

# The core of GIScience

a process-based approach



ITC Educational textbook series

**UNIVERSITY OF TWENTE**

FACULTY OF GEO-INFORMATION SCIENCE AND EARTH OBSERVATION



## Chapter 9

# Analysis and Process modelling

*Rolf de By  
Otto Huisman  
Menno-Jan Kraak*

### Introduction

We know from preceding chapters that the analytical capabilities of a GIS make use of spatial and non-spatial (attribute) data to answer questions and solve problems that are of spatial relevance. We now make a distinction between analysis (or analytical operations) and analytical models (often referred to as “modelling”). And by analysis we actually mean only a subset of what is usually implied by the term: we do not specifically deal with advanced statistical analysis (such as cluster detection or geostatistics), which is beyond the scope of this textbook.

All knowledge of the world is based on models of some kind—whether idiosyncratic interpretations, culturally-based stereotypes or complex equations that describe a physical phenomenon. We have already seen in Chapter 7 that there are different types of models and that the word itself has different meanings in different contexts. Section 8.1 notes that even spatial data are themselves a kind of “model” of some part of the real world. Here, in this chapter, we focus on analytical functions that form the building blocks for application models. Section 9.1 presents a framework that defines four main classes of these functions. Each class is then discussed, with the help of examples, in detail in subsequent sections. The aim is to make clear to the reader that these operations can be combined in various ways to perform increasingly complex analyses.

### 9.1 Classification of analytical GIS capabilities

There are many ways to classify the analytical functions of a GIS. The classification presented here is essentially the one put forward by Aronoff [4], which makes the following distinctions:

- Classification functions allow the assignment of features to a class on the basis of attribute values or attribute ranges (definition of data patterns). On the basis of reflectance characteristics found in a raster, pixels may be classified as representing different crops, e.g. potato or maize.
- Retrieval functions allow selective searching of data. We might, for example, retrieve all agricultural fields on which potato is grown.
- Generalization functions allow different classes of objects with common characteristics to be joined to form a higher-level (generalized) class. For example, we might generalize fields where potato or maize, and possibly other crops, are grown as “food-produce fields”.
- Measurement functions allow the calculation of distances, lengths or areas. All functions in this category are performed on a single (vector or raster) data layer, often using the associated attribute data.

More detail can be found in Section 9.2. The following three function types belong to the above classification functions.

### Overlay functions

Overlay functions is one of the most frequently used functions in a GIS application. They combine two (or more) spatial data layers, comparing them position by position and treating areas of overlap—and of non-overlap—in distinct ways. Many GISs support overlays through an algebraic language, expressing an overlay function as a formula in which the data layers are the arguments. In this way, we can find:

- those potato fields on clay soils (select the “potato” cover in the crop-data layer and the “clay” cover in the soil-data layer and perform an intersection of the two areas found);
- those fields in which potato or maize is the crop (select both areas of “potato” and “maize” cover in the crop-data layer and determine their union);
- those potato fields not on clay soils (use a difference operator of areas with “potato” cover with the areas having clay soil);
- those fields that do not have potato as a crop (determine the complement of the potato areas).

Examples are provided in Section 9.3.

### Neighbourhood functions

Neighbourhood functions evaluate the characteristics of an area surrounding a feature’s location. A neighbourhood function “scans” the neighbourhood of the given feature(s), and performs a computation on it(them)

- Search functions allow the retrieval of features that fall within a given search window. This window may be a rectangle, circle or polygon.
- Buffer zone generation (or buffering) is one of the best-known neighbourhood functions. It determines a spatial envelope (buffer) around a given feature or features. The buffer created may have a fixed width or a variable width that depends on characteristics of the area.

- Interpolation functions predict unknown values using the known values at nearby locations. This typically occurs for continuous fields, e.g. elevation, when the data actually stored does not provide a direct answer for the location/locations of interest.
- Topographic functions determine characteristics of an area by also looking at the immediate neighbourhood. Typical examples are slope computations on digital terrain models (i.e. continuous spatial fields). The slope at a location is defined as the plane tangent to the topography at that location. Various computations can be performed, such as:
  - determination of slope angle;
  - determination of slope aspect;
  - determination of slope length;
  - determination of contour lines.

These functions are discussed more fully in Section 9.4.

### Connectivity functions

Connectivity functions work on the basis of networks, including road networks, water courses in coastal zones, and communication lines in mobile telephony. These networks represent spatial linkages between features. Main functions of this type include:

- Contiguity functions for evaluating a characteristic of a set of connected spatial units. One can think, here, of the search for a contiguous area of forest of a certain size and shape in a satellite image.
- Network analytic functions for computing related to connected line features that make up a network. The network may consist of roads, public transport routes, high-voltage power lines, or other forms of transportation infrastructure. Analysis of such networks may entail shortest path computations (in terms of distance or travel time) between two points in a network for routing purposes. Other forms are to find all points reachable within a given distance or duration from a start point for allocation purposes, or determination of the capacity of the network for transportation between an indicated source location and sink location.
- Visibility functions also fit in this list because they are used to compute the points visible from a given location (viewshed modelling or viewshed mapping) using a digital terrain model.

For more details see Section 9.5.

## 9.2 Measurement, retrieval and classification

### 9.2.1 Measurement

Geometric measurement on spatial features includes counting, distance and area size computations. This subsection discusses such measurements in a planar spatial reference system. We limit ourselves to geometric measurements and do not include attribute data measurement, which is typically performed in a database query language. Measurements on vector data are more advanced (and thus also more complex) than those on raster data.

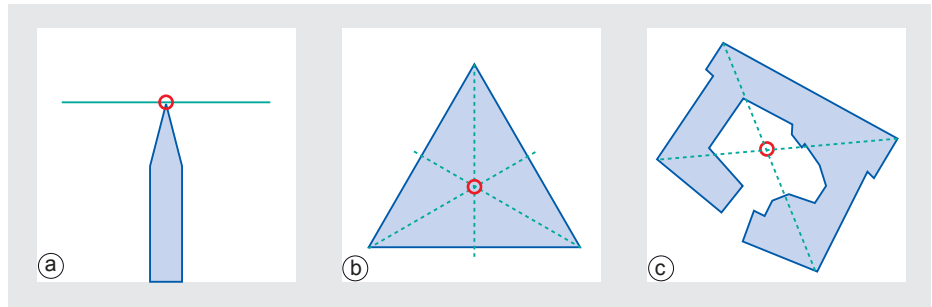


### Measurements on vector data

The primitives of vector data sets are the point, (poly)line and polygon. Related geometric measurements are location, length, distance and area size. Some of these are geometric properties of a feature in isolation (location, length, area size); others (distance) require two features to be identified.

**Figure 9.1**

The centroid: (a) the centroid of an object can be visualized as the point on which the object would balance when placed on the tip of a pencil, (b) the centroid of a polygon, (c) the centroid of the ITC building is actually outside the building (in the courtyard).



centroid

The location property of a vector feature is always stored by the GIS: a single coordinate pair for a point, or a list of pairs for a polyline or polygon boundary. Occasionally, there is a need to obtain the location of the centroid of a polygon (see some examples in Figure 9.1); some GISs store these also, while others compute them on the fly.

Length is a geometric property associated with polylines, by themselves or in their function as polygon boundaries. It can obviously be computed by the GIS—as the sum of lengths of the constituent line segments—but quite often it is also stored with the polyline.

Area size is associated with polygon features. Again, it can be computed, but it is usually stored with the polygon as an extra attribute value. This speeds up the computation of other functions that require area size values. The attentive reader will have noted that all of the above “measurements” do not actually require computation but only retrieval of stored data.

Measurement of distance between two features is another important function. If both features are points, say  $p$  and  $q$ , the computation in a Cartesian spatial reference system is given by the well-known Pythagorean distance function:

$$\text{dist}(p, q) = \sqrt{(x_p - x_q)^2 + (y_p - y_q)^2}$$

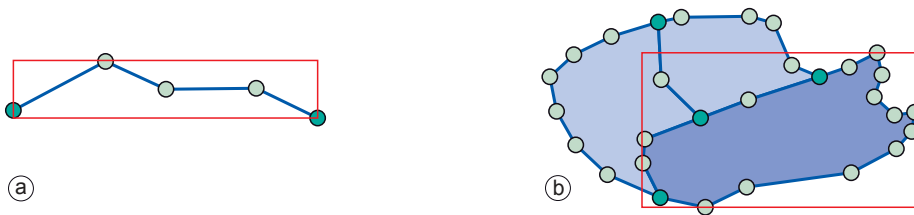
If one of the features is not a point, or both are not, we must be precise in defining what we mean by their distance. All these cases can be summarized as computation of the minimal distance between a location occupied by the first feature and a location occupied by the second feature. This means that features that intersect or meet, or when one contains the other, have a distance of 0. We leave a further case analysis, including polylines and polygons, to the reader as an exercise. It is not possible to store all distance values for all possible combinations of two features in any reasonably-sized spatial database. As a result, the system must compute on-the-fly whenever a distance computation request is made.

Another geometric measurement used by GISs is the minimal bounding box computation. It applies to polylines and polygons and determines the minimal rectangle—with sides parallel to the axes of the spatial reference system—that covers the feature. This is illustrated in Figure 9.2. Bounding box computation is an important support function of GISs: for instance, if the bounding boxes of two polygons do not overlap, we know the polygons cannot possibly intersect each other. Since polygon intersection is

minimal bounding box

a complicated function but bounding box computation is not, a GIS will always first apply the latter as a test to see whether it must do the first.

For practical purposes, it is important to be aware of the unit of measurement that applies to the spatial data layer that one is working on. This is determined by the spatial reference system that has been defined for it during data preparation.



**Figure 9.2**  
The minimal bounding box of  
(a) a polyline, and (b) a  
polygon

A common use of area size measurements is when one wants to sum up the area sizes of all polygons belonging to some class. This class could be crop type: what is the size of the area covered by potatoes? If our crop classification is in a stored data layer, the computation would include (a) selecting the potato areas, and (b) summing up their (stored) area sizes. Clearly, little geometric computation is required in the case of stored features. This is not the case when we are interactively defining our vector features for GIS use and we want measurements to be performed on these interactively defined features. Then the GIS will have to perform complicated geometric computations.

### Measurements on raster data

Measurements on raster data layers are simpler because of the regularity of the cells. The area size of a cell is constant and is determined by the cell resolution. Horizontal and vertical resolution may differ, but typically they do not. Together with the location of what is called an anchor point, this is the only geometric information stored with the raster data, so all other measurements by the GIS are computed. The anchor point is fixed by convention to be the lower-left (or sometimes upper-left) location of the raster.

raster's anchor point

Location of an individual cell derives from the raster's anchor point, the cell resolution, and the position of the cell in the raster. Again, there are two conventions: the cell's location can be its lower-left corner, or the cell's midpoint. These conventions are set by the software in use, and in cases of data of low resolution it becomes more important to be aware of them. The area size of a selected part of the raster (a group of cells) is calculated as the number of cells multiplied by the cell-area size. The distance between two raster cells is the standard distance function applied to the locations of their respective midpoints; obviously the cell resolution has to be taken into account. Where a raster is used to represent line features as strings of cells through the raster, the length of a line feature is computed as the sum of distances between consecutive cells.

### 9.2.2 Spatial selection queries

When exploring a spatial data set, the first thing one usually wants to do is select certain features, to (temporarily) restrict the exploration. Such selections can be made on geometric/spatial grounds or on the basis of attribute data associated with the spatial features. We discuss both techniques in the following two subsections.

### Interactive spatial selection

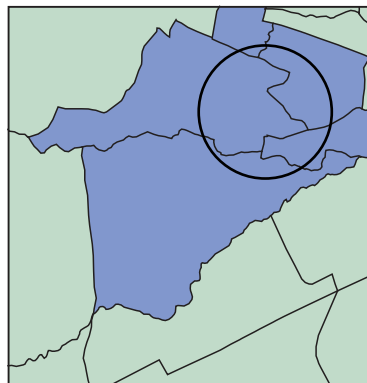
interactive selection

In interactive spatial selection, one defines the selection condition by pointing at or drawing spatial objects on the screen display, after having indicated the spatial data layer(s) from which to select features. The interactively defined objects are called the selection objects; they can be points, lines, or polygons. The GIS then selects the features in the indicated data layer(s) that overlap (i.e. intersect, meet, contain, or are contained in; see Figure 8.13) with the selection objects. These become the selected objects.

As we have seen in Section 8.1, spatial data are usually associated with their attribute data (stored in tables) through a key/foreign key link. Selections of features lead, via these links, to selections on the records. Conversely, selection of records may lead to selection of features.

Figure 9.3

All city wards that overlap with the selection object—here a circle—are selected (left), and their corresponding attribute records are highlighted (right; only part of the table is shown). Data from an urban application in Dar es Salaam, Tanzania.



Area	Perimeter	Ward_id	Ward_name	District	Pop88	Pop92
65420380.0000	41654.940000	1	KUNDUCHI	Kinondoni	22106	27212.00
24813620.0000	30755.620000	2	KAWA	Kinondoni	32854	40443.00
18698500.0000	28403.580000	3	MSASANI	Kinondoni	51225	63058.00
81845610.0000	49645.160000	4	UBUNGO	Kinondoni	47281	58203.00
4486546.000000	13480.130000	5	MANZESE	Kinondoni	59467	73204.00
4999599.000000	10356.850000	6	TANDALE	Kinondoni	58357	71837.00
4102218.000000	8951.096000	7	MWANANYAMALA	Kinondoni	72956	89809.00
3749840.000000	9447.420000	8	KINONDONI	Kinondoni	42301	52073.00
2087509.000000	7502.250000	9	UPANGA WEST	Ilala	9852	11428.00
2268513.000000	9028.788000	10	KIVUKONI	Ilala	5391	6254.00
1400024.000000	6883.288000	11	NDUGUMBI	Kinondoni	32548	40067.00
88896.90000000	4589.110000	12	MAGOMENI	Kinondoni	16938	20851.00
1448370.000000	5651.958000	13	UPANGA EAST	Ilala	11019	12782.00
6214378.000000	14552.080000	14	MABIBO	Kinondoni	43381	53402.00
2496622.000000	7121.255000	15	MAKURUMILA	Kinondoni	54141	66648.00
1262028.000000	4885.793000	16	MZIMUNI	Kinondoni	23989	29530.00
35362240.000000	28976.090000	17	KINYEREZI	Ilala	3044	3531.00
1010613.000000	5393.771000	18	JANGIWANI	Ilala	15297	17745.00
475745.50000000	3043.068000	19	KISUTU	Ilala	8399	9743.00
1754043.000000	7743.187000	20	KIGOGO	Kinondoni	21267	26180.00
29964950.000000	36964.000000	21	KIGAMBONI	Temeke	23203	27658.00
1291479.000000	5187.690000	22	MICHIKICHINI	Ilala	14852	17228.00
720322.10000000	4342.732000	23	MCHAFUKOGE	Ilala	8439	9789.00
9296131.000000	16321.530000	24	TABATA	Ilala	18454	21407.00
483620.70000000	3304.072000	25	KARIAKOO	Ilala	12506	14507.00
3564653.000000	9586.751000	26	BUGURUNI	Ilala	48286	56012.00
2639575.000000	6970.186000	27	ILALA	Ilala	35372	41032.00
912452.80000000	4021.937000	28	GEREZANI	Ilala	7490	8688.00
6735135.000000	13579.590000	29	KURASINI	Temeke	26737	31871.00

Interactive spatial selection answers questions like “What is at . . . ?” In Figure 9.3, the selection object is a circle and the selected objects are the blue polygons; they overlap with the selection object.

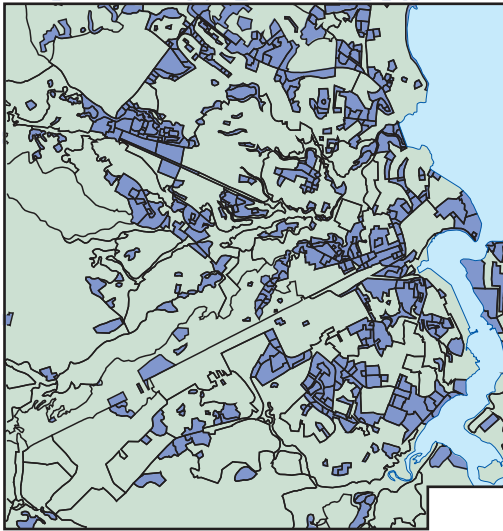
### Spatial selection by attribute conditions

attribute selection

It is also possible to select features by using selection conditions on feature attributes. These conditions are formulated in SQL (if the attribute data reside in a relational database) or in a software-specific language (if the data reside in the GIS itself). This type of selection answers questions like “Where are the features with . . . ?”

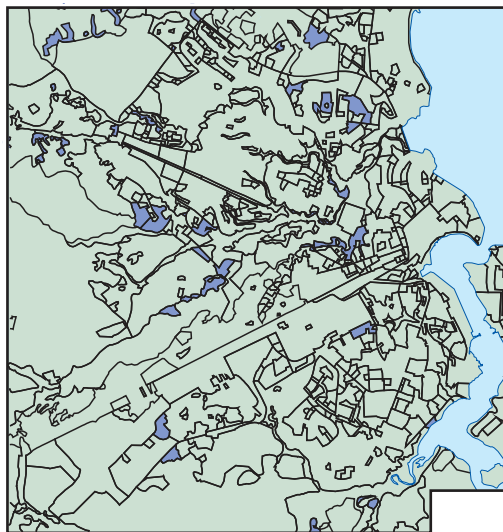
Figure 9.4 shows an example of selection by attribute condition. The query expression is  $Area < 400,000$ , which can be interpreted as “Select all areas of land use of which the size is less than 400,000.” The polygons in red are the selected areas; their associated records are also highlighted in blue.

We can use a set of features already selected as the basis for further selection. For instance, if we are interested in land use areas of size less than 400,000 that are of land use type 80, the selected features of Figure 9.4 are subjected to a further condition,  $Land\ Use = 80$ . The result is illustrated in Figure 9.5.



Area	IDs	LandUse
174308.70	2	30
2066475.00	3	70
214582.50	4	80
29313.86	5	80
73328.08	6	80
53303.30	7	80
614530.10	8	20
1637161.00	9	80
156357.40	10	70
59202.20	11	20
83289.59	12	80
225642.20	13	20
28377.33	14	40
228930.30	15	30
986242.30	16	70

**Figure 9.4**  
Spatial selection using the attribute condition Area < 400,000 on land use areas in Dar es Salaam. Spatial features on the left, associated attribute data (in part) on the right.



Area	IDs	LandUse
174308.70	2	30
2066475.00	3	70
214582.50	4	80
29313.86	5	80
73328.08	6	80
53303.30	7	80
614530.10	8	20
1637161.00	9	80
156357.40	10	70
59202.20	11	20
83289.59	12	80
225642.20	13	20
28377.33	14	40
228930.30	15	30
986242.30	16	70

**Figure 9.5**  
Further spatial selection from the features already selected in Figure 9.4 using the additional condition Land Use = 80 on land use areas. Note that fewer features are now selected.

### Combining attribute conditions

The combination of conditions just dealt with in the previous subsection is fairly common in practice.

When multiple criteria have to be used for selection, we need to carefully express all of these in a single composite condition. The tools for this come from a field of mathematical logic known as propositional calculus.

The example of the previous subsection made use of simple atomic conditions such as Area < 400,000 and Land Use = 80. Atomic conditions use a predicate symbol, such as < (less than) or = (equals). Other possibilities are <= (less than or equal), > (greater than), >= (greater than or equal) and <> (does not equal). Any of these symbols is combined with an expression on the left and one on the right. For instance, Land Use <> 80 can be used to select all areas with a land use class different from 80. Expressions are either constants like 400,000 and 80, attribute names like Area and

atomic condition

Land Use, or possibly composite arithmetic expressions like  $0.15 \times \text{Area}$ , which would compute 15% of the area size.

Atomic conditions can be combined into composite conditions using logical connectives. The most important ones are AND, OR, NOT and the bracket pair (. . .). If we write a composite condition such as  $\text{Area} < 400,000 \text{ AND Land Use} = 80$ , we can use it to select areas for which both atomic conditions hold true. This is the meaning of the AND connective. If we had written  $\text{Area} < 400,000 \text{ OR Land Use} = 80$  instead, the condition would have selected areas for which either condition holds, so effectively those with an area size less than 400,000, but also those with land use class 80. (Included, of course, will be areas for which both conditions hold.)

composite condition

The NOT connective can be used to negate a condition. For instance, the condition NOT ( $\text{Land Use} = 80$ ) would select all areas with a different land use class than 80. (Clearly, the same selection can be obtained by writing  $\text{Land Use} \neq 80$  but this is not the point.) Finally, brackets can be applied to force grouping amongst atomic parts of a composite condition. For instance, the condition  $(\text{Area} < 30,000 \text{ AND Land Use} = 70) \text{ OR } (\text{Area} < 400,000 \text{ AND Land Use} = 80)$  will select areas of class 70 less than 30,000 in size, as well as class 80 areas less than 400,000 in size.

spatial selection

### Spatial selection using topological relationships

Various forms of topological relationship between spatial objects were discussed in the subsection in Chapter 8 titled Topology and spatial relationships (page 249). These relationships can be useful to select features as well. The steps to be carried out are:

1. select one or more features as the selection objects; and
2. apply a chosen spatial relationship function to determine the selected features that have that relationship with the selection objects.

### Selecting features that are inside selection objects

This type of query uses the containment relationship between spatial objects. Obviously, polygons can contain polygons, lines or points, and lines can contain lines or points, but no other containment relationships are possible.

Figure 9.6a illustrates a containment query. Here, we are interested in finding the location of medical clinics in Ilala District. We first selected all areas of Ilala District using the technique of selection by attribute condition  $\text{District} = \text{"Ilala"}$ . Then, these selected areas were used as selection objects to determine which medical clinics (as point objects) were within them.

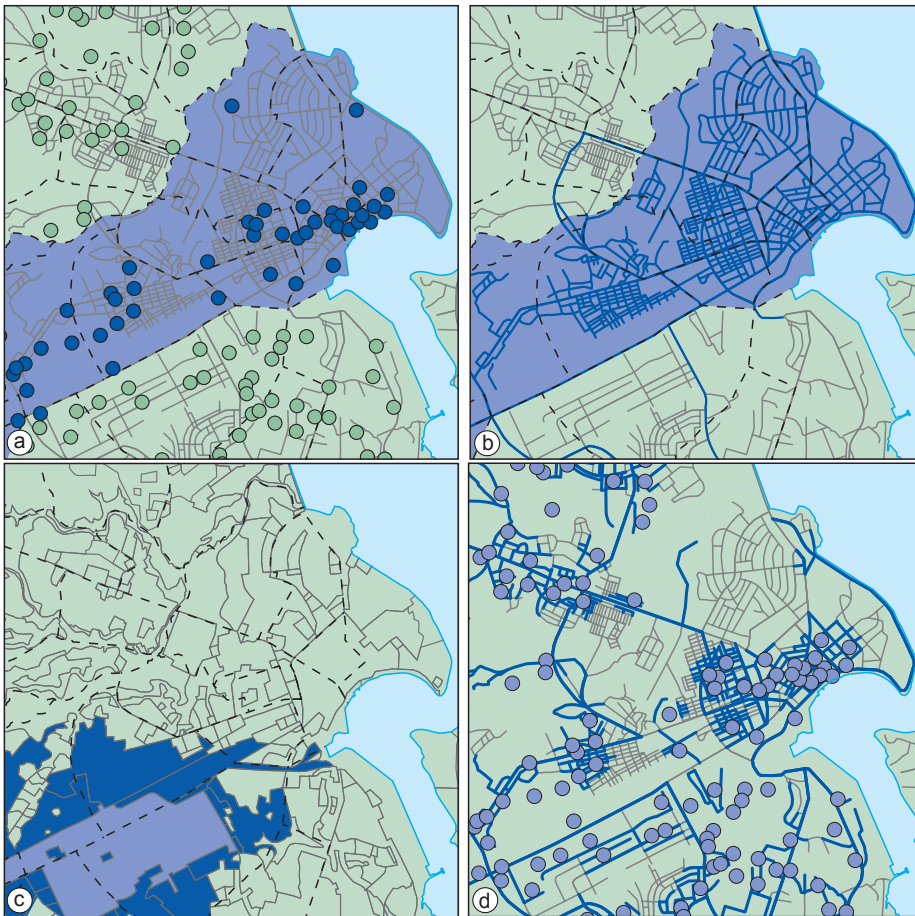
### Selecting features that intersect

The intersect operator identifies features that are not disjoint in the sense of Figure 8.13, but now extended to include points and lines. Figure 9.6b provides an example of spatial selection using the intersect relationship between lines and polygons. All roads intersecting Ilala District were selected.

### Selecting features adjacent to selection objects

Adjacency is the meet relationship described in the subsection Topology and spatial relationships of Chapter 8 (page 249). The adjacency operator identifies those features that share boundaries and, therefore, applies only to line and polygon features. Figure 9.6c illustrates a spatial adjacency query. We want to select all parcels adjacent to an industrial area. The first step is to select the industrial area (in dark blue) and then





**Figure 9.6**  
Spatial selection using topological relationships: (a) selection of medical clinics (dark purple) located inside the selected district (violet); (b) selection of roads (dark purple) that are partially located in the selected district (violet); (c) selection of the areas (dark purple), adjacent to the selected urban industrial area (violet); (d) selection of roads within 200 m of a clinic.

apply the adjacency function to select all land use areas (delineated in light blue) that are adjacent to it.

### Selecting features based on their distance

One may also want to use the distance function of a GIS as a tool for selecting features. Such selections can be searches for features within a given distance of the selection objects, at a given distance, or even beyond a given distance. There is a whole range of selections of this type. For example:

- Which medical clinics are within 2 km of a selected school? (Information needed for the school's emergency procedures.)
- Which roads are within 200 m of a medical clinic? (These roads must have a high priority for road maintenance.)

Figure 9.6d illustrates a spatial selection using distance. In this case, we executed the selection of the second example directly above. Our selection objects were all clinics, and we selected the roads that pass within 200 m of a clinic. For situations in which we know the distance criteria to use—for selections within, at or beyond that distance value—the GIS has many (straightforward) computations to perform. Things

become more complicated if the distance selection condition involves the word “nearest” or “farthest”. The reason is that not only must the GIS compute distances from a selection object A to all potentially selectable features F, but also that it must find the feature F that is nearest to (or farthest away from) object A. So, this requires an extra computational step to determine minimum (maximum) values. Most GIS packages support this type of selection, though the mechanics (“the buttons to use”) differ among packages.

### Afterthought on selecting features

So far we have discussed a number of different techniques for selecting features. We have also seen that selection conditions on attribute values can be combined using logical connectives such as AND, OR and NOT. Other techniques of selecting features can also usually be combined. Any set of selected features can be used as the input for a subsequent selection procedure. This means, for instance, that we can select all medical clinics first, then identify roads within 200 m of them, then select from those only the major roads, then select the nearest clinics to these remaining roads as the ones that should receive our financial support for maintenance. In this way, we are combining various techniques of selection.

### 9.2.3 Classification

Classification is a technique for purposely removing detail from an input data set in the hope of revealing important patterns (of spatial distribution). In the process, we produce an output data set, so that the input set can be left intact. This output set is produced by assigning a characteristic value to each element in the input set, which is usually a collection of spatial features that could be raster cells or points, lines or polygons. If the number of characteristic values in the output set is small in comparison to the size of the input set, we have classified the input set.

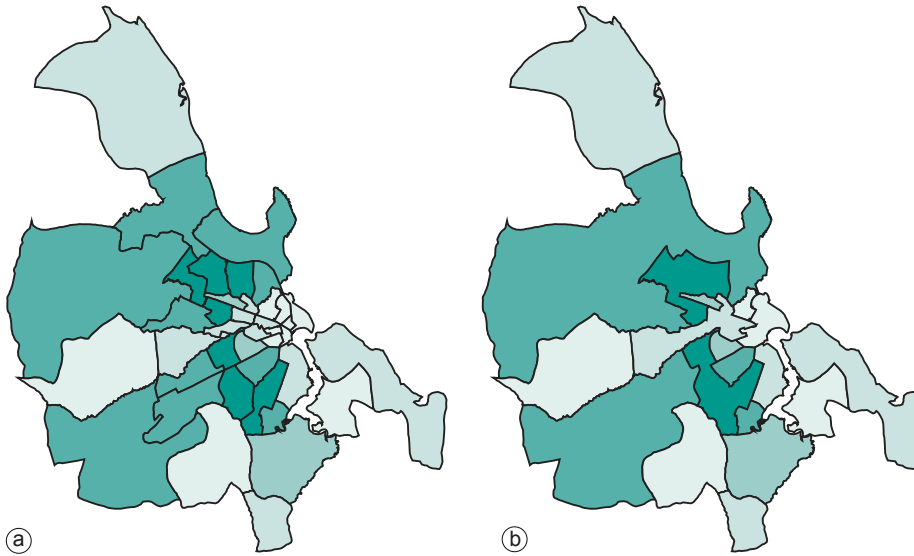
The pattern that we are looking for may be the distribution of household income in a city. In that case, household income is called the classification parameter. If we know for each ward in the city the associated average income, we have many different values. Subsequently, we could define five different categories (or classes) of income: “low”, “below average”, “average”, “above average” and “high”, and provide value ranges for each category. If these five categories are mapped using a sensible colour scheme, this may reveal interesting information. Figure 9.7 illustrates how this has been done in two ways for Dar es Salaam.

The input data set may, itself, have been the result of a classification. In such cases we refer to the output data set as a reclassification. For example, we may have a soil map that shows different soil type units and we would like to show the suitability of units for a specific crop. In this case, it is better to assign to the soil units an attribute of suitability for the crop. Since different soil types may have the same crop suitability, a classification may merge soil units of different type into the same category of crop suitability.

In classification of vector data, there are two possible results. In the first, the input features may become the output features in a new data layer, with an additional category assigned. In other words, nothing changes with respect to the spatial extents of the original features. Figure 9.7a illustrates this first type of output. A second type of output is obtained when adjacent features of the same category are merged into one bigger feature. Such a post-processing function is called spatial merging, aggregation or dissolving. An illustration of this second type is found in Figure 9.7b. Observe that this type of merging is only an option in vector data, as merging cells in an output raster on the basis of a classification makes little sense. Vector data classification can

reclassification

aggregation and merging



**Figure 9.7**  
Two classifications of average annual household income per ward in Dar es Salaam, Tanzania. Higher income areas are in darker greens. Five categories were identified. (a) with original polygons left intact; (b) with original polygons merged when in the same category. The data used for this illustration are *not* factual.

be performed on point sets, line sets or polygon sets; the optional merge phase only makes sense for lines and polygons. In the following two subsections we discuss two kinds of classification: user-controlled and automatic classification.

Household income range	New category value
391–2474	1
2475–6030	2
6031–8164	3
8165–11587	4
11588–21036	5

**Table 9.1**  
Classification table used in Figure 9.7.

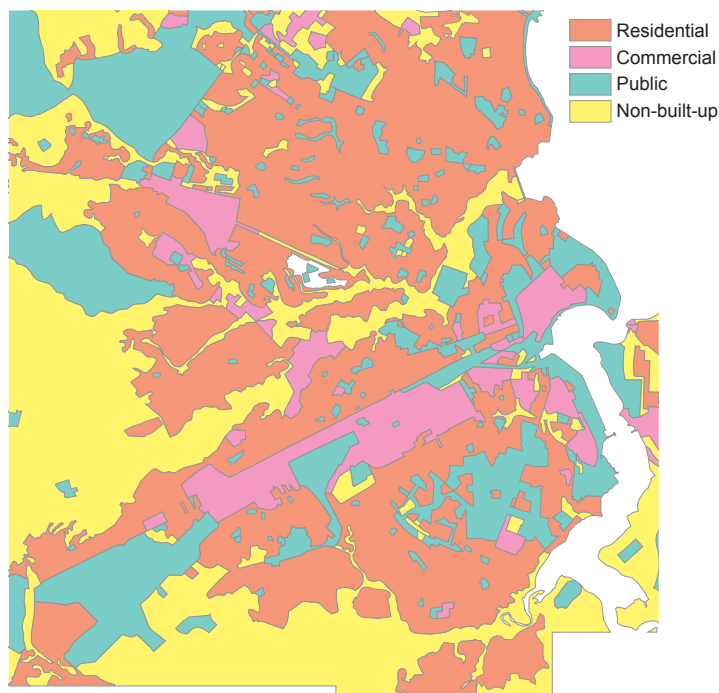
### User-controlled classification

In user-controlled classification, a user selects the attribute(s) that will be used as the classification parameter(s) and defines the classification method. The latter involves declaring the number of classes, as well as the correspondence between the old attribute values and the new classes. This is usually done via a classification table. The classification table used for Figure 9.7 is displayed in Table 9.1. It is rather typical for cases in which the parameter domain used is continuous (e.g. household income). Then, the table indicates value ranges to be mapped to the same category. Note that category values are ordinal data, described in the subsection Geographic fields on (page 239).

Another case exists when the classification parameter is nominal or at least discrete. Such an example is given in Figure 9.8. We must also define the data format of the output as a spatial data layer, which will contain the new classification attribute. The data type of this attribute is always categorical, i.e. integer or string, no matter what the data type of the attribute(s) from which the classification was obtained.

Sometimes, one may want to classify only a selection of features. In such cases, there are two options for the features that are not selected. One option is to keep their original values, while the other is to assign a null value to them in the output data set. A null value is a special value that means that no applicable value is present.

classification table



**Figure 9.8**  
An example of a classification on a discrete parameter, namely land use unit in city of Dar es Salaam, Tanzania.

Care must be taken to deal with these values correctly, both in computations and in visualization.

### Automatic classification

User-controlled classifications require a classification table or user interaction. GIS software can also perform automatic classification, in which a user only specifies the number of classes in the output data set. The system automatically determines the class break points. The two main techniques of determining break points being used are the equal interval technique and the equal frequency technique.

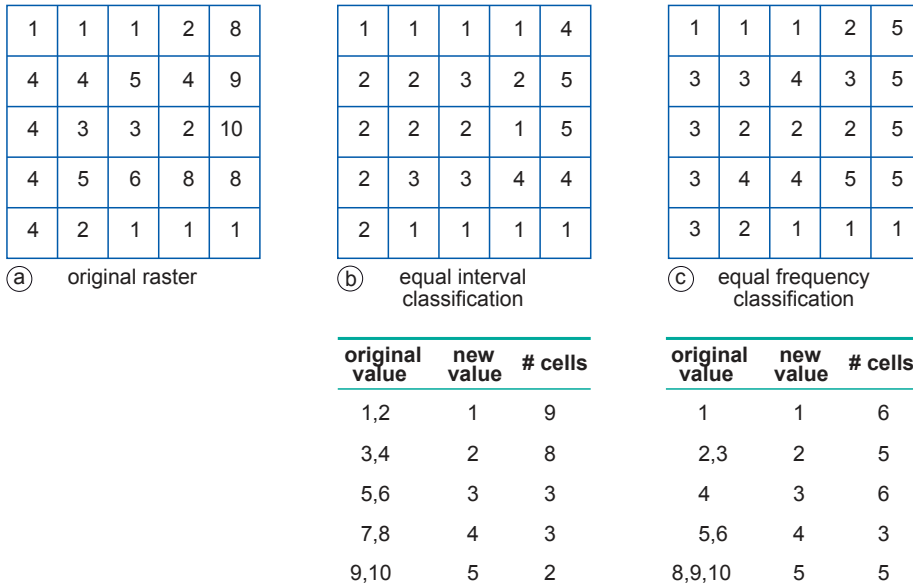
### Equal interval technique

The minimum and maximum values  $v_{min}$  and  $v_{max}$  of the classification parameter are determined and the (constant) interval size for each category is calculated as  $(v_{max} - v_{min})/n$ , where  $n$  is the number of classes chosen by the user. This classification is useful in that it reveals the distribution pattern, as it determines the number of features in each category.

### Equal frequency technique

This technique is also known as quantile classification. The objective is to create categories with roughly equal numbers of features per category. The total number of features is determined first, then, based on the required number of categories, the number of features per category is calculated. The class break points are then determined by counting off the features in order of classification parameter value.

Both techniques are illustrated on a small  $5 \times 5$  raster in Figure 9.9.



**Figure 9.9**  
 Example of two automatic classification techniques: (a) the original raster with cell values; (b) classification based on equal intervals; (c) classification based on equal frequencies. Below, the respective classification tables, with a tally of the number of cells involved.

### 9.3 Overlay functions

In the previous section, various techniques for measuring and selecting spatial data were discussed. We also discussed the generation of a new spatial data layer from an old layer using classification. Now, in this section, we look at techniques for combining two spatial data layers and producing a third layer from them. The binary operators that we discuss are known as spatial overlay operators. Vector overlay operators will be dealt with first, followed by raster operators.

Standard overlay operators take two input data layers and assume that they are georeferenced in the same system and that they overlap in the study area. If either of these requirements is not met, the use of an overlay operator is pointless. The principle of spatial overlay is to compare the characteristics of the same location in both data layers and to produce a result for each location in the output data layer. The specific result to produce is determined by the user. It might involve a calculation or some other logical function to be applied to every area or location. With raster data, as we shall see, these comparisons are carried out between pairs of cells, one from each input raster. With vector data, the same principle of comparing locations applies but the underlying computations rely on determining the spatial intersections of features from each input layer.

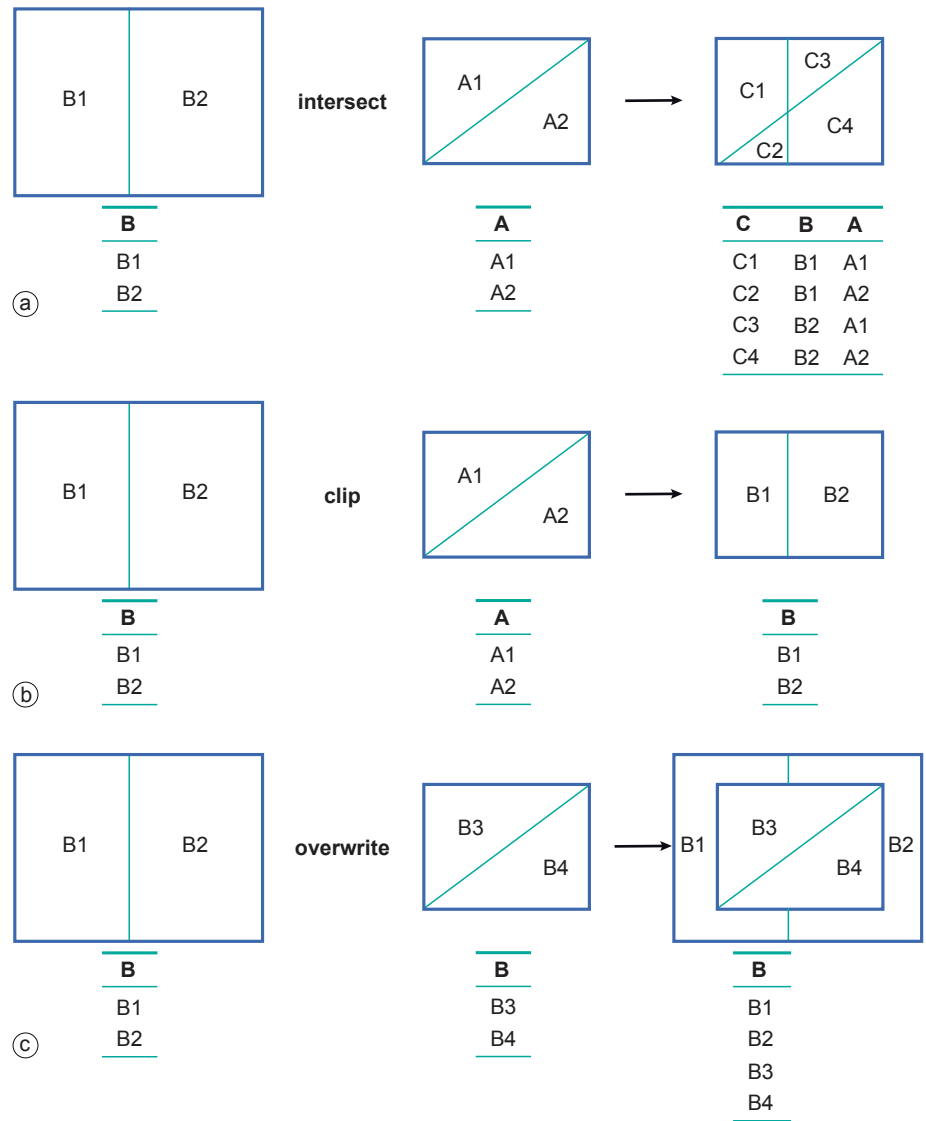
#### 9.3.1 Vector overlay operators

In the vector domain, overlay is computationally more demanding than in the raster domain. Here we will only discuss overlays from polygon data layers, but do note that most of the ideas also apply to overlay operations with point or line data layers. The standard overlay operator for two layers of polygons is the polygon intersection operator. It is fundamental, as many other overlay operators proposed in the literature or implemented in systems can be defined in terms of it. The principles are illustrated in Figure 9.10. The result of this operator is the collection of all possible polygon intersections; the attribute table result is a join—in the relational database sense of Section 8.4—of the two input attribute tables. This output attribute table only con-

intersect



tains one tuple for each intersecting polygon found, which explains why we call this operator a spatial join. A more practical example is provided in Figure 9.11a, which was produced by polygon intersection of the ward polygons with land use polygons classified as in Figure 9.8. This allows us to select the residential areas in Ilala District.



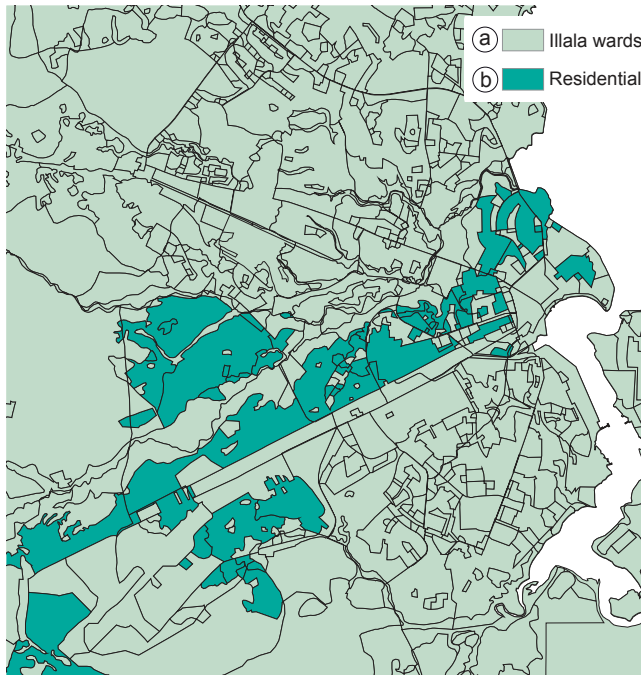
**Figure 9.10**  
 The polygon overlay operators: (a) polygon intersection, (b) polygon clipping, the left hand polygon (*B*) is clipped by polygon *A*, (c) polygon overwrite.

clip

overwrite

Three polygon overlay operators are illustrated in Figure 9.10. The second is known as the polygon clipping operator. It takes a polygon data layer and restricts its spatial extent to the generalized outer boundary obtained from all (selected) polygons in a second input layer. Besides this generalized outer boundary, no other polygon boundaries from the second layer play a role in the result.

The third overlay operator illustrated in Figure 9.10 is referred to as polygon overwrite. The result of this binary operator is a polygon layer with the polygons of the first layer, except where polygons exist in the second layer, as these take priority.



**Figure 9.11**  
The residential areas of Illala District, obtained from polygon intersection. Input for the polygon intersection operator were (a) a polygon layer with all Illala wards, (b) a polygon layer with the residential areas, as classified in Figure 9.8.

Most GISs do not force the user to apply overlay operators to the full polygon data set. One is allowed to first select relevant polygons in the data layer and then use the selected set as an operator argument. The fundamental operator of all these is polygon intersection. The other operators can be defined in terms of it, usually in combination with polygon selection and/or classification. For instance, the polygon overwrite of  $A$  by  $B$  can be defined as polygon intersection between  $A$  and  $B$ , followed by a (well-chosen) classification that prioritizes polygons in  $B$ , followed by a merge. (The reader is asked to verify this.) Vector overlays are usually also defined for point or line data layers. Their definition parallels the definitions of operators discussed above. Different GISs use different names for these operators, so it is advisable to carefully check the documentation before applying any of these operators.

### 9.3.2 Raster overlay operators

Vector overlay operators are useful but geometrically complicated, and this sometimes results in poor operator performance. Raster overlays do not suffer from this disadvantage, as most of them perform their computations cell by cell, and thus they are fast. GISs that support raster processing—as most do—usually have a language to express operations on rasters. These languages are generally referred to as map algebra [111] or, sometimes, raster calculus. They allow a GIS to compute new rasters from existing ones, using a range of functions and operators. Unfortunately, not all implementations of map algebra offer the same functionality. The discussion below is to a large extent based on general terminology; it attempts to illustrate the key operations using a logical, structured language. Again, the syntax often varies among different GIS software packages.

When producing a new raster we must provide a name for it, and define how it is to be computed. This is done in an assignment statement of the following format:

Output raster name := Map algebra expression.

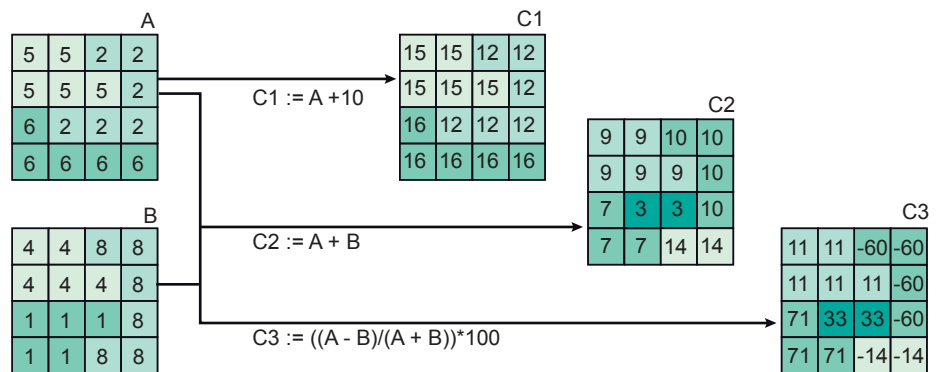
map algebra

The expression on the right is evaluated by the GIS, and the raster in which it results is then stored under the name on the left. The expression may contain references to existing rasters, operators and functions; the format is made clear in each case. The raster names and constants that are used in the expression are called its operands. When the expression is evaluated, the GIS will perform the calculation on a pixel-by-pixel basis, starting from the first pixel in the first row and continuing through to the last pixel in the last row. In map algebra, there is a wide range of operators and functions available, some of which will be discussed in the following subsections.

### Arithmetic operators

Various arithmetic operators are supported. The standard ones are multiplication ( $\times$ ), division ( $/$ ), subtraction ( $-$ ) and addition ( $+$ ). Obviously, these arithmetic operators should only be used on appropriate data values, and, for instance, not on classification values. Other arithmetic operators may include modulo division (MOD) and integer division (DIV). Modulo division returns the remainder of division: for instance, 10 MOD 3 will return 1 as  $10 - 3 \times 3 = 1$ . Similarly, 10 DIV 3 will return 3.

**Figure 9.12**  
Examples of arithmetic map algebra expressions.



Other operators are goniometric: sine (sin), cosine (cos), tangent (tan); and their inverse functions asin, acos, and atan, which return radian angles as real values. Some simple map algebra assignments are illustrated in Figure 9.12. The assignment

$$C1 := A + 10$$

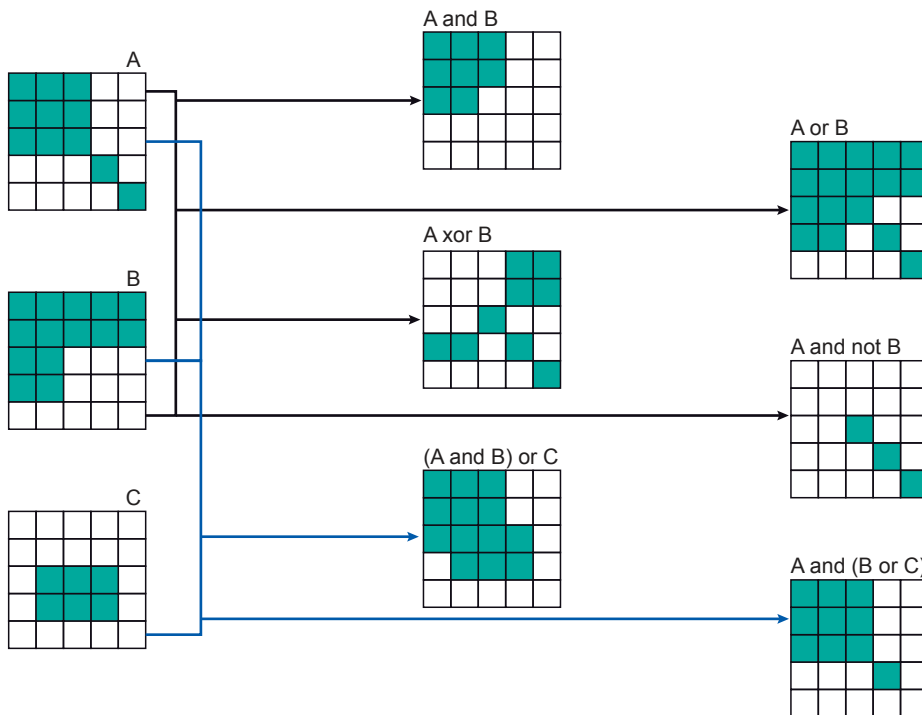
will add a constant factor of 10 to all cell values of raster *A* and store the result as output raster *C1*. The assignment

$$C2 := A + B$$

will add the values of *A* and *B* cell by cell, and store the result as raster *C2*. Finally, the assignment

$$C3 := (A - B)/(A + B) \times 100$$

will create output raster *C3*, as the result of the subtraction (cell by cell, as usual) of *B* cell values from *A* cell values, divided by their sum. The result is multiplied by 100. This expression, when carried out on AVHRR channel 1 (red) and AVHRR channel 2 (near infrared) of NOAA satellite imagery, is known as the NDVI (Normalized Difference Vegetation Index). It has proven to be a good indicator of the presence of green vegetation.



**Figure 9.13**  
Example of logical expression  
in map algebra. Green cells  
represent true values, white  
cells represent false values.

### Comparison and logical operators

Map algebra also allows the comparison of rasters cell by cell. To this end, we may use the standard comparison operators ( $<$ ,  $<=$ ,  $=$ ,  $>=$ ,  $>$  and  $<>$ ) that were introduced in Subsection 9.3.2.

A simple raster comparison assignment is

$$C := A <> B.$$

It will store truth values—either true or false—in the output raster  $C$ . A cell value in  $C$  will be true if the cell's value in  $A$  differs from that cell's value in  $B$ . It will be false if they are the same. Logical connectives are also supported in many raster calculi. We have already seen the connectives of AND, OR and NOT in raster overlay operators (page 327). Another connective that is commonly offered in map algebra is exclusive OR (XOR). The expression  $a \text{ XOR } b$  is true only if either  $a$  or  $b$  is true, but not both.

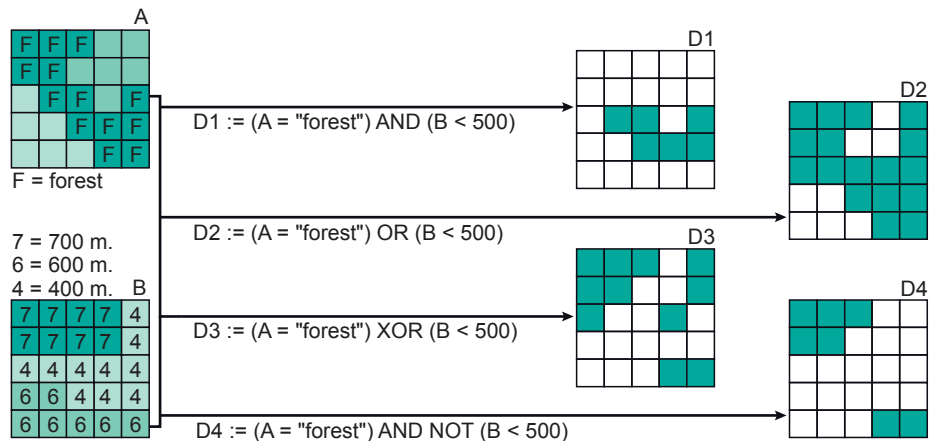
Examples of the use of these comparison operators and connectives are provided in Figure 9.13 and Figure 9.14. The latter figure provides various raster computations in searches for forests at specific elevations. In the figure, raster  $D1$  indicates forest below 500 m,  $D2$  indicates areas below 500 m or that are forests, raster  $D3$  areas that are either forest or below 500 m (but not at the same time), and raster  $D4$  indicates forests above 500 m.

### Conditional expressions

The above comparison and logical operators produce rasters with the truth values true and false. In practice, we often need a conditional expression together with them that allows us to test whether a condition is fulfilled. The general format is:

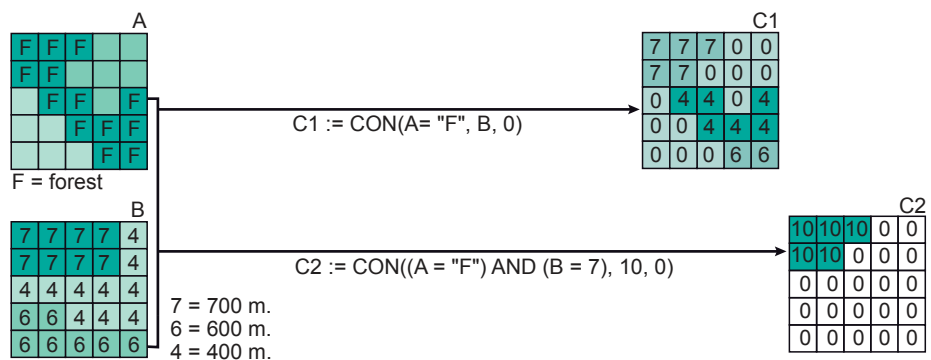
Output raster := CON(condition, then expression, else expression).

**Figure 9.14**  
Examples of complex logical expressions in map algebra. *A* is a classified raster for land use, and *B* holds elevation values.



Here, condition stands for the condition tested, then the expression is evaluated if condition holds, and else the expression is evaluated if it does not hold. This means that an expression such as  $CON(A = \text{"forest"}, 10, 0)$  will evaluate to 10 for each cell in the output raster where the same cell in *A* is classified as forest. For each cell where this is not true, the else expression is evaluated, resulting in 0. Another example is provided in Figure 9.15, showing that values for the *then* expression and the *else* expression can be some integer (possibly derived from another calculation) or values derived from other rasters. In this example, the output raster *C1* is assigned the values of input raster *B* wherever the cells of input raster *A* contain forest. The cells in output raster *C2* are assigned 10 wherever the elevation (*B*) is equal to 7 and the ground cover (*A*) is forest.

**Figure 9.15**  
Examples of conditional expressions in map algebra. Here *A* is a classified raster holding land use data, and *B* is an elevation-value raster.



### Overlays using a decision table

Conditional expressions are powerful tools in cases where multiple criteria must be taken into account. A small example may illustrate this. Consider a suitability study in which a land use classification and a geological classification must be used. The respective rasters are shown on the left-hand side of Figure 9.16. Domain expertise dictates that some combinations of land use and geology result in suitable areas, whereas other combinations do not. In our example, forests on alluvial terrain and grassland on shale are considered suitable combinations, while any others are not.

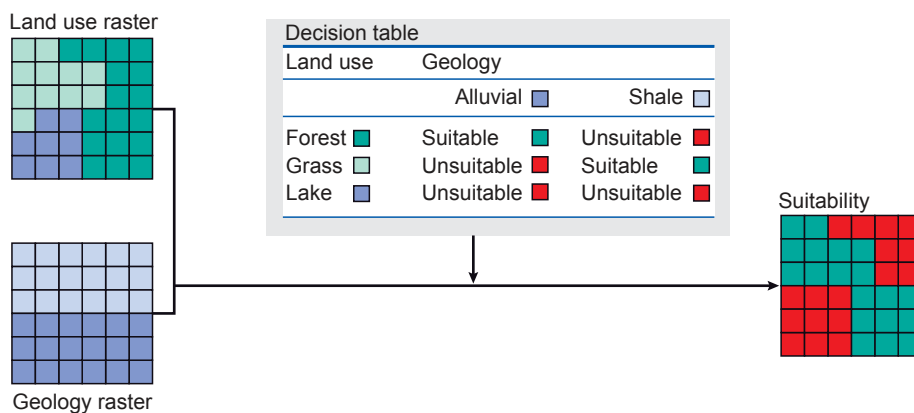
We could produce the output raster of Figure 9.16 with a longish map algebra expression, such as



```
Suitability := CON((Landuse = "Forest" AND Geology = "Alluvial")
OR (Landuse = "Grass" AND Geology = "Shale"),
"Suitable", "Unsuitable")
```

and consider ourselves lucky that there are only two "suitable" cases. In practice, many more cases must usually be covered and, then, writing up a complex CON expression is not an easy task.

To this end, some GISs accommodate setting up a separate decision table that will guide the raster overlay process. This extra table carries domain expertise and dictates which combinations of input raster-cell values will produce which output raster-cell value. This gives us a raster overlay operator using a decision table, as illustrated in Figure 9.16. The GIS will have supporting functions to generate the additional table from the input rasters and to enter appropriate values in the table.



**Figure 9.16**

The use of a decision table in a raster overlay. The overlay would be computed in a suitability study in which land use and geology are important factors. The meaning of values in both input rasters, as well as the meaning of values in the output raster, can be derived from the decision table.

## 9.4 Neighbourhood functions

In the explanation of overlay operators (Subsection 9.3.2), the guiding principle was to compare or combine the characteristic value of a location from two data layers and to do so for all locations. This is what map algebra, for instance, gives us: cell by cell calculations with the results stored in a new raster.

There is another guiding principle in spatial analysis that can be equally useful. The principle in this case is to find out the characteristics of the vicinity, here called neighbourhood, of a location. After all, many suitability questions, for instance, depend not only on what is at a location but also on what is near the location. Thus, the GIS must allow us "to look around locally". To perform neighbourhood analysis, we must:

1. state which target locations are of interest to us and define their spatial extent;
2. define how to determine the neighbourhood for each target; and
3. define which characteristic(s) must be computed for each neighbourhood.

For instance, our target might be a medical clinic. Its neighbourhood could be defined as:

- an area within a radius of 2 km distance as the crow flies; or

- an area within 2 km travelling distance; or
- all roads within 500 m travelling distance; or
- all other clinics within 10 minutes travelling time;
- all residential areas for which the clinic is the closest clinic.

Finally, in the third step we indicate what it is we want to discover about the phenomena that exist or occur in the neighbourhood. This might simply be its spatial extent, but it might also be statistical information such as:

- how many people live in the area;
- what is their average household income;
- are any high-risk industries located in the neighbourhood.

These are typical questions in an urban setting. When our interest is more in natural phenomena, different examples of locations, neighbourhoods and neighbourhood characteristics arise. Since raster data are the more commonly used in this case, neighbourhood characteristics often are obtained via statistical summary functions that compute values such as the average, minimum, maximum and standard deviation of the cells in the identified neighbourhood.

To select target locations, one can use the selection techniques that we discussed in Section 9.3. To obtain characteristics from an eventually-to-be identified neighbourhood, the same techniques apply. So what remains to be discussed here is the proper determination of a neighbourhood. One way of determining a neighbourhood around a target location is by making use of the geometric distance function. Some of these techniques are discussed in Section 9.5. Geometric distance does not take into account direction, but certain phenomena can only be studied by doing so. Think of the spreading of pollution by rivers, groundwater flow or prevailing weather systems. The more advanced techniques for computation of flow and diffusion are discussed in Section 9.5.

geometric distance

Diffusion functions are based on the assumption that the phenomenon in question spreads in all directions, though not necessarily equally easily in each direction. Hence it uses local terrain characteristics to compute local resistances to diffusion. In flow computations, the assumption is that the phenomenon will choose a path of least-resistance and will not spread in all directions. This, as we will see, involves the computation of preferred local direction of spread. Both flow and diffusion computations take local characteristics into account and are, therefore, more easily performed on raster data.

#### 9.4.1 Proximity computations

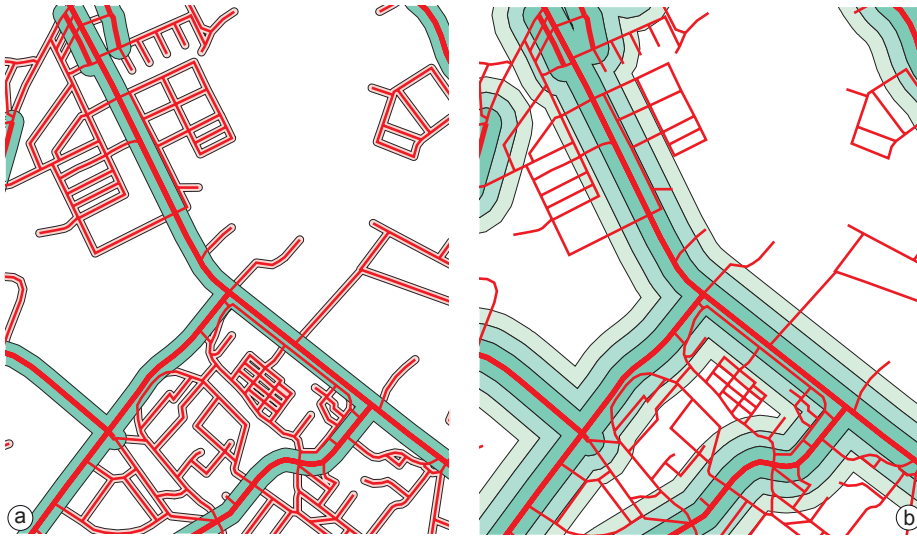
In proximity computations, we use geometric distance to define the neighbourhood of one or more target locations. The most common and useful technique is buffer zone generation. Another technique based on geometric distance that will also be discussed is Thiessen polygon generation.

##### Buffer zone generation

The principle of buffer zone generation is simple: we select one or more target locations and then determine the area around them within a certain distance. In Figure 9.17a, a number of main and minor roads were selected as targets and 75 m and 25 m (respectively) buffers were computed from them.

In some case studies, zoned buffers must be determined, for instance in assessments of the effects of traffic noise. Most GISs support this type of zoned-buffer computation. An illustration is provided in Figure 9.17b.

zoned buffer

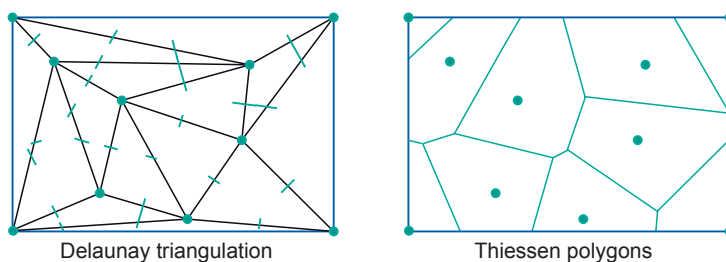


**Figure 9.17**  
Buffer zone generation: (a) around main and minor zones. Different distances were applied: 25 m for minor roads, 75 m for main roads. (b) Zoned buffer zones around main roads. Three different zones were obtained: at 100 m, 200 m and 300 m from the main road.

In vector-based buffer generation, the buffers themselves become polygon features, usually in a separate data layer, that can be used in further spatial analysis. Buffer generation on rasters is a fairly simple function. The target location or locations are always represented by a selection of the raster's cells and geometric distance is defined using cell resolution as the unit. The distance function applied is the Pythagorean distance between the cell centres. The distance from a non-target cell to the target is the minimal distance one can find between that non-target cell and any target cell.

### Thiessen polygon generation

Thiessen polygon partitions make use of geometric distance to determine neighbourhoods. This is useful if we have a spatially distributed set of points as target locations and we want to know the closest target for each location in the study. This technique will generate a polygon around each target location that identifies all those locations that "belong to" that target. We have already seen the use of Thiessen polygons in the context of interpolation of point data. Given an input point set that will be the polygon's midpoints, it is not difficult to construct such a partition. It is even much easier to construct if we already have a Delaunay triangulation for the same input point set (see page 246).



**Figure 9.18**  
Thiessen polygon construction (right) from a Delaunay triangulation; (left): perpendiculars of the triangles form the boundaries of the polygons.

Figure 9.18 repeats the Delaunay triangulation of Figure 8.7b; the Thiessen polygon

partition constructed from it is on the right. The construction first creates the perpendiculars of all the triangle sides; note that a perpendicular of the side of a triangle that connects point A with point B is the imaginary line dividing the area between the area closer to A and the area closer to B. The perpendiculars become part of the boundary of each Thiessen polygon computed by the GIS. (The GIS will work for higher-precision real arithmetic rather than for what is illustrated here.)

### 9.4.2 Computation of diffusion

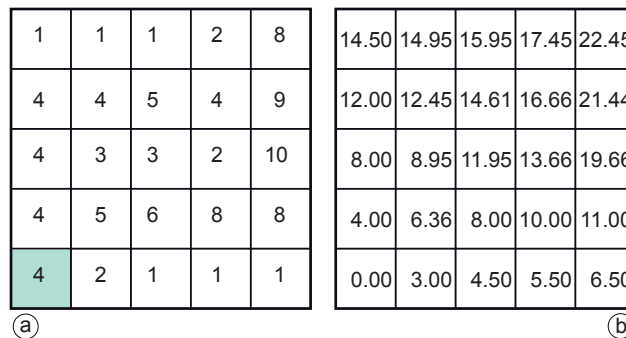
The determination of the neighbourhood of one or more target locations may depend not only on distance—cases of which we have discussed above—but also on direction and differences in the terrain in different directions. This is typically the case when the target location contains “source material” that spreads over time, referred to as *diffusion*. This “source material” may be air, water, soil pollution, commuters exiting a train station, people from an refugee camp that has just been opened up, a natural spring on a hillside, or radio waves emitted from a radio relay station. In all these cases, one will not expect the spread to occur evenly in all directions. There will be local factors that influence the spread, making it easier or more difficult. Many GISs provide support for this type of computation. We will discuss some of the principles here—in the context of raster data.

Diffusion computation involves one or more target locations, which in this context are better called source locations: they are the locations of the source of whatever spreads. The computation also involves a local resistance raster, which for each cell provides a value that indicates how difficult it is for the “source material” to pass through that cell. The value in the cell must be normalized, i.e. valid for a standardized length (usually the cell’s width) of spread path. From the source location(s) and the local resistance raster, the GIS will be able to compute a new raster that indicates how much minimal total resistance the spread has undergone before reaching a raster cell. This process is illustrated in Figure 9.19.

diffusion and spread

resistance raster

**Figure 9.19**  
 Computation of diffusion on a raster. The lower-left green cell is the source location, indicated in the local resistance raster (a). The raster in (b) is the minimal total resistance raster computed by the GIS. (The GIS will work in real arithmetic of higher precision than that illustrated here.)



While computing total resistances, the GIS takes proper care of the path lengths. Obviously, the diffusion from a cell  $c_{src}$  to its neighbour cell to the east  $c_e$  is shorter than to the cell that is its northeastern neighbour  $c_{ne}$ . The distance ratio between these two cases is  $1/\sqrt{2}$ . If  $val(c)$  indicates the local resistance value for cell  $c$ , the GIS computes the total incurred resistance for diffusion from  $c_{src}$  to  $c_e$  as  $1(val(c_{src}) + val(c_e))$ , while the same for  $c_{src}$  to  $c_{ne}$  is  $1(val(c_{src}) + val(c_{ne})) \times \sqrt{2}$ . The accumulated resistance along a path of cells is simply the sum of these incurred resistances from pairwise neighbour cells.

Since “source material” has the habit of taking the easiest route to spread, we must determine at what minimal “cost” (i.e. at what minimal resistance) it may have arrived

in a cell. Therefore, we are interested in the minimal cost path. To determine the minimal total resistance along a path from the source location  $c_{src}$  to an arbitrary cell  $c_x$ , the GIS determines all possible paths from  $c_{src}$  to  $c_x$  and then determines which one has the lowest total resistance. This value is found, for each cell, in the raster of Figure 9.19b.

For instance, there are three paths from the green source location to its northeast neighbour cell (with local resistance 5). We can define them as path 1 (N–E), path 2 (E–N) and path 3 (NE), using compass directions to define the path from the green cell. For path 1, the total resistance is computed as:

$$1/2(4 + 4) + 1/2(4 + 5) = 8.5.$$

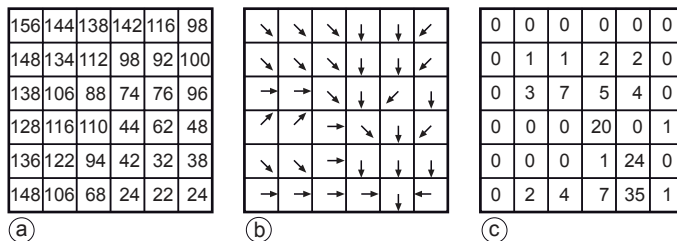
Path 2, in similar style, gives us a total value of 6.5. For path 3, we find

$$1/2(4 + 5) \times \sqrt{2} = 6.36,$$

and, thus, it is obviously the minimal cost path. The reader is asked to verify one or two other values of minimal cost paths that the GIS has produced for the output raster.

### 9.4.3 Flow computation

Spread computations determine how a phenomenon spreads over an area—in principle in all directions, though with varying difficulty or resistance. There are also cases for which a phenomenon does not spread in all directions, but moves or “flows” along a given, least-cost path, determined again by characteristics of local terrain. The typical case arises when we want to determine drainage patterns in a catchment area: rain water “chooses” a way to leave the area.



**Figure 9.20**  
Flow computations on a raster: (a) the original elevation raster, (b) the flow-direction raster computed from it, (c) accumulated flow-count raster.

We can illustrate the principles involved in this typical case with the simple elevation raster provided in Figure 9.20a. For each cell in that raster, the steepest downward slope to a neighbour cell is computed and its direction is stored in a new raster (Figure 9.20b). This computation determines the elevation difference between the cell and the neighbour cell and it takes into account cell distance—1 for neighbour cells in N–S or W–E direction, 2 for cells in a NE–SW or NW–SE direction. From among its eight neighbour cells, it picks the one with the steepest path to it. The directions thus obtained in raster (b) are encoded in integer values; we have “decoded” them for the sake of illustration. Raster (b) can be called the flow-direction raster. From raster (b), the GIS can compute the accumulated flow-count raster, a raster that for each cell indicates how many cells have their water flow into that cell.

Cells with a high accumulated flow count represent areas of concentrated flow and may, thus, belong to a stream. By using some appropriately chosen threshold value in a map algebra expression, we may decide whether they do or not. Cells with an accumulated flow count of zero are local topographic highs and can be used to identify ridges.

minimal cost path

flow direction  
flow accumulation



#### 9.4.4 Raster-based surface analysis

Continuous fields have a number of characteristics not shared by discrete fields. Since the field changes continuously, we can talk of slope angle, slope aspect and concavity/convexity of the slope. These notions are not applicable to discrete fields. The discussions in this subsection use terrain elevation as the prototype example of a continuous field, but all aspects discussed are equally applicable to other types of continuous fields. Nonetheless, we regularly refer to the continuous field representation as a DEM, to conform with the most common situation. Throughout the rest of this subsection we will assume that the DEM is represented as a raster.

#### Applications

There are numerous examples that require more advanced computations on continuous field representations, such as:

- Slope angle calculation—the calculation of the slope steepness, expressed as an angle in degrees or percentages, for any or all locations.
- Slope aspect calculation—the calculation of the aspect (or orientation) of the slope in degrees (between 0 and 360°), for any or all locations.
- Slope convexity/concavity calculation—defined as the change of the slope (negative when the slope is concave and positive when the slope is convex)—can be calculated as the second derivative of the field.
- Slope length calculation—with the use of neighbourhood operations, it is possible to calculate for each cell the nearest distance to a watershed boundary (the upslope length) and to the nearest stream (the downslope length). This information is useful for hydrological modelling.
- Hillshading is used to portray relief difference and terrain morphology of hilly and mountainous areas. The application of a special filter to a DEM produces hillshading. The colour tones in a hillshading raster represent the amount of reflected light at each location, depending on its orientation relative to the illumination source. This illumination source is usually chosen to be to the northwest at an angle of 45° above the horizon.
- Three-dimensional map display—with GIS software, three-dimensional views of a DEM can be constructed in which the location of the viewer, the angle under which he or she is looking, the zoom angle, and the amplification factor of relief exaggeration can be specified. Three-dimensional views can be constructed using only a predefined mesh, covering the surface, or using other rasters (e.g. a hillshading raster) or images (e.g. satellite images) that are draped over the DEM.
- Determination of change in elevation through time—the cut-and-fill volume of soil to be removed or to be brought in to make a site ready for construction can be computed by overlaying the DEM of the site before the work begins with the DEM of the expected modified topography. It is also possible to determine landslide effects by comparing DEMs of before and after a landslide event.
- Automatic catchment delineation—catchment boundaries or drainage lines can be automatically generated from a good quality DEM with the use of neighbourhood functions. The system will determine the lowest point in the DEM, which is considered to be the outlet of the catchment. From there, it will repeatedly

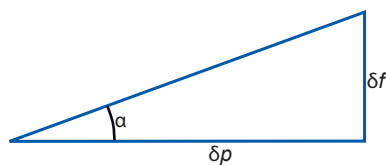
search for the neighbouring pixels with the highest altitude. This process is repeated until the highest location (i.e. the cell with the highest value) is found; the path followed determines the catchment boundary. For delineating the drainage network, the process is reversed. Then the system will work from the watershed downwards, each time looking for the lowest neighbouring cells, which determines the direction of water flow.

- Dynamic modelling—apart from the applications mentioned above, DEMs are increasingly used in GIS-based dynamic modelling, such as the computation of surface run-off and erosion, groundwater flow, the delineation of areas affected by pollution, the computation of areas that will be covered by processes such as flows of debris and lava.
- Visibility analysis—a viewshed is the area that can be “seen” (i.e. it is in the direct line-of-sight) from a specified target location. Visibility analysis can determine the area visible from a scenic lookout or the area that can be reached by a radar antenna, as well as assess how effectively a road or quarry will be hidden from view.

Some of the more important computations mentioned in the list above are discussed further in the following subsections. All of them apply a technique known as filtering, which has been described in Subsection 5.1.7.

#### Computation of slope angle and slope aspect

A different choice of weight factors may provide other information. Special filters exist to perform computations on the slope of the terrain. Before we look at these filters, let us define various notions of slope.



**Figure 9.21**

Slope angle defined. Here,  $\delta p$  stands for length in the horizontal plane,  $\delta f$  stands for the change in field value, where the field is usually terrain elevation. The slope angle is  $\alpha$ .

Slope angle, also known as slope gradient, is the angle  $\alpha$  illustrated in Figure 9.21, between a path  $p$  in the horizontal plane and the sloping terrain. The path  $p$  must be chosen such that the angle  $\alpha$  is maximal. A slope angle can be expressed as elevation gain in a percentage or as a geometric angle, in degrees or radians. The two respective formulas are:

$$\text{slope\_perc} = 100 \frac{\delta f}{\delta p} \text{ and } \text{slope\_angle} = \arctan \left( \frac{\delta f}{\delta p} \right)$$

The path  $p$  must be chosen to provide the highest slope-angle value and thus it can lie in any direction. The compass direction, converted to an angle to North, of this maximal downslope path  $p$  is what is called the slope aspect.

Let us now look at how to compute slope angle and slope aspect in a raster environment.

From an elevation raster, we cannot “read” the slope angle or slope aspect directly. Yet that information can be extracted. After all, for an arbitrary cell, we have its elevation value, plus those of its eight neighbouring cells. A simple approach to slope

angle computation is to make use of  $x$ - and  $y$ -gradient filters (gradient filters were introduced in Subsection 5.1.7). The  $x$ -gradient filter determines the slope increase ratio from west to east: if the elevation to the west of the centre cell is 1540 m and that to the east of the centre cell is 1552 m, then apparently along this transect the elevation increases 12 m per two cell widths, i.e. the  $x$ -gradient is 6 m per cell width. The  $y$ -gradient filter operates entirely analogously, though in the south-north direction.

Observe that both filters express elevation gain per cell width. This means that we must divide by the cell width—given in metres, for example—to obtain the (approximations to) the true derivatives  $\delta f/\delta x$  and  $\delta f/\delta y$ . Here,  $f$  stands for the elevation field as a function of  $x$  and  $y$ , and  $\delta f/\delta x$ , for instance, is the elevation gain per unit of length in the  $x$ -direction.

To obtain the real slope angle  $\alpha$  along path  $p$ , observe that both the  $x$ - and  $y$ -gradient contribute to it. This is illustrated in Figure 9.22. A not-so-simple geometric derivation can show that always:

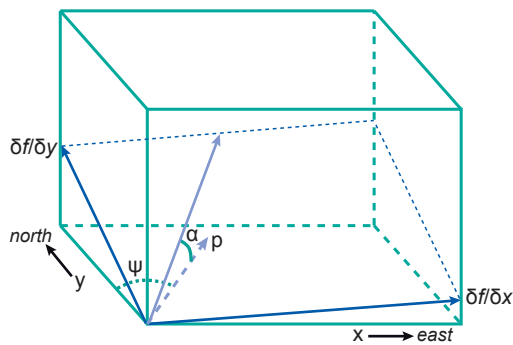
$$\tan(\alpha) = \sqrt{\left(\frac{\delta f}{\delta x}\right)^2 + \left(\frac{\delta f}{\delta y}\right)^2}.$$

In the practice of computing local slope angles from an elevation raster, this means that we must perform the following steps:

1. Compute from (input) elevation raster  $R$  the non-normalized  $x$ - and  $y$ -gradients.
2. Normalize the resulting rasters by dividing by the cell width, expressed in units of length such as metres.
3. Use both rasters for generating a third raster, applying the  $\sqrt{\dots}$  formula above, and possibly even an arctan function to the result, to obtain the slope angle  $\alpha$  for each cell.

It can also be shown that for the slope aspect  $\psi$  we have

$$\tan \psi = \frac{\delta f/\delta x}{\delta f/\delta y},$$



**Figure 9.22**  
Slope angle and slope aspect defined. Here,  $p$  is the horizontal path in maximal slope direction and  $\alpha$  is the slope angle. The plane tangent to the terrain in the origin is also indicated. The angle  $\psi$  is the slope aspect.

Slope aspect can, therefore, also be computed from the normalized gradients. Readers are warned that this formula should not be carelessly replaced by using

$$\psi = \arctan \left( \frac{\delta f / \delta x}{\delta f / \delta y} \right),$$

the reason being that the latter formula does not account for southeast and southwest quadrants, nor for cases where  $\delta f / \delta y = 0$ . (In the first situation, one must add  $180^\circ$  to the computed angle to obtain an angle measured from North; in the latter situation,  $\psi$  equals either  $90^\circ$  or  $-90^\circ$ , depending on the sign of  $\delta f / \delta x$ .)

## 9.5 Network analysis

Computations on networks comprise a different set of analytical functions in GISs. A network is a connected set of lines representing some geographic phenomenon, typically to do with transportation. The “goods” transported can be almost anything: people, cars and other vehicles along a road network, commercial goods along a logistic network, phone calls along a telephone network, or water pollution along a stream/river network.

Network analysis can be performed on either raster or vector data layers, but they are more commonly done on the latter, as line features can be associated with a network and hence can be assigned typical transportation characteristics, such as capacity and cost per unit. A fundamental characteristic of any network is whether the network lines are considered to be directed or not. Directed networks associate with each line a direction of transportation; undirected networks do not. In the latter, the “goods” can be transported along a line in both directions. We discuss here vector network analysis, and assume that the network is a set of connected line features that intersect only at the lines’ nodes, not at internal vertices. (But we do mention under- and overpasses.)

directed and undirected  
networks

For many applications of network analysis, a planar network, i.e. one that can be embedded in a two-dimensional plane, will do the job. Many networks are naturally planar, such as stream/river networks. A large-scale traffic network, on the other hand, is not planar: motorways have multi-level crossings and are constructed with underpasses and overpasses. Planar networks are easier to deal with computationally, as they have simpler topological rules. Not all GISs accommodate non-planar networks, or they can only do so using “tricks”. These tricks may involve the splitting of overpassing lines at the intersection vertex and the creation of four lines from the two original lines. Without further attention, the network will then allow one to make a turn onto another line at this new intersection node, which in reality would be impossible. In some GISs we can allocate a cost for turning at a node—see our discussion on turning costs below—and that cost, in the case of the overpass trick, can be made infinite to ensure it is prohibited. But, as mentioned, this is a work around to fit a non-planar situation into a data layer that presumes planarity. The above is a good illustration of geometry not fully determining the network’s behaviour. Additional application-specific rules are usually required to define what can and cannot happen in the network. Most GISs provide rule-based tools that allow the definition of these extra application rules. Various classical spatial analysis functions for networks are supported by GIS software packages. The most important ones are:

planar network—non-planar  
network

- optimal-path finding, which generates a least-cost path on a network between a pair of predefined locations using both geometric and attribute data.
- network partitioning, which assigns network elements (nodes or line segments) to different locations using predefined criteria.

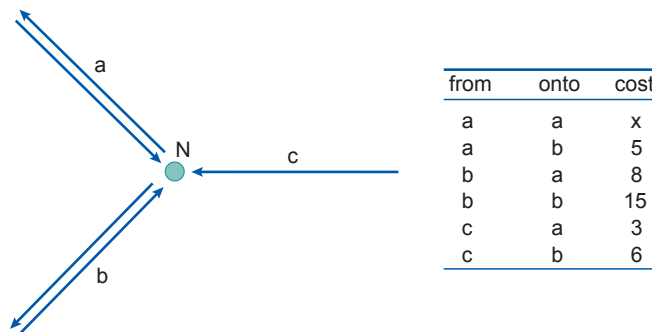
These two typical functions are discussed in the two subsections that follow.

9.5.1 Optimal-path finding

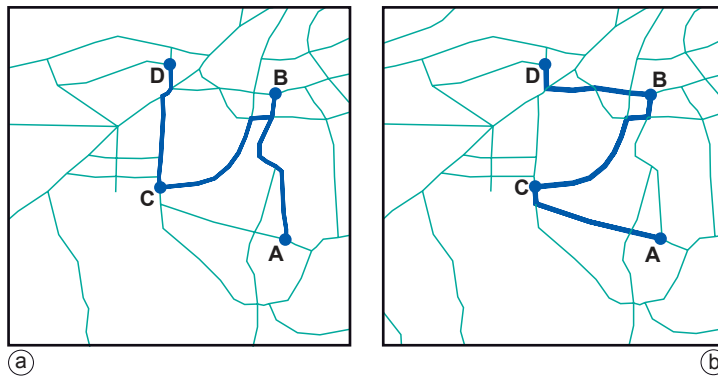
Optimal-path finding techniques are used when a least-cost path between two nodes in a network must be found. The two nodes are called origin and destination. The aim is to find a sequence of connected lines to traverse from the origin to the destination at the lowest possible cost. The cost function can be simple: for instance, it can be defined as the total length of all lines of the path. The cost function can also be more elaborate and take into account not only length of the lines but also their capacity, maximum transmission (travel) rate and other line characteristics, for instance to obtain a reasonable approximation of travel time. There can even be cases in which the nodes visited add to the cost of the path as well. These may be called turning costs, which are defined in a separate turning-cost table for each node, indicating the cost of turning at the node when entering from one line and continuing on another. This is illustrated in Figure 9.23.

cost function

**Figure 9.23**  
Network neighbourhood of node *N* with associated turning costs at *N*. Turning at *N* onto *c* is prohibited because of its direction, so no costs are mentioned for turning onto *c*. A turning cost of infinity ( $\infty$ ) also means that the turn is prohibited.



**Figure 9.24**  
Ordered (a) and unordered (b) optimal-path finding. In both cases, a path had to be found from *A* to *D*: in (a) by visiting *B* and then *C*; in (b) also by visiting both nodes, but in arbitrary order.

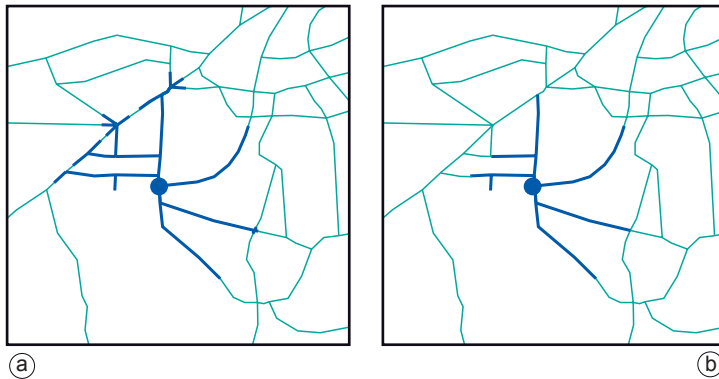


The attentive reader will notice that it is possible to travel on line *b* in Figure 9.23, make a U-turn at node *N*, and return along *a* to where one came from. The question is whether doing this makes sense in optimal-path finding. After all, to go back to where one came from will only increase the total cost. In fact, there are situations where it is optimal to do so. Suppose it is node *M* that is connected by line *b* with node *N*, and that we actually wanted to travel from *M* to another node *L*. The turn at *M* towards node *L* coming via another line may be prohibitively expensive, whereas turning towards *L* at *M* and returning to *M* along *b* may not be so expensive.

ordered—unordered pathfinding

Problems related to optimal-path finding may require *ordered* optimal path finding or *unordered* optimal-path finding. Both have as an extra requirement that a number of additional nodes need to be visited along the path. In ordered optimal-path finding,

the sequence in which these extra nodes are visited matters; in unordered optimal-path finding it does not. An illustration of both types is provided in Figure 9.24. Here, a path is found from node A to node D, via nodes B and C. Obviously, the length of the path found under non-ordered requirements is at most as long as the one found under ordered requirements. Some GISs provide support for these more complicated path-finding problems.



**Figure 9.25**  
Network allocation for a pupil/school assignment problem. In (a), the street segments within 2 km of the school are identified; in (b), the selection of (a) is further restricted to accommodate the school's capacity for the new school year.

### 9.5.2 Network partitioning

In network partitioning, the purpose is to assign lines and/or nodes of the network in a mutually exclusive way to a number of target locations. Typically, the target locations play the role of service centres for the network. This may be any type of service, e.g. medical treatment, education, water supply. This sort of network partitioning is known as a network allocation problem. Another problem is *trace analysis*. Here, one wants to determine that part of a network that is upstream (or downstream) from a given target location. Such problems exist in tracing pollution along river/stream systems, but also in tracking down network failures in energy distribution networks.

### 9.5.3 Network allocation

In network allocation, we have a number of target locations that function as resource centres, and the problem is which part of the network to exclusively assign to which service centre. This may sound like a simple allocation problem, in which a service centre is assigned those line (segments) to which it is nearest, but usually the problem statement is more complicated. The additional complications stem from the requirements to take into account (a) the capacity with which a centre can produce the resources (whether they are medical operations, seats for school pupils, kilowatts or bottles of milk), and (b) the consumption of the resources, which may vary amongst lines or line segments. After all, some streets have more accidents, more children who live there, more industry in high demand of electricity or just more thirsty workers.

The service area of any centre is a subset of the distribution network, in fact a connected part of the network. Various techniques exist to assign network lines, or their segments, to a centre. In Figure 9.25a, the blue dot indicates a primary school and the GIS has been used to assign streets and street segments along the network within 2 km distance of the school. Then, using demographic figures on pupils living along the streets, it was determined that too many potential pupils lived in the area for the school's capacity. So in part (b), the part of the network already selected was reduced to match precisely the school's capacity for pupils in the new school year.

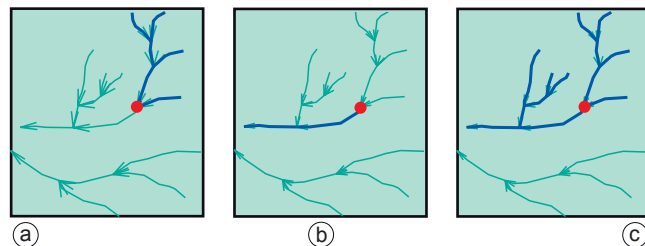
service area

### 9.5.4 Trace analysis

Trace analysis is performed when we want to understand which part of a network is “conditionally connected” to a chosen node on the network, which is known as the “trace origin”. If a node or line is conditionally connected, this means that a path exists from the node/line to the trace origin, and that the connecting path fulfills the conditions set. What these conditions are depends on the application; they may involve the direction of the path, its capacity, its length, or resource consumption along it. The condition is typically a logical expression, as we have seen before:

- the path must be directed from the node/line to the trace origin;
- its capacity (defined as the minimum capacity of the lines that constitute the path) must be above a given threshold; and
- the path’s length must not exceed a given maximum length.

Tracing is the computation that the GIS performs to find the paths from the trace origin that obey the tracing conditions. It is a rather useful function for many network-related problems. In Figure 9.26 the trace origin is indicated in red. In part (a), tracing conditions were set to trace all the way upstream; part (b) traces all the way downstream; and in part (c) there were no conditions set for the direction of the path, thereby tracing all connected lines from the trace origin. More complex conditions are certainly possible in tracing.



**Figure 9.26**  
Tracing functions on a network: (a) tracing upstream, (b) tracing downstream, (c) tracing all connected lines from the origin.

## 9.6 Error propagation in spatial data processing

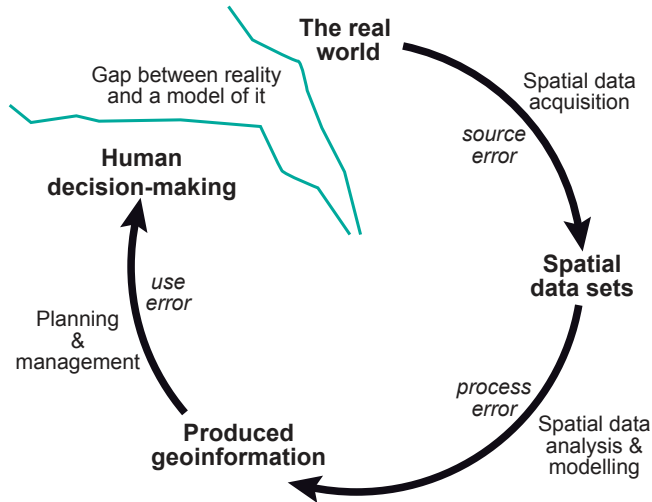
### 9.6.1 How errors propagate

In Section 8.5, a number of sources of error that may be present in source data were discussed. The acquisition of high quality base data still does not guarantee that the results of further, complex processing can be treated with certainty. As the number of processing steps increases, it becomes more difficult to predict the behaviour of such error propagation. These various errors may affect the outcome of spatial data manipulations. In addition, further errors may be introduced during the various processing steps discussed earlier in this chapter (see Figure 9.27).

Table 9.2 lists some common sources of error that may be introduced into GIS analyses. Note that these originate in a wide range of sources and include various common tasks relating to both data preparation and data analysis. It is the combination of different errors that are generated at each stage of preparation and analysis that may result in several errors and uncertainties in the final outputs.

One of the most commonly applied operations in GISs is analysis by overlaying two or more spatial data layers. Each of these layers will contain errors, due to both in-





**Figure 9.27**  
Error propagation in spatial data handling.

**Table 9.2**  
Some of the most common causes of error in spatial data handling.

<b>Coordinate adjustments</b> rubber sheeting/transformations projection changes datum conversions rescaling	<b>Generalization</b> linear alignment line simplification addition/deletion of vertices linear displacement
<b>Feature Editing</b> line snapping extension of lines to intersection reshaping moving/copying elimination of spurious polygons	<b>Raster/Vector Conversions</b> raster cells to polygons polygons to raster cells assignment of point attributes to raster cells post-scanner line thinning
<b>Attribute editing</b> numeric calculation and change text value changes/substitution re-definition of attributes attribute value update	<b>Data input and Management</b> digitizing scanning topological construction / spatial indexing dissolving polygons with same attributes
<b>Boolean Operations</b> polygon on polygon polygon on line polygon on point line on line overlay and erase/update	<b>Surface modelling</b> contour/lattice generation TIN formation Draping of data sets Cross-section/profile generation Slope/aspect determination
<b>Display and Analysis</b> cluster analysis calculation of surface lengths shortest route/path computation buffer creation display and query adjacency/contiguity	<b>Display and Analysis</b> class intervals choice areal interpolation perimeter/area size/volume computation distance computation spatial statistics label/text placement

herent inaccuracies in the source data and errors arising from some form of computer processing—for instance, rasterization. During the process of spatial overlay-

ing, errors in the individual data layers contribute to the final error of the output. The amount of error in the output depends on the type of overlay operation applied and on the amount of error in the individual layers. For example, errors in the results of an overlay using the logical operator AND are not the same as those created using the OR operator.

Consider another example. A land use planning agency is faced with the problem of identifying areas of agricultural land that are highly susceptible to erosion. Such areas occur on steep slopes in areas of high rainfall. The spatial data used in a GIS to obtain this information might include:

- a land use map produced five years previously from 1:25,000 scale aerial photographs;
- a DEM produced by interpolating contours from a 1:50,000 scale topographic map; and
- annual rainfall statistics collected with two rainfall gauges.

The reader is invited to assess what sort of errors are likely to occur in this analysis.

Referring back to Figure 9.27, the reader is also encouraged to reflect on errors introduced in the components of the application models discussed in the previous section, specifically, the methodological aspects of representing geographic phenomena. What might be the consequences of using a random function in an urban transportation model (when, in fact, travel behaviour is not purely random)?

### 9.6.2 Quantifying error propagation

Chrisman [18] noted that “the ultimate arbiter of cartographic error is the real world, not a mathematical formulation”. We will never be able to capture and represent everything that happens in the real world perfectly in a GIS. Hence there is much to recommend the use of testing procedures for assessing accuracy. Various perspectives, motives and approaches for dealing with uncertainty have given rise to a wide range of conceptual models and indices for the description and measurement of error in spatial data. All these approaches have their origins in academic research and have solid theoretical foundations in mathematics and statistics. Here we identify two main approaches for assessing the nature and amount of error propagation:

1. testing the accuracy of each state by measurement against the real world; and
2. modelling error propagation, either analytically or by means of simulation techniques.

Error propagation can be modelled mathematically, although these models are very complex and only valid for certain types of data (e.g. numerical attributes). Rather than explicitly modelling error propagation, it is often more practical to test the results of each step in the process against some independently measured reference data.

### Models of error and error propagation

It is important to distinguish models of error from models of error propagation in GISs. Modelling of error propagation has been defined by Veregin [118] as: “the application of formal mathematical models that describe the mechanisms whereby errors in source data layers are modified by particular data transformation operations.”

In other words, we would like to know how errors in the source data behave under the manipulations that we subject them to in a GIS. If we are able to quantify the error in

the source data as well as their behaviour under GIS manipulations, we have a means of judging the uncertainty of their results.

Initially, error propagation models described only the propagation of attribute error [44], [118]. More recent research has addressed the spatial aspects of error propagation and the development of models incorporating both attribute and location components. These topics are beyond the scope of this book and readers are referred to [2] and [58] for more details.

