CHAPTER

4 Thresholding Techniques

One of the important practical aims of image processing is the demarcation of objects appearing in digital images. This process is called segmentation, and a good approximation to it can often be achieved by thresholding. Broadly, this involves separating the dark and light regions of the image, and thus identifying dark objects on a light background (or vice versa). This chapter discusses the effectiveness of this idea and the means for achieving it.

Look out for:

- the segmentation, region-growing and thresholding concepts.
- the problem of threshold selection.
- the limitations of global thresholding.
- problems in the form of shadows or glints (highlights).
- the possibility of modeling the image background.
- the idea of adaptive thresholding.
- the rigorous Chow and Kaneko approach.
- what can be achieved with simple local adaptive thresholding algorithms.
- more thoroughgoing variance, entropy-based, and maximum likelihood methods.
- the possibility of modeling images by multilevel thresholding.
- the value of the global valley transformation.
- how thresholds can be found in unimodal distributions.

Thresholding is limited in what it can achieve, and there are severe difficulties in automatically estimating the optimum threshold—as evidenced by the many available techniques that have been devised for the purpose. In fact, segmentation is an ill-posed problem, and it is misleading that the human eye appears to perform thresholding reliably. Nevertheless, there are instances where the task can be simplified, for example, by suitable lighting schemes, so that thresholding becomes effective. Hence, it is a useful technique that needs to be included in the toolbox of available algorithms for use when appropriate. However, edge detection (Chapter 5) provides an alternative highly effective means to key into complex image data.

4.1 INTRODUCTION

One of the first tasks to be undertaken in vision applications is to segment objects from their backgrounds. When objects are large and do not possess very much surface detail, segmentation can be imagined as splitting the image into a number of regions each having a high level of uniformity in some parameter such as brightness, color, texture or even motion. Hence, it should be straightforward to separate objects from one another and from their background, and also to discern the different facets of solid objects such as cubes.

Unfortunately, the concept of segmentation presented above is an idealization that is sometimes reasonably accurate, but more often in the real world, it is an invention of the human mind, generalized inaccurately from certain simple cases. This problem arises because of the ability of the eye to understand real scenes at a glance, and hence to segment and perceive objects within images in the form they are known to have. Introspection is not a good way of devising vision algorithms, and it must not be overlooked that segmentation is actually one of the central and most difficult practical problems of machine vision.

Thus, the common view of segmentation as looking for regions possessing some degree of uniformity is to a large extent invalid. There are many examples of this in the world of 3-D objects: one is a sphere lit from one direction, the brightness in this case changes continuously over the surface so that there is no distinct region of uniformity; another is a cube where the direction of the lighting may lead to several of the facets having equal brightness values so that it is impossible from intensity data alone to segment the image completely as desired.

Nevertheless, there is sufficient correctness in the concept of segmentation by uniformity measures for it to be worth pursuing for practical applications. The reason is that in many (especially industrial) applications, only a very restricted range and number of objects are involved, and in addition it is possible to have almost complete control over the lighting and the general environment. The fact that a particular method may not be completely general need not be problematic, since by employing tools that are appropriate for the task in hand, a cost-effective solution will have been achieved in that case at least. However, in practical situations, there is clearly a tension between simple cost-effective solutions and general-purpose but more computationally expensive solutions; this tension must always be kept in mind in severely practical subjects such as machine vision.

4.2 REGION-GROWING METHODS

The segmentation idea outlined in Section 4.1 leads naturally to the region-growing technique (Zucker, 1976b). Here, pixels of like intensity (or other suitable property) are successively grouped together to form larger and larger

regions until the whole image has been segmented. Clearly, there have to be rules about not combining adjacent pixels that differ too much in intensity, while permitting combinations for which intensity changes gradually because of variations in background illumination over the field of view. However, this is not enough to make a viable strategy, and in practice the technique has to include the facility not only to merge regions together but also to split them if they become too large and inhomogeneous (Horowitz and Pavlidis, 1974). Particular problems are noise and sharp edges and lines that form disconnected boundaries, and for which it is difficult to formulate simple criteria to decide whether they form true region boundaries. In remote sensing applications, for example, it is often difficult to separate fields rigorously when hedges are broken and do not give continuous lines: in such applications, segmentation may have to be performed interactively, with a human operator helping the computer. Hall (1979) found that in practice regions tend to grow too far,[1] so that to make the technique work well it is necessary to limit their growth with the aid of edge detection schemes.

Thus, the region-growing approach to segmentation turns out to be quite complex to apply in practice. In addition, region-growing schemes usually operate iteratively, gradually refining hypotheses about which pixels belong to which regions. The technique is complicated because, carried out properly, it involves global as well as local image operations. Thus, each pixel intensity will in principle have to be examined many times, and as a result the process tends to be quite computation intensive. For this reason, it is not considered further here, since we are often more interested in methods involving low computational load that are amenable to real-time implementation.

4.3 THRESHOLDING

If background lighting is arranged so as to be fairly uniform, and we are looking for rather flat objects that can be silhouetted against a contrasting background, segmentation can be achieved simply by thresholding the image at a particular intensity level. This possibility was apparent from Fig. 2.2. In such cases, the complexities of the region-growing approach are bypassed. The process of thresholding has already been covered in Chapter 2, the basic result being that the initial grayscale image is converted into a binary image in which objects appear as black figures on a white background, or as white figures on a black background. Further analysis of the image then devolves into analysis of the shapes and dimensions of the figures: at this stage, object identification should be straightforward. Chapter 9 concentrates on such tasks. Meanwhile, there is one outstanding problem—how to devise an automatic procedure for determining the optimum thresholding level.

[1]Clearly, there is a danger that even one small break could join two regions into a single larger one.

4.3.1 Finding a Suitable Threshold

One simple technique for finding a suitable threshold arises in situations such as optical character recognition (OCR) where the proportion of the background that is occupied by objects (i.e., print) is relatively constant in a variety of conditions. A preliminary analysis of relevant picture statistics then permits subsequent thresholds to be set by insisting on a fixed proportion of dark and light in a sequence of images (Doyle, 1962). In practice, a series of experiments is performed in which the thresholded image is examined as the threshold is adjusted, and the best result ascertained by eye: at that stage, the proportions of dark and light in the image are measured. Unfortunately, any changes in noise level following the original measurement will upset such a scheme, since they will affect the relative amounts of dark and light in the image. However, this is frequently a useful technique in industrial applications, especially when particular details within an object are to be examined: typical examples of this are holes in mechanical components such as brackets (note that the mark–space ratio for objects may well vary substantially on a production line, but the proportion of hole area *within* the object outline would not be expected to vary).

The technique that is most frequently employed for determining thresholds involves analyzing the histogram of intensity levels in the digitized image (Fig. 4.1): if a significant minimum is found, it is interpreted as the required threshold value (Weska, 1978). Clearly, the assumption being made here is that the peak on the left of the histogram corresponds to dark objects, and the peak on the right corresponds to light background (here it is assumed that, as in many industrial applications, objects appear dark on a light background).

This method is subject to the following major difficulties:

1. the valley may be so broad that it is difficult to locate a significant minimum.
2. there may be a number of minima because of the type of detail in the image, and selecting the most significant one will be difficult.
3. noise within the valley may inhibit location of the optimum position.
4. there may be no clearly visible valley in the distribution because noise may be excessive or because the background lighting may vary appreciably over the image.
5. either of the major peaks in the histogram (usually due to the background) may be much larger than the other and this will then bias the position of the minimum.
6. the histogram may be inherently multimodal, making it difficult to determine which is the relevant thresholding level.

Perhaps the worst of these problems is the last point: that is, if the histogram is inherently multimodal, and we are trying to employ a single threshold, then we are applying what is essentially an *ad hoc* technique to obtain a meaningful result. In general, such efforts are unlikely to succeed, and this is clearly a case where full image interpretation must be performed before we could be sure that the results are valid. Ideally, thresholding rules have to be formed after many images

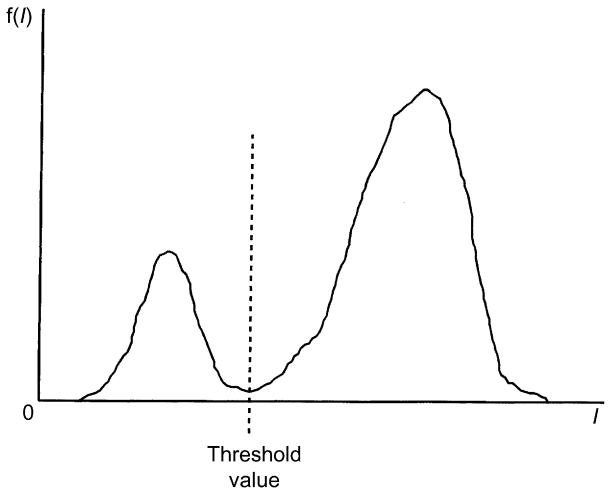


FIGURE 4.1

Idealized histogram of pixel intensity levels in an image. The large peak on the right results from the light background; the smaller peak on the left is due to dark foreground objects. The minimum of the distribution provides a convenient intensity value to use as a threshold.

have been analyzed. In what follows such problems of meaningfulness are eschewed and attention is concentrated on how best to find a genuine single threshold when its position is obscured as indicated by problems 1–5 above (which can be ascribed to image "clutter," noise, and lighting variations).

4.3.2 Tackling the Problem of Bias in Threshold Selection

This section considers problem 5 of Section 4.3.1—that of eliminating the bias in the selection of thresholds that arises when one peak in the histogram is larger than the other. First, note that if the relative heights of the peaks are known, this effectively eliminates the problem, since the "fixed proportion" method of threshold selection outlined above can be used. However, this is not normally possible. A more useful approach is to prevent bias by weighting down the extreme values of the intensity distribution and weighting up the intermediate values in some way. To achieve this, note that the intermediate values are special in that they correspond to object edges. Hence, a good basic strategy is to find positions in the image where there is a significant intensity gradient—corresponding to pixels in the regions of edges—and to analyze the intensity values of these locations while ignoring other points in the image.

One way of dealing with this is to construct "scattergrams" in which pixel properties are plotted on a 2-D map with intensity variation along one axis and

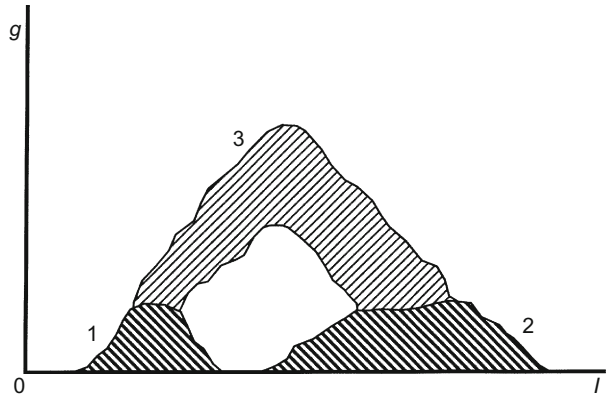


FIGURE 4.2

Scattergram showing the frequency of occurrence of various combinations of pixel intensity *I* and intensity gradient magnitude *g* in an idealized image. There are three main populated regions of interest: (1) a low-*I*, low-*g* region; (2) a high-*I*, low-*g* region; and (3) a medium-*I*, high-*g* region. Analysis of the scattergram sometimes provides useful information on how to segment the image.

intensity gradient magnitude variation along the other. As indicated in Fig. 4.2, there are three main populated regions on the map: (1) a low-intensity, low-gradient region corresponding to the dark objects; (2) a high-intensity, low-gradient region corresponding to the background; and (3) a medium-intensity, high-gradient region corresponding to object edges (Panda and Rosenfeld, 1978). By analyzing how these regions merge into each other, it is sometimes possible to obtain better results than can be obtained using simple thresholding. In particular, by examining the situation for moderate values of gradient, bias may be reduced, as indicated above. However, instead of constructing a scattergram, we can try weighting the plots in the intensity histogram in such a way as to minimize threshold bias: this possibility is discussed in the following section.

4.3.2.1 Methods Based on Finding a Valley in the Intensity Distribution

This section considers how to weight the intensity distribution using a parameter other than the intensity gradient, in order to locate accurately the valley in the intensity distribution. A simple strategy is first to locate all pixels that have a significant intensity gradient, and then to find the intensity histogram not only of these pixels but also of nearby pixels. This means that the two main modes in the intensity distribution are still attenuated very markedly and hence the bias in the valley position is significantly reduced. Indeed, the numbers of background and foreground pixels that are now being examined are very similar, so the bias

from the relatively large number of background pixels is virtually eliminated (note that if the modes are modeled as two Gaussian distributions of equal widths and they also have equal heights, then the minimum lies exactly halfway between them).

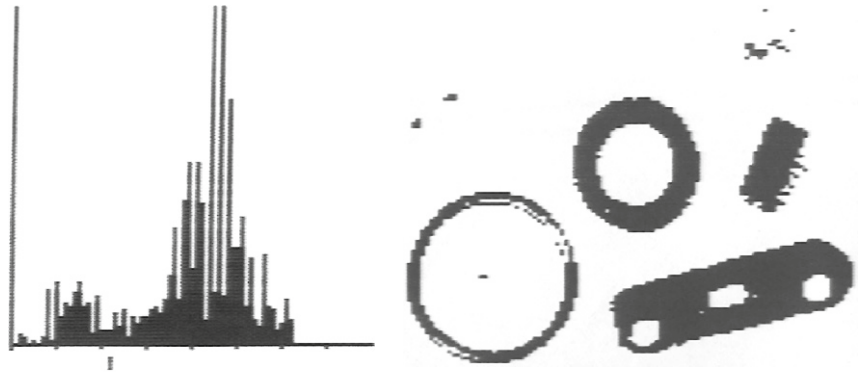
Although obvious, this approach clearly includes the edge pixels themselves, which tend to fill the valley between the two modes. For the best results, the points of highest gradient must actually be removed from the intensity histogram. A well-attested way of achieving this is to weight pixels in the intensity histogram according to their response to a Laplacian filter (Weska et al., 1974). Since such a filter gives an isotropic estimate of the second derivative of the image intensity (i.e., the magnitude of the first derivative of the intensity gradient), it is zero where intensity gradient magnitude is high: hence, it gives such locations zero weight, but it nevertheless weights up those locations on the shoulders of edges. It has been found that this approach is very good at estimating where to place a threshold within a wide valley in the intensity histogram (Weska et al., 1974).

4.3.3 Summary

It has been shown that available techniques are able to provide values at which intensity thresholding can be applied, but they do not themselves solve the problems caused by uneven lighting. They are even less capable of coping with glints, shadows and image clutter. Unfortunately, these artifacts are common in most real situations (Figs. 4.3–4.5) and are only eliminated with difficulty in practice. Indeed, in industrial applications where shiny metal components are involved, glints are the rule rather than the exception, while shadows can seldom be avoided with any sort of object. Even flat objects are liable to have quite strong shadow contours around them because of the particular placement of lights. Lighting problems are studied in detail in Chapter 25. Meanwhile, note that glints and shadows can only be allowed for properly in a two-stage image analysis system, where tentative assignments are made first, and these are firmed up by exact explanation of all pixel intensities. We now return to the problem of making the most of the thresholding technique, by finding how variations in background lighting can be allowed for.

4.4 ADAPTIVE THRESHOLDING

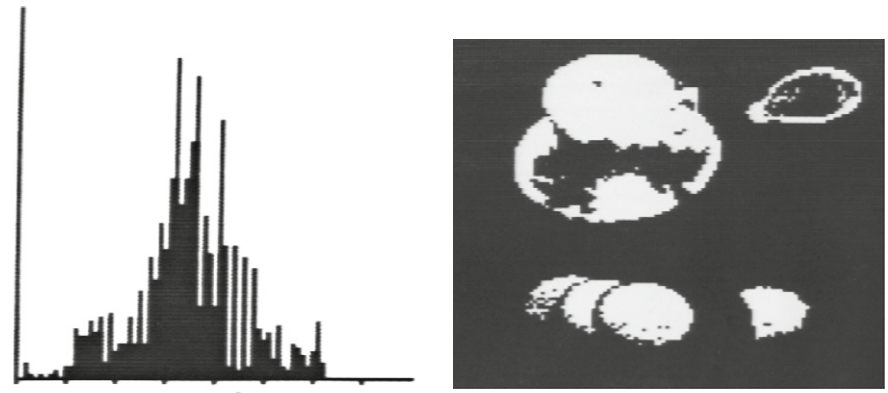
The problem that arises when illumination is not sufficiently uniform may be tackled by permitting the threshold to vary adaptively (or "dynamically") over the whole image. In principle, there are several ways of achieving this. One involves modeling the background within the image. Another is to work out a local threshold value for each pixel by examining the range of intensities in its neighborhood. A third approach is to split the image into subimages and deal



(a) (b)

FIGURE 4.3

Histogram for the image shown in Fig. 2.7(a). Note that the histogram is not particularly close to the ideal form of Fig. 4.1. Hence, the threshold obtained from (a) (indicated by the short line beneath the scale) does not give ideal results with all the objects in the binarized image (b). Nevertheless, the results are better than for the arbitrarily thresholded image of Fig. 2.7(b).



(a) (b)

FIGURE 4.4

Histogram for the image shown in Fig. 2.1(a). The histogram is not at all close to the idealized form, and the results of thresholding (b) are not a particularly useful aid to interpretation.

with them independently. Although "obvious," the last method will clearly run into problems at the boundaries between subimages, and by the time these problems have been solved, it will look more like one of the other two methods.

The problem can sometimes be solved rather neatly in the following way. On some occasions—such as in automated assembly applications—it is possible to

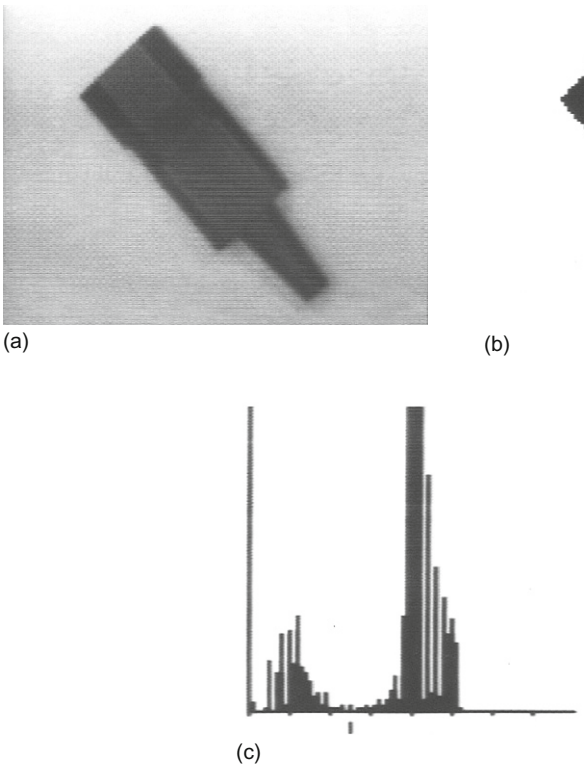


FIGURE 4.5

A picture with more ideal properties. (a) Image of a plug that has been lit fairly uniformly. The histogram (c) approximates to the ideal form, and the result of thresholding (b) is acceptable. However, much of the structure of the plug is lost during binarization.

obtain an image of the background in the absence of any objects. This appears to solve the problem of adaptive thresholding in a rigorous manner, since the tedious task of modeling the background has already been carried out. However, caution is needed because objects bring with them not only shadows (which can in some sense be regarded as part of the objects) but also an additional effect due to the reflections they cast over the background and other objects. This additional effect is nonlinear in the sense that it is necessary to add not only the difference between the object and the background intensity in each case but also an intensity that depends on the products of the reflectances of pairs of objects. These considerations mean that using the no-object background as the equivalent background when several objects are present is ultimately invalid. However, as a first approximation, it is frequently possible to assume an equivalence. If this proves impracticable, there is no option but to model the background from the actual image to be segmented.

On other occasions, the background intensity may be rather slowly varying, in which case it may be possible to model it by the following technique (this is a form of Hough transform—see Chapter 11). First, an equation is selected, which can act as a reasonable approximation to the intensity function, for example, a quadratic variation:

$$I = a + bx + cy + dx^2 + exy + fy^2 \tag{4.1}$$

Next, a parameter space for the six variables a, b, c, d, e, f is constructed; then each pixel in the image is taken in turn and all sets of values of the parameters that could have given rise to the pixel intensity value are accumulated in parameter space. Finally, a peak is sought in parameter space, which represents an optimal fit to the background model. So far it appears that this has been carried out only for a linear variation, the analysis being simplified initially by considering only the differences in intensities of pairs of points in image space (Nixon, 1985). Note that a sufficient number of pairs of points must be considered so that the peak in parameter space resulting from background pairs is sufficiently well populated.

4.4.1 The Chow and Kaneko Approach

As early as 1972, Chow and Kaneko introduced what is widely recognized as the standard technique for dynamic thresholding: the technique performs a thoroughgoing analysis of the background intensity variation, making few compromises to save computation (Chow and Kaneko, 1972). In this method, the image is divided into a regular array of overlapping subimages and individual intensity histograms are constructed for each one. Those that are unimodal are ignored since they are assumed not to provide any useful information that can help in modeling the background intensity variation. However, the bimodal distributions are well suited to this task: these are individually fitted to pairs of Gaussian distributions of adjustable height and width and the threshold values are located. Thresholds are then found, by interpolation, for the unimodal distributions. Finally, a second stage of interpolation is necessary to find the correct thresholding value at each pixel.

One problem with this approach is that if the individual subimages are made very small in an effort to model the background illumination more exactly, the statistics of the individual distributions become worse, their minima become less well defined and the thresholds deduced from them are no longer statistically significant. This means that it does not pay to make subimages too small and that ultimately only a certain level of accuracy can be achieved in modeling the background in this way. Clearly, the situation is highly data dependent, but it might be expected that little would be gained by reducing the subimage size below 32×32 pixels. Chow and Kaneko employed 256×256 pixel images and divided these into a 7×7 array of 64×64 pixel subimages with 50% overlap.

Overall, this approach involves considerable computation, and in real-time applications it may well not be viable for this reason.

4.4.2 Local Thresholding Methods

The other approach mentioned earlier is particularly useful for finding local thresholds. It involves analyzing intensities in the neighborhood of each pixel to determine the optimum local thresholding level. Ideally, the Chow and Kaneko histogramming technique would be repeated at each pixel, but this would significantly increase the computational load of this already computationally intensive technique. Thus, it is necessary to obtain the vital information by an efficient sampling procedure. One simple means for achieving this is to take a suitably computed function of nearby intensity values as the threshold: often the mean of the local intensity distribution is taken because this is a simple statistic and gives good results in some cases. For example, in astronomical images, stars have been thresholded in this way. Niblack (1985) reported a case in which a proportion of the local standard deviation was added to the mean to give a more suitable threshold value, the reason (presumably) being to help suppress noise (clearly, addition is appropriate where bright objects such as stars are to be located, whereas subtraction is more appropriate in the case of dark objects).

Another statistic that is frequently used is the mean of the maximum and minimum values in the local intensity distribution. The justification for this is that whatever the sizes of the two main peaks of the distribution, this statistic often gives a reasonable estimate of the position of the histogram minimum. The theory presented earlier shows that this method will only be accurate if (a) the intensity profiles of object edges are symmetrical, (b) noise acts uniformly everywhere in the image so that the widths of the two peaks of the distribution are similar, and (c) the heights of the two distributions do not differ markedly. Sometimes these assumptions are definitely invalid—e.g., when looking for (dark) cracks in eggs or other products. In such cases, the mean and maximum of the local intensity distribution can be found and a threshold deduced using the statistic

$$T = mean - (maximum - mean) \tag{4.2}$$

where the strategy is to estimate the lowest intensity in the bright background assuming the distribution of noise is symmetrical (Fig. 4.6): use of the mean here is realistic only if the crack is narrow and does not affect the value of the mean significantly. If it does, then the statistic can be adjusted by use of an *ad hoc* parameter:

$$T = mean - k(maximum - mean) \tag{4.3}$$

where k may be as low as 0.5 (Plummer and Dale, 1984).

This method is essentially the same as that of Niblack (1985), but the computational load in estimating the standard deviation is minimized. Each of the last two techniques relies on finding local extrema of intensity. Using these measures helps save computation, but they are clearly somewhat unreliable because of the effects of noise. If this is a serious problem, quartiles or other statistics of the distribution may be used. The alternative of prefiltering the image to remove noise is unlikely to work for crack thresholding, since cracks will almost certainly be removed at the same time as the noise. A better strategy is to form an image of

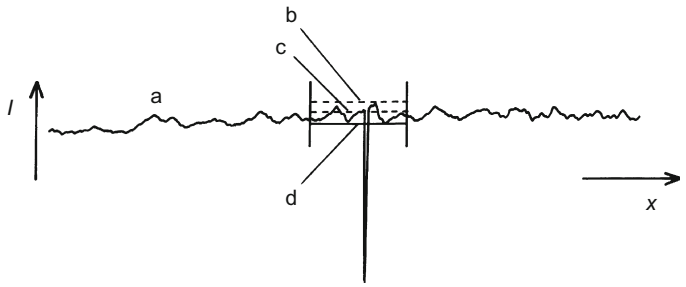


FIGURE 4.6

Method for thresholding the crack in an egg. a, Intensity profile of an egg in the vicinity of a crack: the crack is assumed to appear dark (e.g., under oblique lighting); b, local maximum of intensity on the surface of the egg; c, local mean intensity. Eq. (4.2) gives a useful estimator T of the thresholding level d.

T-values obtained using Eq. (4.2) or (4.3): smoothing this image should then permit the initial image to be thresholded effectively.

Unfortunately, all these methods work well only if the size of the neighborhood selected for estimating the required threshold is large enough to span a significant amount of foreground and background. In many practical cases, this is not possible and the method then adjusts itself erroneously, for example, so that it finds darker spots within dark objects as well as segmenting the dark objects themselves. However, there are certain applications where there is little risk of this occurring. One notable case is that of OCR. Here the widths of character limbs are likely to be known in advance and should not vary substantially. If this is so, then a neighborhood size can be chosen to span or at least sample both character and background, and it is thus possible to threshold the characters highly efficiently using a simple functional test of the type described above. The effectiveness of this procedure (Table 4.1) is demonstrated in Fig. 4.7.

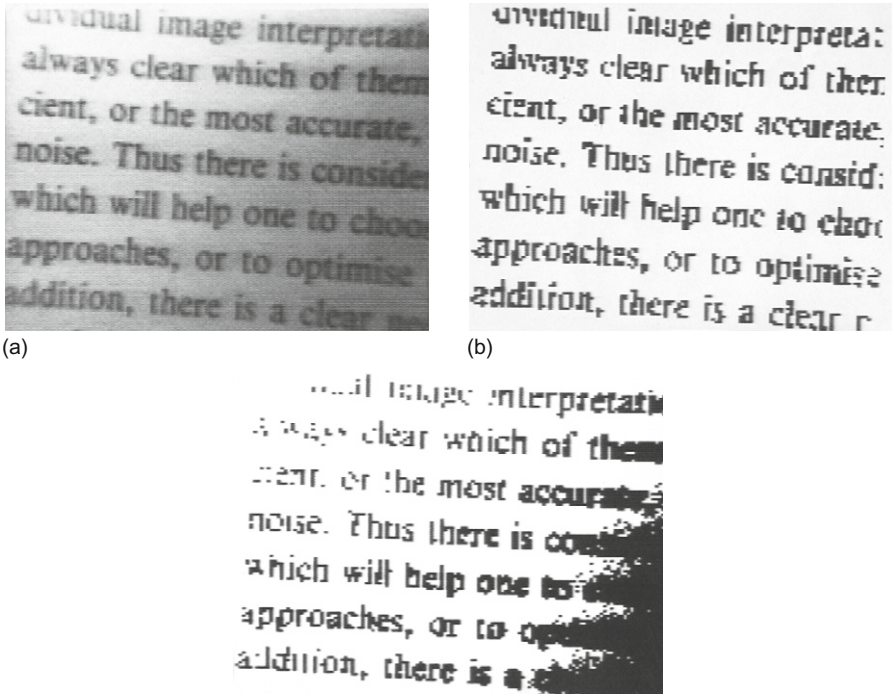
Finally, before leaving this topic, note that hysteresis thresholding is a type of adaptive thresholding—effectively permitting the threshold value to vary locally: this topic is investigated in Section 5.10.

4.5 MORE THOROUGHGOING APPROACHES TO THRESHOLD SELECTION

At this point, we return to global threshold selection and describe some important approaches that have a rigorous mathematical basis. The first of these is variance-based thresholding, the second is entropy-based thresholding, and the third is maximum likelihood thresholding. All three are widely used, the second having achieved an increasingly wide following over the past 20−30 years, and the third is a more broad-based technique that has its roots in statistical pattern recognition—a subject that is covered in Chapter 24.

Table 4.1 A Simple Algorithm for Adaptively Thresholding Print

```
minrange = 255 / 5;
/* minimum likely difference in intensity between print and background:
this parameter can be preset manually or "learnt" by a previous routine */
for all pixels in image do {
   find minimum and maximum of local intensity distribution;
   range = maximum – minimum;
   if (range > minrange)
      T = (minimum + maximum)/2; // print is visible in neighborhood
   else T = maximum – minrange/2; // neighborhood is all white
   if (P0 > T) Q0 = 255; else Q0 = 0; // now binarize print
}
```



(a) (b) (c)

FIGURE 4.7

Effectiveness of local thresholding on printed text. Here, a simple local thresholding procedure (Table 4.1), operating within a 3×3 neighborhood, is used to binarize the image of a piece of printed text (a). Despite the poor illumination, binarization is performed quite effectively (b). Note the complete absence of isolated noise points in (b), while by contrast the dots on all the i's are accurately reproduced. The best that could be achieved by uniform thresholding is shown in (c).

4.5.1 Variance-Based Thresholding

The standard approach to thresholding outlined earlier involved finding the neck of the global image intensity histogram. However, this is impracticable when the dark peak of the histogram is minuscule in size, as it will then be hidden among the noise in the histogram and it will not be possible to extract it with the usual algorithms.

A good many investigators have studied this sort of problem (e.g., Otsu, 1979; Kittler et al., 1985; Sahoo et al., 1988; Abutaleb, 1989): among the most well-known approaches are the variance-based methods. In these methods, the image intensity histogram is analyzed to find where it can best be partitioned to optimize criteria based on ratios of the within-class, between-class, and total variance. The simplest approach (Otsu, 1979) is to calculate the between-class variance, as will now be described.

First, we assume that the image has a grayscale resolution of L gray levels. The number of pixels with gray level i is written as n_i, so the total number of pixels in the image is $N = n_1 + n_2 + \cdots + n_L$. Thus, the probability of a pixel having gray level i is:

$$p_i = \frac{n_i}{N} \tag{4.4}$$

where

$$p_i \geq 0 \quad \sum_{i=1}^{L} p_i = 1 \tag{4.5}$$

For ranges of intensities up to and above the threshold value k, we can now calculate the between-class variance σ_B^2 and the total variance σ_T^2:

$$\sigma_B^2 = \pi_0\left(\mu_0 - \mu_T\right)^2 + \pi_1\left(\mu_1 - \mu_T\right)^2 \tag{4.6}$$

$$\sigma_T^2 = \sum_{i=1}^{L} \left(i - \mu_T\right)^2 p_i \tag{4.7}$$

where

$$\pi_0 = \sum_{i=1}^{k} p_i \qquad \pi_1 = \sum_{i=k+1}^{L} p_i = 1 - \pi_0 \tag{4.8}$$

$$\mu_0 = \sum_{i=1}^{k} ip_i/\pi_0 \qquad \mu_1 = \sum_{i=k+1}^{L} ip_i/\pi_1 \qquad \mu_T = \sum_{i=1}^{L} ip_i \tag{4.9}$$

Making use of the latter definitions, the formula for the between-class variance can be simplified to:

$$\sigma_B^2 = \pi_0\pi_1\left(\mu_1 - \mu_0\right)^2 \tag{4.10}$$

For a single threshold, the criterion to be maximized is the ratio of the between-class variance to the total variance:

$$\eta = \frac{\sigma_B^2}{\sigma_T^2} \tag{4.11}$$

However, the total variance is constant for a given image histogram, so maximizing η simplifies to maximizing the between-class variance.

The method can readily be extended to the dual threshold case $1 \leq k_1 \leq k_2 \leq L$, where the resultant classes, C_0, C_1, and C_2, have respective gray-level ranges of $[1, ..., k_1]$, $[k_1 + 1, \ldots, k_2]$, and $[k_2 + 1, \ldots, L]$.

In some situations (e.g., Hannah et al., 1995), this approach is still not sensitive enough to cope with histogram noise, and more sophisticated methods must be used. One such technique is that of entropy-based thresholding, which has become firmly embedded in the subject (Pun, 1980; Kapur et al., 1985; Abutaleb, 1989; Brink, 1992). For further insight into the performance of the between-class variance method (BCVM), see Section 4.7.

4.5.2 Entropy-Based Thresholding

Entropy measures of thresholding are based on the concept of entropy. The entropy statistic is high if a variable is well distributed over the available range, and low if it is well ordered and narrowly distributed: specifically, entropy is a measure of disorder, and is zero for a perfectly ordered system. The concept of entropy thresholding is to threshold at an intensity for which the sum of the entropies of the two intensity probability distributions thereby separated is maximized. The reason for this is to obtain the greatest reduction in entropy—i.e., the greatest increase in order—by applying the threshold: in other words, the most appropriate threshold level is the one that imposes the greatest order on the system, and thus leads to the most meaningful result.

To proceed, the intensity probability distribution is again divided into two classes—those with gray levels up to the threshold value k and those with gray levels above k (Kapur et al., 1985). This leads to two probability distributions A and B:

$$\text{A:} \quad \frac{p_1}{P_k}, \frac{p_2}{P_k}, \ldots, \frac{p_k}{P_k} \tag{4.12}$$

$$\text{B:} \quad \frac{p_{k+1}}{1 - P_k}, \frac{p_{k+2}}{1 - P_k}, \ldots, \frac{p_L}{1 - P_k} \tag{4.13}$$

where

$$P_k = \sum_{i=1}^{k} p_i \qquad 1 - P_k = \sum_{i=k+1}^{L} p_i \tag{4.14}$$

The entropies for each class are given by:

$$H(A) = -\sum_{i=1}^{k} \frac{p_i}{P_k} \ln \frac{p_i}{P_k} \tag{4.15}$$

$$H(B) = -\sum_{i=k+1}^{L} \frac{p_i}{1-P_k} \ln \frac{p_i}{1-P_k} \tag{4.16}$$

and the total entropy is:

$$H(k) = H(A) + H(B) \tag{4.17}$$

Substitution leads to the final formula:

$$H(k) = \ln\left(\sum_{i=1}^{k} p_i\right) + \ln\left(\sum_{i=k+1}^{L} p_i\right) - \frac{\sum_{i=1}^{k} p_i \ln p_i}{\sum_{i=1}^{k} p_i} - \frac{\sum_{i=k+1}^{L} p_i \ln p_i}{\sum_{i=k+1}^{L} p_i} \tag{4.18}$$

and it is this parameter that has to be maximized.

This approach can give very good results—see, e.g., Hannah et al. (1995). Again, it is straightforwardly extended to dual thresholds, but we shall not go into the details here (Kapur et al., 1985). In fact, probabilistic analysis to find mathematically ideal dual thresholds may not be the best approach in practical situations: an alternative technique for determining dual thresholds sequentially has been devised by Hannah et al. (1995), and applied to an X-ray inspection task—as described in Chapter 20.

4.5.3 Maximum Likelihood Thresholding

When dealing with distributions such as intensity histograms, it is important to compare the actual data with the data that might be expected from a previously constructed model based on a training set: this is in agreement with the methods of statistical pattern recognition (see Chapter 24), which takes full account of prior probabilities. For this purpose, one option is to model the training set data using a known distribution function such as a Gaussian. The latter has many advantages, including its accessibility to relatively straightforward mathematical analysis. In addition, it is specifiable in terms of two well-known parameters—the mean and standard deviation—which are easily measured in practical situations. Indeed, for any Gaussian distribution, we have:

$$p_i(x) = \frac{1}{(2\pi\sigma_i^2)^{1/2}} \exp\left[-\frac{(x-\mu_i)^2}{2\sigma_i^2}\right] \tag{4.19}$$

where the suffix i refers to a specific distribution, and of course when thresholding is being carried out, there is a supposition that two such distributions are involved. Applying the respective *a priori* class probabilities P_1, P_2 (Chapter 24), careful analysis (Gonzalez and Woods, 1992) shows that the condition $p_1(x) = p_2(x)$ reduces to the form:

$$x^2\left(\frac{1}{\sigma_1^2} - \frac{1}{\sigma_2^2}\right) - 2x\left(\frac{\mu_1}{\sigma_1^2} - \frac{\mu_2}{\sigma_2^2}\right) + \left(\frac{\mu_1^2}{\sigma_1^2} - \frac{\mu_2^2}{\sigma_2^2}\right) + 2\log\left(\frac{P_2\sigma_1}{P_1\sigma_2}\right) = 0 \tag{4.20}$$

Note that, in general, this equation has two solutions,[2] implying the need for two thresholds, although when $\sigma_1 = \sigma_2$ there is a single solution:

$$x = \frac{1}{2}(\mu_1 + \mu_2) + \frac{\sigma^2}{\mu_1 - \mu_2}\ln\left(\frac{P_2}{P_1}\right) \tag{4.21}$$

In addition, when the prior probabilities for the two classes are equal, the equation reduces to the altogether simpler and more obvious form:

$$x = \frac{1}{2}(\mu_1 + \mu_2) \tag{4.22}$$

Of all the methods described in this chapter, only the maximum likelihood method makes use of *a priori* probabilities. While this makes it look as if it is the only rigorous method, and indeed that all other methods are automatically erroneous and biased in their estimations, this is not the actual position. The reason lies in the fact that the other methods incorporate actual frequencies of sample data, which embody the *a priori* probabilities (see Section 24.4). Hence, the other methods should give correct results. Nevertheless, it is refreshing to see *a priori* probabilities brought in explicitly, as this gives a greater confidence of getting unbiased results in any doubtful situations.

4.6 THE GLOBAL VALLEY APPROACH TO THRESHOLDING

An important disadvantage of the many approaches to threshold estimation, including particularly entropy thresholding and its variants, is that it is often unclear how they will react to unusual or demanding situations, such as where multiple thresholds have to be found in the same image (Kapur et al., 1985;

[2]The reason for the existence of two solutions is that one solution represents a threshold in the area of overlap between the two Gaussians; the other solution is mathematically unavoidable, and lies at either very high or very low intensities. It is this latter solution that disappears when the two Gaussians have equal variance, as the distributions clearly never cross again. In any case, it seems unlikely that the distributions being modeled would in practice approximate so well to Gaussians that the non-central solution could ever be important—i.e. it is essentially a mathematical fiction that needs to be eliminated from consideration.

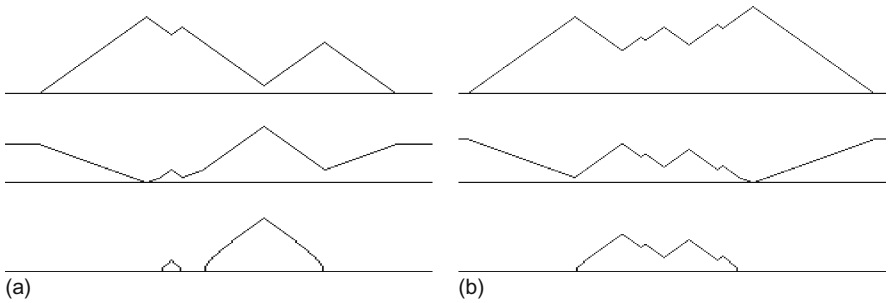


FIGURE 4.8

Result of applying global minimization algorithm to 1-D data sets. (a) A basic two-peak structure. (b) A basic multimode structure. Top trace: original 1-D data sets. Middle trace: results from Eq. (4.23). Bottom trace: results from Eq. (4.24).

Source: © IET 2008

Hannah et al., 1995; Tao et al., 2003; Wang and Bai, 2003; Sezgin and Sankur, 2004). Added to this, there is the risk that the more complex approaches will miss important aspects of the original data. The global valley approach (Davies, 2007a) aimed to provide a rigorous means of going back to basics to find global valleys of intensity histograms in such a way as to embody the intrinsic meaning of the data.

The top trace of Fig. 4.8(a) shows the basic situation—where thresholding is effective and the optimum threshold should be simple to locate. However, the intensity histogram often contains such a welter of peaks and valleys that even the human eye, with its huge capability for analysis "at a glance," can be confused—especially when it is necessary to identify global valley positions rather than local minima of lesser significance. The situation is made clearer by the example shown in the top trace of Fig. 4.8(b). Here valley 1 (numbering from the left) is lower than valley 3, but valley 3 is deeper in the sense that it has two high peaks immediately around it; however, valley 1 also lies between the highest two peaks, and in that sense it is the *globally* deepest valley in the distribution.

Clearly, to judge global valley deepness, we need a mathematical criterion so that comparisons between all the valleys can be carried out unambiguously. To proceed, for any potential global valley point (call it point j), we need to look at all points (i) on the left of it to find the highest peak to the left and all points (k) on the right of it to find the highest peak to the right, before we can construct a suitable criterion value for point j. Hence, we need to take the maximum over all points i and the maximum over all points k. Furthermore, we need to do this for all points j, and for each of them we need to consider only points i ($i < j$) and points k ($k > j$), and take account of the corresponding heights h_i, h_j, h_k in the distribution. The maximum must then be taken for a criterion function C_j of general

form $\max_{i,k} \{Q(h_i - h_j, h_k - h_j)\}$. An obvious criterion function of this form employs the arithmetic mean. However, to avoid complications from negative heights, we introduce a sign function $s(\cdot)$ such that $s(u) = u$ if $u > 0$ and $s(u) = 0$ if $u \leq 0$. The result is the following function:

$$F_j = \max_{i,k} \left\{ \tfrac{1}{2} \left[s(h_i - h_j) + s(h_k - h_j) \right] \right\} \tag{4.23}$$

When this is applied to the top trace of Fig. 4.8(a), the result is a distribution (middle trace of Fig. 4.8(a)) that has a maximum at the required valley position. In addition, the values of i and k corresponding to this maximum are the first and third peak positions in the original intensity distribution. The sign function $s(\cdot)$ has the effect of preventing negative responses that would complicate the situation unnecessarily.

While the function F used above is straightforward to apply and employs linear expressions that are often attractive in permitting in-depth analysis, it results in pedestals at either end of the output distribution: these could complicate the situation when there are many peaks and valleys. Fortunately, the geometric mean is not subject to this problem, and so it is the one that is adopted in the global valley method (GVM). Thus, we use the following function instead of F_j:

$$K_j = \max_{i,k} \left\{ \left[s(h_i - h_j) s(h_k - h_j) \right]^{1/2} \right\} \tag{4.24}$$

Note that the arithmetic and geometric means are very similar when the two arguments are nearly equal, but deviate a lot when the two arguments are dissimilar: it is the dissimilar case that applies at the ends of the distribution, where it is required to suppress a potential valley that has only one peak near to it, and the geometric mean then offers a sound advantage over the arithmetic mean. These ideas are further made clear in Fig. 4.8(b).

Overall, the rationale for this approach is that we are looking for the most significant valley in an intensity distribution, corresponding to an optimum discriminating point between, for example, dark objects and light background in the original image. While in some cases the situation is obvious (Fig. 4.8(a)), in general it is difficult to sort out a confusing set of peaks and valleys and in particular to identify global valleys. So the concept embodied in Eq. (4.24) is that of aiming to guarantee an optimal global solution by automatic means. Clearly, by analysis of the output distribution, it is also possible to find a whole range of maxima corresponding to global valley positions in the input distribution: to this extent, the method is able to cope with multimode distributions and to find multiple threshold positions.

With all histogramming methods, it is necessary to take due account of local noise in the distribution, as it could lead to inaccurate results. Hence, the K distribution is smoothed before proceeding with further analysis to locate thresholds.

Another important factor is the amount of computation required for this approach. While it at first appears that a computationally intensive scan over all possible sets of sampling points i, j, k is required to obtain the optimal solution, it turns out that with care the computational load can be reduced from $O(N^3)$ to $O(N)$, where N is the number of gray levels in the intensity distribution.

4.7 PRACTICAL RESULTS OBTAINED USING THE GLOBAL VALLEY METHOD

The ideas presented in Section 4.6 are next tested using Fig. 4.9(a). Starting with this image, the following sequence of operations is applied: (a) an intensity histogram is generated (top trace in Fig. 4.9(d)); (b) the function K is applied (middle trace in Fig. 4.9(d)); (c) the output distribution is smoothed (bottom trace in Fig. 4.9(d)); (d) peaks are located (see the short vertical lines at the bottom of Fig. 4.9(d)); (e) the most significant peaks are chosen as threshold levels (here all eight are selected); (f) a new image is generated by applying the mean of the adjacent threshold intensity levels. The result (Fig. 4.9(b)) is a reasonably segmented likeness of the original image, albeit with clear limitations in the cloud regions—simply because accurate renditions of these would require a rather full range of gray levels, and thresholding is not appropriate in such regions. However, what is significant is the ease with which the approach automatically incorporates multilevel thresholding of multimode intensity distributions—a point that has been a difficulty with entropy thresholding, for example (Hannah et al., 1995). Finally, Fig. 4.9(c) gives a comparison with the maximum BCVM of Otsu (1979), which has recently undergone something of a resurgence of popularity and use, partly as a result of the ease with which it can be used for the systematic generation of multilevel thresholds (Liao et al., 2001; Otsu, 1979).

The reconstructability of the method (in the sense that much of the image is reconstructed so well that it is difficult to distinguish from the original) is an indication of success in that it is clear that the information removed was by no means arbitrary, but was actually redundant and unhelpful. This property is also evident in Fig. 4.10, which shows the application of the method to the well-known Lena image.

The basic criterion used for smoothing is that of reducing noise as far as possible without eliminating relevant thresholding points. To achieve this, repeated convolutions of the K distributions are made with a three-element $\frac{1}{4}[1\ 2\ 1]$ kernel until an appropriate amount of smoothing is obtained. Note that the GVM peaks are by no means static. In particular, as smoothing progresses, they gradually move and then merge, as can be seen in the bottom traces in Fig. 4.10(f–h). Just before merger, there is often a rapid movement to align the merging peaks. To cope with this and to find suitable thresholding levels, a useful heuristic was to move one quarter of the way from the merged position to the next merger position

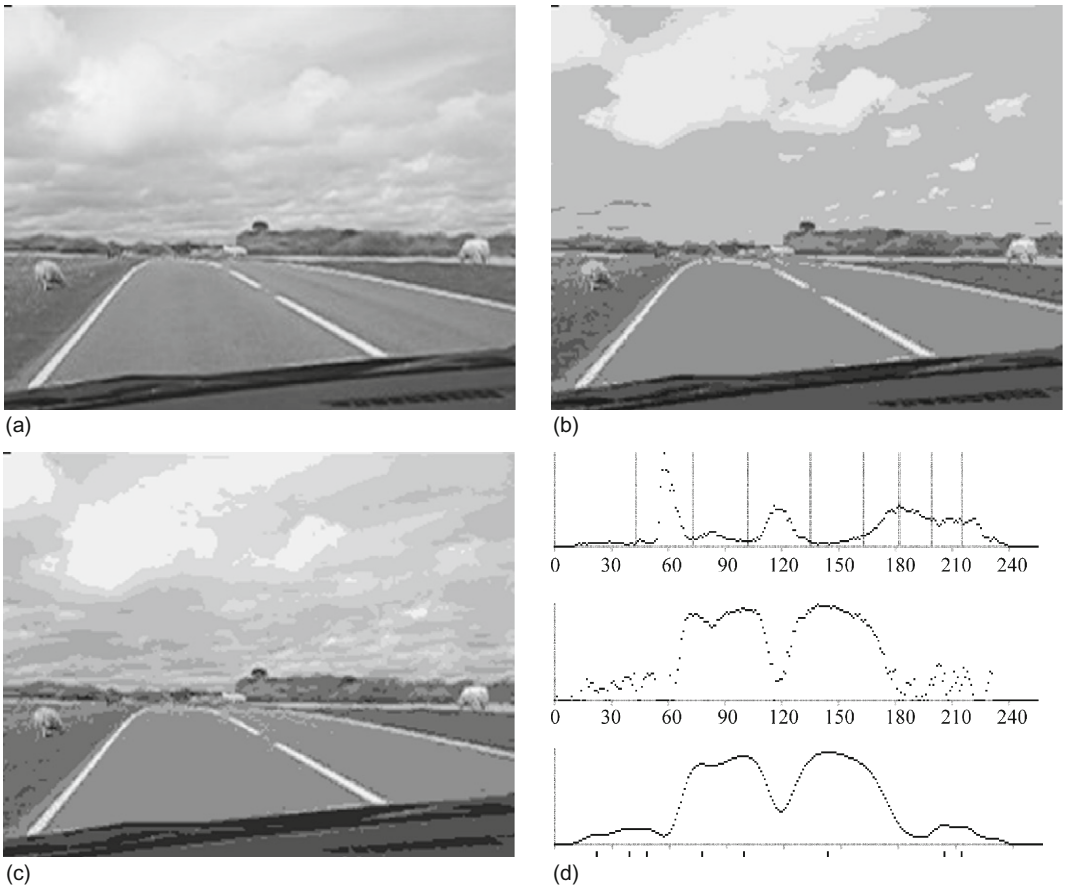


FIGURE 4.9

Result of applying the global valley algorithm to a multimode intensity distribution. (a) Original grayscale image. (b) Reconstituted image after multiple thresholding using the eight peaks in the output distribution. (d) Top: original intensity histogram for (a). Middle: result of applying the global valley transformation. Bottom: result of smoothing. The eight short vertical lines at the very bottom indicate the peak positions. In (d), the intensity scale is 0–255; the vertical scale is normalized to a maximum height indicated by the height of the vertical axis. *Note:* the three traces are computed 25 times more accurately than the rounded values displayed, so the peak locations *are* determined as accurately as indicated. For comparison, (c) shows the result of applying the between-class variance method to the same image: the eight thresholds are indicated by vertical lines in the top trace of (d).

FIGURE 4.10

Multilevel thresholding of the Lena image. For the original Lena grayscale image, see 'Miscellaneous' at the USC-SIPI Image Database*. (a) Result of applying the between-class variance method (BCVM) to original image. (b)–(d) Results of applying the global valley method to original image, producing, respectively, bi-level, tri-level, and five-level images. (e) Top: intensity histogram of original image: the vertical line indicates the bi-level threshold selected by the BCVM. Bottom: the resulting *K* distribution. (f)–(h) The upper traces show smoothed versions of the *K* distribution, with short vertical lines indicating, respectively, one, two, or four threshold positions; the lower traces show threshold positions resulting from progressive smoothing of the *K* distribution: note that these are scaled and some are truncated as indicated by the horizontal gray line at the top; the horizontal dotted lines show how sets of threshold values are selected automatically (see text).

*http://sipi.usc.edu/database/database.php (website accessed 13 December 2011).

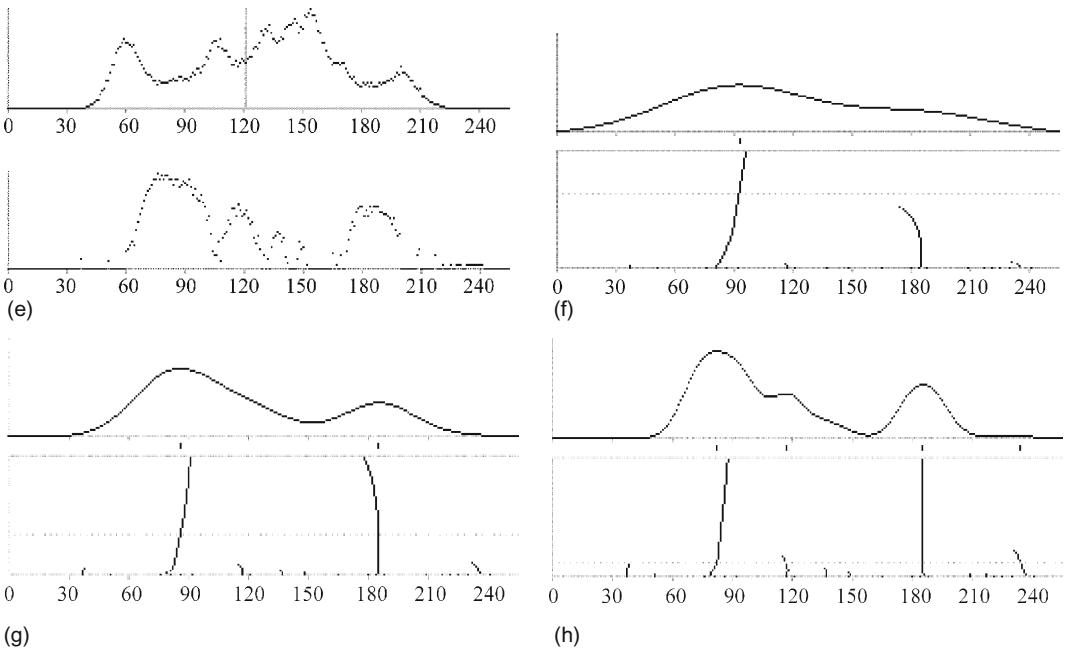


FIGURE 4.10

(Continued)

(see horizontal dotted lines in Fig. 4.10(f–h)). To clarify the process, the basic GVM algorithm is given in Fig. 4.11.

Figure 4.10(f–h) gives three examples of smoothing until 1, 2 or 4 thresholding points are produced (these give bi-level, tri-level, and five-level thresholding).[3] These lead to the images shown in Fig. 4.10(b–d) (note particularly that the light shaded region on Lena's nose is very stable and noise-free).

We concentrate next on a specific advantage of the GVM: that it produces robust judgments of minority intensities at the ends of the intensity range. Effectively, it amplifies such regions of the distribution and provides highly stable image segmentations: see, in particular, the under-vehicle shadows located in Fig. 23.1(d) and the ergot contaminant located in Fig. 21.2(d).[4] That the

[3]There is a potential confusion here: as smoothing proceeds, the number of GVM thresholds progressively *decreases*. Hence the ordering of the respective images and traces in subfigures (f)–(h) appears inverted from this point of view. However, it is the logical order for the BCVM for which computation increases approximately exponentially with the number of thresholds.

[4]Note that these represent important vehicle guidance and inspection tasks: (1) use of under-vehicle shadows is a promising technique for locating vehicles on the road ahead (Liu et al., 2007); (2) ergot is poisonous and it is important to locate it amongst wheat or other grains that are to be used for human consumption (Davies, 2003b).

```
scan = 0;
do {
  numberofpeaks = 0;
   for (all intensity values in distribution) {
      if (peak found) {
         peakposition[scan, numberofpeaks] = intensity;
         numberofpeaks ++;
      }
   }
   if (numberofpeaks == requirednumber) {
      if (previousnumberofpeaks > numberofpeaks) lowestscan = scan;
      else highestscan = scan;
   }
   previousnumberofpeaks = numberofpeaks;
   apply incremental smoothing kernel to distribution;
   scan ++;
} while (numberofpeaks > 0);

optimumscan = (lowestscan*3 + highestscan)/4;
for (all peaks up to requirednumber)
   bestpeakposition[peak] = peakposition[optimumscan, peak];
```

FIGURE 4.11

Basic global valley algorithm. This version of the algorithm assumes that the required number of peaks (*required number*) is known in advance, although the optimum amount of smoothing is unknown. Here, the latter is estimated by taking a weighted mean of the lowest and highest numbers of smoothing scans that yield the required number of peaks. The final line of the algorithm gives the required number of peaks in the best positions. While this form of the algorithm obtains positions for a specific required number of peaks, the underlying process also maps out a complete set of stability graphs because it proceeds until the number of peaks is zero. For further details, see Section 4.7.

GVM is able to make sense of the exceptionally noisy K distribution shown in Fig. 21.2(d) seems rather remarkable.

Comparing the GVM results with those of the BCVM (see Fig. 4.10(a, e)), we see that the bi-level BCVM threshold appears to lie in an *a priori* quite reasonable position in the intensity histogram: however, closer examination shows that the performance of the BCVM approximates to splitting the active area of the histogram into equal parts, corresponding to finding an approximate median. This means that for nearly unimodal histograms, it has much less chance of leading to optimal segmentations. This view of its operation is supported by tests (Fig. 4.12) made on idealized histograms, which show that it is unable to locate the bottom of the valley. It is also noteworthy that, unlike the GVM, the multilevel BCVM sometimes misses thresholds at the ends of the range of intensities (see, e.g., the vertical lines in the top trace of Fig. 4.9(d)).

Overall, it has been found that the GVM produces significantly more stable thresholds than the BCVM, that it is less prone to producing noisy boundaries in the thresholded images, and that its results tend to be more meaningful.

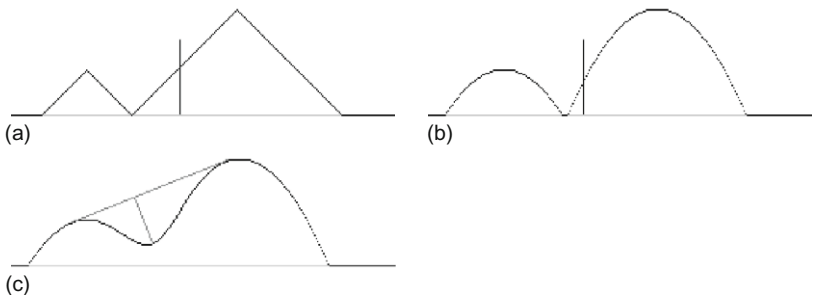


FIGURE 4.12

Results of applying the between-class variance method (BCVM) and concavity analysis in idealized cases. Applying the BCVM to (a) a triangular histogram and (b) a parabolic histogram. The vertical lines indicate the bi-level threshold selected by the BCVM: note that in each case it lies well away from the obvious global minimum of the histogram. (c) Finding thresholds by concavity analysis. The technique forms the convex hull of the distribution, takes each joining line, and uses the foot of the longest normal as an indicator of the threshold position. This approach is often highly effective but tends to give a result closer to the main peak than the optimum minimum location.

Source: © IET 2008

In fact, the BCVM tends to split intensity distributions rather blindly into approximately equal areas: although its mathematical formulation does not explicitly aim at this, it often seems to have essentially this effect.

4.8 HISTOGRAM CONCAVITY ANALYSIS

In this section, we briefly consider previous work on histogram concavity analysis. Rosin (2001) described how a simple geometrical construction (Fig. 4.12(c)) could be used to identify a suitable bi-level threshold. The technique depends on the histogram having a "corner," which is then easily identified, but when the corner is less well defined, bias can creep in and it becomes necessary to model the histogram distributions to obtain systematic corrections to the thresholding point. The approach will work for true unimodal distributions (including those produced by grey-scale edge images) or for "nearly" unimodal distributions where there is a very weak mode in addition to the main mode. For true unimodal distributions, the GVM will not work because one of the component signals in function K is zero: in such cases, it is imperative to use a method such as that described by Rosin—although others have been described over a long period—by Rosenfeld and de la Torre (1983), Tsai (1995), and others. For nearly unimodal distributions, the Rosin approach gives some intrinsic bias, as indicated in Fig. 4.12(c): but it is presumably possible in many applications to perform modeling to overcome this problem. However, the need for modeling does not seem to arise with the GVM—as has already been demonstrated (see particularly Figs. 21.2 and 23.1).

4.9 CONCLUDING REMARKS

Sections 4.3 and 4.4 have revealed a number of factors that are crucial to the process of thresholding. First, the need to avoid bias in threshold selection by arranging roughly equal populations in the dark and light regions of the intensity histogram; second, the need to work with a small subimage (or neighborhood) size so that the intensity histogram has a well-defined valley despite variations in illumination; and third, the need for subimages to be sufficiently large so that statistics are reliable, permitting the valley to be located accurately.

Unfortunately, these conditions are not compatible and compromises are needed in practical situations. In particular, it is generally not possible to find a neighborhood size that can be applied everywhere in an image, on all occasions yielding roughly equal populations of dark and light pixels. Indeed, if the chosen size is small enough to span edges ideally, hence yielding unbiased local thresholds, it will be valueless inside large objects. Attempting to avoid this situation by resorting to alternative methods of threshold calculation does not solve the problem since inherent to such methods is a built-in region size. It is therefore not surprising that a number of workers have opted for variable resolution and hierarchical techniques in an attempt to make thresholding more effective (Wu et al., 1982; Wermser et al., 1984; Kittler et al., 1985).

At this stage, we call into question the complications involved in such thresholding procedures—which become even worse when intensity distributions start to become multimodal. Note that the overall procedure is to find local intensity gradients in order to obtain accurate, unbiased estimates of thresholds so that it then becomes possible to take a horizontal slice through a grayscale image and hence, ultimately, find "vertical" (i.e., spatial) boundaries within the image. Why not use the gradients *directly* to estimate the boundary positions? Such an approach, for example, leads to no problems from large regions where intensity histograms are essentially unimodal, although it would be foolish to pretend that there are no other problems (see Chapters 5 and 10).

On the whole, the author takes the view that many approaches (region-growing, thresholding, edge detection, etc.), *taken to the limits of approximation*, will give equally good results. After all, they are all limited by the same physical effects—image noise, variability of lighting, presence of shadows, etc. However, some methods are easier to coax into working well, or need minimal computation, or have other useful properties such as robustness. Thus, thresholding can be a highly efficient means of aiding the interpretation of certain types of image: but as soon as image complexity rises above a certain critical level, it suddenly becomes more effective and considerably less complicated to rely on edge detection. This is studied in the next chapter. Meanwhile, we must not overlook the possibility of easing the thresholding task by optimizing the lighting system and ensuring that any worktable or conveyor is kept clean and white: this turns out to be a viable approach in a surprisingly large number of industrial applications.

The end result of thresholding is a set of silhouettes representing the shapes of objects: these constitute a "binarized" version of the original image. Many techniques exist for performing binary shape analysis, and some of these are described in Chapter 9. Meanwhile, note that many features of the original scene—e.g., texture, grooves or other surface structure—will not be present in the binarized image. Although the use of multiple thresholds to generate a number of binarized versions of the original image can preserve relevant information present in the original image, this approach tends to be clumsy and impracticable, and sooner or later one may be forced to return to the original grayscale image for the required data.

Thresholding is among the simplest of image processing operations and is an intrinsically appealing way of performing segmentation. While the approach is clearly limited, it would be a mistake to ignore it and its recent developments, which provide useful tools for the programmer's toolkit.

4.10 BIBLIOGRAPHICAL AND HISTORICAL NOTES

Segmentation by thresholding started many years ago from simple beginnings, and in recent years has been refined into a set of mature procedures. Among the notable early methods is the paradigm but computation-intensive Chow and Kaneko method (1972), which has been outlined in Section 4.4.1. Nakagawa and Rosenfeld (1979) studied the method and developed it for cases of trimodal distributions but without improving computational load.

Fu and Mui (1981) provided a useful general survey on image segmentation: which was updated by Haralick and Shapiro (1985). These papers review many topics that could not be covered in this chapter due to space reasons—which also applies for Sahoo et al.'s (1988) valuable survey of thresholding techniques. Nevertheless, it is worth emphasizing the point made by Fu and Mui (1981) that "All the region extraction techniques process the pictures in an iterative manner and usually involve a great expenditure in computation time and memory."

As hinted in Section 4.4, thresholding (particularly local adaptive thresholding) has had many applications in optical character recognition. Among the earliest were the algorithms described by Bartz (1968) and Ullmann (1974): also two highly effective algorithms have been described by White and Rohrer (1983).

During the 1980s, the entropy approach to automatic thresholding evolved (e.g., Pun, 1981; Kapur et al., 1985; Abutaleb, 1989; Pal and Pal, 1989): this approach (Section 4.5.2) proved highly effective, and its development continued during the 1990s (e.g., Hannah et al., 1995).

In the 2000s, the entropy approach to threshold selection has remained important, in respect both of conventional region location and ascertaining the transition

region between objects and background to make the segmentation process more reliable (Yan et al., 2003). In one instance, it was found useful to employ fuzzy entropy and genetic algorithms (Tao et al., 2003). Wang and Bai (2003) have shown how threshold selection may be made more reliable by clustering the intensities of boundary pixels, while ensuring that a continuous rather than a discrete boundary is considered (the problem is that in images that approximate to binary images over restricted regions, the edge points will lie preferentially in the object or the background, not neatly between both). However, in complex outdoor scenes and for many medical images such as brain scans, thresholding alone will not be sufficient, and resort may even have to be made to graph matching (Chapter 14) to produce the best results—reflecting the important fact that segmentation is necessarily a high-level rather than a low-level process (Wang and Siskind, 2003). In rather less demanding cases, deformable model-guided split-and-merge techniques may, on the other hand, still be sufficient (Liu and Sclaroff, 2004).

4.10.1 More Recent Developments

Sezgin and Sankur (2004) give a thorough review and assessment of work on thresholding prior to 2004. More recently, there has been continued interest in thresholding in the case of unimodal (Coudray et al., 2010; Medina-Carnicer et al., 2011) and near-unimodal histograms (Davies, 2007a, 2008b): the latter case is covered fairly fully in Sections 4.6 and 4.7. In the case of Coudray et al. (2010), the aim is to threshold intensity gradient histograms in order to locate edges reliably: the approach taken is to model the contribution from noise as a Rayleigh distribution and then to devise heuristics for analyzing the overall distribution. With the same aim, Medina-Carnicer et al. (2011) show that applying a histogram transformation improves the performance of the Otsu (1979) and Rosin (2001) methods. Li et al. (2011) adopt the novel approach of constraining the gray-level ranges considered by the thresholding algorithm in such a way as to weaken gray-level changes in both foreground and background, thus simplifying the original image and making the intensity histogram more closely bimodal. After that several thresholding methods are found to operate more reliably. Ng (2006) describes a revised version of the Otsu (1979) method that operates well for unimodal distributions, and which is useful for defect detection. This "valley emphasis" method works by applying a weight to the Otsu threshold calculation. Overall, several of the recent developments can be construed as applying transformations or other improvements to older methods to make them more sophisticated and accurate: in fact none is highly complex in any theoretical way. Finally, it may seem somewhat surprising that, after so many decades, thresholding is still something of a "hot" subject: the driving force for this is its extreme simplicity and high level of utility.

4.11 PROBLEMS

1. Using the ideas outlined in Section 4.3.2, model the intensity distribution obtained by finding all the edge pixels in an image and including also all pixels adjacent to these pixels. Show that while this gives a sharper valley than for the original intensity distribution, it is not as sharp as for pixels located by the Laplacian operator.
2. Consider whether it is more accurate to estimate a suitable threshold for a bimodal, dual-Gaussian distribution by (a) finding the position of the minimum, or (b) finding the mean of the two peak positions. What corrections could be made by taking account of the magnitudes of the peaks?
3. Obtain a complete derivation of Eq. (4.20). Show that, in general (as stated in Section 4.5.3), it has two solutions. What is the physical reason for this? How can it have only one solution when $\sigma_1 = \sigma_2$?
4. Prove the statement made in Section 4.6 that the computational load of the histogram analysis for the global value method can be reduced from $O(N^3)$ to $O(N)$. Show also that the number of passes over the histogram required to achieve this is at most 2.