CHAPTER TWO

The spatial database

"When considering any space—a room, a landscape, or a continent—we may adopt several fundamentally different ways to describe what is going on in that subset of the earth's surface." (Burrough and McDonnell 1998: 20)

So far, we have discussed GIS in a very broad sense. From here onwards, we begin to look in more detail at how precisely the GIS works. As mentioned in Chapter 1, central to any archaeological application of GIS is the *spatial database*. We use the term to define the entirety of information we have held in the GIS for a certain study area. Any given spatial database must provide for the storage and manipulation of four aspects of its data:

- A record of the position, in geographic space (the *locational component*) that determines where something is and what form it takes;
- A record of the logical relationships between different geographic objects (the *topological component)*;
- A record of the characteristics of things (the *attribute component*) that determines what geographic objects represent, and what properties they have;
- Thorough documentation of the contents of the overall database (the *metadata component*).

2.1 HOW DOES A SPATIAL DATABASE DIFFER FROM A TRADITIONAL DATABASE?

Traditional (non-spatial) databases are concerned only with the attributes of objects. As a result they make no explicit distinction between the location of an object (*e.g.* an archaeological site) and its other attributes (date range, finds, structural evidence *etc.*). An example would be the common practice of storing a grid reference for each site in an inventory. This distinction is a crucial one and can be illustrated with a hypothetical example. Consider the feature complex shown in Figure 2.1. Here, we have an Iron Age Hillfort comprising two concentric ramparts. Inside the outer rampart, but overlain by the inner is an earlier Neolithic bank barrow. Abutting the Hillfort are the remains of a prehistoric field system. Although fictional, in the UK this is not a particularly artificial or unusual configuration of archaeological remains.

In a traditional monuments inventory or register based on a conventional database, there would be individual entries for the Hillfort, the barrow and the field system, in each case giving some details about the physical remains, the legal status, bibliographic details, history of investigation and so on. At a minimum, the database would also give a map reference for each monument. It is not uncommon for features such as the field system to have several different entries, referring to a number of discrete locations spaced at regular intervals throughout the overall

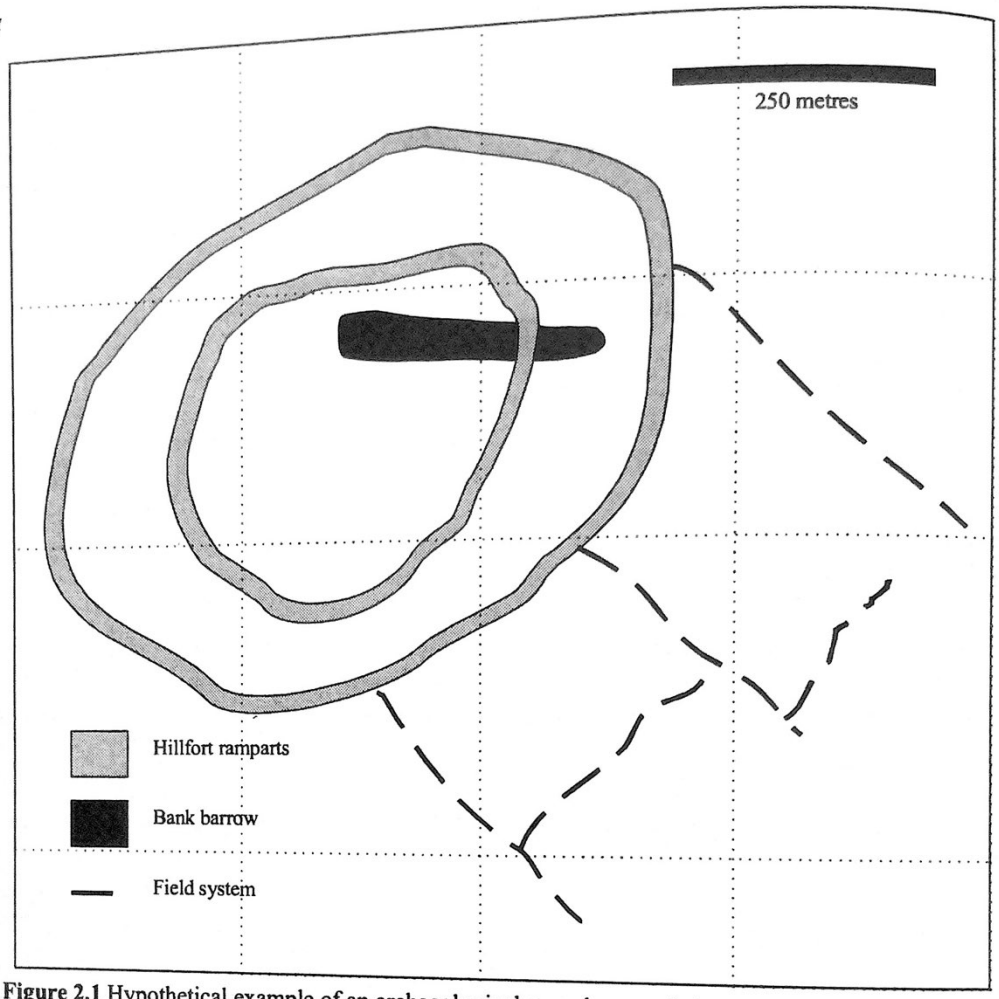


Figure 2.1 Hypothetical example of an archaeological complex consisting of a Hillfort, bank barrow and adjoining field system.

complex. Now consider the following enquiries, which might not unreasonably be made by an archaeologist, resource manager or planner.

- *What is the extent, in hectares, of the hillfort?*
- *Is the bank barrow wholly enclosed by the hillfort?*
- *How do the orientations of barrow and field system compare?*
- *What is the relationship between the field system and the hillfort?*
- *How many other monuments are partly or wholly within 1km of the barrow?*
- *How many other monuments are visible from the hillfort?*

Whatever the query facilities available within the monuments register, it is clear that without reference to maps or more detailed textual descriptions of the sites no traditional database would be able to answer these questions. To answer the first requires a geographic representation of the form of the site—the spatial component—while the remainder require that the database has some representation of the logical relationships between the geographic forms of the features

themselves or information derived from them (*e.g.* they are in view of the zone of land within 1km of the Hillfort).

The situation becomes even more complex when an archive or inventory is required to hold aerial photographs of the sites or to retain a record of fieldwalking data or geophysical survey. Each of these kinds of data requires an approach to storage that is explicitly based on geographic location.

In the following sections we will concentrate on the first two aspects of the spatial database highlighted in the introduction: the locational component (how the GIS records and represents the positions of features in space); and the topological component (how it encodes and interprets the geometrical relationships between features). Attribute and metadata will be discussed in Chapter 3.

2.2 THEMATIC MAPPING AND GEOREFERENCING

Let us begin by looking at how the spatial database organises and manages spatial information. Rather than storing it in the form of a traditional map, GIS store spatial information *thematically*. This can best be illustrated with reference to a typical map sheet. Any given map contains a wealth of information relating to a host of specific themes. These might be: topography (in the form of contours); hydrology (rivers and streams); communications (roads, tracks and pathways); landuse (woods, houses, industrial areas); boundaries (*e.g.* political or administrative zones) not to mention archaeology (the point locations of archaeological sites). Although we think of maps as a highly familiar way of representing the spatial arrangement of the world, closer perusal reveals the everyday map to be an abstracted and deceptively complex entity encoding an enormous breadth of information. Returning to the GIS, an important point to realise is that GIS systems *do not* conceptualise, store and manage spatial information in such a complex and holistic form. As a result, although it is often convenient to talk of the discrete collections of spatial information held within the GIS as 'GIS-maps' it is formally incorrect.

Instead, GIS rely upon the concept of 'thematic mapping'. Instead of storing a single, complex, multiply-themed map sheet, GIS store and manage a collection of individual sheets, each themed to a particular facet of the region under study. The precise terminology used for these discrete slices of thematic data varies, with some systems using the term *layer* and others *theme*, *coverage* or *image*. To avoid confusion the term *layer* will be used exclusively in this book.

In the example of the hypothetical map mentioned earlier, we would no longer have a single map but instead a group of themed layers: topography (holding just the contours); hydrology (the network of rivers and streams); communications (roads and paths); landuse (the areas of woods, houses and industrial activity); boundaries (the administrative zones) and archaeology (the point locations of archaeological sites). Some of these layers are depicted in Figure 2.2.

The important point to emphasise is that a spatial database and a traditional map sheet can contain the same information, but are structured in a very different way: a set of specific thematic layers as opposed to a complex totality. Let us look at some archaeological examples. In their pioneering work on the Dalmatian island of Hvar, Gaffney and Stancic constructed their spatial database using five basic

layers of information: topography; soils; lithology; microclimate and archaeological site locations (Gaffney and Stancic 1991:36). In a recent assessment of archaeological resource sensitivity in Pennsylvania and West Virginia, Duncan and Beckman based their spatial database around the following layers: Prehistoric site location; historically documented Indian trails; roads and disturbance factors; hydrology; soils; and topography (Duncan and Beckman 2000, see Table 2.1).

The use of thematic layers makes a lot of conceptual sense, particularly in the management and organisation of data. For the concept to work we have to be able to combine and overlay the individual layers in our database whilst maintaining the original spatial relationships held between the component features. For example, if on the original map sheet an archaeological site fell 23m from a river and sat exactly on the 92.5m contour, then if we were to ask the GIS to overlay the archaeology, hydrology and topography layers in the spatial database, the same relationship should hold. To ensure that this is so, at the heart of the spatial database is a mechanism whereby every location in each data layer can be matched to a location on the earth's surface, and hence to information about the same location in all the other data layers. This makes it possible for the user of the GIS to identify the absolute location of, for example, a pottery density held in a fieldwalking theme, the limits of a protected archaeological site, a river in a hydrology layer and any other archaeological or non-archaeological information within the overall spatial database.

This registering of the spatial locations of features in the individual thematic layers to the surface of the earth relies upon a concept we term *georeferencing*. The term *georeferenced* refers to data that have been located in geographic space, their positions being defined by a specified co-ordinate system.

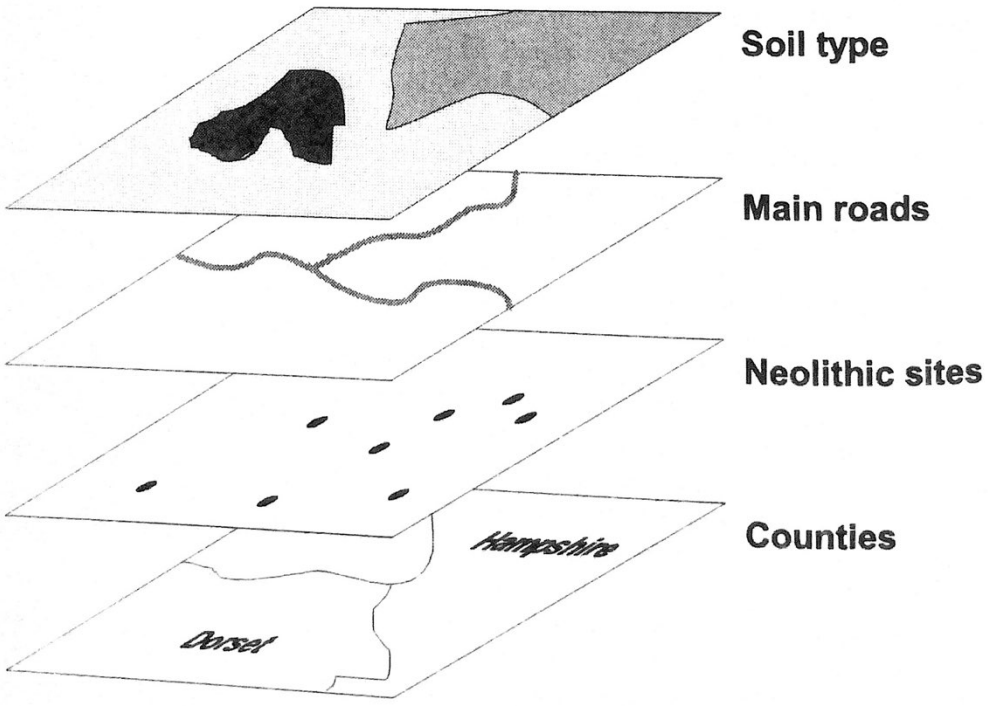


Figure 2.2 Thematic layers.

Table 2.1 Examples of primary thematic layers (after Gaffney and Stancic 1991:36; Duncan and Beckman 2000:36).

Layers	Gaffney and Stancic	Duncan and Beckman
Layer 1	Topography	Prehistoric site locations
Layer 2	Soils	Historically documented Indian trails
Layer 3	Lithology	Roads and disturbance factors
Layer 4	Microclimate	Hydrology
Layer 5	Archaeological site-locations	Soils
Layer 6		Topography

This can be illustrated with reference to the typical process of archaeological survey. Commonly a baseline or site grid is established and a series of measurements are taken with respect to it. This could be to record the locations of finds in an excavation trench or capture the shape of an earthwork feature. Once recorded, the measurements are then used to produce a scaled drawing of the features ready for interpretation and publication. Although the measurements and subsequent drawing are spatially accurate with respect to the trench location or morphology of the features, we have no idea where they are in relation to the wider world. The only spatial reference we have has been internal to the survey itself — the survey grid or baseline. We could not compare the location, size and extent of the earthwork to any other features in the area, as we have no way of relating their relative positions. Such surveys are called *floating* or *divorced* surveys for this very reason. In GIS-based studies, georeferencing is a vital part of any spatial database. This is because it is through georeferencing that data encoded on separate thematic layers can be combined and analysed. It is also vital for the interpretation of the results of GIS-based analyses because it allows the products of GIS to be related to objects and locations in the real world.

So far we have established that the term georeferencing refers to the location of a thematic layer in space, as defined by a known co-ordinate referencing system. Two types of co-ordinate system are currently in general use. The eldest of these is the geographical co-ordinate system, based upon measurements of latitude and longitude. This is the primary locational reference system for the earth, and measures the position of an object on the earth's surface by determining latitude (the North-South angular distance from the equator) and longitude (the East-West angular distance from the prime meridian—a line running between the poles and passing through Greenwich observatory near London). In this system, all points on the earth falling at the same latitude are called a parallel, whilst all points falling at the same longitude are said to be on a meridian.

The other type of co-ordinate system in widespread use is the rectangular or planar co-ordinate system. Planar co-ordinates are used to reference locations not upon the curving surface of the earth but instead upon flat maps. Modern plane co-ordinate systems have evolved from the concept of Cartesian co-ordinates, a mathematical construct defined by an origin and a unit of distance. In a plane, two axes are established running through the origin and spaced so that they run

perpendicular to each other. These are subdivided into units of the specified distance from the origin. Good examples are the international UTM grid system and the Ordnance Survey grid covering the United Kingdom (Gillings *et al.* 1998:14-15). If all of the spatial features of interest, recorded on all of our component thematic layers are referenced using the same co-ordinate system, then they can be overlain and combined maintaining the accuracy and integrity of the spatial relationships encoded upon the source map.

The establishment of a co-ordinate system in turn relies upon a process of what is termed *projection*. Projection refers to the mathematical transformation that is used to translate the irregular curved form of the earth's surface to the two-dimensional co-ordinate system of a map. Such a translation cannot be undertaken without some compromise in properties such as area, shape, distance and compass bearing. Needless to say, the precise projection used has an important influence upon the properties of the map derived from that process. As a result, understanding which projection has been used in the creation of a map is an essential first step in incorporating it into a spatial database.

For very small study areas, it is sometimes acceptable to ignore projection, and to assume that the region of interest does correspond to a flat two-dimensional surface. Most geophysical survey software takes this approach because surveys are rarely larger than a few hectares. However, if the study region is larger than a few kilometres, or if information is to be included from maps which have been constructed with different projections, then to be able to georeference the thematic layers held within the overall spatial database, the GIS needs to understand the projection used for each layer in order to avoid inaccuracies.

2.3 PROJECTION SYSTEMS

The mathematics of projection are beyond the scope of this text, but it is worth considering briefly the major types (or families) of projection in order to understand the essence of the process. Map projections consist of two main stages, which we will examine in turn.

Firstly, the surface of the earth is estimated through the use of a geometric description called an ellipsoid. This is often referred to as a spheroid, though such a designation is not always formally correct (Gillings *et al.* 1998:13). Unlike a circle, which boasts a constant distance from its centre to any point on its edge, in an ellipsoid the distance is not constant. As a result, whereas we define circles on the basis of their radius and diameter, ellipsoids are defined in terms of their equatorial radius (what is formally termed the semimajor axis of the ellipse) and by another parameter, such as the flattening, reciprocal flattening or eccentricity. Another common term applied to geometric figures used to estimate the earth's form is *geoid*. Most of the ellipsoids that have been used to generate maps have names such as the '1849 Airy' ellipsoid used by the Ordnance Survey of Great Britain or the 'Clarke 1866' ellipsoid used with the UTM system in the United States. Fortunately, most GIS provide a list of known ellipsoids that can be selected by the user when data is entered.

Having approximated the shape of the earth, the second stage in the process is to project the surface of the ellipsoid onto a flat surface. There are a huge and varied number of methods for undertaking this process of projection. As intimated

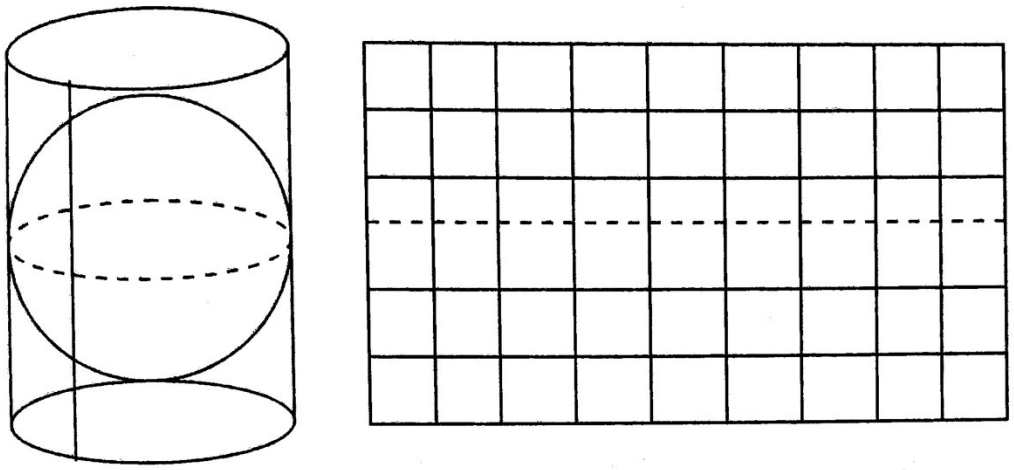


Figure 2.3 Cylindrical projection and graticule.

earlier, each method produces a map with different properties. For example in a *cylindrical projection*, the flat surface is wrapped around the ellipsoid to form a cylinder. An imaginary light source is then placed at the centre of the ellipsoid and used to project the lines of latitude and longitude on the surface of the enclosing cylinder. The lines of latitude (parallels) of the selected spheroid are then drawn as simple straight, parallel lines.

As can be seen in Figure 2.3, the lengths of the lines of longitude (parallels) are progressively exaggerated towards the poles. To maintain the right-angled intersections of the lines of latitude and longitude, the lines of longitude (meridians) are also drawn as parallel lines which means that the meridians never meet at the poles, as they do in reality, so that the polar points of real space become lines the length of the equatorial diameter of the ellipsoid in projection space. The cylindrical projection thus maintains the correct length of the meridians at the expense of areas close to the poles that become greatly exaggerated in an east-west direction. A *transverse cylindrical projection* is created in the same manner, but the cylinder is rotated with respect to the parallels and is then defined by the meridian at which the cylinder touches the ellipsoid rather than the parallel. Reducing the distance between the lines of latitude as they move away from the poles can offset the exaggeration of the areas at the polar regions. This is called a *cylindrical equal-area* projection (Figure 2.4). However, this projection distorts the shapes of regions very badly, and cannot represent the poles at all because they are projected into infinitely small areas.

One other cylindrical projection is worth mentioning. This is the *Mercator* projection, which exaggerates the distance between meridians by the same amount as the parallels are exaggerated in order to obtain what is termed an *orthomorphic* and *conformal* projection. The term orthomorphic describes the fact that shape is preserved so that squares in the projection plane are truly squares. Conformal means that the parallels and meridians in the projection plane always intersect at right angles. The poles cannot be shown because they are infinitely distant from the equator, and the projection exaggerates the apparent areas of northerly areas far more than either of the previous projections—at 60 degrees latitude this exaggeration is 4 times, at 70 degrees 9 times and at 80 degrees it becomes an exaggeration of 32 times the same area if it were shown at the equator. A

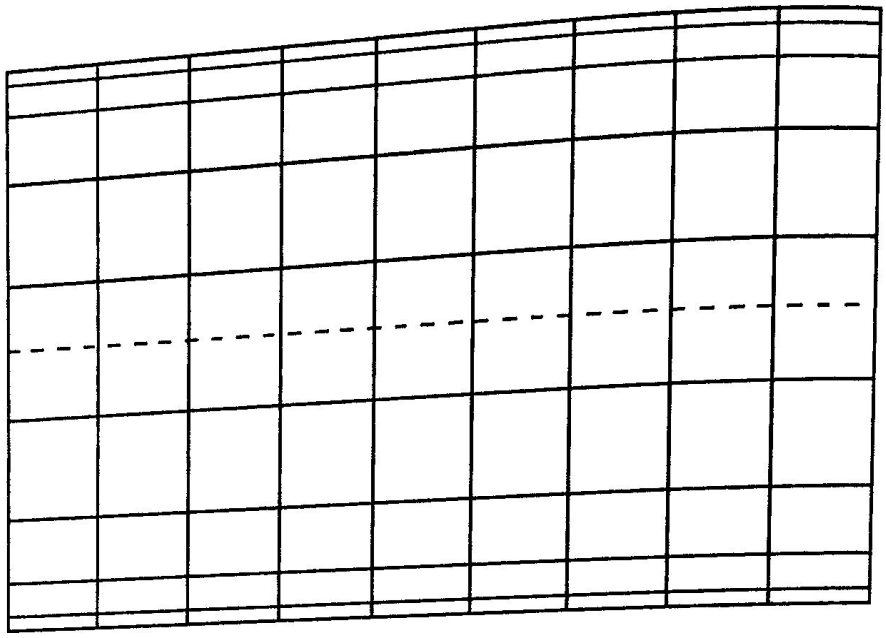


Figure 2.4 Cylindrical, equal-area projection graticule.

transverse Mercator projection is similar except that it is based on the transverse cylindrical projection. This means that instead of the north-south exaggeration of the traditional Mercator projection, the distance of areas to the east and west of the defining meridian are exaggerated in the projection plane.

Cylindrical projections serve to illustrate some of the distortions that are introduced into the representation of space as a result of the projection process, but practitioners should be aware that there are a great many other forms of projection that are not based on the cylinder. Conical projections (such as the *Lambert Conformal Conic*) are based on a cone which intersects the spheroid at one or more locations. Where the point of the cone is above one of the poles, then it intersects at one or more parallels (termed standard parallels). As can be seen in Figure 2.5, conical projections depict the parallels as curved lines, which decrease in size as they approach the poles. The meridians are shown as straight lines that meet at the poles. In other projections, the meridians are curved. A cylindrical projection where all of the parallels are correctly drawn and divided, for example, results in an

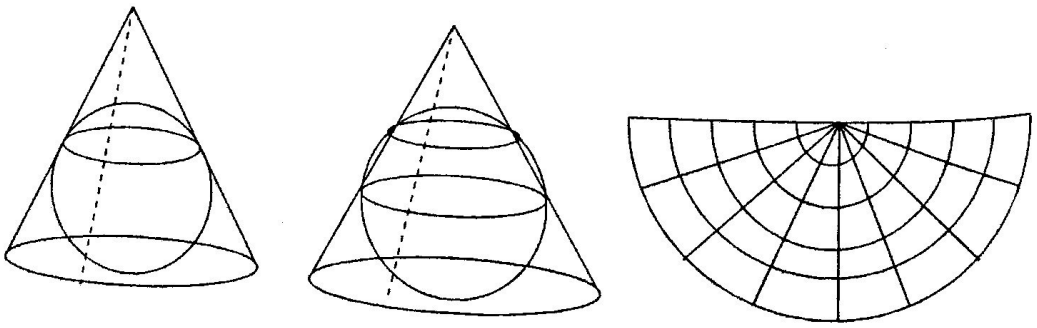


Figure 2.5 Conical projection with one (left) and two standard parallels (middle) and the general form of the resulting projection (right).

equal area projection plane with curved meridians called *Bonne's Projection*. Two entirely separate families of projections called *two-world equal area projections* and *zenithal projections* are also occasionally used for particular purposes.

2.4 FURTHER COMPLICATIONS

To further complicate matters, it is not uncommon for a basic projection to be modified through the use of correction factors. For example, in transverse Mercator projections it is not uncommon for a scaling factor to be incorporated into the east-west axis, which serves to correct for the east-west distortion inherent in the projection itself. It is also common to use a false origin: in other words to arbitrarily define some x, y location on the projection plane to act as the origin point 0,0. If the true origin of the projection is far from the actual mapped area, the false origin can be placed on the positive x and y axes. This makes the axis numbers far shorter and easier to use. Where the origin of the projection occurs within the mapped area, it is not uncommon for a false origin to be defined on the negative side of the x and y projection plane, to ensure that all of the co-ordinates quoted are positive. False origins are used by both the UK Ordnance survey and UTM co-ordinate systems.

The most important point to understand is that all projections result in spatial distortion and different projection systems distort in different ways. If we are aiming to integrate spatial data from a variety of sources into a single GIS-based spatial database, the GIS must be aware of the projections used in their initial creation. Different parameters will need to be given to the GIS depending upon exactly which projection has been used.

To recap: the GIS stores, manages and manipulates spatial data in the form of thematic layers. For the thematic-layer structure to function, the component themes must be georeferenced, ideally to the same co-ordinate system. To be able to integrate spatial data from a range of map-based sources within a single spatial database care must be taken to ensure that they either derive from the same projection or that the GIS holds and understands the relevant georeferencing and projection information about each individual data layer. This is in order to determine how different primary layers which might be represented at different scales or with different projections, relate together in space.

It is also necessary in order for the GIS to be able to undertake calculations of spatial parameters such as distance and area. In some instances it is possible to assume that the layers can be manipulated by using simple Euclidean geometry, either ignoring effects of the curvature of the earth as acceptable errors, or accepting that the selected projection plane has appropriate properties. However, where the locations of entities are recorded in different co-ordinate systems it is necessary for the system to store both the co-ordinates of the entities and the projection through which they are defined. If this is not undertaken, and the characteristics of the projection plane used are not incorporated into the information system, errors in distance and area calculations are introduced.

We can think of the projection information related to a given layer as being critical for us to be able to effectively integrate and use it. Another term for describing such essential data-about-data is *metadata*. We will be looking in more detail at metadata in Chapter 3.

PROJECTION EXAMPLE: THE UK NATIONAL GRID

To understand how projection relates to the use of maps in a GIS, we really need to consider a familiar mapped space. For the authors, the most familiar maps are those made of the United Kingdom by the Ordnance Survey of Great Britain. Superficially, these seem straightforward enough: we can simply digitise them (leaving aside the issue of copyright for a moment) and assume that the shapes and areas will be those of a flat plane. However, the reality is a little more complex than that.

In the United Kingdom, a *transverse Mercator* projection has been used as the basis for all recent maps. Transverse Mercator projections are particularly suitable for areas with a north-south extent as the north-south distortion of areas is removed in favour of an east-west distortion. The UK's projection uses the 1849 *Airy spheroid*, which estimates the earth as an ellipsoid with equatorial radius of 6377563.396 metres and a polar radius of 6356256.910 metres and the projection is made from the 2-degree (west) meridian.

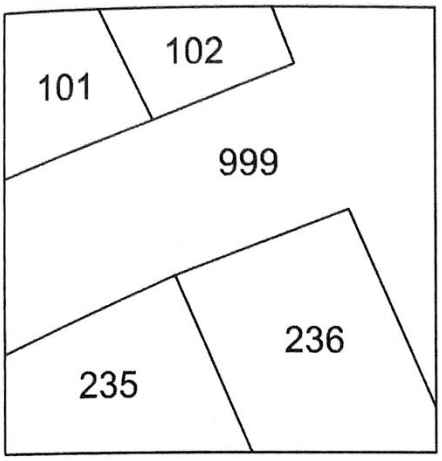
The national grid is defined in this projected space by defining a false origin at crossing of the 2 degree (west) meridian and the 49 degree (north) parallel, in which x is 400000 metres and y is -100000 metres. A scaling factor of 0.996 is also used.

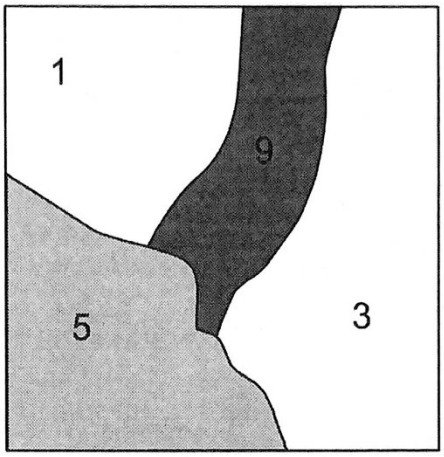
For small areas, any errors of shape and area will be minimal, but if we are concerned with a significant area of or even the whole of the UK then we should represent all of our data in a geodetic (longitude, latitude) system and then instruct the GIS how to translate between them. As you may now appreciate, this is not necessarily straightforward.

2.5 SPATIAL DATA MODELS AND DATA STRUCTURES

The precise way in which a given GIS conceptualises, stores and manipulates spatial information is referred to as its *spatial data model*. Rather than a single, generic GIS, there are currently two types in common archaeological usage, which differ in the precise spatial data model they implement. These two basic types are termed *Vector*-GIS and *Raster*-GIS. The essential difference is that vector data is a formal *description* of something in the real world, usually in the form of geometric shapes. Raster data on the other hand comprises a number of *samples* of something, usually taken at regularly spaced intervals.

Raster systems (Figure 2.6 right) store information as a rectangular matrix of cells, each of which contains a measurement that relates to one geographic location. Raster images thus constitute a 'sampling' approach to the representation of spatial information. The more samples that are taken, the closer the representation will be to the original. One of the most widespread forms of raster data used in GIS is remotely-sensed imagery, from earth orbiting satellites such as SPOT and Landsat Thematic Mapper (see Chapter 3). In this case, each of the raster cells in the resulting matrix is a measurement of the amount of





ID	Use	Lines
101	House	1,2,3,5
102	House	5,2,7,3
235	House	8,92,3,5
236	Pub	12,2,5,7
999	Street	2,6,4,5

1	1	1	9	3
1	1	9	9	3
5	5	9	3	3
5	5	5	3	3
5	5	5	3	3

Figure 2.6 Vector (left) and raster (right) representations of an area map.

electromagnetic radiation in a particular waveband that is reflected from a location to the sensor.

In contrast, in a vector data file (Figure 2.6 left) the data is represented in terms of geometric objects or 'primitives'. These primitives are defined in terms of their locations and properties within a given co-ordinate system, or model space. Vector systems do not so much sample from an original as describe it. The most common origin of vector data is from digitised maps, where the various features inscribed upon the surface of the paper map are converted into co-ordinates, strings of co-ordinates and area structures, usually through the agency of a digitising tablet.

It should be appreciated that within each of these basic data models there are ways in which data models may be implemented, often closely tied to specific software packages. As a result of this diversity, in the following discussion the aim has been to concentrate more on generic properties and features of the models themselves rather than specific details of individual data structures. Let us begin by looking in detail at the vector model.

2.6 VECTOR DATA STRUCTURES

In a vector data structure each geographical entity in a given layer can be represented by a combination of:

- A geometric primitive (fundamental mappable object);
- Attribute data which defines the characteristics of the object;
- Topological data (discussed further below).

Let us look at each of these features individually.

Fundamental mappable objects

Geographical data are usually represented by three basic geometric primitives (Burrough 1986:13):

- point
- line (arc)
- area (polygon)

The first, and simplest way of representing a feature is as a discrete *x*, *y* co-ordinate location, in effect a *point*. The second method of representation builds upon this, representing features as series of *x* and *y* co-ordinates, which are ordered, so as to define a *line*, often referred to as an *arc*. The third method of representation builds upon this, representing features as a series of lines which join to enclose a discrete *area*, normally referred to as a *polygon*. In each case the basic building blocks are discrete *x, y* co-ordinates. In the case of points they are treated in isolation; with lines they are ordered to form a sequence; and in the case of areas ordered to form a set of lines that enclose a discrete area. These basic geometric primitives can be termed *fundamental mappable objects* and are depicted in Figure 2.7.

In the vector-GIS the thematic layers in the overall spatial database comprise sets of georeferenced points, lines and areas. As well as storing the co-ordinate information relating to each discrete feature in a given layer, the vector-GIS assigns each feature a unique code or identification number. This code number is critical as it not only enables the GIS to keep track of the various features within each layer, but also provides a vital link that can be used to relate attribute information to the features.

To illustrate how these fundamental mappable units are applied in practice, a linear earthwork bank might be represented as a line with the attribute 'Earthwork bank' and information about its date, period and legal status. An isolated lithic find-spot may be represented as a point object, whose attributes include the label 'Obsidian', some information about its context, its dimensions and perhaps techno-functional information; while an area of limestone geology could be represented as an area entity with the label 'Limestone' and information about the calcium content or geological period.

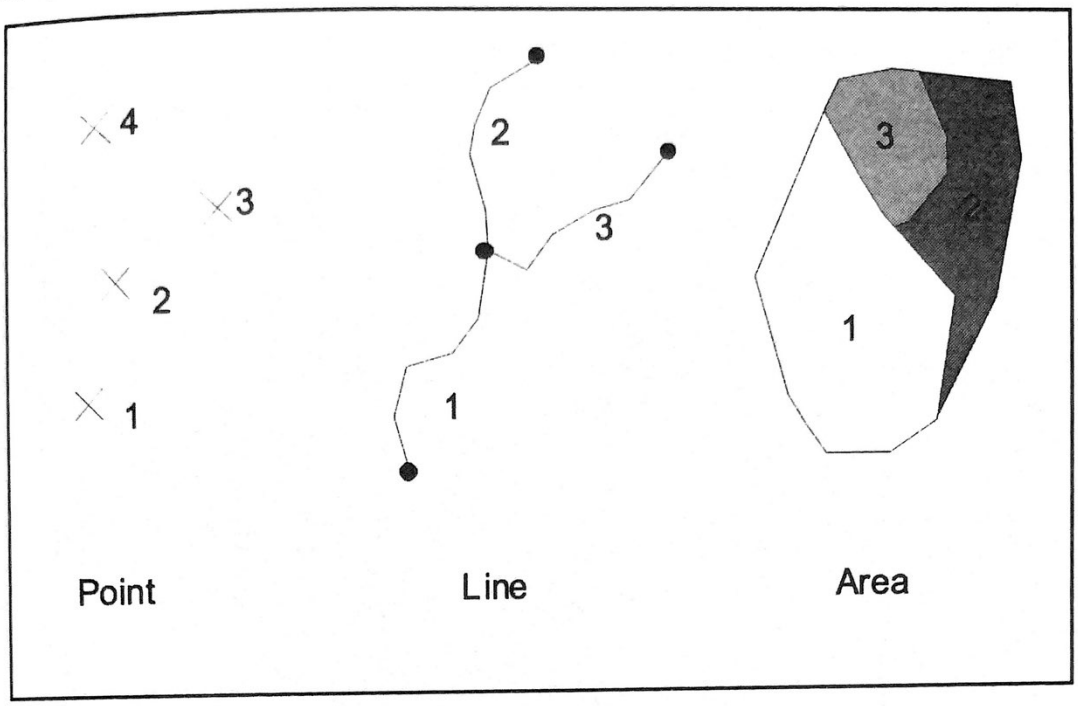


Figure 2.7 Examples of point, line, area.

Spatial dimension of primitives

The basic geographic elements discussed above are often referred to in terms of their dimensionality. A point entity is regarded as having no spatial extent, and is thus referred to as a 'zero-dimensional' spatial entity. Lines connecting two points possess length, but not extent and are therefore referred to as 'one-dimensional' entities. 'Two-dimensional' spatial objects are those that have both length and width—in other words areas. The spatial dimension of an entity is important because it has a direct effect on the logical relationships that are possible between geometric entities.

Imagine a spread of points. Because a point is zero-dimensional, it cannot contain or enclose other entities. Now add a line. Because lines are considered to have only length, but no width—one-dimensional—the points entities must either be on one side of the line or the other.

Intuitively it may be expected that surfaces, such as elevation models, would be three-dimensional. However, surfaces in most GIS systems are stored with the third dimension recorded as an *attribute* of a two-dimensional entity. True 3D representation of objects requires surfaces and solids to be represented through an extension of the *spatial component* of the data model into three dimensions, not merely the use of an attribute as a z co-ordinate. To separate the typical GIS-derived surface representations from true 3D models (we discuss fully 3D GIS in Chapter 12), they are sometimes referred to as 2½-dimensional entities.

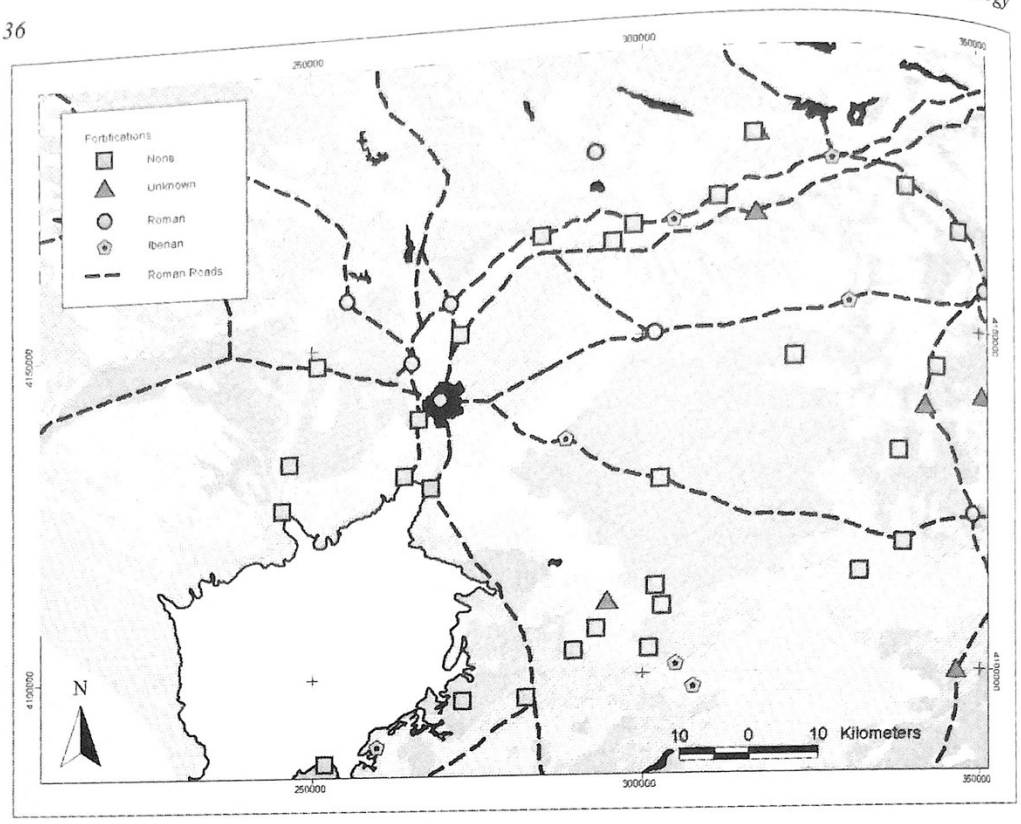


Figure 2.8 Fortified Roman and Iron Age centres in the lower Guadalquivir Valley, Andalucia. This single GIS-produced map combines area, network and point data themes. Point attribute data (nature of the fortifications) is used to control the shapes of the sites and the polygon attribute data (soil classification) to control the type of shading.

2.7 AN EXAMPLE OF A 'SIMPLE' VECTOR STRUCTURE

The simplest method of explaining how vector GIS data files are structured is to work through an example. Rather than adopt a particular proprietary format, however, the description that follows is of a slightly simplified and fictional format. Its structure actually owes something to several real formats used by commercial GIS.

To describe a geographic theme, then, we will need sections of our data file to describe *point*, *line* and *area* features and we will also need sections for their *attributes*. Before that, however, we will start with a simple format for describing the general properties of our data theme in terms of a *documentation section*.

Documentation section

In our format, this consists of two columns, separated by a comma. The left column identifies the data, which is given in the right hand column. A typical documentation section might read:

```
DOCUMENTATION
   title,    Long Barrows
   system,   plane
   units,    metres
   min. X,   400000.0000000
   max. X,   420000.0000000
   min. Y,   160000.0000000
   max. Y,   180000.0000000
```

The *title* field contains the title of the whole data theme, *system* and *units* indicate the type of co-ordinate system and the unit of measurement for the co-ordinates and the next four fields define the maximum and minimum co-ordinates of the data—in other words they define a rectangular region within which all of the objects will be found.

This is, admittedly, a very minimal documentation section. Most systems will also keep a record of the original scale of the data, the source, the projection system used, some estimate of its accuracy and many other facts about the data theme.

Data section for points

Point data is straightforward to store in a spatial database. Point data normally consists of a series of co-ordinate pairs, each associated with an identifier. In our simple vector data structure, the data section for a points file might follow this format:

```
POINTS
identifier, x1, y1
...
identifier_n, xn, yn
end
```

The POINTS line tells the GIS that each of the lines that follows reference a single point entity. This is followed by any number of identifiers preceding the relevant co-ordinate pair. The identifier is a number which uniquely identifies each of the points, and to which any attribute data can be linked (see below). For example, if we were recording the locations of small-finds in an excavation trench, a sensible identifier would be the unique small finds numbers assigned to each artefact as it is excavated. An identifier string 'end' is used to indicate the end of the section.

A complete data section for the four long barrows shown in Figure 2.9 might be as follows:

```
POINTS
1, 103.97,   98.43
2, 143.75,   103.79
3, 110.85,   93.63
4, 114.17,   89.04
end
```

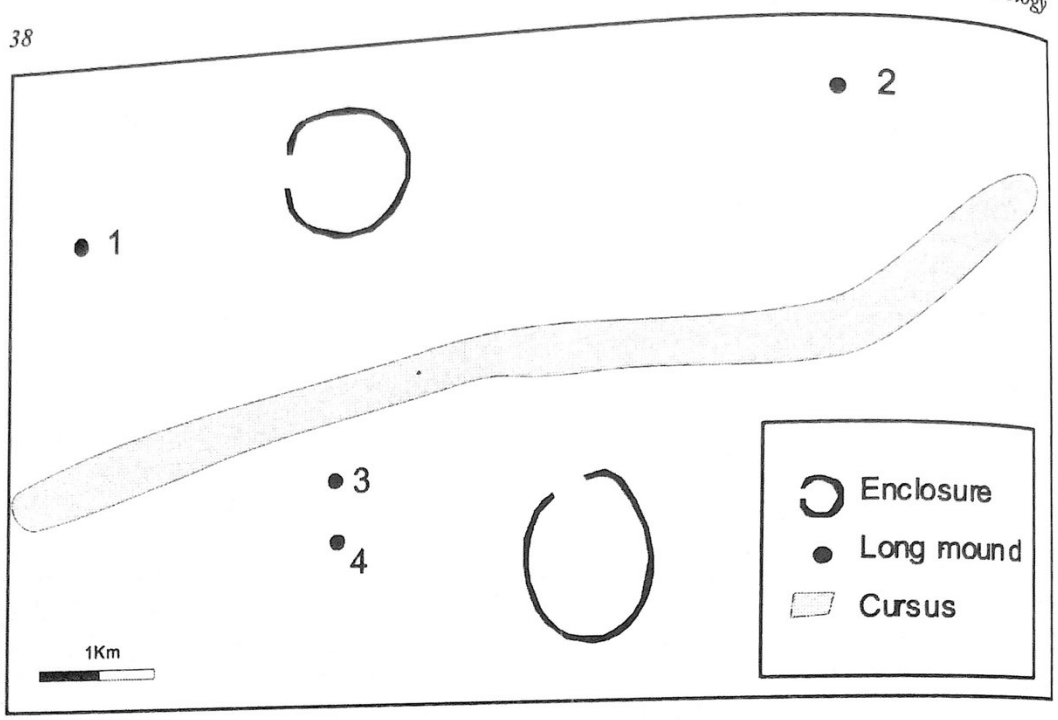


Figure 2.9 A hypothetical landscape.

This contains the codes 1,2,3 and 4 to identify the long barrows and the co-ordinate locations of each of the barrows.

Data section for lines

The additional complication presented by line files (e.g. the cursus shown in Figure 2.9) is that they can contain any number of points. To allow for this, the lines section of our data might require that the identifier be followed by the number of points in the line, and then by that number of co-ordinate pairs like so:

```
LINES
identifier, n
    x1, y1,
    ...
    xn yn
identifier2, n2
    ...
end
```

As for points, we can allow an identifier of 'end' to indicate the end of the lines section. An area section (*e.g.* for the enclosure depicted in Figure 2.9) can be identical to the line section except that the identifier will be AREA and the first and last co-ordinates of the line will be the same so that the line segments close to form an area or closed polygon (in other words $x1,y1 = xn, yn$).

Attribute values files

On a traditional paper map, the attributes of the geographical objects are either written as text next to the objects, or defined by a legend. A legend provides a list of attributes together with the symbols, line-types, hatching or colours which are used to identify them on the map. In the spatial database they are more commonly stored as simple ASCII attribute descriptor files (as shown below) or tables in a DBMS. In practice, the identifiers assigned to each object can be used to link the spatial entities with any number of attributes. Our system will allow attributes to be defined in an attribute section, which associates attribute codes with attributes as follows:

```
ATTRIBUTE
name,  type
code,  attribute
code,  attribute
code,  attribute
end
```

The section begins with the name of the attribute and its data type (*e.g.* character, decimal, or integer) and then contains one line for each identifier for which an attribute is recorded. A simple attribute file containing a single piece of information *about* each of the long barrows in the point file described above, might serve to link the identifier code for the long barrows with a number indicating their length in metres:

```
ATTRIBUTE
length, decimal
1,  39.5
2,  24.3
3,  66.8
4,  86.7
end
```

Exactly the same process can be used for assigning attributes to line and area entities. Additionally, there can be several attribute sections in the file, allowing the spatial entities to have any number of different attributes associated with them—for example, different attributes sections might exist to indicate the name, orientation and type of the long barrows. A complete data file defining the four barrows, each with four attributes is listed in Figure 2.10.

Topological information

When we use a map we are often explicitly interested in the logical geometrical relationships between objects. For example, if we were trying to re-locate a ploughsoil scatter on the basis of a shaded area on a 1:10,000 scale map sheet, we would look for the specific field or block of land the site was located in, and perhaps the nearest recognisable building or intersection of field boundaries to orient ourselves within the field.

```
DOCUMENTATION
   title,    Long Barrows
   system,   plane
   units,    m
   min. X,   400000.0000000
   max. X,   420000.0000000
   min. Y,   160000.0000000
   max. Y,   180000.0000000
POINTS
   1, 103.97,  98.43
   2, 143.75,  103.79
   3, 110.85,  93.63
   4, 114.17,  89.04
   end
ATTRIBUTE
   length, decimal
   1,  39.5
   2,  24.3
   3,  66.8
   4,  86.7
   end
ATTRIBUTE
   name, char
   1,  "Adams Grave"
   3,  "Beckhampton Road"
   end
ATTRIBUTE
   orientation, integer
   1,  1
   2,  4
   3,  3
   4,  1
   end
END
```

Figure 2.10 A complete example of a simple vector data structure, defining four point entities, each of which might have up to three attributes (length, name and orientation).

Encoding topology

Topology therefore refers to the logical geometry of a data theme (we discussed this at the start of this chapter, but see Figure 2.11 for a reminder of why topology is a familiar concept to most archaeologists). It comprises the connections and relationships between objects. In the GIS it has to be explicitly calculated and recorded rather than inferred from their spatial location. Let us return to our geometric primitives: points, lines and areas. As points effectively have 0-dimensions they do not really pose any topological problems. However, with lines a number of important potential topological relationships can be identified.

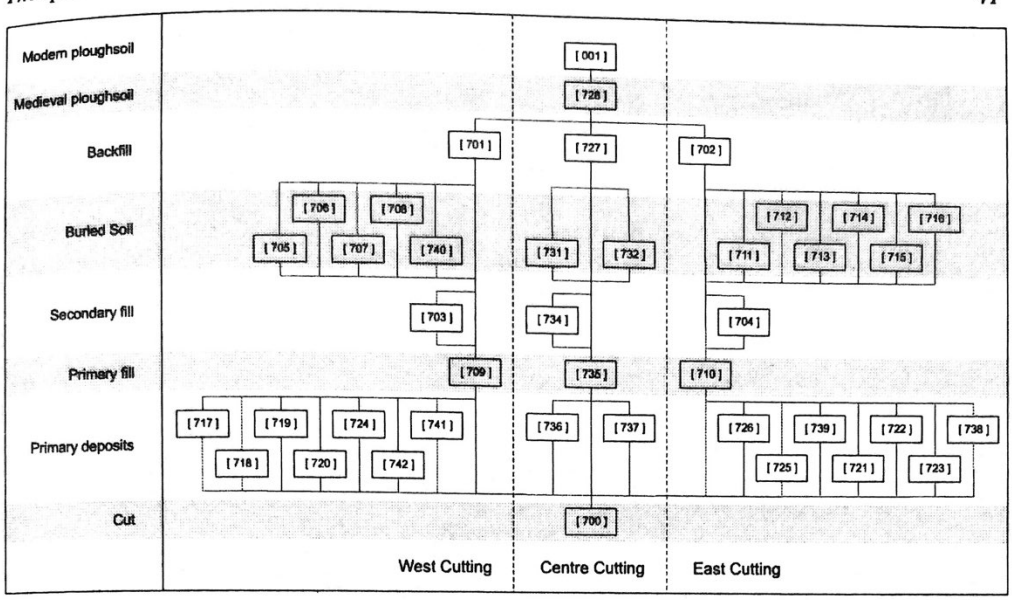


Figure 2.11 A Harris matrix as an example of topology. The true spatial locations of features are irrelevant, what is important are their absolute stratigraphic relationships.

Topology of line data

Consider a simple road network. The locational component of this can be represented as a series of lines. Attributes can be recorded for each of the roads, for example the name of the road as shown in Figure 2.12 (left).

However, this in itself is insufficient to record the full complexity of the road network. If we are interested in the possibilities of transport within this system, we need to know whether we can drive from the A31 onto the M4 for example. To

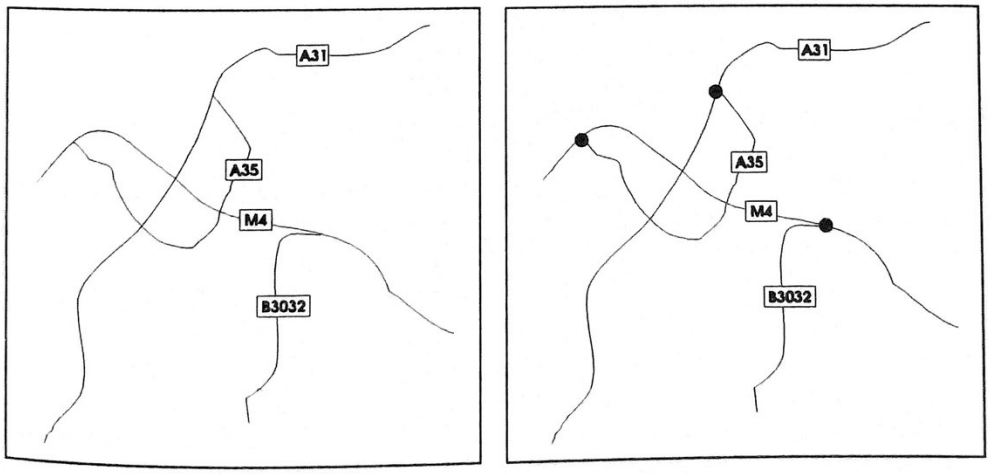


Figure 2.12 Roads represented as line entities with attributes. Right: introducing nodes (black dots) makes the relationships between roads clear.

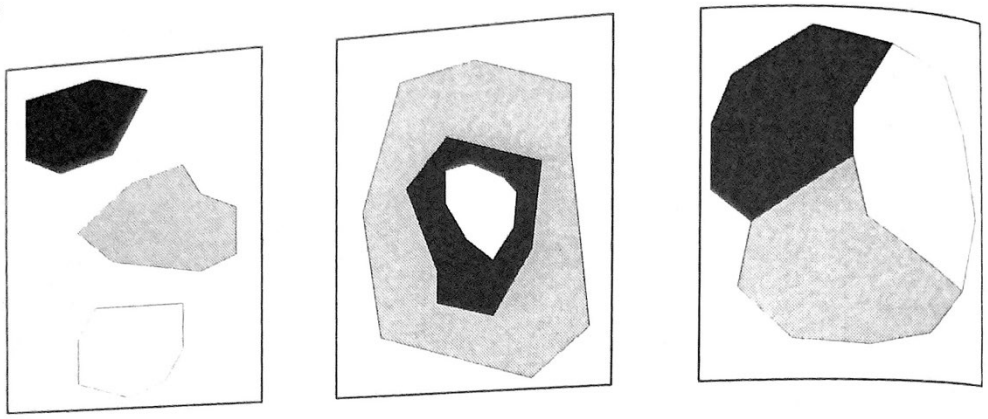


Figure 2.13 Three regions showing some possible topological relationships. Left, unconnected; centre, regions contained island fashion within other regions; and right, regions connected by shared boundaries.

understand this, we need to record where the roads meet as opposed to where they cross each other with bridges. This sort of information about the relationships between objects is part of the topology of the theme. If we simply store the features (in this case individual roads) as little more than ordered lists of co-ordinates (as we did in our simple vector data structure), we have no way of recording whether and where lines connect, cross or simply terminate without a connection.

In the case of the road network it is necessary to introduce another spatial object into the model to record the way in which the roads relate to one another. We can refer to this as a node. A node represents the point at which an arc begins or ends, and records the other arcs to which it is connected. We will be discussing nodes in a little more detail shortly. In Figure 2.12 (right) a series of nodes have been included corresponding to road junctions, and we can now infer that to get from the A31 to the M4 we must use the A35. Note that our roads can be represented by several arcs—in our example the M4 is composed of three separate arcs—and that lines which cross must be broken up into separate arcs, with nodes at the crossings, if we are to use nodes to build up topology.

Topology of area data

With area data even more complex relationships are possible. Consider the three regions shown in Figure 2.13 (these might be soil classes within a site catchment zone, or geological substrates).

The visual relationships between the three regions in each case are straightforward: on the left the regions are unconnected; in the centre the white region is wholly contained by the black region which is itself wholly contained by the grey region; on the right the three regions are adjacent, connected through shared boundaries. Each of these cases or combinations of such cases may occur in particular types of geographic situation and so each must be explicitly coded in the spatial database.

Problems with simple data structures

To enable the vector-GIS to effectively 'understand' the geometrical relationships between the features in the spatial database it has to establish topology for them. This information has to be stored with each data layer or calculated 'on the fly' as and when needed. It should be noted that there are few standards as to how this topological information should be stored, with different commercial vector-GIS systems implementing different, and often highly complex, proprietary solutions.

Although the straightforward method for representing geographic objects described earlier in our simple vector data structure has the benefit of being easy to understand, there are a number of problems with such a data format. Point entities do not present a major problem, and are often stored in a similar way to this in larger vector-GIS systems. This is due to the zero-dimensionality of the point, which means that points cannot cross or contain other entities, and so present few topological problems. On the other hand, the storage of lines and areas in the simple model is extremely restrictive.

Several limitations apply to the lines section of our simple example. The main restrictions are twofold: the connections between lines are not explicitly coded within the data; and there is no directional information recorded in the data. Recording whether lines connect, as opposed to cross or terminate without connection, is vital within network data. Direction within line data may be of considerable importance in the representation of rivers or drainage systems, where the GIS may need to explicitly incorporate flow direction into analyses. For example, if we were to attempt to assess the feasibility of early trade networks based upon river transport, we would need to consider the costs incurred in travelling up and downstream on any given river in the network.

The problems of recording adequate topological information for area data are rather more severe than those of line data. We have already illustrated how the relationships between areas are not always simple. Additional problems are encountered as a result of areas that share boundaries, and of the relationships between areas enclosed by others and their surroundings—so called 'island' polygons. In cases where geographic areas border one another, such as soil,

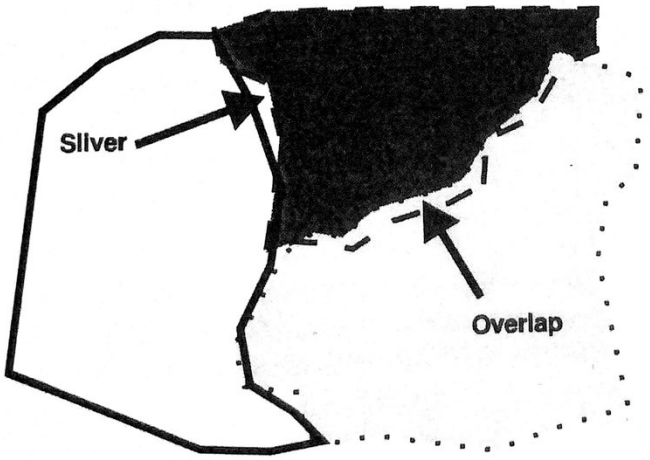


Figure 2.14 Areas represented by whole polygons showing some possible errors that result from shared boundaries.

geological or political maps, the simple model of area data (what is often termed the *whole polygon* model) repeats each shared boundary. This can lead to a variety of problems, particularly if data derives from digitised maps. It effectively doubles the storage requirement for each coverage, and therefore increases the time required to process the data, as two boundaries must be updated whenever a modification is made. In addition to this, at the data acquisition stage, the shared boundary lines may not have been digitised precisely. We will look in detail at the process of digitising in Chapter 3. These differences between the lines will show up as *slivers* or *overlaps* in the data coverage (see Figure 2.14). Additionally, there is no method of checking the data for topological errors such as figure-of-8 shaped polygons, where an error in the order of co-ordinates forces the area boundary to cross. These are frequently referred to in the literature as *weird* polygons and, as the name suggests, are to be avoided.

The simple whole polygon database also contains no record of the adjacency (or otherwise) of different polygons, so there is no way of searching for areas that border one another. Similarly, there is no straightforward way of identifying island polygons.

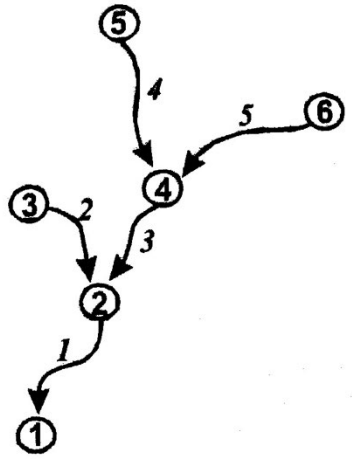
An enriched vector data structure

If analysis is to be carried out on data stored as vectors, the data must be capable of answering basic questions such as *are two polygons adjacent?* or *does this area contain any islands of different value?* Our brief examination of the simple data model should have illustrated that points, lines and whole polygons are not sufficient to represent both the spatial and topological complexity of real-world geographical data. As a result of this, more sophisticated data models are necessary in situations where either (1) analysis is mainly to be carried out on vector data themes or (2) the accuracy and redundancy problems associated with whole polygon data are deemed unacceptable.

The key to solving the problem lies in the incorporation of topological information explicitly into the data structure. As intimated earlier, there are unfortunately few data standards for the storage of full-topology geographic data. Because GIS is a relatively new field, most systems have developed largely in isolation and have adopted slightly different ways of storing and representing data. The result is a plethora of proprietary data standards, which, incidentally, can cause extensive problems for the porting of data from one application to another.

Networks

As implied earlier, simple lines and chains of arcs contain no information about connectivity or direction. To store such information in a digital form it is necessary to introduce the notion of a *node* into the data structure. Nodes are essentially point entities that are used to indicate the end of lines (or chains of lines), and to store the characteristics of the ends. In such a structure the line contained between two nodes is often referred to as an *arc*, and each point that makes up the line may be referred to as a *vertex*. Arcs that share a node are said to be linked.



Line	To	From
1	1	2
2	2	3
3	2	4
4	4	5
5	4	6

Node	Lines		
1	-1		
2	1	-2	-3
3	2		
4	3	-4	-5
5	4		
6	5		

Figure 2.15 A simple river network represented by a list of lines (chain of arcs) and a separate list of nodes.

In the example in Figure 2.15, two tables have been used to represent a simple river network. The 'Node' table holds a record of which lines are incident at each node, and also which end of each line is incident at each node. This has been done by using the number of the line (*i.e.* '1') to indicate that the line starts at this node, or a negative number (*i.e.* '-1') to indicate that it ends at the node. A quick check through the nodes table should then reveal that each line number has one (and only one) negative equivalent so we can use this as a quick check: add up the numbers in the lines section, and it should come to zero. If not, then we certainly have an error. This data structure now represents the direction of each line, which in turn could represent the direction of flow within the river system.

Note that the 'from' and 'to' fields in the lines table are not strictly necessary, because they can be deduced from the nodes table. However, many systems include them to speed up processing. Note also that no *geographic* or *attribute* information has been included in this example. As in the simple data structure presented earlier, these must be stored but they have been left out here for the purposes of clarity.

Area data: arc-node-area data structures

We have seen some of the limitations of simple polygons as a model for storing area data. These have been summarised by Burrough (1986:27) as:

1. Data redundancy because the boundaries of polygons must be stored twice
2. Resultant data entry errors such as slivers, gaps and weird polygons
3. Islands are impossible to store meaningfully
4. There is no way of checking the topology of the data for these errors.

Fortunately, the arc-node model of network data outlined above, can be extended to record area data. The following discussion describes a simplified example of an arc-node structure, loosely based on the Digital Line Graph (DLG) format of the US Geological Survey.

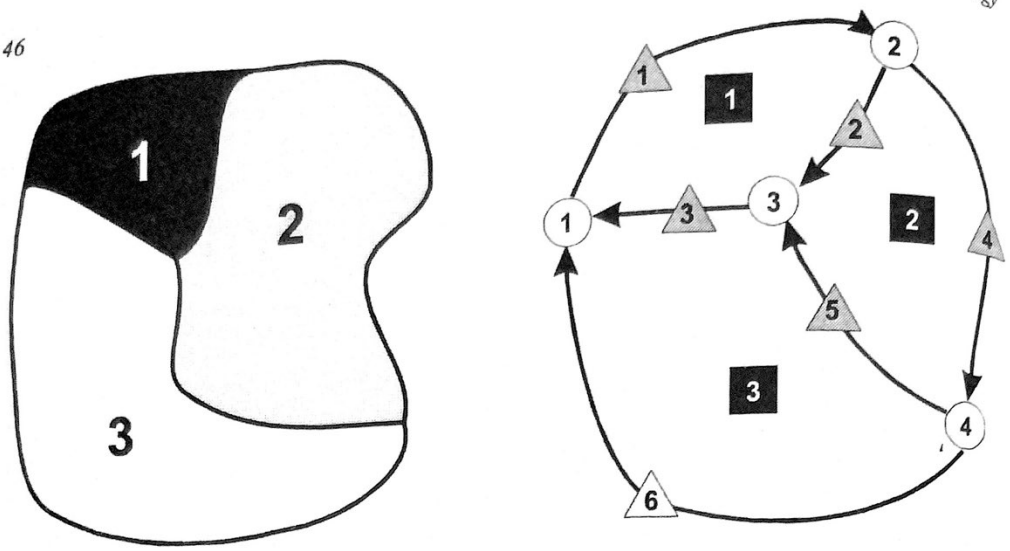


Figure 2.16 A simple topology problem. Showing (left) three areas of value 1, 2 and 3 and (right) a representation of their topology. Line numbers are given in triangles, nodes in circles and areas in the squares.

Figure 2.16 (left) shows a simple example of a configuration of areas that might require encoding. This shows three areas of value 1, 2 and 3, which have shared boundaries. To overcome the shortcomings listed above, what is required is a data model that records the boundaries only once, and preserves the relationships between the areas. As a first step we might give the lines and nodes numbers, and record these as for network data. If the lines and nodes are numbered as in Figure 2.16 (right), a 'first stab' at encoding Figure 2.16 might be as shown in Table 2.2.

Note that Table 2.2 deviates slightly from the network example in that the lines do not record their start and end node (as we said, this information can be deduced from the nodes table) and also that some spatial data has been included (the co-ordinates defining each vertex in the overall line or arc). All the attribute data has once again been left out for the sake of clarity. As before, the lines have been given a direction by recording each as a positive number at the start node and

Table 2.2 Introducing nodes into the data structure.

Line	Co-ordinates defining vertices
1	$x_1, y_1 \ldots x_n, y_n$
2	$x_1, y_1 \ldots x_n, y_n$
3	$x_1, y_1 \ldots x_n, y_n$
4	$x_1, y_1 \ldots x_n, y_n$
5	$x_1, y_1 \ldots x_n, y_n$
6	$x_1, y_1 \ldots x_n, y_n$

Node	No. lines	Lines
1	3	-6, -3, 1
2	3	2, 4, -1
3	3	3, -5, -2
4	3	6, -4, 5

as a negative one at the end node—again this is a common way of allowing the system to quickly check the topology.

In order to fully record the area component of the data, however, we need a third table. Although there is strictly no real need for the area data to be linked directly to any spatial entities, area codes are often recorded by creating special points within the relevant areas and attaching the area data to these. There will be three area entities for this example.We proceed by recording which area is to the left and which to the right of each line, and adding this to the lines section as shown in the top of Table 2.3. A third data section (Table 2.3, bottom) now records the data about each area and we now have somewhere we can put the attribute data for each area (although it is not shown here). This new section records which lines form each area's polygon, coding them as positive if they go clockwise around the polygon or negative for anti-clockwise. As before, this can actually be deduced from the left and right areas of the lines table and thus represents a compromise between data redundancy and processing efficiency.

Island and envelope polygons

So far our example has been restricted to polygons with shared boundaries. In Figure 2.17 we have a much more complicated topological structure that includes an island (4) and a 'complex island' (inside area 3). Initially we can proceed as before, and assign numbers to the lines, nodes and areas as shown in Figure 2.17 (right). This time we have included a special 'bounding polygon' coded 0 (zero) to represent the area that surrounds our geographic regions, and a special line 0 (zero) to describe it. This is sometimes, but not always, required by digitising programs.

Table 2.3 Introducing areas into the data structure.

Lines	Vertices	Left area	Right area
1	x, y ...	4	1
2	x, y ...	2	1
3	x, y ...	3	1
4	x, y ...	4	2
5	x, y ...	3	2
6	x, y ...	4	3

Nodes	X	Y	No. lines	Lines
1	?	?	3	-6, -3, 1
2	?	?	3	2, 4, -1
3	?	?	3	3, -5, -2
4	?	?	3	6, -4, 5

Areas	No. lines	Lines
1	3	2, 3, 1
2	3	4, 5,-2
3	3	-5, 6,-3

The line and node tables are constructed as for the previous example, including these extra lines and nodes. These are shown in the three tables of Table 2.4. The areas table (*i.e.* the information associated with the label points) is now modified so that each area record also includes the codes for any areas it encloses (*islands*), and for any area that encloses it (*envelopes*). In a similar way to the direction of lines, islands are recorded as positive numbers while envelopes are given as negative numbers. In this way the entire hierarchy of islands and envelope polygons can be recorded within the areas table.

These three tables now represent a full description of the area, including the geographic and topological elements. More importantly, this structure does not repeat any of the lines—so is not prone to the problems of 'whole polygon' structures—and it allows a vector-GIS to quickly find the logical relationships between all the areas of a spatial theme. Attribute data can be linked to the area identifiers as for our simple data structure summarised in Figure 2.10.

Although vector-GIS vary considerably in their precise data formats, the basic principles are similar to those described here, with a full geometric description of the geographic and topological components of the data being built up from the relationships between points, lines and areas.

Summary

Having traced the various table relationships around Figure 2.17, you should by now appreciate the complexities faced in calculating and storing topological information for area data. To use and understand the geometric relationships encoded on a map sheet involves a relatively trivial visual task, one that we perform so routinely we tend to take it for granted.

For the GIS to gain a similar understanding is far from trivial. To really understand what makes topological data structures different from the simple models discussed previously, try to think how you could create automated methods

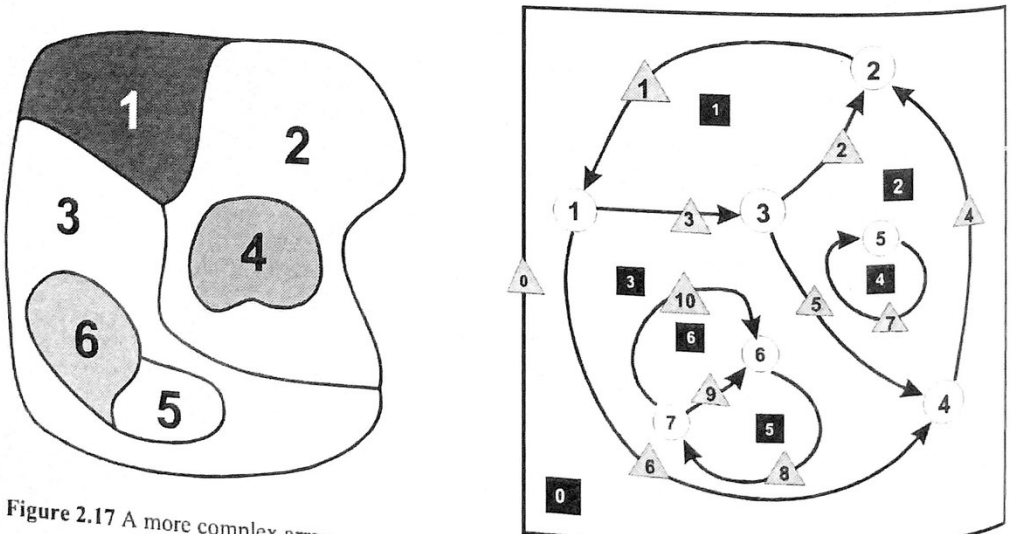


Figure 2.17 A more complex arrangement of polygons (left) and a graph of their topology (right). Line numbers are given in triangles, nodes in circles and areas in the squares.

to answer questions such as: '*is area 1 adjacent to area 2?*' or '*how many square km is area 3?*', without looking at the pictures! These types of questions can only be answered, by a computer program such as a GIS, if the vector data structure employed records the full complexity of the topological relationships present between the constituent geographical objects.

The point of these rather detailed examples is to illustrate the care required to store spatial information in a comprehensive and meaningful way. Fortunately, most of the work of creating this topological information and then checking that the logical structure of the database is coherent can be automated within the data input subsystem of the GIS. For example, in systems such as Arc Info or Cartalinx it is as straightforward as issuing the commands *build* or *clean* on a given vector data layer, or selecting the *create topology* button on a menu bar.

Table 2.4 The topology of Figure 2.17. Lines numbers are given in triangles, nodes in circles and areas in the squares.

Lines	Vertices	Left area	Right area
1	x, y ...	7	1
2	x, y ...	2	1
3	x, y ...	3	1
4	x, y ...	7	2
5	x, y ...	3	2
6	x, y ...	7	3
7	x, y ...	2	4
8	x, y ...	3	5
9	x, y ...	6	5
10	x, y ...	3	6

Nodes	x	y	No. lines	Lines
1	?	?	3	-6, -3, 1
2	?	?	3	2, 4, -1
3	?	?	3	3, -5, -2
4	?	?	3	6 , -4, 5
5	?	?	2	-7, 7
6	?	?	3	-9, 8, -10
7	?	?	3	-8, 9, 10

Areas	No. lines	Encloses (+ve) Is enclosed by (-ve)	Lines
7	4	1, 2, 3	0, -1, -6, -4
1	3	-7	2, 3, 1
2	4	-7, 4	4, 5, -2, 7
3	6	-7, 5, 6	-5, 6, -3, -10, -8
4	1	-2	7
5	2	-3	9, 8
6	2	-3	10, -9

The user then need only remember a few basic rules while digitising a map in order to ensure the complete data structure is maintained. Different GIS digitising programs do this in different ways, and some are better and easier to operate than others. We will discuss the digitising process in more detail in Chapter 3. As with data standards generally, there is at present no one accepted method of digitising and storing this type of data. The worked examples given here present a simplified form of the data models that are employed by many GIS and Spatial Data Builders such as Arc Info, Cartalinx, SPANS, GIMMS or MapInfo. Not all proprietary data structures incorporate all the levels of the model, while some include extra information in each of the tables, building in a degree of data redundancy to speed up subsequent data processing. We firmly believe that to construct coherent and robust vector data layers it is important to realise the complexity of the task the GIS has to perform when we blithely select the option 'create topology' from a pull-down menu.

2.8 RASTER DATA LAYERS

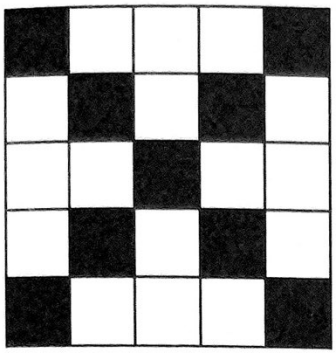
Rasters are relatively simple data structures and raster data is used in a far wider set of contexts than simply GIS. Examples of other kinds of raster applications include geophysical survey software, photo retouching programs and scientific image processing packages. Raster structures store graphical information by representing it as a series of small parts or elements. In the image processing literature, these are referred to as *pixels* or sometimes as *pels* (both derive from a shortened form of the words 'Picture Elements'). In the GIS literature these units are frequently referred to as 'grid cells' or simply 'cells'.

In a raster system, the spatial database once again comprises a series of georeferenced thematic layers. Unlike the vector data model, in raster systems we do not define individual features (whether points, lines or areas) within each thematic layer. Instead, the study area we are interested in is covered by a fine mesh of grid cells—Gaffney and Stancic (1991:26) have likened it to a chess-board—and each cell is coded on the basis of whether it falls upon a feature or not.

As a result, a simple raster representation of an 'X'-shaped conjunction of field boundaries might look like Figure 2.18 (left). Here the junction is represented in terms of a matrix of positive values for the cells that fall upon a boundary and zero values for those that do not. A simple text file to store that data (shown on the right) might therefore consist of a series 1s and 0s, with commas as dividers to separate the cells from one another.

Some file structures (for example the Idrisi ASCII storage format) are almost as simple as that shown in Figure 2.18 (right). Normally text files are not used to represent the pixels, instead the storage is based on individual bits and bytes. This is because raster data files can contain extremely large numbers of cells. Consider, for example, a region of 20km by 20km where each cell represents a 100m x 100m land parcel—even at this course resolution the data layer will contain 40,000 individual cell values.

Returning to the example of the vector data structure, raster systems represent the three fundamental mappable units in the following way: points by a series of single grid cells; lines by a set of connecting cells; and areas by contiguous blocks of adjoining cells.



```
1,  0,  0,  0,  1,
0,  1,  0,  1,  0,
0,  0,  1,  0,  0,
0,  1,  0,  1,  0,
1,  0,  0,  0,  1,
```

Figure 2.18 A simple raster (left) and corresponding data file (right).

Attributes and raster structures

In the vector model, the spatial representation of features and their non-spatial attributes are kept separate, as a series of separate text files or a linked relational database. In the raster model the representation of features and their non-spatial attributes are instead merged into a unified data file as attributes are encoded directly into the thematic layer. It should be noted that this situation is changing as a number of raster systems (for example Idrisi) begin to support some degree of attribute file or external database access. In such systems rather than an attribute value, a numeric identifier is encoded into each cell that acts as a link to an external attribute database file in a similar way to a vector identifier label. This type of raster layer has been referred to as a *data frame*.

Despite such enhancements, it is fair to say that in the raster data model we are recording not a series of discrete spatial features, but the behaviour of an attribute in space. The particular attribute value of the ground surface we are interested in is recorded as the value for each cell. The numeric value assigned to each grid-cell can correspond to: a simple presence-absence (1=archaeological site, 0=no-site); a feature identifier (*e.g.* 1=woodland, 2=urban, 3=lake); a qualitative attribute code (soil type 1, 2, 3 *etc.*); or a quantitative value (73m above sea level).

Locating individual cells

In terms of georeferencing a raster thematic layer, in practice the left, right, top and bottom co-ordinates of the grid-mesh are recorded, as is the resolution (*i.e.* dimensions) of each grid cell. The locations of individual cells are determined by referencing each cell according to its position within the rows and columns of the grid. For example, if the grid cells are 30m square (*i.e.* the grid resolution is 30m) and the corner co-ordinates of the overall mesh are known, it is a trivial task to count the cells along and up/down from any corner adding/subtracting increments of 30m to locate any one cell.

Topology and raster structures

It will no doubt come as some relief to note that unlike the vector model, there are no implicit topological relationships that need to be encoded within raster data. We are after all not recording individual spatial features but the behaviour of a particular attribute in space.

Sampling and resolution

The feature that most distinguishes raster approaches to GIS data from vector ones is the issue of sampling or *quantization,* which occurs because real space is continuous while its computerised representation is discrete. Moving between the two can cause many problems and can generate 'artefacts' that have nothing whatsoever to do with the properties of the real world we are trying to represent.

The choice of grid cell resolution is critical to any analysis we may want to undertake. The finer the resolution, the closer and more detailed the representation is to its ground state. There are no rules to guide us, only the requirements and tolerances of the analyses we would like to perform. It is obvious that the junction of field boundaries depicted in Figure 2.19 (left) is not a very good image of what are probably the narrow, continuous lines of the original. Of course, the representation could be improved by increasing the number of samples that are taken, in other words increasing the *resolution* of the data. The junction in Figure 2.19 (right) shows the effect of increasing the number of pixels along each axis of the image by a factor of 2. Instead of the mesh being 5 pixels deep and 5 wide, the image is now 10 pixels deep and 10 wide. This new 10 x 10 image is still not particularly good, particularly at the junction itself, but it is rather better than the 5 x 5 image.

Ideally, we would want to have as high a spatial resolution as possible. Unfortunately, increasing the resolution has a significant impact on the storage requirement of the resultant data files. If the cells of our example land parcel were 20m by 20m (instead of 100m x 100m), then we would need 1000 x 1000 =1,000,000 cells in the data file rather than the original 40,000. This will result in a file around 1.5 million bytes in size, or around 1.4 Megabytes.

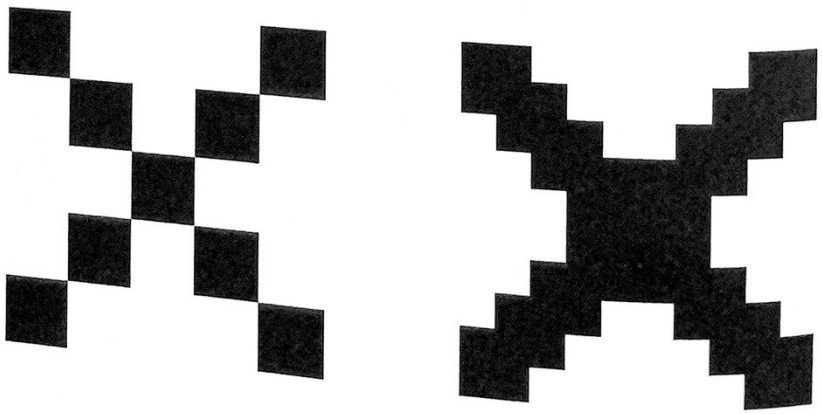


Figure 2.19 The effects of doubling the resolution of an image.

STORAGE REQUIREMENTS FOR RASTER DATA

One of the greatest problems with raster files is the large amount of storage space that they require, so predicting storage requirements is very important. To understand how to predict the storage requirement for a given application, it is first necessary to understand a little of how numbers and data are stored within computers.

At the lowest level, computers store information in *registers*. These are capable of representing only two values: 1 or 0. Using two of these registers (called bits), however, we can store four different binary numbers as follows: 00, 01, 10, and 11. It turns out that n bits can represent 2^n different numbers so that 4 bits can store 2^4, or 16 values and so on. One byte (eight bits) can store 2^8 or 256 different values or the numbers 0 through 255.

Because modern computers have very large memories and disks (at least compared with a decade ago) it is now common to refer to Kilobytes (Kb), Megabytes (Mb) or even Gigabytes (Gb) of memory. There are 1024 bytes in a Kilobyte, 1024 Kb in a Megabyte and 1024 Mb in one Gigabyte and so on (1024 is 2^{10}). Equipped with this arcane knowledge, we can begin to predict storage requirements for any given raster. We need to know two things:

1. the number of cells will there be in the raster;
2. the range of possible values we need to store in each cell.

The first is easy: we can calculate this from the size of our area, and the size of the individual cells. If we were interested in a region of 20km by 20km, and each grid cell represents an area of 100m by 100m, then the region will be represented as 200 cells (200 x 100m = 20,000m) wide by another 200 high. This results in 200 x 200 = 40,000 cells.

Next, we need to understand the range of possible values. This defines the storage allocation that needs to be made *for each cell*. If the data theme will only consist of two possible values—for example 'presence of an archaeological site' and 'absence of site'—then technically we need only allocate one bit for each cell. Usually, however, the cells can have many different values, as is the case in remotely sensed data, so that a larger unit of storage needs to be allocated to each cell.

Let us assume that our 20km by 20km data theme is a remote-sensed image, in which the cell values are scaled from 0 to 255. In this case, each cell needs to be stored not as one bit (2 possible values) but one byte (256 possible values). From this we can work out the storage requirement for this raster—the number of cells multiplied by the storage for each cell. In this case, the data file will be 40,000 bytes, which is a bit less than 40 Kb. Increasing the number of possible values that each cell can contain obviously increases the storage requirement for the cell. For example, if our image contains values that can be distinguished more finely, perhaps scaled between 0 and 4000, we will need to allocate more bits per cell—in this case 12 bits will allow values between 0 and 4096. Now the data file will be 40,000 x 12 bits = 480,000 bits or 470kb. Fortunately, most systems allow large raster files to be *compressed* (see main text) so that they do not, in practice, take up as much disk space as these calculations would suggest.

Compression

While increasing the resolution of a raster grid to say 10 x 10 or 100 x 100 improves the approximation to ground truth, it also dramatically increases the storage space required and increases the redundancy of much of the information in the image file. Unlike vector systems, raster structures explicitly store the absence of information as well as the presence. For example, if we return to the 4 long barrows outlined in Figure 2.10, in a vector structure these would be represented by a very compact set of 4 identifier codes and co-ordinate pairs.

In a raster system we would have to cover the barrow study area in a mesh and then encode not only a positive value for those cells which overlay a barrow, but zeros or negative values in all of the cells which do not—in effect non-data and the higher the resolution, the more redundancy there is likely to be. Each arithmetic increase in the dimensions of the image, results in a geometric increase in the number of pixels which must be stored: a 5 by 5 image contains 25 pixels, 10 by 10 contains 100, while a 100 by 100 image contains 10,000 pixels—so increasing the linear dimension by a factor of 10 actually increases the storage space required by a factor of 100. Obviously the resolution of the image therefore has a rather dramatic effect on the storage space the image requires.

These increases in the space occupied by raster images can be offset by using compression techniques. These encode the data in a more efficient way by removing some of the redundant data in the image. The simplest technique is referred to as run length encoding (RLE). RLE relies on the fact that data files frequently contain sequences of identical values which can be more efficiently stored by giving one example of the value, followed by the number of times it is repeated. For example, the 10 x 10 image in Figure 2.18 could be coded as pairs of numbers, in which the first represents the length of the run, and the second the value—so that 4,0 represents four consecutive zeros, reading right to left. This might look something like this:

```
2,1 6,0 2,1
3,1 4,0 3,1
1,0 3,1 2,0 3,1 1,0
2,0 6,1 2,0
3,0 4,1 3,0
3,0 4,1 3,0
2,0 6,1 2,0
1,0 3,1 2,0 3,1 1,0
3,1 4,0 3,1
2,1 6,0 2,1
```

The data file now only needs to store 68 numbers as opposed to the 100, which would be required for an uncompressed data file, providing a 'compression ratio' of about one third. Obviously little advantage is gained from compression where there are not enough identical values in long runs to make it worthwhile. In addition, a further disadvantage of compressed data files is that they must be decoded before they can be used, which takes additional time. For example, if the 5 x 5 raster on the left is encoded, the result is actually larger than the uncompressed

file because the coding carries a data 'overhead' which must be recouped before any saving in storage can be made. In reality, raster grids are usually much larger than these examples, and sometimes contain a high proportion of repeated values. RLE encoding in images can therefore, depending on the context and on the picture, save considerable storage space.

Because the size of storage in a compressed data file depends on the *content* of the file as well as the dimensions, it is not possible to predict exactly the storage that will be needed for any given application. A good strategy is to obtain an approximation of the ratio, which might be obtained from compression by experimentation with typical data files, and then to use this to estimate the overall storage requirement.

Quadtrees

Some GIS (such as Tydac Technology's SPANS) utilise an alternative to raster storage that allows the user to specify a highly effective resolution whilst mitigating against the commensurate increase in storage space. This is termed a *quadtree data structure* (see *e.g.* Rosenfield 1980, Tobler and Chen 1993). It is based on a successive division of the stored region into quarters, so that the 'resolution' is determined by the number of successive subdivisions used. Values are stored in a hierarchical format, working downwards from the first quartering of the region (Figure 2.20). The advantage of quadtree data is that it can be more efficient than a simple raster format for the storage of some types of data, such as soil type or administrative boundaries. This is because some of the regions need only be subdivided a few times, while the full depth of the tree is used to provide high definition at boundaries. Burrough (1986) gives a fuller description of quadtree data structures, together with a more comprehensive illustration, but it is sufficient for our purposes to realise that quadtrees can be an efficient and elegant storage option. In saying this, it should be acknowledged that, like run length

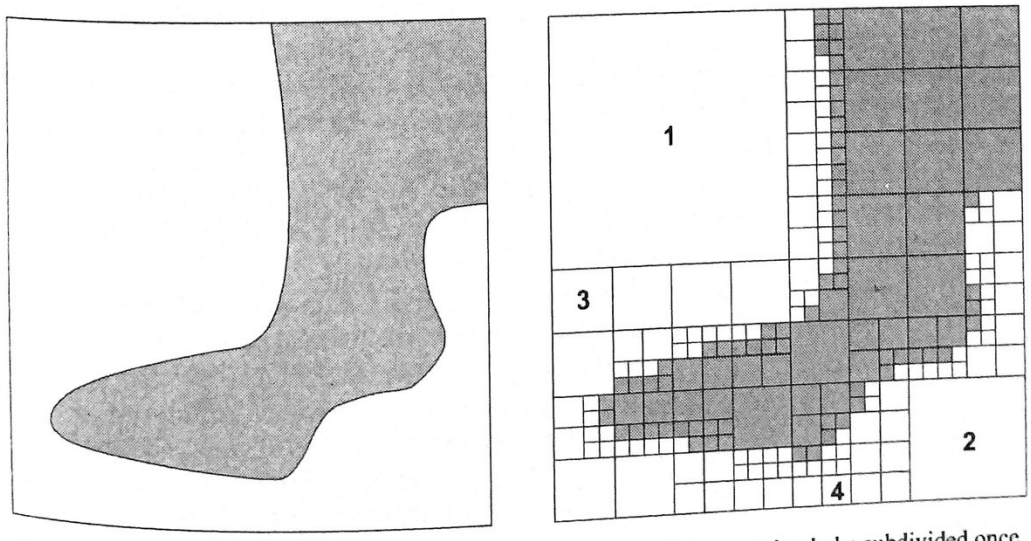


Figure 2.20 Quadtree data, to five levels (fifth not labelled). Some areas need only be subdivided once, while border areas require the full five divisions.

encoding, they require the system to do more work than for simple raster encoding, and so decrease the speed with which the GIS can process data.

Summary

Raster storage is simple but typically data-intensive unless the storage is at a low resolution, or mitigated through the use of compressed data or quadtree structures. Unless the resolution of a data theme is predetermined—for example where the cell samples derive from geophysical survey or fieldwalking—then a compromise is necessary between a high resolution, which provides better definition of features, and the increase in storage requirement and processing time that this generates. Analysis of raster data is relatively straightforward to undertake, although it is more limited than for vector data because the spatial relationships between geographical areas are not explicitly represented.

2.9 WHICH IS BEST—VECTOR OR RASTER?

A question that has often cropped up in archaeological discussions of GIS is *which is best — vector or raster?* As with most issues there is no clear-cut answer, with both approaches being adept at addressing different kinds of problems.

As we have discussed, unlike the vector model, which stores features as highly compact co-ordinate lists, the raster model is much more data intensive. However, the raster representation does define geographical space in a simple and predictable way — the spatial extent of a given attribute is always represented as a rectangular grid. As a result they are ideally suited to problems that require the routine overlay, comparison and combination of locations within thematic layers. As satellite and aerial-photographic data sources are themselves stored in a raster format, raster systems are good at handling and manipulating such data. In addition, the raster model copes better with data that changes continuously across a study area, *i.e.* data that has no clear-cut edge or boundary. Good examples are topography and changing artefact densities in the ploughsoil.

The vector model, with its topologically defined lines and areas, is much better at working with clearly bounded entities and network based analyses (*e.g.* shortest distance between points on a road system). The flexible database linkages make it ideally suited to database intensive applications such as sites and monuments inventory work (*e.g.* '*identify and plot all Woodland period burial sites with evidence for ochre and log coverings*').

Many GIS have facilities for the storage and manipulation of both types of data model, and provide utilities to translate data from raster to vector formats and vice versa. This is becoming increasingly common and offers a flexible way of manipulating spatial data, allowing the user to choose the most appropriate method for the task at hand. As a result, some GIS systems (for example Idrisi and GRASS) are generally regarded as 'raster' systems because they perform the majority of their analysis on raster data even though they have the capability to read and write vector data files. Similarly Arc/Info, which has been regarded

primarily as a 'vector' GIS system, contains powerful modules for creating and manipulating data as rasters. It is also fair to say that a whiff of 'straw man' pervades this sub-section of the text insofar as the question itself is rapidly becoming historical. Most modern GIS have both raster and vector capabilities within them and all spatial databases will comprise a mix of vector and raster data layers.

2.10 A NOTE ON THEMATIC MAPPING

As mentioned earlier, the use of thematic layers makes a lot of conceptual sense, particularly in the management of data. Having discussed in some detail spatial data models, some additional benefits of such an approach should have become obvious. In a vector-GIS the thematic layer structure enables us to overcome any limitations that might arise as a result of having to adhere to specific topological types (point, line or area) in any given layer. In raster systems it allows us to overcome the inherent restriction of only being able to represent a single attribute in any given theme or image.

2.11 CONCLUSION

There are two spatial data models in widespread use within archaeology. Vector systems enable features to be represented using points, lines and areas, and provide a flexible link to related databases of attribute information. Raster systems offer a consistent grid-based mechanism for the recording of attributes over space. Any spatial database created within a GIS will contain a judicious blend of both vector and raster data layers.

The major implication this has for archaeology is that the representation of the spatial form and extent of any archaeological entity has to conform to one of these schema. With some objects, for example a single artefact find in an excavation trench, a single point or—depending upon the resolution—a single cell may be adequate. But what of the partial remains of an enclosure, defined by an interrupted ditch? Does it represent an area feature or block of contiguous cells? If so, how do we cope with the uncertainty and permeability of the defining boundary?

The scale at which data is acquired is also important. A 13th century AD Pueblo or Roman villa may be represented by little more than a dot on a map at a scale of 1:50,000, yet comprise a complex series of areas and lines when transcribed from an excavation plan, geophysical plot or aerial photograph at a scale of 1:500 or better. Problems such as the latter are not restricted to archaeological data — for example, whether a major river be represented as a line or area — and fall within the much broader topic of cartographic generalisation. Problems such as these will be discussed in more detail in the next chapter, when we begin to look in detail at issues relating to the overall design and management of an archaeological spatial database.

2.12 FURTHER INFORMATION

Most general textbooks on GIS cover the material we have discussed in this chapter, and we would recommend Burrough (1986), now substantially re-written as Burrough and McDonnell (1998) (see particularly Chapters 2 and 3). A very comprehensive discussion of spatial data and the spatial database can also be found in Worboys (1995) where many more methods of raster and vector storage are discussed. This also contains detailed discussion of database theory, fundamental spatial concepts, models for spatial information and data structures.

Guidelines for the design of archaeological spatial databases can be found in the AHDS *GIS Guide to Good Practice* (Gillings and Wise 1998), particularly section 4 '*Structuring, Organising and Maintaining Information*'.