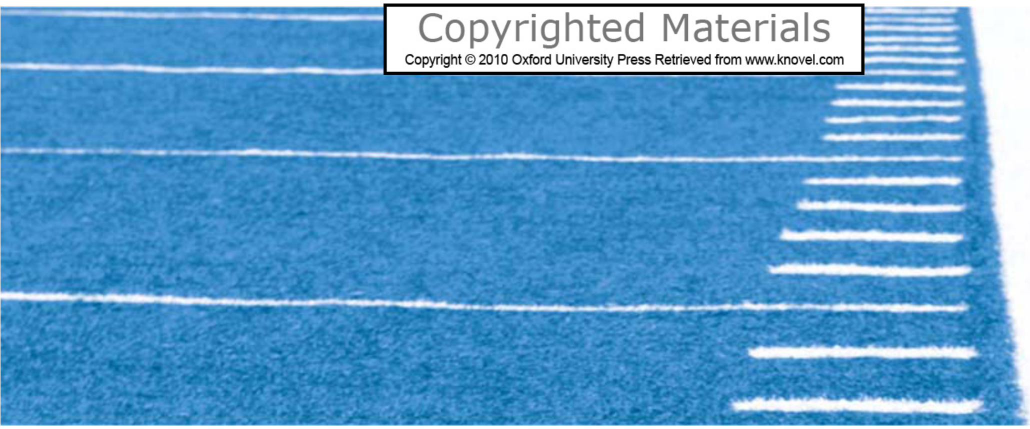

10

Analysis of grids and surfaces

10.1 Introduction

This chapter introduces methods for processing and analysing gridded data in general and gridded elevation data in particular. Topics include map algebra, image processing, and spatial filters. Derivatives of altitude, such as slope and other products derived from surfaces, are also described. Image processing has a central role in GIS contexts. This chapter begins by outlining some key approaches to the analysis of gridded data, of which images are a common example.

10.2 Map algebra

Most GIS software packages include functions for map algebra. In words, using map algebra raster layers can be combined in various ways. For example, the values in overlapping cells may be added together using this format:

OUTPUT = MAP1 + MAP2

Similarly, values could be multiplied or any other arithmetic operation applied. Obviously, in many cases such operations may be meaningless—it makes no sense to add altitude values to precipitation values. However, addition of pollutant scores (as opposed to concentrations), for example, may be sensible. Map algebra provides a means of masking files. For example, if a raster exists which has values of 1 for areas of interest and 0 for areas that are not of interest then this map layer could be multiplied

$$\begin{array}{|c|c|c|}\hline 1 & 0 & 0 \\ \hline 0 & 1 & 0 \\ \hline 0 & 1 & 1 \\ \hline \end{array} \times \begin{array}{|c|c|c|}\hline 7 & 6 & 4 \\ \hline 6 & 5 & 4 \\ \hline 5 & 4 & 3 \\ \hline \end{array} = \begin{array}{|c|c|c|}\hline 7 & 0 & 0 \\ \hline 0 & 5 & 0 \\ \hline 0 & 4 & 3 \\ \hline \end{array}$$

Figure 10.1 Retaining grid cells indicated by 1 in a binary grid.

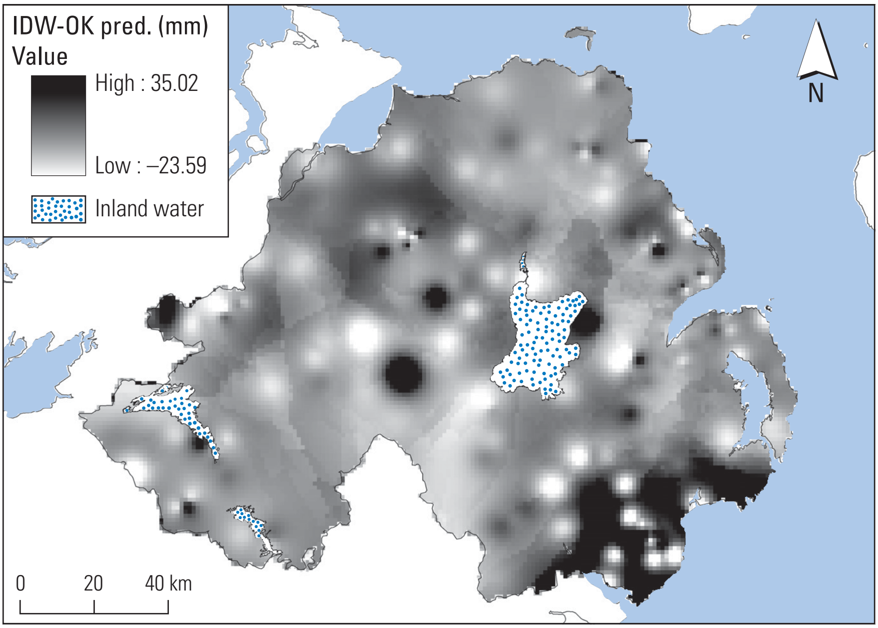

Figure 10.2 IDW minus OK predictions of precipitation in July 2006 (both using 16 nearest neighbours).

by any other layer. The result is that only features in areas of interest are transferred to the new raster layer as all cells of interest are multiplied by 1, whereas cells outside the area of interest are multiplied by 0. Figure 10.1 gives a simple example of this operation.

Figure 10.2 shows the map obtained when IDW predictions of precipitation (Figure 9.7) are subtracted from OK predictions (Figure 9.13). In this case, the map algebra command corresponds to OUTPUT = IDWMAP – OKMAP. Burrough and McDonnell (1998) provide more examples of map algebra.

Positive values indicate that the IDW predictions are larger while negative values indicate that OK predictions are larger. In other words, where values are positive, IDW predicts larger precipitation amounts than OK does and, where they are negative, OK predicts larger precipitation amounts than IDW does. This kind of approach offers a simple means to compare grids. One obvious potential applications area is the comparison of grids for different time periods. If digital elevation models (DEMs, as illustrated in Figure 10.5), for example, are available for two time periods, one could be subtracted from the other to assess rates of erosion. However, it is, of course, necessary to ensure that data sets compared in this way are compatible (e.g. they have the same spatial resolution and the data were collected with instruments which make measurements with a comparable level of accuracy). Matrix algebra is a key tool in multicriteria decision analysis (see Section 5.3).

10.3 Image processing

The analysis of images has, in comparison to GIS, a long history. The need to remove 'noise' from raster grids (or images; e.g. remotely sensed images) or to identify features contained in images has led to the development of a wide range of sophisticated tools. Tools for spatial segmentation and classification are widely used to deal in different ways with the grouping together of pixels that have, in some respect, similar characteristics. Such methods are outside the remit of this account. This section begins by defining different classes of grid operators. Grid operators can be divided into four groups (Chou, 1997):

- local functions: work on every single cell (a cell is treated as an individual object)
- focal functions: derive a new value based on the neighbourhood of a pixel
- zonal functions: work on each group of cells of identical values
- global functions: work on a cell based on the data in the entire grid.

The addition of two grids is a local function since the values in overlapping individual cells are combined. Moving window functions (such as spatial filters, described in the following section) are examples of focal operators. Zonal operators deal with values that fall within particular zones, and an example is given below. A common example of a global function is the computing of Euclidean (straight line) distance from one or more source locations to all cells in a grid (see Section 4.2).

With zonal operators, values from one input grid that fall within zones indicated by a second input grid are combined in some way. The zonal mean is illustrated in Figure 10.3. The zonal minimum or maximum, for example, could be computed instead. Note that the boundaries of the individual cells in the zones layer and the zonal means layer are not shown—the values in both cases would be written to all cell locations indicated in the values grid. As an example, for zone 1: $(45+44+44+43+42+43+42)/7=43.286$.

There are many potential applications areas for zonal operations, for example average erosion in construction zones. Any application which makes use of zones represented as rasters might make use of operators like those just described.

The focus of the following sections is spatial filters for image smoothing or enhancing edges of features.

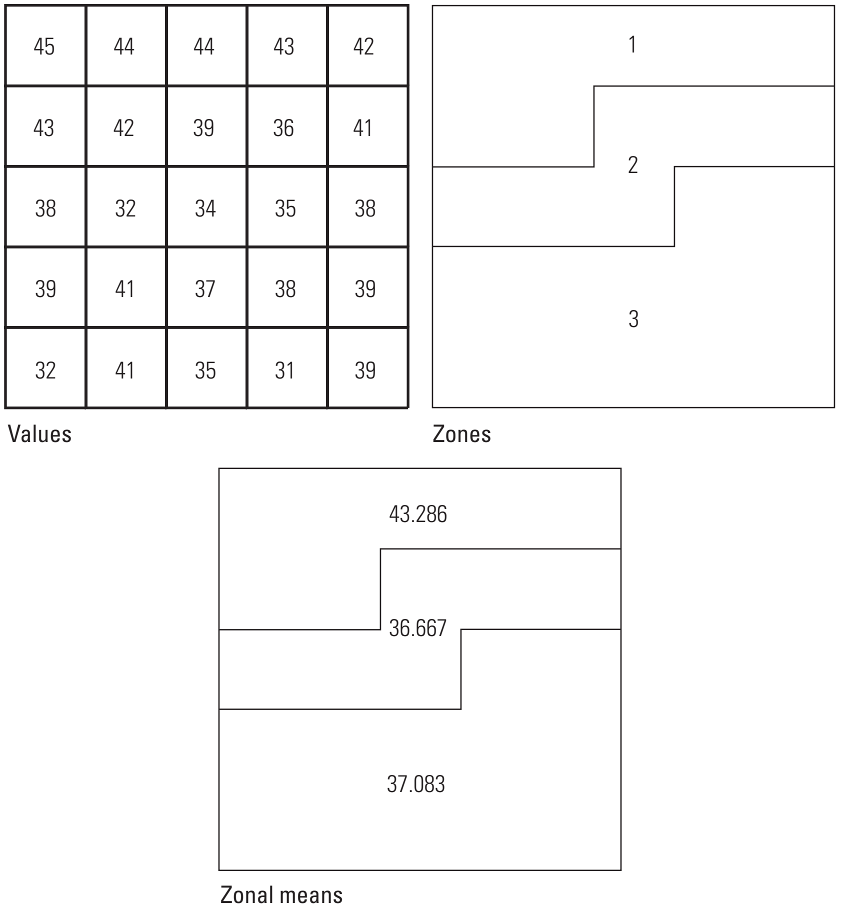

Figure 10.3 Zonal mean.

10.4 Spatial filters

The idea of moving windows for processing images was introduced in Section 4.6. Such operations are called spatial filters. The application of a mean filter, as illustrated in Section 4.6, results in a smoothed version of the original image. In contrast, application of a standard deviation filter will result in a new image that highlights the edges of features in the original image. Figure 10.4 shows an input grid (the same one as used in Section 4.6) and the output of a mean filter and a standard deviation filter.

Comparison of the values of the standard deviation output in Figure 10.4 with the input grid shows how areas with more variable values have larger standard deviations. As with the example in Section 4.6, the edge pixels are not included in the outputs.

Figure 10.5 shows elevations in Northern Ireland. Figure 10.6 shows the mean of elevation for a 3×3 pixel moving window. Comparison with the original elevation map shows that the range of values in the mean map is smaller than in the original map as the effect of outliers (values that are large or small locally) is reduced through

Output grid (3 × 3 cells) location is shaded

45	44	44	43	42
43	42	39	36	41
38	32	34	35	38
39	41	37	38	39
32	41	35	31	39

Input grid

40.11	38.78	39.11
38.33	37.11	37.44
36.56	36.00	36.22

Mean

4.68	4.66	3.62
3.61	3.26	2.19
3.50	3.57	2.68

Standard deviation

Figure 10.4 Mean average and standard deviation computed for a 3×3 pixel moving window.

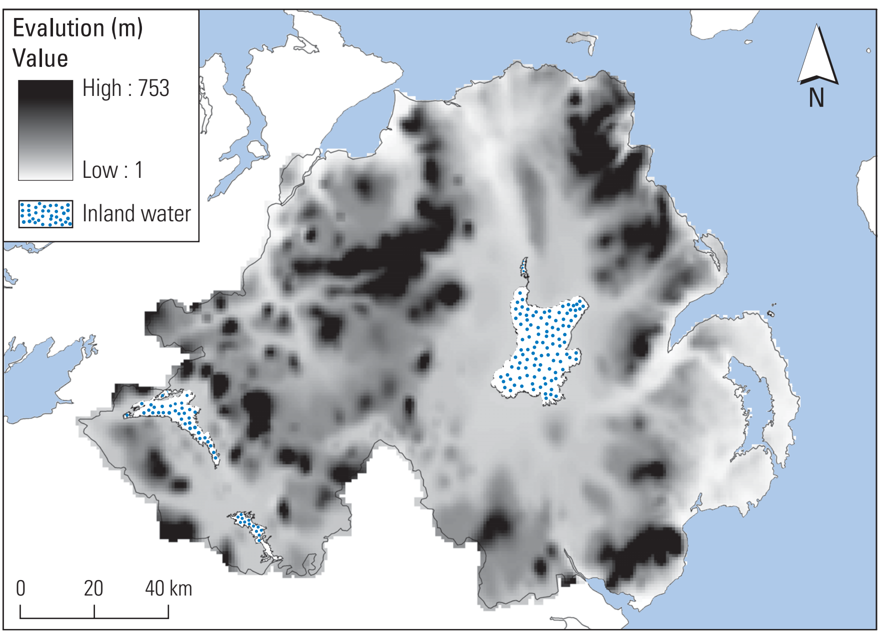

Figure 10.5 Elevation: Northern Ireland. Data available from the United States Geological Survey/EROS, Sioux Falls, SD.

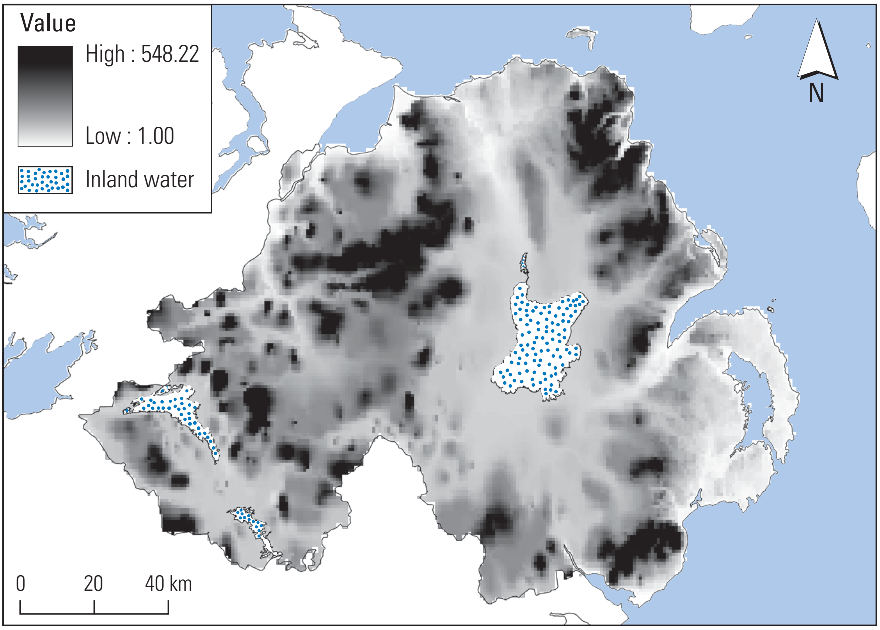

Figure 10.6 Mean of elevation for a 3×3 pixel moving window.

averaging. Figure 10.7 shows the standard deviation, also for a 3×3 pixel moving window. In this case, it is apparent that the edges of more notable topographic features (particularly those with high elevation values) are highlighted.

The mean filtered DEM (recall that this is a grid-based representation of topographic form) in Figure 10.6 is smoother in appearance than the unfiltered DEM in Figure 10.5. This is particularly clear on the edges of areas with higher elevation values. Other filters are defined by Sonka *et al.* (1999) and Lloyd (2006). Smoothing filters (such as the mean filter) are used in many different contexts. For example, smoothing filters are often used to reduce the effect of 'noise' in remotely sensed images (Mather, 2004), although the median filter, for example, may be more suitable in such cases than the mean filter (as the mean is affected by outliers, whereas the median is not). The focus of the following section is the analysis of DEMs specifically (as opposed to other forms of raster grids).

10.5 Derivatives of altitude

The remainder of this chapter focuses on the analysis of DEMs. Various products are often derived from DEMs—derivatives such as gradient and aspect (the two making up slope) are often computed. Gradient refers to the maximum rate of change in altitude while aspect refers to the direction of the maximum rate of change (e.g. gradient may be north facing) (Burrough and McDonnell, 1998). The terms 'gradient' and 'slope' are sometimes used interchangeably, but here the convention of defining slope to comprise

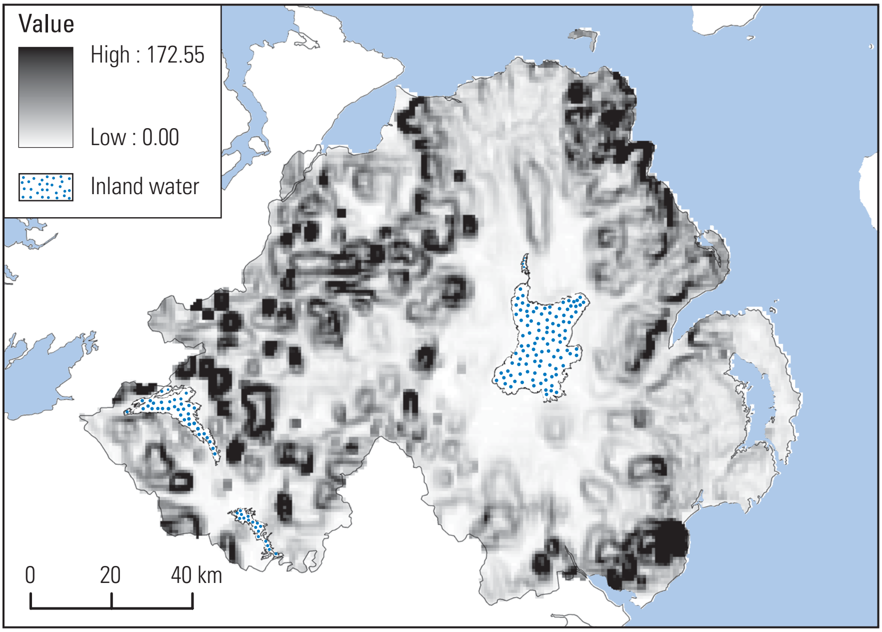

Figure 10.7 Standard deviation of elevation for a 3×3 pixel moving window.

gradient and aspect together is preferred. There are several methods for deriving gradient and one of the most popular is demonstrated here. Given cells labelled as follows:

$$\begin{matrix} z_1 & z_2 & z_3 \\ z_4 & z_5 & z_6 \\ z_7 & z_8 & z_9 \end{matrix}$$

the gradient can be estimated from (Horn, 1981):

$$\frac{\partial z}{\partial x} = \frac{(z_3 + (2z_6) + z_9) - (z_1 + (2z_4) + z_7)}{8h_x} \tag{10.1}$$

$$\frac{\partial z}{\partial y} = \frac{(z_3 + (2z_2) + z_1) - (z_9 + (2z_8) + z_7)}{8h_x} \tag{10.2}$$

where h_x is the grid spacing in the x direction (east–west) and h_y is the grid spacing in the y direction (north–south). The method is illustrated using the following 3×3 cell matrix, which we will assume has a spatial resolution of 10 m (i.e. cells that cover an area in the real world of 10×10 m):

45	44	44
43	42	39
38	32	34

Using the approach detailed by Horn (1981), the gradient of the central cell is computed using Equations 10.1 and 10.2:

$$\frac{\partial z}{\partial x} = \frac{(44+(2\times 39)+34)-(45+(2\times 43)+38)}{(8\times 10)} = 0.1625$$

$$\frac{\partial z}{\partial y} = \frac{(44+(2\times 44)+45)-(34+(2\times 32)+38)}{(8\times 10)} = 0.5125$$

Using these figures, the gradient is calculated from:

$$g = \sqrt{\left(\frac{\partial z}{\partial x}\right)^2 + \left(\frac{\partial z}{\partial y}\right)^2} \qquad (10.3)$$

For this example, this leads to:

$$g = \sqrt{(0.1625)^2 + (0.5125)^2} = 0.5376 \text{ m}$$

This can be converted to gradient in degrees (*gd*):

$$gd = \text{atan}(g) \times 57.29578 \qquad (10.4)$$

where atan (also given by $\tan^{-1}$) is the inverse tangent (see Appendix C). Gradient is often also expressed as percentages. Here gradient in degrees is:

$$gd = 0.4933 \times 57.29578 = 28.264°$$

Figure 10.8 shows gradient (in degrees) derived from a DEM of Northern Ireland. Note that the spatial resolution of the DEM from which gradient was derived is 740.224 m. The gradients are never as large as in the computed example above because any dramatic gradients are, in effect, averaged out because of the coarse spatial resolution.

Note that the gradient map is visually similar to the standard deviation map given earlier in the chapter (Figure 10.7). This is not surprising as the standard deviation picks out the edges of features and gradient does the same. Maps of gradient are used widely for modelling erosion. Mitásová *et al.* (1996) computed gradient and aspect as a part of a set of procedures for modelling erosion and deposition. Li *et al.* (2004) provide a detailed account of gradient and aspect derivation as well as other terrain parameters.

10.6 Other products derived from surfaces

Besides gradient and other derivatives of altitude, many other products are derived from DEMs. These include maps indicating the direction of steepest downhill descent

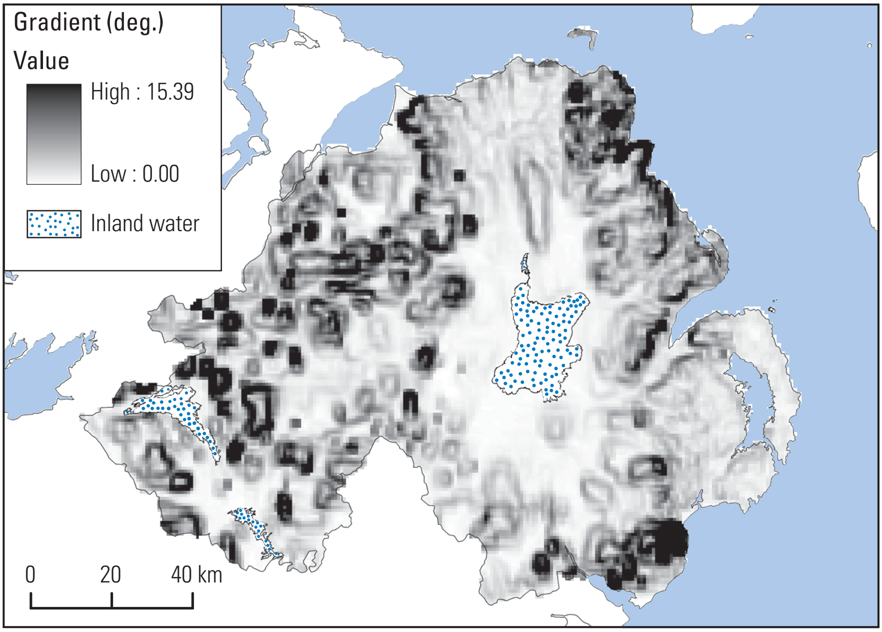

Figure 10.8 Gradient in degrees.

(routing), irradiance, watershed, cost surfaces, and visibility maps. Algorithms also exist for processing DEMs—in particular, removal of 'anomalous' pits or peaks is often a concern. This section considers two classes of widely used DEM-based approaches: visibility analysis and derivation of cost surfaces.

Visibility analysis is concerned with the identification of areas in the landscape that are visible from a given location (or a set of locations). Many GIS offer tools for generating viewsheds—areas that are visible from a given point. Figure 10.9 indicates a viewing cell in black and the pixel that is to be tested for visibility is grey. Figure 10.10 indicates the two possible cases: that the cell is invisible or visible. By testing the line of sight to each cell, a map of visible cells (the viewshed) can be generated.

The line of sight from one location to another can be assessed using:

$$h_{\text{crit}} = \frac{d_{\text{vo}}h_{\text{t}} + d_{\text{to}}h_{\text{v}}}{d_{\text{to}} + d_{\text{vo}}} \tag{10.5}$$

where h_{crit} is the critical value for the height of an obstacle, d_{to} is the distance between the target and the obstacle, d_{vo} is the distance between the viewer and the obstacle, h_{t} is the height of the target, and h_{v} is the height of the viewer. If the height of the obstacle is less than h_{crit} then the target is visible (Li *et al.*, 2004). As an example, for $d_{\text{to}} = 24.5$, $d_{\text{vo}} = 12$, $h_{\text{t}} = 8$, and $h_{\text{v}} = 5$:

$$h_{\text{crit}} = \frac{12 \times 8 + 24.5 \times 5}{24.5 + 12} = \frac{218.5}{36.5} = 5.986$$

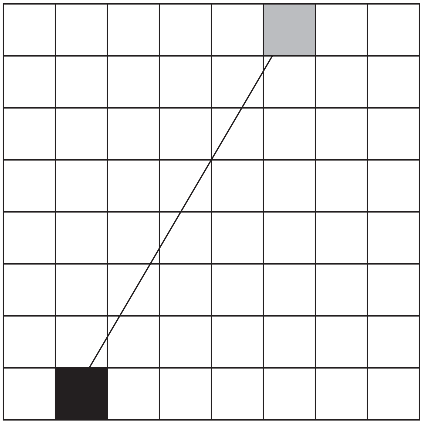

Figure 10.9 Viewing direction.

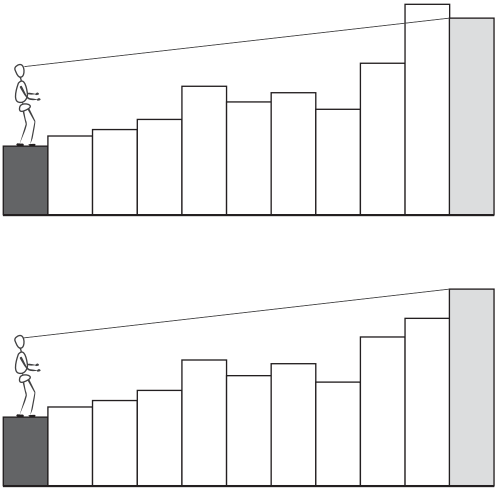

Figure 10.10 Top: invisible cell; bottom: visible cell (both shaded grey).

In this case, if the obstacle is higher than 5.986 units then the target will not be visible to the viewer.

When the height of the obstacle (h_o) is known, the length of the possible line of sight that is blocked by the obstacle can be computed using (Li *et al.*, 2004):

$$d_{to} = \frac{h_o}{h_v + h_o} \times d_{vo} \quad (10.6)$$

This approach could be applied to all cells in a raster image, but more efficient algorithms exist. Figure 10.11 shows a DEM with a viewpoint superimposed (A) and the viewshed from this location (B).

Note that the viewer height must be specified and not simply the elevation or building height at the viewing location. Of course, visibility can only be meaningfully

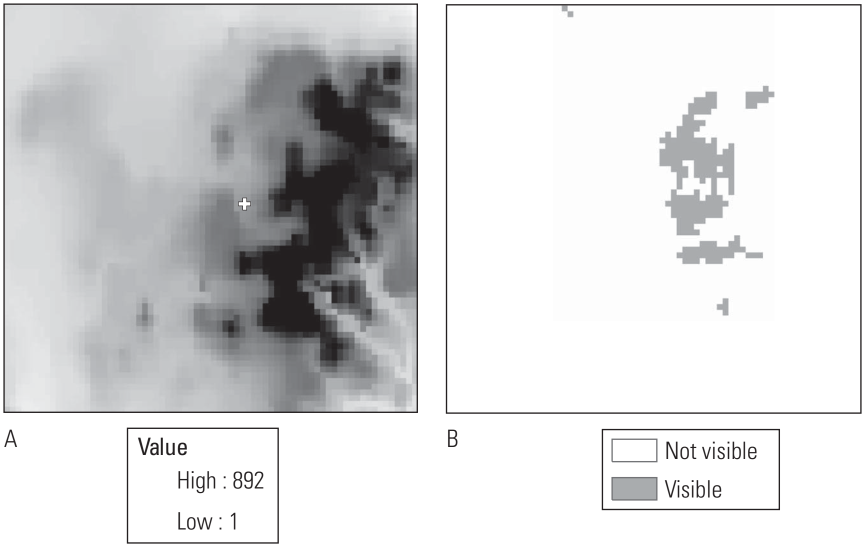

Figure 10.11 DEM (with elevations in metres) with a viewpoint superimposed (A) and the viewshed from this location (B).

measured up to the distance to which objects will be visible and such considerations must be taken into account when generating viewsheds. Viewsheds are used widely in various contexts. One obvious use is generating viewsheds as part of the planning process whereby the visual impact of a development on the landscape can be minimized by identifying areas that are not visible from particular locations. Wheatley (1995) used viewshed analysis to explore intervisibility between Neolithic long barrows in the Danebury region (in south-west England). Fisher (1992) considers the uncertainties inherent in viewsheds.

Another frequently generated product using DEMs is a cost surface. Cost surfaces indicate the minimum cost of reaching a cell from source cells. To generate a cost surface a friction surface and source cells are needed. The friction surface is often simply a gradient map—steep gradients represent a greater degree of friction or higher cost. A cost surface can be generated using Dijkstra's algorithm, as outlined in Section 6.5. When a DEM is generated using spatial interpolation, the interpolation method used will determine how 'rough' the surface is and thus the modelled 'cost' of travelling over the surface. Such factors, which may not relate to variations in the real world, should be taken into account.

An example is given using the cost grid (e.g. gradient) and source cells (cell locations from which distance are measured) shown in Figure 10.12.

Cost distances are computed as follows:

- neighbours sharing an edge = average of costs in the neighbouring cells (e.g. $45+44={}^{89}/_{2}=44.5$)

- neighbours sharing only a corner = average of costs in the neighbouring cells multiplied by 1.4142 (e.g. ${}^{(45+42)}/2 = {}^{87}/2 = 43.5 \times 1.1412 = 61.5$).

The cost surface procedure (note the connections with Section 6.5) can be outlined as follows:

- Compute the cost distances from each source cell to its neighbours.
- Select the smallest cost distance and compute the smallest cost distance to that cell's neighbours—these cells are activated.
- The next activated cell with the smallest accumulative cost distance is selected. Next, compute the smallest cost distance to that cell's neighbours. Every time a cell becomes accessible to a source cell through a different path it is reactivated and its accumulative cost must be recalculated because the new path may have a smaller accumulative cost (Chang, 2008). If it does not, then the accumulative cost value remains the same.
- Continue this process until all smallest accumulative cost distances have been computed.

45	44	44	43
43	42	39	36
38	32	34	35
39	41	37	38

Cost grid

1			
			2

Source cells

Figure 10.12 Cost grid and source cells.

0	44.5		
44	61.5		
		50.9	**36.5**
		37.5	0

Step 1

0	44.5		
44	61.5	87.4	72
		50.9	36.5
		37.5	0

Step 2

0	44.5		
44	61.5	87.4	72
	83.9	50.9	36.5
	76.5	37.5	0

Step 3

0	**44.5**		
44	61.5	87.4	72
84.5	83.9	50.9	36.5
	76.5	37.5	0

Step 4

0	44.5	88.5	
44	61.5	87.4	**72**
84.5	83.9	50.9	36.5
	76.5	37.5	0

Step 5

0	44.5	88.5	111.5
44	61.5	87.4	72
84.5	83.9	50.9	36.5
	76.5	37.5	0

Step 6

0	44.5	88.5	111.5
44	61.5	87.4	72
84.5	83.9	50.9	36.5
116.5	76.5	37.5	0

Step 7: Least accumulative cost distances

Figure 10.13 Cost surface derivation.

The procedure is illustrated in Figure 10.13. The smallest accumulative cost distances that have not yet been included are indicated by bold characters.

In step 1, the cost distances are measured from the source cells, and the smallest cost distance (36.5) is selected. The cost distances are then measured from the selected cell and the smallest non-selected cost distance (37.5) is selected, the end result being step 2. Next, the cost distances are measured from that cell location and the smallest non-selected value is selected (44)—this is step 3. The cost distances are then measured from the selected cell and the smallest non-selected cost distance is selected (44.5), leading to step 4. The cost distances are measured from this cell and the smallest non-selected cost distance (72) is selected—this is step 5. The cost distances are measured from this cell location and the smallest non-selected value (76.5) is selected. After the distances are measured from that cell all accumulative cost distances have been computed and the final grid is step 7.

The direction of the least accumulative cost path for the above example is given in Figure 10.14 while the origin of the least cost path (i.e. the parent cell) is given in Figure 10.15 (1 and 2 being the values assigned to the source cells).

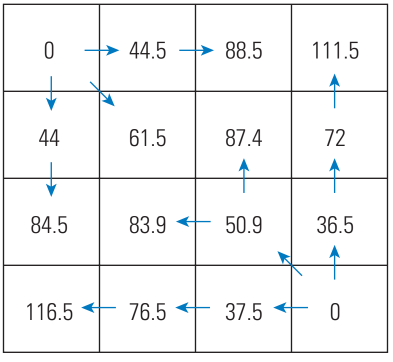

Figure 10.14 Direction of the least accumulative cost path.

1	1	1	2
1	1	2	2
1	2	2	2
2	2	2	2

Figure 10.15 Origin of the least accumulative cost path.

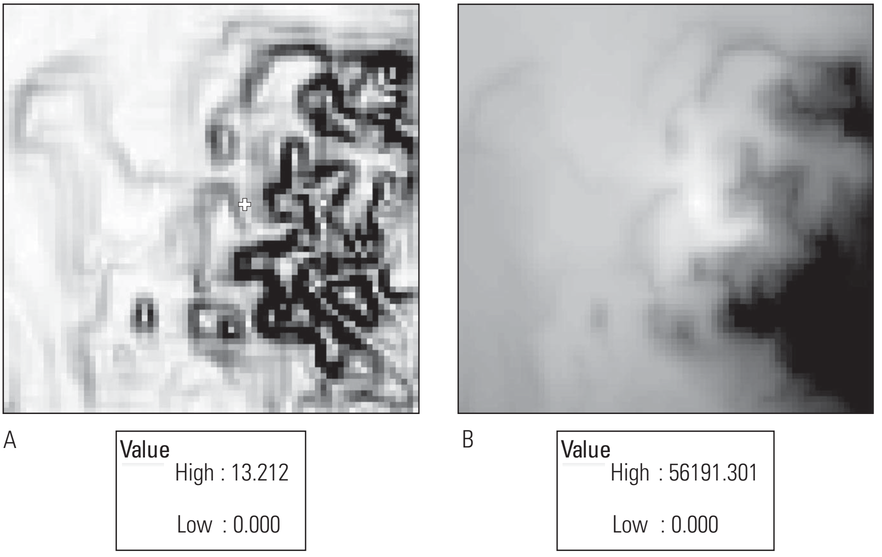

Figure 10.16 Gradient (in degrees) with starting location superimposed (A) and the cost surface from this location (B).

Figure 10.16 shows a gradient map (in degrees) with the starting location (A) and the cost surface derived using these inputs (B).

Cost surfaces are computed in numerous different contexts. In an application concerned with identifying possible routes for the construction of a new railway, Cowen *et al.* (2000) generated cost surfaces accounting for several different factors, including grade (difference between existing and desired elevation), road crossings, stream crossings, and track cost. The generation of alternative routes and comparisons between them has been found to be very useful in many planning projects of this kind.

10.7 Case study

Gradient (in degrees) was computed using ArcGIS™ Spatial Analyst from the DEM (with a spatial resolution of 1009.975 m) illustrated in Figure 10.17. The DEM is described by Dubois (2003). The algorithm used is that detailed by Horn (1981) and the output is shown in Figure 10.18.

The southern part of Switzerland is dominated by the Alps, and this explains the steep gradients in that region; in the north-west the Jura mountains are evident. In terms of, for example, modelling the costs of transporting goods, this output would suggest that it requires more effort to move goods over land in the south of Switzerland than in the north. If a value can be assigned to a gradient derived from a DEM then such data may provide the basis of a useful analysis.

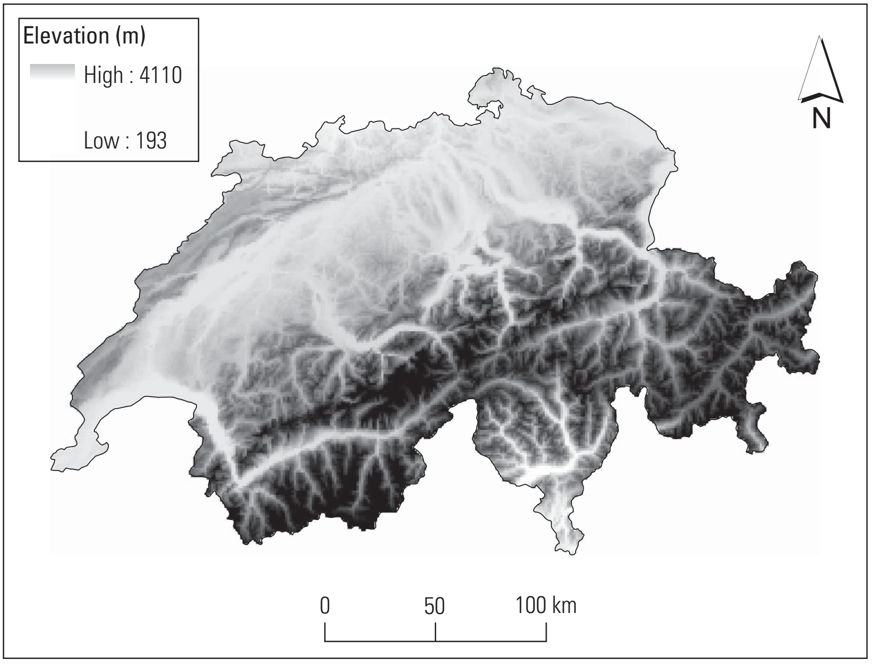

Figure 10.17 Elevation: Switzerland.

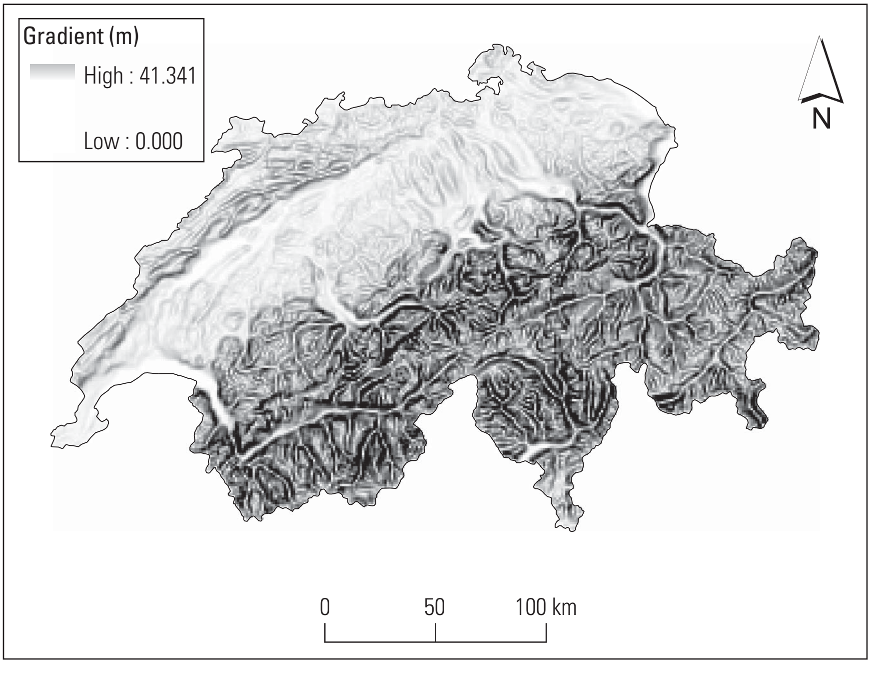

Figure 10.18 Gradient: Switzerland.

Summary

This chapter has dealt with various aspects of surface and analysis. Map algebra, image processing, and spatial filters were introduced. Following these, derivation of gradient was detailed before a summary of some widely used methods for the analysis of surfaces was given. Such approaches are very widely used in many different applications and an understanding of such approaches is essential in developing a rounded knowledge of spatial data analysis. As detailed in Section 2.8.3, remote sensing is a core source of spatial data and processing the imagery, using the kinds of techniques introduced in this chapter, is an important task if these data are to be used to the full.

Further reading

Some basic principles of image processing are outlined by **Burrough and McDonnell (1998)**, who also discuss elevation derivatives. The book by **Sonka *et al.* (1999)** provides a detailed introduction to some key principles of image processing. In remote sensing data contexts specifically, the book by **Mather (2004)** gives an excellent summary of some important issues. A dedicated account of DEMs and their analysis is given by **Li *et al.* (2004)**.

➜ The next chapter revisits some of the themes explored throughout this book and suggests some ways forward for the interested reader.