A Cost Minimization Approach to Edge Detection Using Simulated Annealing

Hin Leong Tan, Saul B. Gelfand, *Member, IEEE*, and Edward J. Delp, *Senior Member, IEEE*

***Abstract*—In this paper, we cast edge detection as a problem in cost minimization. This is achieved by the formulation of a cost function that evaluates the quality of edge configurations. The function is a linear sum of weighted cost factors. The cost factors capture desirable characteristics of edges such as accuracy in localization, thinness, and continuity. Edges are detected by finding the edge configurations that minimize the cost function. We give a mathematical description of edges and analyze the cost function in terms of the characteristics of the edges in minimum cost configurations. Through the analysis, we provide guidelines on the choice of weights to achieve certain characteristics of the detected edges. The cost function is minimized by the simulated annealing method. We present a novel set of strategies for generating candidate states and devise a suitable temperature schedule. Experimental results, which verify the usefulness of our cost minimization approach to edge detection, are given.**

***Index Terms*—Cost minimization, edge detection, edge modeling, low-level vision, segmentation, simulated annealing.**

I. Introduction

The DETECTION of edges in an image is an important task in image processing. A great deal of literature exists on the subject (see [1]–[3] for an overview). The more recent detection algorithms can be generally classified into one of three categories: optimal filtering [4]–[6], surface fitting [7]–[10], and sequential contour tracing techniques [11]–[15]. Despite the tremendous amount of research that has been done, the task of finding the edges in an image that correspond to true physical boundaries remains a difficult problem. Part of the difficulty lies in finding a suitable definition of an edge that is generally applicable. A narrow concept of an edge ultimately restricts the applicability of the detection algorithm. For example, a detection algorithm that assumes that edges are ideal steps will be ineffective in finding roof edges or texture edges [16].

Another difficulty with many detection algorithms is that the decision on the presence of an edge pixel is made without considering the local edge structure in the neighborhood of the pixel. This is particularly true of nonsequential detection algorithms. It should be possible to exploit information from neighboring edges in the decision process. For instance, noise in an image causes many detection algorithms to produce fragmented edges; an algorithm that exploits local edge continuity information should be able to reduce the amount of fragmentation by linking together locally disconnected edges.

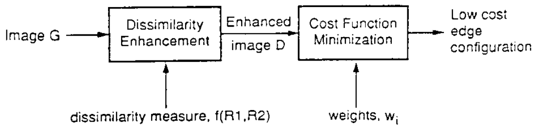


Image G
Dissimilarity Enhancement
Enhanced image D
Cost Function Minimization
Low cost edge configuration
dissimilarity measure, f(R1,R2)
weights, w_i

Fig. 1. Block diagram of the cost minimization approach to edge detection.

In this paper, we cast edge detection as a problem in cost minimization. Although most other detection techniques can also be viewed as some form of cost minimization, our approach is unique in the way the cost function is defined: The function not only uses information from image data but also exploits information from local edge structure. By this, we mean that the function takes into account not only the pointwise presence of edge pixels in an image but the local shape and continuity of the edge as well. Two major difficulties arise in such an approach. The first lies in defining a suitable cost function. The second involves the minimization of such a function. Note that the search space for the minimum cost solution is extremely large as the number of possible solutions is equal to 2^P, where P is the number of pixels in the image.

An important advantage of the cost minimization approach to edge detection presented in this paper is that it assumes no preconceived concept of an edge in terms of the dissimilarity between the regions on either side of the edge. Hence, the approach is flexible in potentially being able to detect various types of edges.

Fig. 1 shows a block diagram of our cost minimization approach to edge detection. The first step in the detection process is dissimilarity enhancement. Here, the points in an image that serve as good candidates for edge points are enhanced. The enhancement is based on a user-defined dissimilarity measure. The cost function is defined in terms of the enhanced image. It is a weighted sum of five cost factors. The edges are detected by finding a suitably low cost solution to the cost function. Simulated annealing is employed as a technique for finding low-cost solutions.

There has been a lot of work on applying simulated annealing (or stochastic relaxation) to image restoration and segmentation [17]–[21]. In this earlier work, the cost function is *model based*. A random field model is specified for the image intensities and/or boundary processes. This prior random field is a Markov random field characterized by a

Manuscript received January 10, 1989; revised May 21, 1991. Recommended for acceptance by R. J. Woodham.

H. L. Tan is with the Image Electronics Center, Kodak Research Laboratories, Rochester, NY.

S. B. Gelfand and E. J. Delp are with the Computer Vision and Image Processing Laboratory, School of Electrical Engineering, Purdue University, West Lafayette, IN 47907.

IEEE Log Number 9104526.

Gibbs distribution. A measurement model is specified for the observed image, and the resulting posterior random field is also a Markov random field characterized by a Gibbs distribution. The cost function is the posterior Gibbs energy, and minimizing the cost corresponds to finding a maximum a posteriori (MAP) estimate of the image. In all of these approaches, the relationship between the random field parameters and the image characteristics is not clear, and the parameters are either assumed known or must be estimated from the data. Furthermore, it is not possible to analyze the dependence of the optimal temperature schedule [22] on the parameters.

In contrast with the above work, our cost function is *heuristic*, and is an attempt to incorporate criteria related to both the image data and local edge structure in a reasonable manner. We are only concerned with detecting edges. In our approach, the relationship between the weight parameters and the detected edge characteristics can be analyzed, and parameters can be selected based on desired characteristics. In addition, we can analyze (to some extent) the dependence of the optimal temperature schedule [22] on the parameters.

We also point out that our approach to edge detection is quite different from relaxation labeling [23], [24]. Relaxation labeling optimizes a linear function that has the form of an "average local support" function. Such a cost function as applied to edge detection depends only on the average consistancy of the local edge structure. Our cost function depends directly on the image data as well as the local edge structure and cannot be optimized by a relaxation labeling process.

The paper is organized as follows. In Section II, we describe our concept of an edge. In Section III, we define a cost function for evaluating edges. In Section IV, we analyze the characteristics of minimum cost edge configurations. In Section V, the simulated annealing method is adapted and applied to the problem of finding (near) minimum cost edge configurations. In Section VI, we present some experimental results.

II. Description of an Edge

A precise notion of an edge is crucial to the formulation of a cost function for evaluating edges. However, it is a difficult task to explicitly define what constitutes an edge in an image. The perception of edges by the human visual system is an extremely complex process that is strongly influenced by prior knowledge. There are a number of visual paradoxes in which an edge is clearly perceived when none physically exists (c.f. p. 51 of [25]). No absolute definition of an edge exists, and the performance of edge detection algorithms are only as good as their inherent assumption as to what edges are.

For our purpose, we will define an edge in a general sense to include a wide variety of edge types. However, we will restrict our attention to those edges that are evident from the image data itself and not from higher level human cognitive processes. With this in mind, we define an edge to be a boundary in an image that separates two regions that have significantly dissimilar characteristics. The cause of the dissimilarity may be due to a combination of several factors, such as the geometry of the object, surface reflectance characteristics, viewpoint, and illumination. The term "dissimilarity" is used in a broad sense to include any characteristic variation in the intensity values in the image. This definition is broad enough to include both intensity and texture edges. In addition to the fundamental property that edges separate dissimilar regions, edges should also possess certain desirable topological properties. In particular, edges should accurately partition dissimilar regions and should be thin, continuous, and of sufficient length.

We next make this intuitive concept of an edge precise by formulating it in mathematical terms. We begin with the following definitions. An *image* G is a 2-D array of pixels

$$G = \{g(i,j); 1 \leq i,j \leq N\}$$

where each pixel $g(i,j)$ is a gray level in the range $0 \leq g(i,j) \leq 255$ (for simplicity, we assume square images and an 8-b gray scale). An *edge configuration* S is also a 2-D array of pixels

$$S = \{s(i,j); 1 \leq i,j \leq N\}$$

where each pixel $s(i,j)$ is a binary value 0 or 1. If $s(i,j) = 1$, the pixel $s(i,j)$ is called an *edge pixel*; otherwise, it is a *nonedge pixel.* Let $\mathbf{S}$ denote the set of all possible edge configurations.

Let $V_{ij}(S)$ denote the set of eight pixels in $S \in \mathbf{S}$, which consists of the neighbors of $s(i,j)$. Let $W_{ij}(S)$ denote the set of nine pixels in $S \in \mathbf{S}$, which consists of the neighbors of $s(i,j)$ and $s(i,j)$ itself. In addition, let L denote the $N \times N$ lattice of pixel sites, and let L_{ij} denote the 3×3 lattice of pixel sites centered at site (i,j). When the specific indices $(i,j) \in L$ of a pixel site are not required, we denote the site by $l \in L$.

A *walk* is a nonnull sequence of edge pixels $e_1, \ldots, e_K$ such that e_k is a neighbor of e_{k+1} for $1 \leq k \leq K-1$. A *path* is walk such that every edge pixel is distinct. A *cycle* is a walk such that the origin and terminus are the same, the origin and internal pixels are distinct, and there is at least one internal pixel. A collection of edge pixels A is *connected* if for any distinct $e, e' \in A$ there is a path from e to e' in A. Let B be a collection of edge pixels (not necessarily connected). Then, there is a partition of B into nonempty connected subsets $B_1, \ldots, B_K$ such that $B_j \cup B_k$ is not connected for $j \neq k$. $B_1, \ldots, B_K$ are the maximally connected subsets or *components* of B. For any set of pixels C, we denote by $\|C\|$ the number of distinct edge pixels in C.

We will now specify what is meant by an edge, a segment of an edge, and thick and thin edges.

Definition 1: An *edge* E is a component of the set of edge pixels in an edge configuration S.

Definition 2: A *segment* of an edge E is a subset of E that is connected.

It will sometimes be convenient to refer to an edge or edge segment as a path or cycle, by which we mean that the elements of the edge or edge segment can be arranged in a sequence that is a path or a cycle.

According to our concept of an edge, the edges in an image should be thin. Consider an edge E that contains a path that connects two edge pixels. Suppose there is a collection of adjacent paths in E that connect the same two pixels. An

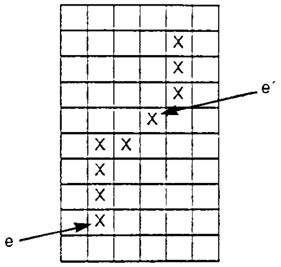


Fig. 2. Edge that contains a cycle of length 3 (note that there are two adjacent paths connecting edge pixels e and e').

examination of edge structures reveals that E will contain at least one cycle of length 3. An example of how such cycles cause adjacent paths is shown in Fig. 2.

Definition 3: An edge pixel that is not contained in any cycle of length 3 is called a *thin edge pixel*; otherwise, it is called a *thick edge pixel.* An edge that contains only thin edge pixels is called a *thin edge*; otherwise, it is called a *thick edge.*

III. A Cost Function for Evaluating Edges

In this section, we turn our attention to the formulation of a cost function for evaluating edges. The function assigns a cost to each edge configuration such that the configuration with the lowest cost corresponds to the best configuration according to our concept of an edge.

In practice, there is often a tradeoff between the different desirable characteristics of an edge. For instance, requiring every edge in an image to be long and continuous may result in poor localization and the appearance of false boundaries. Hence, it is appropriate to associate a measure of importance with each desirable edge characteristic so that situations involving conflicting edge requirements may be resolved. This is achieved here by a cost function that takes the form of a linear combination of weighted cost factors, where each cost factor captures a desirable characteristic of edges. The cost factors are curvature (C_c), dissimilarity (C_d), number of edge points (C_e), fragmentation (C_f), and thickness (C_t). We will first describe the general form of the cost function, and then describe each of the cost factors.

Definition 4: The *point cost* of edge configuration $S \in \mathbf{S}$ at pixel site $l \in L$ is defined as the following linear sum of weighted *point cost factors* $C_i(S, l)$:

$$F(S,l) = \sum_i w_i C_i(S,l) \tag{1}$$

where $w_i \geq 0, 0 \leq C_i \leq 1$, and $i \in \{c, d, e, f, t\}$.

Definition 5: The *total cost* of edge configuration $S \in \mathbf{S}$ is the sum of the point costs at every pixel site in the image:

$$F(S) = \sum_{l \in L} F(S,l). \tag{2}$$

This total cost is the cost function used for evaluating edge quality. This cost function is to be minimized.

Definition 6: For any pair of edge configurations $S, S' \in \mathbf{S}$, the *incremental cost* $\Delta F(S, S')$ from S to S' is given by

$$\Delta F(S,S') = F(S') - F(S). \tag{3}$$

Alternatively, we can write the total cost as a sum of five *total cost factors* $C_i(S)$:

$$F(S) = \sum_i w_i C_i(S) \tag{4}$$

where

$$C_i(S) = \sum_{l \in L} C_i(S,l)$$

and we can write the incremental cost as a sum of five *incremental cost factors* $\Delta C_i(S, S')$:

$$\Delta F(S,S') = \sum_i w_i \Delta C_i(S,S') \tag{5}$$

where

$$\Delta C_i(S,S') = \sum_{l \in L} [C_i(S',l) - C_i(S,l)].$$

A. Determining Region Dissimilarity

We have mentioned in Section II that the fundamental property of an edge is that it separates regions that are dissimilar. In dissimilarity enhancement, we assign large values to those points in an image that possess this fundamental property. We also attempt to ensure that these enhanced points are accurately localized.

The enhanced image

$$D = \{d(i,j); 1 \leq i,j \leq N\}$$

is a collection of pixels where each pixel value is proportional to the degree of region dissimilarity that exists at that pixel site. The pixel values lie in the range $0 \leq d(i,j) \leq 1$. Pixels with large values close to 1 are good candidates for edge points. Two things are required in enhancement: well defined regions of interest on either side of an edge and a function that measures dissimilarity between the regions of interest.

The regions of interest are defined with reference to a set of selected edge structures. We call this set of edge structures the *basis set.* Within the scope of this paper, the basis set is constrained to be 3-pixel thin edge structures contained in a 3×3 window region. In most of our applications, we selected 12 edge structures for the basis set. For each edge structure, we define a pair of regions on either side of the edge. These regions, which shall be labeled $R1$ and $R2$ for each edge structure, are the regions of interest on which a dissimilarity measure will be applied. The 12 edge structures of the basis set, the (circled) edge pixel they are centered around, and their associated regions are shown in Fig. 3. Depending on the application and the specific measure of dissimilarity used, larger (or smaller) regions for $R1$ and $R2$ could be defined. We remark that it may be useful to select the regions $R1$ and $R2$ adaptively for the detection of varying resolution and multiscale edges. This is an area for further research (see [33]).

The function that measures the dissimilarity between regions $R1$ and $R2$ is denoted by $f(R1, R2)$. This measure could be a simple difference of gray level averages in $R1$ and $R2$, or it could be more complicated measures based on statistical or structural properties in the gray levels. There is great flexibility

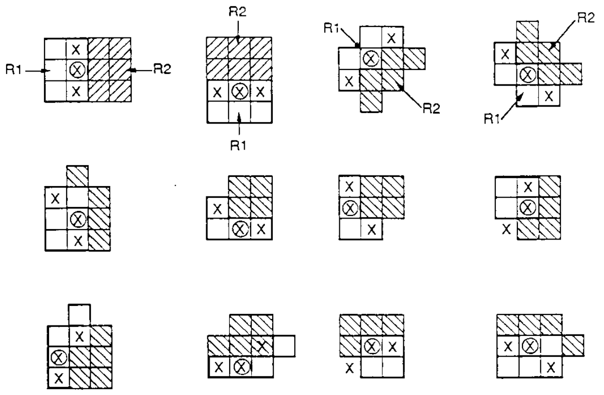


Fig. 3. Twelve edge structures of the basis set, the (circled) edge pixel they are centered around, and their associated regions of interest on each side of the edge.

in such an approach to dissimilarity enhancement as we do not *a priori* restrict the nature of the dissimilarity. This is in contrast to many detection algorithms that assume some specific notion of edges and are devoted to finding only such edges. Several examples of $f(R1, R2)$ are discussed in Section VI.

We now describe a procedure to obtain an enhanced image D from the original image G. The procedure is as follows:

1) Initially, all the pixels $d(l)$ are set equal to zero.
2) At each pixel site l, we perform steps a) and b).
   a) Each of the edge structures of the basis set is fitted onto the site by centering it on the location l in G. The corresponding paired regions $R1$ and $R2$ in G are determined for each structure, and the value of $f(R1, R2)$ is computed. The structure that results in the maximum value of $f(R1, R2)$ is chosen as the best fitted edge structure. Note that each edge structure of the basis set contains exactly three edge pixels; we will denote the sites of the three edge pixels of the best fitted edge structure in G as l, l_1, and l_2.
   b) Next, we perform nonmaximal suppression by shifting the location of the chosen best fitted edge in a direction determined by the edge structure. For vertical, horizontal, and diagonal edge structures, the shifting is performed by moving the edge location by one pixel in each of the directions perpendicular to the edge. For all other edge structures, the shifting is done by moving the edge location one pixel in each of the the four directions: up, down, left, and right. For each shifted edge, we determine the new regions for $R1$ and $R2$ and compute the corresponding value of $f(R1, R2)$. One of the following two cases results:
      i) If no larger value of $f(R1, R2)$ results from shifting the best fitted edge structure, we set

$$\delta = \frac{f(R1, R2)}{3}$$

where $f(R1, R2)$ is determined using the best fitted edge structure. We then increment the value of each of the pixels $d(l)$, $d(l_1)$, and $d(l_2)$ by δ.
      ii) If there is a larger value of $f(R1, R2)$ from one of the shifted edge structures, we do not alter any pixel value.
3) Finally, the values of the pixels $d(l)$ at all sites are truncated to a maximum of 1.

Step 3) is performed to ensure that the dissimilarity values lie in the assumed range $0 \leq d(l) \leq 1$.

B. Defining the Cost Factors

The general form of the point cost function is given in (1). It is a linear combination of five weighted point cost factors, which we now define. In the following discussion, an *endpoint* is an edge pixel that has at most one neighboring edge pixel, a *nonisolated endpoint* has exactly one neighboring edge pixel, and an *isolated endpoint* has no neighboring edge pixels.

Cost for Curvature: The cost for curvature assigns a cost to nonendpoint edge pixels based on a local measure of curvature. This cost factor tends to smooth out or remove curvy edges. To define the cost for curvature, consider an edge pixel $s(l)$ that is not an endpoint. Then, $s(l)$ is the connection point (or common point) of at least one pair of straight edge segments. The direction of each such straight edge segment is uniquely specified by a vector pointing from site l to the site of any other pixel of the segment. Let n be the maximum number of distinct pairs of straight edge segments connected by $s(l)$. Let $\phi_i(l)$ be the larger of the two angles between the ith pair of edge segments connected by $s(l)$, and let $\theta_i(l) = \phi_i(l) - 180$. We define the *curvature* $\theta(l)$ by

$$\begin{aligned}\theta(l) &= \max_{1 \leq i \leq n} \theta_i(l), \quad \text{if } s(l) \text{ is a nonendpoint edge pixel}\\ &= 0, \quad \text{else.}\end{aligned}$$

Assuming that the image lattice is uniformly spaced, the curvature at any site can take on one of four possible values, namely 0, 45, 90, or 135°; in the example in Fig. 4, the curvature is 135°. Finally, the curvature point factor cost $C_c(S, l)$ is given by

$$C_c(S, l) = \begin{cases} 0, & \text{if } \theta(l) = 0 \\ 0.5, & \text{if } \theta(l) = 45 \\ 1.0, & \text{if } \theta(l) \geq 90. \end{cases}$$

Cost for Region Dissimilarity: This cost for region dissimilarity assigns a cost to nonedge pixels that is proportional to the degree of dissimilarity. This cost factor tends to place edge pixels at points of high region dissimilarity. The region dissimilarity point cost factor $C_d(S, l)$ is given by

$$C_d(S, l) = \begin{cases} 0, & \text{if } s(l) \text{ is an edge pixel,} \\ d(l), & \text{if } s(l) \text{ is not an edge pixel.} \end{cases}$$

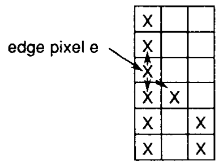


Fig. 4. Edge pixel that is the connection point of three pairs of straight edge segments. The curvature at the denoted edge pixel site is 135°.

Cost for Number of Edge Points: The cost for number of edge points assigns a (unit) cost to each edge pixel. This cost factor tends to avoid an excessive number of edge pixels being detected. The edgepoint point cost factor $C_e(S, l)$ is given by

$$C_e(S, l) = \begin{cases} 1, & \text{if } s(l) \text{ is an edge pixel,} \\ 0, & \text{if } s(l) \text{ is not an edge pixel.} \end{cases}$$

Cost for Fragmentation: The cost for fragmentation assigns a cost to endpoint edge pixels. This cost factor tends to (locally) link up or remove fragmented edges. To define the cost for fragmentation, consider an edge pixel $s(l)$ that is an endpoint. If we view an isolated endpoint as a path connecting two nonisolated endpoints that has shrunk to a single point, then it seems reasonable to take the cost of an isolated endpoint to be twice that of a nonisolated endpoint. The fragmentation point cost factor $C_f(S, l)$ is given by

$$C_f(S, l) = \begin{cases} 1.0, & \text{if } s(l) \text{ is an isolated endpoint} \\ & \text{edge pixel ,} \\ 0.5, & \text{if } s(l) \text{ is a nonisolated endpoint} \\ & \text{edge pixel ,} \\ 0, & \text{else.} \end{cases}$$

Cost for Edge Thickness: The cost for edge thickness assigns a (unit) cost to thick edge pixels. The edge thickness point cost factor $C_t(S, l)$ is given by

$$C_t(S, l) = \begin{cases} 1, & \text{if } s(l) \text{ is a thick edge pixel,} \\ 0, & \text{if } s(l) \text{ is not a thick edge pixel.} \end{cases}$$

The interaction between the various cost factors is highly complex. Nonetheless, it is possible to derive some interesting relationships between the different cost factors that guarantee that detected edges have certain desirable properties. These relationships can be used to establish a principled approach to selecting the weight parameters (see Sections IV and VI).

Computing the Cost

It will be seen in the cost minimization procedure that the point cost and the incremental cost will be used repeatedly in the search for low-cost configurations. Hence, from a computational standpoint, it is of major importance that they can be computed in an efficient way.

Computing the Point Cost: A great deal of reduction in computation time can be achieved by pooling together information affecting each of the different factors and organizing it in a form that will allow for efficient computation. This is achieved by the decision tree structure as shown in Fig. 5.

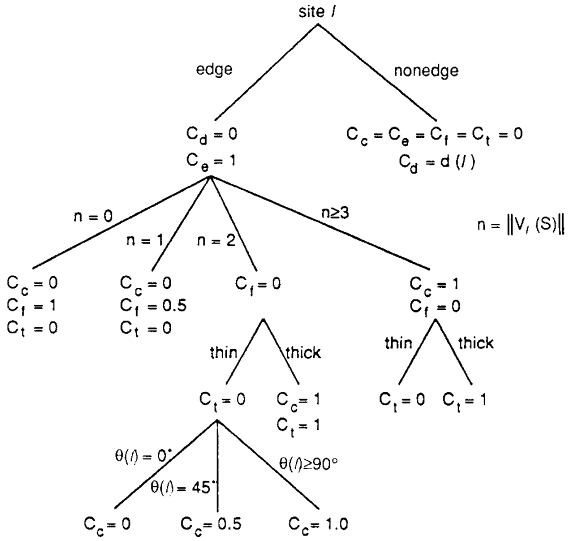


Fig. 5. Computation of point cost $F(S, l)$ using a decision tree.

Computing the Incremental Cost: From (2) and (3), the incremental cost from S to S' can be written as the summation over the lattice L of the difference in point cost $F(S', l) - F(S, l)$. Since there are N^2 sites in L, this represents a total of $2N^2$ times that the point cost has to be computed. The following proposition asserts that by appropriately restricting the choice of S and S', the incremental cost can be reduced to a summation of the difference in point cost over a small subset of L.

Proposition 1: If $S, S' \in \mathbf{S}$ are identical edge configurations except at site $l_0 \in L$, then

$$\begin{aligned} \Delta F(S, S') &= \sum_{l \in L} [F(S', l) - F(S, l)] \\ &= \sum_{l \in L_{l_0}} [F(S', l) - F(S, l)]. \end{aligned} \qquad (6)$$

The proof of Proposition 1 is omitted. Note that an equivalent expression for (6) using incremental cost factors is

$$\Delta F(S, S') = \sum_i w_i \Delta C_i(S, S') \qquad (7)$$

where

$$\begin{aligned} \Delta C_i(S, S') &= \sum_{l \in L} [C_i(S', l) - C_i(S, l)] \\ &= \sum_{l \in L_{l_0}} [C_i(S', l) - C_i(S, l)]. \end{aligned}$$

In addition, note that $C_i(S, l)$ depends only on the pixels in a 3×3 window around site l, and hence, $\Delta C_i(S, S')$ depends only on the pixels in a 5×5 window around site l (we shall use this fact repeatedly in the sequel). Finally, note that Proposition 1 can be used to sequentially compute the incremental cost between two edge configurations that differ at more than a single site, i.e., if edge configurations S, S' now differ at K sites, then there exists a sequence of edge configurations

$S = S_1, S_2, \ldots, S_K = S'$, where S_k and S_{k+1} differ at only a single site, and

$$\Delta F(S, S') = \sum_{k=1}^{K-1} \Delta F(S_k, S_{k+1}).$$

This indirect method of computing the incremental cost is particularly efficient for values of K much less than N^2.

IV. Characteristics of Minimum Cost Configurations

In this section, we analyze certain characteristics of edges in minimum cost configurations. These characteristics include edge thickness, length, and continuity.

The characteristics of the edges is related to the set of weights chosen for the cost factors. Hence, a judicious choice of weights is essential for ensuring good performance. Our choice of weights will be based on several propositions to be presented in this section. The application of these results to the selection of weights in several examples is discussed in Section VI. In the sequel, we will use the terms "state" and "edge configuration" interchangeably.

A. Thin Edges

An important aspect of our edge concept is that edges should be thin. The following proposition gives a sufficient condition for thin edges in a minimum cost state.

Proposition 2: If $w_t > 2w_f - w_c + w_d - w_e$, then there are no thick edges in any minimum cost state.

Proof: See Appendix A.

B. Dissimilarity Threshold

An edge pixel will be detected if and only if the dissimilarity value at the corresponding site is sufficiently large. The following proposition gives lower bounds on the dissimilarity value in a minimum cost state.

Proposition 3: Let S_G be a minimum cost state with no thick edges.

a) S_G has an edge pixel at site l if

$$d(l) > \frac{\max\{7w_c + 7w_t, w_f\} + w_e}{w_d}.$$

b) S_G has an edge pixel at site l only if

$$d(l) \geq \frac{\min\{w_c - 2w_f, -w_f\} + w_e}{w_d}.$$

c) S_G has an endpoint edge pixel at site l only if

$$d(l) \geq \frac{w_e}{w_d}.$$

Proof: See Appendix B.

We remark that if $w_c = w_f = w_t = 0$, then combining parts a) and b) of Proposition 3 shows that S_G has an edge pixel at site l if and only if $d(l) \geq w_e/w_d$ (this is true without the assumption that S_G has no thick edges). Under these conditions, minimization of the cost function corresponds to a simple thresholding approach to the pointwise detection of edge pixels.

Proposition 3 gives information on how to choose the weights to be commensurate with a dissimilarity measure $f(R1, R2)$. To see this, recall that $d(l)$ is incremented by $f(R1, R2)/3$ for each locally best fitted edge structure that includes an edge pixel at site l (see Section III-A). If $d(l)$ is incremented three times and $f(R1, R2)$ is the same for each case, then according to Proposition 3c), it is possible to have a detected endpoint edge pixel at site l only if

$$f(R1, R2) \geq \frac{w_e}{w_d}.$$

We define

$$\xi \stackrel{\triangle}{=} \frac{w_e}{w_d}$$

to be the *dissimilarity threshold*.

C. Minimum Edge Length and Gap Length

Two other important aspects of our concept of an edge is that edges should be sufficiently long and continuous. By requiring the length of an edge and the gap between edges to be large, short locally fragmented edges will be connected up into long continuous edges. The following propositions give lower bounds on the edge length and gap length in a minimum cost state.

Proposition 4: Let S_G be a minimum cost state with no thick edges. If an edge E in S_G contains at least two endpoints, then (assuming $w_d > w_e$)

$$\|E\| \geq \frac{w_f}{w_d - w_e}.$$

Proof: See Appendix C.

Proposition 5: Let S_G be a minimum cost state with no thick edges. If a straight path E can be constructed between two edge endpoints in S_G such that the resulting state also has no thick edges, then

$$\|E\| \geq \frac{w_f - 2w_c}{w_e}.$$

Proof: See Appendix C.

In view of Propositions 4 and 5, we define

$$\eta_e \stackrel{\triangle}{=} \frac{w_f}{w_d - w_e}$$

to be the *minimum edge length*, and

$$\eta_g \stackrel{\triangle}{=} \frac{w_f - 2w_c}{w_e}$$

to be the *minimum gap length.*

The propositions and definitions discussed in this section are used in Section VI to establish a principled approach to selecting the weights for the cost factors.

V. Minimization Using Simulated Annealing

In the previous sections, we have presented a cost function that measures how well an edge configuration fits a given image. The task is to find a configuration that is a (near) global minimum of such a cost function that may have large numbers of local minima. Since there are 2^{N^2} possible edge configurations, it is not possible to implement an exhaustive search.

We will use simulated annealing [26] to perform the minimization. Simulated annealing is a stochastic optimization algorithm derived from Monte Carlo methods in statistical mechanics. The annealing algorithm has been successfully applied to a variety of hard optimization problems in operations research [27], source coding [28], and image processing [29]. Practical implementations of the annealing algorithm will usually yield local minima that are not global minima. However, in the context of edge detection, we find that it is not necessary to obtain a minimum cost edge configuration; very satisfactory results are obtained from relatively low-cost edge configurations.

We are given a cost function F defined over a finite set $\mathbf{S}$. Let $\{T_k\}$ be a sequence of positive numbers. In the context of the annealing algorithm, F is the *energy function*, and $\{T_k\}$ is the *temperature schedule.* For every $S \in \mathbf{S}$, let $Q(S) \geq 0$ and $\Sigma_{S \in \mathbf{S}} Q(S) = 1$, and for every $S, S' \in \mathbf{S}$, let $R(S, S') \geq 0$ and $\Sigma_{S' \in \mathbf{S}} R(S, S') = 1$. The general form of the annealing algorithm is as follows. Let $[x]_+ = \max\{x, 0\}$.

Simulated Annealing Algorithm:

1) Set $k = 0$.
2) Generate an initial state X_0 with probability $Pr(X_0 = S) = Q(S)$.
3) Generate a candidate state Y_k with conditional probability $\Pr(Y_k = S' \mid X_k = S) = R(S, S')$.
4) Let $X_{k+1} = Y_k$ with probability $\exp\left(\frac{-[F(Y_k) - F(X_k)]_+}{T_k}\right)$ and $X_{k+1} = X_k$ otherwise.
5) Set $k = k + 1$, and go to Step 3).

The annealing algorithm generates a discrete time Markov chain $\{X_k\}$. To implement the algorithm, three things are required. First, we need to specify the probability distribution $Q(\cdot)$ for generating the initial state. Second, we need to specify the transition distribution $R(\cdot, \cdot)$ for generating candidate states. Finally, we need a temperature schedule $\{T_k\}$. In our work, we chose $Q(\cdot)$ to be a uniform distribution over $\mathbf{S}$. The choice of $R(\cdot, \cdot)$ and $\{T_k\}$ is discussed below. A number of researchers [17], [22], [30]–[32] have established the asymptotic convergence of the above algorithm and estimated rates of convergence subject to certain conditions on the transition distribution and the temperature schedule. In devising a transition distribution and a temperature schedule, we will focus our attention on a theorem due to Hajek [22].

Before stating Hajek's theorem, some preliminary definitions are in order. Let the set of states in $\mathbf{S}$ at which F attains its minimum value be denoted by $\mathbf{S}^*$, i.e., $\mathbf{S}^*$ is the set of global minima for F.

A state S' is *reachable* from state S if there is a sequence of states $S = S_0, S_1, \ldots, S_K = S'$ such that $R(S_k, S_{k+1}) > 0$ for $0 \leq k < K$. $(\mathbf{S}, R)$ is *irreducible* when for any pair of states S and S', S' is reachable from S.

A state S' is *reachable at height* U from state S if there is a sequence of states $S = S_0, S_1, \ldots, S_K = S'$ such that $R(S_k, S_{k+1}) > 0$ for $0 \leq k < K$ and $F(S_k) \leq U$ for $0 \leq k \leq K$. $(\mathbf{S}, R, F)$ is *weakly reversible* when for any real number U and any pair of states S and S', S' is reachable at height U from S if and only if S is reachable at height U from S'.

A state S is said to be a *local minimum* if no state S' with $F(S') < F(S)$ is reachable from S at height $F(S)$. Suppose S is a local minimum. The *depth* of S is $+\infty$ if S is a global minimum; otherwise, the *depth* of S is the smallest number U, $U > 0$, such that some state S' with $F(S') < F(S)$ can be reached from S at height $F(S) + U$.

Let the temperature schedule $\{T_k\}$ be a sequence of strictly positive numbers such that $T_1 \geq T_2 \geq \ldots$ and $\lim_{k \to \infty} T_k = 0$. Suppose that a discrete time nonstationary Markov chain $\{X_k\}$ with state space $\mathbf{S}$ is generated using the simulated annealing algorithm described above. The convergence in probability of the chain to the set of globally minimum cost states is characterized by the following theorem.

Theorem 1 [22]: Assume that the simulated annealing based on $(\mathbf{S}, R, F)$ is irreducible and weakly reversible. Then

$$\lim_{k \to \infty} \Pr[X_k \in \mathbf{S}^*] = 1 \tag{8}$$

if and only if

$$\sum_{k=1}^{\infty} \exp\left(\frac{-d^*}{T_k}\right) = +\infty \tag{9}$$

where d^* is the maximum of the depths of all states that are local but not global minima.

A. Method of Generating Candidate States

Our method of generating candidate states is based on a combination of five possible strategies that are described in the following paragraphs. In each strategy, we assume that the site $l \in L$ is given; the method of selecting l will be discussed later. Let $X_k = S$ be the current state and $Y_k = S'$ be the candidate state.

Strategy 1: Single pixel change: $S' = M_1(S, l)$.

Here, we generate the candidate state S' by complementing the edge labeling at site l of the present state S.

Strategy 2: Double pixel change: $S' = M_2(S, l)$.

Here, we randomly select a neighboring site $l' \in L_l$ and generate the candidate state S' by complementing the edge labelings at sites l and l' of the present state S.

Strategy 3: Single pixel shift: $S' = M_3(S, l)$.

This strategy of generating a candidate state is based on locally perturbing the edge labelings in the window $W_l(S)$. The candidate state S' is identical to the present state S except for the pixels in $W_l(S)$. The pixels in $W_l(S')$ are labeled according to the transformation of the edge structure in $W_l(S)$ shown in Fig. 6. If the edge structure in $W_l(S)$ is one of the 14 edge structures shown in the figure, the edge structure in $W_l(S')$ is the corresponding structure shown on the right. Where there are two structures possible for the transformation on the right, one of them is selected on an equally likely basis. If the edge structure in $W_l(S)$ does not correspond to one of

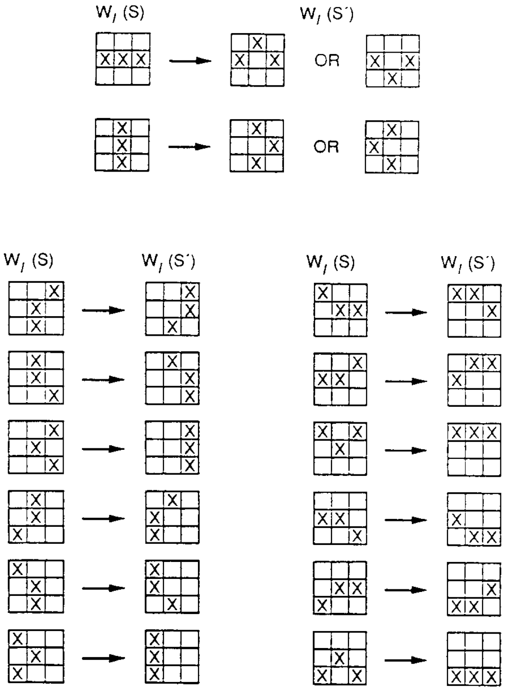


Fig. 6. Fourteen edge structures in $W_l(S)$ and their corresponding transformations to $W_l(S')$ using strategy M_3.

the structures shown in the figure, then the structure in $W_l(S')$ is made identical to that in $W_l(S)$.

Strategy 4: Multiple pixel shift: $S' = M_4(S, l)$.

This strategy of generating a candidate state is very similar to the strategy of M_3 except that the edge perturbation is more significant. Again, the candidate state S' is identical to the present state S except for the pixels in $W_l(S)$. The pixels in $W_l(S')$ are labeled according to the transformation of the edge structure in $W_l(S)$ shown in Fig. 7.

Strategy 5: Window region change: $S' = M_5(S, l)$.

This strategy of generating a candidate state is based on randomly changing edge labelings in the window $W_l(S)$. The candidate state S' is identical to the present state S except for the pixels in $W_l(S)$. The pixels in $W_l(S')$ are labeled at random.

The overall approach to generating a candidate state is as follows. First, a site l is selected at random, and then a strategy M is selected (independent of l) with $M = M_i$ with probability p_i (the specific values of p_i are application dependent). Note that this procedure (implicitly) defines the transition distribution $R(\cdot, \cdot)$. In addition, it is easy to see that the resulting simulated annealing $(\mathbf{S}, R, F)$ is irreducible and weakly reversible.

An alternative to selecting the next site l at random is to select l in a deterministic order. The simplest approach is to select l in raster scan fashion. Experimentally, we have found that the results obtained by selecting l in a raster scan fashion and at random are very similar.

Now, recall that when a pixel is relabeled at site l, the resulting incremental cost depends only on the pixels in a 5×5 window around site l. Consequently, if there are at least four pixels between two sites l_1 and l_2, then the incremental costs corresponding to altering the edge labelings in 3×3 windows around sites l_1 and l_2 are independent. Furthermore, all of the strategies discussed above at most alter the edge labelings in a 3×3 window around the selected site. This suggests a parallel implementation of the simulated annealing algorithm that simultaneously alters the local edge structures at $N^2/25$ sites in an $N \times N$ image, as shown in Fig. 8. Such a parallel implementation is equivalent to a sequential implementation that selects the next site l in a staggered raster scan fashion. Experimentally, we have also found that the results obtained by selecting l in staggered and ordinary raster scan fashions (and also at random) are all very similar.

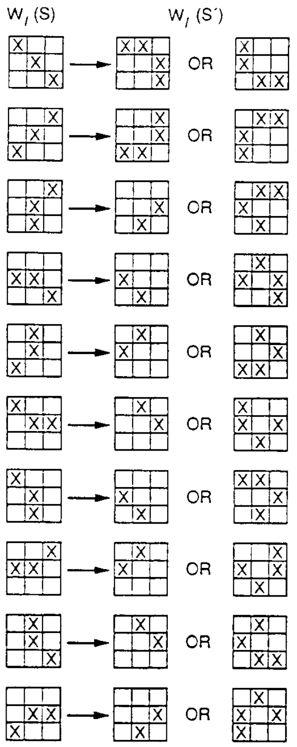


Fig. 7. Ten edge structures in $W_l(S)$ and their corresponding transformations to $W_l(S')$ using strategy M_4.

B. Temperature Schedule

We now focus on the use of Theorem 1 to devise a suitable temperature schedule. Consider a schedule of the form

$$T_k = \frac{c}{\log(k+1)}. \tag{10}$$

Equation (9) and, hence, (8) holds if and only if $c \geq d^*$. For practical purposes, the parameter c in the equation has to be kept as small as possible so that the number of iterations can be held within a reasonable limit. Because of the complex nature of the interaction between the different cost factors and because of the large number of possible edge structures that have to be taken into consideration, we are unable to compute

1	2	3	4	5	1	2	3	4	5
6	7	8	9	10	6	7	8	9	10
11	12	13	14	15	11	12	13	14	15
16	17	18	19	20	16	17	18	19	20
21	22	23	24	25	21	22	23	24	25
1	2	3	4	5	1	2	3	4	5
6	7	8	9	10	6	7	8	9	10
11	12	13	14	15	11	12	13	14	15
16	17	18	19	20	16	17	18	19	20
21	22	23	24	25	21	22	23	24	25

Fig. 8. Parallel implementation simulatenously updates all of the same numbered sites in the image.

d^*. Instead, we will obtain an estimate of d^*, which we denote by d_0.

Our estimate d_0 of d^* will be based in part on the following two propositions:

Proposition 6: Let E be a thin edge that is a path in a global minimum state S_G. Let S be a state that has no edge pixels at the sites of E or its neighboring pixels, and let S' be the state that is identical to S except at the sites of E. Then, there exists a sequence of states $S = S_0, S_1, \ldots, S_K = S'$ that successively add single edge pixels along the sites of E, such that

$$\Delta F(S, S_k) \leq w_f, \ \ k = 0, \ldots, K-1$$

and $\Delta F(S, S') \leq 0$.

Proof: See Appendix D.

Proposition 7: Let E be a thin edge that is a path in a state S. Let S_G be a global minimum state that has no edge pixels at the sites of E or its neighboring pixels, and let S' be the state that is identical to S except at the sites of E. Then, there exists a sequence of states $S = S_0, S_1, \ldots, S_K = S'$ that successively delete single edge pixels along the sites of E, such that

$$\Delta F(S, S_k) \leq w_f, \ \ k = 0, \ldots, K-1$$

and $\Delta F(S, S') \leq 0$.

Proof: See Appendix D.

We remark that there are some useful extensions of Propositions 6 and 7 that can be used to justify the discussion of estimating d^* to follow in more rigorous detail. For example, in Proposition 6, S_G could contain an edge segment E, and S could contain a subsegment E' of E, which gets extended to E in S', whereas in Proposition 7, S could contain an edge segment E, and S_G could contain a supersegment E' of E, which gets contracted to E in S'. Under suitable conditions, we can obtain similar bounds as above, but we do not go into detail here (see [33]).

Our approach to estimating d^* is to analyze successively more complicated edge structures. In the sequel, we will discuss five cases of edge structures. In each case, we consider a hypothetical strict local minimum state S_L with a single edge, and a related global minimum state S_G, also with a single edge; we will refer to these edges as the *suboptimal* and *optimal* edges, respectively. We denote by d the depth of the local minimum state S_L. Note that d will be upper bounded by the maximum cost ascent to reach S_G from S_L by a sequence of transitions that successively change the labeling of only a single pixel at a time.

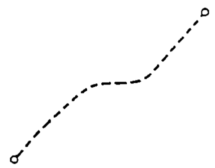


Fig. 9. Edge configuration that contains no edge pixels. The dotted line indicates the optimum edge position.

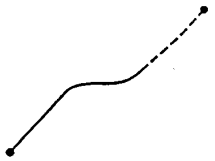


Fig. 10. Suboptimal edge that spans only a portion of the optimal edge position (dotted line).

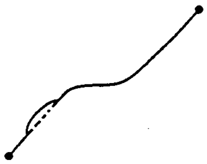


Fig. 11. Suboptimal edge that is slightly displaced from the optimal edge position (dotted line).

Case 1: Here, the optimal edge is a path, and there is no neighboring suboptimal edge, as shown in Fig. 9. By Proposition 6, we can transform S_L to S_G by sequentially adding edge pixels (one at a time) along the sites of the optimal edge, and the depth $d \leq w_f$. It can be shown that the same bound holds when the suboptimal edge is an endsegment of the optimal edge, as shown in Fig. 10.

Case 2: Here, the suboptimal edge is a path, and there is no neighboring optimal edge. By Proposition 7, we can transform S_L to S_G by sequentially deleting edge pixels (one at a time) along the sites of the suboptimal edge, and the depth $d \leq w_f$. It can be shown that the same bound holds when the optimal edge is an endsegment of the suboptimal edge.

Case 3: Here, the suboptimal edge contains a segment that is just slightly displaced from the optimal edge, as shown in Fig. 11. This is possibly the most common local minimum that will be encountered in the minimization process. It is possible to transform S_L to S_G by sequentially locally shifting the position of the edge pixels (one at a time) without breaking the continuity of the edge structure. Consequentially, if there are cost ascents, they will be dominated by curvature costs. The maximum cost ascent will not exceed $2w_c$ as the displaced edge pixels are shifted. Hence, the depth $d \leq 2w_c$.

Case 4: Here, the suboptimal edge contains a segment that is significantly displaced from the position of the optimal edge, as shown in Fig. 12. It is possible to transform S_L

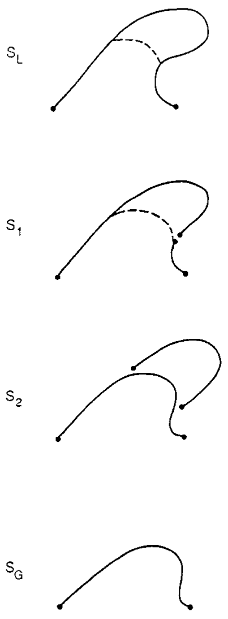


Fig. 12. Suboptimal edge that is significantly displaced from the optimal edge position (dotted line) and a sequence of states transforms the suboptimal edge to the optimal edge.

to S_G as shown in the figure. First, we perturb S_L to S_1 by introducing a local discontinuity in the edge. Clearly, $\Delta F(S_L, S_1) \leq w_f + w_d - w_e$. Next, we generate a sequence of states from S_1 to S_2 by sequentially adding edge pixels along the sites of the optimal edge. Similar to Proposition 6, it can be shown that the maximum cost ascent will not exceed w_f. Finally, we generate a sequence of states from S_2 to S_G by sequentially deleting edge pixels along the sites of the displaced edge. Similar to Proposition 7, it can be shown that the maximum cost ascent will not exceed w_f. Since we are constructing the optimal edge in S_2, it is reasonable to assume that $\Delta F(S_1, S_2) \leq 0$. Hence, we estimate that the depth $d \leq 2w_f + w_d - w_e$.

Case 5: Fig. 13 shows a suboptimal edge that is a combination of the four cases discussed above. It is possible to transform S_L to S_G by iteratively applying cases 1–4 as shown in the figure. Note that each consecutive transition does not increase the cost. Hence, the depth d is the maximum of the four cases above, i.e.

$$d \leq \max\{w_f, 2w_c, 2w_f + w_d - w_e\}. \tag{11}$$

General Case: In the preceding cases, we considered hypothetical local minimum states with a single edge and related global minimum states with a single edge as well. In these cases, the maximum depth of any local (but not global) minimum was (essentially) bounded by the right-hand side of (11). Furthermore, if the states consist of multiple isolated edge structures of the type considered above, then it is clear that the bound in (11) is also valid. Hence, we shall take

$$d_0 = \max\{w_f, 2w_c, 2w_f + w_d - w_e\}$$

as our estimate of d^*. In our view, this estimate is somewhat

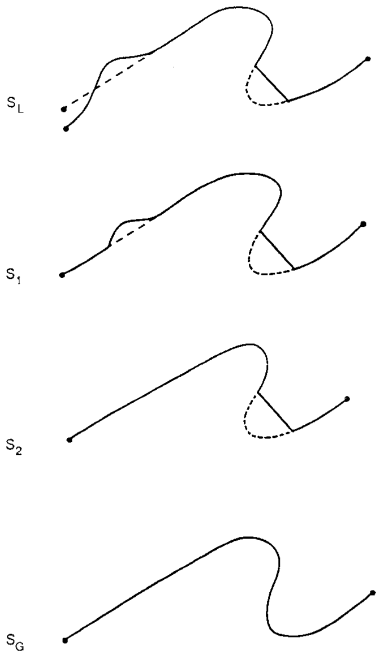


Fig. 13. More general suboptimal edge relative to the optimal edge position (dotted line) and a sequence of states that transforms the suboptimal edge to the optimal edge.

conservative, i.e., for typical images, we may have $d_0 >> d^*$. A precise estimate of d^* for a particular image or class of images seems very difficult to obtain. Nonetheless, our estimate d_0 of d^* does give a qualitative understanding of how d^* depends on the weight parameters.

Finally, in our implementation, we used a temperature schedule of the form

$$T_k = \frac{d_0}{\log(nk_s + 2)} \tag{12}$$

where

$$n = n(k) = \lfloor \frac{k}{N^2} \rfloor \cdot N^2$$

and k_s is a scaling constant ($k_s \geq 0$). Notice that the temperature is changed only after every N^2 iterations of the annealing algorithm. In addition, note that the temperature schedule satisfies (9), and hence, convergence is obtained if and only if $d_0 \geq d^*$.

VI. Experimental Results

In this section, we present some experimental results of detecting edges based on the cost minimization approach using simulated annealing. The ultimate test of any detection technique is in its ability to find edges that correspond to true boundaries in an image. Subjective comparison of the detection performance is made with two other recent edge detection techniques: derivative of Gaussian (∇G) [5] and the facet model approach [7]. Nonmaximal suppression and thresholding was used in the ∇G technique. Nonmaximal suppression was performed by quantizing the edge direction of the ∇G operator output into one of eight possible directions and suppressing the nonmaximum magnitude values in a direction perpendicular to the edge direction. For the facet model approach, we used an 11×11 operator and thresholded the slope of the zero crossings to obtain the edges.

Our selection of weights for the cost function was guided by the results of Section IV on the characteristics of minimum cost edge configurations. We first set $w_c = 1$ to establish a reference. We then set $w_d = 2$ to fix the dissimilarity threshold ξ at a value of 0.5. These two weights are fixed in all of the experiments. In addition, we scaled our choice of dissimiliarity measure f by a constant α such that reasonable edges were detected using a simple thresholding approach (see Proposition 3 and the discussion following it). Next, we selected $w_c \in \{0.25, 0.5, 0.75\}$ and $w_f \in \{2, 3, 4\}$ according to desired values for the minimum edge length η_e and minimum gap length η_g. Finally, we selected $w_t = 2w_f - w_c + w_d - w_e + 0.01$ to avoid thick edges according to Proposition 2. It should be noted that in examples 1–4 (below), we used the *same* values of w_c and w_f and, hence, of all the weights.[1]

Our implementation of simulated annealing to minimize the cost function used the deterministic raster scan method of selecting the next site l to alter the local edge structure. We decided *a priori* on making 200 scans through the image, although as few as 50 scans were required for some of the examples. Note that a single scan through the $N \times N$ image corresponds to N^2 iterations of the annealing algorithm. The probabilities p_i were fixed at $p_1 = \frac{200}{1024}$, $p_2 = \frac{300}{1024}$, $p_3 = \frac{200}{1024}$, $p_4 = \frac{200}{1024}$, and $p_5 = \frac{124}{1024}$. The temperature schedule T_k was as in (12). The value of k_s was selected based on the criterion that T_k should be approximately 0.3 after 200 scans through the image. This value of 0.3 was chosen empirically based on the observation that as the temperature decreased toward 0.3, the annealing process approached a point of "freezing" where very few uphill climbs were made. In the final two scans, the process was quickly "frozen" by dropping the temperature to a value of 0.01.

Example 1*: Rings Image:* Fig. 14 shows a rings image of size of 128×128 pixels, which was made of concentric circles of gray levels 115 and 140, constructed in the manner described in [34]. The image was corrupted with additive zero mean independent identically distributed (i.i.d.) Gaussian noise of standard deviation 25. The detected edges for the noisy image using the ∇G operator, facet model, and cost minimization techniques are shown in the same figure. For the facet model and cost minimization techniques, we found it advantageous to low-pass filter the images with a Gaussian function to suppress the noise effects, although there is a tradeoff here with poorer edge localization [2]. For the ∇G technique, the standard deviation of the Gaussian function was 4.0 pixels. For the facet model and cost minimization techniques, the standard deviation of the Gaussian smoothing function was 3.5. The cost minimization technique used the dissimilarity measure

$$f(R1, R2) = \alpha|m_1 - m_2|.$$

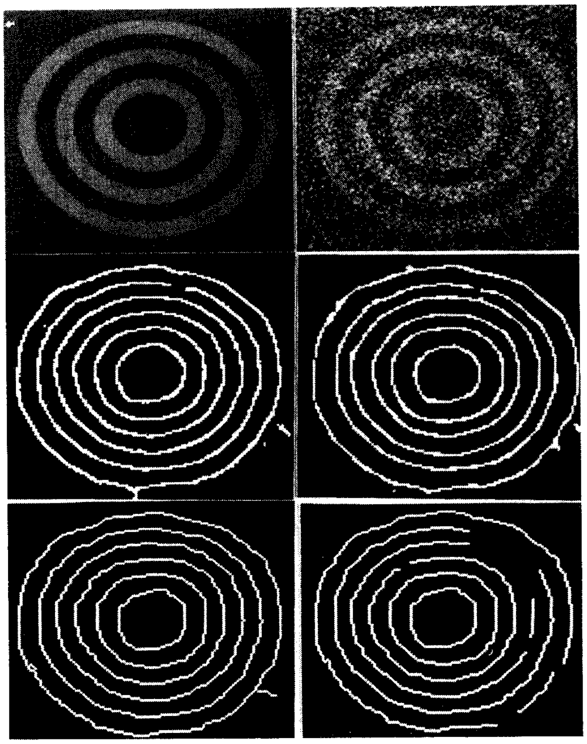


Fig. 14. Rings image. Top left—rings image with grey levels 115 and 140; top right—image and zero-mean additive i.i.d. Gaussian noise of standard deviation 25; middle left—edges of noisy image using ∇G; middle right—edges using facet model approach; bottom left—edges using cost minimization approach with simulated annealing; bottom right—edges using cost minimization approach with strict descent.

where m_1 and m_2 are the sample means of the gray-level values in regions $R1$ and $R2$, respectively. The weights were set to (see above discussion) $w_c = 0.5$, $w_d = 2$, $w_e = 1$, $w_f = 3$, and $w_t = 6.51$, corresponding to a minimum edge length $\eta_e = 3$ and a minimum gap length $\eta_g = 1$. Notice that all the edges detected using the cost minimization technique are thin and continuous. The other two techniques show edges that are slightly fragmented and thick at certain portions.

Fig. 14 also shows the results obtained by using a descent search algorithm to minimize the cost function. The search was performed by using the annealing algorithm and setting the temperature T_k to a constant value of 0.01.

Fig. 15 shows an example of the edges produced at intermediate scans of the cost minimization process. It can be seen that there is significant improvement in the first 50 scans. After about 100 scans, the process reaches comparatively stable low-cost states.

Example 2*: Airport Image:* Fig. 16 shows a 256×256 airport image and the edges detected using the cost minimization technique with the same weights and dissimilarity measure as in Example 1.

Example 3*: Plate Image:* Fig. 17 shows a 256×256 image of a plate and the detected edges using the cost minimization technique with the same weights and dissimilarity measure as in Example 1.

[1] It can be shown that for the weights used in the examples, connectivity is preserved at T junctions.

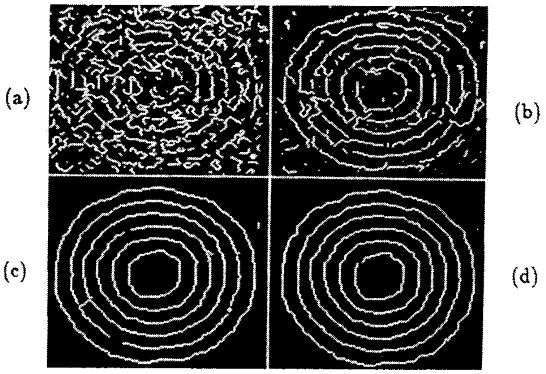


Fig. 15. Intermediate edge configurations in annealing process: (a) Edges at tenth scan through the image; (b) edges at 20th scan; (c) edges at 50th scan; (d) edges at 100th scan.

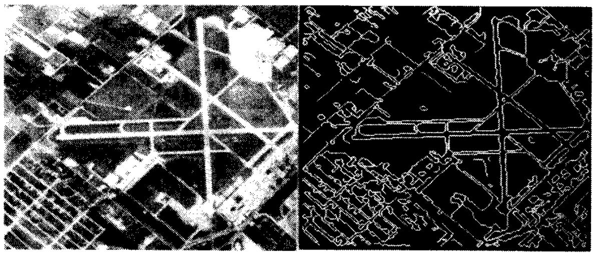


Fig. 16. Airport image. Left—original image; right—edges detected using cost minimization approach.

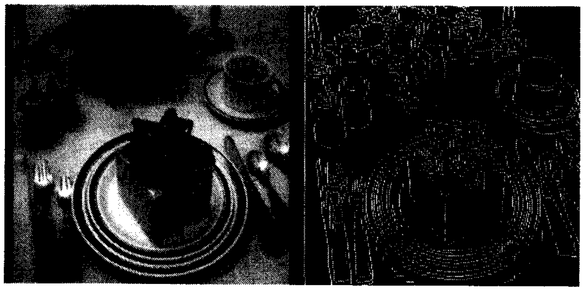


Fig. 17. Plate image. Left—original image; right—edges detected using cost minimization.

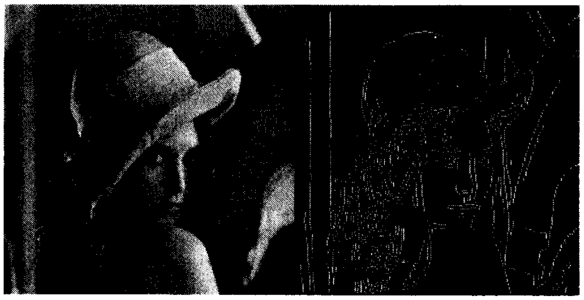


Fig. 18. Lena image. Left—original image; right—edges detected using cost minimization.

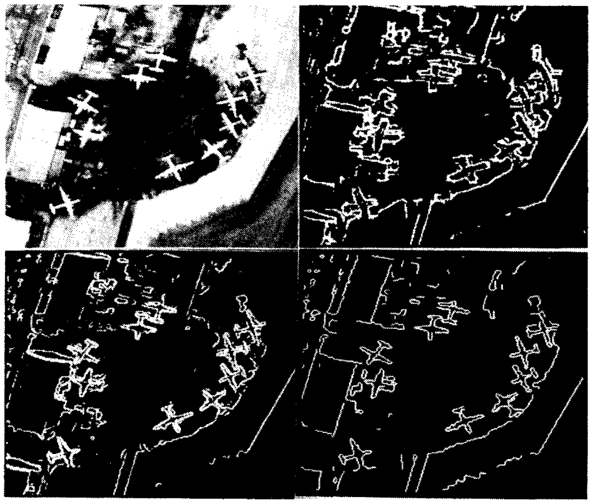


Fig. 19. Airplanes image. Top left—original image; top right—edges detected using facet model approach; bottom left: —edges detected using ∇G technique; bottom right—edges detected using cost minimization approach with new dissimilarity measure.

Example 4: *Lena Image:* Fig. 18 shows a 256×256 image of Lena and the detected edges using the cost minimization technique with the same weights and dissimilarity measure as in Example 1.

Example 5: *Airplanes Image:* Fig. 19 shows an image containing ten airplanes. The important features of interest in this image are the ten airplanes, the two large buildings on the left part of the image, and portions of the tarmac. The edges detected using the ∇G operator and facet model techniques are also shown in the same figure. The ∇G technique used a value of 1.0 for the standard deviation of the Gaussian function. For both of these techniques, the thresholds and relevant parameters were chosen to recover as much of the boundaries of the airplanes and the large buildings as possible, without introducing an excessive number of false edges. However, it was found that selecting a threshold low enough to recover the boundaries of the large buildings resulted in a high degree of false edges being detected.

It can be seen that the important features of interest in the image generally have a lighter shade than the background, which corresponds to higher image intensity values. This information is easily incorporated into the cost minimization technique by modifying the previously defined dissimilarity measure $f(R1, R2)$ to be

$$f(R1, R2) = \alpha|m_1 - m_2|g(\max\{m_1, m_2\})$$

where $g(\bullet)$ is the ramp function shown in Fig. 20. With this measure, two regions will have large dissimilarity when their average intensities are sufficiently different, *and* one of the average intensities is sufficiently high. The weights were set to (see above discussion) $w_c = 0.25$, $w_d = 2$, $w_e = 1$, $w_f = 2$, and $w_t = 4.76$, corresponding to a minimum edge length $\eta_l = 2$ and a minimum gap length $\eta_g = 1.5$. The detected edges using the cost minimization technique are also shown

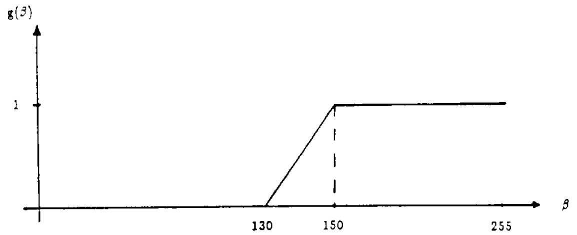


Fig. 20. Piecewise linear function using in the definition of dissimilarity measure for Example 5 (airplanes image).

in Fig. 19. Notice that the edges of the large buildings, the airplanes, and the boundary region of the tarmac on the lower right portion of the image are clearly visible.

Example 6: *Texture Box Image:* Fig. 21 shows an image of size 128 × 128 containing two texture regions. The image was constructed by adding zero-mean i.i.d Gaussian random noise to an image of constant gray level equal to 128. Within the center 64 × 64 box region, the noise standard deviation was 30; outside of the box, the noise standard deviation was 10. Since the boundary of the box does not correspond to a step or ramp, it is not possible to use conventional derivative-based techniques to detect the edges of the box. We applied the cost minimization technique to this example with the dissimilarity measure

$$f(R1, R2) = \alpha|\sigma_1 - \sigma_2|.$$

where σ_1 and σ_2 are the sample standard deviations of the grey-level values in regions $R1$ and $R2$, respectively. The weights were set to (see above discussion) $w_c = 0.75$, $w_d = 2$, $w_e = 1$, $w_f = 4$, and $w_t = 8.26$, corresponding to a minimum edge length $\eta_e = 4$ and a minimum gap length $\eta_g = 1.25$. The detected edges are shown in Fig. 21.

VII. Conclusion

In this paper, we have cast edge detection as a problem in cost minimization. This was achieved by the formulation of a cost function that evaluates the quality of edge configurations. The function is a linear sum of weighted cost factors. The cost factors capture desirable characteristics of edges such as accuracy in localization, thinness, and continuity. Edges were detected by finding the edge configurations that minimize the cost function. This approach is capable of detecting many different types of edges including intensity and texture edges. We have given a mathematical description of edges and stated several propositions concerning the characteristics of edges in minimum cost configurations. The characteristics depend on the weight of the cost factors. These propositions provide guidelines on the choice of weights to achieve certain characteristics of the detected edges.

Simulated annealing was used to minimize the cost function. We have developed a novel set of strategies for generating candidate states and devised a suitable temperature schedule. Experimental results that verify the usefulness of our cost minimization approach to edge detection were given.

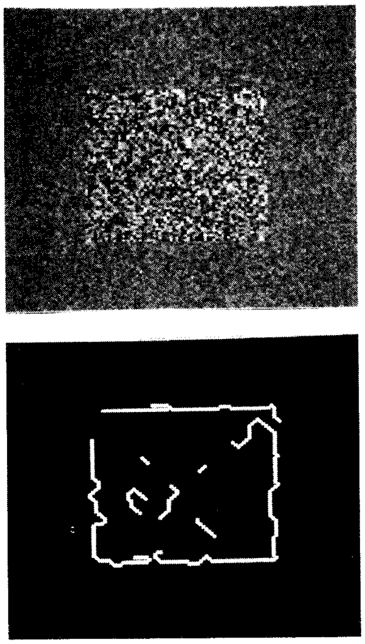


Fig. 21. Texture box image. Top—original image; bottom—edges detected using cost minimization.

Appendix A

Proof of Proposition 2: Let S be a state that contains a thick edge pixel at site l. Let S' be the state that is identical to S except at site l (where it has a nonedge pixel). From (5), the incremental cost can be written as

$$\Delta F(S. S') = \sum_i w_i \Delta C_i(S. S').$$

As pointed out in Section III-C, $\Delta C_i(S. S')$ depends only on the pixels in a 5 × 5 window around site l. Hence, by taking into consideration the various edge structures in such a 5 × 5 window around site l, we can obtain the following bounds for the incremental cost factors: $-7 \leq \Delta C_c \leq -1$, $0 \leq \Delta C_d \leq 1$, $\Delta C_e = -1$, $0 \leq \Delta C_f \leq 2$, and $-7 \leq \Delta C_t \leq -1$. Examples of edge structures that achieve the upper and lower bounds for ΔC_c are shown in Fig. A1. Examples of edge structures that achieve the upper bound for ΔC_f are shown in Fig. A2. Hence,

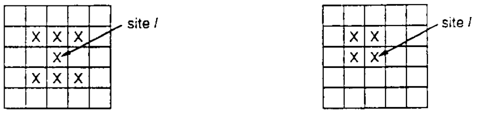
$$\Delta F(S. S') \leq 2w_f + w_d - w_c - w_e - w_t < 0.$$

Hence, for any state that contains a thick edge pixel, we can find a lower cost state, and therefore, there can be no thick edges in any minimum cost state. □

Appendix B

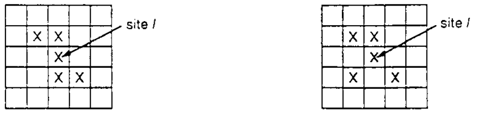
Proof of Proposition 3:

a) Suppose S_G has a nonedge pixel at site l. Let S be the state that is identical to S_G except at site l (where it has an edge pixel). Using (5), the incremental cost from S_G



(a) (b)

Fig. A1. Computation of ΔC_c: (a) Removal of edge pixel at site l results in $\Delta C_c = -7$; (b) removal of edge pixel at site l results in $\Delta C_c = -1$.



(a) (b)

Fig. A2. Computation of ΔC_f: Removal of edge pixel at site l_1 of either (a) or (b) results in $\Delta C_f = 2$.

to S can be written as

$$\Delta F(S_G, S) = \sum_i w_i \Delta C_i(S_G, S).$$

Now $\Delta C_d = -d(l)$ and $\Delta C_e = 1$. By exhaustive examination of the edge structures in a 5×5 window around site l, it can be shown that $w_c\Delta C_c + w_f\Delta C_f + w_t\Delta C_t$ is maximized by one of the two edge structures shown in Fig. A3. Since $\Delta C_c = 0$, $\Delta C_f = 1$ and $\Delta C_t = 0$ for the structure in Fig. A3(a), and $\Delta C_c = 7$, $\Delta C_f = 0$ and $\Delta C_t = 7$ for the structure in Fig. A3(b), it follows that

$$w_c\Delta C_c + w_f\Delta C_f + w_t\Delta C_t \le \max\{7w_c + 7w_t, w_f\}.$$

Hence

$$\Delta F(S_G, S) \le \max\{7w_c + 7w_t, w_f\} - w_d d(l) + w_e.$$

Since S_G is a minimum cost state, we have $\Delta F(S_G, S) \ge 0$, and therefore

$$d(l) \le \frac{\max\{7w_c + 7w_t, w_f\} + w_e}{w_d}$$

as required (this proves the contrapositive of the Proposition statement).

b) Suppose S_G has an edge pixel at site l. Let S be the state that is identical to S_G except at site l (where it has a nonedge pixel). Using (5), the incremental cost from S to S_G can be written as

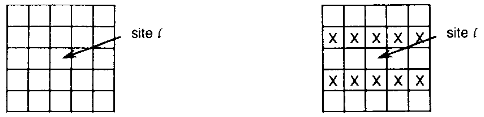
$$\Delta F(S, S_G) = \sum_i w_i \Delta C_i(S, S_G).$$

Now, $\Delta C_d = -d(l)$, $\Delta C_e = 1$ and $\Delta C_t = 0$. By exhaustive examination of the edge structures in a 5×5 window around site l, it can be shown that $w_c\Delta C_c + w_f\Delta C_f$ is minimized by one of the two edge structures shown in Fig. A4. Since $\Delta C_c = 0$, $\Delta C_f = -1$ for the structure shown in Fig. A4(a), and $\Delta C_c = 1$, $\Delta C_f = -2$ for the structure shown in Fig. A4(b), it follows that

$$w_c\Delta C_c + w_f\Delta C_f \ge \min\{w_c - 2w_f, -w_f\}.$$

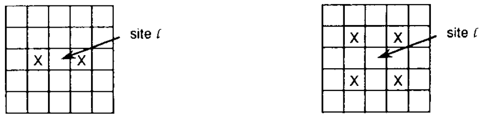
Hence

$$\Delta F(S, S_G) \ge \min\{w_c - 2w_f, -w_f\} - w_d d(l) + w_e.$$



(a) (b)

Fig. A3. $w_c\Delta C_c + w_f\Delta C_f + w_t\Delta C_t$ is maximized by insertion of edge pixel at site l in (a) or (b).



(a) (b)

Fig. A4. $w_c\Delta C_c + w_f\Delta C_f$ is minimized by insertion of edge pixel at site l in (a) or (b).

Since S_G is a minimum cost state, we have $\Delta F(S, S_G) \le 0$ and therefore

$$d(l) \ge \frac{\min\{w_c - 2w_f, -w_f\} + w_e}{w_d}$$

as required.

c) The proof proceeds similarly to part b), except here, $w_c\Delta C_c + w_f\Delta C_f \ge 0$ (trivially). □

APPENDIX C

Proof of Proposition 4: Let S be the state that is identical to S_G except at the sites of E, where it has nonedge pixels. Using (5), the incremental cost from S to S_G is given by

$$\Delta F(S, S_G) = \sum_i w_i \Delta C_i(S, S_G).$$

Clearly, $\Delta C_c \ge 0$, $\Delta C_d \ge -||E||$ (since $d(l) \le 1$), $\Delta C_e = ||E||$, $\Delta C_f \ge 1$, and $\Delta C_t = 0$. Hence

$$\Delta F(S, S_G) \ge -w_d||E|| + w_e||E|| + w_f.$$

Since S_G is a minimum cost state, we have $\Delta F(S, S_G) \le 0$, and therefore

$$||E|| \ge \frac{w_f}{w_d - w_e}$$

as required. □

Proof of Proposition 5: Let S be the state that is identical to S_G except at the sites of E, where it has edge pixels. E does not include the two edge endpoints in S_G; see Fig. A5. Using (5), the incremental cost from S_G to S is given by

$$\Delta F(S_G, S) = \sum_i w_i \Delta C_i(S_G, S).$$

Clearly, $\Delta C_c \le 2$, $\Delta C_d \le 0$, $\Delta C_e = ||E||$, $\Delta C_f = -1$, and $\Delta C_t = 0$. Hence

$$\Delta F(S_G, S) \le 2w_c + w_e||E|| - w_f.$$

Since S_G is a minimum cost state, we have $\Delta F(S_G, S) \ge 0$ and therefore

$$||E|| \ge \frac{w_f - 2w_c}{w_d}$$

as required. □

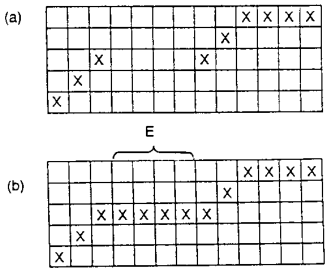


Fig. A5. Two edge endpoints in (a) are linked by a straight path E in (b).

APPENDIX D

Let $I(A)$ denote the sites of the pixels in A (hence $L = I(S)$ and $L_{ij} = I(W_{ij}(S))$).

Proof of Proposition 6: Let E be the path $e_1, \ldots, e_K$, and let E_k be the endsegment $e_1, \ldots, e_k$ of E. Let $S_{G,k}$ be the state identical to S_G except at the sites of E_k, where it has nonedge pixels. Using (5), the incremental cost from $S_{G,k}$ to S_G is given by

$$\Delta F(S_{G,k}, S_G) = \sum_i w_i \Delta C_i(S_{G,k}, S_G).$$

Clearly, $\Delta C_c = \sum_{l \in I(E_{k+1})} C_c(S_G, l)$, $\Delta C_d = -\sum_{l \in I(E_k)} d(l)$, $\Delta C_e = k$, $\Delta C_f = 0$, and $\Delta C_t = 0$. Hence

$$\Delta F(S_{G,k}, S_G) = w_c \sum_{l \in I(E_{k+1})} C_c(S_G, l) - w_d \sum_{l \in I(E_k)} d(l) + w_e k \leq 0 \quad \text{(A1)}$$

since S_G is a global minimum state.

Now, let S_k be the state identical to S except as the sites of E_k, where it has edge pixels. Using (5), the incremental cost from S to S_k is given by

$$\Delta F(S, S_k) = \sum_i w_i \Delta C_i(S, S_k).$$

Clearly, we now have $\Delta C_d = -\sum_{l \in I(E_k)} d(l)$, $\Delta C_e = k$, $\Delta C_f = 1$, and $\Delta C_t = 0$. In addition

$$\begin{aligned}\Delta C_c &= \sum_{l \in I(E_k)} C_c(S_k, l) \\ &= \sum_{l \in I(E_{k-1})} C_c(S_G, l) \\ &\leq \sum_{l \in I(E_{k+1})} C_c(S_G, l)\end{aligned}$$

Hence

$$\begin{aligned}\Delta F(S, S_k) \leq & w_c \sum_{l \in I(E_{k+1})} C_c(S_G, l) \\ & - w_d \sum_{l \in I(E_k)} d(l) + w_e k + w_f \\ \leq & w_f\end{aligned}$$

by (A1), as required. Finally, it is clear that $\Delta F(S, S') = \Delta F(S_{G,0}, S_G) \leq 0$ since S_G is a global minimum state. □

Proof of Proposition 7: Let E be the path $e_1, \ldots, e_K$, and let E_k be the endsegment $e_1, \ldots, e_k$ of E. Let $S_{G,k}$ be the state identical to S_G except as the sites of E_k, where it has edge pixels. Using (5), the incremental cost from $S_{G,k}$ to S_G is given by

$$\Delta F(S_{G,k}, S_G) = \sum_i w_i \Delta C_i(S_{G,k}, S_G).$$

Clearly, $\Delta C_c = -\sum_{l \in I(E_k)} C_c(S_{G,k}, l)$, $\Delta C_d = \sum_{l \in I(E_k)} d(l)$, $\Delta C_e = -k$, $\Delta C_f = -1$, and $\Delta C_t = 0$. Hence

$$\begin{aligned}\Delta F(S_{G,k}, S_G) = & - w_c \sum_{l \in I(E_k)} C_c(S_{G,k}, l) \\ & + w_d \sum_{l \in I(E_k)} -w_e k - w_f \leq 0 \quad \text{(A2)}\end{aligned}$$

since S_G is a global minimum state.

Now, let S_k be the state identical to S except at the sites of E_k, where it has nonedge pixels. Using (5), the incremental cost from S to S_k is given by

$$\Delta F(S, S_k) = \sum_i w_i \Delta C_i(S, S_k).$$

Clearly, we now have $\Delta C_d = \sum_{l \in I(E_k)} d(l)$, $\Delta C_e = -k$, $\Delta C_f = 0$, and $\Delta C_t = 0$. In addition

$$\begin{aligned}\Delta C_c &= - \sum_{l \in I(E_{k+1})} C_c(S, l) \\ &\leq - \sum_{l \in I(E_{k-1})} C_c(S, l) \\ &= - \sum_{l \in I(E_k)} C_c(S_{G,k}, l)\end{aligned}$$

Hence

$$\begin{aligned}\Delta F(S, S_k) \leq & - w_c \sum_{l \in I(E_k)} C_c(S_{G,k}, l) \\ & + w_d \sum_{l \in I(E_k)} d(l) - w_e k \leq w_f\end{aligned}$$

by (A2), as required. Finally, it is clear that $\Delta F(S, S') = \Delta F(S_{G,K}, S_G) \leq 0$ since S_G is a global minimum state. □

REFERENCES

[1] A. Rosenfeld and A. C. Kak, *Digital Picture Processing.* New York: Academic, 1982, vols. 1 and 2.

[2] V. Torre and T. A. Poggio, "On edge detection," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. PAMI-8, no. 2, pp. 147–163, Mar. 1986.

[3] T. Peli and D. Malah, "A study of edge detection algorithms," *Comput. Graphics Image Processing,* vol. 20, pp. 1–21, 1982.

[4] F. M. Dickey and K. S. Shanmugam, "Optimum edge detection filter," *Appl. Optics,* vol. 16, no. 1, pp. 145–148, Jan. 1977.

[5] J. Canny, "A computational approach to edge detection," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. PAMI-8, no. 6, pp. 679–698, Nov. 1986.

[6] D. Marr and E. Hildreth, "Theory of edge detection," in *Proc. Royal Soc. London,* 1980, pp. 187–217., pp. 187–217.

[7] R. M. Haralick, "Digital step edges from zero crossing of second directional derivatives," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. PAMI-6, no. 1, pp. 58–68, Jan. 1984.

[8] V. Nalwa and T. O. Binford, "On detecting edges," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. PAMI-8, no. 6, pp. 699–714, Nov. 1986.

[9] D. Mumford and J. Shah, "Boundary detection by minimizing functionals," in *Proc. IEEE Comput. Vision Patt. Recogn. Conf.* (San Francisco), 1985, pp. 22–26.

[10] A. Blake and A. Zisserman, *Visual Reconstruction.* Cambridge, MA: MIT Press, 1987.

[11] A. Martelli, "An application of heuristic search methods to edge and contour detection," *Commun. ACM,* vol. 19, no. 2, pp. 73–83, Feb. 1976.

[12] G. P. Ashkar and J. W. Modestino, "The contour extraction problem with biomedical applications," *Comput. Graphics Image Processing,* vol. 7, pp. 331–355, 1978.

[13] P. H. Eichel and E. J. Delp, "Sequential edge detection in correlated random fields," in *Proc. IEEE Comput. Vision Patt. Recogn. Conf.* (San Francisco), June 1985, pp. 14–21.

[14] E. J. Delp and C. H. Chu, "Detecting edge segments," *IEEE Trans. Syst. Man Cybern.*, vol. SMC-15, no. 1, pp. 144–152, Jan. 1985.

[15] P. H. Eichel, E. J. Delp, K. Koral, and A. J. Buda," "A method for fully automatic definition of coronary arterial edges from cineangiograms," *IEEE Trans. Med. Imaging,* vol. 7, no. 4, pp. 313–320, Dec. 1988.

[16] A. Rosenfeld, *Image Modeling.* New York: Academic, 1981.

[17] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and the bayesian restoration of images," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. PAMI-6, no. 6, pp. 721–741, Nov. 1984.

[18] J. L. Marroquin, "Probablistic solution of inverse problems," Ph.D. thesis, Dept. of Elect. Eng. Comput. Sci., Mass. Inst. Technol., 1985.

[19] B. Gidas, "A renormalization group approach to image processing problems," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 11, pp. 164–180, Feb. 1989.

[20] S. Lakshmanan and H. Derin, "Simultaneous parameter estimation and segmentation of Gibbs random fields using simulated annealing," *IEEE Trans. Patt. Anal. Machine Intell.,* vol. 11, no. 8, pp. 799–813, Aug. 1989.

[21] D. Geman, S. Geman, C. Graffigne, and P. Dong, "Boundary detection by constrained optimization," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 12, no. 7, pp. 609–628, July 1990.

[22] B. Hajek, "Cooling schedules for optimal annealing," *Math. Oper. Res.*, vol. 13, no. 2, pp. 311–329, May 1988.

[23] B. J. Schachter, A. Lev, S. W. Zucker, and A. Rosenfeld, "An application of relaxation methods to edge reinforcement," *IEEE Trans. Syst. Man Cybern.,* vol. SMC-7, no. 11, pp. 813–816, Nov. 1977.

[24] R. Hummel and S. W. Zucker, "On the foundations of relaxation labeling processes," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. PAMI-5, pp. 267–287, 1983.

[25] D. Marr, *Vision.* New York: W. H. Freeman, 1982.

[26] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Sci.,* vol. 220, no. 4598, pp. 671–680, May 13, 1983.

[27] C. C. Skiscim and B. L. Golden, "Optimization by simulated annealing: A preliminary computational study for the TSP," in *Proc. 1983 Winter Simulation Conf.,* 1983, pp. 523–535.

[28] A. A. El Gamal, L. A. Hemachandra, I. Shperling, and V. K. Wei, "Using simulated annealing to design good codes," *IEEE Trans. Inform. Theory,* vol. IT-33, no. 1, pp. 116–123, Jan. 1987.

[29] P. Carnevali, L. Coletti, and S. Patarnello, "Image processing by simulated annealing," *IBM J. Res. Develop.*, vol. 29, no. 6, pp. 569–579, Nov. 1985.

[30] D. Mitra, F. Romeo, and A. Sangiovanni-Vincentelli, "Convergence and finite-time behavior of simulated annealing," *Advances Appl. Probability,* vol. 18, pp. 747–771, 1986.

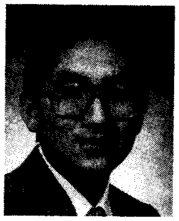
[31] S. B. Gelfand and S. K. Mitter, "Analysis of simulated annealing for optimization," in *Proc. 24th Conf. Decision Contr.* (Ft. Lauderdale, FL), Dec. 1985, pp. 779–786.

[32] B. Gidas, "Nonstationary markov chains and convergence of the annealing algorithm," *J. Stat. Phys.*, vol. 39, nos. 1/2, pp. 73–131, 1985.

[33] H. L. Tan, "Edge detection by cost minimization," Ph.D. thesis, School of Elect. Eng., Purdue Univ., West Lafayette, IN, Dec. 1988.

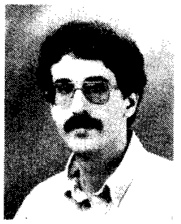
[34] L. Kitchen and A. Rosenfeld, "Edge evaluation using local edge coherence," *IEEE Trans. Syst. Man Cybern.*, vol. SMC-11, no. 9, pp. 597–605, Sept. 1981.

[35] H. L. Tan, S. B. Gelfand, and E. J. Delp, "A cost minimization approach to edge detection using simulated annealing," in *Proc. IEEE Comput. Vision Patt. Recogn. Conf.* (San Diego), June 1989, pp. 86–91.



Hin Leong Tan was born in the Republic of Singapore. He received the B. S. (with high scolastic honors) and M. S. degrees in electrical and computer engineering from Oregon State University, Corvallis, in 1983 and 1984, respectively, and the Ph.D. degree in electrical engineering from Purdue University, West Lafayette, IN, in 1988.

From 1984 to 1985, he worked as a VLSI design engineer at Integrated Devices Technology, Santa Clara, CA. Since 1988, he has been with the Image Electronics Center of Kodak Research Laboratories, Rochester, NY. His current research interests are in optical character recognition, image segmentation, pattern recognition, and image processing.



Saul B. Gelfand (M'88) received the S.B. degree in physics and the S.M. and Ph.D. degrees in electrical engineering from the Massachusetts Institute of Technology, Cambridge, in 1978, 1982, and 1987, respectively.

From 1978 to 1980, he was with Scientific Systems, Inc., Cambridge, MA, where he worked on automatic electrocardiogram analysis. From 1981 to 1983, he held summer positions at Bolt Beranek and Newman, Cambridge, MA, where he worked on multitarget tracking problems. Since 1987, he has been an Assistant Professor in the School of Electrical Engineering, Purdue University, West Lafayette, IN. His research interests include nonlinear and adaptive signal processing, optimization theory, pattern recognition, and neural networks.

Edward J. Delp (SM'86) was born in Cincinnati, OH. He received the B.S.E.E. (cum laude) and M.S. degrees from the University of Cincinnati, Cincinnati, OH, and the Ph.D. degree from Purdue University, West Lafayette, IN.

From 1980–1984, he was with the Department of Electrical and Computer Engineering, the University of Michigan, Ann Arbor. Since 1984, he has been with the School of Electrical Engineering, Purdue University, West Lafayette IN, where he is a Professor of Electrical Engineering. His research interests include ill-posed inverse problems in computational vision, nonlinear filtering using mathematical morphology, image coding, and medical imaging. He has also consulted for various companies and government agencies in the areas of signal and image processing, robot vision, pattern recognition, and secure communications.

Dr. Delp is a member of Tau Beta Pi, Eta Kappa Nu, Phi Kappa Phi, Sigma Xi, the, Optical Society of America, the Pattern Recognition Society, and SPIE. He is an associate editor of the IEEE TRANSACTIONS ON PATTERN AANALYSIS AND MACHINE INTELLIGENCE and is a member of the editorial board of the *International Journal of Cardiac Imaging*. He is also co-editor of the book *Digital Cardiac Imaging* (Martinus Nijhoff). He was the Chairman of the 1990 SPIE/SPSE Conference on Nonlinear Image Processing. He is a member of the program committee for the 1991 IEEE Computer Vision and Pattern Recognition Conference. In 1989, he received the Honeywell Award for excellence in teaching. In 1990, he received a Fullbright Fellowship to teach and perform research at the Universitat Polytecnica de Catalunya, Barcelona, Spain.