CVGIP: GRAPHICAL MODELS AND IMAGE PROCESSING
Vol. 54, No. 2, March, pp. 112–133, 1992

An Optimal Linear Operator for Step Edge Detection

JUN SHEN*

Institut de Géodynamique, Bat. Géologie, Université Bordeaux-III, Avenue des Facultés, 33405 Talence, France

AND

SERGE CASTAN

IRIT, UA CNRS, Université Paul Sabatier, 118 Route de Narbonne, 31062 Toulouse, France

Received July 26, 1990; accepted June 20, 1991

Step edge detection is an important subject in image processing and computer vision and many methods, including some optimal filters, have been proposed. In this paper, we propose an optimal linear operator of an infinite window size for step edge detection. This operator is at first derived from the well-known mono-step edge model by use of a signal/noise ratio adapted to edge detection. Because of the infinite window size of the operator, we propose then a statistic multiedge model and analyze the optimal operator by spectral analysis. It is shown that the Infinite Symmetric Exponential Filter (ISEF) is optimal for both mono- and multiedge detection. Recursive realization of ISEF and the derivatives is presented and generalized to multidimensional cases also. The performance of ISEF is analyzed and compared with that of Gaussian and Canny filters, and it is shown that ISEF has a better performance in precision of edge localization, insensibility to noise, and computational complexity. Edge detection based on the optimal filter ISEF is thus presented and the essential difference between ISEF and some other optimal edge detectors is shown. The experimental results for computer-generated and real images, which confirm our theoretical analysis, are reported. © 1992 Academic Press, Inc.

1. INTRODUCTION

Edge detection has been an important subject in image processing because the edges correspond in general to important changes in physical or geometrical properties of objects in the scene. The derivatives express well the changes in the gray value function; edges can therefore be detected by the maxima of the gradient or the zero crossings of the second derivatives including the Laplacian, calculated by some differential operators.

As differential operators are sensitive to noise, a preprocessing such as smoothing is often necessary to reduce the noise. Many methods for edge detection in noisy images have been proposed, such as the Robert gradient, Sobel operators, Prewitt operators, facet model, and Laplacian operator [1–8]. Note that these differential operators are in general of small size. Because the noise in images is generally random, it is difficult to efficiently remove it from the image data merely in a small window. On the contrary, the attributes extracted from a great number of pixels would be less sensitive to noise, which can explain partially the success of the edge detection theory of Marr and Hildreth, who proposed the use of the well-known Gaussian filters of large window sizes [9]. Filtering an image by Gaussian filters of large windows needs much computation, and fast algorithms have been proposed for their realization [10–13]. The Gaussian distribution is a nonzero function from $-\infty$ to $+\infty$, but in practice, when one processes an image by Gaussian filters, the Gaussian function is considered to be nonzero only in a finite interval $[-w/2, +w/2]$; i.e., a mask of a finite size is used. The success of Marr's theory of edge detection showed that images filtered by a low-pass mask of large size are less sensitive to noise, which inspired us to hypothesize that if we use a filter kernel of an infinite size, still better results would be obtained. Moreover, the limited kernel size constraint will introduce a cutoff effect in the filtered image and produces in turn some Dirac distributions in the derivatives at the boundaries of the finite window, which is equivalent to an introduction of noise in the low-pass and derivative images. With the use of filters of infinite window size, the cutoff effect problem can be avoided.

The essential problem for a Gaussian filter is the contradiction between the noise-suppressing effect and the edge localization precision. As is well known, the larger the standard deviation of a Gaussian kernel, the less sensitive to noise the Gaussian filter. However, on the other hand, the larger the Gaussian kernel, the more planar the Gaussian function at the center and the worse the precision for edge localization. Moreover this loss of localization precision can in turn introduce difficulties for edge point verification afterward; for example, the gradient

* On leave from Southeast University, 210018 Nanjing, China.

1049-9652/92 $3.00

magnitude will be less important at the edge position detected because this position deviates from the true edge.

Given the above considerations, we propose that in order to obtain better results, it would be interesting to use a filter that is of an infinite size for efficiently reducing the noise on the one hand and is sharper at the center than the Gaussian filters for improving the precision for edge localization on the other hand. To the knowledge of the authors, the first results of edge detection by filters of infinite window sizes can be found in [14], where a symmetric exponential filter of an infinite size was used, realized by a recursive algorithm.

Canny was the first to analyze the optimal filter for edge detection by variational calculus [15]. By use of the mono-step edge model and the finite kernel size constraint, he proposed an optimal filter, which can be approximated by a Gaussian filter. A related work was that of Deriche, who proposed a simplified version of Canny's filter by applying directly Canny's results obtained under finite kernel constraints to the case of infinite window size [16]. We think that it is not sufficiently reasonable to analyze a filter of an infinite window size merely on the basis of the mono-step edge model, because when the window size is infinite, we cannot suppose that there exists only one edge in the window. In practice, only in rare cases does one edge exist all over an image.

In the present paper, we analyze the problem of the optimal filter for step edge detection, giving a summary of our research on step edge detection since 1985 [14, 17–20]. As mentioned above, to obtain better results, we prefer to use filters of infinite window size. In Section 2, following the traditional mono-step edge model with an additive independent white noise, we model the problem of edge detection in noisy images as a smoothing filter for removing the noise followed by a differential operator for detecting the change. The analysis is done from the point of view of signal processing. Because the smoothing filter works as a preparation for step edge detection, the energy of its response to the step edge and to the noise as well as to the differential noise is considered in our criterion. Note that the localization error is in fact implied in this criterion because this error is caused by the noise energy in the differential output which has been taken into account. By use of variational analysis and in the case of infinite filter window size, we conclude that an optimal low-pass filter as a preparation for edge detection is a symmetric exponential filter of an infinite size (ISEF, Infinite Symmetric Exponential Filter). The performance of the filter is analyzed and compared with that of Gaussian-like filters and it is shown that our optimal filter has a signal/noise ratio and a precision of localization better than those of Gaussian-like filters. Because we use a filter of infinite size, an analysis based on only the mono-step edge would not be sufficient. In Section 3, we propose a multiedge model described by a stationary stochastic process. By use of the theory of linear filtering of stationary processes, we show that the symmetric exponential filter of an infinite window size is also optimal as a preparation for multiedge detection. Note that the optimal kernel we develop in the present paper is deduced from mono- and multiedge models by use of the criteria determined from the point of view of signal processing rather than as an approximation or simplification of Canny's filter. Because an ideal differentiation operator implies an infinite energy in the frequency domain, it is impossible to realize exactly such operators, and if we use an approximation, additional errors will be introduced. To overcome this difficulty, in Section 4 we present how to calculate the band-limited first- and second-order derivatives of the input image filtered by the optimal symmetric exponential filter. In Section 5, we present the recursive realization of the optimal filter and its first- and second-order derivatives, taking into account the computational complexity, the finite word length effect, and the adaptation to parallel processing. In Section 6, we generalize these results to bi- and multidimensional cases by use of magnitude and maximum value distances, and the calculation of the Laplacian and partial Laplacian is introduced. Edge detection based on the optimal filter is presented in Section 7 and the adaptive gradient is introduced also. To verify the performance of the optimal filter, in Section 8 we present the experimental results of our method and compare them with results of some other optimal filters for edge detection such as the Gaussian and Canny filters. These results show that the optimal filter proposed in the present paper exhibits a better performance in insensibility to noise, precision of edge localization, computational complexity, and adaptation to parallel processing, which confirms the theoretical analysis. The paper is terminated with some concluding remarks and two tables for clarifying the essential relation and difference between some other current filters for edge detection and our optimal filter ISEF.

2. OPTIMAL FILTER BASED ON MONO-STEP EDGE MODEL

2.1. *Optimal Kernel for Monoedge Model*

In this section, we deduce the optimal filter for edge detection based on the monoedge model. An edge detection operator is considered as the cascade of a low-pass smoothing filter for removing the noise in the input image and a differential block for detecting the important gray value changes to localize the edge points (Fig. 1). The advantage of this model is that it clarifies the role that each part plays and shows that the essential characteristic of edge detectors in noisy images is the implied equivalent smoothing kernel. Once we have the optimal smoothing kernel, its derivatives of first or second order, including the Laplacian, can be used to detect edges.

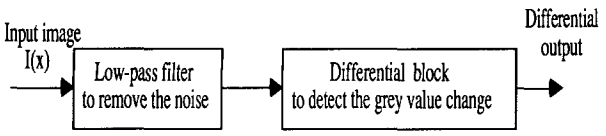

FIG. 1. An edge detector considered as a smoothing filter followed by a differential block.

Let $f(x)$ be the low-pass smoothing filter kernel for removing the noise that we want to find that gives the best results for step edge detection. The input noisy image $S_i(x)$ is modeled as (Fig. 2)

$$S_i(x) = S(x) + N(x), \tag{2.1}$$

where $S(x)$ is the step edge free of noise,

$$S(x) = \begin{cases} A, & \text{for } x > 0, \\ A/2, & \text{for } x = 0, \\ 0, & \text{for } x < 0, \end{cases}$$

and $N(x)$ is the white noise independent of $S(x)$ with

$$E\{N(x)\} = 0 \quad \text{and} \quad E\{N^2(x)\} = n^2,$$

where $E\{\cdot\}$ indicates the expectation.

The output $S_0(x)$ of the low-pass smoothing filter will be

$$\begin{aligned} S_0(x) &= S_i(x) * f(x) \\ &= S(x) * f(x) + N(x) * f(x), \end{aligned} \tag{2.2}$$

where $*$ denotes the convolution.

On the right-hand side of Eq. (2.2), the first term is the response of the filter to the step edge free of noise and the second, that to the noise.

The energy of the noise in $S_0(x)$ can be measured by E_N, with

$$\begin{aligned} E_N &= E\{[N(x) * f(x)]^2\} \\ &= E\left\{\left[\int_{-\infty}^{\infty} N(x-u) \cdot f(u) \cdot du\right)^2\right\}. \end{aligned}$$

Because the noise is white, we have

$$\begin{aligned} E_N &= E\left\{\int_{-\infty}^{\infty} N^2(x-u) \cdot f^2(u) \cdot du\right\} \\ &= n^2 \cdot \int_{-\infty}^{\infty} f^2(x) \cdot dx. \end{aligned} \tag{2.3}$$

Considering that $f(x)$ is the low-pass filter as a preparation for edge detection for removing the noise, we should analyze also the differential property of the filter. Taking the first derivative of $S_0(x)$ with respect to x, we have

$$\begin{aligned} (d/dx)S_0(x) &= (d/dx)\{S_i(x) * f(x)\} \\ &= (d/dx)\{S(x) * f(x)\} + (d/dx)\{N(x) * f(x)\}. \end{aligned}$$

Because

$$(d/dx)\{S(x) * f(x)\} = \{(d/dx)S(x)\} * \{f(x)\}$$

and

$$(d/dx)S(x) = A \cdot \delta(x), \tag{2.4}$$

where $\delta(x)$ is the Dirac distribution, we have

$$\begin{aligned} (d/dx)\{S(x) * f(x)\} &= A \cdot \delta(x) * f(x) \\ &= A \cdot f(x), \end{aligned} \tag{2.5}$$

which gives that the first-derivative response corresponding to the step edge in the output of the filter will be $A \cdot f(0)$ at the edge position $x = 0$, and the energy of this response can therefore be measured by E_S with

$$E_S = A^2 \cdot f^2(0). \tag{2.6}$$

Another part that corresponds to the noise in the first derivative of the filter output is $(d/dx)\{N(x) * f(x)\}$, whose energy can be measured by $E_{N'}$ with

$$\begin{aligned} E_{N'} &= E\{[(d/dx)N(x) * f(x)]^2\} \\ &= E\left\{\left[\int_{-\infty}^{\infty} N(x-u) \cdot f'(u) \cdot du\right]^2\right\}. \end{aligned}$$

As the noise $N(x)$ is white, we have

$$\begin{aligned} E_{N'} &= E\left\{\int_{-\infty}^{\infty} N^2(x-u) \cdot f'^2(u) \cdot du\right\} \\ &= n^2 \cdot \int_{-\infty}^{\infty} f'^2(x) \cdot dx. \end{aligned} \tag{2.7}$$

Because the edges are in general detected by the maximum of the first derivative or the zero crossing of the

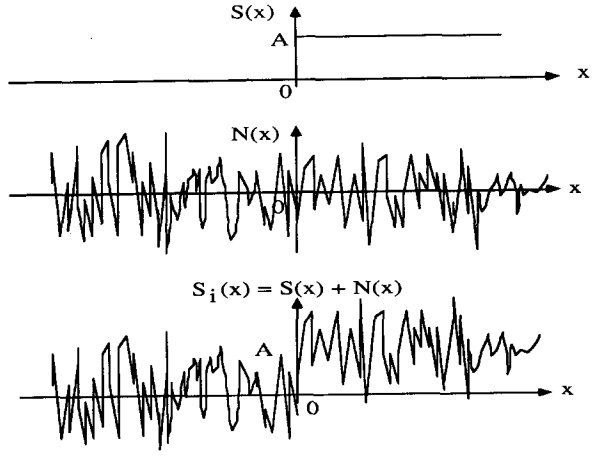

FIG. 2. Noisy mono-step edge model.

second derivative, the optimal low-pass filter $f(x)$ as a preparation for edge detection should maximize the response E_S to the step edge at the edge position and minimize the noise energy E_N and $E_{N'}$ in the smoothed output and the differential output, respectively. Combining the analysis results in Eqs. (2.3), (2.6), and (2.7), we have that the filter kernel $f(x)$ should minimize the criterion C with

$$C = \sqrt{E_N \cdot E_{N'}/E_S^2}.$$

Ignoring the coefficients representing the amplitudes of the step edge and of the noise in the criterion C, we obtain the normalized criterion C_N, which expresses the performance of the filter $f(x)$ for mono-step edge detection and is dependent only on the filter kernel,

$$C_N = \sqrt{\left\{\int_{-\infty}^{\infty} f^2(x) \cdot dx\right\} \cdot \left\{\int_{-\infty}^{\infty} f'^2(x) \cdot dx\right\}} \Big/ f^2(0), \quad (2.8)$$

and the kernel $f(x)$ should minimize C_N.

Before determining the optimal kernel $f(x)$ from the criterion C_N, we extract some properties that $f(x)$ should satisfy.

From Eq. (2.5), we see that the first derivative of the filtered step edge is $A \cdot f(x)$. Because the edge position corresponds to the maximum of the first derivative, $A \cdot f(x)$ should have its maximum at the edge position $x = 0$, and in order to avoid the introduction of additional maxima at the positions other than the edge position, this maximum should be unique; i.e., the filter kernel $f(x)$ should satisfy

$$\max_{x \in (-\infty, +\infty)} f(x) = f(0) \quad (2.9)$$

and

$$f(x_1) > f(x_2) \quad \text{for} \quad |x_1| < |x_2|.$$

Second, because all functions can be decomposed into the sum of the odd and even parts, $f(x)$ can be rewritten as

$$f(x) = f_1(x) + f_2(x), \quad (2.10)$$

where $f_1(x)$ is the odd part, $f_1(-x) = -f_1(x)$, and $f_2(x)$ is the even part, $f_2(-x) = f_2(x)$.

Substituting Eq. (2.10) in the criterion C_N, we have

$$\begin{aligned} C_N^2 = \Bigg\{ & \left[\int_{-\infty}^{\infty} f_2^2(x) \cdot dx + 2 \cdot \int_{-\infty}^{\infty} f_1(x) \cdot f_2(x) \cdot dx \right. \\ & \left. + \int_{-\infty}^{\infty} f_1^2(x) \cdot dx\right] \\ & + \left[\int_{-\infty}^{\infty} f_2'^2(x) \cdot dx + 2 \cdot \int_{-\infty}^{\infty} f_1'(x) \cdot f_2'(x) \cdot dx \right. \\ & \left. + \int_{-\infty}^{\infty} f_1'^2(x) \cdot dx\right] \Bigg\} \Big/ f_2^4(0). \end{aligned}$$

Because $f_1(x)$ is odd and $f_2(x)$ is even, we have

$$\int_{-\infty}^{\infty} f_1(x) \cdot f_2(x) \cdot dx = 0,$$

$$\int_{-\infty}^{\infty} f_1'(x) \cdot f_2'(x) \cdot dx = 0,$$

$$\int_{-\infty}^{\infty} f_1^2(x) \cdot dx \geq 0,$$

and

$$\int_{-\infty}^{\infty} f_1'^2(x) \cdot dx \geq 0.$$

This means the odd part $f_1(x)$ produces no other effect than to increase the effect of noise, so in order to minimize C_N, we should take $f_1(x) = 0$; i.e., $f(x)$ should be an even function. And C_N can be rewritten as

$$C_N^2 = 4 \cdot \int_0^{\infty} f^2(x) \cdot dx \cdot \int_0^{\infty} f'^2(x) \cdot dx \Big/ f^4(0). \quad (2.11)$$

Now we determine $f(x)$ that minimizes C_N. For the convenience of analysis, we first analyze the filter kernel of a finite window size $2W$ and then take the limit case $W \to +\infty$ to find the optimal solution.

Consider a filter $f(x)$ with a limited window size $2W$; we have from Eq. (2.11)

$$C_N^2 = 4 \cdot \int_0^{W} f^2(x) \cdot dx \cdot \int_0^{W} f'^2(x) \cdot dx \Big/ f^4(0). \quad (2.12)$$

In order to find the optimal function $f(x)$, $x \in [0, W]$, that minimizes C_N, we create

$$\Phi(f, f') = f'^2 + \lambda \cdot f^2,$$

where λ is a constant, $\lambda > 0$.

The optimal function $f(x)$ defined in the interval $[0, W]$ that satisfies the conditions of continuity and differentiation for variational calculus will be the solution of Euler's equation

$$(d/dx)\Phi_{f'} - \Phi_f = 0; \quad \text{i.e., } 2 \cdot f''(x) - 2 \cdot \lambda \cdot f(x) = 0,$$

which gives

$$f(x) = C_1 \cdot \exp(p \cdot x) + C_2 \cdot \exp(-p \cdot x) \quad (2.13)$$

with $p = \sqrt{\lambda} > 0$, and C_1 and C_2 are constants which will be determined by the boundary condition.

As was mentioned above, from the signal processing theory, in general, a filter with a larger window size will have a better performance in removing the random noise. In addition, the limited window size filters often introduce a cutoff effect in their derivatives at the boundary of

the window, which is in fact equivalent to introducing a noise effect in the differential output. So for our optimal filter for edge detection, we do not limit the window size; i.e., the limit case $W \to +\infty$ should be considered. Taking the case $W \to +\infty$, from Eq. (2.13), we see that in order to obtain a convergent filter, $f(x)$ should satisfy the boundary condition

$$\lim_{x \to +\infty} f(x) = 0,$$

which gives

$$C_1 = 0; \quad \text{i.e., } f(x) = C_2 \cdot \exp(-p \cdot x),\ x \in [0, +\infty).$$

Because $f(x)$ is an even function as mentioned above, the optimal filter $f(x)$ is

$$f(x) = C_2 \cdot \exp(-p \cdot |x|) \quad \text{with } p > 0,\ x \in (-\infty, +\infty).$$

As is well known, filters $f(x)$ and $C \cdot f(x)$, where C is a constant, have essentially the same performance in removing the noise. We would like to always take a normalized filter kernel with the amplitude gain 1, i.e.,

$$\int_{-\infty}^{\infty} f(x) \cdot dx = 1,$$

which gives

$$f(x) = (p/2) \cdot \exp(-p \cdot |x|),$$

rewritten as

$$f(x) = a \cdot b^{|x|}, \tag{2.14}$$

where $a = (-\ln b)/2$ and $0 < b < 1$. In the discrete case, we have $a = (1 - b)/(1 + b)$ and $0 < b < 1$, which implies $0 < a < 1$.

We conclude that on the basis of the mono-step edge model, the optimal low-pass linear filter as a preparation for edge detection for removing the noise is a symmetric exponential filter of an infinite window size, briefly called the infinite size symmetric exponential filter. And edges can be detected from the differentiation of the output of this filter, such as by maxima of the first derivative or zero crossings of the second derivative or Laplacian.

2.2. *Performance of the Optimal Filter*

2.2.1. *Noise/Signal Ratio (NSR).* From Eq. (2.11), we can calculate the NSR for the optimal filter:

$$C_N = \left\{4 \cdot \int_0^{\infty} a^2 \cdot b^{2x} \cdot dx \cdot \int_0^{\infty} (a \cdot \ln b)^2 \cdot b^{2x} \cdot dx\right\}^{1/2} \Big/ a^2 = 1. \tag{2.15}$$

And for the Gaussian filter $G(x, \sigma) = [1/(2\pi)^{1/2} \cdot \sigma] \cdot \exp(-x^2/2\sigma^2)$,

$$C_N = \left\{4 \cdot \int_0^{\infty} G^2(x, \sigma) \cdot dx \cdot \int_0^{\infty} G'^2(x, \sigma) \cdot dx\right\}^{1/2} \Big/ (1/2\pi\sigma^2) = (\pi/2)^{1/2} \approx 1.253. \tag{2.16}$$

We see that when the energy of the noise in the output of the filter and in its first derivative is considered with respect to the energy of the response of the filter to the step edge, the optimal filter ISEF shows a performance better than that of the Gaussian filters; the difference between them is to a factor of 25% or so.

2.2.2. *Precision for Edge Localization.* Because the edge points are detected by the maximum of the gradient or the zero crossing of the second derivative, the precision of edge localization can be analyzed from the sign change and the slope of the second derivative of the filtered image.

Consider the second derivative of the image filtered by the kernel $f(x)$,

$$(d^2/dx^2)\{S_i(x) * f(x)\} = (d^2/dx^2)\{S(x) * f(x)\} + (d^2/dx^2)\{N(x) * f(x)\}. \tag{2.17}$$

Obviously, if there is no noise, i.e., $N(x) = 0$, we have

$$(d^2/dx^2)\{S_i(x) * f(x)\} = (d^2/dx^2)\{S(x) * f(x)\} = S(x) * (d^2/dx^2)f(x) = (d/dx)S(x) * (d/dx)f(x). \tag{2.18}$$

For the optimal filter $f(x) = a \cdot b^{|x|}$, we have

$$(d/dx)f(x) = \begin{cases} a \cdot \ln b \cdot b^x & \text{for } x > 0, \\ -a \cdot \ln b \cdot b^{-x} & \text{for } x < 0, \end{cases} \tag{2.19}$$

which gives, from Eqs. (2.4) and (2.18),

$$(d^2/dx^2)\{S(x) * f(x)\} = \begin{cases} A \cdot a \cdot \ln b \cdot b^x & \text{for } x > 0, \\ -A \cdot a \cdot \ln b \cdot b^{-x} & \text{for } x < 0, \end{cases} \tag{2.20}$$

We see from Eqs. (2.18) and (2.20) that $x = 0$ is a zero crossing of the second derivative of the image filtered by $f(x)$. That is, if there is no noise, the second derivative of $f(x)$ localizes the zero crossing at the edge position with no error. Figure 3 shows the symmetric exponential filter of infinite size and its first derivative.

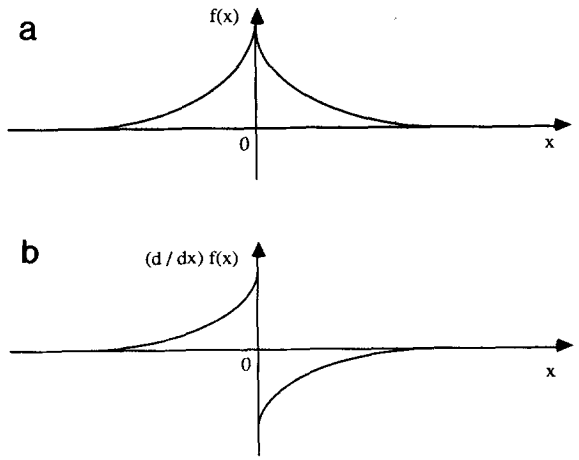

FIG. 3. (a) ISEF and (b) its first-order derivative.

Now we analyze the case where the noise exists. As in Eq. (2.17), in the second derivative of the filtered output, there exist two parts: one corresponds to the step edge and another to noise. So the zero crossing will not be detected at the position where $(d^2/dx^2)\{S(x) * f(x)\}$ changes the sign but rather at the position where $(d^2/dx^2)\{S_i(x) * f(x)\}$ changes the sign. That is, the edge pixel will be localized at $x = x_e$ such that

$$(d^2/dx^2)\{S_i(x) * f(x)\} > 0 \quad \text{for } x = x_e^+$$

and

$$(d^2/dx^2)\{S_i(x) * f(x)\} < 0 \quad \text{for } x = x_e^-,$$

or

$$(d^2/dx^2)\{S_i(x) * f(x)\} < 0 \quad \text{for } x = x_e^+$$

and

$$(d^2/dx^2)\{S_i(x) * f(x)\} > 0 \quad \text{for } x = x_e^-.$$

Because

$$(d^2/dx^2)\{S_i(x) * f(x)\} = (d^2/dx^2)\{S(x) * f(x)\} + R_n \quad (2.21)$$

with $R_n = (d^2/dx^2)\{N(x) * f(x)\}$, the zero crossing of $(d^2/dx^2)\{S_i(x) * f(x)\}$ will be the level crossing of $(d^2/dx^2)\{S(x) * f(x)\}$ passing through the level $-R_n$. From Eq. (2.20), we see that $(d^2/dx^2)\{S(x) * f(x)\}$ has a jump from $-A \cdot a \cdot \ln b \cdot b^{-x}$ to $A \cdot a \cdot \ln b \cdot b^x$ when x changes from 0^- to 0^+ (Fig. 4), so the error introduced by the noise R_n in the second derivative of the filtered image produces no effect on zero-crossing localization; i.e., the zero crossing will still be detected at $x = 0$ and no error in edge localization will be introduced even when there is noise.

To compare our optimal filter with other filters such as Gaussian filters, we further discuss this problem in an analytical form. Suppose R_n will introduce a small error in the edge localization, i.e., the zero crossing will be detected at $x = x_e$ in the noisy case rather than at $x = 0$, where x_e is the localization error. Obviously, x_e will depend on the slope of the second derivative of the filtered image at $x = 0$; the larger the slope, the smaller x_e. And x_e can be determined by

$$\begin{aligned} -R_n &\approx x_e \cdot \lim_{\varepsilon \to 0} \left\{ (1/\varepsilon) \cdot \int_{-\varepsilon/2}^{\varepsilon/2} (d/dx)(d^2/dx^2)[S(x) * f(x)] \cdot dx \right\} \\ &= x_e \cdot \lim_{\varepsilon \to 0} \left\{ (A/\varepsilon) \cdot \int_{-\varepsilon/2}^{\varepsilon/2} f''(x) \cdot dx \right\} \end{aligned} \quad (2.22)$$

with

$$R_n = (d^2/dx^2)\{N(x) * f(x)\}|_{x=0} \quad \text{and} \quad \varepsilon > 0. \quad (2.23)$$

We have from Eqs. (2.22) and (2.23)

$$x_e = \frac{-N(x) * f''(x)|_{x=0}}{\lim_{\varepsilon \to 0} \left\{ (A/\varepsilon) \cdot \int_{-\varepsilon/2}^{\varepsilon/2} f''(x) \cdot dx \right\}}. \quad (2.24)$$

The error for edge localization can therefore be measured by

$$\bar{x}_e = E\{|x_e|\},$$

which gives from Eq. (2.24)

$$\begin{aligned} \bar{x}_e &= E\{|N(x) * f''(x)|\} \cdot \left| \lim_{\varepsilon \to 0} \frac{\varepsilon}{A \cdot \int_{-\varepsilon/2}^{\varepsilon/2} f''(x) \cdot dx} \right| \\ &\le E\{|N(x)|\} \cdot \left| \lim_{\varepsilon \to 0} \frac{\varepsilon}{\int_{-\varepsilon/2}^{\varepsilon/2} |f''(x)| \cdot dx} \right| \\ &\quad \cdot \left[\int_{-\infty}^{+\infty} |f''(x)| \cdot dx \right] \Big/ |A|. \end{aligned}$$

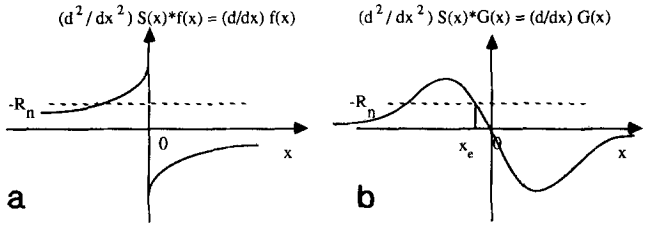

FIG. 4. Precision of zero crossing localization. (a) Symmetrical exponential filter. (b) Gaussian filter.

Removing the coefficients corresponding to the amplitudes of the step edge and of the noise, we obtain the measure of the edge localization error L_e:

$$L_e = \left| \lim_{\varepsilon \to 0} \frac{\varepsilon}{\int_{-\varepsilon/2}^{\varepsilon/2} |f''(x)| \cdot dx} \right| \cdot \left[\int_{-\infty}^{+\infty} |f''(x)| \cdot dx \right]. \tag{2.25}$$

For our optimal filter $f(x) = a \cdot b^{|x|}$, we have from Eq. (2.19) that $f''(x)$ is the distribution

$$f''(x) = \begin{cases} a \cdot (\ln b)^2 \cdot b^{|x|} & \text{for } x \neq 0, \\ (2a \cdot \ln b) \cdot \delta(x) & \text{for } x = 0; \end{cases} \tag{2.26}$$

i.e.,

$$f''(x) = a \cdot \ln b \cdot [(\ln b) \cdot b^{|x|} + 2 \cdot \delta(x)]. \tag{2.27}$$

Substituting Eq. (2.27) in Eq. (2.25), we have for the optimal filter

$$L_e = 0. \tag{2.28}$$

For the Gaussian filter $G(x, \sigma) = [1/(2\pi)^{1/2} \cdot \sigma] \cdot \exp(-x^2/2\sigma^2)$, we have

$$\lim_{\varepsilon \to 0} \frac{\varepsilon}{\int_{-\varepsilon/2}^{\varepsilon/2} |G''(x)| \cdot dx} = (2 \cdot \pi)^{1/2} \cdot \sigma^3 \tag{2.29}$$

and

$$\int_{-\infty}^{+\infty} |G''(x)| \cdot dx = 4 \cdot (e)^{1/2} \cdot \sigma^{-2}. \tag{2.30}$$

Substituting Eqs. (2.29) and (2.30) in Eq. (2.25), we have for the Gaussian filter

$$L_e = 4 \cdot (2 \cdot e \cdot \pi)^{1/2} \cdot \sigma. \tag{2.31}$$

We see that the edge localization error of a Gaussian filter is proportional to the standard deviation of the Gaussian kernel; the larger the standard deviation, the more important the edge localization error. In fact, when the noisy step edge is filtered by a Gaussian filter of a large standard deviation, the slope of the second derivative of the filtered image will be decreased, and the response to the noise in the second derivative, i.e., R_n, will cause a more important localization error (Fig. 4). As for the symmetric exponential filter of an infinite size, because the second derivative of the filtered step edge changes its value from $-A \cdot a \ln b \cdot b^{-x}$ to $A \cdot a \ln b \cdot b^{x}$ when x changes from 0^- to 0^+ (Fig. 4), the error introduced by the noise R_n in the second derivative causes no error for zero-crossing localization. Comparing Eq. (2.28) with Eq. (2.31), we conclude that our optimal filter ISEF localizes the step edge with a precision better than that of the Gaussian filters. It should be noted that in the ISEF method, the output is the convolution of the input image with the second derivative of a symmetric exponential filter of infinite window size which includes a Dirac distribution at the center of the kernel (see Eq. (2.27)). It is this Dirac distribution for the ISEF filter that ensures a very good precision for edge localization.

Note that in noisy cases, because of the remaining noise, we have, in general, another zero crossing farther from the real edge position than the one detected, regardless of which smoothing filter is used (Fig. 4). This false zero crossing will be removed by gradient thresholding and/or the "derivative sign correspondence principle," which is presented in Section 7.1.

3. OPTIMAL FILTER FOR MULTIEDGE DETECTION

In Section 2, we have analyzed the optimal filter by use of the mono-step edge model. As is well known, the mono-step edge model is widely used in edge detection studies such as [15]. When Canny deduced his optimal filter for step edge detection, he used the mono-step edge model and analyzed the optimal filter with the constraint that the filter kernel is of a finite window size. It is reasonable to use the mono-step edge model in the case of filters of limited window sizes because we can suppose that there exists only one step edge within the filter window when the window size is sufficiently small. But, in our case, because we prefer to use filters of infinite window size to remove efficiently the noise and to avoid the cutoff effect and because the optimal filter we found is a symmetric exponential filter of an infinite window size, an analysis based uniquely on the mono-step edge model evidently will be insufficient. In practical cases, we always have many edge points in an image and the case where there exists only one unique edge point on all the image is rarely encountered. So we think that for the filters of an infinite window size, an analysis based on multiedge models would be absolutely necessary.

In this section, we analyze the optimal filter problem for edge detection based on multiedge models.

Our task is to find an optimal filter that can eliminate the noise and preserve as well as possible the step edges so that its derivatives can be used for edge detection. Because the edge positions in images change from one to another, our analysis is statistical.

What we are interested in is the step edges rather than the DC or low-frequency components, so we model the step edge sequences by the following stationary stochastic processes.

The multiedge model SM(x) is shown in Fig. 5. SM(x) takes values $-A$ or A with $A > 0$, and a jump of SM(x) from $-A$ to A (respectively from A to $-A$) corresponds to a step edge. Suppose SM(x) is a stationary stochastic process satisfying the following conditions:

(1) *Stability*. The probability that we have a step edge in an arbitrary interval $(x_0, x_0 + \Delta x)$ is independent of x_0.

(2) *Orthogonality*. The number of step edges in an interval (x_0, x_1) is independent of that in another interval (x_2, x_3) if $(x_0, x_1) \cap (x_2, x_3) = \emptyset$, which is an assumption much used in image analysis [1].

(3) *Finite edge density*. The probability that there exist two or more step edges in an interval $(x, x + \Delta x)$ is very small when $\Delta x \to 0$; i.e., $P_2(\Delta x)/\Delta x \to 0$ when $\Delta x \to 0$, where $P_2(\Delta x)$ denotes the probability that we have two edge points in the interval $(x, x + \Delta x)$.

The noisy step edge sequence $\mathrm{SM}_i(x)$ is

$$\mathrm{SM}_i(x) = \mathrm{SM}(x) + N(x),$$

where SM(x) is the step edge sequence free of noise as described above and $N(x)$ is the white noise independent of SM(x) with $E\{N(x)\} = 0$ and $E\{N^2(x)\} = n^2$.

From the model SM(x) above, the probability that there exist k step edges in an interval Δx is

$$P_k(\Delta x) = \exp(-\lambda \cdot \Delta x) \cdot (\lambda \cdot \Delta x)^k/k! \tag{3.2}$$

with $\lambda > 0$, the average density of edge points in the image, and $k! = k \cdot (k-1) \cdot \cdots \cdot 1$; $P_k(\Delta x)$ is therefore a Poisson distribution.

Obviously, when there exists an even number of edge points in the interval $(x - \tau, x)$, $\mathrm{SM}(x) \cdot \mathrm{SM}(x - \tau)$ will be equal to A^2, and when there exists an odd number of edge points in the interval $(x - \tau, x)$, $\mathrm{SM}(x) \cdot \mathrm{SM}(x - \tau)$ will be equal to $-A^2$, so we have the autocorrelation of the multiedge sequence SM(x) as follows,

$$\begin{aligned} R_S(\tau) &= E\{\mathrm{SM}(x) \cdot \mathrm{SM}(x - \tau)\} \\ &= A^2 \cdot \left[\sum_{k=0}^{+\infty} P_{2\cdot k}(\tau) - \sum_{k=0}^{+\infty} P_{2\cdot k+1}(\tau)\right] \\ &= A^2 \cdot \exp(-2 \cdot \lambda \cdot |\tau|), \end{aligned} \tag{3.3}$$

which gives the spectral density of SM(x),

$$\begin{aligned} \rho_S(\omega) &= (1/\pi) \cdot \int_0^{+\infty} R_S(\tau) \cdot \cos(\omega \cdot \tau) \cdot d\tau \\ &= (2/\pi) \cdot (A^2 \cdot \lambda)/(\omega^2 + 4 \cdot \lambda^2). \end{aligned} \tag{3.4}$$

Because $N(x)$ is the white noise, its spectral density will be constant:

$$\rho_N(\omega) = (2/\pi) \cdot B^2 \quad (B > 0). \tag{3.5}$$

We therefore have from Eqs. (3.4) and (3.5) that the spectral density of the noisy step edge sequence $\mathrm{SM}_i(x)$ is

$$\rho_{SN}(\omega) = (2/\pi) \cdot [(A^2 \cdot \lambda)/(\omega^2 + 4 \cdot \lambda^2) + B^2]. \tag{3.6}$$

The optimal linear filter $f(x)$ removing the noise in the input noisy image should give an optimal estimation $M(x)$ of SM(x) from the noisy image $\mathrm{SM}_i(x)$ such that the difference between the estimation $M(x)$ and SM(x) is

$$E\left\{\int_{-\infty}^{+\infty} [M(x) - \mathrm{SM}(x)]^2 \cdot dx\right\} = \text{Minimum}.$$

According to the theory of linear filtering of stationary stochastic processes [21, 22], the spectral characteristic of the optimal filter $f(x)$ should satisfy

$$\int_{-\infty}^{+\infty} \exp(j \cdot T \cdot \omega) \cdot [\rho_S(\omega) - \phi_M(\omega) \cdot \rho_{SN}(\omega)] \cdot d\omega = 0$$

$$\text{for } -\infty < T < +\infty,$$

with $\phi_M(\omega) = \mathcal{F}\{f(x)\}$, the Fourier transform of $f(x)$. The necessary and sufficient condition is therefore

$$\phi_M(\omega) = \rho_S(\omega)/\rho_{SN}(\omega),$$

which gives from Eqs. (3.4) and (3.6)

$$\phi_M(\omega) = (A^2 \cdot \lambda/B^2)/(\omega^2 + \alpha^2) \tag{3.7}$$

with

$$\alpha = (4 \cdot \lambda^2 + A^2 \cdot \lambda/B^2)^{1/2}. \tag{3.8}$$

The optimal filter kernel $f(x)$ can therefore be found by

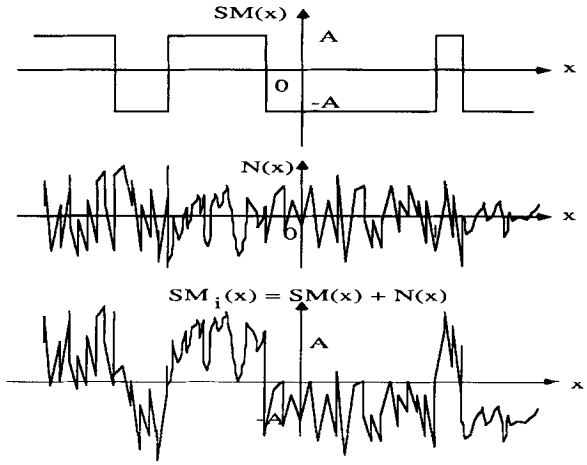

FIG. 5. Multiedge model.

$$f(x) = \mathcal{F}^{-1}\{\phi_M(\omega)\} = [(A^2 \cdot \lambda)/(2 \cdot B^2 \cdot \alpha)] \cdot \exp(-\alpha \cdot |x|). \tag{3.9}$$

We conclude that based on the multiedge model that we proposed in this section, the optimal linear smoothing filter is still a symmetric exponential filter of an infinite window size, i.e., the same as that found based on the mono-step edge model.

Note that in Eqs. (3.8) and (3.9), λ is the density of edge points, i.e., the average number of edge points in an interval of unit length. So when λ is increased, i.e., the average distance between neighboring edge pixels is decreased, α will be increased and the optimal filter will be sharper to reduce the influence between neighboring edge pixels. And, on the other hand, when the signal/noise ratio of the noisy image is decreased, the optimal filter will become planar to effectively remove the noise. But in all cases, we have always $\alpha \geq \lambda$, otherwise the filter kernel $f(x)$ would be too planar to avoid an important influence of neighboring edges.

4. DERIVATIVE COMPUTATION FOR EDGE DETECTION

We have shown that based on the mono-step edge and multi-step edge models, the optimal low-pass smoothing filter for removing the noise as a preparation for edge detection is a symmetric exponential filter of an infinite window size. Edges can therefore be detected by use of the first or second derivatives of the smoothed image. The scheme in Fig. 1 is convenient for the analysis of the optimal filter for edge detection, but for practical realization, it is not necessary to realize the low-pass filtering and the differentiating block separately. One reason is that it is almost impossible to exactly realize a differential operator without smoothing; for example, the first-order differentiation will have $j \cdot \omega$ as the Fourier transform, which has infinitely important high-frequency components and is not convergent, so it cannot be realized physically. If we use an approximation of a finite window size to realize the differential operator, errors may be introduced by the approximation and an operator of a limited window size will in general produce the cutoff effect, so the advantage of using an infinite window filter will be lost. Another reason is from the point of view of computational complexity. It will be desirable to realize, if possible, the low-pass filtering and the differential operations together by a simple algorithm, which is important for practical applications.

Rewrite the symmetric exponential filter and its derivatives as

$$f(x) = a \cdot b^{|x|} \tag{4.1}$$

with $a = (-\ln b)/2$ and $0 < b < 1$;

$$f'(x) = (d/dx)f(x) = \begin{cases} a \cdot \ln b \cdot b^x & \text{for } x > 0, \\ -a \cdot \ln b \cdot b^{-x} & \text{for } x < 0; \end{cases} \tag{4.2}$$

and

$$f''(x) = (d^2/dx^2)f(x) = a \cdot \ln b \cdot [(\ln b) \cdot b^{|x|} + 2 \cdot \delta(x)]. \tag{4.3}$$

Consider the Fourier transform of $f(x)$ in Eq. (4.1); we have

$$\mathcal{F}\{f(x)\} = 4 \cdot a^2/(\omega^2 + 4 \cdot a^2) \tag{4.4}$$

and Eq. (4.4) can be rewritten as

$$\mathcal{F}\{f(x)\} = [2 \cdot a/(2 \cdot a + j \cdot \omega)] \cdot [2 \cdot a/(2 \cdot a - j \cdot \omega)], \tag{4.5}$$

where $j = \sqrt{-1}$.

Taking the inverse Fourier transform, we have

$$f(x) = f_L(x) * f_R(x), \tag{4.6}$$

where

$$f_L(x) = \begin{cases} 2 \cdot a \cdot \exp(-2 \cdot a \cdot x) = 2 \cdot a \cdot b^x, & \text{for } x \geq 0, \\ 0, & \text{for } x < 0; \end{cases} \tag{4.7}$$

and

$$f_R(x) = \begin{cases} 0 & \text{for } x > 0, \\ 2 \cdot a \cdot \exp(2 \cdot a \cdot x) = 2 \cdot a \cdot b^{-x}, & \text{for } x \leq 0. \end{cases} \tag{4.8}$$

Comparing Eqs. (4.7) and (4.8) with Eq. (4.2), we have

$$f'(x) = a \cdot [f_R(x) - f_L(x)] \tag{4.9}$$

and another representation for $f'(x)$ is

$$f'(x) = \begin{cases} a \cdot \ln b \cdot b^x = \ln b \cdot [2 \cdot a \cdot b^x - a \cdot b^x], & \text{for } x > 0, \\ -a \cdot \ln b \cdot b^{-x} = -\ln b \cdot a \cdot b^{-x}, & \text{for } x < 0, \end{cases}$$

which gives

$$f'(x) = (-\ln b) \cdot [f(x) - f_L(x)]. \tag{4.10}$$

From Eqs. (4.3), (4.7), and (4.8), we have

$$f''(x) = 2 \cdot a \cdot \ln b \cdot [\delta(x) - f(x)] = 4 \cdot a^2 \cdot [f(x) - \delta(x)]. \tag{4.11}$$

From Eqs. (4.2), (4.9), (4.10), and (4.11), with the coefficients neglected, the low-pass output of an input image $I(x)$ filtered by the symmetric exponential filter $f(x)$ of an infinite window size and the band-limited first- and second-order derivatives can be calculated as

$$\text{Low-pass filtered image: } I(x) * f(x) = I(x) * f_L(x) * f_R(x) \tag{4.12}$$

First-order derivative:

$$(d/dx)[I(x) * f(x)] \sim I(x) * f_R(x) - I(x) * f_L(x) \tag{4.13}$$

$$(d/dx)[I(x) * f(x)] \sim I(x) * f(x) - I(x) * f_L(x) \tag{4.14}$$

$$\text{Second-order derivative: } (d^2/dx^2)[I(x) * f(x)] \sim I(x) * f(x) - I(x), \tag{4.15}$$

where left side ~ right side means that the left side can be calculated by the right side to a factor, so they are equivalent as the absolute amplitude is not concerned.

We see that on the basis of the one-side exponential filters $f_L(x)$ and $f_R(x)$, the low-pass image and the band-limited first- and second-order derivatives can be easily calculated.

5. RECURSIVE REALIZATION

From the preceding section, we conclude that given an input noisy image, the low-pass and the band-limited first- and second-order derivative images filtered by the optimal filter, i.e., the symmetric exponential filter of infinite window size, can be calculated by the essential one-side exponential filters $f_L(x)$ and $f_R(x)$. Evidently, to realize these infinite impulse response filters, recursive algorithms should be used. In this section, we present how to realize them by a very simple recursive algorithm.

5.1. *Recursive Realization of One-Side Exponential Filters*

Consider a recursive filter as

$$Y_1(i) = a_0 X(i) + a_1 Y_1(i-1), \quad i = 1, \ldots, N, \tag{5.1}$$

where Y_1 is the output, X is the input, and a_0 and a_1 are the coefficients, with $a_0, a_1 > 0$.

By use of Z-transform [23], we have

$$Y_1(i) = f_L(i) * X(i), \tag{5.2}$$

where $f_L(i)$ is the equivalent linear filter with

$$F_L(i) = \begin{cases} a_0 \cdot a_1^i & \text{for } i \geq 0 \\ 0 & \text{for } i < 0. \end{cases} \tag{5.3}$$

By comparing Eq. (4.7) with Eq. (5.3) and letting

$$a_0 = 2 \cdot a \qquad a_1 = b = \exp(-2 \cdot a), \tag{5.4}$$

the one-side filter $f_L(x)$ in Eq. (4.7) is realized.

Similarly, another one-side exponential filter $f_R(x)$ defined in Eq. (4.8) can be realized by the algorithm

$$Y_1(i) = a_0 X(i) + a_1 Y_1(i+1), \quad i = N, \ldots, 1, \tag{5.5}$$

where Y_1 is the output, X is the input, $a_0 = 2 \cdot a$, and $a_1 = b = \exp(-2 \cdot a)$, as is mentioned above.

Normalizing the filters $f_L(i)$ and $f_R(i)$ to have an amplitude gain 1 in the discrete cases, we take

$$a_0 + a_1 = 1 \quad \text{with } 0 < a_0 < 1. \tag{5.6}$$

Substituting Eq. (5.6) in Eqs. (5.1) and (5.5), we obtain the following recursive algorithms to realize the one-side exponential filters $f_L(i)$ and $f_R(i)$:

$$f_L(i): Y_1(i) = a_0 \cdot [X(i) - Y_1(i-1)] + Y_1(i-1), \quad i = 1, \ldots, N, \tag{5.7}$$

and

$$f_R(i): Y_1(i) = a_0 \cdot [X(i) - Y_1(i+1)] + Y_1(i+1), \quad i = N, \ldots, 1. \tag{5.8}$$

By use of the recursive filters in Eqs. (5.1) and (5.5) or in Eqs. (5.7) and (5.8), the symmetric exponential filter of infinite size and its first and second derivatives can be realized by their cascading or parallel combination, as is mentioned in Section 4.

5.2. *Recursive Realization Taking Account of Finite Word Length Effect*

Different combinations based on the above filters for realizing the ISEF filters and their derivatives have been proposed [16–20, 29]. Note that these algorithms were proposed with no special consideration of the errors introduced by finite word length effect. Here we present a new algorithm for realizing the ISEF filters and their derivatives that shows less complexity of computation. And because the images are stored in general in 8-bit memories, i.e., each pixel has only 8 bits, special consideration

of reducing the error propagation of the finite word length effect is also taken into account in our new algorithm.

We present first the realization of the ISEF filter. Let a_0 in Eq. (5.1) be $(1 - b)/(1 + b)$ and a_0 in Eq. (5.5) be $b \cdot (1 - b)/(1 + b)$; i.e., we use the recursive filters

$$Y_{Ls}(i) = [(1 - b)/(1 + b)] \cdot X(i) + b \cdot Y_{Ls}(i - 1), \quad i = 1, \ldots, N, \tag{5.9}$$

and

$$Y_{Rs}(i) = [b \cdot (1 - b)/(1 + b)] \cdot X(i) + b \cdot Y_{Rs}(i + 1), \quad i = N, \ldots, 1. \tag{5.10}$$

By taking the sum

$$Y(i) = Y_{Ls}(i) + Y_{Rs}(i + 1) \tag{5.11}$$

it is easy to show that $Y(i)$ is the output corresponding to the input image filtered by the discrete normalized symmetric exponential filter $f(i)$ with

$$f(i) = [(1 - b)/(1 + b)] \cdot b^{|i|}, \quad i = -\infty, \ldots, -1, 0, 1, \ldots, +\infty,$$

and $\sum_{i=-\infty}^{+\infty} f(i) = 1$.

Note that the algorithm above is well adapted to realization in parallel: Y_{Ls} and Y_{Rs} can be calculated in parallel, the structure and access to memory is simple, and the overall time to calculate an output pixel is only one multiplication and two additions.

The scheme above is well adapted for calculating the smoothed image, but it is not convenient for calculating the derivatives with precision by this scheme. In the following, we present a scheme for the computation of the derivatives.

Consider the filters in Eqs. (5.1) and (5.5) as

$$Y_{Ld}(i) = (1 - b) \cdot X(i) + b \cdot Y_{Ld}(i - 1), \quad i = 1, \ldots, N, \tag{5.12}$$

and

$$Y_{Rd}(i) = (1 - b) \cdot X(i) + b \cdot Y_{Rd}(i + 1), \quad i = N, \ldots, 1, \tag{5.13}$$

By taking the differences

$$D_1(i) = Y_{Rd}(i + 1) - Y_{Ld}(i - 1) \tag{5.14}$$

$$D_2(i) = Y_{Rd}(i + 1) + Y_{Ld}(i - 1) - X(i) - X(i), \tag{5.15}$$

it is easy to show that D_1 and D_2 are the first- and second-order derivatives of the image filtered by the symmetric exponential filter divided by a factor $p \cdot e^{-p}/2$ and $p^2 \cdot e^{-p}/2$, respectively.

Compared with the algorithms proposed in [20], the new algorithms proposed above need no more bits of memory but the amplitudes of the calculated derivatives are amplified $1/b$ times. Considering the finite word length effect [24], the errors introduced by the finite bits of memory are thus reduced by a factor b. Note that $b < 1$, and for many practical images, b is in general less than 0.5; a double precision is gained because of the new algorithm and no more memory is needed. Note that the one-side exponential filters used in the implementation are first-order recursive filters, whose limit-cycle behavior has been analyzed in the literature [24]. If the coefficients of the filters are floating-point numbers and a floating-point multiplier is used, floating-point operations can therefore be used in the calculation of the recursive filtering but the resulting image is stored in 8-bit memory; the dead band effect is avoided.

6. GENERALIZATION IN MULTIDIMENSIONAL CASES

As shown in the preceding sections, based on the mono- and multi-step edge models, the optimal filter as a preparation for edge detection is a symmetric exponential filter of an infinite window size:

$$f(x) = a \cdot e^{-p \cdot |x|} \quad (p > 0).$$

In this section, we generalize the result obtained for monodimensional cases to multidimensional cases, especially to bidimensional cases, which are the most used in image processing.

Evidently, we have two possibilities for this generalization. One is to use directional filters in different directions and the filtered directional derivatives (first or second order) can then be calculated for edge detection. Discussions on directional derivative operators can be found in the literature [1, 6]. Another possibility is to use nondirectional operators such as the band-limited Laplacian [9]. The advantage of such operators is that calculations in different directions are not needed and less computational complexity can therefore be achieved. A discussion on nondirectional operators can be found in [25]. In both these two cases, the essential problem is the determination of the corresponding bidimensional low-pass filter and its realization, because the band-limited derivatives can be easily calculated from the smoothed image, as we see later.

6.1. *Using Euclidean Distance?*

A direct generalization of a one-dimensional filter $f(x)$ to bidimensional cases is to take $f(D(x, y))$, where $D(x, y)$

is a distance measure defined for (x, y) such that $D(x, y)$ degenerates to x in one-dimensional cases, and $f(D(x, y))$ will be rotationally invariant for $D(x, y)$; i.e., all (x, y) having the same distance value will correspond to the same filter coefficient. For the symmetric exponential filter, we have the bidimensional filter $f(D(x, y))$ as

$$f(D(x, y)) = a \cdot e^{-p \cdot D(x,y)} \quad (p > 0). \tag{6.1}$$

A natural choice of $D(x, y)$ is to take the Euclidean distance

$$D_e(x, y) = \sqrt{(x^2 + y^2)}, \tag{6.2}$$

which gives

$$f_e(x, y) = a \cdot \exp[-p \cdot \sqrt{(x^2 + y^2)}]. \tag{6.3}$$

The filter $f_e(x, y)$ is circularly symmetric for Euclidean distance, as is desirable in most cases. The inconvenience of using $f_e(x, y)$ is that no fast realizations have yet been found, so one must take an approximate filter mask of a finite size to realize it, which will introduce a cutoff error, and a large mask size implies an important computational complexity. It is for this reason that we propose in the following the use of some other distance measures to deduce the bidimensional symmetric exponential filters of which a fast realization is possible.

6.2. *Magnitude Distance Symmetric Exponential Filter*

One of the most utilized distance measures in image processing is known as the magnitude distance [1], defined by

$$D_m(x, y) = |x| + |y|. \tag{6.4}$$

Substituting Eq. (6.4) in Eq. (6.1), we have

$$f(x, y) = a \cdot e^{-p \cdot (|x| + |y|)}. \tag{6.5}$$

Of course, an inconvenience of the use of the magnitude distance is that the 2D filter will no longer be isotropic for Euclidean distance measure (the isovalue line of the filter will be a square rather than a circle). But considering the important advantage that a very simple implementation can be obtained and the filter will be isotropic for the magnitude distance measure, we propose using the magnitude distance measure to generalize the symmetric exponential filter from 1D to 2D and M-dimensional ($M > 2$) cases.

Given a 2D filter $f(i, j)$ to realize, we show now how to calculate the smoothed image and the first- and second-order derivatives of a 2D image by 1D symmetric exponential filters. Obviously, the filter $f(i, j)$ is separable; i.e.,

$$f(i, j) = f(i) * f(j), \tag{6.6}$$

where $f(i)$ and $f(j)$ are the 1D symmetric exponential filters respectively in the dimensions i and j.

The directional derivatives can be calculated as

First derivative in dimension i:

$$f_i(i, j) = f(j) * f_i(i) \tag{6.7}$$

Second derivative in dimension i:

$$f_{ii}(i, j) = f(j) * f_{ii}(i) \tag{6.8}$$

First derivative in dimension j:

$$f_j(i, j) = f(i) * f_j(j) \tag{6.9}$$

Second derivative in dimension j:

$$f_{jj}(i, j) = f(i) * f_{jj}(j), \tag{6.10}$$

where $f_x(i, j)$ and $f_{xx}(i, j)$ mean the first- and second-order partial derivatives in the dimension x, respectively.

The filter $f(i, j)$ and its derivatives can therefore be decomposed into the cascade of the one-dimensional symmetric exponential filters respectively in the dimensions i and j; each can be realized in turn by the combination of two one-side exponential filters realized by first-order recursive algorithms in inverse directions, as is mentioned above. Because the lines (respectively the columns) are independent of each other for horizontal (vertical) operations, this decomposed realization can be easily implemented by parallel line processors, for example, in a parallel system SYMPATI [32].

6.3. *Maximum Value Distance Symmetric Exponential Filter*

Another kind of distance measure in image processing is the maximum value distance $D_x(x, y)$ [1], defined by

$$D_x(x, y) = \text{Max}(|x|, |y|), \tag{6.11}$$

which gives the Maximum Value Symmetric Exponential filter (MVSE filter) as

$$f_x(x, y) = a_x \cdot e^{-p_x \cdot \text{Max}(|x|, |y|)}. \tag{6.12}$$

By use of the Z-transform, it is easy to see that a maximum distance filter is in fact a rotation of $\pi/4$ of a magnitude distance filter [19]. So the separated realization of the maximum distance filter will be similar to that of the magnitude distance filter except that one-dimensional fil-

tering should be executed in two diagonal directions respectively rather than in line and column directions, and we must take the sampling interval $\sqrt{2}$ pixels in two diagonal directions because of the rectangularly sampled image.

6.4. *Laplacian Calculation in Bidimensional Cases*

As for the Laplacian, we have

$$\Delta f(i, j) = f_{ii}(i, j) + f_{jj}(i, j), \tag{6.13}$$

which can easily be calculated from the partial derivatives.

Another possibility for calculating the Laplacian is to use the difference between the bidimensionally smoothed image and the original image.

Let

$$I_2(x, y) = I(x, y) * f_2(x, y),$$

where $I(x, y)$ and $I_2(x, y)$ are respectively the input and bidimensionally smoothed output images, and $f_2(x, y)$ is the bidimensional symmetric exponential filter with

$$f_2(x, y) = a^2 \cdot b^{|x|+|y|}.$$

We have

$$\begin{aligned} I_2(x, y) - I(x, y) = {} & I_2(x, y) - I(x, y) * f(x) + I(x, y) \\ & * f(x) - I(x, y), \end{aligned} \tag{6.14}$$

where $f(x)$ is the one-dimensional symmetric exponential filter in dimension x, as is described in Eq. (4.1).

From Eqs. (6.14) and (4.11), we have

$$\begin{aligned} I_2(x, y) - I(x, y) &= I(x, y) * \{f(x) * [f(y) - \delta(y)] \\ &\quad + [f(x) - \delta(x)]\} \\ &= I(x, y) * \{f(x) * (1/4a^2) \\ &\quad \cdot (\partial^2/\partial y^2) f(y) + (1/4a^2) \\ &\quad \cdot (\partial^2/\partial x^2) f(x)\} \\ &= (1/4a^2) \cdot I(x, y) * \{(\partial^2/\partial y^2) f_2(x, y) \\ &\quad + (\partial^2/\partial x^2) f(x)\} \\ &= (1/4a^2) \cdot I(x, y) * \{[(\partial^2/\partial y^2) f_2(x, y) \\ &\quad + (\partial^2/\partial x^2) f_2(x, y)] + [(\partial^2/\partial x^2) f(x) \\ &\quad - (\partial^2/\partial x^2) f_2(x, y)]\} \\ &= (1/4a^2) \cdot I(x, y) * \{\Delta f_2(x, y) \\ &\quad - (1/4a^2) \cdot (\partial^4/\partial x^2 \partial y^2) f_2(x, y)\}. \end{aligned}$$

Neglecting the higher-order derivatives, we obtain

$$I_2(x, y) - I(x, y) \approx (1/4a^2) \cdot I(x, y) * \Delta f_2(x, y). \tag{6.15}$$

Equation (6.15) indicates that the difference between the input and the output of a 2D symmetric exponential filter is a measure of the band-limited Laplacian of the input image; we call this method the DRF (Difference of Recursive Filters) method for the Laplacian of ISEF.

Let all the pixels of a positive Laplacian take the value 1 and the others take the value 0; we obtain a Binary Laplacian Image (BLI), which is a feature image of the original image [30, 31] and can serve edge detection also.

6.5. *M-Dimensional Cases*

It is direct to generalize the algorithms above to M-dimensional (M > 2) cases.

6.5.1. Low-Pass Image and Derivatives. Given an image in the M-dimensional space $S_M(x_1, \ldots, x_M)$, the magnitude distance in the space S_M is defined by

$$D_m = \|\mathbf{X}\| = |x_1| + \cdots + |x_M|,$$

where $\mathbf{X} = (x_1, \ldots, x_M)$, and an M-dimensional symmetric exponential filter is

$$f_m(\mathbf{X}) = a \cdot \exp[-p \cdot (|x_1| + \cdots + |x_M|),$$

which is separable; i.e.,

$$\begin{aligned} f_m(\mathbf{X}) = {} & [a^{1/M} \cdot \exp(-p \cdot |x_1|) \\ & * \ldots * [a^{1/M} \cdot \exp(-p \cdot |x_M|)]. \end{aligned} \tag{6.16}$$

In Fig. 6 the isovalue surface of a 3D ISEF filter is given. Evidently, $f_m(\mathbf{X})$ can be realized by the cascade of M one-dimensional exponential filters respectively in dimensions x_i, $i = 1, \ldots, M$.

For the derivatives, we have from the convolution theorem

$$\begin{aligned} (\partial/\partial x_j) f_m(\mathbf{X}) = {} & \left\{ \prod_{i=1, i\neq j}^{M} * [a^{1/M} \cdot \exp(-p \cdot |x_i|)] \right\} \\ & * [(\partial/\partial x_j) f_m(\mathbf{x}_j)], \end{aligned} \tag{6.17}$$

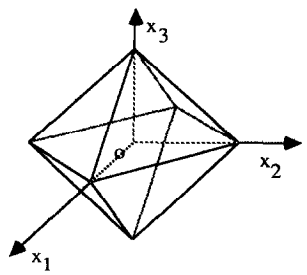

FIG. 6. Isovalue surface of a tridimensional ISEF.

in which $\prod_{i=1}^{M} * e_i = e_1 * \cdots * e_M$, because the first partial derivative $(\partial/\partial x_j) f_m(x_j)$ in the dimension x_j and the low-pass exponential filters in the other dimensions can be realized by the cascade of one-dimensional filters. The multidimensional derivative filter is thus realized by the cascade of them. The calculation of higher-order partial and mixed derivatives is similar by use of the convolution theorem, which is trivial.

6.5.2. Laplacian and Partial Laplacian. Laplacian of an M-dimensional image I(**X**) filtered by an M-dimensional exponential filter $f_m(\mathbf{X})$ is defined by

$$\Delta[f_m(\mathbf{X}) * I(\mathbf{X})] = \sum_{i=1}^{M} [(\partial^2/\partial x_i^2) f_m(\mathbf{X}) * I(\mathbf{X})],$$

which can be calculated by the sum of the second-order partial derivatives in each dimension.

Similar to 2D cases, it is easy to show that we can calculate the M-dimensional Laplacian by the difference between the output and the input of the M-dimensional exponential filter; i.e.,

$$\Delta[f_m(\mathbf{X}) * \mathrm{I}(\mathbf{X})] \sim f_m(\mathbf{X}) * I(\mathbf{X}) - I(\mathbf{X}). \quad (6.18)$$

Note that in M-dimensional cases, edge detection by zero crossings of Laplacian shows more advantages than 2D cases. First, using the M-dimensional Laplacian, the computational complexity is much more reduced than that of directional derivatives as the dimension number is increased. Second, the zero crossings of the M-dimensional Laplacian form closed supersurfaces in the M-dimensional space, which will be difficult to obtain if only directional derivatives were used. Of course, the shortcoming of the use of the Laplacian for edge detection is that it is more sensitive to noise if the linear variation condition is not satisfied [9].

It is interesting to introduce the partial Laplacian in multidimensional cases defined by

$$\Delta(x_{k1}, \ldots, x_{kn})[f_m(\mathbf{X}) * I(\mathbf{X})] = (\partial^2/\partial x_{k1}^2) f_m(\mathbf{X}) * I(\mathbf{X}) + \cdots + (\partial^2/\partial x_{kn}^2) f_m(\mathbf{X}) * I(\mathbf{X}) \quad (6.19)$$

with $x_{ki} \in \{x_1, \cdots, x_M\}$, for $i = 1, \ldots, n$. Evidently, this partial Laplacian can be calculated by the sum of the directional second-order derivatives in the dimensions concerned, but it can also be calculated by the difference between the output and input of multidimensional filters

$$\Delta(x_{k1}, \ldots, x_{kn})[f_m(\mathbf{X}) * I(\mathbf{X})] \sim f_m(\mathbf{X}) * I(\mathbf{X}) - f_m(x_{k1}, \ldots, x_{kn}) * I(\mathbf{X}),$$

which can be easily realized by the cascade of 1D exponential filters.

Edges can be detected by the zero crossings of the partial Laplacian. The advantage of the use of partial Laplacian is that we can detect and obtain closed edge supersurfaces in different subspaces adapted to the problem concerned. Moreover, the low-pass filtering in the complement of the subspace smooths the M-dimensional image to suppress the noise but does not blur the edges in the subspace in which we are interested, if the complement and the subspace are orthogonal to each other, as is the case in many practical images.

7. EDGE DETECTION BY ISEF AND ADAPTIVE GRADIENT

With the optimal linear filter for edge detection and its realization proposed above, edges can be detected by maxima of gradient, zero crossings of directional second derivatives, or zero crossings of Laplacian, always filtered by the optimal symmetric exponential filter. The edge candidates thus obtained are then verified by gradient thresholding with or without hysterisis. Note that the Laplacian will be more sensitive to noise than directional derivatives if the linear variation condition is not satisfied, as was mentioned by Haralick and Marr [6, 9], so the use of directional derivatives can give better results than the Laplacian. For ISEF, this is also the case [20]. On the other hand, use of the Laplacian shows the following advantages: (a) it is rotationally symmetric to corresponding distance measure (for example, magnitude distance or maximum value distance); (b) derivative calculation in different directions is not needed and a less important computational complexity can therefore be achieved; and (c) the edges detected will be closed and of 1 pixel width.

7.1. *False Zero-Crossing Suppression by Sign Correspondence*

As is well known, step edges can be detected by the zero crossings of the second derivatives. In Fig. 7 the two different cases of smoothed step edges and the first- and second-order derivatives are shown. We see that in case 1, the first derivative is positive and the second derivative will change its sign from positive to negative when x is increased to pass through the zero crossing. For simplicity, we call such a zero crossing a *positive zero crossing*. In case 2, the first derivative is negative, which corresponds to a *negative zero crossing* (i.e., when x is increased to pass through the zero crossing, the second derivative changes its sign from negative to positive). We conclude that for a step edge, if it has a positive first derivative, it must correspond to a positive zero crossing;

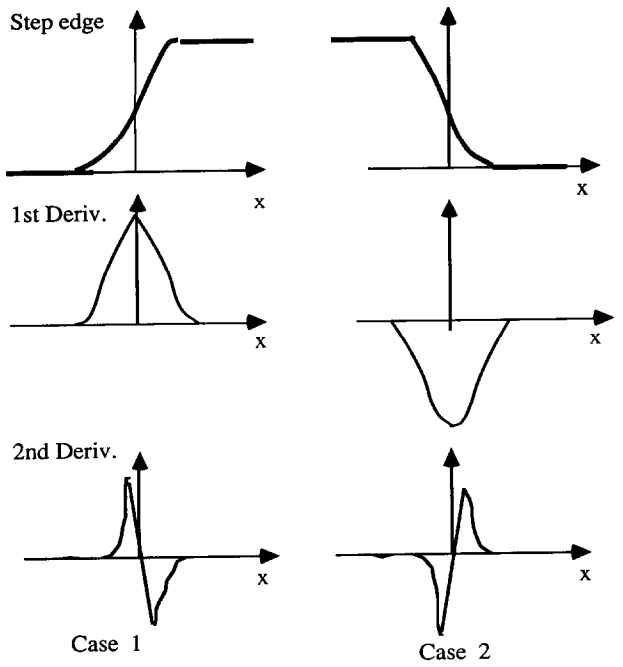

FIG. 7. Sign correspondence between the first and second derivatives of a step edge.

otherwise if it has a negative first derivative, it must correspond to a negative zero crossing. We call this correspondence between the signs of the first and second derivatives of a step edge the "zero-crossing sign correspondence principle." And a zero crossing violating this principle cannot be a step edge.

Because of the existence of noise and the rounding or truncation errors, in practice when we calculate the derivatives of the input image to detect step edges, the results are in general not as simple as those in Fig. 7. For the step edges of case 1 and case 2, we obtain some false zero crossings as shown in Fig. 8.

Sometimes it is difficult to remove these false zero crossings by gradient thresholding, because they are close to the true edge and can therefore correspond to a large gradient value (note that the calculated gradient is a smoothed one). A high threshold will cause a considerable loss of true edges and a low threshold will not be sufficient for false zero-crossing suppression. But if we examine the sign of the derivatives, we see that these false zero crossings violate the sign correspondence principle and can therefore be easily removed even when they correspond to a high gradient value. That is why we propose removing them with the use of the sign correspondence for zero crossings before the gradient thresholding. According to our experiments, we find that the use of sign correspondence helps much in suppressing false zero crossings that are sometimes difficult to remove by gradient thresholding.

7.2. *Adaptive Gradient*

We present here in more details edge detection by zero crossings of the Laplacian. The BLI is first calculated by use of ISEF (see Section 6.4). The zero crossings, i.e., boundary pixels between different regions in BLI, are then verified by gradient thresholding. In cases where the input image is very noisy, when necessary, a process of "noisy zone suppression in BLI" can be applied before gradient thresholding. Because the random noise shows in general an important high frequency with respect to the object in the scene and the optimal filter ISEF localizes the edges with a good precision, the zones in BLI produced by noise will be very small patches, which is against the continuity of objects, so one can suppress small isolated patches (which are in general produced by the random noise) to remove the false noisy zones in BLI.

For gradient thresholding, we can use, of course, the first derivatives filtered by the ISEF filter. Another possibility is to use the adaptive gradient presented in the following.

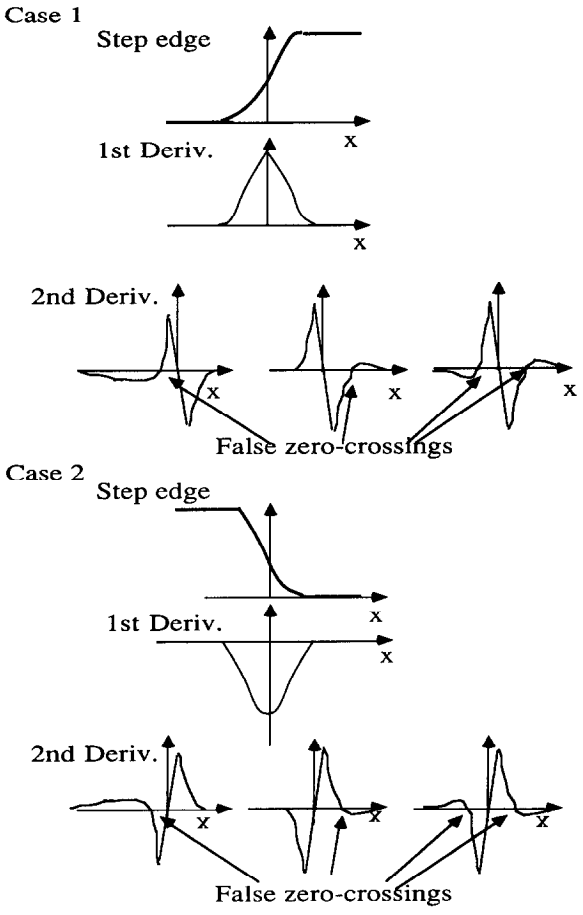

FIG. 8. False zero crossings close to a step edge.

Because zero crossings of the band-limited Laplacian are verified by gradient thresholding, the quality of the gradient estimate plays an important role in edge detection. Obviously, a shift-invariant band-limited gradient operator will smooth together the two regions of different gray values separated by the edge pixels and give therefore an estimate smaller than the real gradient magnitude as the edges in a 2D image can have arbitrary forms. In this manner difficulties for edge pixel verification arise. We show here how to give an estimate that best approximates the real gradient magnitude with the help of BLI. Our deduction shows that in 2D cases, this estimator is not shift-invariant.

Let W be a window centered at the pixel P which is a zero crossing of the band-limited Laplacian. Suppose that W is composed of N regions R_i $(i = 1, \ldots, N)$ of different gray values g_i $(i = 1, \ldots, N)$ with an additive white noise $N(x, y)$ of mean 0 and R_I, R_J, the two regions separated by P (Fig. 9). We have

$$G(x, y) = g_i + N(x, y), \quad (x, y) \in R_i \in W, \quad i = 1, \ldots, N, \tag{7.1}$$

and

$$E[N(x, y)] = 0 \tag{7.2}$$

$$E[N^2(x, y)] = n_0^2 \tag{7.3}$$

where $G(x, y)$ is the gray value of the pixel (x, y) and $E[\cdot]$ indicates the expectation.

A linear gradient estimator $M(x, y)$ gives the estimated gradient $\hat{}\mathrm{gr}(P)$ for the pixel P by

$$\hat{}\mathrm{gr}(P) = \sum_{i=1}^{N} \sum_{(x,y)\in R_i} M(x, y) \cdot G(x, y) \tag{7.4}$$

and $\hat{}\mathrm{gr}(P)$ should satisfy for all possible gray value distributions in W,

$$E\{\hat{}\mathrm{gr}(P)\} = \mathrm{gr}(P) \tag{7.5}$$

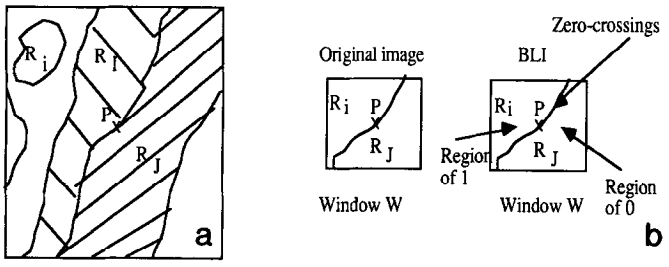

FIG. 9. Adaptive gradient. (a) Window W composed of N regions. (b) When W is small, it is, composed of only two regions corresponding to 1 and 0 in BLI separated by the zero crossings.

and

$$E\{[\hat{}\mathrm{gr}(P) - \mathrm{gr}(P)]^2\} = \text{Minimum}, \tag{7.6}$$

where $\mathrm{gr}(P)$ is the gradient magnitude at P free of noise; i.e.,

$$\mathrm{gr}(P) = g_I - g_J. \tag{7.7}$$

Substituting Eqs. (7.1), (7.4), and (7.7) in Eq. (7.5), we have

$$\sum_{i=1}^{N} \sum_{(x,y)\in R_i} M(x, y) \cdot g_i + E\left\{\sum_{i=1}^{N} \sum_{(x,y)\in R_i} M(x, y) \cdot N(x, y)\right\} = g_I - g_J. \tag{7.8}$$

We have from Eqs. (7.2) and (7.8)

$$\begin{aligned} &M(x, y) = 0 \quad \text{for } (x, y) \in R_i \ (i = 1, \ldots, N \text{ and } i \neq I, J) \\ &\sum_{(x,y)\in R_I} M(x, y) = 1 \\ &\sum_{(x,y)\in R_J} (M(x, y) = -1. \end{aligned} \tag{7.9}$$

Because $N(x, y)$ is white, substituting Eq. (7.9) in Eq. (7.6), we have

$$E\{[\hat{}\mathrm{gr}(P) - \mathrm{gr}(P)]^2\} = n_0^2 \cdot \left[\sum_{(x,y)\in R_I} M^2(x, y) + \sum_{(x,y)\in R_J} M^2(x, y)\right] = \text{minimum}. \tag{7.10}$$

From Eqs. (7.9) and (7.10), the optimal gradient estimator is

$$M(x, y) = \begin{cases} 1/n_I & \text{for } (x, y) \in R_I \\ -1/n_J & \text{for } (x, y) \in R_J \\ 0 & \text{for } (x, y) \in W \text{ and } (x, y) \notin R_I \cup R_J, \end{cases} \tag{7.11}$$

where n_I and n_J are respectively the numbers of pixels in regions R_I and R_J. Equations (7.10) and (7.11) give the error of the estimate:

$$E\{[\hat{}\mathrm{gr}(P) - \mathrm{gr}(P)]^2\} = n_0^2 \cdot [1/n_I + 1/n_J]. \tag{7.12}$$

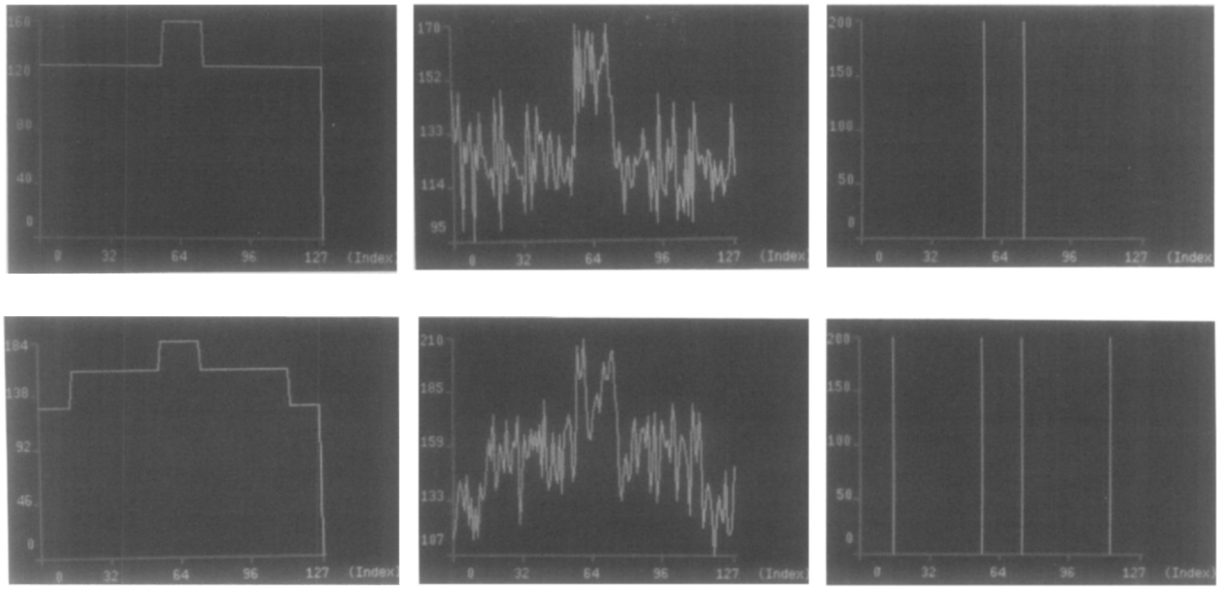

FIG. 10. Experimental results of ISEF filter for 1D noisy edge signals.

We see that the optimal gradient estimate is the difference between the averages of gray values of the regions R_I and R_J, which is evidently not shift-invariant. And the more pixels of regions R_I and R_J contained in window W, the less important the estimate error. As for practical applications, to make the calculation simple, we take window size 5×5 or 7×7 and experiments show that the result is satisfactory. In this case, window W contains in general only two regions separated by the zero crossings (Fig. 9). These two regions will correspond respectively to zones of values 1 and 0 in W of the binary band-limited Laplacian image (BLI), because the ISEF method detects zero crossings with a good precision [17]. So the optimal gradient estimate can be calculated by the difference between the gray value averages in the original image corresponding to zones 1 and 0 in window W of the BLI. This estimate is known as the adaptive gradient proposed by Shen and Castan [17].

It should be noted that the adaptive gradient can be used only when the possible edge distribution in window W is known. This is why we propose using it for gradient thresholding after the detection of zero crossings. And it is the use of the knowledge of zero crossings obtained before the gradient calculation that makes the adaptive gradient give an estimate better than that of those that do not make use of this already obtained knowledge. Compared with the shift-invariant operators, the adaptive gradient operator separately smoothes the regions of different gray values to remove the noise but at the same time no blurring effect is produced because it processes different regions separately by use of BLI. In general, the adaptive gradient gives a stronger response to edge points; i.e., it shows the advantage of being sensitive to weak edges. On the other hand, because of the small window size used for adaptive gradient calculation, it may be more sensitive to noise than the first derivatives calculated by ISEF, especially for very noisy images. Of course, the adaptive gradient can be improved by use of a larger window, but in this case this will imply a greater computational complexity.

8. EXPERIMENTAL RESULTS

Our optimal edge detection filter is deduced from one- and multiedge models. Obviously, real images will be still more complicated than these models. To examine the robustness of the models and of the theoretical results, the methods of edge detection by ISEF have been implemented by the recursive algorithms presented and tested for different types of images, including computer-generated and real ones. The experimental results are very satisfactory. In Fig. 10 experimental results for 1D noisy edge signals are given. Some examples for 2D images are given in Fig. 11 and Fig. 12. To compare the performance of different filtering techniques for edge detection, we first use the noisy images created by computer. Because in the artificial image, we know exactly where we do and do not have the edges with this kind of image, it is convenient and confirmatory to examine and compare the performance of different techniques, such as the sensibil-

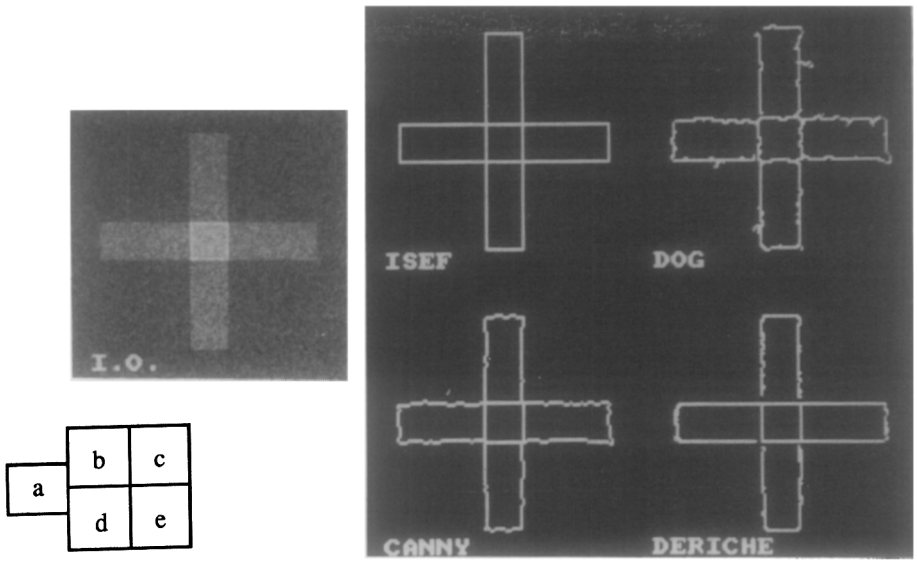

FIG. 11. Experimental results for a computer-generated noisy image. (a) Original image; (b) edges detected by ISEF filter; (c) best result for edges detected by DOG; (d) best result for edges detected by Canny–Gaussian filter; (e) best result for edges detected by simplified version of Canny by Deriche.

ity to noise, the precision of localization, and the ability to detect real edges. An example is given in Fig. 11. We see that with the optimal filter presented in this paper, the result is excellent: all the edge points are detected with no errors of localization, and there exist no false edge points caused by noise. With Gaussian filters or the simplified version of Canny's filters, we have never obtained such a result, though many different parameters were tested. In Fig. 12 some examples for real images are given. The experimental results show the good performance of the optimal filter proposed for edge detection, which confirms the theoretical analysis.

9. CONCLUSION

An essential difficulty with Gaussian filters for edge detection is the contradiction between the insensibility to noise and the precision of edge localization. In the present paper, we deduce the optimal linear operator for edge detection from the point of view of signal processing. A linear edge detector is considered as a low-pass smoothing filter to remove the noise followed by a differential element to detect the changes. The optimal operator is at first deduced from the well-known monostep edge model, based on a combined signal/noise ratio criterion adapted to edge detection, i.e., maximizing the response to the step edge and minimizing those to the noise and to the derivative of the noise. It is shown that an optimal low-pass filter as a preparation for edge detection is the infinite symmetric exponential filter. The performance of this optimal filter is analyzed and compared with that of the well-known Gaussian-like filters. We show that the optimal filter proposed in the present paper has a better performance in terms of insensitivity to noise and precision of edge localization.

Because the ISEF filter is of an infinite window size and many edge points always exist in a real image, an analysis based on only the mono-step edge model is not sufficient. Thus we propose a noisy multiedge model that is described by a stationary stochastic process with an additive white noise. On the basis of the spectral analysis, we show that the optimal linear filter for multiedge detection is still an ISEF filter.

We propose also the recursive realization of the ISEF and its first and second derivatives, taking into account the computational complexity, the finite word length effect, the facility for hardware implementation, and the adaptation to parallel processing.

These algorithms are generalized to bidimensional and multidimensional cases. It is shown that the ISEF filters and their derivatives are separable for magnitude and maximum value distances, the two most used distances in image processing. So a multidimensional ISEF and the derivatives, including the Laplacian, can be easily calculated from the cascade of one-dimensional ISEF filters. The use of the Laplacian exhibits the advantage of extremely little computational complexity and gives closed edge supersurfaces even in high-dimensional cases. We introduced also the partial Laplacian for high-dimensional cases, which shows the advantages of obtaining closed edge surfaces in some subspaces of interest, and the smoothing of the complement of the subspace suppresses the noise without blurring the subspace edges.

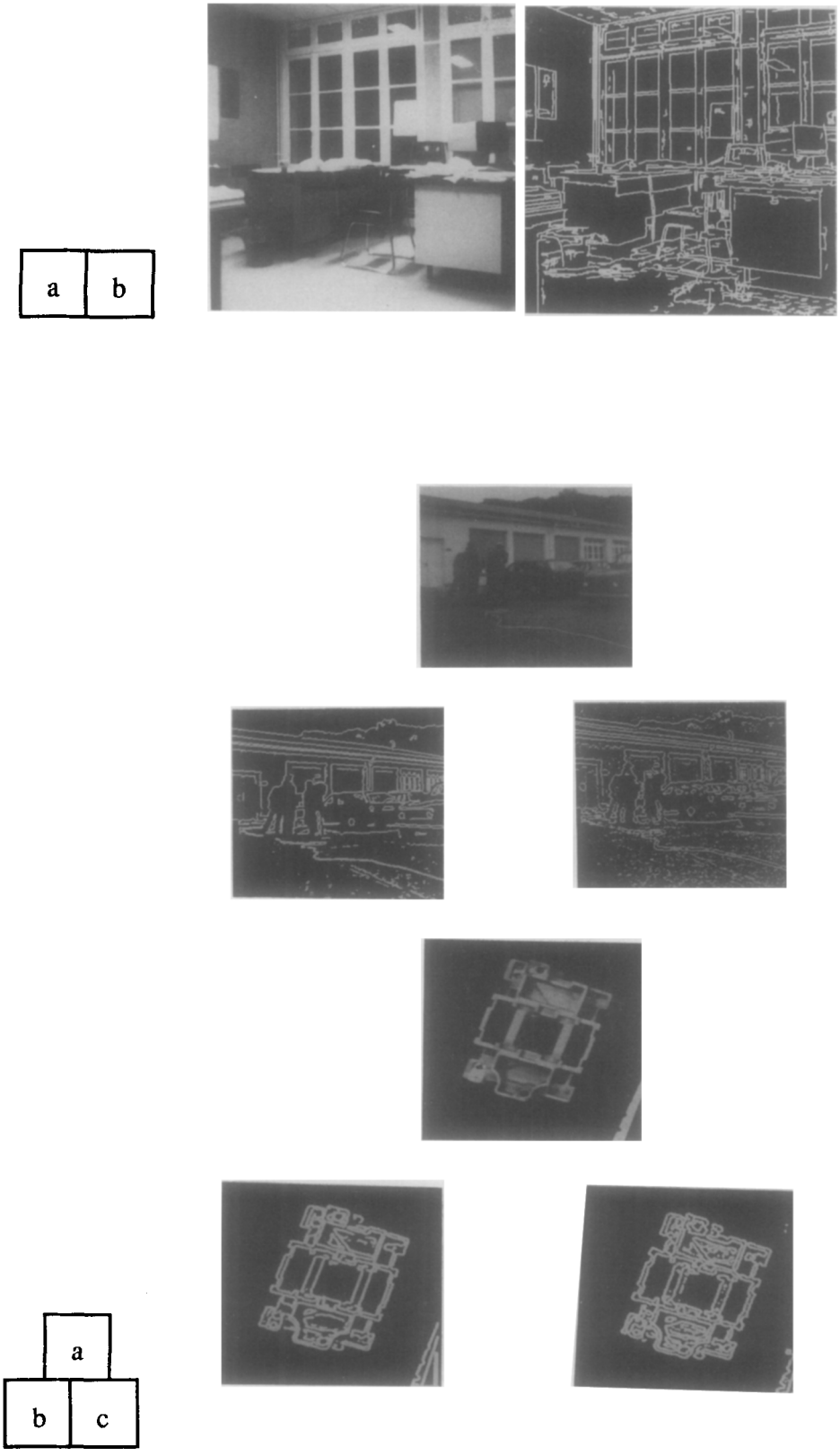

FIG. 12. Edges detected by ISEF methods. Top: (a) original image; (b) edges detected by ISEF (gradient threshold 5). Bottom: (a) original image; (b) edges detected by the DRF method (gradient thresholds 5 and 8); (c) edges detected by Laplacian of Gaussian filter (gradient thresholds 5 and 8).

With the algorithms presented, edges can therefore be detected by the zero crossings of the second derivatives or Laplacian, or by the maxima of the gradient, always filtered by the ISEF filter. A technique using the sign correspondence between the first and second derivatives is presented to remove some false zero crossings. The edge candidates thus detected are then verified by gradient thresholding with or without hysteresis. This gradient can be calculated from the ISEF filter or from the adaptive gradient if non-shift invariant operators are considered. Note that the adaptive gradient can be used only after the BLI is obtained. The advantage of the adaptive gradient is that it is more sensitive to weak edges because regions of different gray values are smoothed separately.

The optimal linear operator for edge detection proposed is implemented and tested for computer-generated and real images and compared with some other optimal operators such as the Gaussian filter, Canny's filter, and its simplified version by Deriche. The experimental results show a significantly better performance by the ISEF filters, which confirms the theoretical analysis of optimization and performance.

It may be interesting to explain qualitatively the optimal operator for edge detection as the following:

In a strict mathematical sense, the differentiation at a pixel should concern only its very close neighbors; the closer a pixel to the pixel of interest the more important the role that it should play in differential operation. Because of the existence of noise, an additional smoothing will be necessary, but we should always keep in mind the principle that closer pixels should play a more important role in the differentiation operation. Obviously, in examining the first derivative of the symmetric exponential filters, we see that this principle is always respected no matter how great the smoothing factor b for removing the noise (Fig. 3). So even when one adjusts the smoothing factor b to adapt it to different noisy images, this property is preserved and always gives a Dirac distribution in the second derivative of the filter kernel at the center of the kernel, which assures that on the one hand the more

TABLE 1
Comparison between ISEF and Canny Filter

	Canny filter	ISEF filter
Model	Mono-step edge	Mono-step edge
	—	Multiedge
Type	First deriv.	Low pass followed by differentiation (first and/or second order)
Criterion	S/N ratio	Point of view of signal processing
1. Monoedge	Precision of local.	Max. resp. to step edge
	Max. unique	Min. resp. to noise
		Min. resp. to noise in deriv.
		Max. unique
2. Multiedge	—	Optimal estimation
Analysis method	Variational calculus	Variational calculus
		Spectral analysis for stochastic process
Constraint	Limited window size, continuous deriv.	Window size nonlimited
		Discontinuous deriv. permitted
Problem	For limited size, mono-step edge model, reasonable	For infinite window size, multiedge model analysis necessary, (never only one edge)
Optimal filter obtained	$f(x) = a_1 e^{\alpha x} \sin \omega x + a_2 e^{\alpha x} \cos \omega x + a_3 e^{-\alpha x} \sin \omega x + a_4 e^{-\alpha x} \cos \omega x + c$	Low pass: $f(x) = ce^{-\alpha\lvert x\rvert}$, first and second deriv.
Separability	Nonseparable	Separable for distances of four and of eight neighbors
Realization	Approximated by first deriv. of Gaussians	Strictly optimal and realized by recursive algorithms (first and/or second deriv., Laplacian)
Particularity	Continuous second deriv.	δ distribution in second deriv. at the center
		Jump in first deriv. at the center
Edge localization error	$4 \cdot (2 \cdot e \cdot \pi)^{1/2} \cdot \sigma$ (zero error impossible)	0
S/N ratio (Shen–Castan's criterion: $E_S/(E_N \cdot E_N')^{1/2}$)	$(2/\pi)^{1/2} < 1$	1

TABLE 2
Comparison between Gaussian, Canny, and Deriche Filters and ISEF

1. Edge localization error x_e [17]

Gaussian filter	ISEF filter
$x_{eG} = 4 \cdot (2 \cdot e \cdot \pi)^{1/2}/\alpha$	$x_{eE} \approx 0$
Canny filter	
$x_{eC} \approx 0.81/\alpha$	
Deriche filter	
$x_{eD} = 4 \cdot e^{-1}/\alpha = 1.47/\alpha$	

2. Canny's signal/noise ratio at the edge point detected [20][a]

Gaussian filter	ISEF filter
$SNR_G = 2 \cdot \sigma \cdot e^{-32\pi\sigma}/\tau^{1/2}$	$SNR_E = 1/\alpha$
Canny filter	
$SNR_C = 0.39/\alpha$	
Deriche filter	
$SNR_D = 2 \cdot (1 + 4/e)^2 \cdot e^{-8/e}/\alpha$ $\approx 0.64/\alpha$	

3. Computational complexity for one pixel in a 2D image[b]

Deriche filter	ISEF filter
Low-pass filtering	Low-pass filtering
16× and 14+	4× and 9+
First-derivative operator	First-derivative operator
13× and 13+	4× and 10+
Second-derivative operator	Second-derivative operator
15× and 14+	4× and 12+
Laplacien operator	Laplacien operator
14× and 16+	4× and 10+

[a] SNR $= f^2(x_e)/\int_{-\infty}^{\infty} f'^2(x)\,dx$ (x_e, estimation of edge position detected).

[b] We only give a comparison between the exponential filter and the Deriche filter [16], because they are implemented by recursive algorithms needing less computation.

or less important smoothing in an infinite window efficiently removes the noise, and on the other hand this smoothing will not disturb the edge localization; a very good precision for edge localization is thus obtained. As for other edge detectors of smoothed first-derivative type, such as the first derivative of the Gaussian or Canny filter, the kernel changes its value slowly at the center and the maximum value in the kernel of its first derivative will correspond to a position more or less far from the center, so the differentiation principle is violated and the noise remaining after the smoothed differentiation will cause an error in edge localization. This error in turn decreases the signal/noise ratio at the edge position detected because it deviates from the real edge. When one needs to use an important smoothing to remove the noise, the maximum of the kernel of first-derivative type will be far from the center and the contradiction between the precision of localization and noise suppressing arises. It is thus difficult and sometimes impossible to have an ideal compromise to solve this contradiction. With the optimal symmetric exponential filters, this problem is avoided. It should be noted that there exist other edge detection operators of first-derivative type having a Dirac distribution in its derivative at the center, such as the box difference filters (average difference filters) [28]. But because first the form of this kind of kernel is not optimal for edge detection in complicated noisy images and second there also exist Dirac distributions in the derivatives at the boundary of these kernels that serve to introduce a noise effect in the second derivatives, they show a performance worse than that of the ISEF filters.

In order to show more clearly the relation and the essential difference between our optimal ISEF filter for edge detection and the Gaussian and Canny filters, we list them in Table 1 as a summary. A comparison between the Gaussian, Canny, and Deriche filters and the ISEF is given in Table 2. We conclude that the optimal ISEF filter for edge detection shows a better performance in insensibility to noise, precision of edge localization, and reduced computational complexity and is well adapted to parallel processing.

REFERENCES

1. W. K. Pratt, *Digital Image Processing*, Wiley, New York, 1978.
2. R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*, Wiley, New York, 1973.
3. J. Prewitt, Object enhancement and extraction, in *Picture Processing and Psychopictories* (B. Lipkin and A. Rosenfeld, Eds.), pp. 75–149, Academic Press, New York, 1970.
4. R. Haralick, Edge and region analysis for digital image data, *Comput. Vision Graphics Image Process.* **12,** 1980, 60–73.
5. R. Haralick and L. Watson, A facet model for image data, *Comput. Vision Graphics Image Process.* **15,** 1981, 113–129.
6. R. Haralick, Digital step edges from zero-crossings of second directional derivatives, *PAMI* **6**(1), 1984, 58–68.
7. M. Hueckel, An operator which locates edges in digitized pictures, *J. Assoc. Comput. Mach.* **18,** 1971, 113–125.
8. J. Shen and S. Castan, Stereo vision using perspective projections, *Proceedings 1st Image Symposium, Biarritz, May 1984,* pp. 1017–1022.
9. D. Marr and E. C. Hildreth, Theory of edge detection, *Proc. R. Soc. London B* **207,** 1980, 187–217.
10. P. J. Burt, Fast filter transform for image processing, *Comput. Vision Graphics Image Process.* **17,** 1981, 33–51.
11. J. Shen and S. Castan, Fast filter transform theory and design for image processing, *Proceedings IEEE Conference on Computer Vision and Pattern Recognition (CVPR), San Francisco, 1985.*
12. J. Shen and S. Castan, 2-D Gaussian type image filter decomposition and fast realization, *Proceedings, 4th Scandinavian Conference on Image Analysis, Trondheim, 1985.*
13. J. Shen and S. Castan, Box filtering technique for Gaussian type image filters by use of the B-spline functions, *Proceedings of the 4th Scandinavian Conference on Image Analysis, Trondheim, 1985.*
14. J. Shen and S. Castan, A new algorithm for edge detection, *Proceedings, 5th Conference on P.R.&A.I., Grenoble, 1985.* [In French]

15. J. F. Canny, Finding edges and lines in images. *MIT Tech. Rep. 720,* 1983.
16. R. Deriche, Optimal edge detection using recursive filtering, *Proceedings, 1st International Conference on Computer Vision, London, June 8–12, 1987.*
17. J. Shen and S. Castan, An optimal linear operator for edge detection, *Proceedings CVPR'86, Miami, 1986.*
18. J. Shen and S. Castan, Edge detection based on multi-edge models, *Proceedings SPIE'87, Cannes, 1987.*
19. J. Shen and S. Castan, Further results on DRF method for edge detection, *Proceedings, 9th ICPR, Rome, 1988.*
20. S. Castan, J. Zhao, and J. Shen, New edge detection methods based on exponential filter, *Proceedings, 10th ICPR, June 1990.*
21. A. Papoulis, *Probability, Random Variables And Stochastic Process, McGraw-Hill, New York, 1965.*
22. A. Papoulis, *Signal Analysis,* McGraw–Hill, New York, 1972.
23. K. Ogata, *Modern Control Engineering,* Prentice-Hall, Englewood Cliffs, NJ, 1970.
24. A. V. Oppeinheim and R. W. Schafer, *Digital Signal Processing,* Prentice–Hall, Englewood Cliffs, NJ, 1975.
25. M. Brady and B. Horn, Rotationally symmetric operators for surface interpolation, *Comput. Vision Graphics Image Process.* **22,** 1983, 70–94.
26. V. Torre and T. A. Poggio, On edge detection, *PAMI* **8**(2), Mar. 1986.
27. J. S. Chen and G. Medioni, Detection, localization, and estimation of edges, *PAMI* **11**(2), Feb. 1989.
28. A. Rosenfeld and M. Thurston, Edge and curve detection for visual scene analysis, *IEEE Trans. Comput.* 1971.
29. R. Deriche, Fast algorithms for low-level vision, *PAMI* **12**(1), 1989.
30. H. K. Nishihara, PRISM: A practical real-time imaging stereo matcher, A.I. Memo 780, MIT, 1984.
31. J. Shen and S. Castan, A new fast algorithm of stereo vision, *Proceedings, 2nd SPIE, Cannes,* 1985.
32. J. Basille and S. Castan, Multilevel architecture for image processing, *Proceedings, 2nd SPIE, Cannes, 1985.*