LOCAL THRESHOLD AND BOOLEAN FUNCTION BASED EDGE DETECTION

Muhammad Bilal Ahmad and Tae-Sun Choi, *Senior Member, IEEE*
Kwangju Institute of Science and Technology (K-JIST), Department of Mechatronics,
1-Oryong Dong, Puk Gu, Kwangju, 500-712, Korea

ABSTRACT

Localization of edge points in images is one of the most important starting steps in image processing. Many varied edge detection techniques have been proposed. Different edge detectors present distinct and different responses to the same image, showing different details. This work will present a new approach for edge detection. The actual gray level image is locally thresholded using local mean value to make a binary image. The binary image is checked for edges by comparing with the known edge like patterns, utilizing the Boolean algebra. This approach recognizes nearly all-actual edges and edges due to noise. For removing edges due to noise, we adopt another approach. This time the actual image is globally thresholded by variance value of the image. The two resulting images are logically ANDed to get the final edge map.

1. INTRODUCTION

Edge detection is one of the most important areas in lower level computer vision. The success of higher level processing heavily relies on good edges. To define an edge is difficult, as it does not correspond entirely to an image feature. Edge is defined as an image point, where gradient of image intensity function reaches its local maximum. Edges are the curves where rapid changes occur in brightness or in the spatial derivatives of brightness [3]. The changes in the brightness are places where surface orientation changes discontinuously, where one object occludes another, where a cast shadow line appears, or where there is a discontinuity in surface reflectance properties. In each case, we have to locate the discontinuity in image brightness, or its derivatives. Edge detection is the technique that yields pixels lying only on the boundary between regions. In practice, this set of pixels seldom characterizes a boundary completely because of noise, breaks in the boundary from non-uniform illumination, and other effects that introduce spurious intensity discontinuities.

Various early edge detectors [1,2] present distinct and different responses to the same image, showing different details. The edge detection methods can be classified into two types, namely, directional operators, and non-directional operators. Directional operators, look for local zero-crossings. In this process, two masks and two convolutions are used. Non-directional or gradient-based operators use single mask and convolution, but they are sensitive to noise due to gradient nature of the operators. Several edge detectors have been proposed in the literature [1,4]. The popular gradient operators are that of Sobel, Prewitt, Robert, Laplacian, etc. The operators based on surface fittings are that of Hueckel, Hartly, Haralicks facet model. The operator based on derivatives of Gaussian is Laplacian of Gaussian. Gradient based operators use thresholding for edge detection.

Types of edge detectors based on thresholding can be grouped into two classes: (a) local techniques, which use operators on local image neighborhoods and (b) global techniques, which use global information and filtering methods to extract edge information. Both methods have their advantages and disadvantages on various types of images. Nearly all detectors utilize thresholding of the image for edge detection. Each pixel in the image is compared with this threshold value. If the pixel's intensity is higher than the threshold value, the pixel is set to, say, white in the output image. If it is less than the threshold, it is set to black. The efficient selection of single threshold value is the most important and the most difficult process in edge detection technique. Edge detectors based on local techniques, use local feature for selecting threshold value. Similarly, edge detectors based on global techniques, use global feature for selecting threshold value. We, however, utilize both local and global feature for thresholding the image.

The most popular methods are based on gradients of the image, which are sensitive to changes in the intensity values of the image. The original image is filtered with a suitable mask in the gradient operators. Edge points are detected in places where the original image has a high rate of intensity change. Edge points are detected by selecting an optimum global threshold value. An image contains variations at different levels. Use of a single global threshold over the whole image gives poor results. To avoid this we use a locally derived threshold [6] based on the local mean value. But local threshold method also detects false edges due to noise. To avoid this, a global threshold based on variance filtering is used [5]. The global threshold value depends on the presence of

Manuscript received June 28, 1999

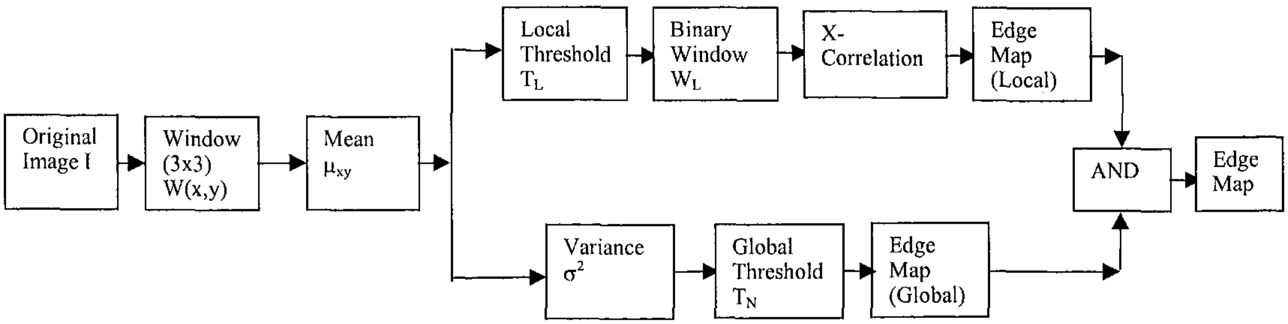


Fig. 1 A block diagram of the proposed method.

noise in the image. Our method exploits the advantages of local threshold methods, as well as global threshold methods. Boolean algebra is used to achieve some part of the algorithm, so this algorithm can be efficiently implemented in hardware.

We divide our work in four sections. Section 2 contains information about the proposed algorithm. Section 3 contains the results of the method. Finally, section 4 contains the conclusion. At the end we show some references.

2. PROPOSED ALGORITHM

Our algorithm is based on local operations, global operations, and Boolean algebra. We take window of size (3x3) of the original gray-level image. Local threshold is found on the basis of local mean value. This local threshold is used for thresholding the image points. This converts the gray-level image into binary image. The resulting binary image window (3x3) is cross-correlated with sixteen edge-like binary patterns. We use Boolean functions in the cross-correlation of the image window. The resulting intermediate edge map contains true edges as well as false edges. The false edges are generated by noise in the image. These false edges are removed by another parallel path. In this path, the same window (3x3) of the image is thresholded globally, but now on the basis of variance of the window. The global threshold is pre-selected, considering the presence of noise in the image. The resulting intermediate edge map is logically ANDed with the intermediate edge map from local threshold. The block diagram is shown in Fig. 1. Each step is explained in details now.

2.1 Thresholding (Local operation)

The idea of global and multiple thresholding has been applied in the various edge detectors. Not all images can be neatly processed for edges using simple thresholding. The intensity histogram of an image has different picture for different types of images. In global thresholding, we expect to see a distinct peak in the histogram for determining the global threshold for the entire image. But in most cases, such peaks do not exist, then it is unlikely that simple thresholding will produce a good result. In this case, adaptive thresholding may be a better answer.

Adaptive thresholding changes the threshold dynamically over the image. This more sophisticated version of thresholding can accommodate changing lighting conditions in the image, e.g., those occurring as a result of a strong illumination gradient or shadows. In adaptive thresholding, a threshold is calculated for each pixel in the image.

There are different methods for adaptive thresholding. One of them is local thresholding. To find the local threshold, we statistically examine the intensity values of the local neighborhood of each pixel. The statistic, which is most appropriate, depends largely on the input image. Simple and fast functions include the mean of the local intensity distribution.

Local threshold (T) value for each center pixel is selected either as

T = Mean, Or

T = Median Or

T = (Max+Min)/2, Or

T = (Max-Min)/2.

We use the mean value approach, as it is already calculated for the variance function.

On the margin, however, the mean of the local area is not suitable as a threshold, because the range of intensity values within a local neighborhood is very small and their mean is close to the value of the center pixel [7]. The situation can be improved if the threshold employed is not the mean, but (mean - C), where C is a constant. We use (mean - C) for local thresholding.

For image window W (x,y) of size 3x3, we calculate mean as

$$\begin{bmatrix}0&1&1\\0&1&1\\0&1&1\end{bmatrix}\begin{bmatrix}0&0&0\\1&1&1\\1&1&1\end{bmatrix}\begin{bmatrix}1&1&0\\1&1&0\\1&1&0\end{bmatrix}\begin{bmatrix}1&1&1\\1&1&1\\0&0&0\end{bmatrix}\quad\begin{bmatrix}1&1&1\\0&1&1\\0&0&1\end{bmatrix}\begin{bmatrix}1&0&0\\1&1&0\\1&1&1\end{bmatrix}\begin{bmatrix}0&0&1\\0&1&1\\1&1&1\end{bmatrix}\begin{bmatrix}1&1&1\\1&1&0\\1&0&0\end{bmatrix}$$

$$\begin{bmatrix}0&1&1\\0&1&1\\0&0&1\end{bmatrix}\begin{bmatrix}1&0&0\\1&1&0\\1&1&0\end{bmatrix}\begin{bmatrix}1&1&1\\0&1&1\\0&0&0\end{bmatrix}\begin{bmatrix}1&1&1\\1&1&0\\0&0&0\end{bmatrix}\quad\begin{bmatrix}0&0&0\\0&1&1\\1&1&1\end{bmatrix}\begin{bmatrix}0&0&0\\1&1&0\\1&1&1\end{bmatrix}\begin{bmatrix}0&0&1\\0&1&1\\0&1&1\end{bmatrix}\begin{bmatrix}1&1&0\\1&1&0\\1&0&0\end{bmatrix}$$

Fig. 2 Sixteen Masks corresponding to edge-like patterns.

$$Mean\mu = \frac{1}{N \times N} \sum_{x=0,y=0}^{x=N,y=N} W(x,y), and$$

local threshold $T_L(x,y)$ for each center pixel of window W (x,y) is selected as

$T_L(x,y) = (\mu - C)$, where C is a constant.

The window W(x,y) is thresholded as

$W_L(x,y) = 1$

if $W(x,y) > T_L(x,y)$,

otherwise, $W_L(x,y) = 0$.

2.2 Boolean Functions (Local operation)

For edge finding, the window $W_L(x,y)$ is cross-correlated with sixteen edge like patterns. Any pattern which matches the window $W_L(x,y)$ is called an edge at the center of the window W(x,y). The sixteen patterns, as shown in Fig. 2, are like Prewitt compass masks [2]. These patterns cover nearly all-possible edge patterns in every direction. The cross-correlation of the window $W_L(x,y)$ with edge patterns are accomplished by Boolean functions. One Boolean equation is expressed in eq.1, for the first mask. All similar equations are ORed to get either one or zero at the center of the window W(x,y).

B0 = !B(0,0)xB(0,1)xB(0,2)x!B(1,0)xB(1,1)xB(1,2)x !B(2,0)xB(2,1)xB(2,2) eq. (1).

2.3 False Edge Rremoval (Global Thresholding)

So far in the work, false edges are detected due to the presence of noise. We now remove the false edges with global threshold approach. We take a new threshold T_N, whose value is related with the noise level in the image. As variance function has its maximum value at an edge. So, the variance is calculated again locally for each window W (3x3) and thresholded as

If $\sigma^2_{xy} > T_N$, B(x,y) = 1, otherwise B(x,y) = 0, and

$$\sigma^2_{x,y} = \frac{1}{N \times N} \sum_{x=0}^{x=N-1} \sum_{y=0}^{y=N-1} \left[g(x,y) - \mu_{x,y}\right]^2,$$

where g(x,y) is the intensity value of the window W(x,y), μ is the mean of the neighbors (3x3) at (x,y) position, and NxN is the window size.
The local threshold (mean - C) gives better edge localization, while global threshold (variance) limits the spread of pixel intensity, above which edge presence is maximum.

3. RESULTS

We apply the proposed method on various real gray-level images. Results are shown in Fig. 3. The results are obtained with conditions of global noise Tn = 20, and the constant C = 5. Fig. 3 (a) shows the original image. Fig 3

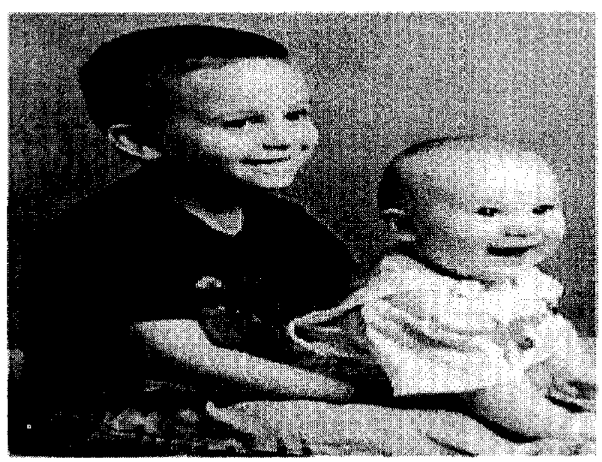


Fig. 3 Using Proposed Algorithm, with Tn = 20, C = 5; (a) original gray-level image

(b) shows the image after applying only the local threshold (mean – C). Fig 3 (c) shows the image after applying the global threshold (variance). Finally, in Fig 3 (d), the two operations are combined to get the final edge image of the original image. Fig 3 (b) shows that the maximum possible edges in the image are detected. This edge map contains both true as well as false edges. Fig 3 (b) is obtained by using only local threshold (mean – C), and after cross-correlation with the sixteen edge-like patterns. Fig 3 (c) shows the image obtained by only global threshold, and variance of the image. The variance function limits the spread of pixel intensity, above which edge presence is maximum. Fig 3 (d) shows after logically ANDing the two edge maps obtained from local and global operations. It means we ANDed the Fig 3 (b) and Fig 3 (c) to get the final edge detection in Fig 3 (d).The results of our proposed algorithm on 'Lenna' image are shown in Fig. 4. The results for 'Pepper' image are shown in Fig 5.

The results of proposed method are good. Noise is minimized, and lines are thinner, as compared to the

Fig. 3 (b) Using only (mean – C) on the original image

Fig 3 (c) Using only Variance on the image

Fig. 3 (d) Using complete algorithm on the image

Fig.4 (a) 'Lenna' Image (original)

Fig 4 (b) 'Lenna' Image after applying the proposed algorithm results of gradient based detectors. The proposed method is somehow complex, but the overall performance is better.

Fig 5 (a) Using proposed algorithm with Ten = 50, C=3, original "peppers" image

Fig 5 (b) Using only (mean –C) on the image

Fig 5 (c) Using only Variance on the image

Fig 5 (d) Using complete algorithm on the image

4. CONCLUSIONS

The proposed method detects edges in two processes. In first process, image is locally thresholded, and cross-correlated with edge like patterns, using Boolean algebra. This process detects true edges as well as false edges. The second process detects the existence of true edges only. Local spread of intensity values in the image is measured. Image is thresholded with global threshold, based on the noise level in the image. The two processes are combined to give rise to the final edge map. First process detects maximum possible edges, while second process determines the location of true edges only.

We use (mean – C) for local thresholding, as it is already calculated for variance function in our method. We use local edge detection by cross-correlating with the edge-like patterns, using boolean functions. So our method can also be implemented easily in hardware.

The results of our method are good. Our method minimizes the noise, and also edge lines are thinner, as compared to the results of gradient based detectors. The proposed method is somehow complex, but the overall performance is better.

References

[1] Rapheal C.Gonzalez and Richard E. Woods "Digital Image Processing ". Addison Wesley, 1992.

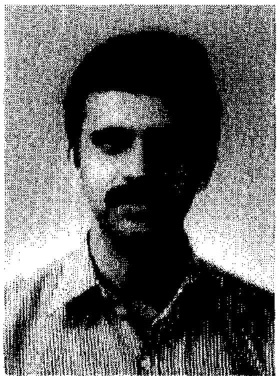
[2] M A. Sid-Ahmed, "Image Processing", McGraw-Hill, Inc.

[3] Berthold Klaus Paul Horn "Robot Vision" The MIT Electrical Engineering and Computer Science Series.

[4]Marek Brejil, "Comprehensive Exam", Dept. of Electrical and Computer Engineering University of Iowa, March 9, 1998.

[5] Robert M. Haralick, Linda G. Shapiro "Computer and Robot Vision", vol. 1.

[6] A. Elmabrouk and A. aggoun, "Edge detection using local histrogram analysis", Electronics Letters, 11th June 1998, Vol. 34 No. 12.
[7] "Adaptive Thresholding", at http://image.korea.ac.kr/HIPR/HTML/adpthrsh.htm.



Muhammad Bilal Ahmad was born in D.I.Khan, Pakistan in 1970. He received the B.S. degree in electrical engineering from N_W.F.P, University of Engineering and Technology, Peshawar, Pakistan, in 1993. He also received a post graduate diploma in computer systems software and hardware. He is currently a graduate student in the Department of Mechatronics at Kwangju Institute of Science and Technology, (KJIST) Kwangju, Republic of Korea. His research interests include image processing, digital signal processing, machine/robot vision, data communications.

Tae-Sun Choi received the B.S. degree in electrical engineering from Seoul National University, Korea, in 1976, the M.S. degree in electrical engineering from Korea Advanced Institute of Science and Technology in 1979, and the Ph.D. degree in electrical engineering from the State University of New York at Stony Brook in 1993. He is currently an assistant professor in the Department of Mechatronics at Kwangju Institute of Science and Technology in Korea.
His research interests include image processing, machine/robot vision, and visual communications. He is a senior member of IEEE.