Reprinted from O.D.Faugeras (ed.) Fundamentals in Computer Vision

Printed in Great Britain

FEATURE-BASED 2-D SHAPE MODELS

T. C. Henderson
Department of Computer Science
The University of Utah
Salt Lake City, Utah 84112

The solution to a complex shape analysis problem generally requires the design of shape modeling techniques and procedures for organizing unknown shapes according to such models. Once a shape modeling mechanism has been chosen, specific models for the classes of shapes to be analyzed can be constructed. This might involve simply determining values for specific shape features or detailing the spatial organization of a shape. The process of choosing a modeling mechanism and then constructing shape models is complicated by a variety of factors that influence both the appearance of specific shapes in images and the segmentation of shapes of individual objects from images. These factors include: geometric transformations, obstruction, agglomeration and noise. In this chapter, we consider shape models based on features of the area, boundary or special axes of the objects to be modeled.

AREA METHODS

Area methods are based on the knowledge of all points belonging to the shape, that is, both interior and boundary points. One commonly used area model is the method of moments. The theory of 2-D moment invariants for planar geometric figures is described by Hu (1962). The method provides for recognition of shapes independent of position, size, and orientation. The (p,q)th moment of a shape is defined as:

$$m(p,q) = \iint x^p y^q r(x,y)\,dxdy \qquad p,q = 0,1,2...$$

where r(x,y) is the characteristic equation for a bounded shape, i.e., r(x,y) = 1 if (x,y) is in the shape and 0 otherwise. The sequence {m(p,q)} is uniquely determined by r(x,y), and vice versa. Definitions and properties of moment invariants under translation, scaling, orthogonal transformations and general linear transformations are developed.

These moment invariants rather than the actual moments can be used for shape modeling.

Thus, the method of moments presents many desirable feature However, the method has several drawbacks. A significant amount of computation is required to compute them. Also, although the first few moments may suffice to model a simple shape, more complicated shapes require many more terms of the sequence $\{m(p,q)\}$. Moreover, distortion of a shape by noise and poor segmentation is not easily modeled in terms of transformations of the moments. Finally, if parts of the shape are obscured, or if agglomeration occurs, then the moments of the resulting shapes are radically different from those of the original shape.

In a digital image, $f(x,y)$, the projection of that image (see Rosenfeld (1976)) may be defined as a function:

$$p(i) = \sum_{x,y \in I_i} f(x,y)$$

where I_i describes a family of curves, e.g., a set of lines or circles. Some standard projections include the x-projection of f, and the y-projection of f, which are just the column sums and the row sums, respectively. These projections can be used to detect blobs by merely looking for plateaus in the projection function. Linear objects can be detected if they run perpendicular to the projection axis. Thus, if their orientation is unknown, several projection axes might be tried. If several objects must be recognized, and the objects have a known orientation, then a projection function might suffice to distinguish between them (e.g., character identification).

A different approach to shape decomposition is to segment the area of a shape into convex subregions. These area-based shape models use convex polygons as the primitive elements of the representation. Given a set of points, S, the usual definitions for convexity include:

- For every two points p and q in S, the line segment from p to q lies entirely in S, or

- For every two points p and q in S, the midpoint of the line segment from p to q lies in S.

However, these definitions require some care if they are to be implemented for a digital image. Namely, line segments must be digital line segments, and the midpoint must be compared to points within half a unit of the midpoint. The definition of convex objects in a digital setting remains a problem for the application of these methods.

Once a suitable definition of convexity has been chosen, a set of points must be broken into its convex subsets. In general, there is no unique convex segmentation. One approach to solve this problem is to start with several (say 3) close points, and "grow" a convex face containing those parts, i.e., points of S are added until no point can be added and still satisfy the definition of convexity. A structural approach to this problem is described by Pavlidis (977). Such an approach is complicated by the fact that a polygon cannot, in general, be expressed uniquely as the finite union of its convex subsets. Moreover, decompositions into convex subsets may not necessarily correspond to a natural organization of the shape.

BOUNDARY METHODS

Several classes of shape models based on the boundary of a shape have been proposed, including, polygonal approximations, Fourier descriptors, B-splines, Hough transforms and shape numbers. All of these methods depend on extracting the object in terms of the boundary between the object and the background.

Chain codes, and more generally, boundary segments determined by piecewise functional approximations provide a slope intrinsic representation of shape. A grid is superimposed on the shape, and one of two methods is used to encode the boundary of the shape. Some starting point on the boundary is arbitrarily chosen; then for each point of intersection of the shape boundary with a grid line, the nearest point of the grid is included in the encoding of the shape boundary. If the elements of the encoding are joined together, they define a polygonal arc which is an approximation to the shape boundary. This arc is called a chain and can be completely specified by giving a starting point and the successive directions necessary to follow the chain. For an eight neighbor grid, these directions can be efficiently encoded with 3 bits corresponding to directions 0 degrees to 315 degrees in 45 degree increments. Such a list is called a chain code, and each element is a chainlet.

Various properties of chain codes can be easily derived and used in defining shapes (see Freeman (1974)). For example, the Euclidean length of a chain is the sum of the number of even chainlets and the /2 times the number of odd chainlets. Some other properties which can be easily computed include the maximum cross section length in any of the eight given orientations and the first moments in any of these orientations. This is quite attractive in many applications.

When the approximating functions are line segments, then this is a generalization of the chain code which allows for arbitrary lengths and directions of chain elements (also see Freeman (1979) on generalized chain codes). In general, the shape boundary is segmented into pieces described by arbitrary functions. It is usually sufficient for the functions to be restricted to low-order polynomials (of degree less than three).

Perhaps the most frequently used shape boundary representation is the piecewise linear approximation of the boundary. Many algorithms have been proposed for computing various piecewise linear approximations - see, e.g., Pavlidis (1973a, 1974, 1975), Ramer (1972), Rosenfeld & Johnston (1973) and Davis (1977). These procedures can be categorized as one of two types: (1) those that attempt to find the lines directly, and (2) those that attempt to find the break points directly. The first class of procedures search for boundary segments which are well fit by lines, while the second class search for boundary points with locally high curvature, that is, they are angle detectors. Of all these algorithms, the split-and-merge algorithm proposed by Pavlidis (1974) seems to be the most robust (i.e., it has very slight sensitivity to small changes in the underlying shape) yet is at the same time computationally efficient.

Piecewise linear approximation can be used not only for feature extraction, but also for noise filtering and data compaction (see Pavlidis (1973b)). Let $S = \{(x_i,y_i)\}$, $i = 1,N$, be a set of points; the problem is to find a minimum partition of S into n subsets $s_1,\ldots,s_n$ where $s_i = \{(x_j,y_j)\}$ $j = K_i,l_i$ and s_i is approximated by a straight line with an error norm less than a prespecified threshold, E_{max}. Given an error norm E and a segmentation $s_1,\ldots,s_r$, let E_i be the value of the norm evaluated on s_i, and let $E = E_i$, $i = 1,r$. Then the split -and-merge algorithm is:

1. For $i = 1,r$, if $E_i > E_{max}$, then split s_i into two subsets, set $r = r + 1$ and calculate the error norms of the two subsets. The breakpoint for obtaining the two new sets can be chosen in one of several ways. Pavlidis proposes using the point which contributes the most to the error value or the midpoint.

2. For $i = 1,r$, if E_i and E_{i+1} can be merged and the error norm of the new segment is less than E_{max}, then merge E_i and E_{i+1} and reduce r by one. Compute the error norm of the new segment.

3. Adjust the segment end points to minimize E. If no changes were made in steps (2) or (3), then terminate, else go to (1).

There are many reasonable choices for the error norm. Pavlidis describes the Euclidean distance between $\{x_i, y_i\}$ and the approximating curve, while Horowitz (1977) demonstrates the use of the mean square error norm. To minimize the latter norm means to find:

$$\min_{a,b} \{ \Sigma [y_i - (ax + b)]^2 \}$$

There exist unique closed forms for a and b:

$$a = (n\Sigma x_i y_i - \Sigma x_i \cdot \Sigma y_i)/(n\Sigma x_i^2 - (\Sigma x_i)^2)$$

$$b = (\Sigma y_i - a \cdot \Sigma x_i)/n.$$

An important practical advantage of this error norm is that the various sums used in defining a and b need not be recalculated completely for updating purposes; they can be directly added or subtracted as sums.

One can obtain higher order approximations using the split-and-merge algorithm with higher degree polynomials, but as Pavlidis discusses (1973a), the computational cost increases dramatically as one raises the degree of the approximating curves. Furthermore, the algorithms become numerically less stable. Pavlidis suggests using the results of the piecewise linear approximations to selectively guide the application of higher order approximation procedures to pieces of the shape boundary.

A description of 2-dimensional non-intersecting closed curves proposed by Bribiesca & Guzman (1979) called the shape number is associated with the boundary of simply connected regions. This shape number is obtained by laying a grid over a shape and encoding the border around the grid squares that fall (at least 50% of the square) within the shape. In particular, the shape number has several chain code representations, namely, one for each starting point on the boundary. If the derivative of the chain code is used, i.e., replace each convex corner of the chain by a 1, each straight corner by a 2, and each concave corner by a 3, then the chain with the minimum value when viewed as an integer can be used to represent the shape. In order to normalize the shape, a grid is chosen after the shape is surrounded by its basic rectangle (i.e., its orientation coincides with the major axis of the shape). Then the shape can be represented to any level of detail by refining the grid. In this way, a shape description is obtained that is independent of size, orientation and position.

The boundary of a 2-D shape can be expressed in term of slope of the boundary as a function of arc length, Zahn (1972), or as a complex parametric function, Granlund (1972). In either case, the function is periodic and can be expanded in a Fourier series. The shape can be approximated to any desired degree of accuracy by retaining a sufficient number of terms of the series. For example, suppose that the boundary of a shape is expressed in parametric form:

$$Z(t) = (x(t), y(t)), \quad 0 \leq t < L,$$

where L is the length of the boundary. Let T(t) be the angular direction at point t, and let T(0) = C. Define P(t) to be the amount of angular bend between the starting point and t. Then, P(t) + C = T(t). The function P(t) can be normalized over the interval [0,2] as follows:

$$P^*(t) = P(Lt/2) + t.$$

Then, $P^*(0) = P^*(2) = 0$. There exists a sequence $\{A_k, a_k\}$, k=1 to ∞, and u such that

$$P^*(t) = u + \Sigma A_k \cos(kt - a_k).$$

The A_k, a_k k = 1 to ∞ are the Fourier descriptors.

Transforms other than the Fourier transform can be used, e.g., the Walsh transform. However, one of the main advantages of the Fourier transform is the well-developed theory and software to implement it. Fourier shape models can be made independent of position, orientation and scale. The major disadvantages of the method are that local features of the shape are difficult to describe without taking many coefficients and that obstruction or agglomeration produce coefficients unrelated to those of the original shape.

An approach to boundary representation that has received much attention in the areas of computer graphics and CAD/CAM is the use of B-splines. As described by Gordon & Riesenfeld (1974), a polynomial spline is a generalized polynomial with specified points of derivative discontinuity. Usually, a spline function of degree m-1 is defined over a sequence of intervals and is a polynomial of degree m-1 on each subinterval, and its derivatives of orders 1,2,...,m-2 are everywhere continuous. Various bases exist for the space of all such spline functions, however, the B-spline basis functions are a commonly used basis. Arbitrary primitive functions can be represented by a B-spline approximation which consists of a weighted sum of the basis functions. B-spline approximations are variation diminishing representations and provide local approximations. Although these are convenient properties of general splines, only first order splines have had much success in shape analysis since the theoretical and computational problems are much more complex for higher-order splines; moreover, in shape perception, linear approximations capture most relevant information and continuity conditions are usually not too important (see Pavlidis (1977) for a discussion of the use of splines in shape analysis).

Another approach to 2-D shape modeling is to transform the elements of the shape into a parameter (or transform) space. The Hough transform is perhaps the most important example of this approach. Originally, the Hough transform was used to detect simple curves, e.g., lines or circles. Usually this method is applied to an edge image, i.e., a binary or thresholded description of the edges in the original image. The (x,y) location of an edge response restricts the set of possible lines that this edge could lie on. This set of lines can be represented by a couple of quantized parameters, e.g., slope and intercept. The complete set of lines possible for the whole image can be represented by an accumulator array whose axes are slope and intercept.

Then, for every (x,y) location of an edge response the accumulator array is incremented for every possible line through (x,y). If many edge responses lie on the same line then this results in a high value at the position in the accumulator array corresponding to that line.

The Hough transform has been generalized for arbitrary shapes in the plane by Merlin & Farber (1975). Given a list (x_i, y_i) i=1,n of locations designating a shape, then the shape is modeled by choosing a reference point, (x_o, y_o), and keeping list of displacement vectors $D=(\delta x_i, \delta y_i)$ i = 1 to n where $\delta x_i = x_o - x_i$ and $\delta y_i = y_o - y_i$. This list constitutes the model for that shape. In order to detect the shape in an image, an accumulator array H is initialized to zero and for each edge location (x,y) detected in the image, H $(x+\delta x_i, y+\delta y_i)$ is incremented by one. Maxima in H should represent the location of the reference point (x_o, y_o). This algorithm is actually an efficient binary convolution of the shape model and the edge image. Moreover, this approach can be extended to account for orientation and scale changes (see Davis & Yam (1981)). If one is willing to pay the overhead of a complete edge description, e.g., edge likelihood and orientation, then one can use even more efficient generalization of the Hough transform has been proposed by Ballard (1981

SPECIAL AXES METHODS

Sweep representations describe shape in terms of special axes which can be given as a set of points or as a function. Associated with each point of the axis is either a geometric object, e.g., a circle, or some deformation of that object. The two major examples of this approach are the medial axis transform and ribbons: the 2-D version of generalized by cylinders.

The medial axis transform proposed by Blum (1964 differs from the previously discussed methods in that a new object is derived from the given one. This is one of the earliest proposed shape modeling techniques and has been widely studied. Let R be the set of points defining an object and let B be the set of boundary points of R. Let N_x be the set of points that are in B and whose distance from x is less than or equal to the distance of x from any other point of B. Then medial axis transform consists if two parts:

-The set $S = \{x : N_x$ has more than one member$\}$, and

-The radius of the largest disk contained in R and centered at x for each x in S.

Thus, a spatial decomposition of R is given in terms of S, also known as the skeleton of R. The medial axis transform has many desirable properties; for example, to determine whether or not a point, p, is in the interior or R, one need only compute the distance of p from each point s of the skeleton and see if that distance is less than the radius of the disk associated with s.

The medial axis transform has several disadvantages. The skeleton of an arbitrary object is not as economical a representation as the boundary of the object. Moreover, digital approximations to the skeleton may not be connected and are very sensitive to noise. Finally, there is no obvious way to compute properties of the original shape directly from the skeleton.

Another successful sweep representation is that of ribbons. A ribbon is a 2-D restriction of the 3-D shape modeling method of generalized cylinders. Basically, a ribbon is a means of describing the projection of a generalized cylinder. 3-D objects are described by a basic 2-D shape, an axis along which the shape is moved and a description of the transformation of the 2-D shape as a function of position along this axis. A ribbon describes the relations between lines in an image in the same way, except that the special axis is restricted to stay in the plane, as is the 2-D shape (see Brook (1981)).

REFERENCES

Ballard, D.H. (1981). Generalizing the Hough Transform to Detect Arbitrary Shapes. Pattern Recognition, 13, no.2, 111-122.

Blum, H. (1964). A Transformation for Extracting New Descriptions of Shape. Symp. on Models for the Perception of Speech and Visual Form, MIT Press.

Bribiesca, E. & Guzman, A. (1979). How To Describe Pure Form and How To Measure Differences in Shape Using Shape Numbers. Proc. PRIP, Chicago, 427-436.

Brooks, R. (1981) Symbolic Reasoning Among 3-D Models and 2-D Images. Rept. No. STAN-CS-81-861, June.

Davis, L. (1977). Understanding Shape, I: Angles and Sides. IEEE T. on Computers, C-26, 236-242.

Davis, L. & Yam, S. (1980). A Generalized Hough-like Transformation for Shape Recognition. TR-134, Univ. of Texas.

Freeman, H. (1974). Computer Processing of Line-Drawing Images. Computing Surveys, March, 57-97.

Freeman, H. (1979). Use of Incremental Curvature for describing and Analyzing 2-D Shape. Proc. PRIP, Chicago, 437-444.

Gordon, W.J. & Riesenfeld, R.F. (1974). B-Spline Curves and Surfaces. In Computer Aided Geometric Design, ed. R.E. Barnhill & R.F. Riesenfeld, pp. 95-126. New York: Academic Press.

Granlund, G.H. (1972). Fourier Preprocessing for Hand Print Character Recognition. IEEE T. on Computers, C-21, 195-201.

Horowitz, S. (1977). Peak Recognition in Waveforms. In Syntactic Pattern Recognition Applications, ed. K.S. Fu, pp. 1-31. Berlin: Springer-Verlag.

Hu, M. (1962). Visual Pattern Recognition by Moment Invariants. IRE T. on Inf. Theory, IT-8, 179-187.

Merlin, P.M. & Farber, D.J. (1975). A Parrallel Mechanism for Detecting Curves in Pictures. IEEE T. on Computers, C-24, 96-98.

Pavlidis, T. (1973a). Waveform Approximation Through Functional Approximation. IEEE T. on Computers, C-22, 689-697.

Pavlidis, T. & Horowitz, S. (1973b). Piecewise Approximation of Plane Curves. Pattern Recognition, 346-405.

Pavlidis, T. & Horowitz, S. (1974). Segmentation of Plane Curves. IEEE T. on Computers, C-23, 860-870.

Pavlidis, T. & Ali, F. (1975). Computer Recognition of Handwritten Numerals Using Polygonal Approximations. IEEE T. on Systems, man, and Cybernetics, SMC-5, 610-614.

Pavlidis, T. (1977). Structural Pattern Recognition. Berlin: Springer-Verlag.

Ramer, U. (1972). An Iterative Procedure for Polygonal Approximation of Plane Curves. CGIP, 1, 244-256.

Rosenfeld, A. & Johnston, E. (1973). Angle Detection in Digital Curves. IEEE T. on Computers, C-22, 874-878.

Rosenfeld, A. & Kak, A. (1976). Digital Picture Processing. New York: Academic Press.

Zahn, C.T. & Roskies, R.Z. (1972). Fourier Descriptors for Plane Closed Curves. IEEE T. on Computers, C-21, 269-281.