CHAPTER

3 Basic Image Filtering Operations

Image filtering involves the application of window operations that achieve useful effects, such as noise removal or image enhancement. This chapter is concerned particularly with what can be achieved with quite basic approaches, such as application of local mean, median, or mode filters to digital images. The focus is on grayscale images, although some aspects of color processing are also covered.

Look out for:

- what can be achieved by low-pass filtering in the spatial frequency domain.
- how the same process can be carried out by convolution in the spatial domain.
- the problem of impulse noise and what can be achieved with a limiting filter.
- the value of median, mode, and rank order filters.
- how computational load can be reduced.
- the distinction between image enhancement and image restoration.
- the distortions produced by standard filters—mean, Gaussian, median, mode, and rank order filters.

This chapter takes pain to delve into the properties of a variety of standard types of filter, because it is necessary to know both what they can achieve and what their limitations are. In fact, the edge shifts produced by most of these filters are small but predictable, and therefore correctable in principle. The exception is the rank order filter, for which the shifts can be large—but this is the *advantage* of this type of filter, which is at the core of mathematical morphology (see Chapter 7).

3.1 INTRODUCTION

Chapter 2 is concerned with simple imaging operations, including such problems as thresholding grayscale images and suppressing noise in binary images. In this chapter, the discussion is extended to noise suppression and enhancement in

Computer and Machine Vision.

© 2012 Elsevier Inc. All rights reserved.

grayscale images. Although these types of operation can for the most part be avoided in industrial applications of vision, it is useful to examine them in some depth because of their wide use in a variety of other image processing applications and because they set the scene for much of what follows. In addition, some fundamental issues come to light which are of vital importance.

It has already been seen that noise can arise in real images and it is hence necessary to have sound techniques for suppressing it. Commonly, in electrical engineering applications, noise is removed by means of low-pass or other filters that operate in the frequency domain (Rosie, 1966). Applying these filters to 1-D time-varying analog signals is straightforward, since it is necessary only to place them at suitable stages in the sequence of black boxes through which the signals pass. For digital signals the situation is more complicated, since the frequency transform of the signal must first be computed, then the low-pass filter applied, and finally the signal obtained from the modified transform by converting back to the time domain. Thus, two Fourier transforms have to be computed, although modifying the signal while it is in the frequency domain is a straightforward task (Fig. 3.1). In fact, the amount of processing involved in computing the discrete Fourier transform of a signal represented by N samples is of order N^2 (we shall write this as $O(N^2)$), although the amount of computation can be cut down to $O(N \log_2 N)$ by employing the fast Fourier transform (FFT) (Gonzalez and Woods, 1992). This then becomes a practical approach for the elimination of noise.

When applying these ideas to images we must first note that the signal is a spatial rather than a time-varying quantity and must be filtered in the spatial frequency domain. Mathematically this makes no real difference, but there are nevertheless significant problems. First, there is no satisfactory analog shortcut and the whole process has to be carried out digitally (we here ignore optical processing methods despite their obvious power, speed, and high resolution, because they are by no means trivial to marry with digital computer technology). Second,

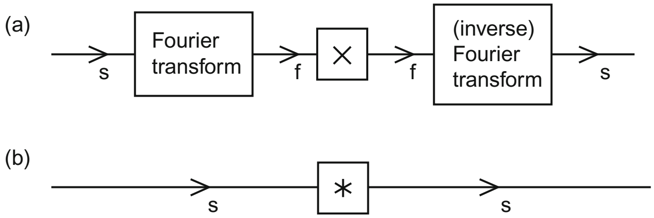

FIGURE 3.1

Low-pass filtering for noise suppression: s, spatial domain; f, spatial frequency domain; ×, multiplication by low-pass characteristic; *, convolution with Fourier transform of low-pass characteristics. (a) Low-pass filtering achieved most simply, by a process of multiplication in the (spatial) frequency domain; (b) low-pass filtering achieved by a process of convolution. Note that (a) may require more computation overall, because of the two Fourier transforms that have to be performed.

for an $N \times N$ pixel image, the number of operations required to compute a Fourier transform is $O(N^3)$ and the FFT only reduces this to $O(N^2 \log_2 N)$, so the amount of computation is quite considerable (here it is assumed that the 2-D transforms are implemented by successive passes of 1-D transforms: see Gonzalez and Woods, 1992). Note also that two Fourier transforms are required for the purpose of noise suppression (Fig. 3.1). Nevertheless, in many imaging applications it is worth proceeding in this way, because not only noise can be removed but also TV scan lines and other artifacts can be filtered out. This situation applies particularly in remote sensing and space technology. However, in industrial applications the emphasis is always on real-time processing, so in many cases it is not practicable to remove noise by spatial frequency domain operations. A further problem is that low-pass filtering is suited to removing Gaussian noise, but distorts the image if it is used to remove impulse noise.

Section 3.2 discusses Gaussian smoothing, in both the spatial frequency and the spatial domains. The subsequent three sections introduce median filters, mode filters, and then general rank order filters, and contrast their main properties and uses. In Section 3.6, consideration is given to reducing computational load, with particular reference to the median filter. Section 3.7 introduces the sharp−unsharp masking technique, which provides a rather simple route to image enhancement. Then follow a number of sections that concentrate on the edge shifts produced by the various filters. In the case of the median filter, the discrete theory (Section 3.9) is much more exact than the continuum model (Section 3.8). All edge shifts are quite small, except for rank order filters (Section 3.12): these are treated fairly fully because of their relevance to widely used morphological operators (Chapter 7) where the shifts are turned to advantage. Finally, Section 3.14 gives a brief discussion on the application of filters to color images.

3.2 NOISE SUPPRESSION BY GAUSSIAN SMOOTHING

Low-pass filtering is normally thought of as the elimination of signal components with high spatial frequencies, and it is therefore natural to carry it out in the spatial frequency domain. Nevertheless, it is possible to implement it directly in the spatial domain. That this is possible is due to the well-known fact (Rosie, 1966) that multiplying a signal by a function in the spatial frequency domain is equivalent to convolving it with the Fourier transform of the function in the spatial domain (Fig. 3.1). If the final convolving function in the spatial domain is sufficiently narrow, then the amount of computation involved will not be excessive: in this way a satisfactory implementation of the low-pass filter can be sought. It now remains to find a suitable convolving function.

If the low-pass filter is to have a sharp cut-off, then its transform in image space will be oscillatory: an extreme example of this is the sinc ($\sin x/x$) function, which is the spatial transform of a low-pass filter of rectangular profile (Rosie, 1966). Oscillatory convolving functions are unsatisfactory since they can introduce halos around objects, hence distorting the image quite grossly. Marr and Hildreth (1980) suggested that the right types of filter to apply to images are those that are well-behaved (nonoscillatory) both in the frequency and in the spatial domain. Gaussian filters are able to fulfill this criterion optimally: they have identical forms in the spatial and spatial frequency domains. In 1-D, these forms are:

$$f(x) = \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\left(-\frac{x^2}{2\sigma^2}\right) \tag{3.1}$$

$$F(\omega) = \exp\left(-\frac{1}{2}\sigma^2\omega^2\right) \tag{3.2}$$

Thus, the type of spatial convolving operator required for the purpose of noise suppression by low-pass filtering is one that approximates to a Gaussian profile. Many such approximations appear in the literature: these vary with the size of the neighborhood chosen and in the precise values of the convolution mask coefficients.

One of the most common is the following mask, first introduced in Chapter 2, which is used more for simplicity of computation than for its fidelity to a Gaussian profile:

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Another commonly used mask, which approximates more closely to a Gaussian profile, is the following:

$$\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

In both cases, the coefficients that precede the mask are used to weight all the mask coefficients: as mentioned in Section 2.3, these weights are chosen so that applying the convolution to an image does not affect the average image intensity. These two convolution masks probably account for over 80% of all discrete approximations to a Gaussian. Note that as they operate within a 3×3 neighborhood, they are reasonably narrow and hence incur a relatively small computational load.

Let us next study the properties of this type of operator, deferring for now consideration of Gaussian operators in larger neighborhoods. First, imagine such an operator is applied to a noisy image whose intensity is inherently uniform. Then clearly noise is suppressed, as it is now averaged over nine pixels. This averaging model is obvious for the first of the two masks above but in fact applies equally to the second mask, once it is accepted that the averaging effect is differently distributed in accordance with the improved approximation to a Gaussian profile.

Although this example shows that noise is suppressed, it is clear that the signal will also be affected. This problem arises only where the signal is initially nonuniform: indeed, if the image intensity is constant, or if the intensity map approximates to a plane, there is again no problem. However, if the signal is uniform over one part of a neighborhood and rises in another part of it, as is bound to occur adjacent to the edge of an object, then the object will make itself felt at the center of the neighborhood in the filtered image (see Fig. 3.2). As a result, the edges of objects become somewhat blurred. Looking at the operator as a "mixing operator" that forms a new picture by mixing together the intensities of pixels fairly close to each other, it is intuitively obvious why blurring occurs.

It is also apparent from a spatial frequency viewpoint why blurring should occur. Basically, we are aiming to give the signal a sharp cut-off in the spatial frequency domain, and as a result it will become slightly blurred in the spatial domain. Clearly, the blurring effect can be reduced by using the narrowest possible approximation to a Gaussian convolution filter, but at the same time the noise suppression properties of the filter are lessened. Assuming that the image was initially digitized at roughly the correct spatial resolution, it will not normally be appropriate to smooth it using convolution masks larger than 3×3 or at most 5×5 pixels (here we ignore methods of analyzing images that use a number of versions of the image with different spatial resolutions: see for example, Babaud et al., 1986).

Overall, low-pass filtering and Gaussian smoothing are not appropriate for the applications considered here because of the blurring effects they introduce. Note also that where interference occurs, it can give rise to impulse or "spike" noise (corresponding to a number of individual pixels having totally the wrong intensities): merely averaging this noise over a larger neighborhood can make the situation worse, since the spikes will be smeared over a sizeable number of pixels and will distort the intensity values of all of them. This consideration is important as it leads naturally to the concepts of limit and median filtering.

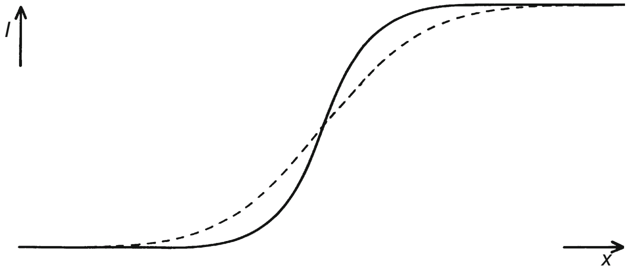

FIGURE 3.2

Blurring of object edges by simple Gaussian convolutions. The simple Gaussian convolution can be regarded as a grayscale neighborhood "mixing" operator, hence explaining why blurring arises.

3.3 MEDIAN FILTERS

The idea explored here is to locate the pixels in the image that have extreme and therefore highly improbable intensities and to ignore their actual intensities, replacing them with more suitable values. This is akin to drawing a graph through a set of plots and ignoring the plots that are evidently a long way from the best fit curve. An obvious way of achieving this is to apply a "limit" filter that prevents any pixel having an intensity outside the intensity range of its neighbors:

```
for all pixels in image do {
  minP = min (P1, P2, P3, P4, P5, P6, P7, P8);
  maxP = max(P1, P2, P3, P4, P5, P6, P7, P8);
  if (P0 < minP) Q0 = minP;                          (3.3)
  else if (P0 > maxP) Q0 = maxP;
  else Q0 = P0;
}
```

To develop this technique, it is necessary to examine the local intensity distribution within a particular neighborhood. Points at the extremes of the distribution are quite likely to have arisen from impulse noise. So it is sensible not only to eliminate these points, as in the limit filter, but also to try taking the process further, removing equal areas at either end of the distribution and ending with the median. Thus, we arrive at the median filter, which takes all the local-intensity distributions and generates a new image corresponding to the set of median values. As the preceding argument indicates, the median filter is excellent at impulse noise suppression and this is amply confirmed in practice (see Fig. 3.3).

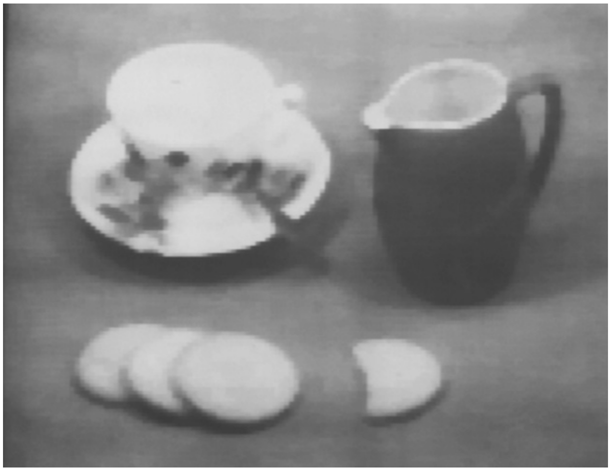

FIGURE 3.3

Effect of applying a 3 × 3 median filter to the image of Fig. 2.1(a). Note the slight loss of fine detail and the rather "softened" appearance of the whole image.

In view of the blurring caused by Gaussian smoothing operators, it is pertinent to ask whether the median filter also induces blurring. In fact, Fig. 3.3 shows that any blurring is only marginal, although there is some slight loss of fine detail that can give the resulting pictures a "softened" appearance. Theoretical discussion of this point is deferred for now; the lack of blur makes good the main deficiency of the Gaussian smoothing filter and results in the median filter being perhaps the most widely used filter in general image processing applications.

There are many ways of implementing the median filter: Table 3.1 reproduces only an obvious algorithm that essentially implements the above description. The notation of Chapter 2 is used but is augmented in order to permit the nine pixels in a 3 × 3 neighborhood to be accessed in turn with a running suffix (specifically, P0 to P8 are written as P[*m*] where *m* runs from 0 to 8).

The operation of the algorithm is as follows: first, the histogram array is cleared and the image is scanned, generating a new image in Q-space; then, for each neighborhood, the histogram of intensity values is constructed; then the median is found; and, finally, points in the histogram array that have been incremented are cleared. This last feature eliminates the need to clear the whole histogram and hence saves computation. Unlike the general situation in which the median of a distribution is being located, only one (half) scan through the distribution is required, since the total area is known in advance (in this case it is 9).

As is clear from the above discussion, methods of computing the median involve pixel intensity sorting operations. If a bubble sort (Gonnet, 1984) were used for this purpose, then up to $O(n^4)$ operations would be required for an $n \times n$ neighborhood, compared with some 256 operations for the histogram method described above. Thus, sorting methods such as the bubble sort are faster for small neighborhoods where n is 3 or 4 but not for neighborhoods where n is greater than 5, or where pixel intensity values are more restricted.

Much of the discussion of the median filter in the literature is concerned with saving computation (Narendra, 1978; Huang et al., 1979; Danielsson, 1981). In

Table 3.1 An Implementation of the Median Filter

```
for (i = 0; i <= 255; i++) hist[i] = 0;
for all pixels in image do {
  for (m = 0; m <= 8; m++) hist[P[m]]++;
  i = 0; sum = 0;
  while (sum < 5) {
    sum = sum + hist[i];
    i = i + 1;
  }
  Q0 = i - 1;
  for (m = 0; m <= 8; m++) hist[P[m]] = 0;
}
```

particular, it has been noticed that, on proceeding from one neighborhood to the next, relatively few new pixels are encountered: this means that the new median value can be found by updating the old value rather than starting from scratch (Huang et al., 1979).

3.4 MODE FILTERS

Having considered the mean and the median of the local intensity distribution as candidate intensity values for noise smoothing filters, it also seems relevant to consider the mode of the distribution. Indeed, we might imagine that this is if anything more important than the mean or the median, since the mode represents the most probable value of any distribution.

However, a tedious problem arises as soon as we attempt to apply this idea. The local intensity distribution is calculated from relatively few pixel intensity values (Fig. 3.4). This means that instead of a smooth intensity distribution whose mode is easily located, we are almost certain to have a multimodal distribution whose highest point does not indicate the position of the *underlying* mode. Clearly the distribution needs to be smoothed out considerably before the mode is computed. Another tedious problem is that the width of the distribution varies widely from neighborhood to neighborhood (e.g., from close to zero to close to 256), so that it is difficult to know quite how much to smooth the distribution in any instance. For these reasons, it is likely to be better to choose an indirect measure of the position of the mode rather than to attempt to measure it directly.

In fact, the position of the mode can be estimated with reasonable accuracy once the median has been located (Davies, 1984a, 1988c). To understand the technique, it is necessary to consider how local intensity distributions of various sorts arise in practical situations. At most positions in an image, variations in pixel intensity are generated by steady changes in background illumination, or by steady variations in surface orientation, or else by noise. Thus, a symmetrical unimodal local intensity distribution is to be expected. It is well known that the mean, median, and mode are coincident in such cases. More problematic is what

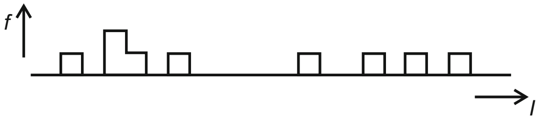

FIGURE 3.4

The sparse nature of the local intensity histogram for a small neighborhood. This situation clearly causes significant problems for estimation of the mode. It also has definite implications for rigorous estimation of the underlying median, assuming that the observed intensities are only noisy samples of the ideal intensity pattern (see Section 3.8.3).

Source: © *IEEE 1984*

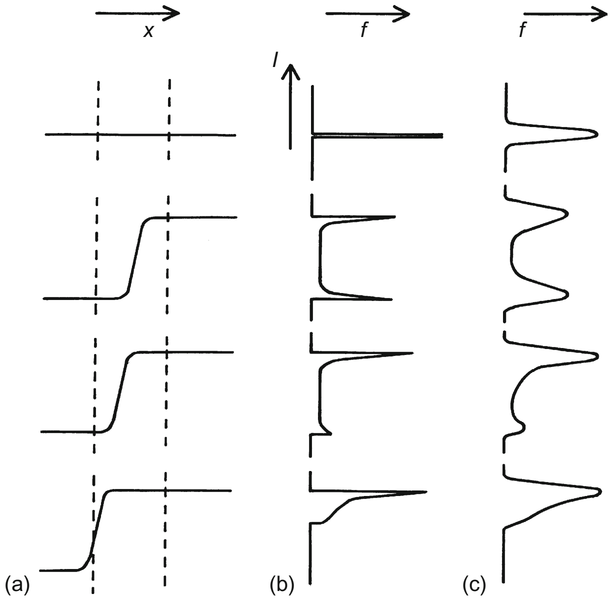

FIGURE 3.5

Local models of image data near the edge of an object: (a) cross-sections of an edge falling in the vicinity of a filter neighborhood; (b) corresponding local intensity distributions when very little image noise is present; (c) situation when the noise level is increased.

Source: © *IEEE 1984*

happens to the intensity variation near the edge of an object in the image. Here the local intensity distribution is unlikely to be symmetrical and, more important, it may not even be unimodal. In fact, near an edge the distribution is in general inherently *bimodal*, since the neighborhood contains pixels with intensities corresponding to the values they would have on either side of the edge (Fig. 3.5). Considering the image as a whole, this will be the most likely alternative to a symmetrical unimodal distribution, any further possibilities such as trimodal distributions being rare and of varied causes (e.g., odd glints on the edges of metal objects) which are outside the scope of the present discussion.[1]

If the neighborhood straddles an edge and the local intensity distribution is bimodal, the larger peak position should clearly be selected as the most probable intensity value. A good strategy for finding the larger peak is to eliminate the smaller peak. If we knew the position of the mode, we could find where to truncate the smaller peak by first finding which extreme of the distribution was closer to the mode, and then moving an equal distance to the opposite side of the mode

[1] Here we are ignoring the effects of noise and just considering the underlying image signal.

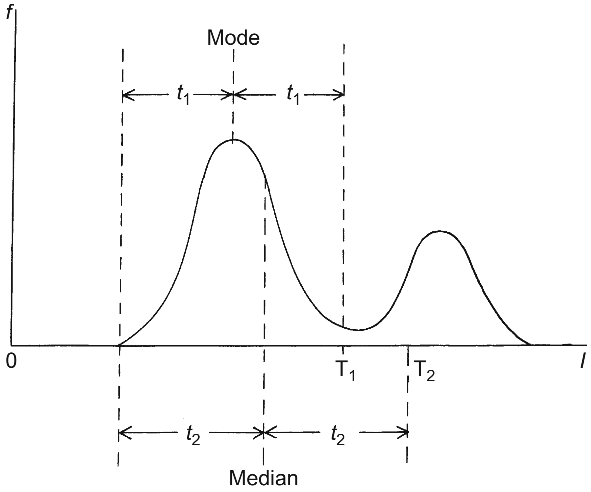

FIGURE 3.6

Rationale for the method of truncation. The obvious position at which to truncate the distribution is T_1. Since the position of the mode is not initially known, it is suboptimal but safe to truncate instead at T_2.

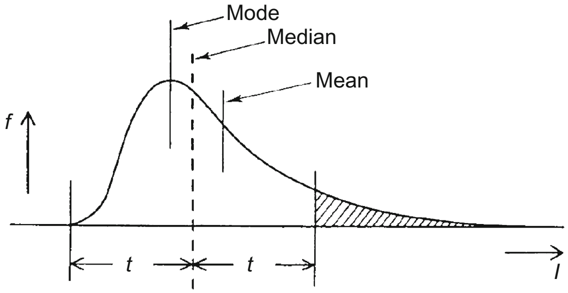

FIGURE 3.7

Relative positions of the mode, median, and mean for a typical unimodal distribution. This ordering is unchanged for a bimodal distribution, as long as it can be approximated by two Gaussian distributions of similar width.

Source: © *IEEE 1984*

(Fig. 3.6). Since we start off *not* knowing the position of the mode, one option is to use the position of the median as an estimator of the position of the mode, and then to use that position to find where to truncate the distribution. Since it invariably happens that the three means take the order mean, median, and mode (see Fig. 3.7), except when distributions are badly behaved or multimodal, this method

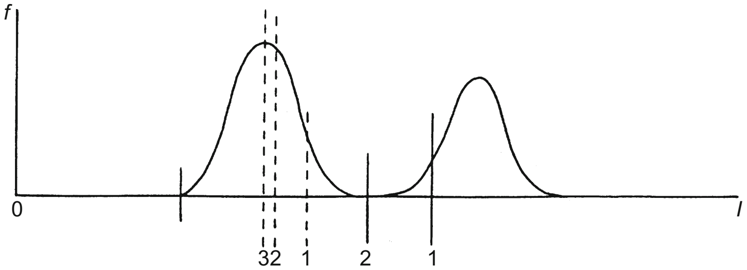

FIGURE 3.8

Iterative truncation of the local intensity distribution. Here the median converges on the mode within three iterations of the truncation procedure. This is possible since at each stage the mode of the new truncated distribution remains the same as that of the previous distribution.

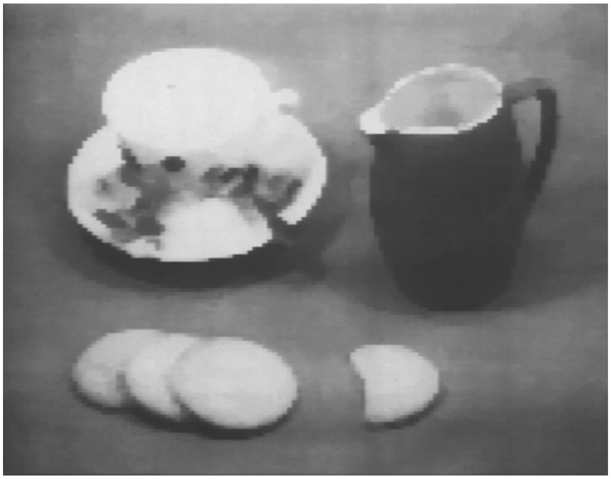

FIGURE 3.9

Effect of a single application of 3 × 3 truncated median filter to the image of Fig. 2.1(a).

is cautious in the sense that it truncates less of the distribution than the required amount: this makes it a safe method to use. When we now find the median of the truncated distribution, the position is much closer to the mode than the original median was, a good proportion of the second peak being removed (Fig. 3.8). Iteration could be used to find an even closer approximation to the position of the mode. However, the method gives a marked enhancement in the image even when this is not done (Fig. 3.9).

We next examine more closely the properties of the "truncated median filter" (TMF) described above. The median filter is highly successful at removing noise,

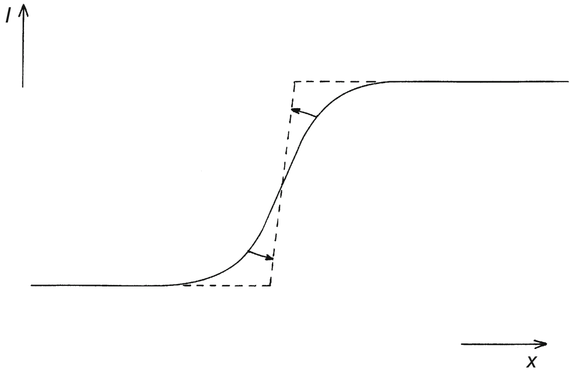

FIGURE 3.10

Image enhancement performed by the mode filter. Here the onset of the edge is pushed laterally by the action of the mode filter within one neighborhood; since the same happens from the other side within an adjacent neighborhood, the actual position of the edge is unchanged in first order. The overall effect is to sharpen the edge.

whereas the TMF not only removes noise but also enhances the image so that edges become sharper. Figure 3.10 makes it clear why this should happen. Basically, at a location even very slightly to one side of an edge, a majority of the pixel intensities contribute to the larger peak and the TMF ignores the pixel intensities contributing to the smaller peak. Thus, the TMF makes an informed binary choice about which side of the edge it is on. At first this seems to mean that it pushes a nearby edge further away. However, it must be remembered that it actually “pushes the edge away” from both sides, and the result is that its sides are made sharper and object outlines are crispened up. Particularly striking is the effect of applying the TMF to an image a number of times, when objects start to become segmented into regions of fairly uniform intensity (Fig. 3.11). The complete algorithm for achieving this is outlined in Table 3.2.

This problem has been dealt with at some length for a number of reasons. First, the mode filter has not hitherto received the attention it deserves. Second, the median filter seems to be used fairly universally, often without very much justification or thought. Third, all these filters show what markedly different characteristics are available merely by analyzing the contents of the local intensity distribution and ignoring totally where in the neighborhood the different intensities appear: it is perhaps remarkable that there is sufficient information in the local intensity distribution for this to be possible. All this shows the danger of applying operators that have been derived in an *ad hoc* manner without first making a specification of what is required and then designing an operator with the required characteristics. In fact, the situation appears to be that if we want a filter that has maximum impulse noise suppression capability, then we should use a median filter; and if we want a filter that enhances images by sharpening edges,

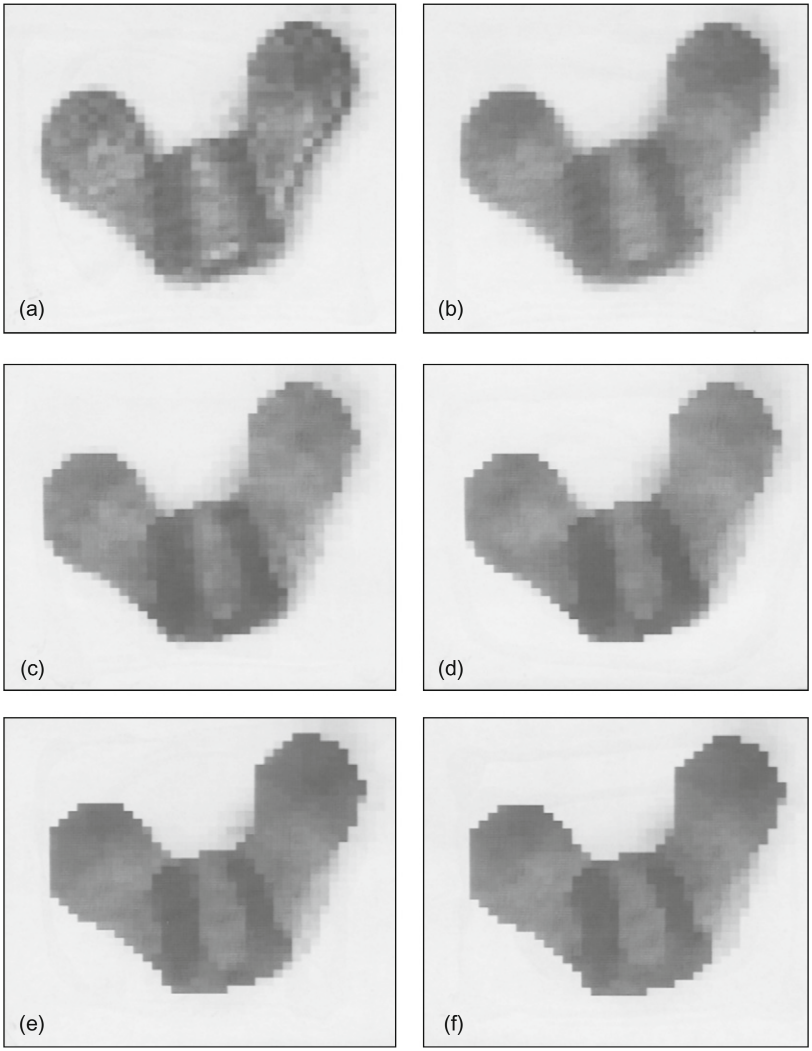

FIGURE 3.11

Results of repeated action of the truncated median filter: (a) the original, moderately noisy picture; (b) effect of a 3 × 3 median filter; (c)–(f) effect of 1–4 passes of the basic truncated median filter, respectively.

Source: © *IEEE 1984*

then we should use a mode filter or TMF (note that the TMF should be an improvement on the mode filter in that it is more cautious very close to an edge transition, where noise prevents an exact judgement being made as to which side of the edge a pixel is on: see Davies (1984a, 1988c)).

Table 3.2 Outline of Algorithm for Implementing the Truncated Median Filter

```
do { // as many passes over image as necessary
    for all pixels in image do {
        compute local intensity distribution;
        do { // iterate to improve estimate of mode
            find minimum, median, and maximum intensity values;
            decide from which end local intensity distribution should
              be truncated;
            deduce where local intensity distribution should be
              truncated;
            truncate local intensity distribution;
            find median of truncated local intensity distribution;
        } until median sufficiently close to mode of local distribution;
        transfer estimate of mode to output image space;
    }
} until sufficient enhancement of image;
```

Comments:

(i) *The outermost and innermost loops can normally be omitted (i.e., they need to be executed once only).*

(ii) *The final estimate of the position of the mode can be performed by simple averaging instead of computing the median: this has been found to save computation with negligible loss of accuracy.*

(iii) *Instead of the minimum and maximum intensity values, the positions of the outermost octiles (for example) may be used to give more stable estimates of the extremes of the local intensity distribution.*

While considering enhancement, attention has been restricted to filters based on the local intensity distribution: there are many filters that enhance images without the aid of the local intensity distribution (Lev et al., 1977; Nagao and Matsuyama, 1979), but they are not within the scope of this chapter. Note that the method of "sharp−unsharp masking" (Section 3.7) performs an enhancement function, although its main purpose is to restore images that have inadvertently become blurred, e.g., by a hazy atmosphere or defocussed camera.

Finally, while this section has concentrated on the grayscale properties of mode filters, Charles and Davies (2003a, 2004) have shown how to devise versions of the TMF that operate on color images. Typical results are shown in Fig. 3.12. In addition, Fig. 3.13 shows that the TMF has the useful property of being able to eliminate very large amounts of impulse noise from images—significantly more than a median filter—in spite of being designated as an image enhancement filter.

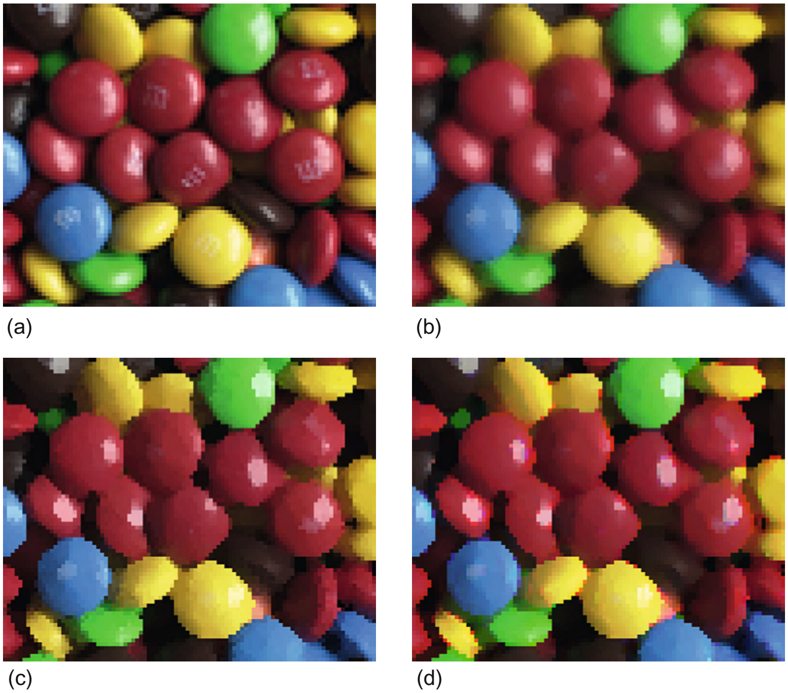

FIGURE 3.12

Color filtering of brightly colored objects. (a) Original color image of some sweets. (b) Vector median filtered version. (c) Vector mode filtered version. (d) Version to which a mode filter has been applied to each color channel separately. Note that (b) and (c) show no evidence of color bleeding, although it is strongly evident in (d). It is most noticeable as isolated pink pixels, plus a few green pixels, around the yellow sweets. For further details on color bleeding, see Section 3.14

Source: © *RPS 2004*

3.5 RANK ORDER FILTERS

The principle employed in rank order filters is to take all the intensity values in a given neighborhood, to place these in order of increasing value, and finally to select the *r*th of the *n* values and return this value as the filter local output value. Clearly, *n* rank order filters can be specified in terms of the value *r* that is used, but these filters are intrinsically nonlinear, i.e., the output intensity cannot be expressed as a linear sum of the component intensities within the neighborhood. In particular, the median filter (for which $r = (n+1)/2$, and which is only defined

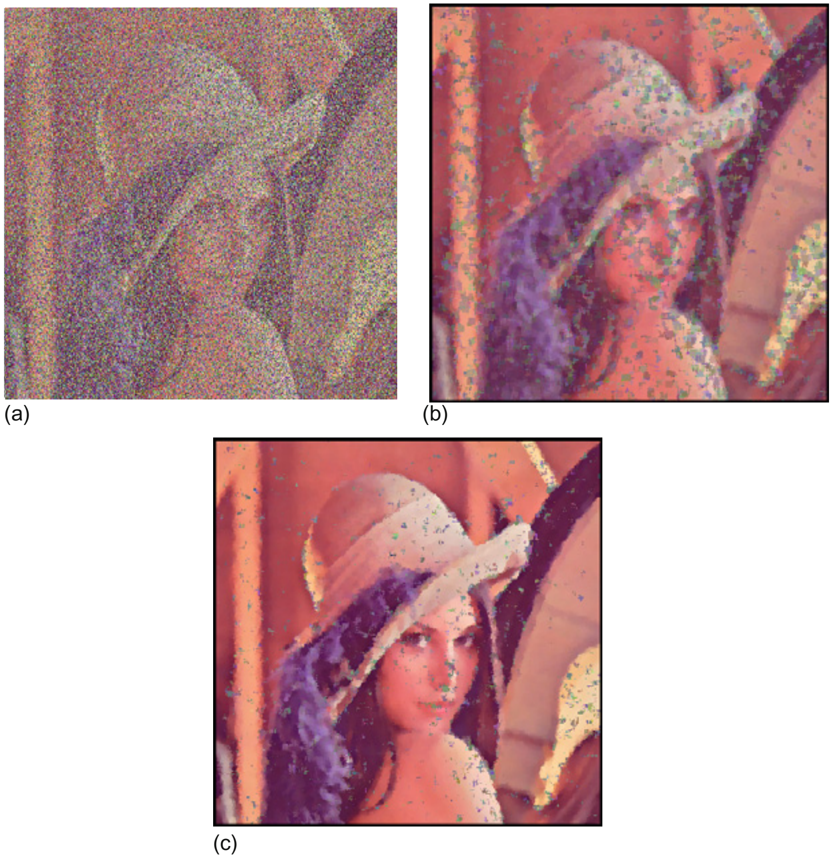

FIGURE 3.13

Color filtering of images containing substantial impulse noise. (a) Version of the Lena image containing 70% random color impulse noise. (b) Effect of applying a vector median filter, and (c) effect of applying a vector mode filter. While the mode filter is designed more for enhancement than for noise suppression, it has been found to perform remarkably well at this task when the noise level is very high

Source: © *RPS 2004*

if n is odd)[2] does not normally give the same output image as a mean filter: indeed, it is well known that the mean and median of a distribution are in general only coincident for symmetrical distributions. Note that minimum and maximum filters (corresponding to $r = 1$ and $r = n$, respectively) are also often classed as morphological filters (see Chapter 7).

[2] If n is even, it is usual to take the mean of the central two values in the distribution as representing the median.

3.6 REDUCING COMPUTATIONAL LOAD

Significant efforts have been made to speedup the operation of the Gaussian filter since implementations in large neighborhoods require considerable amounts of computation (Wiejak et al., 1985). For example, smoothing an image of 256×256 pixels using a 30×30 Gaussian convolution mask involves 64 million basic operations. For such a basic operation as smoothing, this is unacceptable. However, it is possible to cut down the amount of computation drastically, since a 2-D Gaussian convolution can be factorized into two 1-D Gaussian convolutions, which can be applied in turn:

$$\exp\left(-\frac{r^2}{2\sigma^2}\right) = \exp\left(-\frac{x^2}{2\sigma^2}\right)\ \exp\left(-\frac{y^2}{2\sigma^2}\right) \tag{3.4}$$

It is important to realize that the decomposition is rigorously provable and is not an approximation: we shall refer to this below in the context of the median filter. Meanwhile, the decompositions for the two 3×3 Gaussian filters we discussed earlier are:

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} = \frac{1}{3}\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \frac{1}{3}\begin{bmatrix} 1 & 1 & 1 \end{bmatrix} \tag{3.5}$$

and

$$\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} = \frac{1}{4}\begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \frac{1}{4}\begin{bmatrix} 1 & 2 & 1 \end{bmatrix} \tag{3.6}$$

Overall, this approach replaces a single $n \times n$ operator whose load is $O(n^2)$ with two operators of load $O(n)$, and ignoring scanning and other overheads, the saving factor is $n/2$. Hence, for $n > 2$, there will always be a useful saving.

In fact, it is not possible to decompose the median filter in the same way without making approximations. However, it is quite common to try to perform a similar function by applying two 1-D median filters in turn (Narendra, 1978). Although the effect is similar in its outlier rejection properties to the standard 2-D median filter, the following example confirms that the two are not exactly equivalent:

Original image segment:

$$\begin{matrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 2 & 0 \\ 0 & 0 & 2 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{matrix}$$

After applying a 3 × 3 filter:

$$\begin{array}{cccccc}
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0
\end{array}$$

After applying a 3 × 1 and then a 1 × 3 filter:

$$\begin{array}{cccccc}
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 1 & 0 \\
0 & 0 & 2 & 2 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0
\end{array}
\qquad
\begin{array}{cccccc}
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0
\end{array}$$

After applying a 3 × 1 and then a 1 × 3 filter:

$$\begin{array}{cccccc}
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0
\end{array}
\qquad
\begin{array}{cccccc}
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0
\end{array}$$

By chance, the final results in the last two cases are the same, but this is a relatively rare occurrence.

In general, spots and streaks not more than one pixel wide are eliminated quite effectively by the original or by the separated forms of the filter. Larger filters should effectively eliminate wider spots and streaks, although exact functional equivalence between the original and its separated forms is not to be expected, as has been indicated.

Finally, the problem of inexact decomposition is not an exclusive property of nonlinear filters; many linear filters cannot be decomposed exactly either: this is evident because the number of independent coefficients in an $n \times n$ mask is n^2, which is much greater than the total number in an $n \times 1$ and a $1 \times n$ component mask.

3.7 SHARP–UNSHARP MASKING

When images are blurred either before or as part of the process of acquisition, it is possible frequently to restore them substantially to their ideal state. Properly, this is achieved by making a model of the blurring process and applying an inverse transformation that is intended to cancel the blurring. This is a complex task to carry out rigorously, but in some cases a rather simple method called sharp–unsharp masking is able to produce significant improvement (Gonzalez and Woods, 1992). As indicated in Fig. 3.14, this technique involves first obtaining an even more blurred version of the image (e.g., with the aid of a Gaussian filter) and then subtracting this image from the original. Note that the amount of

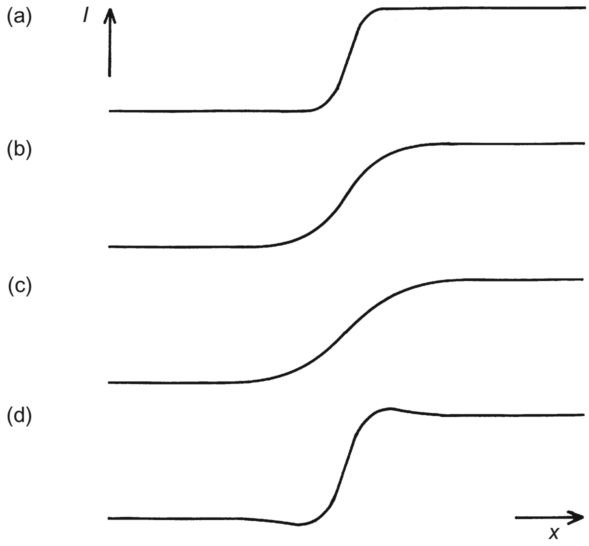

FIGURE 3.14

The principle of sharp–unsharp masking: (a) cross-section of an idealized edge; (b) observed edge; (c) artificially blurred version of (b); (d) result of subtracting a proportion of (c) from (b).

artificial blurring to apply and the proportion of the blurred image to subtract are rather arbitrary quantities that are normally adjusted by eye. Thus, the method is better categorized under the heading "enhancement" than "restoration," as it is not the precise mathematical technique normally understood by the latter term. Of such enhancement techniques, Hall (1979) states: "Much of the art of enhancement is knowing when to stop."

3.8 SHIFTS INTRODUCED BY MEDIAN FILTERS

Despite knowing the main characteristics of the different types of filter there are still some unknown factors. In particular, it is often important (especially when making precision measurements on manufactured components) to ensure that noise is removed in such a way that object locations and sizes are unchanged. However, at this point the following two problems arise.

First, it has been assumed that the intensity profile of an edge is symmetrical. If this is so, then the mean, median, and mode of the local intensity distribution will be coincident and there will clearly be no overall bias for any of them. However, when the edge profile is asymmetrical, it will not be obvious in the absence of a detailed model of the situation what the result will be for any of the

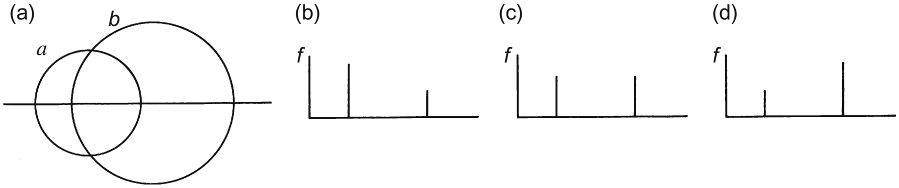

FIGURE 3.15

Variation in local intensity distribution with position of neighborhood: (a) neighborhood of radius *a* overlapping a dark circular object of radius *b*; (b)–(d) intensity distributions *I* when the separations of the centers are respectively less than, equal to, or greater than the distance *d* for which the object bisects the area of the neighborhood.

filters. The situation is even more involved when significant noise is present (Yang and Huang, 1981; Bovik et al., 1987). Since the problem is so data-dependent, it is not profitable to consider it further here.

The second problem concerns the situation for a curved edge. In this case, there is again a variety of possibilities, and filters employing the different means will modify the edge position in ways that depend markedly on its shape. In robot vision applications, the median filter is the one we are most likely to use because its main purpose is to suppress noise without introducing blurring. Hence, the bias produced by this type of filter is worth considering in some detail. This is done in the next subsection.

3.8.1 Continuum Model of Median Shifts

This section takes the case of a continuous image (i.e., a nondiscrete lattice), assuming (i) that the image is binary, (ii) that neighborhoods are exactly circular, and (iii) that images are noise-free. To proceed we notice that binary edges have symmetrical cross-sections, while straight edges extend this symmetry into 2-D: hence applying a median filter in a (symmetrical) circular neighborhood cannot pull a straight edge to one side or the other.

Now consider what happens when the filter is applied to an edge that is not straight. If, for example, the edge is circular, the local intensity distribution will contain two peaks whose relative sizes will vary with the precise position of the neighborhood (Fig. 3.15). At some position the sizes of the two peaks will be identical. Clearly, this happens when the center of the neighborhood is situated at a point where the output of the median filter changes from dark to light (or *vice versa*). Thus, the median filter produces an inward shift toward the center of a circular object (or the center of curvature), whether the object is dark on a light background or light on a dark background.

To calculate the magnitude of this effect, we need to find at what distance *d* from the center of a circular object (of radius *b*) the area of a circular neighborhood (of radius *a*) is bisected by the object boundary.

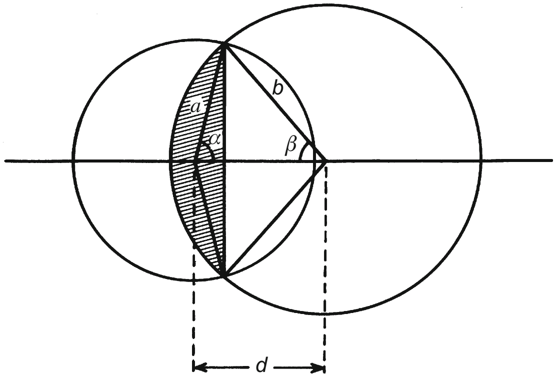

FIGURE 3.16

Geometry for calculating neighborhood and object overlap.

From Fig. 3.16 the area of the sector of angle 2β is βb^2, while the area of the triangle of angle 2β is $b^2\sin\beta\cos\beta$. Hence, the area of the segment shown shaded is:

$$B = b^2(\beta - \sin\beta\cos\beta) \tag{3.7}$$

Making a similar calculation of the area A of a circular segment of radius a and angle 2α, the area of overlap (Fig. 3.16) between the circular neighborhood of radius a and the circular object of radius b may be deduced as:

$$C = A + B \tag{3.8}$$

For a median filter this is equal to $\pi a^2/2$. Hence:

$$F = a^2(\alpha - \sin\alpha\ \cos\alpha) + b^2(\beta - \sin\beta\ \cos\beta) - \frac{\pi a^2}{2} = 0 \tag{3.9}$$

where

$$a^2 = b^2 + d^2 - 2bd\ \cos\beta \tag{3.10}$$

and

$$b^2 = a^2 + d^2 - 2ad\cos\alpha \tag{3.11}$$

To solve this set of equations, we take a given value of d, deduce values of α and β, calculate the value of F, and then adjust the value of d until $F = 0$. Since d is the modified value of b obtained after filtering, the shift produced by the filtering process is:

$$D = b - d \tag{3.12}$$

The results of doing this computation numerically have been found by Davies (1989b). As expected, $D \to 0$ as $b \to \infty$ or $a \to 0$. Conversely, the shift becomes very large as a first approaches and then exceeds b. Note, however, that when

$a > \sqrt{2}b$ the object is ignored, being small enough to be regarded as irrelevant noise by the filter: beyond this point it has no effect at all on the final image. The maximum edge shift before the object finally disappears is $(2 - \sqrt{2})b \approx 0.586b$.

It is instructive to approximate the above equations for the case when edge curvature is small, i.e., $a \ll b$. Under these conditions, β is small, $\alpha \approx \pi/2$ and $d \approx b$. Hence, we find:

$$\beta \approx \frac{a}{b} \tag{3.13}$$

After some manipulation the edge shift D is obtained in the form:

$$D \approx \frac{a^2}{6b} = \frac{\kappa a^2}{6} \tag{3.14}$$

$\kappa = 1/b$ being the local curvature. In Chapter 6, this equation is found to be useful for estimating the signals from a median-based corner detector.

3.8.2 Generalization to Grayscale Images

To extend these results to grayscale images, first consider the effect of applying a median filter near a smooth step edge in l-D. Here the median filter gives zero shift, since for equal distances from the center to either end of the neighborhood there are equal numbers of higher and lower intensity values and hence equal areas under the corresponding portions of the intensity histogram. Clearly this is always valid where the intensity increases monotonically from one end of the neighborhood to the other—a property first pointed out by Gallagher and Wise (1981) [for more recent discussions on related “root” (invariance) properties of signals under median filtering, see Fitch et al. (1985) and Heinonen and Neuvo (1987)].

Next, it is clear that for 2-D images, the situation is again unchanged in the vicinity of a straight edge, since the situation remains highly symmetrical. Hence the median filter gives zero shift, as in the binary case.

For curved boundaries, the situation has to be considered carefully for grayscale edges, which, unlike binary edges, have finite slope. When boundaries are roughly circular, contours of constant intensity often appear as in Fig. 3.17. To find how a median filter acts, we merely need to identify the contour of median intensity (in 2-D the median intensity value labels a whole contour) that divides the area of the neighborhood into two equal parts. The geometry of the situation is identical to that already examined in Section 3.8.1: the main difference here is that for every position of the neighborhood, there is a corresponding median contour with its own particular value of shift depending on the curvature. Intriguingly, the formulae already deduced may immediately be applied for calculating the shift for each contour. Figure 3.17 shows an idealized case in which the contours of constant intensity have similar curvature, so that they are all moved

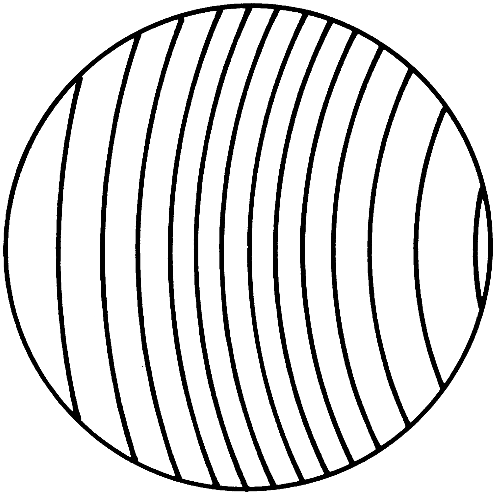

FIGURE 3.17

Contours of constant intensity on the edge of a large circular object, as seen within a small circular neighborhood.

inward by similar amounts. This means that, to a first approximation, the edges of the object retain their cross-sectional profile as it becomes smaller.

For grayscale images, the shifts predicted by this theory (with certain additional corrections: see Davies (1989b)) agree with experimental shifts within approximately 10% for a large range of circle sizes in a discrete lattice (see Fig. 3.18). Paradoxically, the agreement is less perfect for binary images, since circles of certain sizes show stability effects (akin to median root behavior): these effects tend to average out for grayscale images, owing to the presence of many contours of different sizes at different gray levels. Overall, the edge shifts obtained with median filters are now quite well understood. Figures 3.19 and 3.20 give some indication of the magnitudes of these shifts in practical situations. Note that once image detail such as a small hole or screw thread has been eliminated by a filter, it is not possible to apply any edge shift correction formula to recover it, although for larger features such formulae are useful for deducing true edge positions.

3.8.3 Problems with Statistics

Thus far it has been seen that computations of the position of the mode are made more difficult because of the sparse statistics of the local intensity distribution. In fact this also affects the median calculation. Suppose the median value happens to be well spaced near the center of the distribution (Fig. 3.4). Then a small error in

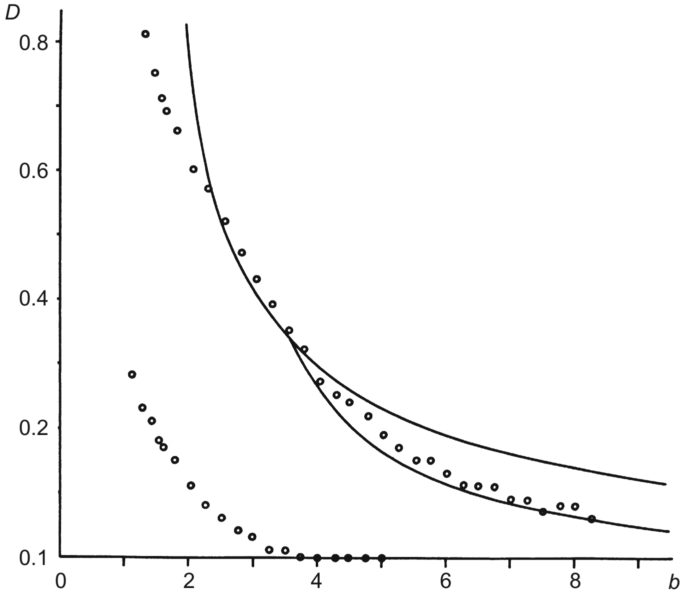

FIGURE 3.18

Edge shifts for 5 × 5 median filter applied to a grayscale image. The upper set of plots represents the experimental results and the upper continuous curve is derived from the theory of Section 3.8.1. The lower continuous curve is derived from a more accurate model (Davies, 1989b). The lower set of plots represents the much reduced shifts obtained with the "detail-preserving" type of filter (see Section 3.16).

this one intensity value is immediately reflected in full when calculating the median: i.e., the poor statistics have biased the median in a particular way. Ideally, what is required is a stable estimator of the median of the underlying distribution. Thus, the distribution should be made smoother before arriving at a specific value for the median. In practice, this procedure adds significant computational load to the filter calculation and is commonly not carried out. As a consequence the median filter tends to result in runs of constant intensity, thereby giving images the "softened" appearance noted earlier. This is apparent on studying the following 1-D example:

Original:	0	0	1	0	0	1	1	2	1	2	2	1	2	3	3	4	3	2	2	3
Filtered:	?	0	0	0	0	1	1	1	2	2	2	2	2	3	3	3	3	2	2	?

Although histogram smoothing is not commonly carried out, some workers have felt it necessary to adjust the relative weights of the various pixels in the neighborhood according to their distance from the central pixel (Akey and Mitchell, 1984). This mimics what happens for a Gaussian filter and is theoretically

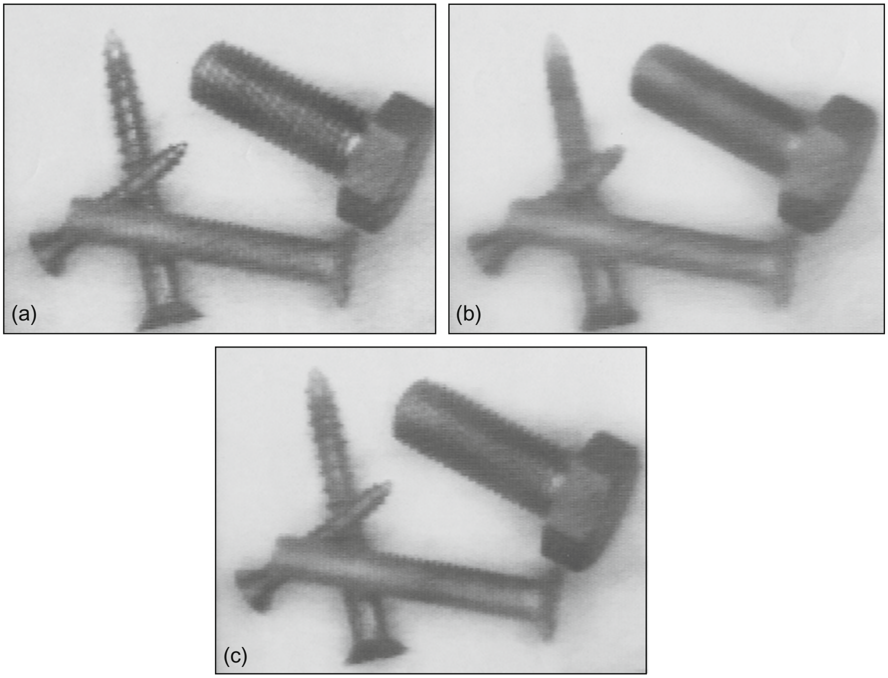

FIGURE 3.19

Edge smoothing property of the median filter: (a) Original image; (b) median filter smoothing of irregularities, in particular those around the boundaries (note how the threads on the screws are virtually eliminated although detail larger in scale than half the filter area is preserved), using a 21-element filter operating within a 5 × 5 neighborhood on a 128 × 128 pixel image of 6-bit gray scale; (c) effect of the detail-preserving filter (see Section 3.16).

necessary, although it is not generally implemented. However, there have recently been further developments on this front (e.g., see Charles and Davies, 2003b).

3.9 DISCRETE MODEL OF MEDIAN SHIFTS

To produce a discrete model of median shifts we need to recognize explicitly the positions of the pixels within the chosen neighborhood. We approximate by assuming that the intensity of any pixel is the mean intensity over the whole pixel and is represented by a sample positioned at the center of the pixel. We start by examining the case of a 3×3 neighborhood, and proceed by taking the underlying analog intensity variation to have contours of curvature κ, as shown in

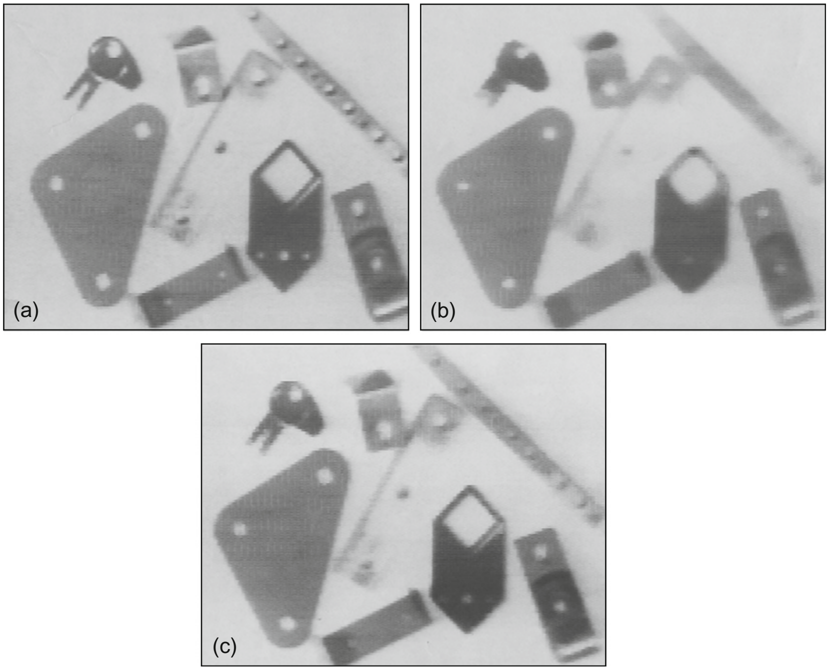

FIGURE 3.20

Circular holes in metal objects before and after filtering: (a) original 128 × 128 pixel image with 6-bit gray scale; (b) 5 × 5 median-filtered image: the diminution in size of the hole is clearly visible and such distortions would have to be corrected for when taking measurement from real filtered images of this type; (c) result using a detail-preserving filter: some distortions are present although the overall result is much better than in (b).

Fig. 3.17. Following what happened in the continuum case, it will not matter whether the contours of constant intensity are those of a step edge or those of a slowly varying slant edge: it is what happens at the median contour that determines the shift that arises.

The starting point is that zero shift occurs for $\kappa = 0$. Next, if κ is even minutely greater than zero, the center pixel will not necessarily be the median pixel. Consider the case when the circular median intensity contour passes close to the center of the neighborhood at a small angle θ to the positive x-axis (Figs. 3.21 and 3.22(a)). In that case, the filter will produce a definite shift, whose value is:

$$D_\theta \approx \frac{1}{2}\kappa a_0^2 - a_0\theta = \frac{1}{2}\kappa - \theta \qquad (3.15)$$

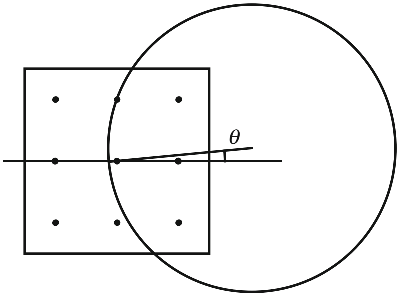

FIGURE 3.21

Geometry for calculation of median shifts on the discrete model.

Source: © *IEE 1999*

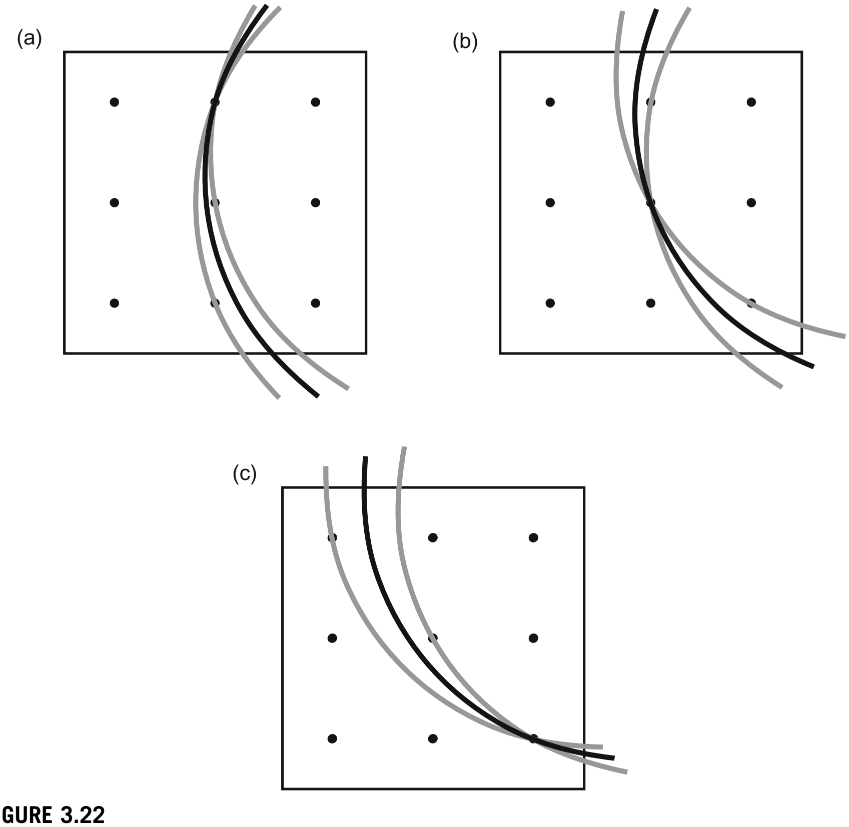

FIGURE 3.22

Geometry for calculation of median shifts at low κ. These three diagrams show the positions of the median pixels and the ranges of orientations of circular intensity contours for which they apply, (a) for low θ, (b) for intermediate θ, and (c) for high θ.

Source: © *IEE 1999*

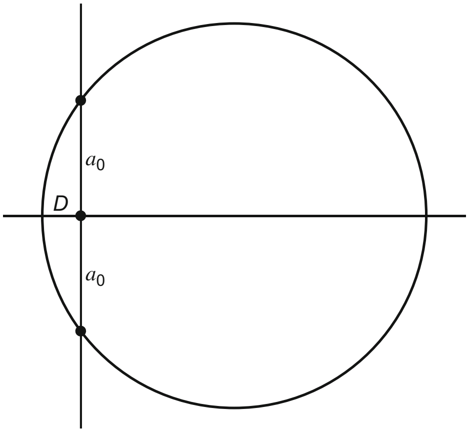

FIGURE 3.23

Geometry for calculation of shift when the median contour passes through the centers of two pixels.

where a_0 is the inter-pixel separation—here taken as unity. The first term arises from the following simple result for the geometry of the circle of radius $b = 1/\kappa$ in Fig. 3.23:

$$a_0^2 = D \times (2b - D) \approx 2Db \tag{3.16}$$

It is too tedious to recount here the complete analysis for the variation in D_θ for all θ. Suffice it to say that when it has been carried out, it is necessary to average it over all θ for each value of κ. When this is done (Davies, 1999f) the agreement between theory and experiment is essentially exact over a wide range of values of κ, as shown in Fig. 3.24: the reason for the discrepancy of high values of κ is due to the limited intensity gradients that occur at edges in grayscale images.

Overall, the problem of median shifts is now well understood, and is fully explained using the discrete model. The continuum model turns out to be capable of giving accurate results only in the limiting case where a and b $(= 1/\kappa)$ are many pixels in size.

3.10 SHIFTS INTRODUCED BY MODE FILTERS

In this section we consider the shifts produced by mode filters in continuous images. As in the cases of median filters, straight edges with symmetrical profiles cannot be shifted by mode filters, because of symmetry. We proceed to the two paradigm cases—step edges and slant edges with circular boundaries. Again, the effects of noise will be ignored as we are considering the intrinsic rather than the noise-induced behavior of the mode filtering operation.

The situation for a curved step edge can again be understood by appealing to Fig. 3.15. The result for the mode also has to be identical to that for the median,

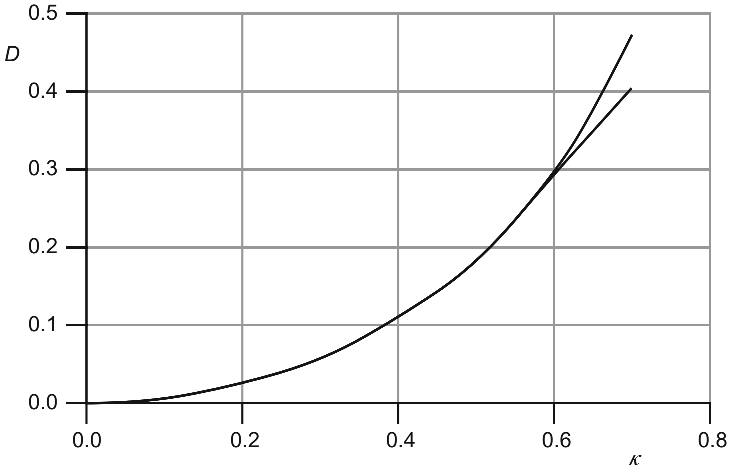

FIGURE 3.24

Comparisons of 3 × 3 median shifts. The lower solid curve shows the nonapproximated results of the discrete model (Davies, 1999f): the upper solid curve shows the results of experiments on grayscale circles.

Source: © IEE 1999

because the local intensity distribution is exactly symmetric and bimodal at the point where the median filter switches from a left-hand to a right-hand decision: at that point the mode must give the same result, since the median and the mode are coincident for a symmetric distribution. Hence, we conclude that the mode also gives a shift of $1/6\,\kappa a^2$ for a curved step edge.

Next, we calculate edge shifts in the case where smoothly varying intensity functions exist—or within the confines of a small neighborhood—*appear* to be smoothly varying. In this case the calculation is especially simple (Davies, 1997a). Using the geometry of Fig. 3.17, we consider the intensity pattern within a circular neighborhood C. Of all the circular intensity contours appearing within C, the one possessing the most frequently occurring intensity, as selected by a mode filter, is the longest. Clearly, this is the one (M) whose ends are at opposite ends of a diameter of C. To estimate the shift in this case, all we need to do is to calculate the position of M, and determine its distance from the center of C. To proceed, we use the well-known formula relating the lengths of parts of intersecting chords of a circle, which gives:[3]

$$a^2 = D(2b - D) \approx 2Db \tag{3.17}$$

[3] This equation is a more general form of equation (3.16), as a can have any value; the proof follows similarly, on replacing a_0 by a in Fig. 3.23.

Table 3.3 Summary of Edge Shifts for Neighborhood Averaging Filters

Edge Type	Filter		
	Mean	Median	Mode
Step	$\frac{1}{6}\kappa a^2$	$\frac{1}{6}\kappa a^2$	$\frac{1}{6}\kappa a^2$
Intermediate	$\sim\frac{1}{7}\kappa a^2$	$\frac{1}{6}\kappa a^2$	$\frac{1}{2}\kappa a^2$
Linear	$\frac{1}{8}\kappa a^2$	$\frac{1}{6}\kappa a^2$	$\frac{1}{2}\kappa a^2$

© IEE 1999.

Hence:

$$D = \frac{1}{2}\kappa a^2 \tag{3.18}$$

i.e., there is a *right* shift of the contour, toward the local center of curvature, of $1/2\ \kappa a^2$. If we regard this set of contours as forming part of a grayscale edge profile, then the mode filter shifts the edge through $1/2\ \kappa a^2$ toward the center of curvature.

Some comments on the marked difference between the cases of step edges and linear intensity profiles are called for. This is all the more interesting as the median filter produces identical shifts, of $1/6\ \kappa a^2$, for the two profiles (see Table 3.3). In fact, of all the cases listed in Table 3.3, the outstanding one is the large shift for a mode filter operating on a linear intensity profile: what is special in this case is that the result relies on a single extreme contour length rather than an average of lengths amounting to an area measure. Hence, it is not surprising that the mode filter gives an exceptionally large shift in this case.

Finally, we note that edge shifts are not avoided merely by choosing an alternative method of neighborhood averaging, but rather that they are intrinsic to the averaging process, and can be avoided only by specially designed operators (e.g., see Greenhill and Davies, 1994a).

3.11 SHIFTS INTRODUCED BY MEAN AND GAUSSIAN FILTERS

In this section, we consider the shifts produced by mean and Gaussian filters in continuous images. As in the cases of median and mode filters, straight edges with symmetrical profiles cannot be shifted by mean and Gaussian filters, because of symmetry. We again consider the two paradigm cases—step edges and slant edges with circular boundaries. Again, we will ignore the effects of noise as we are considering the intrinsic rather than the noise-induced behavior of the filters.

The situation for a curved step edge can again be understood by appealing to Fig. 3.15, and is identical to that for the median and mode filters, and follows because of the symmetry of the local intensity distribution at the point where the filter switches from a left-hand to a right-hand decision: at that point the mean filter must give the same result, since all three statistics coincide for a symmetric distribution. Hence, it also gives a shift of $1/6\ \kappa a^2$ for a curved step edge.

In the case of a smoothly varying slant edge, the result for the mean filter has to be calculated by integrating over the area of the neighborhood. The results cannot be obtained by intuitive or simple geometric or intuitive arguments, and here we merely quote the shift for the mean filter as being $1/8\ \kappa a^2$.

These considerations complete the entries in Table 3.3. The results for Gaussian filters do not differ substantially from those for mean filters, but have to be obtained by integration, taking account of the Gaussian weighting function (Davies, 1991b). A general point is that all such filters have similar shifting effects because they all incorporate a measure of signal averaging: the shifting effect is not avoided simply by employing a different central-seeking statistic to perform the averaging.

3.12 SHIFTS INTRODUCED BY RANK ORDER FILTERS

This section is particularly concerned with rank order filters (Bovik et al., 1983), which form a whole family of filters that can be applied to digital images—often in combination with other filters of the family—in order to give a variety of effects (Goetcherian, 1980; Hodgson et al., 1985): other notable members of the family are max and min filters. Because rank order filters generalize the concept of the median filter, it is relevant to study the types of distortion they produce on straight and curved intensity contours. It should also be pointed out that these filters are of central importance in the design of filters for morphological image analysis and measurement. In addition, it has been pointed out that they have some advantages when used for this purpose in that they help to suppress noise (Harvey and Marshall, 1995) (although the effect vanishes in the special cases of max and min filters).

Section 3.12.1 examines the reasons underlying the shifts produced by rank order filters and makes calculations of their extent for rectangular neighborhoods. It then generalizes these results to circular neighborhoods and goes on to examine the extent to which the theoretical predictions are borne out in practice by measurements of the shifts produced by 5×5 rank order filters on circular disks of varying sizes. It will be taken as axiomatic that the application of rank order filters produces edge shifts on real images (they are well attested in the case of max, min, and median filters): the main question to be answered here is the exact numerical extent of these shifts and how they may be modeled for general rank order filters.

3.12.1 Shifts in Rectangular Neighborhoods

In common with previous work in this area we here concentrate on the ideal noiseless case, in which the filter operates within a small neighborhood, over which the signal is basically a monotonically increasing intensity function in some direction. The most complex intensity variation that will be considered is that in which the intensity contours are curved with curvature κ. In spite of this simplified configuration, valuable statements can still be made about the level of distortion likely to be produced in practice by rank order filters.

Because of the complexity of the calculations that arise in the case of rank order filters, which involve an additional parameter *vis-à-vis* the median filter, it is worth studying their properties first for the simple case of rectangular neighborhoods (Davies, 2000f). Let us presume that a rank order filter is being applied in a situation in which straight intensity contours are aligned parallel to the short sides of a rectangular neighborhood that we initially take to be a $1 \times n$ array of pixels (Fig. 3.25). In this case, we can assume without loss of generality that the successive pixels within the neighborhood will have *increasing* values of intensity. We next take the basic property of the rank order filter (effectively or in fact) to construct an intensity histogram of the local intensity distribution and return the value of the rth of the n intensity values within the neighborhood. This means that the rank order filter selects an intensity that has physical separation B from the lowest intensity pixel of the neighborhood and C from the highest intensity pixel, where:

$$B = r - 1 \tag{3.19}$$

$$C = n - r \tag{3.20}$$

$$A = B + C = n - 1 \tag{3.21}$$

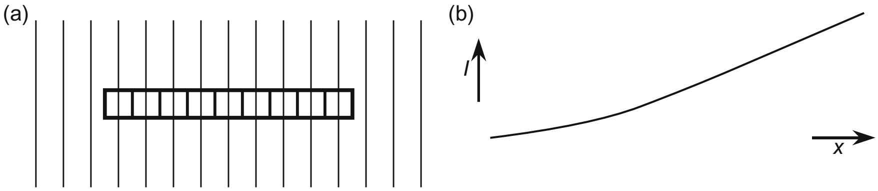

FIGURE 3.25

Basic situation for a rank order filter in a rectangular neighborhood. This figure illustrates the problem of applying a rank order filter within a rectangular neighborhood consisting of a $1 \times n$ array of pixels. The intensity is taken to increase monotonically from left to right, as in (b); the intensity contours in (a) are taken to be parallel to the short sides of the neighborhood.

Source: © RPS 2000

Table 3.4 Properties of the Three Paradigm Filters

Filter	r	η	B	C	D
Median	$\frac{1}{2}(n+1)$	0	$\frac{1}{2}A$	$\frac{1}{2}A$	0
Max	n	-1	A	0	$-\frac{1}{2}(n-1)$
Min	1	1	0	A	$\frac{1}{2}(n-1)$

© RPS 2000.

These definitions underline that a rank order filter will in general produce a D-pixel shift, whose value is:

$$D = \frac{1}{2}(n+1) - r \tag{3.22}$$

Before proceeding further, it will be useful to introduce a parameter η that is more symmetric than r, and has value $+1$ at $r = 1$ and -1 at $r = n$:

$$\eta = (n - 2r + 1)/(n - 1) \tag{3.23}$$

Using this parameter in preference to r, we can write down new formulae for B, C, D:

$$B = \frac{1}{2}A(1-\eta) \tag{3.24}$$

$$C = \frac{1}{2}A(1+\eta) \tag{3.25}$$

$$D = \frac{1}{2}\eta(n-1) \tag{3.26}$$

The properties of the three paradigm filters are summarized in Table 3.4 in terms of these parameters.

We now proceed to a continuum model, assuming a large number of pixels in any neighborhood (i.e., $n \to \infty$). The main difference will be that we shall specify distance in terms of the half-length a of the neighborhood rather than in terms of numbers of pixels:

$$D = \eta a \tag{3.27}$$

Next note that this formulation is independent of the width of the neighborhood, so long as the latter is rectangular. We now generalize the situation by taking the neighborhood to be rectangular and of dimensions $2a$ by $2\tilde{a}$ (Fig. 3.26).

The next task is to determine the result of a curvature $\kappa = 1/b$ in the intensity contours. Here we approximate the equation of a circle of radius b, with its diameter on the positive x-axis and passing through the origin, as:

$$x = \frac{y^2}{2b} \tag{3.28}$$

We can integrate the area under an intensity contour (see Fig. 3.26) as follows:

$$\begin{aligned} K &= \int_{-\tilde{a}}^{\tilde{a}} x\,dy = \left(\frac{1}{2b}\right) \int_{-\tilde{a}}^{\tilde{a}} y^2\,dy = \left(\frac{1}{2b}\right) \left[\frac{y^3}{3}\right]_{-\tilde{a}}^{\tilde{a}} \\ &= \frac{\tilde{a}^3}{3b} = \frac{1}{3}\kappa\tilde{a}^3 \end{aligned} \tag{3.29}$$

We deduce that the shift D is given by:

$$B = 2\tilde{a}\,(a - D) + \frac{1}{3}\kappa\tilde{a}^3 \tag{3.30}$$

$$C = 2\tilde{a}(a + D) - \frac{1}{3}\kappa\tilde{a}^3 \tag{3.31}$$

$$\eta = \frac{(C - B)}{A} = \frac{4\tilde{a}D - (2/3)\kappa\tilde{a}^3}{A} \tag{3.32}$$

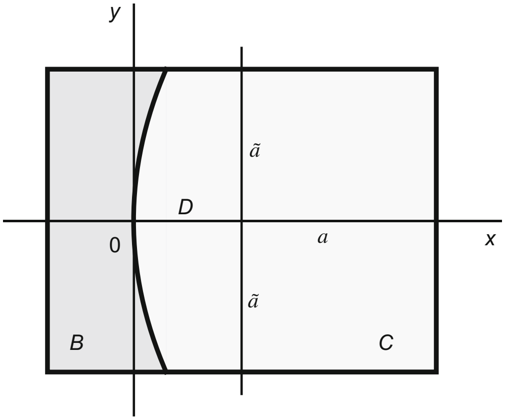

FIGURE 3.26

Geometry of a rectangular neighborhood with curved intensity contours. Here the neighborhood is a general rectangular neighborhood of dimensions $2a \times 2\tilde{a}$. Again, the intensity is taken to increase monotonically from left to right; the intensity contours are taken to be parallel and in this case are curved with identical curvature κ. x and y axes needed for area calculations are also shown. B and C represent the areas of the two shaded regions on either side of the thick intensity contour.

Source: © *RPS 2000*

where

$$A = 4a\tilde{a} \tag{3.33}$$

Hence:

$$D = \frac{\eta A}{4\tilde{a}} + \frac{1}{6}\kappa\tilde{a}^2 = \eta a + \frac{1}{6}\kappa\tilde{a}^2 \tag{3.34}$$

What is important about this equation is that it shows that the effects of rank order and curvature can be calculated and summed separately, the first term being that obtained above for the case of zero curvature and the second term being exactly that calculated for a median filter when the intensity contour is of length $2\tilde{a}$ (the earlier calculation (Davies, 1989b) related to a circular neighborhood). Thus, in principle we merely need to recompute the first term for any appropriate shape of neighborhood. However, various complications arise, particularly in the case of high curvature contours. These have been dealt with successfully, with the results shown in Figs. 3.27 and 3.28 (Davies, 2000f).

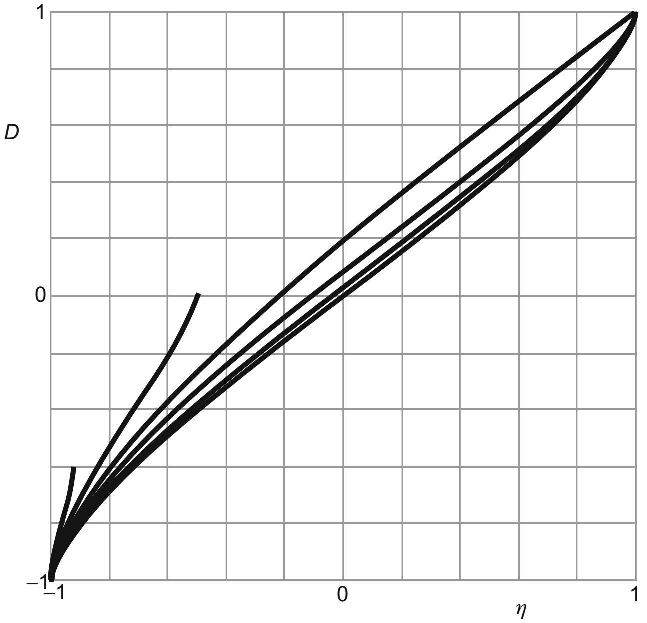

FIGURE 3.27

Graphs of shift D against rank order parameter η for various κ. This diagram summarizes the operation of rank order filters, with graphs, bottom to top, respectively, for $\kappa = 0$, $0.2/a$, $0.5/a$, $1/a$, $2/a$, $5/a$. Note that graphs for which $b < a$ ($\kappa > 1/a$) apply for restricted ranges of η and D (see Section 3.12.1). A multiplier of a must be included in the D-values.

Source: © *RPS 2000*

As the above theory is based on a continuum model, it is not perfect, and it does indicate only the main features of the practical situation. In particular, when the curvatures are very high, they may arise from spots that are entirely within the neighborhood, and then there is the possibility that they will be completely eliminated by the rank order filter (note that noise points are entirely eliminated by a median filter, which indeed is the prime practical use of that type of filter). Correspondingly, the assumptions made in the model break down when there is no intersection of the circular neighborhood and the intensity contour of radius $b = 1/\kappa$.

Some of the conclusions of this work are quite important. In particular, the result for a median filter is the special case that arises when $\eta = 0$ and is in agreement

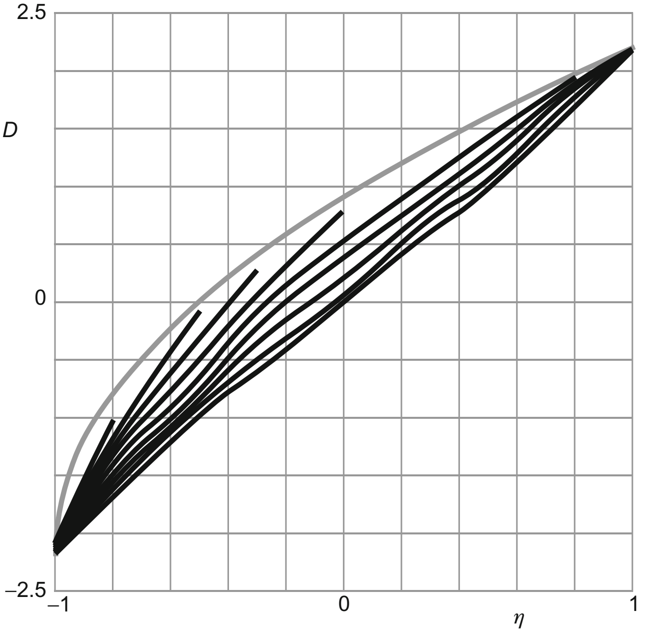

FIGURE 3.28

Shifts obtained for a typical discrete neighborhood. These shifts were obtained for rank order filters operating within a truncated 5 × 5 neighborhood when applied to eight discrete circular disks with radii ranging from 10.0 down to 1.25 pixels, the mean curvatures being 0.1–0.8 in steps of 0.1; the lowest curve was obtained by averaging the responses from circular disks of radius ±20.0 pixels, with curvatures ±0.05, and to the given scale are indistinguishable from the result that would be obtained with zero curvature. The uppermost curve represents the theoretical limiting value. However, because of the directional effects that occur in the discrete case, the upper limit is actually lower than indicated by this curve (see text).

Source: © *RPS 2000*

with the calculations of Section 3.8. Next, the max and min filters are also special cases and occur for $\eta = -1$ and 1, respectively. In these limiting cases, the shifts are $D = -a$ and a, respectively, the results being independent of κ: this is as might be expected since the value of $\tilde{a}$ is zero in each case. Between the max and min filters and the median filter, there is a continuous gradation of performance, with very significant but opposite shifts for the max and min filters, and the two basic effects canceling out for median filters—although the cancellation is only exact for straight contours. The full situation is summarized in Fig. 3.27.

3.13 THE ROLE OF FILTERS IN INDUSTRIAL APPLICATIONS OF VISION

It has been shown above how the median filter can successfully remove noise and artifacts such as spots and streaks from images. Unfortunately, many useful features such as fine lines and important points and holes are effectively indistinguishable from spots and streaks. In addition, it has been seen that the median filter "softens" pictures by removing fine detail. It is also found to clip corners of objects—another generally undesirable trait (but see Chapter 6). Finally, although it does not blur edges, it can still shift them slightly. In fact, shifting of curved edges seems to be a general characteristic of noise suppression filters.

Such distortions are quite alarming and mitigate against the indiscriminate use of filters. If applied in situations where accurate measurements are to be made on images, particular care must be taken to test whether the data are being biased in any way. Although it is possible to make suitable corrections to the data, it seems a good general policy to employ noise removal filters only where they are absolutely essential for object visibility. The alternative is to employ edge detection and other operators that automatically suppress noise as an integral part of their function. This is the general approach taken in subsequent chapters: indeed, it is one of the principles underlined in this book that algorithms should be "robust" against noise or other artifacts that might upset measurements. There is quite significant scope for the design of robust algorithms, since images contain so much information that it is normally possible to arrange for erroneous information to be ignored.

3.14 COLOR IN IMAGE FILTERING

In Chapter 2, it was indicated that color often adds to the complexity of image analysis algorithms, and could also add to the associated computational costs. From these points of view color might, except for applications such as assessing the ripeness of fruit, be regarded as an irrelevant luxury. Nevertheless, in the field of image processing and image filtering, where good quality images have to be presented to human operators, it is a vital concern. In fact, in recent years much effort has been devoted to the development of effective color filtering algorithms. Here we shall consider mainly median and related impulse noise filtering procedures.

Perhaps the first point to note is that median filtering is defined in terms of sorting operations and is thus undefined in the color domain, which normally contains three dimensions. However, a simple solution is to apply a standard median filter to each of the color channels, and then to reassemble the color image. Unfortunately, this approach leads to certain problems, the most obvious one being that of color "bleeding" (Fig. 3.12). This occurs when an impulse noise point appears in just one of the channels and is situated near an edge or other image feature. The case of an impulse noise point near an edge is hereby illustrated in simplified form:

$$\begin{array}{lcccccccccccc} \text{Original:} & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ \text{Filtered:} & ? & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & ? \end{array}$$

We see that a 3-element median filter eliminates the impulse noise point but at the same time moves the edge toward it. The end result for a color image is that the edge will be tinted with the color of the impulse noise point.

Fortunately, there is a standard solution to this problem. First, note that it is possible to express single-channel median filtering as the minimization of a distance metric, and this metric is trivially extendible to three color channels (or indeed any number of channels). The relevant single channel metric is:[4]

$$\text{median} = \arg\ \min_i \sum\nolimits_j |d_{ij}| \tag{3.35}$$

where d_{ij} is the distance between sample points i and j in the single-channel (gray scale) space. In the three-color domain, the metric is readily extended to:

$$\text{median} = \arg\ \min_i \sum\nolimits_j |\tilde{d}_{ij}| \tag{3.36}$$

where $\tilde{d}_{ij}$ is the generalized distance between sample points i and j, and we typically take the L_2 norm to define the distance measure for three colors:

$$\tilde{d}_{ij} = \left[\sum_{k=1}^{3} (I_{i,k} - I_{j,k})^2 \right]^{1/2} \tag{3.37}$$

Here $\mathbf{I}_j$, $\mathbf{I}_j$ are RGB vectors and $I_{i,k}$, $I_{j,k}$ ($k = 1, 2, 3$) are their color components.

[4] "arg min" is a standard mathematical term that means the argument (here pixel intensity) corresponding to the index (here i) giving rise to the minimum value of the expression in Eq. (3.36) (here $\Sigma_j\ |d_{ij}|$).

While the resulting vector median filter (VMF) no longer treats the individual color components separately, it is by no means guaranteed to eliminate color bleeding completely. In fact, like the standard median, it replaces any noisy intensity $\mathbf{I}_\mathrm{n}$, (including color) by the intensity $\mathbf{I}_j$ of another pixel that exists in the same window—rather than by an ideal intensity $\mathbf{I}$. Hence, color bleeding is only reduced, but not eliminated. If indeed there is a confluence of colors at any one point in an image, even in the absence of any impulse noise there is the possibility that these sorts of algorithms will become confused and inadvertently introduce small amounts of color bleeding: ultimately, the effect is due to the increased dimensionality of the data, which means that the algorithm has to contend with a greatly increased number of possible outcomes in spite of being an *ad hoc* procedure that does not embody specific understanding of images.

Figure 3.12 demonstrates the nature of color bleeding, albeit in the case of mode filtering: this figure shows vector median and vector mode filters to be remarkably free from color bleeding, but the same does not apply to scalar mode filters—for similar reasons to those indicated above for median filters.

3.15 CONCLUDING REMARKS

Although this chapter has dwelt on the implementation of noise suppression and image enhancement operators based on the local intensity distribution, it has made certain other points. In particular, it has shown the need to make a specification of the required imaging process and only then to work out the algorithm design strategy. Not only does this ensure that the algorithm will perform its function effectively, but also it should make it possible to optimize the algorithm for various practical criteria including speed, storage, and other parameters of interest. In addition, this chapter has demonstrated that any undesirable properties of the particular design strategy chosen (such as the inadvertent shifting of edges) should be sought and dealt with. Next, it has demonstrated a number of fundamental problems to do with imaging in discrete lattices—not least being problems of statistics that arise with small pixel neighborhoods. Finally, the large edge shifts of certain types of rank order filter are particularly important because they are turned to advantage in morphological operators (Chapter 7).

The next chapter moves on to a particularly vital problem in machine vision—that of segmenting images in order to find where objects are situated. This work builds on what has been learnt in the present chapter about edge profiles and how they are "seen" by neighborhood operators.

> Median filters have long been used to eliminate impulse noise without blurring edges. However, this chapter has shown that significant shifting of edges can result from use of median filters, and this property extends to mode filters and a fortiori to rank order filters—so much so that the latter form the basis of morphological processing.

3.16 BIBLIOGRAPHICAL AND HISTORICAL NOTES

Much of the work of this chapter has built on a paper by the author (Davies, 1988c), which rests on considerable earlier work on Gaussian, median, and other rank order filters (Hodgson et al., 1985; Duin et al., 1986). Note that the edge shifts that occur for median filters are not limited to this type of filter but apply almost equally to mean filters (Davies, 1991b). In addition, other inaccuracies have been found with median filters and methods have been found to correct them (Davies, 1992e).

The early literature hardly mentions mode filters, presumably because of the difficulty of finding simple mode estimators that are not unduly confused by noise and which still operate rapidly. Indeed, only one early reference has been found (Coleman and Andrews, 1979), although it has been backed up by later work (e.g., Evans and Nixon, 1995; Griffin, 2000). Other work referred to here is that on decomposing Gaussian and median filters (Narendra, 1978; Wiejak et al., 1985), and the many papers on fast implementation of median filters (e.g., Narendra, 1978; Huang et al., 1979; Danielsson, 1981; Davies, 1992a).

Considerable efforts have been devoted to studying the "root" behavior of the median filter, i.e., the result of applying median filtering operations until no further change occurs. In fact, much of this work has been carried out on 1-D signals, including cardiac and speech waveforms, rather than on images (Gallagher and Wise, 1981; Fitch et al., 1985; Heinonen and Neuvo, 1987). Root behavior is of interest as it relates to the underlying structure of signals, although its realization involves considerable amounts of processing. Some of the work on filtering aims to improve on rather than to emulate the median filter. Work of this type includes the detail-preserving filters of Heinonen and others (Nieminen et al., 1987) and relates to the lower set of plots in Fig. 3.18. See also the neural network approach to this topic (e.g., Greenhill and Davies, 1994a). More recent work on nonlinear filtering appears in Marshall et al. (1998): see Marshall (2004) for a new design method for weighted order statistics filters.

The author (Davies, 1987c) has reported methods of optimizing linear smoothing filters in small neighborhoods by minimizing the total error in fitting them to a continuous Gaussian function: a balance has to be struck between subpixel errors within the neighborhood and errors that arise from the proportion of the distribution that lies outside the neighborhood (Fig. 3.29).

With the advent of extremely low cost color frame grabbers on PCs, and the widespread use of digital cameras, digital color images have become ubiquitous, and this has extended to (or even necessitated) much research on color filtering. A useful summary of work in this area up to 1998 appears in Sangwine and Horne (1998). More recent work on vector (color) filtering includes that of Lukac (2003). Charles and Davies (2003b) describe new distance-weighted median filters and their application to color images. They also extend the author's earlier mode filter work to color images (Charles and Davies, 2003a, 2004). Davies's (2000e) theorem shows that restricting a multichannel (color) filter output to the

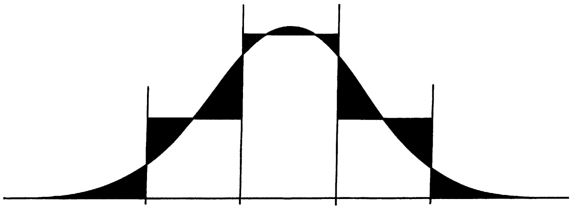

FIGURE 3.29

Approximating a discrete to a continuous Gaussian. This diagram shows how a balance needs to be struck between subpixel errors and those arising from the truncated part of the function.

vector value of one of the input sample points (i.e., from the current window in the image) will increase the inaccuracy present in the final image, for a large proportion of pixels: since this represents the usual vector median strategy that is employed to minimize color bleeding, the effectiveness of the current generation of color filter algorithms needs to be looked at further.

Davies has further analyzed the distortions and edge shifts produced by a range of rank order, mean, and mode filters, and has produced a unified review of the subject (Davies, 2003e). In the case of median filters, it proved possible, and necessary for high accuracy, to produce a discrete model of the situation (Davies, 2003c), rather than extending the continuum model described much earlier (Davies, 1989b).

3.16.1 More Recent Developments

The 2000s have seen a new approach to filtering via "switched" types of filter that judge whether or not any pixel is corrupted by impulse noise: if the latter, they use a method such as the median or vector-median filter to eliminate it; if the former, they adopt a policy of zero change by using the original pixel intensity or color. The zero change policy is useful because it helps maintain image sharpness and fidelity. An early example of this approach was the work of Eng and Ma (2001): see Chen et al. (2009) and Smolka (2010) for recent, more sophisticated versions of this concept (Smolka's version falls in the category of a "peer group switching filter").

Davies (2007b) has studied the properties of the generalized (nonvector) median filter that has the capability for eliminating even more noise than the VMF while not being targeted so specifically at eliminating color bleeding. He demonstrates ways of implementing the filter so that it runs sufficiently rapidly to make it a viable alternative to the VMF.

Celebi (2009) has shown how to reduce the computational needs of directional vector filters based on order statistics without significant loss of accuracy. At another end of the scale, Rabbani and Gazor (2010) have shown how to reduce additive Gaussian noise by using local mixture models; they find that of the wavelet types of local representation, the discrete complex wavelet transform is preferable in terms of both peak noise performance and computational cost.

3.17 PROBLEMS

1. Draw up a table showing the numbers of operations required to implement a median filter in various sizes of the neighborhood. Include in your table (i) results for a straight bubble sort of all n^2 pixels, (ii) results for bubble sorts in separated $1 \times n$ and $n \times 1$ neighborhoods, and (iii) results for the histogram method of Section 3.3. Discuss the results, taking account of possible computational overheads.
2. Show how to perform a median filtering operation on a binary image. Show also that if a set of binary images is formed by thresholding a grayscale image at various levels, and each of these binary images is median filtered, then a grayscale image can be reconstructed that is a median filtered version of the original grayscale image. Consider to what extent the reduced amount of computation in filtering a binary image compensates for the number of separate thresholded images to be filtered.
3. An "extremum" filter is an image-parallel operation that assigns every pixel the intensity value closer to the two extreme values in its local intensity distribution. Show that it should be possible to use such a filter to enhance images. What would be the *disadvantage* of such a filter?
4. Under what conditions is a 1-D signal that has been filtered once by a median filter a root signal? What truth is there in the statement that a straight edge in an image is neither shifted nor blurred by a median filter, whatever its cross-section?
5. **a.** Explain the action of the following median filtering algorithm:

```
for all pixels in image do{
 for (i = 0; i <= 255; i ++) hist[i] = 0;
 for (m = 0; m <= 8; m ++) hist[P[m]] ++;
 i = 0; sum = 0;
 while (sum < 5) {
   sum = sum + hist[i];
   i = i + 1;
 }
 Q0 = i - 1;
}
```

   b. Show how this algorithm can be speeded up (i) by a more efficient histogram clearing technique and (ii) by calculating the minimum intensity in each 3×3 window. In each case, estimate approximately how much the algorithm will be speeded up.

   c. Explain why a median filter is able to smooth images without introducing blurring.

d. A 1-D cross-section of an image has the following intensity profile:

1 2 1 1 2 3 0 2 2 3 1 1 2 2 9 2 2 8 8 8 7 8 8 7 9 9 9

Apply (i) a 3-element median filter and (ii) a 5-element median filter to this profile. With the aid of these examples, show that median filters tend to produce "runs" of constant values in 1-D profiles. Show also that under some circumstances an edge in the profile can be shifted by a nearby spike: give a rule showing when this is likely to occur for an *n*-element median filter in one dimension.

6. a. A *mode* filter is defined as one in which the new pixel intensity at any pixel takes the most probable value in the local intensity distribution of a window placed around that pixel in the original image space. Show for a grayscale image that a mode filter will, if anything, sharpen the image, while a *mean* filter will tend to blur the image.

b. A *max* filter is one that takes the maximum value of the local intensity distribution in a window around each pixel. Explain what will be seen when a max filter is applied to an image. Consider whether any similar effects are liable to happen when a mode filter is applied to an image.

c. Explain the purpose of a *median* filter. Why are 2-D median filters sometimes implemented as two 1-D median filters applied in sequence?

d. Contrast the behavior of 5-element 1-D mean, max, and median filters as applied to the following waveform:[5]

0 1 1 2 3 2 2 0 2 3 9 3 2 4 4 6 5 6 7 0 8 8 9 1 1 8 9

e. Work out what would happen if the 1-D median filter were applied many times, starting with this waveform.

7. a. Determine the effect of applying (i) a 3 × 3 median filter and (ii) a 5 × 5 median filter to the portion of an image shown in Fig. 3.30.

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	2	0	0
0	1	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0
0	0	0	0	9	9	9	9	9	9
0	0	0	0	9	9	8	9	9	9
0	0	1	0	9	8	9	9	7	9
0	0	0	0	7	9	9	8	9	9
0	1	0	8	9	9	9	9	9	9
0	0	0	0	9	9	9	9	9	9
0	0	0	0	8	9	9	9	9	9

FIGURE 3.30

Portion of image for tests of median filter.

[5] For the mean filter, give the nearest integer value in each case.

b. Show that it should be possible to develop a corner detector based on the properties of these median filters. What advantages or disadvantages might result from employing this design strategy?

8. a. Distinguish between *mean* and *median filtering*. Explain why a mean filter would be expected to blur an image, while a median filter would not have this effect. Illustrate your answer by showing what happens in the following 1-D case with a window of size 1 × 3:

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

b. Give a complete median filter algorithm based on histograms and operating within a 3 × 3 window. Explain why it operates relatively slowly.

c. A computer language has the *max(a, b)* operation as standard. Show how it may be used to find the maximum intensity within a 3 × 3 window. Show also how it may be used to find the median by successively replacing the maximum values by zeros. If the *max(a, b)* operation is about the same speed as the $a + b$ operation, determine whether the median can be found any faster by this method.

d. Discuss whether splitting a 3 × 3 median operation into 1 × 3 and 3 × 1 median operations is likely to be effective at eliminating impulse noise in images. How would the speed of this approach be affected by use of the *max(a, b)* operation?

9. a. Determine the result of applying a 3-element median filter to the following 1-D signals:

i. 0 0 0 0 0 1 0 1 1 1 1 1 1 1
ii. 2 1 2 3 2 1 2 2 3 2 4 3 3 4
iii. 1 1 2 3 3 4 5 8 6 6 7 8 9 9

b. What general lessons can be learnt from the results? In the first case, consider also the corresponding situation for a grayscale edge in a 2-D image.

c. 2-D median filters are sometimes implemented as two 1-D median filters applied in sequence in order to improve the speed of processing. Estimate the gain in speed that could be achieved in this way for (i) a 3 × 3 median filter, (ii) a 7 × 7 median filter, and (iii) in the general case.