COMPUTER VISION, GRAPHICS, AND IMAGE PROCESSING 45, 167–195 (1989)

A Nonlinear Laplace Operator as Edge Detector in Noisy Images

LUCAS J. VAN VLIET AND IAN T. YOUNG

Pattern Recognition Group of the Faculty of Applied Physics, Delft University of Technology, Lorentzweg 1, 2628 CJ Delft, The Netherlands

AND

GUUS L. BECKERS

Department of Medical Informatics, Erasmus University, The Netherlands

Received October 10, 1987; revised July 5, 1988

An edge detection scheme is developed robust enough to perform well over a wide range of signal-to-noise ratios. It is based upon the detection of zero crossings in the output image of a nonlinear Laplace filter. Specific characterizations of the nonlinear Laplacian are its adaptive orientation to the direction of the gradient and its inherent masks which permit the development of approximately circular (isotropic) filters. We have investigated the relation between the locally optimal filter parameters, smoothing size, and filter size, and the SNR of the image to be processed. A quantitative evaluation shows that our edge detector performs at least as well—and in most cases much better—than other edge detectors. At very low signal-to-noise ratios, our edge detector is superior to all others tested.

1. INTRODUCTION

One of the major problems in digital image processing and scene analysis is the extraction of useful and relevant edges. This problem may be approached as either a problem in image filtering (producing a binary image with single pixel thick edges starting from a grey-level input image) or a problem in image analysis (producing a data list—coordinate positions—where edge pixels can be found) starting from the same grey-level input image. In the research to be reported here we will take the former approach, that of image filtering. To obtain such a result it is necessary either to segment an image into regions separated by edges (*the region approach*) or to detect the edges directly (*the edge approach*). A discussion of the region approach is beyond the scope of this article. Surveys with applications have been given by Rosenfeld [1], Pratt [2], Duda [3], and Ballard [4].

The success of the edge approach to image segmentation is dependent upon the quality of the edges in the image itself as well as the specific algorithm used to locate them. Thus edge improvement (or *enhancement* as we shall refer to it here) as well as the specific edge detection algorithm is an important component in an edge detection system.

1.1. Edge Enhancement

In many applications we are confronted with low-quality images. Improving the quality of the image may thus be seen as a first step towards the edge detection phase. If we are dealing with motion-blurred or out-of-focus images, we can frequently make these more acceptable by image sharpening techniques. This

0734-189X/89 $3.00

approach is based upon a number of research models including physiological ones (Rosenfeld [1], Pratt [2], Ballard [4], Attneave [5], and Gonzalez [6]).

One example of edge enhancement is to create a small overshoot at the top and a small undershoot at the bottom of an edge slope, offering a sharper intensity change than actually occurs. This is similar to the Mach-effect in the human visual system and has been discussed in detail by Stockham [7].

Since edges are "high-frequency phenomena," edge enhancement can be achieved by emphasizing the high spatial frequencies. The Laplacian operator achieves this emphasis and is, therefore, frequently used as an edge enhancement operator:

$$I_{\text{enhanced}}(x, y) = I_{\text{blur}}(x, y) - k\nabla^2 I_{\text{blur}}(x, y). \tag{1}$$

Subtracting a positive fraction, **k**, of the Laplace from the blurred image will amplify the quality of the edges in the image. The choice of **k** is dependent upon the type of blur and its size. The Laplacian itself is a mathematical operator in continuous space and must be represented by an appropriate digital filter. This digital approximation, its rotation invariance, and its sensitivity to high-frequency noise will be discussed in Section 2.

1.2. Edge Detection

During the past two decades the development, implementation, and use of edge detection algorithms have been major topics of research. Many algorithms have been found that perform well in given applications but poorly in others. We can divide these algorithms into roughly two classes:

(1) Sequential algorithms. The already-detected edge pixels exert an influence on both the position of the next potential edge pixel and the result of its acceptance test. This is most vividly illustrated in the use of heuristic search algorithms for edge tracking (Lester [8], Montanari [9]). The disadvantage of many sequential techniques is that it is often necessary to have a considerable body of *a priori* knowledge about the shape or the position of the edges in the image to be processed. Examples of edge tracking algorithms have been presented by Rosenfeld [1], Davis [10], Martelli [11], and Gerbrands [12].

(2) Parallel algorithms. The detection of a pixel as an edge element is independent of the *results* elsewhere in the image.

The earliest and most popular edge detection schemes begin with the gradient operator. They use the fact that the absolute value of the first derivative will be maximal near the position of the steepest descent. Seeking the maximum of the gradient leads, of course, to a second derivative. A number of parallel edge detectors have been developed and they can be partitioned into four major types: (1) detectors that approximate the mathematical first derivative (gradient) or the second derivative (Laplacian); (2) template matching operators that use multiple templates at different orientations, of different sizes, and of different shape; (3) edge fitting operators that search for the best fit between the local neighborhood of an image and a parametric edge model; (4) morphologic operators that use an approach to image processing based upon set theoretic concepts of shape. The first three types cover many well-known edge detectors. For extended surveys we refer to Rosenfeld [1], Pratt [2], Ballard [4], and Peli [13]. The fourth type, however, is rather

new in the field of edge detection. In 1980 Goetcherian [14] proposed using the grey level equivalents of the binary erosion and dilation operators for this purpose.

1.3. Motivation for Study

It is known that the human visual system uses edge detection techniques in early vision. Marr [15, 16] and Hildreth [17] tried to understand and model this process and on the basis of neurophysiological studies they developed a computational model for edge detection. It has also been shown that human beings are able to recognize objects starting from a very crude outline (Attneave [5]). Edge detection may be the most important method of feature extraction in low level vision.

Despite their great importance, most edge detectors perform poorly in images with low signal-to-noise ratios. The objective of our research is, therefore, to develop an edge detection scheme robust enough to perform well over a wide range of signal-to-noise ratios. It is based upon the detection of zero crossings in the output of a "Laplace" filtered image.

In 1986 Beckers [18] proposed a *nonlinear* Laplace-like operator in a local 3×3 neighborhood which may perform better than a linear Laplace filter. In this project we have developed an edge detection scheme based upon the Marr–Hildreth model combined with the nonlinear Laplace operator. Further, we have extended the nonlinear Laplacian to larger filter sizes. In the following sections both the various Laplace filters considered as well as the edge detection model will be presented.

1.4. Proposal for Evaluation

What we seek as an edge corresponds in a physical world to an abrupt change in intensity. It is often the consequence of a change in some physical property such as reflection, illumination, shading, orientation, etc. In this article we will formally define an edge as:

> **edge**—a simply-connected contour, one pixel thick, at the center of the slope between two adjacent regions with a considerable difference in grey level.

Detecting edges satisfying this condition is not a trivial task and it will often occur that the detected edge has gaps at places where the transition between two adjacent regions is not sufficiently abrupt. Clearly the proficiency of our edge detection technique—the ability to minimize the number of gaps as well as place the edge contours at the "right" position—is a matter of central concern. Further, this proficiency must be compared to other techniques that have been proposed in the literature.

The choice of an evaluation (comparison) criteria is difficult and depends upon the accuracy requirements for the detected edges. We can, however, distinguish two classes of performance measures: quantitative and qualitative. Quantitative performance measures are needed for comparing various edge detectors as objectively as possible. In our evaluation we have chosen Pratt's figure-of-merit (Pratt [2]). A detailed description of this evaluation technique will be given in Section 4. The qualitative approach, being intrinsically subjective, leaves much to be desired. For the purposes of this article qualitative assessments can be performed by comparing the resulting figures.

2. LAPLACE FILTERS

When we consider Laplace filters we actually mean a discrete approximation to the mathematical Laplace operator:

$$\nabla^2 f(x, y) = \frac{\partial^2 f(x, y)}{\partial x^2} + \frac{\partial^2 f(x, y)}{\partial y^2}. \tag{2}$$

Note that the 2-dimensional Laplacian given by Eq. (2), is the second-order partial derivative in the orthogonal directions of a continuous space. In this section we present several forms of the discrete Laplacian operator designed for image processing tasks. The extraction of the zero crossings (as the edge position) from the filtered image will not be considered in this section. The task of the Laplace filter in the final edge detection model will be dealt with in a later phase.

2.1. Classical Laplace filter

The Laplace filter of Eq. (3), frequently used in digital image processing, is a crude approximation of its mathematical equivalent defined in Eq. (2):

$$\begin{aligned}\nabla^2 I(x, y) \approx\ & I(x+1, y) + I(x-1, y) + I(x, y+1) \\ & + I(x, y-1) - 4I(x, y).\end{aligned} \tag{3}$$

Alternatively, this digital filter can be viewed as the following 3×3 set of filter coefficients:

0	1	0
1	−4	1
0	1	0

(4)

To understand the nature and effect of this approximation we look at the behavior of this linear, shift-invariant operator in the Fourier domain. Taking the Fourier transform of the Laplace operator gives

$$\mathbf{F}\left\{\frac{\partial^2 I(x, y)}{\partial x^2} + \frac{\partial^2 I(x, y)}{\partial y^2}\right\} = -\left(u_x^2 + u_y^2\right) I(u_x, u_y), \tag{5}$$

where u_x and u_y are the spatial frequencies in the x and y directions, respectively. Computing the inverse Fourier transform of $H(u_x, u_y) = -\{(u_x)^2 + (u_y)^2\}$ results in an infinite impulse response (IIR) digital filter $h(x, y)$.

$$h(x, y) = \begin{cases} -\dfrac{2\pi^2}{3} & \text{for } x = 0,\ y = 0 \\ \dfrac{2(-1)^{x+1}}{x^2} & \text{for } x \neq 0,\ y = 0 \\ \dfrac{2(-1)^{y+1}}{y^2} & \text{for } x = 0,\ y \neq 0. \end{cases} \tag{6}$$

Equation (3) represents a specific FIR digital filter approximation to Eq. (6). A more general approach minimizes the mean square error between the frequency response of an FIR digital filter, $H_{\text{FIR}}(u_x, u_y)$, and the ideal Laplacian frequency response, $H(u_x, u_y)$, by determining the filter H_{FIR} that minimizes

$$\text{MSE} = \frac{1}{(2\pi)^2} \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} \left| H(u_x, u_y) - H_{\text{FIR}}(u_x, u_y) \right|^2 du_x \, du_y. \tag{7}$$

The Laplace filter in Eq. (3) is produced by applying a square window of 3×3 samples to the IIR filter $h(x, y)$. It can be proven easily that simple truncation gives the best mean square error approximation for a fixed filter size (Oppenheim, Willsky, and Young [19, p. 434]). The remaining filter coefficients are approximated by small integers for computational efficiency with the important constraint that the sum of the filter coefficients must be zero. A better approximation to the IIR filter in Eq. (5) can be obtained by taking a window size larger than three.

The Laplacian is rotation invariant. This has the advantage that high spatial frequencies in all orientations are equally enhanced. At the same time this may be a disadvantage; useful directional information is not available. Another disadvantage of the linear Laplace filter, designed in this way, is its sensitivity to high frequency noise. The Laplacian is generally a useless tool in images with medium and low signal-to-noise ratios [4].

In order to avoid these problems, filters can be designed which do not emphasize the highest spatial frequencies. Examples of filters based on the Laplacian but with a different frequency response are algorithmically designed Laplace filters[1] and the Marr–Hildreth operator.

2.2 The Marr–Hildreth Operator

The Marr–Hildreth operator [15–17] consists of a combination of a band pass filter and a high pass filter. The theory behind this filtering technique is based upon research in the field of neurophysiology. The digital filter is a "best fit" to DOG functions (difference of Gaussians) used in modelling low-level human vision. Marr and Hildreth argue that the 2-dimensional Gaussian is the optimal smoothing filter because it optimally satisfies two conflicting constraints. The first requires a smooth, bandlimited filter in the frequency domain. Thus the width (for Gaussians, standard deviation) in this domain should be small. The second constraint, however, requires a smooth, but localized filter in the spatial domain in order to preserve a good visual interpretation of the scene. This is equivalent to a small standard deviation in the spatial domain. A Gaussian filter minimizes the product of these two widths; it is a filter with minimum spatial-bandwidth product (Papoulis [21, p. 273]).

The first step in the Marr–Hildreth formulation is thus the smoothing:

$$I_{\text{smooth}}(x, y) = G_\sigma(x, y) * I_{\text{input}}(x, y), \tag{8}$$

where $G_\sigma(x, y)$ is the Gaussian filter with standard deviation σ, and $*$ the

[1]In this research we evaluated an "enhanced" Laplacian based upon the Parks–McClellan [20] filter design technique. While the filter itself gave a significantly better Laplacian—second derivative—behavior, the total performance of the edge detector was not improved. The reason for this was the Gaussian filtering (see text) that was used in combination with the Laplacian to achieve the Marr–Hildreth model.

convolution operator. The second step is filtering with the classical Laplace filter to emphasize the high-spatial frequencies.

$$I_{\text{output}}(x, y) = \nabla^2 I_{\text{smooth}}(x, y). \tag{9}$$

Since both filters are linear and shift invariant they can be combined into one filter, the Marr–Hildreth operator, the Laplacian of a Gaussian,

$$\nabla^2 G_\sigma(x, y) = \frac{1}{2\pi\sigma^4}\left(2 - \frac{(x^2 + y^2)}{\sigma^2}\right)\exp\left[-\frac{(x^2 + y^2)}{2\sigma^2}\right]. \tag{10}$$

The operator $\nabla^2 G_\sigma(x, y)$ is often called the "Mexican-hat" operator , due to its characteristic shape. This filter has only one parameter σ, the standard deviation of the Gaussian. From this we can derive the width of the central positive region $w = 2\sqrt{2}\,\sigma$, and the approximate filter size $3w$.

The advantage of the Mexican-hat approach in the sequential processing of the Gaussian and the Laplace filter is the improved accuracy. In Section 2.1 we saw that Eq. (3) is only a rough approximation to the ideal Laplacian. Using Eq. (3) and a separate filter for the Gaussian smoothing permits a propagation of the Laplacian approximation error. By using a single filter based upon Eq. (10), a significantly better approximation to the ideal mathematical Laplace operator in combination with the Gaussian can be achieved. The frequency response of the Laplacian/Gaussian filter, the Marr–Hildreth operator, is given by

$$\begin{aligned}\mathbf{F}\{\nabla^2 G_\sigma(x, y)\} &= \mathbf{F}\{\nabla^2\}\mathbf{F}\{G_\sigma(x, y)\} \\ &= -\left(u_x^2 + u_y^2\right)\exp\left[-\frac{\left(u_x^2 + u_y^2\right)\sigma^2}{2}\right],\end{aligned} \tag{11}$$

where the width of the band and the position of its maximum are determined by the parameter σ. From the above expression we see that for low frequencies the response approximates the Laplacian, especially for small values of σ.

Many articles have pointed out the relationship between the size of the Gaussian and the spatial accuracy of detected edge positions obtained with the Marr–Hildreth operator. (Marr [15, 16], Hildreth [17], Grimson [22], Huertas [23], Eklundh [24], Berzins [25], Zhixian Xu [26].) A large operator—a large Gaussian filter—can be used to find the approximate position of strong edges, while a small operator can be used to obtain high positional accuracy. In general, we may say that the assigned edge position of regions having a size smaller or equal to $2\sqrt{2}\,\sigma$ are displaced. As long as the edge features are smaller than the total size of the operator, the slope of the zero crossings will be fused. Berzins [25] has studied the displacement in relation to spatial features such as the size of the regions, the curvature of the regions, the presence of sharp corners, and the occurrence of nonlinear illumination along the edge. The conclusion of his research is to use the smallest possible smoothing filter to obtain the highest possible spatial accuracy.

Canny [27] has distinguished three performance criteria in judging the ability of an operator to find edges: (1) Good detection. The error types I and II—false

rejection and false detection—should be as low as possible. These probabilities are monotonic decreasing functions of the signal to noise ratio. (2) Good localization. This has been mentioned above when we discussed the accurate spatial position of the detected edge pixels. (3) Only one response to a single edge.

We can easily understand that the first two criteria are conflicting. A large filter, like a large smoothing filter, has a good signal to noise ratio. but a rather poor localization ability. This phenomenon is equivalent to the minimal product of the variances in both the spatial and the spatial frequency domain. Canny has proven that, in the case of step edges disturbed by additive Gaussian white noise with zero mean, the derivative of a Gaussian is a very good approximation to the numerically optimal edge detector for these edges. It is suboptimal according to the above criteria and can be replaced by the zero crossings of the second derivative of a Gaussian operator derived from an input image.

A further disadvantage of a large Marr–Hildreth operator is that the operation is slow on a conventional digital computer. Due to its symmetry, however, the Marr–Hildreth operator can be decomposed into separate row and column filters that permit a significant increase in computational speed. Decomposition is allowed when three conditions are satisfied: (1) the sum of the resulting filter coefficients is zero, (2) the DOG approximation is not disturbed, and (3) the filter remains accurate. The implementation proposed by Huertas fulfills all of the above requirements. First in Eq. (10) the normalization factor is replaced by a proper scale factor K, whose value is chosen according to the desired number of bits of accuracy. Equation (10) can then be rewritten as the sum of two separable filters:

$$\nabla^2 G_\sigma(x, y) = h_{12}(x, y) = h_{21}(x, y), \tag{12}$$

where

$$h_{12}(x, y) = h_1(x)h_2(y) \tag{13}$$

$$h_{21}(x, y) = h_2(x)h_1(y) \tag{14}$$

and with

$$h_1(\xi) = \sqrt{K}\left(1 - \frac{\xi^2}{\sigma^2}\right)\exp\left[-\frac{\xi^2}{2\sigma^2}\right] \tag{15}$$

$$h_2(\xi) = \sqrt{K}\exp\left[-\frac{\xi^2}{2\sigma^2}\right]. \tag{16}$$

With the help of the above expressions it is easy to implement an accurate Marr–Hildreth operator with the standard deviation of the Gaussian as the only filter parameter.

2.3. Nonlinear Laplace Filter

Another approach to removing the disadvantages of the linear Laplace filter is the design of a nonlinear Laplacian, a nonlinear combination of neighborhood pixels. Just as in the linear case, each of these filters has associated with it, one or more

filter parameters. In the case of our nonlinear Laplace filter this might be the size of the filter and the effective shape of its 2-dimensional domain.

In his Ingenieur's (Master's) thesis, Beckers [18] proposed a nonlinear Laplace operator in a 3×3 neighborhood. Given the neighborhood N, the nonlinear Laplace filter–NLLAP–is given by

$$\mathbf{NLLAP}(x, y) = \{\text{maximum}(\mathbf{N}) - b(x, y)\} - \{b(x, y) - \text{minimum}(\mathbf{N})\} \tag{17a}$$

$$= \text{maximum}(\mathbf{N}) - 2 \cdot b(x, y) + \text{minimum}(\mathbf{N}), \tag{17b}$$

where

$$\text{maximum}(\mathbf{N}) = \text{the maximum brightness value in } \mathbf{N},$$

$$\text{minimum}(\mathbf{N}) = \text{the minimum brightness value in } \mathbf{N};$$

and

$$b(x, y) = \text{the brightness value at central position } (x, y).$$

The first term in Eq. (17a) can be described as a gradient in the direction of the maximum, and the second term in Eq. (17a) can be described as a gradient from the direction of the minimum. While the classical Laplacian always computes the second derivative[2] along both axes of the grid, this nonlinear Laplacian adapts its orientation to the local gradient direction. Due to its flexibility, this operator takes the second-order derivative in the most relevant direction—perpendicular to the local edge direction. The maximum as well as the minimum gradient can be found with the help of a window search or a histogram search algorithm (Huang [28], Duin [29]).

Assume we want to develop a circular, more isotropic approximation to this filter. With this idea in mind, we draw a conceptual unit circle centered at the central pixel of the window. In order to approximate the circular shape all the gradients derived from pixels outside the circle have to be normalized by their Euclidean distance to the central pixel.

From the above discussion a formal definition of the nonlinear Laplace operator of arbitrary size and shape is

$$\mathbf{NLLAP}(x, y) = \text{gradmax}(x, y) + \text{gradmin}(x, y), \tag{18}$$

[2]Equation (17), describing the nonlinear Laplacian's raw form, is similar to the filer with second-order derivative behavior presented by Bernsen [50, p. 22, Eq. (2.2)]. However, his resulting edge detection scheme is totally different from the one presented in Section 3 of this article.

where

$$\text{gradmax}(x, y) = \mathbf{max}\{C_n(x', y')[I(x', y') - I(x, y)] \mid (x', y') \in d_n(x, y)\} \tag{19}$$

$$\text{gradmin}(x, y) = \mathbf{min}\{C_n(x', y')[I(x', y') - I(x, y)] \mid (x', y') \in d_n(x, y)\}, \tag{20}$$

where $d_n(x, y)$ is an $n \times n$ square centered at (x, y) and $C_n(x', y')$ is a multiplication mask consisting of scale coefficients dependent on the filter size and the desired filter shape, for example circular or square.

2.3.1. *Square Shaped Filter*

The square shape of the nonlinear Laplace filter can be obtained with the multiplication mask

$$C_n(x', y') = \{1 \mid (x', y') \in d_n(x, y)\}. \tag{21}$$

According to Eq. (21) the multiplication mask in Eqs. (19), (20) can now be skipped. **gradmax** and **gradmin** can be rewritten as

$$\text{gradmax}(x, y) = \mathbf{max}\{I(x', y') \mid (x', y') \in d_n(x, y)\} - I(x, y) \tag{22}$$

$$\text{gradmin}(x, y) = \mathbf{min}\{I(x', y') \mid (x', y') \in d_n(x, y)\} - I(x, y). \tag{23}$$

In the first terms we recognize the formal definition of two simple nonlinear filters the local maximum and the local minimum filter with an $n \times n$ square window as structuring element. Using these definitions Eqs. (22), (23) can be simplified to

$$\text{gradmax}(x, y) = \text{MAX}_n\{I(x, y)\} - I(x, y) \tag{24}$$

$$\text{gradmin}(x, y) = \text{MIN}_n\{I(x, y)\} - I(x, y). \tag{25}$$

It is easy to see that this expression is completely consistent with the less formal description of Eq. (17a). The appearance of MAX_n and MIN_n in the above expression offers the possibility of fast algorithms, with a computation time independent of the filter size n. The square max/min filter is separable, i.e., it can be obtained by a line filter in the x-direction followed by a line filter in the y-direction. These line filters can be implemented in almost size-independent time by updating. The filter window is a line interval. Windows at subsequent positions share most of their pixels. Thus, in going from one position to the next, it suffices to remove one pixel and to add one pixel to update the window. The filter time is thus made independent of the filter size. (See Duin [29], Groen [52], and Verbeek [51].)

2.3.2. *Circular Shaped Filter*

Let us now focus our attention on the development of a nonlinear Laplace operator with a circular shape and arbitrary filter size. This can be done in one of two ways.

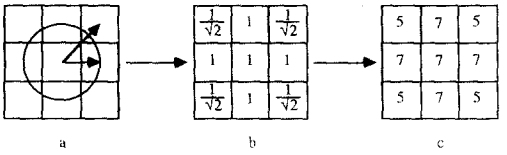

FIG. 1. Design stages of a 3 × 3 nonlinear Laplace operator with quasi-circular shape. (a) A 3 × 3 neighborhood with the unit circle and arrows indicating the distance to the neighbor pixels lying "partially" outside the superimposed circle. (b) Weight factors for gradients originating from these pixels. (c) Integer approximations of the weight factors for faster implementation.

Technique 1. Circular filter shapes can be approximated by choosing the proper multiplication masks $C_n(x', y')$. The general idea has already been given for the smallest filter size. The gradients calculated with corner pixels must be scaled to the unit circle. The resulting floating point coefficients are replaced by integers for computational speed. Fig. 1 shows the three design stages for the coefficients of the 3 × 3 multiplication mask.

Designing larger filters is *not* a straightforward extension of this idea. For an arbitrary filter size n, the conceptual circle becomes a circle with radius $(n - 1)/2$. In this case all gradients calculated with a grey value from outside the new *unit* circle must be scaled to it. Thus normalization is equivalent to a multiplication by

$$\frac{\text{Radius of the imaginary circle}}{\text{Euclidean distance to the center of the window}}.$$

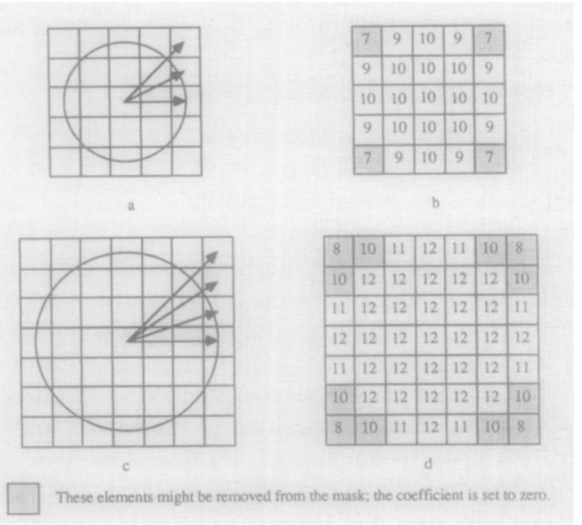

FIG. 2. Multiplication masks $C_5(x, y)$ and $C_7(x, y)$ for nonlinear Laplace operator with quasi-circular shape. (a) A 5 × 5 neighborhood with a circle of radius 2 and arrows indicating the distance to the neighbor pixels lying "partially" outside the superimposed circle. (b) Integer approximations of the weight factors for faster implementation. (c) Idem a for a 7 × 7 neighborhood with a circle of radius 3. (d) Idem b.

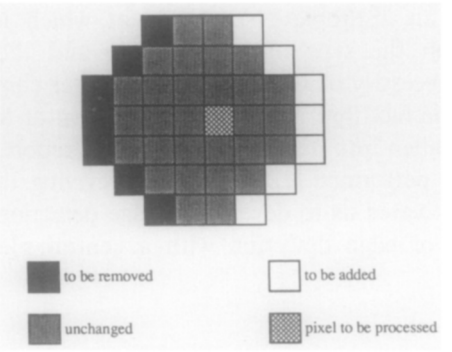

FIG. 3. Two-dimensional crescent updating for a circular maximum or minimum filter with filter size 7.

The gradients originating entirely from within the unit circle should all be viewed as lying exactly on the circle; this corresponds to a coefficient of 1 in the multiplication mask. Possible multiplication masks for the 5×5 and the 7×7 filters are shown in Fig. 2.

In addition, mask coefficients which originate completely from outside this circle may be set to zero. In the masks presented in Figs. 1, 2, and 3 the central pixel is also used in the gradient calculations presented in Eqs. (19), (20). These masks will be used in the following sections when we evaluate the nonlinear Laplace operator.

Technique 2. Circular filter shapes can be achieved by choosing a circular support for the maximum and minimum filters (Eqs. (24), (25)). As the radius of the support increases, this produces a better (smoother) digital approximation to a circle. Although a round full filter requires a proper 2D window, updating—as described in the previous section—can be used again. Now the pixels to be removed constitute a crescent and those to be added another crescent (see Fig. 3). The number of pixels in the crescents is proportional to the filter size n as is the pixel fetching time. This updating technique can be applied to all filter shapes that have a considerable overlap between subsequent positions (Verbeek [51]).

The first technique seems to be more isotropic, especially for small filter sizes. The second technique, however, implies the possibility of even simpler and faster implementation on general purpose computers.

3. EDGE DETECTION MODEL

It is our purpose to evaluate the nonlinear Laplace operator as edge detector. As discussed in Section 2, a Laplace filter without pre- and/or post-processing will not result in a desired edge image. To be able to transform these images into a final edge image, we have to understand how the edge information has been derived. At every place where an intensity change occurs, there will be a corresponding peak in the first derivative perpendicular to the local edge direction. This is, however, equivalent to a zero crossing in the output of the second directional derivative *along* the gradient. Hence the extraction of intensity changes can be performed by finding the zero crossings in the Laplace filtered image.

Marr and Hildreth have improved this simple scheme by combining the Laplace filter with a Gaussian smoothing filter as described in Section 2.3. This filter offers

the ability to choose an appropiate size scale at which intensity changes can occur—dependent upon the type of input image and the noise level of that particular image. The necessity of such a smoothing filter combined with a Laplace filter is confirmed by the fact that before the publication of Marr [15], the classical Laplace operator had fallen into disuse [4]. After their report, a significant number of similar studies were performed [22–27, 30–32] reviving the use of the Laplace operation. This has motivated us to develop an edge detection system based on the Marr–Hildreth theory of edge detection with a central place for the nonlinear Laplace operator.

3.1. Edge Detection System

The complete Marr–Hildreth model for edge detection consists of a smoothing filter, a Laplace filter, and a zero-crossing detector performed in sequence. We have extended this model with an edge strength filter and a threshold operator. An outline of the system is shown in Fig. 4.

We can distinguish five major functions and each of them can be accomplished by different operators. In order to obtain the best overall performance, all of these operators must be optimal. Each operator has associated with it, parameters whose values depend upon the type of input image and the signal to noise ratio. With all the possible choices of parameters, we can, in principle, construct many edge detectors with different properties, characteristics, and performance.

In this section the possible choices for each function—smoothing, Laplace filtering, edge strength detection, zero-crossing detection, and thresholding—will be summarized with a brief motivation.

3.2. Smoothing Filter

In many applications the input image is first smoothed in order to reduce the noise level. We will study the influence of a smoothing filter on edge detection in detail. Two types of smoothing filters will be considered—the uniform filter $u_n(x, y)$

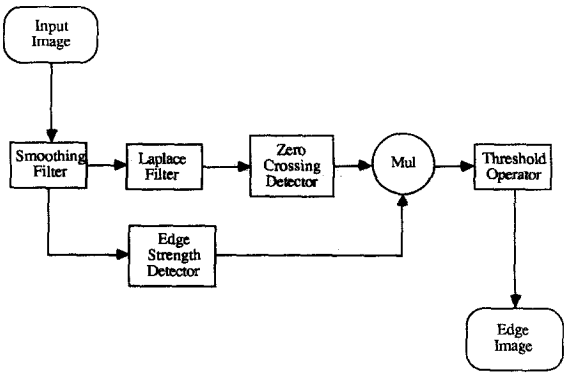

FIG. 4. The edge detection model.

and the Gaussian filter $G_\sigma(x, y)$. The uniform filter with filter size n is defined by

$$u_n(x, y) = \frac{1}{n^2} \sum_{x'=x-S_u}^{x+S_u} \sum_{y'=y-S_u}^{y+S_u} I(x', y'), \tag{26}$$

where S_u is half the filter size of the uniform filter ($(n-1)/2$). The Gaussian filter with standard deviation σ is defined by

$$G_\sigma(x, y) = \frac{1}{2\pi\sigma^2} \sum_{x'=x-S_g}^{x+S_g} \sum_{y'=y-S_g}^{y+S_g} \exp\left[-\frac{x'^2 + y'^2}{2\sigma^2}\right] I(x', y') \tag{27}$$

with a window size of $2S_g + 1$ in each direction. We shall choose S_g equal to the smallest integer larger than 2σ for the following reasons: (1) The digital filter must be a good fit relative to the continuous Gaussian function. Choosing S_g too small produces artifacts attributable to the side lobes that occur in the frequency domain. (2) Large values of S_g slow the computational speed considerably. (3) At least 95% of the area under a 1-dimensional Gaussian lies between plus and minus 1.96σ from the mean. The symmetry of the Gaussian smoothing filter (as well as the uniform filter), means that it can be separated in two independent 1-dimensional filters—one for the rows and one for the columns.

Smoothing filters are able to suppress noise by applying some form of averaging to the input image. We must be careful with using this kind of filter, however, because averaging blurs the image and thus decreases resolution.

Consider that the noise in the image consists of independent samples from a distribution with zero mean and variance σ_n^2. A uniform filter of size n (n^2 samples) will reduce the noise variance to

$$\text{Var[smoothed noise]} = (\sigma_n)^2/n^2. \tag{28}$$

Thus averaging increases the signal to noise ratio. A uniform filter has associated with it the parameter size n while the Gaussian filter standard deviation σ. It can be shown that the noise suppression factors are equal if the standard deviation of the filters are equal. Hence, if we desire to develop a Gaussian filter with the same noise suppression factor as a corresponding uniform filter, then the standard deviation of the Gaussian filter must be chosen equal to the calculated standard deviation of the original uniform filter. For simplicity let us consider the 1-dimensional case; the 2-dimensional case is a straightforward extension. For the 1D uniform filter with total width n we have

$$\text{Var[unifrom}_n\text{]} = (n^2 - 1)/12. \tag{29}$$

The relation between these two filter types is shown in Table 1. In this context it is important to understand that our Gaussian filter (due to the finite window size) has an effective standard deviation which is a little less than that of the underlying continuous function. This also reduces (slightly) the effective noise suppression factor. Even when the effective noise suppression factors of the uniform and Gaussian filters are equal, however, the ultimate result is quite different. This is due to the different frequency responses of these filters.

TABLE 1
Comparison of the Equivalent Noise Suppression of the Uniform and the Gaussian Filters Based upon Their Standard Deviations

Uniform filter with size	Uniform filter with standard deviation	Gaussian filter with standard deviation
3	0.82	0.8
5	1.41	1.4
7	2.00	2.0
9	2.58	2.6

3.3. Laplace Filters

In Section 2 we presented a few Laplace filters in detail. In our evaluation of Laplace filters we have considered the following:

- The classical Laplace filter,
- The Marr–Hildreth operator with three different sizes of the underlying Gaussian: $\sigma = 1.0$, 2.0, and 3.0. (In the case of the Marr–Hildreth operator, the smoothing filter will be removed from the system.)
- The square shaped nonlinear Laplace filter with sizes 3, 5, and 7,
- The circular shaped nonlinear Laplace filter with sizes 3, 5, and 7.

Each of these "second-derivative" operators produces an image in which the zero crossings must be detected.

3.4. Zero-Crossing Detector

It is the task of the zero-crossing detector to assign the changes in sign of a Laplace filtered image as the location of the edge. It is not likely, however, that most of the zero crossings will coincide with the specific spatial coordinates called pixels. Therefore the nearest neighbor in the direction of the gradient should be assigned as the edge pixel to achieve accuracy. Still higher accuracy can be achieved by subpixel interpolation.

Another possibility is to fit the data of a local neighborhood with discrete orthogonal polynomials. The calculated parameter values of this model determine the gradient direction in which the second derivative ought to be computed. With the help of the transformed model the zero crossings establish the proper edge position. This has been done by Haralick [33–34].

These methods have a maximum displacement of 0.5 pixel in the orientation perpendicular to the local edge direction. A disadvantage is that these techniques require rather complex calculations in contrast to the approach presented below.

At all places where regions of opposite sign touch each other, the border pixels of the positive region will be assigned as possible edge pixels. When these regions have been separated by a 1-pixel width strip of zero value pixels, then these pixels mark the exact edge position. In the case of large ramp-like edges, the "Laplacian" transition strip will become wider. Each zero value pixel will be assigned to the nearest region, either positive or negative. The border of the positive region represents the zero crossing. This zero crossing detector has a maximum displace-

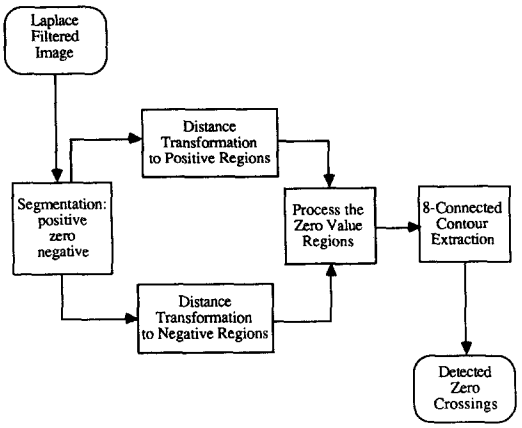

FIG. 5. The zero crossing detection scheme.

ment of one pixel on the downward side of the edge slope. It will, however, offer a better performance when dealing with unsharp images where long ramp edges often occur.

3.4.1. *Implementation*

A flow diagram of the zero-crossing detector is shown in Fig. 5. The input is a Laplace filtered image, which is first segmented into three regions dependent upon the pixel value: positive, zero, or negative.

Next the minimum distance for a zero value pixel to both the positive and negative regions is computed with Borgefors' [35] distance transformation. Her algorithm computes, in two passes through the image, a pseudo-Euclidean distance from the given (zero) pixels to the nearest (positive or negative) regions. The masks for these two passes contain the local chamfer distances. Dorst [36] has proven that the best mean square error approximation to the Euclidean distance for small integers is obtained by using the chamfer distances 5 and 7 in a 3×3 neighborhood. This corresponds to a step with weight 5 along the grid and with weight 7 in diagonal directions. Thus the distance between points P and Q is computed as the minimum distance along a path connecting P and Q. The path consists of n_g "grid" steps and n_d "diagonal" steps.

The zero value pixels are now assigned to the nearest adjacent region. This results in a binary image of positive and negative regions. Finally the 8-connected contour of the positive region is extracted by an exclusive-or with the 4-connect eroded image. This is an important step in that closed contours are guaranteed without additional computational effort.

3.5. Edge Strength Detector

The most plausible detector for computing the edge strength is a gradient operator. A gradient operator is maximal near the position of maximum edge slope. A few common gradient operators are: Roberts [37], Prewitt [38], and Sobel [3]. Performance comparisons by Abdou [39] and Peli [13] have pointed out that these operators perform poorly in images with medium and low signal to noise ratios.

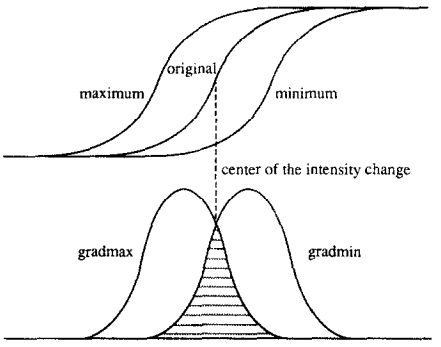

FIG. 6. The blur minimum morphologic edge detector Lee [40] on a blurred edge profile.

Another edge strength detector is the local grey level contrast (maximum–minimum). This filter performs similarly to the above-mentioned gradient operators.

Lee [40] proposed an edge strength detector based on grey scale morphological operators. We have chosen this operator because its inherent noise sensitivity is much less than the more conventional gradient operators. It is composed of the dilation residue and the erosion residue operators and has been called the "blur minimum morphologic edge detector." Lee [40] has defined it as

$$I_{\text{Edge-Strength}} = \mathbf{min}\{I_1 - \text{Erosion}(I_1), \text{Dilation}(I_1) - I_1\}, \tag{30}$$

where $I_1 = \text{Blur}\{I_{\text{input}}\}$ and $\text{Blur}\{I_{\text{input}}\}$ is the input image with a blurring operation. The grey scale erosion and dilation operators can be implemented by local minimum and maximum filters. The blurring or smoothing beforehand is essential as this operator only assigns a pixel as an edge element when it has a grey value between the two extremes—when the pixel lies on the edge slope. Thus—curiously enough—step edges can only be detected after the input image has first been smoothed. The blurred edge profile of Fig. 6 offers an impression of how this edge strength detector works. We see that the minimum of two gradients is maximal at the position with the steepest edge slope. This coincides with a zero crossing in the output of the nonlinear Laplace operator.

When we consider this operator in detail we notice a strong parallel with the nonlinear Laplace operator. According to Eqs. (18)–(20) the definition given by Eq. (30) can be rewritten as

$$I_{\text{Edge-Strength}}(x, y) = \mathbf{min}\{\text{gradmax}(x, y), -\text{gradmin}(x, y)\}. \tag{31}$$

This edge-strength filter includes the possibility of using circularly shaped supports. The above equation also implies a reduction in computational effort. Namely, gradmax and gradmin have to be computed only once and are used twice.

Lee has pointed out that the blurring filter should be of the same size as the grey scale erosion and dilation. Its lower sensitivity to single noise pixels in a homogeneous region is shown in Fig. 7. It shows that when using a uniform smoothing filter for blurring, the output image will not respond to an isolated noise sample in the input image.

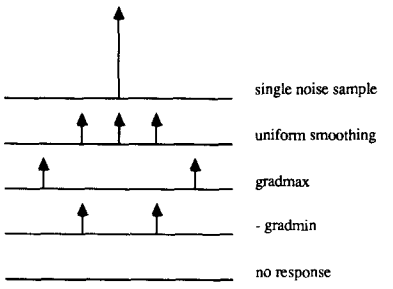

FIG. 7. The response of a single noise sample by Lee's [40] minimum morphologic edge detector.

3.6. Threshold Operator

We did not use an automatic threshold selection algorithm to identify the appropriate threshold level indicated in Fig. 4. It may be possible to separate automatically the noise response and the real edge response for medium and high signal to noise ratios, but an algorithm robust enough for images with low signal to noise ratio is not available. We have, instead, chosen the threshold level by "hand." In order to obtain this we estimate an initial threshold level based on the grey value histogram and this is then adjusted by visual inspection.

4. EVALUATION TECHNIQUE

Quantitative performance measures are often used to compare the results obtained by different edge detectors. In this study these measures are also quite useful for evaluating the various configurations of our edge detection model, developed with choices from the previous section. While the zero-crossing detector and the threshold operator are essentially fixed, the smoothing filter and the Laplace filter must be optimized as to type and size. The size of the edge strength filter should also be related to the smoothing size.

4.1. Parameter Choice as f(SNR)

It is possible to describe a relationship between the SNR (signal to noise ratio) of the input image and the parameter choices of the edge detection model. The SNR is defined as

$$\text{SNR} = g^2/(\sigma_n)^2, \tag{32}$$

where g is the grey level difference between the two sides of the edge, and σ_n the standard deviation of the additive Gaussian noise with zero mean. From Eq. (28) we see that the SNR improves by smoothing the noisy input image. Hence, the noise level of the input image determines roughly the minimum size of the smoothing filter. The lower the SNR, the larger the minimum size of the smoothing filter must be. Within a given range of SNR, however, the final filter configuration ought to be the one with the smallest possible total support in the spatial domain; this permits the detection of relatively small spatial features.

4.2. *Figure of Merit*

Quantitative performance measures can be grouped into two types. The first type —probabilistic measures—is based upon the statistics of false detection and false rejection. The second type—distance measures—is based upon edge deviation or error distance. This is the minimum distance between the detected edge and ground truth. A few measures based upon distance are:

- the mean square deviation:

$$\text{MSD} = \frac{1}{I_D} \sum_{i=1}^{I_D} (d_i)^2 \tag{33}$$

- the mean absolute deviation:

$$\text{MAD} = \frac{1}{I_D} \sum_{i=1}^{I_D} |d_i| \tag{34}$$

- Pratt's figure of merit:

$$\text{FOM} = \frac{1}{\max\{I_D, I_I\}} \sum_{i=1}^{I_D} \frac{1}{1 + \alpha(d_i)^2}, \tag{35}$$

where I_D is the number of detected edge points, I_I the number of ideal edge points (ground truth), α (> 0) a scaling constant, and d_i the edge deviation or error distance for the ith detected edge pixel.

In our study we favor the distance measures, in particular, Pratt's FOM. It fulfills our requirements and it has been used often in edge detector evaluations: Pratt [2], Abdou [39], Peli [13], Gerbrands [12], and Bailey [41]. For a perfect match between the detected edge and the ground truth, FOM = 1; as the detected edge deviates more and more from the truth, the FOM goes to zero. In all cases $0 < \text{FOM} \leq 1$. The advantage of the FOM over the first type of performance measure is the more carefully balanced appraisal of the detected edge. For example, the edge detection algorithm that gives a 1-pixel offset between the detected edge and its ground truth will receive a poor rating based upon the false detection probability of 1.0. Using Pratt's FOM ($\alpha = \frac{1}{9}$) the same algorithm achieves an excellent rating of 0.9. Unfortunately, Pratt's figure-of-merit gives no bonus for finding closed contours and small erroneous edge fragments far from the true contour are heavily penalized. When the ratings obtained with the FOM for different detection techniques or different parameter choices are very close, the measures given in Eqs. (33), (34) can be used to decide which one performs incrementally better.

The implementation of the distance measures is simple. With Borgefors' distance transformation we compute the distance from each detected edge pixel to the nearest actual edge element simultaneously. Again the chamfer distances 5 and 7 are used.

FIG. 8. Edge profiles: (a) an ideal step edge; (b) an ideal ramp edge.

4.3. Test Images

For the comparison of the performance of the different configurations examined, we used two test images: one containing a circular edge and one containing a vertical edge.

A circle is a complex object for those edge detectors that are not able to adapt their orientation to the local edge direction. The local direction of the edge varies rapidly. We used a circle with 180 pixels in diameter. The edge of the circle is an ideal step edge, meaning no transition region between the object and the background (Fig. 8a).

We have chosen this particular test object because it has been used by Peli, who has provided an extensive survey of edge detector performance. The vertical edge is a ramp edge with a transition region of one pixel lying just between the object and background grey level (Fig. 8b). This test image is equivalent to the one used by Pratt, Abdou, and Gerbrands.

Like the above authors we have considered the test images with SNR from 1 to 100. For all test images, g is the grey level change across the edge and σ_n is the standard deviation of the additive Gaussian noise.

4.4. Noise Generator

The Gaussian noise is generated using the polar method for normal deviates proposed by Knuth [42]. This algorithm requires a uniform random number generator for which we have used the one available on our system. According to the *Unix Programmer's Manual* [43] this generator uses a multiplicative random number generator with period 2^{32} to return successive pseudo-random numbers in the range from 0 to $2^{31} - 1$. All of the programming was performed in the C language [44] on a Motorola 680 × 0-based computer system running under Unix.

5. TEST RESULTS

In this section we present the results of our evaluations. In the first part the quantitative results—Pratt's figure of merit—are shown for the different experiments we have performed. In the second part we offer the visual results of our edge detection model.

5.1. Quantitative Evaluation

Figure 9 contains plots of the FOM as a function of SNR for different configurations of our edge detection model. We have used the following abbreviations to

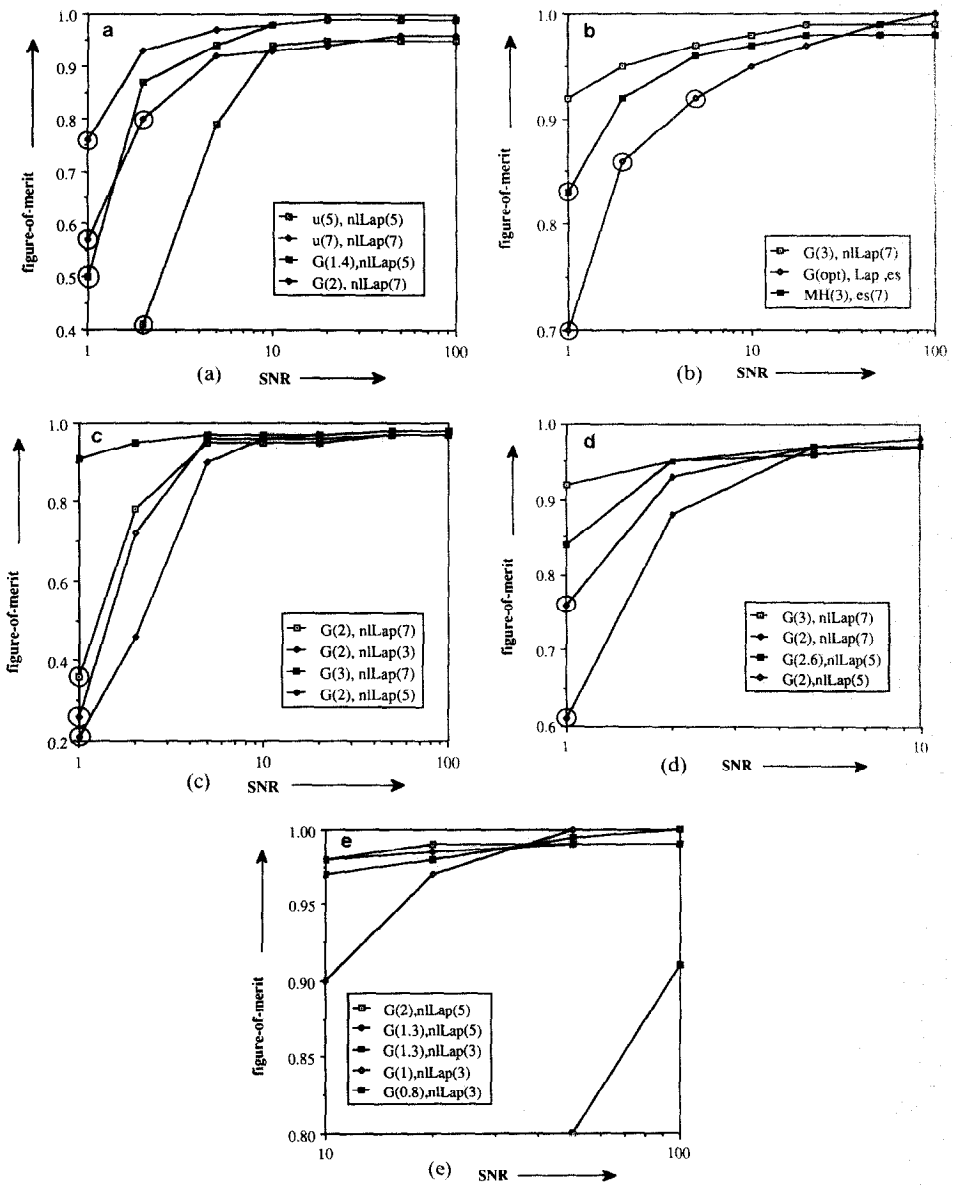

FIG. 9. (a) The figure-of-merit as a function of SNR obtained with different smoothing filters (Gaussian filter, $G(\sigma)$; and uniform filter, $u(n)$) on a circular test image. All nonlinear Laplace filters as well as the edge strength filters have a pseudo-circular support. (b) The figure-of-merit as a function of SNR obtained with different Laplace filters (nonlinear Laplace filter, nlLap(n); classic Laplace filter, Lap; and the Marr–Hildreth operator, MH(σ)) on a circular test image. All nonlinear Laplace filters as well as the edge strength filters have a pseudo-circular support. The edge results achieved with the classical Laplace filter at SNRs below 10 are more disturbed by noise than one would expect from the corresponding figure-of-merits. Large sections of the contour are missing and the rest exhibit a "high-frequency" character around the true contour. (c) The figure-of-merit as a function of SNR obtained with different parameters for Gaussian filter and nonlinear Laplace filter on a vertical ramp edge. All nonlinear Laplace filters as well as the edge strength filters have a pseudo-circular support. (d–e) The figure-of-merit as a function of SNR obtained with several configurations of the edge detection model on a circular test image. All nonlinear Laplace filters as well as edge strength filters have a pseudo-circular support. (d) low SNRs; (e) high SNRs.

indicate the associated parameter values where **c** means circular and **s** means square:

Gaussian filter w/standard deviation σ	$G(\sigma)$
uniform filter of size $\mathbf{n} \times \mathbf{n}$	$u(\mathbf{n})$
classical Laplace	Lap
nonlinear Laplace w/shape (**c** or **s**) and size **n**	nlLap(**c**/**s**, **n**)
Marr–Hildreth operator w/standard deviation σ	MH(σ)
edge strength detector w/shape (**c** or **s**) and size	es(**c**/**s**, **n**)

In the configurations with the nonlinear Laplace filter we always use the edge strength detector with the same parameters (**c**/**s**, **n**). Thus the filter will not be indicated in those cases. A circle around a point on the lines drawn in Fig. 9 marks a point where a broken edge (contour) was detected instead of a closed one. The values of SNR evaluated are 1, 2, 5, 10, 20, 50, and 100.

5.1.1. *Smoothing Filter*

The first function to be investigated is the smoothing filter. In Fig. 9a we show the FOMs obtained with two different types of smoothing for which the noise reduction factor is almost the same. (Remember that from Table 1 in Section 3 with respect to noise reduction: $G(1.4) \approx u(5)$ and $G(2.0) \approx u(7)$.) Although this test was executed for three equivalent noise reduction sizes ($u(5)$, $u(7)$, $u(9)$), we have plotted only two of them in order to keep Fig. 9a readable. The result of the third, however, is similar to the ones plotted in Fig. 9a. The other functions of the edge detection model (Fig. 4) were all equal for the corresponding configurations with different smoothing filters.

As can be seen in Fig. 9a, the Gaussian filter was found to have the best performance over the entire range of SNR. Even with the somewhat smaller effective noise reduction factor of our FIR Gaussian filter, the results are superior to those obtained with uniform filtering. Based upon our results from Fig. 9a as well as the Marr–Hildreth results, we will focus exclusively in the following experiments on the use of Gaussian smoothing filters.

5.1.2. *Laplace Filters*

In this section we evaluate the various possible choices for Laplacian-like filtering with the Gaussian filter as smoothing filter. In Fig. 9b we show the FOM for three different Laplace-like filters: the classical Laplace (Lap), the Marr–Hildreth (MH), and the nonlinear Laplace (nlLap). Before judging the performance one should know that: (1) For the classical Laplace filter we have chosen the optimal smoothing size for *each* SNR; for the others the smoothing size was constant. By "optimal Gaussian" filter we mean the value for the standard deviation that leads to the best ultimate edge result according to the FOM. (2) The edge strength (from Eq. (31)) was also computed with this optimal smoothing filter.

From Fig. 9b, we conclude that the nonlinear Laplace operator has the best performance for medium and low SNR (that is, SNR $\leq$ 50). At high SNR ($= 100$) the classical Laplacian performs better due to its optimally chosen Gaussian filter. At very low SNR the nonlinear Laplacian is far superior; it is the only Laplace-like filter able to detect closed contours at SNR = 1. Experiments with other values for

the standard deviation (Marr–Hildreth and Gaussian filter) and the size of the nonlinear Laplace filter offer similar results in comparison to the other two types of Laplace filters.

5.1.3. *Nonlinear Laplace Filter*

For the test image containing a vertical ramp edge (Fig. 8b) as used by Pratt, Abdou, and Gerbrands, we have studied four configurations with a Gaussian filter, a nonlinear Laplace filter, and an edge strength filter with the same parameters as the nonlinear Laplacian. The FOM plots are shown in Fig. 9c. For the vertical ramp edge we see that the larger the filters (G, nlLap, and es) the better the achieved FOM for low SNR. For SNR > 5 all filters achieve similar performance.

A characteristic of the nonlinear Laplace filter is its shape in the spatial domain—circular or square. It has not been possible for us to distinguish their performance with the two test images considered. For these images the performance of the circularly shaped filter is just slightly better than that of the square-shaped one.

5.1.4. *Locally Optimum Configurations*

To identify the optimal parameter choice for each *interval* of SNR, we divide the total range in two parts: $1 \leq \text{SNR} \leq 10$, and $10 < \text{SNR} \leq 100$. We have studied many configurations arranged with a Gaussian smoothing filter, a nonlinear Laplace filter, and an edge strength filter of different sizes. Figure 9d shows the FOM results for low SNRs and Fig. 9e for high SNR.

From Fig. 9e we may conclude there is a lower bound for the smoothing width somewhere between $\sigma = 0.8$ and $\sigma = 1.0$—just above the standard deviation of a uniform smoothing filter with size 3 (see Table 1). The lower bound for the nonlinear Laplace filter is size 3. As the SNR decreases we see a monotonic increase in the width of the FIR Gaussian filter required to achieve a respectable FOM. This result confirms the proposal for relating the sizes of the smoothing and nonlinear

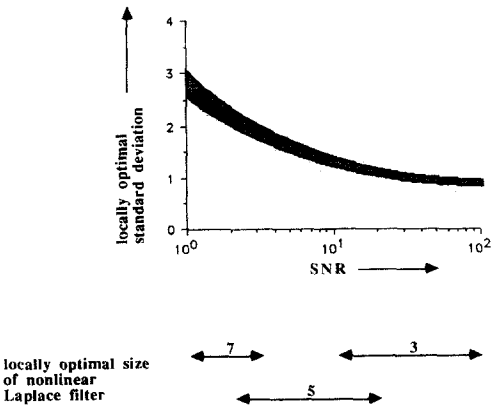

FIG. 10. Locally optimal standard deviation of the Gaussian smoothing filter and locally optimal size of the nonlinear Laplace filter as function of the signal-to-noise ratio.

Laplace filters to the signal to noise ratio. A rough estimation of the locally optimal standard deviation is shown in Fig. 10. (By locally optimal we mean the value of the filter parameter required to achieve optimal performance over a local range of SNR.) An estimate of the uncertainty associated with these parameter choices is indicated by the grey region in Fig. 10. At the bottom of the figure a rough indication for the proper size of the nonlinear Laplace filter has been presented. These results provide another indication of the appropriate sizes for the combination of Gaussian smoothing, nonlinear Laplace filtering and edge strength measurement.

The locally optimal smoothing size also depends on the edge strength detector (Lee [40]) used. This detector always requires a certain minimum amount of blurring in order to be able to detect sharp edges such as step edges.

5.2. Qualitative Evaluation

In addition to the objective analysis presented above, it is worthwhile to offer a qualitative examination by presenting standard pictures processed with these procedures. The reasons for this are twofold: First, an impression of the performance is given by a sequence of pictures made during the various stages of the edge detection process. This has been done for SNR = 1, and the results are shown in Fig. 11. Second, despite the subjectivity associated with simply viewing pictures, the topic remains *image processing* and we would be less than responsible if we did not permit the reader to draw his or her own conclusions as to the efficacy of our method.

6. DISCUSSION

In the previous section we have evaluated a number of configurations of our edge detection model. The evaluation has pointed out a (locally) optimal configuration for our edge detector as a function of the SNR in the input image. In order to get a perspective on its performance, it is useful to compare our results with those reported in the literature.

Pratt, Abdou, and Gerbrands compared the FOM on a vertical ramp edge in the presence of noise. In their comparisons the threshold level was set to achieve the maximum FOM value. We also could have chosen to set our thresholds by the "objective" method of maximizing the FOM. We chose instead thresholds that provided closed contours and we took our chances with the resulting FOM. We did so, because the edge results with maximum figure-of-merit consist of many small disconnected line elements from which hardly any information can be derived. They examined the Kirsch template matching operator [45] of different sizes and levels and the Prewitt, Sobel, and Roberts' gradients combined with both the square root and the magnitude operation. Gerbrands evaluated a sequential edge detector which searched for a minimum cost path through a desired region of interest with the help of dynamic programming. His method showed significantly better performance than the best of the above-mentioned parallel schemes for low values of the SNR. His results are shown in [12, Fig. 4]. A disadvantage of his technique is that *a priori* knowledge is required about the shape and position of the edge to be detected. On

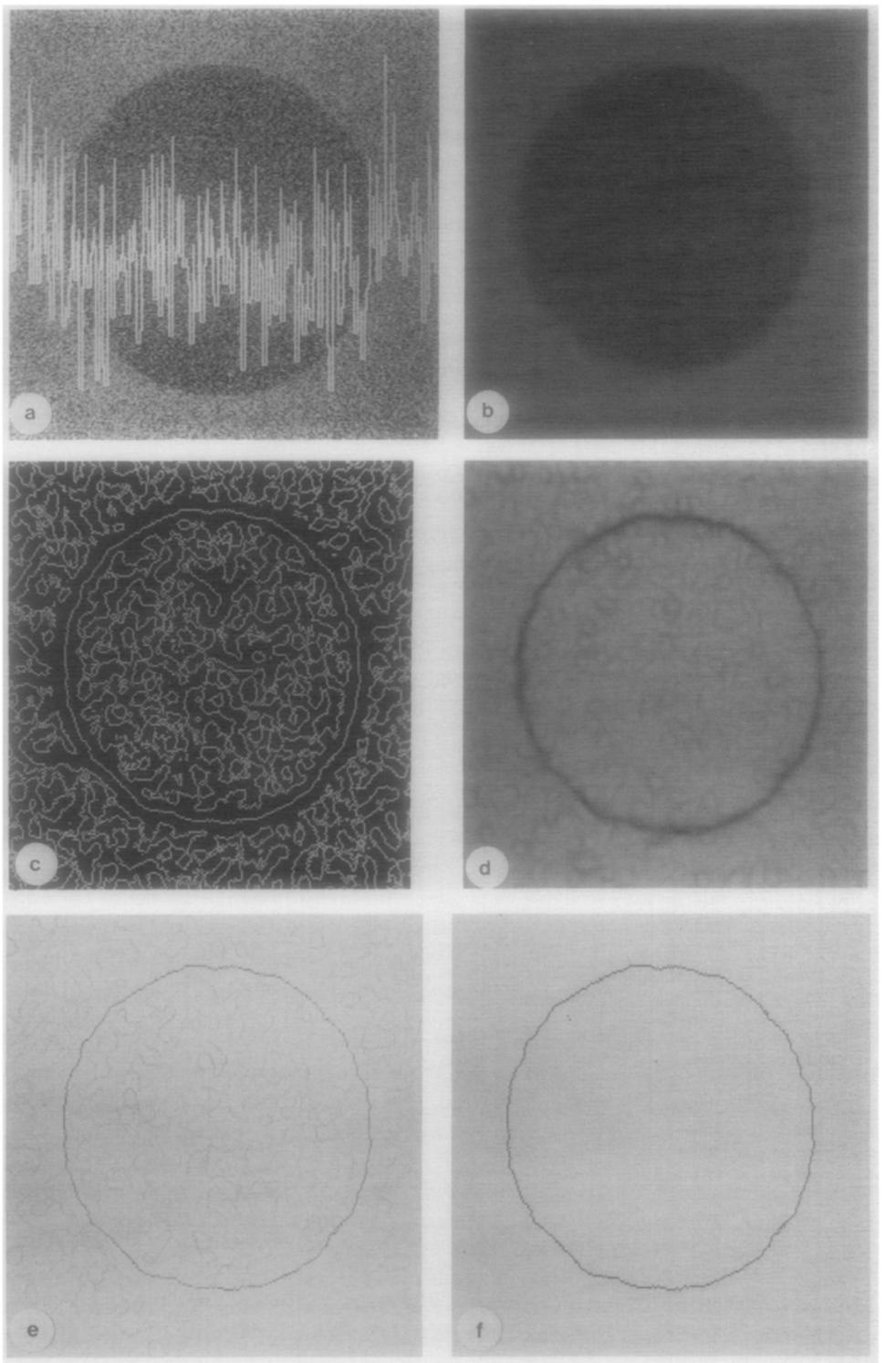

FIG. 11. Pictures showing the intermediate results during the various states of the edge detection process for a circle of SNR = 1: (a) circular test image of SNR = 1 (0 dB) with a superimposed line plot; (b) Gaussian filtered image, G(3.0); (c) zero crossings after nonlinear Laplace filtering, nlLap(c, 7); (d) edge strength image, es(c, 7); (e) zero crossings with grey values representing the strength of the underlying edge; (f) thresholded edge result, a simply connected 1-pixel thick contour.

the same ramp edge our nonlinear Laplace filter exhibits better performance than the minimum cost search algorithm even at low values of SNR (Fig. 9c).

Peli carried out an extensive evaluation of edge detection performance on a circular test image. His results are shown in [13, Fig. 11b]. He pointed out that Roberts' gradient is a useful edge detector for an image with a very small amount of noise because of its low number of computations. When the image is noisy, however, the Rosenfeld algorithm [46–48] performs better. The general idea behind his technique is that of computing differences between averages of nonoverlapping neighborhoods that meet at the same point. Our edge detection model with a suitable combination of Gaussian smoothing and nonlinear Laplace filtering is superior to the above results for the entire range of SNR.

Haralick [33, 49] and Grimson compared edge detectors based upon the detection of zero crossings in the output of the second-order derivative. They used a synthetic chessboard image with base grey levels 75 to 175 to which was added independent Gaussian noise of zero mean and standard deviation 50. This test image had a SNR = 4 according to our definition of the SNR in Eq. (34). The measures they used were the conditional probabilities of true edges given assigned edges and vice versa and an edge distance criteria which was the average distance from the detected edge pixels to the closest true edge pixels which were assigned nonedge but which were true. The edge detectors analyzed were: the cubic polynomial edge detector which computes the second-order derivative by a model approximation of the underlying grey level surface (Haralick [33]), the Marr–Hildreth operator (Grimson and Hildreth [22]), and a truncated version of the Marr–Hildreth operator (Haralick [33]). Following Fig. 10 we have constructed our edge detector with a Gaussian smoothing of standard deviation 1.8 in combination with a circular shaped nonlinear Laplace filter of size 5. The results of this comparison are shown in Table 2.

It appears that the conditional probabilities for the nonlinear Laplace filter are slightly better than the Haralick cubic polynomial fitting model, but slightly worse than the Marr–Hildreth operator. This may be explained by the fact that in their implementation the edge pixels are assigned to the nearest neighbor in the output of the Marr–Hildreth operator, instead of the nearest positive neighbor in the nonlinear Laplace filtered image. This latter method offers a faster implementation in exchange for a slightly larger displacement of the edge. Truncation of the Marr–Hildreth operator leads to disastrous results. For the measured error distance our detector has the smallest value. Figure 12 shows the noisy chess-board and the edge result achieved.

TABLE 2

Conditional Probabilities and the Mean Absolute Edge Deviation as Performance Measures for a Chessboard Image with SNR = 4. IE Stands for Ideal Edge and AE Stands for Assigned Edge

Performance measures	Cubic polynomial model (Haralick)	Truncated Marr–Hildreth	Marr–Hildreth	Nonlinear Laplace filter
Prob(AE|IE)	0.7207	0.3977	0.8887	0.82
Prob(IE|AE)	0.7197	0.4159	0.9237	0.81
Error distance	1.16	1.76	1.17	1.07

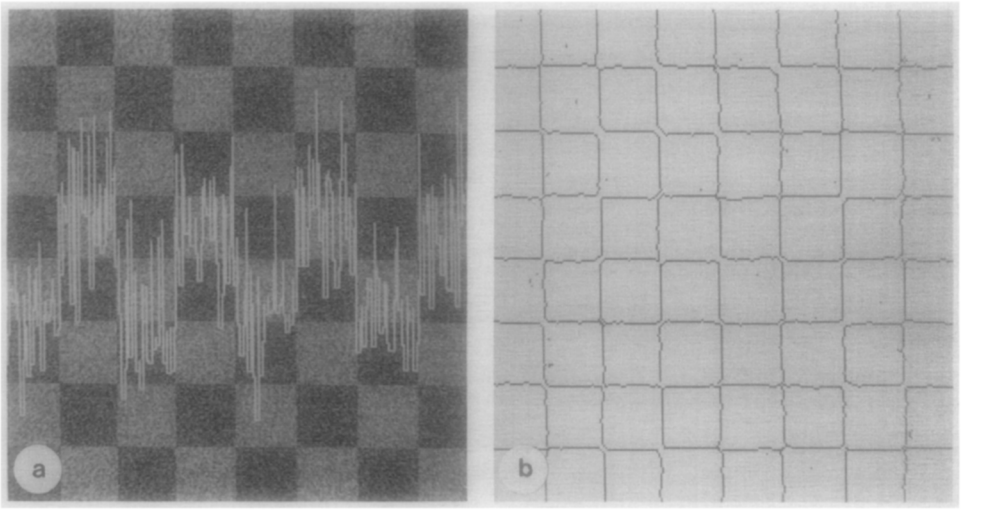

FIG. 12. (a) A synthetic chessboard image of SNR = 4 with a superimposed line plot. (b) Detected edges achieved by an edge detector consisting of: A Gaussian filter with standard deviation 1.8, G(1.8); a nonlinear Laplace filter of quasi-circular shape and filter size 5, nlLap(c, 5); and a Lee's edge strength filter of the same shape and filter size, es(c, 5).

7. CONCLUSIONS

In this article we have evaluated extensively the performance of an edge detection scheme developed around the nonlinear Laplace operator. The complete edge detection model is based upon the Marr–Hildreth theory of edge detection and is extremely effective and flexible in detecting 1-pixel thick edges. We have argued that the parameters of the Gaussian smoothing filter as well as the nonlinear Laplacian depend upon the SNR in the input image. Using an optimal configuration for minimum spatial extension leads to a result where even in noisy images most of the edge information is retained. We have compared our results with those in the literature. For the test images we considered, our configuration performs at least as well—and in most cases far better—than other edge detectors. For these comparisons we have used Pratt's figure of merit as a quantitative performance measure. At low SNRs (< 10) our detector performed far better than the others and produced closed contours for the test images used.

Specific characterizations of the nonlinear Laplacian are its adaptive orientation to the direction of the gradient, its inherent masks which permit the development of approximately circular (isotropic) filters, and its easy and fast implementation in software. For a finishing touch we present the edge results achieved by applying our detector to three "real world" images (Fig. 13).

ACKNOWLEDGMENTS

The authors thank Dr. Piet Verbeek for his valuable comments during various stages of this project. This article describes the research done at the Pattern Recognition Group of the Faculty of Applied Physics at Delft University of Technology, The Netherlands, under partial sponsorship of the Netherlands Foundation for Fundamental Research on Materials (FOM) Grant DTN95.0329.

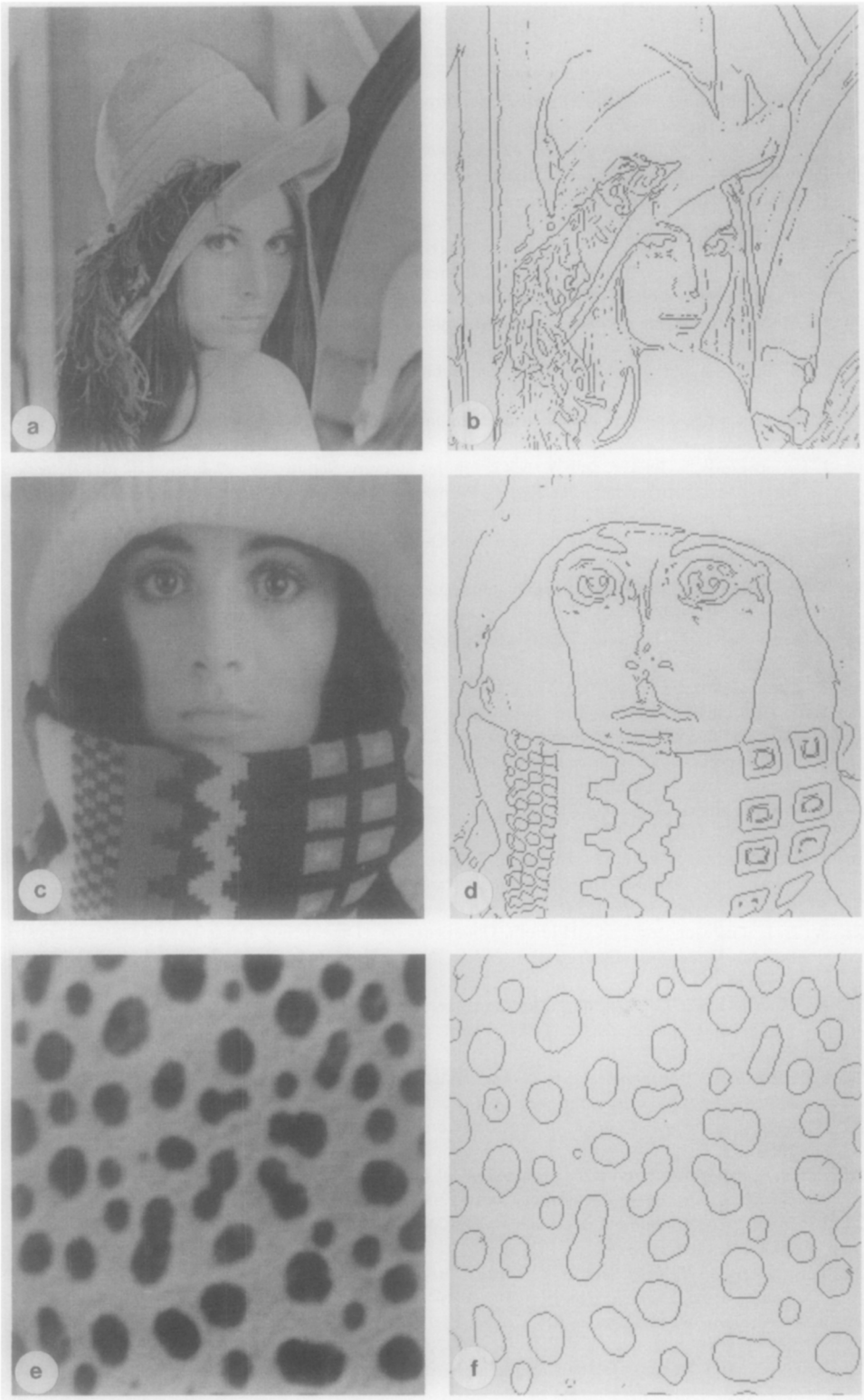

FIG. 13. Presentation of the edge results achieved by applying the proposed edge detection model to some "real world" images. (a) Picture of a woman wearing a hat ("lena"). (b) Detected edges achieved by: G(1.5), nlLap(c, 5), es(c, 5). (c) Picture of a woman wearing a highly structured scarf ("trui"). (d) Detected edges achieved by: G(1.4), nlLap(c, 5), es(c, 5). (e) Electron microscope image containing gold particles embedded in glass ("cermet"). (f) Detected edges achieved by: G(1.0), nlLap(c, 3), es(c, 3).

REFERENCES

1. A. Rosenfeld and A. C. Kak, *Digital Picture Processing*, Academic Press, New York, 1976.
2. W. K. Pratt, *Digital Image Processing*, Wiley, New York, 1977.
3. R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*, Wiley, New York, 1973.
4. D. A. Ballard and C. M. Brown, *Computer Vision*, Prentice-Hall, Englewood Cliffs, NJ, 1982.
5. F. Attneave, Some information aspects of visual perception, *Psychol. Rev.* **61**, 1954, 183–193.
6. R. C. Gonzalez and P. Wintz, *Digital Image Processing*, Addison–Wesley, Reading, MA, 1977.
7. T. G. Stockham, Image processing in the context of a visual model, *Proc. IEEE* **60**, 1972, 828–842.
8. J. Lester, H. Williams, B. Weintraub, and J. Brenner, Two graph searching techniques for boundary finding in white blood cell images, *Comput. Bio. Med.* **8**, 1978, 293–308.
9. U. Montanari, On the optimal detection of curves in noisy pictures, in *Commun. ACM* **14**, 1971, 335–345.
10. L. S. Davis, A survey of edge detection techniques, *Comput. Graphics Image Process.* **4**, 1975, 248–270.
11. A. Martelli, Edge detection using heuristic search methods, *Comput. Graphics Image Process.* **1**, 1971, 335–345.
12. J. J. Gerbrands, E. Backer, and W. A. G. van der Hoeven, Quantitative evaluation of edge detection by dynamic programming, in *Pattern Recognition in Practice II* (Gelsema and Kanal, Eds.) pp. 91–99, Elsevier, North-Holland, New York, 1986.
13. T. Peli and D. Malah, A study on edge detection algorithms, *Comput. Graphics Image Process.* **20**, 1982, 1–21.
14. V. Goetcherian, From binary to grey tone image processing using fuzzy logic concepts, *Pattern Recognit.* **12**, 1980, 7–15.
15. D. Marr and E. C. Hildreth, Theory of edge detection, *Proc. Roy. Soc. London Ser. B* **207**, 1980, 187–217.
16. D. Marr, *Vision*, Freeman, New York, 1982.
17. E. C. Hildreth, The detection of the intensity changes by computer and biological vision systems, *Comput. Vision Graphics Image Processing* **22**, 1983, 1–27.
18. A. L. D. Beckers, *Metingen van Parameters voor Niet-lineaire Objectgrootte-filters in beelden*, Ingenieur's thesis, Department of Applied Physics, Delft University of Technology, 1986 [Dutch].
19. A. V. Oppenheim, A. S. Willsky, and I. T. Young, *Signals and Systems*, Prentice–Hall, Englewood Cliffs, NJ, 1983.
20. J. H. McClellan, T. W. Parks, and L. R. Rabiner, A computer program for designing optimum FIR linear phase digital filters, *IEEE Trans. Audio Electroacoust.* **AU-21**, No. 6, 1973, 506–525.
21. A. Papoulis, *Signal Analysis*, McGraw–Hill, New York, 1977.
22. W. E. L. Grimson and E. C. Hildreth, Comments on "Digital Step Edges from Zero Crossings of Second Directional Derivatives," *IEEE Trans. Pattern Anal. Mach. Intell.* **PAMI-7**, 1985, 121–127.
23. A. Huertas and G. Medioni, Detection of intensity changes with subpixel accuracy using Laplacian–Gaussian masks, *IEEE Trans. Pattern Anal. Mach. Intell.* **PAMI-8**, No. 5, 1986, 651–664.
24. J. O. Eklundh, T. Elfving, and S. Nyberg, Edge detection using the Marr–Hildreth operator with different sizes, in *Proceedings, 6th Inter. Conf. Pattern Recognition, Munich, 1982*, pp. 1109–1111.
25. V. Berzins, Accuracy of Laplacian edge detectors, *Comput. Vision Graphics Image Process.* **27**, 1984, 195–210.
26. Z. Xu, A further study on error probabilities of Laplacian–Gaussian edge detection, in *Proceedings, 8th Int. Conf. Pattern Recognition, Paris, France*, 1986, pp. 601–603.
27. J. Canny, A computational approach to edge detection, *IEEE Trans. Pattern Anal. Mach. Intell.* **PAMI-8**, 1986, 679–698.
28. T. S. Huang, G. J. Yang, and G. Y. Tang, A fast two-dimensional median filtering algorithm, *IEEE Trans. Acoust. Speech Signal Process.* **ASSP-27**, 1979, 13–18.
29. R. P. W. Duin, W. Haringa, and R. Zeelen, Fast percentile filtering, *Pattern Recognit. Lett.* **4**, 1986, 269–272.
30. V. Torre and T. Poggio, On edge detection, *IEEE Trans. Pattern Anal. Mach. Intell.* **PAMI-8**, 1986, 147–163.
31. W. H. H. J. Lunscher and M. P. Beddoes, Optimal edge detector design I: Parameter selection and noise effects, *IEEE Trans. Pattern Anal. Mach. Intell.* **PAMI-8**, 1986, 164–177.

32. J. Besuijen, Edge detection by combining directional derivatives, in *EUSIPCO-86* (I. T. Young *et al.*, Eds.), pp. 887–890.
33. R. M. Haralick, Digital step edges from zero crossings of second directional derivatives, *IEEE Trans. Pattern Anal. Mach. Intell.* **PAMI-6**, No. 1, 1984, 56–68.
34. R. M. Haralick, Cubic facet model edge detector and ridge–valley detector: Implementation details, in *Pattern Recognition in Practice II* (Gelsema and Kanal, Eds.), pp. 81–90, Elsevier, North-Holland, New York, 1986.
35. G. Borgefors, Distance transformations in arbitrary dimensions, *Comput. Vision Graphics Image Process.* **27**, 1984, 321–345.
36. L. Dorst, *Discrete Straight line segments: Parameters, Primitives and Properties*, Ph.D. thesis, Department of Applied Physics, Delft University of Technology, 1986.
37. L. G. Roberts, Machine perception of three-dimensional solids, in *Optical and Electrooptical Information Processing* (J. T. Tippet *et al.*, Eds.), pp. 159–197, MIT Press, Cambridge, MA, 1965.
38. J. M. S. Prewitt, Object enhancement and extraction, in *Picture Processing and Psychopictorics* (B. S. Lipkin and A. Rosenfeld, Eds.), Academic Press, New York, 1970.
39. I. E. Abdou and W. K. Pratt, Quantitative design and evaluation of enhancement/thresholding edge detectors, *Proc. IEEE* **67**, No. 5, 1979, 753–763.
40. J. S. L. Lee, R. M. Haralick, and L. S. Shapiro, Morphologic edge detection, in *Proceedings, 8th Int. Conf. Pattern Recognition, Paris*, 1986, pp. 369–373.
41. D. G. Bailey and R. M. Hodgson, Range filters: Intensity subrange filters and their properties, *Image Vision Comput.* **3**, 1985, 99–110.
42. D. E. Knuth, *The Art of Computer Programming Volume 2: Seminumerical Algorithm*, Addison–Wesley, Reading, MA, 1981.
43. *UNIX Programmer's Manual*, 4.1 Berkeley Software Distribution, 1983.
44. B. W. Kernighan and D. M. Ritchie, *The C Programming Language*, Prentice-Hall, Englewood Cliffs, NJ, 1978.
45. R. Kirsch, Computer determination of the constituent structure of biological images, *Comput. Biomed. Res.* **4**, 1971, 315–328.
46. A. Rosenfeld, A nonlinear edge detection technique, *Proc. IEEE* **58**, 1970, 814–816.
47. A. Rosenfeld and M. Thurston, Edge and curve detection for visual scene analysis, *IEEE Trans. Comput.* **C-20**, 1971, 562–569.
48. A. Rosenfeld, M. Thurston, and Y. H. Lee, Edge and curve detection: Further experiments, *IEEE Trans. Comput.* **C-21**, 1972, 677–715.
49. R. M. Haralick, Author's reply, *IEEE Trans. Pattern Anal. Mach. Intell.* **PAMI-7**, 1985, 127–129.
50. J. Bernsen, Dynamic thresholding of grey-level images, in *Proceedings, 8th Int. Conf. Pattern Recognition, Paris, France*, 1986, pp. 1251–1255.
51. P. W. Verbeek, H. A. Vrooman, and L. J. van Vliet, Low level image processing max–min filters, *Signal Proc.* **15**, 1988, 249–258.
52. F. C. A. Groen, P. P. Jonker, and R. P. W. Duin, Hardware versus software implementations of fast image processing algorithms, in *Real-time Object Measurement and Classification* (A. K. Jain, Ed.), NATO ASI Series Vol. F42, Springer-Verlag, BerlinHeidelberg, 1988.