Moment-preserving edge detection and its application to image data compression

Hua-Sheng Hsu*
National Chiao Tung University
Institute of Computer Science and
Information Engineering
Hsinchu, Taiwan 300

Wen-Hsiang Tsai
National Chiao Tung University
Department of Computer and Information
Science
Hsinchu, Taiwan 300

Abstract. In contrast to the numerous edge-detection techniques that detect edges either point by point or using overlapping circular windows, an edge detector using nonoverlapping rectangular windows is proposed. The detector examines the pixels within each rectangular window of an image, and decides whether an edge element is present or not in the window. Based on the gray and mass moment-preserving principles, the step edge is estimated locally to subpixel accuracy using analytical formulas. To apply the edge detection results to image compression, the detected edge elements are then tracked and grouped based on proximity and orientation. Using the line parameters of the grouped edge elements, region boundaries are approximated in a piecewise linear manner. This reduces the amount of data required to describe region shapes and is useful for compressing some types of images. Good experimental results of compressing character and trademark images are also included to show the feasibility of the proposed approach.

Subject terms: visual communication; edge detection; image data compression.

Optical Engineering 32(7), 1596–1608 (July 1993).

1 Introduction

Edge detection plays an important role in many applications, such as industrial inspection, pattern recognition, scene analysis, image coding, etc. Since human visual perception is sensitive to edges and contours, we can simplify shape analysis and perform more accurate object recognition if the boundary of an object can be extracted successfully. In addition, image data compression is crucial to reducing the amount of storage and transmission time in the applications of database management and network communication, respectively. Therefore, some effective data reduction techniques that do not destroy object shape boundaries are desired. The goal of this study is to develop a new edge-detection technique with subpixel accuracy that can be used in image data compression.

For a survey of edge-detection techniques with ''pixel'' accuracy, see Ref. 1. The gradient and the Laplacian are two basic edge operators for noiseless pictures, which respond to changes in gray levels.[1] To reduce the effect of noise on the response of an operator, some operators that compute differences of local averages were proposed.[1] In addition, many approaches to modifying the gradient and the Laplacian operators have also been provided. Nevatia and Babu[2] correlated windowed data with masks corresponding to square-aperture-sampled ideal step edges oriented in six selected directions. The magnitude of the correlated output and the highest output at each pixel are recorded as the edge magnitude and angle, respectively. Frei and Chen[3] developed a set of orthogonal functions that were closely related to distinct image features and allowed efficient extraction of object boundary elements.

Statistical analysis is another approach to edge detection. Suk and Hong[4] described a new edge-extraction technique specifically developed for noisy images that eliminates the necessity of noise-removal preprocessing or postprocessing. The algorithm was based on parallel statistical tests for which indeterministic decisions were allowed. From the viewpoint of statistical classification, Kundu and Mitra[5] assumed a two-valued region model. Whenever the sliding window enters the decision process, only three parameters are varied to achieve the optimal performance even in the presence of

*Current affiliation: Industrial Technology Research Institute, Computer and Communications Laboratories, Hsinchu, Taiwan 300.

Paper VCI-14 received Sep. 5, !992; revised manuscript received Feb.6, 1993; accepted for publication Feb. 10, 1993.

noise. Huang and Tseng[6] exploited the statistical theory of hypothesis testing based on the likelihood ratio test to filter the noise and detect the edges in the mean time.

Haralick[7] proposed the use of the facet model for edge detection. The facet model fits a polynomial surface (plane, quadratic, etc.) to the image data in a neighborhood of each pixel. This polynomial surface is used as a model of the underlying continuous image. Thus, any processing to be done on the image data is instead done on this polynomial surface. An edge was defined to occur at a pixel if at some point in the pixel's neighborhood there is a negatively sloped zero crossing of the second directional derivative taken in the direction of a nonzero gradient at the pixel center. But some problems are associated with the use of zero crossings of the second derivative to detect and/or localize step edges. The reason is that the derivative operation is very sensitive to noise; besides, if the surface-fitting basis is inadequately selected, then the zero crossing can result in extremely bad localization.

Because some problems need accurate measurements, several researchers have tried to develop methods to obtain edge locations with "subpixel" accuracy. Hueckel[8] first derived a regional operator of this type for 2-D images. He developed an algorithm to determine the presence of edges by fitting a unit disk of data to a Hilbert space spanned by a set of nine orthonormal Fourier-type basis functions. An ideal edge of arbitrary contrast, orientation, etc. was also projected onto this space. The parameters of the ideal edge whose coefficients, in a sum of square error sense, most closely matched those of the input image was used to estimate the edge location to subpixel accuracy. Instead of using Hueckel's weighting function over a bounded disk,[8] Hartley[9] took a data structure called *pyramid* in which the basis has an approximately Gaussian weighting function over the plane. Using subsets of this basis, the operator becomes equivalent to either a gradient edge operator or a zero-crossing edge operator.

Similar to Haralick's method,[7] a problem with Hueckel's method[8] is the choice of an adequate basis. In addition, it often fails to exploit the directional characteristics of edges. Unlike the previous work, the basis proposed by Nalwa and Binford[10] was constrained to be directional. The edge-detection process is to fit a series of oriented 1-D surfaces to each window in the sense of the least squares error with the fewest parameters. From the above three methods, it should be emphasized that there cannot be any unique basis that is appropriate to describe the image data in all windows. If we attempt to do this, we will obtain noise-sensitive results if the basis is not minimal and extremely bad localization if the basis is inadequate.

Tabatabai and Mitchell[11] presented an operator that locates edges by fitting the first three gray-level moments to the image data. They derived a line equation to calculate a local edge location to subpixel accuracy. Since a line equation is a degenerate case of conics, Chen and Tsai[12] proposed an advanced detector to estimate curved edge locations locally using a parabolic equation. They used a decent method without derivatives to solve the coefficients as a minimization problem, and curved edge points could be detected to subpixel accuracy, too. Moreover, Liu and Tsai[13] introduced the principle of mass moment preserving to detect a general corner in a given circular area, with the first three gray-level moments preserved also. By the assumption that the edges are locally straight, Hyde and Davis[14] provided another subpixel line edge detector that best predicts the intensities of the pixels along the edge.

Image compression is significant for mass image transmission and storage. Certain applications of image processing require the storage of a large number of images. An example is the identification of trademark images to prevent a newly registered trademark from being identical or similar to existing ones. For this application, a huge image database for all existing marks is required. Hence, image compression is useful.

Since one of the most striking features of human visual perception is the sensitivity to edges and contours, recent neurophysiological results suggest the use of a general contour/texture model for image processing and coding. Some techniques attempt to use explicit representations of geometric structures to code images. The coding problem includes (1) efficient extraction and coding of contour information and (2) coding of texture between contours. The work of Graham[15] starts by noting that if the spatial frequency spectrum of an image is split into a low-frequency and a high-frequency part, the high-frequency part will reproduce the sharp intensity changes in the image, i.e., the contours marking the transitions between the object and the background. On the other hand, the low-frequency part approximates the smooth areas between contours. By using explicit representations of contour geometry, the efficiency of the coding of the high-frequency part can be increased. Kunt, Ikonomopoulos, and Kocher[16] used the duplicate approach to contour extraction in trying to define the smooth areas between contours. Their approach is based on a region-growing and split-and-merge scheme to find the smooth regions. These regions are then approximated using low-order polynomials. The contours are obtained as the boundaries between the smooth regions.

The methods of edge detection mentioned above can be classified into two categories. Each method in the first category should be repeated over the whole image by shifting the window in one-pixel steps in the x and y directions.[2-10,17] A method of the second category aims to divide a digital picture into a set of contiguous overlapping 4.5-unit circles, the distance between the centers of every two neighboring circles in the horizontal or vertical direction being 5 pixels.[11-13] However, each of them suffers from a prominent weakness. That is, a rather heavy processing burden necessary to achieve this way of edge detection or contour extraction.

In this paper, an edge detector is proposed to scan the entire image by nonoverlapping rectangular windows instead of overlapping circles and examine whether an edge element is present within each window by statistical analysis of gray-value variations. If a rectangular window contains an edge element, the edge operator then uses the gray and mass moment-preserving principle to obtain an analytical solution that determines the step edge location to subpixel accuracy. Once a group of short linear edge elements are detected, a tracking technique is used to stretch an edge element along its orientation as long as possible under a given linearity criteria, and the final tracked elements are approximated by piecewise line segments to obtain a more compact representation. In this way, data compression of certain types of im-

103	101	99	100	102	105	104	100
101	104	102	99	100	102	102	101
99	101	102	101	102	100	101	104
98	98	97	99	97	96	95	91
99	101	101	101	96	79	60	44
101	99	97	84	60	37	20	8
97	89	71	49	29	16	9	6
93	70	42	18	5	0	0	0

(a)

97	97	97	97	97	97	97	97
97	97	97	97	97	97	97	97
97	97	97	97	97	97	97	97
97	97	97	97	97	97	15	15
97	97	97	97	15	15	15	15
97	97	15	15	15	15	15	15
15	15	15	15	15	15	15	15
15	15	15	15	15	15	15	15

(b)

Fig. 1 Step edge detection in an 8 × 8 rectangular window: (a) input data and (b) output data with $h_1 = 15$ and $h_2 = 97$.

ages, such as Chinese character images, trademark images, etc., can be achieved. Because the brightness contrast of the images in our specific applications is assumed to be apparent, the gray-level information on both sides of each contour need not be coded along with the contour location.[18] The coordinates of the set of fitted boundary segments and a classification map about the whole set of windows, together with the representative gray values, are recorded to achieve image compression. Reconstruction of an image is performed simply by regenerating boundary segments from the stored coordinates and referencing the decoded classification map to fill appropriate gray values into reproduced regions.

The remainder of the paper is organized as follows: we describe the proposed step edge operator in Sec. 2, edge tracking is described in Sec. 3 for use in the application of some types of image data compression, experimental results with brief discussions are presented in Sec. 4 to show the effectiveness and efficiency of this approach, and conclusions are given in Sec. 5.

2 Proposed Edge Detector

2.1 *Principle of Proposed Method*

The proposed edge operator estimates an edge location by a line equation. Based on the principle of 2-D spatial moment preserving, an analytical definition of edge location is then given. This new approach, different from that of Tabatabai and Mitchell,[11] uses nonoverlapping rectangular windows and accepts as input a set of grid squares equal to the size of the window [see Fig. 1(a)]. In this paper, we address only step edges. We assume that the rectangular window is small enough so that a step edge element (briefly called an *edgel* in the sequel) existing in the detection window will lie between two neighboring regions with nearly constant gray levels. The proposed detector generates as output an ideal step edgel defined over a rectangular window with two gray values h_1 and h_2 (assume $h_2 > h_1$). The edgel separates the window into two regions A_1 and A_2 with h_1 and h_2 being the representative gray values for A_1 and A_2, respectively, as shown in Fig. 1(b).

To obtain the areas of A_1 and A_2, as well as their corresponding gray values h_1 and h_2, define the first three spatial moments of empirically obtained data in the $n \times m$ window as

$$m_j = \frac{1}{nm} \sum_{j=1}^{nm} I_j^i \ , \qquad i = 1,2,3 \ ,$$

where I_j is the intensity associated with the j'th grid. Since all of the information of an image comes from the gray values of the pixels, by preserving the first three gray moments in the detection window (this is reasonable because the window is small), we can obtain four equalities as follows:

$$\sum_{j=1}^{2} P_j h_j^i = m_i \ , \qquad i = 1,2,3 \ ,$$

and

$$P_1 + P_2 = 1 \ ,$$

where P_1 and P_2 are the fractions of A_1 and A_2, respectively. Now, we have four equations for four unknown parameters P_1, P_2, h_1, and h_2. These unknowns can be solved by Ref. 19, whereby the areas of A_1 and A_2 can be readily computed as $a_1 = mnP_1$ and $a_2 = mnP_2$.

In this approach, no rotation of the window is needed. Such rotations are found in several previous approaches.[11–13] After checking each of the four corners of the window (see Fig. 2), if the average gray value of the four pixels of the corner is less than the following threshold:

$$T_d = (3 \cdot h_1 + h_2)/4 \ ,$$

then the corner is flagged to be "black"; otherwise, it is flagged to be "white." According to the geometric configurations in the window, twelve distinct ways for the edgel to intersect the window are shown in Fig. 3.

For a window with an edgel, the proposed step edge operator identifies the location of the edgel by two intersection

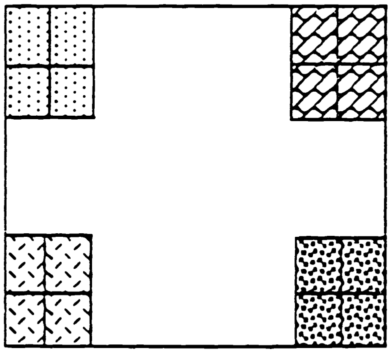


Fig. 2 Checking the four corners of a window. Each corner consists of 4 pixels.

points on the window boundary, say $P=(x_1,y_1)$ and $Q=(x_2,y_2)$. These points determine the line equation of the edgel. Let the line equation be described by

$$Ax+By=C \ , \tag{1}$$

where

$$A = y_1-y_2,$$
$$B = x_2-x_1,$$
$$C = x_2y_1-x_1y_2.$$

Since the location of a window is known at any time, each of the two intersection points P and Q has only one unknown parameter, say p and q, respectively, which determines the edgel position. For example, for the case of Fig. 4, line Eq. (1) becomes

$$(p-q)x+(s_1-s_0)y=s_1p-s_0q \ . \tag{2}$$

We derive next some equations for computing p and q by the mass moment-preserving principle.[20] Let R be a lamina of constant intensity $\rho(x,y)=h>0$ in the xy plane, where the x axis is drawn horizontally pointing to the right, and the y axis is drawn vertically pointing downward. The general rule about the mass moment is

$$M_{ij}(R)=\iint_R x^iy^i\rho(x,y)\ \mathrm{d}x\ \mathrm{d}y \ . \tag{3}$$

In the following, we only investigate the case of Fig. 3(d); the other cases can be derived in the same manner. Note that when the window slides over the entire image, the coordinates of the four corners in it are always kept track of. Let A_1 and A_2 be the two constant subregions separated by edgel **l** with areas a_1 and a_2 and representative gray values h_1 and h_2, respectively. Moreover, assume that edgel **l** with equation $P_ax+P_bY=P_c$ intersects the window at two points, (s_0,p) and (s_1,q) (see Fig. 4), where p and q are two unknown parameters to be solved. By preserving the zeroth-order and the first-order mass moments of A_1 and A_2, we can derive the following lemma to solve the unknown parameters p and q analytically, and so determine the location of the edgel to subpixel accuracy.

Lemma. The unknown parameters p and q satisfy the following relations:

$$\frac{p+q}{2}=\frac{[M_{00}(A)/(s_1-s_0)]+h_2t_0-h_1t_1}{h_2-h_1} \tag{4}$$

and

$$\frac{q-p}{2}=\frac{6\{M_{10}(A)-[(s_1+s_0)/2]M_{00}(A)\}}{(s_1-s_0)^2(h_2-h_1)} \ . \tag{5}$$

Proof. Let ∂T_1 and ∂T_2 be the boundaries (the polygons enclosing the areas) of A_1 and A_2, respectively. Since ∂T_1 and ∂T_2 are two piecewise simple closed curves, by using a modified theorem,[13,20] the total mass moment of the window A is

$$\begin{aligned}
M_{00}(A)&=\iint_A \rho(x,y)\ \mathrm{d}x\ \mathrm{d}y\\
&=\iint_{A_1}\rho(x,y)\ \mathrm{d}x\ \mathrm{d}y+\iint_{A_2}\rho(x,y)\ \mathrm{d}x\ \mathrm{d}y\\
&=h_1\iint_{A_1}\ \mathrm{d}x\ \mathrm{d}y+h_2\iint_{A_2}\ \mathrm{d}x\ \mathrm{d}y\\
&=-h_1\int_{\partial T_1} y\ \mathrm{d}x+\left(-h_2\int_{\partial T_2} y\ \mathrm{d}x\right)\\
&=h_1(s_1-s_0)\left(t_1-\frac{p+q}{2}\right)+h_2(s_0-s_1)\left(t_0-\frac{p+q}{2}\right)\\
&=(s_1-s_0)\left[h_1t_1-h_2t_0+\frac{p+q}{2}(h_2-h_1)\right] \ ,
\end{aligned} \tag{6}$$

where parameters t_0 and t_1 are the y coordinates of the four corners of the detection window (see Fig. 4 for an illustration).

The y mass moment of A is

$$\begin{aligned}
M_{10}(A)&=\iint_A \rho(x,y)\ \mathrm{d}x\ \mathrm{d}y\\
&=\iint_{A_1}x\rho(x,y)\ \mathrm{d}x\ \mathrm{d}y+\iint_{A_2}x\rho(x,y)\ \mathrm{d}x\ \mathrm{d}y\\
&=h_1\iint_{A_1}x\ \mathrm{d}x\ \mathrm{d}y+h_2\iint_{A_2}x\ \mathrm{d}x\ \mathrm{d}y\\
&=M_{10}(A_1)+M_{10}(A_2)\\
&=-h_1\int_{\partial T_1} xy\ \mathrm{d}x+\left(-h_2\int_{\partial T_2} xy\ \mathrm{d}x\right)
\end{aligned}$$

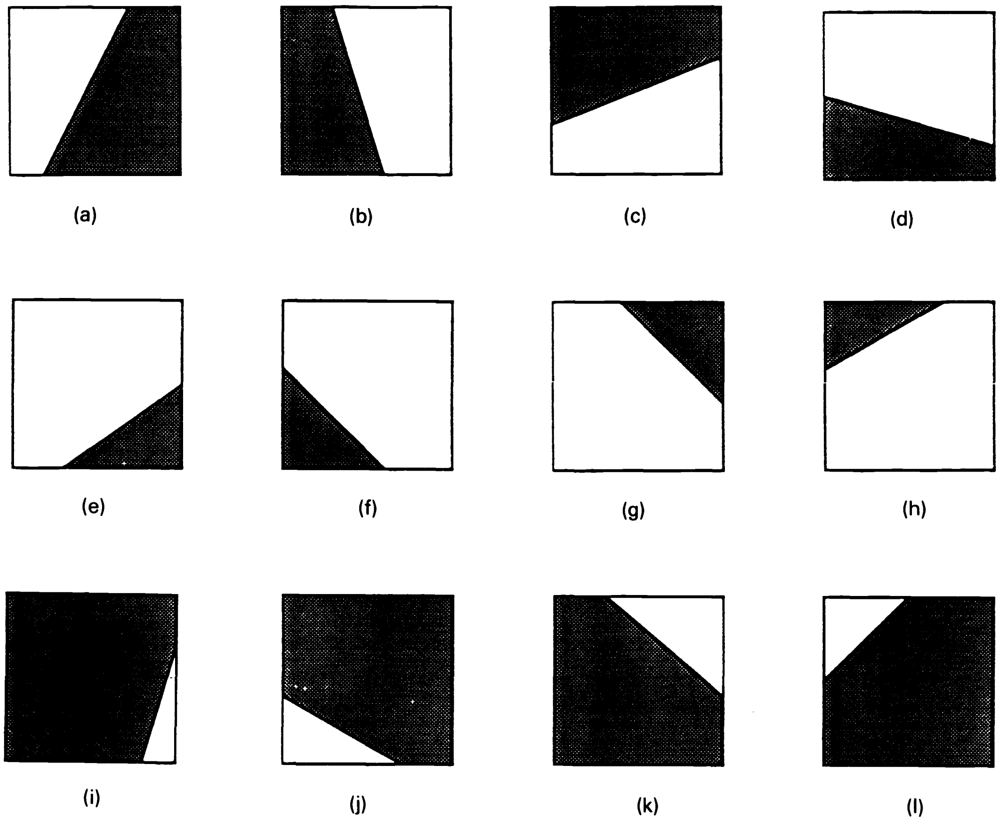


Fig. 3 An edgel may intersect a window in 12 ways.

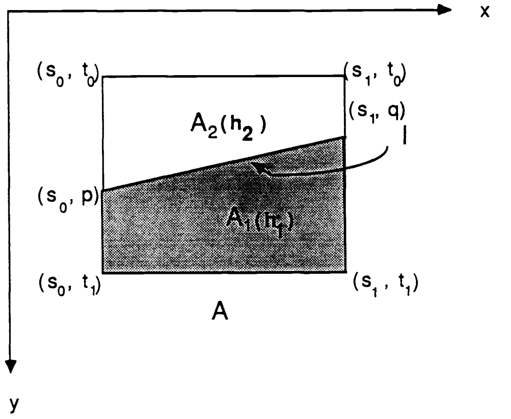


Fig. 4 An edgel intersecting the rectangular window at two points (s_0, p) and (s_1, q).

$$= h_1\left[\frac{s_1+s_0}{2}a_1+\frac{1}{12}(s_1-s_0)^2(p-q)\right]$$

$$+ h_2\left[\frac{s_0+s_1}{2}a_2+\frac{1}{12}(s_1-s_0)^2(p-q)\right]$$

$$= \frac{s_1+s_2}{2}M_{00}(A)+\frac{1}{12}(s_1-s_0)^2(p-q)(h_2-h_1) \ . \qquad (7)$$

For the discrete representation of an image, the mass moment of A can be computed as follows:

$$M_{ij}(A)=\sum\sum_{(x,y)\in A} x^i y^j I(x,y) \ , \qquad (8)$$

where $I(x,y)$ is the observed gray value at location (x,y), which is constant over the grid centered at (x,y). From Eq. (8), we can reformulate Eq. (6) to be

$$\frac{p+q}{2}=\frac{[M_{00}(A)/(s_1-s_0)]+h_2t_0-h_1t_1}{h_2-h_1} \ ,$$

and Eq. (7) to be

$$\frac{q-p}{2}=\frac{6\{M_{10}(A)-[(s_1+s_0)/2]M_{00}(A)\}}{(s_1-s_0)^2(h_2-h_1)} \ ,$$

and the lemma is proven.

Now, we can solve the analytical Eqs. (4) and (5) for p and q. It is found that p and q may be nonintegers, so the edgel location in a window can be estimated to subpixel accuracy. For display purposes, by rounding p and q to the nearest integers, we can obtain the line equation $P_{ai}x+P_{bi}y=P_{ci}$ for edgel $\mathbf{l}_i$ where $P_{ai}=p-q$, $P_{bi}=s_1-s_0$, and $P_{ci}=s_1p-s_0q$.

An edgel-map is introduced in the study as a look-up table to be referenced by the edge-tracking process described in Sec. 3. Each entry in the map corresponds to a window in the original picture and is declared to be a *no-edgel* type initially. By a decision strategy described later, if a window contains an edgel, we apply the appropriate formula and fill the corresponding entry with the edgel type and the computed edgel location.

2.2 *Decision Making about Existence of an Edgel*

One inherent property of edge detection is that the operation requires the examination of several pixels within continuous or overlapping subareas, followed by a decision as to whether an edge is present or not within each subarea. We proceed now to analyze this decision process and suggest certain solutions to the problem. By considering the evidence that noise usually corrupts the edges, the decision making about the existence of an edgel are divided into two stages. In the first stage, the gray-level distribution of the pixels in the window is analyzed. Two parameters T_0 and T_1 are preselected: T_0 is used to determine the existence of unimodal windows (i.e., constant regions), whereas T_1 is used to determine the existence of bimodal windows (i.e., two-valued regions). The mechanism of this simple test is described in the following.

Divide the gray-level range [0, 255] into 16 intervals, say I_j, $j=1, 2, ..., 16$, and let O_j represent the number of pixels with gray levels falling in interval I_j. If the tested window is bimodal, the values of O_j will become diverse and small. On the other hand, if the tested window is unimodal, the values of O_j will concentrate on several larger values. Compute O_j for each window, where $j=1, 2, ..., 16$. And compute the index difference

$$\text{Diff\_indx} = \text{Max\_indx-Min\_indx} \ , \tag{9}$$

where Max_indx and Min_indx are the index numbers of $\max_i\{O_i\}$ and $\min_i\{O_i\}$ for $i=1, 2, ..., 16$, respectively. In the experiments, the critical values of the two parameters T_0 and T_1 are assigned to be 1 and 3, respectively. If Diff_indx is less than or equal to T_0, the window is determined to be uniform and contain no edgel; if Diff_indx is larger than or equal to T_1, the window is determined to contain an edgel; and if Diff_indx is greater than T_0 and less than T_1, the window is regarded as an "ambiguous" region. This simple parametric test can speed up the decision process of edgel existence by eliminating those windows whose gray values are clearly unimodal or bimodal distributions and reduce the unnecessary computation time significantly.

If it fails to make a decision in the first stage, a simple test about the gray level variations using Student's t-distribution is introduced in the second stage. If an edgel exists in a window, let the constant subregions separated by the edgel be denoted as A_1 and A_2, respectively. Among them, one is "darker" with respect to the other. To implement this test, we assume that the picture is a 2-D discrete random field that is a collection of random samples. Each random sample is supposed to have a normal distribution and represents the gray value of each pixel. We also assume that the random samples are stochastically independent with their variances being unknown, but equal. Under these assumptions, we can regard subregions A_1 and A_2 as two independent normal distributions, say $n(\mu_1, \sigma^2)$, and $n(\mu_2, \sigma^2)$, where μ_1 and μ_2 are two unknown means of A_1 and A_2, respectively. For convenience, it is hoped that μ_2 is larger than μ_1 (i.e., subregion A_2 is "brighter" than A_1) and a hypothesis testing for $\mu_2-\mu_1$ may be obtained as follows.

Let $X_1, X_2, ..., X_{n1}$ and $Y_1, Y_2, ..., Y_{n2}$ denote, respectively, the independent random samples from the two independent distributions having, respectively, the probability density functions $n(\mu_1, \sigma^2)$ and $n(\mu_2, \sigma^2)$. Denote the means of the samples by $\bar{X}$ and $\bar{Y}$, and the variances of the samples by S_1^2 and S_2^2, respectively. According to the stochastic independence property described in Ref. 21, $\bar{X}$ and $\bar{Y}$ are normally and stochastically independently distributed with means μ_1 and μ_2 and variances $\sigma^2/n1$ and $\sigma^2/n2$, respectively. Thus, the difference $\bar{Y}-\bar{X}$ is normally distributed with mean $\mu_2-\mu_1$ and variance $(\sigma^2/n1)+(\sigma^2/n2)$. Since variance σ^2 is unknown, an unbiased estimation of σ^2, denoted by S^2, is needed, which can be formulated as

$$S^2=\frac{n1*S_1^2+n2*S_2^2}{n1+n2-2}$$

$$=\frac{\sum_{i=1}^{n1}(X_i-\bar{X})^2+\sum_{i=2}^{n2}(Y_i-\bar{Y})^2}{n1+n2-2} \ ,$$

and the estimated standard derivation $\hat{S}_D$ of the difference between $\bar{X}$ and $\bar{Y}$ is then derived to be

$$\hat{S}_D=\left(\frac{S^2}{n1}+\frac{S^2}{n2}\right)^{1/2}=S\left(\frac{1}{n1}+\frac{1}{n2}\right)^{1/2} \ . \tag{10}$$

Now consider the hypothesis testing

$$H_0{:}\mu_2-\mu_1<T_2 \ ,$$

against

$$H_1{:}\mu_2-\mu_1\geqslant T_2 \ ,$$

where T_2 is a preselected tolerance upper bound of gray-level similarity such that two pixels are declared to be in the same region if their absolute gray-value difference is equal to or less than T_2. In our experiments, T_2 is set to be 15. By Eq. (10), the decision rule

$$\xi=\frac{(\bar{Y}-\bar{X})-T_2}{\hat{S}_D} \tag{11}$$

has a Student's t-distribution with $(n1+n2-2)$ degrees of freedom. The critical t value at the 0.5% significance level in our experiments for $(n1+n2-2)$ degrees of freedom can be taken from the tables listed in the appendix of Ref. 21. If the observed value of ξ exceeds this postulated critical t value, then H_0 is rejected and we confirm that the tested window is bimodal (i.e., we confirm that the window contains an edgel). Otherwise, the tested window is regarded as a region with no edgel.

3 Edge Tracking and Application to Image Data Compression

3.1 *Edge-Tracking Principle*

The proposed edge-tracking method tends to extend an edgel along its orientation. In connecting the next boundary edgel

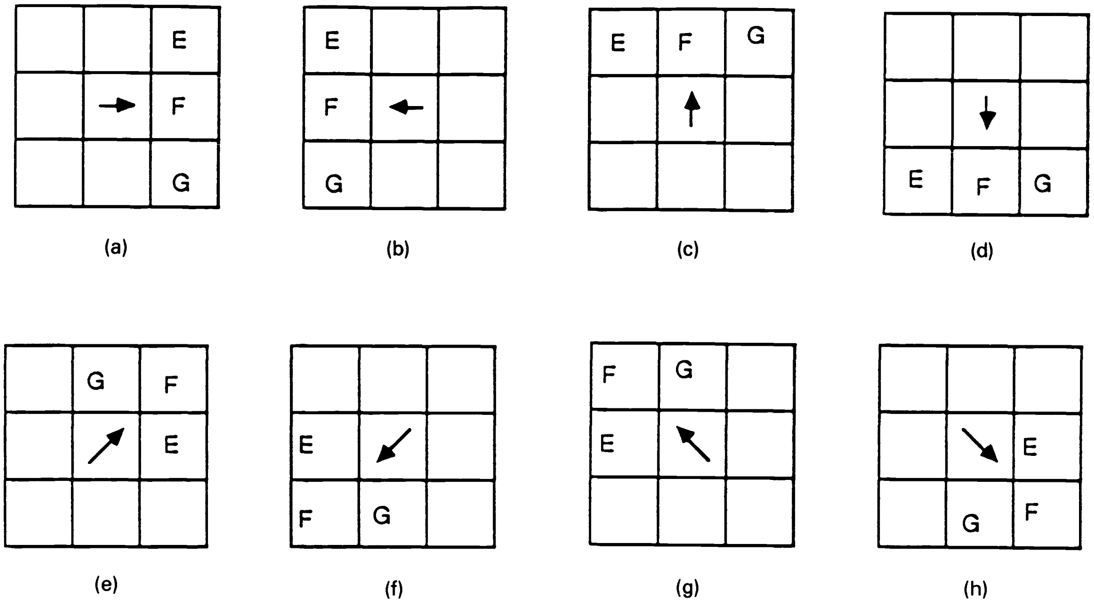


Fig. 5 Eight different search areas for the next boundary edgel. The arrow represents the direction to which the direction of the current edgel is similar. The search areas are denoted by E, F, and G.

from the current edgel **l**, only three of its eight neighbors, called search areas, are examined. Although there are 12 distinct ways in which an edgel can exist in a window, we can further classify them into eight edgel orientations, and eight possible search areas are thus derived, as shown in Fig. 5. From the edgel map obtained from the edge detector, an unfitted edgel is selected as the "start edgel." An object boundary may be traced by following the successive edgel with an orientation that agrees with the current checking orientation and with the collinearity measurement $d_{\max}$ (defined later) that does not exceed some preset threshold ε_1. A temporarily fitted segment l is obtained simply by connecting the start edgel and the current successive edgel. The Euclidean distance is used to measure the collinearity:

$$d_{\max} = \max_{i=2}^{2m-1} \frac{|P_a x_i + P_b y_i - P_c|}{(P_a^2 + P_b^2)^{1/2}} , \qquad (12)$$

where (P_a, P_b, P_c) are the line parameters of the temporarily fitted segment l and (x_i, y_i) are coordinates of the end points of the edgels that will be approximated by l. This tracking process is repeated until an edgel with no successor or until a previously fitted edgel is reached. Hence, a newly fitted boundary segment can be formed by connecting the start edgel and the end edgel to approximate a group of ordered edgels just passed through. After all of the detected edgels are traced, the measure of fitting error ε_{angl} that determines the fitting quality of the whole tracking process is defined as

$$\varepsilon_{angl} = \overline{\Theta} + \sigma_{angl} , \qquad (13)$$

where $\overline{\Theta}$ and σ_{angl} are the mean and the standard deviation of the intersection angles, respectively, between the fitting line segments and the corresponding edgels. If ε_{angl} is no less than some preset limit ε_2, then decrease ε_1 by a fixed value and restart the tracking process; otherwise, the tracking stage is terminated, resulting in an acceptable compact representation for the object contour. Formally, the algorithm for edge tracking is given below.

Algorithm 1. Edge tracking.

```
Input.    An edgel map.
Output.   A set of fitted boundary segments, SEG_SET.
Step 1.   Create an empty segment set, SEG_SET.
Step 2.   Create an empty queue SEG.
Step 3.   Choose an unfitted edgel from the edgel map as the current edgel. Push it into SEG.
Step 4.   Examine the possible search areas of the current edgel to determine a successive edgel $\mathbf{l}_{js}$ that is not fitted yet.
Step 5.   If no successive edgel exists, go to Step 8.
Step 6.   Connect the first edgel in SEG and the successive edgel $\mathbf{l}_{js}$ to form a temporary fitted segment $l_t$.
Step 7.   Check the collinearity of the edgels in SEG and $\mathbf{l}_{js}$. For each end point $P$ of any edgel in SEG, compute the Euclidean distance from $P$ to $l_t$ and choose the maximum, say $d_{\max}$. If $d_{\max} \leq \varepsilon_1$, push the successive edgel $\mathbf{l}_{js}$ into SEG, let $l_t$ be the current edgel, and go to Step 4.
Step 8.   If SEG contains only one edgel, remove it from SEG and go to step 2; otherwise, connect the first and the last edgels in SEG to form a new fitted segment $l$ and push it into SEG_SET.
Step 9.   Compute the intersection angle between each edgel in SEG and $l$.
Step 10.  If any edgel remains unfitted, go to Step 2.
Step 11.  Compute the mean $\overline{\Theta}$ and standard deviation $\sigma_{angl}$ of the intersection angles between
```

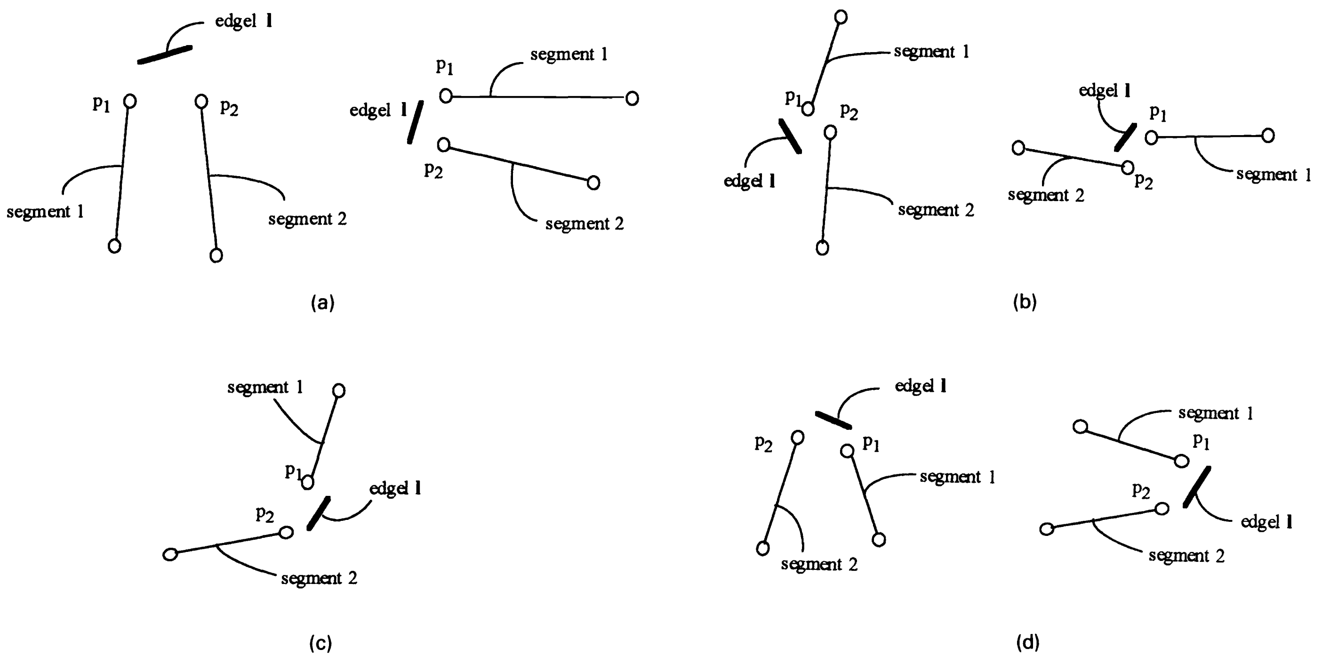


Fig. 6 A gap may exist between two fitted segments in four ways.

the fitted segments and the corresponding edgels.

Step 12. Measure the fitting quality ε_{angl} of all the fitted segments in SEG_SET. If $\varepsilon_{angl} < \varepsilon_2$, output all of the fitted segments in G and exit; otherwise, reinitialize SEG_SET as empty and adjust ε_1 to a smaller value, say $\varepsilon_1 - \zeta$. Go to Step 1 for the next iteration.

Since some edgels may not be included into any fitted segments during the tracking process, these untouched edgels correspond to the gaps that significantly reduce the visual acceptability. To solve this problem, each of the gaps can be bridged simply by examining 24 neighbors of the corresponding untouched edgel $\mathbf{l}_j$, and picking two of the fitted segments, say τ_1 and τ_2, nearest to $\mathbf{l}_j$. The geometric relations of τ_1, τ_2, and $\mathbf{l}_j$ can be classified into four classes, as shown in Fig. 6. If τ_1 and τ_2 are nearly parallel [see Figs. 6(a) and 6(b)], choose two end points P_1 and P_2 nearest to edgel $\mathbf{l}_j$ and reassign the coordinates of P_1 and P_2 to their midpoint Q [see Figs. 7(a) and 7(b)]; otherwise, as seen in Figs. 6(c) and 6(d), stretch τ_1 and τ_2 along their individual orientations to obtain an intersection point Q as the new end point, and properly reassign Q to the end points of τ_1 and τ_2, respectively, such that τ_1 and τ_2 have a common end point Q [see Figs. 7(c) and 7(d)]. An advantage of using fixed end points in the approximation under the preset maximum Euclidean distance criteria is that, in practical applications, slight deviations from the minimum error norm are generally acceptable and usually preferred because they can result in considerable savings of computational time without sacrificing the accuracy of the result.

3.2 *Application to Image Data Compression*

Recently developed image data compression techniques, such as segmentation-based coding,[16] directional decomposition-based coding,[21] etc., aim at reducing storage space by the use of explicit representations of geometric structures. However, they usually require substantial amounts of computing time. This restricts their applications, especially when fast compression/decompression is needed. The set of fitted line segments, resulting from the edge-tracking process, can be used to represent shape boundaries that obviously are useful for data reduction or compression, because line segments, in general, can be represented by their equation coefficients using a much smaller number of bits.

Once object contours are represented by a set of fitted line segments, the entire image is then separated into distinct constant regions. Since the brightness contrasts of images are assumed to be prominent in our application, each of these segmented constant regions can be colored simply by a single gray value. A blockwise connected component labeling method is employed to distinguish these regions by sequential labels that are recorded in a resulting classification map. Instead of coding gray levels of contour pixels as proposed by Carlsson,[18] image compression can be achieved by recording the coordinates of these boundary segments and using fewer bits to store the classification map and the label table. However, the method of constructing the classification map and the label table still needs improvement because of the following two reasons: (1) the distinct labels may not be indexed in sequential numbers and (2) there may be two or more label indices in the label table whose corresponding average gray values are very much close.

For reason (1), we rearrange the distinct labels in sequential numbers. This reduces the size of the label table. For reason (2), a threshold T_4 is selected to allow a tolerance in judging the similarity among the calculated average gray values; the gray values of any two labels are determined to be of the same class if their difference is less than T_4. In our

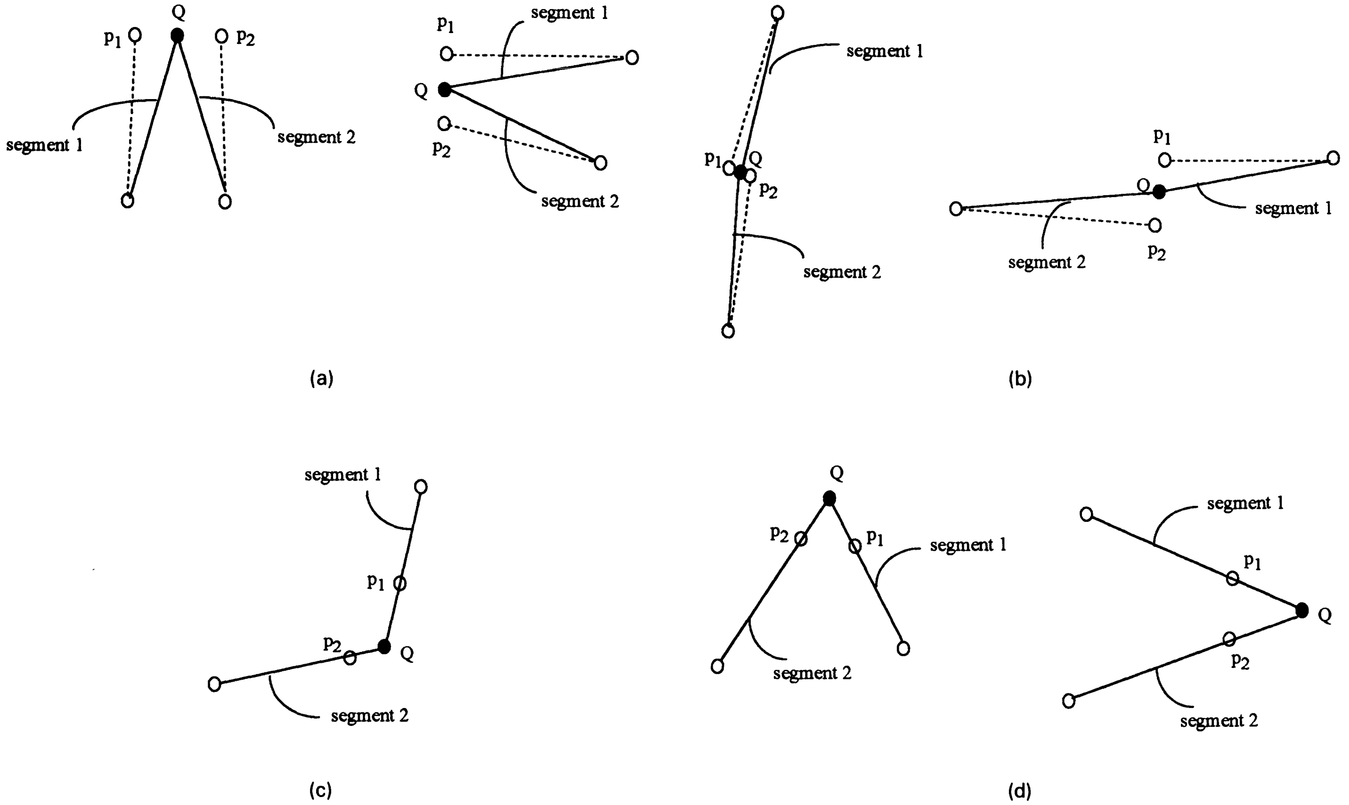


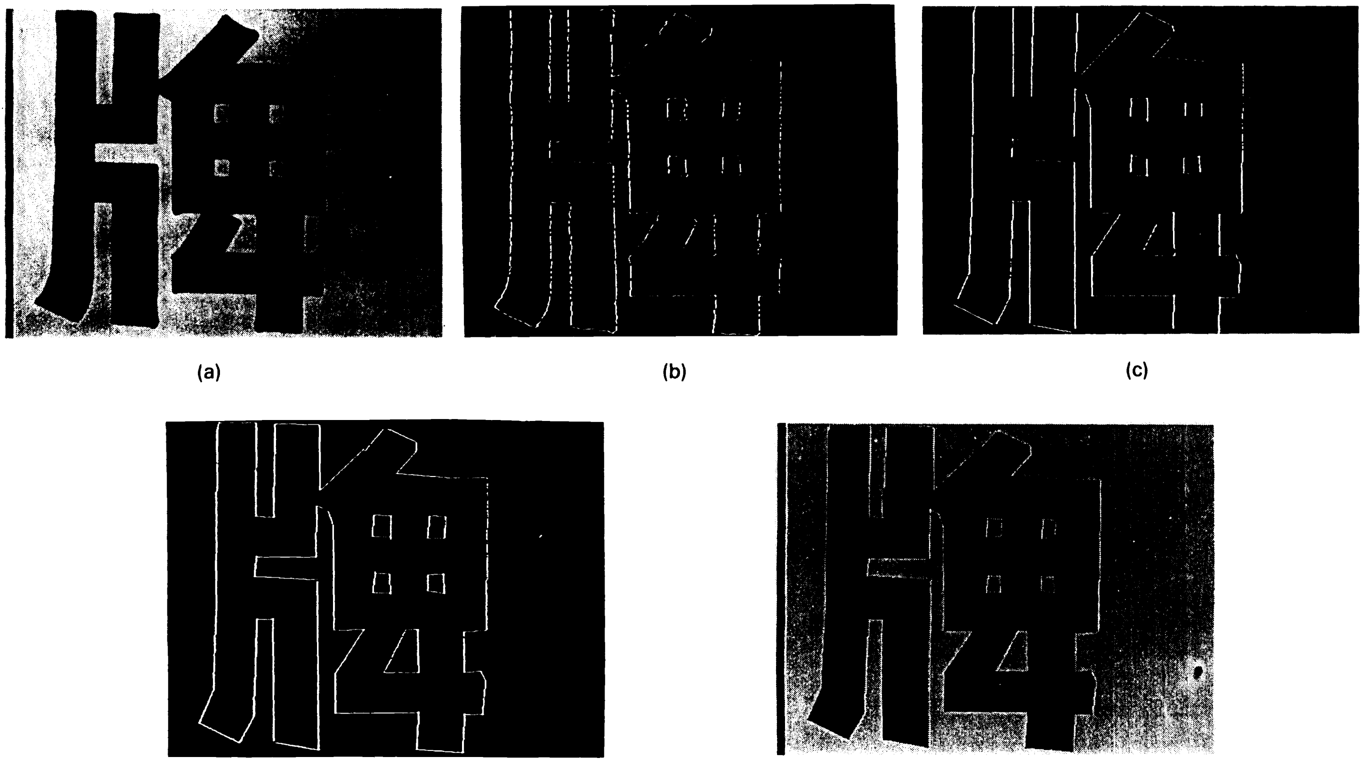
Fig. 7 Two ways of gap filling: (a) and (b) are for two nearly parallel fitted segments and (c) and (d) for two nonparallel fitted segments.

experiments, T_4 is set to be 25. A detailed description of the modified blockwise connected component labeling algorithm is described below.

Algorithm 2. Connected component labeling.

```
Input.     An edgel map.
Output.    A classification map C_MAP and a label table T_LAB.
Step 1.    Create a classification map C_MAP, with each entry corresponding to a window in the original picture.
Step 2.    An entry of C_MAP is set to 1 if the corresponding rectangular window contains an edgel; otherwise, it is set to 0.
Step 3.    Examine each entry of C_MAP in a row-majored top-down manner.
Step 4.    Search for the first nonzero entry. If it is not found and the map has been scanned over, go to Step 8.
Step 5.    Find the set ADJ of four adjacent labeled entries.
Step 6.    If ADJ is empty, assign a new label to this entry; otherwise, choose the minimum label in ADJ as the new label and record that the other labels in ADJ and the minimum label are equivalent, i.e., they represent an identical class.
Step 7.    Go to Step 4 to determine the label for the next entry.
Step 8.    Relabel entries based on the recorded equivalent labels, such that a class is represented by only a label.
Step 9.    If any label change occurs, do the process from Step 4 to Step 8 in a row-majored bottom-up manner.
Step 10.   Number the distinct labels in sequence and then modify the corresponding entries in C_MAP.
Step 11.   Construct a label table T_LAB with its size equal to the number of labels in C_MAP.
Step 12.   Group the entries having an identical label in C_MAP and compute the average gray value G_AVE of the corresponding uniform windows. Record each G_AVE value and the corresponding label number on T_LAB.
Step 13.   If there exist two or more labels in T_LAB with the difference of the corresponding average gray values being less than $T_4$, mark these labels as being equivalent and record them as belonging to an identical class; otherwise, output C_MAP and T_LAB with the label numbers in sequence, and exit.
Step 14.   Relabel the entries of C_MAP based on the equivalence classes. Go to Step 12.
```

To measure the compression ratio (CR), the following notations are employed:



(a) (b) (c)

(d) (e)

Fig. 8 The result of applying the proposed step edge detector to a Chinese character image: (a) original input image; (b) output image with 6 × 7 nonoverlapping squares as output areas and with detected short edge element drawn in each square; (c) output image resulting from edgel tracking and line segment approximation, in which $\varepsilon_1 = 8.8$, $\bar{\theta} = 13.5$ and $\sigma_{angl} = 9.2$; (d) output image identical to (c) except that the gaps around the corners are bridged; and (e) reconstructed image with CR = 248.83.

ROW = number of rows in the image

COL = number of columns in the image

leng = length of the rectangular window

wid = width of the rectangular window

no = number of elements contained in the label table

seg = number of fitted boundary segments

nbit = number of bits required to represent a pixel.

Now, we briefly summarize the total number of bits required to code an image of 512 × 512 pixels. The bits come from the following sources.

1. Size of the classification map, map_size:

$$\text{map\_size} = \left(\frac{\text{ROW}}{\text{leng}} \times \frac{\text{COL}}{\text{wid}}\right) \times \lceil \log_2(\text{no}+1) \rceil \ .$$

2. Size of the label table, lab_size:

$$\text{lab\_size} = \text{nbit} \times \lceil \log_2(\text{no}+1) \rceil \ .$$

3. Size of coordinates for two end points of an edgel, coor_size:

$$\text{coor\_size} = 2 \times (\lceil \log_2 \text{ROW} \rceil + \lceil \log_2 \text{COL} \rceil) \ .$$

Let the gray level of each pixel be coded with 8 bits (denoted as nbit), then the CR is

$$\text{CR} = \frac{\text{ROW} \times \text{COL} \times \text{nbit}}{\text{map\_size} + \text{lab\_size} + \text{seg} \times \text{coor\_size}} \ .$$

To summarize, the CR depends on the content of an image. In our case, it is the amount of fitted boundary segments that mainly determines the CR. Therefore, the preset limit of the maximum Euclidean distance and the size of the rectangular window chosen in the edge detection process can be varied to obtain various CR.

4 Experimental Results

The proposed approach has been tested on several images using a PC/AT 386. Some results are shown in Figs. 8 and 9. The step edge detector is applied to digital images taken from an image capture board with resolution 512 × 512 × 8 bits. Given a window size, an image is divided into a set of contiguous nonoverlapping quadrilaterals. For the purpose of display, the pixels drawn by these edge line equations are set to a predetermined white intensity value, say 255.

Figures 8(a) and 9(a) are the original images of a Chinese character and a trademark, respectively. The edge detection results with fixed window sizes of 6 × 7 and 7 × 8 are given in Figs. 8(b) and 9(b), respectively. After all of the edgels in

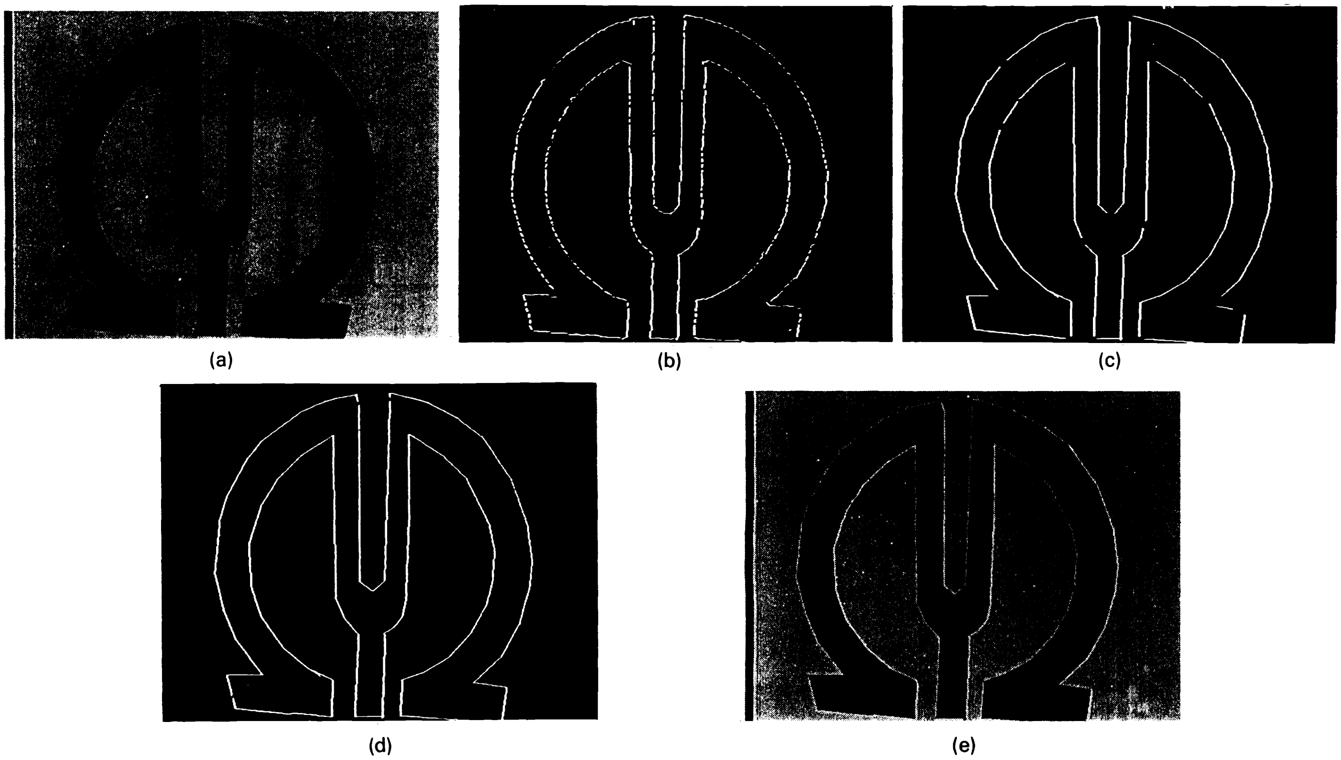


Fig. 9 The result of applying the proposed step edge detector to a trademark image with two colors: (a) original input image; (b) output image with 7 × 8 nonoverlapping squares as output areas and detected short edge element drawn in each square; (c) output image resulting from edgel tracking and line segment approximation, in which $\varepsilon_1 = 4.3$, $\bar{\theta} = 9.7$, and $\sigma_{angl} = 8.9$; (d) output image identical to (c) except that the gaps around the corners are bridged; and (e) reconstructed image with CR = 269.5.

an image are identified, the available line parameters of the edgels together with the edgel map are then exploited in the stage of edge tracking. At the end of this tracking process, a more compact representation of the contours of regions or objects is thus constructed. Figures 8(c) and 9(c) show the set of fitted boundary segments in which some gaps occur around the corners. An improving step used to bridge these gaps is then performed, and the final results are displayed in Figs. 8(d) and 9(d).

After performing the edgel-tracking process, the information extracted from an original image includes: (1) the coordinates of the set of available fitted boundary segments and (2) a classification map and a label table about the set of constant windows. Such information forms a simple and perspicuous representation and can be used for image reconstruction. Figures 8(e) and 9(e) exhibit the reconstructed images with CRs 248.83 and 269.5, respectively. Figure 10 shows the experimental results of a third image with a CR of 324.1. For various choices of the preset values of the collinearity controlling threshold ε_1 and the window size, the resulting CRs together with the mean and the standard deviation of intersection angles are listed in Tables 1 through 3. Tables 1 through 3 show that the larger the window size, the larger the CR that can be obtained, under the same value of ε_1. Besides, when the value of ε_1 grows larger (i.e., enduring more fitting errors in the edge-tracking process), the mean value of the intersection angles, as well as the CR in general, are increased. In most cases, the fitting quality is highly dependent on the window size, and the influences resulting from various values of ε_1 are less prominent.

It takes about 2 to 3 min to process an image, the amounts of time being dependent on the complexity of the processed image contents. Most of the time (more than half) was spent in edge detection and the *t*-distribution test if applied. If faster computers such as workstations are used, the processing time can be greatly reduced.

5 Conclusions

The moment-preserving principle is a useful technique in image processing to keep as much information of the input image as possible. A new method for step edge detection has been proposed. This method can be applied directly to an observed picture. For each rectangular window, the proposed detector analytically computes a line equation for the location of an edgel to a good accuracy level, if the decision strategy claims that an edgel is contained in it, by the use of the gray and mass moment-preserving principles, as well as various geometric relations in the detection window.

The derived edgel equations are useful as the input data for the work of boundary tracking. Based on a linearity measure using Euclidean distances and intersection angle variations, piecewise line segment approximations with fixed end points are employed to represent the contours of regions or objects in a compact way, and data reduction is thus achieved. The original picture can then be segmented into a set of constant regions. Using the modified blockwise connected

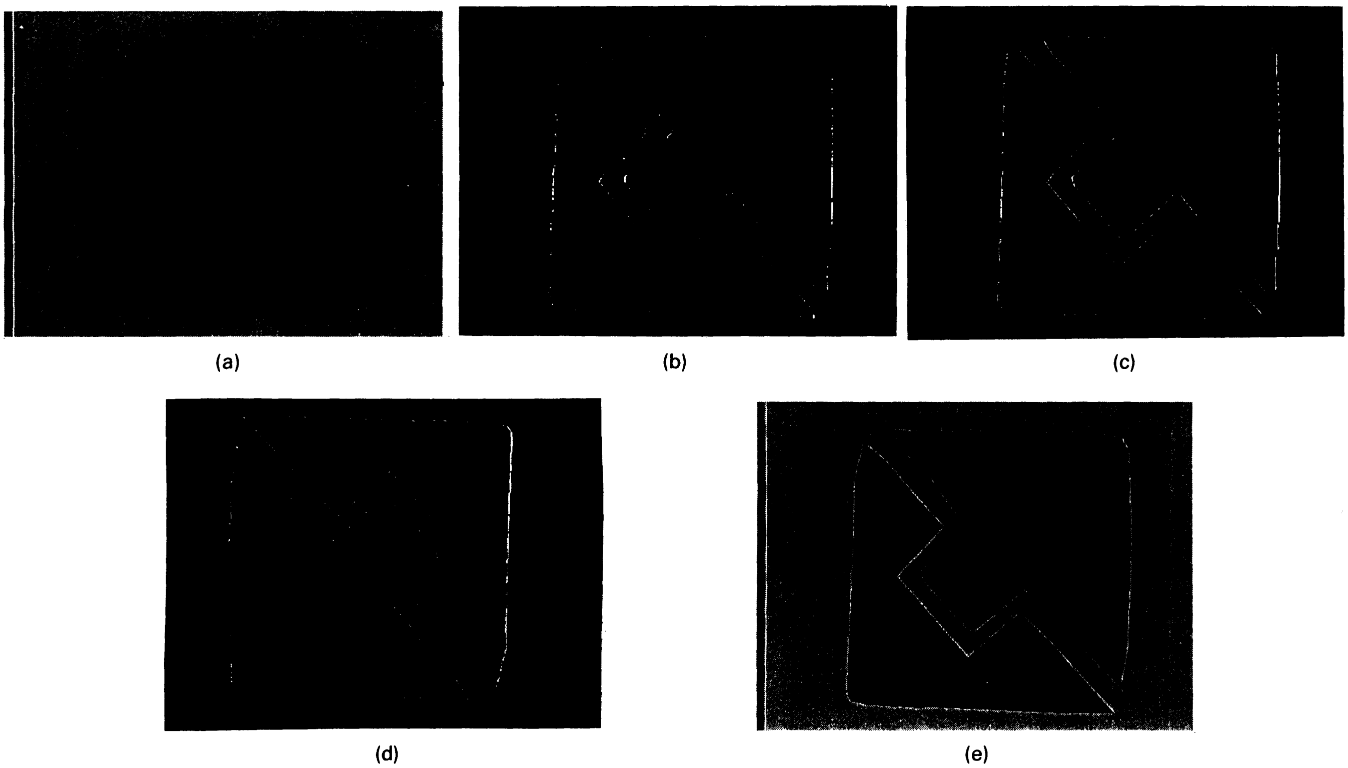


Fig. 10 The result of applying the proposed step edge detector to a trademark image with three colors: (a) original input image; (b) output image with 6 × 7 nonoverlapping rectangles as output areas with a detected short edge element drawn in each area; (c) output image resulting from edgel tracking and line segment approximation, in which $\varepsilon_1=3.6$, $\bar{\theta}=10.1$, and $\sigma_{angl}=11.9$; (d) output image identical to (c) except that the gaps around the corners are bridged; (e) reconstructed image with CR = 324.1.

Table 1 Evaluation of the CR and the corresponding $\bar{\theta}$ and σ_{angl} values with various preset limits ε_1 for the image of Fig. 9 with a 6 × 6 window.

ε_1	3.5	4.6	5.5	6.6	7.5	8.3	9.5	10.8	15.6
$\bar{\theta}$	9.5	11.0	11.1	11.0	11.5	11.4	12.4	13.9	15.1
σ_{angl}	5.4	6.38	6.33	6.41	6.65	5.7	8.2	9.94	9.52
CR	199.4	203.4	204.0	207.4	211.9	211.1	216.0	216.1	223.9

Table 2 Evaluation of the CR and the corresponding $\bar{\theta}$ and σ_{angl} values with various preset limits ε_1 for the image of Fig. 9 with a 6 × 7 window.

ε_1	3.5	4.6	5.5	6.6	7.5	8.3	9.5	10.8	15.6
$\bar{\theta}$	9.53	11.0	11.3	12.4	12.0	12.4	13.0	13.2	14.1
σ_{angl}	4.7	5.84	5.75	7.03	6.67	7.0	7.74	7.7	9.05
CR	217.1	221.0	222.7	228.3	235.0	238.2	238.9	240.0	246.0

Table 3 Evaluation of the CR and corresponding $\bar{\theta}$ and σ_{angl} values with various preset limits ε_1 for the image of Fig. 9 with a 8 × 8 window.

ε_1	3.5	4.6	5.5	6.6	7.5	8.3	9.5	10.8	15.6
$\bar{\theta}$	8.3	9.3	10.2	10.2	10.6	11.5	11.8	11.9	13.4
σ_{angl}	4.93	5.2	6.0	6.0	6.05	6.36	6.33	6.0	6.02
CR	270.3	268.0	273.1	275.7	277.0	282.8	281.3	283.5	296.0

component labeling scheme, each region can be identified just by a label. Using the compact representations of the object contours, we have shown examples of compression with CRs larger than 195.5 for pictures with gray-level resolutions of 8 bits/pixel.

The proposed edge detector does not need a thinning process. Although it is sensitive to noise similar to most image processing techniques using moments, the process of bilevel gray moment-preserving thresholding can remove the noise of low intensities without preprocessing and postprocessing. Besides, no iteration is required and only algebraic computations are involved in processing the entire image.

Compared with other similar approaches using the moment-preserving principle (like the Tabatabai and Mitchell approach[11]), the proposed approach of edge detection has at least the following advantages. (1) Window overlapping, which is encountered in the other approaches using overlapping circular windows, is avoided, and this speeds up the edge-detection process applied to the entire image. (2) The detected edges will not overlap or be disconnected by gaps since the rectangular windows are contiguous and nonoverlapping. Note that this is not the case for the other approaches using overlapping circular windows that do not completely cover the entire image, resulting in more difficulty in the postprocessing steps of edge tracking and linking.

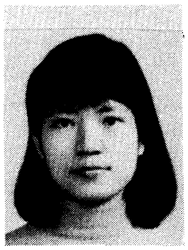
The acquired range of the CRs has been shown to be satisfactory for pictures with clear regions. For textured pictures like natural scenes, the performance of the proposed method will be worse, but the method is still applicable. The CRs are in the range of 20 to 30, as revealed by a similar study by Cheng and Tsai.[27]

Acknowledgment

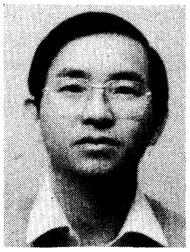
This work was supported partially by the National Science Council, Republic of China, under contract number NSC81-04W-EOO9-17.

References

1. A. Rosenfeld and A. C. Kak, *Digital Picture Processing,* Vol. 2, Academic Press, New York (1982).
2. R. Nevatia and K. R. Babu, "Linear feature extraction and description," *Comput. Vis. Graph. Image Process.* **13,** 257–269 (1980).
3. W. Frei and C. C. Chen, "Fast boundary detection: a generalization and a new algorithm," *IEEE Trans. Comput.* **C-26,** 988–998 (Oct. 1977).
4. M. Suk and S. Hong, "An edge extraction technique for noisy images," *Comput. Vis. Graph. Image Process.* **25,** 24–45 (1984).
5. A. Kundu and S. K. Mitra, "A new algorithm for image edge extraction using a statistical classifier approach," *IEEE Trans. Patt. Anal. Mach. Intell.* **PAMI-9,** 569–577 (July 1987).
6. J. S. Huang and D. H. Tseng, "Statistical theory of edge detection," *Comput. Vis. Graph. Image Process.* **43,** 337–346 (1988).
7. R. M. Haralick, "Digital step edges from zero crossing of second directional derivatives," *IEEE Trans. Patt. Anal. Mach. Intell.* **PAMI-6,** 58–68 (Jan. 1984).
8. M. H. Hueckel, "An operator which locates edges in digitized pictures," *J. Assoc. Comput. Math.* **18,** 113–125 (1971).
9. R. Hartley, "A gaussian-weighted multiresolution edge detector," *Comput. Vis. Graph. Image Process.* **30,** 70–83 (1985).
10. V. S. Nalwa and T. O. Binford, "On detecting edges," *IEEE Trans. Patt. Anal. Mach. Intell.* **PAMI-8,** 699–714 (Nov. 1986).
11. A. J. Tabatabai and O. R. Mitchell, "Edge location to subpixel values in digital imagery," *IEEE Trans. Patt. Anal. Machine Intell.* **PAMI-6,** 188–201 (March 1984).
12. L. H. Chen and W. H. Tasi, "Moment-preserving curve detection," *IEEE Trans. Syst. Man Cybern.* **18,** 148–158 (Jan. 1988).
13. S. T. Liu and W. H. Tsai, "Corner detection and pattern clustering by moment-preserving techniques for image analysis," Ph.D. Dissertation, National Chiao Tung University (June 1989).
14. P. D. Hyde and L. S. Davis, "Subpixel edge estimation," *Patt. Recog.* **16,** 413–420 (1983).
15. D. N. Graham, "Image transmission by two-dimensional contour coding," *Proc. IEEE* **55,** 336–346 (March 1967).
16. I. Kunt, A. Ikonomopoulos, and M. Kocher, "Second generation image coding techniques," *Proc. IEEE* **73,** 549–574 (April 1985).
17. C. H. Chen, "Note on a modified gradient method for image analysis," *Patt. Recog.* **10,** 261–264 (1973).
18. S. Carlsson, "Sketch based coding of gray level images," *Sig. Process.* **15,** 57–83 (1988).
19. W. H. Tsai, "Moment-preserving thresholding: a new approach," *Comput. Vis. Graph. Image Process.* **29,** 377–393 (1985).
20. J. Stewart, *Calculus,* Brooks/cole, Monterey, CA (1987).
21. R. V. Hogg and A. T. Craig, *Introduction to Mathematical Statistics,* 4th ed., Sec. 4.7, Macmillan Publishing Co., Inc. New York (1978).
22. A. Ikonomopoulos and M. Kunt, "High compression image coding via directional filtering," *Sig. Process.* **8**(2), 179–203 (Apr. 1985).



Hua-Sheng Hsu received the BS and MS degrees in computer science and information engineering from National Chiao Tung University, Hsinchu, Taiwan, in 1988 and 1990, respectively. From July 1988 to June 1990, she worked as a research assistant in the Computer Vision Laboratory of the Department of Computer and Information Science at National Chiao Tung University. In September 1990, she joined the Computer and Communication Research Laboratories of the Industrial Technology Research Institute, Hsinchu, Taiwan, where she works as a design engineer/programmer for color image processing and compression, special visual effects, and synchronization in controlling media devices, etc. Her current research interests include pattern recognition, video compression, media communication, and object-oriented databases.



Wen-Hsiang Tsai received the BS degree from National Taiwan University in 1973, the MS degree from Brown University in 1977, and his PhD degree from Purdue University in 1979, all in electrical engineering. Since November 1979, he has been on the faculty of the Institute of Computer Science and Information Engineering at National Chiao Tung University, Hsinchu, Taiwan. He joined the Department of Computer and Information Science at National Chiao Tung University in August 1984, acted as the head of the department from 1984 to 1988, and is currently a professor there. His current research interests include computer vision, image processing, pattern recognition, and neural networks. Dr. Tsai is an associate editor for several journals and is the recipient of many research awards. He was the winner of the 13th Annual Best Paper Award of the U.S. Pattern Recognition Society. Dr. Tsai has published more than sixty papers in international journals. Tsai is a senior member of the IEEE, and a member of the Chinese Image Processing and Pattern Recognition Society, the Computing Linguistics Society of the Republic of China, and the Medical Engineering Society of the Republic of China.