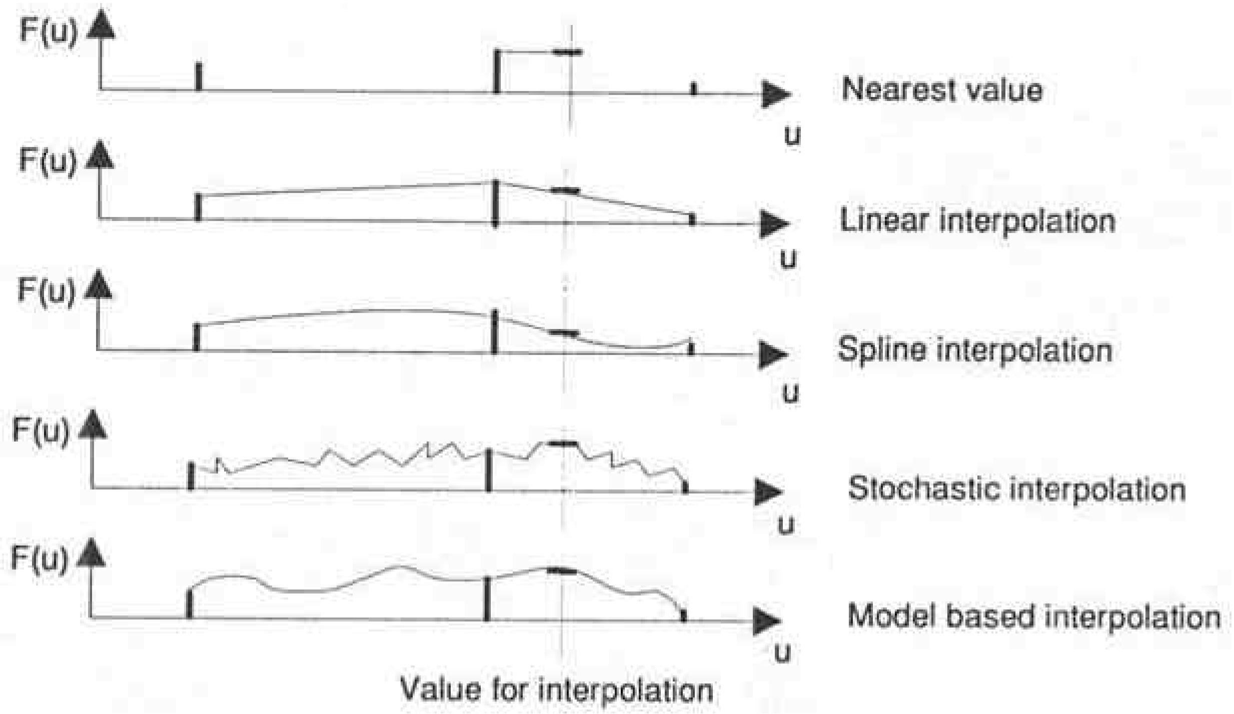
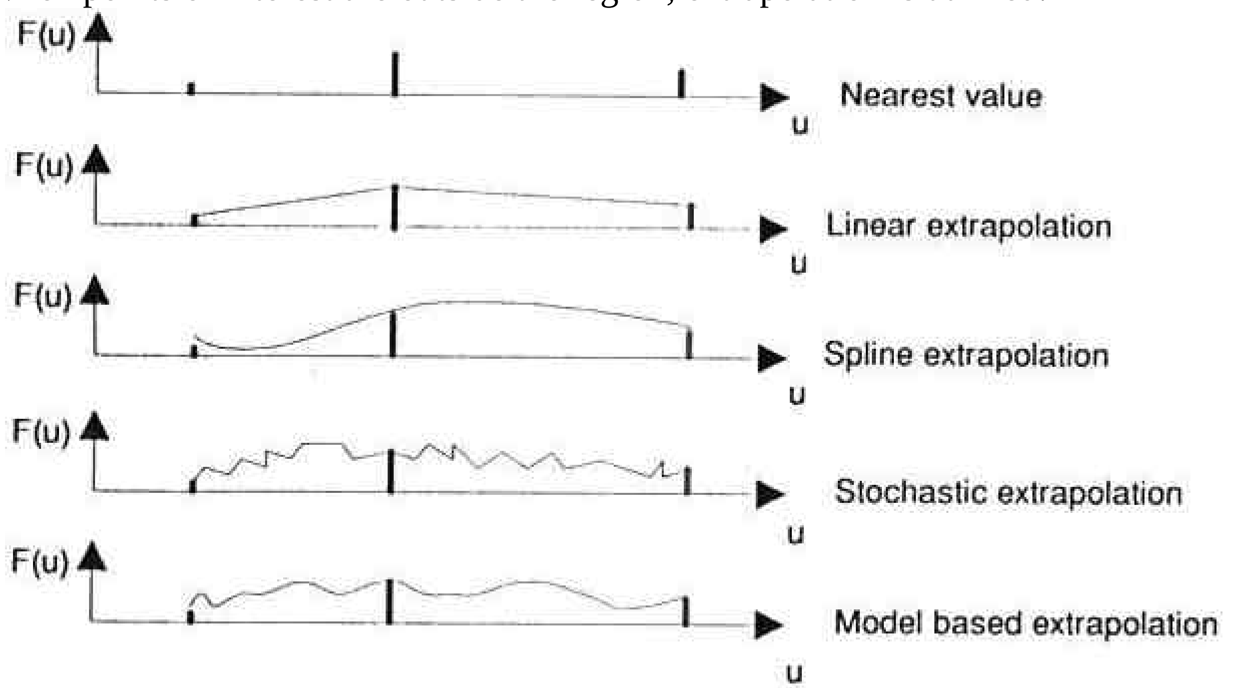
Manipulations

Interpolation (in situations with limited information), especially point-oriented interpolation:

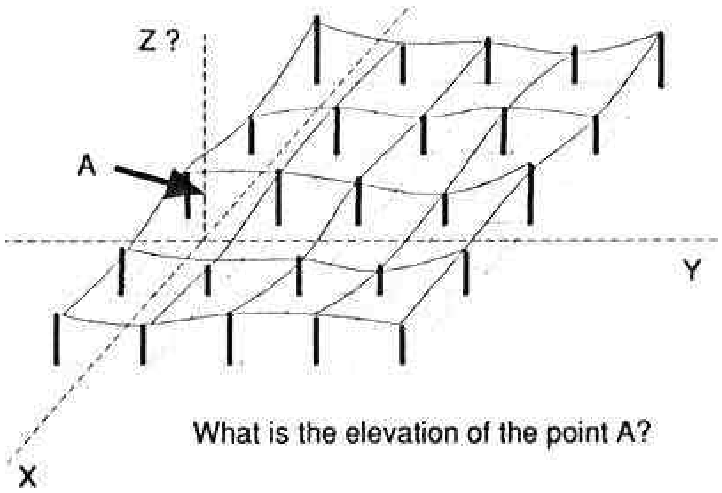


When points of interest are outside the region, extrapolation is utilized:



The model approach is perhaps the most prudent one, and provides more confidence in the results. When extrapolating, as for interpolating a certain level of precision must be given, although in particular context it may be difficult to establish how much error can be tolerated.

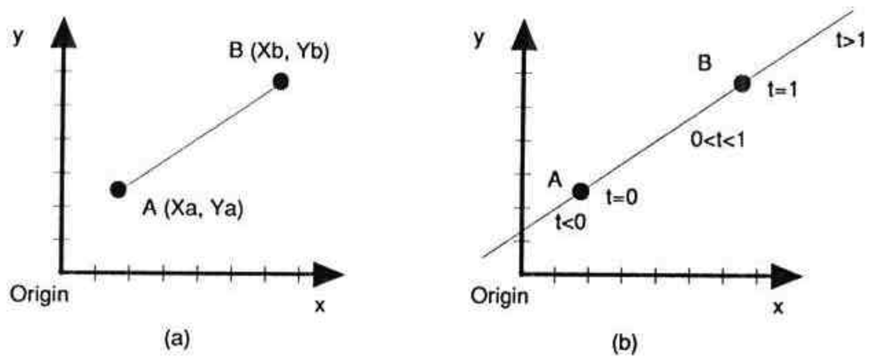
When considering more dimensions, there are different possibilities, even mixture of them: we have to interpolate in one dimension and extrapolate in the other. We have so-called **geometric inference:**



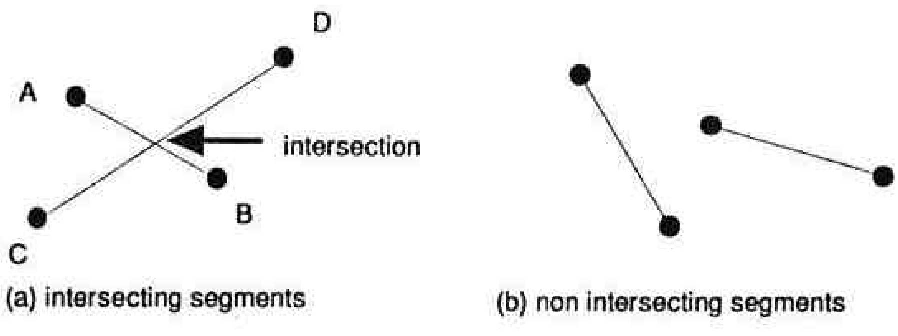
Basic Operations on Lines and Points

Many contexts are making necessary to find the intersections of line entities. Usually we have linear functions, in Cartesian coordinates. Formulas differ according also to the representation of segments:

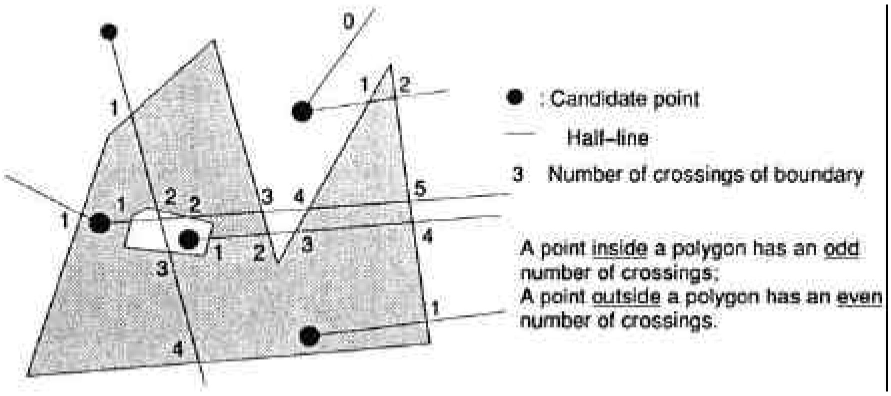
- by end point coordinates,
- by end point coordinate and parameter.



Segment intersections usually refer to the use of so-called parameter expression, calculating intersections or non-intersections by linear formulas:



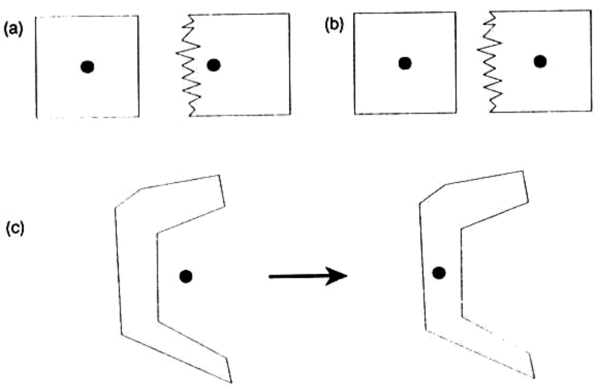
Point-in-polygon procedure:



Centroid definitions

If we need to represent polygon by one point, we refer to centroid. There are various definitions of centroid:

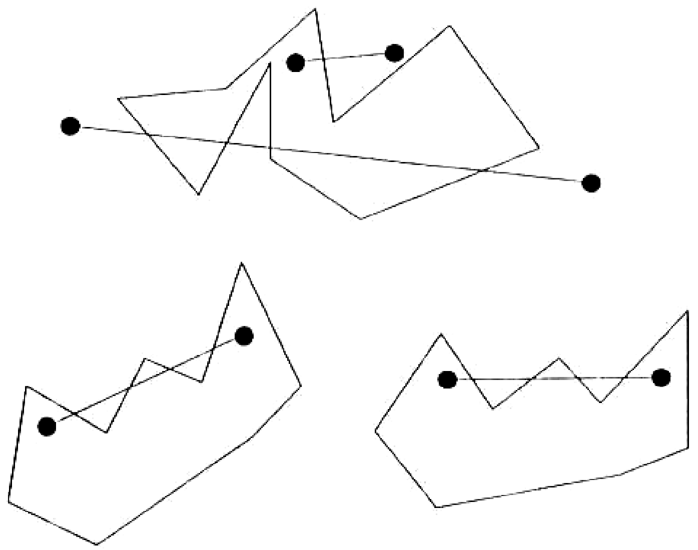
- defined from vertices,
- obtained as a statistical bivariate median or the centre of gravity,
- computed as the centre of an enclosing or enclosed rectangle or of an enclosing or inscribing circle,
- obtained as the peak value of a surface fitted within the polygon,
- choosen intuitively:



Some Operations for Polygons

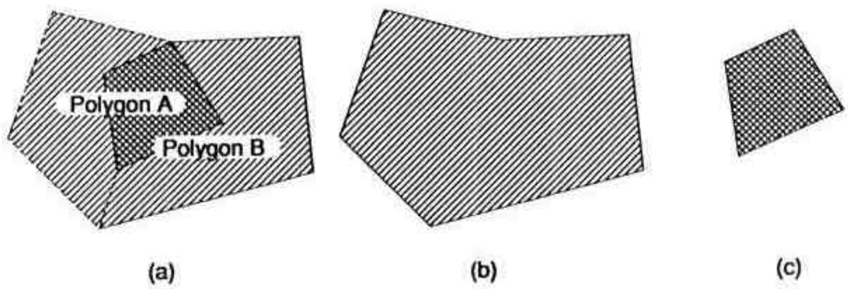
Intersection of lines with polygons

A common situation comprises the intersection of a segment with a polygon, or in the opposite way, the computation of what part of the segment lies inside a polygon. In order to determine the segment intersection, we can compute the intersection of the straight line for the segment and the line segments for the polygon boundaries. If there is no intersection with the boundaries, there is no intersection of the segment ind the polygon. If there are intersections of the straight line and the polygon, we can determine what parts are inside by a repetitive use of the **Jordan theorem**. For checking whether a segment is totally included in a polygon, it is not sufficient to test end-points because the polygons can have strange concavities intersecting this segment (a). Obviously we cannot examine the infinity of segment points to see whether they are all inside or outside the polygon. In order to facilitate the testing and to re-use the result of the Jordan theorem, a nice step is to rotate the polygon so that the segment under study is parallel to an axis (b). After this, it is easy to count the number of intersections only by comparing coordinates. If the number is different from zero, we know that a part of the polygon is inside and a part is outside. By re-using the half-line procedure on the end-points when the number of intersections is zero, we then know whether the end-points are inside or outside, and, consequently, whether the totality of the line segment is inside or outside:

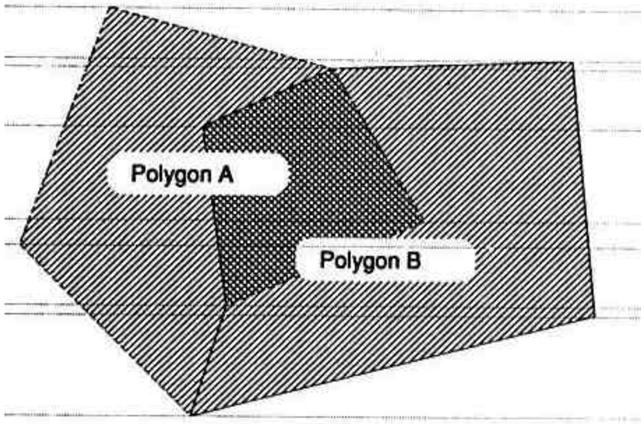


Union and intersection of polygons

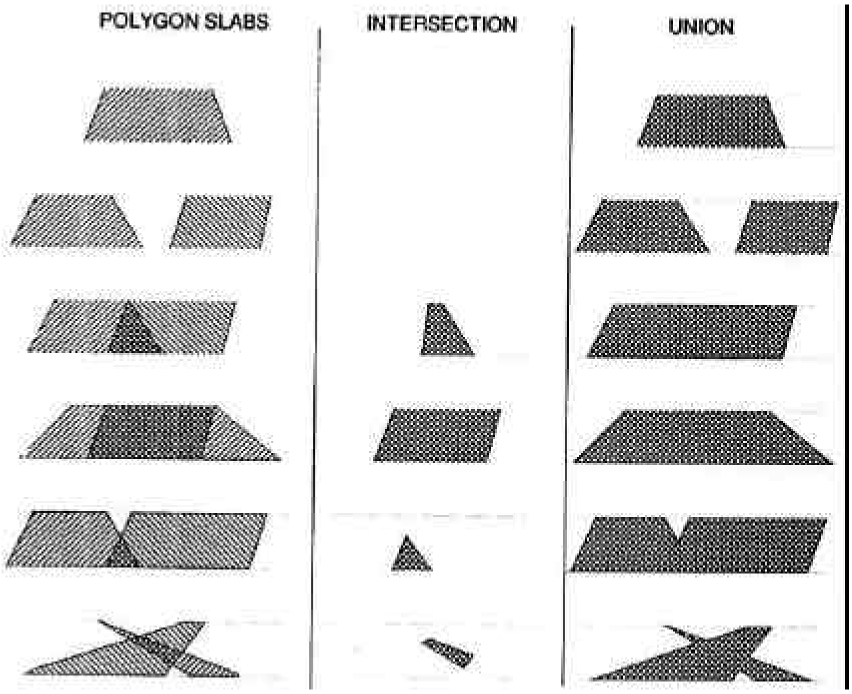
One of the major needs and challenging problems in spatial information systems is to compute the difference, union and the intersection of polygons.



We present only possibly the simplest way, based on the **slab technique.** Each polygon is divided into parallel slabs, usually horizontal for the convenience of parallelism with the coordinate axis, created by drawing lines through the polygon vertices. This procedure creates trapezoids which are easy to compare. When the edges are not intersecting inside the slabs, the comparison is straightforward; otherwise, as in the last row in the diagram, some other slabs can be created passing through these intersection points:



The union and intersection to be calculated



The method of slabs for finding the union and intersection of two polygons.

Area computation

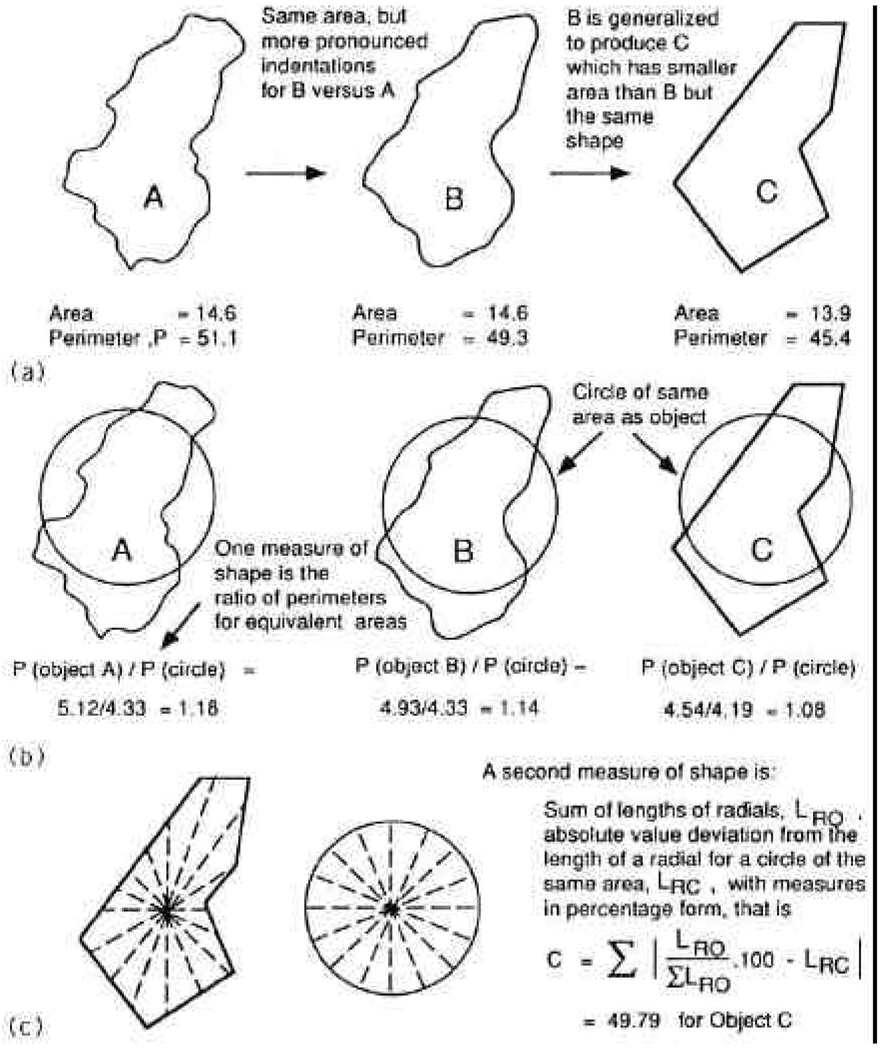
Area computation sometimes uses the trapezoid idea although a procedure using geometric cross-products is usually preferred. Lines perpendicular to the *x*-axis (alternatively, the *y*-axis could be used) are dropped to it from the vertices of polygons. This process creates a set of overlapping figures, with four line segments: the base on the *x*-axis, two vertical lines and an edge of a polygon. After the area of each trapezoid is computed, they are summed, including subtractions for the lower pieces. Enclaves and exclaves can be handled by this procedure by using encoded data showing the gaps or separate pieces of the polygons.

However, the classical way in computational geometry is to use the **geometric cross-product** to compute areas, and mixed products for volumes. For computing the area of any polygon with or without holes, if we have in total *N* vertices ordered in a counterclockwise sequence from *1* to *N*, the area is given by:

$$Area = \frac{1}{2}\left(\sum_{i=1}^{N-2} (x_i y_{i+1} - x_{i+1} y_i) + (x_N y_1 - x_1 y_N) \right)$$

This process, established via vector algebra using the cross-products, is more rapid than the trapezoid decomposition. The dot product device is also useful for obtaining angles between vectors, and for concatenating boundary lines for polygons using a centroid as origin for the vectors.

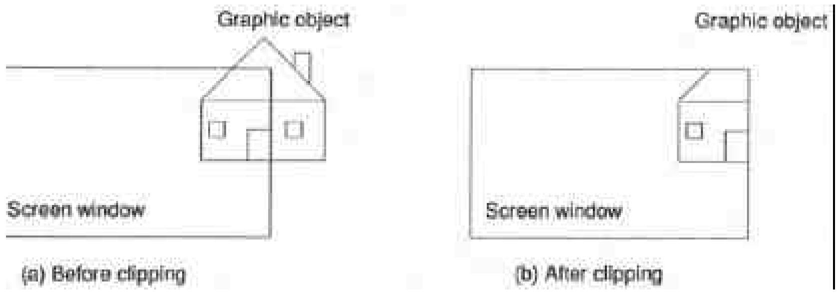
Shape Measures for Polygons



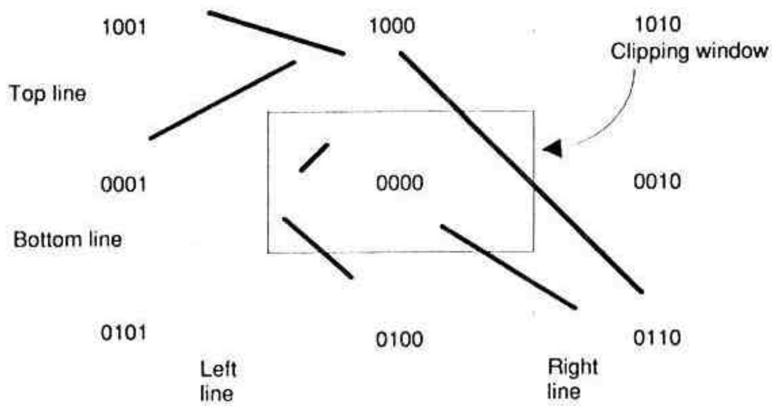
Polygon clipping

A clipping situation occurs commonly when displaying or retrieving spatial entities within a defined area like a rectangle. When one has a rectangular window on a screen or on another graphic device, it is necessary to know what part of the object (say, a house) has to be displayed on the screen. Similarly, when using enclosing rectangles for overlay purposes, particular polygon or line objects may be cut. Three situations arise, illustrated for the display context:

- object is completely inside the window,
- object is totally outside the window,
- the object is overlapped by the window.



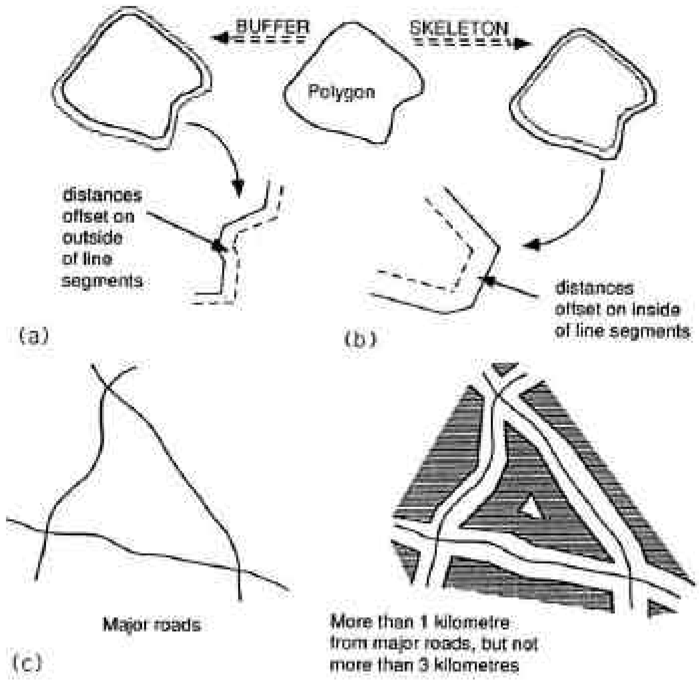
Clipping of the object by a rectangular window.



Clipping a segment (an illustration of the algorithm).

Buffer zones

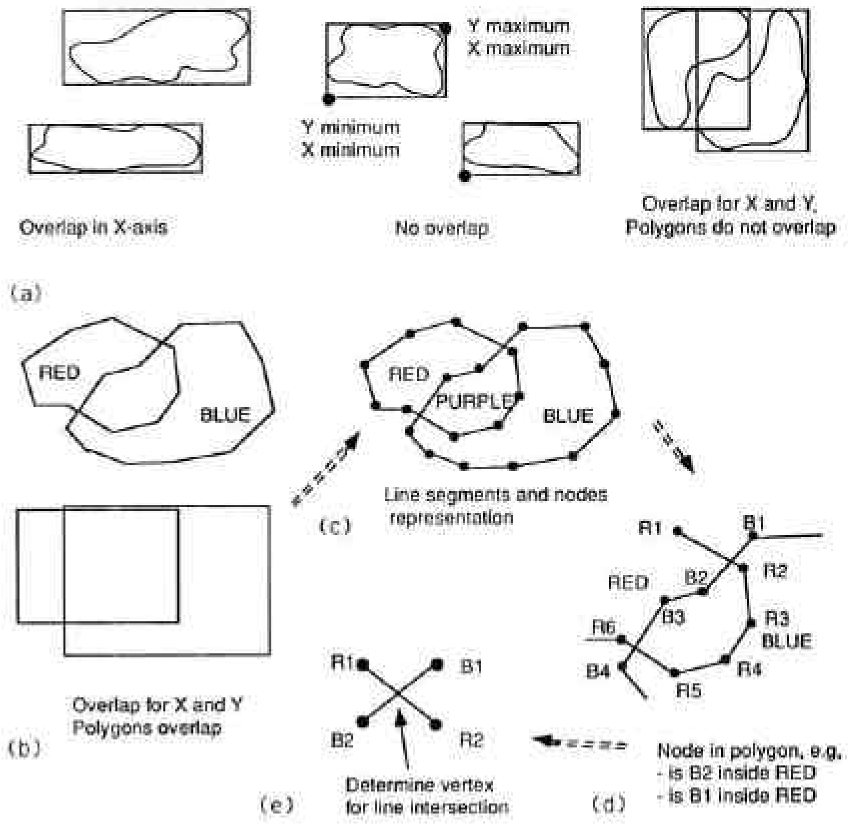
One other class of spatial operations includes the creation of boundaries, inside or outside the existing polygon, offset by a certain distance, and parallel to the boundary: skeleton zones and buffer zones.



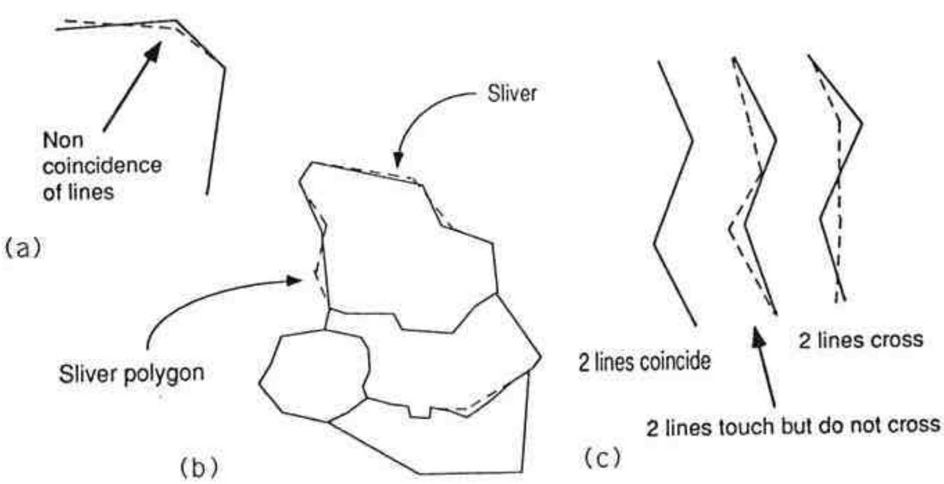
Polygon Overlay Process

The general process of overlay to create new polygons consist of:

identifying line segments, preferably having topology,
establishing minimun enclosing rectangles for the polygons,
ascertaining if line segment(s) of one polygon are inside a polygon of the overlay map by a point-in-polygon process,
finding intersections of segments representing boundaries,
creating records for new line segments and their associated topology,
assembling the new polygons from the appropriate line segments,



Line disrepancies and sliver polygons



Spatial Data Transformations

Transformations among spatial unit types - changes in dimensionality:

Conceptual model		Representation				
	FROM: TO:	CELL	POINT	LINE	POLYGON	VOLUME
	POINT	D	Ideal	S	S	S
	LINE	D	R	Ideal	S	S
	POLYGON	D	R	R	Ideal	S
	VOLUME	D	R	R	R	Ideal

Ideal = optimum match
Cell = regular figure, e.g. square or triangle or cube
D = possible distortion and imprecision
S = possible substitution and symbolisation
R = information reduction

(a)

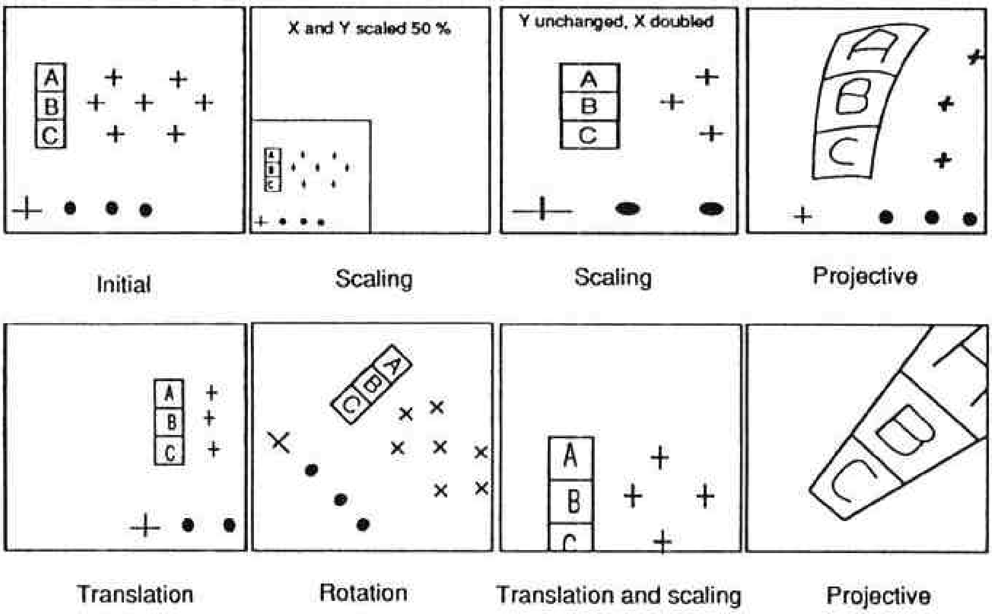
From Encoding	To Encoding CELL	POINT	LINE	POLYGON
CELL	Scale change	Make lattice	Boundary estimation	Areal interpolation
POINT	Encoding	Interpolation	Triangulation	Thiessen polygon
LINE	Encoding	Collapse (e.g. contours)	Generalization	Link chain
POLYGON	Encoding	Centroid	Profile	Distort shape

(b)

Changes in position - types of geometric transformations:

GEOMETRY	PROPERTIES	TRANSFORMATION ALLOWED	TRANSFORMATION NOT ALLOWED
Equi-area	Area of an object is preserved (congruence)	For example, rotation	[Similar, but smaller area]
Similarity	Shape but not necessarily area is preserved	For example, translation and scaling	[Same area, but different shape]
Affine	Angular distortion; parallelism of lines is preserved	For example, tilting and scaling	[lines not parallel]
Projective	Angular and length distortion	For example. rubber-sheeting	[Does not have 4 corners]
Topology	Neighbourhood is preserved but shape is irrelevant	For example, rubber-sheeting	[Has a hole in the disk]

Rubber sheeting - topological, projective, affine transformations:



Conversion between raster and vector representations: