Branch Edge Detection Using Three-Arrow-Shaped Laplacian Gaussian Filter

Mohammad Reza MAHZOUN*, Kozo OKAZAKI*, Shin-ichi TAMURA**, and Okihiko ISHIZUKA***

(Received Feb. 28, 1994)

Making a good mask to extract the edges of objects in an image correctly, is a problem that many researches have done around it. To detect the shape of Corneal Endithelial Cells (CEC), which is important for clinical diagnosis, we use the combination of conventional LGF and newly developed TAS-LGF (Three-Arrow-Shaped Laplacian-Gaussian Filter) and we show that its rersult will be better than that of using a two dimensional LGF.

I. Introduction

A problem of fundamental importance in image analysis is edge detection. Edges characterize object boundaries and are therefore useful for segmentation , registration, and identification of objects in scenes. Edge points can be thought of as pixel locations of abrupt gray-level change. For example , it is reasonable to define edge points in binary images as black pixels with at least one white nearest neighbor, that is , pixel locations (m,n) such that u(m,n)=0 and g(m,n)=1, where

$$g(m,n)= [u(m,n) \oplus u(m\pm 1,n] \ .OR.\ [u(m,n) \oplus u(m,n\pm 1)] \qquad (1)$$

where u(m,n) indicates a digital image and + denotes the logical exclusive_OR operation.

* Dept. of Electrical and Electronics Eng. Fukui Univesity
** Medical School, Osaka University
*** Dept. of Electronics Eng. Miyazaki University

In summary a good edge detector should have three performance criteria as follows:

1) Good detection. There should be a low probability of failing to mark real edge points, and low probability of falsely marking nonedge points. Since both these probabilities are monotonically decreasing functions of the output signal-to-noise ratio, this criterion corresponds to maximizing signal-to-noise ratio.

2) Good localization. The points marked as edge points by the operator should be as close as possible to the center of the true edge.

3) Only one response to a single edge. This is implicitly captured in the first criterion since when there are two responses to the same edge, one of them must be considered false. However, the mathematical form of the first criterion did not capture the multiple response requirement and it had to be made explicit.

II. Edge Detection Methods

For a continuous image f(x,y) its derivative assumes a local maximum in the direction of the edge. Therefore, one edge detection technique is to measure the gradient of f along r in a direction θ (Fig. 1), that is

$$\frac{\partial f}{\partial r} = \frac{\partial f}{\partial x}\frac{\partial x}{\partial r} + \frac{\partial f}{\partial y}\frac{\partial y}{\partial r} = f_x \cos\theta + f_y \sin\theta \qquad (2)$$

The maximum value of $\partial f/\partial r$ is obtained when $(\partial/\partial\theta)(\partial f/\partial r) = 0$. This gives

$$-f_x \sin\theta_g + f_y \cos\theta_g = 0 \quad \Rightarrow \quad \theta_g = \tan^{-1}\left(\frac{f_y}{f_x}\right) \qquad (3.a)$$

$$\left(\frac{\partial f}{\partial r}\right)_{\max} = \sqrt{f_x^2 + f_y^2} \qquad (3.b)$$

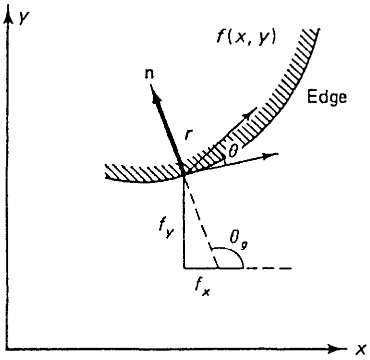


Figure 1. Gradient of $f(x, y)$ along r direction.

where θ_g is the direction of the edge. Based on these concepts, two types of edge detection operators have been introduced, gradient operators and compass operators. For digital images these operators, also called masks, represent finite-difference approximations of either the orthogonal gradients f_x, f_y or the directional gradient $\partial f/\partial r$. Let **H** denote a pxp mask and define, for an arbitrary image **U**, their inner product at location (m,n) as the correlation (★)

$$\langle \mathbf{U},\mathbf{H}\rangle_{m,n} = \sum_{i}\sum_{j} h(i,j)u(i+m,j+n) = u(m,n)\bigstar h(-m,-n) \qquad (4)$$

a) Gradient Operators

These are represented by a pair of masks H_1, H_2, which measure the gradient of the image u(m,n) in two orthogonal directions (Fig. 2). Defining the bidirectional gradients $g_1(m,n)=<U,H_1>_{m,n}$, $g_2(m,n)=<U,H_2>_{m,n}$ the gradient vector magnitude and direction are given by;

$$g(m,n) = \sqrt{g_1^2(m,n) + g_2^2(m,n)} \tag{5}$$

$$\theta_g(m,n) = \tan^{-1}\frac{g_2(m,n)}{g_1(m,n)} \tag{6}$$

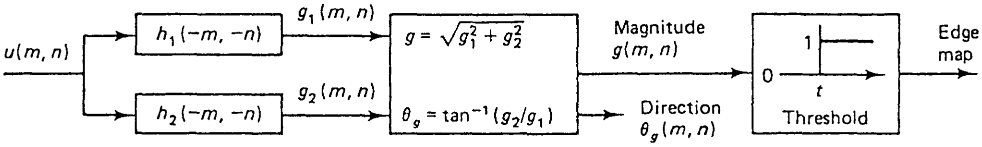


Figure 2. Edge detection via gradient operators.

Often the magnitude gradient is calculated as

$$g(m,n) = |g_1(m,n)| + |g_2(m,n)| \tag{7}$$

rather than as in 5. This calculation is easier to perform and is preferred especially when implemented in digital hardware.

Table 1 lists some of the common gradient operators. The Prewitt, Sobel, and isotropic operators compute horizontal and vertical differences of local sums. This reduces the effect of noise in the data. Note these operators have the desirable property of yielding zeros for uniform regions.

TABLE 1 Some Common Gradient Operators.
Boxed element indicates the location of the origin

	H_1	H_2
Roberts	$\begin{bmatrix} \boxed{0} & 1 \\ -1 & 0 \end{bmatrix}$	$\begin{bmatrix} \boxed{1} & 0 \\ 0 & -1 \end{bmatrix}$
Smoothed (Prewitt)	$\begin{bmatrix} -1 & 0 & 1 \\ -1 & \boxed{0} & 1 \\ -1 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} -1 & -1 & -1 \\ 0 & \boxed{0} & 0 \\ 1 & 1 & 1 \end{bmatrix}$
Sobel	$\begin{bmatrix} -1 & 0 & 1 \\ -2 & \boxed{0} & 2 \\ -1 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} -1 & -2 & -1 \\ 0 & \boxed{0} & 0 \\ 1 & 2 & 1 \end{bmatrix}$
Isotropic	$\begin{bmatrix} -1 & 0 & 1 \\ -\sqrt{2} & \boxed{0} & \sqrt{2} \\ -1 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} -1 & -\sqrt{2} & -1 \\ 0 & \boxed{0} & 0 \\ 1 & \sqrt{2} & 1 \end{bmatrix}$

The pixel location (m,n) is declared an edge location if g(m,n) exceeds some threshold t. The locations of edge points constitute an edge map ε (m,n), which is defined as

$$\varepsilon(m,n) = \begin{cases} 1, & (m,n) \in I_g \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

where

$$I_g \triangleq \{(m,n); g(m,n) > t\} \tag{9}$$

The edge map gives the necessary data for tracing the object boundaries in an image. Typically, **t** may be selected using the cumulative histogram of g(m,n) so that 5 to 10% of pixels with largest gradients are declared as edges. (Although the gradient operators act as good detectors in many cases but Photos 18,19 of the Appendix shows their poor results to detect CEC edges.)

b) Compass Operators

Compass Operators measure gradients in a selected number of directions (Fig.3). Table 2 shows four different compass gradients for north-going edges. An anti-clockwise circular shift of the eight boundary elements of these masks gives a 45° rotation of the gradient direction. For example, the eight compass gradients corresponding to the third operator of Table 2 are

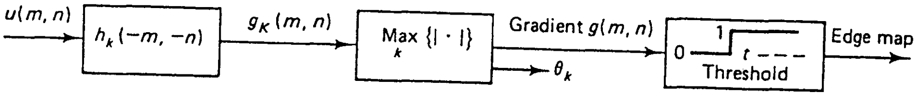


Figure 3 Edge detection via compass operators.

$$\begin{array}{rrr}1&1&1\\0&0&0\\-1&-1&-1\end{array}\ \overset{\uparrow}{(N)}\qquad \begin{array}{rrr}1&1&0\\1&0&-1\\0&-1&-1\end{array}\ \overset{\nwarrow}{(NW)}\qquad \begin{array}{rrr}1&0&-1\\1&0&-1\\1&0&-1\end{array}\ \overset{\leftarrow}{(W)}\qquad \begin{array}{rrr}0&-1&-1\\1&0&-1\\1&1&0\end{array}\ \overset{\swarrow}{(SW)}$$

$$\begin{array}{rrr}-1&-1&-1\\0&0&0\\1&1&1\end{array}\ \overset{\downarrow}{(S)}\qquad \begin{array}{rrr}-1&-1&0\\-1&0&1\\0&1&1\end{array}\ \overset{\searrow}{(SE)}\qquad \begin{array}{rrr}-1&0&1\\-1&0&1\\-1&0&1\end{array}\ \overset{\rightarrow}{(E)}\qquad \begin{array}{rrr}0&1&1\\-1&0&1\\-1&-1&0\end{array}\ \overset{\nearrow}{(NE)}$$

Let $g_K(m,n)$ denote the compass gradient in the direction $\theta_K = \pi/2 + k\pi/4$, k=0,...,7. The gradient at location (m,n) is defined as

$$g(m,n) = \max_{K} \{ \mid g_K(m,n) \mid \tag{10}$$

which can be thresholded to obtain the edge map as before. Note that only four of the preceding eight compass gradients are linearly independent. Therefore, it is possible to define four **3x3** arrays that are mutually orthogonal and span the space of these compass gradients. These arrays are called orthogonal gradients and can be used in place of the compass gradients. Compass gradients with higher angular resolution can be designed by increasing the size of the mask.

TABLE 2 · Compass Gradients (North). Each Clockwise Circular Shift of Elements about the Center Rotates the Gradient Direction by 45°

1)	$\begin{bmatrix} 1 & 1 & 1 \\ 1 & \boxed{-2} & 1 \\ -1 & -1 & -1 \end{bmatrix}$	3)	$\begin{bmatrix} 1 & 1 & 1 \\ 0 & \boxed{0} & 0 \\ -1 & -1 & -1 \end{bmatrix}$
2)	$\begin{bmatrix} 5 & 5 & 5 \\ -3 & \boxed{0} & -3 \\ -3 & -3 & -3 \end{bmatrix}$ (Kirsch)	4)	$\begin{bmatrix} 1 & 2 & 1 \\ 0 & \boxed{0} & 0 \\ -1 & -2 & -1 \end{bmatrix}$

c) Laplace operators and Zero Crossings

The forgoing methods of estimating the gradients work best when the gray-level transition is quite abrupt, like a step function. As the transition region gets wider (Fig. 4), it is more advantageous to apply the second-order derivatives. One frequently encountered operator is the Laplacian operator, defined as

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \tag{11}$$

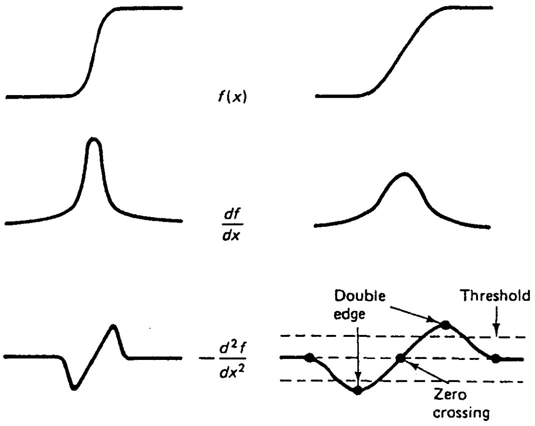


Figure 4 First and second derivatives for edge detection

Table **3** gives three different discrete approximations of this operator. Because of the second-order derivatives, this gradient operator is more sensitive to noise than those previously defined. Also, the thresholded magnitude of $\nabla^2 f$ produces double edges. For these reasons, together with its inability to detect the edge direction, the Laplacian as such is not a good edge detection operator. There is also **Stochastic Gradients** method that is powerful in the presence of noise.

TABLE 3 Discrete Laplace Operators

1) $\begin{bmatrix} 0 & -1 & 0 \\ -1 & \boxed{4} & -1 \\ 0 & -1 & 0 \end{bmatrix}$	2) $\begin{bmatrix} -1 & -1 & -1 \\ -1 & \boxed{8} & -1 \\ -1 & -1 & -1 \end{bmatrix}$	3) $\begin{bmatrix} 1 & -2 & 1 \\ -2 & \boxed{4} & -2 \\ 1 & -2 & 1 \end{bmatrix}$

In order to detect intensity changes efficiently, one should search for a filter that has two salient characteristics. First and foremost, it should be a differential operator, taking either a first or second spatial derivative of the image. Second, it should be capable of being tuned to act at any desired scale, so that large filters can be used to detect blurry shadow edges, and small ones to detect sharply focused fine detail in the image.

Marr and Hildreth argued that the most satisfactory operator fulfilling these conditions is the filter $\nabla^2 G$, where ∇^2 is the Laplacian operator $(\partial^2/\partial x^2 + \partial^2/\partial y^2)$ and G stands for the two-dimensional Gaussian distribution

$$G(x,y) = \exp -(x^2+y^2)/2\pi\sigma^2 \qquad (12)$$

which has standard deviation σ. $\nabla^2 G$ is a circularly symmetric Mexican-hat-shaped operator whose distribution in two dimensions may be expressed in terms of the radial distance **r** from the origin by the formula

$$\nabla^2 G(r) = (-1/\pi\sigma^4)(1 - r^2/2\sigma^2)\exp(-r^2/2\sigma^2) \qquad (13)$$

There are two basic ideas behind the choice of the filter $\nabla^2 G$. The first is that the Gaussian part of it, G, blurs the image, effectively wiping out all structure at scales much smaller than the space constant σ of the Gaussian. The reason why one chooses the Gaussian, rather than blurring with a cylindrical pillbox function (for instance), is that the Gaussian distribution has the desirable characteristic of being smooth and localized in both the spatial and frequency domains and, in a strict sense, being the unique distribution that is simultaneously optimally localized in both domains. And the reason, in turn, why this should be desirable property of blurring function is that if the blurring is as smooth as possible, both spatially and in the frequency domain, it is least likely to introduce any changes that were not present in the original image.

The second idea concerns the derivative part of the filter, ∇^2. The great advantage of using it is economy of computation. First-order directional derivatives, like $\partial/\partial x$ or $\partial/\partial y$, could be used, in which case one would subsequently have to search for their peaks or troughs at each orientation (as illustrated in Figure 4); or, second-order directional derivatives, like $\partial^2/\partial x^2$ or $\partial^2/\partial y^2$, could be used, in which case intensity changes would correspond to their zero-crossing (Figure 4). However, the disadvantage of all these operators is that they are directional; they all involve an orientation. In order to use the first derivatives, for example, both $\partial I/\partial x$ and $\partial I/\partial y$, have to be measured, and the peaks and troughs in the overall amplitude have to be found, This means that the signed quantity $[(\partial I/\partial x)^2 + (\partial I/\partial y)^2]^{-1/2}$ must also be computed.

Using second-order directional derivative operators involves problems that are even worse than the ones involved in using first-order derivatives. The only way of avoiding these extra computational burdens is to try to choose an orientation-independent operator. The lowest-order isotropic differential operator is the Laplacian ∇^2, and fortunately it so happens that this operator can be used to detect intensity changes provided the blurred image satisfies some quite weak requirements. Images on the whole do satisfy these requirements locally, so in practice one can use the Laplacian. Hence,in practice, the most satisfactory way of finding the intensity changes at a given scale in an image is first to filter it with the operator $\nabla^2 G$, where the space constant of G is chosen to reflect the scale at which the changes are to be detected, and then to locate the zero-crossings in the filtered image.

Unfortunately there are some errors in edge detection using the above method. Clark proofs that this method can produce phantom, or spurious, edges, which have no correspondence to significant changes in image intensity. It is seen that the phantom edges occur, for the domain of smoothed step edges, when two spatially consecutive edges have the same sense (i.e., dark to light or light to dark). For Gaussian smoothing, the strength of a phantom edge increases as the filter scale constant σ increases, while the strength of an authentic edge decreases with σ. From this observation comes the description of these edges as phantom; as we reduce σ and are looking at these edges more closely, they fade away, vanishing altogether at $\sigma=0$

III. Contour enhancement using three-arrow-shaped LGF

Our interest is to detect and enhance the edges of a noisy and low contrast images, particularly their crossing points (branches). Normally there are some irregularities in detected branch edges, using conventional explained filters (joint section of three adjacent hexagons as our interest). To detect these kinds of edges we present an special mask name it three-arrow-shaped Laplacian Gaussian Filter (TAS-LGF). The numbers and their distribution in this mask are the result of combination of läplacian of gaussian function and the shape of mentioned edges (Fig. 6). As a working sample we selected a C.E.C image (defected by operation) which consist of hexagonal cells (Photo 1). First we convolved the image with the two kinds of TAS-LGF (Figs. 6 & 7). Photos 5 and 6 show their result after convolution , and photos 10,11 show these results after the process of binarization, isolated point removement, expansion and line thinning. Almost all of corners are detected by these two filters while the results are weak in normal straight edges. So it is difficult to extract the contour by only simple algorithm. T.Saga presented a method, enhancing the image by combining FIR and smoothing filters. Extraction of the C.E.C. contour is carried out by discriminating the vertical angle information of hexagon. To complete the edges, we used more three one-directional LGF (Figs. 5,8,9). Here, the filter is composed of stepwise weighted second-order differential operator, therefore the shape is not a pure LGF. However, the original image is degraded and has few high frequency component. Stepwise shape will affect only high frequency components. Therefore, this approximation gives no effective difference to the output shape. This situation is the same as the operation of the following TAS-LGF. The image is faded naturally. Since the 1D-LGF is fitted to extract only the simple edges the vertex of the CEC hexagon is difficult to extract by that, as it may produce the missing vertices, sometimes. So we specially devised the TAS-LGF to extract the crossing parts. Photos 2,3,4 show the result of convolution of original image with three one-directional LGFs. Photos 7,8,9 indicate the thinned-line images of these results. Here is our algorithm to detect strongly the noisy hexagonal shape edges with poor contrast (like CEC). Three types of 1D-LGF's (vertical, left and right oblique) and two types of TAS-LGF's are applied to the original image. Their five outputs are summed to extract contours irrelevant to edge direction. Outputs of above filters are shown in Photos 2 to 6, while Photo 12 shows their summation. The binarization processing is applied to the summation result. In order to eliminate the noise, we used the reduction and expansion processing while we removed the isolated points, also. Finally with exerting line thinning process we achieved the result which contain most parts of edges (Photo 14). We did the same process using two dimensional LGF (Fig. 10 and Photo 13), then compared its result (Photo 15) with the result of our algorithm (Photos 16 and 17). The comparisons show that the better result belongs to the TAS-LGF algorithm that Fig. 11 shows its flow chart.

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
1	1	1	-2	-2	-2	1	1	1
1	1	1	-2	-2	-2	1	1	1
1	1	1	-2	-2	-2	1	1	1
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

Fig.5 One-directional LGF (for Ver. lines)

1	1	0	0	0	0	0	1	1
1	1	1	0	0	0	1	1	1
1	1	1	1	0	1	1	1	1
0	1	1	-4	-4	-4	1	1	0
0	0	1	-4	-4	-4	1	0	0
0	0	0	-4	-4	-4	0	0	0
0	0	0	1	1	1	0	0	0
0	0	0	1	1	1	0	0	0
0	0	0	1	1	1	0	0	0

Fig.6 Three-Arrow-Shapped LGF

0	0	0	1	1	1	0	0	0
0	0	0	1	1	1	0	0	0
0	0	0	1	1	1	0	0	0
0	0	0	-4	-4	-4	0	0	0
0	0	1	-4	-4	-4	1	0	0
0	1	1	-4	-4	-4	1	1	0
1	1	1	1	0	1	1	1	1
1	1	1	0	0	0	1	1	1
1	1	0	0	0	0	0	1	1

Fig.7 Three-Arrow-Shapped LGF

1	1	1	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0
1	1	1	1	1	0	0	0	0
0	1	1	1	-3	-3	0	0	0
0	0	1	-3	-3	-3	-3	0	0
0	0	0	-3	-3	-3	1	1	0
0	0	0	0	-3	1	1	1	1
0	0	0	0	0	1	1	1	1
0	0	0	0	0	0	1	1	1

Fig.8 Oblique one-directional LGF(Left)

0	0	0	0	0	0	1	1	1
0	0	0	0	0	1	1	1	1
0	0	0	0	1	1	1	1	1
0	0	0	-3	-3	1	1	1	0
0	0	-3	-3	-3	-3	1	0	0
0	1	1	-3	-3	-3	0	0	0
1	1	1	1	-3	0	0	0	0
1	1	1	1	0	0	0	0	0
1	1	1	0	0	0	0	0	0

Fig.9 Oblique one-directional LGF(Right)

0	0	0	0	0	0	0	0	0
0	0	-1	-2	-3	-2	-1	0	0
0	-1	-4	-8	-8	-8	-4	-1	0
0	-2	-8	2	21	2	-8	-2	0
0	-3	-8	21	60	21	-8	-3	0
0	-2	-8	2	21	2	-8	-2	0
0	-1	-4	-8	-8	-8	-4	-1	0
0	0	-1	-2	-3	-2	-1	0	0
0	0	0	0	0	0	0	0	0

Fig. 10 Two Dimensional LGF

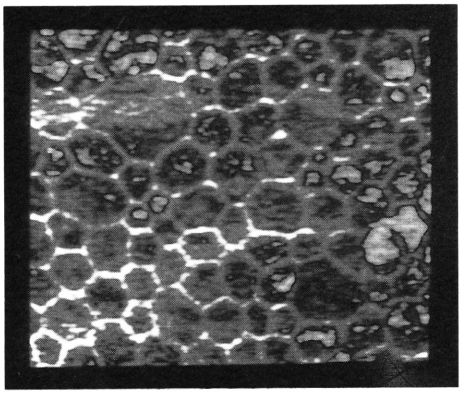


Photo.1 Original Image

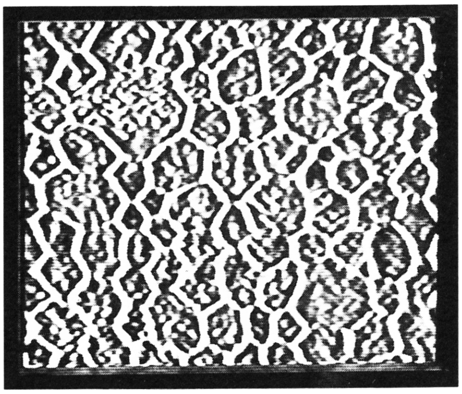


Photo.2 Output of 1D-LGF (Ver.)

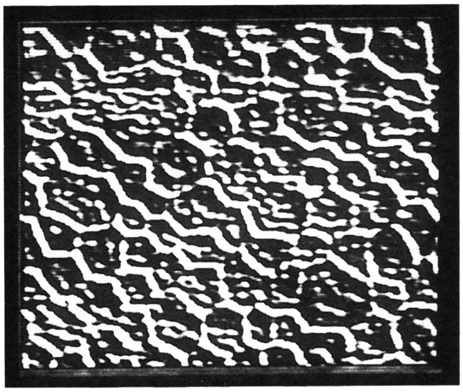


Photo 3 Output of Oblique LGF (Right)

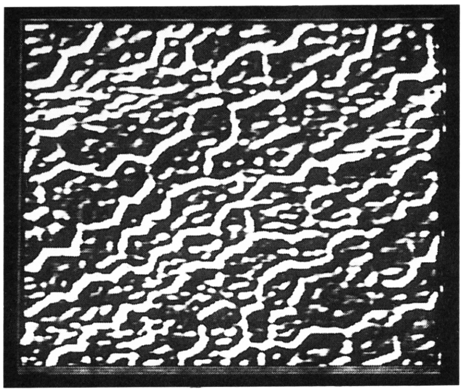
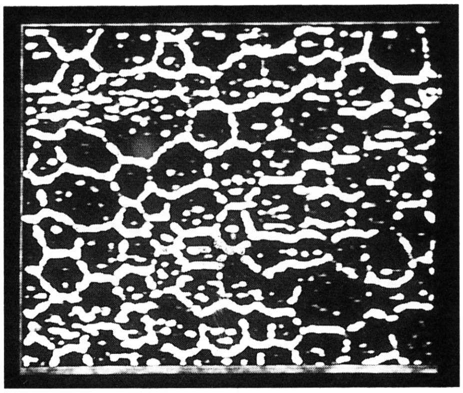
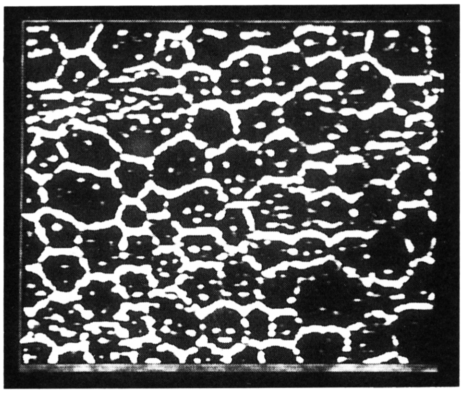


Photo 4 Output of Oblique LGF (Left)





Photos 5,6 Outputs of Three-Arrow-Shaped LGF (Two types)

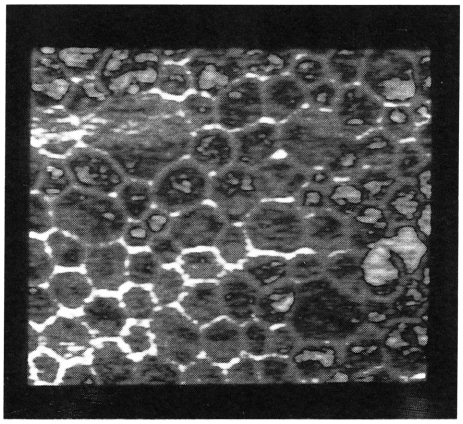


Photo 1 Original Image

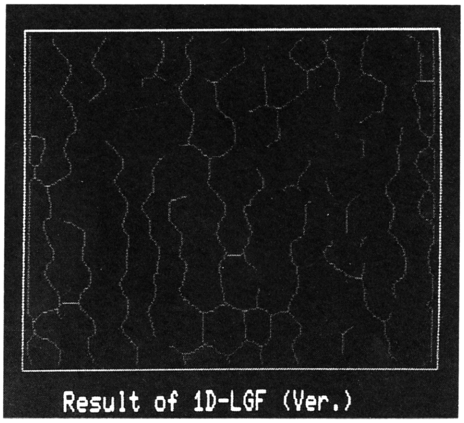


Photo 7

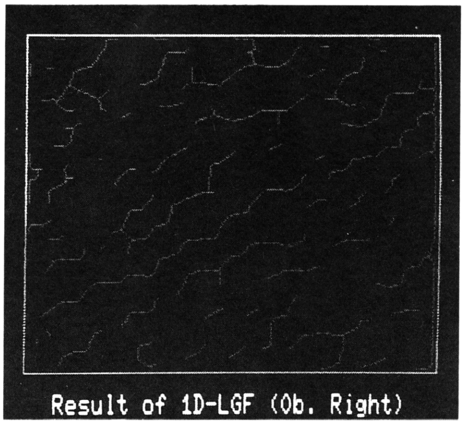


Photo 8

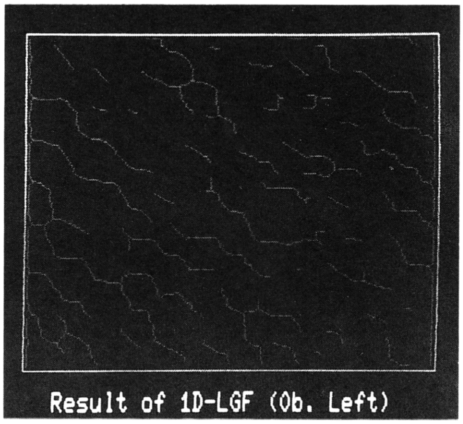


Photo 9

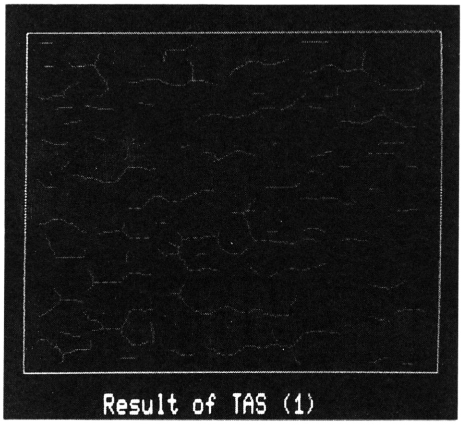


Photo 10

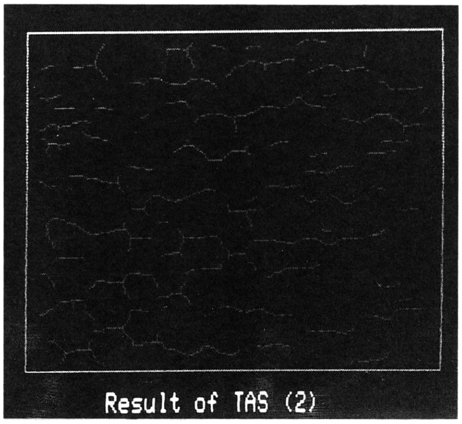


Photo 11

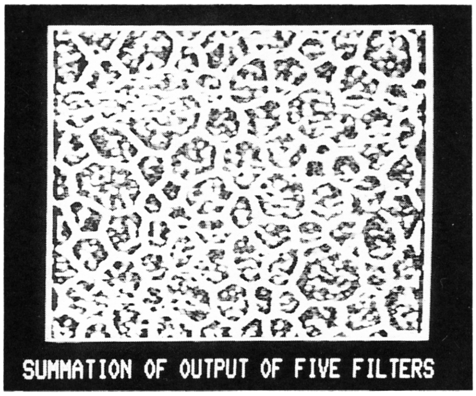


Photo 12

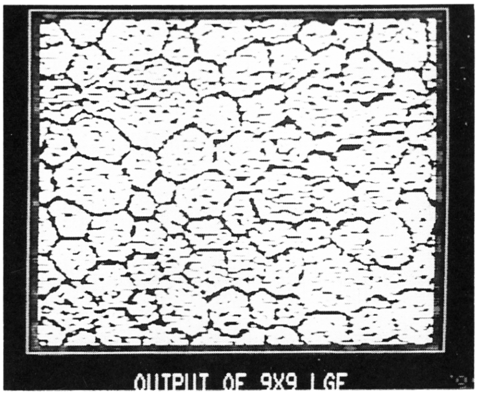


Photo 13

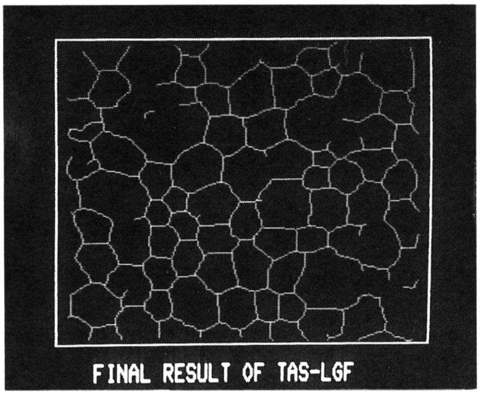


Photo 14

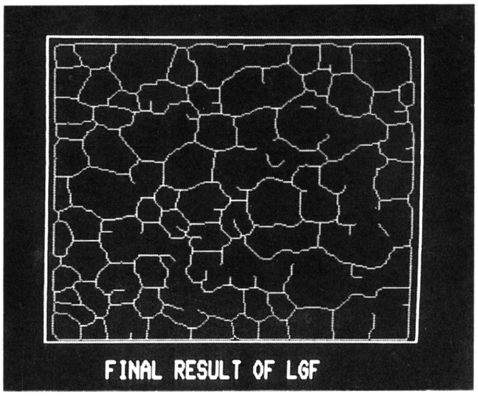


Photo 15

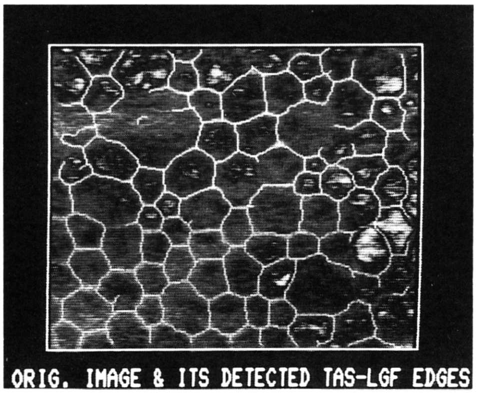


Photo 16

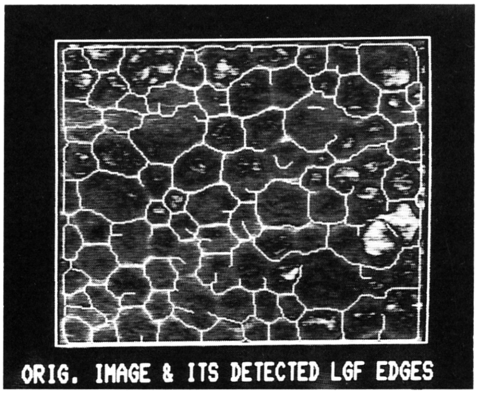


Photo 17

IV. Conclusion

We have enhanced the contour of CEC image which is low resolution, low contrast and noisy image, using Three-Arrow-Shaped LGF. Although its result still have some weaknesses, but in comparison with the result of powerful Two-Dimensional Laplacian Gaussian Filter, it seems that it is closer to real edges. However the images obtained in daily clinical diagnosis, are with less contrast and often include more noise. To cope with such situations the algorithm should be reformed to more generalized form.

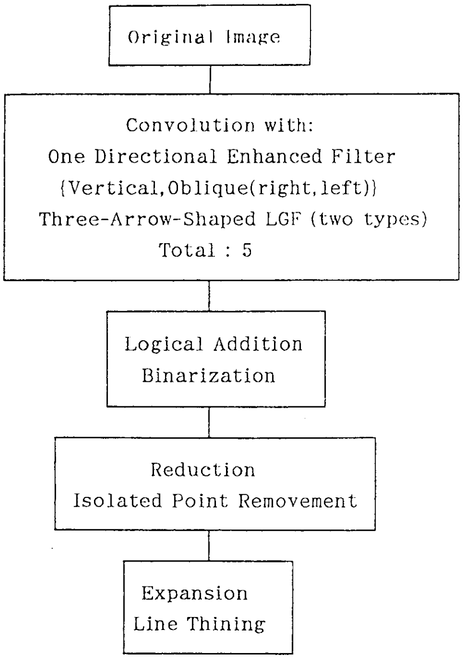

Original Image
Convolution with:
One Directional Enhanced Filter
{Vertical,Oblique(right,left)}
Three-Arrow-Shaped LGF (two types)
Total : 5
Logical Addition
Binarization
Reduction
Isolated Point Removement
Expansion
Line Thining

Fig.11 Flow Chart of Contour Extraction

References

1) J. Canny, Computational Approach to Edge Detection, PAMI-8, No. 6, pp.679-698, Nov.1986

2) J. Clark, Authenticating Edge Produced by Zero-Crossing Algorithms, PAMI-11, No.1, PP.43-57, Jan. 1989

3) A. Jain, Fundamentals of Digital Image Processing, 1989.

4) D. Marr and E. Hildreth, Theory of Edge Detection, Proc. R. Soc. Lond, B 207, PP 187-217, 1980

5) T. Saga et.al: Computer Analysus of Human Corneal Endothelial Specular micrograms, Trans of IEICE, Vol.J.65-D, No.11, PP.1451-1458, 1982.

Appendix

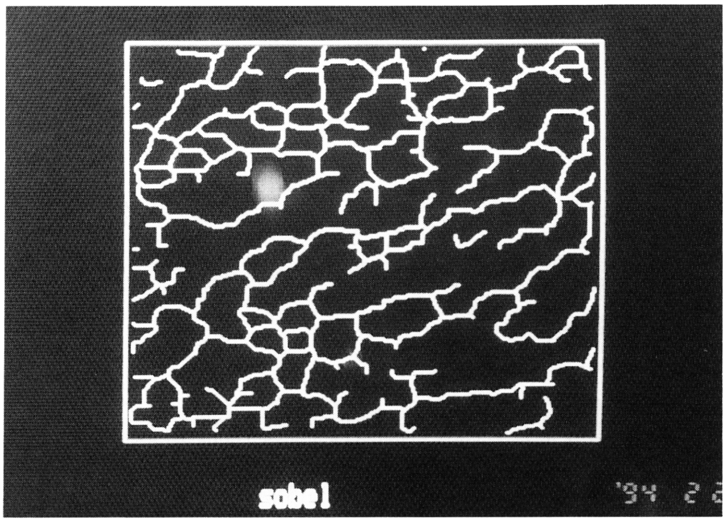


Photo 18 CEC edges, using Sobel Operator

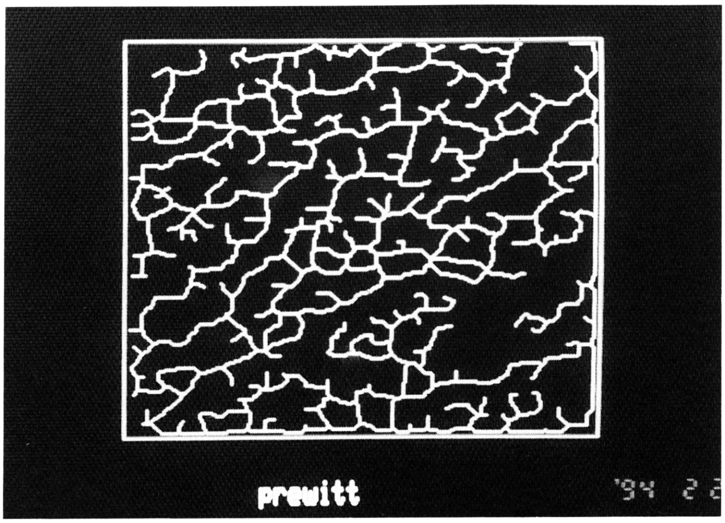


Photo 19 CEC edges, using Prewitt Operator