Volume 5, No. 10, October 2014
Journal of Global Research in Computer Science
ISSN-2229-371X
RESEARCH PAPER
Available Online at www.jgrcs.info

A NOVEL APPROACH TO GRADIENT EDGE DETECTION

[1]Naorah Saad Al Harbi. Supervisor: Dr. Msater Prince Syed[2]
College of Computer & Information Technology, Department of Computer Science, Qasim University, Kingdom of Saudi Arabia.
[1]joody20082010@hotmail.com, syedprinceiqbal@hotmail.com[2]

Abstract: Edges characterize boundaries and are therefore a problem of fundamental importance in image processing. Image Edge detection significantly reduces the amount of data and filters out useless information, while preserving the important structural properties in an image. The aim of our proposed method is to obtain thin edges so that the result is more suitable for further applications such as boundary detection image segmentation, object identification and so on. In this study, we propose a new two approaches based on gradient edge detection method. We only focus on traditional Canny Edge Detection in our methods and introduce convolution masks to obtain better edges. The proposed project is implemented using MATLAB 7.0.

Keywords: Edge Detection, Digital Image Processing, Canny Edge Detector.

INTRODUCTION

The purpose of image processing is to get better image and enhance and detect/ classify/segment the objects in the image which are not visible. In order to understand the contents in image during image processing, it is vital to partition the image into objects in an image and background of the image. Partition of the image into object and background is a severe step in Image interpretation. Edge detection is a fundamental tool used in most image processing applications to obtain information from the frames.[8]

Edge detection is the process of finding edge in image. Edge refers to pixel positions of the image where significant abrupt local changes of intensity (either gray or color) occur.[6] There are many ways to perform edge detection.

However, the majority of different methods may be grouped into two categories: First-Order Derivatives and Second-Order Derivatives. The first-order derivatives in an image are computed using the gradient,[3] and the second-order derivatives are obtained using the computation of second-order directional derivatives to identity locations with zero crossing.[5] some of the early methods include the Robert, Prewitt, Sobel and canny edge operators .They used small convolution masks to approximate the first derivative of the image brightness function. Literature survey shows that the Canny's edge detection algorithm performs better than all these operators under almost all scenarios. In practice, the optimal filter can be closely approximated by the first-order derivatives of a Gaussian function which is a 2-D convolution operator that is used to `blur' images and remove detail and noise.[7]

LITERATURE REVIEW

The specialists consider the process of edge detection as a significant and important procedure that contributes to the analysis of different visual scenes. The basic edge detection operator is a matrix area gradient operation that determines the level of variances between different pixels. Examples of gradient-based edge detectors are the Robert, Prewitt, Sobel and canny edge operators. All the gradient -based algorithms have kernel operators that calculate the strength of the slope in directions which are orthogonal to each other, commonly vertical and horizontal.

The realization of adaptation of Canny's method includes, therefore, four consequent steps: convolution operation, differentiation, non-maximum suppression technique, and thresholding segmentation. Thus, convolution starts with Gaussian filter for reducing the noise in the source image.[4] The Canny operator can use the filter which is close to the first derivative of a Gaussian.[2]

$$G(x,y) = \frac{1}{\sqrt{2\pi\sigma}} e^{\frac{-1}{2}\left(\frac{x^2}{\sigma_x^2}+\frac{y^2}{\sigma_y^2}\right)} \quad (1)$$

and

$$\frac{\partial G(x,y)}{\partial x} = \frac{-x}{\sigma_x^2} G(x,y) \quad (2)$$

$$\frac{\partial G(x,y)}{\partial y} = \frac{-y}{\sigma_y^2} G(x,y)$$

The 2-Dconvolution operation is described in the following equation:

$$G = H \times F \quad (3)$$

$$G_{k,l} = \sum_{i=k}^{k+M-1} \sum_{j=l}^{l+N-1} H_{i,j} \times F_{i-k+1,j-l+1}$$

H=Original image.
F=Convolution kernel.
G=Filtered image.

The non-maximal suppression step finds the local maxima in the direction of the gradient and suppresses all others, minimizing false edges. The local maximum is found by comparing the pixel with its neighbors along the direction of the gradient. This helps to maintain the single pixel thin edges before the final threshod step. Instead of using a single static threshold value for the entire image, the canny algorithm introduced hysteresis threshold, which has some adaptively to the local content of the image. Hysteresis uses two thresholds and if the magnitude is below the first

threshold, it is set to zero (a non-edge).If the magnitude is above the high threshold, it is made an edge. And if the magnitude is between the two thresholds, then it is set zero, unless there is a path from this pixel with a gradient above the high threshold. Canny edge detector uses two convolution masks one for horizontal direction and one for vertical direction they are shown in figure.1.[1]

-1	-1	-1
0	0	0
+1	+1	+1

a) horizontal direction

-1	0	+1
-1	0	+1
-1	0	+1

b) vertical direction

Figure 1: Convolution masks Canny method.

METHODOLOGY

Methodology is divided in two parts:

a. Proposed Quad Edge Detection method, working algorithm using four convolution masks .
b. Proposed Enhanced Edge Detection method, working algorithm using six convolution masks.

We proposed Quad Edge Detection method based on Canny Edge Detection .We introduce two more additional convolution masks in addition to two convolution masks used in Canny Edge Detection method. First one for horizontal, second one for vertical, third one for positive left diagonal, and fourth one for positive right diagonal .They are shown in figure.2.

-1	-1	-1
0	0	0
+1	+1	+1

c) Positive left diagonal

-1	0	+1
-1	0	+1
-1	0	+1

d) Positive right diagonal

Figure 2: Convolution masks for proposed Quad method.

The detailed algorithm is given below:

Step1: In order to implement the proposed edge detector, the first to filter out any noise using Gaussian filter in the original image. Then convolve the image with four masks; looking for horizontal, vertical, positive left diagonal, positive right diagonal edges. The direction producing the largest result at each pixel point is marked. Record the convolution result and the direction of edge at each pixel.

Step2: To find the edge strength by taking the gradient of the image. Norm vector of the gradient image (NVI) is calculated as:

$$NVI = SQRT(Ix^2 + Iy^2 + Pld^2 + Prd^2) \quad (4)$$

Step3:Once the edge direction is known, the next step is to relate the edge direction to a direction that is to traced in an image.

Step4: Apply non maximum suppression. Non maximum suppression is used to trace along the edge in the edge direction and suppress any pixel value(sets if equal to 0) that is not considered to be an edge. This will produce a thin line in the output image. The algorithm of non maximum suppression is given below.

Algorithm

For each pixel (x,y) do:

if $magn(i, j) < magn(i_1, j_1)$ or $magn(i, j) < magn(i_2, j_2)$
then $I_N(i, j) = 0$
else $I_N(i, j) = magn(i, j)$

Step5: Apply hysteresis, it is used as a means of eliminating streaking. Streaking is the breaking up of an edge contour caused by the operator output fluctuating above and below the threshold.

The element of the thresholded image matrix (Ibw) is selected either from pixel values of image after calculating Norm vector of the gradient it (NVI),or newly generated pixel values, which one is larger, expressed in equation (5).

(Ibw)= max(NVI,level.*ones(size(NVI)) (5)

where level is the threshold value, computed using equation (6).

Level = ((alfa*(I_max -I_min))+Imin) (6)

Step6: Finally, Thinning operation using interpolation to find the pixels where the norms of gradient are local maximum. Local maximum of
the contour of the processed image is determined. The pixels are retained whose values are same as the local maxima while the rest are masked.

All the gradient edge detection methods and the proposed Quad edge detection methods uses only limited number of convolution masks. Because of this, the edges detected by these methogods are not accurate. We propose a new method to overcome this problem called Enhanced Gradient Edge detection method.

We introduce two more convolution masks in addition to the four convolution masks used in the proposed Quad edge detection method. The additional convolution masks are negative left diagonal and Negative right diagonal. They are shown in figure3.

0	-1	-1
+1	0	-1
+1	+1	0

(a)

+1	+1	0
+1	0	-1
0	-1	-1

(b)

Figure 3: (a)Negative left diagonal (b) Negative right diagonal

RESULTS

The performance of Canny and the proposed edge detection methods depend heavily on the adjustable parameter sigma (σ), which is the standard deviation for the Gaussian filter and the threshold parameter alpha (α).

The standard deviation σ also controls the size of the Gaussian filter.

The greater the value for σ, the size of Gaussian filter becomes larger. This implies more blurring which is necessary for noisy images as well as detecting larger edges. Smaller values of σ imply a smaller Gaussian filter that limits the amount of blurring and maintaining finer edges in the image.[2]

Experimental results were tested with various sets of normal images, considering different values of standard deviation sigma (σ).

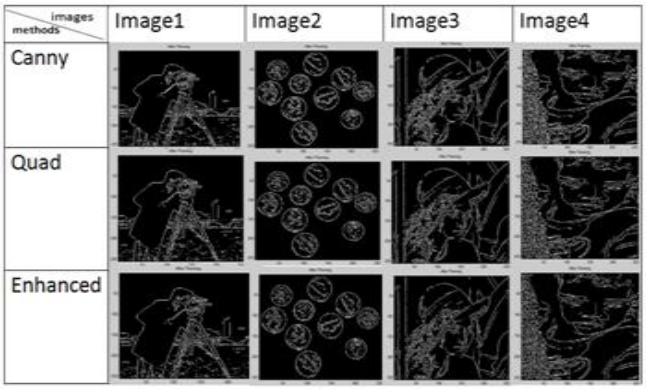
We will now proceed to implement the canny, Quad and Enhanced Edge Detection methods in MATLAB with several σ values on several images that have size 256x256 pixels . To illustrate the effect of the Canny, Quad Edge Detection method and Enhanced Edge Detection method on images.

Case1:
The parameters of the Gaussian filter are n = 10, sigma=3 and threshold parameter alpha = 0.1.
Case 2:
In this case the parameters of the Gaussian filter are n=10, sigma = 1 and threshold parameter alpha = 0.1.

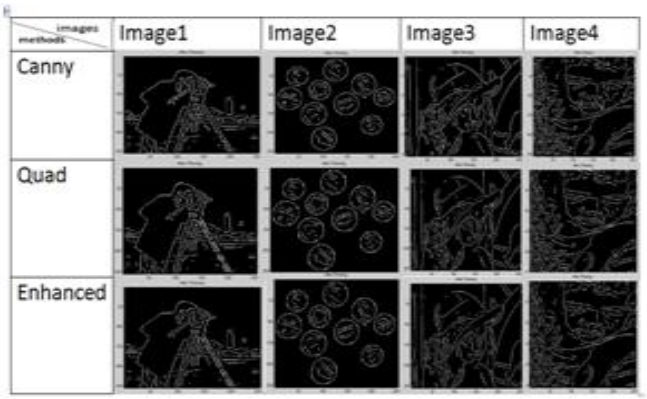
We used the four original images in the figure 4 .The images obtained by using Canny ,Quad Edge Detection method and Enhanced Edge Detection method are given in Table 1 . and Table 2. respectively.

Figure 4: Samples input images.

Tables 1: After using case 1 of the Canny, Quad and Enhanced methods.



Tables 2: After using case 2 of the Canny, Quad and Enhanced methods.



In the table 3, the experimental result of case1 (where sigma=3) shows that the average of computational time for applying the Canny edge detection algorithm is 30.313225 seconds, Enhanced gradient edge detection method is 99.19637 seconds and by Quad edge detection is 70.84442 seconds.

In case2 (where sigma=1) found from the experimental results, the average of computational time for the canny is 21.718975 the Enhanced gradient edge detection method is 58.96587 seconds and by the Quad edge detection method is 42.55435 seconds.

Tables 3: Computational time of 256×256 pixel images.

			Gradient Edge Detection Methods		
			canny	quad	enhanced
Computational time of 256×256 pixel images	1-Cameraman	Sigma=1	20.4154	39.6813	55.3711
		Sigma=3	27.4843	65.7382	92.041
	2-coins	Sigma=1	23.296	38.4696	45.6228
		Sigma=3	29.3372	72.0626	92.5995
	3-lena	Sigma=1	17.3415	46.0418	59.4586
		Sigma=3	32.3709	59.585	107.83
	4-boy	Sigma=1	25.823	46.0246	75.411
		Sigma=3	32.0605	85.9919	104.315
	Average seconds	Sigma=3	30.313225	70.84442	99.19637
		Sigma=1	21.718975	42.55435	58.96587
Computational complexity			O (N^2)	O ($2N^2$)	O ($3N^2$)

We used Mean square error (MSE) and peak signal-to-noise ratio (PSNR) to measure Edge detector performance. Mean square error (MSE) is given by:[4]

$$MSE = \sum_{i=j=1}^{N} \frac{[f(i,j)-F(i,j)]^2}{N^2} \quad (7)$$

Where, f is the original image F is the filtered image and N is the size of image.
PSNR in decibels is easily defined from MSE as given below:[4]

$$PSNR = 10 \log_{10}(255^2 / MSE) \quad (8)$$

Higher the PSNR, the better degraded image has been reconstructed to match the original image and the better the reconstructive algorithm.

Table 4 gives the Comparison of the PSNR values for the Canny, Quad and Enhanced edge detection method. This table indicate the resulting PSNR values of Quad Edge Detection on normal images more accurate than Canny, and The resulting PSNR values of Enhanced Edge Detection on normal images, gives more accurate than Canny and it more accurate than Quad Edge Detection.

Table 4: PSNR of 256×256 pixel images.

			Gradient Edge Detection Methods		
			Canny	quad	enhanced
PSNR OF 256×256 PIXELS IMAGE	1-Cameraman	Sigma=1	13.9811	14.0471	14.1251
		Sigma=3	13.9928	14.0856	14.191
	2-coins	Sigma=1	12.2609	12.2975	12.3419
		Sigma=3	12.2787	12.35	12.432
	3-lena	Sigma=1	10.6581	10.7084	10.7717
		Sigma=3	10.6791	10.7624	10.8658
	4-boy	Sigma=1	10.7643	10.8263	10.9053
		Sigma=3	10.7858	10.8832	11.0052
	Average performance	Sigma=3	11.9341	12.0203	12.036
		Sigma=1	11.9161	11.932425	12.1235

a. ***Summary of results:*** After simulating different images on the canny, Quad and Enhanced Edge Detection methods we conclude that the enhanced Edge Detection method finding the edges of images with greater efficiency as show from figures and tables as compared to quad and canny methods.

Clearly, the proposed gradient edge detection methods accurately detects edges like straight lines, elliptic and circular edges.

Experimental results reveal that the Enhanced gradient edge detection method is perhaps the most elaborate edge detection method, which gives very good results. However, this method takes longer time due to its higher degree of complexity.

CONCLUSION

The enhanced and quad proposed methods were tested on gray level test images and compared the results and performance between tow proposed methods and Canny. Experiment results have demonstrated that the enhanced and quad proposed methods for edge detection produces satisfactory results for different gray level images more than canny. And the enhanced Edge Detection method finding the edges of images with greater efficiency when compared with quad and canny methods. However, this method takes longer time to compute results due to its higher degree of complexity.

REFERENCES

[1] Canny. J, "A computational approach to edge detection", IEEE Trans. Pattern Analysis and Machine Intelligence, vol 8, pp. 679-714, 1986.

[2] Fisher .R, Perkins .S, Walker. A, Wolfart .E, 2003, Retrieved from http://homepages.inf.ed.ac.uk/rbf/HIPR2/gsmooth.htm.

[3] Juneja. Mamta , Sandhu .Parvinder Singh, "Performance Evaluation of Edge Detection Techniques for Images in Spatial Domain", International Journal of Computer Theory and Engineering, Vol. 1, No. 5, December, 2009, 1793-8201.

[4] Kaur .Beant,"Comparative study of different edge detection techniques",International Journal of Engineering Science and Technology (IJEST) ISSN : 0975-5462 , Vol. 3 no, 3 March 2011.

[5] Lee .Tzu-Heng Henry," Edge Detection Analysis". National Taiwan University.

[6] Nadernejad .Ehsan, Hassanpour .Hamid, Sharifzadeh .Sara, "Edge Detection Techniques: Evaluations and

[7] Torre. V , T. A, Poggio, "On edge detection". IEEE Trans. Pattern Anal. 1986.

[8] Umbaugh. S.E, Digital Image Processing and Analysis: Human and Computer Vision Applications with CVIPtools , 2nd Edition, CRC Press, Taylor & Francis Group, Boca Raton, 2010,pp. 5-12,pp. 341-347.

[9] Comparisons", Applied Mathematical Sciences, Vol. 2, 2008, no. 31, pp.1507 – 1520.

