International Journal of Advanced Computer Research (ISSN (print): 2249-7277 ISSN (online): 2277-7970)
Volume-3 Number-4 Issue-13 December-2013

Novel Method for Edge Detection for Gray Scale Images using VC++ Environment

Harsimran Singh[1], Tajinder kaur[2]

Abstract

Edge detection is the most common approach for detecting discontinuities in gray scale images. By using edge detection important features can be extracted from an image in the form of edges. The number of researches has found so many techniques to detect edges. It is very important to locate the edges which are true as every operator is sensitive to certain type of edges. It is important to choose an edge detector that is best suited for the application. Out of these techniques canny operator is the best method to detect true edges. In this paper all the techniques are reviewed and a novel technique is proposed to detect true edges by using the concept of Adaptive thresholding. Each pixel value of gradient magnitude is compared with adaptive threshold value and edges are detected. We also compared canny operator with the proposed algorithm in terms of PSNR and MSE.

Keywords

Thresholding, Mask, Sobel, Canny, Suppression, kernel

1. Introduction

Edge detection is a name for a set of mathematical methods which are used to identify points in a digital image. Points are identified at the locations where image brightness changes sharply or has discontinuities. The points at which image brightness changes sharply are organized into a set of curved line segments known as edges [13]. Different Operators are used for Edge detection. Each operator is designed to be sensitive to certain types of edges.
Edge detection is a fundamental tool in image processing, machine vision and computer vision and in the areas of feature detection and feature extraction particularly. Each operator works better in different conditions. So while detecting edges we have to select any one of the operator that identify true edges. So, it depends on different variables that are involved in the selection of an edge detection operator.

These variables are Edge orientation, Good localization, Noise environment and the structure of an edge. The direction of an edge should be determined in such a way that it should be most sensitive to edge.

Edges can be detected in different directions like horizontal, vertical, or diagonal edges. It is difficult to detect an edge in noisy image. So to reduce the amount of noise from an image we have to convert it to noiseless. A noisy image results in less accurate localization of the detected edges [8]. Localization means that distance between edge pixels and edge lines should be minimum.

All the edges do not involve a step change in intensity. Effects of refraction or poor focus can result in objects with boundaries defined by a gradual change in intensity. The operator that is to be chosen should be responsive to such a gradual change in those cases. So, while detecting edges there may be the possibility of false edge detection where true edges are missing. High computational time and problems can occur due to noise etc. So we have to analyze every edge detection operator that how it works in different conditions. All edges are locally directional. Therefore, the goal in edge detection is to find out what occurred perpendicular to an edge. [12]

It has been found from the previous researches that Canny gives better results for true edges. In this paper a novel method is proposed to detect edges of a gray scale images which gives better results than canny operator.

2. Techniques for Edge Detection

There are two techniques that are used to detect edges in gray scale images:

First Order Derivative: [4, 11]

Sobel Operator: The operator consists of a pair of 3×3 convolution kernels as shown in Figure. One kernel is simply the other rotated by 90°. [1, 2, 3]

-1	**0**	**+1**
-2	**0**	**+2**
-1	**0**	**+1**

Gx

+1	**+2**	**+1**
0	**0**	**0**
-1	**-2**	**-1**

Gy

Figure 1: Convolution masks of sobel operator

These kernels are designed to respond maximally to edges running vertically and horizontally relative to the pixel grid, one kernel for each of the two perpendicular orientations. The kernels can be applied separately to the input image, to produce separate measurements of the gradient component in each orientation (call these *Gx* and *Gy*). These can then be combined together to find the absolute magnitude of the gradient at each point and the orientation of that gradient. The gradient magnitude is given by:

$$|G| = \sqrt{Gx^2 + Gy^2} \quad (1)$$

Typically, an approximate magnitude is computed using:

$$|G| = |Gx| + |Gy| \quad (2)$$

which is much faster to compute.
The angle of orientation of the edge (relative to the pixel grid) giving rise to the spatial gradient is given by:

$$\theta = \arctan(Gy/Gx) \quad (3)$$

Robert cross operator: The Roberts Cross operator performs a simple, quick computation to compute, 2-D spatial gradient measurement on an image. Pixel values at each point in the output represent the estimated absolute magnitude of the spatial gradient of the input image at that point. The operator consists of a pair of 2×2 convolution kernels as shown in Figure. One kernel is simply the other rotated by 90°.

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

Gx

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

Gy

Figure 2: Convolution masks of Robert cross operator

These kernels are designed to respond maximally to edges running at 45° to the pixel grid, one kernel for each of the two perpendicular orientations. The kernels can be applied separately to the input image, to produce separate measurements of the gradient component in each orientation (call these *Gx* and *Gy*). These can then be combined together to find the absolute magnitude of the gradient at each point and the orientation of that gradient. The gradient magnitude is given by (1). Although typically, an approximate magnitude is computed using (2) which is much faster to compute.

The angle of orientation of the edge giving rise to the spatial gradient (relative to the pixel grid orientation) is given by:

$$\theta = \arctan(Gy/Gx) - 3\pi/4 \quad (4)$$

Prewitt operator: Prewitt operator is similar to the Sobel operator and is used for detecting vertical and horizontal edges in images. The convolution kernel used for this is shown below:

-1	**0**	**1**
-1	**0**	**1**
-1	**0**	**1**

Gx

1	**1**	**1**
0	**0**	**0**
-1	**-1**	**-1**

Gy

Figure 3: Convolution masks of Prewitt operator

Second Order Derivative: [4, 11]

Laplacian of Gaussian: The Laplacian is a 2-D isotropic measure of the 2nd spatial derivative of an image. The Laplacian of an image highlights regions of rapid intensity change and is therefore often used for edge detection. The Laplacian is often applied to an image that has first been smoothed with something approximating a Gaussian Smoothing filter in order to reduce its sensitivity to noise. The operator normally takes a single gray level image as input and produces another gray level image as output. [5, 6, 7]

The Laplacian *L(x,y)* of an image with pixel intensity values *I(x,y)* is given by:

$$L(x,y) = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2} \quad (5)$$

Since the input image is represented as a set of discrete pixels, we have to find a discrete convolution kernel that can approximate the second derivatives in the definition of the Laplacian. [9, 10]

-1	**2**	**-1**
2	**-4**	**2**
-1	**2**	**-1**

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

1	**1**	**1**
1	**-8**	**1**
1	**1**	**1**

Figure 4: Three commonly used discrete approximations to the Laplacian filter.

Canny Operator: The Canny edge detection algorithm is known to many as the optimal edge detector. It is used to detect wide range of edges. In the very first stage noise get removed from the image. So the raw image gets convoluted with the Gaussian filter. A convolution mask is usually much smaller than the actual image. As a result, the mask slid over the image, manipulating a square of pixels at a time. The larger the width of the Gaussian mask, the lower is the detector's sensitivity to noise. The localization error in the detected edges also increases slightly as the Gaussian width is increased.

An edge in an image may point in different directions. The edge detection operators are used which returns a value for first derivative in both horizontal and vertical directions. The edge direction angle is rounded to one of four angles representing vertical, horizontal and two diagonals.

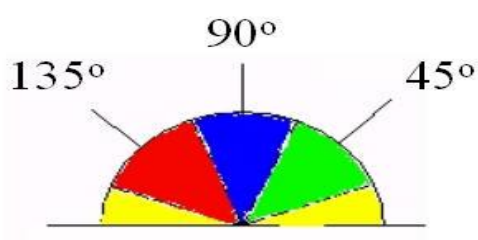

Figure 5: Possible directions for edge detection

Therefore, any edge direction falling within the yellow range (0 to 22.5 & 157.5 to 180 degrees) is set to 0 degrees. Any edge direction falling in the green range (22.5 to 67.5 degrees) is set to 45 degrees. Any edge direction falling in the blue range (67.5 to 112.5 degrees) is set to 90 degrees. And finally, any edge direction falling within the red range (112.5 to 157.5 degrees) is set to 135 degrees.

Then a search is carried out to detect the direction of edges. If the rounded gradient angle is 0, 90 , 135 and 45 degrees the point will be considered to be on edge if its gradient magnitude is greater than the magnitude in its opposite directions respectively. This stage is refereed as non maximum suppression. It is used to trace along the edge in the edge direction and suppress any pixel value (sets it equal to 0) that is not considered to be an edge. This will give a thin line in the output image. A set of edge points in the form of binary image is obtained.

In First Order Derivatives, with a difference in the convolution mask both Robert cross operator and Prewitt operator work as the same way a Sobel operator does. Sobel is the best method to detect edges from first order derivative, but it cannot produce good edge detection with thin and smooth edge. The quality of the edge totally depends on quality of the picture, in other words the raw picture must be totally filtered from noisy pixels.

In Second Order Derivatives, Canny operator is defined as the best operator to detect true edges. Its features are use of hysteresis thresholding rather than using fixed thresholding. It maximizes Signal to Noise Ratio and localization, more robust to noise, avoid multiple edge lines. The localization is not good in LOG, edges are thick and it also retains more noise. [8]

3. Parameters for Comparison

Peak Signal to Noise Ratio (PSNR) is defined as the ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation. The operator with higher PSNR is considered as best for edge detection. [6]

$$PSNR = 10\log_{10}(MAX_i^{\,2} / MSE)$$

Mean square error (MSE) of an estimator is to quantify the difference between an estimator and the true value of the quantity being estimated. [6]

$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I(i,j) - K(i,j)]^2$$

4. Proposed Method

We have proposed an efficient edge detection method using adaptive thresholding approach.

Algorithm-

Step 1: Start
Step 2: Take an image f (i,j) of size 256*256 that is free from noise.
Step 3: Find Gradient magnitude G (i,j).
Step 4: Find direction theta (i,j) in degrees.
Step 5: Apply non maximum suppression on the image.
Step 6: Detect edges by using Adaptive thresholding approach.
Step 7: Set variable i and j to 0.
Step 8: Repeat step 9 to 18 until i is less than equal to 255.
Step 9: Repeat step 10 to 17 until j is less than equal to 255.
Step 10: Check whether f (i,j) is greater than 0 AND less than equal to 50. If yes go to step 13 otherwise go to step 11.
Step 11: Check whether f (i,j) is greater than 50 AND less than equal to 100. If yes go to step 14 otherwise go to step 12.
Step 12: Check whether f (i,j) is greater than 100 AND less than equal to 200 . If yes go to step 14 otherwise go to step 15.
Step 13: Check whether G (i,j) is less than 50 . If yes go to step 16 otherwise go to step 17.
Step 14: Check whether G (i,j) is less than 100 . If yes go to step 16 otherwise go to step 17.
Step 15: Check whether G (i,j) is less than 150 . If yes go to step 16 otherwise go to step 17.
Step 16: Set output pixel value r (i,j) equal to 0 and increase j by 1.
Step 17: Set output pixel value r (i,j) equal to 255 and increase j by 1.
Step 18: Increase value of i by 1.
Step 19: Calculate PSNR and MSE value of the output image.
Step 20: Display the output image on the screen.
Step 21: Stop.

5. Experimental Results and Comparison

Table 1: shows the experimental results of Sobel operator, Canny operator and Proposed method.

Image (a) Image (b) Image (c)

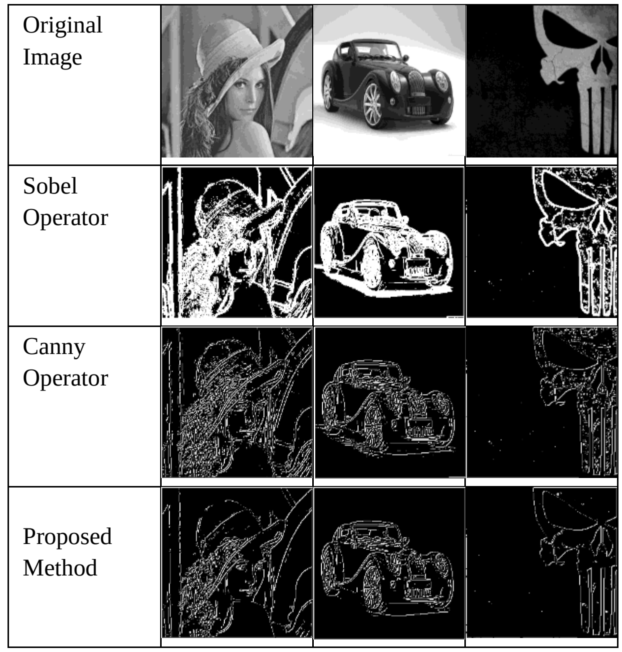

Table 2: shows the comparison of Sobel operator, Canny operator and Proposed method

Image (a)		
Operator	**MSE**	**PSNR**
Sobel	20009.041901	5.118541
Canny	19044.741440	5.333053
Proposed	18883.428726	5.369995
Image (b)		
Operator	**MSE**	**PSNR**
Sobel	36286.560593	2.533346
Canny	35412.056427	2.639292
Proposed	35322.773743	2.650256
Image (c)		
Operator	**MSE**	**PSNR**
Sobel	8070.961700	9.061551
Canny	5944.784027	10.389443
Proposed	5911.663910	10.413706

6. Conclusion

In this paper, a novel approach is proposed using adaptive thresholding for detecting true edges. We also compared our proposed approach with traditional canny operator and sobel operator. As in the previous researches they are considered to be best in second order derivatives and first order derivatives respectively. After comparing them with our proposed algorithm it has been observed that PSNR of proposed technique is higher than other two operators and less in case of MSE. So, it is concluded that our novel algorithm is a better technique to detect true edges in gray scale images. In the future work this technique can be applied on coloured images to detect edges.

References

[1] Rafael c. Gonzalez and richard e. Woods "Digital Image Processing", Third Edition, Pearson Prentice Hall, 2002.
[2] Ireyuwa. E. Igbinosa (2013), "Comparison of Edge Detection Technique in Image Processing Techniques", International Journal of Information Technology and Electrical Engineering, ISSN 2306-708X Volume 2, Issue 1, February 2013.
[3] Pooja Sharma, Gurpreet Singh, Amandeep Kaur (2013), " Different Techniques Of Edge Detection In Digital Image Processing" , International Journal of Engineering Research and Applications (IJERA), ISSN: 2248-9622, Vol. 3, Issue 3, May-Jun 2013.
[4] Ravi S, A M Khan (2012), "Operators Used In Edge Detection Computation: A Case Study", International International Journal of Applied Engineering Research, ISSN 0973-4562 Vol.7 No.11.
[5] Mr. Manoj K.Vairalkar , Prof. S.U.Nimbhorkar, "Edge Detection of Images Using Sobel Operator", International Journal of Emerging Technology and Advanced Engineering , ISSN 2250-2459, Volume 2, Issue 1, January 2012.
[6] Beant Kaur, Anil Garg (2011), "Comparative study of different edge detection techniques", International Journal of Engineering Science and Technology, ISSN 0975-5462 Vol. 3 No. 3.
[7] Mehdi Ghasemi Naraghi, Mahdi koohi, abbas shakery(2011), "Edge detection in multispectral Images based on structural Elements".
[8] Mamta Juneja , Parvinder Singh Sandhu (2009), "Performance Evaluation of Edge Detection Techniques for Images in Spatial Domain", International Journal of Computer Theory and Engineering, Vol. 1, No. 5.
[9] Mohamed Roushdy (2006), "Comparative Study of Edge Detection Algorithms Applying on the Grayscale Noisy Image Using Morphological Filter", GVIP Journal, Volume 6, Issue 4.
[10] Mohsen Sharifi, Mahmoud Fathy, Maryam Tayefeh Mahmoudi (2002), "A Classified and Comparative Study of Edge Detection Algorithms", Proceedings of the International Conference on Information Technology: Coding and Computing.
[11] Harsimran Singh, Tajinder Kaur (2013), "Empirical Study of various Edge Detection Techniques For Gray Scale Images" , International Journal of Advanced Research in Computer Science and Software Engineering (IJARCSSE), ISSN 2277 128X, Volume 3, Issue 8, August 2013.
[12] Amiya Halder, Nilabha Chatterjee, Arindam Kar, Swastik Pal and Soumajit Pramanik(2011), "Edge Detection: A Statistical approach", International Conference on Electronics Computer Technology (ICEC).
[13] Masoud Nosrat, Ronak Karimi, Mehdi Hariri, Kamran Malekian (2013), "Edge Detection Techniques in Processing Digital Images: Investigation of Canny Algorithm and Gabor Method", World Applied Programming, ISSN: 2222-2510, Volume 3, Issue 3, March 2013.

Er. Harsimran Singh received his B.Tech degree from S.S.C.E.T Badhani, Pathankot and currently he is pursuing his M.Tech from S.B.B.S.I.E.T Padhiana, Jalandhar, Punjab, India. He is presently working as Lecturer in Lovely Professional University, Phagwara. His research interests focus on Digital Image Processing and Data Mining.

Er. Tajinder Kaur has received her M.Tech degree from MMU Mullana, Ambala. She is presently working as Assistant Professor in S.B.B.S.I.E.T Padhiana, Jalandhar, Punjab, India. Her research interests focus on Digital Image Processing and Software Engineering.