Thresholding in Edge Detection: A Statistical Approach

Rishi R. Rakesh, Probal Chaudhuri, and C. A. Murthy

***Abstract*—Many edge detectors are available in image processing literature where the choices of input parameters are to be made by the user. Most of the time, such choices are made on an *ad-hoc* basis. In this article, an edge detector is proposed where thresholding is performed using statistical principles. Local standardization of thresholds for each individual pixel (*local thresholding*), which depends upon the statistical variability of the gradient vector at that pixel, is done. Such a standardized statistic based on the gradient vector at each pixel is used to determine the eligibility of the pixel to be an edge pixel. The results obtained from the proposed method are found to be comparable to those from many well-known edge detectors. However, the values of the input parameters providing the appreciable results in the proposed detector are found to be more stable than other edge detectors and possess statistical interpretation.**

***Index Terms*—Fixed and adaptive choices for parameters, local standardization for thresholding, nonmaxima suppression, smoothing techniques, thresholding with hysteresis.**

I. Introduction

An EDGE IS characterized by an abrupt change in intensity indicating the boundary between two regions in an image. It is a local property of an individual pixel and is calculated from the image function in a neighborhood of the pixel. Edge detection is a fundamental operation in computer vision and image processing. It concerns the detection of significant variations of a gray level image. The output of this operation is mainly used in higher-level visual processing like three-dimensional (3-D) reconstruction, stereo motion analysis, recognition, scene segmentation, image compression, etc. Hence, it is important for a detector to be efficient and reliable.

Edge detection has been an active research area for more than 35 years [31]. Several reviews of work on edge-detection are available in literature [1], [13], [28], [38], [40]. Surface fitting approach for edge detection is adopted by several authors [26], [18], [34], [8]. Bergholm's [2] edge detector applies a concept of edge focusing to find significant edges. Detectors based on some optimality criteria are developed in [4], [32], and [33]. Use of statistical procedures are illustrated in [14], [11], and [30]. Other approaches on edge detection include the use of genetic algorithms [3], [5], neural networks [36], the Bayesian approach [21], and residual analysis-based techniques [9], [35]. Some authors have tried to study effect of noise in images on the performance of edge detectors [39].

In spite of the aforementioned work, the need for general purpose edge detector is still felt. Edge detectors based on zero crossings of second order derivatives are generally simple but suffer from phantom edges [12]. Moreover, the use of higher order derivatives makes the detector susceptible to high frequency noise and also results in poorer localization of edges. Detectors based on optimality criteria are often derived in continuous one dimensional domain and are extended to two dimensional domains in a subjective way that lacks firm logical justification [4], [23], [37]. Also, a problem very commonly faced by detectors is the choice of threshold values, which are often chosen on heuristic basis. Prewitt's [17], Roberts' [17], and Sobel's [17] operators and zero-crossing [25] edge detectors use thresholds which are generally selected without any precise objective guideline. In the *Matlab* version of Canny's edge detector—the most popular among all edge detectors—the default value of upper threshold is suggested to be 75th percentile of the gradient strength.

Several authors [19], [20] have done extensive study on the performance of different edge detection algorithms by applying them on large number of real life images.[1] These authors have observed that the performance of the well-known edge detectors, like Canny, Nalwa Binford [26], [19], [20], Rothwell [19], [20], Bergholm [2], [19], [20], Iverson [22], [19], [20], etc., depend critically on the choice of the input parameters. They have also reported striking improvement of the performance of some of the edge detectors, especially Canny's, when the choice of the input parameters is done adaptively, instead of using some fixed default values.

Whether a pixel is an edge pixel or not depends upon the gray values of that pixel and its surrounding pixels. Smoothing is necessary to remove the noise when it is present in the image as well as to estimate the image surface in analogue domain. Most of the well-known edge detectors, e.g., Canny's (which uses a Gaussian filter), use some smoothing filter for this purpose. Further, the detectors based on gradient need a smooth estimate of the image surface, and smoothing techniques have been used in the literature for this [25], [4]. After estimating the gradient vector, one should not use the magnitudes of the derivatives alone for determining the eligibility of a pixel to be an edge pixel, though it has been done in that way with many edge detectors. The variations in the neighborhood of a pixel need to be

Manuscript received June 3, 2002; revised April 29, 2003. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Giovanni Ramponi.

R. R. Rakesh is with American Express, Gurgaon, India (e-mail: rishi_r_r@hotmail.com).

P. Chaudhuri is with the Theoretical Statistics and Mathematics Unit, Indian Statistical Institute, Kolkata-700 108, India (e-mail: probal@isical.ac.in).

C. A. Murthy is with the Machine Intelligence Unit, Indian Statistical Institute, Kolkata-700 108, India (e-mail: murthy@isical.ac.in).

Digital Object Identifier 10.1109/TIP.2004.828404

[1]http://marathon.csee.usf.edu/edge/edge_detection.html

incorporated in this analysis. Several authors [24] have pointed out this issue. Threshold values from image to image may vary since the variations in the gray values in the neighborhoods of pixels vary from image to image. In order to automatically find a threshold, standardization of gradient magnitudes is to be done relative to the surrounding pixels' gradient magnitudes, and, then, it is to be tested whether the obtained value is large or not. A natural way of doing such standardization in any procedure is to use appropriate statistical principles. A way of accomplishing the above objective is given in this paper. Our method of standardizing the gradient strength at each pixel locally before thresholding results in the removal of the ambiguity and inappropriateness in choosing global threshold values, and thereby produces reliable, robust, and smooth edges. Local image statistics have been used earlier by Chow and Kaneko [10] to get boundaries in images, and their algorithm was modified by Peli and Lahav [27] for the purpose of detecting bright objects in darker backgrounds.

II. Details of the Methodology

In this section, we first estimate the image surface by a bivariate smoother. The specific smoother used is a bivariate version of the well-known Priestly–Chao [29] kernel smoother. Priestly–Chao kernel smoother is used for fitting curves when the independent variable is equally spaced as in time-series analysis. Since, in image data, the pixels are equally spaced rowwise and columnwise, this particular smoother is used here. Moreover, its use facilitates easy computation of partial derivatives of the fitted surface in horizontal and vertical directions and convenient estimation of their statistical variations.

In the proposed procedure, the gradient vector at each pixel of the fitted surface is calculated. The variation in the gradient vector at each pixel position is estimated using its variance covariance matrix based on standard statistical formulae. This variance covariance matrix is used to standardize the gradient vector at each pixel. This leads to a statistic, which is used to extract regions in the image where the gradient magnitudes are significantly large. Edge pixels are extracted from this region using algorithms similar to *nonmaxima suppression* and *thresholding with hysteresis* [4].

A. Formulae and Mathematical Details

Consider a digital gray level image of size $m \times n$. Let Z_{ij} be the gray level value at pixel (i,j) of this image. Then, the Priestly–Chao kernel smoother f with smoothing parameter h at pixel (x,y) is given by

$$f(x,y) = \frac{1}{2\pi mnh^2} \sum_{i=1}^{m} \sum_{j=1}^{n} K\left(\frac{x-i}{h}\right) K\left(\frac{y-j}{h}\right) Z_{ij} \tag{1}$$

where $K(x)$ is some appropriate kernel function. Note that for any given value of h, the function $f(x,y)$ is a representation of the image surface in the analogue domain at a given scale of smoothing [6], [7], [16]. Let us denote $K((x-i)/h)$ as $a_{i,h}(x)$ and $\sigma^2/(4\pi^2 m^2 n^2 h^4)$ by ξ. Then, the partial derivatives of $f(x,y)$ in x, y directions, denoted as $f_x(x,y)$ (or simply f_x) and $f_y(x,y)$ (or f_y), respectively are calculated as follows:

$$f_x(x,y) = \frac{1}{2\pi mnh^2} \sum_{j=1}^{n} a_{j,h}(y) \left\{ \sum_{i=1}^{m} a'_{i,h}(x) Z_{ij} \right\} \tag{2}$$

and

$$f_y(x,y) = \frac{1}{2\pi mnh^2} \sum_{i=1}^{m} a_{i,h}(x) \left\{ \sum_{j=1}^{n} a'_{j,h}(y) Z_{ij} \right\} \tag{3}$$

where $a'_{i,h}(x)$ is the derivative of $a_{i,h}(x)$.

Note that $(f_x(x,y), f_y(x,y))$ is the gradient vector of the smooth image function $f(x,y)$ at the pixel position (x,y). The estimate of data variability in the image is given by

$$\sigma^2 = \frac{1}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} (Z_{ij} - f(i,j))^2. \tag{4}$$

For every pixel location (x,y), the variance-covariance matrix of $(f_x(x,y), f_y(x,y))$ is given by $\Sigma(x,y)$, where

$$\Sigma(x,y) = \begin{pmatrix} \sigma_{11}(x,y) & \sigma_{12}(x,y) \\ \sigma_{12}(x,y) & \sigma_{22}(x,y) \end{pmatrix}. \tag{5}$$

Here, the expressions for $\sigma_{11}(x,y), \sigma_{22}(x,y)$, and $\sigma_{12}(x,y)$ can easily be calculated from (2), (3), and (4), and they are given as follows:

$$\sigma_{11}(x,y) = \xi \sum_{i=1}^{m} \sum_{j=1}^{n} a^2_{j,h}(y) a'^2_{i,h}(x) \tag{6}$$

$$\sigma_{22}(x,y) = \xi \sum_{i=1}^{m} \sum_{j=1}^{n} a^2_{i,h}(x) a'^2_{j,h}(y) \tag{7}$$

and

$$\sigma_{12}(x,y) = \xi \sum_{i=1}^{m} \sum_{j=1}^{n} a_{i,h}(x) a_{j,h}(y) a'_{i,h}(x) a'_{j,h}(y). \tag{8}$$

Define for every pixel location (x,y) the statistic $S(x,y)$ as

$$S(x,y) = (f_x, f_y)\, \Sigma^{-1}(x,y)\, (f_x, f_y)^T \tag{9}$$

which, after simplification, becomes

$$S(x,y) = \frac{f_x^2 \sigma_{22}(x,y) + f_y^2 \sigma_{11}(x,y) - 2\sigma_{12}(x,y) f_x f_y}{\sigma_{11}(x,y)\sigma_{22}(x,y) - \sigma_{12}^2(x,y)}. \tag{10}$$

$S(x,y)$ denotes the standardized gradient magnitude at the pixel (x,y). For each pixel, $S(x,y)$ is calculated and if its value is found to be sufficiently large, pixel (x,y) is taken to be an edge pixel.

It may be noted that the output obtained after implementing the above procedure may contain thick edges. These thick edges are explained by noticing the fact that the procedure has extracted *a region around an edge pixel* and has not exactly detected *an edge pixel* as shown in Fig. 1.

As already mentioned, an edge pixel is that pixel for which the rate of change of intensity is maximum. So, a way of detecting an edge pixel from the region extracted is to use nonmaxima suppression algorithm. In order to obtain smooth and continuous edges, we suggest the use of two threshold values (S_1 and S_2, $S_1 < S_2$, say) for the statistic S and implementing the idea of

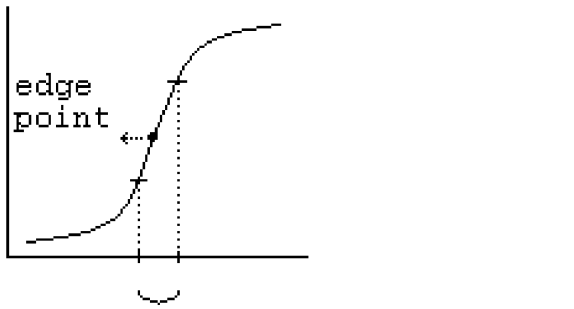


Fig. 1. Region containing edge point as extracted by the procedure.

thresholding with hysteresis. The exact algorithm is now given in the next section.

III. Algorithm for the Proposed Methodology

An algorithm for the proposed methodology is described as follows.

1) Input the gray level image I_{inp} of size $m \times n$. Also, input the two threshold values of S_1 and S_2 and the smoothing parameter h.
2) Calculate the two matrices of size $m \times n$, $((f_x(i,j)))$ and $((f_y(i,j)))$, as in **(2) and (3)**. Here, $f_x(i,j)$ and $f_y(i,j)$ denote the two partial derivatives in the directions of x and y axes of the smoothed image function $f(i,j)$ at the pixel position (i,j), $1 \le i \le m$, $1 \le j \le n$.
3) Define matrices I_{temp} and I_{out}, each of size $m \times n$, with all elements equal to zero.
4) For $i = 1,2,\ldots,m$,
for $j = 1,2,\ldots,n$,
apply nonmaxima suppression algorithm using $f_x(i,j)$ and $f_y(i,j)$. If the pixel (i,j) is not suppressed through nonmaxima suppression, then
calculate the value of S as in **(10)** for the pixel (i,j).
If $S \ge S_2$ then $I_{\text{temp}}(i,j) \leftarrow 2$, else if $S \ge S_1$ then $I_{\text{temp}}(i,j) \leftarrow 1$.
5) For $i = 1,2,\ldots,m$,
for $j = 1,2,\ldots,n$,
if $I_{\text{temp}}(i,j) = 2$ then Track-edge(i,j).

The subprogram Track $-$ edge() declares a pixel with gradient magnitude lying between upper and lower thresholds as edge pixel by checking its connectivity with already declared edge pixels. This forms a subpart of *thresholding with hysteresis* module and results in a smooth and connected edge map. The main steps of the algorithm for this is given as follows.

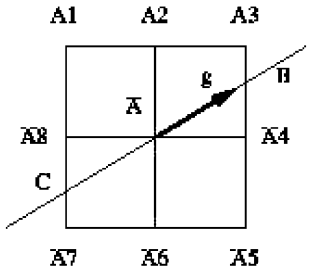


Fig. 2. Gradient magnitude at pixel A is checked for being maxima in a 3×3 neighborhood during *nonmaxima suppression* phase. The arrow mark denotes the direction perpendicular to the edge direction.

Track-edge(i,j)
If $I_{\text{out}}(i,j) \ne 1$, then
1) $I_{\text{out}}(i,j) \leftarrow 1$.
2) For $a = i-1, i, i+1$,
for $b = j-1, j, j+1$,
if $(a,b) \ne (i,j)$ and (a,b) does not belong to the set of considered pixels for (i,j) and $I_{\text{temp}}(a,b) > 0$, then Track-edge(a, b).

Nonmaxima Suppression:
1) Interpolate the magnitude of gradients at hypothetical pixels that lie along the direction perpendicular to the edge direction at pixel (x,y) in a 3×3 neighborhood around it. The gradient vector at (x,y) is obtained using $f_x(x,y)$ and $f_y(x,y)$ as in **(2) and (3)**.
2) If the magnitude of the gradient at (x,y) is not maximum among the interpolated magnitudes, then it is not an edge point.

In Fig. 2, the value at C is interpolated between the values at A_7 and A_8 and the value at B between those at A_3 and A_4.

IV. Some Edge Maps

In all the experiments conducted for this paper, in this section and for the next section, too, we have used the Gaussian kernel $K(x) = (\sqrt{2\pi})^{-1} \exp(-(1/2)x^2)$ for smoothing. This section shows the edge maps obtained after implementing the proposed algorithm on Lena image (Fig. 4) and an artificially created image of smoothed concentric circles. The three regions of the concentric circles are filled with three different gray level values and is smoothed five times by applying mean filter over a window of size 3×3 for each pixel.

Fig. 5 shows edges detected from Lena image (Fig. 4) and superimposition of this edge map on the actual image. It can be seen from the Fig. 5(a) that many of the edges visible to the eye are detected by our algorithm and Fig. 5(b) suggests that the edges detected are correctly localized. It is to be noted that the values of the parameters are taken to be same for both the images under consideration. It is seen from Fig. 6 that the algorithm is able to provide edges in every direction. In the next section, the choice of the values of the parameters is discussed extensively with the help of several examples and the results are compared with the most widely used edge detection method of Canny.

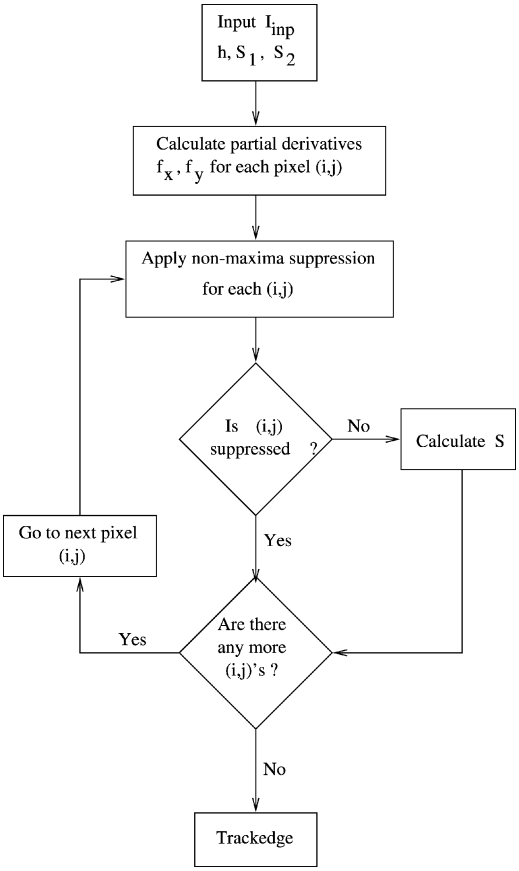


Fig. 3. Flow chart of the algorithm.

Fig. 4. A 256×256 Lena image.

V. Comparative Evaluation of the Proposed Edge Detector

We begin the evaluation by comparing the output of our algorithm with those obtained using Canny's algorithm [2] for three images: X-ray image of hand[3] (Fig. 7), dome image[4], and Lena image corrupted with Gaussian noise. The reason for choosing Canny's methodology for comparison is that it is considered as

[2]Canny's algorithm is taken from Matlab *version* 5.2.0.3084; package available on PCWIN.

[3]The hand image is downloaded from the site *www.prip.tuwien.ac.at/ Teaching/SS/GdBA/LU/Images*, Fig. hand.pgm.

[4]The dome image is downloaded from site *www.cim.mcgill.ca/ wangfang/ canny/node25.html*, fig.(4.1).

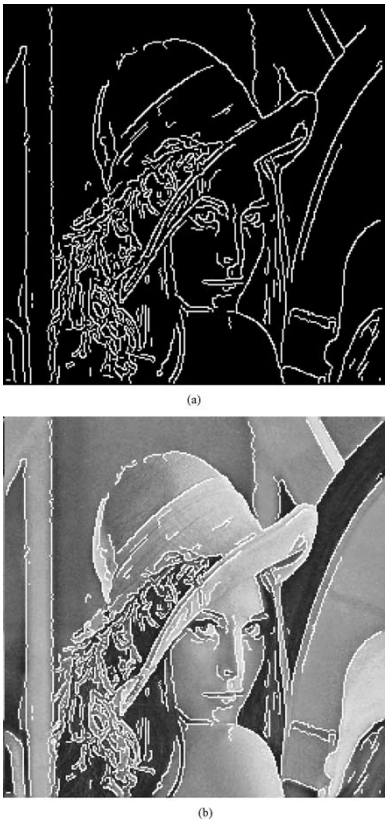


Fig. 5. (a) Edge detected using the proposed method from Lena image (Fig. 4) for $h = 1$, $S_1 = 5$, and $S_2 = 15$. (b) Edge map (a) superimposed on the Lena image (Fig. 4).

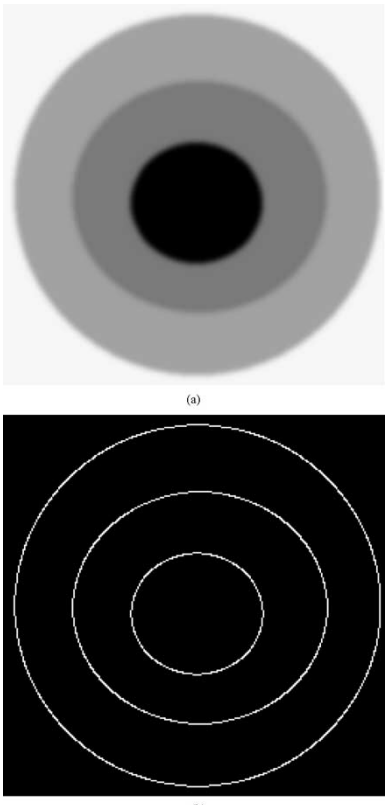


Fig. 6. (a) A 256×256 image of smoothed concentric circles. (b) Edge detected from circle image (a) using the proposed method for $h = 1$, $S_1 = 5$, and $S_2 = 15$.

Fig. 7. X-ray image of hand.

a standard method in edge detection and its usage is reinforced after empirical evaluation done by Doughtry and Bowyer [15]. Heath *et al.* [19], [20] have demonstrated that, while the performance of Canny's algorithm often critically depends on the choices of the input parameters, the quality of the edge map produced by this algorithm, run with appropriate adaptive choices of input parameters, is extremely good for a wide range of real life images.

The output of the proposed and Canny's edge detectors for the hand image (Fig. 7) is shown in Fig. 8. The values of input parameters for the new algorithm are: level of smoothing $h = 1$, lower critical point $S_1 = 5$ and higher critical point $S_2 = 15$ and for Canny's algorithm, the values of the input parameters, chosen to get a visibly very good result, are $\sigma = 1$, $\text{lower thresh} = 0.03$ and $\text{upper thresh} = 0.04$.

The results obtained from the proposed algorithm and Canny's algorithm for different values of parameters are now compared for the dome image (Fig. 9). Edge maps of the dome image extracted by the proposed algorithm and Canny's algorithm at the same values of the parameters as used in hand image (Figs. 7 and 8) are shown in Fig. 10. It is to be noted that the output of Canny's algorithm produced many spurious edges [Fig. 10(b)] when parameter values used were the same as those used in the hand image. On the other hand, our procedure with the same set of parameter values, used in the case of hand image, produced a much cleaner edge map [Fig. 10(a)] free from many such spurious edges.

It may be noted that a visibly nicer edge map is obtained for the dome image by Canny's algorithm with appropriate choices of parameters, as shown in Fig. 11(b). When some small changes are made in the values of input parameters for our procedure, it also produced slightly better results [Fig. 11(a)] than the earlier edge map of Fig. 10(b).

For the case of Lena image (Fig. 4), none of the sets of parameter values at which best results were obtained using Canny's algorithm for previous images (Figs. 7 and 9) could produce visibly clean edge maps—the edge maps produced were all full of many spurious edges. This is evident from Fig. 12(a) and (b). However, the new algorithm remained consistent by providing

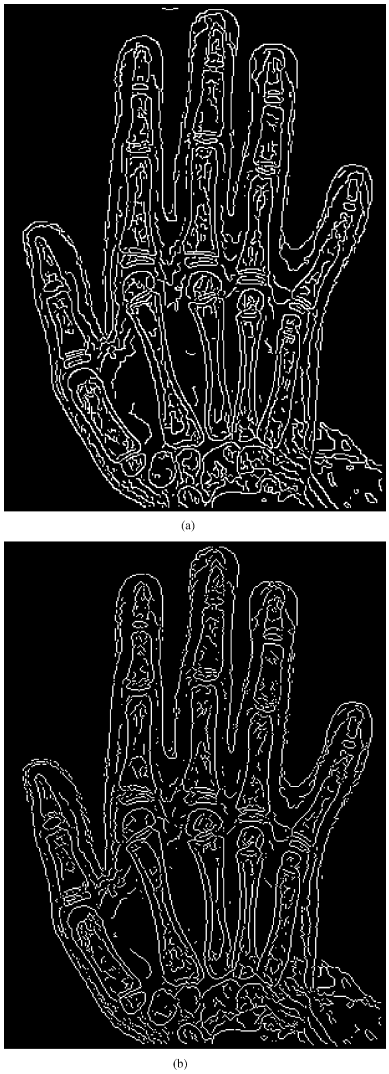


Fig. 8. (a) Edge detected from hand image (Fig. 7) using the new method for $h = 1$, $S_1 = 5$, and $S_2 = 15$. (b) Edge detected from the hand image (Fig. 7) by Canny's method for $\sigma = 1$, $\text{lower thresh} = 0.03$, and $\text{upper thresh} = 0.04$.

Fig. 9. A 350 × 250 image of dome.

visibly clean results at *default values* $h = 1$, $S_1 = 5$, and $S_2 = 15$ for the input parameters (Fig. 5).

We now try our algorithm on a noisy image to judge the extent of stability of its input parameter values. For this purpose, the Lena image added with Gaussian noise with standard deviation 20 [Fig. 13(a)] is considered. The output of the proposed algorithm at the default set of values for the parameters is shown in

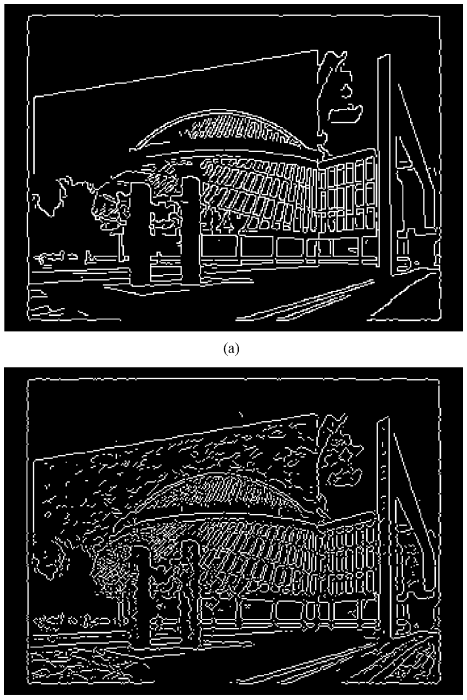


Fig. 10. (a) Edge detected from dome image (Fig. 9) using the new method for $h = 1$, $S_1 = 5$, and $S_2 = 15$. (b) Edge detected from the dome image (Fig. 9) by Canny's method for $\sigma = 1$, $\text{lower thresh} = 0.03$, and $\text{upper thresh} = 0.04$.

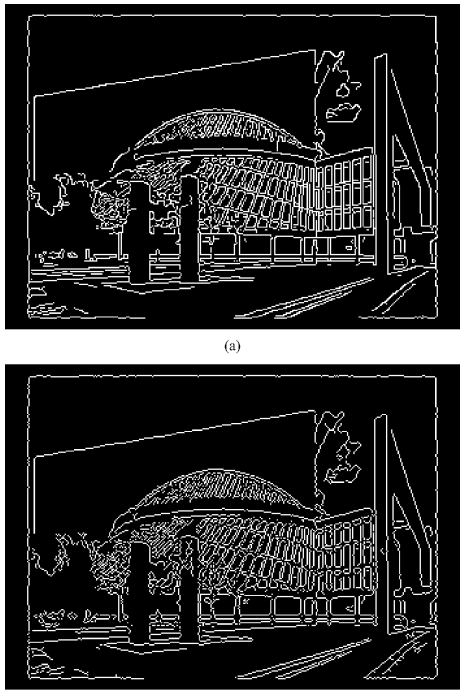


Fig. 11. (a) Edge detected from dome image (Fig. 9) using the new method for $h = 0.75$, $S_1 = 2$, and $S_2 = 12$. (b) Edge detected from the dome image (Fig. 9) by Canny's method for $\sigma = 0.75$, $\text{lower thresh} = 0.0438$, and $\text{upper thresh} = 0.1094$.

Fig. 13(b). It is clear from the figure that the proposed method with default values produced an edge map containing many visibly important edges even for this noisy image, though the edge positions are slightly distorted. Further, there are only a few spurious edges in this edge map. However, at this juncture, we do not intend to carry out a detailed study on the effect of noise on the values of the input parameters of our method.

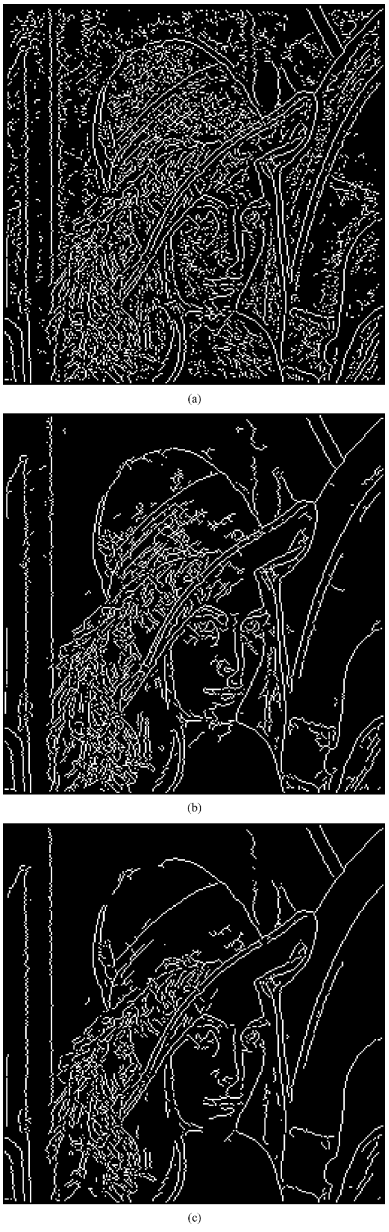


Fig. 12. Edges are detected from Lena image (Fig. 4) using Canny's method. Edge map (a) is obtained by taking $\sigma = 1$, $\text{lower thresh} = 0.03$, and $\text{upper thresh} = 0.04$, which was used to get visibly good result for hand image [Fig. 8(b)]. Edge map (b) is obtained by taking $\sigma = 0.75$, $\text{lower thresh} = 0.0438$, and $\text{upper thresh} = 0.1094$, which was used to get a visibly good result for dome image [Fig. 11(b)]. Edge map (c) is a very good result obtained for the parameter values $\sigma = 1$, $\text{lower thresh} = 0.0563$, and $\text{upper thresh} = 0.1406$.

It may be noted that at each pixel location (i, j), the proposed method involves calculation of S statistic which itself requires computation of variance-covariance matrix and its inverse; however, the matrix is only of size 2×2. The other calculations like computation of gradient vector and those involved in nonmaxima suppression and thresholding with hysteresis in Canny

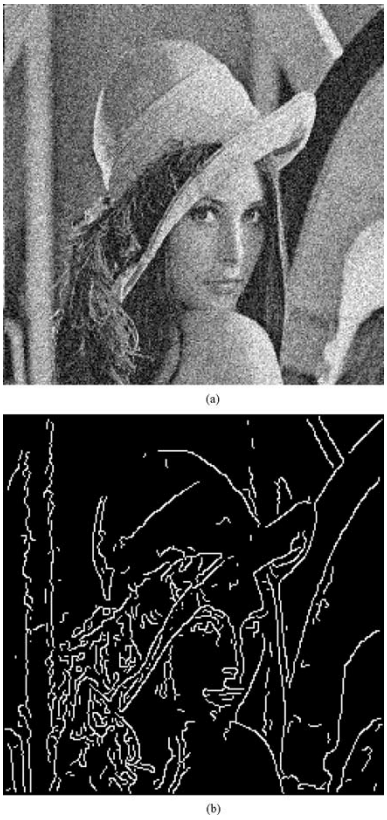


Fig. 13. (a) Lena image added with Gaussian noise with standard deviation 20. (b) Edge detected from the corrupted Lena image (a) by the new method using default parameter values $h = 1$, $S_1 = 5$, and $S_2 = 15$.

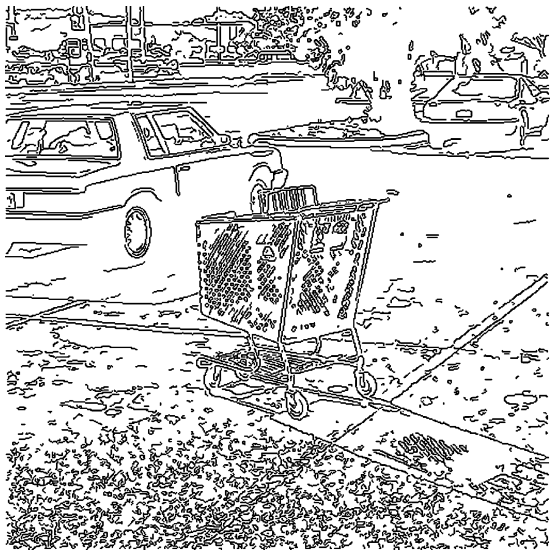


Fig. 14. Edges obtained on the shopping_cart image with the proposed method using default parameter values.

and the proposed algorithms are of similar computational complexities. It is observed that the computational time taken by the proposed method is around four times more than that taken by Canny's method. Thus the robustness of the proposed algorithm is achieved at the cost of computational complexity.

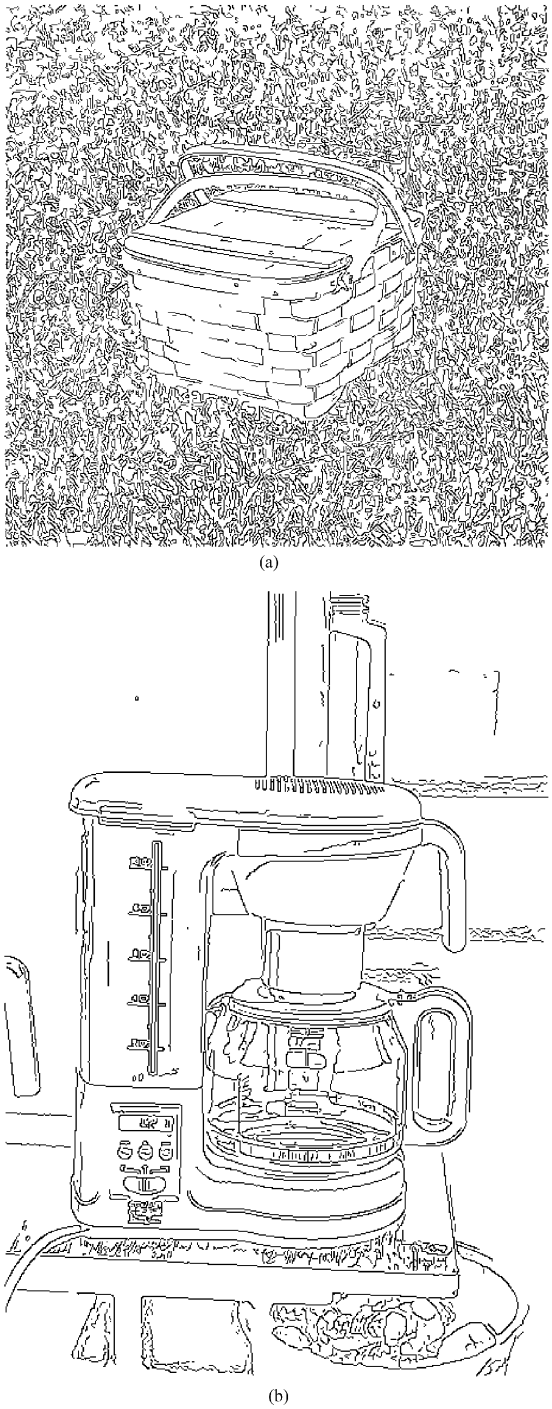


Fig. 15. Edges obtained on (a) picnic_basket and (b) coffee_maker images with the proposed method using default parameter values.

A. Detailed Comparison With Some Standard Algorithms

In order to judge the performance of the proposed algorithm *vis a vis* the other well-known algorithms, comparisons are made on the same set of images with different choices of input parameters. For this purpose, six images, namely coffee_maker, trash_can, shopping_cart, banana, traffic_cone and picnic_basket, are obtained from an internet website at the University of South Florida. These are the images which have been primarily discussed in great detail in [19], [20]. On the same website, the best results obtained using different standard

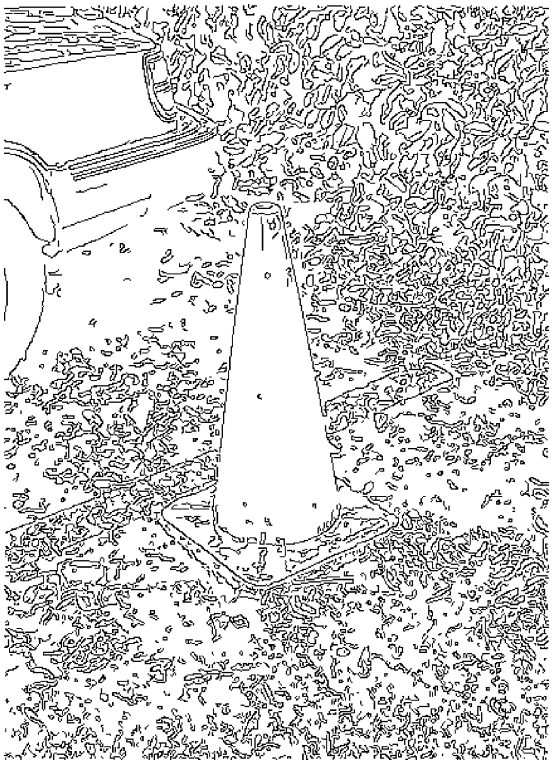


Fig. 16. Edges obtained on traffic_cone image with the proposed method using default parameter values.

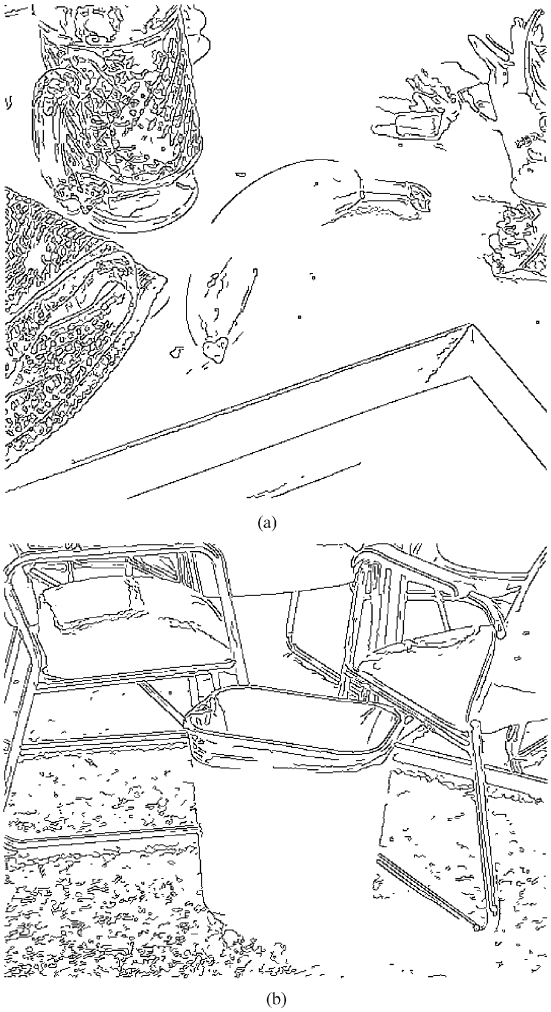


Fig. 17. Edges obtained on (a) banana and (b) trash_can images with the proposed method using default parameter values.

algorithms, namely Canny, Nalwa Binford, Iverson, Bergholm, and Rothwell edge detectors, by changing the parameter values are available for these images. These are described as *best adaptive* results. The results obtained using the *best fixed* set of parameters for the same five methods are available for the same images. Obviously, the *best adaptive* results are at least as good as, if not better than, the *best fixed* results. We shall discuss the performance of our proposed algorithm for each image for a default set of input parameters $h = 1$, $S_1 = 5$, and $S_2 = 15$. We have arrived at these parameter values after extensive numerical experiments. Since the original images as well as the results of the well-known standard algorithms applied to them are available in the same website, we are not reproducing them here in order to save space.

With fixed parameters, the other algorithms except that of Canny's produced too many spurious edges for the shopping_cart image (Fig. 14). These spurious edges get eliminated with the adaptive choices for parameters for these methods. Our method with the default choice of parameter values produced good edge map that is comparable with that obtained by Canny's method. For the three images, namely, picnic_basket, coffee_maker, traffic_cone (Figs. 15 and 16), the proposed method produces edge maps which are visibly at least as good as, if not better than, those produced by other methods with adaptive choice of parameters. On the other hand, for the banana and trash_can images, many important edges are missing in the edge maps produced by all the above-mentioned algorithms, including our method (Fig. 17), with fixed choices of parameter values. Those edges are regained with the adaptive choices of parameter values for other algorithms, and for our algorithm, these edges are regained with parameter values $h = 1.5$, $S_1 = 2$, and $S_2 = 8$ (Fig. 18).

Though, in the case of trash_can and banana images, our method with default parameter values produced edge maps missing some of the important edges, the overall performance of the method with the default parameter values was at least as good as, if not better than, those obtained by best fixed parameter values for the other algorithms stated above.

VI. Concluding Remarks and Discussion

The results of the proposed algorithm have been compared with various methods in the preceding section. The comparison is performed on 1) real life images without noise and 2) a real life image with noise. It is to be noted that on all the real life images considered, the proposed algorithm with a fixed set of input parameters produced fairly good results. This suggests that the proposed methodology is robust over the choice of input parameters (especially the threshold values). This is of great advantage because most of the real life images are of complex nature, and we are generally ignorant about the edges present. Fiddling with the input parameters will not serve the purpose because we may never know whether the edges gained or lost are spurious or

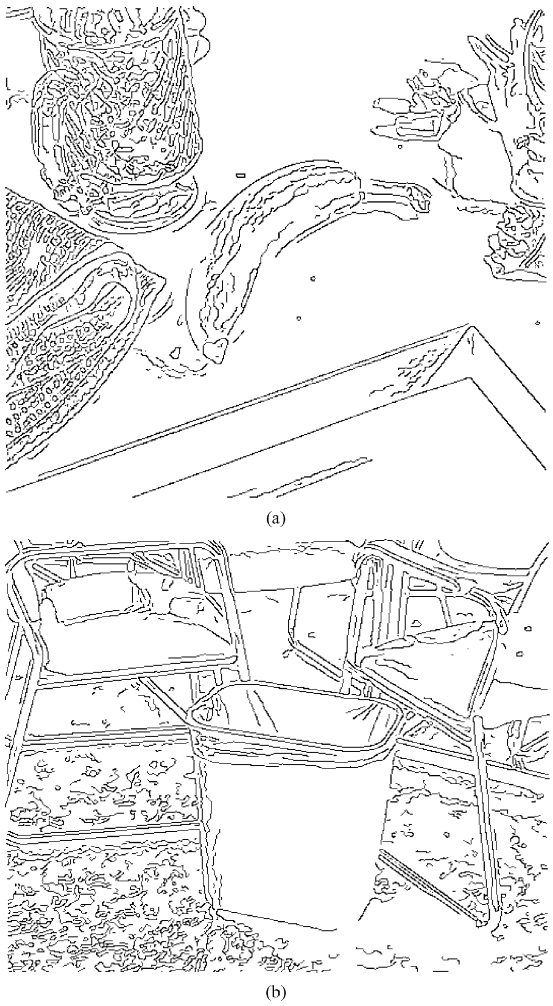


Fig. 18. Edges obtained on (a) banana and (b) trash_can images with the proposed method using $h = 1.5$, $S_1 = 2$, and $S_2 = 8$.

the genuine ones. For this reason, one needs a reliable detector, which, for the same set of input parameter values, would produce appreciable results.

Regarding $S(x, y)$ defined in (10) of Section II, it may be noted that one can use χ^2 distribution with two degrees of freedom as an approximate distribution for it. This approximation is justified if we assume that the Z_{ij}s are a set of random variables having some very general distributions. Note that, since $(f_x(x, y), f_y(x, y))$ is formed using weighted sums of the Z_{ij}s, applying central limit theorem to it will be in order. Thus, we have a Gaussian approximation for the distribution of $(f_x(x, y), f_y(x, y))$, and, as a consequence χ^2, approximation for $S(x, y)$ follows. Our threshold values can be viewed as critical points S_1, S_2 of an approximate χ^2 distribution with two degrees of freedom at two specified levels of significance. Thus, the procedure also associates some confidence levels to the edges extracted. Further, edge being a local property of a pixel, the local standardization used in forming the statistic S is natural and meaningful in this context.

The statistical nature of the analysis conducted makes the choice of input parameters robust for noisy images, too. The methodology provides a way to estimate the variability in the image data locally at each pixel. Then, it is used to get locally standardized gradient magnitudes. This yields our statistic S, which can efficiently handle random noise present in an image. This explains the reason why the proposed algorithm produced appreciable results for the two images Lena and noisy Lena for the same set of values for input parameters.

It is evident from our extensive numerical investigations involving a variety of real life images that the values $S_1 = 5$, $S_2 = 15$ are generally producing results containing almost all important edges at the right locations of the images. A central χ^2 distribution with two degrees of freedom is actually an exponential distribution with $\text{mean} = 2$. $S_1 = 5$ and $S_2 = 15$ correspond to approximately the upper 92nd percentile and the upper 99.95th percentile, respectively, of that distribution. Here, the 92nd percentile possibly implies that normally more than 8%–10% of the pixels in a small neighborhood in the image cannot be appreciated as edge pixels by our vision system. On the other hand, if the strength of the locally standardized gradient of a pixel is more than the 99.95th percentile, one would expect our vision system to surely detect it to be an edge pixel.

ACKNOWLEDGMENT

The authors would like to thank B. L. Narayan for his help in preparing this manuscript.

REFERENCES

[1] I. E. Abdou and W. K. Pratt, "Quantitative design and evaluation enhancement/thresholding edge detectors," *Proc. IEEE*, vol. 67, pp. 753–763, 1979.

[2] F. Bergholm, "Edge focusing," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-9, pp. 726–741, June 1987.

[3] S. M. Bhandarkar, Y. Zhang, and W. D. Potter, "An edge detection technique using genetic algorithm based optimization," *Pattern Recognit.*, vol. 27, no. 9, pp. 1159–1180, 1994.

[4] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-8, pp. 679–697, June 1986.

[5] L. Caponetti, N. Abbattists, and G. Carapella, "A genetic approach to edge detection," in *Int. Conf. Image Processing*, vol. 94, 1994, pp. 318–322.

[6] P. Chaudhuri and J. S. Marron, "Sizer for exploration of structures in curves," *J. Amer. Stat. Assoc.*, vol. 94, no. 447, pp. 807–823, 1999.

[7] ——, "Scale space view of curve estimation," *Ann. Stat.*, vol. 28, no. 2, pp. 408–428, 2000.

[8] G. Chen and Y. H. H. Yang, "Edge detection by regularized cubic B-spline fitting," *IEEE Trans. Syst., Man, Cybern.*, vol. 25, pp. 636–643, Apr. 1995.

[9] M. H. Chen, D. Lee, and T. Pavlidis, "Residual analysis for feature detection," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 13, pp. 30–40, Jan. 1991.

[10] C. K. Chow and T. Kaneko, "Automatic boundary detection of the left ventricle from cineangiograms," *Comput. Biomed. Res.*, vol. 5, pp. 388–410, 1972.

[11] E. Chuang and D. Sher, "Chi-square test for feature extraction," *Pattern Recognit.*, vol. 26, no. 11, pp. 1673–1683, 1993.

[12] J. J. Clark, "Authenticating edges produced by zero-crossing algorithms," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 2, pp. 43–57, Jan. 1989.

[13] L. S. Davis, "A survey of edge detection techniques," *Comput. Graph. Image Process.*, vol. 4, no. 3, pp. 248–270, 1976.

[14] P. deSouza, "Edge detection using sliding statistical tests," *Comput. Vis., Graph. Image Process.*, vol. 23, no. 1, pp. 1–14, 1983.

[15] S. Dougherty and K. W. Bowyer, "Objective evaluation of edge detectors using a formally defined framework," in *Emperical Evaluation Techniques in Computer Vision*, K. W. Bowyer and J. Phillips, Eds. Los Alamitos, CA: IEEE Comput. Soc. Press, 1998, pp. 211–228.

[16] F. Godtleibsen, J. S. Marron, and P. Chaudhuri, "Significance in scale space for bivariate density estimation," *J. Comput. Graph. Statist.*, vol. 11, pp. 1–21, 2002.

[17] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. Reading, MA: Addison-Wesley, 1993.
[18] R. M. Haralick, "Digital step edges from zero crossings of second directional derivatives," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-6, pp. 58–68, Jan. 1984.
[19] M. D. Heath, S. Sarkar, T. Sanocki, and K. W. Bowyer, "A robust visual method for assessing the relative performance of edge-detection algorithms," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 19, pp. 1338–1359, Dec. 1997.
[20] ——, "Comparison of edge detectors, a methodology and initial study," *Comput. Vis. Image Understanding*, vol. 69, pp. 38–54, 1998.
[21] T. J. Hebert and D. Malagre, "Edge detection using a priori model," in *Int. Conf. Image Processing*, vol. 94, 1994, pp. 303–307.
[22] L. A. Iverson and S. W. Zucker, "Logical/linear operators for image curves," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 17, pp. 982–996, Sept. 1995.
[23] J. Koplowitz and V. Greco, "On the edge location error for local maximum and zero-crossing edge detectors," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 16, pp. 1207–1212, Dec. 1994.
[24] M. K. Kundu and S. K. Pal, "Thresholding for edge detection using human psychovisual phenomena," *Pattern Recognit. Lett.*, vol. 4, pp. 433–441, 1986.
[25] D. Marr and E. C. Hildreth, "Theory of edge detection," *Proc. Roy. Soc. London B*, no. 207, pp. 187–217, 1980.
[26] V. S. Nalwa and T. O. Binford, "On detecting edges," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-8, pp. 699–714, June 1986.
[27] E. Peli and M. Lahav, "Drusen measurements from fundus photographs using computerized image analysis," *Opthalmology*, vol. 93, pp. 1575–1580, 1986.
[28] T. Peli and D. Malah, "A study of edge detection algorithms," *Comput. Graph. Image Process.*, vol. 20, no. 1, pp. 1–21, 1982.
[29] M. B. Priestly and M. T. Chao, "Non-parametric function fitting," *J. Roy. Stat. Soc. B*, no. 4, pp. 384–392, 1972.
[30] P. Qie and S. M. Bhandarkar, "An edge detection technique using local smoothing and statistical hypothesis testing," *Pattern Recognit. Lett.*, vol. 17, no. 8, pp. 849–872, 1996.
[31] L. G. Roberts, "Machine perception of three-dimensional solids," in *Optical and Electro-Optical Information Processing*, D. A. Berkowitz, L. C. Clapp, C. J. Koester, and A. Vanderburgh, Jr., Eds. Cambridge, MA: MIT Press, 1965, pp. 159–197.
[32] S. Y. Sarkar and K. L. Boyer, "Optimal infinite impulse response zero-crossing based edge detectors," *Comput. Vis. Graph. Image Process.: Image Understanding*, vol. 54, no. 9, pp. 224–243, 1991.
[33] J. Shen and S. Castan, "An optimal linear operator for step edge detection," *Graph. Models Image Process.*, vol. 54, no. 1, pp. 112–133, 1992.
[34] S. S. Sinha and B. G. Schunk, "A two stage algorithm for discontinuity-preserving surface reconstruction," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 14, pp. 36–55, Jan. 1992.
[35] C. Spinu, C. Garbay, and J. M. Chassery, "Edge detection by estimation and minimization of errors," in *Proc. Int. Conf. Image Processing*, vol. 94, 1997, pp. 303–307.
[36] V. Srinivasan, P. Bhatia, and S. H. Ong, "Edge detection using neural network," *Pattern Recognit.*, vol. 27, no. 12, pp. 1653–1662, 1995.
[37] H. D. Tagare and R. J. P. deFigueiredo, "On the localization performance measure and optimal edge detection," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 12, pp. 1186–1190, Dec. 1990.
[38] V. Torre and T. Poggio, "On edge detection," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-8, pp. 147–163, Feb. 1986.
[39] F. L. Valverde, N. Guil, J. Munoz, R. Nishikawa, and K. Doi, "An evaluation criterion for edge detection techniques in noisy images," in *Int. Conf. Image Processing*, 2001, pp. 766–769.
[40] D. Ziou and S. Tabbone, "Edge Detection Techniques—An Overview," Dept. Math Informatique, Univ. Sherbrooke, Sherbrooke, QC, Canada, Tech. Rep. no. 195, 1997.

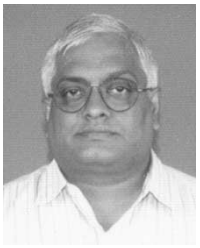
Rishi R. Rakesh was born in India in 1979. He received the B.Sc. degree in statistics from Delhi University, Delhi, India, in 1999 and the M.S. degree in bio-statistics and data analysis from the Indian Statistical Institute, Kolkata, in 2001.

His research interests include pattern recognition, regression techniques, image processing, and data mining. He is currently a Business Analyst with American Express, Gurgaon, India, where he is responsible for the analytics and models required to reduce the risk factor of the American Express credit card business.

Probal Chaudhuri received the B.S. and M.S. degrees in statistics from the Indian Statistical Institute (ISI), Kolkata, in 1983 and 1985, respectively, and the Ph.D. degree in statistics from University of California, Berkeley, in 1988.

He was on the faculty at the University of Wisconsin, Madison, for some time before joining the the faculty of the ISI in 1990. He is currently a Professor at the Theoretical Statistics and Mathematics Unit, ISI. His fields of research interest include nonparametric and robust statistics, pattern recognition, and image processing.

Dr. Chaudhuri was elected a Fellow of the Indian Academy of Sciences in 2003.



C. A. Murthy was born in Ongole, India, in 1958. He received the B.Stat. (Hons.), M.Stat., and Ph.D. degrees from the Indian Statistical Institute (ISI), Kolkata.

He visited Michigan State University, East Lansing, for six months from 1991 to 1992 and Pennsylvania State University, University Park, for 18 months from 1996 to 1997. He is a Professor in the Machine Intelligence Unit, ISI. His fields of research interest include pattern recognition, image processing, machine learning, neural networks, fractals, genetic algorithms, wavelets, and data mining.

Dr. Murthy is a fellow of the National Academy of Engineering, India. He received the best paper award in 1996 in computer science from the Institute of Engineers, India. He received the Vasvik award along with his two colleagues in electronic sciences and technology in 1999.