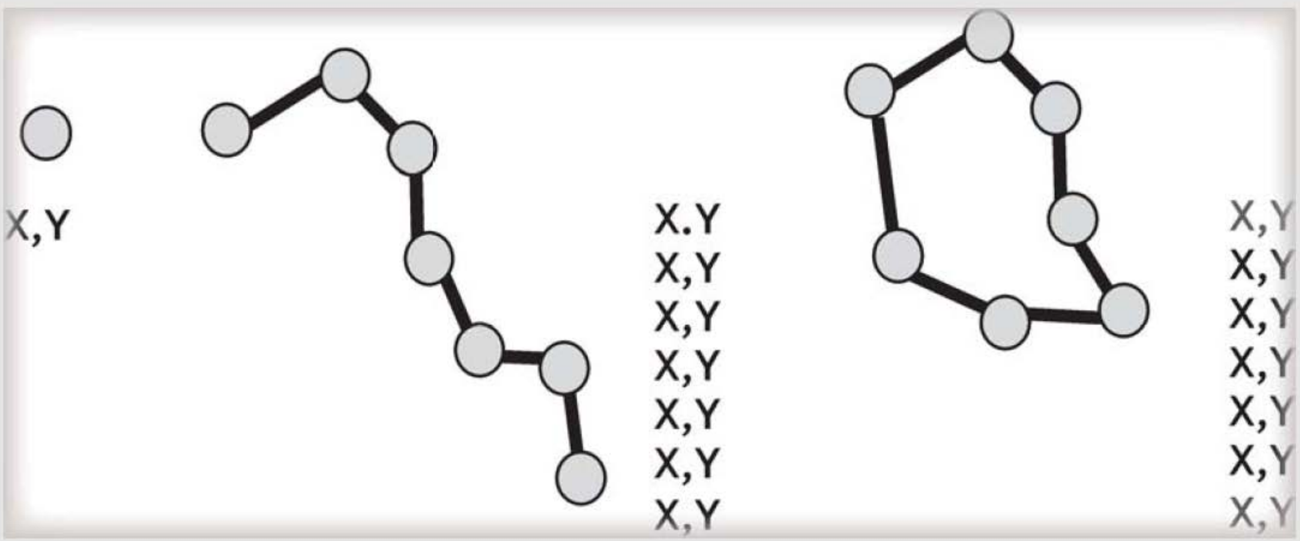
Chapter 7

Basic Data Models in GIS



VECTOR DATA MODEL

- The basis of the vector model is the assumption that the real world can be divided into clearly defined elements.
  - where each element consists of an identifiable object with its own geometry of points, lines, or areas (Figure 7.1).

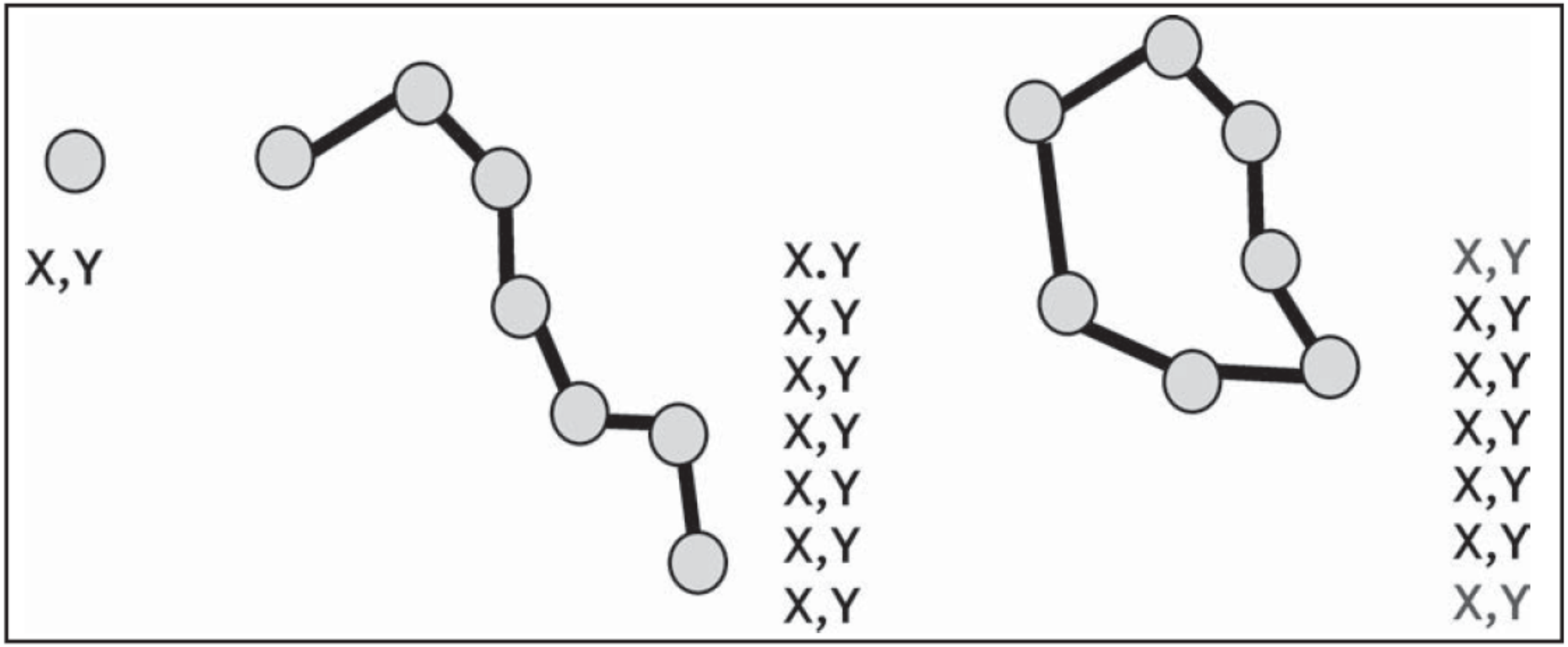


Figure 7.1: The vector data model, objects are described as points, lines or areas (polygons). These three geometric phenomena are described individually by a single point in a coordinate system and with connected lines (lines and area features).

- With few exceptions, digital representations of spatial information in a vector model are based on individual points and their coordinates.
  - The exceptions include cases where lines or parts of lines (*e.g.,* those representing roads or property boundaries) may be described by mathematical functions, such as those for circles or parabolas. In these cases, GIS data include equation parameters
- points, lines, and areas (polygons) are the homogeneous and discrete units that carry information.

CODING DIGITAL DATA FOR MAP PRODUCTION

- In thematic coding, which may be compared to the overlay separation of conventional map production (Figure 7.2), data are divided into single-topic groups, such as all property boundaries.

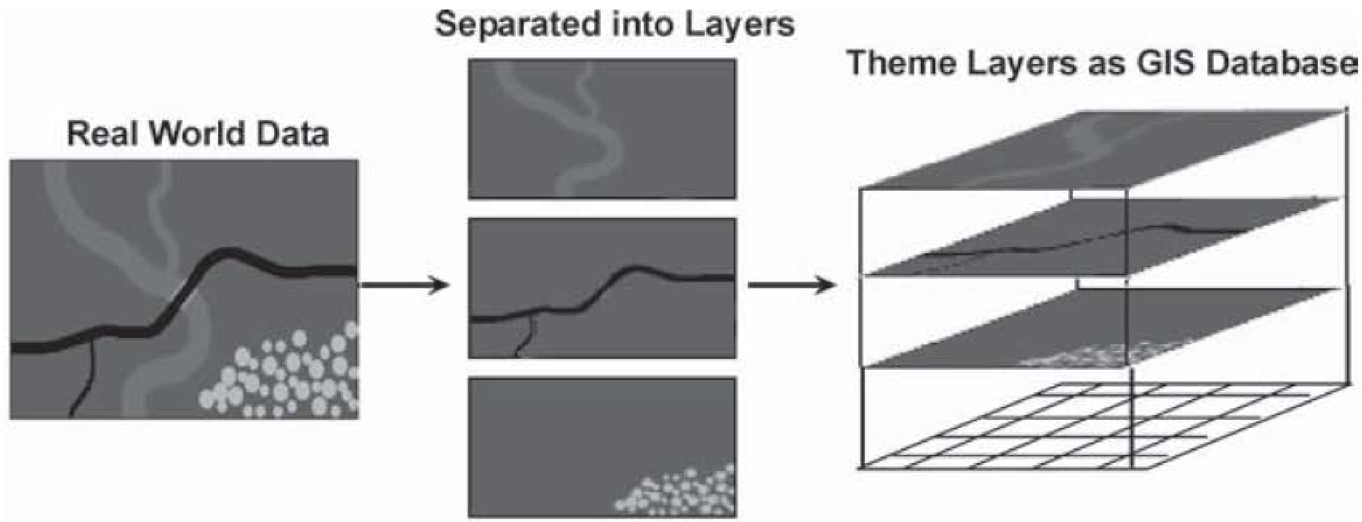
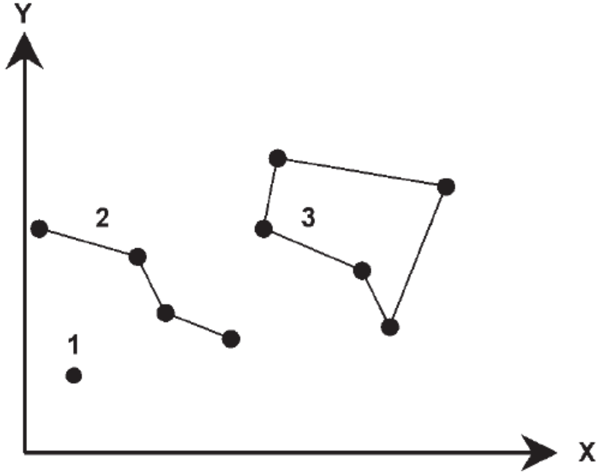


Figure 7.2: Theme codes in the digital map data can be send to separate data into thematic layers.

CODING DIGITAL DATA FOR GIS

- Point objects may easily be realized in a database because a given number of attributes and coordinates is associated with each point (Figure 7.3).



Geometry	ID	Attributes		Coordinates
Point	1	A	B	(Single Point) 2,3
Line	2	C	D	(String) 1,7 3,6 5,5 7,4
Area	3	E	F	(Closed Polygon) 8,7 11,6 12,5 14,9 9,15

Figure 7.3: Each object is assigned attributes and coordinates. The number of coordinates for lines and polygons depends on the length or circumference of the object.

- Line and polygon objects are more difficult to realize in a database because of the variation in the number of points composing them.
- Object spatial information and object attributes are often stored in different databases to ease the manipulation of lines and areas, but in some systems they are stored together.
  - Dividing the databases conserves memory by precluding duplicate storage of the same data.
- Key features of some of the many data structures of vector data are shown in Figure 7.4.
  - Ideas from mathematics are very important in the development of these data structures.

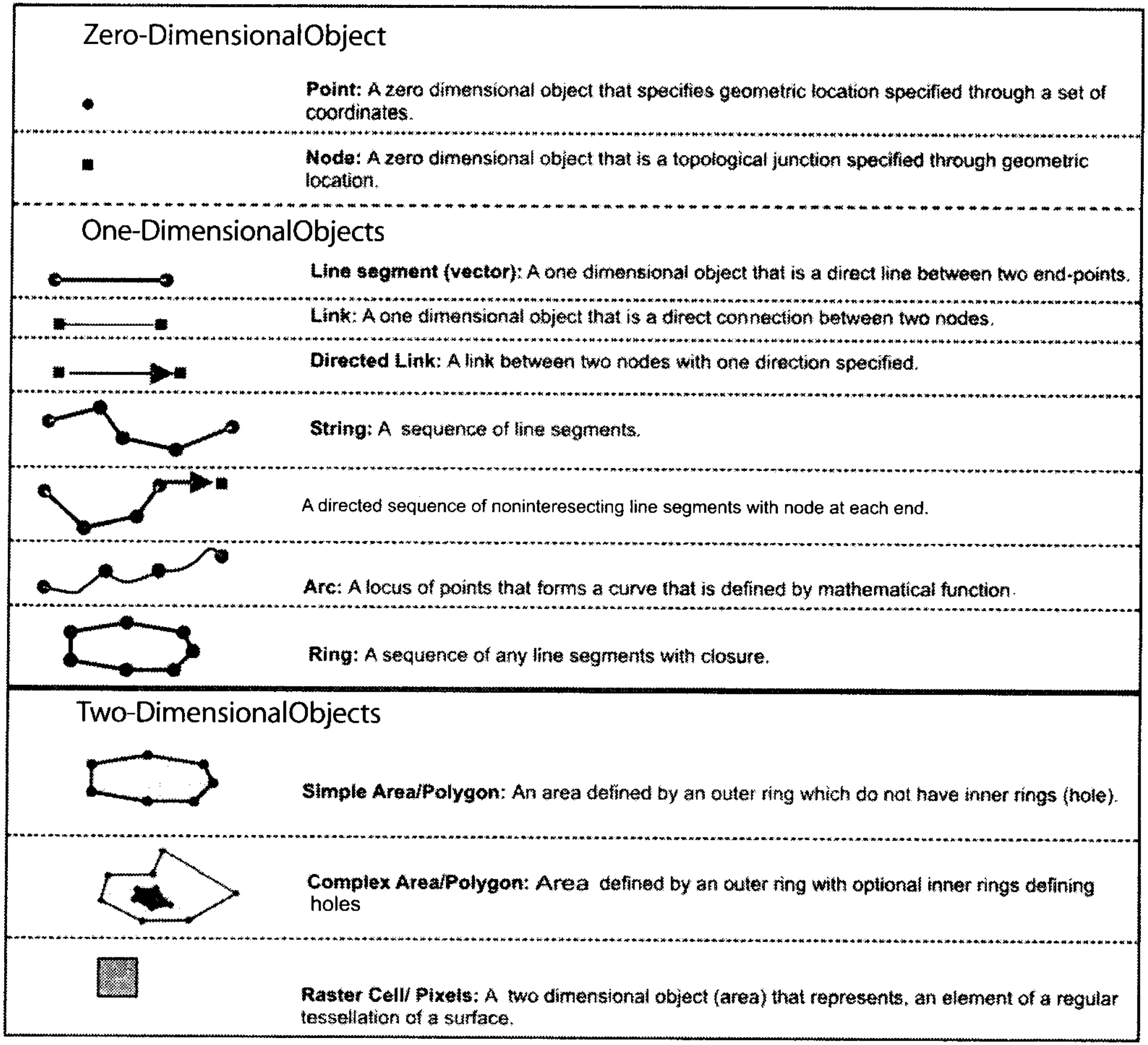


Figure 7.4: Some geometric objects.

STORING POINTS AND LINES

- Figure 7.5 shows a simple vector layer containing examples of all three vector data types: points (*e.g.*, the houses), lines (*e.g.*, the roads and rivers) and areas (*e.g.*, the forests, agricultural fields).

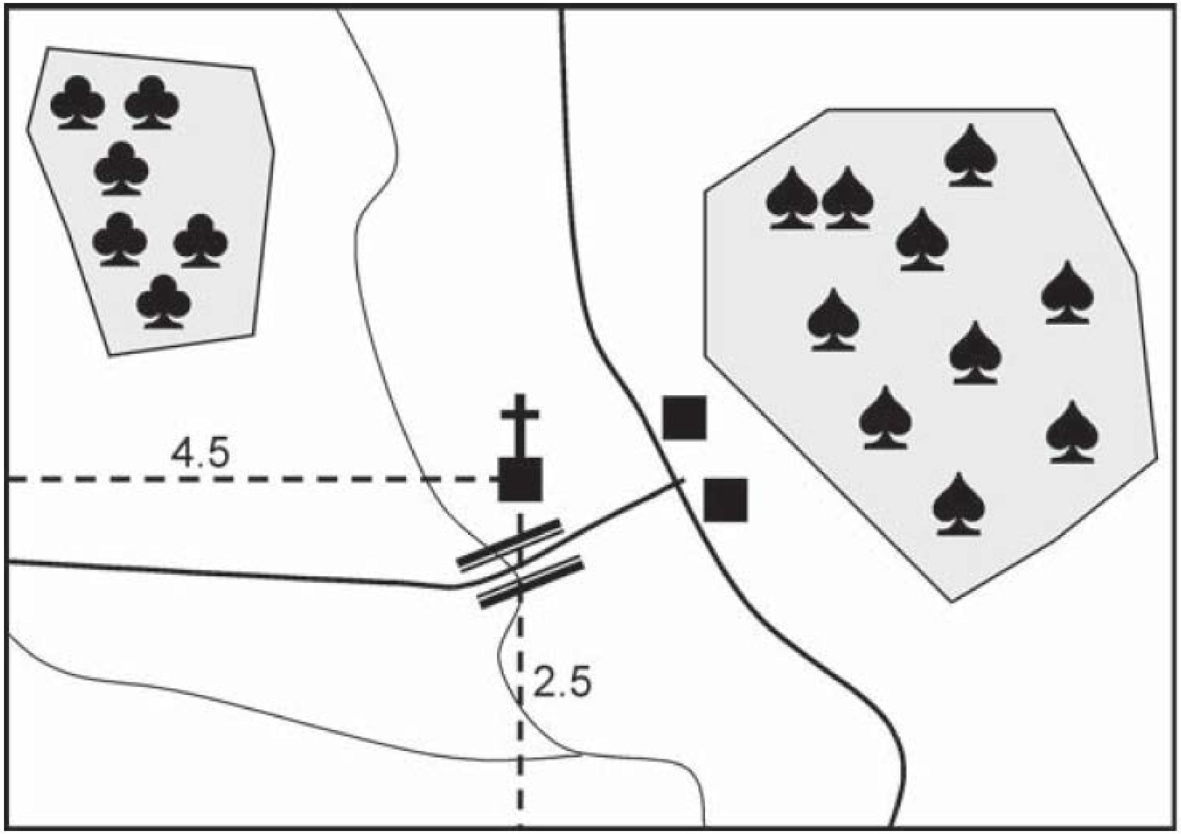


Figure 7.5: Imaginary map.

Points

- The vector representation of the point is as a pair of coordinates.
  - To store the positions of points, we measure its distance from the left hand corner of the map both horizontally and vertically as shown and this will give us a pair of figures.(Table 7.1).

Table 7.1: Coordinate data for points.

Building 1	4.5	2.5
Building 2	5.8	2.9
Building 3	6.0	2.2

- By storing the location of the points as a pair of coordinates, we can do three things:
  - 1. Store the locational data from the map on a computer.
  - 2. Use this information to reproduce the map.
  - 3. Make simple calculations from the data.

- In practice, measuring the coordinates from the origin of the map is not very useful, especially if the area we are interested in covers more than one map sheet.
  - Many, for example use the UTM (Universal Transverse Mercator) system which gives a standard way of representing positions for any point in the region.
- Storing the attributes of each point.
  - Table 7.2 indicates one possible way of doing this – using extra columns in the table.

Table 7.2: Attribute data for points.

	X Coordinate	Y Coordinate	Feature Code	Building Material	Name
Building 1	4.5	2.5	Temple	Stone	Krishna's
Building 2	5.8	2.9	House	Brick	Shaan's
Building 3	6.0	2.2	House	Stone	Sameer's

Lines

- To store the location of the lines, we simply need the coordinates of the points between each straight section.

Table 7.3: Coordinate data for part of a line.

4.5	10.0
4.5	5.7
5.5	2.5
6.5	0.3
6.8	0.0

- With a curved line, such as a river, this representation by a series of straight sections will only be an approximation of course as shown in Figure 7.6.

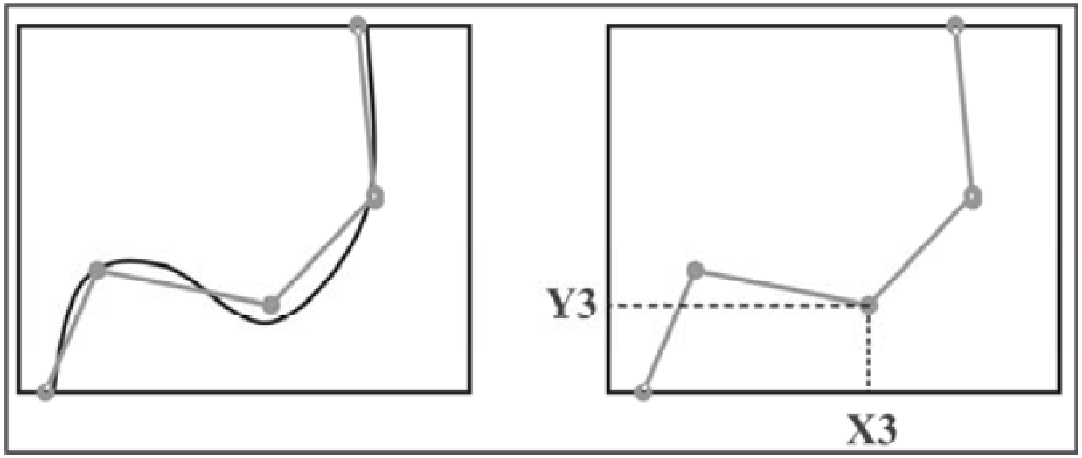


Figure 7.6: Approximating a curved line with a series of straight segments.

- The only way to improve the match between the real line and our series of straight segments is to add extra points along the line.
  – Each extra point means more data to be stored and hence larger files on the computer.

- Storing attributes for each line.
  - Imagine that we wish to store the following information about our roads.

Table 7.4: Attribute data for line's.

Name	Surface Quality	Peak Traffic Flow
Khair Road	Fair	600
Anupshahar Road	Good	1000

  - The problem comes when we try and combine both sets of information into one table.

Table 7.5: Adding locational information to attribute table for lines. Rows containing data for Khair Road are shaded light grey.

Name	Surface Quality	Peak Traffic Flow	X Coordinate	Y Coordinate
Khair Road	Fair	600	4.5	10.0
			4.5	5.7
			5.5	2.5
			6.5	0.3
			6.8	0.0
Anupshahar Road	Good	1000	0.0	1.5
			3.6	1.5
			5.5	2.5

– Each feature is now represented by a different number of rows, depending on how many X Y coordinate pairs we have.

- To keep each feature in a single row, we could add more columns to each row as shown in table 7.6.

Table 7.6: Alternative method of adding locational information to attribute table for lines.

Name	Surface Quality	Peak Traffic Flow	X_1	Y_1	X_2	Y_2	X_3	Y_3	X_4	Y_4	X_5	Y_5
Khair Road	Fair	600	4.5	10.0	4.5	5.7	5.5	2.5	6.5	0.3	6.8	0.0
Anupshahar Road	Good	1000	0.0	1.5	3.6	1.5	5.5	2.5				

  – Now we have different numbers of columns for each feature.

- Because of this, many vector GIS systems solve the problem by storing the locational and attribute data separately.
  - The attribute data is stored in a standard database package,
  - The locational data is stored using specially written software which can handle its more complicated structure.

- The identity (ID) codes and identifiers
  - The ID codes are used to label and connect spatial information and attribute table data.
  - They are most often numerical, but may be alphanumerical.
  - ID codes allow differentiation between objects.
    - Whereas theme codes allow for differentiation between different groups of objects.
  - Spatially defined objects without attributes need no identifiers, but they are required for all objects that are listed in attribute tables, and manipulated spatially.
  - Identifiers are normally entered together with the relevant data, but they may also be entered later.
- Let us first see how this works with the point data.
  - In the original table (Table 7.1) we had a column which simply identified each point as Building 1, Building 2 etc.

- Instead we will now have a column labelled Building – ID which will contain the Identification number of each building.
- We can then split our original table into two tables (see Tables 7.7 and 7.8), one each for the locational and attribute data.

Table 7.7: Locational data for buildings.

Building – ID	X Coordinate	Y Coordinate
1	4.5	2.2
2	5.8	2.9
3	6.0	2.2

Table 7.8: Attribute data for buildings.

Building – ID	Feature Code	Building Material	Name
1	Temple	Stone	Krishna's
2	House	Brick	Shaan's
3	House	Stone	Sameer's

- Let us see how this works with the road data.
  - We might use the road identification number as our unique ID but this will not be a good idea if we wish to distinguish between different parts of the Khair road for example.
  - So again we will simply use a number starting at 1.
  - Our attribute table will now be as shown in Table 7.9.

Table 7.9: Modified attribute table for roads.

Road – ID	Name	Surface Quality	Peak Traffic Flow
1	Khair Road	Fair	600
2	Anupshahar Road	Good	1000

STORING AREA BOUNDARIES

- Figure 7.7 shows a simple area of forest, and one way to store this area is by storing the line which defines its boundary, as shown in Figure 7.7B.

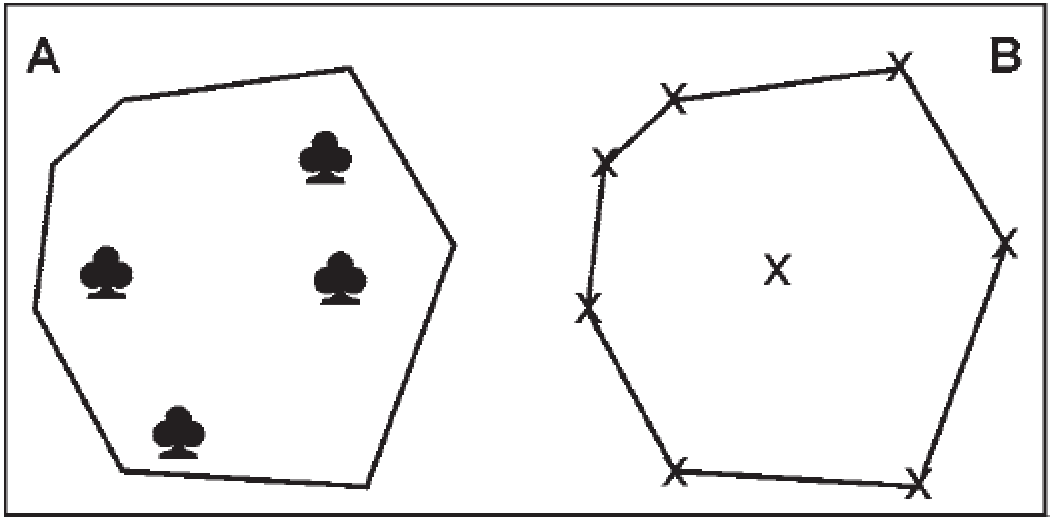


Figure 7.7: Storage of area feature.

- Figure 7.7B is what mathematicians call a polygon a closed shape described by a series of straight lines – and in the GIS literature the term polygon is often used to refer to areas.
- It is very common to store a centroid for each area, which is a point that is located inside the polygon as shown in Figure 7.7.
- The use of centroids means that to store a single area, we need to store two things – the line defining the boundary and the point defining the centroid.

- Figure 7.8 shows our original forest area as part of a land use map.

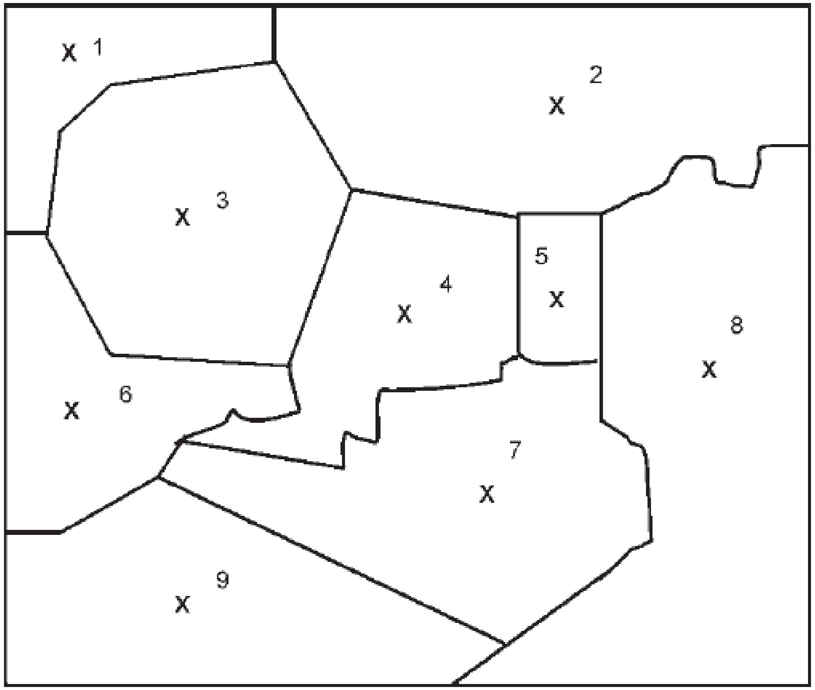


Figure 7.8: Land use map – an example of multiple areas.

- Each area has a centroid, with an identifier associated with it, and this identifier is used as the link to a table containing the attributes for the areas (Table 7.10).

Table 7.10: Attribute table for land use map.

Polygon ID	Land Use
1	Vacant Land
2	Vacant Land
3	Forest
4	Urban
5	Agriculture
6	Vacant Land
7	Vacant Land
8	Forest
9	Agriculture

- We can still use the simple method of storing the areas shown on Figure 7.7 but we will run into a number of problems.
  - The majority of the boundary lines lie between two areas, and will be stored twice in this way-the result is that we will store nearly twice as much data as necessary.
  - When the same line is digitized a second time, slightly different points will be chosen, with the result shown in Figure 7.9.

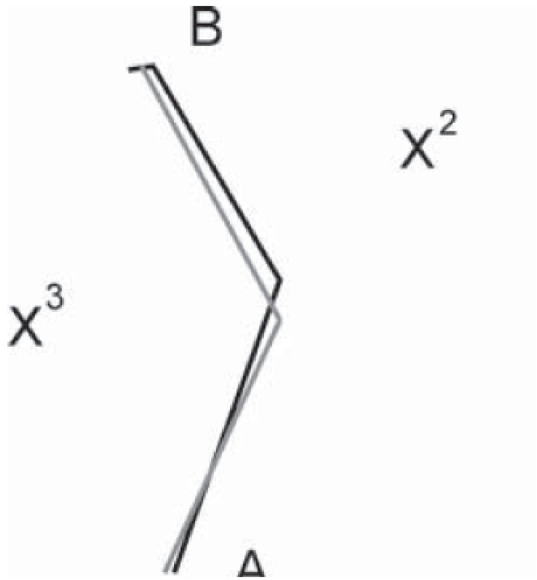


Figure 7.9: Sliver polygons as a result of digitizing the same line twice.

  - These mismatches areas are called sliver polygons, because they are usually very small and thin.
- There is a third problem with this method of storing area boundaries which arises if we wish to use our data for analysis.
  - We do not have any information about the contiguity of our polygons (which ones are neighbours to each other) and to store this information we need a different method of storing our area boundaries.

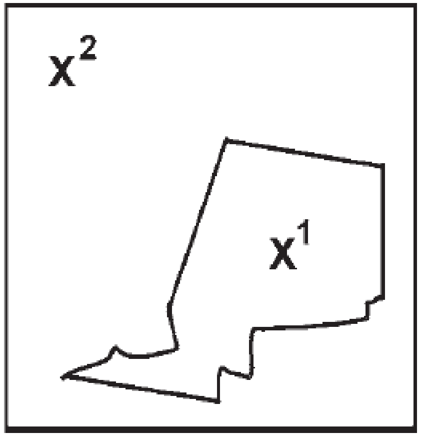
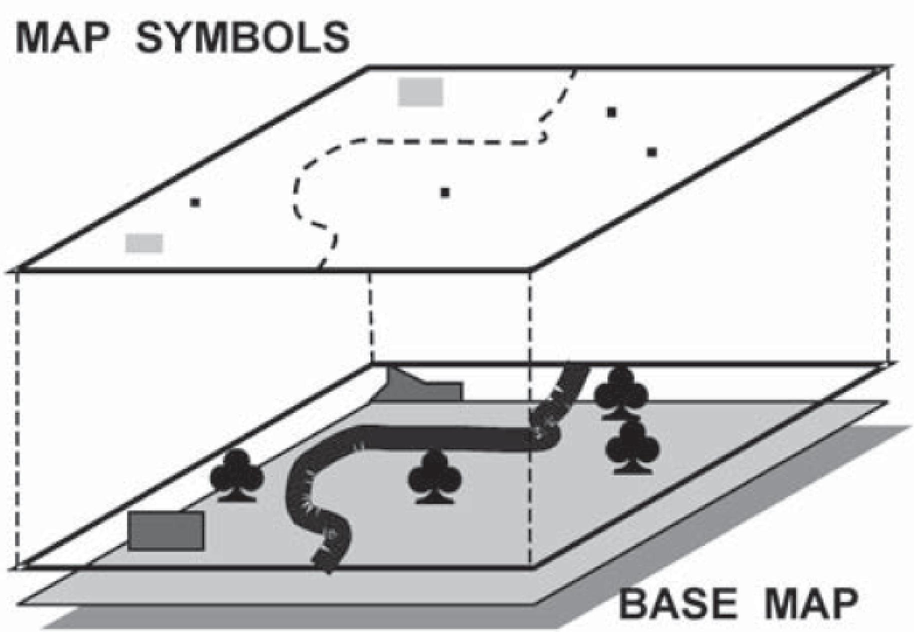


Figure 7.10: Map of urban and non-urban areas created using a polygon dissolve operation.

- Some systems tie a polygon's ID code to a characteristic point in the polygon, known as the label point.



Look-up Table

Thematic Code	Map Symbols
10	Dots
20	Dotted Lines
30	Shaded Area

Table

ID	Thematic Code	Map Coordinates
Tree	10	X_1, Y_1
River	20	X_2, Y_2
Builtup Land	30	X_3, Y_3

Figure 7.11: Drawing instructions are designated in look-up tables. Thematic code values or attribute values are input values in the tables, while output values can be symbol types, colours, line thickness etc.

SPAGHETTI MODEL

- Spaghetti data are a collection of points and line segments with no real connection (Figure 7.12).

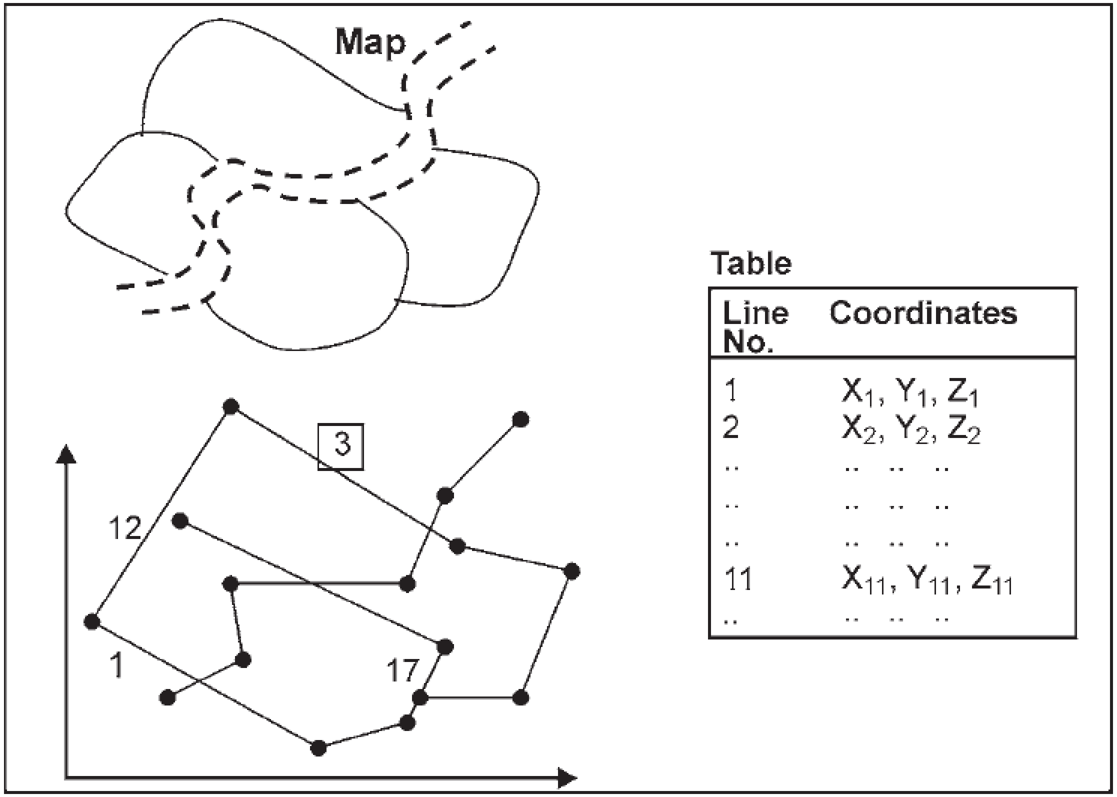


Figure 7.12: Spaghetti data is a term used to describe digital map data with crossing lines, loose ends, double digitization of common boundaries between adjacent polygons, etc. These data lie in a pile, just like spaghetti. Several line segments are found at odd places in the data file.

- Unlinked (spaghetti) data usually include data derived either from the manual digitizing of maps or from digital photogrammetric registration.
- Drawbacks of spaghetti data
  - Both data storage and data searches are sequential. Hence search times are often unduly long for routine operations.
  - Other operations such as overlaying and network analysis, are intractable.
  - Unlinked data require an inordinate amount of storage memory because all polygons are stored as independent coordinate sequences.

STORING AREA BOUNDARIES: THE TOPOLOGICAL APPROACH

- To overcome the limitations of the simple method of storing polygons, GIS systems draw on ideas called *topology.*
- Topology can be broadly explained as the way in which area data is stored in GIS systems.
- In Figure 7.13, we can see that each area boundary is made up of a series of line sections, which meet at junctions.
- If we identify each of those junctions, and store the location of the line section joining them, we will have stored all the boundary information, but without duplication.
  - The junctions are called nodes.
  - The sections of lines between them are links.
    - They are also called as arcs, chains, segments and edges.

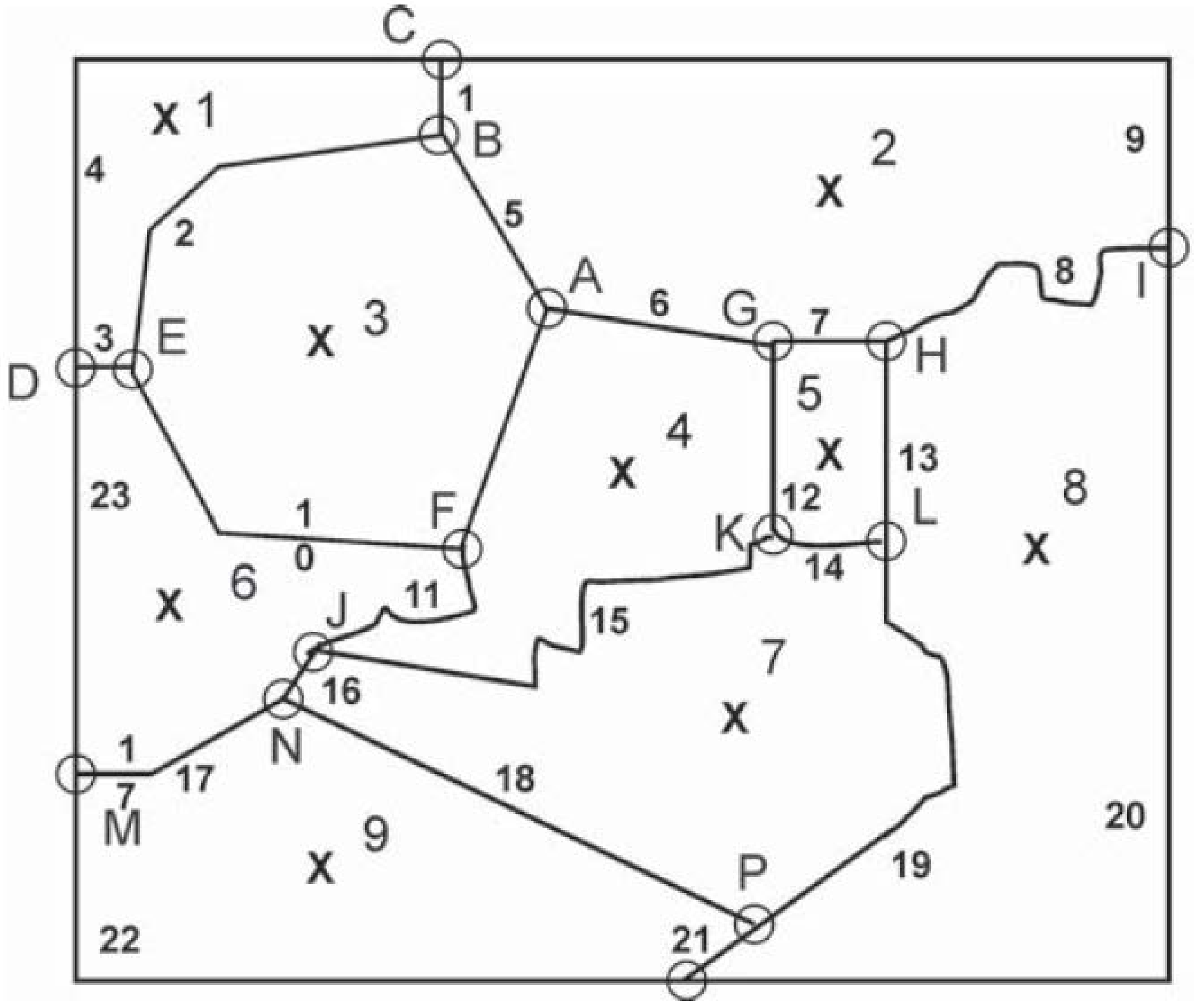


Figure 7.13: Land use map as stored in a topological GIS.

- For each link, we can store the location of the end points (the nodes) and of a series of positions along the line recording its location; and for each link we store the identifier of the polygons on either side.

Table 7.11: Information stored for a link in a link and node data structure.

Link ID	Start Node	End Node	Left Polygon	Right Polygon	X_1	Y_1	X_2	Y_2	X_n	Y_n
5	A	B	3	2						

- The direction of the link is not important – we could equally well store this link as running from B to A with polygon 2 on the left and 3 on the right.

- One of the strengths of the link and node structure is that it can be used to check for errors in the data.
- The structure also solves our other problems.
  - First, each link is only stored once, thus saving the problem of duplicating data.
  - Second, we now have information about the relationship between areas, which can be useful for analysis.
- The key to the link and node method of storing area boundaries is in the information which describes the relationships between objects on the map – the left/right identifiers and the to/from node identifiers for each link.

SO WHAT ACTUALLY IS TOPOLOGY?

- The study of relationships such as contiguity is part of the mathematical subject of topology, which is concerned with those characteristics of objects which do not change when the object is deformed.
- The connection between areas and containment is a topological property of the map.
- The Konigsberg bridge problem, has a direct relevance to the use of topological ideas in GIS.
  - Euler realized that the problem had nothing to do with the distances or directions involved, but depended solely on the connections between places.

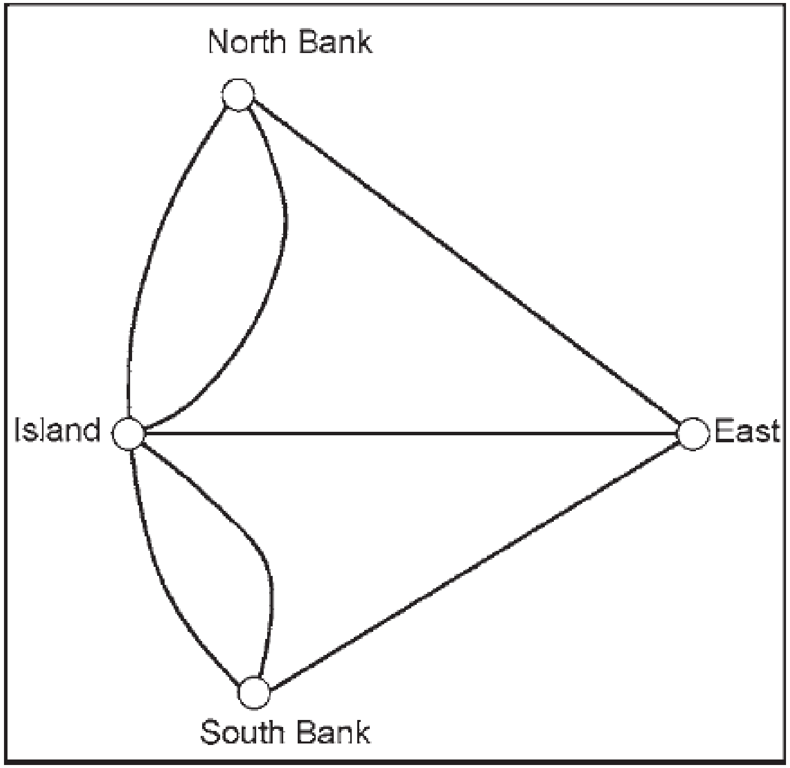


Figure 7.14: Sketch of the Konigsberg bridge problem.

- Dual graph model of maps by Poincaré
  - The graph theory of Henri Poincaré could be applied to maps in general and used at the US Bureau of the Census to help in processing the data for the 1980 census.
  - In figure 7.15, we can construct two graphs – one based on the streets surrounding a block, one on the blocks surrounding a street intersection.
  - Mathematically, the two graphs are exactly the same, since both will consist of a single closed loop.
  - If we can automatically create these graphs from our street map, and check that they do form closed loops we will have a way of checking the data.

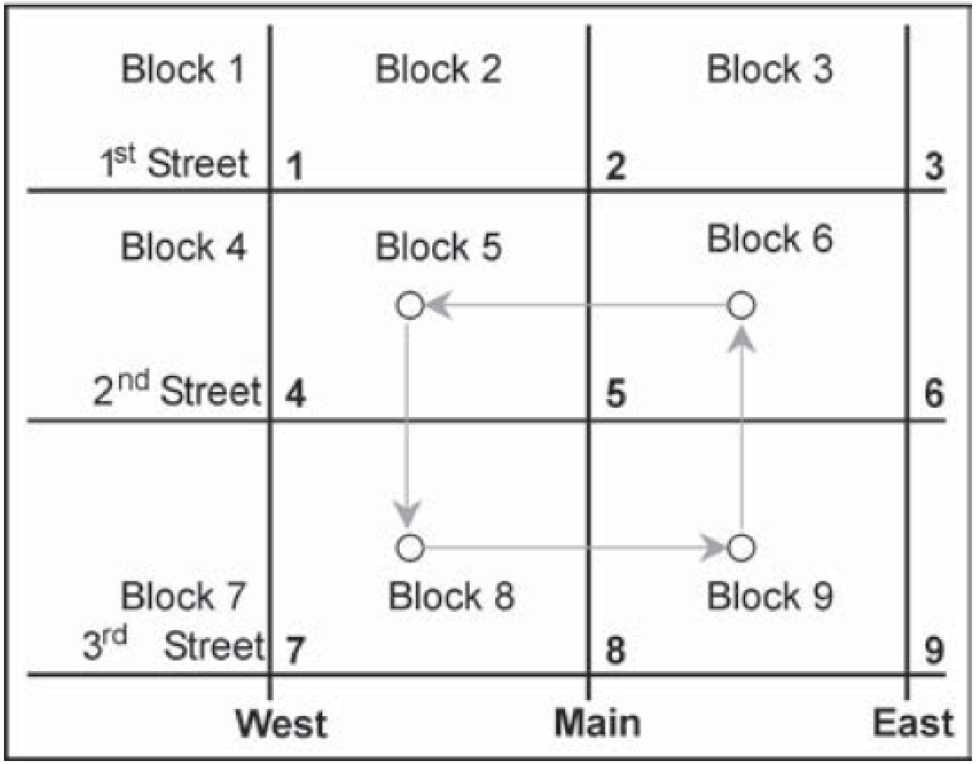


Figure 7.15: Fictitious city blocks illustrating Poincarés dual graph model of maps.

HOW IT WORKS? THE DIME EXAMPLE

- DIME (Dual Independent Map Encoding) was based upon a data structure in which one record was created for each street segment the part of a street between two junctions.
- For the map in Figure 7.15, a portion of the DIME – file might be as shown in Table 7.12.

Table 7.12: DIME data structure for fictious city map.

Segment	From	To	Block Left	Block Right	Street Name
1	1	2	2	5	1st
2	2	3	3	6	1st
3	4	5	5	8	2nd
4	5	6	6	9	2nd
5	7	8	8	11	3rd
6	8	9	9	12	3rd
7	4	1	4	5	West
8	7	4	7	8	West
9	5	2	5	6	Main
10	8	5	8	9	Main

- We first find those records in the DIME file which have 5 on either their left or right; In this case, we will find segments 1, 3, 7 and 9.
- We then need to check that these segments form a closed loop.
- If they all have block 5 on their right hand side, this means they will form a clockwise loop.
- So we change the direction of the segments 3, 9.
  - Switch the left and right blocks and then to and from nodes to produce the records shown in Table 7.13.

Table 7.13: Records from DIME file related to block 5. All records are modified so that block 5 is on right hand side of the street.

Segment	From	To	Block Left	Block Right	Street Name
1	1	2	2	5	1st
3	5	4	8	5	2nd
7	4	1	4	5	West
9	2	5	6	5	Main

- We can now start at any segment and try and trace round the loop.
- If for any reason we can't complete this loop, we know there is an error in the data.
  - Such as a segment completely missed out, or one in which the block numbers or node numbers were wrong.

- This checking process can, also be carried out using the graph based around the street junctions (Table 7.14).

Table 7.14: Records from DIME file relating to junction 5.

Segment	From	To	Block Left	Block Right	Street Name
3	4	5	5	8	2nd
4	5	6	6	9	2nd
9	5	2	5	6	Main
10	8	5	8	9	Main

- In both the block and junction checking, we are using the left/right and from/to information to trace around the topological graph.
  - There must be one closed graph in each case.

Table 7.15: Records from DIME file relating to junction 5 modified so that the street ends at junction 5.

Segment	From	To	Block Left	Block Right	Street Name
3	4	5	5	8	2nd
4	6	5	9	6	2nd
9	2	5	6	5	Main
10	8	5	8	9	Main

- In DIME file, the geographical referencing was done via addresses.

Table 7.16: Storage of address information in the DIME data structure.

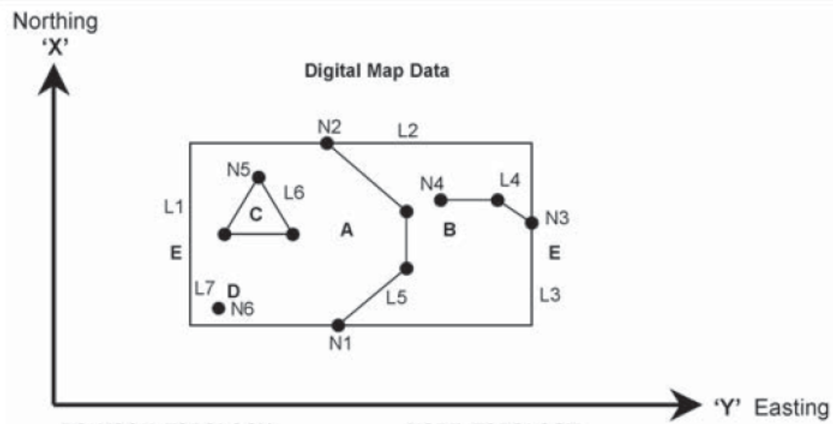
Segment	From	To	Block Left	Block Right	Street Name	Left Address Low	Left Address High	Right Address Low	Right Address High
1	1	2	2	5	1st	12	24	13	25

- In order to develop this into a more general purpose data structure, it was necessary to allow the lines between the nodes to take on any shape, as described by a set of XY coordinates, in this way we reach the link and node data structure.

Topology Model – Connections and Relationships between Object

- The topology model is based on mathematical graph theory and employs nodes and links.
  - A node can be a point where two lines intersect, an endpoint on a line, or a given point on a line.
  - A link is a segment of a line between two nodes.
  - Links connect to each other only at nodes.
  - A closed polygon consisting of alternating nodes and links forms an area.
  - Single points can be looked upon as a degenerate node and as a link with zero length (Laurini and Thompson, 1992).
- Unique identities are assigned to all links, nodes, and polygons, and attribute data describing connections are associated with all identities.

- Topology can therefore be described in three tables (Figure 7.16):
  - i. The polygon topology table lists the links comprising all polygons, each of which is identified by a number.
  - ii. The node topology table lists the links that meet at each node.
  - iii. The link topology table lists the nodes on which each link terminates and the polygons on the right and left of each link, with right and left defined in the direction from a designated start node to a finish node.
- The geometry of the objects is stored in its own subordinate table (see Figure 7.16).
- Topological attribute data may be used directly in contiguity analyses and other manipulations with no intervening, time – consuming geometric operations.



POLYGON TOPOLOGY

Polygon	Links
A	L1, L5
B	L2, L3, L5
C	L6
D	L7
E	L1, L2, L3

NODE TOPOLOGY

Node	Links
N1	L1, L3, L5
N2	L1, L2, L5
N3	L2, L3, L4
N4	L4
N5	L6
N6	L7

LINK COORDINATES

Link	Coordinates			
L1	4, 10	4, 4	11, 4	11, 9
L2	11, 9	11, 16	8, 16	
L3	8, 16	4, 16	4, 10	
L4	8, 16	9, 15	9, 13	
L5	11, 9	8, 11	6, 11	4, 10
L6	10, 7	7, 8	7, 5	10, 7
L7	5, 5			

LINK TOPOLOGY

Link	Start Node	End Node	Left Polygon	Right Polygon
L1	N1	N2	E	A
L2	N2	N3	E	B
L3	N3	N1	E	B
L4	N3	N4	B	B
L5	N2	N1	B	A
L6	N5	N5	A	C
L7	N6	N6	A	A

Figure 7.16: Topology model have geometric objects. Digital map data are represented by nodes and links. The objects attributes and relationships are described by storing nodes and links in tables, i.e. Polygon Table; Node Topology Table; Link Topology Table and an additional table showing Objects Geographical Coordinates.

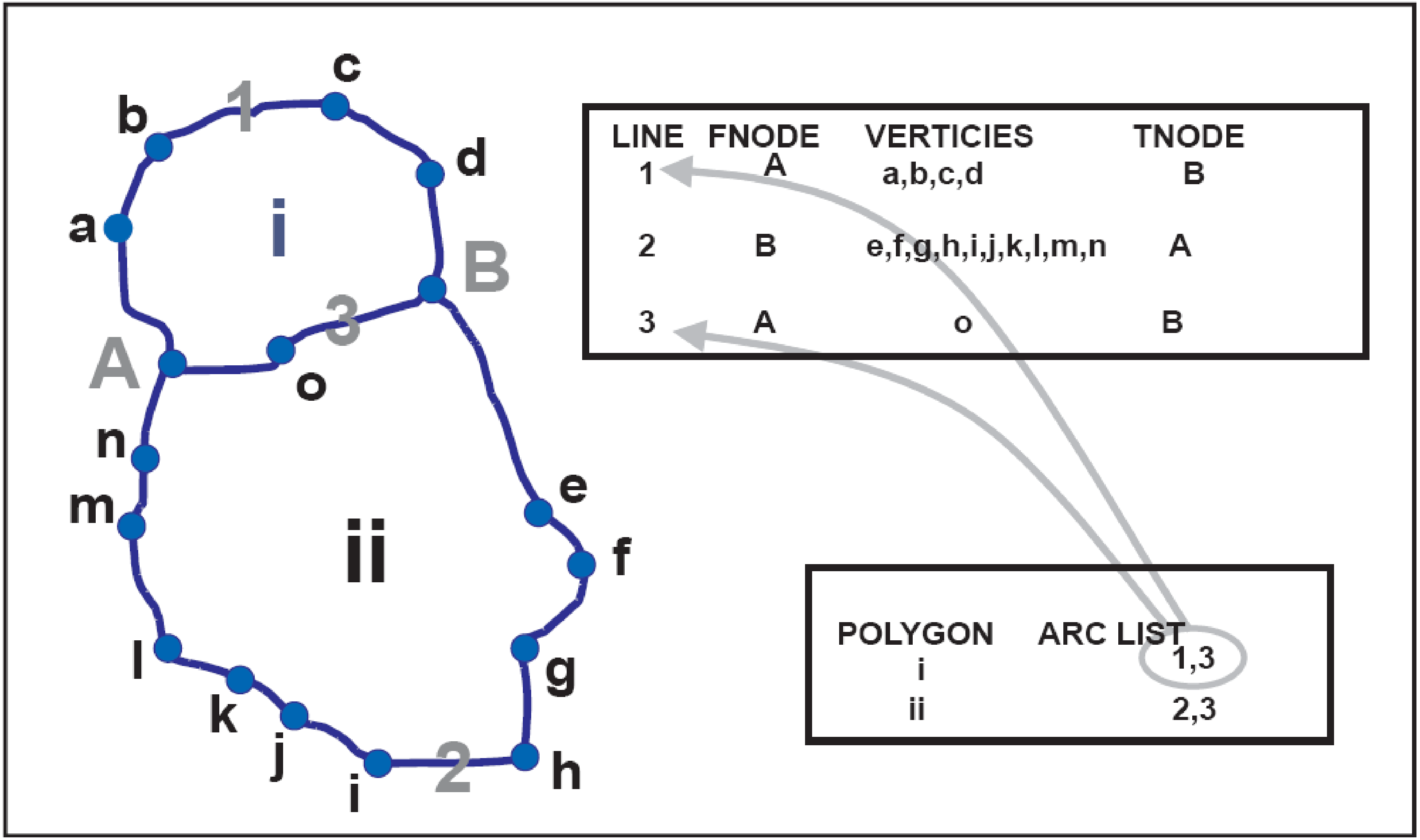


Figure 7.17: Representing polygon in a topology.

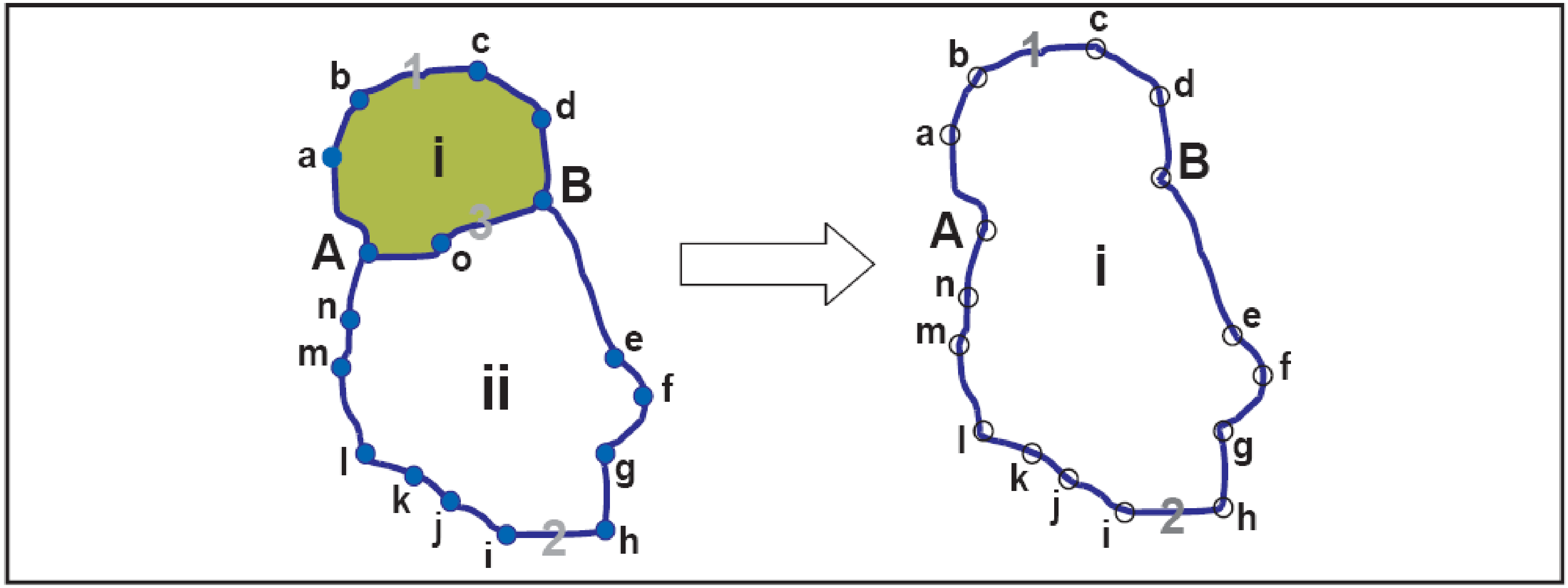


Figure 7.18: Impact of editing in topology.

- Topology requires that all lines should be connected, all polygons closed, and all loose ends removed.
  - Using the snap function with a defined tolerance, the two points will be snapped together to form a common node, thereby closing the polygon.
  - A node can also be created in existing data by calculating the point of intersection between lines.
  - Meaningless loose ends can be removed by testing with a given minimum length.
- The topological model has a few drawbacks.
  - The computational time required to identify all nodes may be relatively long.
  - Uncertainties and errors may easily arise in connection with the closing of polygons and formation of nodes in complex networks.
- When new data are entered and existing data updated, new nodes must be computed and the topology tables brought up to date.

DATA COMPRESSION

- The amount of memory needed can be reduced by using data compression techniques.
- Most of these automatic techniques are based on removing points from continuous lines (contour lines, etc.).
- Good data compression techniques are those that preserve the highest possible degree of geometric accuracy.
- The most basic technique involves the elimination of repetitive characters.
  - for example, the first character of all coordinates along a particular axis.
  - Savings in characters stored are illustrated in Table 7.17.

Table 7.17: Simple data compression. The volume of data to be stored is reduced to a single entry, assigning the value common to all coordinate values.

Original Data		Compacted Data	
Northing	Easting	10000	80000
10,234	80565	234	565
10245	80598	245	598
10167	80324	167	324
--	--	--	--
--	--	--	--
--	--	--	--

- Douglas-Peucker algorithm
  - Draw a straight line between the first and last points on a curved stretch of line and calculate the orthogonal distance from each point on the curved line below the straight one.
  - Points that are closer than a given distance from the straight line will be removed.
  - The endpoint of the straight line is then moved to the point with the greatest distance and the same procedure for removing points is repeated.
  - This continues until all the relevant points are removed.

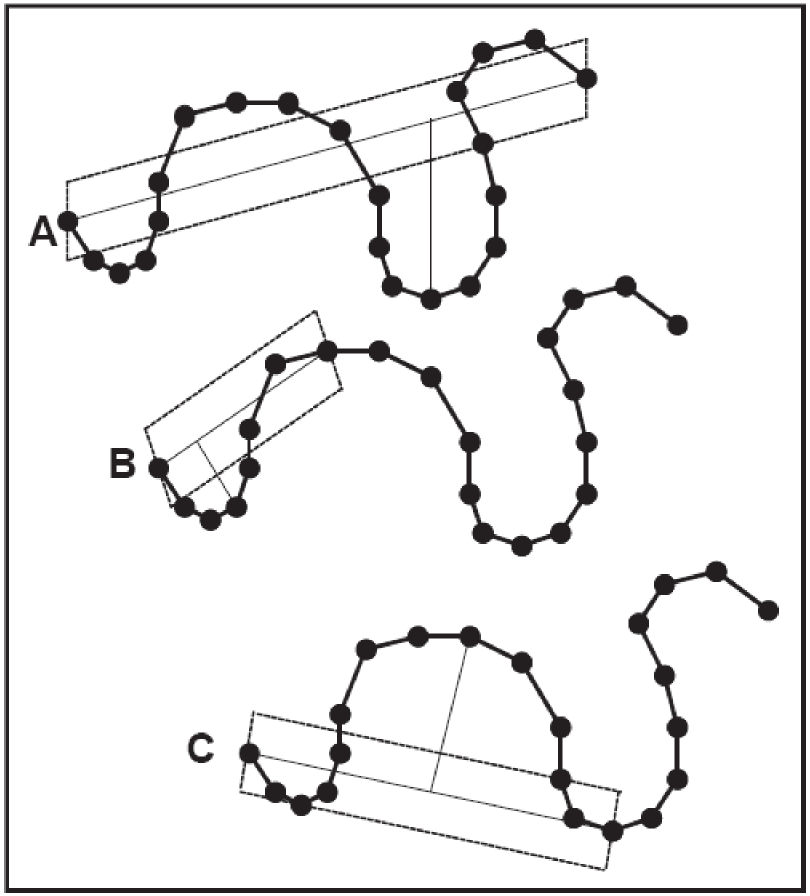


Figure 7.19: Douglas-Peucker Method, this helps in saving storage space.

- Points of little or no value in describing a line may be eliminated by moving a corridor step by step along a line and deleting points that are closer to the neighbouring point than a given value or where the vectors create an angle that is smaller than the given value (Figure 7.20).

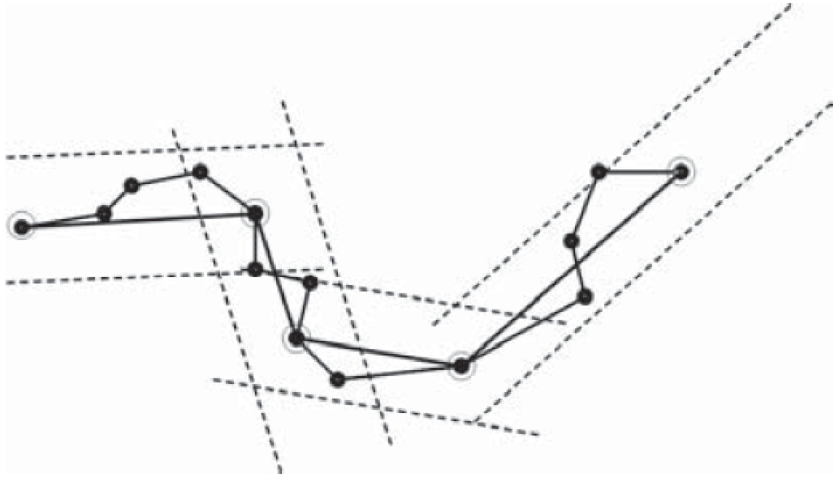


Figure 7.20: Example of reducing lines by the use of corridor. The numbers of points needed to describe a line are reduced by moving forward a corridor of a given width until it touches the digitized line. All the points on the line in the corridor, apart from the first and last, are deleted. This process is repeated until the entire line is trimmed.

STORING VECTOR DATA

- Database storage of cartographic data involves standardization of data through common reference systems and uniform formats.
  - Standardized storage makes the presentation of data compiled from dissimilar sources much easier.
- The selection and use of DBMS are vital in GIS applications because they determine the speed and flexibility with which data may be accessed.
  - The various DBMSs differ primarily in the ways in which the data are organized.
- It is usual to split topological data into different thematic layers to simplify storage and to improve access to data.
  - This division is done so that no overlap occurs between polygons within each thematic layer.
  - The disadvantage of this system is that common lines between objects that are stored in different layers have to be removed several times; This problem can be avoided by using object-based storage.

THE CHOICE BETWEEN SPAGHETTI AND TOPOLOGY MODELS

- The problems related to spaghetti and topology are based on;
  - The assumption that a class of area entities is always a tiling of the plane in which every point lies in exactly one polygon.
  - The assumption of needing to avoid computation for area coverage/layer model and use of topology.
- However, the problems related to spaghetti and topology have changed somewhat during recent years;
  - New GIS software treats polygons as independent objects that may overlap and need not fill the plane, and systems permit shapes.
  - New and more powerful computers eliminate the need for reduction in calculation time.
- Today, topology can easily be built on-the-fly.

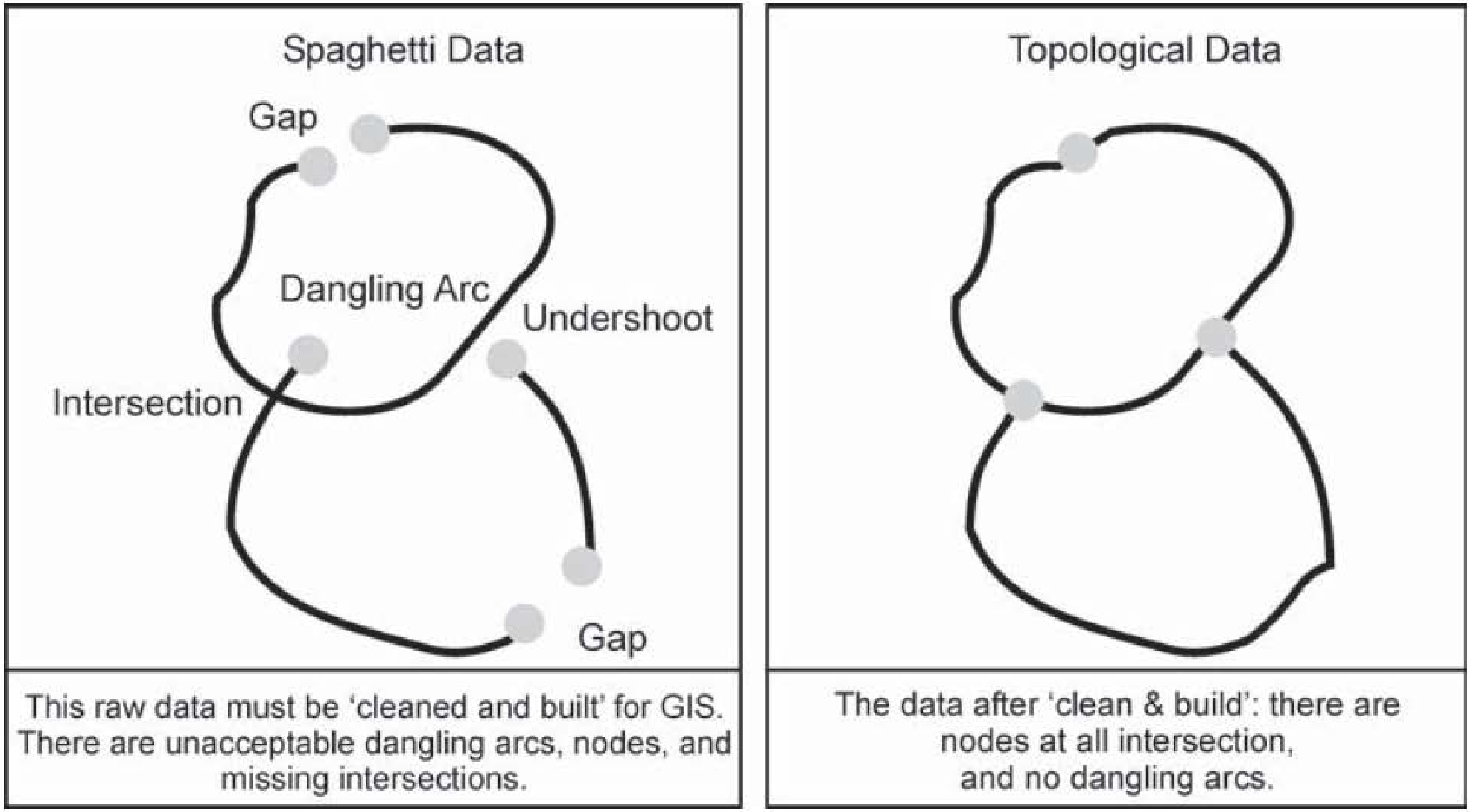


Figure 7.21: Spaghetti data versus topological data.

RASTER DATA MODELS

- Raster data are applied in at least four ways:
  - i. Models describing the real world
  - ii. Digital image scans of existing maps
  - iii. Compiling digital satellite and image data
  - iv. Automatic drawing driven by raster output units

RASTER MODEL

- The raster model is in many ways a mathematical model, as represented by the regular cell pattern (Figure 7.22).

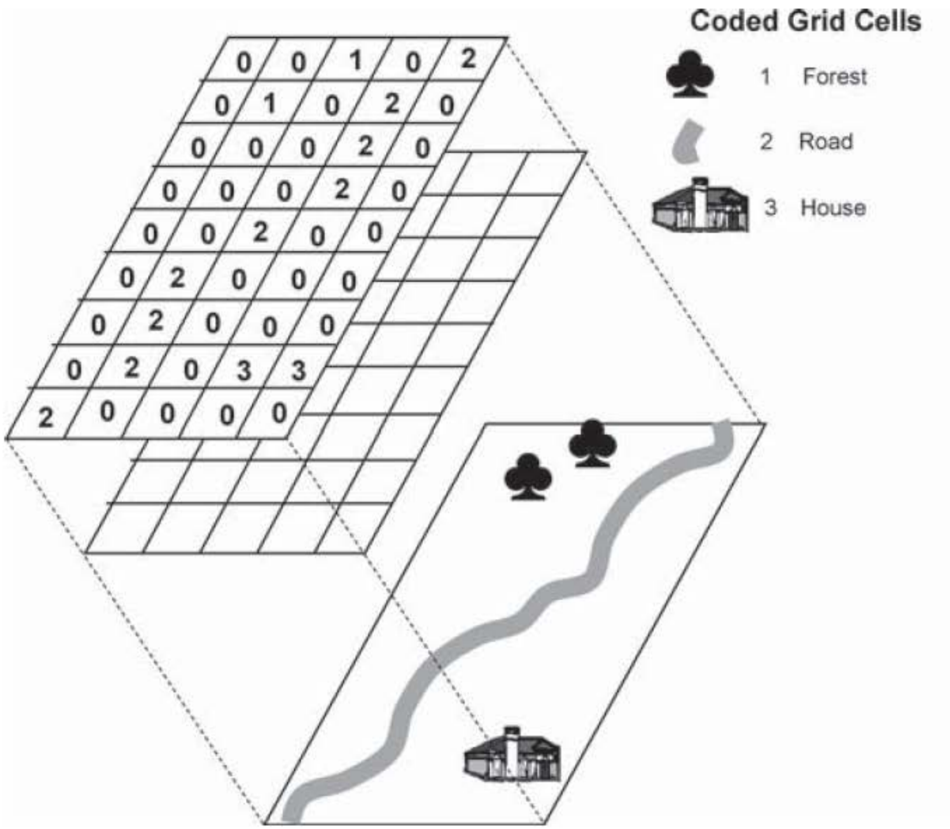
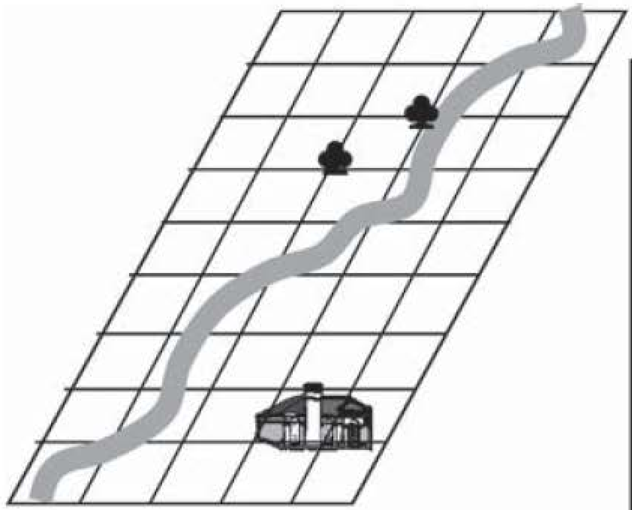


Figure 7.22: Raster data can be visualized as a grid lying over the terrain. Each grid cell has a code stored in the database describing the terrain within that particular cell.

- Because squares are often used and a pictorial view of them resembles a classic grid of squares, it is sometimes called the grid model.
- The cells of a model are given in a sequence determined by a hierarchy of rows and columns in a matrix, with numbering usually starting from the upper left comer (Figure 7.23).
- The cells are often called pixels (picture elements)
  - A pixel is the smallest element of an image that can be processed and displayed individually.



Cell Table

Cell No.	Cell Value
00	0
01	0
02	1
03	0
04	2
10	0
11	1
–	–
–	–
–	–
–	–
70	0
71	2
72	0
73	3
74	4
–	–
–	–

Cell Code

	0	**1**	**2**	**3**	**4**
0	0	0	1	0	2
1	0	1	0	2	0
2	0	0	0	2	0
3	0	0	0	2	0
4	0	0	2	0	0
5	0	2	0	0	0
6	0	2	0	0	0
7	0	2	0	3	3
8	2	0	0	0	0

Code List

0	Unmapped
1	Forest
2	Road
3	House

Figure 7.23: A line number and column number define the cell's position in the raster data. The data are then stored in a table giving the number and attribute value of each cell.

REALIZING THE RASTER MODEL

- Raster models are created by assigning real-world values to pixels (Figure 7.23).
- Cell values may represent numerous phenomena, including:
  - Physical variables, such as precipitation and topography, respectively, with amounts and elevations assigned to the cells
  - Administrative regions, with codes for urban districts, statistical units, and so on
  - Land use, with cell values from a classification system
  - References to tables of information pertaining to the area(s) the cells cover, such as references to attribute tables
  - Distances from a given object
  - Emitted and/or reflected energy as a function of wavelength – satellite data.

- A single cell may be assigned only one value, so dissimilar objects and their values must be assigned to different raster layers, each of which deals with one thematic topic (Figure 7.24).
- Raster models usually have more layers than those in vector models.
  - In a vector model, attributes are assigned directly to objects.
  - In a raster model, attributes are assigned in a second thematic layer.
- In practice, a single cell may cover parts of two or more objects or values.
- Cell locations, defined in terms of rows and columns, may be transformed to rectangular ground coordinates
  - The coordinates of all cell corners and centers can then be computed using the known cell shapes and sizes.

- Raster data are normally stored as a matrix. However, they can also be stored in tabular form, where each individual cell in a raster forms a line in the table.

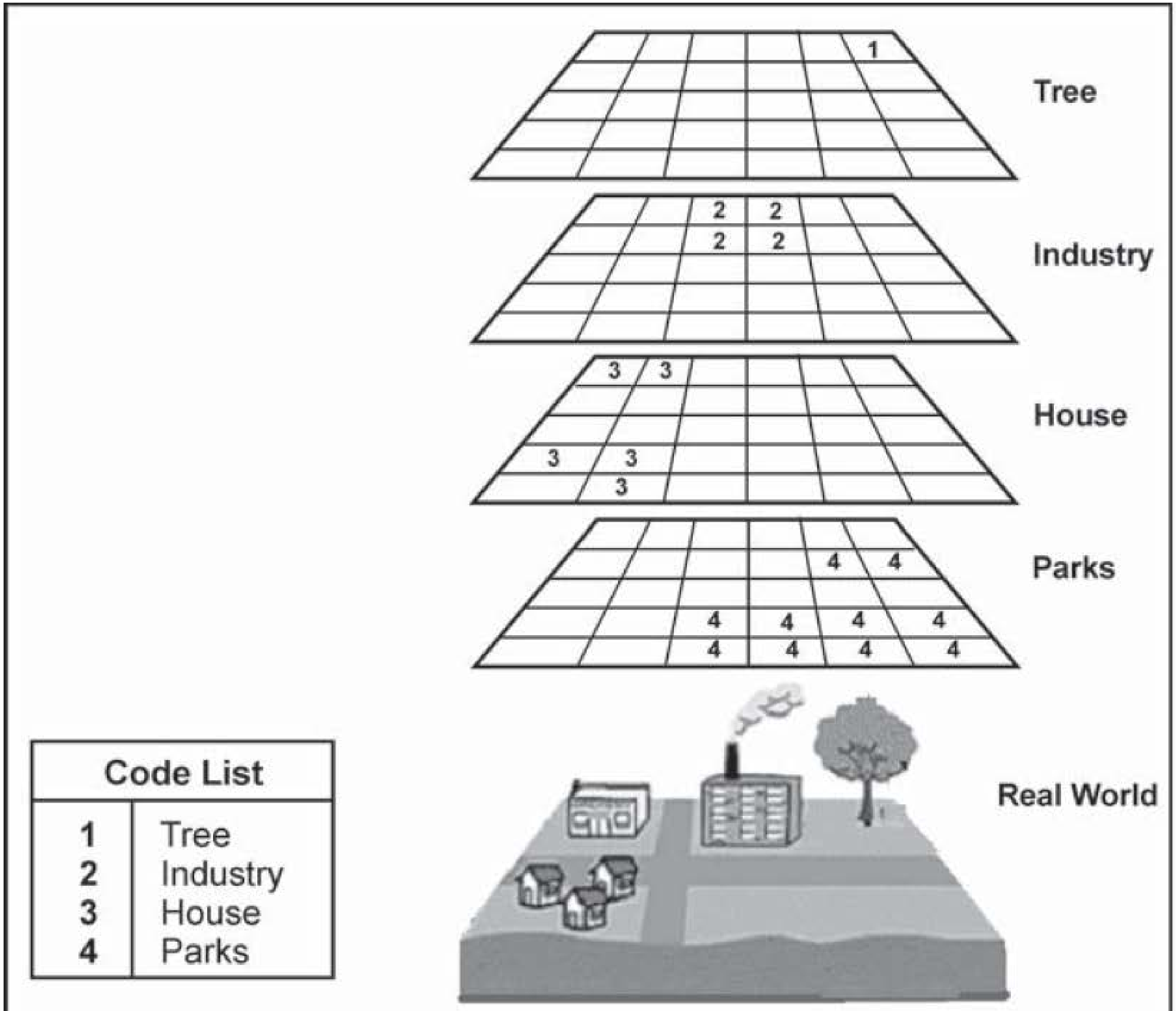


Figure 7.24: Only one attribute value is assigned to each cell. Objects that have several attributes are therefore represented with a number of raster layers, one for each attribute.

STORING RASTER DATA STRUCTURES

- Using a raster GIS we could store a set of spatial data in the form of a grid of pixels.
  - I. Binary
    - A value which indicates the presence or absence of a feature of interest.
    - For example, in a layer representing roads, we might use 1 for pixels which contained part of road, and 0 for pixels which did not.
  - II. Enumeration
    - A value from some classification.
    - For example, a layer representing soils might contain codes representing the different soil types–1 for alluvial, 2 for red soil etc. Since the values are not directly related to the soil type, there would have to be a key of some sort indicating the meaning of each value.

- III. Numerical
  - An integer or floating point number recording the value of a geographical phenomenon.
  - In the soil example, we might have measurements of soil moisture content. A common example of this kind of raster layer is Digital Elevation Model (DEM).

- The raster data model has the great virtue of simplicity but it can produce very large files.
- The precision with which, 1 raster layer can represent spatial data is related to the size of the pixel.
  - we cannot represent anything which is smaller than a pixel.
  - Halving the pixel size would increase the number of pixels by a factor of 4.

RASTER DATA STRUCTURE: THE ARRAY

- The simplest method of storing a raster layer in the memory of the computer is using a data structure called an array.
- We consider alternative methods for searching through a list of entries in a telephone book to find one that matched with a particular name.
  - 1. Array NAMES [1…... 64]
  - 2. Read names from file into names array
  - 3. i = 1
  - 4. FOUND = false
  - 5. repeat until FOUND = = true or i > 64
  - 6. if NAMES [I] = = ‘Sameer’ then FOUND = true
  - 7. i = i + 1

- The first few elements of the array might contain the following:

NAMES [1]	Sameer
NAMES [2]	Shaan
NAMES [3]	Krishna

- So why do we need memory?
- Memory is organized so that every box has an *address*, which is basically a number which uniquely identifies it. So our list of names might look like this in memory (Table 7.18).

Table 7.18: Storage of an array in computer memory.

Address	1295	1296	1297	1298
Contents	Sameer	Shaan	Krishna	

- So when the program refers to NAMES [2] what the computer actually does is as follows:
  - 1. The index value of this element is 2.
  - 2. The first element in this array has an index value of 1.
  - 3. This element is therefore (2–1) = 1 box on from the start.
  - 4. The address of the fi rst element is 1295.
  - 5. The address of this element is therefore 1296.
- To see how this relates to GIS, let us consider the simple raster layer shown in figure 7.25.

3	A	A	A	A
2	A	B	B	B
1	A	A	B	B
0	A	A	A	B
	0	1	2	3

Figure 7.25: An example of simple raster array.

- We still store our array in a sequence of memory locations in exactly the same way as for our list of names, as shown in table 7.19.

Table 7.19: Storage of array in computer memory.

Address	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Value	A	A	A	B	A	A	B	B	A	B	B	B	A	A	A	A

- So when a program refers to a particular pixel, such as IMAGE [2, 3] how does the computer know which memory location to go to?
  - a four by four array would be declared as follows:

    Array IMAGE [0 . . 3, 0. . 3]
  - address = (nrow*rowsize) +ncolumn

    address = (2*4) +3 = 11

- Arrays can be very efficient in terms of processing time.
  - No matter how large the array held in memory, the retrieval of an item from it will take exactly the same amount of time.
- However, they are very inefficient in terms of storage, since every single pixel takes one element of storage.

COMPRESSION OF RASTER DATA

- Various devices may be employed to reduce data volume and, consequently, storage memory requirements.
- Cells of the same value in a row may be compacted by stating the value and their total. This type of compacting, called run – length encoding.
- Further compacting may be achieved by applying the same process recursively to subsequent lines.

SAVING SPACE: THE RUN LENGTH ENCODING AND QUAD-TREES

- The main disadvantage of the array is the size of the files when data is stored.
  - First, smaller files means quicker execution times.
  - Second, the smaller the file size, the more images can be held in memory at one time.
- The simplest strategy for reducing file sizes is to use the smallest possible amount of storage for each pixel.
  - The use of single byte integers is commonly available, and where appropriate will reduce the file size, and hence memory usage by a factor of 4 compared with using 32 bit words.

Table 7.20: Examples of storage of integers in bytes.

Binary	Decimal
00000000	0
00000001	1
11111111	255

- A second strategy for dealing with large files, is to hold only part of the layer in memory at any one time.
  - In order to assess the efficiency of this approach, we have to consider two issues – how much memory will be needed, and how many times will we have to transfer data between memory and disk storage.
  - If we hold the whole array in memory then this uses $O(n^2)$ storage, but only requires 1 read and write operation between the disk and the memory.
    - It is very quick, but uses a lot of memory.
  - At the other extreme, we could read each pixel as we need it and write it back to disk afterwards. This now uses 1 unit of storage but $O(n^2)$ read/write operations.
    - This uses almost no memory but would be painfully slow.
  - A compromise is to read one row at a time into memory, process it and write it out to disk
    - This uses $O(n)$ storage, and also $O(n)$ read/write operations.

- A third strategy to save storage space is to use run length encoding.
  - The raster layers contain large number of pixels with the same value next to one another.

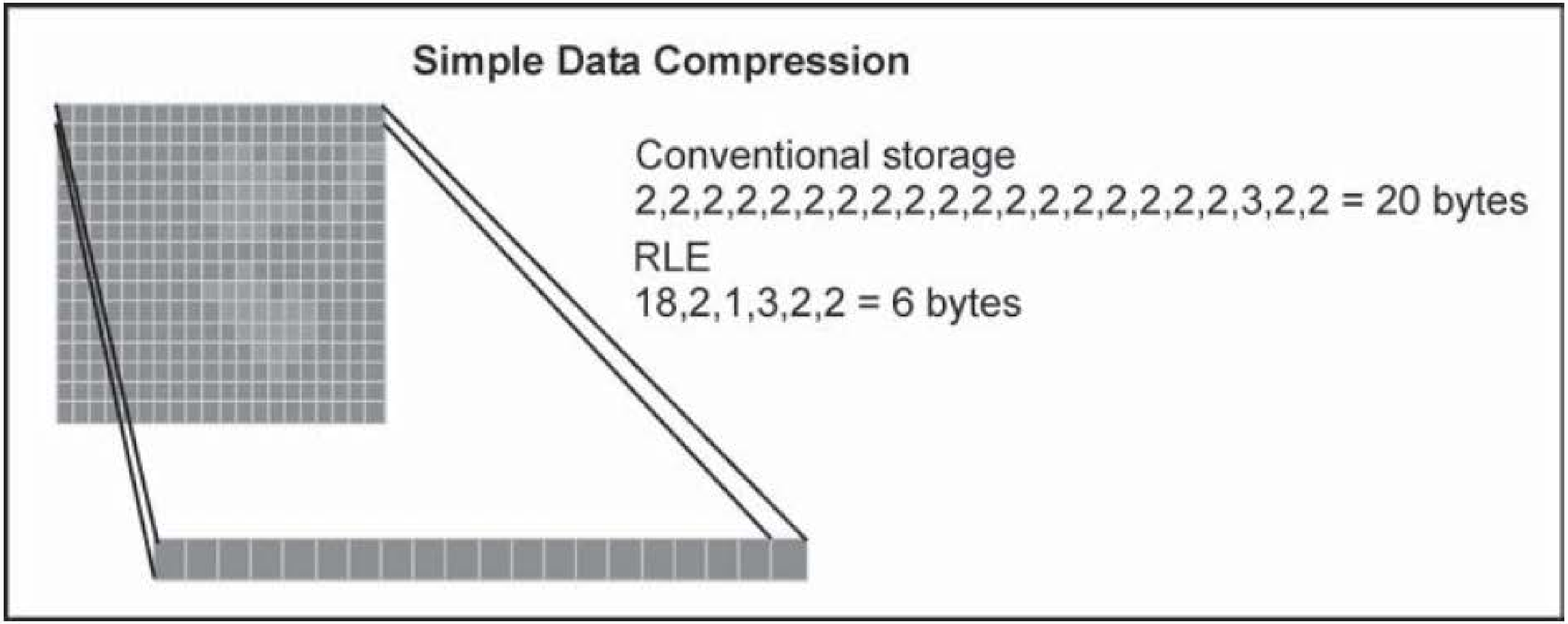


Figure 7.26: Much of the raster consists of areas which contain pixels of same value adjacent to each other.

- Consider the simple raster layer we used earlier, which is repeated as shown in Figure 7.27.

3	A	A	A	A
2	A	B	B	B
1	A	A	B	B
0	A	A	A	B
	0	1	2	3

Figure 7.27: An example of simple raster array.

- Instead of storing each one we can store the information about the run – how long it is and what value the pixels have.

Table 7.21: Storage of a run length encoded layer in computer memory.

Address	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Value	3	A	1	B	2	A	2	B	1	A	3	B	4	A		

- But, In the case of a layer in which every pixel was different from its neighbours, such as a DEM, the file size would double.

- The final raster data structure to save storage is called the quadtree, and it extends the idea of run length encoding to 2D.
  - The basis of the quadtree method is that the pixel size is allowed to vary across the image, so that uniform areas are stored using a few large pixels, but small pixels are used in areas of variation.

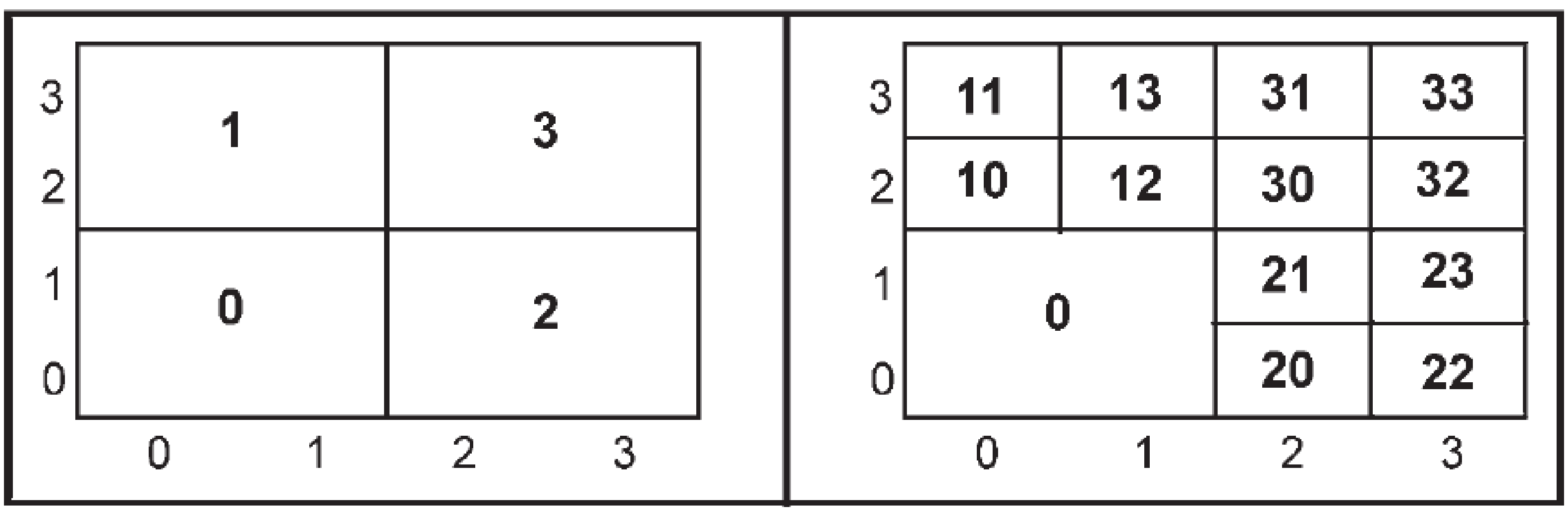


Figure 7.28: Quadtree subdivision of layer shown in figure 7.28.

Table 7.22: Storage of quadtree in memory.

Address	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Quadtree Address	0	1	2	3	10	11	12	13	20	21	22	23	30	31	32	33
Value	A	4	8	12	A	A	B	A	A	B	B	B	B	A	B	A

- So why is it called a quadtree?

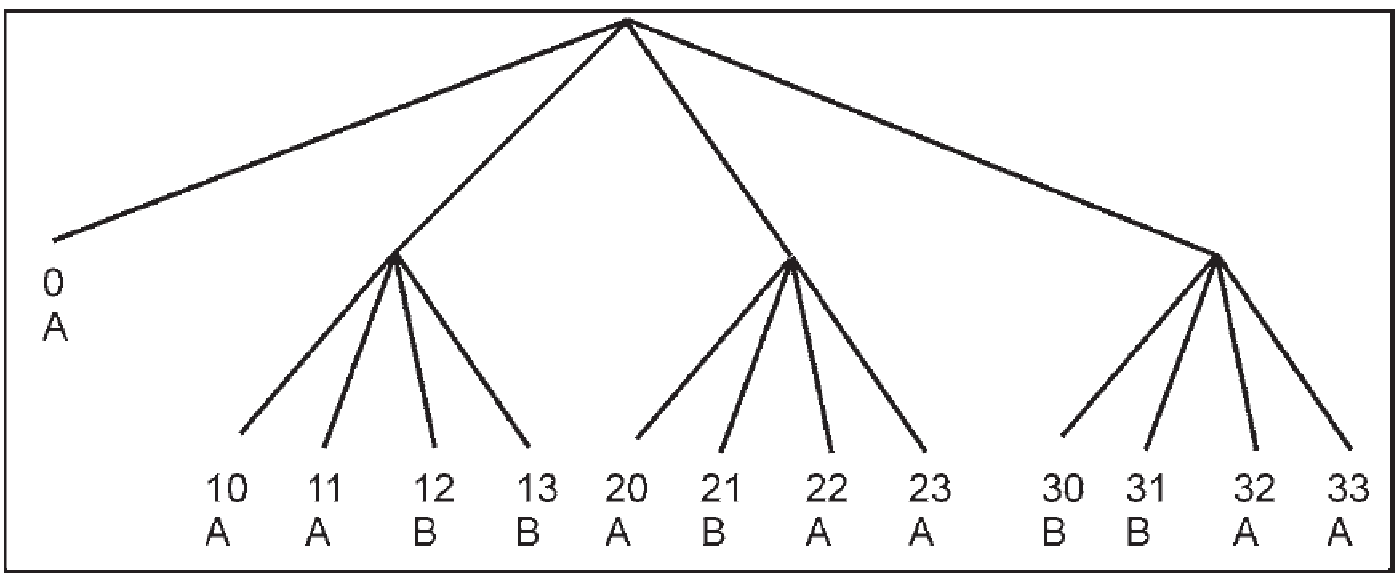


Figure 7.29: Graphical representation of quadtree.

- The advantages of the quad-tree model are
  - Rapid data manipulation because homogeneous areas are not divided into the smallest cells used
  - Rapid search because larger homogeneous areas are located higher up in the point structure
  - Compact storage because homogeneous squares are stored as units
  - Efficient storage structure for certain operations, including searching for neighboring squares or for a square containing a specific point

- The disadvantages of the quad-tree model are
  - Establishing the structure requires considerable processing time.
  - Protracted processing may prolong alterations and updating.
  - Data entered must be relatively homogeneous.
  - Complex data may require more storage capacity than ordinary raster storage.

AUTOMATIC CONVERSION BETWEEN VECTOR AND RASTER MODELS

- Vectorization.
  - Converting raster data to vector data.
  - Areas containing the same cell values are converted to polygons with attribute values equivalent to the pre – conversion cell values (Figure 7.30).
- Rasterization
  - Converting vector data to raster data.
  - Each cell falling within a polygon is assigned a value equal to the polygon attribute value (Figure 7.31).
- Vectorization is the more complex and time consuming of the two processes.
- Different conversion programs can yield differing results from the same set of raster data.
- Normally, some information/data are lost in conversions. Consequently, converted data are less accurate than original data.

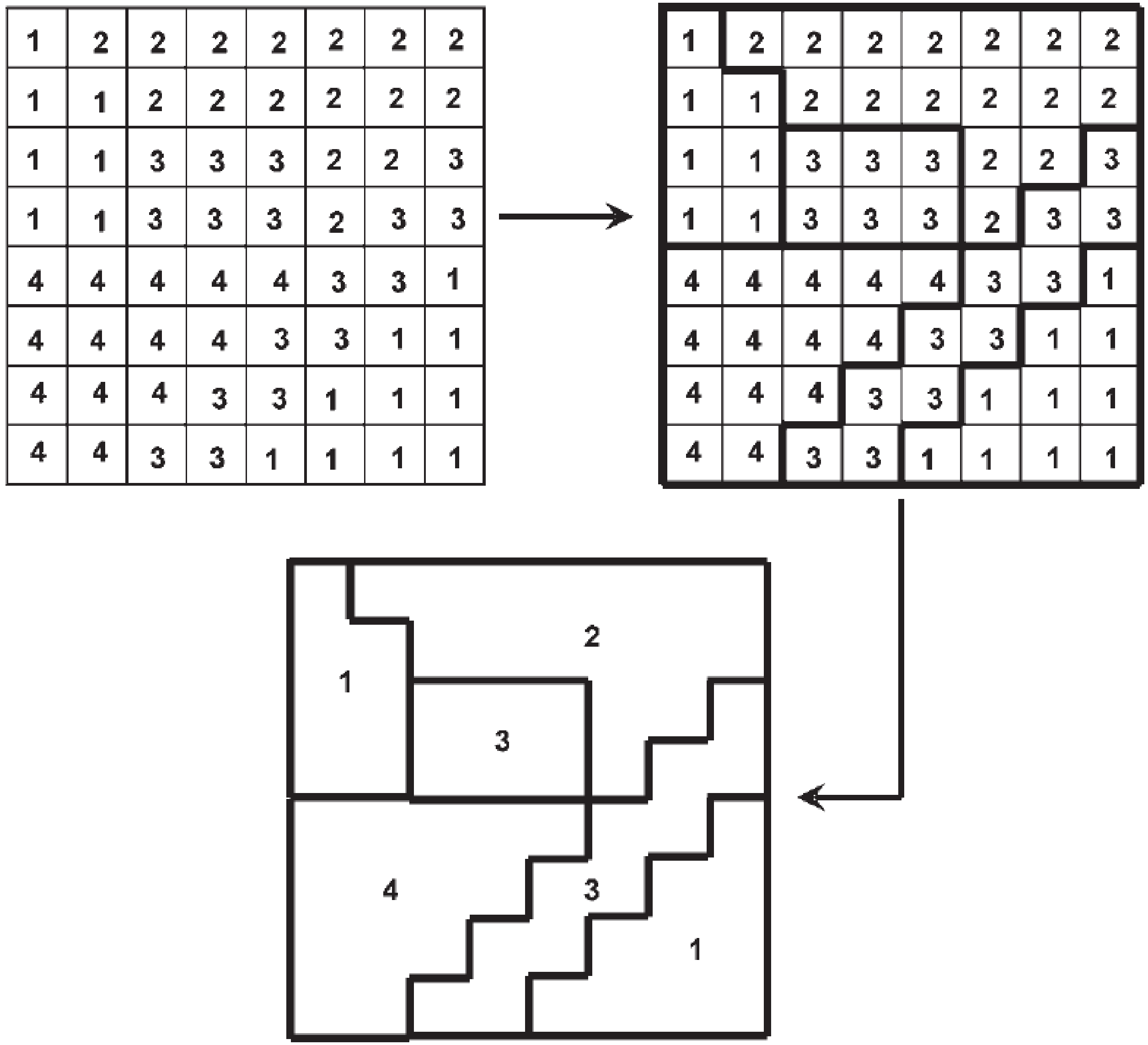


Figure 7.30: Conversion of raster data to vector data; first, each raster cell is assigned an attribute value; secondly, boundaries are set up between different attribute classes and finally, polygons are created by storing X and Y coordinates.

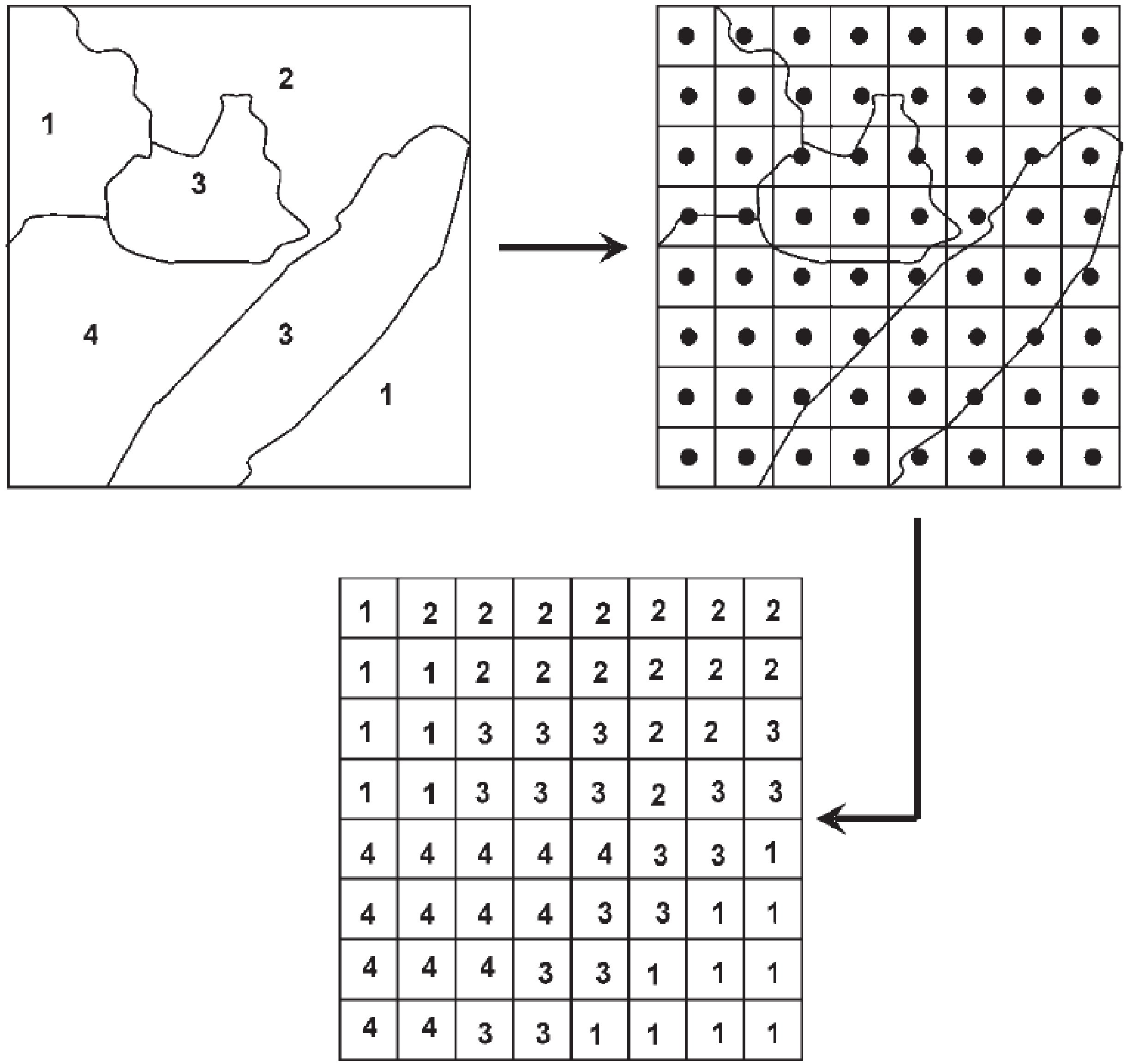


Figure 7.31: Conversion of vector data to raster data; first, polygons are coded; secondly, a grid with the right cell size overlays the polygons; here, the polygons that contain the centre of the individual cells are Identified; finally each cell is assigned the attribute code of the polygon to which it belongs.

VECTOR VERSUS RASTER MODELS

- Raster model
  - Raster data are more adept at showing the geographical variation of phenomena.
  - Raster data are superior for area presentations.
  - The raster model has been used more frequently in natural resource planning and management.
  - Raster GIS emphasizes properties: here, the basic units of observation are regular cells in a raster.
  - The accessibility of raster data may be a major problem and perhaps the greatest drawback of a raster GIS in comparison with a vector GIS.
- Vector model
  - Vector data may be considered best suited for documentation.
  - Vector data are preferable for line presentations.
  - To date, the vector model has been dominant in commercial GIS implementations.

- Many newer GIS can manipulate both vector and raster models. With dual capability, a GIS can exploit the respective advantages of both.

ATTRIBUTE DATA AND COMPUTER REGISTERS

- With the advantages of easy updating, rapid search, and the flexible superimposition of data, the computerized filing of information has become commonplace in administrative work.
- Computerization permits a simplification and coordination of registration efforts and can eliminate duplication and rationalize the overall filing process.
- In the public sector, central registers have been established as a common resource for numerous users.
- In many countries, work is under way to make public registers available to GIS users.

- Upon entry, register data are selected (structured) so that registers contain uniform and limited data.
- As for digital map data, register data are stored using formats.
- There is no general pattern for register content, but usually the items registered will have identities, locations, descriptive details, time and date notations, and sometimes references to other registers.

Table 7.23: Geometric content is often limited to identifications geometry/coordinates and topology, while attribute content often comprises location (address), various descriptions of the object and timing.

Georeferencing	Attributes
ID	ID
Geometry/	Location
Coordinates	Description
Topology	Time period

CODING AND ENTERING ATTRIBUTE DATA

- Attribute data may be coded for several reasons in order to:
  - Establish an ID code between geometry and attributes
  - Conserve computer memory
  - Ease input work
  - Ease verification of data entered
  - Simplify subsequent searches for data in databases
- Coding of attribute data often includes data structuring.
  - Codes are often assigned according to a hierarchical classification system devised to ease data operations.
  - Examples include the official codes widely used for addresses, names of towns, highways, and so on.
- The type of data may be specified for each field, such as integer (land-use code), decimal (area), and text (name).
- Code tables may be compiled and used with the main table to produce more meaningful printouts from the system.

- ID codes are usually entered together with the attribute data.
  - They may also be registered or edited into compilations of attribute data which initially have no codes.

STORING ATTRIBUTE DATA

- Attribute data are usually most easily and expediently stored in tabular form.
  - Each line in a table represents an object, each column an attribute.
- Attribute data are therefore often called tabular data and are normally stored in a relational database.
- Data on different types of objects are usually stored in separate tables, each dedicated to a single object type.
- In each table, line formats and lengths are identical throughout.
- Table content must be relevant to the objects, so each object or line must have a stable identity or access key.
- Data available in existing computerized registers are not always in convenient tabular form.
  - Conversions and round about methods must often be used to access such data for GIS uses.

LINKING DIGITAL MAP AND REGISTER INFORMATION

- Common identifiers in map data and attribute data permit moving from map data to attribute data, and vice versa.
- As illustrated in Table 7.24, this is possible if the attribute data that lack georeferencing have access keys in common with attribute data that have other access keys in common with map data.
- In some instances, the geometry can be stored directly together, with the attribute data linked to each register object.
  - For example, a building register with coordinates representing each building

Table 7.24: Data elements in one data set can be used as an access key to another data set, thereby acting as a link between other data.

Digital Map Database

Point no.	Theme code	Building	X Coordinate	Y Coordinate
2971	2.1	Nishat	10211300	527330

Property Register

Building	Plot no.	Owner	Year of construction	Use
Nishat	11/1	Shaad	1998	Residential

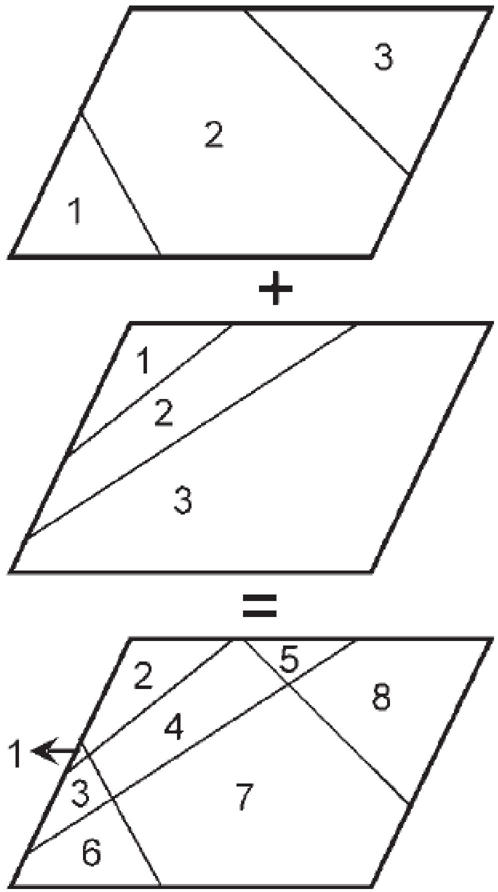
Land Register

Plot no.	Land owner	Occupation	Area	Address
11/1	Fawwad	Business	10000 sq. m	Shamshad Market

Personal Register

Land owner	Age	Education	Sex	Address
Fawwad	35 Yrs.	Graduate	Male	859/6 Azad Enclave

- Map data may be used not just to link maps and attributes, but also geographically to link dissimilar attributes.
  - Superimposing dissimilar data can be solved by using cartographic integration, in which overlay techniques are used to combine geometry from two dissimilar thematic maps into a single synthesized map.
  - The synthesized map contains numerous new objects and areas, all of which are related to the two original thematic maps.
  - the objects in the synthesized map comprise the least common units between the original maps and are therefore called *integrated terrain units* (ITUs; Figure 7.32).
  - The ITU is also called a *basic spatial unit* (BSU) and defined as a fundamental area unit which has homogeneous properties in the context of a particular subject.



ID	Attribute
•	Cultivated
•	Grazing
•	Forest

+

ID	Attribute
•	Development
•	Recreation
•	Preservation

=

ID		Attribute		Attribute
1	1	Cultivated	1	Development
2	2	Grazing	1	Development
3	1	Cultivated	2	Recreation
4	2	Grazing	2	Recreation
5	3	Forest	2	Recreation
6	1	Cultivated	3	Preservation
7	2	Grazing	3	Preservation
8	3	Forest	3	Preservation

Figure 7.32: Attribute data can be made comparable by superimposing geometry from dissimilar geographical units to get the integrated data.