COMPUTER GRAPHICS AND IMAGE PROCESSING (1975) 4, (248–270)

SURVEY

A Survey of Edge Detection Techniques

LARRY S. DAVIS

University of Maryland,
College Park, Maryland 20742

Communicated by A. Rosenfeld

Received October 16, 1974

Methods of detecting "edges," i.e., boundaries between regions in a picture, are reviewed. Included are both parallel (linear, nonlinear, optimal) and sequential methods, as well as methods using planning or a priori knowledge.

1. THE PROBLEM OF EDGE DETECTION

In a grey-level picture containing homogeneous (i.e., untextured) objects, an edge is the boundary between two regions of different constant grey level. The ideal step edge in such a picture has the cross section shown in Fig. 1. Depending upon the class of pictures being analyzed we get a variety of edge cross sections. For example, if we look at solid objects, which contain surfaces at different orientations, meeting at sharp angles, then roof-type edges (Fig. 2) and spike edges (Fig. 3) are also present [9]. In the remainder of this paper, though, the word "edge" will refer to step edges, since they are by far the most common type of edge encountered.

The ideal edge, however, is not what one finds in the images produced by image dissectors, flying spot scanners, and other imaging devices. There are several factors that degrade the edges that are actually found:

1. photon noise (quantum effects);
2. blurring, or defocusing (this is especially pronounced with the image dissector);
3. irregularities of the surface structure of the objects.

The effects of (1) and (2) on the output of an image dissector are discussed quantitatively by Herskovitz and Binford [9]. Of course, in any particular application there may be other sources of noise, but these three are universal and must be dealt with by any edge detection scheme.

The net effect of the constraints on the imaging process and the departure from homogeneity of the objects is that the perfect edge of Fig. 1 may be degraded to something similar to Fig. 4. Here the step is replaced by a noisy

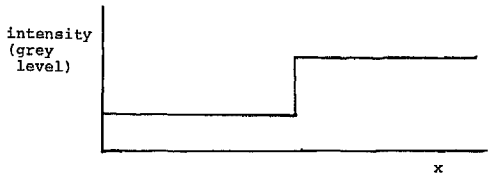

FIG. 1. An ideal step edge; cross section orthogonal to the direction of the edge.

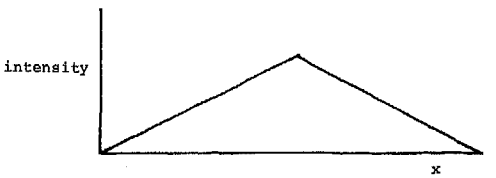

FIG. 2. An ideal roof edge.

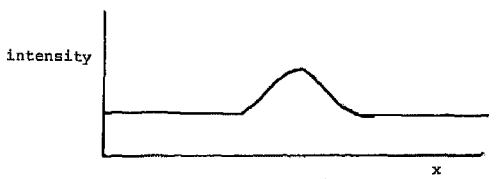

FIG. 3. An ideal spike edge.

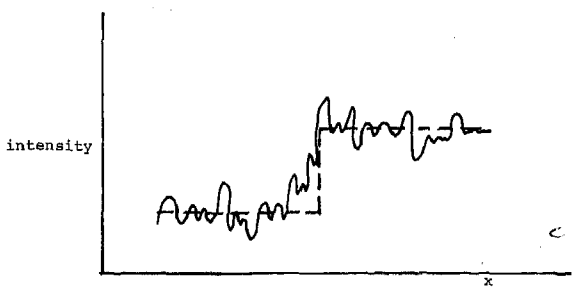

FIG. 4. A real noisy edge superimposed on an ideal edge.

"ramp"; the angle the ramp makes with the x-axis defines the steepness, or sharpness, of the edge.

2. LINEAR PARALLEL EDGE DETECTORS

By a parallel solution to the edge detection problem we mean that the decision of whether or not a set of points is on an edge is made on the basis of the grey level of the set and some set of its neighbors; but the decision is not dependent on first deciding if other sets of points lie on an edge. So the edge detection operator may in principle be applied simultaneously everywhere in the picture.

Let $\mathcal{O}$: $X \to Y$, where X and Y are classes of functions. Then $\mathcal{O}$ is called a *linear operator* if

$$\mathcal{O}(ax + bx') = a\mathcal{O}(x) + b\mathcal{O}(x'), \qquad x, x' \in X.$$

A. High-Emphasis Spatial Frequency Filtering

Perhaps the classical linear operator for picture processing is the spatial frequency filter. The relation between Fourier analysis and edge detection is that high spatial frequencies are associated with sharp changes in intensity. So, one can enhance edges by performing high-pass filtering: i.e., take the Fourier transform of the picture, say $\mathcal{F}(f(x,y)) = F(x,y)$. Multiply F by the linear spatial filter H: $E(x,y) = F(x,y) \cdot H(x,y)$. Here H is designed to attenuate the low spatial frequencies and enhance the higher ones. Blurring is a low-frequency phenomenon and so its effect is minimized by the filtering operation. Similarly, noise fills up the very high frequencies and so one designs the filter to attenuate these frequencies also. Since the frequency domain operator is a convolution in the space domain, the filtering operator is linear and parallel. The real problem here, then, is filter design. Duda and Hart [3] give a short introduction to optimal filtering.

B. Directional Differentiation

We are more interested in operations in the space domain than in the frequency domain.

Let $g(i,j)$ be the grey level at point (i,j). Then the simplest edge detector for, say, vertical edges would be:

> Compute $|g(i,j) - g(i,j+1)|$; if this value is high, then there is a vertical edge between the two points.

This is the digital analog of taking a directional derivative of the picture along the direction orthogonal to the edges we are looking for. Intuitively, this is reasonable because the derivative will be infinite at a step and will be high along the entire length of a ramp.

C. The Gradient

A related, but more versatile, operation was proposed by Roberts [15], namely the *"gradient"* $|g(i,j) - g(i+1,j+1)| + |g(i,j+1) - g(i+1,j)|$, which would detect either a horizontal or vertical edge. This operator involves only four points and is therefore extremely sensitive to noise and surface ir-

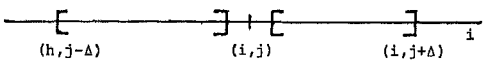

FIG. 5. Neighborhoods of length Δ of point (i,j).

regularities. A simple extension of this is to compute the difference of the average grey levels of two one-dimensional neighborhoods on opposite sides of a point (see Fig. 5):

$$\left|\frac{1}{\Delta}\sum_{n=1}^{\Delta} g(i,j+n)-\frac{1}{\Delta}\sum_{n=1}^{\Delta} g(i,j-n)\right| = \frac{1}{\Delta}\left|\sum_{n=1}^{\Delta}\left(g(i,j+n)-g(i,j-n)\right)\right|.$$

Averaging over many points reduces the effect of noise to a great extent. One can further extend this notion and compute the difference of the averages of two nonoverlapping two-dimensional neighborhoods on opposite sides of the point [16]. See Fig. 6 for an illustration of such a vertical edge detector. Using two-dimensional neighborhoods gives greater sensitivity to the edge detection operation. One would like to use circular neighborhoods, but they are computationally expensive. However, the average grey level of a $2^k \times 2^k$ square neighborhood can be computed by only $2k$ adds [8].

The size of the neighborhood is critical. One would like small neighborhoods to detect microedges, but large neighborhoods are necessary to overcome the effects of digital noise, surface irregularities, and (sometimes) textural properties of the regions [17].

3. NONLINEAR PARALLEL EDGE DETECTORS

The fundamental weakness of linear operators is that they weigh surface irregularities not only by their extent but also by their amplitudes so that, e.g., a bright point in the right neighborhood of a point can erroneously lead to the "detection" of an edge at that point.

Also, the method just described detects edges very sloppily; i.e., the value of the edge detector will be high not only on the edge, but also at points close to the edge.

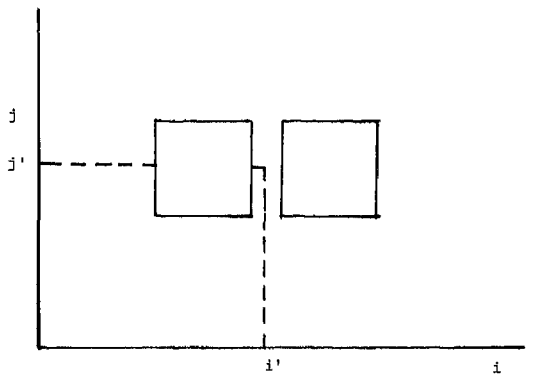

FIG. 6. Pair of nonoverlapping neighborhoods of point (i',j').

A.

Rosenfeld [16,18] has offered two solutions to the problem, both based on using neighborhoods of many sizes at every point. Computationally, this is efficient because the intermediate results of the calculation of the average grey level of a $2^k \times 2^k$ neighborhood gives the average grey level of all $2^j \times 2^j$ neighborhoods, $j < k$.

The first of the two methods takes products of the differences between the average grey levels of pairs of neighborhoods of all sizes [17,1]. That is, if E_{ij}^k is the difference of the average grey levels of a pair of $2^k \times 2^k$ neighborhoods centered at (i,j), then

$$M(i,j) = \prod_{k=1}^{m} E_{ij}^k$$

is the edge detector. $M(i,j)$ will be large only if all of the E_{ij}^k's are large. Involving large k's in the product ensures the detection of only major edges and reduces the effect of noise. At the same time, the small k's favor large values of $M(i,j)$ only very close to the actual edge. Notice that as the contrast, sharpness (i.e., steepness of the ramp), and extent of the edge increase, so does $M(i,j)$, so that a more conspicuous edge will tend to have a higher value.

A still simpler approach is not to use the same size neighborhood at each point, but to determine a largest-size neighborhood at each point and use just the E_{ij}^k of largest size. This way, minor edges due to noise near major edges (edges that are the boundaries of large regions) will not be detected, but isolated minor edges from real objects will be detected. The scheme for deciding upon the best size is: Choose the largest k such that E_{ij}^k is not significantly smaller than E_{ij}^{k-1}.

Conspicuous edges can be sharply determined by computing

$$\begin{aligned} E'_{ij} &= E_{ij}^{k_{\max}} \text{ if } E_{ij}^{k_{\max}} \geq E_{i'j'}^{k'_{\max}} \quad \text{for all } (i',j') \text{ within distance } k_{\max/2} \text{ of } (i,j), \\ &= 0 \qquad \text{otherwise.} \end{aligned}$$

where $k_{\max}$ and $k'_{\max}$ are the best sizes at (i,j) and (i',j'). Hayes and Rosenfeld [8] report good success with this edge detection scheme.

The elegance of Rosenfeld's approach to edge detection lies in its generalization to the more complex problem of finding edges between textured objects [17]. Two adjacent textured objects may share a very conspicuous edge even if the average grey levels of both objects are the same. What this means is that the average grey levels of neighborhoods about a point is not always the optimal *local property* to evaluate in order to detect an edge. But other functions of the grey level can give a measure of the degree to which a neighborhood has some textural property (e.g., we might measure the "dottedness" of a neighborhood by applying Laplacian-like operators to it). So, a reasonable approach to finding edges between textured objects might be:

1. At each point in the picture compute a local operation that measures some textural property for a neighborhood of the point. The value of the local property can be thought of as the "grey level" of the point in some texture space.

2. Apply the edge detection procedure to the preprocessed picture. Now, a point whose adjacent neighborhoods have different average "grey levels" really signifies a point whose adjacent neighborhoods differ in some average textural measure. The success of this approach, of course, depends greatly upon the choice of local properties. For a detailed discussion of the algorithm, and examples, see [8].

B.

Herskovitz and Binford [9] were part of a large M.I.T. effort to attain a thorough understanding of the world of polyhedra. Their procedure for the detection of edges in scenes containing untextured polyhedral objects is very similar to Rosenfeld's edge detector. The major difference in their approach is motivation—Herskovitz wants to find the best procedure she can for detecting edges in a very particular domain; so there is an in-depth analysis of the sources and characteristics of noise in their imaging process and of the specific edge cross sections found in the world of polyhedra. Rosenfeld, on the other hand, takes a more general approach to the problem. The Herskovitz edge detection procedure, however, is still a very general, and powerful, tool.

Consider the step function shown in Fig. 7. At every point compute the function

$$D(x) = -2g(x) + g(x+\Delta) + g(x-\Delta) = [g(x+\Delta) - g(x)] - [g(x) - g(x-\Delta)],$$

which is the difference of two slopes about the point x. Qualtitatively, $D(x)$ should be zero on flat regions, and have high absolute value when within Δ of an edge. Figure 8 shows $D(x)$ for the step function of Fig. 7. In practice, a two-sided cutoff is put on $D(x)$; i.e., if $|D(x)| < \alpha$, where α is the cutoff point, then $D(x)$ is set to 0.

Next, the function $F_s(x)$ is computed at every point.

$$F_s(x) = \sum_{i=1}^{\Delta} sg(D(x+i)) - \sum_{i=1}^{\Delta} sg(D(x-i)),$$

where $sg(x) = 1$ if $x > 0$, -1 if $x < 0$, and 0 if $x = 0$. (This is all really done of course using two-dimensional neighborhoods, so that

$$F_s(x) = \sum_{\substack{x \in \text{right} \\ \text{neighborhood}}} sg(D(x)) \quad - \quad \sum_{\substack{x \in \text{left} \\ \text{neighborhood}}} sg(D(x)).$$

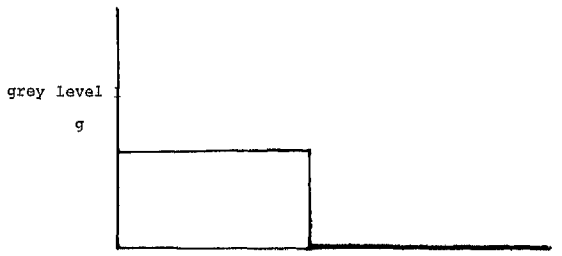

FIG. 7. A step function of height h.

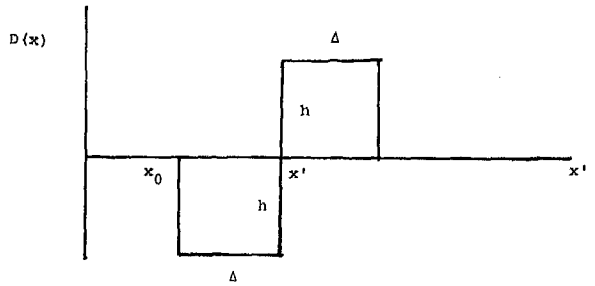

FIG. 8. $D(x)$ for the step function of Fig. 7.

This is the difference between the excess of positive over negative $D(x)$ values in the two neighborhoods of x. Figure 9 shows $F_s(x)$ for the step function of Fig. 7. Finally, local maxima of $F_s(x)$ are found, and a line fitting procedure builds the complete edge. This approach is also insensitive to minor edges that are close to major edges.

Another interesting point is that Herskovitz uses edge detectors of one size only. In general, this is not sufficient because, as Rosenfeld and Thurston show, one should use edge detectors of different sizes in order to discern both major and minor edges.

Although she does not discuss extensions to textured objects, Herskovitz' approach can be generalized to deal with texture (in the same way as Rosenfeld's). Call the edge of Fig. 1 a positive edge, and its mirror image a negative edge. Call the top step the plateau, and the bottom the valley. A high positive value of $D(x)$ is a bit of "evidence" that x is in the valley, close to the plateau (i.e., within a distance Δ) and a high negative value is "evidence" that x is on the plateau, within Δ of the valley. Now, when we compute

$$\sum_{x \in \text{right neighborhood}} Sg(D(x)),$$

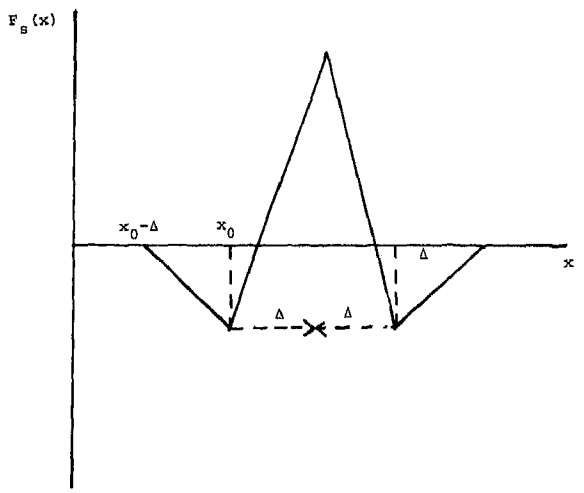

FIG. 9. F_s for the step function of Fig. 7.

we are collecting all the evidence from the right neighborhood and getting a weighted "vote" of whether the right neighborhood is a valley, a plateau, or sees no edge in sight (of course, within distance Δ). Similarly we take the vote for the left neighborhood. If both votes are high in absolute value, but different in sign, then their difference will signify an edge near x. The sign of their difference distinguishes between positive and negative edges. If $D(x)$ is replaced by some function that evaluates a textural property (e.g., a Laplacian of x and $x + \Delta x$ can allow x to vote "I am spottier than $x + \Delta x$," "$x + \Delta x$ is spottier than I," or "No spots in sight"), then evaluating $\sum_x Sg(D(x))$ can still be meaningful, and can be used to detect textural edges.

4. OPTIMAL APPROACHES TO EDGE DETECTION

All of the previously discussed edge detectors were heuristic; i.e., there was no formal model of "edge" associated with them. They were all based on the extraction of the most prevalent features of edges—high derivatives; large extent along the direction of the edge. So, one could not discuss their optimality; only informal arguments could be provided to analyze their performance. In this section we will discuss three approaches to boundary detection based on formal models of edges.

A. Griffith's Operator

Griffith [4,5,6] discusses the optimal use of intensity information to detect edges in the blocks world. His analysis assumes no knowledge of a scene other than the types of edges (step, spike) that one should expect.

Let $I(x,y)$ be a real picture function, and let $J(x,y)$ be the noisy, blurred version of the real picture that the computer gets to see. It is assumed that the noise is white noise, and is normally distributed about the real intensity with a constant variance independent of intensity, and that the blurring (spatial distortion) can be accounted for by the convolution of $I(x,y)$ with $g(x,y)$, the point spread function of the imaging system. Let $I^*(x,y)$ denote the distorted, but noise-free, picture.

The key question then is: What is the value of $P[J(x,y)|I(x,y)]$ (from now on abbreviated $P(J|I)$), i.e., what is the probability of getting the noisy, distorted picture J_i given the original picture I? Note that this probability is nonzero for all i, since the noise process is Gaussian.

In particular, we are going to let I be the set of all long, narrow rectangular bands in a larger picture (and J will be their noisy counterparts) and we will develop an optimal decision procedure to determine whether or not an edge is exactly centered in one of these regions. Now, since the decision procedure will depend on the values of the intensities within a region, and not on intensity values outside the region, we should discuss to what extent such a regional process is less optimal than a global one that uses information from the entire picture. Qualitatively, the regional predicate will be as good as a global one only if intensity values far from the position of a proposed edge do not yield relevant information as to the existence of the edge. Two factors do limit the usefulness of remote information:

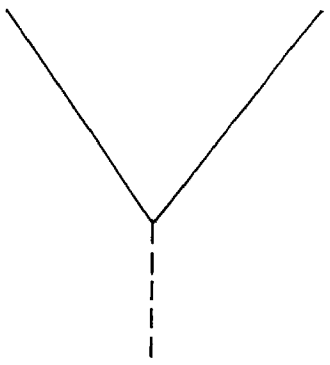

FIG. 10. The downfall of the regional predicate.

1. Due to nonuniformities in illumination and reflectivity, one cannot make reliable predictions about remote intensities.
2. The effect of an edge on the derivatives falls off drastically as one leaves the vicinity of an edge, and it is derivative information that characterizes an edge.

Implicit in the above discussion is that if one considers intensity values only, then edge is a local property of a picture. Since an edge is really not such a local phenomenon, the regional predicate will not be as optimal as a global one. For example, in Fig. 10, the regional predicate would miss the faint dotted line, but a more informed global predicate would not miss it.

In the following analysis, it is assumed that at most one edge or line exists in any one rectangular band. We need the following definitions.

1. $P(I_i,J_j)$: the joint probability of occurrence of the ith noise-free sample and the jth noisy sample.
2. $P(I_i)$, $P(J_j)$: the a priori probabilities of I_i and J_j.
3. $P(I_i|J_j)$: the conditional probability of I_i given J_j.
4. False Positive Error (FPE): the decision procedure says there is an edge in a region when in fact there is none.
5. True Negative Error (TNE): missing an edge that is in the picture.
6. $G:\{I\} \rightarrow \{0,1\}$: G maps all I_i with edges or lines down their centers into 1, and all other I_i into 0. G is called an identification function.
7. $Q(J) = \sum_{G(I_i)=1} P(I_i|J)$: the probability that the noisy region J corresponds to a line or edge.

Griffith proves the following theorem: We can threshold $Q(J)$ at a point that will keep the FPE rate constant while minimizing the TNE rate. So the problem is to determine a closed form for $Q(J)$.

First, using Bayes' rule we can rewrite $Q(J)$ as

$$Q(J) = \frac{\sum_{G(I_i)=1} P(J|I_i) \cdot P(I_i)}{\sum_{G(I_i)=1} P(J|I_i) \cdot P(I_i) + \sum_{G(I_i)=0} P(J|I_i) \cdot P(I_i)}. \tag{1}$$

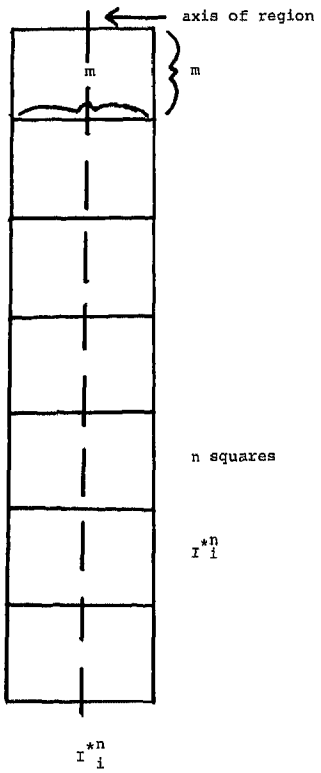

FIGURE 11

We want to be able to determine Eq. (1).

1. The relationship of $I(x,y)$ and I^* (x,y). Since there is a deterministic relationship between I and I^*, we can achieve a solution for (1) if we can determine $P(J|I^*_i)$ and $P(I^*_i)$. In this section we will derive a closed form for $P(I^*_i)$.

Let us view a region as composed of n $m \times m$ squares, as in Fig. 11. We will denote the distorted version of such a region by I_i^{*n}.

We need a model for $\{I_i^{*n}\}$. The set will be partitioned into four subsets:

(1) CL: regions with lines down the center,
(2) CE: regions with edges down the center,
(3) CH: homogeneous regions,
(4) CS: regions with line or edge not centered.

We can then analyze the class CL as follows.

(i) We assume that the intensities across the jth subsquare of I_i^{*n} can be described by

$$a_{ij}\, g(x) + b_{ij},$$

where a_{ij} is the relative amplitude of the line in the jth subsquare, x is the perpendicular distance between the point x and the axis of the region, and b_{ij} is the background intensity. By the relative amplitude we mean the difference in intensity between the line and the background.

Also, since we will be dealing with relative amplitudes we will ignore the b_{ij}.

(ii) It is also assumed that a_{ij} is uniform for the jth subsquare, so we can denote I_i^{*n} by $I_i^{*n}(a_{ij}, \ldots, a_{in})$.

(iii) The a_{ij}'s are normally distributed about the real amplitude with a very small variance, σ_N^2, i.e., there are minor perturbations of relative amplitudes due to factors like nicks on a block.

So, let $F(a)$ be the a priori probability of a line in the real world with ideal amplitude a; then the probability of a particular distorted line $I_i^{*n}(a_{i1}, \ldots, a_{in})$ arising from this ideal line is

$$F(a) \prod_{j=1}^{n} \frac{1}{\sigma_N(2\pi)^{1/2}} \exp\{-[(a - a_{ij})/\sigma_N]^2\} da_{i1} \cdots da_{in}.$$

So

$$P(I_i^{*n}(a_{i1}, \ldots, a_{in})) = \sum_a F(a) \prod_{j=1}^{n} \frac{1}{\sigma_N(2\pi)^{1/2}} \exp\{-[(a - a_{ij})/\sigma_N]^2\} da_{i1} \cdots da_{in}. \quad (2)$$

Assuming that $F(a)$ is normal, with variance ρ_n^2, we have

$$F(a) = [P(CL)/\rho_n(2\pi)^{1/2}] \exp[-(a/\rho_n)^2] da,$$

where $P(CL)$ is the a priori probability of a member of CL. The analysis assumes that $P(CL)$ is given.

A similar analysis can be made for the sets CE and CH.

2. The relationship between J and $I_i^{*n}(a_{i1}, \ldots, a_{in})$. We assume that the intensity at a point is normally distributed with variance r about the real intensity, and that the noise induced deviations at different points are statistically independent. So, the probability of getting the set of intensities $\{V_{i1}, \ldots, V_{im}\}$ from the set $\{U_{i1}, \ldots, U_{im}\}$ is

$$P(J|I^*) = \sum_{i=1}^{n} \sum_{j=1}^{m} \frac{1}{r(2\pi)^{1/2}} \exp\{-[(U_{ij} - V_{ij})/r]^2\} dV_{11} dV_{12} \cdots dV_{nm}. \quad (3)$$

Here, we are measuring the intensities in the jth subsquare with one scan normal to the axis of the region. This is reasonable because we have assumed a high level of uniformity within each subsquare.

The assumption of statistical independence of the individual deviations is reasonable because we are restricting our attention to white, untextured blocks with very nominal surface irregularity. In a less restricted environment, this assumption would be suspect.

We can make a similar analysis for the classes CE and CH.

Now, make the following definitions.

(1) $$CL(J_j^n) = \sum_i P(J_j^n | I_i^{*n}(a_{i1}, \ldots, a_{in})) \cdot P(I_i^{*n}(a_{i1}, \ldots, a_{in})),$$

where $a_{i1}, \ldots, a_{in}$ are the relative amplitudes of some line centered in a region

$$(2)\quad CE(J_j^n) = \sum_i P(J_j^n | I_i^{*n}(b_{i1}, \ldots, b_{in})) \cdot P(I_i^{*n}(b_{i1}, \ldots, b_{in})),$$

where $b_{i1}, \ldots, b_{in}$ are the relative amplitudes of some edge centered in a region

$$(3)\quad CH(J_j^n) = \sum_i P(J_j^n | I_i^{*n}(c_{i1}, \ldots, c_{in})) \cdot P(I_i^{*n}(c_{i1}, \ldots, c_{i1})),$$

where $c_{i1}, \ldots, c_{in}$ are the relative amplitudes of some homogeneous region

$$(4)\quad CS(J_j^n) = \sum_i P(J_j^n | I_i^{*n}) \cdot P(I_i^{*n}),$$

where the notation I_i^{*n} will from here on refer exclusively to regions in the class *CS*.

Because of the deterministic correspondence between distorted and undistorted noise-free samples, we see that

$$\sum_{G(I_i)=1} P(J_j^n | I_i) P(I_i) = CL(J_j^n) + CE(J_j^n)$$

and

$$\sum_{G(I_i)=0} P(J_j^n | I_i) \cdot P(I_i) = CH(J_j^n) + CS(J_j^n).$$

So, we can rewrite Eq. (1) as

$$Q(J_j^n) = \frac{CL(J_j^n) + CE(J_j^n)}{CL(J_j^n) + CE(J_j^n) + CH(J_j^n) + CS(J_j^n)}.$$

Griffith offers no model for the class *CS*. Instead, he argues that the class can be ignored on the grounds that

i. when the region J_j^n does not contain a skewed edge or line, the value of $CS(J_j^n)$ will be low.
ii. when the region J_j^n does contain a skewed line or edge, the value of $CS(J_j^n)$ is high, but Q will take on a local maximum in the region almost exactly centered on that line or edge. So by a procedure very similar to Rosenfeld's nonmaximum suppression, we can still detect the line or edge sharply.

The problem with not constructing a model for the class *CS* is that it results in a very high FPE rate in the edge detector and necessitates the use of a global "noise cleaning" operation after the predicate is applied. The optimality of the procedure can no longer be determined analytically.

Griffith ran into computational difficulties in determining values for Eqs. (2) and (3) possibly due to having to enumerate all possible edges, lines, etc. So, he had to simplify his procedure, and instead of computing, e.g., $P(J_j^n / I_i^{*n}(a_{i1}, \ldots, aa_{in}))$ for all possible n-tuples arising from lines centered in a region, he just computes the similarity between J_j^n and some paradigm intensity profile of a distorted line (similarly for an edge and a homogeneous region).

In summary, then, the major weaknesses of Griffith's approach to edge detection are:

1. Many of the assumptions hold only for the restricted block world, and it is not clear that the analysis could be extended to objects that were inherently noisy, or to textured objects.
2. The extension to include curved surfaces is also not clear.
3. His arguments for ignoring the class *CS* are not compelling and result in a very high (0.7) FPE rate.

B. Hueckel's Operator

An alternative to Griffith's approach is: Instead of asking whether or not a region contains a centered line or edge, ask what edge element will best fit the intensities in a given region.

This is Hueckel's [10] approach to edge detection. Hueckel's model of an edge is a step function F in a circular disc. Let

$$F(x,y,c,s,\rho,b,d) = \begin{cases} b, & cx + sy \leq \rho, \\ b + d, & cx + sy > \rho, \end{cases}$$

where the x–y coordinate system has its origin at the center of the circular region. Clearly, there is a 1–1 correspondence between ideal edges F and the quintuples (c,s,ρ,b,d).

Let $E(x,y)$ be an obtained intensity function in the circular disc. Then we approximate the empirical edge element E with the ideal edge F' that minimizes the distance between E and F'; i.e., we minimize:

$$\iint [E(x,y) - F(x,y,c,s,\rho,b,d)]^2 dxdy. \tag{4}$$

Hueckel's operator is an efficient solution to the minimization problem (4). The technique used is series expansion and truncation in the frequency domain. Specifically, he takes as his basis functions that are separable into a product of an angular and radial component; i.e., he does Fourier analysis in polar coordinates.

Let $\{H_i\}_{i=0}^{\infty}$ be the basis. Then define

$$a_i = \int H_i(x,y)\, E(x,y)\, dxdy,$$

$$s_i = \int H_i(x,y)\, F(x,y,c,s,\rho,b,d)\, dxdy.$$

Notice that the a_i are constants and the s_i are variables.

Then minimizing (4) is equivalent to minimizing

$$\sum_{i=0}^{\infty} (a_i - s_i)^2. \tag{5}$$

However, only the first eight coefficients are used because

1. real edges are blurred, and blurring removes high spatial frequencies,
2. noise predominates in the high spatial frequencies.

The minimization procedure returns both the best edge and a measure of the goodness of the edge.

Possible criticisms of Hueckel's approach are that he offers no analytical model for the relationship between the noise process and noise level and the performance of the operator (in fact no examples of raw edge pictures are given). So one cannot qualitatively determine a priori the error rates of the operator, or any other measures of its sensitivity.

C. Chow's Operator

Chow [2] has studied the problem of detecting the boundary of the heart in a cineagiogram (a motion picture of the heart). The input to the procedure is an average of several frames of the cineangiogram.

The method used is a variation of a very standard approach. Take a grey-level histogram of the entire picture. Ideally, the form of the histogram should be as in Fig. 12, a bimodal histogram. One of the hills represents the background and the other the object (depending on whether we have bright objects on a dark background or vice versa). We can then threshold the picture at the valley of the histogram, and any transition from a point in the picture whose grey level is on one side of the threshold to an adjacent point whose grey level is on the other side of the threshold defines a boundary position. Equivalently we can construct a binary picture once the threshold is selected by setting all points whose grey level is less than the threshold to 0, and all others to 1. A boundary position is then some 0–1 transition. There are two weak assumptions in this approach:

1. that the histogram will be bimodal—in fact for many classes of pictures this will not be the case (see Weszka, Nagel, and Rosenfeld [21]);
2. that even if the histogram were bimodal, there might not be any good global threshold. Here, consider a scene with a high luminance gradient in one direction.

So, we have to abandon the simple-minded approach and attempt to devise a dynamic threshold method based on statistical principles.

First, notice that we can view the grey-level histogram as a probability frequency distribution simply by dividing the absolute frequency of a grey level by the number of points in the picture to transform it into a relative frequency. Now, we make the assumption that the probability distribution (relative

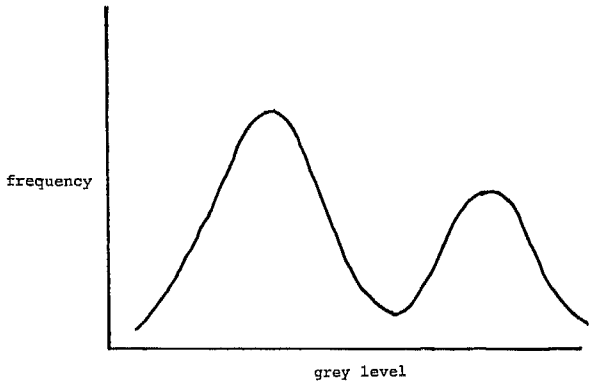

FIGURE 12

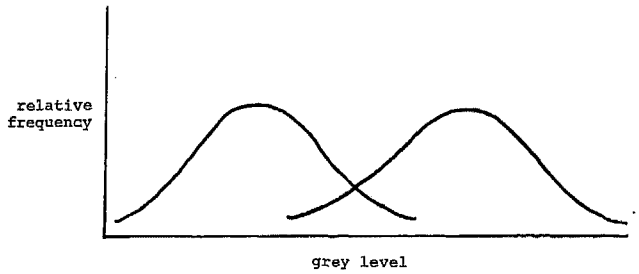

FIGURE 13

frequency histogram) of any small region of the picture that contains only object or background is unimodal.

So a region that contains both object and background will be a mixture of the two distributions—the one that defines the background and the one that defines the object (see Fig. 13). In order to separate the mixture into its components we must know:

i. the functional form of the unimodal distributions. Here it is assumed they are normal, and empirical investigations justify this assumption.
ii. If it is possible to determine two distributions from their mixture. For the case of two normal distributions, it is.

The threshold determination method starts by separating the picture into regions with 50% overlap. Since the frequency distribution of any region is the sum of two normal distributions, we can describe the rth region by

$$f_r(x) = \sum_{k=1}^{2} \frac{\rho_{kr}}{\sigma_{kr}(2\pi)^{1/2}} \exp\left[-(X - \mu_{kr})^2/2\sigma_{kr}^2\right], \tag{6}$$

where ρ_{1r} and ρ_{2r} are the theoretical fraction of area occupied by the object and background, respectively, $(\rho_{1r} + \rho_{2r} = 1)$; μ_{1r}, μ_{2r}, σ_{1r}, σ_{2r} are the mean and standard deviation of the object and background, respectively.

It can be shown that the mean and variance of the mixture are given by

$$\mu_r = \rho_{1r}\,\mu_{1r} + \rho_{2r}\,\mu_{2r}$$

and

$$\sigma_r^2 = \rho_{1r}\,\sigma_{1r}^2 + \rho_{2r}\,\sigma_{2r}^2 + \rho_{1r}\,\rho_{2r}\,(\mu_{1r} - \mu_{2r})^2$$

Empirically, it was determined that regions with large values of σ_r^2 (which of course can be directly computed from the grey-level frequency distribution) usually contain boundaries. For each region with large variance, the five parameters of (6) are least-squares fitted by a hill climbing method. Next, the bimodality of the regions is measured by computing the valley-to-peak ratios

$$\delta = \frac{\text{minimum of } f(\mu_r) \text{ in } [\mu_{1r}, \mu_{2r}]}{\text{minimum } [\mathrm{F}(\mu_{1r}), f(\mu_{2r})]},$$

where $f(\mu_r)$ is the obtained distribution in the rth region and $f(\mu_{1r})$ and $f(\mu_{2r})$ are the two approximated distributions.

A threshold t_r is determined for each region that has a sufficiently high δ-measure. Any point whose intensity is greater than or equal to t_r will be called object; any point whose intensity is less than t_r will be called background. We want to pick t_r in a way that minimizes the probability of misclassifying a point (i.e., minimizes the sum of the FPE's—calling a background point object; and TNE's—calling an object point background). It is well known that if the cost of a FPE is the same as the cost of a TNE, then t_r is the value of x that satisfies the equation

$$\log f(\mu_{1r}) - \log f(\mu_{2r}) = 0.$$

This turns out to be a simple quadratic equation in x.

The next step is to compute thresholds for regions that did not exhibit the required amount of bimodality. Here a simple interpolation scheme is used: The threshold of a region is set to a weighted average of its own threshold (if it already had one) and its neighbors'. Finally, thresholds are assigned to each point by another interpolation procedure that depends on the distance of a point from nearby regions.

The two interpolation procedures are heuristic, and so the final threshold assignment is not guaranteed to be optimal Also, the procedure is not directly extendable to scenes with many objects—the thresholding procedures would tend to smooth objects into each other and one would not know a priori the type of mixture for any one region. Finally, more efficient and simpler threshold detection techniques exist (Weszka, Nagel, and Rosenfeld [21]) that seem to work on images of quality comparable to cardiac cineangiograms.

5. SEQUENTIAL EDGE DETECTION

In contrast to parallel edge detectors, when applying an edge operator sequentially to be a picture, the result at a point is contingent upon the results of the operator at previously examined points.

The critical components of a sequential edge detection procedure are:

1. Picking a good initial point. The performance of the entire procedure will depend upon the choice of a good starting point. Ordinarily, some simplifying assumption is made to facilitate that choice (e.g., the edge begins in the top row of the picture).

2. The dependence structure. How do the results obtained at previously examined points affect both the choice of the next point to be examined and the result at that next point?

3. A termination criterion. There must be a way for the procedure to determine that it is finished.

A. Heuristic Search

A classical sequential search procedure is the heuristic search of a state-space for some minimal cost path to a goal using the merit-ordering function $f(n) = g(n) + h(n)$ to decide which unexamined node to examine next [14]. Ordinarily, f is interpreted as a cost function where g is the cost to get to the state n from some starting state and h is (an estimate of) the cost to get from state n to some

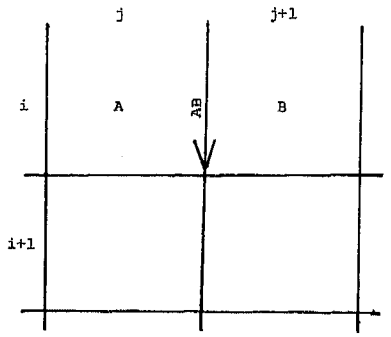

FIG. 14. Definition of the edge element AB.

goal state. (Note: The assumption that g and h are additive means that they must be measured in comparable units).

Martelli [12] has attempted to embed an edge detection problem into a state-space representation. Specifically, he is looking for a high-contrast edge that starts in the top row of the picture (initial points are therefore trivial to find) and ends in the bottom row (providing a termination criterion).

The states are edge elements defined by two points (see Fig. 14). For example, the points $A = (i,j)$, $B = (i,j+1)$ define the directed edge element AB. The direction of the edge element is clockwise from the first point to the second. An edge is a sequence of adjacent edge elements that starts in the top row, ends in the bottom row, contains no loops, and has no element whose direction is "up." So an edge is a path in the graph that represents the state-space and the problem of finding the best edge in a picture reduces to the problem of finding an optimal path in the graph.

Suppose Max is the maximum gray level in the picture. Martelli defines the cost associated with a node (state) $n = A_r A_s$ as $C(n) = (\text{Max} - g(A_r) + g(A_s))$ where g is the gray level at point A. Notice that $C(n)$ is low if A_r has high gray level and A_s low gray level. The cost, $g(n)$, of getting to state n from some starting state is the sum of the costs (C's) encountered along the path from that start state to state n. The cost $g(n)$ of state n is low if the edge defined by the path to n is of high contrast. The particular choice of measurement unit for g, a sum of grey-level differences, makes it very difficult to construct a heuristic function h because we can characterize neither the set of goal states nor a good edge with respect to this unit of measurement. The problem is compounded by a poor choice of representation—the goal of the search is to find the edge, and yet the states in the space do not represent partial edges, but only microedge elements. If the states did represent partial edges then one could also redefine the g function as the average contrast and this would facilitate defining a heuristic function. But Martelli does not do this; consequently his search is not guided by a heuristic function and he therefore has to impose restrictions on the search procedure that transcend the merit ordering; namely, a state is not expanded if a state already exists three rows closer to the bottom of the picture than the first state. There is thus also no guarantee of finding an optimal path with respect to the cost function, but only a "good" one.

The root of some of these problems is the assumption that a reasonable measurement scheme for $f(n)$ is an additive one. It would perhpas be wiser to use a different measurement structure based on vectors

$$f(n_1) = (g(n_1),h(n_1)),$$
$$f(n_2) = (g(n_2),h(n_2)).$$

Here one could say that if $g(n_1) < g(n_2)$, then $f(n_1) < f(n_2)$, (or it might be better to say $f(n_1) < f(n_2)$ if $g(n_1) < g(n_2) + t$). If the first test fails, i.e., $g(n_1) \not< g(n_2)$, *then* if $h(n_1) < h(n_2)$, then $f(n_1) < f(n_2)$. Here we have formally given h the role as a "tie-breaker." So, e.g., $g(n)$ might be the average contrast of the partial edge n and $h(n)$ might be the number of rows between the end of the partial edge n and the bottom of the picture. In this way, the fact that the criterion for determining the "goodness" of a partial edge and the "difference" between a partial edge and goal are not comparable is made explicit. All of this is not intended to suggest that this is a good measurement structure for edge detection (it certainly is not), but that Martelli's choice of merit ordering is unsatisfying.

A certain amount of globality is introduced using heuristic search. Suppose some path in the graph represents a very good partial edge, but that all of the successors of the last node on the path are of low contrast. The heuristic search method would force the best continuation of the partial edge, while a parallel scheme (say the Roberts gradient) would have rejected all of the successors.

While it is true that by progressing from a parallel scheme to heuristic search we have replaced a local decision procedure by a global one and that a global decision procedure is in general a more informed and optimal procedure than a local one, the standard notion of heuristic search in a state space is an incomplete model because:

1. the notion of a goal node is ill-defined with respect to edges. Martelli's definition of goal node is a contrivance, and, basically, finding the ideal edge form is an illusion because
   a. an edge is a probabilistic phenomenon, and
   b. a large variety of distinct and incomparable features may be the distinguishing characteristics of inherently incomparable edges; which leads us to the fact that
2. a static merit ordering affords us little more flexibility than a static local predicate; i.e., the criterion that should be employed to determine the best edge should also be contingent upon the previous results of the search process.

What all of this seems to imply is that a conventional state-space representation is too structured an environment for edge detection. The construction of evaluation fuctions is a complex and sometimes hopeless task (How do you build into the evaluation function, e.g., assurance that an edge will never intersect itself?).

B. Dynamic Programming

Martelli's work draws heavily on Montanari's study [13], which used dynamic programming to detect systems of curves. Dynamic programming as used here is analogous to uniform cost, or breadth-first search. By representing all possible sequences of points that can define, e.g., smooth curves as paths through a graph, the combinatorics of the problem can be reduced because many curves share identical initial segments. Dynamic programming is a technique for doing the bookkeeping for a "reverse" breadth-first search. A cost function, or figure of merit, determines the relative value of different paths, but the cost function is not used to guide the search, but rather to determine the best path once they have all been enumerated.

Now, under certain conditions dynamic programming is not the most efficient way to achieve an optimal solution. For example, if the merit function and length of the solution path are bounded above by F and L, respectively, and below by 0, then at stage k of the optimization procedure no path that differs from the path of maximal merit at stage k by $(L-k) \times F$ need be considered at the next stage because it can obviously not be extended to an optimal path. On the other hand, if nothing is known about, say, the length of an optimal solution, then no cutoffs can be made and optimality can be achieved only by extending all paths (of course, there must be some termination criteria).

Montanari specifically discusses finding a smooth, dark curve of fixed length. The curve is embedded in a noise background, but since the merit function does not guide the search, the computation time is independent of the noise level (which would not be the case if the merit function guided the search). The figure of merit of a path $z_1, \ldots, z_n$ is

$$f(z_1, \ldots, z_n) = \sum_{i=1}^{n} g(z_i) - q \sum_{i=2}^{n-1} (d(z_{i+1},z_i) - d(z_i,z_{i-1}) \bmod 8),$$

where $g(x)$ is the grey level of x, $d(x,y)$ is the slope between points x and y, so the second term is proportional to curvature. The following constraints are placed on the solution:

$$\left.\begin{array}{l} \max(|x_{i+1} - x_i|, |y_{i+1} - y_i|) = 1, \\ (d(z_{i+1},z_i) - d(z_i,z_{i-1}) \bmod 8) \leq 1, \end{array}\right\} \quad z_i = (x_i,y_i).$$

The procedure can be directly extended to detecting smooth high contrast (either positive or negative) edges of fixed length by replacing the original picture with, say, the gradient of the picture or a directional derivative (with a positive or negative cutoff at zero). In this case, since the absolute value of the gradient is bounded, we can cut off consideration of some paths at each stage.

Designing a figure of merit is an art and suffers from the same difficulties encountered in constructing cost functions. The price of global optimality using dynamic programming is high execution time and large memory requirements; these can be reduced if one is willing to settle for a locally optimal procedure (e.g., instead of finding the best curve in an $m \times n$ picture, find the best curve in each of the two $m/2 \times n$ pictures—now the entire procedure can be carried out

in parallel on the two halves of the picture but the solutions are only locally optimal).

C. Guided Edge Detection

The fundamental fallacy of all the edge detection schemes discussed so far is the assumption that the edge detector is a box in a hierarchical scene analysis system and that its decisions are not dynamically influenced by the results of other components in the system. This is not to say that autonomous (or bottom up) edge detection has no place in a scene analysis system (see Rosenfeld [19]), but that it is not sufficient to consider only bottom-up edge detection.

An extensive study of control structures, backtracking criteria, "model" design, etc., is beyond the scope of this paper. However, we can informally and intuitively discuss several schemes for "guiding" the edge detection process. None of the ideas that will be discussed are grounded in any established theory (e.g., the theory of heuristic search) but rather reflect the state of the art and derive their justification from their performance.

A simple, yet powerful, guidance scheme introduced by Kelly [11] is "planning." Suppose we wish to find the outline of a face, in a picture that contains just the face, as a first stage in face recognition. Since the outline of the face is of relatively large extent, if the picture is shrunk (by, say, taking averages over nonoverlapping regions) and the reduced image is searched for edges, then the edges detected in the reduced image can serve as a plan, or guide, for finding the facial outline and other features in the original picture.

The search combinatorics are also reduced. The outline of a face gives powerful clues to the positions of other spatial features. If the original picture were searched for the outline, many more points would have to be examined. Also,

1. there will be more false edges due to noise and quantization in the original picture than in the shrunken one, and
2. a real edge will sometimes temporarily vanish in a digital picture. Once the image is reduced, however, the probability of having a hole in the outline of the face is diminished.

Kelly notes that planning is essentially equivalent to running coarse edge detectors on the original picture, but that when you think about what is being done in terms of forming plans rather than doing coarse edge detection, your insight into the problem is increased.

A supplementary guidance mechanism is to give the edge detector a structural, topological description of the generic scene it will encounter (Harlow [7]). For example, the description could be in the form of a tree where the nodes of the tree represent features and the arcs represent relations between features (e.g., "to the left of," "below," etc.). Of course, this approach is not completely general, but it does provide an added degree of flexibility to a scene analysis system that would otherwise have to have a description of a particular type of scene built into it, rather than accept that description as input.

Shirai [20] has successfully constructed a heterarchical system that "under-

stands" the blocks world. His effort is the culmination of the many years of study of polyhedra at M.I.T.

The two aspects of Shirai's work that particularly apply to edge detection are:

1. the way he handles blurred edges, and
2. dynamic threshold selection.

Recall that blurred edges are a fact of life in TV scenes of the blocks world and were treated empirically by Herskovitz and Binford [9] and theoretically by Griffith [4]. Shirai adopts a variation of the Rosenfeld–Thurston edge detector, where instead of assigning the edges to local maxima, he uses the maxima to determine a shoulder, or plateau, in the edge detector outputs and then assigns the edge to the midpoint of the plateau. This is called feature point selection.

The rationale for doing this is that the location of the peak is more noise sensitive than the extent of the shoulder. Unlike Rosenfeld, Shirai uses only one size edge detector.

The heterarchical nature of the system provides for and dictates that the edge detection component be in continuous communication with other parts of the system, once sequential processing begins. Shirai initiates scene analysis with a "planning" stage to extract the outline of a cluster of objects in the scene. After the first step, the system becomes heterarchical. The outline obtained is a plan that guides the proposal of other borders in the scene. Borders (between objects or between faces of the same object) are proposed by applying certain heuristics to the plan (e.g., if two of the contour lines in the plan form a concavity, find a collinear extension of them); when new borders are found, the plan is updated to reflect the new knowledge and then another line is proposed. The edge finding and tracking algorithms are the particularly relevant features of the procedure.

The line proposer specifies a point which, according to the heuristics, is a prime candidate for the beginning of a new border. The direction of the proposed line is unknown, but the range of directions is determined by the heuristic that led to the proposal. Lines are then searched for in overlapping angular segments of size α and area β (see Fig. 15) by determining all feature points in each segment and fitting these points with a least-squares line. The deviation of the fit yields a measure of the goodness of the line segment. The threshold for the

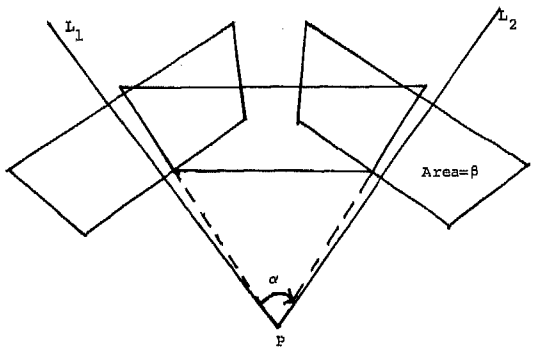

FIG. 15. Circular search segments of size α and area β.

acceptance of feature points is not the same threshold used to extract the original plan, but is set according to local context. Once a good initial segment is found by this circular search process, the system predicts the location of feature points to extend the new line. It then searches for a new feature point to extend the line, and there are four possible outcomes of the search:

1. There is no acceptable feature point in the predicted search area. In this case a "nervousness" parameter, m, is incremented. If m ever gets too high during tracking, then tracking terminates.
2. A feature point is on the line. What this really means is that the perpendicular distance, D, between the new feature point and the least-squares line fitted to previous feature points is less than a dynamically determined threshold $D1$; $D1 = C1 + C2n$, where the value of n is initially zero. When a new point is found, any points that were previously classified as doubtful (see Eq. (4)) are put on the line, and a new least-squares line is fitted, and m and n are reset to 0.
3. A feature point is off the line; i.e., $D \geq D2$, where $D2$ is a static threshold.
4. An ambiguous feature point is found; $D1 < D < D2$. Here n is incremented and m is decremented. If the sum, $m + n$, ever gets too high, then tracking terminates.

It should be noted in closing that the integration of edge detection techniques into a heterarchical system (community of experts) that does not possess as neat a hierarchy of structures as the blocks domain (line–corner–object) is as yet an unexplored problem and is part of the larger unsolved problem of the design of scene analysis systems that combine both goal-directed, or modal driven, processes; and bottom-up, or data driven, processes.

REFERENCES

1. M. F. Abbamonte, G. Johnston, Y. Lee, R. Nagel, A. Rosenfeld, and M. Thurston, Edge and curve enhancement in digital pictures, 2, Univ. of Maryland Tech. Rep. 70–103, 1970.
2. C. K. Chow and T. Kaneko, Automatic boundary detection of the left ventricle from cineangiograms, Comput. Biomed. Res. **5**, 388–410, 1972.
3. R. Duda and P. Hart, *Pattern Classification and Scene Analysis*, Wiley, New York, 1973.
4. A. K. Griffith, Computer recognition of prismatic solids, MAC-TR-73, 1970.
5. A. K. Griffith, Edge detection in simple scenes using a priori information, *IEEE Trans. Computers* **C-22**, 1973, 371–381.
6. A. K. Griffith, Mathematical models for automatic line detection, *J. Assoc. Comput. Mach.* **20**, 1973, 62–80.
7. C. Harlow and S. Eisenbeis, The analysis of radiographic images, *IEEE Trans. Computers* **C-22**, 1973, 678–689.
8. K. Hayes and A. Rosenfeld, Efficient edge detectors and applications, Univ. of Maryland Tech. Rep. 207, 1972.
9. A. Herskovitz and T. Binford, On boundary detection, M.I.T., AI Memo 183, 1970.
10. M. Hueckel, An operator which locates edges in digital pictures, *J. Assoc. Comput. Mach.* **18**, 1971, 113–125.
11. M. Kelly, Edge detection by computer using planning, in *Machine Intelligence VI*, Edinburgh University Press, Edinburgh, 1971, pp. 397–409.
12. A. Martelli, Edge detection using heuristic search methods, *Computer Graphics Image Processing* **1**, 1972, 169–182.

13. U. Montanari, On the optimal detection of curves in noisy pictures, *Comm. Assoc. Comput. Mach.* **14**, 1971, 335–345.
14. N. Nilsson, *Problem Solving Methods in Artificial Intelligence*, McGraw–Hill, New York, 1971.
15. L. G. Roberts, Machine perception of three dimensional solids, in *Optical and Electro-Optical Information Processing* (J. Tippett, D. Berkowitz, L. Clapp, C. Koester, A. Vanderburgh, Eds.), M.I.T. Press, 1965, pp. 159–197.
16. A. Rosenfeld, R. Thomas, and Y. Lee, Edge and curve enhancement in digital pictures, Univ. of Maryland Tech. Rep. 1969, pp. 69–93.
17. A. Rosenfeld and M. Thurston, Edge and curve detection for visual scene analysis, *IEEE Trans. Computers* **C-20**, 1971, 562–569.
18. A. Rosenfeld and Y. Lee, Edge and curve detection, further experiments, *IEEE Trans. Computers* **C-21**, 1972, 677–715.
19. A. Rosenfeld, Non-purposive perception in computer vision, Univ. of Maryland Tech. Rep. 219, 1973.
20. Y. Shirai, Understanding intensity arrays, M.I.T. A.I. Memo 263, 1973.
21. J. Weszka, R. Nagel, and A. Rosenfeld, A technique for facilitating threshold selection for object extraction from digital pictures, Univ. of Maryland Tech. Rep. 243, 1973.