# A Differentiation / Enhancement Edge Detector and Its Properties

BHABATOSH CHANDA, STUDENT MEMBER, IEEE,
B. B. CHAUDHURI, MEMBER IEEE, AND D. DUTTA MAJUMDER

*Abstract*—In this report a differentiation / enhancement edge detector for noisy situations is proposed. The definition of the gradient has been initiated from the algorithm for finding a border in a binary picture. The merit of this operator has been compared with those of other widely used operators. A measurement of error in extracting edges by thresholding the gradient has also been suggested, and for the detection of acceptable edges the optimum threshold is chosen corresponding to the minima in the error function.

## I. Introduction

Recognition or classification of pictures by digital computer is generally done by using the information contained in the edges between different regions. Success of the recognition algorithm, to some extent, depends on the edges extracted in the previous stage of processing. Edges may be defined as abrupt changes in characteristic features like, brightness, color, texture, etc. Detection is performed either by frequency or spatial domain techniques [1], [2]. In this report a spatial domain technique has been considered for edge detection which is nothing but the differentiation of the gray level of the pixels in the discrete domain. However, results obtained using this method are very much affected by the noise present in the picture. To solve this problem, a noisy picture is first enhanced by smoothing and then the spatial gradients are computed. Methods proposed by Rosenfeld and Thurston [3], Sobel [4], and Prewitt [5] belong to this class and are considerably immune to spurious noise. Though the above-mentioned operators reduce the effect of noise, they also thicken the edges.

This report presents a conceptually different edge detector for noisy pictures. The new operator can be characterized as a differentiation/enhancement scheme. Here the definition of the gradient is yielded by generalizing an algorithm devised for finding the border of a binary picture using parallel processing. The gradient at selected pixels of the image is enhanced or modified using local properties, so that the effect due to noise is reduced and the edges are kept reasonably thin. In Section II, the border finding algorithm for a binary picture and its evolution to the measure for gradient in a multilevel image is described. The properties of the new operator are presented in Section III and the performance of this operator is compared with that of other widely used operators based on these properties. Another important problem in edge detection is to select a threshold that transforms the gradient picture to a two-level picture containing optimum edges between the regions. In Section IV, selection of such a threshold is given depending on some measures of errors in thresholding. Section V describes the implementation of the algorithm and discusses the results.

## II. Algorithm for Finding the Border in a Binary Image and Its Generalization to Finding the Gradient in a Multilevel Image

Edge detection segments a picture by finding the border between different regions, say, objects and background. Binary (having gray-level one for objects and zero for background) pictures can be described as the optimum representation of such regions with the least ambiguity. In this section, a parallel algorithm for finding the borders in a binary picture is described and then generalized to find the gradient in a multilevel picture.

### A. Algorithm for Border Finding

By the border of a binary picture subset or object $S$, we mean the set $S_b$ of pixels of $S$ that have neighbors in $\bar{S}$ or background. Given the overlay $0_S$ representing $S$, it can be said that $0_S$ is the entire binary picture. Hence, the border-finding algorithm can be described as follows.

*Algorithm 1:*

*Step 1:* Compute the logical exclusive-OR between $0_S(i,j)$ and $0_S(i-k,j-l)$, which is $0_S(i,j)$ shifted by $k, l$. If we consider all the eight-neighbors, values of $k$ will be $-1$, 0, and 1, but both will not be equal to zero. Let us call these pictures $0_S(k,l)$.

*Step 2:* Compute the logical OR of $0_S(k,l)$ for all possible values of $k$ and $l$ (except $k = l = 0$) to obtain $0_{S_e}$, the overlay of $S_e$.

*Step 3:* To obtain $0_{S_b}$, the overlay containing $S_b$, compute the logical AND between $0_{S_e}$ and $0_S$.

### B. Generalization to the Measure of the Gradient

Before the generalizing Algorithm 1 to multilevel pictures, some (natural) algebraic operations that are equivalent to Boolean operations are presented. Using a truth table, it can be verified easily that

$$a \text{ AND } b \equiv \min\{a,b\}$$
$$a \text{ OR } b \equiv \max\{a,b\}$$
$$a \text{ exclusive-OR } b \equiv |a-b|$$
$$\bar{a} \equiv m - a,$$

where $m$ is the maximum value that $a$ can attain. Hence, using these equivalent operations Algorithm 1 can be expressed for a multilevel picture $f(i,j)$ as given below.

*Algorithm 2:*

*Step 1:* Compute $|f(i,j) - f(i-k,j-l)|$ and denote it by $f_{k,l}(i,j)$, where $k = -1, 0, 1$ and $l = -1, 0, 1$ (except $k = l = 0$).

*Step 2:* Compute $g(i,j) = \max_{k,l}\{f_{k,l}(i,j)\}$.

*Step 3:* Find $b(i,j)$ analogous to $0_{S_b}$ in Algorithm 1 by computing $\min\{g(i,j), f(i,j)\}$.

Since there exist some basic differences in characteristics of binary and multilevel pictures, algorithms developed for them should also reflect the same. Algorithm 1 extracts a set of pixels whereas Algorithm 2 finds the gradient of graylevels. So for convenience, Step 3 should be omitted from Algorithm 2. Now the outcome of this algorithm will be equivalent to the outcome of Step 2 of Algorithm 1, which gives the edge, $S_e$, of thickness two. $S_e$ consists of the set of pixels, which are $S$ border of $\bar{S}$ as well as, the set of pixels, which are $\bar{S}$ border of $S$. Thresholding of $g(i,j)$, similarly, will try to give edges of thickness two. To obtain the edges of thickness one, any three consecutive shifts, only one of which is a diagonal shift, can be used instead of eight shifts as described in Step 1.

However, for some noisy situations in multilevel pictures extracted edges may become broken or discontinuous. Since in the next stage of processing, the broken edge results in more trouble than a thicken one, all the eight shifts have been used to define the gradient. Hence, in short, the gradient $g(i,j)$ for a multilevel picture can be defined as

$$g(i,j) = \max_{k,l}\{|f(i,j) - f(i-k,j-l)|\} \tag{1a}$$

Manuscript received May 22, 1984; revised September 24, 1984.
The authors are with the Electronics and Communication Science Unit, Indian Statistical Institue, Calcutta, 70035, India.

0018-9472/85/0100-0162$01.00 ©1985 IEEE

where, $k, l \in \{-1, 0, 1\}$, or in a more general way

$$g(i,j) = \max_{u,v}\{|f(i,j) - f(u,v)|\} \tag{1b}$$

where $(u, v)$ is the neighboring pixel of $(i, j)$.

### C. Enhancement of the Gradient

Now the gradient $g(i, j)$ is to be modified to make it immune to spurious noise. Algorithm 3 is developed to achieve the required modification using some local properties.

Initially a coarse threshold $T_c$ is chosen to reduce the number of computations required for Algorithm 3 considerably. It is inferred that the pixels $(i, j)$, for which $g(i, j) < T_c$, cannot lie on the edges. Let us define a set $A$ that contains the gray levels of all the neighbors $(u, v)$ of the candidate pixel $(i, j)$; that is

$$A = \{a_m | a_m = f(u,v), \text{ for } m = 1, 2, \cdots M\}. \tag{2}$$

*Algorithm 3:*

*Step 1:* If $g(i, j) < T_c$, go to Step 5.
*Step 2:* $R(i, j) \leftarrow \max_m\{a_m\} - \min\{a_m\}$.
*Step 3:* If $R(i, j) = 0$, go to Step 5.
*Step 4:* $\mu(i, j) \leftarrow 1/M \sum_{m=1}^{M} a_m$
$\sigma(i, j) \leftarrow 1/M \sum_{m=1}^{M} |a_m - \mu(i, j)|$
$w(i, j) \leftarrow 2\sigma(i, j)/R(i, j)$
$G(i, j) \leftarrow w(i, j) g(i, j)$
go to Step 6.
*Step 5:* $G(i, j) \leftarrow 0$.
*Step 6:* Output $G(i, j)$.

It will be shown later that the value of $w(i, j)$ varies between 0.75 and 1.0 for a step edge and is much less than 0.75 when isolated spurious noise induces false edge points.

From Algorithm 2 and Algorithm 3 it is clear that the technique to obtain $G(i, j)$ is hybrid in nature, so it will be called the *hybrid operator*.

## III. Quantitative Comparison of Edge Detectors

In this section, the performance of the hybrid edge detector with those of some other widely used edge detectors due to Rosenfeld and Thurston and Sobel and Prewitt will be compared. All these operators are more or less immune to noise. Before finding differences, these popular edge detectors enhance the picture by smoothing. The Sobel and Prewitt operators smooth the picture over a $3 \times 1$ neighborhood. The Rosenfeld and Thruston operator does the same over an $I \times I (I = 3, 5, 7, \cdots)$ neighborhood. Increasing mask size decreases the noise sensitivity because of the inherent noise averaging performed by the operator. However, use of a large size of neighborhood is associated with the performance penalty which makes the edges thick. So we have taken here $I = 3$.

Now

$$G_1 = \frac{1}{2+w}|f(i-1,j+1) + w \cdot f(i,j+1) + f(i+1,j+1) - f(i-1,j-1) - w \cdot f(i,j-1) - f(i+1,j-1)|$$

$$G_2 = \frac{1}{2+w}|f(i+1,j-1) + w \cdot f(i+1,j) + f(i+1,j+1) - f(i-1,j-1) - w \cdot f(i-1,j) - f(i-1,j+1)|. \tag{3}$$

For the Sobel operator $w = 2$ and for the Prewitt operator $w = 1$. For the operator proposed by Rosenfeld and Thurston (where $I = 3$)

$$G_1 = \frac{1}{9}\left|\sum_{l=1}^{3}\sum_{k=1}^{3} f(i-2+k, j-1+l) - \sum_{l=1}^{3}\sum_{k=1}^{3} f(i-2+k, j-3+l)\right|$$

$$G_2 = \frac{1}{9}\left|\sum_{l=1}^{3}\sum_{k=1}^{3} f(i-1+k, j-2+l) - \sum_{l=1}^{3}\sum_{k=1}^{3} f(i-3+k, j-2+l)\right| \tag{4}$$

Ultimately the gradient $G(i, j)$ is obtained by root mean square or average or maximum of $G_1$ and $G_2$ as given in (3) and (4). Here the max operator is used; i.e., $G(i, j) = \max\{G_1, G_2\}$, to achieve a meaningful comparison with the hybrid operator. Because of the nature of the point operator, $G_1$ and $G_2$ (in (3) and (4)) can be defined interchangably without affecting the value of $G(i, j)$. An edge is deemed present if $G(i, j)$ exceeds a predefined threshold, say, $T$.

### A. Edge Orientation and Displacement Sensitivity

Desirable properties of an edge detector are 1) amplitude response invariance to the choice of origin or reference axes; 2) amplitude response invariance to edge orientation; and 3) amplitude of the edge detector response where it decays rapidly as the edge moves away from the center of the mask. All of the edge detectors considered here possess the first property. Regarding other properties, the edge detectors' merits are compared for an ideal noise-free step edge. Fig. 1 shows one such edge passing through the center of a $5 \times 5$ mask and is inclined to the vertical axis by an angle $\theta$. Fig. 2 shows the variation of response with the edge orientation, that is, $\theta$ for different operators. It is seen from the figure that the variation in amplitude for the hybrid operator is much less than those for other operators. Fig. 3 presents models of (a) vertical and (b) diagonal edge displaced from the center of the mask. Displacement sensitivity for different operators is shown in Fig. 4. Fig. 4 reveals that the amplitude response for the hybrid operator is maximum when the edge passes through the boundary of two adjacent pixels and decays more rapidly than the amplitude response curves for the other operators as the edge recedes from the boundary in either direction. This characteristic is desirable for the extraction of weak edges or streaks.

### B. Noise Sensitivity

Another important property that an edge detector should have is low sensitivity to random noise. The study of this property is complicated by the fact that the edges depend on the choice of appropriate operator (root mean square or magnitude average or maximum of the magnitude) as well as the threshold. For noisy images the threshold selection becomes a trade-off between the missing of valid edges and creation of noise-induced false edges. Abdou and Pratt [6] have studied the noise sensitivity of some widely used operators using statistical analysis. However, this type of analysis is not followed here for the reason that the notion of valid edges is somewhat fuzzy, and the conditional probability densities $p(G|\text{ edge})$ and $p(G|\text{ no edge})$ are to be modeled rather than estimated.

We have already said that the edge detectors proposed by Rosenfeld and Thurston and Sobel and Prewitt reduce the effects of spurious noises by smoothing over a predefined neighborhood. So noise sensitivity of the hybrid operator will be considered, in this subsection, based on the risk of missing valid edge pixels and including false edge pixels induced by noise. Consider a mask that contains $(M + 1)$ pixels within its area. Among these a

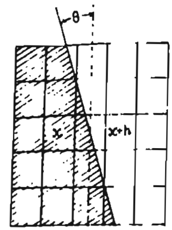



Fig. 1. Model of an ideal step edge passing through the center of mask and having inclination θ with vertical axis.

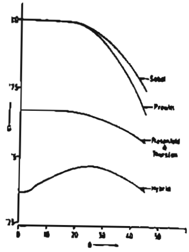



Fig. 2. Amplitude response of different operators to inclination θ in max sense.

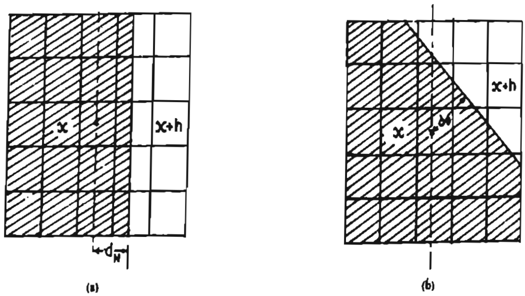



Fig. 3. Models of step edge displaced from the center of mask. (a) vertical step edge. (b) diagonal step edge.

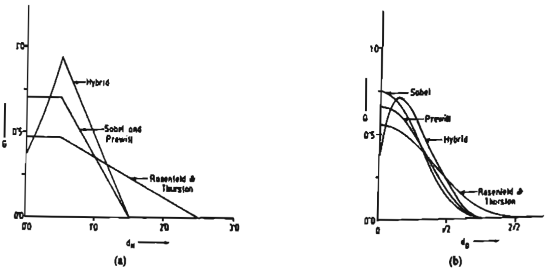



Fig. 4. Displacement sensitivity of different operators. (a) For a vertical edge. (b) For a diagonal edge.

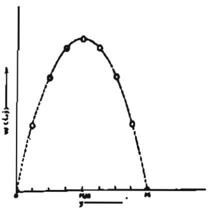


Fig. 5. Plot of weighting factor $w(i, j)$ versus $y$(see text).

candidate pixel and $y$ number of pixels have gray-level $a$ and the rest $(M - y)$ number of pixels have gray-level $a + h$. It can be readily assumed that if any edge really exists in the vicinity of the candidate pixel then (for $M \geq 2$)

$$\frac{1}{3} \leq \frac{Y}{M-y} \leq 3.$$

Now the amplitude of the hybrid edge detector can be found as follows:

$$g(i,j) = |a + h - a| = |h|$$

$$\mu(i,j) = \frac{ya + (M-y)(a+h)}{M}$$

$$\sigma(i,j) = \frac{1}{M}[|\mu(i,j) - a|y + |\mu(i,j) - a - h|(M-y)]$$

$$= \frac{2y(M-y)}{M^2}|h|$$

$$R(i,j) = h$$

$$w(i,j) = \frac{2\sigma(i,j)}{R(i,j)} = \frac{4y(M-y)}{M^2}.$$

Hence, the gradient, $G(i, j)$ finally can be given as

$$G(i,j) = \frac{4y(M-y)}{M^2}|h|.$$

Fig. 5 represents the variation of the weighting factor $w(i, j)$ with $y$. It is seen from the figure that the value of $w(i, j)$ decays very quickly outside the interval $1/3 \leq y/(M - y) \leq 3$. Hence it can be inferred that for isolated spurious noises, the value of $w(i, j)$ will be very low and consequently the effect of noise on $G(i, j)$. The amplitude of the hybrid edge detector will also be sufficiently reduced.

## IV. Selection of Optimum Threshold

Another basic problem of edge detection in case of multilevel pictures is to select the value of the threshold that transforms gradients of the picture onto another picture. That is

$$\mathcal{T}: G(i,j) \rightarrow B(i,j)$$

$\mathcal{T}$ denotes the thresholding operator, and $B(i, j)$ is a binary picture where picture subset $S_e$ contains only edge points of $f(i, j)$. Operator $\mathcal{T}$ satisfies the following two conditions

a) $\mathcal{T}$ is not invertible since it is not one-one.
b) $\mathcal{T}$ can take any value $T_k$ as threshold from the interval

$$\left[\min_{i,j}\{G(i,j)\}, \max_{i,j}\{G(i,j)\}\right].$$

Edges between the regions (that is, subset $S_e$ of $B(i, j)$) can be extracted in three different ways. 1) If the shape of the picture subset $S_e$ of $B(i, j)$ is known (that means shape or type of the objects whose edges are to be extracted are known] *a priori*, then $T_k$ can be chosen in an interactive way so that subset $S_e$ of $B(i, j)$ represents the nearest approach to the ideal edge. 2) If threshold $T_k$ is known *a priori*, then the picture subset $S_e$, obtained by the thresholding operation, gives the edges of the objects whatever they might be. This approach is used when edges of the objects from a large class of similar type pictures are to be extracted. 3) If neither the shape of $S_e$ nor the threshold $T_k$ is known *a priori*, then $T$ is so chosen such that $S_e$ satisfies some properties most closely.

*Notations:*

| | |
|---|---|
| $\{T_k, k = 1, 2, \cdots r_1\}$ | A nonempty set of threshold values. |
| $\{P_l, l = 1, 2, \cdots r_2\}$ | A nonempty set of properties that an ideal edge should possess. |
| $E_{P_l}(T_k)$ | An error-function characterizing how loosely $S_e$ satisfies the property $P_l$ for the threshold $T_k$. |
| $\mathcal{E}(T_k)$ | Some convex combination of $E_{P_1}(T_k)$, $E_{P_2}(T_k), \cdots E_{P_{r_2}}(T_k)$. |

Hence, $T_o$ can be taken as optimum threshold if

$$\mathcal{E}(T_o) = \min_k \{\mathcal{E}(T_k)\}$$

and picture subset $S_e$ is accepted as an optimum edge.

### *A. Implementation*

Here the third approach is implemented to select the optimum threshold. It is an adaptive technique using an incomplete set of properties.

- $P_1$ Edge or picture subset $S_e$ should be of thickness one.
- $P_2$ A string of pixels in $S_e$ takes a spatial turn of $0°$ or $\pm 45°$ at most of the elements (pixels).

Let a pixel $(i, j)$ and its neighboring $n$ pixels are first selected as elements of $S_e$ due to thresholding $T_k$. For thin and connected edges (except the cases when $(i, j)$ is a terminal pixel or it lies on the boundary of the picture frame or a genuine bifurcation of edge occurs at $(i, j)$) the value of $n$ will be $n_0$ (say). For eight-connectivity, $n_0$ is equal to two and for four-connectivity, $n_0$ may be two, three, or four. Here we use eight-connected definition of edges, so a measurement of error can be given by $|n - 2|$. Since a discontinuity in an edge causes a more severe error than the thickness of it in the context of preservation of information, the error due to the property $P_1$ can be expressed as

$$E_{P_1}(T_k) = \sum_{(i,j) \in S_e} q^n |n-2|$$

where $q$ is a constant and less than one.

The property $P_2$ states that the distance between at least one pair of pixels, from the set of $n$ neighboring pixels, will be two. The notion of error measurement due to this property can be formulated as

$$E_{P_2}(T_k) = \sum_{(i,j) \in S_e} \left[2 - \max_{(i_1,j_1),(i_2,j_2)} \{|i_1 - i_2|, |j_1 - j_2|\}\right]$$

where $(i_1, j_1)$ and $(i_2, j_2)$ are the elements of the set of $n$ neighboring pixels.

Finally, the error-function $\mathcal{E}(T_k)$ takes the form

$$\mathcal{E}(T_k) = \frac{\lambda E_{P_1}(T_k) + (1-\lambda) E_{P_2}(T_k)}{n_t} \times 100$$

where $n_t$ is the total number of elements in $S_e$ and $0 < \lambda < 1$. It is observed that the plot of $\mathcal{E}(T_k)$ versus $T_k$ takes the shape of a convex function. The value of $T_k$ that corresponds to the valley of the plot has been taken as the optimum threshold.

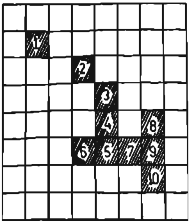


Fig. 6. An example of subset $S_e$ whose elements are indexed.

TABLE I

| Index of the Pixel | $n$ | $0.93^n\|n-2\|$ | $2-\max_{(i_1,j_1),(i_2,j_2)}(\|i_1-i_2\|,\|j_1-j_2\|)$ |
|---|---|---|---|
| 1 | 0 | 2.00 | 2 |
| 2 | 1 | 0.93 | 2 |
| 3 | 2 | 0.00 | 0 |
| 4 | 4 | 1.496 | 0 |
| 5 | 3 | 0.804 | 0 |
| 6 | 2 | 0.00 | 1 |
| 7 | 5 | 2.088 | 0 |
| 8 | 2 | 0.0 | 1 |
| 9 | 3 | 0.804 | 0 |
| 10 | 2 | 2.00 | 1 |
| | | $E_{P_1}(T_k)=10.122$ | $E_{P_2}(T_k)=7$ |

## B. Examples

Let us consider an example that shows how we compute errors $E_{P_1}(T_k)$ and $E_{P_2}(T_k)$ in practice. Before proceeding, we choose the value of $q$ which will be required to compute $E_{P_1}(T_k)$. Since $E_{P_1}(T_k)$ increases with the difference between the number of neighboring pixels $n$ and $n_0$, so $q$ must satisfy the following conditions:

$$q^{n_0}|n_0-n_0| < q^{n_0-1}|n_0-1-n_0| < q^{n_0-2}|n_0-2-n_0| < \cdots < q^0|0-n_0|$$

and

$$q^{n_0}|n_0-n_0| < q^{n_0+1}|n_0+1-n_0| < q^{n_0+2}|n_0+2-n_0| < \cdots < q^8|8-n_0|. \quad (5)$$

The value of $q$ can be found iteratively using inequations (5). In our case, $n_0=2$. This leads us to pick up $q=0.93$ as one of the possible solutions of inequation (5). Now consider Fig. 6, where indexed pixel represent the elements of $S_e$ obtained due to the threshold $T_k$. For the pixel with index 2, the error due to property $P_1$ is $0.93^1|1-2|=0.93$ and the error due to property $P_2$ is 2 (since, it has only one neighbor; $\max_{(i_1,j_1)(i_2,j_2)}\{|i_1-i_2|,|j_1-j_2|\}$ cannot be computed and is assumed to be zero). Similarly for the pixel with index 9 having three neighbors with index 7, 8, and 10, error due to property $P_1$ is $0.93^3|3-2|=0.804$. Now the checker-board distance between the pixels with index 7 and index 8 is 1, with index 8 and index 10 is 2, and with index 7 and index 10 is 1; therefore, the error due to property $P_2$ is $2-\max\{1,2,1\}=0$. $E_{P_1}(T_k)$ and $E_{P_2}(T_k)$ are obtained by adding the errors for all the pixels due to properties $P_1$ and $P_2$, respectively. Table I summerizes errors due to different properties at different pixels.

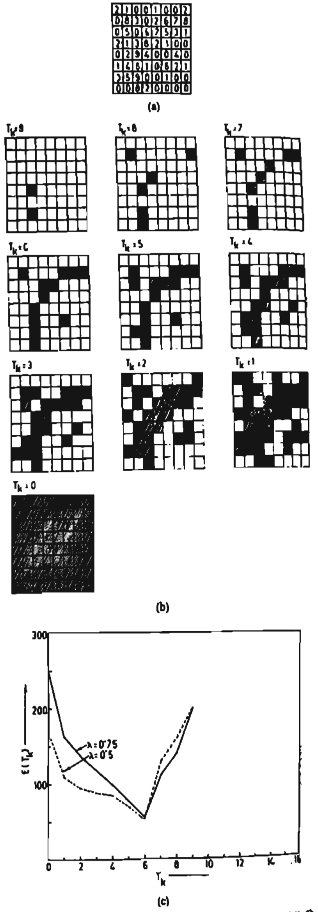



Fig. 7(a). An array of integer numbers representing $G(i,j)$. (b) $B(i,j)$'s obtained by thresholding $G(i,j)$ of Fig. 7(a) at different values of $T_k$. (c) Plot of $\mathscr{E}(T_k)$ versus $T_k$ corresponding to $B(i,j)$'s of Fig. 7(b) for $q=0.93$ and for different values of λ.

Consider another example to show the shape of the plot of $\mathscr{E}(T_k)$ versus $T_k$ and also to show the effect of λ on the selection of optimum threshold $T_e$. Fig. 7(a) shows an array of integer number representing $G(i,j)$ (where $0 \leq G(i,j) \leq 15$, for all

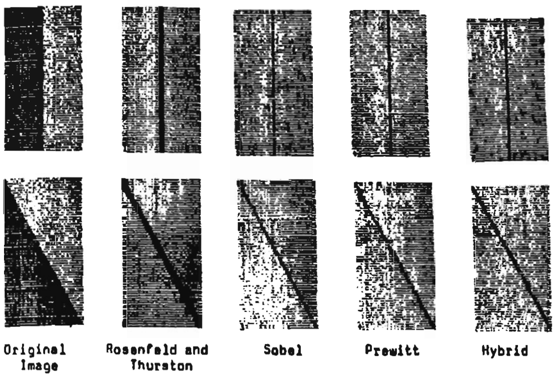



Fig. 8. Gradient picture obtained by using different operators for vertical and diagonal edges.

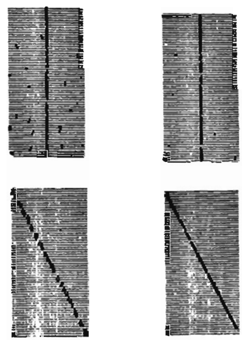
$i, j$). Fig. 7(b) gives the corresponding $B(i, j)$ for different values of $T_k$. The plots of $\mathscr{E}(T_k)$ versus $T_k$, for $\lambda = 0.5$ and $\lambda = 0.75$ are shown in Fig. 7(c). Here also $q = 0.93$ is taken. It is seen that for both the values of $\lambda$ minima of the plots appear at the same $T_k$ [i.e., $T_m$]. This indicates that the selection of optimum threshold is not much sensitive to the value of $\lambda$.

## V. Results and Discussion

To test the effectiveness of the hybrid operator as an edge detector, it has been applied to two artificial pictures having vertical and diagonal step edges, respectively. Other popular edge detectors are also applied to the same pictures for comparison of their merits. The algorithms have been simulated in a general purpose EC 1033 computer using the Fortran language. Both the input and output hard copies of the pictures were generated by overprinting of common characters on a computer line printer. Gradient pictures obtained from using different operators are shown in Fig. 8. Edges obtained due to the operator proposed by Rosenfeld and Thurston are least sensitive to noise but the edges are certainly thicker than those obtained by other operators. Regarding noise sensitivity, performance of the hybrid operator is superior to that of the Sobel operator and inferior to that of the Prewitt operator. Abdou and Pratt [6] have shown that the merit of the Prewitt operator is better than that of the Sobel operator for vertical edges and both are comparable for diagonal edges. However, the hybrid operator is quite insensitive to edge orientation, since actual measurement of edge strength, that is $g(i, j)$ (equation (1)) is completely insensitive to edge orientation.

Improvement of the performance of the hybrid operator in the presence of spurious noise is shown in Fig. 9. Thresholding operations are applied to the $g(i, j)$ and $G(i, j)$, the gradients before and after modification, respectively. Threshold values were chosen using the technique described in Section IV. Fig. 9 reveals that the hybrid measure of the gradient is sufficiently immune to spurious noise.

It is evident that the number of computations required for the hybrid operator is considerably larger than for the other operators. However, a coarse threshold $T_c$, reduces considerably the number of pixels for which the weighting factor $w(i, j)$ is to be computed (see algorithm 3). Threshold $T_c$ can be selected most easily from a wide range of values. Therefore the efficiency of hybrid operator is comparable to the other operators.



(a) (b)

Fig. 9(a). Results of optimum thresholding of $g(i, j)$ (equation (1)). (b) Results of optimum thresholding of $G(i, j)$ (algorithm 3)).

## VI. Conclusion

This report describes a new edge detection scheme for noisy pictures. The approach is conceptually different from other such edge detectors. Here the operations are implemented in the hierarchy: differentiation–enhancement–thresholding, whereas for other edge detectors the hierarchy is enhancement–differentiation–thresholding. Another basic difference between the hybrid operator and other operators due to Rosenfeld and Thurston and Sobel and Prewitt is the weighting factors. In the latter group of operators weighting factors are constant throughout the picture. On the other hand, for hybrid operator the weighting factor $w(i, j)$ is space-variant. The merit of this new operator, in terms of some fundamental properties that an ideal edge detector should possess, is quite satisfactory and the performance on noisy pictures is equally good. The scheme is not computationally expensive as well.

## Acknowledgment

The authors like to express their thanks to Smt. S. De Bhowmick and Mr. J. Gupta for typing the manuscript.

## References

[1] A. Rosenfeld and A. C. Kak, *Digital Picture Processing*. New York: Academic, 1976.
[2] R. C. Gonzalez and P. Wintz, *Digital Image Processing*. Massachusetts: Addison-Wesley, 1977.
[3] A. Rosenfeld and M. Thurston, "Edge and curve detection for visual scene analysis," *IEEE Trans. Comput.*, vol. C-20, pp. 562–569, 1971.
[4] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*. New York: Wiley, 1973.
[5] J. M. S. Prewitt, "Object enhancement and extraction," in *Picture Processing and Psychopictorics*, B. S. Lipkin and A. Rosenfeld, Eds. New York: Academic, 1970.
[6] I. E. Abdou and W. K. Pratt, "Quantitative design and evaluation of enhancement/thresholding edge detectors," *Proc. IEEE*, vol. 67, pp. 753–763, 1979.