7

EDGE AND CURVE DETECTION

Detection of object boundaries is an important part of the perception process. There is much psychological evidence, such as in our ability to understand cartoons, that boundaries are sufficient for perception of a broad class of objects. In this chapter, techniques for detecting boundaries by aggregating evidence from local discontinuities are described. A different approach will be discussed in the next chapter, that of finding areas in an image over which some attributes are constant.

Edge-detection techniques aim to find local discontinuities in some image attribute, such as intensity or color. These discontinuities are of interest because they are likely to occur at the boundaries of objects. But, local edges may also occur due to variations in surface characteristics of an object, changes in illumination and shadows, and the like. An important process of perception is to organize the local edges into aggregates that lead to a scene segmentation. Thus, the process of edge detection may be viewed as one stage of abstracting descriptions from the image data.

7.1 EDGE DETECTION

We first concentrate on the detection of local edge elements. An edge is said to occur at a point in the image if some image attribute changes in value discontinuously at that point. Here we will discuss intensity edges; edges in other attributes can be defined similarly.

An ideal edge, in one dimension, may be viewed as a step change in intensity. In real signals, the step change is likely to be mixed with "noise" caused by sensor, surface, or illumination variations, as shown schematically in Fig. 7-1. In two dimensions, the ideal step occurs along a line of certain length, the intensity values on the two sides of the line being different. Also, edges of interest are not necessarily limited to step edges.

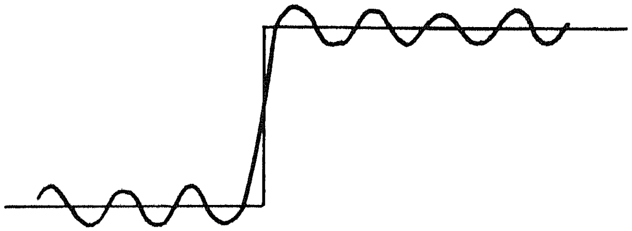


Figure 7-1: A one-dimensional edge

Detection of ideal step edges, without noise, would be simple. In real images, with noise and surface imperfections, a compromise must be achieved between maximizing the detection of the desired edges and minimizing the detection of the undesired noise edges. (Some of the "undesired" edges are due to legitimate surface variations, such as texture, and the local edge detection process is not intended to discriminate against them.)

The literature on techniques of edge detection is rather large. Only the prominent classes of techniques are discussed below. Good surveys of edge-detection techniques are also given in [1, 2].

7.1.1 Edge Enhancement and Differentiation

If an operator enhances the edges in a picture—that is, it outputs higher values for points at the desired edges than for the other points, then edge-detection can be performed by simple thresholding of the enhanced picture. While enhancement can be by any general operator, linear filtering techniques, such as Wiener filtering applied uniformly to the entire image, are often suggested. The difficulty with the use of a

X

global filtering technique is in the choice of proper enhancing functions. As objects in a given picture may be of different sizes and at different ranges, it is unlikely that a uniform filter function for the entire picture is appropriate.

Differentiation may also be viewed as a local edge enhancement operator. For application to two-dimensional images, an appropriate derivative operation is that of the gradient. Robert's gradient operator, using a 2-by-2 neighborhood, was described in Section 3.1.1 [3]. Better immunity to noise is achieved if a larger neighborhood is used for approximating the magnitude of the gradient.

Consider a 3-by-3 neighborhood of a certain pixel, say (i, j), with intensity values as shown in Fig. 7-2. We can define the magnitude of the gradient by

$$S = \sqrt{S_x^2 + S_y^2} \qquad (7\text{-}1)$$

where

$$S_x = (a_2 + ca_3 + a_4) - (a_0 + ca_7 + a_6) \qquad (7\text{-}2)$$

$$S_y = (a_6 + ca_5 + a_4) - (a_0 + ca_1 + \underset{a_2}{a_7}) \qquad (7\text{-}3)$$

and c is constant. c was chosen to be 1 in an edge detector described by Prewitt [4] and 2 by Sobel. (No published source is available for Sobel operator, but the described operator is generally associated with this name.)

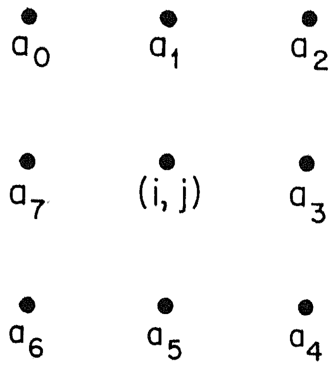


Figure 7-2: A pixel and its eight neighbors

The direction of the edge, θ, in these operations is given by

$$\theta = \tan^{-1}\left(\frac{S_x}{S_y}\right) \qquad (7\text{-}4)$$

Note that these operators are not perfectly isotropic; that is, edges of the same strength but differing directions give different edge magnitude outputs.

7.1.2 Edge Fitting

Another approach to edge detection is to have models of ideal edges and to determine how closely these models fit to a given image neighborhood. A popular analytical procedure due to Hueckel is described below.

Hueckel operator. A simplified model of an ideal edge used in this operator is as shown in Fig. 7-3, where an edge at an angle θ and distance r from the center separates two regions of brightness b and $b + h$. We desire to compute the ideal step function that matches best with the picture function by varying the position, orientation, and intensity values of the ideal step—that is, b, h, r, and θ of Fig. 7-3—comprising a vector ξ. Define the difference, N, between a neighborhood of an image and the ideal step by

$$N^2 = \sum_{\mathbf{x} \in \mathbf{R}} (A(\mathbf{x}) - S(\mathbf{x}, \xi))^2 \qquad (7\text{-}5)$$

where $A(\mathbf{x})$ is the image intensity at point $\mathbf{x}$ and $S(\mathbf{x}, \xi)$ is the intensity of the ideal step function for a chosen vector of parameters ξ. The sum is to be computed over the entire neighborhood, $\mathbf{R}$, of the ideal step. ξ is to be chosen such that N^2 is minimized. The decision as to the presence of an edge is based on N being small and the step height, h, being large.

To simplify this minimization computation, Hueckel suggested approximating the $A(.)$ and $S(.)$ functions by an orthogonal series of functions, say $H(.)$. The series chosen was a radial Fourier series where only the first nine functions shown in Fig. 7-4 were used. We now wish to minimize

$$N'^2 = \sum_{i=0}^{8} (a_i - s_i(\xi))^2 \qquad (7\text{-}6)$$

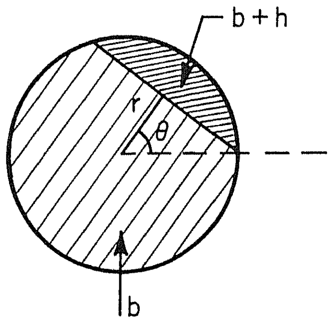


Figure 7-3: An ideal edge in a circular neighborhood

where a_i and s_i are the coefficients of expansion in the above series:

$$a_i = \sum_{\mathbf{x} \in \mathbf{R}} H_i(\mathbf{x}) \cdot A(\mathbf{x}) \tag{7-7}$$

$$s_i = \sum_{\mathbf{x} \in \mathbf{R}} H_i(\mathbf{x}) \cdot S(\mathbf{x}) \tag{7-8}$$

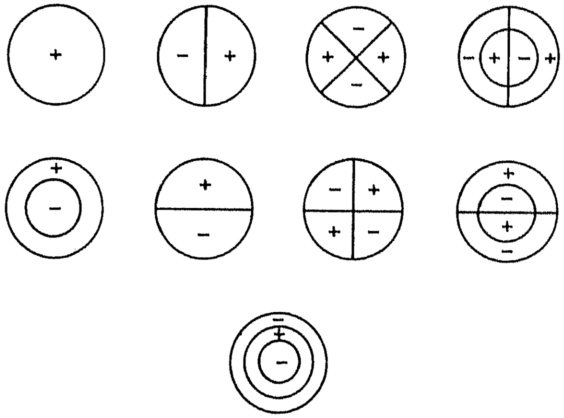


Figure 7-4: Basis functions for approximation in the Hueckel operator

Hueckel was successful in obtaining an analytical solution to determine ξ. The algebraic details are involved and may be found in [5] and [6].

Once ξ and N' are determined, the presence of an edge is determined by requiring step height, h, to be large and N' to be small. Actually, the required step height is allowed to be a function of N', a larger step being required if the fit is poor.

A generalized Hueckel operator uses an edge model with intensity profiles as shown in Fig. 7-5. This allows the operator to detect step as well as "line" edges (defined to be edges where $b_1 \ulcorner b_3$). Vector ξ now contains two additional parameters, one for an additional intensity value and another for edge width.

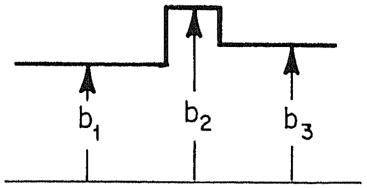


Figure 7-5: Intensity profile of a generalized edge

Mero and Vassy [7] and Nevatia [8] have described attempts to simplify this operator by using fewer terms for approximation of a step. However, such simplifications result in marked loss of performance in the presence of noise.

Hueckel's operator is claimed to be optimal under certain assumptions. However, further analysis has indicated some deficiencies in his optimization process. One is that the optimal analysis was done for continuous domain and does not necessarily apply to the digital approximations [9]. Abdou [9] and Shaw [10] have described techniques of direct edge fitting in the digital image domain without resorting to any approximations.

7.1.3 Edge Detection by Template Matching

A straightforward approach is to detect edges by matching with templates of desired ideal edges, also called *edge masks*. Two 3-by-3 edge masks, corresponding to vertical and diagonal edges, are shown in Fig. 7-6. Eight such masks were used in an edge detector described by Prewitt [4]. At each pixel in the image, a match with each of the eight masks is computed; the mask with the highest output determines the edge magnitude at that pixel, and the orientation of this mask gives the orientation of the edge. Kirsch [11] has used an edge detector with slightly different masks, as shown in Fig. 7-7. Again, eight masks are used. (This is a slightly modified description of the Kirsch operator from that given in [11].)

1	1	-1
1	-2	-1
1	1	-1

(a) Vertical

1	-1	-1
1	-2	-1
1	1	1

(b) Diagonal

Figure 7-6: Prewitt edge masks in two directions: (a) vertical, (b) diagonal

3	3	-5
3	0	-5
3	3	-5

(a)

3	-5	-5
3	0	-5
3	3	3

(b)

Figure 7-7: Masks for Kirsch operator (two directions)

The edge masks are not limited to 3-by-3 neighborhoods. A general edge mask is shown in Fig. 7-8. For arbitrary orientations, the analog mask can be approximated by using values determined by the proportion of a cell being on either side of an ideal edge. 5-by-5 masks for 30-degree increments are shown in Fig. 7-9 (from [12]).

McLeod has suggested the use of masks where the weights drop exponentially as the distance from the edge increases. The weights also drop along the edge, as the distance from the center increases [13]. Such masks de-emphasize the effects of points away from the center.

An important parameter to select is the size of the edge mask. A larger mask offers more immunity to noise, but less discrimination between nearby edges. An adaptive way to choose an optimal mask size is to compute the convolution outputs with masks of increasing size and choose the largest size such that the next larger size causes a significant reduction in the output value, presumably due to the introduction of a second edge. Experiments with masks of varying size have been described by Rosenfeld and Thurston [14] and by Marr [15]. The number of masks depends on the desired angular resolution and also on the mask size and shape. Long and thin masks need finer angular resolution so that the edges with orientation between two edge masks are not missed.

-1	+1

Figure 7-8: A schematic edge mask

-100	-100	0	100	100
-100	-100	0	100	100
-100	-100	0	100	100
-100	-100	0	100	100
-100	-100	0	100	100

(a) 0°

-100	32	100	100	100
-100	-78	92	100	100
-100	-100	0	100	100
-100	-100	-92	78	100
-100	-100	-100	-32	100

(b) 30°

100	100	100	100	100
-32	78	100	100	100
-100	-92	0	92	100
-100	-100	-100	-78	32
-100	-100	-100	-100	-100

(c) 60°

100	100	100	100	100
100	100	100	100	100
0	0	0	0	0
-100	-100	-100	-100	-100
-100	-100	-100	-100	-100

(d) 90°

-100	100	100	100	100
-100	100	100	78	-32
-100	92	0	-92	-100
32	-78	-100	-100	-100
-100	-100	-100	-100	-100

(e) 120°

100	100	100	32	-100
100	100	92	-78	-100
100	100	0	-100	-100
100	78	-92	-100	-100
100	-32	-100	-100	-100

(f) 150°

Figure 7-9: Edge masks in six directions: (a) 0°,(b) 30°, (c) 60°, (d) 90°, (e) 120°, (f) 150°

For large edge masks, the template match output is high not only at an edge, but also in its vicinity. The output as a function of the mask position for the mask of Fig. 7-8 is shown schematically in Fig. 7-10. A simple thresholding of the output will thus give a "thick" edge with uncertainty in position. Figure 7-11(a) shows an aerial image of an airport and Fig. 7-11(b) the magnitude of the edge output using the masks of Fig. 7-9. Simple thresholding of the edge output gives thick edges as shown in Fig. 7-11(c). The edge output can be "thinned" by selecting only the local peaks. This operation is sometimes called *nonmaxima suppression* [14]. Improved results can be obtained by thinning in a direction normal to the edge direction; a scheme to combine thinning and thresholding is described below.

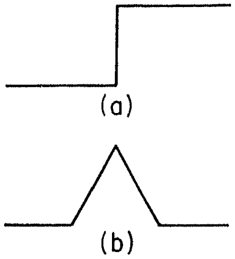


Figure 7-10: (a) Step edge, (b) output profile

Precise location and characterization of edges can be more complex than simple peak detection of the convolved output. Nevatia and Babu used step masks as shown in Fig. 7-9 and examined magnitudes and directions of pixels in a small neighborhood [12]. At each pixel, let the maximum outputs of the convolution with these masks, and the direction of the mask producing the maximum output, be known as the edge magnitude and direction, respectively, at that pixel. They required that for a pixel to be an edge pixel, its edge magnitude be higher than the edge magnitudes of the pixels on either side of it, in a direction normal to its edge direction. Further, these neighboring pixels were required to have edge orientations similar to that of the central pixel. Figure 7-11(d) shows the edges detected by this procedure in the image of Fig. 7-11(a). Note that the two sides of the vertical taxiway in the right of the image are connected in Fig. 7-11(c) and yet are separated by the described thinning process.

More complex decision procedures have used both edge and *bar* masks. A bar mask, shown schematically in Fig. 7-12, approximates second derivatives. At first, the bar masks may seem to be ideally suited for the detection of "line" like edges. However, these bar or line masks also respond strongly to bright points and to step edges.

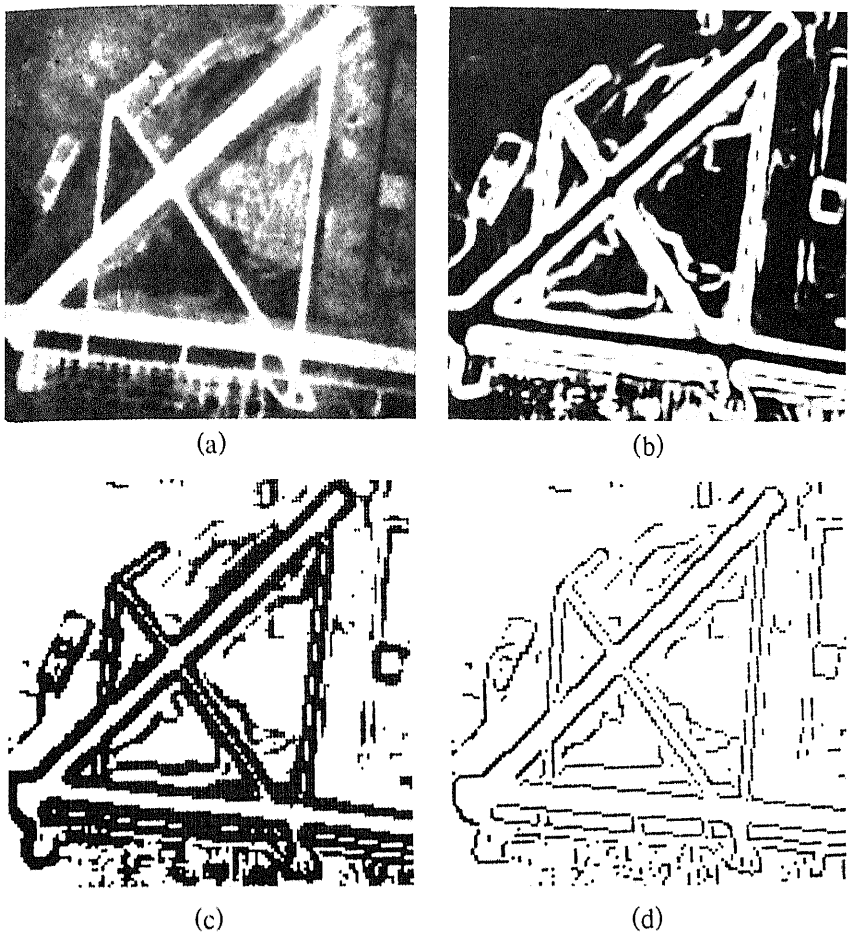


Figure 7-11: Steps in edge detection: (a) image, (b) edge magnitude, (c) unthinned edge, (d) thinned edges

VanderBrug has suggested a "nonlinear" line detector that requires the presence of a line to be indicated not only by the total mask output being high but also by a determination along each row of the bar mask that the values in the positive part are higher (or lower) than in the neighboring negative parts [16].

Herskovits [17] and Marr [15] have used the outputs of both the edge and the bar masks for the location and description of the edges. Herskovits has defined several types of edges that occur in scenes of polyhedra, as shown in Fig. 7-13. Each edge type produces a characteristic output profile when convolved with the edge, and the bar

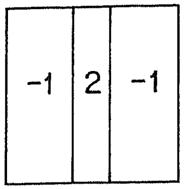


Figure 7-12: A schematic bar mask

masks and these profiles can be used to determine the edge type [15]. As an example, for a step edge, the edge mask produces a triangular output with peak at the edge location whereas the bar mask output has a zero value at the edge location, surrounded by a positive peak on one side and a negative peak on the other.

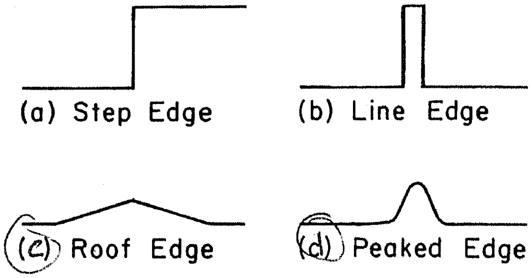


Figure 7-13: Different types of edges: (a) step, (b) line, (c) rook, (d) peaked
roof

In later work, Marr and Hildreth used an isotropic second derivative operator, instead of the directed bar masks [18]. The simplest such operator is a Laplacian ($\partial^2/\partial x^2 + \partial^2/\partial y^2$). The operator used by Marr and Hildreth is a Laplacian of a Gaussian and has an intensity profile similar to that of a lateral inhibition operator (Section 6.2). The operator is given formally by

$$\nabla^2 G(x, y) = \frac{-1}{2\pi\sigma^4} \cdot \left(2 - \frac{x^2 + y^2}{\sigma^2}\right) e^{-\frac{x^2+y^2}{2\sigma^2}} \qquad (7\text{-}9)$$

where $w = 2\sigma$ determines the width of the excitatory region of the operator, and $\lambda = 2\pi\sigma$ is the diameter.

As in the case of a bar mask, this operator produces a zero-crossing at the edge location for a step edge. Note that the second-derivative operators are insensitive to gradual changes in

illumination. The above operator is also conjectured to have the property that zero-crossing information is sufficient to reconstruct the original image. Marr and Hildreth suggest use of four different mask sizes to correspond to observations from the human visual system.

7.1.4 Statistical Edge Detectors

Edge detection has also been viewed as a problem of statistical decision making. Griffith has described an analysis that uses the a priori probability of grey-level distributions in an image in the presence of an edge [19, 20]. For step and line edges, the optimal decision reduces to a procedure like template matching. This statistical analysis, however, requires many simplifying assumptions.

A different approach, by Yakimovsky, views edge detection as choosing between the following two hypotheses [21]:

H_0: The samples on the two sides of a line are taken from the same object.

H_1: The samples on one side are from one object and on the other side from a different object.

Under the assumptions that the object grey levels are uniform, except for added white Gaussian noise, the optimal decision reduces to a likelihood-ratio computation, depending only on the measured variances on the two sides of the hypothesized edge.

7.1.5 Choice of Thresholds

Most of the above techniques require that a certain measure of edge strength exceed a threshold for an edge presence to be declared. The choice of such thresholds should depend on the expected properties of the desired edges and the undesired variations including random noise and changes in illumination. The threshold selection should also be dependent on context; for example, it is easier for us to perceive low-contrast edges if they belong to an elongated boundary segment.

In general, however, the properties of a new scene are not known a priori, and hence the threshold must be selected by measurements on the image. Owing to an absence of adequate mathematical models for the structures in an image, the common approach is to choose an "optimal" threshold by empirical analysis of a set of typical images. These thresholds may then be changed by feedback from further analysis of the resulting edges.

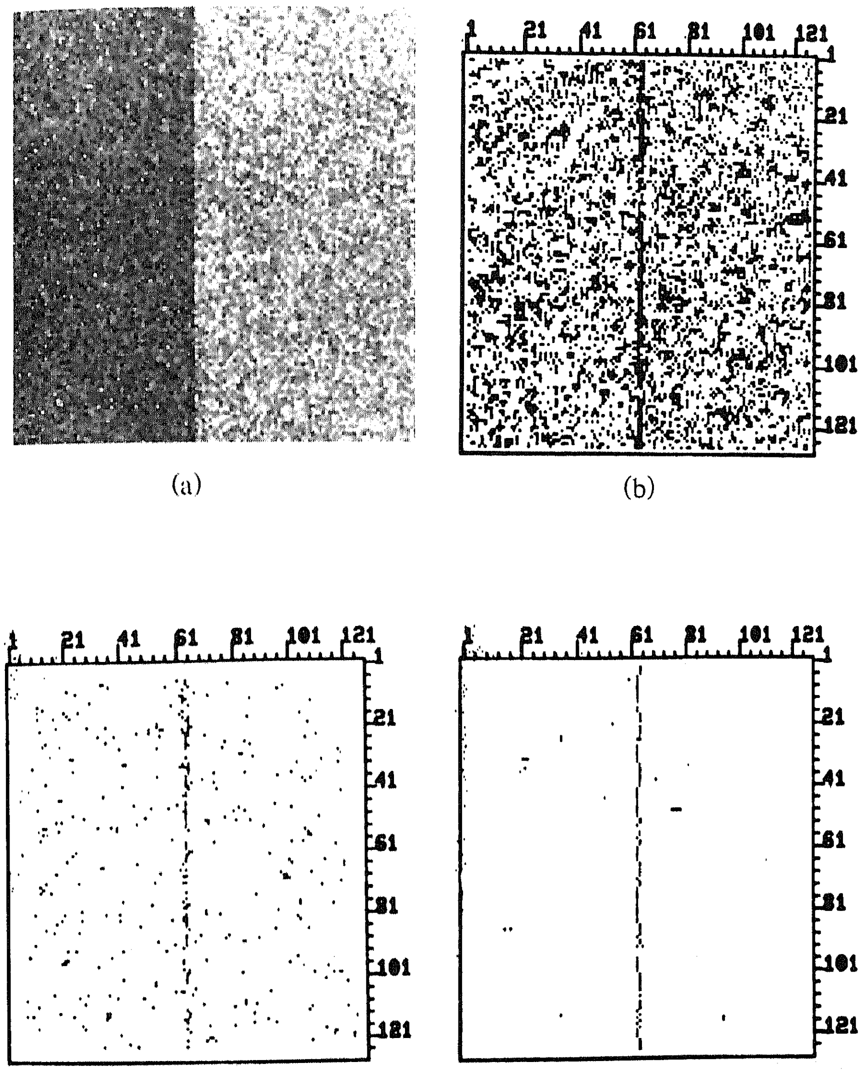
For specific and restricted scenes, optimal thresholds may be determined by a mathematical analysis. Herskovits devised a technique for choosing a threshold for scenes of polyhedra viewed through a sensor with known noise characteristics to maximize the sensitivity of edge detection while minimizing the response to noise and smooth illumination gradients [17]. Abdou and Pratt have described analytical techniques with different optimality criteria using statistical decision theory methods [9, 22].

Note that the edge detectors using second-derivative masks need not use a threshold, as only a zero-crossing must be detected. The operator described by Nevatia and Babu [12] can also be used without thresholds as other tests on the output profile are used.

7.2 RESULTS OF EDGE DETECTION

How successful are the various edge detectors in providing the desired results for a given image? The answer is complicated by the dependence of the results on the particular image and the lack of a suitable model for images to use for predictions. Also, in complex images, the desired response for the edge detection process is difficult to specify, as edges are likely to occur in many positions other than the desired object boundaries because of the presence of surface texture. Mathematical analysis for even simple images is difficult owing to the complexity and the nonlinearity of the edge-detection algorithms. Some analytical results may be found in [9, 17, 22]. Abdou's results indicate that template matching is optimal, if the type of the edge is known and the noise is additive Gaussian.

Alternatives to empirical analysis are the use of a set of typical real images or synthetic images with controlled parameters. The results are easier to evaluate in the latter case, but not necessarily indicative of performance on real images. Figure 7-14(a) shows a step with Gaussian noise (the ratio of step size to signal variance is 2). The image looks noisy, but the vertical edge is distinctly visible to us. Figures 7-14(b), (c), and (d) show the outputs of the Sobel, Hueckel, and the Nevatia-Babu edge detector using six masks shown earlier in Fig. 7-9. Note that the last output is thinned, as thinning is integral with thresholding, whereas the others are not. In each case, the thresholds were adjusted for best subjective response, picking up the maximum number of desired edges without being swamped with the noise edges. For this example, the 5-by-5 mask edge detector clearly is much superior to the other two. For similar images with less noise, the differences are less pronounced, and all work nearly perfectly. A systematic and comprehensive evaluation of many edge detectors with



(a) (b)

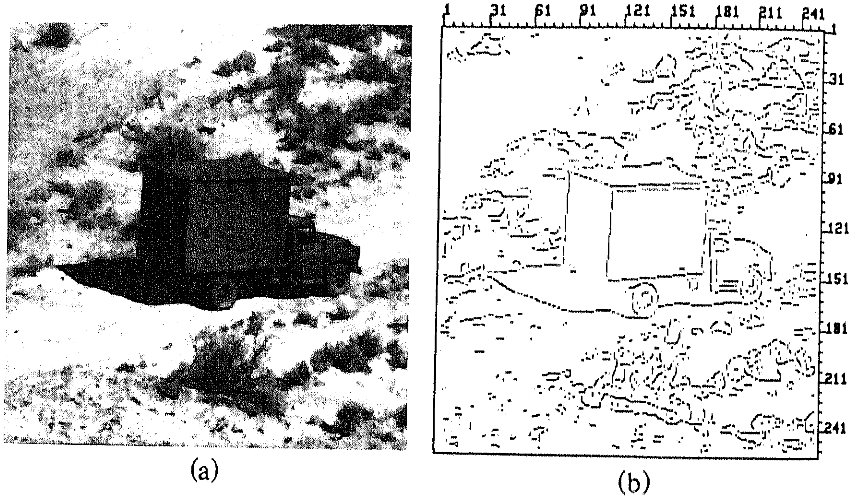
(c) (d)

Figure 7-14: An image and the edges detected in it: (a) image, (b) Sobel output, (c) Hueckel output, (d) Nevatia-Babu output

synthetic images may be found in [9, 23, 24].

More interesting are the results of edge detection on real images. Even for simple scenes, as the polyhedral object scene of Fig. 3-4 and the slightly more complex scene of Fig. 7-11 indicate, some of the desired edges are missed and several undesired edges are present. The undesired edges are primarily due to real surface markings or texture (and may, in fact, be useful for texture analysis, as in Section 8.3.2). Two more examples are discussed below.

Figure 7-15(a) shows an image of a truck with bushes in the foreground and the background. Figure 7-15(b) shows the edges detected by the Nevatia-Babu edge detector. It is clear that some elaborate structuring of the edges is needed before the desired object boundaries can be derived. Adequacy of edge detection cannot be evaluated separately from these processes. The edge detection is far from perfect, compared to our perception. However, many of the undesired edges, such as those in the background, are real edges in the image. Many of the missing edges are due to inadequacy of the edge detector, but some, particularly those near the top of the boundary of the truck with the background, are very low-contrast edges and distinctly visible to humans only when viewed with the surrounding high-contrast edges. Such edges may be called perceptual edges and are not expected to be detected by a local edge detector.



(a)

(b)

Figure 7-15: (a) A truck image and (b) edges detected in it

Figure 7-16 shows edges detected (by the algorithm described in [12]) in the aerial image of the San Francisco area shown earlier in Fig. 1-9. Here, we can recognize many important features of the image, but further analysis of this large number of edges by machine is going to be complex. However, such edge output has been used for the detection of roads, bridges, and other linear features, and along with a region segmentation technique described in Chapter 8 has also been used for matching of the image with a symbolic map; some results are given in Chapter 10. (Such processing has applications in navigation and map-updating.)

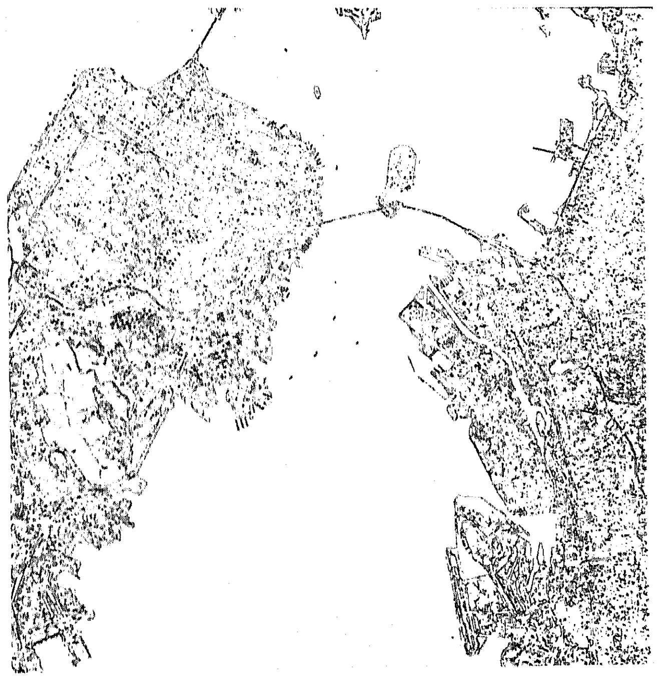


Figure 7-16: Edges detected in the aerial image of Fig. 1-9

Marr has suggested that the basic purpose of low-level processing, such as edge detection, is to provide a description of discontinuities that is adequate for higher-level processing without recourse to original intensity data [15]. In his system, the descriptions are derived by convolving the image with edge and bar masks (or Laplacian-Gaussian masks) of different sizes. In addition to the presence of a significant discontinuity, the type of the edge (for example, step or roof), its contrast and sharpness are included in the description. Such a description, termed a *primal sketch*, was conjectured by Marr to be the main result of low-level processing for humans and other primates.

7.3 LINE AND CURVE DETECTION

An important level of processing of edges detected in an image (possibly with additional descriptions as in Marr's primal sketch) is to aggregate the edge elements to form object boundaries. The shape of these boundaries may not be known a priori, but in many cases they can be approximated well by piecewise linear segments. However, it is not feasible to simply fit linear segments to all the edges in an image and discard the poor fits. It is first necessary to aggregate the edges lying along a single line or another known curve. Proximity and the directions of edges, and perhaps more detailed descriptions of the edges, can be used for such aggregation. In the following, a few such techniques are described.

7.3.1 Hough Transform

It is simpler to first describe this technique applied to the detection of a set of points lying on a straight line. The general equation of a straight line can be written as

$$x \cos \theta + y \sin \theta = r \qquad (7\text{-}10)$$

where θ is the angle made by a normal to the line with the x axis and r is the length of this normal (see Fig. 7-17). For a given point (x_i, y_i) on this line, Eq. (7-10) becomes

$$x_i \cos \theta + y_i \sin \theta = r \qquad (7\text{-}11)$$

Considering r and θ as the new variables, the above equation gives the relation between the parameters of a line constrained to go through the point (x_i, y_i). Equation (7-11) corresponds to a sinusoidal curve in the (r, θ) space (see Fig. 7-18). This relationship between the image plane and the (r, θ) plane is known as the Hough transform after the inventor [25]. For a collinear group of points in the image plane, the curves determined by Eq. (7-10) in the (r, θ) space must all intersect in one common point (corresponding to the actual parameters of the line through the image points).

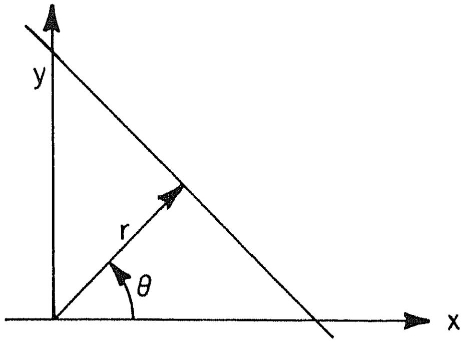


Figure 7-17: The (r, θ) representation of a straight line

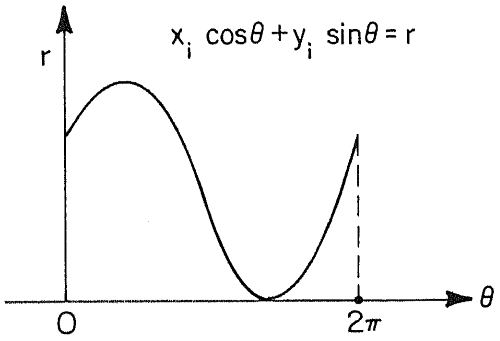


Figure 7-18: Hough transform~~n~~ of a point (x_i, y_i)

Thus, to detect clusters of collinear points, we could construct the transform curves in the (r, θ) space for each point and pick the points corresponding to coincidence of three or more such curves. For implementation in a digital computer, the r and θ parameters must be quantized. θ varies between 0 and 2π, and r can be limited to the size of the image plane. Construction of a curve in (r, θ) space now requires incrementing the counters of cells along the curve. To find coincidences, we simply pick the cells with high counts. The performance is affected by the chosen quantization. A coarse digitization will fail to distinguish between nearby lines, and a very fine resolution would allow little error in collinearity.

This procedure can be made more efficient if the directions of the edges are known. Then, a single point, rather than a sinusoid, can be computed in the (r, θ) space for a given edge point. Note that the Hough transform technique does not examine the proximity of the clustered points. Groups of contiguous points must be further separated from a cluster of collinear points.

The Hough transform technique easily generalizes to detection of arbitrary curves described by a function such as

$$f(a_1, a_2, \ldots, a_n, x, y) = 0 \tag{7-12}$$

where a_1, a_2, ..., a_n are the parameters of the curve. For each point(x_i, y_i), we now associate a surface $f(a_1, a_2, ..., a_n)$ in the $(a_1, a_2, ..., a_n)$ space. The complexity of the computation increases with the dimensionality of this parameter space.

Hough transform techniques became widely known through work of Duda and Hart [26]. Extensions and generalizations of the basic technique, including the detection of circle and other nonlinear curves, may be found in [26-32]. An interesting generalization for nonanalytical shapes can be found in [33]. In this approach, the parameters of the Hough space are the origin of a local coordinate system for the figure and its scale and rotation.

7.3.2 Graph-Theoretic Techniques

A simple linking technique is to connect each point to one or more of its neighbors (say 8-neighbors), if they have similar orientations. This technique is feasible only if the number of branches at each step can be kept low. Nevatia and Babu have described an implementation where the number of branches is at usually one and at most two [12]. Their edge-detection method was by convolution with edge-shaped masks, followed by thinning in a direction normal to the edge. Such edge detection tends to yield edges that form lines with few gaps and few small connecting branches. If only small neighborhoods, say the 8-neighbors, are considered for possible linking, an exhaustive list of rules for allowable connections can be made.

Generally, the edge points in an image may be viewed as the nodes of a graph and a cost function associated with the connection of two edge points. The desired boundaries can then be defined as the minimal-cost (or low-cost) paths through this graph. The cost of connecting two edge points is defined to be a function of the distance between them, difference in their directions, and desired similarities in other descriptions. Various cost functions and graph search techniques have been used. Montanari, and Martelli have described dynamic programming approaches that are best suited if only a single desired curve is to be extracted [34, 35]. These techniques are efficient if starting and end points of the curve are also given. Zahn [36] has suggested the use of the longest branch of a *minimal spanning tree* for a more efficient search.

Ramer has described a graph search technique that applies to detection of one or more curves [37, 38]. His technique is a simple tree search, where at each node the number of alternatives is restricted to a few nearest neighbors only. Ramer calls the basic edge elements *strokes* and the resulting connected segments *streaks.* His program also maintains information about streak intersections and possible loops. His technique was applied to simple curved objects, such as a table lamp, in relatively clear environments.

Graph-theoretic techniques are useful when the graph search can be constrained. For special applications, specific curves have been successfully detected even in presence of large noise (for example, see [34]).

7.3.3 Method of Projections

Another approach to detection of straight-line segments is to search for these lines in a number of quantized directions over the entire range of directions. For a given direction, say vertical, the image plane is divided in strips or buckets parallel to this direction. Only the edge elements in the same bucket, and the resulting segments from neighboring buckets are linked together (see Fig. 7-19). The linking is merely in the order of increasing (or decreasing) coordinate along the direction of linking. Such a linking algorithm has been used by the author [39]. A similar algorithm is also described by Marr [15] (called the process of "theta aggregation").

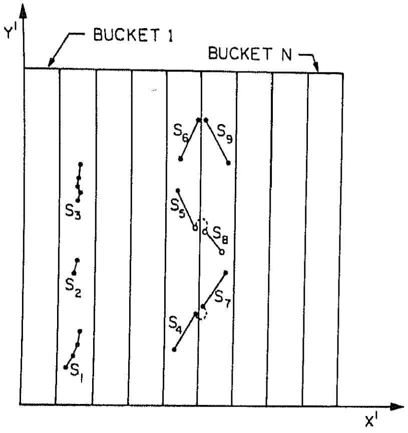


Figure 7-19: Linking by projections

7.4 CONTOUR FILLING AND FEEDBACK

The different line-detection techniques of the previous section differ in their computational costs, but the results are rather similar and largely dictated by the performance of the local edge detector itself. The detected boundaries are generally fragmented and not closed, except for some simple scenes. The gaps in the boundary further complicate the separation of object boundary segments from the "background" boundary segments. Some of the gaps are due to the failings of the local edge detector, particularly near corners and in areas of high texture. However, in real images, some of the boundaries that are perceptually continuous to us have local segments with little discontinuity in local attributes. These segments are difficult to perceive, if seen in isolation from the rest of the scene.

Small gaps in the boundaries can be filled simply by extrapolation. These extrapolations may then be "verified" by the use of a more sensitive edge detector. Many systems use such contour filling, starting from the early work of Roberts [3], but they are ad hoc and not of general applicability.

Zucker, Hummel, and Rosenfeld have described an application of stochastic relaxation labeling, described previously in Chapter 5, to improving detection of curves and lines in images [40]. Each point in the image is assigned a probability of being on a boundary or not. Initial probabilities may be computed from the output of a local edge detector. These probabilities are then modified iteratively, based on "compatibility" with neighbors. An edge point is viewed as being more compatible with other edge points of similar orientation in the neighborhood. This technique tends to cause longer lines to grow and isolated points to be removed. However, the performance is dependent on the choice of the compatibility criteria.

Shirai developed an interesting feedback mechanism for the limited domain of polyhedral objects [41]. The outer boundaries of such object assemblies are of high contrast and easily detected. The interior boundaries can have poor contrast, depending on the sources of illumination and may be hard to detect. Shirai's system detects the outer boundaries first and then proposes hypotheses for the presence of other lines, which are verified (or rejected) by a more sensitive edge detector. These rules are specifically for the domain of polyhedral objects but do not require knowledge of the particular objects present. Figure 7-20 shows the successive steps for a typical example. One rule proposes extension of any existing line at a concave vertex, such as line *KJ* at vertex *J*. Another rule suggests new lines that are parallel to other existing lines such as lines *G'M'* and *O'P'*. A total of ten such rules is used. Verification by a sensitive edge detector is possible since a long line of specific orientation and position is being evaluated. This system was successful in obtaining complete boundaries for fairly complex polyhedral scenes. The term "heterarchy" was first coined to describe this technique.

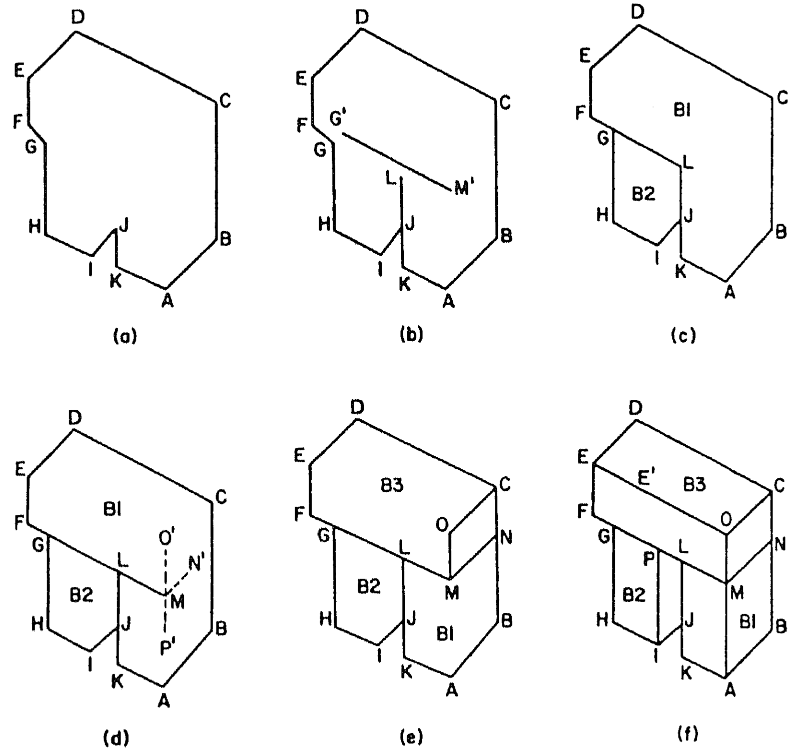


Figure 7-20: Steps in Shirai's line-detection system (from Shirai [41])

Kelley used a *planning* approach to boundary detection [42]. A boundary is first detected in a lower-resolution image, obtained by averaging the larger image. This boundary is then expanded in scale for a higher-resolution image, and detailed analysis between the gaps can be performed. This technique can be helpful in reducing the computation time, but finer lines, such as thin roads, may be lost in the reduced image.

Several suggestions have been made to use a sequence of reduced images forming a *pyramid* structure, with processing at one level influencing the processing at higher or lower levels [43-46]. Such structures are used not only for edge and curve detection, but for all levels of processing. The levels in the pyramid defined by varying resolution should be distinguished from the levels of abstraction in processing (for one resolution level). The pyramid structures are likely to lead to efficient processing if the level with minimum adequate resolution can be isolated, without elaborate processing. Also, for certain images, the processing may be easier at certain low-resolution levels, owing to averaging of fine texture and small details.

In many instances, missing boundary segments may be obtained only by other, more difficult processing—for example, in discrimination of surfaces of different textures or of discontinuities in three-dimensional surface orientations. In some images, humans perceive contours in addition to those formed by edge discontinuities. For example, a circle connecting the end points of lines is seen in Fig. 7-21(a) and an extra triangle is perceived in Fig. 7-21(b). It has been suggested that the end points of detected boundary segments should be examined for such *subjective contours*, based on their proximity, collinearity of the segments, or the end points forming regular patterns [15].

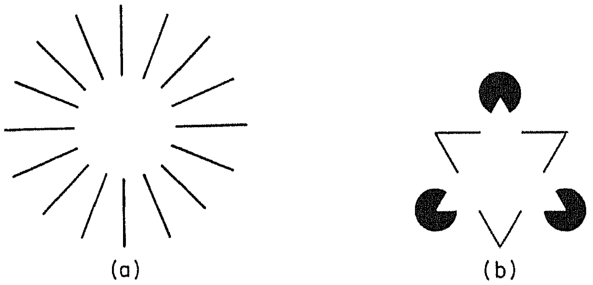


Figure 7-21: Two instance of subjective contours

Experience with edge- and line-detection techniques indicates that complete object boundaries are difficult to obtain, and thus many of the previously described techniques of higher-level scene analysis are difficult to use directly. One point of view even contends that higher-level descriptions must be obtained from incomplete boundaries and, in turn, used to complete the boundaries. In spite of limited development of such techniques, incomplete boundaries have been used for recognition and higher-level descriptions in limited domains. Perkins, for example, has described recognition of industrial parts by matching of boundary descriptions [47]; Nevatia and Babu have described detection of roads by locating *antiparallel* lines, which are parallel lines of opposite contrast [12]. (The latter may be viewed as a very simple case of computing generalized cones, as defined in Chapter 5, from incomplete boundaries.)

7.5 COLOR EDGES

A color image is given by a triple of numbers for each pixel (in one of the many representations). To detect color edges, we can detect the edges in the three components separately and then combine the results. (In some attributes, such as hue, the range of values may be circular, or have a discontinuity. Edges can not be easily detected in these components directly.)

The simplest way to combine the edges in three color components is to form a new edge whose strength is a linear combination of the strengths of the edges in the three components. (Reconciling different edge orientations is more complicated.) However, such combination is likely to lead to results very similar to the processing of a single grey-level image, which itself is derived by a linear combination of three component images. An alternative is to accept an edge in the color image, if an edge is present in any of the components. Another alternative is to require concurrence in all components.

The author has experimented with a generalization of the Hueckel edge detector, to compute a best-fit edge in three-dimensional color space [48]. The orientation of the edge in the three components is constrained to be the same, but other edge parameters, such as brightness and the intensity profile, may vary. The three components of the resulting edge were combined in different ways, as discussed above. An analysis of many images pointed to the following conclusions (these are believed to be independent of the edge detector used):

1. Intensity and chromatic edges tend to co-occur. This may be expected as two objects of different color are unlikely to have the same intensity.

2. Chromatic edges may be useful in boundary detection in low contrast images where intensity edges are weak.

3. Requiring co-occurrence of intensity and chromatic edges may provide higher-confidence edges, useful in a multipass system.

Color has been found to be more useful in region-segmentation techniques described in the next chapter. This is consistent with the hypothesis that humans use color for distinguishing large surfaces rather than high-resolution details.

7.6 SUMMARY

This chapter has covered several techniques of edge detection, mainly distinguished by their computational definitions of local discontinuities. Various techniques for grouping local edges into boundaries have also been described. A major conclusion of experience with such techniques has been that while the local techniques can be improved, boundary detection also requires global processing, and that higher-level descriptions must be able to use incomplete boundaries.

REFERENCES

[1] L. S. Davis, "A Survey of Edge Detection Techniques," *Computer Graphics and Image Processing*, Vol. 4, No. 3, September 1975, pp. 248-270.

[2] W. K. Pratt, *Digital Image Processing*, John Wiley & Sons, New York, 1978.

[3] L. G. Roberts, "Machine Perception of Three-Dimensional Solids," in *Optical and Electro-Optical Information Processing*, J. T. Tippett, et al. (eds.), MIT Press, Cambridge, Mass., 1965, pp. 159-197.

[4] J. M. S. Prewitt, "Object Enhancement and Extraction," in *Picture Processing and Psychopictorics*, B. S. Lipkin and A. Rosenfeld (eds.), Academic Press, New York, 1970.

[5] M. H. Hueckel, "A Local Visual Edge Operator Which Recognizes Edges and Lines," *Journal of the ACM*, Vol. 20, 1973, pp. 634-647.

[6] M. H. Hueckel, "An Operator Which Locates Edges in Digitized Pictures," *Journal of the ACM*, Vol. 18, 1971, pp. 113-15.

[7] L. Mero and Z. Vassy, "A Simplified and Fast Version of the Hueckel Operator for Finding Optimal Edges in Pictures," in *Advance Papers of the Fourth International Joint Conference on Artificial Intelligence*, Tbilisi, Georgia, U.S.S.R., September 1975, pp. 650-655.

[8] R. Nevatia, "Evaluation of a Simplified Hueckel Edge-Line Detector," *Computer Graphics and Image Processing*, Vol. 6, 1977, pp. 582-588.

[9] I. Abdou, "Quantitative Methods of Edge Detection," University of Southern California, Report USCIPI 830 (Ph.D. thesis), July 1978.

[10] G. B. Shaw, "Local and Regional Edge Detectors: Some Comparisons," *Computer Graphics and Image Processing*, Vol. 9, No. 2, February 1979, pp. 135-149.

[11] R. A. Kirsch, "Computer Determination of the Constituent Structure of Biological Images," *Computers and Biomedical Research*, Vol. 4, No. 3, June 1971, pp. 315-328.

[12] R. Nevatia and K. R. Babu, "Linear Feature Extraction and Description," *Computer Graphics and Image Processing*, Vol. 13, 1980, pp. 257-269.

[13] I. D. G. Macleod, "Comments on Techniques for Edge Detection," *Proceedings of the IEEE*, Vol. 60, No. 3, March 1972, pp. 344.

[14] A. Rosenfeld and M. Thurston, "Edge and Curve Detection for Visual Scene Analysis," *IEEE Transactions on Computers*, Vol. C-20, May 1971, pp. 562-569.

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

[16] G. J. VanderBrug, "Semilinear Line Detectors," *Computer Graphics and Image Processing*, Vol. 4, 1975, pp. 287-293.

[17] A. Herskovits, "On Boundary Detection," MIT Project MAC Memo 183, Cambridge, Mass., 1970.

[18] D. Marr and E. Hildreth, "Theory of Edge Detection," *Proceedings of the Royal Society of London*, B207, 1980, pp. 187-217.

[19] A. K. Griffith, "Mathematical Models for Automatic Line Detection," *Journal of ACM*, Vol. 120, January 1973, pp. 62-80.

[20] A. K. Griffith, "Edge Detection in Simple Scenes Using A PriorInformation," *IEEE Transactions on Computers*, Vol. 22, No. 4, April 1973, pp. 371-380.

[21] Y. Yakimovsky, "Boundary and Object Detection in Real World Images," *Journal of ACM*, Vol. 23, 1976, pp. 599-618.

[22] I. Abdou and W. K. Pratt, "Quantitative Design and Evaluation of Enhancement Edge Detector Schemes," *Proceedings of IEEE*, Vol. 67, 1979, pp. 753-763.

[23] J. R. Fram, and E. S. Deutsch, "On the Quantitative Evaluation of Edge Detection Schemes and Their Comparison with Human Performance," *IEEE Transactions on Computers*, Vol. 24, No. 6, June 1975, pp. 616-628.

[24] E. S. Deutsch and J. R. Fram, "A Quantitative Study of the Orientation Bias of Some Edge Detector Schemes," *IEEE Transactions on Computers*, Vol. 27, 1978, pp. 205-213.

[25] P. V. C. Hough, "Method and Means for Recognizing Complex Patterns," U.S. Patent 3069654, December 18, 1962.

[26] R. O. Duda and P. E. Hart, "Use of the Hough Transformation to Detect Lines and Curves in Pictures," *Communications of the ACM*, Vol. 15, January 1972, pp. 11-15.

[27] C. Kimme, D. Ballard, and J. Sklansky, "Finding Circles by an Array of Accumulators," *Communications of ACM*, Vol. 18, February 1975, pp. 120-122.

[28] S. D. Shapiro, "Transformations for the Computer Detection of Curves in Noisy Pictures," *Computer Graphics and Image Processing*, December 75, pp. 328-338.

[29] S. D. Shapiro, "Properties of Transforms for Detection of Curves in Noisy Pictures," *Computer Graphics and Image Processing*, October 1978, Vol. 8, No. 2, pp. 219-236.

[30] J. Sklansky, "On the Hough Techniques for Curve Detection," *IEEE Transactions on Computers*, Vol. 27, No. 10, October 1978, pp. 923-926.

[31] S. Tsuji and F. Matsumoto, "Detection of Ellipses by a Modified Hough Transform," *IEEE Transactions on Computers*, Vol. 27, No. 8, August 1978, pp. 777-781.

[32] F. O'Gorman and M. B. Clowes, "Finding Picture Edges Through Collinearity of Feature Points," in *Proceedings of the Third International Joint Conference on Artificial Intelligence*, August 1973, Stanford, Calif., pp. 543-555.

[33] D. H. Ballard and D. Sabbah, "On Shapes," *Proceedings of the Seventh International Joint Conference on Artificial Intelligence*, Vancouver, B.C., Canada, August 1981, pp. 607-612.

[34] U. Montanari, "On the Optimal Detection of Curves in Noisy Pictures," *Communications of ACM*, Vol. 14, May 1971, pp. 335-345.

[35] A. Martelli, "An Application of Heuristic Search Methods to Edge and Contour Detection," *Communications of ACM*, February 1976, pp. 73-83.

[36] C. T. Zahn, "Graph-theoretical Methods for Detecting and Describing Gestalt Clusters," *IEEE Transactions on Computers*, Vol. 20, pp. 68-86, January 1971.

[37] U. Ramer, "Extraction of Lines Structures from Photographs of Curved Objects," *Computer Graphics and Image Processing*, Vol. 4, pp. 81-103, June 1975.

[38] U. Ramer, "The Transformation of Photographic Images into Stock Arrays," *IEEE Transactions on Circuits and Systems*, Vol. 22, 1975, pp. 363-374.

[39] R. Nevatia, "Locating Object Boundaries in Textured Environments," *IEEE Transactions on Computers*, Vol. 25, No. 11, November 1976, pp. 1170-1175.

[40] S. W. Zucker, R. A. Hummel and A. Rosenfeld, "An Application of Relaxation Labeling to Line and Curve Enhancement," *IEEE Transactions on Computers*, Vol. 26, No. 4, April 1977, pp. 394-403.

[41] Y. Shirai, "Analyzing Intensity Arrays Using Knowledge About Scenes," in *The Psychology of Computer Vision*, P. H. Winston (ed.), McGraw-Hill, New York, 1975.

[42] M. Kelly, "Edge Detection by Computer Using Planning," in *Machine Intelligence VI*, B. Meltzer and D. Michie (eds.), Edinburgh University Press, Edinburgh, 1971, pp. 397-409.

[43] A. R. Hanson and E. M. Riseman, "VISIONS: A Computer System for Interpreting Scenes," in *Computer Vision Systems*, A. R. Hanson and E. M. Riseman (eds.), Academic Press, New York, 1978, pp. 303-333.

[44] S. Tanimoto and T. Pavlidis, "A Hierarchical Data Structure for Picture Processing," *Computer Graphics and Image Processing*, Vol. 4, No. 2, June 1975, pp. 104-119.

[45] L. Uhr, "Layered Recognition Cone Networks That Preprocess, Classify and Describe," *IEEE Transactions on Computers*, Vol. 21, July 1972, pp. 758-768.

[46] A. Klinger and C. R. Dyer, "Experiments on Picture Representation Using Regular Decomposition," *Computer Graphics and Image Processing*, Vol. 5, 1976, pp. 68-105.

[47] W. A. Perkins, "A Model-Based Vision System for Industrial Parts," *IEEE Transactions on Computers*, Vol. 27, 1978, pp. 126-143.

[48] R. Nevatia, "A Color Edge Detector and Its Use in Scene Segmentation," *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 7, No. 11, November 1977, pp. 820-826.