

10 Boundary Pattern Analysis

Recognition of binary objects by boundary pattern analysis should be a straightforward process, but this chapter shows that there are a number of problems to be overcome. In particular, any boundary distortions such as those due to breakage or several objects being in contact may result in total failure of the matching process. This chapter discusses the problems and their solution.

Look out for:

- the centroidal profile approach and its limitations.
- how the method may be speeded up.
- how recognition based on the (s, ψ) boundary plot is significantly more robust.
- how the (s, ψ) plot leads on to the more convenient (s, κ) plot.
- the relation between κ and ψ .
- more rigorous ways of dealing with the occlusion problem.
- discussion of the accuracy of boundary length measures.

Disparate ways are available for representing object boundaries and for measuring and recognizing objects using them. All the methods are subject to the same ultimate difficulties—particularly that of managing occlusion (which necessarily removes relevant data), and that of inaccuracy in the pixel-based description. Sound ways of managing occlusion are indicated in [Section 10.6](#). This work presages later chapters where methods such as the Hough transform are widely employed for robust object recognition.

10.1 INTRODUCTION

Chapter 4 has shown how thresholding may be used to binarize grayscale images and hence to present objects as 2-D shapes. However, that method of segmenting objects is successful only when considerable care is taken with lighting and when the object can be presented conveniently, e.g., as dark patches on a light

background. In other cases, adaptive thresholding may help to achieve the same end: as an alternative, edge detection schemes can be applied, which are generally rather more resistant to problems of lighting. Nevertheless, thresholding of edge-enhanced images still gives certain problems: in particular, edges may peter out in some places and may be thick in others (Fig. 10.1). For many purposes, the output of an edge detector is ideally a connected unit-width line around the periphery of an object and steps need to be taken to convert edges to this form—if this has not already been achieved using a Canny or other operator employing nonmaximum suppression (see Chapter 5).

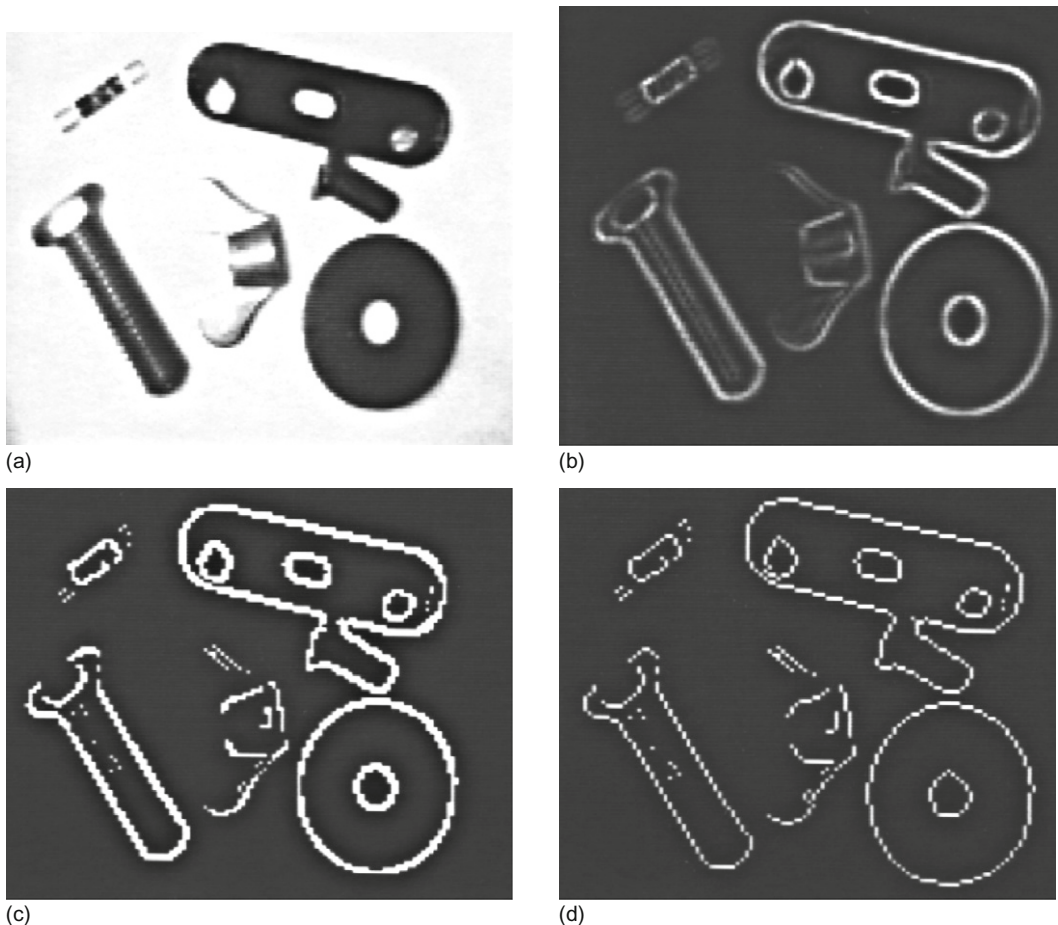


FIGURE 10.1

Some problems with edges. The edge-enhanced image (b) from an original image (a) is thresholded as in (c). The edges so detected are found to peter out in some places and to be thick in other places. A thinning algorithm is able to reduce the edges to unit thickness (d) but *ad hoc* (i.e., not model-based) linking algorithms are liable to produce erroneous results (not shown).

Thinning algorithms can be used to reduce edges to unit thickness while maintaining connectedness (Fig. 10.1(d)). Many algorithms have been developed for this purpose, and the main problems here are: (a) slight bias and inaccuracy due to uneven stripping of pixels, especially in view of the fact that even the best algorithm can only produce a line that is locally within $1/2$ pixel of the ideal position; and (b) introduction of a certain number of noise spurs. The first problem can be minimized by using grayscale edge thinning algorithms, which act directly on the original grayscale edge-enhanced image (e.g., Paler and Kittler, 1983). Noise spurs around object boundaries can be eliminated quite efficiently by removing lines that are shorter than (say) 3 pixels. Overall, the major problem to be dealt with is that of reconnecting boundaries that have become fragmented during thresholding.

A number of rather *ad hoc* schemes are available for relinking broken boundaries: e.g., line ends may be extended along their existing directions—a *very* limited procedure since there are (at least for binary edges) only eight possible directions, and it is quite possible for the extended line ends not to meet. Another approach is to join line ends that are sufficiently close together and point in similar directions, both to each other *and* to the direction of the vector between the two ends. In fact, this approach can be made quite credible in principle, but in practice it can lead to all sorts of problems as it is still *ad hoc* and not model driven. Hence, adjacent lines that arise from genuine surface markings and shadows may be arbitrarily linked together by such algorithms. In many situations, it is therefore best if the process is model driven—e.g., by finding the best fit to some appropriate idealized boundary such as an ellipse. Yet another approach is that of relaxation labeling, which iteratively enhances the original image, progressively making decisions as to where the original gray levels reinforce each other. Thus, edge linking is permitted only where evidence is available in the original image that this is appropriate. A similar but computationally more efficient line of attack is the hysteresis thresholding method described in Chapter 5. Here intensity gradients above a certain upper threshold are taken to give definite indication of edge positions, whereas those above a second, lower threshold are taken to indicate edges only if they are adjacent to positions that have already been accepted as edges (for a more detailed analysis, see Section 5.10).

It may be thought that the Marr–Hildreth and related (Laplacian-based) edge detectors do not run into these problems, because they give edge contours that are necessarily connected. However, the result of using methods that force connectedness is that sometimes (e.g., when edges are diffuse, or of low contrast, so that image noise is an important factor) parts of a contour will lack meaning; indeed, a contour may meander over such regions following noise rather than useful object boundaries. Furthermore, it is as well to note that the problem is not merely one of pulling low-level signals out of noise, but rather that sometimes there is no signal at all present that could be enhanced to a meaningful level. Reasons for this include lighting being such as to give zero contrast (as, e.g., when a cube is lit in such a way that two faces have equal brightness) and occlusion. Lack of spatial resolution can also cause problems by merging together several lines on an object.

In what follows, it is assumed that all of these problems have been overcome by sufficient care with the lighting scheme, appropriate digitization, and other means. It is also assumed that suitable thinning and linking algorithms have been applied so that all objects are outlined by connected unit-width boundary lines. At this stage, it should be possible to locate the objects from the boundary image, and to identify and orientate them accurately.

10.2 BOUNDARY TRACKING PROCEDURES

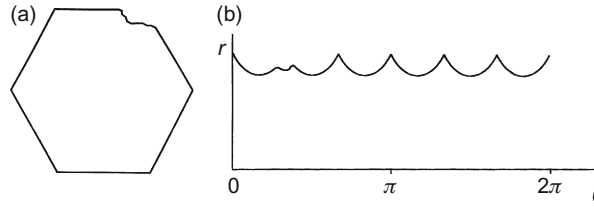
Before objects can be matched with their boundary patterns, means must be found for tracking systematically around the boundaries of all the objects in an image. Means have already been demonstrated for achieving this in the case of regions such as those that result from intensity thresholding routines (Chapter 9). However, if a connected unit-width boundary is formed by an alternative process such as edge detection, the problem of tracking is much simpler, since it is necessary only to move repeatedly to the next edge pixel. Clearly, it is necessary to ensure that we (a) never reverse direction, (b) know when we have been round the whole boundary once, and (c) record which object boundaries have been encountered. As when tracking around regions, we must ensure that in each case we end by passing through the starting point in the same direction.

10.3 CENTROIDAL PROFILES

The substantial matching problems that occur with 2-D template matching make it attractive to attempt to locate objects in a less demanding search space. In fact, it is possible to achieve this very simply by matching the boundary of each object in a single dimension. Perhaps the most obvious such scheme uses an (r, θ) plot. Here the centroid of the object is first located,¹ and then a polar coordinate system is set up relative to this point and the object boundary is plotted as an (r, θ) graph—often called a “centroidal profile” (Fig. 10.2). Next, the 1-D graph so obtained is matched against the corresponding graph for an idealized object of the same type. Since objects generally have arbitrary orientation, it is necessary to “slide” the idealized graph along that obtained from the image data until the best match is obtained. The match for each possible orientation α_j of the object is commonly tested by measuring the differences in radial distance between the boundary graph B and the template graph T for various values of θ and summing their squares to give a difference measure D_j for the quality of the fit:

$$D_j = \sum_i [r_B(\theta_i) - r_T(\theta_i + \alpha_j)]^2 \quad (10.1)$$

¹Note that the position of the centroid is deducible directly from the list of boundary pixel coordinates—there is no need to start with a region-based description of the object for this purpose.

**FIGURE 10.2**

Centroidal profiles for object recognition and scrutiny: (a) hexagonal nut shape in which one corner has been damaged; (b) centroidal profile, which permits both straightforward identification of the object and detailed scrutiny of its shape.

Alternatively, the absolute magnitudes of the differences are used:

$$D_j = \sum_i |r_B(\theta_i) - r_T(\theta_i + \alpha_j)| \quad (10.2)$$

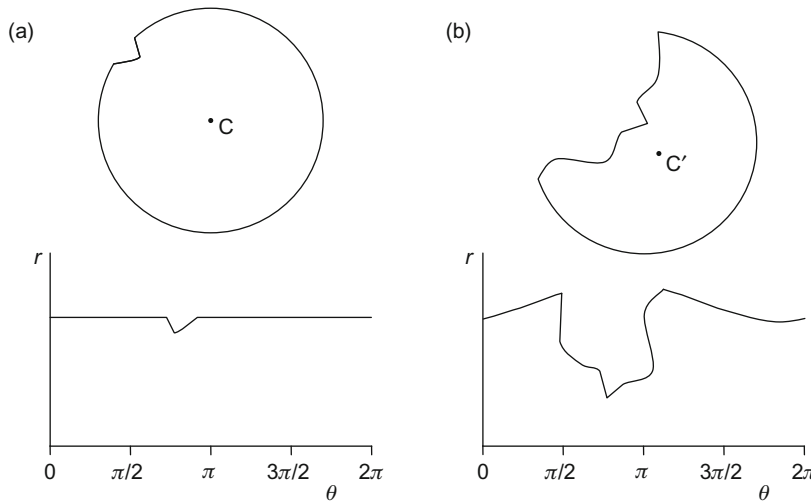
The latter measure has the advantage of being easier to compute and less biased by extreme or erroneous difference values. Note that the basic 2-D matching operation has now been reduced to 1-D, and if we need work in 1° steps, the orientation indices i and j will each have to range over 360 values. The result is that the number of operations that are required to test each object drops to around 360^2 (i.e., $\sim 100,000$), so boundary pattern analysis should give a very substantial saving in computation.

The 1-D boundary pattern matching approach described above is able to identify objects and also to find their orientations. In fact, initial location of the centroid of the object also solves one other part of the problem as specified at the end of [Section 10.1](#). At this stage, it may be noted that the matching process also leads to the possibility of inspecting the object's shape as an inherent part of the process ([Fig. 10.2](#)). In principle, this combination of capabilities makes the centroidal profile technique quite powerful.

Finally, note that the method is able to cope with objects of identical shapes but different sizes. This is achieved by using the maximum value of r to normalize the profile, giving a variation (ρ, θ) where $\rho = r/r_{\max}$.

10.4 PROBLEMS WITH THE CENTROIDAL PROFILE APPROACH

In practice, there are several problems with the procedure outlined in [Section 10.3](#). First, any major defect or occlusion of the object boundary can cause the centroid to be moved away from its true position, and the matching process will be largely spoiled ([Fig. 10.3](#)). Thus, instead of concluding that this is an object of type X with a specific part of its boundary damaged, the algorithm will most probably not recognize it at all. Such behavior would be inadequate in many

**FIGURE 10.3**

Problems with the centroidal profile descriptor. (a) A circular object with a minor defect on its boundary; its centroidal profile appears beneath it. (b) The same object, this time with a gross defect: because the centroid is shifted to C' , the *whole* of the centroidal profile is grossly distorted.

automated inspection applications, where positive identification and fault-finding are required, and the object would have to be rejected without a satisfactory diagnosis being made.

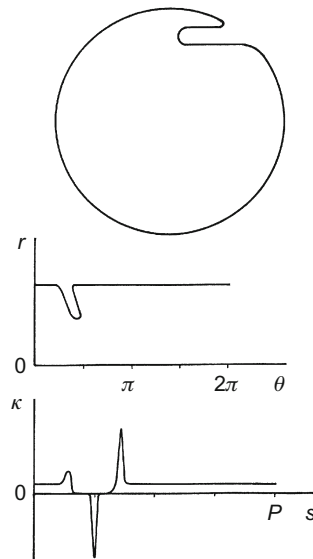
Second, the (r, θ) plot will be multivalued for a certain class of object (Fig. 10.4). This has the effect of making the matching process partly 2-D and leads to complication and excessive computation.

Third, the very variable spacing of the pixels when plotted in (r, θ) space is a source of complication. It leads to the requirement for considerable smoothing of the 1-D plots, especially in regions where the boundary comes close to the centroid—as for elongated objects such as spanners or screwdrivers (Fig. 10.5); however, in other places accuracy will be greater than necessary and the overall process will be wasteful. The problem arises because quantization should ideally be uniform along the θ -axis so that the two templates can be moved conveniently relative to one another to find the orientation of best match.

Finally, computation times can still be quite significant, so some timesaving procedure is required.

10.4.1 Some Solutions

All four of the above-described problems can be tackled in one way or another, with varying degrees of success. The first problem, that of coping with occlusions

**FIGURE 10.4**

Boundary pattern analysis via (r, θ) and (s, κ) plots.

**FIGURE 10.5**

A problem in obtaining a centroidal profile for elongated objects. This figure highlights the pixels around the boundary of an elongated object—a scanner—showing that it will be difficult to obtain an accurate centroidal profile for the region near the centroid.

and gross defects, is probably the most fundamental and the most resistant to satisfactory solution. For the method to work successfully, a stable reference point must be found within the object. The centroid is naturally a good candidate for this since the averaging inherent in its location tends to eliminate most forms of

noise or minor defect: however, major distortions such as those arising from breakages or occlusions are bound to affect it adversely. The centroid of the boundary is no better, and may also be less successful at suppressing noise. Other possible candidates are the positions of prominent features such as corners, holes, centers of arcs, and so on. In general, the smaller such a feature, the more likely it is to be missed altogether in the event of a breakage or occlusion, although the larger such a feature, the more likely it is to be affected by the defect. In fact, circular arcs can be located accurately (at their centers) even if they are partly occluded (see Chapter 12), so these features are very useful for leading to suitable reference points. A set of symmetrically placed holes may sometimes be suitable, since even if one of them is obscured, one of the others is likely to be visible and can act as a reference point.

Clearly, such features can help the method to work adequately but their presence also calls into question the value of the 1-D boundary pattern matching procedure, since they make it likely that superior methods can be used for object recognition (see later chapters in Part 2). For the present, we therefore accept that (a) severe complications arise when part of an object is missing or occluded and (b) it may be possible to provide some degree of resistance to such problems by using a prominent feature as a reference point instead of the centroid. Indeed, the only significant *further* change that is required to cope with occlusions is that difference ($r_B - r_T$) values greater than (say) 3 pixels should be ignored, and the best match then becomes one for which the greatest number of values of θ gives good agreement between B and T.

The second problem, of multivalued (r, θ) plots, is solved very simply by employing the heuristic of taking the smallest value of r for any given θ and then proceeding with matching as normal (here it is assumed that the boundaries of any holes present within the boundary of the object are dealt with separately, information about any object and its holes being collated at the end of the recognition process). This *ad hoc* procedure should in fact be acceptable when making a preliminary match of the object to its 1-D template, and may be discarded at a later stage when the orientation of the object is known accurately.

The third problem described above arises because of uneven spacing of the pixel boundaries along the θ dimension of the (r, θ) graph. To some extent this problem can be avoided by deciding in advance on the permissible values of θ and querying a list of boundary points to find which has the closest θ to each permissible value. Some local smoothing of the ordered set of boundary points can be undertaken but this is in principle unnecessary, since for a connected boundary, there will always be 1 pixel, which is closest to a line from the centroid at a given value of θ .

The two-stage approach to matching hinted in [Section 10.3](#) can also be used to help with the last of the problems mentioned above—the need to speed up the processing. First, a coarse match is obtained between the object and its 1-D template by taking θ in relatively large steps of (say) 5° and ignoring intermediate angles in both the image data and the template, and then a better match is

obtained by making fine adjustments to the orientations, obtaining a match to within 1° . In this way, the coarse match is obtained perhaps 20 times faster than the previous full match, whereas the final fine match takes a relatively short time, since very few distinct orientations have to be tested.

This two-stage process can be optimized by making a few simple calculations. The coarse match is given by increasing the θ incrementation step to $\delta\theta$, so the computational load is proportional to $(360/\delta\theta)^2$, whereas the load for the fine match is proportional to $360\delta\theta$, giving a total load of:

$$\lambda = \left(\frac{360}{\delta\theta}\right)^2 + 360\delta\theta \quad (10.3)$$

This should be compared with the original load of:

$$\lambda_0 = 360^2 \quad (10.4)$$

Hence, the load is reduced (and the algorithm speeded up) by the factor:

$$\eta = \frac{\lambda_0}{\lambda} = \frac{1}{(1/\delta\theta)^2 + (\delta\theta/360)} \quad (10.5)$$

This is a maximum for $d\eta/d\delta\theta = 0$, giving:

$$\delta\theta = \sqrt[3]{2 \times 360} \approx 9^\circ \quad (10.6)$$

In practice, this value of $\delta\theta$ is rather large and there is a risk that the coarse match will give such a poor fit that the object will not be identified: hence, values of $\delta\theta$ in the range $2-5^\circ$ are more usual (see, e.g., Berman et al., 1985). Note that the optimum value of η is 26.8 and this reduces only to 18.6 for $\delta\theta = 5^\circ$, although it goes down to 3.9 for $\delta\theta = 2^\circ$.

Another way of approaching the problem is to search the (r, θ) graph for some characteristic feature such as a sharp corner (this step constituting the coarse match), and then to perform a fine match around the object orientation so deduced. Clearly, there are possibilities of error here, in situations where objects have several similar features—as in the case of a rectangle: however, the individual trials are relatively inexpensive and so it is worth invoking this procedure if the object possesses appropriate well-defined features. Note that it is possible to use the position of the maximum value, r_{\max} , as an orientating feature, but this is frequently inappropriate because a smooth maximum gives a relatively large angular error.

10.5 THE (s, ψ) PLOT

It can be seen from the above considerations that boundary pattern analysis should usually be practicable except when problems from occlusions and gross defects can be expected. However, these latter problems do give motivation for

employing alternative methods if these can be found. In fact, the (s, ψ) graph has proved particularly popular since it is inherently better suited than the (r, θ) graph to situations where defects and occlusions may occur. In addition, it does not suffer from the multiple values encountered by the (r, θ) method.

The (s, ψ) graph does not require prior estimation of the centroid or some other reference point since it is computed directly from the boundary, in the form of a plot of the tangential orientation ψ as a function of boundary distance s . The method is not without its problems and, in particular, distance along the boundary needs to be measured accurately. The commonly used approach is to count horizontal and vertical steps as unit distance and to take diagonal steps as distance $\sqrt{2}$; in fact, this idea must be regarded as a rather *ad hoc* solution, and the situation is discussed further in Section 10.7.

When considering application of the (s, ψ) graph for object recognition, it will immediately be noticed that the graph has a ψ value that increases by 2π for each circuit of the boundary, i.e., $\psi(s)$ is not periodic in s . The result is that the graph becomes essentially 2-D, i.e., the shape has to be matched by moving the ideal object template both along the s -axis and along the ψ -axis directions. Ideally, the template could be moved diagonally along the direction of the graph. However, noise and other deviations of the actual shape relative to the ideal shape mean that in practice the match must be at least partly 2-D, hence adding to the computational load.

One way of tackling this problem is to make a comparison with the shape of a circle of the same boundary length P . Thus, an $(s, \Delta\psi)$ graph is plotted, which reflects the difference $\Delta\psi$ between the ψ expected for the shape and that expected for a circle of circumference P :

$$\Delta\psi = \psi - \left(\frac{2\pi s}{P}\right) \quad (10.7)$$

This expression helps to keep the graph 1-D, since $\Delta\psi$ automatically resets itself to its initial value after one circuit of the boundary (i.e., $\Delta\psi$ is periodic in s).

Next note that the $\Delta\psi(s)$ variation depends on the starting position where $s = 0$ and this is randomly sited on the boundary. It is useful to eliminate this dependence, and this may be achieved by subtracting from $\Delta\psi$ its mean value μ . This gives the new variable:

$$\tilde{\psi} = \psi - \left(\frac{2\pi s}{P}\right) - \mu \quad (10.8)$$

At this stage, the graph is completely 1-D and is also periodic, being similar in these respects to an (r, θ) graph. Matching should now reduce to the straightforward task of sliding the template along the $\tilde{\psi}(s)$ graph until a good fit is achieved.

At this point, there should be no problems so long as (a) the scale of the object is known and (b) occlusions or other disturbances cannot occur. Suppose next that the scale is unknown: then the perimeter P may be used to normalize the value of s . If, however, occlusions *can* occur, then no reliance can be placed on P for

normalizing s and hence the method cannot be guaranteed to work. This problem does not arise if the scale of the object is known, since a standard perimeter P_T can be assumed. However, the possibility of occlusion gives further problems, which are discussed in the [Section 10.6](#).

Another way in which the problem of nonperiodic $\psi(s)$ can be solved is by replacing ψ by its derivative $d\psi/ds$. Then the problem of constantly expanding ψ (which results in its increase by 2π after each circuit of the boundary) is eliminated—the addition of 2π to ψ does not affect $d\psi/ds$ locally, since $d(\psi + 2\pi)/ds = d\psi/ds$. Note that $d\psi/ds$ is actually the local curvature function $\kappa(s)$ (see [Fig. 10.4](#)), so the resulting graph has a simple physical interpretation. Unfortunately, this version of the method has its own problems in that κ approaches infinity at any sharp corner. For industrial components, which frequently have sharp corners, this is a genuine practical difficulty and may be tackled by approximating adjacent gradients and ensuring that κ integrates to the correct value in the region of a corner (Hall, 1979).

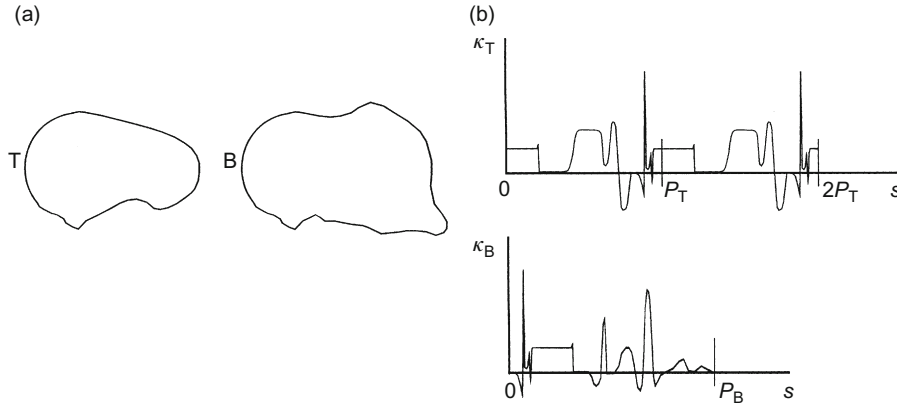
Many workers take the (s, κ) graph idea further and expand $\kappa(s)$ as a Fourier series:

$$\kappa(s) = \sum_{n=-\infty}^{\infty} c_n \exp\left(\frac{2\pi i n s}{P}\right) \quad (10.9)$$

This results in the well-known Fourier descriptor method. In this method, shapes are analyzed in terms of a series of Fourier descriptor components, which are truncated to zero after a sufficient number of terms. Unfortunately, the amount of computation involved in this approach is considerable and there is a tendency to approximate curves with relatively few terms. In industrial applications, where computations have to be performed in real time, this can generate problems, so it is often more appropriate to match to the basic (s, κ) graph. In this way, critical measurements between object features can be made with adequate accuracy in real time.

10.6 TACKLING THE PROBLEMS OF OCCLUSION

Whatever means are used for tackling the problem of continuously increasing ψ , problems still arise when occlusions occur. However, the approach is not immediately invalidated by missing sections of boundary as it is for the basic (r, θ) method. As noted in [Section 10.5](#), a major effect of occlusions is that the perimeter of the object is altered, so P can no longer be used to indicate its scale. Hence, the latter has to be known in advance: this is assumed in what follows. Another practical result of occlusions is that certain sections correspond correctly to parts of the object, whereas other sections correspond to parts of occluding objects; alternatively, they may correspond to unpredictable boundary segments where damage has occurred. Note that if the overall boundary is that of two overlapping objects, the observed perimeter P_B will be greater than the ideal perimeter P_T .

**FIGURE 10.6**

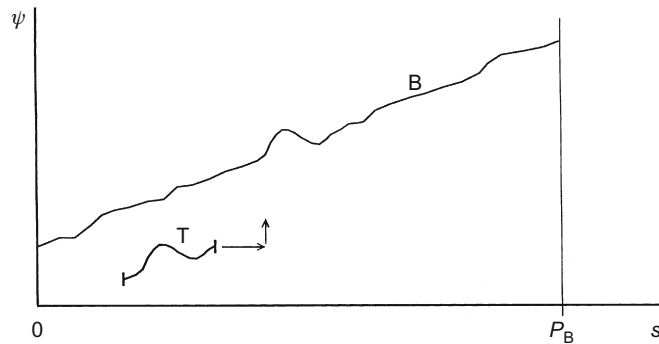
Matching a template against a distorted boundary. When a boundary B is broken (or partly occluded) but continuous, it is necessary to attempt to match between B and a template T that is doubled to length $2P_T$, to allow for T being severed at any point: (a) the basic problem and (b) matching in (s, κ) space.

Segmenting the boundary between relevant and irrelevant sections is, *a priori*, a difficult task. However, a useful strategy is to start by making positive matches wherever possible and to ignore irrelevant sections—i.e., try to match as usual, ignoring any section of the boundary that is a bad fit. We can imagine achieving a match by sliding the template T along the boundary B. However, a problem arises since T is periodic in s and should not be cut off at the ends of the range $0 \leq s \leq P_T$. As a result, it is necessary to attempt to match over a length $2P_T$. At first sight, it might be thought that the situation ought to be symmetrical between B and T. However, T is known in advance, whereas B is partly unknown, there being the possibility of one or more breaks in the ideal boundary into which foreign boundary segments have been included. Indeed, the positions of the breaks are unknown, so it is necessary to try matching the whole of T at all positions on B. Taking a length $2P_T$ in testing for a match effectively permits the required break to arise in T at any relevant position: (see Fig. 10.6).

When carrying out the match, we basically use the difference measure:

$$D_{jk} = \sum_i [\psi_B(s_i) - (\psi_T(s_i + s_k) + \alpha_j)]^2 \quad (10.10)$$

where j and k are the match parameters for orientation and boundary displacement, respectively. Note that the resulting D_{jk} is roughly proportional to the length L of the boundary over which the fit is reasonable. Unfortunately, this means that the measure D_{jk} appears to *improve* as L decreases; hence, when variable occlusions can occur, the best match must be taken as the one for which the greatest length L gives good agreement between B and T (this may be measured

**FIGURE 10.7**

Matching a short template to part of a boundary. A short template T , corresponding to part of an idealized boundary, is matched against the observed boundary B . Strictly speaking, matching in (s, ψ) space is 2-D, although there is very little uncertainty in the vertical (orientation) dimension.

as the greatest number of values of s in the sum of Eq. (10.10), which gives good agreement between B and T , i.e., the sum over all i such that the difference in square brackets is numerically less than, say, 5°).

If the boundary is occluded in more than one place, then L is at most the largest single length of unoccluded boundary (not the total length of unoccluded boundary), since the separate segments will in general be “out of phase” with the template. This is a disadvantage when trying to obtain an accurate result, since extraneous matches add noise, which degrades the fit that is obtainable—hence adding to the risk that the object will not be identified and reducing accuracy of registration. This suggests that it might be better to use only short sections of the boundary template for matching. Indeed, this strategy can be advantageous since speed is enhanced and registration accuracy can be retained or even improved by careful selection of salient features (note that nonsalient features such as smooth curved segments could have originated from many places on object boundaries and are not very helpful for identifying and accurately locating objects: hence, it is reasonable to ignore them). In this version of the method, we now have $P_T < P_B$, and it is necessary to match over a length P_T rather than $2P_T$, since T is no longer periodic (Fig. 10.7). Once various segments have been located, the boundary can be reassembled as closely as possible into its full form, and at that stage defects, occlusions, and other distortions can be recognized overtly and recorded. Reassembly of the object boundaries can be performed by techniques such as the Hough transform and relational pattern matching techniques (see Chapters 13 and 14). Work of this type has been carried out by Turney et al. (1985), who found that the salient features should be short boundary segments where corners and other distinctive “kinks” occur.

Before leaving this topic, note that $\tilde{\psi}$ can no longer be used when occlusions are present, since although the perimeter can be assumed to be known, the mean value of $\Delta\psi$ (Eq. (10.8)) cannot be deduced. Hence, the matching task reverts to

a 2-D search (although, as stated earlier, very little unrestrained search in the ψ direction need be made). However, in the case when small salient features are being sought, it is a reasonable working assumption that no occlusion occurs in any of the individual cases—a feature is either entirely present or entirely absent. Hence, the average slope $\bar{\psi}$ over T can validly be computed (Fig. 10.7) and this again reduces the search to 1-D (Turney et al., 1985).

Overall, missing sections of object boundaries necessitate a fundamental rethink as to how boundary pattern analysis should be carried out. For quite small defects the (r, θ) method is sufficiently robust but in less trivial cases it is vital to use some form of the (s, ψ) approach, while for really gross occlusions it is not particularly useful to try to match for the full boundary: rather it is better to attempt to match small salient features. This sets the scene for the Hough transform and relational pattern matching techniques of later chapters.

10.7 ACCURACY OF BOUNDARY LENGTH MEASURES

Next we examine the accuracy of the idea expressed earlier, that adjacent pixels on an 8-connected curve should be regarded as separated by 1 pixel if the vector joining them is aligned along the major axes and by $\sqrt{2}$ pixels if the vector is in a diagonal orientation. In general, this estimator overestimates the distance along the boundary. The reason for this is quite simple to see by appealing to the following pair of situations:

```

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
...
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
...

```

In either case we are considering only the top of the object. In the first example, the boundary length along the top of the object is exactly that given by the rule. However, in the second case the estimated length is increased by amount $\sqrt{2} - 1$ because of the step. Now as the length of the top of the object tends to large values, say p pixels, the actual length approximates to p , whereas the estimated length is $p + \sqrt{2} - 1$; thus, a definite error exists. Indeed, this error initially increases in importance as p decreases, since the actual length of the top of the object (when there is one step) is still:

$$L = (1 + p^2)^{1/2} \approx p \quad (10.11)$$

so the fractional error is:

$$\xi \approx \frac{\sqrt{2} - 1}{p} \quad (10.12)$$

which increases as p becomes smaller.

This result can be construed as meaning that the fractional error ξ in estimating boundary length increases initially as the boundary orientation ψ increases from zero. A similar effect occurs as the orientation decreases from 45° . Thus, the ξ variation possesses a maximum between 0° and 45° . This systematic overestimation of boundary length may be eliminated by employing an improved model in which the length per pixel is s_m along the major axis directions and s_d in diagonal directions. A complete calculation (Kulpa, 1977; see also Dorst and Smeulders, 1987) shows that:

$$s_m = 0.948 \quad (10.13)$$

and

$$s_d = 1.343 \quad (10.14)$$

It is perhaps surprising that this solution corresponds to a ratio s_d/s_m that is still equal to $\sqrt{2}$, although the arguments given above make it obvious that s_m should be less than unity.

Unfortunately, an estimator that has just two free parameters can still permit quite large errors in estimating the perimeters of individual objects. To reduce this problem, it is necessary to perform more detailed modeling of the step pattern around the boundary (Koplowitz and Bruckstein, 1989), which seems certain to increase the computational load significantly.

It is important to underline that the basis of this work is to estimate the length of the original continuous boundary rather than that of the digitized boundary: furthermore, it must be noted that the digitization process loses information, so the best that can be done is to obtain the best estimate of the original boundary length. Thus, employing the values 0.948 and 1.343 given above, rather than the values 1 and $\sqrt{2}$, reduces the estimated errors in boundary length measurement from 6.6% to 2.3%—but then only under certain assumptions about correlations between orientations at neighboring boundary pixels (Dorst and Smeulders, 1987).

10.8 CONCLUDING REMARKS

This chapter is concerned with boundary pattern analysis. The boundary patterns were imagined to arise from edge detection operations that have been processed to make them connected and of unit width. However, if intensity thresholding methods were employed for segmenting images, boundary tracking procedures would also permit the boundary pattern analysis methods of this chapter to be used. Conversely, if edge detection operations led to the production of connected boundaries, these could be filled in by suitable algorithms (which are more tricky to devise than might at first be imagined) (Ali and Burge, 1988) and converted to regions to which the binary shape analysis methods of Chapter 9 could be applied. Hence, shapes are representable in region or boundary form: if they initially arise in one representation, they may be converted to the alternate representation. This means that boundary or regional means may be employed for shape analysis, as appropriate.

An important factor here is that a positive advantage is often gained by employing boundary pattern analysis, since computation should be inherently lower (in proportion to the numbers of pixels that are required to describe the shapes in the two representations). Another important determining factor that is discussed in the present chapter is that of occlusion. If occlusions are present, then several of the methods described in Chapter 9 will operate incorrectly—as also happens for the basic centroidal profile method described in [Section 10.3](#). The (s, ψ) method then provides a good starting point. As has been seen, this is best applied to detect small salient boundary features, which can then be reassembled into whole objects by relational pattern matching techniques (see especially Chapter 14).

A variety of boundary representations is available for shape analysis. However, this chapter has shown that intuitive schemes raise fundamental robustness issues: these will only be resolved later on by forgoing deduction in favor of inference. Underlying analog shape estimation in a digital lattice is also an issue.

10.9 BIBLIOGRAPHICAL AND HISTORICAL NOTES

Many of the techniques described in this chapter have been known since the early days of image analysis. Boundary tracking has been known since 1961 when Freeman introduced his chain code. Indeed, Freeman (1974) is responsible for much subsequent work in this area. Freeman (1978) introduced the notion of segmenting boundaries at “critical points” in order to facilitate matching: suitable critical points were corners (discontinuities in curvature), points of inflection, and curvature maxima. This work is clearly strongly related to that of Turney et al. (1985). Early work on Fourier boundary descriptors using the (r, θ) and (s, ψ) approaches was carried out by Rutovitz (1970), Barrow and Popplestone (1971), and Zahn and Roskies (1972); another notable paper in this area is by Persoon and Fu (1977). In an interesting development, Lin and Chellappa (1987) were able to classify partial (i.e., nonclosed) 2-D curves using Fourier descriptors.

At the beginning of the chapter it was noted that there are significant problems in obtaining a thin connected boundary for every object in an image. Since 1988, the concept of active contour models (or “snakes”) solved many of these problems. See Chapter 5 for an introduction to snakes and Chapter 22 for their application to vehicle location.

It is worth remarking on the increased attention to accuracy evident over the past 20–30 years: this is seen, e.g., in the length estimators for digitized boundaries discussed in [Section 10.7](#) (see Kulpa, 1977; Dorst and Smeulders, 1987; Beckers and Smeulders, 1989; Koplowitz and Bruckstein, 1989; Davies, 1991). For a later update on the topic, see Coeurjolly and Klette (2004).

In recent times, there has been an emphasis on characterizing and classifying *families* of shapes rather than just individual isolated shapes: (see in particular Cootes et al. (1992), Amit (2002), and Jacinto et al. (2003)). Klassen et al. (2004)

provide a further example of this in their analysis of planar (boundary) shapes using geodesic paths between the various shapes of the family. In their work, they employ the Surrey fish database (Mokhtarian et al., 1996). The same general idea is also manifest in the self-similarity analysis and matching approach of Geiger et al. (2003), which they used for human profile and hand motion analysis. Horng (2003) describes an adaptive smoothing approach for fitting digital planar curves with line segments and circular arcs. The motivation for this approach is to obtain significantly greater accuracy than can be achieved with the widely used polygonal approximation, yet with lower computational load than the spline fitting type of approach. It can also be imagined that any fine accuracy restriction imposed by a line plus circular arc model will have little relevance in a discrete lattice of pixels. da Gama Leitão and Stolfi (2002) have developed a multiscale contour-based method for matching and reassembling 2-D fragmented objects. Although this method is targeted at reassembly of pottery fragments in archeology, the authors imply that it is also likely to be of value in forensic science, in art conservation and in assessing the causes of failure of mechanical parts following fatigue and the like.

Two useful books are available that cover the subjects of shape and shape analysis in rather different ways: one is by Costa and Cesar (2000) and the other is by Mokhtarian and Bober (2003). The former is fairly general in coverage, but emphasizes Fourier methods, wavelets, and multiscale methods. The latter sets up a scale-space (especially curvature scale-space) representation (which is multiscale in nature), and develops the subject quite widely from there.

10.9.1 More Recent Developments

Ghosh and Petkov (2005) have described problems relating to the robust interpretation of incomplete object boundaries. They discuss the problems in relation to the ICR test—*viz.* assessing recognition rate performance as a function of the percentage of the contour retained, where deletions may occur either as segment deletions, or as occlusions, or as random pixel deletions. Experiments showed that occlusion was the most, and random pixel deletion the least, serious problem. Mori et al. (2005) considered problems relating to 3-D shape recognition from multiple 2-D views. They found that “shape contexts” were particularly important for efficient matching in such situations, shape contexts corresponding to representing shapes by a set of n samples on an object and examining the distribution of relative positions. This technique permitted shape matching to take place efficiently in two stages—fast pruning of possibilities followed by detailed matching.

10.10 PROBLEMS

1. Devise a program for finding a thinned (8-connected) boundary of an object in a binary image.

2.
 - a. Describe the centroidal profile approach to shape analysis. Illustrate your answer for a circle, a square, a triangle, and defective versions of these shapes.
 - b. Obtain a general formula expressing the shape a straight line presents in the centroidal profile.
 - c. Show that there are two means of recognizing objects from their centroidal profiles, one involving analysis of the profile and the other involving comparison with a template.
 - d. Show how the latter approach can be speeded up by implementing it in two stages, first at low resolution, and then at full resolution. If the low resolution has $1/n$ of the detail of the full resolution, obtain a formula for the total computational load. Estimate from the formula for what value of n the load will be minimized. Assume that the full angular resolution involves 360 one-degree steps.
3.
 - a. Give a simple algorithm for eliminating salt and pepper noise in binary images, and show how it can be extended to eliminate short spurs on objects.
 - b. Show that a similar effect can be achieved by a “shrink” + “expand” type of procedure. Discuss how much such procedures affect the shapes of objects: give examples illustrating your arguments, and try to quantify exactly what sizes and shapes of object would be completely eliminated by such procedures.
 - c. Describe the (r, θ) graph method for describing the shapes of objects. Show that applying 1-D median filtering operations to such graphs can be used to smooth the described object shapes. Would you expect this approach to be more or less effective at smoothing object boundaries than methods based on shrinking and expanding?
4.
 - a. Outline the (r, θ) graph method for recognition of objects in two dimensions, and state its main advantages and limitations. Describe the shape of the (r, θ) graph for an equilateral triangle.
 - b. Write down a complete algorithm, operating in a 3×3 window, for producing an approximation to the convex hull of a 2-D object. Show that a more accurate approximation to the convex hull can be obtained by joining humps with straight lines in an (r, θ) graph of the object. Give reasons why the result for the latter case will only be an approximation, and suggest how an exact convex hull might be obtained.
5. An alternative approach to shape analysis involves measuring distance around the boundary of any object and estimating increments of distance as 1 unit when progressing to the next pixel in a horizontal or vertical direction and $\sqrt{2}$, units when progressing in a diagonal direction. Taking a square of side 20 pixels, which is aligned parallel to the image axes, and rotating it through small angles show that distance around the boundary of the square is not estimated accurately by the $1:\sqrt{2}$, model. Show that a similar effect occurs when the square is orientated at about 45° to the image axes. Suggest ways in which this problem might be tackled.