Chapter 5
Geometric Processing and Enhancement:
Image Domain Techniques

5.1 Introduction

This chapter presents methods that allow us to analyse or modify the geometric properties of an image. Our attention, first, is on techniques for filtering images to remove noise or to enhance geometric detail. We will then look at means by which we can characterise geometric properties like texture, and processes that allow us to analyse objects and shapes that appear in imagery.

Our methods here are called image domain techniques because the results are generated by working on the pixel brightness values directly. An alternative approach is based on the spatial frequency domain, in which the images are transformed first using Fourier and wavelet methods. Those are the subject of Chap. 7.

5.2 Neighbourhood Operations in Image Filtering

In contrast to the point operations used for radiometric modification of image data, techniques for geometric processing are characterised by operations over local neighbourhoods of pixels, as illustrated in Fig. 5.1. The result of a neighbourhood operation is still a modified brightness value for the single pixel at the centre of the neighbourhood; however the new value is determined by the brightnesses of all the local neighbours rather than just the original brightness value of the central pixel alone.

The neighbourhood shown in Fig. 5.1 is square and of dimension $3 \times 3$. It is defined by a template or window that is laid over the image. In practice the template can be any shape and size and, as we shall see, that partly defines the outcome of the resulting operation; usually though it has an odd number of cells horizontally and vertically so that it has a natural centre to place over the image pixel being processed.
While the template has been introduced as a means for defining the image neighbourhood of interest it most often has numbers associated with each of its cells as seen in Fig. 5.2; those numbers contribute to the outcome of the operation. The result is most easily expressed in the \((m, n)\) coordinate system of the template. The new brightness value for the pixel \((i, j)\) at the centre of the template is

\[
\begin{align*}
\text{in which } \phi(m, n) & \text{ are original pixel brightness values addressed according to their positions under the template; } t(m, n) \text{ is the corresponding template entry with its centre located at } (i, j). \\
The origin for the template coordinates is at the upper left hand corner. Sometimes the template entries collectively are referred to as its kernel and the template itself is sometimes called a mask. Different kernels will implement different geometric modifications of the image. Note that the coordinate system we have used here to address image pixels has its origin at the top left hand corner, consistent with referring to the first row as the “first line of pixels” as normally displayed.
\end{align*}
\]

If the template is filled entirely with zeros except for an entry of “1” at its centre then application of (5.1) will leave the image unchanged. While this may seem to be a strange and unproductive operation it will feature in compound template-based operations designed to achieve specific image processing outcomes, as we will see later.

Equation (5.1) shows the template operating algebraically on the image pixels. Non-algebraic operations are also possible as we will see later in this chapter. We could write (5.1) more generally as
new brightness value = $\phi(1,1)t(1,1) + \phi(1,2)t(1,2) \ldots \phi(3,3)t(3,3)$

Fig. 5.2 Demonstrating the operation of a numerical template of dimension $3 \times 3$ and the coordinate convention used, based on the template

\[ r(i,j) = T_{m,n}\{\phi(i,j)\} \quad (5.2) \]

in which the operator $T_{m,n}$ represents how the template processes the brightness of the pixels covered by the template and centred on $(i,j)$. It could be an operator whose kernel selects the maximum value of the pixels, or calculates their median value for example.

To produce a geometrically modified image the template is run over the original image line by line and column by column, centred on each pixel in turn, as illustrated in Fig. 5.3, creating new brightness values for the central pixels. Note that the border pixels cannot, in principle, be modified because they do not have a full set of neighbours. Often they are left unprocessed and removed because there are usually sufficiently many pixels in the original image that the lost edges are not a problem. Sometimes artificial borders are created outside the image to allow the actual borders to be processed. They are used in the generation of new edge pixel values but are not themselves replaced by a template response. The external artificial borders can be made up by replicating the actual border pixels. A theoretically more correct method for handling the borders derives from sampling theory, which we treat in Chap. 7. That considers an image as though it were just one cycle within an infinitely repeating sequence of images in both the horizontal and vertical directions. In that case the lines and columns of pixels outside the borders are those on the opposite sides of the image. Although appealing this is rarely done in practice.

The template based approach to the geometric modification of an image has a theoretical basis in the mathematical operation of convolution, which is treated in Sect. 5.8 following. Before looking at that we first examine some simple templates and see how they modify imagery. We will then look at convolution in a little detail so that we can develop a fuller understanding of the template method, and how it can be generalised.
5.3 Image Smoothing

Images often contain noise, which usually shows as random variations of the brightnesses of the pixels about the actual values associated with real features and objects in the scene. The noise can originate in the sensors that were used to acquire the data, from any communications channels used to carry the information, and from quantisation noise introduced when the raw signal data is digitised. One of the most common geometric processing operations carried out with imagery is to smooth the data to reduce the effect of noise. In this section we look at three approaches for smoothing.

5.3.1 Mean Value Smoothing

If the template has dimensions $M \times N$ and each of its entries has the value $1/MN$ then (5.1) becomes

$$r(i,j) = \frac{1}{MN} \sum_m \sum_n \phi(m,n)$$

(5.3)

which is the mean value of the pixels covered by the template. The new brightness for the pixel at the centre of the template is then the average brightness value in the neighbourhood of that pixel.

Figure 5.4 shows the result of running a $1 \times 3$ smoothing template over a single line of image data. The data contains an edge between bright (left hand) and dark (right hand) regions. Either side of the boundary the noise fluctuations have been reduced by the smoothing template. However the edge has been smeared out over several pixels. That degradation can be avoided if a threshold is applied to the
smoothing operation such that if the new brightness value is significantly different from its old value then the old value is used. That is implemented by the following algorithm. Let

\[
q_{i,j} = \frac{1}{MN} \sum_m \sum_n \phi(m,n)
\] (5.4a)

then

\[
\begin{align*}
    r(i,j) &= \rho(i,j) \quad \text{if } |\phi(i,j) - \rho(i,j)| < T \\
    &= \phi(i,j) \quad \text{otherwise}
\end{align*}
\] (5.4b)

where \(T\) is a user-specified threshold, which could be determined from a knowledge or estimate of the signal to noise ratio of the image. Figure 5.4 shows how the use of such a threshold can preserve boundaries and edges.

Templates of any shape and size can be used. Larger templates give more smoothing and greater loss of fine detail. We describe fine detail in terms of high spatial frequencies (see Chap. 7). If the detail changes rapidly across or down the scene it is said to be high frequency, whereas low spatial frequency features vary slowly across or down an image. Horizontal templates will smooth horizontal noise and detail but leave detail in the vertical direction unaffected. In Fig. 5.5 the results of applying several different smoothing templates to the same image are shown, to illustrate these points. Sometimes smoothing with a simple averaging template is called box car filtering.

**Fig. 5.4** Illustration of \(1 \times 3\) smoothing along a single line of image data showing the degradation of an edge, and its preservation if a threshold is used in the smoothing operation.
5.3.2 Median Filtering

An alternative to applying thresholding for avoiding edge deterioration when smoothing is to use a median filter. In this case the kernel of the operator in (5.2) is designed to find the median brightness of the pixels covered by the template, which is then used as the new brightness for the central pixel. Whereas the mean of a set of numbers is their average, the median is that number which sits in the middle of the set. For example, the median of the set 4, 6, 3, 7, 9, 2, 1, 8, 8 is 6, whereas its mean is 5.3. Figure 5.6 shows the effect of applying a median filter to the data of Fig. 5.4 compared with simple mean value smoothing. It can be seen that most of the boundary is preserved.
An application for which median filtering is well suited is the removal of impulse like noise. That is because pixels corresponding to noise spikes are atypical among their neighbours and will be replaced by the most typical pixel in the neighbourhood. Figure 5.7 shows the value of median filtering on an image that contains impulsive black and white noise (sometimes called salt and pepper noise).

![Figure 5.6](image_url)

**Fig. 5.6** Demonstration of the value of median filtering for preserving edges while smoothing noise, compared with mean value smoothing

5.3.3 Modal Filtering

Another form of smoothing filter replaces the brightness of the central pixel by that most commonly occurring among the pixels covered by the template. That is referred to as the *mode* of the set and is illustrated in Fig. 5.8.

5.4 Sharpening and Edge Detection

The opposite to smoothing, in which high spatial frequency detail is reduced, is image sharpening in which detail, including edges and lines, is enhanced. Two techniques are in common use and are treated in the following. While the procedures to be covered sharpen all high spatial frequency detail, the examples used are based on edge detection and enhancement. We also consider edge detection explicitly in Sect. 5.5.
5.4.1 Spatial Gradient Methods

There are several implementations of this approach but all depend on calculating the local gradient in the brightness of an image in a pair of orthogonal directions. Let $\nabla_1$ and $\nabla_2$ be measures of how the brightness changes in those directions, at right angles to each other. We define the magnitude of the change as

$$|\nabla| = \sqrt{\nabla_1^2 + \nabla_2^2}$$  \hspace{1cm} (5.5a)

and its direction as

$$\angle \nabla = \tan^{-1}\left(\frac{\nabla_2}{\nabla_1}\right)$$  \hspace{1cm} (5.5b)
The direction will not concern us any further here since it does not feature in sharpening operations. It is of interest when finding contours in imagery and for deriving slope and aspect models.

Application of (5.5a) will extract the high spatial frequency detail. It is usual to add that detail back to the original data so that the high frequencies are enhanced and the image appears sharper. Usually a weighted combination is employed along the lines of

\[
\text{sharpened image} = \text{original image} + c \times \text{high spatial frequency detail}
\]

where \( c \) is a constant that controls the degree of sharpening.

Different spatial gradient methods for sharpening are distinguished by how they estimate the orthogonal gradients \( r_1 \) and \( r_2 \):

We now look at the most commonly adopted definitions.

5.4.1.1 The Roberts Operator

If the estimates of \( r_1 \) and \( r_2 \) are produced from the differences between the brightnesses of pixels diagonally separated, i.e.

\[
\nabla_1 = \phi(i,j) - \phi(i + 1, j + 1) \tag{5.6a}
\]

\[
\nabla_2 = \phi(i + 1,j) - \phi(i,j + 1) \tag{5.6b}
\]

then the Roberts Operator is generated from (5.5a). In principle, it computes the gradient at the point \((i + \frac{1}{2}, j + \frac{1}{2})\), although the result is generally associated with the pixel at \((i,j)\). The application of the Roberts Operator to the artificial image in Fig. 5.9a, which has sharp horizontal and vertical transitions in brightness, is shown in Fig. 5.9b.
5.4.1.2 The Sobel Operator

Perhaps a better spatial gradient estimator than the Roberts Operator is the Sobel Operator which generates $\nabla_1$ and $\nabla_2$ in the horizontal and vertical directions centred on the pixel at $(i,j)$. While it derives most of its weight from the pixels immediately above, below and to the sides of that at $(i,j)$, it is also sensitive to the pixels diagonally about the pixel of interest. The individual orthogonal components of gradient are

\[
\nabla_1 = \{ \phi(i+1,j-1) + 2\phi(i+1,j) + \phi(i+1,j+1) \} - \{ \phi(i-1,j-1) + 2\phi(i-1,j) + \phi(i-1,j+1) \} \quad (5.7a)
\]

and

\[
\nabla_2 = \{ \phi(i-1,j+1) + 2\phi(i,j+1) + \phi(i+1,j+1) \} - \{ \phi(i-1,j-1) + 2\phi(i,j-1) + \phi(i+1,j-1) \} \quad (5.7b)
\]

which are equivalent to applying the $3 \times 3$ templates:

\[
\begin{array}{ccc}
-1 & -2 & -1 \\
0 & 0 & 0 \\
1 & 2 & 1 \\
\end{array}
\quad \begin{array}{ccc}
-1 & 0 & 1 \\
-2 & 0 & 2 \\
-1 & 0 & 1 \\
\end{array}
\]

These templates are separately run over the image in the manner of Fig. 5.3 and the results combined in (5.5a).

Applying the Sobel Operator to the image of Fig. 5.9a generates the result shown in Fig. 5.9c. Again, a threshold would be specified to highlight major transitions in brightness. In this case the boundaries in the image are highlighted by two rows of pixels, one either side of the boundary.

5.4.1.3 The Prewitt Operator

If the weightings of two for the pixels directly above and below, and to either side of, the central pixel in the calculations of (5.7a, b) are changed to 1 then the result is the Prewitt Operator, which has the template equivalents:

\[
\begin{array}{ccc}
-1 & -1 & -1 \\
0 & 0 & 0 \\
1 & 1 & 1 \\
\end{array}
\quad \begin{array}{ccc}
-1 & 0 & 1 \\
-1 & 0 & 1 \\
-1 & 0 & 1 \\
\end{array}
\]

Again, the templates are applied to the image and the results combined in (5.5a).

Note that for the Roberts, Sobel and Prewitt operators the template entries sum to zero. That is a feature of all operators that seek to highlight high spatial
frequency detail such as lines and edges, including the Laplacian operator below. The reason is that there should be zero response when those templates are applied to areas of constant image brightness. In contrast, the entries in the templates used for smoothing encountered earlier sum to a non-zero value, so that original image brightnesses are preserved in homogeneous regions.

5.4.1.4 The Laplacian Operator

Examination of (5.6a, b) shows that its two components are in fact the classical definition of the first derivative of the image brightness value function $\phi(i,j)$, albeit in the diagonal directions. It is, of course, possible to look at the first derivatives as incremental differences in the horizontal and vertical directions:

$$\nabla_{ij}^1 = \phi(i,j + 1) - \phi(i,j)$$ (5.8a)

$$\nabla_{ij}^2 = \phi(i + 1,j) - \phi(i,j)$$ (5.8b)

in which the superscripts have been added to indicate the pixel address from which the difference is computed.

The Laplacian operator\(^1\) is based on estimating the second derivatives in the horizontal and vertical directions, and summing them. The second derivative is the derivative of the first derivative—it is a measure of how rapidly the first derivative changes. Consider the horizontal direction first. The first derivative from the $j$th to the $(j + 1)$th pixel is given by (5.8a). In a similar way the first derivative (the gradient) from the $(j - 1)$th to the $j$th pixel is

$$\nabla_{ij}^{1j-1} = \phi(i,j) - \phi(i,j - 1)$$

The change in those first derivatives is the difference

$$\nabla_{ij}^1 - \nabla_{ij}^{1j-1} = \phi(i,j + 1) - \phi(i,j) - \phi(i,j) + \phi(i,j - 1)$$

$$= \phi(i,j + 1) - 2\phi(i,j) + \phi(i,j - 1)$$ (5.9a)

which is the second derivative at the point $i,j$. Similarly in the vertical direction the change from the $i$th to the $(i + 1)$th pixel, less that from the $(i - 1)$th to the $i$th pixel, i.e., the vertical second derivative at the point $i,j$, is

$$\nabla_{ij}^2 - \nabla_{ij}^{1i-1j} = \phi(i + 1,j) - 2\phi(i,j) + \phi(i - 1,j)$$ (5.9b)

The Laplacian operator is the sum of (5.9a, b) and given the symbol

$$\nabla^2 \phi(i,j) = \phi(i - 1,j) + \phi(i,j - 1) - 4\phi(i,j) + \phi(i,j + 1) + \phi(i + 1,j)$$ (5.10)

---

which is equivalent to the template

\[ \nabla^2 \equiv \begin{bmatrix}
0 & +1 & 0 \\
+1 & -4 & +1 \\
0 & +1 & 0
\end{bmatrix} \quad \text{or} \quad \begin{bmatrix}
0 & -1 & 0 \\
-1 & +4 & -1 \\
0 & -1 & 0
\end{bmatrix} \]

The second version, with the signs reversed, is often encountered in practice, particularly in the context of unsharp masking treated in the next section.

Sometimes the second derivatives in the two diagonal directions are added as well to give the templates

\[ \nabla^2 \equiv \begin{bmatrix}
+1 & +1 & +1 \\
+1 & -8 & +1 \\
+1 & +1 & +1
\end{bmatrix} \quad \text{or} \quad \begin{bmatrix}
-1 & -1 & -1 \\
-1 & +8 & -1 \\
-1 & -1 & -1
\end{bmatrix} \]

### 5.4.2 Subtractive Smoothing (Unsharp Masking)

As seen in Figs. 5.4a, b and 5.5a, b a smoothed image retains low spatial frequency detail but has its high spatial frequency features such as edges and lines attenuated (unless thresholding has been employed to preserve sharp transitions). If a smoothed image were subtracted from the original we would be left, therefore, with just the high spatial frequency detail—largely the edges and lines. That is illustrated in Fig. 5.10, using the data of Fig. 5.4. If the high frequency detail so detected is then added back to the original data the result is an image in which the higher spatial frequencies, including edges and lines, are enhanced, as seen in Fig. 5.10c.

The difference operation can result in negative values as seen in Fig. 5.10b. Provided the result is not for display that is not a problem. If display is required, a fixed brightness value offset can be added to all pixels and the results rescaled to the display brightness range so that mid grey represents no difference. Positive differences will be brighter than the mid-range and negative differences will be darker. The same scaling approach is adopted for the final result in which the difference image has been added back to the original. Again, the result is scaled to fit within the allowed brightnesses of the display device.

Figure 5.11 shows the technique applied to each of three bands of a Landsat Multispectral Scanner colour composite image. As seen, the sharpened image has clearer high spatial frequency detail, although there is also a tendency for noise to be enhanced.
The technique commonly goes by the name of unsharp masking; the high frequency image that results from subtracting a smoothed version from the original is sometimes referred to as a mask. Although it apparently requires three steps (smoothing, subtraction, adding to the original) it can in fact be implemented by a single template that combines those steps into a single operation, as the template arithmetic shown in Fig. 5.12 demonstrates. Sharpening of a colour image product can also be performed using the pan sharpening procedure treated in Sect. 6.8.

5.5 Edge Detection

The operators defined in Sect. 5.4 effectively detect edges, although as we have seen they will enhance high spatial frequency detail in general. If the gradient estimators $\nabla_1$ and $\nabla_2$ are kept separate and not combined in the magnitude
operation of (5.5a) they will individually detect edges in the vertical and horizontal
directions. That can be seen by looking at the structures of the templates for the
Roberts, Sobel and Prewitt operators.

**Fig. 5.11** Subtractive smoothing (unsharp masking) for image sharpening

this template leaves the
image unchanged, apart
from loss of borders

<table>
<thead>
<tr>
<th>0 0 0</th>
<th>1/9 1/9 1/9</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 1 0</td>
<td>1/9 1/9 1/9</td>
</tr>
<tr>
<td>0 0 0</td>
<td>1/9 1/9 1/9</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>0 0 0</th>
<th>-1/9 -1/9 -1/9</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 1 0</td>
<td>-1/9 17/9 -1/9</td>
</tr>
<tr>
<td>0 0 0</td>
<td>-1/9 -1/9 -1/9</td>
</tr>
</tbody>
</table>

original—smoothed
(high frequency mask)

original

equivalent single
kernel

**Fig. 5.12** The steps in unsharp masking, and a single equivalent template
It is possible to build templates with kernels that detect edges in diagonal directions as well. Suitable Roberts and Sobel diagonal edge detectors are

\[
\begin{align*}
\nabla_{45^\circ}^+ & \equiv \begin{bmatrix} +1 & +1 & 0 \\ +1 & 0 & -1 \\ 0 & -1 & -1 \end{bmatrix} & \nabla_{-45^\circ}^+ & \equiv \begin{bmatrix} 0 & +1 & +1 \\ -1 & 0 & +1 \\ -1 & -1 & 0 \end{bmatrix} \\
\nabla_{45^\circ}^- & \equiv \begin{bmatrix} +2 & +1 & 0 \\ +1 & 0 & -1 \\ 0 & -1 & -2 \end{bmatrix} & \nabla_{-45^\circ}^- & \equiv \begin{bmatrix} 0 & +1 & +2 \\ -1 & 0 & +1 \\ -2 & -1 & 0 \end{bmatrix}
\end{align*}
\]

Other, more sophisticated forms of edge detector are available including the Marr-Hildreth and Canny operators; however they are not often encountered in remote sensing applications.

### 5.6 Line and Spot Detection

Linear features such as rivers and roads are usually detected as pairs of edges if they are more than a pixel in width. If they are almost exactly one pixel wide then they can be extracted with the templates

\[
\begin{align*}
\text{vertical} & \equiv \begin{bmatrix} -1 & 2 & -1 \\ -1 & 2 & -1 \\ -1 & 2 & -1 \end{bmatrix} & \text{horizontal} & \equiv \begin{bmatrix} -1 & -1 & -1 \\ 2 & 2 & 2 \\ 2 & 2 & 2 \end{bmatrix} & \text{diagonal} & \equiv \begin{bmatrix} -1 & -1 & 2 \\ -1 & 2 & -1 \\ -1 & 2 & -1 \end{bmatrix}
\end{align*}
\]

Examination of the form of the Laplacian operator in Sect. 5.4.1.4 suggests it is suited to the detection of spots in imagery, characteristically about a single pixel in size.

---

2 Gonzalez and Woods, *loc. cit.*
5.7 Thinning and Linking

The outputs from the above operators can often contain breaks in the edges and lines detected because of noise and other local variations in the original image data. Also, as we have seen, some lines and edges may be thicker than necessary. Should an edge or line map of high integrity be required the operator outputs may need tidying up by linking edge and line features that are separated by breaks, and by thinning others. Thinning and linking are not operations encountered regularly in remote sensing. Good treatments can be found in standard, more general image processing texts.\(^3\) The morphological operations given in Sect. 5.11 can also be employed for these purposes.

5.8 Geometric Processing as a Convolution Operation

While the template method for smoothing and sharpening presented above is intuitive, it does in fact have a theoretical basis. In this section we develop the more formal convolution approach to geometric processing. Having done so we will understand more clearly the origins of the methods just treated and will be able to devise still more sophisticated templates. The convolution approach also allows a direct comparison of these image domain techniques with those based on the Fourier transformation that operate in the spatial frequency domain.

Convolution is a process that occurs surprisingly often in physical systems including optical imaging devices, imaging radar, and the transmission of signals through electronic circuits. We introduce the concept here for a signal passing through some unspecified system, and then generalise it to the case of images.

Suppose we have a function of time \( f(t) \). If it is passed through some system,\(^4\) such as an amplifier, a telephone network, a link to a satellite in space or similar, as shown in Fig. 5.13, then the signal \( y(t) \) that emerges from the system is given by the convolution integral

\[
y(t) = \int_{-\infty}^{\infty} f(\tau) h(t - \tau) d\tau \quad \text{or} \quad f(t) \ast h(t)
\]

in which \( h(t) \) describes the properties of the system through which the signal passes. It is sometimes called its transfer function, but is more properly called its

---


\(^4\) This assumes the system is linear, which is almost always the case for those we encounter in image processing. A linear system is one in which adding two inputs gives a response which is the sum of the individual responses.
impulse response. The convolution operation also features strongly in the handling of digital signals and images as seen in the material of Sects. 7.6 and 7.10.

Note that there is a dummy variable $\tau$ in (5.11) which describes the various functions inside the integral and disappears once the integral is performed. Also, note that we adopt the symbol $\ast$ to represent the convolution of two functions. Detailed treatments of the concept of convolution in the context of processing images will be found in Castleman\textsuperscript{5} and Gonzalez and Woods.\textsuperscript{6}

Equation (5.11) is the convolution integral in just one dimension—time. For images we have two independent variables—the spatial coordinates that describe pixel location. Consequently we need a two dimensional version of the convolution integral.

Even though the pixels in a remotely sensed image are described by discrete image coordinates, for the moment assume that any point in a scene can be described by the pair of continuous variables $x$ and $y$ and that the scene properties are represented by the brightness function $\phi(x,y)$. Now suppose we image or view that scene through some form of filter or lens system that has the two dimensional impulse response $k(x,y)$. Then the resulting image is given by the two dimensional convolution

$$r(x,y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \phi(u,v)k(x-u,y-v)\,dudv$$  \hspace{1cm} (5.12)

We have referred to $k(x,y)$ here as the impulse response of the system that operates on the signal; we have given it the symbol $k$ to emphasise that it is a kernel, similar to the templates used earlier. It is also often referred to as the system function. If the system represents an imaging device then $k(x,y)$ would be the device’s point spread function; (5.12) then describes the degradation of the properties of the scene observed in an image. Alternatively, if $k(x,y)$ is one of the kernels specified earlier\textsuperscript{7} for geometrically processing an existing image, then (5.12) is a mathematical specification of the sliding template operation illustrated in Fig. 5.3; in this case $\phi(x,y)$ and $r(x,y)$ represent the original and processed versions of the image respectively.

For digital image data $r(x,y)$, $\phi(x,y)$ and $k(x,y)$ are discrete rather than continuous and have limited ranges in the two coordinate directions, not extending to $\pm \infty$ as implied by the integrals in (5.12). It is necessary therefore to modify (5.12) so it can be applied to imagery. The integrals will be replaced by discrete sums and

\textsuperscript{5} Castleman, \textit{ibid.}
\textsuperscript{6} Gonzalez and Woods, \textit{loc. cit.}
\textsuperscript{7} With one small modification, seen in the following paragraphs.
the continuous functions by their digital counterparts. If we let \((i,j)\) be the discrete versions of \((x,y)\) and \((m,n)\) be discrete values of the integration variables \((u,v)\) then (5.12) is written

\[
  r(i,j) = \sum_m \sum_n \phi(m,n)k(i - m,j - n)
\]  

(5.13)

The sums are taken over all values of the dummy variables \(m,n\). Strictly, the ranges of \(m,n\) are the same as the ranges of \(i,j\) and they have the same origin in the manner in which (5.13) is expressed. Similarly, the template is, in principle, the same size as the image and its origin is the same as the image origin.\(^8\)

To see how (5.13) would be used in practice it is necessary to interpret the sequence of operations it incorporates. The negative signs on \((m,n)\) in (5.13) imply reflections through each of the \(m\) and \(n\) axes. That is equivalent to a rotation of the system function \(k(i,j)\) by \(180^\circ\) before it is used in (5.13). We call the rotated function \(t(m - i,n - j)\).

Next, (5.13) says that the value of \(r(i,j)\) is given by multiplying, for all values of \(m, n\), the image and the rotated system function and then summing the result. That gives the new value for the pixel at the specific location \((i,j)\)—in other words just for one pixel in the new image. To get the modified values for all pixels we need to adopt in turn all values of \((i,j)\) in (5.13), which has the effect of relocating the template to each \((i,j)\), pixel by pixel, as in Fig. 5.3.

The pixel and template origins in (5.13) are in their respective upper left hand corners. It is more convenient in practice, when we restrict the templates or kernels to a small neighbourhood about \((i,j)\), to address pixel and template entries by coordinates which have their origin at the upper left hand corner of the finite sized template. That allows (5.13), with the kernel \(k(i - m,j - n)\) replaced by the rotated system function \(t(m - i,n - j)\), to be re-expressed in simple form as (5.1).

The templates of the previous sections, and in (5.1) in particular, are equivalent to the rotated version of the system function \(k(m,n)\). Consequently, any geometric processing operation that can be modelled by convolution can also be expressed in template form. For example, if the point spread function of an imaging device is known then an equivalent template can be derived for computing what the image of a scene will look like, noting that the \(180^\circ\) rotation is important if the system function is not symmetric.

Templates for altering the geometric properties of an image can be chosen intuitively, as with smoothing in Sect. 5.3, or can be designed using a knowledge of filtering in the spatial frequency domain, based on the Fourier transformation discussed in Chap. 7.

\(^8\) There is a further subtlety that only becomes apparent when sampling theory is understood (see Chap. 7). When a continuous image is sampled to convert it to digital form it effectively becomes just one cycle in each dimension of an infinite periodic repetition of samples; the same is the case for the template. In practice, we often ignore that property. It is however important when considering the interaction with a template that is comparable in size to the image.
5.9 Image Domain Techniques Compared with Using the Fourier Transform

Most geometric processing operations can be implemented using either the image domain procedures of this chapter or the Fourier transformation to be treated in Chap. 7. Which option to adopt depends on several factors, such as user familiarity and processing limitations. The Fourier transform method is more flexible and allows a much greater range of processing operations to be applied. Another consideration relates to processing time. This matter is pursued here in order to indicate, from a cost viewpoint, when one method might be chosen in favour of the other.

The Fourier transform spatial frequency domain process and the template approach both consist only of sets of multiplications and additions. No other mathematical operations are involved. It is sufficient, therefore, to make a time comparison based upon the number of multiplications and additions necessary to achieve a result. Here we will ignore the additions since they are generally faster than multiplications in most cases and also since they are comparable in number to the multiplications involved. For an image of $K \times K$ pixels, and a template of size $M \times N$, the total number of multiplications necessary to evaluate (5.1) for every image pixel (ignoring any difficulties with the edges of the image) is

$$N_C = MNK^2$$

From the material presented in Sect. 7.9 it can be seen that the number of (complex) multiplications required in the frequency domain approach is

$$N_F = 2K^2 \log_2 K + K^2$$

A processing time comparison is, therefore, given by

$$\frac{N_C}{N_F} = MN/(2\log_2 K + 1)$$

When this figure is below unity it is more economical to use the template operator approach. Otherwise the Fourier transform method is more cost effective. That does not take into account program overheads, such as the bit shuffling required in the frequency domain approach and the added cost of complex multiplications; however it is a reasonable starting point in choosing between the methods.

Table 5.1 shows values of $N_C/N_F$ for various image and template sizes, from which it is seen that, provided a $3 \times 3$ template will implement the operation required, it is always more cost-effective than processing based on the Fourier transformation. Similarly, a rectangular $3 \times 5$ template is more cost effective for practical image sizes. The spatial frequency domain technique is seen to be economical if very large templates are needed, although only marginally so for large images. Note, however, that the frequency domain method is able to implement processes not possible (or at least not viable) with template operators. Removal of periodic noise is one example. That is particularly simple in the spatial frequency
domain but requires unduly complex templates or even nonlinear operators (such as median filtering) in the image domain. Nevertheless, the template approach is a popular one since often $3 \times 3$ and $5 \times 5$ templates are sufficient in many cases.

5.10 Geometric Properties of Images

The grey level histograms treated in Chap. 4 summarise the radiometric properties of images. We now look at a number of measures that characterise geometric structure. Image geometry is effectively characterised by the inter-relationships between pixels at different locations. For example, sets of pixels in a given neighbourhood may describe an object, such as a field or roadway, while repeating patterns of pixels define any texture-like qualities in a scene. We now look at both of those concepts.

5.10.1 Measuring Geometric Properties

When thinking about the geometric nature of an image a logical consideration is how related adjacent pixels might be. In agricultural regions, for example, it is highly likely that neighbouring pixels will be of the same ground cover type and thus will be similar in brightness in each of the recorded data channels. Also, road and river systems consist of sets of connected pixels of comparable brightness. We are then led to think about means by which we might describe the spatial relationships between pixels. In Chap. 8, when we look at image classification, we will do that using a set of conditional probabilities that describe the likelihoods of neighbouring pixels being from the same ground cover class. Here we are interested in measures that apply to the pixel brightnesses in a single channel of image data.

A simple, and obvious measure, is to compare the brightnesses of pixels by taking their differences. Although logically we might look at adjacent pixels there is also value in comparing pixels further apart. Among other things that will lead us to definitions for texture. If we use the symbol $k \in \{i,j\}$ to represent either the

---

### Table 5.1 Time comparison of geometric processing using templates compared with the Fourier transformation approach; this is based upon a comparison of multiplications (real or complex)

<table>
<thead>
<tr>
<th>Image size</th>
<th>Template size</th>
<th>3 x 3</th>
<th>3 x 5</th>
<th>5 x 5</th>
<th>5 x 7</th>
<th>7 x 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>128 x 128</td>
<td></td>
<td>0.60</td>
<td>1.00</td>
<td>1.67</td>
<td>2.33</td>
<td>3.27</td>
</tr>
<tr>
<td>256 x 256</td>
<td></td>
<td>0.53</td>
<td>0.88</td>
<td>1.47</td>
<td>2.06</td>
<td>2.88</td>
</tr>
<tr>
<td>512 x 512</td>
<td></td>
<td>0.47</td>
<td>0.79</td>
<td>1.32</td>
<td>1.84</td>
<td>2.58</td>
</tr>
<tr>
<td>1024 x 1024</td>
<td></td>
<td>0.43</td>
<td>0.71</td>
<td>1.19</td>
<td>1.67</td>
<td>2.33</td>
</tr>
<tr>
<td>2048 x 2048</td>
<td></td>
<td>0.39</td>
<td>0.65</td>
<td>1.09</td>
<td>1.52</td>
<td>2.13</td>
</tr>
<tr>
<td>4096 x 4096</td>
<td></td>
<td>0.36</td>
<td>0.60</td>
<td>1.00</td>
<td>1.40</td>
<td>1.96</td>
</tr>
</tbody>
</table>
row or column index of a pixel, then the brightness difference of two pixels from the same channel, spaced $h$ apart along a given row or down a given column, is

$$\phi(k) - \phi(k + h)$$

Usually single pixel measures are not that helpful. Instead, we might be interested in the difference between all pixels in the image and their neighbours $h$ away as some sort of measure of similarity or correlation over a given separation. To do so we could average the brightness differences, but of course that risks the cancellation of positive and negative differences, so we average the squared distances instead, to get

$$\text{var} = \frac{1}{K} \sum_{k=1}^{K} \left( \phi(k) - \phi(k + h) \right)^2$$

which will be recognised as a variance-like measure—indeed it is the variance in pixel brightness in the direction of interest; $K$ is the number of brightness pairs chosen for the computation. Often that might be all the available pixel pairs in that direction.

If we vary the separation $h$ then we can construct a graph that shows the variance (essentially how different the pairs of pixels are on the average) as a function of separation. The graph is called a variogram. Sometimes half the variance is plotted; we then have a semivariogram. As the semivariance increases the correlation or similarity of pixels on the average decreases. Conversely, highly correlated pixel pairs will exhibit small semivariance and will plot low on a semivariogram. If there is spatial periodicity in the landscape the semivariogram will reflect that behaviour.

Several properties can be derived from the semivariogram, which are best illustrated on the idealised form shown in Fig. 5.14; they include the sill (its asymptotic maximum value, if it exists), the nugget variance (the extrapolated point of intersection with the ordinate), sometimes taken to indicate the noise properties of the image since it represents variance that is not related to the spatial properties of the scene, and the range, which is the lag or separation at which the sill is reached.

The image of Fig. 5.15 is a Landsat ETM+ image of a region surrounding Canberra, the federal capital of Australia. Figure 5.16 plots horizontal semivariograms for the four regions indicated.

### 5.10.2 Describing Texture

Many of the surfaces we observe about us in our day to day life are textured—that is they seem to have some form of quasi-repeating pattern that readily tells us whether the surface is moderately smooth, rough but nominally repetitive (such as a carpet) or rough and strongly repetitive, such as a piece of linen cloth. The same is the case for satellite imagery. Grassland, crops and forests all appear differently textured to our observation—that is they are composed of some form of natural
scale that seems to repeat on the average. While we can describe textures qualitatively, and can certainly discriminate among them using our own eyes, quantitative characterisation of texture is not so simple. We have to start by finding a measure that captures the spatial properties of a scene which reveal texture. Those properties have to do with repetition, on the average.

A long-standing spatial measure is the grey level co-occurrence matrix (GLCM) defined in the following way. To make the development simple, imagine we want to detect a component of texture just in the horizontal direction in a particular region. To

---

do that we could see how often two particular grey levels occur in that direction in the selected region, separated by a specified distance. Let \( g(\phi_1, \phi_2|h, \theta) \) be the relative occurrence of pixels with grey levels \( \phi_1 \) and \( \phi_2 \) spaced \( h \) apart in the \( \theta \) direction—for the moment chosen as the horizontal direction. Relative occurrence is the number of times each grey level pair occurs, divided by the total possible number of grey level pairs. The GLCM for the region of interest is the matrix of those measurements over all grey level pairs; there will be as many GLCMs as there are values of \( h \) and \( \theta \). If the pixels are described by \( L \) possible brightness values then the matrix will be \( L \times L \). Given that \( L \) can be quite large, the brightness value range is often reduced by looking for co-occurring pairs of brightness value ranges. Alternatively, the dynamic range of the data can be reduced (e.g., from 10 bit to 5 bit data) before processing. We generally look for similar behaviour in other directions, such as vertically and diagonally, in which case there would be four matrices for the chosen values of \( h \):

Often \( h \) is used as a variable to see whether texture exists on a local or more regional scale. Sometimes the GLCMs computed for various values of \( \theta \) are kept separate to see whether the texture is orientation dependent; alternatively they are averaged on the assumption that texture will not vary significantly with orientation.

Once we have the GLCMs for the region of interest it is then usual to set up a single metric computed from each matrix to use as a texture descriptor. A range of measures is possible one of which is to describe the entropy of the information contained in the GLCM, defined by

\[
H(h, \theta) = - \sum_{\phi_1=1}^{L} \sum_{\phi_2=1}^{L} g(\phi_1, \phi_2|h, \theta) \log \{ g(\phi_1, \phi_2|h, \theta) \} 
\]

Entropy will be highest when all entries in the GLCM are equiprobable—when the image is not obviously textured—and will be low when there is a large disparity in the probabilities, as happens when significant texture is present. Another measure is energy which is the sum of the squared elements of the GLCM. It will be small when the GLCM elements are small, indicating low texture.
Figure 5.17 shows graphs of entropy and energy for the four regions indicated in Fig. 5.15, within which just the horizontal GLCMs were computed for a range of values of lag, \( h \). Those calculations used just the first ETM+ band—the visible blue, which was reduced in dynamic range to 5 bits before any calculations were performed. Two points are noteworthy. First, entropy increases with and energy decreases with lag, indicating that the texture is falling away at larger spacing. Secondly the four cover types chosen—grass, forest, mountainous and suburban—are separable by their texture, with grassland exhibiting the strongest texture. The suburban and mountainous regions are seen to be low in texture by comparison, and are comparable to each other for the range of scales chosen. Note that entropy and energy behave oppositely to each other as expected.

5.11 Morphological Analysis

While most interest in digital processing of remotely sensed imagery relates to enhancing radiometric and geometric properties, or to interpreting cover types using the mapping and labelling procedures to be treated in later chapters, there are occasions when we are interested in specific objects. Those objects will be defined by connected groups of pixels and could represent agricultural fields, river systems and road networks, or the building blocks of an urban scene.

Often those objects are not well defined. The image of an agricultural field may have less regular boundaries than is the case in reality because of system noise, or limitations in the analytical procedures used to identify the field. Similarly, a homogeneous field may end up with incorrect inclusions of other cover types. Roads and river networks, likewise, might exhibit gaps and have variable thicknesses, again because of limitations in the processing algorithms, such as the line detectors of Sect. 5.6, and the nature of the data available. The morphological operations to be introduced in this section are helpful in cleaning up such problems by operating on the objects themselves.
Morphological processing is a template based operation of the form described in (5.2). As with the template operations we looked at earlier, the choice of the operator $T_{m,n}$ establishes how the object in an image is modified.

The images that contain the objects in which we are interested in this treatment have binary brightness values for each pixel. Said another way, the background has one value and all the pixels that define the object have a different brightness value. Figure 5.18 contains such an object, which might be a field. Whether it consists of a single field or a long narrow field adjacent to a larger rectangular field is hard to discern because of the join between what could be two objects; that may be the result of the processing operation that led to the object(s) being identified in the image in the first place. Also shown in Fig. 5.18 is an elongated object that emulates a river, that might have been extracted from an image using a sharpening template.

In morphological processing the template operation $T_{m,n}$ is defined in terms of a structuring element (SE), which is effectively a template in which the elements are present or not present, often represented respectively by template entries of 1 or 0. As with the geometric processing operations considered earlier, the SE is placed over each image pixel in turn, similar to the process shown in Fig. 5.3. For each location, the result of applying the structuring element to the image is a decision as to whether the pixel under the centre of the SE is a member of the object or not. That decision will depend on how the entries in the structuring element are used with respect to the object pixels being examined. This is a logical, rather than an algebraic, operation.

Note that there is no concept of generating a new brightness value for the pixel. Instead, the outcome of the operation is whether the pixel is a member of the set of pixels that define the object, or a member of the set of pixels that define the background. That leads us naturally to consider morphological processing in the language of set theory.

The structuring element can, in principle, be any shape or size. We will choose a $3 \times 3$ example to illustrate the essential points about morphological processing but it will become obvious how to use other variants. Two $3 \times 3$ examples are shown in Fig. 5.19. Both can be represented as a box with binary elements. In image processing it is common to define the SE by rectangles and to indicate by 1
those cells which take part in the morphological processing and by 0 those cells which do not, as shown.

We will now introduce some specific processing operations.

### 5.11.1 Erosion

As its name implies this operation has the effect of eroding, and thus reducing the size of, an object. It has the particular advantage that it can help to reduce ragged edges. It is defined by deciding that a pixel is part of an object if the SE, when centred on that pixel, is completely enclosed by the object. As illustrated in Fig. 5.20, if any part of the SE is outside the object, even though centred on an object pixel, that pixel is removed from the object. Thus the size of the object is reduced.

We can express this operation in the notation of set theory. Let $S$ represent the set of members of the structuring element and $O$ represent the set of pixels from which the object is composed. If the structuring element is centred on the image pixel at $(i, j)$ then we represent that by adding the pixel address as subscripts to the SE, viz. $S_{i,j}$. If, at a particular location, the structuring element is completely inside the object then it can be said to be a sub-set of the object, which is written $S_{i,j} \subseteq O$. The eroded object is the set of pixels with addresses $(i,j)$ that satisfy that subset condition. If we let $E$ represent the set of pixels that describe the eroded object then we express that object as

$$E = O \ominus S = \{i,j | S_{i,j} \subseteq O\} \quad (5.16)$$

In words, this equation says that the eroded object is the set of points $(i,j)$ that, when used as the centre of the SE, satisfy the condition that the SE is a sub-set of (is completely enclosed within) the original object. The symbol $\ominus$ is used to represent erosion\(^{10}\). Figure 5.21 shows the effect of eroding the objects of Fig. 5.18 with a square $3 \times 3$ SE. As seen the outcome is a greatly reduced object, with protrusions and thin linkages reduced. In the case of the river object, since its width is no greater than three pixels at any point, it is totally eroded away.

---

\(^{10}\) Castleman uses $\odot$, which some other authors use for a thinning operation.
5.11 Morphological Analysis

5.11.2 Dilation

Dilation has the opposite effect on an object to erosion; it has the tendency to grow the object’s size and to fill in holes. It is defined by deciding that a pixel is part of an object if the SE, when centred on that pixel, partly overlaps the object. This is illustrated in Fig. 5.22, showing that the size of the object is expanded. How can we write this in set notation? We start by saying that the only SE centre locations that are not part of the dilated object are those for which the SE lies entirely outside the original object. If the set of SE members lies outside the set of object members then their intersection will be the null set. In dilation we are happy to accept any location as part of a dilated object provided the SE has a non-null intersection with the original object. Using the same set notation as above, but calling the set of pixels that describe the dilated object \( D \), then

\[
D = O \oplus S = \{i, j\mid S_{ij} \cap O \neq \emptyset\}
\]  

(5.17)
in which $\emptyset$ is the null set. The symbol $\oplus$ is used to signify dilation. Figure 5.23 shows the effect of dilating the objects of Fig. 5.18 with a square $3 \times 3$ SE. As seen, the outcome is a greatly expanded object, with small holes closed up. The gaps in the river object are closed but at the expense of a broadened line overall.

**5.11.3 Opening and Closing**

Erosion and dilation are not the inverse of each other; in other words the original objects cannot be recovered after dilation by applying an erosion operation, and vice versa. However, the sequential application of erosion and dilation, or the sequential application of dilation and erosion, give interesting operations in their own right.
Erosion followed by dilation is referred to as opening, while dilation followed by erosion is called closing. The reasons for these names will be apparent from the results of their application. Using these definitions we can write the set of pixels (the object) that results from an opening operation, in which the bracketed operation is performed first, as

$$P = O \circ S = (O \oplus S) \ominus S$$  \hspace{1cm} (5.18)$$

and the object that results from a closing operation as

$$C = O \cdot S = (O \ominus S) \oplus S$$  \hspace{1cm} (5.19)$$

The results of applying these to the objects in Fig. 5.18 are shown in Fig. 5.24.

5.11.4 Boundary Extraction

Since erosion shrinks the boundaries of an object, subtracting an eroded version from its original will effectively isolate the boundaries. In the notation of sets this is written
in which $B(O)$ is the set of pixels different between the object and its eroded version. Using this approach the boundaries of the field object in Fig. 5.18 are shown in Fig. 5.25.

### 5.11.5 Other Morphological Operations

A range of other morphological operations is possible and finds wide application in fields in which visual interpretation of imagery is common, including medical imaging, astronomy and handwriting analysis.\(^\text{11}\) It is also possible to set up morphological techniques to operate on grey scale images, as against the binary image data considered here.

In remote sensing morphological processing has been extended to a high degree of sophistication, including its application in detailed analysis of high spatial and spectral resolution imagery.\(^\text{12}\)

---


5.12 Shape Recognition

Analysis of shapes in remote sensing imagery has not been as common as in other fields such as robotics and computer vision. Presumably that is because the resolution generally available in the past was insufficient to define shape with any degree of precision. However with pixels of the order of 1 m resolution now available, shapes such as rectangular and circular pivotal irrigation fields in agriculture, and urban features, are now easily discerned.

Shape analysis can be carried out using template techniques, in which the templates are chosen according to the shape of interest. The operation required is one of correlation and not the convolution operation of (5.1) and (5.13). We met correlation in Sect. 2.20.1; it is defined by (5.13) but with additions in place of subtractions. That yields an operation that has a maximum response when the kernel matches as closely as possible the underlying segment of image data in shape, size and orientation. The major difficulty with this approach, which as a consequence renders the technique of limited value in practice, is the need to match shape, size and orientation exactly. Other methods therefore are often employed. They include the adoption of shape factors, moments of area and Fourier transforms of shape boundaries. In each of these the shape must first be delineated from the rest of the image. That is achieved by histogram slicing (to separate objects from backgrounds), quantitative analysis and edge and line detection processes.13

5.13 Bibliography on Geometric Processing and Enhancement: Image Domain Techniques

Many of the books that cover radiometric enhancement of images also include good treatments of geometric processing. They include


The classic reference on texture analysis and processing is

13 See S. Loncaric, A survey of shape analysis techniques, Pattern Recognition, vol. 31, no. 8 August 1988, pp. 983–1001
For a comprehensive overview of recent advances in morphological processing applied to remote sensing images, and which includes a good reference list of salient contributions in morphology, see


The range of shape analysis techniques in common use in image processing will be found in


Castleman, loc. cit., contains a short overview of common shape analysis methods.

5.14 Problems

5.1 The template entries for line and edge detection sum to zero whereas those for smoothing do not. Why?

5.2 Repeat the example of Fig. 5.10 but by using a \([1 \times 5]\) smoothing operation in part (a), rather than \([1 \times 3]\) smoothing.

5.3 Repeat the example of Fig. 5.10 but by using a \([1 \times 3]\) median filtering operation in part (a) rather than \([1 \times 3]\) mean value smoothing.

5.4 An alternative smoothing process to median and mean value filtering using template methods is modal filtering. Apply \([1 \times 3]\) and \([1 \times 5]\) modal filters to the image data of Fig. 5.6. Note differences in the results compared with mean value and median smoothing, particularly around the edges.

5.5 Suppose \(S\) is a template operation that implements smoothing and \(O\) is the template operator that leaves an image unchanged. Then an edge enhanced image created by unsharp masking (Sect. 5.4.2) can be expressed

\[
\text{new image} = O\ (\text{old image}) + O\ (\text{old image}) - S\ (\text{old image})
\]

Rewrite this expression to incorporate two user defined parameters \(A\) and \(B\) that will cause the formula to implement any of smoothing, edge detection or edge enhancement.

5.6 This requires a vector algebra background. Show that template methods for line and edge detection can be expressed as the scalar product of a vector composed from the template entries and a vector formed from the neighbourhood of pixels currently covered by the template. Show how the angle between the template and pixel vectors can be used to assess the edge or line feature to which a current pixel most closely corresponds.
5.7 The following kernel is sometimes convolved with image data. What operation will it implement?

\[
\begin{array}{ccc}
0 & -1 & 0 \\
-1 & +4 & -1 \\
0 & -1 & 0 \\
\end{array}
\]

5.8 Consider the middle pixel shown in the figure below and calculate its new value if

- a [3 x 3] median filter is applied,
- a [3 x 3] unsharp mask is applied,
- a [1 x 3] image smoothing template with a threshold of 2 is applied, and
- the Sobel operator is applied.

\[
\begin{array}{ccc}
3 & 7 & 0 \\
8 & 1 & 1 \\
7 & 2 & 9 \\
\end{array}
\]

5.9 Image smoothing can be performed by template operators that implement averaging or median filtering. Compare those methods, particularly as they affect edges. Would you expect median filtering to be useful in edge enhancement using unsharp masking?

5.10 If a [3 x 3] smoothing template is applied to an image twice in succession how many neighbours will have played a part in modifying the brightness of a given pixel? Design a single template to achieve the same result in one pass.

5.11 The kernel function \( k(\cdot, \cdot) \) in either (5.12) or (5.13) can be used to demonstrated the degrading effect of the point spread function (PSF) of an imaging sensor on a scene being recorded. If \( \phi(\cdot, \cdot) \) is the ideal image of the scene and \( k(\cdot, \cdot) \) is the instrument PSF, what form should \( k(\cdot, \cdot) \) take in order that the instrument cause minimum degradation to the image data?

5.12 Inspection of Figs. 5.18, 5.21 and 5.23 demonstrate that erosion of an object is the same as dilation of the background for a binary image, and vice versa. Can you demonstrate that using set theory?

5.13 Repeat the examples of Figs. 5.20 and 5.22 using structuring elements with shapes: 1 x 3, 3 x 1, 5 x 3, 3 x 5.