

Design of FIR bilevel Laplacian-of-Gaussian filter

Soo-Chang Pei^{a,*}, Ji-Hwei Horng^b

^a*Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan, ROC*

^b*Department of Information Management, National Kaohsiung University of Applied Sciences at Kinmen, Taiwan, ROC*

Received 27 October 2000; received in revised form 21 September 2001

Abstract

The Laplacian-of-Gaussian (LoG) filter is an optimal edge detector, but it is computationally inefficient. In this paper, we propose the bilevel Laplacian-of-Gaussian (BLoG) filter to approximate the LoG filter. This approximation is formulated as an optimization problem and solved by the gradient descent algorithm. Only two multiplications per pixel are required in convolving with the BLoG filter. The computation is fast and it produces satisfactory results. The applications of 1D BLoG filter to corner detection and 2D BLoG filter to edge detection are also presented. © 2002 Elsevier Science B.V. All rights reserved.

Keywords: Laplacian-of-Gaussian; Gradient descent; McClellan transformation; Corner detection; Edge detection

1. Introduction

Edge detection is an important topic in the research area of image analysis. Many optimal filters for edge detection based on different cost functions have been proposed [2,8,11,12]. Dickey and Shanmugam [2] derived an optimal filter by maximizing the portion of output image energy in the vicinity of the edge location for a given resolution requirement. Marr and Hildreth [8] found that the Laplacian-of-Gaussian (LoG) filter minimizes the product of spatial localization and bandwidth. Shen and Castan [11] proposed an optimal filter based on one-step model (a step edge and the white noise). This filter can be realized by first-order recursive algorithm and executed efficiently. Siuzdak [12] derived a novel 2D filter based on three conditions: (1) circular symmetry, (2) zero mask correlation with a constant background, and (3)

maximizing the ratio of the mask correlation with an edge to the noise output variance.

The LoG filter proposed by Marr and Hildreth [8] is a famous edge detector. The input image was convolved with LoG mask. Then, zero-crossings of the filtered image were detected as the edges. It has two additional advantages. Firstly, the LoG filter is an orientation-independent operator. Secondly, the detected edges form closed curves. The LoG filter has shortcomings. Ulupinar and Medioni [14] proposed a simple test to eliminate false zero-crossings and a technique to correct the position of zero-crossings. Gunn [5] analyzed the sampling and truncation effects of the LoG filter and proposed a discrete formulation for implementation. In addition, the LoG filter is computationally inefficient. Many algorithms have been proposed to improve the speed of convolution [1,4,6,7,10,13,15].

Previous implementations [4,6,10] have relied on approximating the LoG operator by a difference of two Gaussian functions (DoG) having different space

* Corresponding author. Fax: +886-2-23671909.
E-mail address: pei@cc.ee.ntu.edu.tw (S.-C. Pei).

constants. The DoG filter has the advantage of separability, but it is only an approximation to the LoG filter. Instead of approximation, Wiejak [15] showed that the LoG filter may be decomposed into sum of two separable filters:

$$\text{LoG}(x, y) = -G''(x)G(y) - G(x)G''(y), \quad (1)$$

where G and G'' are the 1D Gaussian and the second derivative of 1D Gaussian, respectively. The number of multiplications per pixel is twice that of the DoG approximation. However, this filter exactly equals to the LoG filter.

Huertas and Medioni [7] reduced the image resolution by a factor K . Then, the reduced image was convolved with a smaller LoG filter with a space constant of σ/K . The result is very similar to the result of convolving the full resolution image with the LoG filter with space constant σ . However, this method introduces the aliasing effect. To improve this method, Chen et al. [1] decomposed the LoG filter into a Gaussian and a smaller LoG filter. The Gaussian filter was applied before decimation to control aliasing. Sotak and Boyer [13] further improved this method by implementing a separated LoG filter, establishing a criterion for selecting reconstruction constant, and establishing limits on the LoG and Gaussian mask sizes.

In this paper, the bilevel Laplacian-of-Gaussian (BLoG) filter is proposed to approximate the LoG filter. Only two levels are allowed in the filter impulse response. This approximation is formulated as an optimization problem. The parameters of the BLoG filter are adapted to minimize error norm of the approximation. The BLoG filter is just an approximation of the LoG filter, but it can be computed very efficiently, because only two multiplications are required for computing each pixel.

The design method of 1D BLoG filter is proposed in Section 2. Two methods of 2D filter design are given in Section 3. The performance of the BLoG filter in applying to the corner and edge detections are discussed in Section 4. In Section 5, the computational complexity of the proposed BLoG filter is analyzed. Conclusions are presented in Section 6.

2. Design of FIR bilevel LoG filter

The LoG filter is defined by

$$\text{LoG}(n, \sigma) = \frac{1}{\sqrt{2\pi}\sigma^3} \left(1 - \frac{n^2}{\sigma^2}\right) \exp\left(-\frac{n^2}{2\sigma^2}\right), \quad (2)$$

where σ is the filter scale (space constant). The BLoG is specified by the four parameters N_1, N_2, F_1 , and F_2 , and is defined by

$$\text{BLoG}(n, N_1, N_2, F_1, F_2) = \begin{cases} F_1, & |n| \leq N_1, \\ F_2, & N_1 < |n| \leq N_2, \\ 0, & |n| > N_2. \end{cases} \quad (3)$$

Our purpose is to determine the parameters appropriately, so that the BLoG filter best approximates the LoG filter.

In conventional optimal FIR filter design techniques, the filter coefficients are adapted to minimize the error between frequency response of the FIR filter and the desired frequency response. It requires an FFT computation to obtain frequency response of the FIR BLoG filter from its filter parameters. Therefore, an FFT computation is required for each adaptation of the filter parameters. This is very time consuming.

The desired impulse response in our problem has the property of good locality, which is completely different from the impulse response of any ideal frequency-selective filter in conventional design problems. So that, the approximation error between impulse responses of the BLoG filter and the LoG filter can be calculated easily. In this paper, the approximation criterion for the optimization procedure is to minimize the error between impulse responses of the BLoG filter and the LoG filter. This will save the time for computing the FFT. The optimization problem is formulated as follows:

Determine the parameters N_1, N_2, F_1 , and F_2 of the BLoG filter to minimize the error function

$$E_p(N_1, N_2, F_1, F_2) = \|\text{BLoG}(n, N_1, N_2, F_1, F_2) - \text{LoG}(n, \sigma)\|_p, \quad (4)$$

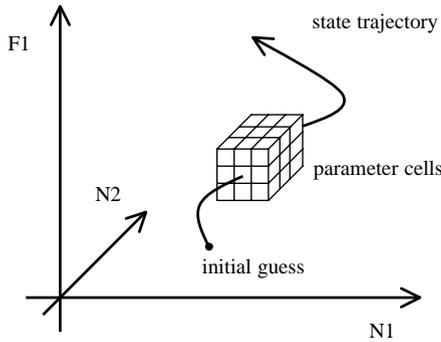


Fig. 1. Illustration of the gradient descent algorithm.

where $\|\cdot\|_p$ is the p -norm, subject to the constraint that the BLoG filter has zero DC response.

This is a highly nonlinear problem. A discrete version of the gradient descent algorithm is developed to minimize the error function. We adapt N_1 , N_2 , and F_1 to minimize the error function. F_2 is reserved to meet the constraint of zero DC response. As shown in Fig. 1, the parameter space of N_1 , N_2 , and F_1 is quantized into discrete cells. Each cell has a corresponding error value. We guess an initial state of the parameters. Then, apply the gradient descent algorithm to minimize the error value. The initial guess plays a very important role in the convergence of the gradient descent algorithm. An unsuitable initial state may trap the process into local minimum of the error function. An proper initial guess is given as follows:

1. $N_1 = \sigma$. If σ is not an integer, round to its nearest integer.
2. F_1 is determined by different optimization criteria:

• L_1 -norm

$$F_1 = \text{LoG}\left(\frac{N_1}{2}, \sigma\right). \quad (5)$$

• L_2 -norm

$$F_1 = \frac{1}{2N_1 + 1} \sum_{n=-N_1}^{N_1} \text{LoG}(n, \sigma). \quad (6)$$

• L_∞ -norm

$$F_1 = \frac{1}{2} \text{LoG}(0, \sigma). \quad (7)$$

3. N_2 is given by

$$N_2 = 3N_1. \quad (8)$$

4. F_2 is given by

$$F_2 = \frac{-F_1(2N_1 + 1)}{2(N_2 - N_1)}. \quad (9)$$

$N_1 = \sigma$ is the zero-crossing of the LoG filter. The values of F_1 are optimal provided that $N_1 = \sigma$ is fixed. To obtain zero DC response, F_2 is determined by the equation

$$(2N_1 + 1)F_1 + 2(N_2 - N_1)F_2 = 0. \quad (10)$$

The initial states corresponding to different optimization criteria are plotted in Fig. 2, where the LoG filter with $\sigma = 10$ is approximated.

As shown in Fig. 2(a), the L_1 -norm of approximation error is the sum of Regions I, II, and III. Its minimum value occurs at $F_1 = \text{LoG}(N_1/2, \sigma)$, where the lower boundary length of Region III equals to sum of the upper boundary lengths of Regions I and II. A lower or a higher F_1 value will lead to a positive sum of incremental areas. The optimal values of F_1 corresponding to L_2 -norm criterion can be obtained by

$$\begin{aligned} \frac{d\{\sum_{n=-N_1}^{N_1} [F_1 - \text{LoG}(n, \sigma)]^2\}^{1/2}}{dF_1} &= 0, \\ \frac{d\sum_{n=-N_1}^{N_1} [F_1 - \text{LoG}(n, \sigma)]^2}{dF_1} &= 0, \\ \sum_{n=-N_1}^{N_1} 2[F_1 - \text{LoG}(n, \sigma)] &= 0, \end{aligned} \quad (11)$$

$$\sum_{n=-N_1}^{N_1} F_1 = \sum_{n=-N_1}^{N_1} \text{LoG}(n, \sigma),$$

$$F_1 = \frac{1}{2N_1 + 1} \sum_{n=-N_1}^{N_1} \text{LoG}(n, \sigma).$$

The optimal values of F_1 corresponding to L_∞ -norm criterion can be obtained by

$$\begin{aligned} \min_{F_1} [&|F_1 - \text{LoG}(0, \sigma)|, |F_1 - \text{LoG}(N_1, \sigma)|] \\ &= \min_{F_1} [|F_1 - \text{LoG}(0, \sigma)|, |F_1 - 0|], \\ F_1 &= \text{LoG}(0, \sigma)/2. \end{aligned} \quad (12)$$

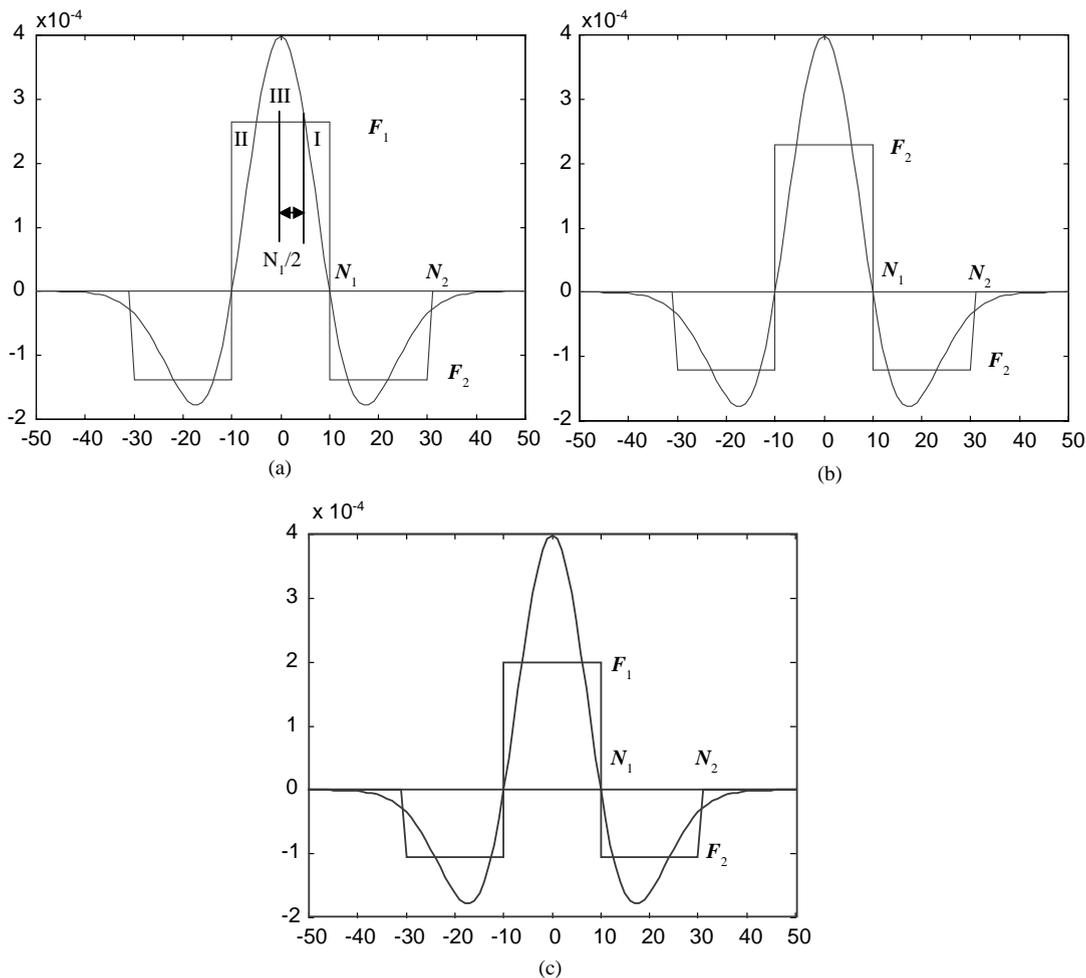


Fig. 2. The initial states for different optimization criteria: (a) L_1 -norm. (b) L_2 -norm. (c) L_∞ -norm.

The discrete version of gradient descent algorithm is summarized as follows:

1. Set initial state according to the above formula.
2. Quantize the parameter space (N_1, N_2, F_1) into discrete cells.
3. Calculate the error values for the $3 \times 3 \times 3$ neighboring parameter cells.
4. Move to the minimum error state among the $3 \times 3 \times 3$ parameter cells.
5. If the state is changed, go to step 3; else, stop.

The parameters N_1 and N_2 are discrete. Therefore, the only parameter to quantize is F_1 . The size of pa-

parameter cell relates to the precision of F_1 . In order to obtain the accurate value of F_1 efficiently, we recursively reduce the step size of F_1 until a predefined accuracy. The initial step size of F_1 is ten percent of its initial value. Then, each time when the gradient descent process converges, the step size is reduced to ten percent of the current size. This recursion proceeds until the step size is smaller than the error tolerance.

A design example of approximating the LoG filter with $\sigma = 10$ is shown in Fig. 3. The impulse responses of the resulting BLoG filters for different optimization criteria are plotted in Fig. 3(a), (c), and (e). Their

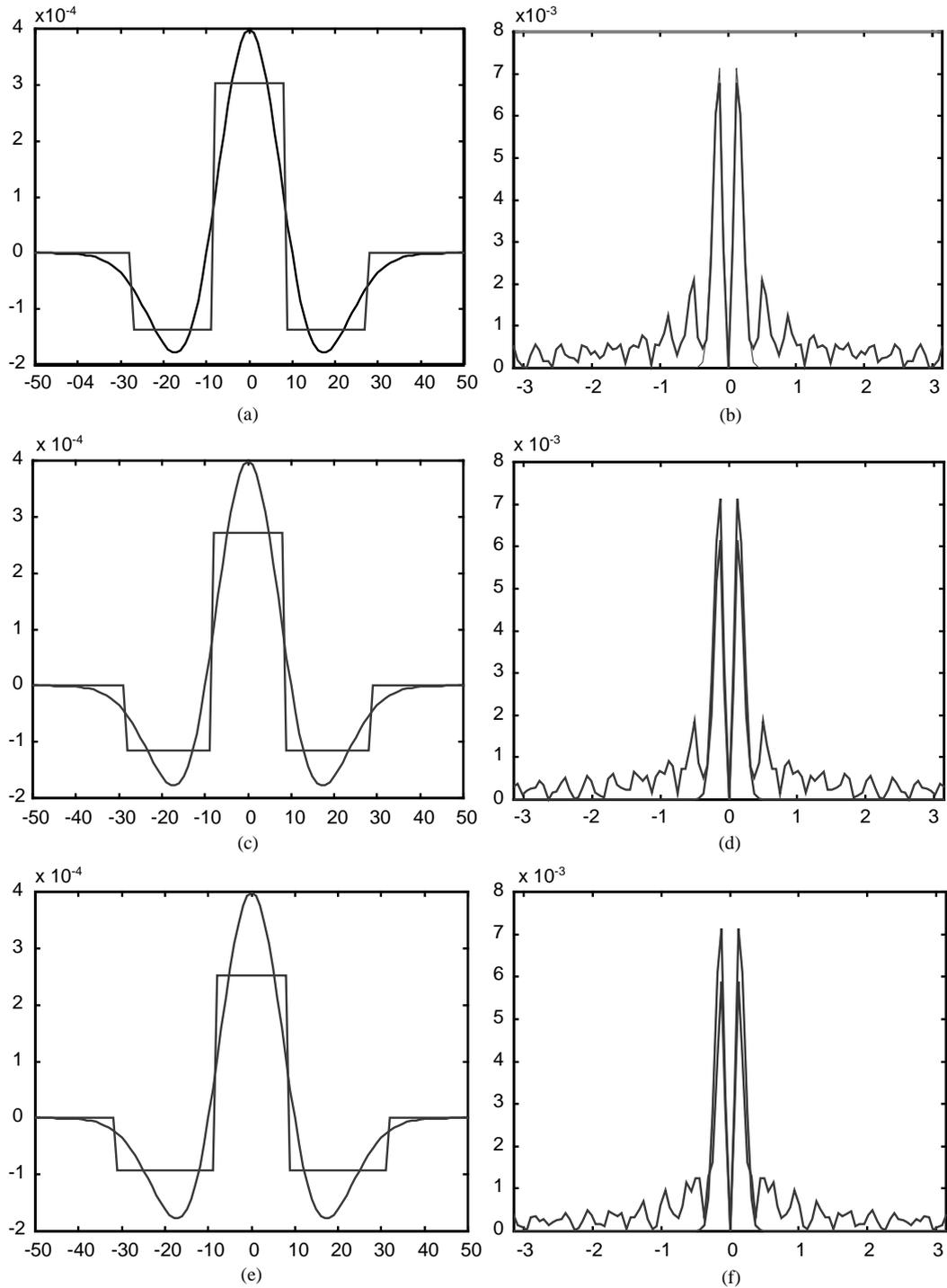


Fig. 3. Impulse and frequency responses of the BLoG filter obtained by different optimization criteria. In all cases, the LoG filter with $\sigma = 10$ is approximated. (a), (b) L_1 -norm criterion. (c), (d) L_2 -norm criterion. (e), (f) L_∞ -norm criterion.

Table 1

The initial and optimal parameter values of the 1D BLoG filter for different optimization criteria. The 1D LoG filter with $\sigma = 10$ is approximated

Criterion	N_{1_ini}	N_{2_ini}	F_{1_ini}	F_{2_ini}	N_{1_opt}	N_{2_opt}	F_{1_opt}	F_{2_opt}
L_1 -norm	10	30	2.64	-1.39	8	27	3.04	-1.36
L_2 -norm	10	30	2.30	-1.21	8	28	2.71	-1.15
L_∞ -norm	10	30	1.99 $\times 10^{-4}$	-1.05 $\times 10^{-4}$	8	31	2.52 $\times 10^{-4}$	-0.93 $\times 10^{-4}$

Table 2

The initial and optimal parameter values of the 1D BLoG filter for approximating different scales of 1D LoG filter. The L_1 -norm optimization criterion is applied

Scale	N_{1_ini}	N_{2_ini}	F_{1_ini}	F_{2_ini}	N_{1_opt}	N_{2_opt}	F_{1_opt}	F_{2_opt}
7	7	21	6.65	-3.56	6	19	7.22	-3.61
8	8	24	5.16	-2.74	6	21	6.22	-2.69
9	9	27	3.24	-1.71	8	24	3.43	-1.82
10	10	30	2.64	-1.39	8	27	3.04	-1.36
11	11	33	1.81	-0.95	9	29	2.14	-1.02
12	12	36	1.53 $\times 10^{-4}$	-0.80 $\times 10^{-4}$	10	32	1.57 $\times 10^{-4}$	-0.75 $\times 10^{-4}$

corresponding frequency responses are plotted in Fig. 3(b), (d), and (f), respectively. The desired impulse response or frequency response is also provided in each figure. The parameters of the resulting BLoG filters are listed in Table 1. Because the initial states are properly assigned, only several iterations are required in all cases. The parameters of the BLoG filters for approximating the LoG filters with different σ values under L_1 -norm criterion are provided in Table 2.

3. Design of 2D FIR bilevel LoG filter

To design the 2D BLoG filter, two approaches are proposed. The first is to generalize the algorithm for 1D case. The second is to perform the McClellan transformation of the 1D impulse response.

3.1. Generalizing the 1D algorithm

The 2D LoG filter is given by

$$\text{LoG}(x, y, \sigma) = \frac{1}{2\pi\sigma^4} \left(2 - \frac{x^2 + y^2}{\sigma^2} \right) \exp\left(-\frac{x^2 + y^2}{2\sigma^2} \right). \quad (13)$$

The 2D BLoG filter is specified by the four parameters $R_1, R_2, F_1,$ and F_2 (see Fig. 4), and is defined by

$$\text{BLoG}(x, y, R_1, R_2, F_1, F_2) = \begin{cases} F_1, & x^2 + y^2 \leq R_1^2, \\ F_2, & R_1^2 < x^2 + y^2 \leq R_2^2, \\ 0, & x^2 + y^2 > R_2^2. \end{cases} \quad (14)$$

The error function for the 2D filter design is defined by

$$E_p(R_1, R_2, F_1, F_2) = \|\text{BLoG}(x, y, R_1, R_2, F_1, F_2) - \text{LoG}(x, y, \sigma)\|_p. \quad (15)$$

According to the same concepts with the 1D case, the initial states for different criteria are provided as follows:

- $R_1 = \sqrt{2}\sigma$. If $\sqrt{2}\sigma$ is not an integer, round it to the nearest integer.
- F_1 is determined by different optimization criteria:
 - L_1 -norm

$$F_1 = \text{LoG}\left(\frac{R_1}{\sqrt{2}}, 0, \sigma \right). \quad (16)$$

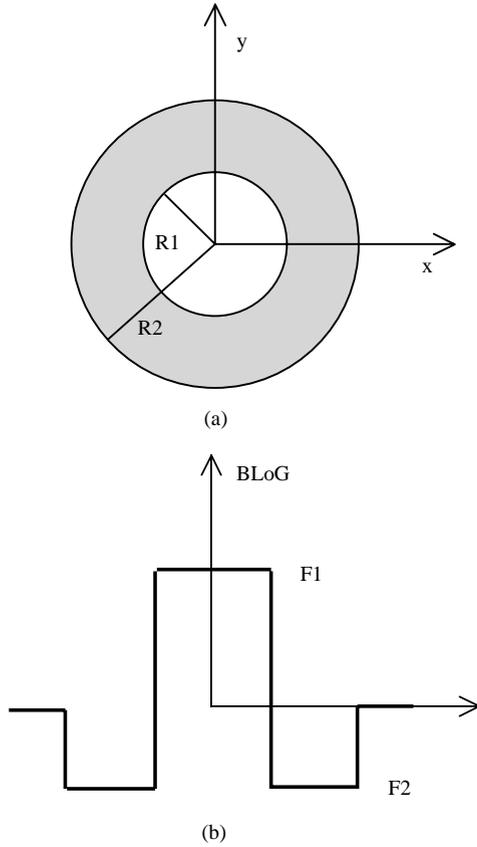


Fig. 4. The two-dimensional BLoG filter: (a) top-view, and (b) the cross section of $y = 0$.

- L_2 -norm

$$F_1 = \frac{\sum \sum_{x^2+y^2 \leq R_1^2} \text{LoG}(x, y, \sigma)}{\sum \sum_{x^2+y^2 \leq R_1^2} 1}. \quad (17)$$

- L_∞ -norm

$$F_1 = \frac{1}{2} \text{LoG}(0, 0, \sigma). \quad (18)$$

- R_2 is given by

$$R_2 = 2R_1. \quad (19)$$

- F_2 is given by

$$F_2 = \frac{-F_1 \sum \sum_{x^2+y^2 \leq R_1^2} 1}{\sum \sum_{R_1^2 < x^2+y^2 \leq R_2^2} 1}. \quad (20)$$

The gradient descent algorithm is then applied to adapt the parameters R_1 , R_2 , and F_1 . The 2D LoG

filter with $\sigma = 10$ is plotted in Fig. 5(a). The impulse responses of the resulting 2D BLoG filters for different optimization criteria are plotted in Fig. 5(b), (c), and (d). The parameters of the resulting 2D BLoG filters are listed in Table 3. Because the initial states are properly assigned, only several iterations are required in all cases. The parameters of the 2D BLoG filters for approximating the LoG filters with different σ values under L_1 -norm criterion are provided in Table 4.

3.2. Using the McClellan transformation

The McClellan transformation [3] is a transformation that converts an 1D zero-phase FIR filter into a multidimensional one through a substitution of variables. Assume that the impulse response $h[n]$ of the 1D zero-phase FIR filter is real. Then, $h[-n] = h[n]$ and the frequency response can be written as

$$\begin{aligned} H(\omega) &= h[0] + \sum_{n=1}^N h[n][\exp(-j\omega n) + \exp(j\omega n)] \\ &= \sum_{n=0}^N a[n] \cos(\omega n) \\ &= \sum_{n=0}^N a[n] T_n[\cos(\omega)], \end{aligned} \quad (21)$$

where

$$a[n] \triangleq \begin{cases} h[0], & n = 0, \\ 2h[n], & n > 0 \end{cases}$$

and $T_n[\cdot]$ is the n th Chebyshev polynomial given by

$$\begin{aligned} T_0[x] &= 1, \\ T_1[x] &= x, \\ T_n[x] &= 2xT_{n-1}[x] - T_{n-2}[x]. \end{aligned}$$

If we make a substitution of variables

$$F(\omega_x, \omega_y) \rightarrow \cos(\omega), \quad (22)$$

we obtain the 2D frequency response

$$H(\omega_x, \omega_y) = \sum_{n=0}^N a[n] T_n[F(\omega_x, \omega_y)]. \quad (23)$$

Note that, the contours of $F(\omega_x, \omega_y)$ are also the contours of $H(\omega_x, \omega_y)$.

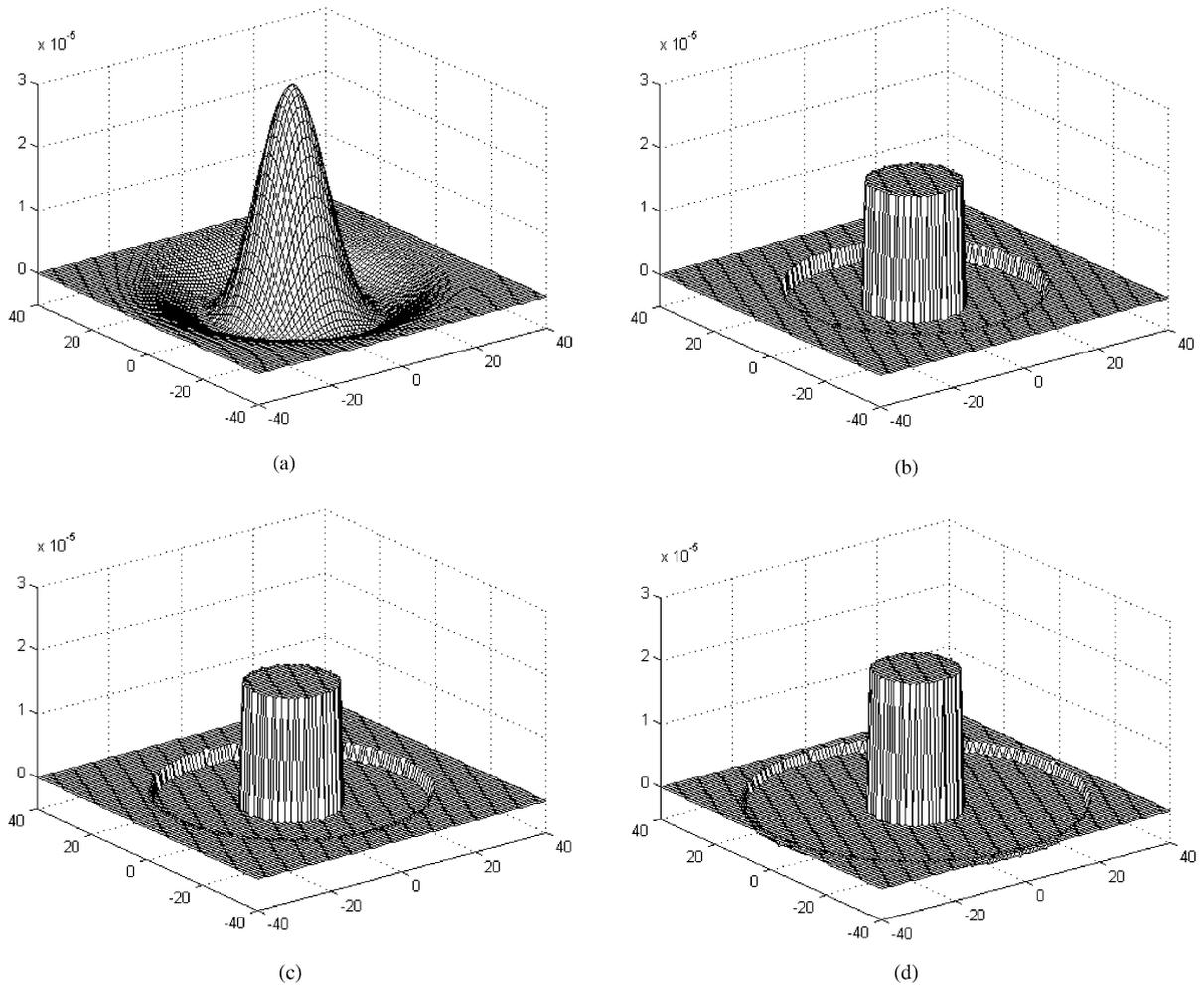


Fig. 5. Impulse response of the 2D BLoG filter obtained by different optimization criteria. The 2D LoG filter with $\sigma = 10$ is approximated: (a) The 2D LoG filter with $\sigma = 10$, (b) L_1 -norm criterion, (c) L_2 -norm criterion, and (d) L_∞ -norm criterion.

Table 3

The initial and optimal parameter values of the 2D BLoG filter for different optimization criteria. The 2D LoG filter with $\sigma = 10$ is approximated

Criterion	R_1 _ini	R_2 _ini	F_1 _ini	F_2 _ini	R_1 _opt	R_2 _opt	F_1 _opt	F_2 _opt
L_1 -norm	14	28	0.97	-0.32	11	29	1.69	-0.28
L_2 -norm	14	28	1.20	-0.40	11	31	1.70	-0.24
L_∞ -norm	14	28	1.59	-0.53	10	38	2.07	-0.16
			$\times 10^{-5}$	$\times 10^{-5}$			$\times 10^{-5}$	$\times 10^{-5}$

The proposed 1D BLoG filter is a 1D zero-phase FIR filter with real impulse response. If we perform McClellan transformation of the 1D BLoG filter with the transformation

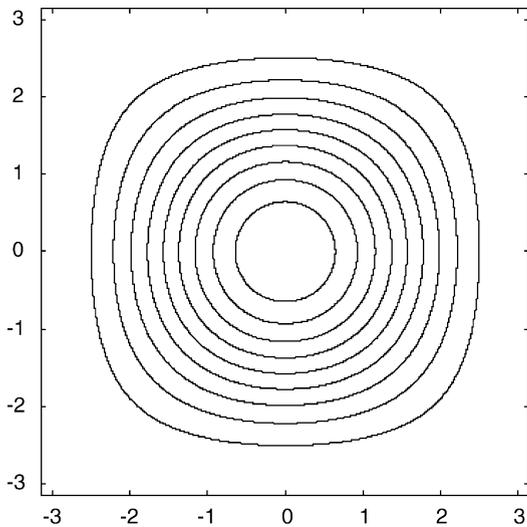
function

$$F(\omega_x, \omega_y) = \frac{1}{2}[-1 + \cos(\omega_x) + \cos(\omega_y) + \cos(\omega_x)\cos(\omega_y)], \tag{24}$$

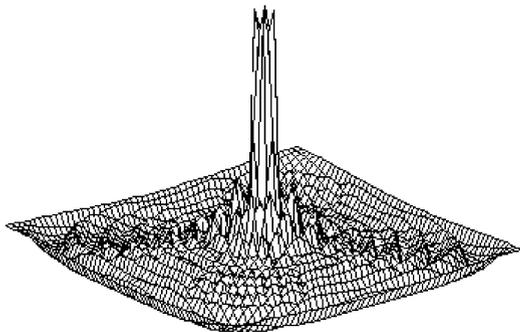
Table 4

The initial and optimal parameter values of the 2D BLoG filter for approximating different scales of 2D LoG filter. The L_1 -norm optimization criterion is applied

Scale	R_{1_ini}	R_{2_ini}	F_{1_ini}	F_{2_ini}	R_{1_opt}	R_{2_opt}	F_{1_opt}	F_{2_opt}
5	7	14	1.54×10^{-4}	-0.50×10^{-4}	6	15	2.35×10^{-4}	-0.45×10^{-4}
8	11	22	2.36	-0.78	9	23	3.90	-0.71
10	14	28	0.97×10^{-5}	-0.32×10^{-5}	11	29	1.69×10^{-5}	-0.28×10^{-5}



(a)



(b)

Fig. 6. Illustration of the McClellan transformation: (a) the contours of the transformation function, and (b) the frequency response of the 2D filter obtained by the McClellan transformation.

we obtain a 2D filter. Fig. 6(a) gives the contours of the transformation function. The frequency response

of the resulting 2D filter along the horizontal or the vertical axes is exactly the same as that of the 1D BLoG filter. The frequency response of the resulting 2D filter is plotted in Fig. 6(b). Realization of the 2D filter designed by the McClellan transformation is shown in Fig. 7, which is a special case of the network proposed by McClellan and Chan [9]. In this example, the impulse response of the 1D filter is given by

$$h[n] = \begin{cases} h+, & |n| \leq 3, \\ h-, & 3 < |n| \leq 6, \\ 0, & |n| > 6. \end{cases} \quad (25)$$

The convolution mask of the network block F in Fig. 7 is given by

$$\begin{bmatrix} +\frac{1}{8} & +\frac{1}{4} & +\frac{1}{8} \\ +\frac{1}{4} & -\frac{1}{2} & +\frac{1}{4} \\ +\frac{1}{8} & +\frac{1}{4} & +\frac{1}{8} \end{bmatrix}.$$

Only three power-of-two multiplications and eight additions per pixel are required in convolving with this mask. Although the resulting 2D filter is not a BLoG filter by definition, it operates in a way that is similar to the BLoG filter. That is, only two multiplications, except for power-of-two multiplications, are required for computing each pixel.

4. Applications

Corner detection and edge detection are analogue problems. Corners are the abrupt changes of tangent angle in the 1D arc length space, while edges are the abrupt changes of intensity in the 2D image space.

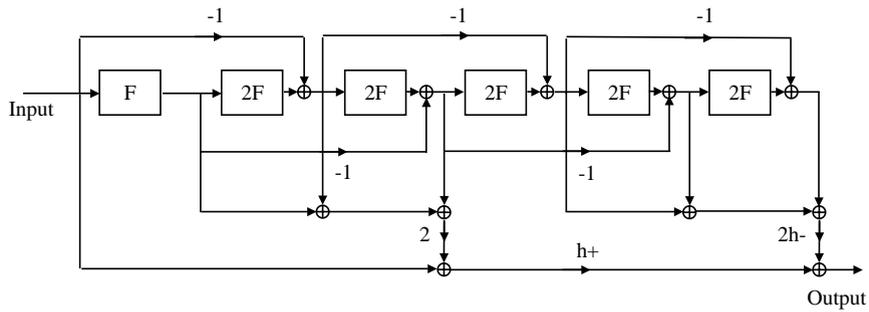


Fig. 7. Realization of the McClellan transformation. In this example, the parameters of the 1D filter are $N_1 = 3$, $N_2 = 6$, $F_1 = h+$, $F_2 = h-$.

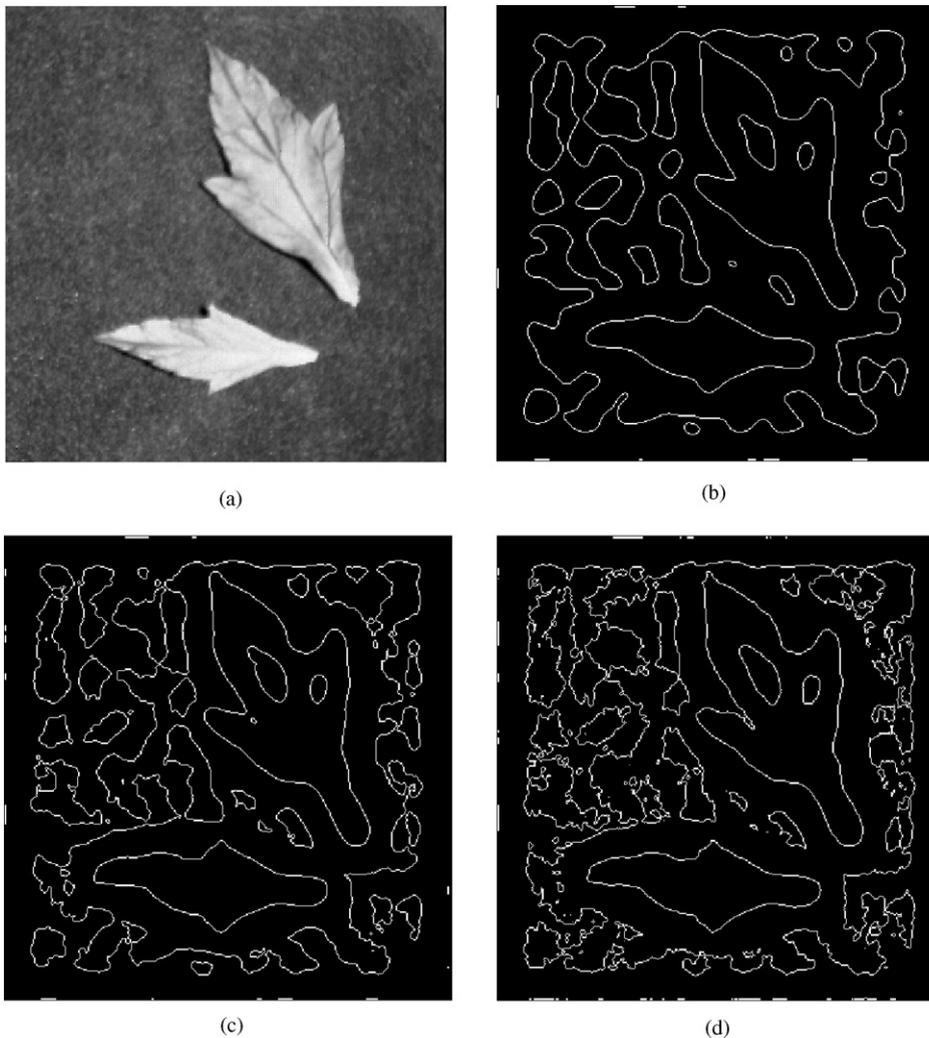


Fig. 8. Comparison of edge images detected by different filters: (a) a gray level image of two leaves, (b) the edges detected by 2D LoG filter with $\sigma = 10$, (c) the edges detected by 2D BLoG filter, and (d) the edges detected by the McClellan transformation of 1D BLoG filter.

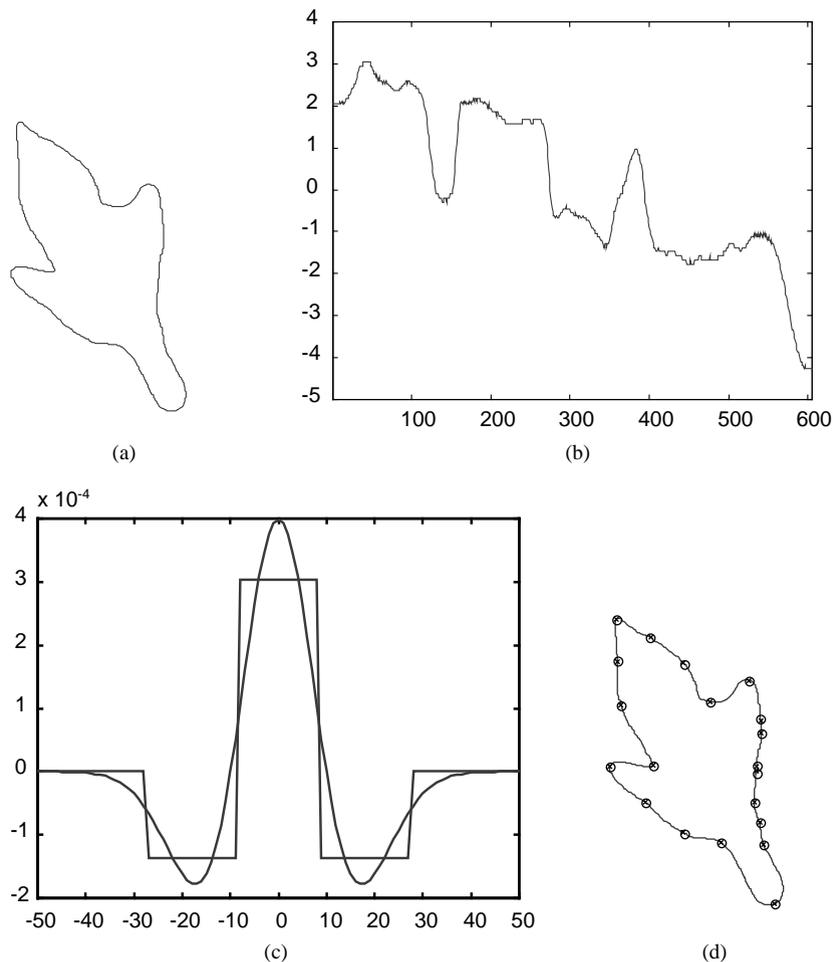


Fig. 9. Comparison of the corners detected by LoG and BLoG filters: (a) A leaf boundary extracted from (b), (b) the tangent angle versus arc length, (c) the applied 1D LoG and BLoG filters, and (d) the detected corners, where crosses represent the corners obtained by LoG filtering and circles represent the corners obtained by BLoG filtering.

The LoG filter is a band-pass filter that can detect abrupt changes of a signal and suppress noise in the high-frequency band. The 1D LoG filter can be applied to detect corners, while the 2D LoG filter can be applied to detect edges. We verify the applicability of the proposed BLoG filter by applying to the detection of corners and edges.

4.1. Detecting edges of gray level image

A real image of two leaves captured by digital camera is shown in Fig. 8(a). The image size is

338×347 pixels with 256 gray levels. This image is convolved with the 2D LoG filter with $\sigma = 10$. Then, zero-crossings of the resulting image are detected as edge points. The edge image is shown in Fig. 8(b), where the edges are represented by white pixels. The edge images obtained by applying the 2D BLoG filter and the McClellan transformation of 1D BLoG filter are shown in Fig. 8(c) and (d), respectively. In all three cases, the boundaries of the two leaves are successfully detected. However, the 2D BLoG filter gives better result than the McClellan transformation of 1D BLoG filter in details.

4.2. Detecting corners of planar shape

The boundary of the larger leaf in Fig. 8(b) is extracted and plotted in Fig. 9(a). The tangent angle versus arc length is plotted in Fig. 9(b). The 1D LoG filter with $\sigma = 10$ and its approximating BLoG filter (see Fig. 9(c)) are convolved with the tangent angle function of the leaf boundary. The zero-crossings of the LoG and the BLoG filtering results are marked by small crosses and circles, respectively (see Fig. 9(d)). As shown in the figure, the corners detected by the LoG and the BLoG filter coincide with each other.

5. Computational complexity

In this section, a very efficient edge detector called the infinite size symmetric exponential filter (ISEF) [11] is introduced first. Then, we compare the computational complexity of the LoG filter, the ISEF, and the proposed BLoG filter.

5.1. The infinite size symmetric exponential filter

Although the ISEF is different from the LoG filter, it is an efficient edge detector that can be realized by a recursive algorithm. The 1D ISEF is given by

$$f_L(x) = Ca_0(1 - a_0)^{|x|} = f_1(x) * f_2(x), \tag{26}$$

where

$$f_1(x) = \begin{cases} a_0(1 - a_0)^x, & x \geq 0, \\ 0, & x < 0, \end{cases}$$

$$f_2(x) = \begin{cases} 0, & x > 0, \\ a_0(1 - a_0)^{-x}, & x \leq 0. \end{cases}$$

Supposing $I(x)$ as the input signal, $I_1(x) = I(x) * f_1(x)$, and $I_2(x) = I(x) * f_2(x)$, we have the recursive algorithm:

$$I_1(x) = I_1(x - 1) + a_0(I(x) - I_1(x - 1)), \tag{27}$$

$$I_2(x) = I_2(x + 1) + a_0(I(x) - I_2(x + 1)). \tag{28}$$

In convolving with $f_1(x)$ or $f_2(x)$, only one multiplication and two addition operations are

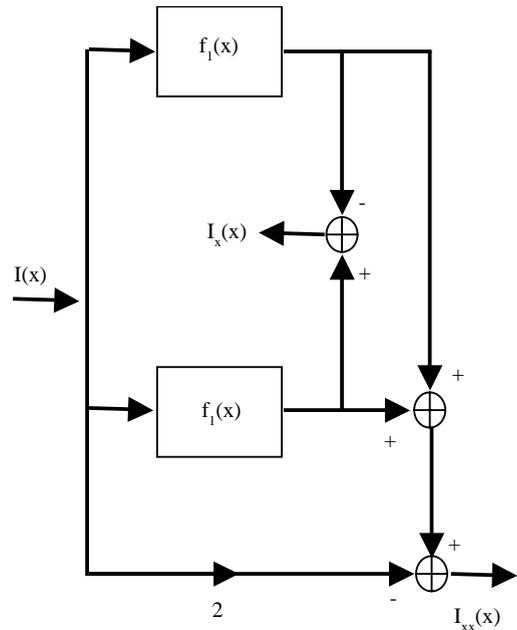


Fig. 10. Realization of the 1D ISEF.

required. The first and the second derivative $I_x(x) = \partial(I(x) * f_L(x))/\partial x$ and $I_{xx}(x) = \partial^2(I(x) * f_L(x))/\partial x^2$ are given by

$$I_x(x) = I(x) * (f_2(x) - f_1(x)), \tag{29}$$

$$I_{xx}(x) = I(x) * (f_2(x) + f_1(x)) - 2I(x) * f_1(x) * f_2(x). \tag{30}$$

The realization of the 1D ISEF is shown in Fig. 10. For the 2D case, the first and the second derivative $I_x(x, y) = \partial(I(x, y) * f_L(x, y))/\partial x$ and $I_{xx}(x, y) = \partial^2(I(x, y) * f_L(x, y))/\partial x^2$ are given by

$$I_x(x, y) = I(x, y) * f_1(y) * f_2(y) * (f_2(x) - f_1(x)), \tag{31}$$

$$I_{xx}(x, y) = I(x, y) * f_1(y) * f_2(y) * (f_2(x) + f_1(x)) - 2I(x, y) * f_1(x) * f_2(x). \tag{32}$$

Similarly,

$$I_y(x, y) = I(x, y) * f_1(x) * f_2(x) * (f_2(y) - f_1(y)). \tag{33}$$

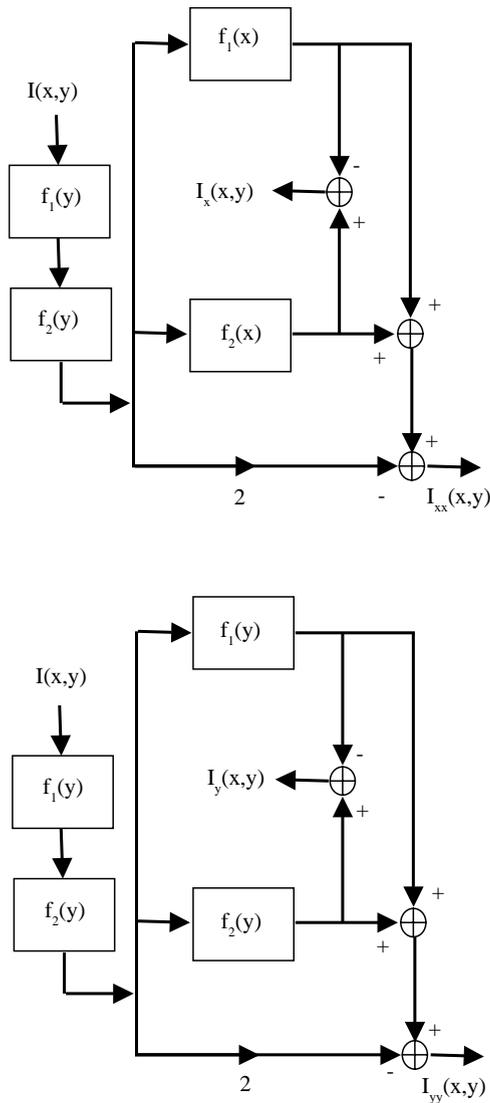


Fig. 11. Realization of the 2D ISEF.

$$I_{yy}(x, y) = I(x, y) * f_1(x) * f_2(x) * (f_2(y) + f_1(y)) - 2I(x, y) * f_1(y) * f_2(y). \quad (34)$$

The realization of the 2D ISEF is shown in Fig. 11. The edges of an image can be detected from the maxima of gradient or zeros of the second directional derivative along the gradient by use of the differential operators of exponential filter.

Table 5

The comparison of computational complexity for the 1D LoG filter, the ISEF, and the BLoG filter. The numbers of addition and multiplication operations for different filter scales are listed

	Filter scale	Window size	Addition	Multiplication
1D LoG	5	20	40	41
	10	40	80	81
	15	60	120	121
ISEF	—	—	7	3
1D BLoG	5	13	5	2
	10	27	5	2
	15	41	5	2

5.2. Computational complexity

The numbers of addition and multiplication operations for the LoG filter, the ISEF, and the proposed BLoG filter are listed as follows:

LoG

- The convolution with 1D LoG filter of length $2N + 1$ requires $2N$ additions and $2N + 1$ multiplications per point.
- The convolution with 2D LoG filter of size $(2N + 1) \times (2N + 1)$ requires $4N^2 + 4N$ additions and $4N^2 + 4N + 1$ multiplications per pixel.

ISEF

- The 1D ISEF requires only three multiplications and seven additions per point.
- The 2D ISEF requires only 10 multiplications and 22 additions per pixel.

BLoG

- The convolution with 1D BLoG filter of length $2N + 1$ requires only $2N$ additions and two multiplications per point. If we incrementally compute the signal points entering and leaving a constant region of the mask, it requires only five additions (two points for inner region and four points for outer region) and two multiplications per point.
- The convolution with 2D BLoG filter of size $(2N + 1) \times (2N + 1)$ requires only $4N^2 + 4N$ additions and two multiplications per pixel. If we incrementally compute the image pixels entering and leaving a constant region of the mask, it requires only $4R_2 + 8R_1$ additions ($4R_1$ points for

Table 6

The comparison of computational complexity for the 2D LoG filter, the ISEF, the BLoG filter, and the McClellan filter. The numbers of addition, multiplication, and shift operations for different filter scales are listed

	Filter scale	Window size	Addition	Multiplication	Shift
2D LoG	5	20	1680	1681	—
	10	40	6560	6561	—
	15	60	14640	14641	—
ISEF	—	—	22	10	—
2D BLoG	5	15	108	2	—
	10	29	204	2	—
	15	44	304	2	—
McClellan	5	13	129	2	40
	10	27	269	2	82
	15	41	409	2	124

inner region and $4R_2 + 4R_1$ points for outer region) and two multiplications per pixel.

- The convolution with the McClellan transformation of 1D BLoG filter (see Fig. 8) requires only $10N - 1$ additions, $3N + 1$ power-of-two multiplications, and two ordinary multiplications per pixel.

To illustrate this, some design examples are presented in Tables 5 and 6. In fact, the window size of LoG filter is larger than the window size of its approximating BLoG filter. Therefore, the number of additions for the LoG filter is larger than the number of additions for its approximating BLoG filter. The region of support for LoG filter is given by 4σ , and that for BLoG filter is automatically determined by the design algorithm. In convolving with the 1D LoG filter with $\sigma = 10$, the region of support is 40, 80 addition and 81 multiplication operations are required. In convolving with its approximating 1D BLoG filter, the region of support is 27, five addition and two multiplication operations are required. In convolving with 2D LoG filter with $\sigma = 10$, 6560 addition and 6561 multiplication operations are required. Only 204 addition and two multiplication operations are required in convolving with its approximating BLoG filter; 269 addition, two ordinary multiplication, and 82 power-of-two multiplication operations are required in convolving with the corresponding 2D filter designed by the McClellan transformation. Considerable reduction in computational complexity is achieved. In both 1D and 2D cases, the ISEF requires more multiplications and less additions than the new proposed BLoG filter.

Note that although we compare the computational complexity of the BLoG filter and the ISEF, they are different in design criterion (see the Introduction). The main difference is that the ISEF is an IIR filter while the LoG filter is an FIR filter. Therefore, the ISEF does not have the property of finite support. Its edge detection result will be influenced by neighboring edges.

6. Conclusions

In this paper, the bilevel Laplacian-of-Gaussian filter and its design method are proposed. In convolving with the BLoG filter, only two multiplication operations per pixel are required. If the McClellan transformation is applied, the number of addition operations can be largely reduced by performing a small number of shift operations (power-of-two multiplications) instead. The BLoG filter is applied to detect corners of planar curves and edges of gray level images, satisfactory results are obtained.

References

- [1] J.S. Chen, A. Huertas, G. Medioni, Fast convolution with Laplacian-of-Gaussian masks, *IEEE Trans. Pattern Anal. Machine Intell.* PAMI-9 (4) (July 1987) 584–590.
- [2] F.M. Dickey, K.S. Shanmugam, Optimal edge detection filter, *Appl. Opt.* 16 (1) (1977) 145–148.
- [3] D.E. Dudgeon, *Multidimensional Digital Signal Processing*, Prentice-Hall, Englewood Cliffs, NJ, 1984, Section 3.5.

- [4] W.E.L. Grimson, Aspects of a computational theory of human stereo vision, Proceedings DARPA Image Understanding Workshop, April 1980, pp. 128–149. College Park, Maryland, USA.
- [5] S.R. Gunn, On the discrete representation of the Laplacian of Gaussian, *Pattern Recognition* 32 (1999) 1463–1472.
- [6] E.C. Hildreth, The detection of intensity changes by computer and biological vision system, *Comput. Graphics Image Process.* 22 (1983) 1–27.
- [7] A. Huertas, G. Medioni, Detection of intensity changes with subpixel accuracy using Laplacian–Gaussian masks, *IEEE Trans. Pattern Anal. Machine Intell. PAMI-8* (5) (September 1986) 651–664.
- [8] D. Marr, E. Hildreth, Theory of edge detection, *Proc. Roy. Soc. London B207* (1980) 187–217.
- [9] J.H. McClellan, D.S.K. Chan, A 2D FIR filter structure derived from the Chebyshev recursion, *IEEE Trans. Circuits Systems CAS-24*(7) (1977) 372–378.
- [10] H.K. Nishihara, N.G. Larson, Towards a real-time implementation of the Marr and Poggio stereo matcher, Proceedings DARPA Image Understanding Workshop, April 1981, pp. 114–120. Washington DC, USA.
- [11] J. Shen, S. Castan, An optimal linear operator for edge detection, *Proc. CVPR'86*, Miami.
- [12] J. Siuzdak, A single filter for edge detection, *Pattern Recognition* 31 (11) (1998) 1681–1686.
- [13] G.E. Sotak, K.L. Boyer, The Laplacian-of-Gaussian kernel: a formal analysis and design procedure for fast, accurate convolution and full-frame output, *Comput. Graphics Image Process.* 48 (1989) 147–189.
- [14] F. Ulupinar, G. Medioni, Refining edges detected by a LoG operator, *Comput. Graphics Image Process.* 51 (1990) 275–298.
- [15] J.S. Wiegak, H. Buxton, B.F. Buxton, Convolution with separable masks for early image processing, *Comput. Graphics Image Process.* 32 (1985) 279–290.