Ambiguous Trihedral Solids," *Computer Graphics and Image* Processing, Vol. 6, 1977, pp. 61-89.
[9] T. Kanade, "A Theory of the Origami World," *Artificial Intelligence*, Vol. 13, 1980, pp. 279-311.
[10] K. Sugihara, "Picture Language for Skeletal Polyhedra," *Computer Graphics and Image Processing*, Vol. 8, 1978, pp. 382-405.
[11] G. R. Grape, "Model Based (Intermediate-level) Computer Vision," Stanford Artificial Intelligence Laboratory, Memo AIM-201, May 1973.
[12] A. K. Mackworth, "Interpreting Pictures of Polyhedral Scenes," *Artificial Intelligence*, Vol. 4, 1973, pp. 121-137.
[13] B. Horn, "Understanding Image Intensities," *Artificial Intelligence*, Vol. 8, 1977, pp. 201-231.
[14] P. H. Winston, "Learning Structural Descriptions from Examples," MIT Report MAC-TR-76, September 1970.
[15] P. H. Winston, "Learning Structural Descriptions from Examples," in *The Psychology of Computer Vision,* P. H. Winston (ed.), McGraw-Hill, New York, 1975, pp. 157-209.

5

SHAPE ANALYSIS AND RECOGNITION

Scenes of interest rarely contain solely polyhedral objects. The motive in studying the problems of the blocks world was to better understand some aspects of the perceptual process. The main problems encountered were of the low-level processes to obtain object boundaries, segmentation of scenes containing multiple occluding objects, recognition of objects under the changes of scale, rotation, perspective, and varying amounts of occlusion, and descriptions of object assemblies. We now examine the generalization of these processes for nonpolyhedral objects, concentrating on the problems of shape analysis and recognition based on these descriptions.

Boundary-extraction techniques of polyhedral objects generalize to other objects, provided that the surfaces of objects are still homogeneous and shading of the surfaces due to curvature is not strong. These techniques must now be more local, as boundaries are not necessarily straight. If the objects or the background are textured, boundary extraction becomes a complex problem (for polyhedral objects as well). These low level processes will be discussed in detail in the succeeding chapters. For now, we assume that perfect boundaries, corresponding to discontinuities in the object surfaces or their slopes, are available.

Segmentation techniques for polyhedral objects were based on effective utilization of the knowledge of constraints placed on the images by the nature of the objects and the image-formation process.

Huffman-Clowes-Waltz techniques used generic knowledge common to trihedral objects, whereas Roberts' technique required knowledge of specific potential objects in a scene (the segmentation being performed simply by recognition). Huffman has generalized his line-labeling theories to apply to arbitrary polyhedral objects and also to objects of zero-Gaussian-curvature surfaces [1]. Chien and Chang have attempted to generate a catalog of vertex types for scenes of simple curved objects such as circular cylinders and cones [2]. However, no similar segmentation techniques have been developed for more general objects.

Scene segmentation is greatly simplified if three-dimensional positions of points on the *visible* surfaces of the objects are available. Such information, sometimes known as *two-and-a-half* dimensional data, can be obtained from multiple views of a scene, as in stereo vision, artificial range measuring devices, and to a certain extent from examination of the variations in surface brightness. These techniques are discussed in Chapter 9. Availability of such information is assumed in some of the techniques described in this chapter.

For most of the remainder of this chapter we assume that perfect object boundaries are available, and that the objects have been segmented or are to be segmented by recognition (that is, knowledge of specific objects is available). 3-D information of the visible surfaces will be assumed where indicated. In this chapter we will concentrate on descriptions of the shape properties of an object and their recognition, based solely on these shape descriptions. 2-D shape analysis is also included for completeness.

5.1 REPRESENTATION OF COMPLEX SHAPES

A good shape representation should allow recognition from partial views of an object, and small changes in object shape should cause only small changes in the shape description. Representation of articulation of parts of an object should be convenient, and the representation should allow a comparison of differences and similarities of two objects rather than just simple classification. The latter property is important if the machine must deal with new objects that are similar to previously seen objects. An encounter with a purple, five-legged "cow" should result in just such a description of differences and not simply an answer that this object is unknown.

The above requirements are largely satisfied if the complex objects are represented by segmentation into simpler parts and the interrelationships of these parts. As example, a human shape could be described to consist of various limbs such as head, arms, body and legs, and the manner in which they are connected to each other. Each limb

may again be described in more detail in a similar fashion—for example, the arms consisting of an upperarm, forearm, and a hand, and the hand in turn being made of a palm and the fingers, and so on. It is sometimes useful to view such descriptions as comprising a graph structure with the parts being the nodes of the graph and relations between the parts being the arcs of this graph. The relations may be of part/whole and connectivity or the more complex relations such as similarities of some parts, for example, left and right arms, or similarities of groups of parts such as bilateral symmetry of a human shape. Such descriptions are also called *relational descriptions* or *structural descriptions.*

Recognition is by matching of two relational descriptions. Partial views of an object generate description graphs that are subgraphs of complete object descriptions and can be accommodated in the matching process. The variations of an object, such as extra parts or articulation of parts, are described naturally.

Such descriptions should be contrasted with representations based on properties of the complete surfaces, such as a Fourier series or moments expansion of a surface. Changes caused by partial views in such descriptions are not easily described, nor do they allow a useful comparison of similarities and differences between two objects. However, they are easily computed and are useful for applications where the above considerations are not important.

In the following, various representations, both segmented and otherwise, are discussed for line, area, and volume shapes.

5.2 LINE DESCRIPTIONS

Descriptions of curves are important for special objects such as the characters of an alphabet, and also for objects in three-dimensional scenes such as roads in an aerial photograph. Further, shape descriptions of three-dimensional objects are sometimes reduced to "skeleton" line structures, as described in Sections 5.3 and 5.4.

5.2.1 Storage of Lines

A line is most easily described by an ordered list of the coordinates of the successive points along it. A significant savings in storage can be obtained if only the coordinates of the starting point and incremental changes for the successive points are stored. A popular technique due to H. Freeman [3], known as chain coding, operates by assigning an integer code to each of the eight neighbors of a pixel as shown in Fig. 5-1. An arbitrary curve is described by a starting point and the code corresponding to each successive point on the curve. The chain code of the line shown in Fig. 5-2, for example, is 0, 1, 2, 2, 3, 2.

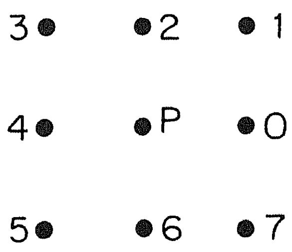

Figure 5-1: Chain codes of the eight neighbors of a pixel

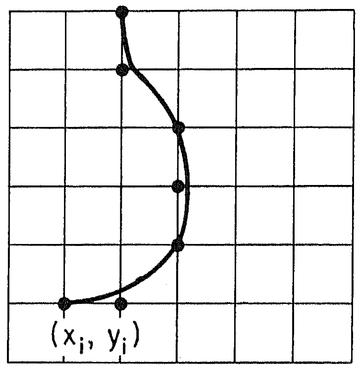

Figure 5-2: A chain-coded curve

Similarities of two curves with chain codes of $\mathbf{a} = (a_1, a_2, \ldots a_n)$ and $\mathbf{b} = (b_1, b_2, \ldots, b_n)$ can be defined by

$$C_{ab} = \frac{1}{n} \sum_{i=1}^{n} a_i \cdot b_i \qquad (5\text{-}1)$$

where $a_i \cdot b_i = \cos(\text{angle } (a_i) - \text{angle } (b_i))$, where angle (α) stands for the angle denoted by code α. This measure is useful only if the two curves are of the same scale, length, and orientation but may have different starting positions. Similarity of two curves of different length can be measured by "sliding" one curve with respect to another and choosing the maximum value of the above similarity measure—that is, picking the maximum value of $C_{ab}(j)$ for all values of j where

$$C_{ab}(j) = \frac{1}{k} \sum_{i=1}^{n} a_i \cdot b_{i+j} \qquad (5\text{-}2)$$

where k is the length of the smaller curve.

5.2.2 Line Approximations

Compact and structured descriptions of a curve are obtained by approximating it by expansion in an orthogonal series of functions or by piecewise segments of simpler curves. Approximation by piecewise linear segments is common, and splines, which are piecewise polynomials with continuity conditions defined at the junctions, are a generalization.

A very simple and effective technique for piecewise linear segment approximation is that of *iterative end-point fitting*, which operates by connecting the end points of a given curve by a straight line and searching for the point on the curve that is farthest from this line. If this distance is unacceptably large, the curve is segmented in two at the point of the maximum excursion and the process iterated for the two segments, as shown in Fig. 5-3. To apply this technique to closed curves, an initial segmentation into two parts, usually derived arbitrarily, is needed. Implementation of piecewise linear approximations is described in [4-7]. Details of spline approximations may be found in [8].

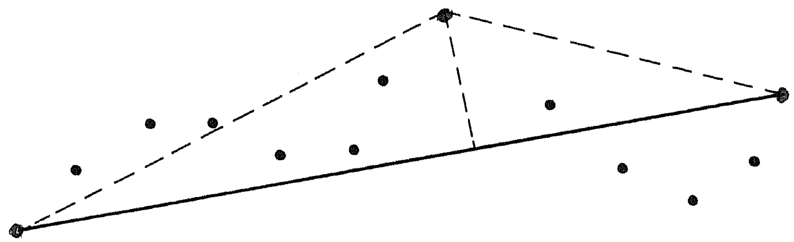

Figure 5-3: Iterative end-point fit

The above technique does not always produce segmentation points or corners that correspond to the human perception. For our perception, the most natural points of segmentation seem to be the points of maximum curvature. To be useful, computation of curvature in a digital curve requires differencing and averaging over a range. A comparative survey of corner-finding methods is given in [9].

Qualitative descriptions of a curve can be generated in terms of its segmentation, by using the number of corners and inflection points, and descriptions of the simpler segments such as being straight or circular.

5.2.3 Analytical Line Shape Measures

The coefficients associated with an analytical approximation of a curve can be used as shape features. Curves of different shape will have different coefficients. However, variations with scale, rotation, and occlusion may alter the coefficients in complex ways. Hence, such measures are useful only if the number of curves and the expected variations in them are small.

A plane curve is generally a multivalued function of two variables [for example, $f(x, y) = 0$]. Analytical approximation of a curve is simplified by transformation to a related single-valued function. One such transformation is a new function $\theta(s)$, defined to be the rotation of the tangent at a point on a curve with the arclength, s, in comparison to the tangent at the starting point, as shown in Fig. 5-4. (An arbitrary starting point may be chosen for a closed curve.) Note that $\theta(0) = 0$ and $\theta(L) = -2\pi$, for a closed curve, where L is the arclength of the curve. A modified function given by Zahn and Roskies [10] is defined as follows:

$$\theta'(t) = \theta\left(\frac{Lt}{2\pi}\right) + t \qquad (5\text{-}3)$$

where t is in range $[0, 2\pi]$ and related to s by $t = (2\pi / L)\, s$.

$\theta'(t)$ is invariant to translation, rotation, and scaling of the curve. Analytic shape measures can now be obtained by approximating this function; a common approach is to expand in a Fourier series and use the lower-order coefficients (for example, see [10]). Curvature of the curve, $K(s)$, provides an alternative transformation. However, this function is ill-behaved for curves with corners.

Another transformation for a closed curve is a function $\phi(s)$, defined to be the angle made by line joining a point on the curve with its centroid, where s is the arclength as before (see Fig. 5-5). This transformation is convenient to use only if the resulting $\phi(s)$ is single-valued.

A curve can also be represented parametrically by two equations of the form $x = f(t)$, $y = g(t)$ and the two functions approximated directly. Again arclength s is a suitable parameter.

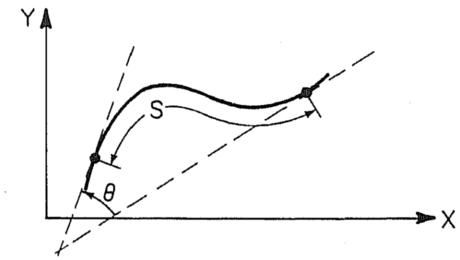

Figure 5-4: Arc length and rotation of the tangent of a curve

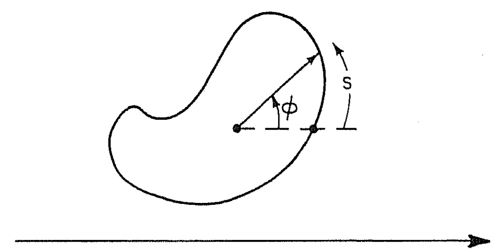

Figure 5-5: Polar transformation of a closed curve

5.3 AREA DESCRIPTIONS

The shape of a plane figure can, of course, be described by its enclosing boundary using the methods of the previous section. This section covers the techniques of describing a figure using the points in its interior and not just on the boundary. Such descriptions are likely to be more robust, as small changes in area can cause rather large changes in the boundary (imagine a jagged boundary rather than a smooth one).

5.3.1 Simple Shape Measures

Rough measures of the shape of a plane figure can be obtained simply from its area and perimeter. *Area*/(*perimeter*)^2 is a measure invariant with the size, position, and orientation of a figure. This measure is maximum for a circle and becomes smaller for elongated shapes. However, this ratio is not necessarily different for two different shapes (the reader may try to construct examples). A better measure for elongation is the ratio of major to minor axes of the minimal bounding rectangle of a figure, defined to be a rectangle completely enclosing the figure but not itself enclosed in any other such rectangle (see Fig. 5-6).

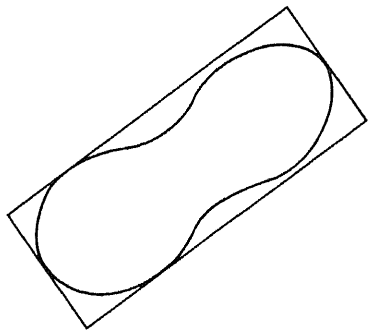

Figure 5-6: Minimum bounding rectangle

An improved approximation to the figure shape is by its *convex hull*, defined to be the minimal convex figure enclosing the given figure. The original figure is now described by the shape of the convex hull and by the number and the shapes of the concavities or the *concave deficiencies* in the figure (see Fig. 5-7). Qualitative shape measures can also be based on the topology of the figure and include the number of connected components and holes.

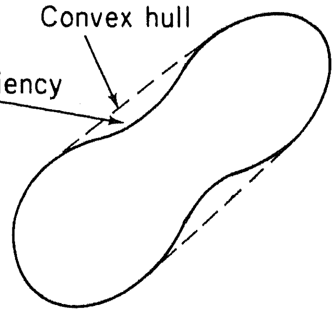

Figure 5-7: Convex hull of a figure

The simple shape measures described here can be expected to differentiate only among a small number of widely different shapes and do not account for the changes caused by perspective and occlusion, if the figure is a projection of a 3-D object.

5.3.2 Analytical Measures

As in the case of line descriptions, the coefficients obtained by expansion or approximation of a figure in terms of some basis functions, such as, 2-D Fourier series, may be used as analytical shape measures. For some basis functions, it is possible to combine the coefficients to obtain invariance to scale, position, and rotation (but not to perspective or occlusion changes). These methods have been extensively applied to recognition in limited domains, mainly character recognition of the English alphabet. Use of approximations by moments is considered in detail below.

The *pq*th order moment of given figure, *R*, is defined to be

$$m_{pq} = \sum_{x,y \in R} x^p y^q \qquad (5\text{-}4)$$

where (x, y) is a point in or on the boundary of R. The zero order moment, m_{00}, is simply the number of points in the figure (that is, the area) and m_{10} and m_{01} give the position of the centroid. The moments can be made invariant to position of the figure by translating the origin to the centroid and defining new coefficients as

$$\mu_{pq} = \sum\sum_{x,y \in R} (x - \bar{x})^p (y - \bar{y})^q \qquad (5\text{-}5)$$

where $\bar{x} = m_{10}/m_{00}$ and $\bar{y} = m_{01}/m_{00}$. Note that $\mu_{10} = \mu_{01} = 0$ and μ_{11}, μ_{02}, and μ_{20} are the usual moments of inertia.

Invariance to scale can be obtained by using

$$\mu'_{pq} = \frac{\mu_{pq}}{\mu_{00}^{((p+q)/2)+1}} \qquad (5\text{-}6)$$

Invariance to rotation can be obtained by rotating the coordinate axes by an angle θ, where

$$\tan 2\theta = \frac{2\mu_{11}}{\mu_{20} - \mu_{02}} \qquad (5\text{-}7)$$

A set of functions of the second- and third-order moments, invariant to rotation and reflection, are given below (from [11]). (The last M_7 is invariant to reflection in magnitude only.)

$$M_1 = (\mu_{20} + \mu_{02})$$
$$M_2 = (\mu_{20} - \mu_{02})^2 + 4\mu^2_{11}$$
$$M_3 = (\mu_{30} - 3\mu_{12})^2 + (3\mu_{21} - \mu_{03})^2$$
$$M_4 = (\mu_{30} + \mu_{12})^2 + (\mu_{21} + \mu_{03})^2 \qquad (5\text{-}8)$$
$$M_5 = (\mu_{30} - 3\mu_{12})(\mu_{30} + \mu_{12}) \cdot [(\mu_{30} + \mu_{12})^2 - 3(\mu_{21} + \mu_{03})^2] + (3\mu_{21} - \mu_{03})(\mu_{21} + \mu_{03}) \cdot [3(\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2]$$
$$M_6 = (\mu_{20} - \mu_{02})[(\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2] + 4\mu_{11}(\mu_{30} + \mu_{12})(\mu_{21} + \mu_{03})$$
$$M_7 = (3\mu_{21} - \mu_{03})(\mu_{30} + \mu_{12}) \cdot [(\mu_{30} + \mu_{12})^2 - 3(\mu_{21} + \mu_{03})^2] - (\mu_{30} - 3\mu_{12})(\mu_{21} + \mu_{03}) \cdot [3(\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2]$$

Dudani et al. used these moment functions for the recognition of different aircraft shapes [12]. The perspective changes were accommodated by storing the seven moment values for each separate view every few degrees apart.

5.3.3 A Medial Axis Transform

An intuitive description of area shape is by a curve in the "middle" of the figure and the varying width of the figure along this curve. Blum formalized this notion by defining a *medial axis transform* [13]. The transform is most easily explained by imagining the given figure to consist of flammable grass and a fire started at its perimeter. Those points in the interior at which two fire fronts meet and extinguish each other are the desired points of a medial axis (see Fig. 5-8). The time of fire extinction, for a given velocity, gives the "width" or the distance from the axis of the figure at that point. From such an axis and distance function, the original figure can be reconstructed accurately.

Two equivalent and precise definitions for the transform are as follows:

1. The points on the medial axis are the centers of the maximal circular neighborhoods totally contained in the figure—that is, those neighborhoods not entirely contained within any other circular neighborhood. The radii of the circles give the distance

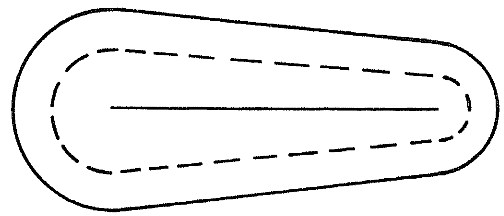

Figure 5-8: Medial axes by Blum transform

function. In Fig. 5-9, the points A and B are centers of such neighborhoods, but not the points C and D.

2. For each point x in the interior of the figure, let a quench function, q, be defined to be

$$q(x, \mathbf{B}) = \min(d(x, y)), \; y \text{ in } \mathbf{B}$$

where y is another point on the boundary $\mathbf{B}$, and d is the Euclidean distance between x and y. Each point x in the interior for which $q(x, \mathbf{B})$ is nonsingular (that is, two points on the boundary are at equal minimum distance from x) belongs to the medial axis, and the quench function is also the desired distance function.

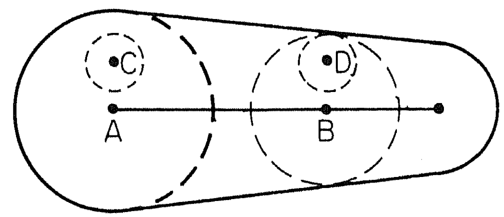

Figure 5-9: Maximal circular neighborhoods

Blum's medial axis transform gives intuitively agreeable descriptions for smooth objects but suffers from some deficiencies. The axes for a rectangle, for example, are as shown in Fig. 5-10, rather than a single line along the major axis. Even more serious is the effect of small changes in the boundary on the derived descriptions, as shown by the axes of a rectangle with a small notch in Fig. 5-11. These problems can be partially alleviatedby removing those points along the axis where the speed of quenching is high—that is, the points with large values of

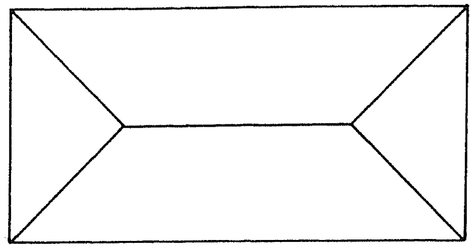

Figure 5-10: Medial axes of a rectangle

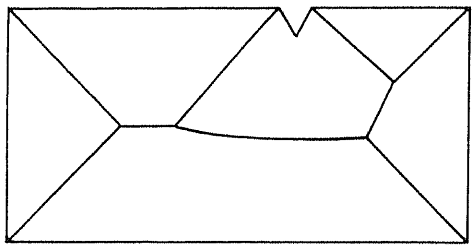

Figure 5-11: Medial axes of a rectangle with a notch (after Agin [23])

$\Delta s/\Delta q$, where s is the distance along the axis. Computational algorithms for the Blum medial axis transform and its modifications may be found in [14-16]. Another medial-axis-like representation using *generalized cones* that avoids these problems, and which also applies to 3-D objects, is described in Section 5.4.

If the figure is subject to changes caused by occlusion and the representation is to meet the other requirements of Section 5.1, it should be described by segmentation into simpler shapes. One criterion is to segment into parts that are convex. Another technique of segmentation into simple generalized cones is described in the next section. Other descriptions based on detecting symmetry may be found in [17].

5.4 DESCRIPTIONS OF 3-D OBJECTS

Shape description of 3-D objects, may be in terms of their exterior surfaces or the volume enclosed by these surfaces. (Holes can be described as negative volumes.) Volume descriptions are likely to be more robust, as relatively large surface changes, such as a nick or a

fold, may result only in small changes in volume and the perceived shape. For certain objects, however, primarily those constructed of thin sheetlike material, such as clothing or pressed metal objects, surface descriptions may be more natural.

An additional difficulty with 3-D object descriptions is that the 3-D surfaces or volumes need to be inferred from a 2-D image. If the 3-D positions of the visible surfaces are available, partial surface descriptions can be derived directly, but volume descriptions still require inferences about the invisible surfaces.

Surface description, given 3-D positions, can be viewed as a problem of approximation and segmentation by simpler surfaces, such as planes or multi-dimensional splines. A method using surface patches, called Coons surfaces, is given in [18]. Surface descriptions are common for computer graphics applications, and details may be found in [19, 20]. Volume descriptions, given 3-D positions of *all* surface points, could proceed by analogous volume approximation techniques, say by polyhedra. We will concentrate on the problem of volume descriptions from a 2-D image, with or without the 3-D positions of the visible points, as it is most common in normal perception.

5.4.1 Generalized Cones

A *generalized cone* is a volume generated by sweeping an arbitrarily shaped planar figure, called a cross section, along an arbitrary 3-D space curve called the axis. The axis passes through the centroids of the cross sections and is normal to them. The size and the shape of the cross section may change along the axis, as-specified by a cross section function. These cones may be viewed as a generalization of regular, right circular cones which are generated by sweeping a circular cross section along a straight line axis, the cross section function being a linear scale change. Generalized cones were introduced by Binford [21].

While a single generalized cone can describe an arbitrary volume, complex shapes are more naturally described by segmentation into a number of simpler generalized cones. For example, the screwdriver shown in Fig. 5-12 is described by four generalized cones, one corresponding to the blade with a varying rectangular cross section, a stem with constant circular cross section and the handle consisting of two generalized cones, as shown. Criteria for simplicity of a generalized cone may be no abrupt change in the size or shape of the cross section, or in the direction of its axis. Techniques for segmentation are discussed later.

Generalized cones give simple descriptions for many natural shapes, such as, animals and tree trunks and also manufactured objects.

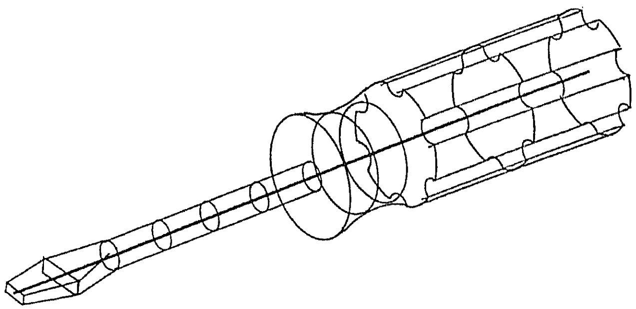

Figure 5-12: Generalized cone representation of a screwdriver (from Agin [23])

Manufacturing processes of extrusion and turning have natural correspondences to generalized cones. Fastening of parts is like cutting and pasting of generalized cones. Generalized cones are not well suited to descriptions of non-elongated objects, such as, spheres which have no preferred axis, or objects of arbitrarily deformed surfaces enclosing little volume.

Simplified generalized cone descriptions still retain many of the important shape properties. Simply a "stick figure" composed of cone axes is adequate for gross shape recognition, without any knowledge of the cross sections along it. For example, humans have little difficulty recognizing the shapes of the figure shown as a stick figure in Fig. 5-13. More subtle distinctions, say between animals of similar skeletonal shape, may require detailed knowledge of the cross sections. The cross sections themselves may be represented by 2-D generalized cones; here the axis is a curve in a plane and the cross sections simply straight-line segments normal to the axis.

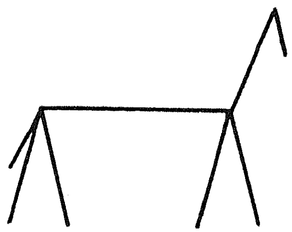

Figure 5-13: A stick figure

The structure of the axes offers some invariance to perspective

changes. The relative lengths of the axes change, and not all of the parts of an object are always visible, but the connectivity of axes remains largely unchanged, except for widely different views, such as a horse viewed from the side, from the front, or the top. In such cases, multiple models for an object need to be stored, but the number of such views is small.

Marr argues that a generalized cone interpretation may be implicit in our normal perception of 2-D line drawings [22]. Two reasonable assumptions in perceiving 2-D contours as 3-D objects are that the contiguous positions of contours arise from contiguous parts of the viewed surfaces (that is, there are no invisible occluding edges) and that the convexities and concavities correspond to real properties of the viewed surfaces. These two assumptions are shown to be equivalent to assuming the viewed surfaces to be generalized cones with fixed-shape cross sections.

The axis and cross-section representation of generalized cones has similarities with Blum's medial axis transform, for 2-D figures. However, the 3-D generalization of the Blum transform, requiring axes to consist of centers of maximal spherical neighborhoods, yields axes that are surfaces rather than curves. Also, small boundary irregularities, have only small effects on generalized cone representations; for example, the notch in the rectangle of Fig. 5-11 causes only a small dip in the generalized cone axis and a small local change in the cross-section width.

First implementations of the generalized cone representation were by Agin and Binford [23, 24], and Nevatia and Binford [25, 26]. Marr and Nishihara also discuss their use as general shape descriptors [27]. Hollerbach used them to describe pottery patterns for anthropological descriptions [28], and Soroka has applied them to biological cell descriptions [29].

5.4.2 Computation of Generalized Cones

The generalized cone representation is not a transform representation, and many alternative descriptions are possible for the same input. We need to choose one or more preferred descriptions among the alternatives. A unique choice is not necessary, and a small number of multiple descriptions may be carried to higher levels for recognition. Some alternative techniques of computing generalized cone descriptions are given below.

Fitting surface data. Optimal generalized cones can be fitted, given 3-D positions of visible surface, and restrictions on the axis and the cross section shapes. A simple iterative solution is possible for cross sections of known shape. Consider a straight circular cylinder. Initially, the orientations of the axes and the cross sections are unknown. Choosing an arbitrary orientation, elliptical cross sections can be fitted to the visible surface. An axis passing through the centroid of these cross sections is not necessarily normal to them. New cross sections can now be constructed normal to the derived axis and the process repeated until only small changes are observed. For straight, circular cylinders and cones the process converges rapidly. Convergence for arbitrary shapes is unclear. This technique was used by Agin and Binford, assuming cross sections could be approximated by ellipses [24].

Using object boundaries. 2-D cones can be computed from the object boundaries. If the 2-D contours are the projection of a 3-D object, the computed cones are the projections of the desired 3-D cones. A brief description of a technique developed by Nevatia and Binford is given below; details may be found in [25, 26]. (Another technique using concavities in the boundary is given in [27].)

In the Nevatia-Binford technique; local cones with straight axes are computed first; these cones are then extended by allowing smooth curving of the axes, and preferred cones are chosen among the various alternatives. Structured descriptions are generated from the properties of the computed cones and their connectivity.

The local cones are computed simply by choosing various directions (say eight equally spaced directions) and examining if any parts of the boundary fit the requirements of the generalized cones. These requirements are that the chosen axis direction pass through midpoints of the cross sections, defined by lines perpendicular to the axis, and that the width of the cross sections be continuous. Figure 5-14 shows an object and the local cone axes in eight directions (note that the object is rotated for the eight views).

These local cones are then extended by extrapolating the axes of the local cones and constructing new cross sections. The midpoint of the new cross section defines a new point on the axis (see Fig. 5-15). This process allows the axis to curve smoothly. Extension of cones terminates by defined criteria of axis and cross-section continuity.

This process may result in the same parts of an object being described by more than one cone. Preferred descriptions are selected by choosing elongated and cylindrical descriptions over less elongated and more conical descriptions.

Figure 5-16 shows the cones computed from the boundary of a doll using the above technique (from [26]). Note that one of the legs is segmented into two cones (*P5* and *P6*), which are merged at this level to generate an alternative description.

The segmented generalized cones, representing parts or pieces of an object, and their connectivity relations constitute structured

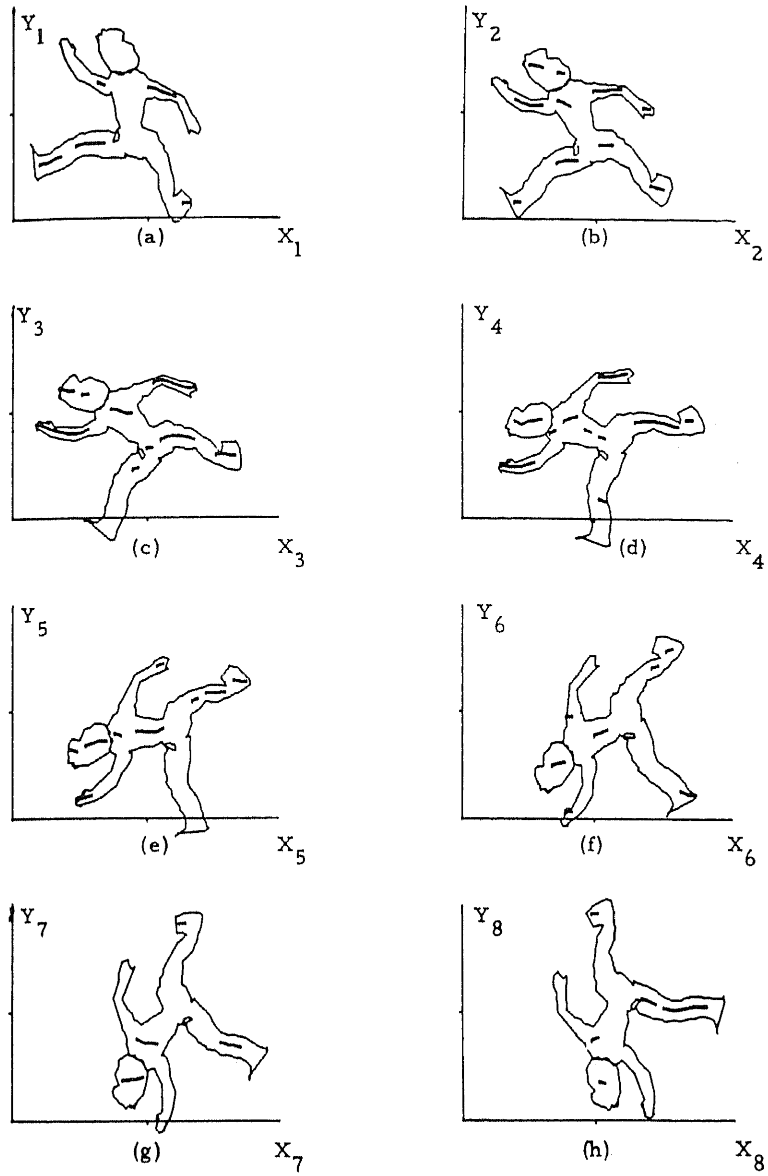

Figure 5-14: Axes of local cones in eight directions for a doll (from Nevatia [26])

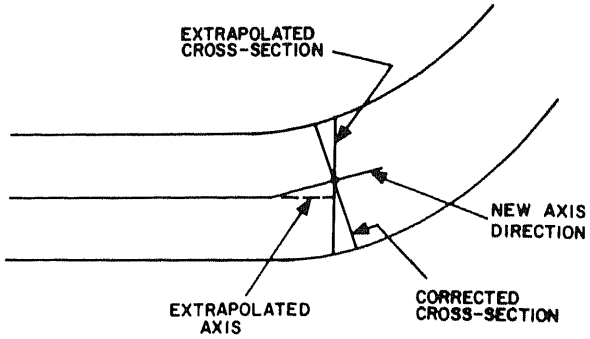

Figure 5-15: Extension of a local cone

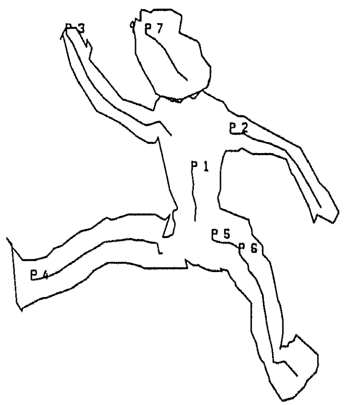

Figure 5-16: Selected cones in a doll

descriptions of the object. This description may be viewed as a graph with joints as nodes and pieces as arcs or vice versa. Figure 5-17 shows the graph corresponding to the segmentation of Fig. 5-16 (assuming *P5* and *P6* were merged).

The structured descriptions also include summary descriptions of the parts and their joints. The part descriptions may include the approximate axis shape, the length of axis to average cross-section width ratio, and approximation of the cross-section function. Joints are characterized by the parts connected to them and the interrelationships of these parts. Some joint types are shown schematically in Fig. 5-18. Additional descriptions may include properties relating to the whole structure - for example, bilateral symmetry and the axis of this symmetry. Recognition of objects from such descriptions is described in the next section.

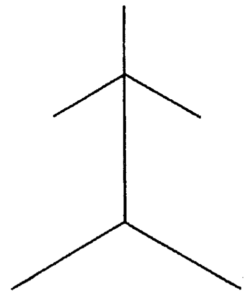

Figure 5-17: Graph representation of the doll axes

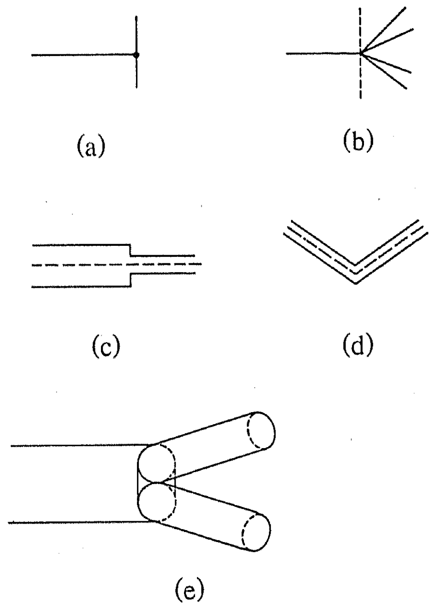

Figure 5-18: Some types of joints: (a) T, (b) fork, (c) neck, (d) elbow, (e) cross-section-conserving

5.5 RECOGNITION OF OBJECTS

Objects, or structures consisting of several objects, can be recognized by comparing their descriptions with the descriptions of models stored in a memory. The models may be acquired by storing machine descriptions of them from a previous encounter, by a directed *learning* sequence of a number of views, as in Winston's method of Section 4.4.2, or they may be simply supplied by a human operator.

If object descriptions are just a list of properties—that is, a feature vector—recognition can use standard mathematical pattern-recognition techniques (as in Chapter 2). For structured descriptions, more elaborate matching techniques are required. Additionally, it is undesirable to match a description with each stored model in a large memory, and *indexing* to select a suitable subclass without complete matching is needed.

5.5.1 Graph Matching

Structured descriptions may be viewed as graphs (or networks). We are interested in evaluating the similarity of two graphs. Some measures of similarity are introduced below.

Let a graph **G**: <**N**, **P**, **R**> be defined to consist of a set of nodes **N** (representing parts of an object), a set of properties **P** of these nodes, and a set of relations **R** between the nodes. Given two graphs **G**: <**N**, **P**, **R**> and **G′**: <**N′**, **P′**, **R′**>, nodes n in **N** and n' in **N′** are said to form an *assignment* if and only if $P(n)$, property of node n, is similar to $P'(n')$, property of node n', by a given similarity measure. Two assignments (n_1, n_1') and (n_2, n_2') are said to be *compatible* if $r(n_1, n_2) = r'(n_1', n_2')$, for all relations r in **R** and r' in **R′** (relations are assumed to be binary).

The two graphs **G** and **G′** are said to be *isomorphic* if there exists a one-to-one assignment of nodes in **G** and **G′** such that all assignments are mutually compatible. [We also require $P(n) = P'(n')$ if (n, n') is an assignment]. **G** and **G′** are said to be subisomorphic if a subgraph of **G** is isomorphic to a subgraph of **G′**.

Graph isomorphism could be determined by an exhaustive search of all assignments and a test of their mutual compatibilities. Subgraph isomorphism could be determined by computing the isomorphism of all subgraphs. A more efficient technique is given in [30]. (Note that the graph-isomorphism problem belongs to the computational complexity class of *NP*-complete problems.)

For application to object recognition the isomorphism or sub-isomorphism measures are likely to be too stringent, as errors are

made at the various levels of the decription processes. Some applications of these techniques to object recognition are given in [31].

A less stringent measure is that of determining *maximal cliques.* From the two graphs to be matched, let us define a new graph, called the match graph, such that the nodes of the match graph consist of an assignment of a pair of nodes from **G** and **G′**, and an arc exists between two nodes of the match graph if the two corresponding assignments are compatible. A clique (of the graphs **G** and **G′**) is a totally connected subgraph of the match graph. A clique is *maximal* if it is not included in any other clique.

Figures 5-19(a) and (b) show two graphs to be matched. Three types of nodes are present, those marked with light circles, dark circles, and a square. Only like nodes can be matched. The two graphs are quite similar and would be isomorphic if nodes B and D were of the light circles type rather than the rectangle type. However, the only isomorphic connected subgraphs are the isolated nodes. Maximal cliques can help find larger compatible matching sub-structures. Figure 5-20 shows the matching graph for the two graphs of Fig. 5-19. Two cliques are present, one with the match $((A, 1), (C, 3), (E, 5))$ and the other $((A, 5), (C, 3), (E, 1))$.

Figure 5-19: Two graphs to be matched

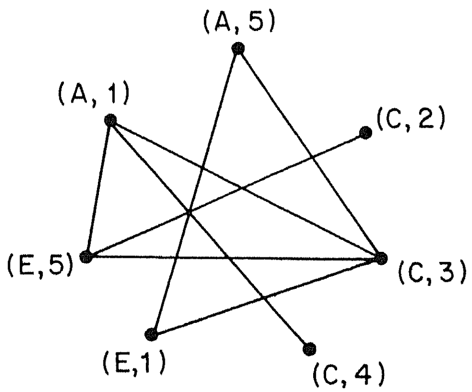

Figure 5-20: Match graph for Fig. 5-19 graphs

A simple procedure to find maximal cliques is given below (from [32]). A function clique (X, Y) generates the set of all cliques that include nodes in clique X and are included in the set Y. Cliques $(0, N)$, N being the set of all nodes, will find all cliques. It is defined as:

cliques (X, Y) := *If* no node in $Y - X$ is connected
to all elements of X
then $\{X\}$
else cliques $(X \cup \{y\}, Y) \cup$ cliques $(X, Y - \{y\})$
where y is such a node

A clique of a minimum size k can be found by stopping the recursion if the size of X plus the number of nodes in Y-X connected to all of X is less than k. To find maximal cliques, start with large k and reduce k by one until some cliques are found.

In a worst case, the maximal-clique computation can be expensive, and the number of maximal cliques can be as large as $(n/2)^{n/2}$, n being the number of nodes. The maximal-clique technique can be modified to include nonbinary relationships, by defining a modified compatibility criterion.

5.5.2 Relaxation Labeling

A labeling problem can be defined to be assignment of a set of labels to a set of nodes (or units) such that the label assignments are consistent according to given constraints. Such labeling has many applications and includes the problem of graph matching (the labels are now nodes of the other graph).

Let **N** be the set of nodes to be labeled and **L** be the set of allowed labels. To each node n_i we wish to assign a set of labels $\mathbf{L}_i$, such that $\mathbf{L}_i$ is a subset of **L**, and the labels are consistent according to the given constraints. For unambiguous cases, each set $\mathbf{L}_i$ contains one element only. Simplest constraints are *unary*, restricting the labels that may be assigned to a certain node, without consideration of the other nodes in the network. *Binary* constraints specify relations between labels of a pair of nodes. A set of labels $\mathbf{L}_i$ for node n_i may be said to be consistent with a set of labels $\mathbf{L}_j$ for node n_j if each label in $\mathbf{L}_i$ is consistent with at least one label in $\mathbf{L}_j$ and vice versa. Such consistency is called *arc consistency.*

In general, the constraints are n-ary, and arc consistency may not result in global consistency. Figure 5-21 shows an example (from [32]), where the unary constraints are that each node be labeled red or green, and that adjacent nodes be of a different color. For each assignment of

red or green to one node, we can assign a consistent label to the neighboring nodes, but we can not satisfy the global constraint simultaneously at the three nodes.

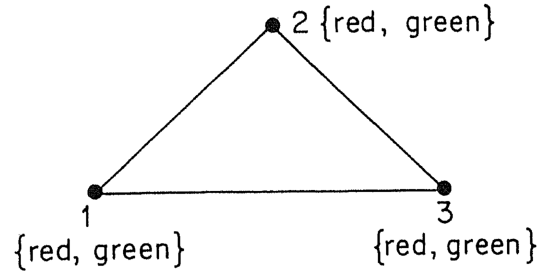

Figure 5-21: An arc-consistent but globally inconsistent labeling

A more powerful constraint is that of *path consistency.* Two nodes n_i and n_j with labels l_k and l_l are path inconsistent, if there exists a path in the network from n_i to n_j, such that there is no set of labels, one for each node along the path, that is simultaneously consistent (in a binary way) with the labels l_k and l_l at the two ends. Note that the network of Fig. 5-21 is not path consistent.

However, path consistency also does not guarantee global consistency. The problem of global consistency is computationally expensive (*NP*-complete), though efficient solutions may be achieved for some networks. In the following we will consider only arc consistency, as it is often helpful in reducing the number of alternatives; more complex labeling methods may be found in [33-36].

Rosenfeld, Hummel, and Zucker have described an iterative *relaxation* scheme for computing arc-consistent labels [37]. Initially, all labels satisfying unary constraints are assigned to each node. At any iteration of the algorithm, those labels of a node that are not arc consistent with the other nodes are removed. Note that removal of some labels may create new inconsistencies that are removed at the next iteration. It can be shown to converge to a consistent labeling, if one exists. It is a generalization of the Waltz labeling method described in Chapter 4 and can be implemented as a parallel algorithm.

Rosenfeld, Hummel, and Zucker have also described a probabilistic or stochastic version of the relaxation labeling, in an effort to account for the variations that are common in image descriptions. A weight or probability is associated with each label assigned to each node. The compatibility between the labels of two nodes is also given by a range of values. At each iteration of the algorithm, the probability of each assignment is updated, based on their compatibility with the neighboring assignments. Generally, the probability of a label should be increased if other high-probability labels are highly compatible with it. Let $P_i^t(k)$ be the probability of label l_k being assigned to node n_i, at the tth iteration, such that

$$\sum_k P_i^t(k) = 1 \qquad (5\text{-}9)$$

These probabilities may be updated based on labels of other nodes by the following expression (according to [38]):

$$P_i^{t+1}(k) = \frac{P_i^t(k)[1 + q_i^t(k)]}{\sum_k \{P_i^t(k) \cdot (1 + q_i^t(k))\}} \qquad (5\text{-}10)$$

where $q_i^t(k)$ gives the correction to the assignment probability and the denominator guarantees that the new probabilities still sum to one. $q_i^t(k)$ is defined by

$$q_i^t(k) = \sum_j d_{ij}[\sum_{k'} r_{ij}(k, k') \cdot P_j^t(k')] \qquad (5\text{-}11)$$

d_{ij} is a weight to determine the effect of node j on node i. $r_{ij}(k, k')$ is a measure of compatibility between the nodes i and j having labels k and k', respectively. Statistical correlation between the two labels has been suggested as one appropriate measure.

Hopefully, in an unambiguous case, the various assignment probabilities will converge to a value of 1 or 0. However, the convergence of this process is not guaranteed for the probabilistic case, as it was for the discrete case. Performance of the algorithm is likely to be highly dependent on the choices of the weights and the compatibility measures.

Experiments with different compatibility measures and updating functions are given in [37, 38]. An approach, that seeks to maximize a certain function, and thus is converging by definition, is described in [39]; an application to labeling of aerial images is given in [40]. Another approach to using nonbinary weights for relaxation may be found in [41].

5.5.3 Multilevel Matching

Graph-matching and scene-labeling techniques discussed above are general. However, they do not provide satisfactory descriptions of similarities and differences. Use of numerical weights, combining unrelated features, such as , color and size, may be meaningless. An alternative is multilevel matching. Here, the result of matching two descriptions is itself a description of their similarities and differences, as, for example, in Winston's learning program (Chapter 4). Of course, eventually a decision for recognition must be made, but the results of matching can now be examined with more context; for example, the particular model may have associated information as to the relative importance of color and size. If matching with two models yields similar differences, the scene may now be reexamined to find specific finer details. Marr has argued that deferrng of decisions, by carrying along additional information, until more contextual information is available is an important organization rule for visual processes; he has called this the "principle of least commitment" [42].

Nevatia and Binford [25, 26] used such an approach for recognizing objects by matching generalized cone descriptions, as in Section 5.4.1. The matching was facilitated by marking the wide pieces (the body and the head) as being *distinguished* and matching them to other distinguished pieces only. Alternative matches of parts, each consistent with the connectivity relations, produces a difference description. Some matches are clearly inferior to others, such as those containing more pieces in the observed object than the model, and are discarded.

In some cases two models (for example, a doll and a horse) may have similar connectivity. In such cases the models may be distinguished by properties of individual parts. In general, a more detailed analysis may be required, such as looking at the limb extremities in the above example.

When the number of models is large, matching with each model is prohibitive, and indexing into memory to retrieve only a small number of likely models is required. Indexing may be by use of context, such as the knowledge of observer location and expected objects in the environment. However, humans are able to quickly perceive objects out of context as well, as in a collage of unrelated objects, and indexing in such cases seems to be based solely on object descriptions.

An indexing procedure must accommodate the usual changes in object descriptions due to different viewing conditions and also the variability caused by the description processes themselves. Indexing is largely ignored, as most systems usually deal with a small number of objects only. An indexing procedure using the properties of distinguished pieces in a generalize cone description is described in [25, 26]. Variability in the descriptions is accommodated by indexing with the observed descriptors as well as by perturbing these descriptions according to expected changes.

5.6 SUMMARY

In this chapter we have considered analysis of nonpolyhedral objects and concentrated mostly on shape analysis. The described techniques should be adequate for a wide range of applications, if the number of objects is small and perfect boundaries are available. However, as will be seen in the next few chapters, perfect boundaries are difficult to extract from a single image of a 3-D scene, and the description mechanisms need to be modified to work with imperfect data. Some examples of systems that operate on real data are given in Chapter 10.

REFERENCES

[1] D. A. Huffman, "Curvature and Creases: A Primer on Paper," *IEEE Transactions on Computers*, Vol. 25, 1976, pp. 1010-1019.

[2] R. T. Chien, and Y. H. Chang, "Recognition of Curved Objects and Object Assemblies," *Proceedings of the Second International Joint Conference on Pattern Recognition*, Copenhagen, 1974, pp. 496-510.

[3] H. Freeman, "On the Encoding of Arbitrary Geometric Configurations," *IRE Transactions on Electronic Computers*, June 1961, pp. 260-268.

[4] T. Pavlidis, *Structural Pattern Recognition*, Springer-Verlag, New York, 1977.

[5] T. Pavlidis and S. L. Horowitz, "Segmentation of Plane Curves," *IEEE Transactions on Computers*, Vol. 23, No. 8, August 1974, pp. 860-870.

[6] U. Ramer, "An Iterative Procedure for the Polygonal Approximation of Plane Curves," *Computer Graphics and Image Processing*, Vol. 1, 1972, pp. 244-256.

[7] L. Davis, "Shape: Angles and Sides," *IEEE Transactions on Computers*, March 1977, pp. 236-242.

[8] C. de Boor, "On Calculating with B-Splines," *Journal of Approximation Theory*, Vol. 6, 1972, pp. 50-62.

[9] W. S. Rubkowski and A. Rosenfeld, "A Comparison of Corner-Detection Techniques for Chain-Coded Curves,"

University of Maryland, Computer Science Department, Report TR-623, January 1978.

[10] C. T. Zahn and R. Z. Roskies, "Fourier Descriptions for Plane Closed Curves," *IEEE Transactions on Computers*, Vol. 21, March 1972, pp. 269-281.

[11] M. K. Hu, "Visual Pattern Recognition by Moment Invariants," *IRE Transactions on Information Theory*, February 1962, pp. 179-187.

[12] S. A. Dudani, K.J. Breeding, R. B. McGhee, "Aircraft Identification by Moment Invariants," *IEEE Transactions on Computers*, Vol. 26, No. 1, January 1977, pp. 39-46.

[13] H. Blum, "A Transformation for Extracting New Descriptions of Shape," in *Symposium on Models for Perception of Speech and Visual Form*, W. Wathen-Dunn (ed.), MIT Press, Cambridge, Mass., 1967, pp. 362-380.

[14] A. Rosenfeld and J. L. Pfalz, "Distance Functions on Digital Pictures," *Pattern Recognition*, Vol. 1, July 1968, pp. 33-62.

[15] U. Montanari, "A Method for Obtaining Skeletons Using a Quasi-Euclidean Distance," *Journal of the ACM*, Vol. 15, October 1968, pp. 600-624.

[16] U. Montanari, "Continuous Skeletons from Digitized Images," *Journal of the ACM*, Vol. 16, October 1969, pp. 534-549.

[17] L. S. Davis, "Understanding Shape: Symmetry," University of Maryland, Computer Science Department, Report TR-441, February 1976.

[18] S. A. Coons, "Surfaces for Computer Aided Design of Space Forms," MIT Project MAC, MAC-TR-41, June 1967 (AD 663504).

[19] W. M. Newman and R. F. Sproull, *Principles of Interactive Computer Graphics*, 2d ed., McGraw-Hill, New York, 1979.

[20] D. F. Rogers and J. A. Adams, *Mathematical Elements for Computer Graphics*, McGraw-Hill, New York, 1976.

[21] T. O. Binford, "Visual Perception by a Computer," *IEEE Conference on Systems and Controls*, Miami, Florida, December 1971.

[22] D. Marr, "Analysis of Occluding Contour," *Proceedings Royal Society of London*, B200, 1977, pp. 441-475.

[23] G. J. Agin, "Representation and Description of Curved Objects," Stanford Artificial Intelligence Laboratory Memo AIM-173, Ph.D. Thesis (1972).

[24] G. J. Agin and T. O. Binford, "Computer Description of Curved Objects," *IEEE Transactions on Computers*, Vol. 25, April 1976, pp. 439-440.

[25] R. Nevatia and T.O. Binford, "Description and Recognition of Curved Objects," *Artificial Intelligence*, Vol. 8, No. 1, February 1977, pp. 77-98.

[26] R. Nevatia, *Computer Analysis of Scenes of 3-D Curved Objects*, Birkhauser-Verlag, Basel, Switzerland, 1976.

[27] D. Marr and K. Nishihara, "Representation and Recognition of the Spatial Organization of Three-Dimensional Shapes," *Proceedings Royal Society of London*, B200, 1977, pp. 269-294.

[28] J. M. Hollerbach, "Hierarchical Shape Representation of Objects by Selection and Modification of Prototypes," MIT AI-TR-346, November 1975.

[29] B. I. Soroka, "Generalized Cones from Serial Sections," *Computer Graphics and Image Processing*, Vol. 15, No. 2, February 1981, pp. 154-166.

[30] J. R. Ullman, "An Algorithm for Subgraph Isomorphism," *Journal of the ACM*, Vol. 23, January 1976, pp. 31-42.

[31] H. G. Barrow, A. P. Ambler, and R. M. Burstall, "Some Techniques for Recognizing Structure in Pictures," in S. Watanabe (ed.), *Frontiers of Pattern Recognition*, Academic Press, New York, 1972, pp. 1-29.

[32] A. P. Ambler, H. G. Barrow, C. M. Brown, R. M. Burstall, and R. J. Popplestone, "A Versatile Computer Controlled Assembly System," *Proceedings of the Third International Joint Conference Artificial Intelligence*, Stanford, California, August 1973, pp. 298-307.

[33] E. C. Freuder, "Synthesizing Constraint Expressions," *Communications of the ACM*, Vol. 21, 1978, pp. 958-966.

[34] U. Montanari, "Networks of Constraints: Fundamental Properties and Applications to Picture Processing," *Information Science*, Vol. 7, No. 2, April 1974, pp. 95-132.

[35] A. K. Mackworth, "Consistency in Networks of Relations," *Artificial Intelligence*, Vol. 8, No. 1, 1977, pp. 99-118.

[36] R. M. Haralick and L. G. Shapiro, "The Consistent Labeling Problem: Part 1," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 1, No. 2, April 1979, pp. 173-184.

[37] A. Rosenfeld, R. A. Hummel, and S. W. Zucker, "Scene Labeling by Relaxation Operations," *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 6, No. 6, June 1976, pp. 420-453.

[38] L. Kitchen, "Relaxation Applied to Matching Quantitative Relational Structures," *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 10, No. 2, February 1980, pp. 96-101.

[39] O. D. Faugeras and M. Berthod, "Scene Labeling: An Optimization Approach," *Proceedings of IEEE Conference on

Pattern Recognition and Image Processing, Chicago, August 1979, pp. 318-326.

[40] O. D. Faugeras and K. Price, "Semantic Description of Aerial Images Using Stochastic Labeling," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, November 1981.

[41] J. M. Tenenbaum and H. G. Barrow, "MSYS: A System for Reasoning about Scenes," SRI Technical Note 121, March 1976.

[42] D. Marr, "Early Processing of Visual Data," *Philosophical Transactions of the Royal Society of London*, B275, 1976, pp. 483-524.