PERGAMON

Pattern Recognition 35 (2002) 689–700

PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY

www.elsevier.com/locate/patcog

Efficient facet edge detection and quantitative performance evaluation

Qiang Ji[a,]*, Robert M. Haralick[b]

[a]Department of Electrical, Computer, and System Engineering, Rensselaer Polytechnic Institute, Troy, NY 12180, USA
[b]Department of Electrical Engineering, University of Washington, Seattle, WA 98115, USA

Received 5 June 2000; received in revised form 26 December 2000; accepted 26 December 2000

Abstract

In this paper, we first introduce a recursive procedure for efficiently computing cubic facet parameters for edge detection. The procedure allows to compute facet parameters in a fixed number of operations independent of kernel size. We then introduce an image independent quantitative criterion for analytically evaluating different edge detectors (both gradient and zero-crossing based methods) without the need of ground-truth information. Our criterion is based on our observation that all edge detectors make a decision of whether a pixel is an edgel or not based on the result of convolution of the image with a kernel. The variance of the convolution output therefore directly affects the performance of an edge detector. We propose to analytically compute the variance of the convolution output and use it as a measure to characterize the performance of four well-known edge detectors. © 2001 Pattern Recognition Society. Published by Elsevier Science Ltd. All rights reserved.

Keywords: Edge detection; Facet model; Performance evaluation; Feature extraction; Low level image processing

1. Introduction

There has been an abundance of work on different approaches to edge detection. These approaches can be grouped into two main categories: gradient and zero-crossing based methods. While the gradient approach uses the first-order directional derivatives of the image to compute a quantity related to edge contrast for edge detection, the zero-crossing approach requires computation of the second-order directional derivatives to identify locations with zero crossings.

Some of the early gradient operators include the Roberts [1], Sobel [2], and Prewitt [3] edge operators. They all involve using small kernels to convolve with an image to estimate the first-order directional derivatives of the image brightness function. While simple to use, these filters give very little control over image noise and edge location. To overcome these problems, Canny [4] described a gradient-based edge-finding algorithm that has since become one of the most widely used edge detectors. The algorithm uses an optimal filter kernel for computing the first-order directional derivatives of an image in two orthogonal directions. The optimal filter is obtained by maximizing a criterion function consisting of signal-to-noise ratio of the image, the edge location accuracy, and the false positive probability. In practice, the optimal filter can be closely approximated by the first-order derivative of a Gaussian function which allows noise suppression and an efficient implementation. Zuniga and Haralick [5] introduced the integrated directional derivative gradient operator based on the cubic facet model. Here the gradient strength is the maximum value of the integral of the first directional derivative taken over a small neighborhood and over all possible directions. The direction that maximizes the integral defines the estimated gradient direction. Experimental study [5] demonstrated the superiority of this type of gradient operator to others like Roberts in reducing noise sensitivity and estimate bias.

*Corresponding author. Tel.: +1-775-784-6974; fax: +1-775-784-1877.
E-mail address: qji@ecse.rpi.edu (Q. Ji).

While the gradient approach approximates the first-order directional derivatives of pixel values in an image, it is also possible to use the second-order derivatives to detect edges since an edge may generate zero-crossings for the second-order derivatives. A very popular second-order derivative operator is the Laplacian operator. The problem with the second-order derivative approach is that the second-order derivatives tend to exaggerate noise twice as much. Some sort of noise suppression is needed. Marr and Hildreth [6] proposed the use of zero-crossings of the Laplacian of a Gaussian (LOG) for edge detection. The Gaussian serves the purpose of smoothing for noise reduction. Isotropic digital LOG kernels are used to convolve with an image to compute the Laplacian value at each pixel. Haralick [7] proposed the use of the zero-crossings of the second directional derivatives of the image brightness function. Unlike the LOG approach where data smoothing is accomplished using a Gaussian filter, data smoothing is achieved with approximation. The image data are approximated by a smooth function and directional derivatives are then obtained analytically from differentiating the smooth function. A pixel is declared an edgel if its second directional derivative, taken in the direction of the gradient, has a negatively sloped zero-crossing located near the center of the pixel. In spite of the use of discrete Chebyshev polynomials, a problem with the facet model based edge detection is that it is rather time consuming. It requires to convolve a 2D kernel with image to compute each facet coefficient. In this paper, we will introduce an efficient method that implements the 2D convolution with two 1D convolutions and that allows the 1D convolution to be implemented recursively. This significantly reduces the computation time by rendering the computation independent of kernel size.

Despite the enormous amount of literature on edge detectors, there are only a few papers on evaluating and/or comparing the performance of different edge detectors. Such a study is important since it helps us understand the strengths and the weakness of an edge operator as well as its applicability to a particular application. Abdou and Pratt [8] proposed Pratt's figure of Merit criterion for analytically evaluating the edge detectors for synthetic images or real images with ground-truth data. Kitchen and Rosenfeld [9] proposed an edge detection evaluation criterion based on edge coherence without requiring knowledge of the ideal edge position. Ramesh and Haralick [10] proposed a method for evaluating a facet-based edge detector with known perturbation model for the input image. From the input perturbation model, the output perturbation model and the output perturbation can be analytically derived, based on which probabilities of misdetection and false alarm rates are computed analytically. Wang and Binford [11] analytically evaluated the performance of a step-edge detection method by fitting a surface to the gradient magnitude values. Heath et al. [12] recently proposed an empirical method for evaluating the edge detectors for real images based on subjective evaluation of edge images by people without the need of groundtruth information.

In this paper, we describe a new image independent measure for analytically evaluating different edge detectors (both gradient and zero-crossing based methods) without the need of ground-truth information. The criterion, called kernel-variance, evaluates each edge detector based on the variance of its output quantity. For any edge detector, its decision as to whether a pixel is an edgel or not is largely determined by the quantity (gradient or zero-crossing) it computes. The variance of the computed quantity can therefore significantly affect its decision, which, in turn, affects the misdetection and false alarm rates. For comparison, we will evaluate the performance of two gradient operators: the Canny edge detector and the integrated gradient operator; and two zero-crossing edge detectors: the facet zero-crossing edge detector (hereafter referred to as the Haralick edge detector) and the Laplacian of Gaussian (LOG) using both synthetic and real images. This paper is arranged as follows. In Section 2, we introduce the cubic facet model and the conventional method for computing facet coefficients. Section 3 introduces a separable and recursive procedure to efficiently compute the facet parameters. The theory of the integrated gradient edge detector and the Haralick edge detector are briefly covered in Section 4. The performance comparison of the four edge detectors is reported in Section 5. This paper ends with a summary and discussion in Section 6.

2. Cubic facet model

2.1. Discrete orthogonal polynomials

The cubic facet model assumes that the underlying gray level intensity surface of a small neighborhood can be approximated by a bivariate cubic function f. In canonical form, f can be represented as

$$f(r,c) = k_1 + k_2 r + k_3 c + k_4 r^2 + k_5 rc + k_6 c^2 + k_7 r^3 + k_8 r^2 c + k_9 rc^2 + k_{10} c^3, \quad (1)$$

where $k_1, \dots, k_{10}$ are the fitting coefficients obtained through a least-squares surface fitting.

Least-squares surface fitting using the canonical form, however, is computationally intense. An efficient implementation of the surface fitting can be realized if f can be represented as a linear combination of a discrete orthogonal polynomial (DOP) basis set. Also referred to as the *discrete Chebyshev polynomials*, discrete orthogonal polynomials allows independent estimation of each fitting coefficient. In one dimension, the DOP can be summarized as follows. Let $P_0(r), \dots, P_N(r)$ be the 0 to

N order of orthogonal polynomials defined over a domain R. In general, $P_N(r) = r^N + a_{N-1}r^{N-1} + \cdots + a_1 r + a_0$ and it must be orthogonal to each of the $P_0(r), \dots, P_{N-1}(r)$ polynomials, i.e.,

$$\sum_{r \in R} P_i(r)(r^N + a_{N-1}r^{N-1} + \cdots + a_1 r + a_0) = 0, \tag{2}$$

where $i = 0, \dots, N-1$.

Given $N-1$ equations, we can easily solve for the $N-1$ unknowns $a_0, \dots, a_{N-1}$. In practice, for a symmetric domain R, the DOP basis set can be computed recursively using the following relation [13]:

$$P_{i+1}(r) = rP_i(r) - \beta_i P_{i-1}(r), \tag{3}$$

where

$$\beta_i = \frac{\sum_{r \in R} P_i(r) P_{i-1}(r)}{\sum_{r \in R} P_{i-1}(r)^2}.$$

For a cubic polynomial function, the four orthogonal basis functions are therefore

$$P_0(r) = 1,$$

$$P_1(r) = r,$$

$$P_2(r) = r^2 - \frac{\mu_2}{\mu_0},$$

$$P_3(r) = r^3 - \frac{\mu_4}{\mu_2} r, \tag{4}$$

where $\mu_k = \sum_{s \in R} s^k$.

For example, given a symmetric neighborhood $R = \{-2\ -1\ 0\ 1\ 2\}$, the discrete orthogonal polynomial set for a cubic function is

$$P_0(r) = 1,$$

$$P_1(r) = r,$$

$$P_2(r) = r^2 - 2,$$

$$P_3(r) = r^3 - \tfrac{17}{5} r.$$

The two dimensional DOP basis set can be constructed from the tensor product of the two sets of one dimensional discrete polynomials [13]. Let R and C be the index sets that satisfy the symmetric conditions, i.e., $r \in R$ and $c \in C$ imply $-r \in R$ and $-c \in C$. Let $\{P_0(r), \dots, P_N(r)\}$ be a set of discrete polynomials on R and $\{Q_0(c), \dots, Q_M(c)\}$ be a set of discrete polynomials on C, then the set $\{P_0(r)Q_0(c), \dots, P_n(r)Q_m(c), \dots, P_N(r)Q_M(c)\}$ is a set of discrete polynomials on $R \times C$. For example, let R be defined as $R = \{-2\ -1\ 0\ 1\ 2\}$ and C be defined as $C = \{-2\ -1\ 0\ 1\ 2\}$, the set of 2D discrete orthogonal polynomials defined over $R \times C$ is

$$\{1, r, c, r^2-2, rc, c^2-2, r^3 - r\tfrac{17}{5}, (r^2-2)c, r(c^2-2),$$
$$c^3 - c\tfrac{17}{5}, (r^3 - c\tfrac{17}{5})c, (r^2-2)(c^2-2), r(c^3 - c\tfrac{17}{5}),$$
$$(r^3 - c\tfrac{17}{5})(c^2-2), (r^2-2)(c^3 - c\tfrac{17}{5}),$$
$$(r^3 - r\tfrac{17}{5})(c^3 - c\tfrac{17}{5})\}.$$

For a cubic function, the polynomial basis with an order higher than 3 can be ignored. Hence, the set of discrete orthogonal polynomials for a cubic function for the example above is

$$\{1, r, c, r^2-2, rc, c^2-2, r^3 - r\tfrac{17}{5}, (r^2-2)c,$$
$$r(c^2-2), c^3 - c\tfrac{17}{5}\}.$$

As a result, the bivariate cubic function $f(r,c)$, expressed using discrete orthogonal polynomials, is

$$\begin{aligned} f(r,c) = {} & K_1 + K_2 r + K_3 c + K_4(r^2-2) + K_5 rc \\ & + K_6(c^2-2) + K_7(r^3 - \tfrac{17}{5}r) + K_8(r^2-2)c \\ & + K_9 r(c^2-2) + K_{10}(c^3 - \tfrac{17}{5}c), \end{aligned} \tag{5}$$

where K_i, $i = 1, 2, \dots, 10$ are coefficients for bivariate cubic function expressed in discrete orthogonal polynomials. From Eqs. (1) and (5), the coefficients k_i and K_i are related. For $i = 4, 5, \dots, 10$, $k_i = K_i$. The remaining coefficients are related as follows:

$$k_1 = K_1 - 2K_4 - 2K_6,$$

$$k_2 = K_2 - \tfrac{17}{5}K_7 - 2K_9,$$

$$k_3 = K_3 - \tfrac{17}{5}K_{10} - 2K_8. \tag{6}$$

2.2. Least-squares surface fitting with discrete orthogonal polynomials

With the 2D discrete polynomials defined, least-squares surface fitting using the 2D discrete orthogonal polynomials can be described as follows. Let S be a symmetric 2D neighborhood defined on $R \times C$ and $I(r,c)$ be the observed intensity value at $(r,c) \in S$. Let $\{g_0(r,c),\ g_1(r,c), \dots, g_N(r,c)\}$ be the set of 2D DOP basis functions. Then the fitting function f can be expressed as

$$f(r,c) = \sum_{i=1}^{N} K_i g_i(r,c), \tag{7}$$

where K_i are the fitting coefficients. The least-squares surface fitting problem is to determine the coefficients $K_0, \dots, K_N$ such that

$$\varepsilon^2 = \sum_{(r,c) \in S} [I(r,c) - f(r,c)]^2 \tag{8}$$

is minimized. Using the orthogonal property that $\sum_{(r,c)\in S} g_j(r,c)g_i(r,c) = 0$ for $i \neq j$, we have

$$K_i = \frac{\sum_{(r,c)\in S} g_i(r,c)I(r,c)}{\sum_{(r,c)\in S} g_i^2(r,c)}. \tag{9}$$

Eq. (9) shows that each fitting coefficient K_i can be computed individually as a linear combination of the intensity values $I(r,c)$. The weight associated with each $I(r,c)$ for ith coefficient is determined by

$$W_i = \frac{\sum_{(r,c)\in S} g_i(r,c)}{\sum_{(r,c)\in S} g_i^2(r,c)}. \tag{10}$$

Each coefficient K_i can therefore be computed independently by convolving the image with the corresponding weight kernel computed using Eq. (10). Given the estimated fitting coefficients $\hat{K}_i$, the estimated function $\hat{f}$ is

$$\hat{f} = \sum_{i=1}^{N} \hat{K}_i g_i(r,c), \tag{11}$$

$\hat{f}$ is a well-defined function and allows to estimate the directional derivatives at each pixel analytically.

For a rectangular neighborhood, the weights for each fitting coefficient can be represented by a kernel of the same size using Eq. (10). For example, the weights for K_2 and K_3 for the example given above are shown in Table 1. Given W_i, the ith coefficient K_i can be obtained simply by convolving the weight kernel with the image.

3. Separable recursive implementation

Despite that each coefficient can be computed independently, it is still rather computationally intense to perform a 2D convolution for each coefficient. That is especially true considering *we need compute* 10 coefficients for each image pixel. It is therefore necessary to further reduce time associated with the computation of each coefficient. The fact that a 2D DOP basis can be obtained from the tensor product of two 1D polynomial function implies that the 2D weight kernel is separable, i.e., its convolution with an image can be accomplished using two separate 1D convolutions, leading to substantial saving in computation. For example, the polynomial basis corresponds to K_2 in the above example is r, which is obtained by multiplication of zero-order polynomial in C and a first-order polynomial for R. Let w_0^1 be the 1D weight kernel for the zero degree polynomial in C and w_1^1 be the 1D weight kernel for the first-degree polynomial in R, then from Eqs. (4) and (10), we have two vectors

Table 1
Weight kernels W_1 and W_2 for coefficients K_2 and K_3 for a 5×5 neighborhood

	−2	−2	−2	−2	−2		−2	−1	0	1	2
	−1	−1	−1	−1	−1		−2	−1	0	1	2
$\frac{1}{50}$	0	0	0	0	0	$\frac{1}{50}$	−2	−1	0	1	2
	1	1	1	1	1		−2	−1	0	1	2
	2	2	2	2	2		−2	−1	0	1	2

$$w_0^1 = \tfrac{1}{5}[1\,1\,1\,1\,1]^t,$$

$$w_1^1 = \tfrac{1}{10}[-2\ -1\,0\,1\,2]^t.$$

It is clear that W_2 can be obtained from the tensor product[1] of w_1^1 and w_0^1 while W_3 can be obtained from the vector product of w_0^1 and w_1^1. With this K_2 can be obtained by two separate 1D convolutions, i.e., convolving the image with w_0^1, followed by convolving the results with w_1^1. This is applicable to other coefficients too. This can drastically reduce the computational complexity for each pixel from $O(n^2)$ to $O(n)$, where n is the kernel size. This computational saving is substantial for large weight kernels and images.

Our further study shows that the computation for the one dimensional convolution can be further reduced by a recursive computation involving only a fixed number of operations independent of the size of the neighborhood. The recursive procedure can be described as follows. Let $a_1, \ldots, a_n$ be given input. We desire to compute for each $n = 1, \ldots, N$, the following quantities, representing the convolution of the four basis functions (Eq. (4)) of a cubic function with the input data:

$$x_n = \sum_{r=-K}^{K} a_{n+r},$$

$$y_n = \sum_{r=-K}^{K} r a_{n+r},$$

$$z_n = \sum_{r=-K}^{K} (r^2 - a) a_{n+r},$$

$$v_n = \sum_{r=-K}^{K} (r^3 - br) a_{n+r},$$

where a and b may be computed from the input using Eq. (4). It is apparent from the above equations that a direct implementation of this computation would require $2K+1$ operations per parameter. The above quantities can be computed recursively using the procedures described below. First, we consider recursive

[1] Tensor product of two vectors v_1 and v_2 is defined by $v_2 v_1^t$.

computation of x_n. We have

$$x_{n+1} = \sum_{r=-K}^{K} a_{n+1+r} \quad \text{and} \quad x_n = \sum_{r=-K}^{K} a_{n+r},$$

let $r' = r + 1$ and substitute it into x_{n+1}, we have

$$\begin{aligned} x_{n+1} &= \sum_{r'=-K+1}^{k+1} a_{n+r'} \\ &= a_{n+K+1} - a_{n-K} + \sum_{r'=-K}^{K} a_{n+r'} \\ &= x_n + a_{n+K+1} - a_{n-K}. \end{aligned}$$

Next let us consider the recursive computation of y_n. We have

$$y_{n+1} = \sum_{r=-K}^{K} r a_{n+1+r} \quad \text{and} \quad y_n = \sum_{r=-K}^{K} r a_{n+r},$$

let $r' = r + 1$ and substitute it into y_{n+1}, we have

$$\begin{aligned} y_{n+1} &= \sum_{r'=-K+1}^{K+1} (r'-1) a_{n+r'} \\ &= (K+1)a_{n+K+1} + K a_{n-K} + \sum_{r'=-K}^{K} r' a_{n+r'} - x_{n+1} \\ &= y_n - x_{n+1} + (K+1)a_{n+K+1} + K a_{n-K}. \end{aligned}$$

The recursive computation for Z_{n+1}, and v_{n+1} can be obtained by applying the similar derivations, yielding

$$\begin{aligned} z_{n+1} &= z_n - x_{n+1} - 2y_{n+1} + [(1+K)^2 - a]a_{n+K+1} \\ &\quad - (K^2 - a)a_{n-K} \\ v_{n+1} &= v_n - 3(z_{n+1} - y_{n+1}) - (3a - b + 1)x_{n+1} \\ &\quad - (K^3 - bK)a_{n-K} + [(K+1)^3 - b(K+1)]a_{n+K+1}. \end{aligned}$$

It is clear from above equations that each convolution can be performed with fixed computations independent of the kernel size.

4. The facet-based edge detectors

Given $\hat{f}$ from Eq. (11), the first- and second-order directional derivatives of each pixel are then analytically approximated by the derivatives of $\hat{f}$. In this section, we briefly describe two edge detectors due to Haralick that determine if a pixel is an edgel based on $\hat{f}$, the facet approximation of image intensity.

4.1. The Haralick edge detector

Based on the image directional derivatives computed from $\hat{f}$, Haralick [7] proposed that the second directional derivative of the pixel must have a negatively sloped zero crossing, taken in the direction of the gradient and sufficiently near the center of the pixel. This criterion requires the second directional derivative to be zero and the third directional derivative to be negative. For ideal step edge, the detection criterion can be refined using a new quantity called contrast C, which measures the height difference between the relative extrema of the fitted cubic function. Experimentally, C is found to be constant independent of the neighborhood size.

4.2. Integrated gradient edge detector

Given a graytone intensity $I(r,c)$ defined in the row and column coordinate system, the gradient operator $\nabla I(r,c)$ is defined as

$$\nabla I(r,c) = \left(\frac{\partial I(r,c)}{\partial r}, \frac{\partial I(r,c)}{\partial c} \right) \tag{12}$$

where $\partial I(r,c)/\partial r$ and $\partial I(r,c)/\partial c$ represent partial derivatives in two orthogonal directions (row and column directions), respectively.

Under the cubic model, each surface facet centered about a given pixel may be approximated by the bivariate cubic in canonical form

$$\begin{aligned} I(r,c) &= k_1 + k_2 r + k_3 c + k_4 r^2 + k_5 rc + k_6 c^2 + k_7 r^3 \\ &\quad + k_8 r^2 c + k_9 r c^2 + k_{10} c^3. \end{aligned}$$

Evaluating the first row and column partial derivatives at the neighborhood center (0,0) yields

$$\frac{\partial I(r,c)}{\partial r} = k_2,$$

$$\frac{\partial I(r,c)}{\partial c} = k_3.$$

According to Zuniga and Haralick [5], the integrated directional derivative can be defined as follows. Let $I'_\theta(r,c)$ represent the first directional derivative along the direction θ, then we have

$$I'_\theta(r,c) = \frac{\partial I(r,c)}{\partial r} \sin\theta + \frac{\partial I(r,c)}{\partial c} \cos\theta.$$

The integrated directional derivative F_θ for an $N \times N$ neighborhood along the θ direction is defined as

$$F_\theta = \frac{1}{4N^2} \int_{-L}^{L} \int_{-W}^{W} I'_\theta(r,c)\,\mathrm{d}r\,\mathrm{d}c, \tag{13}$$

where the integral limits L and W are parameters that need to be adjusted to achieve the best performance. When $L = W = 0$, the integrated gradient degrades to the standard cubic gradient operator. As L and W increase, the performance is expected to improve and reaches a maximum before they are equal to half of the kernel size. The θ that maximizes F_θ is the estimated gradient direction, i.e., the integrated gradient estimate is

$$G = F_{\theta_{\max}} u_{\theta_{\max}},$$

where $F_{\theta_{\max}} = {}^{\max}_{\theta} F_\theta$, $\theta_{\max} = \arg^{\max}_{\theta} F_\theta$, and $u_{\theta_{\max}}$ is the unit vector in the direction that maximizes F_θ. For $L = W$, the θ that maximizes F_θ is

$$\theta = \tan^{-1}\frac{B}{A}, \tag{14}$$

$$F_{\theta_{\max}} = \sqrt{A^2 + B^2}, \tag{15}$$

where $A = L^2k_7 + \frac{1}{3}L^2k_9 + k_2$ and $B = L^2k_{10} + \frac{1}{3}L^2k_8 + k_3$. The integrated gradient operator, however, treats each pixel equally while performing integration.

From Eq. (6), we know that k_i can be computed as a linear combination of K_i. Like K_i, k_i can therefore be computed by convolving its kernel w_i with an image, where w_i can be obtained from a linear combination of kernels for K_i using Eq. (6). For example, weight kernel w_2 for coefficient k_2 is

$$w_2 = W_2 - \tfrac{17}{5}W_7 - 2W_9,$$

where W_2, W_7, and W_9 are weight kernels for coefficients K_2, K_7, and K_9, respectively. The weight kernels W_A and W_B for A and B can obtained as a linear combination of weight kernels w_i. For example, for a 5×5 neighborhood, $L = 2$ and $W_A = 4w_7 + \frac{4}{3}w_9 + w_2$.

5. Performance evaluation

Most edge detectors, be the gradient-based methods or zero-crossing approaches, require convolving an image with a kernel to compute gradients or zero-crossings. A decision is then made as to whether a pixel is an edgel or not based on the result of the convolution. The performance of an edge detector therefore largely depends on the result of the convolution, which is determined by the kernel. The variance of the convolution output therefore directly affects the performance of the edge detector. A larger variance with the convolution result usually leads to a higher misdetection and false alarm rate. Based on the above analysis, we adopt the kernel-variance criterion for comparing different edge detectors. An optimal edge detector has a convolution kernel that yields the smallest variance on its output given the same input perturbation subject to constraints imposed on the kernel *such symmetry*. Using this criterion, we studied the performance of four edge detectors, two of which are gradient-based edge detectors and two of which are zero-crossing based edge detectors. The two gradient-based edge detectors are the Canny edge detector and the integrated gradient edge detector. The two zero-crossing based edge detectors are the Haralick edge detector and Marr's Laplacian of a Gaussian (LOG) operator. The results of this study are summarized in Sections 5.1 and 5.2, respectively.

5.1. Performance comparison of gradient edge detectors

This section discusses the results from a quantitative performance analysis and characterization of the Canny gradient edge detector and the integrated gradient edge detector using the minimum-variance criterion. To have a fair comparison between the two gradient operators, the gradient information obtained from the integrated gradient operator is input to the Canny's hysteresis linking procedure. The resulting edge image is then compared with the output of the Canny edge image. Since both use the same edge linker, any performance difference in the edge detection must be due to the performance difference of the two gradient operators.

Given a $2M \times 2N$ kernel, let y be its output and $w(r,c)$ be the entries of the kernel, then its response to the image $I(r,c)$ is

$$y = \sum_{r=-M}^{M} \sum_{c=-N}^{N} w(r,c)I(r,c) = W'I,$$

where W is a $4MN \times 1$ vector whose elements are $w(r,c)$ and I is a vector containing the corresponding intensity values $I(r,c)$. σ_y^2, the variance of y, is

$$\sigma_y^2 = E(W'II'W)$$
$$= W'\Sigma_I W,$$

where Σ_I is the covariance matrix of vector I.

If elements of I are contaminated by an independently and identically Gaussian distributed noise with zero mean and a standard deviation of σ, then we have $\sigma_y^2 = \sigma^2 W'W$. Fig. 1 plots the output variances of the horizontal Canny gradient kernel and the horizontal integrated gradient kernel (W_A) versus kernel size.

The Canny kernel is approximated with the first order derivative of a 1D Gaussian function. Tables 2 and 3 give examples of two 5×5 horizontal Canny and integrated gradient kernels. For the Canny kernel, kernel size is related to the smoothing factor σ. Increasing kernel size requires increasing σ accordingly to avoid a truncated kernel.

In reality, the iid perturbation assumption may not hold. Perturbations on each pixel may be correlated, especially for neighboring pixels. To model the inherent correlation between pixel perturbations resulted from image acquisition, we assume that the iid perturbed image is subsequently convolved with a Gaussian kernel. The Gaussian kernel introduces correlations to image pixel perturbation. Please note that the Gaussian kernel is for modeling local correlation among image pixels. It is not used for noise smoothing. Let the Gaussian kernel be g, the gradient kernel be h, and the iid perturbed image be I, then the process of introducing correlation and subsequent convolution with a gradient kernel can be described as

$$y = h*(I*g),$$

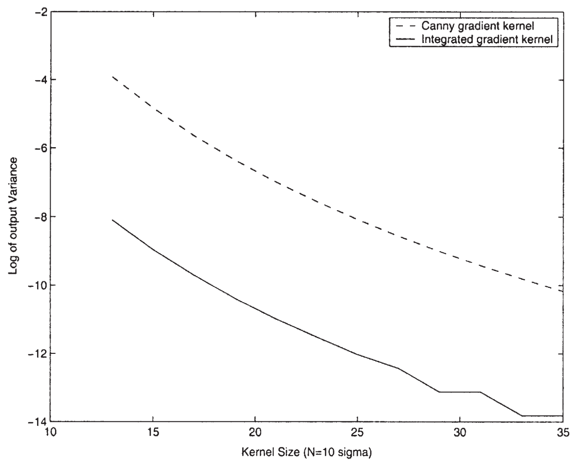

Fig. 1. Kernel output variance versus kernel size for iid input perturbation.

where $*$ represents convolution and y is output. Since convolution is communicative, we have

$$y = (h*g)*I. \tag{16}$$

Eq. (16) suggests that convolving h with a non-independently perturbed image is the same as convolving the gradient kernel h first with g and then convolving an independently contaminated image with the resulting kernel $h' = h*g$. To study the Canny gradient and integrated gradient operators on non-iid perturbed images, we can convolve each gradient kernel with a Gaussian g and then study the output variance of the resulting kernels. Tables 5 and 6 give examples of two kernels resulted from convolving kernels in Tables 2 and 3 with the Gaussian kernel shown in Table 4.

Fig. 2 shows the performance of the two kernels for an image smoothed by Gaussian kernels of different sizes 3 and 5, respectively. An example of 5×5 Gaussian kernel is shown in Table 4.

Both Figs. 1 and 2 show that as kernel size increases, the output variance reduces, i.e., larger kernel window yields better estimate. This agrees with our intuition. However, larger window size introduces more locational errors and requires more computation. In reality, a balance must be stricken between the estimate precision, the locational errors, and the computational complexity. From the minimum-variance point of view, both Figs. 1 and 2 show the integrated gradient operator is superior to the Canny optimal gradient operator. This result agrees with the conclusions drawn by Zuniga and Haralick [5]. Fig. 2 also shows that the impact of the size of Gaussian smoothing kernel on output variance is not significant, especially when the edge kernel is large.

To validate this conclusion, we perform further performance evaluation of the two edge detectors using both synthetic and real images. Fig. 3 shows the synthetic test image used. Downloaded from the Internet, the SUSAN image [14] is selected because it contains different types of edges such as step edge, roof edge, and ramp edge.

The test performed is to evaluate the robustness of the edge detectors under different noise levels. The input test image was perturbed with independently and identically a Gaussian distributed noise with zero mean and a standard deviation of σ. Amount of perturbation is controlled by varying σ. For the integrated gradient operator and Canny edge detector, the low and high thresholds for hysteresis linker are fixed at 1 and 3. The smoothing factor for Canny is set at 0.8. All other parameters are optimally tuned. Figs. 4 and 5 show edge detection results for the two edge detectors when noise level is at 5.

We also applied the two edge detectors to real images. Figs. 6 and 7 give the sample outputs from a real image. We can conclude from both the synthetic and real images that Canny tends to generate more false edges but fewer missing edges than the integrated gradient method. This echos the conclusion drawn from the minimum-variance analysis.

5.2. Haralick facet zero-crossing operator versus Laplacian zero-crossing operator

In this section we study the performance difference between the LOG zero-crossing operator and Haralick's

Table 2
An example of 5×5 horizontal Canny kernel

0.000001	0.000231	0.001708	0.000231	0.000001
0.000116	0.046640	0.344628	0.046640	0.000116
0.000000	0.000000	0.000000	0.000000	0.000000
– 0.000116	– 0.046640	– 0.344628	– 0.046640	– 0.000116
– 0.000001	– 0.000231	– 0.001708	– 0.000231	– 0.000001

Table 3
An example of 5×5 horizontal integrated gradient kernel

− 0.011048	− 0.050476	− 0.063619	− 0.050476	− 0.011048
− 0.012190	− 0.031905	− 0.038476	− 0.031905	− 0.012190
0.000000	0.000000	0.000000	0.000000	0.000000
0.012190	0.031905	0.038476	0.031905	0.012190
0.011048	0.050476	0.063619	0.050476	0.011048

Table 4
An example of 5×5 Gaussian kernel

2.2795779e − 05	0.00106058409	0.00381453967	0.00106058409	2.2795779e − 05
0.00106058409	0.0493441855	0.177473253	0.0493441855	0.00106058409
0.00381453967	0.177473253	0.638307333	0.177473253	0.00381453967
0.00106058409	0.0493441855	0.177473253	0.0493441855	0.00106058409
2.2795779e − 05	0.00106058409	0.00381453967	0.00106058409	2.2795779e − 05

Table 5
An example of 5×5 correlated Canny kernel after convolving a Canny kernel with a Gaussian kernel

0.002735	0.025790	0.066938	0.025790	0.002735
0.009621	0.090713	0.235447	0.090713	0.009621
− 0.000000	0.000000	− 0.000000	0.000000	− 0.000000
− 0.009621	− 0.090713	− 0.235447	− 0.090713	− 0.009621
− 0.002735	− 0.025790	− 0.066938	− 0.025790	− 0.002735

Table 6
An example of 5×5 correlated integrated gradient kernel after convolving a integrated gradient kernel with a Gaussian kernel

− 0.020031	− 0.053859	− 0.068612	− 0.053859	− 0.020031
− 0.018028	− 0.041999	− 0.052058	− 0.041999	− 0.018028
− 0.000000	0.000000	0.000000	0.000000	0.000000
0.018028	0.041999	0.052058	0.041999	0.018028
0.020031	0.053859	0.068612	0.053859	0.020031

facet zero-crossing operator in terms of the minimum variance criterion we established in the previous section. The Laplacian of each pixel can be approximated using a LOG kernel or can be obtained analytically from the fitted facet coefficients. We first analyze the difference between the two methods in terms of the variance of the estimated Laplacian coefficients, assuming each pixel is perturbed by an iid Gaussian distributed noise of mean 0 and variance σ^2. Any difference reveals the difference between the two different data smoothing approaches,

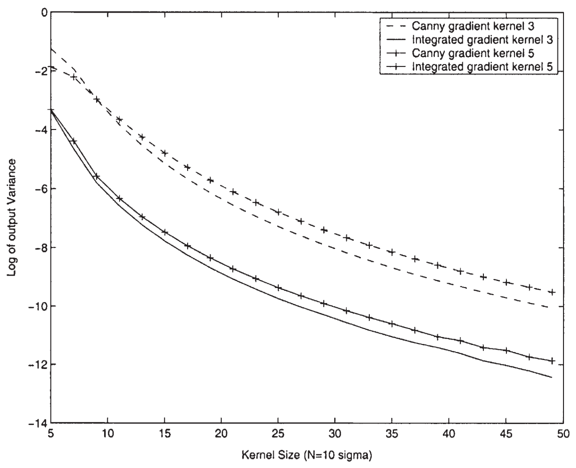

Fig. 2. Kernel output variance versus kernel size with correlated input perturbations generated by two different sizes of Gaussian kernels.

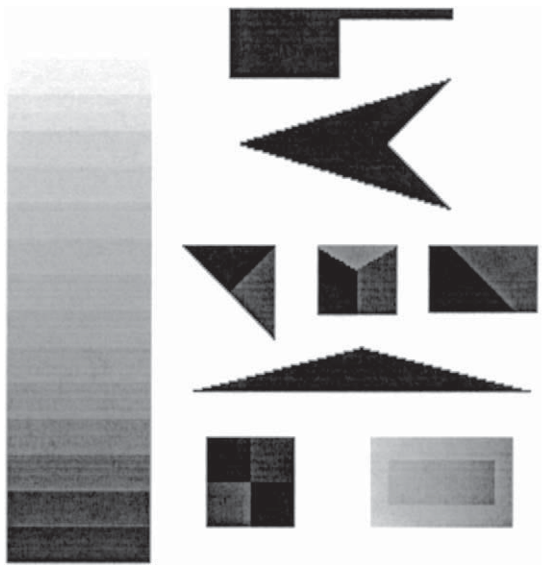

Fig. 3. The synthetic test image.

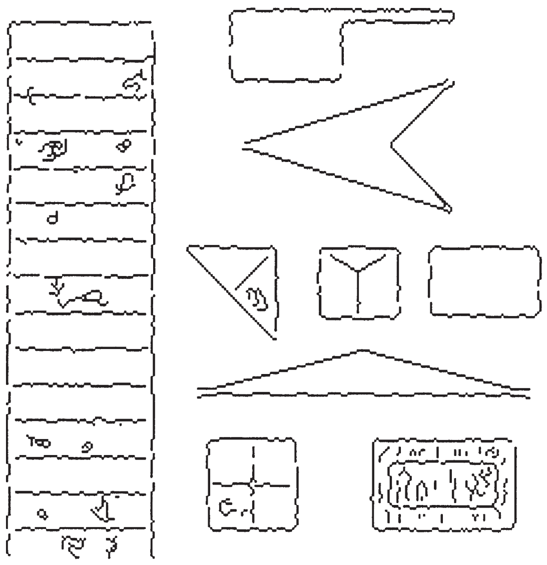

Fig. 4. Output of integrated edge detector when noise level is 5.

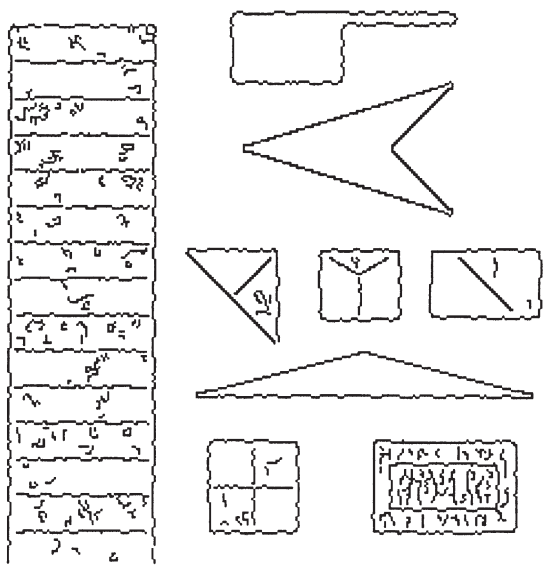

Fig. 5. Output of Canny edge detector when noise level is 5.

i.e., Gaussian smoothing and smoothing via surface fitting.

A 2D LOG kernel can be obtained from

$$\mathrm{LOG}(r,c) = \frac{-1}{2\pi\sigma^4}\left(2 - \frac{r^2+c^2}{\sigma^2}\exp^{(-1/2)((r^2+c^2)/\sigma^2)}\right). \quad (17)$$

Under the cubic model, each surface facet centered about a given pixel may be approximated by the bivariate cubic in canonical form

$$I(r,c) = k_1 + k_2 r + k_3 c + k_4 r^2 + k_5 rc + k_6 c^2 + k_7 r^3 + k_8 r^2 c + k_9 rc^2 + k_{10} c^3.$$

From the cubic fit $I(r,c)$, its Laplacian is $2k_4 + 2k_6$, where k_4 and k_6 are obtained from the cubic facet fitting procedure described in Section 2.1. Figs. 8 and 9 plot the output variances of the LOG kernel and the facet Laplacian kernel versus the kernel sizes, with iid input perturbation and correlated input perturbation

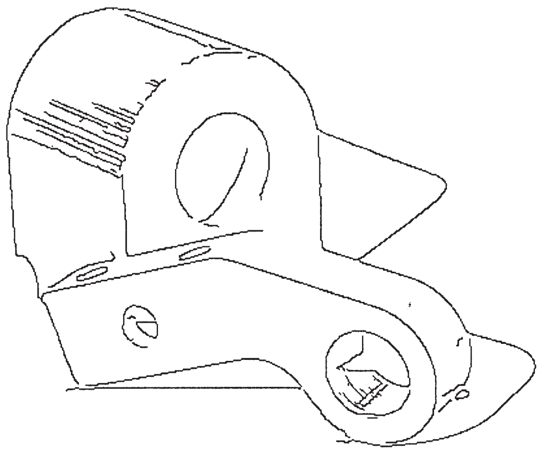

Fig. 6. Output of Canny edge detector.

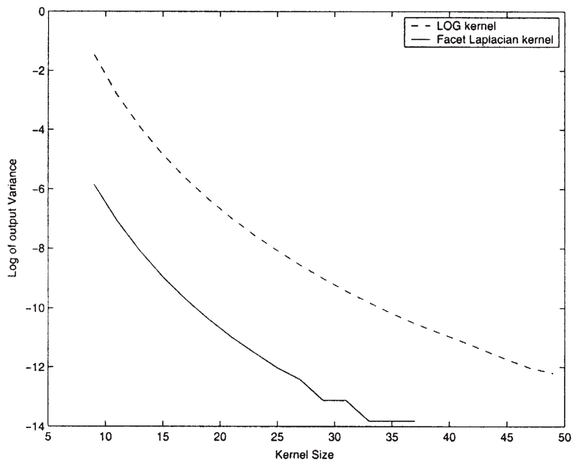

Fig. 8. Kernel output variance versus kernel size for LOG and facet Laplacian kernel with iid perturbation.

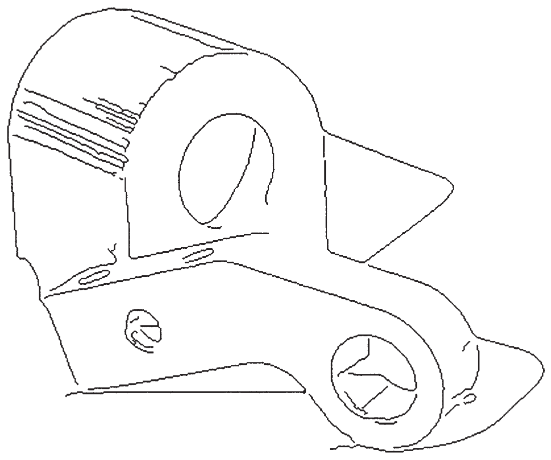

Fig. 7. Output of the integrated gradient edge detector.

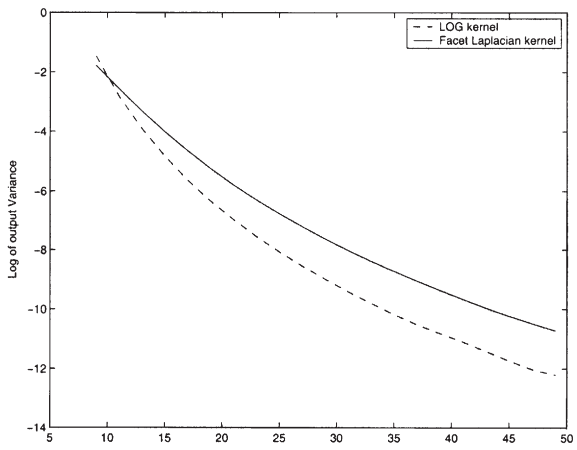

Fig. 9. Kernel output variance versus kernel size for LOG and facet Laplacian kernel with correlated input perturbation.

respectively. It is clear from the two figures that for iid input perturbation, the Laplacian of a pixel computed using facet parameters has a much lower variance than that obtained using LOG kernel, especially when kernel size is small. However, for real images (image with correlated pixel perturbations), the two techniques generate comparable variance. They also show that if LOG must be used, do not use LOGs with kernel size less than 11 since they may yield very unreliable results. For kernel sizes larger than 30 pixels, the two methods yield very comparable results.

5.3. Edge detector performance comparison via ROC analysis

To further study the performance of the edge detectors under different parameters settings, we performed an receiver operating characteristics (ROC) analysis of different edge detectors using the synthetic image shown in Fig. 3. The analysis yields the ROC curves of each edge detector. A ROC curve plots the false alarm probability versus the misdetection probability. It defines the inherent tradeoff between false alarm and misdetection. Each point on the ROC curve results from a particular combination of selected parameters. For the two gradient detectors, the parameters that vary are the high and low thresholds. For the Haralick edge detector, edge contrast is the parameter that changes. Furthermore, the smoothing factor of the Canny edge detector is fixed, consequently the kernel size. The kernel size for the integrated gradient operator is fixed at 5×5. Perturbation for all three edge detectors are set at 5. Fig. 10 plots the ROC curves (false alarm versus misdetection) of each edge detector investigated.

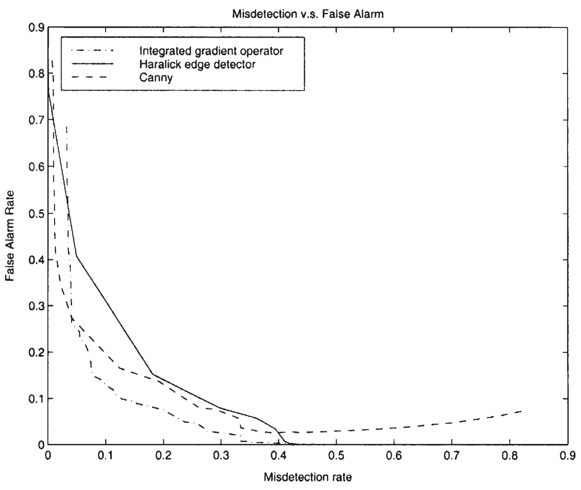

Fig. 10. The ROC curves: false alarm versus misdetection.

The area under the ROC curve (estimated visually) can be used as an index for measuring the performance of the edge detectors. The smaller the area under the ROC curve is, the better performance of the detector is. From Fig. 10, we can see integrated gradient detector yields the best performance, followed by the Canny and Facet edge detector. This basically echos the conclusions we drew from kernel variance analysis.

6. Summary

In this paper, we first describe the cubic facet model and related edge detectors. We then introduce a separable and recursive procedure for efficiently computing the facet parameters. The procedure allows to compute each facet parameter in a fixed number of operations independent of kernel size. To evaluate and compare the performance of different edge detectors, we propose a quantitative measure based on the variance of the edge detector's output. The comparative study based on this measure reveals that the integrated gradient operator coupled with Canny's hysteresis linking procedure can yield better edge detection result than the Canny edge detector. While computationally Canny is still more efficient, the recursive procedure introduced in the paper can dramatically reduce the computations associated with the integrated gradient operator. The study also shows the superiority of facet Laplacian kernel to the LOG kernel in terms of the variance with the computed zero-crossings.

Like other proposed evaluation criteria without the use of groundtruth [6,15–17], the proposed kernel-variance criterion, however, has a drawback. While it can accurately characterize the performance of an edge detector in terms of its misdetection and false alarm rates, it fails to consider location accuracy. Locational errors are affected by the window size. While large window size tend to generate small variance for any type kernel, it, however, tends to lead to a large location errors. The use of Canny's hysteresis linking procedure can yield an optimal edge location for a given kernel size.

References

[1] L.G. Roberts, Machine perception of three-dimensional solids, In Optical and Electrooptical Information Processing, MIT Press, Cambridge, MA, 1965.

[2] I. Sobel, Neighborhood coding of binary images for fast contour following and general array binary processing, Comput. Graphics Image Process. 8 (1978) 127–135.

[3] J.M.S. Prewitt, Object enhancement and extraction, in: B.S. Lipkin, A. Rosenfeld (Eds.), Picture Analysis and Psychopictorics, Academic Press, New York, 1970.

[4] John Canny, A computational approach to edge detection, IEEE Trans. Pattern Anal. Mach. Intell. 8 (6) (1986) 679–698.

[5] O.A. Zuniga, R.M. Haralick, Integrated directional derivative gradient operator, IEEE Trans. Syst. Man Cybern. SMC-17 (3) (1987) 508–517.

[6] D. Marr, E.C. Hildreth, Theory of edge detection, Proc. Roy. Soc. London B-207 (1980) 187–217.

[7] R.M. Haralick, Digital step edges from zero crossing of second directional derivatives, IEEE Trans. Pattern Anal. Mach. Intell. (1984) 58–68.

[8] K.E. Abdou, W.K. Pratt, Quantitative design and evaluation of enhancement/thresholding edge detectors, Proc. IEEE 67 (5) (1979) 753–763.

[9] L. Kithchen, A. Rosenfeld, Edge evaluation using local edge coherence, IEEE Trans. Syst. Man Cybern. 1 (1981) 597–605.

[10] V. Ramesh, R.M. Haralick, Random perturbation model and performance characterization in computer vision. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 1992, pp. 521–527.

[11] S. Wang, T. Binford, Local step edge estimation: a new algorithm, statistical model, and performance evaluation, Proceedings of the 1993 ARPA IUW 1993, pp. 1063–1170.

[12] K.W. Boywer, M. Health, S. Sarkar, T. Sanocki, A robust visual method for assessing the relative performance of edge-detection algorithms, IEEE Trans. Pattern Anal. Mach. Intell. 19 (12) (1997) 1063–1170.

[13] Robert M. Haralick, Linda G. Shapiro, Computer and Robot Vision, Vol. 2, Addison-Wesley Publishing Company, Reading, MA, 1993.

[14] S.M. Smith, Susan – a new approach to low level image processing, DRA Technical Report TR95SMS1b, Department of Engineering Science, Oxford University, Oxford, UK, 1995.

[15] Q. Zhu, Efficient evaluation of edge connectivity and width uniformity, Image Vision Comput. 14 (1996) 21–34.

[16] D.J. Bryant, D.W. Biuldin, Evaluation of edge operators using relative and absolute grading, Proceedings of IEEE Computer Society Conference on Pattern Recognition And Image Processing 1979, pp. 138–145.

[17] K. Cho, P. Meer, J. Cabrera, Quantitative evaluation of performance through bootstrapping: edge detection, IEEE International Symposium on Computer Vision 1996, pp. 491–496.

About the Author—QIANG JI received a MS degree in Electrical Engineering from the University of Arizona in 1993 and his Ph.D. degree in Electrical Engineering from the University of Washington in 1998. His areas of research include computer vision, image processing, pattern recognition, and robotics. Dr. Ji is currently an Assistant Professor at the Department of Electrical, Computer, and System Engineering at the Rensselaer Polytechnic Institute. Previously, he was an Assistant Professor with the Department of Computer Science at University of Nevada at Reno. Between May 1993 and May 1995, he was a research engineer with Western Research Company, Tucson, Arizona, where he served as a principle investigator on several NIH funded research projects to develop computer vision and pattern recognition algorithms for biomedical applications. In summer 1995, he was a visiting technical staff with the Robotics Institute, Carnegie Mellon University, where he developed computer vision algorithms for industrial inspection. Dr. Ji has published numerous papers in referred journals and conferences in these areas. His research has been funded by local and federal government agencies such as NIH, ARO and AFOSR and by private companies including Boeing and Honda.

About the Author—ROBERT M. HARALICK received from the University of Kansas a BA degree in Mathematics in 1964, a BS degree in Electrical Engineering in 1966, and a MS degree in Electrical Engineering in 1967. In 1969, after completing his Ph.D. at the University of Kansas, he joined the faculty of the Electrical Engineering Department where he served as Professor from 1975 to 1978. In 1979, Dr. Haralick joined the Electrical Engineering Department at Virginia Polytechnic Institute and State University where he was a Professor and Director of the Spatial Data Analysis Laboratory. From 1984 to 1986, Dr. Haralick served as Vice President of Research at Machine Vision International, Ann Arbor, Michigan. Dr. Haralick now occupies the Boeing Clairmont Egtvedt Professorship in the Department of Electrical Engineering at the University of Washington. Professor Haralick was elected Fellow of IEEE for his contributions in computer vision and image processing. He serves on the Editorial Boards of Machine Vision and Applications and Real Time Imaging, and he is an Associate Editor for IEEE Transactions on Image Processing and Journal of Electronic Imaging. Professor Haralick's recent work is in shape analysis and extraction using the techniques of mathematical morphology, and in robust pose estimation, techniques for making geometric inferences from perspective projection information, propagation of random perturbations through image analysis algorithms, and document analysis. He has developed the morphological sampling theorem that establishes a sound shape/size basis for the focus of attention mechanisms that can process image data in multiresolution mode, thereby making some of image feature extraction processes execute more efficiently.