COMPUTER GRAPHICS AND IMAGE PROCESSING 20, 1–21 (1982)

A Study of Edge Detection Algorithms*

TAMAR PELI

Lincoln Laboratory, Massachusetts Institute of Technology, Lexington, Massachusetts 02173

AND

DAVID MALAH

Department of Electrical Engineering, Technion-Israel Institute of Technology, Haifa, Israel

Received July 29, 1981; revised October 26, 1981

Edges are image attributes which are useful for image analysis and classification in a wide range of applications. The numerous applications and the subjective approach to edge definition and characterization have promoted the development of a large number of edge detectors (or operators) which may perform well in given applications but poorly in others. In this work we describe a variety of edge detectors and evaluate their performance in terms of some known performance measures. The study is based on computer simulated edges of different shapes, slopes and background noise levels. The performed evaluation together with results in other related works helps to categorize the different edge detection schemes, as well as to better understand the usefulness and limitations of the performance measures used.

1. INTRODUCTION

1.1. Background

An important problem in image processing is the detection of edges in a given image. Since the problem is difficult (to solve and to define) a large number of schemes have been presented in the literature. That information has been summarized in surveys by Davis [1] and in books by Rosenfeld [2] and Pratt [3]. Analysis and comparison of edge detector performance were done by using visual judgment as in Modestino and Fries [4, 5] or by using quantative measures as detailed below. Fram and Deutsch [6, 7] developed two quantative measures: The first reflects the distribution of the detected true edge points (in contrast to false ones which result from noise) along the edge. The second measure can be regarded as a maximum likelihood estimate of the ratio of the total number of true edge points to the total number of detected edge points. Shaw [8] examined some edge detectors by using as a measure the signal-to-noise ratio, with the original edge (for a perfect edge) as the signal and the added edge points in the output due to the distorted image (noisy, blurred, etc.) as the noise. Pratt [3] suggested a measure for evaluating the performance of an edge detector, which corresponds to the weighted and normalized deviation of the real edge from the ideal edge line. Abdou and Pratt [9, 10] presented a statistical analysis of enhancement/thresholding edge detectors and examined the

*This paper is based on the principal author's Master's thesis research conducted at the Technion–Israel Institute of Technology, Haifa. A portion of the work was done at the M.I.T. Lincoln Laboratory with the support of the Department of the Air Force.

The U.S. Government assumes no responsibility for the information presented.

0146-664X/82/090001-21$02.00/0

response of some known edge detectors with the measure described in [3]. Another statistical analysis was done by Panda and Dubitzki in [11].

1.2. Objectives

The first objective of this work was to gather edge detection methods which do not assume any a priori knowledge of the image, such as edge probabilities or knowledge from a previous training set of images, and study their performance. A subset of the following methods was examined:

(a) Roberts' algorithm [12].

(b) Hale's operators [13].

(c) Rosenfeld's algorithms [2, 14, 15, 16].

Roberts' algorithm was examined in other works [9, 10] and was compared to a number of edge detectors; Rosenfeld's algorithms were examined and compared to some other edge detection methods [6, 7] and to knowledge of the authors; Hale's operators were not compared to other edge detectors. These algorithms use different window sizes and as a result their performance should be evaluated while bearing in mind that fact. The above choice is expected to achieve a more complete performance categorization of edge detection algorithms. The second objective was to examine the usefulness of existing measures (Pratt [3]).

1.3. Edge Detector Performance Evaluation

In order to compare the performance of edge detectors a decision should be made as to the definition of an edge point. In this work an edge is defined to be an abrupt change of gray levels and its location as the midpoint of the edge slope. This widely used definition of an edge is one of the reasons for not using Fram's and Deutsch's [6, 7] measures. In their work all the points on the slope were defined as the edge. A second reason for not using their measures was that they are most suitable for the type of images used by them, and were difficult to implement for the set of images used in this study. The basic information given by most edge detectors is the "magnitude" of the edge and sometimes its direction. The comparison here was performed only for magnitude detection. The output edge magnitude varies for different operators. To compare different operators, there is a need to bring the output into a standard form. That standard form was chosen to be a binary plane: '1'—the point is an edge point and '0'—it is not. In most of the algorithms the output must be thresholded, but some of them (e.g., Hale's [13]) already use this output form.

Two objects were chosen as test images: A square, which is characterized by long straight edges, and a circle, which has a curved edge. These two objects were constructed with ideal (Fig. 1a), ramp (Fig. 1b), and noisy edges. Two types of noisy edges were examined: binary noise edges [2] (this type of image was included to examine the response to texture edges as well) and edges with additive Gaussian noise (Fig. 1c).

In the following section a brief survey of edge detectors is presented with an emphasis on the algorithms evaluated in this work. The third section describes the various measures used and the fourth describes the tested images. The results of the study are given in Section 5 and a short summary of other results reported in the

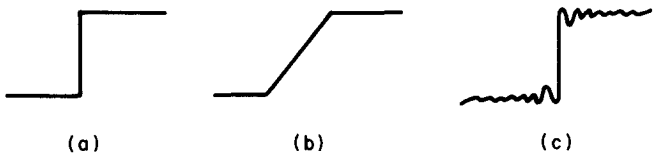

FIG. 1. Different types of edges: (a) ideal; (b) ramp; (c) noisy.

literature is given in Section 6. The last section presents the conclusions on the performance of the edge detectors examined and discusses the measures used.

2. TYPES OF EDGE DETECTORS

Three major types of known edge detection methods are: linear edge enhancement, nonlinear edge enhancement, and the best fit method.

2.1. Linear Methods

A common linear edge enhancement method is differencing. The discrete approximation of the first order derivative in a given direction [2] is one of the simplest methods. In a similar way Pratt [3] defined partial derivatives for the four directions: 0°, 45°, 90°, and 135°. A two-dimensional discrete differencing scheme is suggested by Prewitt [17] based on a set of gradient masks called the compass gradient. The second-order derivative is also used for edge sharpening. A discrete form along a given line is described in [3]. If the direction of the edge is irrelevant the Laplacian, which has a number of representations [2, 3], can be applied in its discrete form. Another operator that uses statistical information about the image is mentioned by Prewitt [17]. There are two operators that use a combination of Gaussian shaped functions; Argyle's operator [18] and Macleod's operator [19].

2.2. Nonlinear Methods

Most of the nonlinear edge detectors use a 2×2 or 3×3 window. One of the popular methods is the gradient which is defined as the maximum over θ_n of the magnitude of the partial derivative in direction θ_n. The gradient has a number of approximations [2]. Roberts [12] uses an approximation, which was simulated in this work and is defined as

$$\max\{|f(i, j) - f(i + 1, j + 1)|, |f(i, j + 1) - f(i + 1, j)|\}. \tag{1}$$

Sobel's operator [2, 20, 21] is also an approximation to the gradient but uses a 3×3 window. Prewitt used the same operator as Sobel except that one of the terms had a different scaling. The algorithms given by Kirsch [22] and Robinson [20] (who presented two methods called 3-level and 5-level) are examples of template matching techniques which use a set of masks to determine the existence of an edge and its direction. Abdou [10] extended the 3-level and 5-level masks to larger window sizes (5×5, 7×7, 9×9). Wallis [3] implemented the Laplacian on the logarithm of the intensity. Smith and Davis [23] developed two operators for binary and gray level images that measure the ratio between the balance of a bi-modal distribution (of binary values) and a measure of "disorder."

A group of operators, introduced by Hale [13], implement a two-dimensional operator by rotating a one-dimensional operator. For the whole group the process of detection has three phases:

(a) *prediction*—computation of a one dimensional function in each direction on part of the image for which a decision has to be made.

(b) *comparison*—between the computed value and a threshold. The result is a Boolean value.

(c) *decision*—according to the result in each direction.

The main difference between the operators in the group is in phase (a). In this work the orientations shown in Fig. 2 were chosen.

One of Hale's operators that was implemented (it is called here "Minmax") is sensitive to the range of the intensity function $f(i, j)$. The operator uses a 9×9 window and is described by the following steps:

STEP 1. For each direction d and for each point (i, j) compute

$$r_d(i, j) = \text{Max}\{f(x, y)\} - \text{Min}\{f(x, y)\},$$

where (x, y) is a point along a line in direction d centered at the point (i, j).

STEP 2. For each point (i, j) and for each direction d compute $R_d(i, j)$ such that

$$\text{if } r_d(i, j) \geq T \quad \text{then } R_d(i, j) = \text{'0'}$$
$$\text{else } R_d(i, j) = \text{'1'},$$

where T—a given threshold,

$R_d(i, j)$—a boolean variable.

STEP 3. An edge exists at the point (i, j) if $[R_D(i, j) \prod_{k \neq D} \bar{R}_k(i, j)] = \text{'1'}$ and its direction is D.

From the above three steps we can see that when the gray level is almost constant in one direction and changing in the others an edge is declared. A disadvantage of the operator is that it cannot detect low-contrast objects. The reason for that is that T—the threshold—should be higher than the maximum amplitude of the noise.

A second Hale's operator, which was implemented in this work, is sensitive to the variance of the intensity function (it is called here "Variance"). The only difference from the previous operator is that instead of $r_d(i, j)$ we compute $S_d^2(i, j)$, which is given by

$$S_d^2(i, j) = \left[\sum_{(x, y) \in d} \left(f(x, y) - \bar{f}_d \right)^2 \right] \Big/ (2N + 1), \tag{2}$$

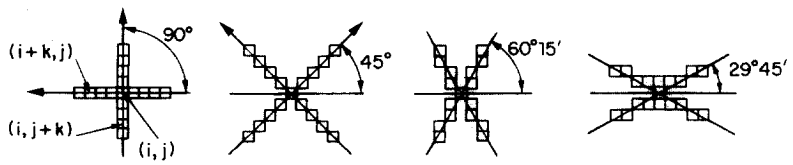

FIG. 2. The eight directions chosen for the one-dimensional operators.

where

$$\bar{f}_d = \left[\sum_{(x,y)\in d} f(x, y) \right] \Big/ (2N + 1),$$

(x, y)—a point along the direction d,

$2N + 1$—number of points taken along the direction d.

In practice the absolute value of $(f(x, y) - \bar{f}_d)$ can be used instead of its square.

A third Hale's operator is sensitive to the curvature of the intensity function. It is similar to the former operator with the only difference that instead of $r_d(i, j)$ one computes $L_d(i, j)$, the general formula of which is given by

$$L_d(i, j) = \sum_{k=-N}^{k=+N} a_k f(k), \tag{3}$$

where the a_k's are constants which satisfy

(a) $\sum_{k=-N}^{k=+N} a_k = 0,$

(b) $a_k = a_{-k},$

and $f(k)$ is a point along the direction d. Condition (a) ensures zero response for a constant region, and the meaning of condition (b) is that the operator is symmetric.

In this work three versions of this operator were used.

VERSION 1 (called "Curvature (1)"). A 1-D difference operation which uses a 3×3 window with

$$a_0 = 2, \qquad a_1 = a_{-1} = -1.$$

VERSION 2 (called "Curvature (2)"). A 1-D second-order derivative [13] which uses a 5×5 window with

$$a_0 = -2/7 \quad , \qquad a_1 = a_{-1} = -1/7, \qquad a_2 = a_{-2} = 2/7.$$

VERSION 3 (called "Curvature (3)"). An optimal 1-D first-order smoothed derivative (a detailed description is given in Appendix A), which uses a 9×9 window and is defined by

$$a_0 = 0, \qquad a_1 = -a_{-1} = 1, \qquad a_2 = -a_{-2} = 0.4,$$

$$a_3 = -a_{-3} = 0.059, \qquad a_4 = -a_{-4} = 0.026.$$

Condition (b) is not satisfied in this version; instead, a condition of antisymmetry exists, which is characteristic of first-order derivatives. In this work the first and second operators ("Minmax" and "Variance") use all eight directions and the third, in its three versions, uses only four directions: vertical, horizontal, and the two diagonals.

Rosenfeld suggested an algorithm that can be implemented in three ways [2, 14–16]. One of them was implemented in this work and will be described. The general idea is that of computing differences between averages of nonoverlapping

neighborhoods that meet at the same point. The details of the algorithm used are as follows:

STEP 1. For each point (i, j) compute the average of the neighborhood centered at the point.

$$f^{(r)}(i, j) = \left[\sum_{m=i-r}^{m=i+r} \sum_{n=j-r}^{n=j+r} f(m, n) \right] \Big/ (2r + 1)^2, \tag{4}$$

where $(2r + 1) \times (2r + 1)$ is the size of neighborhood used, and is an integer.

STEP 2. For each point (i, j) compute the difference between two adjacent neighborhoods that meet at the point.

$$\begin{aligned} \text{Horizontal difference} = L^{(r,H)}(i, j) &= |f^{(r)}(i + r, j) - f^{(r)}(i - r - 1, j)|, \\ \text{Vertical difference} = L^{(r,V)}(i, j) &= |f^{(r)}(i, j + r) - f^{(r)}(i, j - r - 1)|. \end{aligned} \tag{5}$$

The detection can be made by using only horizontal and vertical differences; in this work, however, the two diagonal differences were also used. The computation can be done in two ways: rotating the image by 45°, computing horizontal and vertical differences and rotating the result back; or, as done in this work, defining the two diagonal differences as in (6),

$$\begin{aligned} L^{(r,45°)}(i, j) &= |f^{(r)}(i + r, j + r) - f^{(r)}(i - r - 1, j - r - 1)|, \\ L^{(r,135°)}(i, j) &= |f^{(r)}(i + r + 1, j - r + 1) - f^{(r)}(i - r, j + r)|. \end{aligned} \tag{6}$$

STEP 3. Repeat STEP 1 and STEP 2 for $r = r_1 + 1, \ldots, r_2$.

STEP 4. For each point and for each direction d compute

$$L^{(d)}(i, j) = L^{(r,d)}(i, j). \tag{7}$$

STEP 5. To each point assign the value

$$\begin{aligned} L(i, j) &= \max_d \{L^{(d)}(i, j)\}, \\ &\text{where } d \in \{0°, 45°, 90°, 135°\}. \end{aligned} \tag{8}$$

STEPS 4–5 are a simple but not completely efficient way to narrow the edge line. Two versions of the algorithm were simulated in this work: one using $r_1 = 0$, $r_2 = 4$ (called "Rosenfeld (0–4)") and the other $r_1 = 1$, $r_2 = 4$ (called "Rosenfeld (1–4)"). The last was chosen because it requires less computation and is expected to have a lower sensitivity to noise because of only using averages of neighborhoods.

2.3. *Best Fit*

The third approach involves the best fit of a function to a given image. One way [2] is fitting to a $n \times n$ neighborhood a surface of degree $m < n^2$. The best fit is formed by minimizing the error between the surface of degree m and the actual image. The m coefficients are a function of the gray levels in the $n \times n$ neighborhood, centered at the point (i, j). The gradient to the surface at its center is the best

approximation to the gradient at the point (i, j) (instead of taking the gradient the Laplacian can be used).

Hueckel's method [24, 25] is based on the idea of finding the best match between part of the image (a circle for the continuous case and a sampled version of it for sampled images) and an ideal 2-D edge. The image and the ideal edge are described by a series of orthogonal functions. Using that description of the image and the ideal edge the parameters are computed.

Abdou [10] suggested an optimal edge fitting algorithm based on the discrete image model, while Hueckel's operator is optimal for continuous images. Modestino and Fries [4, 5] used the idea of best fit in a different way. The Laplacian is a good enhancement method for sharp noiseless images. Since real images are noisy, Modestino and Fries determined a filter such that its operation on the noisy image is the best approximation to the operation of the Laplacian on the ideal image.

In this work the following algorithms were implemented: Roberts; Hale "Minmax," "Variance," "Curvature (1)," "Curvature (2)" and "Curvature (3)"; Rosenfeld (0–4) and (1–4).

3. PERFORMANCE MEASURES

Known performance measures can be divided into two classes: quantitative and qualitative. The quantitative measures considered were:

(a) Percentage of edge points detected on the ideal (desired) edge.

(b) Number of detected edge points which do not coincide with the ideal edge (normalized by the number of points on the edge).

(c) *N*oise-to-*S*ignal *R*atio—defined as the ratio of the number of detected edge points, which do not coincide with the ideal edge, to the number of detected edge points which coincide with the ideal edge.

(d) Mean width of a detected edge—defined as the ratio of the total number of detected edge points to the number of ideal edge points.

(e) Weighted and normalized deviation of an actual edge point from the ideal edge, as defined by Pratt [3]:

$$R1 = \frac{1}{I_N} \sum_{1}^{I_A} \frac{1}{1 + \delta e^2},$$

where

I_A—the actual number of detected edge points.
I_I—the number of edge points on the ideal edge.
I_N—$\max\{I_I, I_A\}$.
δ—scaling constant (taken as $1/9$ in this work).
e—distance between an edge point and the ideal edge.

(f) Average squared deviation of a detected edge point from the ideal edge

$$R2 = \frac{1}{I_A} \sum_{1}^{I_A} e^2.$$

(g) Mean absolute value of the deviation

$$R3 = \frac{1}{I_A} \sum_{1}^{I_A} |e| .$$

Measure (b) gives an indication of the smearing effect (thick edges) for an ideal or ramp edge and the addition of false edge points due to noise in the image.

The second class, the qualitative measures, are mainly suited for nonnoisy images such as ideal and ramp edges. This class includes the following measures:

(i) Type of contour: *P*—perfect edge; *B*—broken edge; *PB*—perfect but broken at critical points. Critical points are those that are not uniquely defined as edge points, like the four corners of a square.

(ii) Single or double edge—a single edge or two separate edges.

(iii) Distortion—inherent distortion emanating from the algorithm itself, as a shift of the edge.

Measure (iii) (distortion) is important when the contour is used for computing the volume of an object.

These two classes were used by us in [26]. To simplify the presentation, the summary of the results, which is given here, include only the following measures: type of contour (*P*, *B*, *PB*—as defined in (i)), *R*1 and *R*2, as defined in (e) and (f), respectively.

4. TEST IMAGES

The comparison of the performance of the different edge detectors examined was based on the results obtained using two objects: a square and a circle. The square is a very simple object. The directions of its edges are examined by all the algorithms (vertical and horizontal), and each edge is long, i.e., does not change its orientation rapidly. The circle is a more complex object. Its edge orientation changes more rapidly and includes directions which are not search directions of some of the algorithms, and hence it is harder to detect.

The above objects were used to generate two sets of images. Each set includes three basic types of edges: ideal edge, ramp edge and noisy edge. The ideal edge is a result of a black object (gray level 15) on a white background (gray level 0). As in [3] the definition of an edge point used in this work is given by the point (x', y') in Fig. 3 (another possible choice is the point (x, y)) for both ideal and ramp edges.

The noisy images include two types of noise: binary noise and Gaussian noise. The binary noise (usually called "salt and pepper" noise) creates an edge between textures. For example, an image with 10% binary noise means that pixels are black

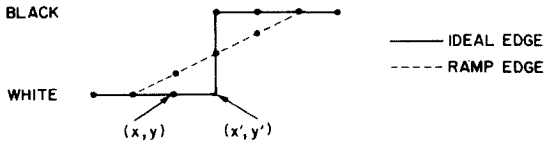

FIG. 3. Position of the edge point for ideal and ramp edges.

or white and are randomly distributed in the image where the probability of a pixel being black in the object is 0.9, and in the background 0.1. In this work, images with 10% and 20% binary noise were examined. For those algorithms which already failed in the 10% binary noise image, the 5% binary noise image was examined.

Gaussian noise images were created from an image with an ideal edge, but with additive zero mean Gaussian noise with standard deviations of 3, 6. 9, and 12. Since only 16 gray levels were used (0–15), all values above 15 or below zero were clipped.

5. TEST RESULTS

For each set of images (ideal, ramp, binary and Gaussian noise) a summary of the results is presented using the following performance measures: type of edge, maximum value of $R1$ and the corresponding $R2$.

5.1. Ideal Edge

Even for an ideal edge most of the algorithms had problems in finding all edge points (especially for the circle). Tables 1 and 2 give a summary of the main measures for each of the algorithms (in Table 2 $\sigma_n = 0$ is the case of an ideal edge).

From Table 1, Table 2 and Fig. 4, we see that $R1$ does not reflect in all cases the performance of an edge detector. In algorithm "Variance" the value of $R1$ is larger for the circle than for the square, although the edge of the square is perfect except for critical points and that of the circle is broken (more details of the reason for such a phenomenon are given in Section 7).

In conclusion we can say that Roberts' algorithm and the two versions of the Rosenfeld algorithm are the best in terms of perfect edge lines for both objects, high value of $R1$ and low value of $R2$.

5.2. Ramp Edge

In general we get a wider edge than in the ideal case. As with the ideal edge, most of the algorithms have problems with the ramp edge in finding the edge of the circle. From Tables 2 and 3 we can see that the best response achieved is by Roberts' algorithm and the two versions of Rosenfeld's algorithm. The response is a perfect edge with a high value of $R1$ and a low value of $R2$. Roberts' algorithm has a perfect

TABLE 1
Type of Contour for an Ideal Edge

Algorithm	Square	Circle
Minmax	*P*	*B*
Roberts	*P*	*P*
Variance	*PB*	*B*
Curvature (1)	*PB*	*B*
Curvature (2)	*PB*	*B*
Curvature (3)	*PB*	*B*
Rosenfeld (0–4)	*P*	*P* or *PB*
Rosenfeld (1–4)	*P*	*P* or *PB*

P—perfect edge; *B*—broken edge, *PB*—perfect but broken at critical points.

TABLE 2

$R1$ and $R2$ for the Various Levels of Gaussian Noise and for a Ramp Edge

Square				$R1$ $R2$		
σ_n / Algorithm	0	3	6	9	12	Ramp Edge
Minmax	0.87	0.89	0.89	0.81	0.49	0.95
	1.52	1.3	1.2	13.3	68.	0.55
Roberts	0.95	0.93	0.57	—	—	0.9
	0.52	0.54	81.8	—	—	1.11
Variance	0.95	0.92	0.91	0.75	0.61	0.93
	0.48	0.9	1.0	30.6	55.3	0.76
Curvature (1)	0.95	0.78	0.46	—	—	0.77
	0.5	32.3	91.6	—	—	2.87
Curvature (2)	0.87	0.80	0.47	—	—	0.75
	1.49	3.7	89.8	—	—	3.64
Curvature (3)	0.95	0.95	0.89	0.55	—	0.94
	0.49	0.47	15.9	85.7	—	0.65
Rosenfeld (0–4)	0.95	0.95	0.95	0.93	0.92	0.95
	0.52	0.52	0.55	0.8	1.0	0.55
Rosenfeld (1–4)	0.95	0.95	0.95	0.94	0.93	0.95
	0.52	0.52	0.54	0.57	0.75	0.54

Circle				$R1$ $R2$		
σ_n / Algorithm	0	3	6	9	12	Ramp Edge
Minmax	0.87	0.93	0.91	0.78	0.42	0.92
	1.44	0.7	0.6	26.5	109.8	0.8
Roberts	0.95	0.94	0.56	—	—	0.89
	0.66	0.6	142.5	—	—	1.35
Variance	0.96	0.96	0.90	0.72	0.52	0.86
	0.32	0.5	6.7	58.0	96.1	1.71
Curvature (1)	0.68	0.61	0.38	—	—	0.88
	0.46	94	166.4	—	—	1.52
Curvature (2)	0.89	0.85	0.38	—	—	0.73
	1.27	12.7	153.9	—	—	3.92
Curvature (3)	0.89	0.75	0.72	0.47	—	0.94
	1.24	0.5	73.1	137.7	—	0.61
Rosenfeld (0–4)	0.93	0.93	0.92	0.90	0.87	0.92
	0.77	0.76	1.0	1.2	1.66	0.9
Rosenfeld (1–4)	0.92	0.91	0.89	0.89	0.87	0.92
	0.84	1.0	1.4	1.4	1.4	0.83

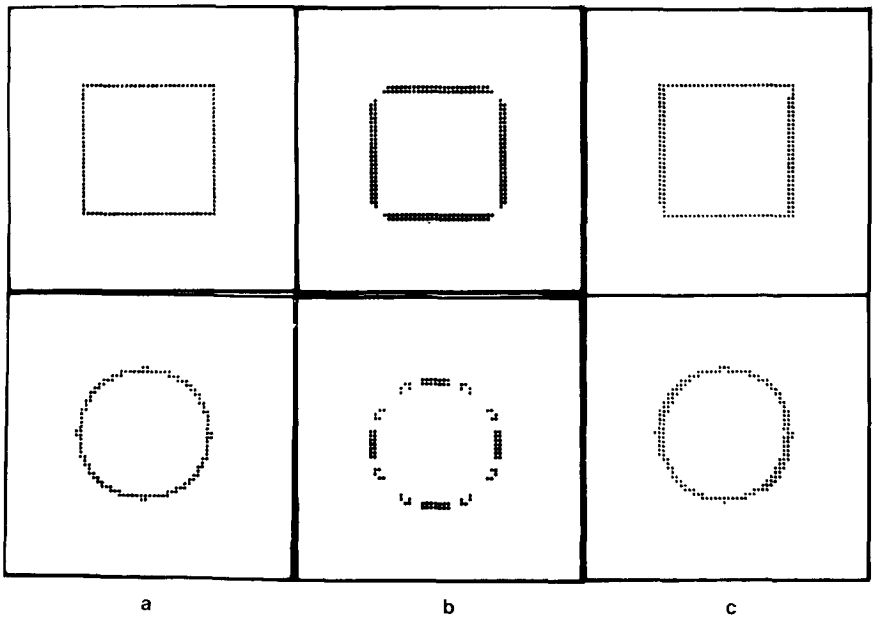

FIG. 4. Response for an ideal edge: (a) Roberts; (b) "Variance"; (c) Rosenfeld (0–4).

edge along the whole range of threshold values, while the algorithms of Rosenfeld have a perfect edge for almost the whole range of threshold values. Only for the very high levels of T, the edge becomes *PB* (perfect but broken at critical points) and then broken. The "Variance" algorithm also has a *PB* edge for the circle, but only at very low values of T. For the ideal image there was no performance difference between the two versions of Rosenfeld's algorithm. For the ramp edge, however, a difference is noticed. Rosenfeld (1–4) responds for higher values of T than Rosenfeld (0–4) and this is a very useful property for edge detection in noisy images. The phenomenon of obtaining a high value of $R1$ for a broken edge is repeated also here for "Curvature (1)" (Table 2 and Fig. 5b).

5.3. Binary Noise Images

Experimentally in this work the point at which an algorithm is declared as failed is when $R1 \leq 0.5$. The starting noise level to be examined is 10%. Those algorithms that fail at that noise level are examined with 5% binary noise. The surviving algorithms from 10% binary noise are also examined in 20% noise level.

The results for 10% binary noise (Table 4) show that Rosenfeld's algorithms and the "Variance" algorithm are capable of finding an edge in this type of image, which can be considered to contain two types of textures. On the other hand Roberts' algorithm has a tendency to enhance noise (Fig. 6b), although visually the edge is quite clear. Because of its tendency to enhance noise, the output at 5% binary noise is also expected to be very noisy. From Table 4 we see that the following algorithms should be examined for 5% binary noise images: "Minmax," "Curvature (1)," "Curvature (2)," and "Curvature (3)" (because a high value of $R2$ means a noisy output).

TABLE 3
Type of Contour for Ramp Edge

Algorithm	Square	Circle
Minmax	*PB*	*B*
Roberts	*P*	*P*
Variance	*PB*	*PB* → *B* (*PB* for low value of *T*)
Curvature (1)	*PB*	*B*
Curvature (2)	*PB*	*B*
Curvature (3)	*P* → *PB*	*B* → *P* → *B* (*P* for small range of *T*)
Rosenfeld (0–4)	*P* → *B* (*B* for high values of *T*)	*P* → *B* (*B* for high values of *T*)
Rosenfeld (1–4)	*P* → *B* (*B* for high values of *T*)	*P* → *PB* → *B* (*B* for high values of *T*)

For images with a 5% binary noise level "Curvature (1)" and "Curvature (3)" show a fair amount of noise immunity, which results in a high value of $R1$ (Table 4). The "Minmax" and especially "Curvature (2)" algorithms are more sensitive to noise in the background and on the edge. This is partly demonstrated in Fig. 6d.

For 20% binary noise there is no use in simulating other than the two versions of Rosenfeld's algorithm and the "Variance" algorithm. Algorithms "Curvature (1)" and "Curvature (3)" have for 10% binary noise a value of $R1$ that is still higher than 0.5 but since the value of $R2$ is very high (higher than 10) it is useless to examine them at a 20% binary noise level. From Table 4 we can see that for the "Variance" algorithm 20% noise is a breaking point. The value of $R1$ is higher than 0.5 but the value of $R2$ is high (12.4 for the square, 42.0 for the circle), and therefore for higher levels of noise it will fail. The two algorithms of Rosenfeld still have a good response (high value of $R1$ and low value of $R2$). As in the 10% binary noise, the edge line of Rosenfeld (1–4) is less noisy than the edge line of Rosenfeld (0–4). This is partly demonstrated in Figs. 6f–g.

5.4. Gaussian Noise

The tested images were ideal edges with different levels of Gaussian noise. The added Gaussian noise was with zero mean and a standard deviation σ_n of 3, 5, 9 and 12. Each of the algorithms was examined starting from the lowest noise level till the point where the value of $R1$ was less or equal to 0.5 (or possibly higher if a high value of $R2$ is obtained (> 10–20) meaning a high level of noise in the output).

For $\sigma_n = 3$, "Curvature (1)" and "Curvature (2)" are already sensitive. The "Variance" algorithm is not sensitive to noise in the background but has some sensitivity to noise on the edge. The shape of the response for some threshold values of "Minmax" and "Curvature (3)" algorithms is very similar to the ideal. However, at low values of T they are sensitive to noise. Robert's algorithm and the two versions of Rosenfeld's algorithm have the best response—a perfect edge, high value of $R1$, low value of $R2$ and a very small amount of noise, or none at all. Again here we can see that the Rosenfeld (1–4) algorithm is less sensitive to noise on the edge than the Rosenfeld (0–4) algorithm. Some of the results are presented in Fig. 7.

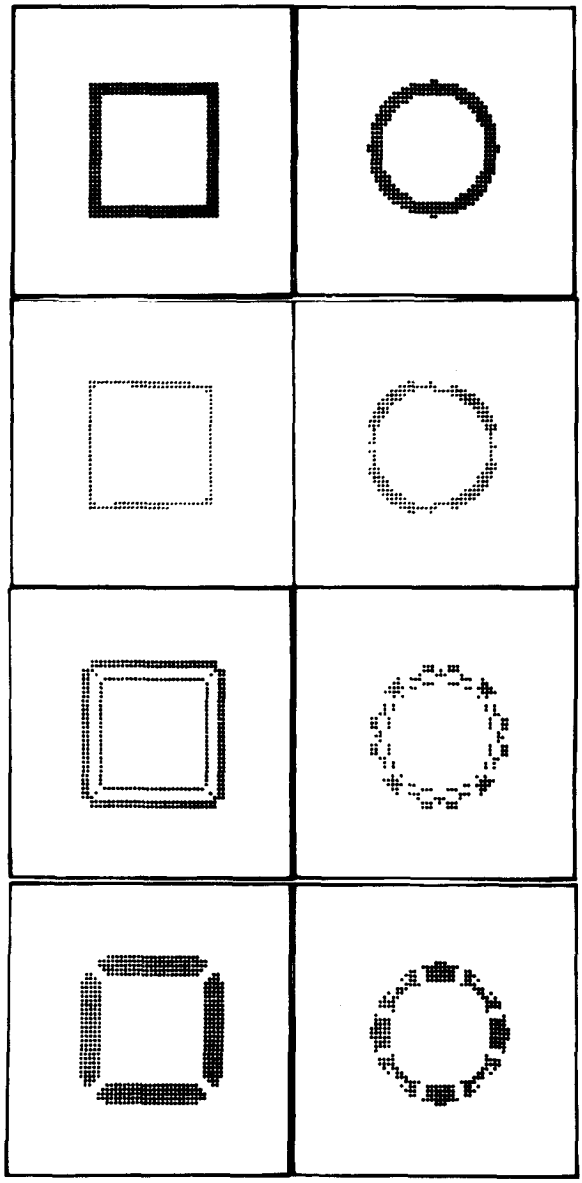

FIG. 5. Response for a picture with a ramp edge: (a) "Variance"; (b) "Curvature (1)"; (c) Rosenfeld (0–4); (d) Roberts.

TABLE 4

Maximum Value of $R1$ and the Corresponding $R2$ for Binary Noise Images

Algorithm \ $R1$ / $R2$	Square			Circle		
Binary noise	5	10	20	5	10	20
Minmax	0.76	0.47	—	0.68	0.32	—
	9.79	75.3	—	24.2	148.1	—
Roberts	—	0.25	—	—	0.21	—
	—	127.1	—	—	193	—
Variance	—	0.92	0.79	—	0.94	0.75
	—	0.9	19.4	—	0.6	42.6
Curvature (1)	0.90	0.69	—	0.78	0.55	—
	8.5	50.3	—	31.2	108.9	—
Curvature (2)	0.70	0.39	—	0.56	0.28	—
	38.2	100.2	—	88.2	171.6	—
Curvature (3)	0.95	0.72	—	0.74	0.59	—
	0.48	47	—	39.46	101.2	—
Rosenfeld (0–4)	—	0.94	0.92	—	0.91	0.88
	—	0.7	1.0	—	1.1	1.5
Rosenfeld (1–4)	—	0.93	0.93	—	0.89	0.87
	—	0.7	0.8	—	1.52	1.7

For $\sigma_n = 6$, Roberts' algorithm has a broken response (for high values of T) or a perfect response with too much noise (for low values of T). $\sigma_n = 6$ is the break point for Roberts' algorithm, because $R1$ equals 0.57 but $R2$ equals 81.8 for the square, and is much higher for the circle. The "Minmax," "Variance," and "Curvature (3)" algorithms start showing some sensitivity to noise at the edge. $\sigma_n = 6$ is also the break point for "Curvature (1)" and "Curvature (2)" algorithms, their response is very noisy and the edge is almost unseen. Again the two algorithms of Rosenfeld have the best response. There is no noise in the background and the edge is thick, noisy but not broken. Also here the edge response for Rosenfeld (0–4) is more noisy than for Rosenfeld (1–4). Some results are shown in Fig. 8.

For $\sigma_n = 9$, "Curvature (3)" has a very noisy response and the edge is almost unseen. The "Minmax" and "Variance" algorithms have a noisy edge line and less noise in the background. The different versions of Rosenfeld's algorithm are still the best. The edge line is noisy, thick but not broken, there is no noise in the background and the noise on the edge is low. Between the two algorithms, Rosenfeld (1–4) is less noisy on the edge than Rosenfeld (0–4). Some results are shown in Fig. 9.

For $\sigma_n = 12$, the "Minmax" algorithm has a very noisy response and the edge is almost invisible. The edge of the "Variance" algorithm is very noisy and broken, but still can be noticed. It is obvious that for $\sigma_n = 15$ it will fail. The two algorithms of Rosenfeld respond with a thick unbroken edge line, high value of $R1$ and low value of $R2$. The Rosenfeld (0–4) algorithm results in a noisier edge than Rosenfeld (1–4), and it has a very small amount of background noise at low values of T. Figure 10 illustrates some of the results.

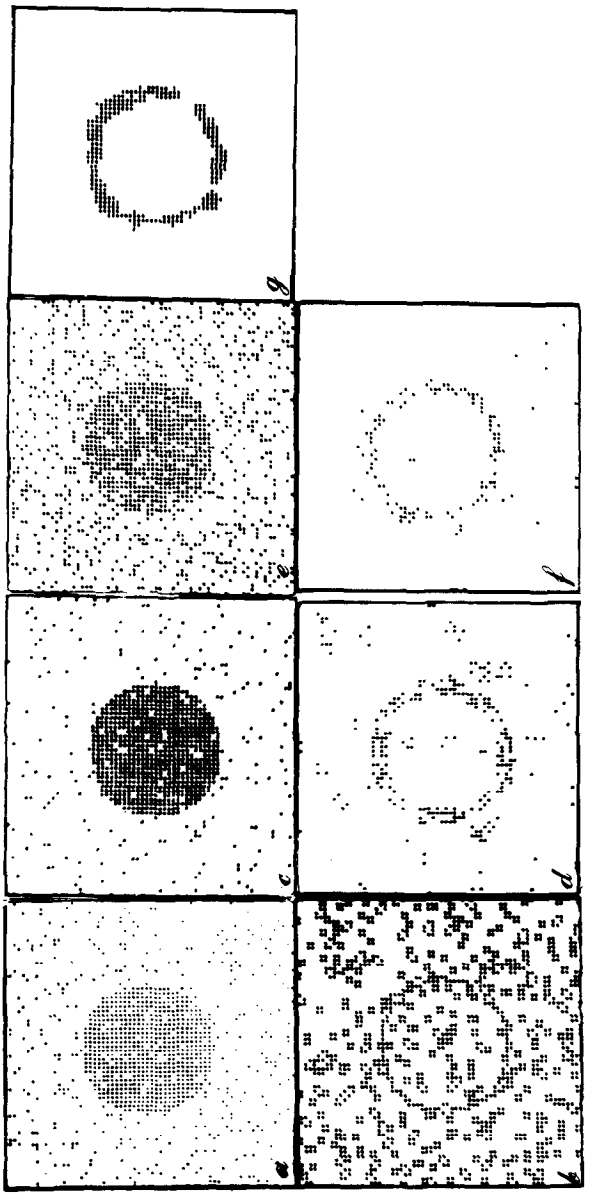

FIG. 6. Response for binary noise. (a) Original noisy image (10%); (b) Roberts' response; (c) original noisy image (5%); (d) "Curvature (2)" response; (e) original noisy image (20%); (f) "Variance" response; (g) Rosenfeld (1–4) response.

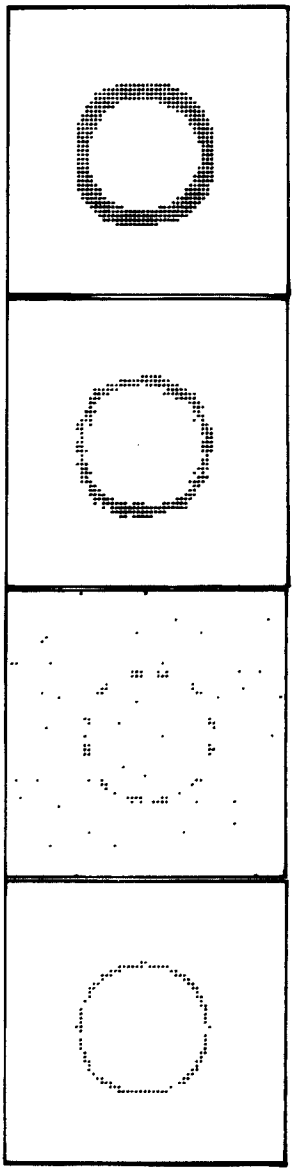

FIG. 7. Response for an ideal edge with Gaussian noise at $\sigma_n = 3$. (a) Roberts; (b) "Curvature (1)"; (c) Rosenfeld (0–4); (d) Rosenfeld (1–4).

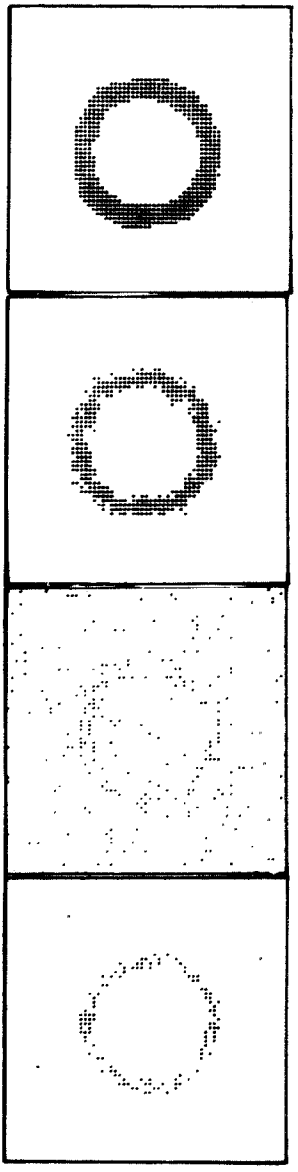

FIG. 8. Response for an ideal edge with additive Gaussian noise at $\sigma_n = 6$. (a) "Variance"; (b) "curvature (2)"; (c) Rosenfeld (0–4); (d) Rosenfeld (1–4).

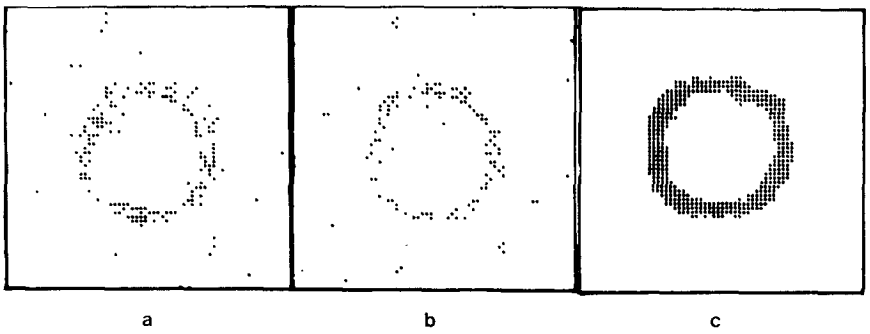

FIG. 9. Response to an ideal edge with additive Gaussian noise at $\sigma_n = 9$. (a) "Minmax"; (b) "Variance"; (c) Rosenfeld (1–4).

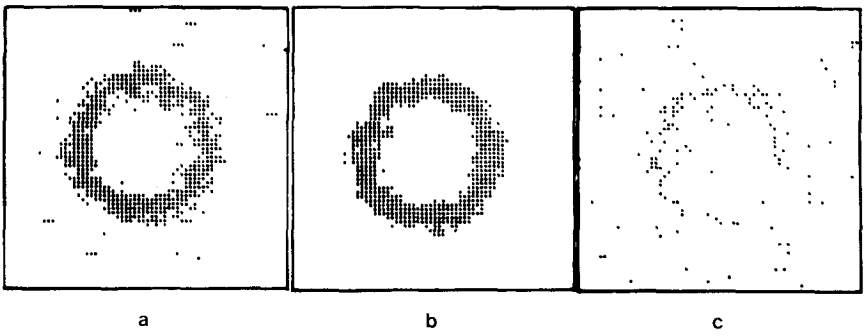

FIG. 10. Response to an ideal edge with additive Gaussian noise at $\sigma_n = 12$. (a) Rosenfeld (0–4); (b) Rosenfeld (1–4); (c) "Variance."

Table 2 summarizes for each algorithm the behavior of the best value of $R1$ and the appropriate $R2$, as a function of σ_n. Figures 11a and 11b show the value of $R1$ as a function of the signal-to-noise ratio (SNR), which is defined as h^2/σ_n^2, where h is the height of the edge (in this work h is equal to 15) for the square and circle images.

6. SUMMARY OF OTHER COMPARISONS

Since other comparisons were reported in the literature, it is useful to summarize the results in order to get an overall perspective. Abdou and Pratt [9, 10] have done a number of comparisons using $R1$ as a performance measure. In the first experiment the following algorithms were examined in the presence of noise on vertical and diagonal edges: 3-level and 5-level (with 3×3, 5×5, 7×7, 9×9 mask sizes), Kirsch's operator, compass gradient, Prewitt's operator (square root, magnitude), Sobel's operator (square root, magnitude) and Roberts' operator (square root, magnitude). The 3-level operator was found to have the best performance in that group. Its performance can be compared only to the performance of Prewitt's operator. As the 3-level and Prewitt operator mask size increases, their noise immunity increases. The extended Prewitt operator and the extended 3-level operator were compared to the 69-point Hueckel operator and found to perform better. The 3-level was also compared to the optimal discrete edge fitting algorithm with the same mask sizes on vertical and diagonal edges. For low SNR and for small mask

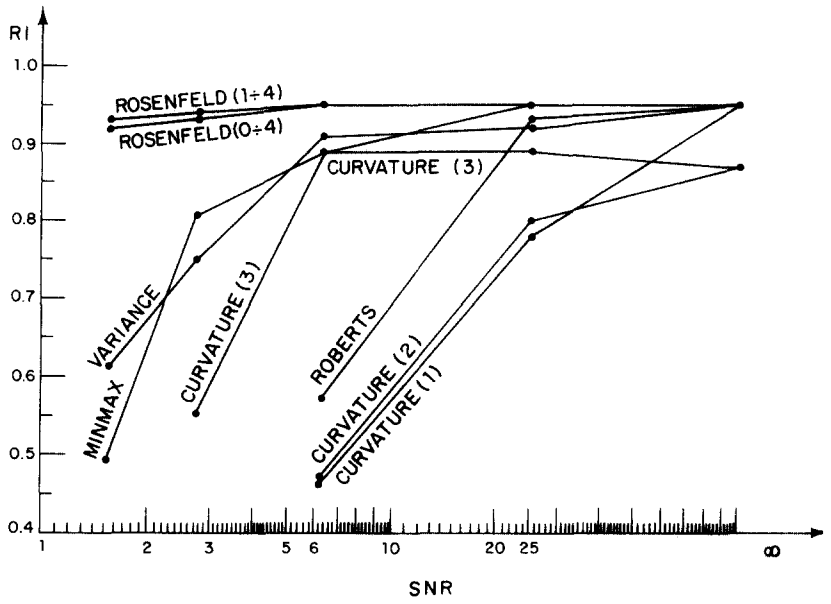

FIG. 11a. $R1$ as a function of signal-to-noise ratio for a square.

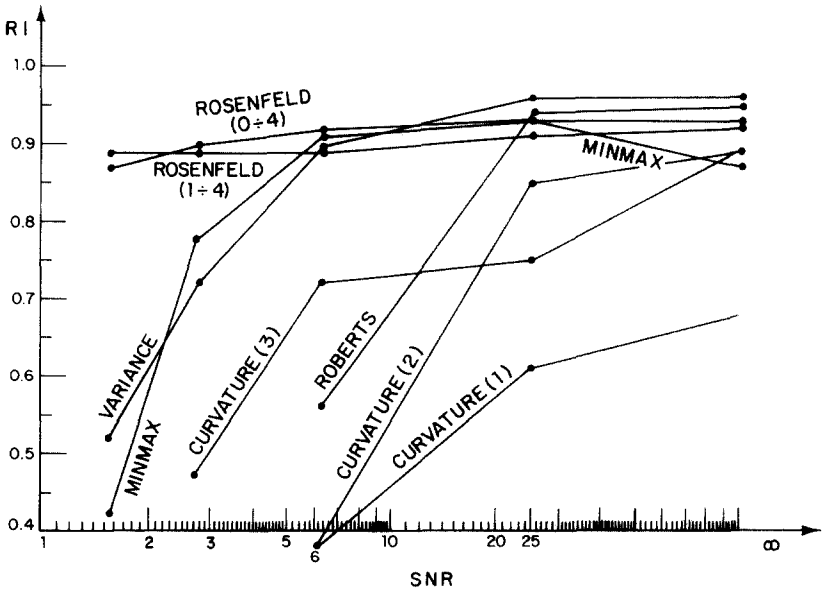

FIG. 11b. $R1$ as a function of signal-to-noise ratio for a circle.

sizes, the edge fitting algorithm is not as good as the 3-level; however, its performance is better for high SNR and for large mask sizes. Abdou and Pratt's last experiment was on pyramid and polynomial operators, Argyle's and Macleod's operators with a 7×7 mask size on only a vertical edge. The conclusion was that all of them have about the same good performance except Macleod's operator which is slightly inferior, but its performance can be improved by changing its parameters.

Modestino and Fries visually evaluated the performance of Hueckel's operator, the four point Laplacian and their algorithm (a Wiener filtering algorithm). It was found that at low to moderate SNR the Laplacian is practically useless. The Wiener filtering algorithm is able to detect a considerable amount of edge structure while preserving a considerable amount of noise immunity. But for a high density of edges, it tends to produce a missing edge structure even at high SNR. Hueckel's operator has a tendency to produce broken lines at a high density of edges but it has a better noise immunity.

Deutsch and Fram [6, 7] evaluated the performance of Hueckel's, Rosenfeld's, and Macleod's operators by using two measures. The first reflects the distribution of the detected true edge points along the edge, and the second is a maximum liklihood estimate of the ratio of the total number of true edge points to the total number of detected edge points. Hueckel's operator had the poorest performance within that group, and had roughly a performance independent of the edge orientation. Rosenfeld's and Macleod's operators were very close in their performance. Macleod's operator had better performance for low-contrast edges while Rosenfeld's operator had less angular bias.

7. CONCLUSIONS

The presented results have shown that in some instances Pratt's figure of merit rating does not provide sufficient information about the performance of the edge detector tested. This is particularly so when the value of $R1$ is high, but visually it is seen that the edge is broken, and the detector should not get such a high figure of merit. In some of the cases the edge of the square was perfect but thick, and its value of $R1$ was much lower than the value of $R1$ for a thick, broken circle edge. The reason is that $R1$ does not take into consideration any information on the distribution of the edge points along the edge. For example edge detection algorithms applied to Fig. 12a, and resulting in Fig. 12b or c, give the same values of $R1$. This kind of information (the distribution of the edge points along the edge) is measured by Fram and Deutsch's first measure. Unfortunately that measure was developed for a specific type of edge (straight lines; the measure estimates the number of rows covered by the edge).

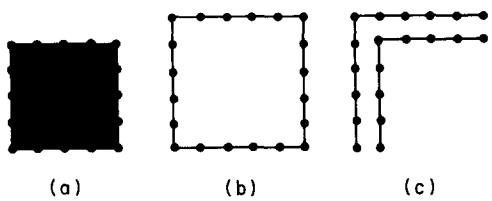

FIG. 12. An object (a) and two different edge detection-outputs (b) and (c) which have the same value of $R1$.

From the simulations in this work we can conclude the following. For an image with a small amount of noise and not too slowly changing ramp edge Roberts' algorithm is the best one to use. This is because of its low number of computations and its capability to detect curved edges. If the image is noisy, the Rosenfeld (1–4) algorithm is particularly suitable because of its high noise immunity (resulting from a large neighborhood) and its ability to detect even curved edges. Still, compared to the other algorithms, the number of computations required by Rosenfeld (1–4) is not much higher and in some cases (the "Variance" algorithm, "Curvature" algorithm, etc.) even lower.

From all this information and the information in Section 6, we can conclude that we need a better measure for edge detection evaluation, and that some of the operators should be examined on a curved edge, including optimal discrete edge fitting (Abdou's version of Hueckel's operator), extended 3-level (9×9) and extended Prewitt (9×9). At present Rosenfeld's algorithm gives the best response, has the ability to detect curved edges and is highly immune to noise.

APPENDIX A

An optimal 1-D first-order derivative was designed for the requirements in the frequency domain (Fig. A1).

The resulting filter in the space domain was windowed to nine coefficients (Fig. A2).

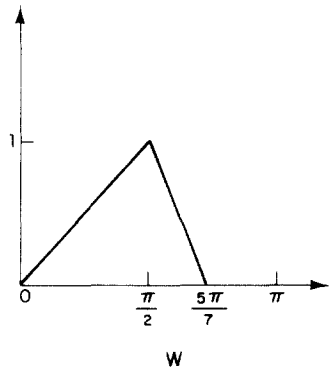

FIG. A1. A first-order smoothed derivative function in the frequency domain.

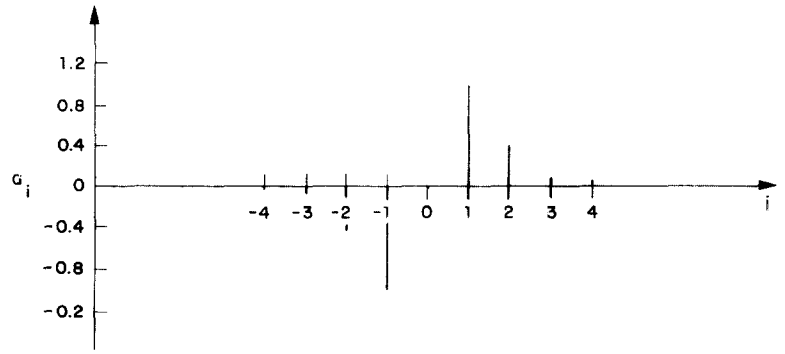

FIG. A2. A first-order smoothed derivative function in the space domain.

REFERENCES

1. L. S. Davis, A survey of edge detection techniques, *Computer Graphics and Image Processing* **4**, 1975, 248–270.
2. A. Rosenfeld and A. C. Kak, *Digital Picture Processing*, Academic Press, New York, 1976.
3. W. K. Pratt, *Digital Image Processing*, Wiley, New York, 1977.
4. R. W. Fries and J. W. Modestino, An Empirical Study of Selected Approaches to the Detection of Edges in Noisy Digitized Images, TR 77-1 Electrical and Systems Engineering Department, Rensselaer Polytechnic Institute, 1977.
5. J. W. Modestino and R. W. Fries, Edge detection in noisy images using recursive digital filtering, *Computer Graphics and Image Processing* **6**, 1977, 409–433.
6. J. R. Fram and E. S. Deutsch, On the quantitative evaluation of edge detection schemes and their comparison with human performance, *IEEE Trans. Comput.* **C-24**, 1975, 616–627.
7. E. S. Deutsh and J. R. Fram, A quantitative study of the orientation bias of some edge detector schemes, *IEEE Trans. Comput.* **C-27**, 1978, 205–213.
8. G. B. Shaw, Local and regional edge detectors: Some comparisons, *Computer Graphics and Image Processing* **9**, 1979, 135–149.
9. I. E. Abdou and W. K. Pratt, Quantitative design and evaluation of enhancement/thresholding edge detectors, *Proc. IEEE* **67**, 1979, 753–763.
10. I. Abdou, Quantitative Methods of Edge Detection, Image Processing Institute, Univ. Southern California, Los Angeles, USCIPI Rep. **830**, 1978.
11. D. P. Panda and T. Dubitzki, Statistical analysis of some edge operators, *Computer Graphics and Image Processing* **11**, 1979, 313–348.
12. L. G. Roberts, Machine perception of three dimensional solids, in *Optical and Electro Optical Information Processing* (J. T. Tippett *et al.*, Eds.), pp. 159–197, M.I.T. Press, Cambridge, Mass., 1965.
13. J. H. G. Hale, Detection of Elementary Features in a Picture by Non-Linear Local Numerical Processing, Proc. Third Int. Joint Conf. on Pattern Recognition, 1976, pp. 764–768.
14. A. Rosenfeld, A nonlinear edge detection technique, *Proc. IEEE*, **58**, 1970, 814–816.
15. A. Rosenfeld and M. Thurston, Edge and curve detection for visual scene analysis, *IEEE Trans. Comput.* **C-20**, 1971, 562–569.
16. A. Rosenfeld and M. Thurston and Y. H. Lee, Edge and curve detection: Further Experiments, *IEEE Trans. Comput.* **C-21**, 1972, 677–715.
17. J. M. S. Prewitt, Object enhancement and extraction, in *Picture Processing and Psychopictorics* (B. S. Lipkin and A. Rosenfeld, Eds.), Academic Press, New York, 1970.
18. E. Argyle, Techniques for edge detection, *Proc. IEEE* **59**, 1971, 285–287.
19. I. D. G. Macleod, Comments on techniques for edge detection, *Proc. IEEE*, **60**, 1972, 344.
20. G. S. Robinson, Edge detection by compass gradient masks, *Computer Graphics and Image Processing* **6**, 1977, 492–501.
21. R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*, Wiley, New York, 1973.
22. R. Kirsch, Computer determination of the constituent structure of biological images, *Computers and Biomedical Research* **4**, 1971, 315–328.
23. M. W. Smith and W. A. Davis, A New Algorithm for Edge Detection, Proc. Second Int. Joint Conf. on Pattern Recognition, 1974, pp. 150–151.
24. M. H. Hueckel, An operator which locates edges in digitized pictures, *J. Assoc. Comput. Mach.* **18**, 1971, 113–125.
25. M. H. Hueckel, A local visual operator which recognizes edges and lines, *J. Assoc. Comput. Mach.* **20**, 1973, 634–647.
26. T. Peli, A Study of Algorithms for Edge Detection in Images, M.Sc. Thesis, Technion–Israel Institute of Technology, Department of Electrical Engineering, June 1979.