7

An overview of techniques for shape description

Many machine vision applications, and robot vision applications in particular, require the analysis and identification of relatively simple objects which may have to be manipulated. While it is very desirable that a vision system should be able to deal with random three-dimensional presentation of objects, it is also, in general, beyond the current capabilities of most commercial systems. If, however, the objects are (to an approximation) two-dimensional, the problem is more tractable, requiring the description and classification of planar shapes which may, or may not, be partially occluded. In Chapter 9, we will return to the more complex issues of three-dimensional object description and representation when we discuss image understanding. For the present, however, we will stay with the simpler two-dimensional approaches and this chapter introduces some of the more common techniques in shape description. Note well, though, that this chapter is *not* intended to be a complete and rigorous survey of the area of shape description. It is, rather, intended to provide a useful overview of some popular techniques and to identify a taxonomy, or classification, to facilitate the discussion of shape descriptors.

Before proceeding, a brief digression is in order. As we saw in the section on edge detection, topics of fundamental importance in computer vision typically give rise to a very large number of widely differing techniques. Unlike edge detection, however, the issue of shape description and representation is very far from being resolved; although edge detection is not a closed subject by any means, it is a much more mature topic than that of shape. The difficulty with shape is that it is not clear exactly what we are trying to abstract or represent: it seems to depend strongly on the use to which we intend putting the resulting representation (see, for example, Brady, 1983). Practitioners in some areas (e.g. mathematical morphology) go a little further and put forward the idea that shape is not an objective entity, in the sense of having an independent existence, and depends both on the observed and observer. Although these issues are important, it would not be wise to get locked into a deep discussion of them in a book such as this. It is sufficient to note here that *shape* as an issue in computer vision is both ubiquitous and ill-defined; the

solution to the riddle of a general understanding of shape will probably shed light on many other areas in advanced computer vision, if only because it epitomizes the stone wall with which we are currently presented when we wish to develop truly adaptive robust visual systems.

7.1 A taxonomy of shape descriptors

Perhaps the most useful and general taxonomy of shape description is that introduced by Pavlidis in 1978 in which he classifies shape descriptors according to whether they are based on external or internal properties of the shape and according to whether they are based on scalar transform techniques or on space domain techniques. External descriptors are typically concerned with properties based on the boundary of the shape, while internal descriptors, as you would expect, take cognizance of the complete region comprising the shape. Scalar transform techniques generate vectors of scalar features while space domain techniques generate spatial or relational descriptions of certain characteristics of features of the shape. Thus, Pavlidis identifies the following four distinct types of shape descriptor:

(a) external scalar transform techniques utilizing features of the shape boundary;
(b) internal scalar transform techniques utilizing features of the shape region;
(c) external space domain techniques utilizing the spatial organization of the shape boundary;
(d) internal space domain techniques utilizing the spatial organization of the shape region.

7.2 External scalar transform descriptors: features of the boundary

External scalar transform descriptors are based on scalar features derived from the boundary of an object. Simple examples of such features include the following:

- the perimeter length;
- the ratio of the major to minor axis of the minimal bounding rectangle of the shape (see Figure 7.1);
- the number and size of residual concavities lying within the bounds of the shape's convex hull (see Figure 7.2);
- the ratio of the area of a shape to the square of the length of its perimeter (A/P^2): this is a measure of circularity which is maximized by circular shapes;
- the ratio of the area of a shape to the area of the minimal bounding rectangle: this is a measure of rectangularity and is maximized for perfectly rectangular shapes.

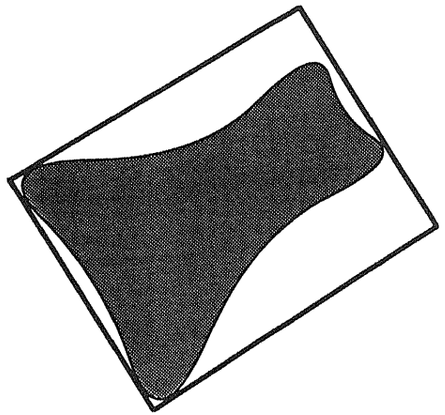


Figure 7.1 Bounding rectangle of a simple two-dimensional shape.

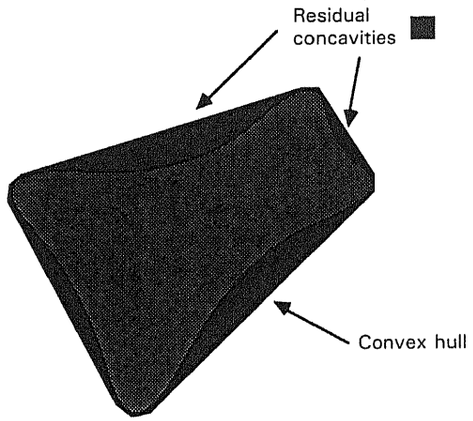


Figure 7.2 Residual concavities within the convex hull of a simple two-dimensional shape.

More sophisticated scalar transform techniques are often based on the Fourier series expansion of a periodic function derived from the boundary. For example, consider the shape depicted in Figure 7.3. The rotation θ of the tangent at the boundary of the object will vary between 0 and 2π radians as the boundary is traversed. In particular, θ will vary with the distance, s, around the perimeter and can be expressed as a function $\theta(s)$. If L is the length of the boundary of the shape, $\theta(0) = 0$ and $\theta(L) = -2\pi$. Unfortunately, this is obviously not a periodic function and so we cannot express it in terms of a Fourier series expansion. However, an

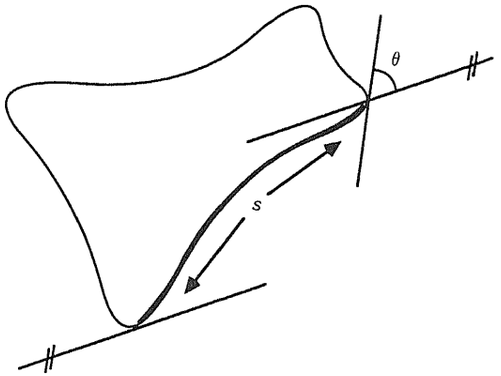


Figure 7.3 Rotation of tangent to a boundary of a shape.

alternative formulation, suggested by Zahn and Roskies, can be. Let $t = (2\pi/L)s$ (thus: $0 \leqslant t \leqslant 2\pi$) and define a new function $\phi(t)$:

$$\phi(t) = \theta\left(\frac{Lt}{2\pi}\right) + t$$

Now, $\phi(0) = \phi(2\pi) = 0$. This function is not dependent on the size, position, and orientation of a shape and, hence, the low-order coefficients of its Fourier series expansion can be used as features for translation, rotation, and scale invariant shape recognition. Unfortunately, it suffers from the disadvantage, common to all transform techniques, of difficulty in describing local shape information, e.g. it would have difficulty in discriminating between two shapes where the only dissimilarity is a small notch in the perimeter.

7.3 Internal scalar transform descriptors: features of the region

Internal scalar transform techniques generate shape descriptors based on the entire shape. One of the most popular is the method of moments. The standard two-dimensional moments m_{uv} of an image intensity function $g(x, y)$ are defined:

$$m_{uv} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} g(x, y)\ x^u y^v \,\mathrm{d}x\,\mathrm{d}y \qquad u, v = 0, 1, 2, 3 \ldots$$

which, in the discrete domain of digital images becomes:

$$m_{uv} = \sum_x \sum_y g(x, y)\ x^u y^v \qquad u, v = 0, 1, 2, 3 \ldots$$

summed over the entire sub-image within which the shape lies.

Unfortunately, these moments will vary for a given shape depending on where the shape is positioned, i.e. they are computed on the basis of the absolute position of the shape. To overcome this, we can use the central moments:

$$\mu_{uv} = \sum_x \sum_y g(x, y)(x - \bar{x})^u (y - \bar{y})^v \qquad u, v = 0, 1, 2, 3 \ldots$$

where:

$$\bar{x} = \frac{m_{10}}{m_{00}} \text{ and } \bar{y} = \frac{m_{01}}{m_{00}}$$

That is, $\bar{x}$ and $\bar{y}$ are the coordinates of the centroid of the shape. Thus, these moments take the centroid of a shape as their reference point and hence are position invariant.

Assuming that the intensity function $g(x, y)$ has a value of one everywhere in the object (i.e. one is dealing with a simple segmented binary image), the computation of m_{00} is simply a summation yielding the total number of pixels within the shape, since the terms in x and y, when raised to the power of zero, become unity. If one also assumes that a pixel is one unit area, then m_{00} is equivalent to the area of the shape. Similarly, m_{10} is effectively the summation of all the x-coordinates of pixels in the shape and m_{01} is the summation of all the y-coordinates of pixels in the shape; hence m_{10}/m_{00} is the average x-coordinate and m_{01}/m_{00} is the average y-coordinate, i.e. the coordinates of the centroid.

The central moments up to order three are:

$$\begin{aligned}
\mu_{00} &= m_{00} \\
\mu_{10} &= 0 \\
\mu_{01} &= 0 \\
\mu_{20} &= m_{20} - \bar{x}m_{10} \\
\mu_{02} &= m_{02} - \bar{y}m_{01} \\
\mu_{11} &= m_{11} - \bar{y}m_{10} \\
\mu_{30} &= m_{30} - 3\bar{x}m_{20} + 2\bar{x}^2 m_{10} \\
\mu_{03} &= m_{03} - 3\bar{y}m_{02} + 2\bar{y}^2 m_{01} \\
\mu_{12} &= m_{12} - 2\bar{y}m_{11} - \bar{x}m_{02} + 2\bar{y}^2 m_{10} \\
\mu_{21} &= m_{21} - 2\bar{x}m_{11} - \bar{y}m_{20} + 2\bar{x}^2 m_{01}
\end{aligned}$$

These central moments can be normalized, defining a set of normalized central moments η_{ij}:

$$\eta_{ij} = \frac{\mu_{ij}}{(\mu_{00})^k}$$

where:

$$k = ((i + j)/2) + 1 \qquad i + j \geqslant 2$$

However, moment invariants (linear combinations of the normalized central moments) are more frequently used for shape description as they generate values

which are invariant with position, orientation, and scale changes. These seven invariant moments are defined as:

not invariant to rotation? Need to rotate into place?

$$\begin{aligned}
\phi_1 &= \eta_{20} + \eta_{02} \\
\phi_2 &= (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \\
\phi_3 &= (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \\
\phi_4 &= (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \\
\phi_5 &= (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})\{(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2\} \\
&\quad + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})\{3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\} \\
\phi_6 &= (\eta_{20} - \eta_{02})\{(\eta_{30} + \eta_{12}) - (\eta_{21} + \eta_{03})^2\} \\
&\quad + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}) \\
\phi_7 &= (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})\{(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2\} \\
&\quad - (3\eta_{12} - \eta_{30})(\eta_{21} + \eta_{03})\{3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\}
\end{aligned}$$

The logarithm of ϕ_1 to ϕ_7 is normally used to reduce the dynamic range of the values when using these moment invariants as features in a feature classification (i.e. shape recognition) scheme.

Shape descriptors based on moment invariants convey significant information for simple objects but fail to do so for complicated ones. Since we are discussing internal scalar transform descriptors, it would seem that these moment invariants can only be generated from the entire region. However, they can also be generated from the boundary of the object by exploiting Stokes' theorem or Green's theorem, both of which relate the integral over an area to an integral around its boundary. We will return to this in the next section on external space domain shape descriptors when describing the BCC (boundary chain code).

7.4 External space domain descriptors: spatial organization of the boundary

One popular technique is the use of syntactic descriptors of boundary primitives, e.g., short curves, line segments, and corners. Thus, the shape descriptor is a list or string of primitive shapes and the formation of the list or string must adhere to given rules: the shape syntax or grammar. A polar radii signature, encoding the distance from the shape centroid to the shape boundary as a function of ray angle, is another much simpler external space domain descriptor (see Figure 7.4).

Descriptors based on external space domain techniques are generally efficient, require minimal storage requirements, and, in the case of the more sophisticated syntactic techniques expounded by Fu *et al.* are based on well-developed general methodologies such as the theory of formal languages: syntactic patterns are recognized by parsing the string of primitive patterns in a manner somewhat similar to the way a compiler parses a computer program to check its syntax (see Fu, 1982). In the case of the simple external space domain descriptors, e.g. radii signatures, the recognition strategy is based on correlation or matching of template signatures, rather than on sophisticated parsing techniques.

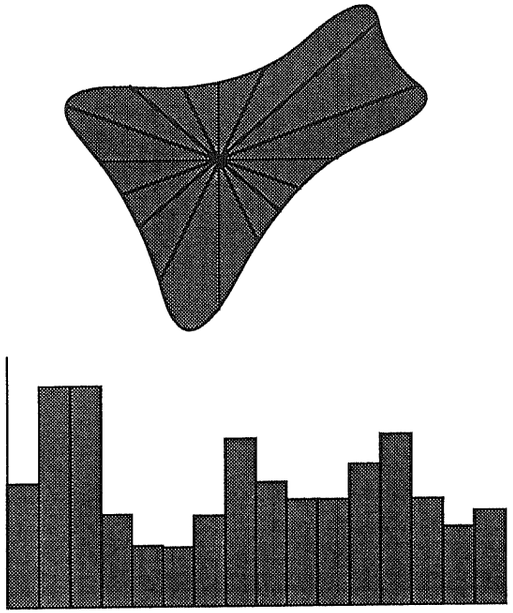


Figure 7.4 Radii signature.

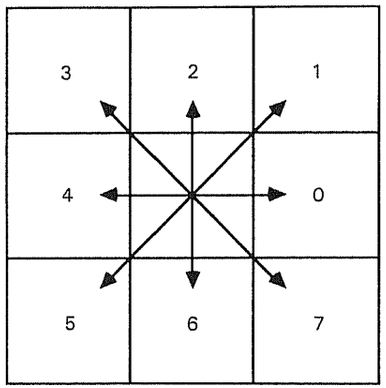


Figure 7.5 Boundary chain code (BCC) directions.

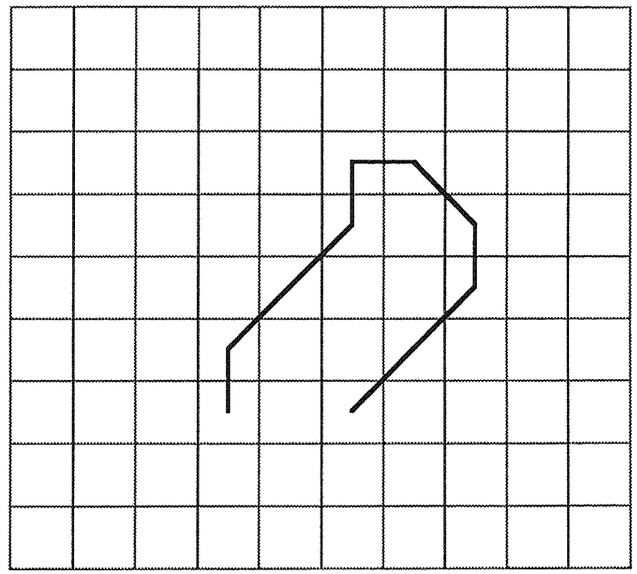
One of the most common external space domain descriptors is the boundary chain code (BCC), introduced in Chapter 5. It should be emphasized, however, that the BCC is more useful for *shape representation* rather than *shape recognition*. The BCC encodes piecewise linear curves as a sequence of straight-line segments called links. A link a_i is a directed straight-line segment of length $T(\sqrt{2})^p$ and of angle

$a_i \times 45^\circ$ (referenced to the X-axis of a right-handed Cartesian coordinate system); T is the grid spacing and is normally set equal to unity; a_i may be any integer in the range 0–7 and represent the (coarsely quantized) direction of the link; p is the modulo two value of a_i; i.e. $p = 0$ if a_i is even and $p = 1$ if a_i is odd. Thus, the link length in directions 1, 3, 5, and 7 is equal to $\sqrt{2}$ and is equal to 1 in directions 0, 2, 4, and 6 (see Figure 7.5). However, you should bear in mind the discussion on inter-pixel distances in Chapter 3. Freeman also proposed the inclusion of signal codes in the chain to facilitate, for example, descriptive comments, chain termination, and the identification of the chain origin. Figure 7.6 illustrates a BCC representation of a simple shape.

Although it is widely used as a technique for shape representation, a BCC is *a non-uniformly sampled* function, that is, the distance between the sample points along the boundary may be either 1 or $\sqrt{2}$ depending on whether the neighbouring boundary points are horizontal/vertical neighbours or diagonal neighbours respectively. Thus, a BCC is dependent on the orientation of the object boundary on two distinct bases:

- Each link encodes the *absolute* direction of the boundary at that point.
- The link length (1 or $\sqrt{2}$) varies with the boundary direction.

Any shape descriptor which is derived from this non-uniformly sampled BCC is



BCC: 2 1 1 2 0 7 6 5 5

Figure 7.6 A BCC representation of a simple shape.

inherently sensitive to changes in orientation. To alleviate this rotational variance it is necessary to remove the dependency on link length, ensuring that the link lengths of the BCC are all equal, specifically by resampling the BCC in uniformly spaced intervals. Note that this resampling technique will generate non-integer coordinate values for the pixels they represent; the actual image pixel values can be obtained for shape reconstruction, which is of interest when displaying the segmented shape, simply by rounding the real-valued coordinates. The fact that the node coordinate values are non-integer is not important if the descriptor is being used for the purposes of shape recognition since one is interested only in correspondence between link values (or the corresponding transformed property) and not the absolute position that they might represent in the image.

Since BCCs are so popular, one simple algorithm for transforming a non-uniformly sampled BCC into a uniformly sampled BCC is presented below. The algorithm, while not as general as others which have been suggested in the computer vision literature, has proven to be adequate for the purposes of constructing rotation-invariant shape descriptors.

7.4.1 An algorithm for resampling boundary chain codes

Let `NUS_BCC` and `US_BCC` represent the non-uniformly sampled and uniformly sampled BCCs, respectively. Let a point given by a non-uniformly sampled BCC be represented by `(nus_x, nus_y)` and let a point given by a uniformly sampled BCC be represented by `(us_x, us_y)`.

`NUS_BCC` and `US_BCC` start at the same point on the contour: `(nus_x, nus_y) = (us_x, us_y)` initially:

```
WHILE there are more NUS_BCC links to be resampled DO

  Generate the next NUS_BCC points: (nus_x, nus_y)

  /* resample */

  REPEAT

     /* generate new US_BCC link and append to US_BCC */

     Generate three candidate uniformly sampled
points:
     (us_x1, us_y1), (us_x2, us_y2), (us_x3, us_y3)
     at a distance of 1 unit from the current uniformly
     sampled point (us_x, us_y)
     in directions corresponding to the NUS_BCC link
     direction, ±1.
```



```
     Test all three points and choose the point which
     is closest to
        (nus_x, nus_y) (using the Euclidean distance
        metric)

     Reassign the current (us_x, us_y) to be this point

     Append to the US_BCC a link with a direction
     corresponding to this chosen point

  UNTIL |us_x - nus_x| < 0.5 AND |us_y - nus_y| < 0.5

  /* (us_x, us_y) now lies within the bounds of the     */
  /* grid pixel given by (nus_x, nus_y)                 */
```

Figure 7.7 illustrates this resampling process: the original non-uniformly sampled

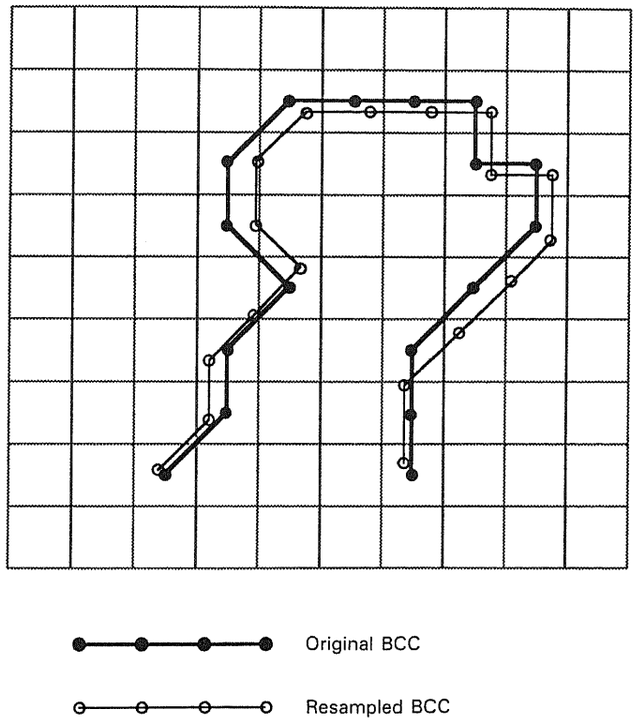


Figure 7.7 Resampling a BCC.

boundary is depicted by dots joined by a dotted line and the resampled boundary is depicted by circles joined by a dashed line.

As mentioned above, the BCC is most useful as a method for the representation of shapes and recognition is normally based upon other descriptors derived from the BCC. For example, the moment shape descriptors discussed previously can be generated from the boundary points given by a BCC. Thus, after Wilf (1981), we have:

$$m_{00} = \frac{1}{2} \sum_{i=1}^{n} A_i$$

$$m_{10} = \frac{1}{3} \sum_{i=1}^{n} A_i(y_i - \tfrac{1}{2}\Delta y_i)$$

$$m_{01} = \frac{1}{3} \sum_{i=1}^{n} A_i(x_i - \tfrac{1}{2}\Delta x_i)$$

$$m_{20} = \frac{1}{4} \sum_{i=1}^{n} A_i(x_i^2 - x_i \Delta x_i + \tfrac{1}{3}\Delta x_i^2)$$

$$m_{11} = \frac{1}{4} \sum_{i=1}^{n} A_i(x_i y_i - \tfrac{1}{2} x_i \Delta y_i - \tfrac{1}{2} y_i \Delta x_i + \tfrac{1}{3}\Delta x_i \Delta y_i)$$

$$m_{02} = \frac{1}{4} \sum_{i=1}^{n} A_i(y_i^2 - y_i \Delta y_i + \tfrac{1}{3}\Delta y_i^2)$$

Where x_{i-1} and y_{i-1} are the coordinates of a point on the perimeter of the shape and x_i and y_i are the coordinates of the subsequent point on the perimeter, as given by the BCC; Δx_i is defined to be $(x_i - x_{i-1})$; Δy_i is defined to be $(y_i - y_{i-1})$; and A_i is defined to be $(x_i \Delta y_i - y_i \Delta x_i)$; n is the number of points on the boundary (i.e. the number of BCC links).

7.5 Internal space domain descriptors: spatial organization of the region

Internal space domain techniques comprise descriptors which utilize structural or relational properties derived from the complete shape. The medial axis transform (MAT), which was mentioned in the section on thinning in Chapter 5, is an example of a commonly used space domain descriptor in that it generates a skeletal line-drawing from a two-dimensional shape. A point in the shape is on the medial axis if and only if it is the centre of a circle which is a tangent to the shape boundary at two non-adjacent points. Each point on the medial axis has a value associated with it which indicates the radius of this circle. This represents the minimum distance to the boundary from that point and thus facilitates the reconstruction of the object. There are various methods for generating the medial axis, the most

intuitive of which is one that is often referred to as the 'prairie fire technique'; it is analogous to setting fire to the boundary of a dry grassy field and letting the flame burn inwards. The points at which the flame fronts meet are on the medial axis. The MAT is sensitive to local distortions of the contour and small deviations can give rise to extraneous skeletal lines. For example, Figure 7.8 illustrates the MAT of a rectangle with a 'bump'.

Other descriptors can be derived using integral geometry: for example, an object shape can be intersected by a number of chords in different directions and the locations and the length of the intersection can be used in various ways as a shape descriptor. One example of this type of descriptor is the normal contour distance (NCD) shape descriptor (see Vernon, 1987). It is essentially a one-dimensional signature in which each signature value represents an estimate of the distance from a point on an object's boundary, with a local orientation of m, to the opposing boundary point which lies on a path whose direction is normal to m. The NCD signature is evaluated over the length of the available boundary. The NCD signature does not require knowledge of the complete boundary and can be used for recognition of partially occluded objects. Note that, if the contour represents a partial boundary, there may not be another boundary point which lies on a path which is normal to the contour and, hence, some segments of the NCD may be undefined.

To conclude this section on internal space domain descriptors, we will discuss the smoothed local symmetries (SLS) representation, which was introduced by Brady. This is a sophisticated and interesting technique because it represents shape with both region-based descriptors and contour-based descriptors. As such, it is more than an internal space domain descriptor and does not require the entire shape to be visible for extraction of useful shape description primitives. An SLS representation has the following three components:

1. A set of *spines* which are the loci of local symmetries of the boundary of the shape.

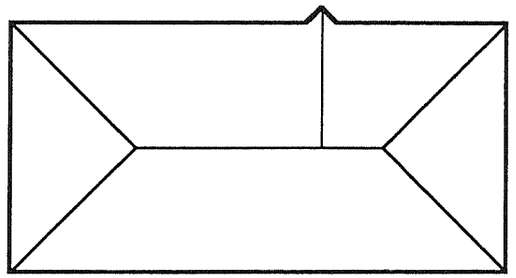


Figure 7.8 Medial axis transform (MAT) of a rectangular shape with a 'bump'.

2. A description of the local shape of the boundary contour in terms of parametric curves (Brady uses circles and spirals) and in terms of primitives of curvature discontinuity. These primitives effectively describe the manner in which the local contour curves are joined together to form the complete boundary shape.
3. A description of the region subtended by two locally symmetric curves. This is effected by a small number of region labels (e.g. cup, sector, wedge, and

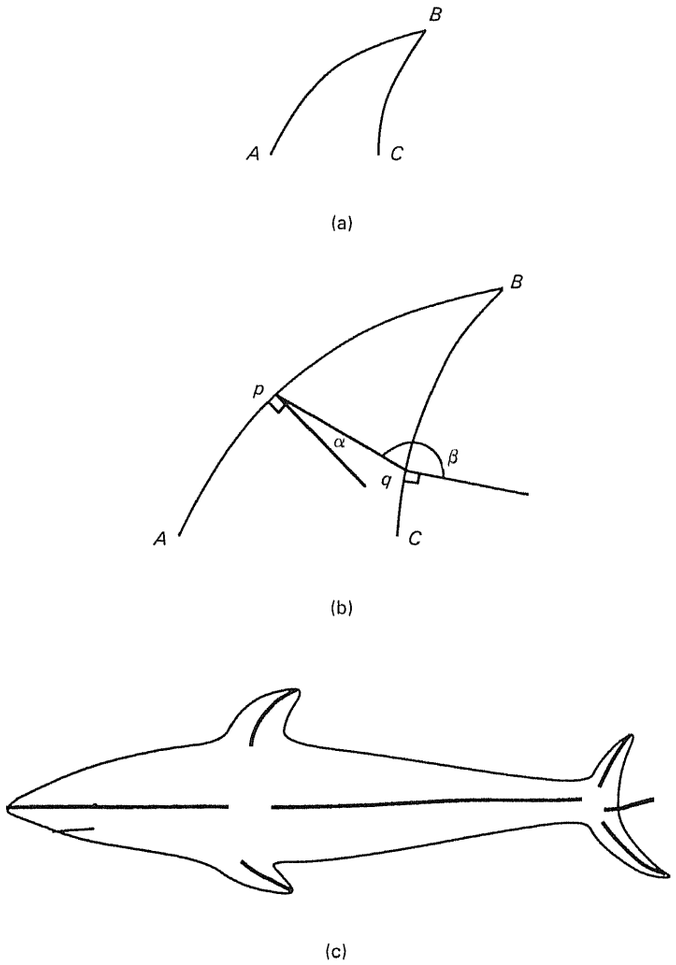


Figure 7.9 (a) Dorsal fin of a shark. (b) Geometry defining a smoothed local symmetry (SLS). (c) Schematic SLS for a complex shape.

so on), each of which has several attributes associated with it. They include the width of the region and the curvature of the associated spine.

For example, consider Figure 7.9(a) which depicts the dorsal fin of a shark. The shape of the fin is defined by two curves AB and BC:

- The curve AB is convex and is described by a circular arc.
- The curve BC is concave and is also described by a circular arc.
- The junction between the two curves at B is an acute-angled *corner*.

These three items form the contour-based description. The descriptor of the region between the two curves AB and BC is labelled a *beak* (the labelling that is used is based on the relative concavity/convexity of the two sides of a local symmetry) and the spine curves to the right. This raises the issue of how the spine is identified. Referring to Figure 7.9(b), let α be the angle subtended by the inward normal to the contour at a point p on the curve AB and let β be the angle subtended by the outward normal to the contour at a point q on the curve BC. A point q on the curve BC forms a local symmetry with the point p on the curve AB if the sum of their respective angles is equal to 180°, i.e. if $\alpha + \beta = 180^\circ$. Several such points q can exist for a given point p. The spine of the SLS is effectively formed by the set mid-points of chords pq which satisfy this condition for local symmetry (for an exact formulation, see Brady and Asada, 1984).

At this stage, we might remark on the similarity between the SLS and the MAT. However, the SLS differs in several important ways: the local symmetry which is implicit in the MAT is made explicit in the SLS; global symmetries, such as that which lies between a long fork (see Figure 7.10), are not made explicit by the MAT; and the SLS includes the attributed contour and regions descriptors discussed above – the SLS is more than just a collection of spines. Finally, let us note that the SLS representation is also intended to be used as a descriptor of complex shapes, which can be viewed as a network of locally symmetric parts (see Figure 7.9(c)). The connections in the network are relationships between the sub-parts and carry information (or attributes) which can help disambiguate between similar objects.

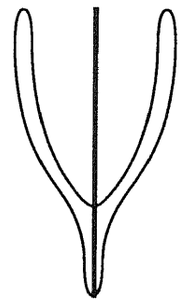


Figure 7.10 Global SLS of a forked object.

Exercises

1. Discuss how an object skeleton would be used as a shape descriptor, paying particular attention to issues of shape recognition.
2. Generate a BCC representing a circle of diameter 10 pixels. Derive a uniformly sampled version and compare the perimeter length of both versions with the theoretical value.
3. Would a radii signature shape descriptor be useful for the recognition of occluded shapes? Why?
4. Bearing in mind that the minimum bounding rectangle of an arbitrarily oriented shape requires the prior estimation of its major axis, how would you go about computing this shape descriptor?

References and further reading

Anderson, R.L. 1985 'Real-time grey-scale video using a moment-generating chip', *IEEE Journal of Robotics and Automation*, Vol. RA-1, No. 2, pp. 79–85.

Ballard, D.H. and Sabbah, D. 1983 'Viewer independent shape recognition', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-5, No. 6, pp. 653–60.

Bamieh, B. and De Figueiredo, R.J.P. 1986 'A general moment-invariants/attributed-graph method for three-dimensional object recognition from a single image', *IEEE Journal of Robotics and Automation*, Vol. RA-2, No. 1, pp. 31–41.

Berman, S., Parikh, P. and Lee, C.S.G. 1985 'Computer recognition of two overlapping parts using a single camera', *IEEE Computer*, pp. 70–80.

Bolles, R.C. 1980 *Locating Partially Visible Objects: The Feature Focus Method*, SRI International, Technical Note No. 223.

Bolles, R.C. and Cain, R.A. 1982 'Recognizing and locating partially visible objects: the local-feature-focus method, *The International Journal of Robotics Research*, Vol. 1, No. 3, 1982, pp. 57–82.

Brady, M. 1983 'Criteria for representations of shape', *Human and Machine Vision*, A. Rosenfeld and J. Beck (eds), Academic Press, New York.

Brady, M. and Asada, H. 1984 'Smoothed local symmetries and their implementation', *The International Journal of Robotics Research*, Vol. 3, No. 3, pp. 36–61.

Freeman, H. 1961 'On the encoding of arbitrary geometric configurations', *IRE Transactions on Electronic Computers*, pp. 260–8.

Freeman, H. 1974 'Computer processing of line-drawing images', *ACM Computing Surveys*, Vol. 6, No. 1, pp. 57–97.

Fu, K-S. (ed.) 1980 *Digital Pattern Recognition*, Springer Verlag, Berlin.

Hu, M.K. 1962 'Visual pattern recognition by moment invariants', *IRE Transactions on Information Theory*, Vol. IT-8, pp. 179–87.

Lin, C.C. and Chellappa, R. 1987 'Classification of partial 2-D shapes using Fourier descriptors', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-9, No. 5, pp. 686–90.

Nackman, L.R. and Pizer, S.M. 1985 'Three-dimensional shape description using the symmetric axis transform 1: Theory', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-7, No. 2, pp. 187–202.

Pavlidis, T. 1978 'A review of algorithms for shape-analysis', *Computer Graphics and Image Processing*, Vol. 7, pp. 243–58.

Pavlidis, T. 1980 'Algorithms for shape analysis of contours and waveforms', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-2, No. 4, pp. 301–12.

Shahraray, B. and Anderson, D.J. 1985 'Uniform resampling of digitized contours', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-7, No. 6, pp. 674–81.

Tang, G.Y. 1982 'A discrete version of Green's theorem', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-4, No. 3, pp. 242–9.

Vernon, D. 1987 'Two dimensional object recognition using partial contours', *Image and Vision Computing*, Vol. 5, No. 1, pp. 21–7.

Wiejak, J.S. 1983 'Moment invariants in theory and practice', *Image and Vision Computing*, Vol. 1, No. 2, pp. 79–83.

Wilf, J.M. 1981 'Chain-code', *Robotics Age*, Vol. 3, No. 2, pp. 12–19.

Wong, R.Y. 1978 'Scene matching with invariant moments', *Computer Graphics and Image Processing*, Vol. 8, pp. 16–24.

Zhan, C.T. and Roskies, R.Z. 1972 'Fourier descriptors for plane closed curves', *IEEE Trans. Computers*, Vol. C-21, pp. 269–81.