Common Spatial Data Models

All spatial data models are based on a conceptualization. As an example, consider a regional map that defines roads as lines. We conceive of each road as a linear feature that fits into a small number of categories. These lines connect cities and towns that are shown as discrete points or polygons on the map. Road properties may include only the road type, e.g., highway or local road. The roads have a width represented by a line symbol on the map; however, the scaled road width may not represent the true road width. All state highways are represented equally although they may vary. Some may have wide shoulders, others not, or dividing barriers of concrete, versus a broad vegetated median, but we may choose to omit this variation, fitting all highways into one class.

There are two main conceptualizations used for digital spatial data. The first defines discrete objects using a *vector data model*. This model uses discrete elements such as points, lines, and polygons to represent the geometry of real-world entities (Figure 2-19, left).

Farm fields, roads, wetlands, cities, and census tracts are examples of entities that are often represented by discrete vector objects. Points are often used to define the locations of "small" objects such as wells, buildings, or ponds. Lines may be used to represent linear objects, for example, rivers or roads, or to enclose polygons, which identify area objects. Starting points and ending points for a line are sometimes referred to as *nodes*, while intermediate points in a line are referred to as *vertices*.

Vector objects are discrete. A forest may share an edge with a pasture, and this boundary is represented by lines. In truth, a forest edge may grade into a mix of trees and shrubs, then shrubs and grass, then pure grass; however, in the vector conceptualization, a line between two land cover types will be drawn to indicate a discrete, abrupt transition. Lines and points have coordinate locations, but points have no dimension, and lines have no dimension perpendicular to their direction. Area features are defined by a closed, connected set of lines.

The second common conceptualization identifies and represents grid cells for a given region of interest. This conceptualization employs a *raster* data model (Figure 2-19, right). Raster cells are arrayed in a row and column pattern to provide "wall-to-wall" coverage of a study region. Cell values are used to represent the type or quality of

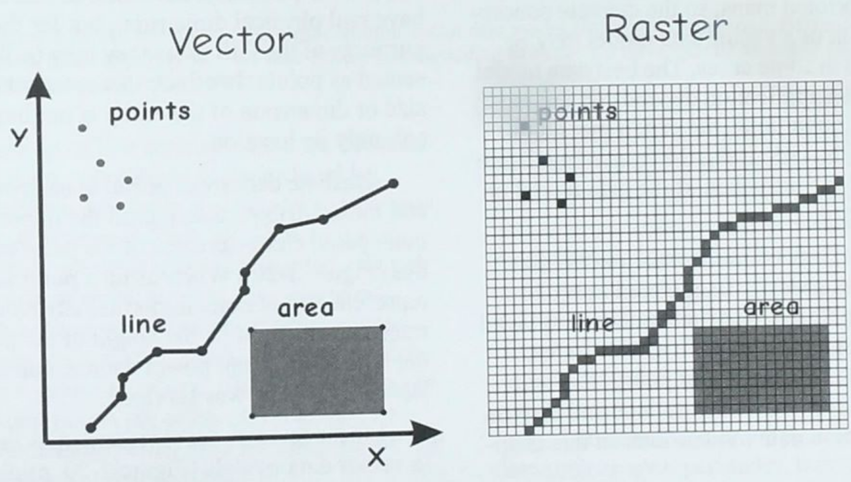


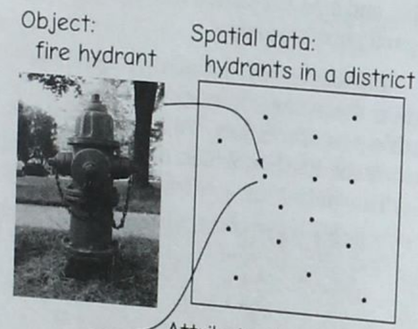
Figure 2-19: Vector and raster data models.

slightly, within a circle about 9 meters (30 feet) across, so the northern pole location is always within this circle. The nutation has a period of 433 days, with the pole returning back to its original location over that time.

Attribute Data and Types

Attribute data are used to record the non-spatial characteristics of an entity. Attributes, also called *items* or *variables*, may be envisioned as a list of characteristics that describe features. Color, depth, weight, owner, vegetation type, or land use are examples of variables that may appear as attributes. Attributes record values; for example, a fire hydrant may be colored red, yellow, or orange, have 1 to 4 flanges, and a pressure rating of any real number from 0 to 12,000.

Attributes are often presented in tables and arranged in rows and columns (Figure 2-18). Each row corresponds to a spatial object, and each column corresponds to an attribute. Tables are often organized and managed using a specialized computer program called a database management system (DBMS, described more fully in Chapter 8).



Attribute data:
properties of hydrants

ID	Flanges	Color	Rating
1	3	red	high
2	3	red	high
3	2	green	low
4	3	red	mid
.	.	.	.

Figure 2-18: Attributes are typically envisioned in a table, with objects arranged in rows and attributes aligned in columns.

All attributes can be categorized as nominal, ordinal, or interval/ratio attributes. *Nominal attributes* are variables that provide descriptive information about an object. The color is recorded for each hydrant in Figure 2-18. Other examples of nominal data are vegetation type, a city name, the owner of a parcel, or soil series. There is no implied order, size, or quantitative information contained in nominal attributes.

Nominal attributes may also be images, film clips, audio recordings, or other descriptive information, for example, GIS for real estate often have images of the buildings as part of the database. Image, video, or sound recordings stored as attributes are sometimes referred to as "BLOBs" for *binary large objects*.

Ordinal attributes imply a ranking by their values. An ordinal attribute may be descriptive, such as high, mid, or low, or it may be numeric; for example, an erosion class with values from 1 to 10. The order reflects only rank, and not scale. An ordinal value of four has a higher rank than two, but we can't infer that the attribute value is twice as large.

Interval/ratio attributes are used for numeric items where both rank order and absolute difference in magnitudes are represented, for example, the number of flanges in the second column of Figure 2-18. These data are often recorded as real numbers on a linear scale. Area, length, weight, height, or depth are a few examples of attributes that are represented by interval/ratio variables.

Items have a *domain*, a range of values they may take. Colors might be restricted to red, yellow, and green; cardinal direction to north, south, east, or west; and size to all positive real numbers.

Common Spatial Data Models

All spatial data models are based on a conceptualization. As an example, consider a regional map that defines roads as lines. We conceive of each road as a linear feature that fits into a small number of categories. These lines connect cities and towns that are shown as discrete points or polygons on the map. Road properties may include only the road type, e.g., highway or local road. The roads have a width represented by a line symbol on the map; however, the scaled road width may not represent the true road width. All state highways are represented equally although they may vary. Some may have wide shoulders, others not, or dividing barriers of concrete, versus a broad vegetated median, but we may choose to omit this variation, fitting all highways into one class.

There are two main conceptualizations used for digital spatial data. The first defines discrete objects using a *vector data model.* This model uses discrete elements such as points, lines, and polygons to represent the geometry of real-world entities (Figure 2-19, left).

Farm fields, roads, wetlands, cities, and census tracts are examples of entities that are often represented by discrete vector objects. Points are often used to define the locations of "small" objects such as wells, buildings, or ponds. Lines may be used to represent linear objects, for example, rivers or roads, or to enclose polygons, which identify area objects. Starting points and ending points for a line are sometimes referred to as *nodes*, while intermediate points in a line are referred to as *vertices*.

Vector objects are discrete. A forest may share an edge with a pasture, and this boundary is represented by lines. In truth, a forest edge may grade into a mix of trees and shrubs, then shrubs and grass, then pure grass; however, in the vector conceptualization, a line between two land cover types will be drawn to indicate a discrete, abrupt transition. Lines and points have coordinate locations, but points have no dimension, and lines have no dimension perpendicular to their direction. Area features are defined by a closed, connected set of lines.

The second common conceptualization identifies and represents grid cells for a given region of interest. This conceptualization employs a *raster* data model (Figure 2-19, right). Raster cells are arrayed in a row and column pattern to provide "wall-to-wall" coverage of a study region. Cell values are used to represent the type or quality of

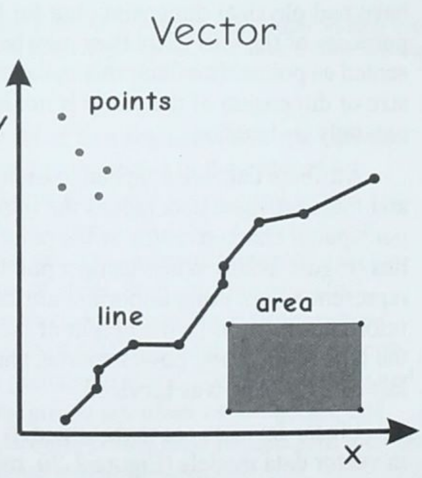


Figure 2-19: Vector and raster data models.

mapped variables. Raster models are often used with variables that may change continuously across a region. Elevation, mean temperature, slope, average rainfall, cumulative ozone exposure, or soil moisture are examples of phenomena that are often represented as continuous fields. Raster representations are also sometimes used to represent discrete features, for example, class maps of vegetation or political units.

Data models are often interchangeable in that many phenomena may be represented by many data models. For example, elevation may be represented as a raster surface (continuous field) or as a series of lines representing contours of equal elevation (discrete objects). Data may be converted from one model to another; for example, the location of contour lines may be determined by evaluating the raster surface, or a raster data layer may be derived from a set of contour lines. These conversions entail some costs both computationally and perhaps in data accuracy.

The decision to use either a raster or vector model often depends on our conceptualization of the objects and the most frequent operations performed. We think of elevation as a continuous variable and slope is more easily determined when elevation is represented in a raster data set. However, discrete contours are often the preferred format for printed maps, so the discrete conceptualization of a vector data model may be preferred in some cases. The best data model for a given application depends on the most common operations, the experiences and views of the GIS users, the form of available data, and the influence of the data model on data quality.

Other, less common data models are sometimes used. A triangulated irregular network (TIN) is one such model, employed to represent surfaces such as elevations, through a combination of point, line, and area features. We will introduce and discuss less common data models later in this chapter.

Vector Data Models

A vector data model uses sets of coordinates and associated attribute data to define discrete objects. Groups of coordinates define the location and boundaries of discrete objects, and these coordinate data plus their associated attributes are used to create vector objects representing the real-world entities (Figure 2-20). In the most common vector models, there is an attribute table associated with each vector layer, and a single row in the table corresponding to each feature in the data layer. These vector layers are said to contain *single-part features*, because there is a single geographic object for each row in the table, with one to several columns in each row. All values in a column have the same type, so for any given column, all entries might be ordinal, or interval ratio, or a BLOB, or some other defined type. An identifier value, or ID, is typically included, and this value is often unique within the table, with an unrepeated value assigned for each row and corresponding feature.

There are three basic types of vector objects: points, lines, and polygons (Figure 2-20, top). A point uses a single coordinate pair to represent the location of an entity that is considered to have no dimension. Gas wells, light poles, accident location, and survey points are examples of entities often represented as point objects. Some of these have real physical dimension, but for the purposes of the GIS users they may be represented as points. In effect, this means the size or dimension of the entity is not important, only its location.

Attribute data are attached to each point, and these attribute data record the important non-spatial characteristics of the point entities (Figure 2-20). When using a point to represent a light pole, important attribute information might be the height of the pole, the type of light and power source, and the last date the pole was serviced.

Linear features are represented as lines in vector data models (Figure 2-20, mid). Lines are most often represented as an ordered set of coordinate pairs. Each line is

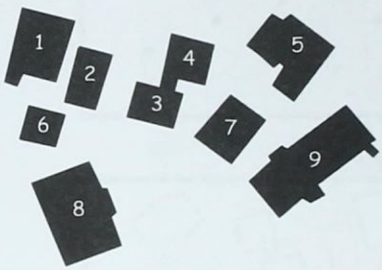
Points

ID	Tower Name	Height	Format
1	WKRP	101.0	Pop
2	WYOU	55.5	Oldies
3	TPT	486.0	Public TV
4	WQXR	99.5	Classical
5	BBC	212.1	News

Lines

ID	Name
1	Tuckaseegee River
2	Pigeon Branch
3	Poplar Run
4	Shope Fork
5	Mel's Brook
6	Merdesansrame Creek
7	Longue Arm
8	Arroyo Grande

Polygons



ID	Building Name	Floors	Roof Type
1	Hodson Hall	6.0	flat, sealed tar
2	Borlaug Hall	5.5	pitched 9/12, tile
3	Guilford Technology Bldg.	4.0	flat, gasket
4	Shop Annex	2.5	flat, sealed tar
5	Animal Sciences Bldg.	1.0	pitched 12/12, tile
6	Administration Bldg.	14.0	pitched 6/12, metal
7	Climate Sciences Center	6.0	flat, sealed tar
8	Grantham Tower	1.0	pitched 9/12, tile
9	Biological Sciences Bldg.	9.0	pitched 12/12, tile

Figure 2-20: An example of the most common vector data model structures. Geographic features consist of points, lines, or polygons, with each feature corresponding to a row in a table with an identifier (ID) and a set of attributes arrayed in columns.

made up of line segments that run between adjacent coordinates in the ordered set. Attributes in a table correspond to line segments (Figure 2-20, mid). Curved linear entities are most often represented as a collection of short, straight, line segments, although curved lines are at times represented by a mathematical equation describing a geometric shape. The line starting and ending points are often called nodes, and intermediate points used to represent the line shape are called vertices.

Area entities are most often represented by closed polygons (Figure 2-20, bottom). These polygons are formed by a set of connected lines, either one line with an ending point that connects back to the starting point, or as a set of lines connected start-to-end. Polygons have an interior region and may entirely enclose other polygons in this region. Polygons may be adjacent to other polygons and thus share "bordering" or "edge" lines with other polygons. Attribute data such as area, perimeter, land cover type, or county name may be linked to each polygon (Figure 2-20, bottom).

Note that there is no uniformly superior way to represent features, and we may represent the same features as points, lines, or polygons (Figure 2-21). Some feature types may appear to be more "naturally" represented one way: manhole covers as points, roads as lines, and parks as polygons. However, in a very detailed data set, the manhole covers may be represented as circles, and both edges of the roads may be drawn and the roads represented as polygons. The best representation depends on the detail, accuracy, and intended use of the data set.

Vector layers sometimes have a many-to-one relationship between geographic features and table rows (Figure 2-22), defining *multi-part features*. In these instances, many spatially distinct features are matched with a row, and the row attributes apply to all the distinct features. This is common when representing islands, groups of buildings, or other clusters of features that make up a perceived whole thing. These multi-part features may have multiple geographic objects that correspond to one row.

Multi-part features may also be used for large data sets, for example, when millions of point observations are collected automatically with laser scanners. Tables are often slower to process than point geographies, and so reducing the table size by grouping points into multi-part features may shorten many operations.

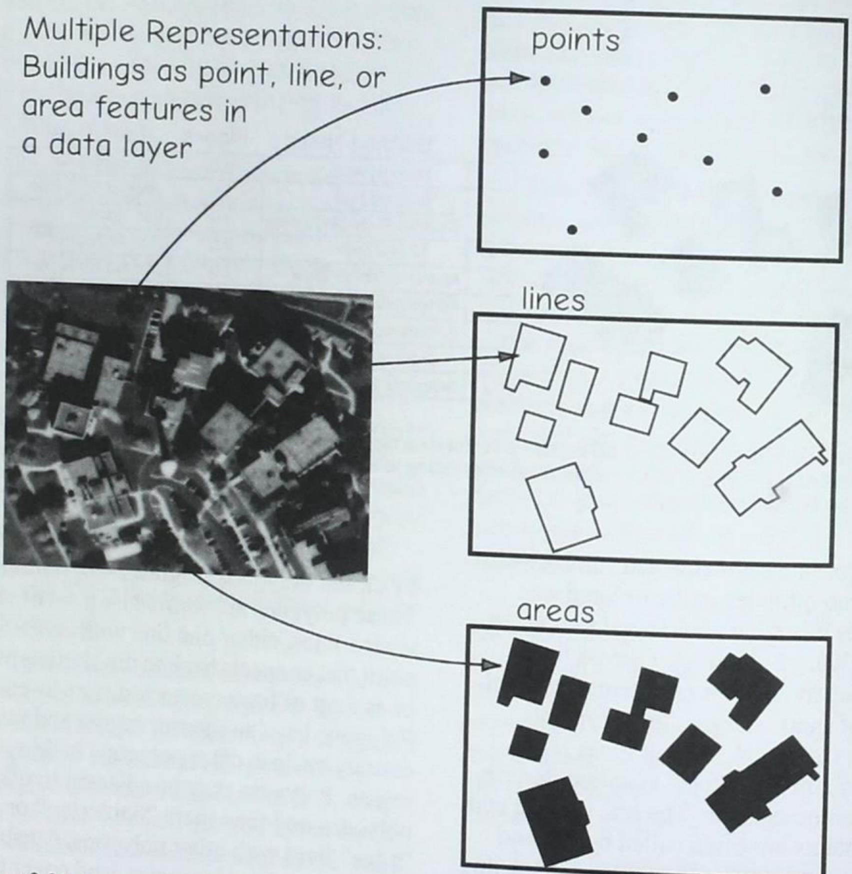
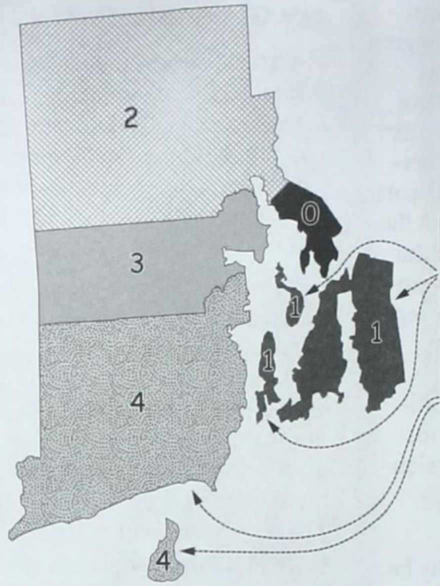


Figure 2-21: The same objects may be represented by points, lines, or polygons, depending on our view of and intended use for the data.



Multi-part Features

ID	County	Area
0	Bristol	73.5
1	Newport	286.5
2	Providence	1,121.9
3	Kent	455.6
4	Washington	912.3

Single-part Features

ID	County	Area
0	Bristol	73.5
1	Newport	243.5
2	Newport	14.8
3	Newport	24.8
4	Providence	1,121.9
5	Kent	912.3
6	Washington	884.7
7	Washington	27.6

Figure 2-22: Example of multi-part and single-part features. Here, counties for Rhode Island, a state in the Eastern U.S.A., are shown with one table entry for each county (top), with multi-part features in the layer, and with one table entry for each distinct polygon (bottom), with only single-part features in the layer. Note that calculations, analysis, and interpretation may differ for multi-part v.s. single-part features.

Care is warranted when converting multi-part features to single-part features. The most common problems arise for aggregate variables in polygon layers, such as total counts. For example, population data are often delivered by census areas such as states. Many states, e.g., Hawaii, have several parts and are represented by a multi-part shape. The population is associated with the aggregated set of polygons comprising the state (Figure 2-23). When converted to single-part shapes, the attributes are often copied for each component polygon. In our example, all single-part polygons will be assigned the attribute values for the multi-part feature, in effect repeating counts for each part. Subsequent aggregation or calculation across the population column may result in error.

Attributes for converted shapes may be corrected. If component data are available, they can be assigned to each of the single-part features. If not, then some weighting scheme may be available, for example, if there is a correlation between area and count. Until they are reviewed and appropriately adjusted, single-part attributes derived from multi-part features should be used with caution.

Polygon Inclusions and Boundary Generalization

Vector data frequently exhibit two characteristics: polygon inclusions and boundary generalization. These characteristics are oft-ignored, but may affect the use of vector data. These concepts must be understood, their presence evaluated, and effects weighed in the use of vector data sets.

Polygon inclusions are areas in a polygon that are different from the rest of the polygon, but still part of it. Inclusions occur because we typically assume an area represented by a polygon is homogeneous, but this is often untrue, as illustrated in Figure 2-24. The figure shows a vector polygon layer representing raised landscaping beds (a). The general attributes for the polygon may be coded; for example, the surface type may be recorded as cedar mulch. The area noted in Figure 2-24b shows a walkway that is an inclusion in a raised bed. This walkway has a concrete surface. Hence, this walkway is an unresolved inclusion within the polygon.

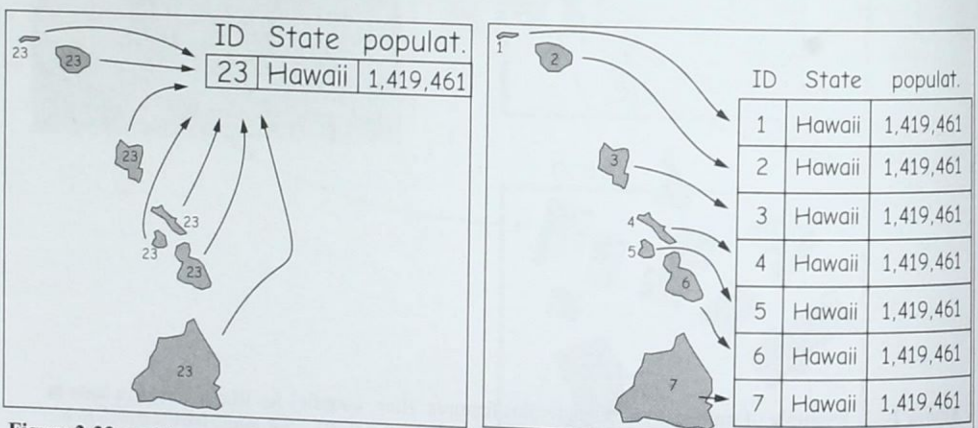


Figure 2-23: multi-part to single-part conversion may lead to errors in subsequent analysis because attributes may be copied from the original, multi-part cluster (left, above), to each single-part component (right, above). Density, sums, or other derived variables often should be re-calculated for single-part features, but often are not, resulting in errors.

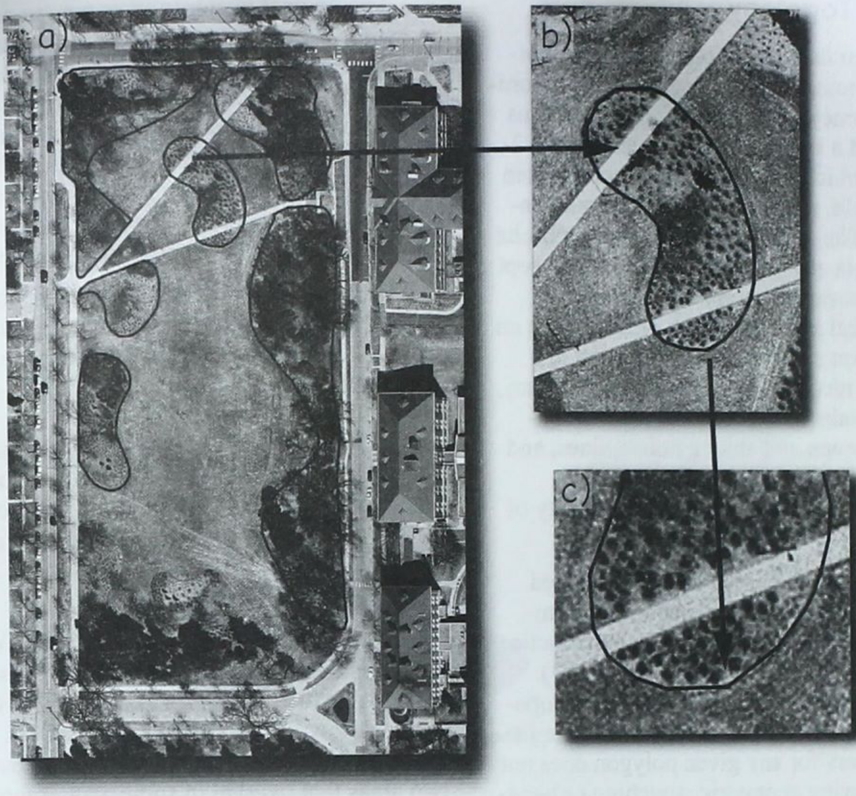


Figure 2-24: Examples of polygon inclusions (sidewalk inclusion in flower bed shown in a and b), and boundary generalization (c) in a vector data model. These approximations typically occur as a consequence of adopting a vector representation, and their impacts must be considered when using vector data.

One solution creates a polygon for each inclusion. This often is not done because it may take too much effort to identify and collect the boundary location of each inclusion, and there typically is some lower limit, or *minimum mapping unit*, on the size of objects we care to record in our data. Inclusions are present in some form in many polygon data layers.

Boundary generalization is the incomplete representation of boundary locations. This problem stems from the typical way we represent linear and area features in vector data sets. As shown in Figure 2-24c, polygon boundaries are represented as a set of connected straight-line segments. The segments are a means to trace the boundaries separating different area features. For curved lines, these straight line segments may be viewed as a sampling of the true curve, and there is typically some deviation of the line segment from the "true" curved boundary. The amount of generalization depends on many factors, and should be so small as to be unimportant for any intended use of the spatial data. However, since many data sets may have unforeseen uses or may be obtained from a third party, the boundary generalization should be recognized and evaluated relative to the specific requirements of any given spatial analysis. There are additional forms of generalization in spatial data, and these are described more thoroughly in Chapter 4.

Vector Topology

Vector data often contain *vector topology,* enforcing strict connectivity and recording adjacency, and planarity. Early systems employed a spaghetti data model (Figure 2-25a), in which lines may not intersect when they should, and may overlap without connecting. The spaghetti model severely limits spatial data analysis and is little used except for very basic data entry or translation. Topological models (Figure 2-25b) create an intersection and place a node at each line crossing, record connectivity and adjacency, and maintain information on the relationships between and among points, lines, and polygons in spatial data. This greatly improves the speed, accuracy, and utility of many spatial data operations.

Topological properties are conserved when converting vector data among common coordinate systems, a common practice in GIS analysis (described in Chapter 3). Polygon adjacency is an example of a topologically invariant property, because the list of neighbors for any given polygon does not change during geometric stretching or bending (Figure 2-25, b and c). These relationships may be recorded separately from the coordinate data.

Topological vector models may vary, and enforce particular types of topological relationships. *Planar topology* requires that all features occur on a two-dimensional surface. There can be no overlaps among lines or polygons in the same layer (Figure 2-26). When planar topology is enforced, lines may not cross over or under other lines. At each line crossing there must be an intersection.

The left side of Figure 2-26 shows nonplanar graphs. In the top left figure, four line segments coincide. At some locations the lines intersect at a node, shown as white-filled circles, but at some locations a line passes over or under another line segment. These lines are nonplanar. The top right of Figure 2-26 shows planar topology enforced for these same four line segments. Nodes are found at each line crossing.

Polygons can also be nonplanar, as shown at the bottom left of Figure 2-26. Two polygons overlap slightly at an edge. This may be due to an error; for example, the two polygons share a boundary but have been recorded with an overlap, or there may be two areas that overlap in some way. If topological planarity is enforced, these two polygons must be resolved into three separate, nonoverlapping polygons. Nodes are placed at the intersections of the polygon boundaries (lower right, Figure 2-26).

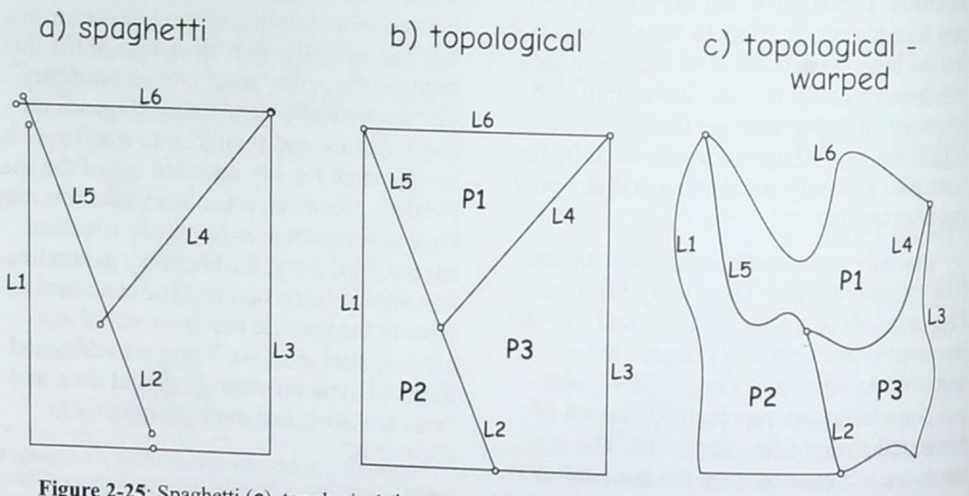


Figure 2-25: Spaghetti (a), topological (b), and topological warped (c) vector data. Figures b and c are topologically identical because they have the same connectivity and adjacency.

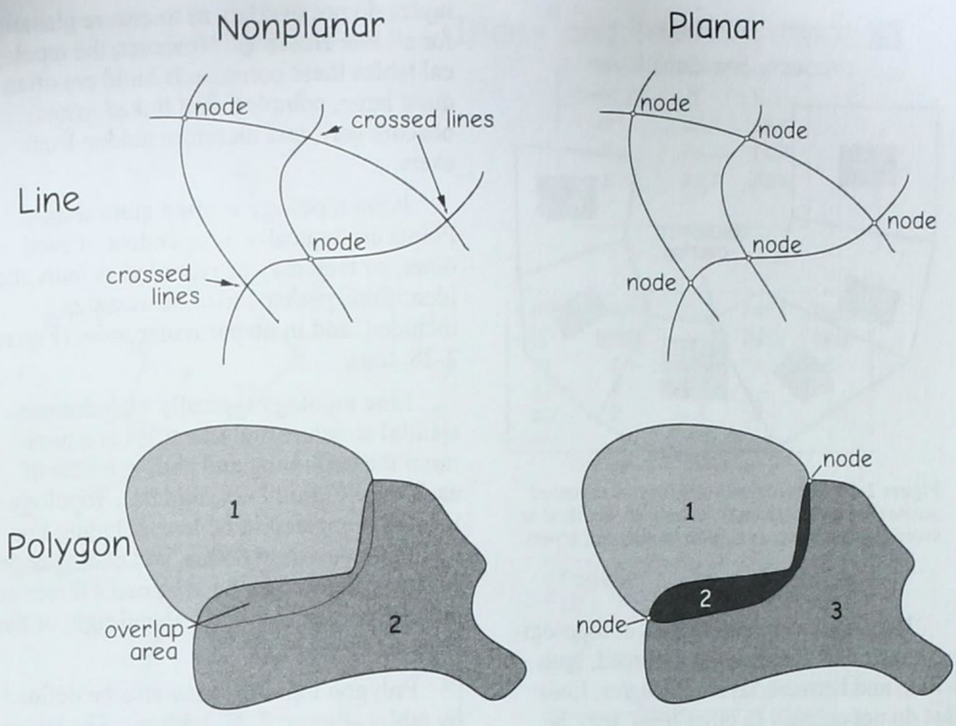


Figure 2-26: Nonplanar and planar topology in lines and polygons.

There are additional topological constructs besides planarity that may be specified. For example, polygons may be exhaustive, in that there are no gaps, holes, or "islands" allowed. Line direction may be recorded, so that a "from" and "to" node are identified in each line. Directionality aids the representation of river or street networks, where there may be a natural flow direction.

There is no uniform set of topological relationships that are included in all topological data models. Different vendors have incorporated different topological information in their data structures. Planar topology is often included, as are representations of *adjacency* (which polygons are next to which) and *connectivity* (which lines connect to which).

Some GIS software create and maintain detailed topological relationships in their data. This results in more complex and perhaps larger data structures, but access is often faster, and topology provides more consistent, "cleaner" data. Other systems maintain little topological information in the data structures, but compute and act upon topology as needed during specific processing.

Topology may also be specified between layers, because we may wish to enforce spatial relationships between entities that are stored separately. As an example, consider a data layer that stores property lines (cadastral data), and a housing data layer that stores building footprints (Figure 2-27). Rules may be specified that prevent polygons in the housing data layer from crossing property lines in the cadastral data layer. This would indicate a building that crosses a property line. Most such instances occur as a result of small errors in data entry or misalignment among data layers. Topological restrictions between two data layers avoid these inconsistencies. Exceptions may be granted in those few cases when a building truly does cross property lines.

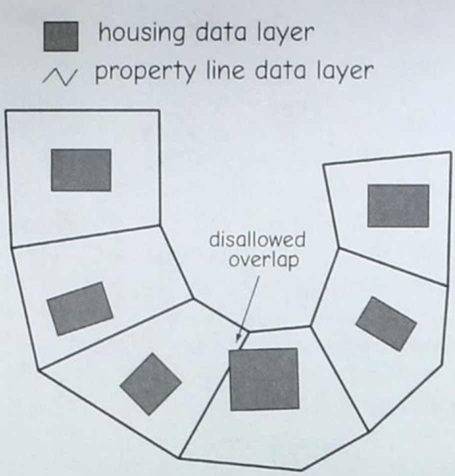


Figure 2-27: Topological rules may be enforced across data layers. Here, rules may be specified to avoid overlap between objects in different layers.

There are many other types of topological constraints that may be enforced, both within and between layers. *Dangles*, lines that do not connect to other lines, may be proscribed, or limited to be greater or less than some threshold length. Lines and points may be required to coincide, for example, water pumps as points in one data layer and water pipes as lines in another, or lines in separate layers may be required to intersect or be coincident. While these topological rules add complexity to vector data sets, they may also improve the logical consistency and value of these data.

Topological vector models often use codes and tables to record topology. As described above, nodes are the starting and ending points of lines. Each node and line is given a unique identifier. Sequences of nodes and lines are recorded as a list of identifiers, and point, line, and polygon topology recorded in a set of tables. The vector features and tables in Figure 2-28 illustrate one form of this topological coding.

Many GIS software systems are written such that the topological coding is not visible to users, nor directly accessible by them. Tools are provided to ensure the topology is created and maintained, that is, there may be directives that require that polygons in two layers do not overlap, or to ensure planarity for all line crossings. However, the topological tables these commands build are often quite large, complex, and linked in an obscure way, and therefore hidden from users.

Point topology is often quite simple. Points are typically independent of each other, so they may be recorded as individual identifiers, perhaps with coordinates included, and in no particular order (Figure 2-28, top).

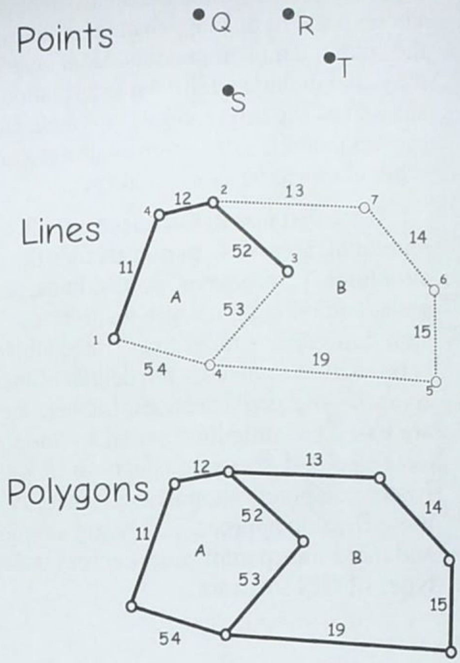
Line topology typically includes substantial structure and identifies at a minimum the beginning and ending points of each line (Figure 2-28, middle). Topology may be organized in tables, including line identifiers, starting nodes, and ending nodes for lines. Lines may be assigned a direction, and the polygons to the left and right of the lines recorded.

Polygon topology may also be defined by tables (Figure 2-28, bottom). The tables may record the polygon identifiers and the ordered list of connected lines that define the polygon.The lines for a polygon form a closed loop, so the starting node of the first line in the list also serves as the ending node for the last line in the list.

Topological models greatly enhance many vector operations. Adjacency analyses are reduced to a "table look-up", a quick and easy operation in most software systems. Assume the city is represented as a single polygon, and we seek all neighboring polygons. Adjacency analysis reduces to 1) scanning the polygon topology table to find the city polygon and reading the list of lines that bound the polygon, and 2) scanning this list of lines, accumulating a list of all left and right polygons. Polygons adjacent to the city may be identified from this list. List searches on topological tables are typically much faster than searches involving coordinate data.

Topological data models often have an advantage of smaller file sizes, largely because coordinate data are recorded once. For example, a nontopological approach

Features | Index and Topology Tables



Point ID	X	Y
Q	32.7	45.6
R	76.3	19.5
S	22.7	15.8
...		

Line ID	Begin node	End node	Left poly	Right poly
11	1	4	...	A
12	4	2	...	A
52	2	3	B	A
...				

Polygon ID	Lines
A	11,12,52,53,54
B	52,53,19,15,14,13
...	

Figure 2-28: An example of vector features and corresponding topology tables. Information on the adjacency, connectivity, and other spatial relationships may be stored in topology tables, and joined to features by indices, here represented by values in the ID columns.

often stores polygon boundaries twice. Lines 52 and 53 at the bottom of Figure 2-28 will be recorded for both polygon A and polygon B. Long, complex boundaries in polygon data sets may double their size. This increases both storage requirements and processing.

There are limitations and disadvantages to topological vector models. First, there are computational costs in defining the topological structure of a vector data layer. Software must determine the connectivity and adjacency information, assign codes, and build the topological tables. Computational costs are typically quite modest with current computer technologies.

Second, the data must be very "clean", in that all lines must begin and end with a node, all lines must connect correctly, and all polygons must be closed. Unconnected lines or unclosed polygons will cause errors during analyses. Significant human effort may be required to ensure clean vector data because each line and polygon must be checked. Software may help by flagging or fixing "dangling" nodes that do not connect to other nodes, and by automatically identifying all polygons. Each dangling node and polygon may then be checked, and edited as needed to correct errors.

Limitations and the extra editing are far outweighed by the gains in efficiency and analytical capabilities provided by topological vector models. Many current vector GIS packages use topological vector models in some form.

Vector Features, Tables, and Structures

As described earlier, geographic features are associated with nonspatial attributes in vector models; tables are used to organize the attributes. In most GIS software, we can most easily view the tables and a graphic representation of the spatial data as a linked table and digital map (Figure 2-29, top).

Most GIS employ underlying file structures to organize components of the spatial data. An example organization is shown in the bottom half of Figure 2-29, where the topological elements are recorded in a linked set of tables, in this example one for each of the polygons, lines, and nodes and vertices. Most GIS maintain the spatial and topological data as a single or cluster of linked files. This internal file structure is often insulated from direct manipulation by the GIS user, but underlies nearly all spatial data manipulations. A user may directly edit or otherwise manipulate table values, usually with the exception of the ID, and the underlying topology and coordinate data are accessed via requests to display, change, or analyze the spatial data components. Data layers may also include additional information (not shown) on the origin, region covered, date of creation, edit history, coordinate system, or other characteristics of a data set.

Note that not all GIS store coordinate and topological data in non-tabular file structures. Coordinates, points, lines, polygons, and other composite features may be stored in tables similar to attribute tables. It is premature to discuss the details of these *spatially enabled* databases, because they are based on something called a *relational data model*, described in detail in Chapter 8. Faster computers support this generally more flexible approach, allowing simpler and more transparent access across different types of GIS software.

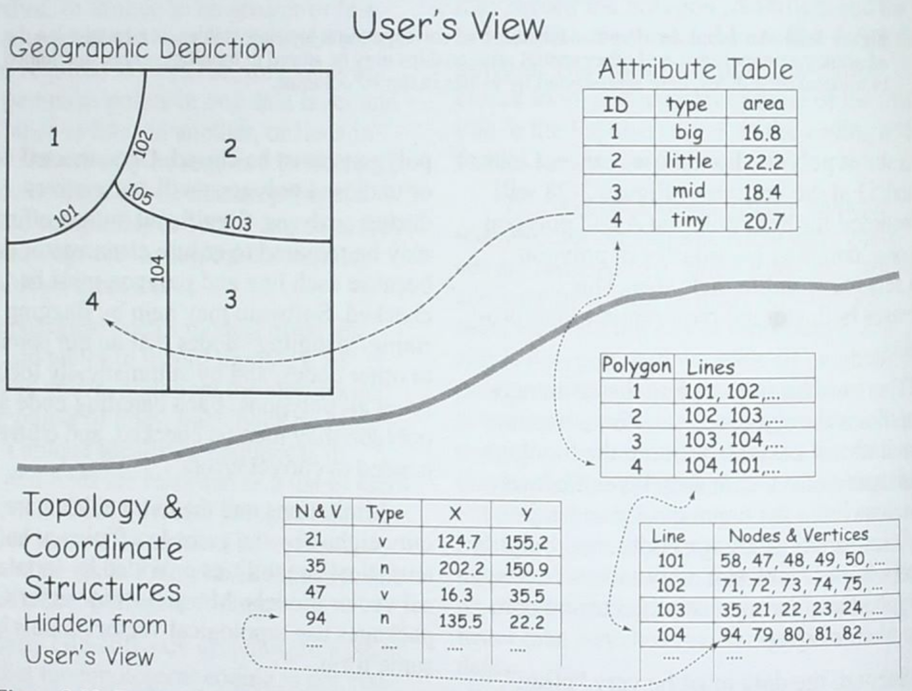


Figure 2-29: Features in a topological data layer typically have a one-to-one relationship with entries in an associated attribute table. The attribute table typically contains a column with a unique identifier, or ID, for each feature. Topology and coordinate data are often hidden from the user, but linked to the attribute and geographic features through pointers and index variables, described in the Data and File Structures section, later in this chapter.

as a
ure
nd
The
re-
h

fea-
a-
are
s.
sig-
as.

quite
els
val-

on,
cell
en-
ters
ted
e
i-

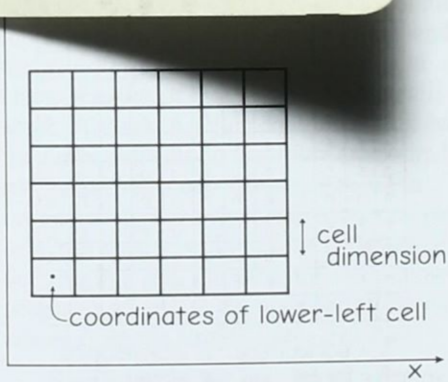


Figure 2-30: Important defining characteristics of a raster data model.

When the cells are square and aligned with the coordinate axes, the calculation of a cell location is a simple process of counting and multiplication. A cell location may be calculated from the cell size, known corner coordinates, and cell row and column number. For example, if we know the lower-left cell coordinate, all other cell coordinates may be determined by the formulas:

$$N_{cell} = N_{low\text{-}left} + row * cell\ size \qquad (2.2)$$

$$E_{cell} = E_{low\text{-}left} + column * cell\ size \qquad (2.3)$$

where N is the coordinate in the north direction (y), E is the coordinate in the east direction (x), and the row and column are counted starting with zero from the lower left cell.

There is often a trade-off between spatial detail and data volume in raster data sets. The number of cells needed to cover a given area increases four times when the cell size is cut in half (Figure 2-31). Smaller cells provide greater spatial detail, but at the cost of larger data sets.

The cell dimension also affects the spatial precision of the data set, and hence positional accuracy. The cell coordinate is usually defined at a point in the center of the cell. The coordinate applies to the entire area covered by the cell. Positional accuracy is typically expected to be no better than approximately one-half the cell size. No matter the true location of a feature, coordinates are truncated or rounded up to the nearest cell center coordinate. Thus, the cell size should be no more than twice the desired accuracy and precision for the data layer represented in the raster, and often it is specified to be smaller.

Each raster cell represents a given area on the ground and is assigned a value that