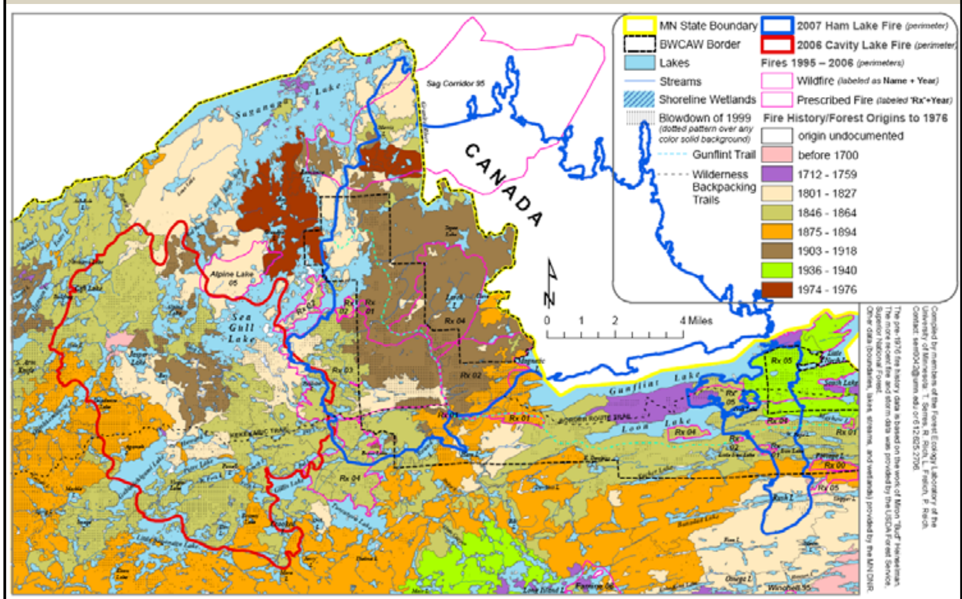
Reclassification
Dissolving
Buffering
Overlay
Cartographic Modeling
(a combination of the above)

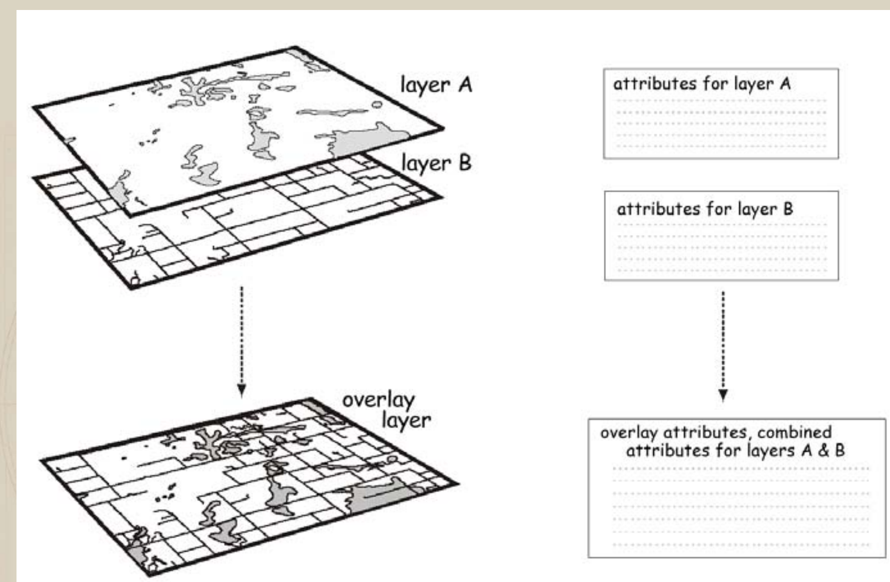
Spatial Analysis: Overlay

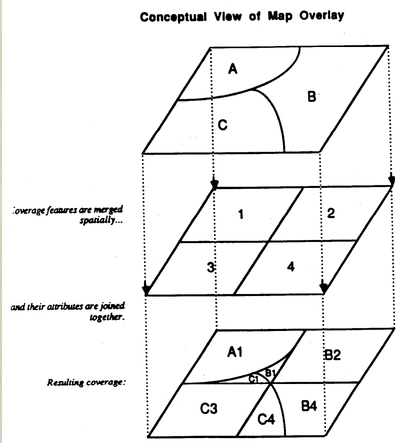
Combination of different data layers
Both spatial and attribute data is combined

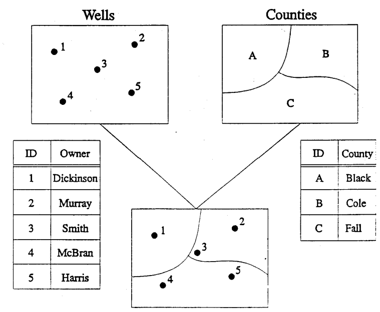
Requires that data layers use a common coordinate system

A new data layer is created


MN State Boundary
BWCAW Border
Lakes
Streams
Shoreline Wetlands
Blowdown of 1999
Gunflint Trail
Wilderness Backpacking Trails
2007 Ham Lake Fire (perimeter)
2006 Cavity Lake Fire (perimeter)
Fires 1995 – 2006 (perimeters)
Wildfire
Prescribed Fire
Fire History/Forest Origins to 1976
origin undocumented
before 1700
1712 - 1759
1801 - 1827
1846 - 1864
1875 - 1894
1903 - 1918
1936 - 1940
1974 - 1976
CANADA
0 1 2 4 Miles


layer A
layer B
overlay layer
attributes for layer A
attributes for layer B
overlay attributes, combined attributes for layers A & B





ID	Owner
1	Dickinson
2	Murray
3	Smith
4	McBran
5	Harris

ID	County
A	Black
B	Cole
C	Fall

ID	County	Owner
1	Black	Dickinson
2	Cole	Murray
3	Cole	Smith
4	Fall	McBran
5	Fall	Harris

Overlay - Point in Polygon

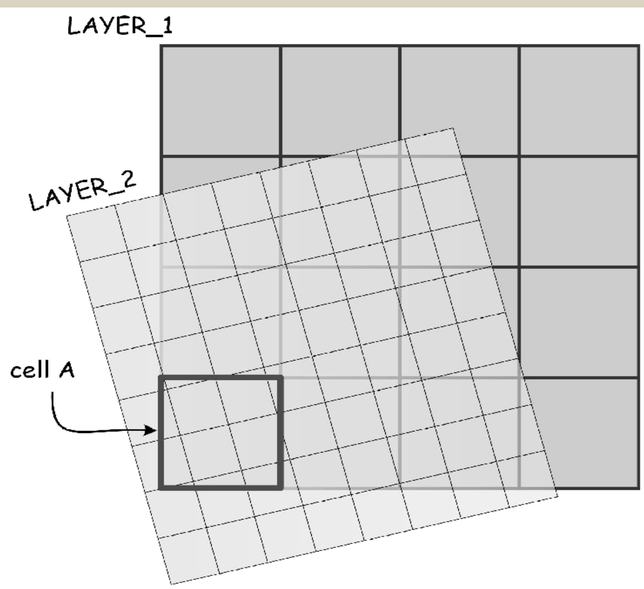


Figure 10-30: Overlain raster layers should be compatible to ensure unambiguous overlay. Cell orientation should be coincident and cell size should be compatible. In the overlay depicted here it is not clear which cells from Layer_2 should be combined with cell A in Layer_1.

Overlay

Raster Overlay

Typically applied to nominal or ordinal data

Cell by cell process which results in the combination of the two input layers

Pay attention to the the number of possible combinations that may be possible and understand the effect on the output layer

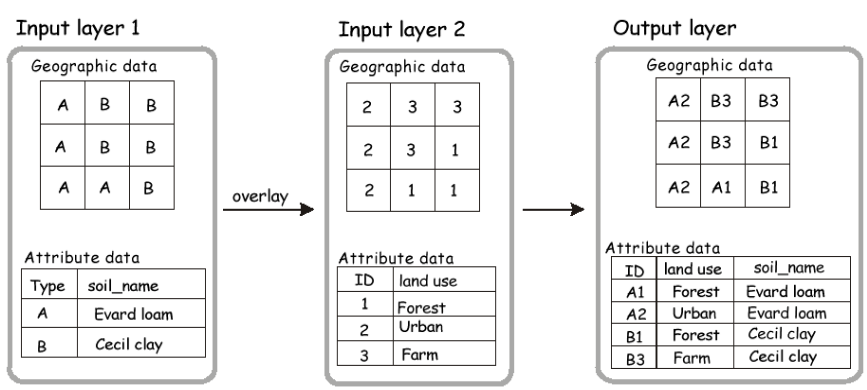
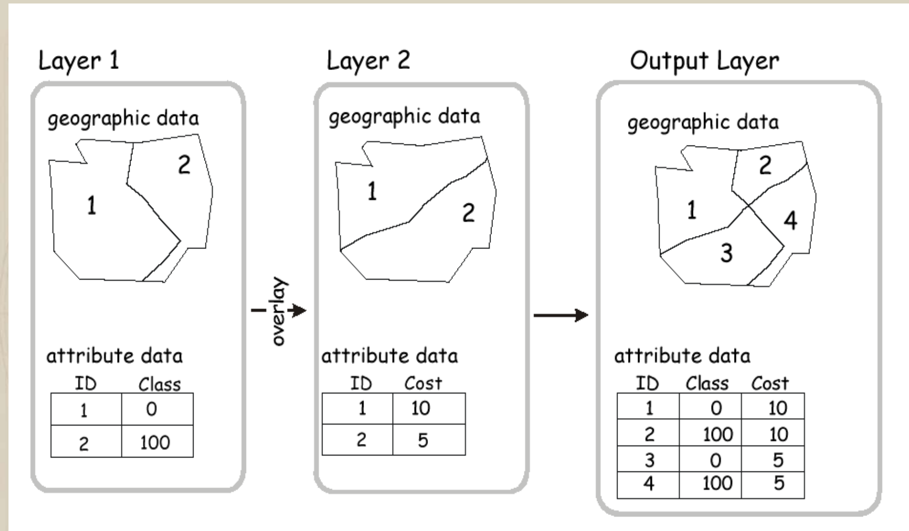


Figure 10-31: Cell-by-cell combination in raster overlay. Two input layers are combined in raster overlay. Nominal variables for corresponding cells are joined, creating a new output layer. In this example a soils layer (left) is combined with a land use layer (center) to create a composite output layer (right).

Feature numbers increase in overlay



Vector Overlay

- Topology is likely to be different
- Vector overlays often identifies line intersection points automatically.
- Intersecting lines are split and a node placed at the intersection point
- Topology must be recreated for later processing

Any type of vector may be overlain with any other type
Output typically takes the lowest dimension of the inputs

For example: Point on Polygon results in a point

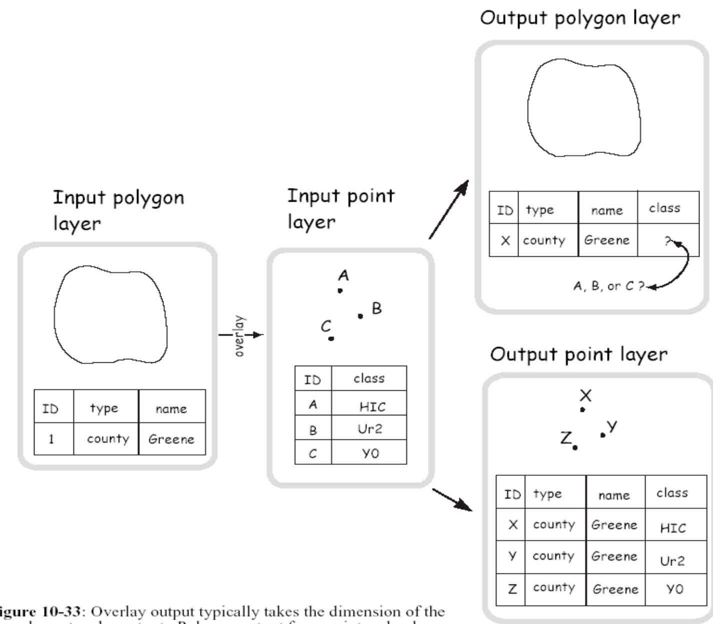
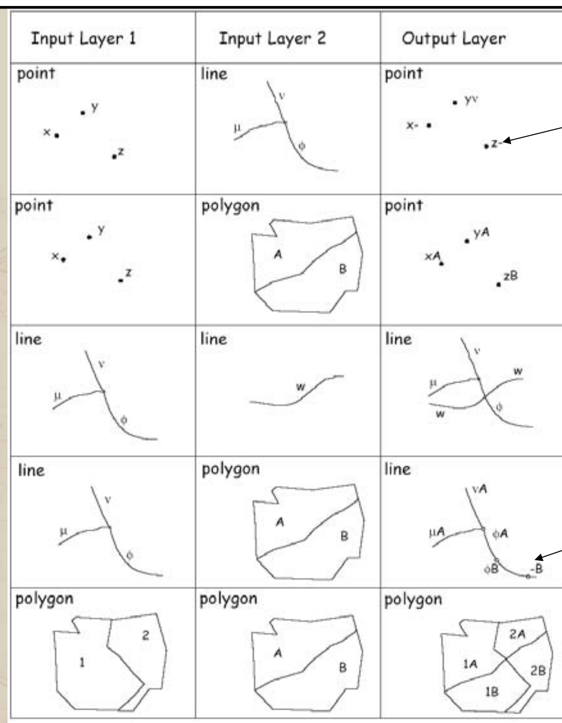


Figure 10-33: Overlay output typically takes the dimension of the lowest-order output. Polygon output from point and polygon overlay (right, top) results in uncertainty regarding the source for combined attribute data because there may be multiple points corresponding to each output polygon. Point output is unambiguous.

Ambiguous result

Un-ambiguous result



note

null or flag values

note

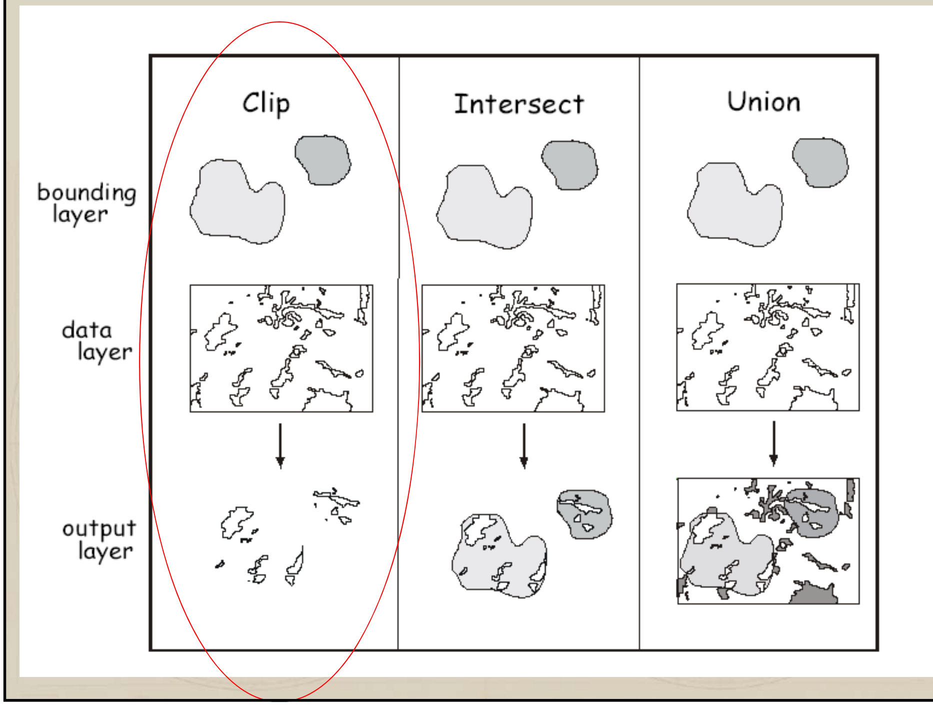
null or flag values

Vector Overlay
(common ways applied)

•CLIP
•INTERSECTION
•UNION

CLIP

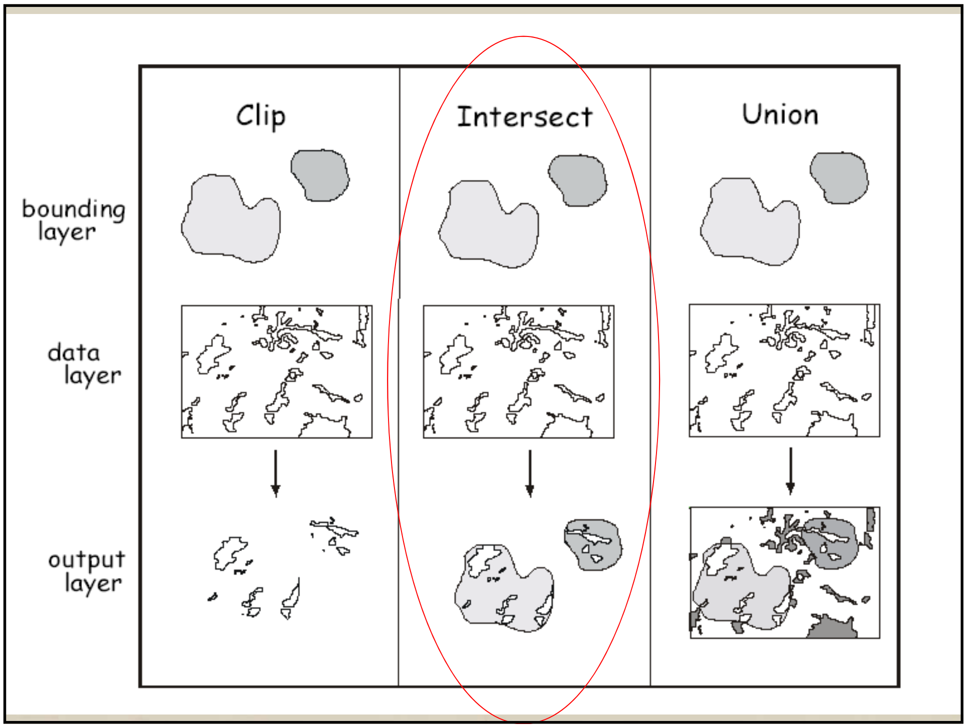
•Cookie cutter approach
•Bounding polygon defines the clipped second layer
•Neither the bounding polygon attributes nor geographic (spatial data) are included in the output layer



INTERSECTION

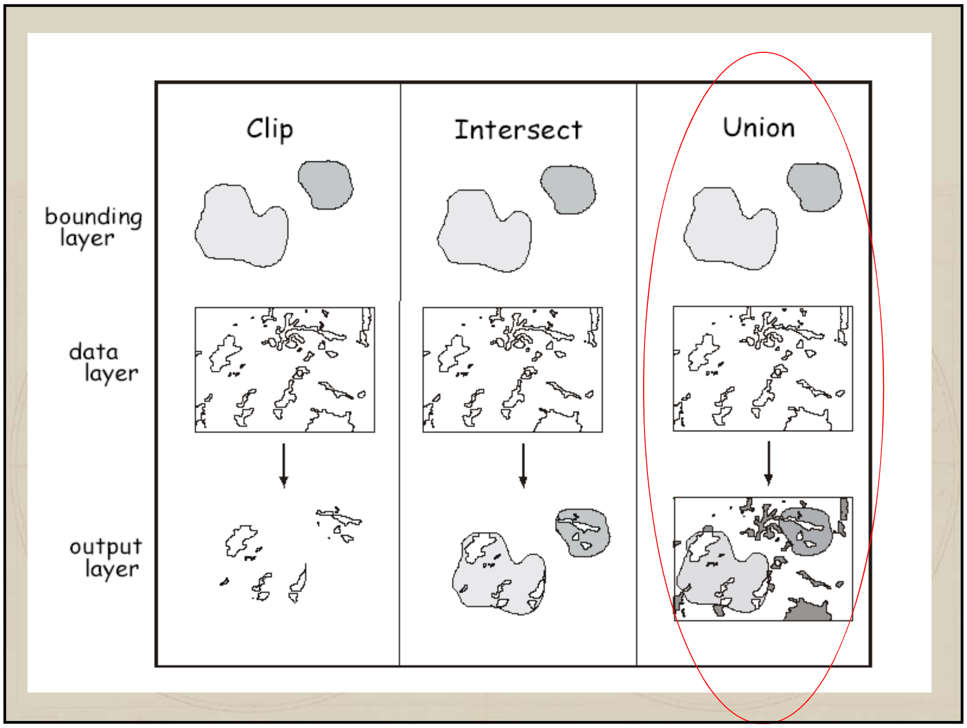
•Combines data from both layers
Features or portions of features which overlap in all layers will be written to the output feature.

•Order of intersection is not important



UNION

- Includes all data from both the bounding and data layers
- New polygons are formed by the combinations of the coordinate data from each layer

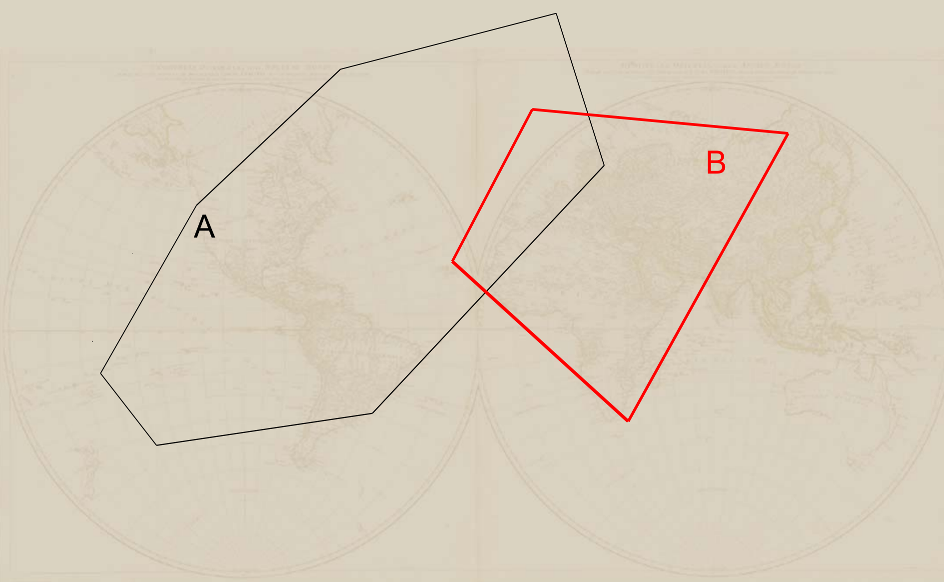


Why do buffering and vector overlay often take so long?

Because a time consuming line intersection test must be performed for all lines in the data layers

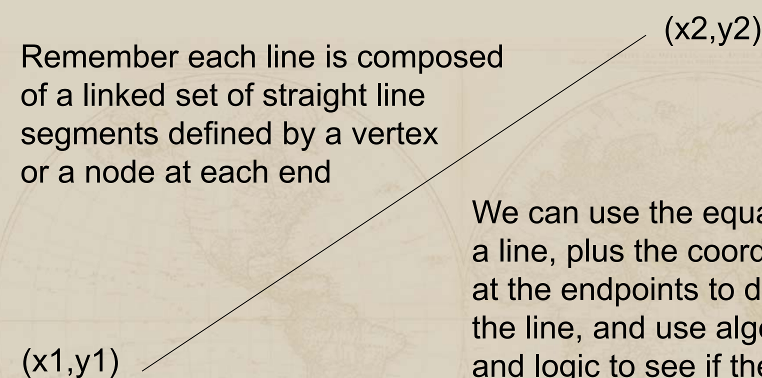
Then, inside vs. outside regions must be identified for all new polygons

Does polygon A intersect/overlap/overlay polygon B?



We must check each line in one data layer against every other line in the second data layer to see if they intersect

Remember each line is composed of a linked set of straight line segments defined by a vertex or a node at each end



We can use the equation for a line, plus the coordinates at the endpoints to define the line, and use algebra and logic to see if the lines intersect

Equation of a line: $y = m * X + b$

Line Intersection Calculations

Line Equation
$y = m_1 x + b_1$

Line Equation
$y = m_2 x + b_2$

1) Calculate Equation Parameters

m_1 = slope = (12-1)/(10-2)
= 1.375

$b_1 = y - m_1 x$
= 12 - 10 * 1.375
= - 1.75

y = 1.375 * x - 1.75

m_2 = slope = (4-2)/(7-9)
= -1

$b_2 = y - m_2 x$
= 4 - (-1) * 7
= 11

y = -1 * x + 11

2) Find Intersection Point

Y =1.375 * x -1.75 y = -1 * x +11

Set y values equal

y =1.375 * x -1.75 = -1 * x +11

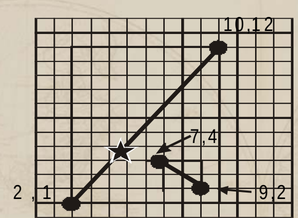
(1.375 + 1) * x = 11 + 1.75
x = 12.75/2.375
= 5.37

y = 1.375 * 5.37 –1.75 = 5.63

Potential Intersection Point at x = 5.37, y = 5.63

(

3) Verify Intersection: Is it Within the Boxes'



Test X:
is 5.37 > 2 and < 10 Yes
is 5.37 > 7 and < 9 No

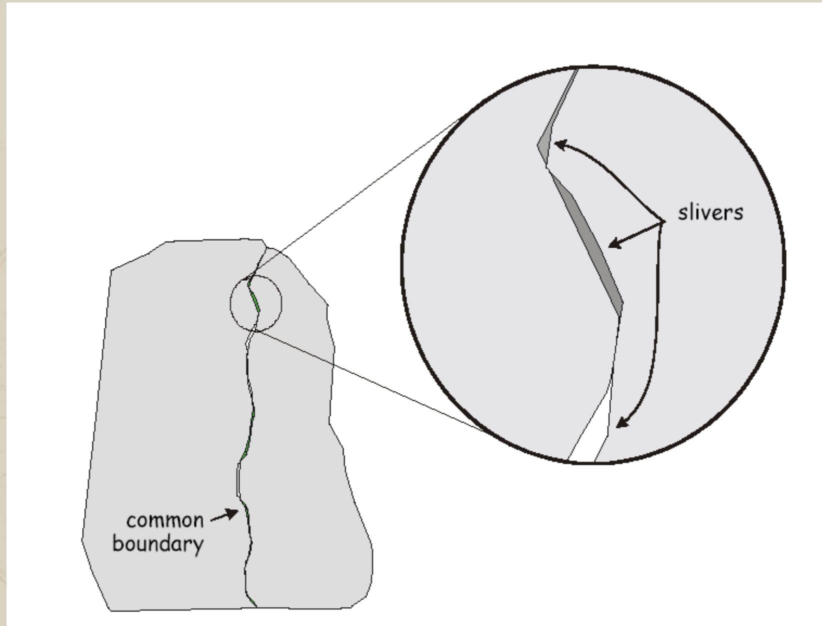
Test Y:
is 5.63 > 1 and <12 Yes
is 5.63 > 2 and < 4 No

Answer: No, the lines do not intersect

Vector Overlay

Common features in Vector overlays create "Slivers" or "Sliver polygons"

A common feature in both layers. The problem is that each definition is very subtly different *(different time, source, materials)* so the polygons don't line up. They can only be seen a very large display scale but can represent over half the output polygons. They take very little space but affect analytical results.



Methods to reduce/remove slivers:

- Redefine the common boundaries with highest coordinate accuracy and replace them in all layers before overlay
- Manually identify and remove
- Use snap distance during overlay